

DISSENY I DESENVOLUPAMENT DE XARXES NEURONALS ARTIFICIALS

RESUM

Partint de l'interès del qual gaudeixen les xarxes neuronals artificials –generat, en gran part, per l'enorme potencial que sembla tenir l'emulació del cervell–, s'ha plantejat conèixer el màxim de detalls possible sobre la computació neuronal, posant especial atenció en el seu funcionament, per tal de dissenyar i desenvolupar diverses xarxes neuronals artificials i comprovar, d'aquesta manera, si efectivament constitueixen un mètode vàlid per a la resolució de problemes reals.

Després d'analitzar els diferents programes que permeten fer això i de fer una tria del que semblava més adequat –el programa anomenat Matlab– s'ha procedit a desenvolupar dos projectes de xarxes neuronals. El primer permet reconèixer els dígitos pertanyents a l'interval comprès entre el 0 i el 9 (tots dos inclosos); el segon, en canvi, permet reconèixer en teixits cancerosos la tipologia dels tumors –és a dir, si aquests són benignes o malignes–.

A partir d'aquests dos projectes, s'ha realitzat una anàlisi dels resultats amb les eines que proporciona el Matlab i s'ha pogut concloure que, efectivament, les xarxes neuronals artificials són un mètode vàlid –i fins i tot recomanable– per resoldre determinats problemes reals. De fet, mostren un gran potencial i constitueixen una inversió necessària en aquest camp si se'n vol afavorir el desenvolupament.

ÍNDEX:

INTRODUCCIÓ	5
COS TEÒRIC:	
1.- XARXES NEURONALS BIOLÒGIQUES	9
1.1.- EL PAPER DEL SISTEMA NERVIÓS	9
1.2.- LES PARTS DEL SISTEMA NERVIÓS	10
1.3.- LA NEURONA	11
1.3.1- Definició	11
1.3.2- La transmissió de l'impuls nerviós i la sinapsi.....	11
1.4.- COMPARACIÓ ENTRE CERVELL I ORDINADOR	12
1.4.1- Semblances	12
1.4.2- Diferències.....	13
2.- INTEL·LIGÈNCIA ARTIFICIAL	14
2.1.- DEFINICIÓ.....	14
2.2.- EL TEST DE TURING	15
2.3.- LES POSTURES ENVERS LA POSSIBILITAT DE SIMULAR LA CONSCIÈNCIA A TRAVÉS DE LA I.A	17
3.- XARXES NEURONALS ARTIFICIALS	18
3.1.- DEFINICIÓ.....	18
3.2.- FUNCIONAMENT BÀSIC DE LES XNA	18
3.3.- ENTRENAMENT I ARQUITECTURA	19
4.- COMPUTACIÓ TRADICIONAL I COMPUTACIÓ NEURONAL	20
4.1- LA RESOLUCIÓ DE PROBLEMES	20
4.2.- LA MEMÒRIA DE LES XARXES NEURONALS ARTIFICIALS	20
4.3.- ARQUITECTURA TRADICIONAL I ARQUITECTURA NEURONAL	20
4.4.- TAULA COMPARATIVA	21
5.- FUNCIONAMENT DE LES XARXES NEURONALS ARTIFICIALS	22
5.1.- LA NEURONA ARTIFICIAL	22
5.2.- PROCESSAMENT D'INFORMACIÓ: REGLES I FUNCIONS MATEMÀTIQUES	23
5.2.1.- Regla de propagació.....	23
5.2.2.- Funció d'activació	24
5.2.3.- Funció de transferència o de sortida.....	24
5.3.- CLASSIFICACIÓ DE L'ARQUITECTURA.....	25
5.3.1.- Segons el número de capes	25
5.3.2.- Segons el tipus de connexions	26
5.3.3.- Segons el grau de connexió	26
5.4.- ENTRENAMENT	26
5.5.- ALGORITMES D'APRENTATGE.....	27
5.5.1.- Aprenentatge supervisat.....	28
5.5.2.- Aprenentatge no supervisat.....	28
6.- HISTÒRIA DE LES XARXES NEURONALS ARTIFICIALS	29
6.1.- EL PREDECESSOR DE LES XNA	29
6.2.- MCCULLOCH I PITTS	29
6.2.1.- La neurona de McCulloch-Pitts.....	29
6.3.- APRENTATGE HEBBIÀ	30
6.4.- EL PERCEPTRÓ	30
6.5.- ADALINE I MADALINE	31
6.6.- MINSKY I PAPERT: "TOY PROBLEMS"	31
6.7.- A CONTRACORRENT: ANDERSON, KOHONEN I GROSSBERG	31
6.8.- JOHN HOPFIELD	32
6.9.- RUMELHART, HINTON I LA RETROPROPAGACIÓ (BACKPROPAGATION).....	32
7.- ENTREVISTA AMB UN EXPERT	34

8.- AVANTATGES I DESAVANTATGES	37
9.- APLICACIONS	39
10.- L'ART PROVINENT DEL RECOONEIXEMENT D'IMATGES	40
10.1.- ELS "SOMNIS" DE LES XARXES NEURONALS DE GOOGLE.....	40
10.2.- INSTAPAINTING	41
11.- EL PERCEPTRÓ	42
11.1.- EL PERCEPTRÓ SIMPLE	42
11.1.1.- Arquitectura	42
11.1.2.- Funcionament	43
11.1.3.- Algorisme d'aprenentatge del perceptró simple.....	43
11.1.4.- Limitacions del perceptró simple.....	44
11.2.- ADALINE	44
11.2.1.- LMS o regla de mínims quadrats, comparació amb l'algorisme d'aprenentatge del perceptró	44
11.3.- EL PERCEPTRÓ MULTICAPA O MLP (MULTI-LAYER PERCEPTRON)	45
11.3.1.- Funcionament	46
11.3.2.- La regla d'aprenentatge del perceptró multicapa: l'algorisme de retropropagació (Back-Propagation)	46
COS PRÀCTIC:	
12.- ELS DIFERENTS TIPUS DE PROGRAMES	48
13.- MATLAB	48
13.1.- PERQUÈ EL MATLAB?	48
13.2.- QUÈ ÉS EL MATAB?.....	49
13.3.- FUNCIONAMENT BÀSIC	50
13.4.- CAIXA D'EINES DE XARXES NEURONALS (NEURAL NETWORK TOOLBOX)	51
13.5.- GUI	52
13.5.1- Descripció	52
13.5.2- Objectes gràfics en el GUI	53
14.- PROJECTES ASSOLITS	54
14.1.- DIFICULTATS.....	54
14.1.1.- Dificultats generals	54
14.1.2.- Dificultats específiques	55
14.1.2.1.- Reconeixement de caràcters	55
14.1.2.2.- Reconeixement de teixits tumorals	56
14.2.- CARACTERISTIQUES COMUNES ENTRE ELS DOS PROBLEMES	56
14.3.-RECOONEIXEMENT DE DÍGITS.....	57
14.3.1- Descripció.....	57
14.3.2.- Script.....	58
14.3.2.1- Bloc 1	59
14.3.2.2.- Bloc 2.....	60
14.3.2.3.- Bloc 3.....	60
14.3.2.4.- Bloc 4.....	62
14.3.2.5.- Anàlisi dels resultats: Matriu de confusió	64
14.3.2.6.- Interfície gràfica (GUI).....	65
14.4.- RECOONEIXEMENT DE TEIXITS TUMORALS	67
14.4.1- Descripció.....	67
14.4.2.- La base de dades	67
14.4.3.- Explicació del mètode	68
14.4.4.- Anàlisi dels resultats: Matriu de confusió	71
CONCLUSIÓ	73
FONTS DOCUMENTALS CONSULTADES	75
ANNEX I	78
ANNEX II	80

AGRAÏMENTS

M'agradaria agrair, en primer lloc, a tot el conjunt de persones que, sense saber-ho, han fet possible aquest treball ja sigui influint-me a través de vies acadèmiques o a través de vies més emocionals.

Al meu tutor, que ha respost els meus dubtes quan ho he necessitat i ha fet el possible per ajudar-me en un camp en el que els dos no en teníem ni idea, el de la computació neuronal.

A la meva família, que m'ha donat ànims i suport de manera incondicional, fins i tot quan els hi demanava que fessin lletra de doctor.

A la meva germana, que ara no em queda cap dubte que és una mestra de la redacció i que com a tal no s'ha estalviat les crítiques que tant requeria el meu treball.

I finalment, cal dir que aquest treball no estaria a les mans del lector sense el Dr. Jordi Solé, que amb molta paciència i de manera desinteressada –que no és dir poc– ha aconseguit que un ignorant de la programació aconseguís fer alguna cosa útil.

Moltes gràcies a tots, de veritat!

INTRODUCCIÓ

PRESENTACIÓ

A continuació es descriurà, a mode d'introducció, tot el procés de raonament que porta a utilitzar les xarxes neuronals artificials.

L'ésser humà és, per naturalesa, un ésser intel·ligent. Aquesta característica, indiscutiblement, ha estat la gran triomfadora del procés evolutiu, ja que la població humana ha crescut al llarg del temps fins a arribar a la impactant xifra d'uns 7000 milions d'éssers humans. Tanmateix, també tendim a preguntar-nos el perquè de tot el que ens envolta, som curiosos. Així doncs, la combinació entre intel·ligència i curiositat condueix a un aprenentatge continu i a una millor comprensió del món. El cas és que la intel·ligència ens permet abstroure, qüestionar, criticar i raonar. Aquestes habilitats ens capaciten per resoldre problemes, la qual cosa possibilita la supervivència de l'espècie i, fins i tot, permet una millora molt significativa en la qualitat de vida. És, definitivament, el tret diferencial més rellevant de la nostra espècie.

Què passaria, doncs, si una capacitat tan útil com és la intel·ligència fos aplicable a les màquines? Mentre que els que tenen més temor que sentit comú opinen que això significaria el final de la humanitat, experts en el camp de la intel·ligència artificial desacrediten l'afirmació anterior amb arguments com constata que només s'ha de témer un robot si es tem, també, al seu creador, és a dir, qui l'ha dissenyat. I és que les màquines es creen amb l'objectiu de ser controlades per humans, i no al contrari.

La realitat, però, és més inhòspita que la ficció. S'ha aconseguit desenvolupar màquines amb una capacitat de processament impressionant que poden fer càlculs d'elevada complexitat en qüestió de segons, però que, tanmateix, no són capaces de dur a terme la majoria de coses que sí que pot fer un nen de 5 anys, com ara reconèixer cares, paraules, etc. De fet, alguns entesos en el camp opinen que és impossible aconseguir una màquina amb una intel·ligència equivalent a la humana.

Quan es va detectar aquest problema, va sorgir la idea de simular l'òrgan per excel·lència relacionat amb la intel·ligència, el cervell. Això va representar l'inici de les xarxes neuronals artificials, és a dir, els mètodes de computació que es basen en les xarxes neuronals biològiques presents al cervell. Ara bé, com es pot intuir, el cervell no és precisament fàcil d'imitar –i menys encara en un entorn computacional–, de manera que les xarxes neuronals artificials (tot i conservar la seva essència en la biologia) van haver d'adquirir un enfoc més matemàtic; es basen en una idea senzilla: donats uns certs paràmetres, existeix una forma de combinar-los per tal d'obtenir un resultat determinat.

Així doncs, aquest treball consistirà en el procés de disseny i desenvolupament de diverses xarxes neuronals per tal d'observar el potencial que tenen aquestes i per descobrir si es poden aplicar o no en problemes reals. No obstant, per tal d'assolir aquest objectiu, serà necessari fer en primera instància una definició acurada dels conceptes propis de la computació neuronal amb la finalitat que el lector compregui millor la idea de xarxa neuronal artificial.

MOTIVACIONS

És curiós saber els motius que han portat a l'autor a fer el treball que ha fet. Aquests són els meus:

El simple mot "xarxes neuronals artificials" ja desperta tota la meua atenció. Combina dos dels àmbits que més em fascinen des que era petit; la biologia i la tecnologia. Mentre que el primer intenta desentrellar els misteris de la vida i busca possibles explicacions sobre aquests, el segon es basa en la comprensió del món que ens envolta per tal de crear aparells o objectes amb una finalitat pràctica.

Per part biològica, el cervell i la intel·ligència en general són temes que, personalment, trobo molt interessants. Això és degut al fet que, malgrat les nombroses investigacions dutes a terme al llarg dels anys, són elements d'extrema complexitat i en l'actualitat s'està considerablement lluny de conèixer i entendre amb precisió com funcionen. Sabent això, triar un treball que em permetés aprofundir més en aquesta matèria em semblava una molt bona opció. Addicionalment, quan vaig saber que a través de les xarxes neuronals artificials resulta possible determinar si un tumor és de tipus benigne o maligne, vaig pensar que aquesta seria una aplicació molt pràctica i em va fer força il·lusió desenvolupar una xarxa capaç de resoldre aquest tipus de problemes.

Per part tecnològica, la visualització tant de la pel·lícula "The imitation game" (2014, dirigida per Morten Tyldum) com de la sèrie "Real humans" (2012, dirigida per Stefan Baron), va fer néixer en mi una creixent curiositat sobre la intel·ligència artificial. La pel·lícula –que, de fet, està basada en fets reals– mostra com una màquina ideada per Alan Turing és capaç de descriptar el codi utilitzat pels nazis, mentre que la sèrie planteja tot un seguit de qüestions ètiques com ara la següent: Haurien de rebre el mateix tracte humans i robots, si aquests gaudissin de consciència?

Així doncs, de tots els treballs que tenia en ment, aquest és el que em va convèncer més a l'hora de fer una decisió, ja que les altres opcions només contenien o bé l'àmbit de la biologia o bé l'àmbit de la tecnologia. Definitivament, no era el que semblava més fàcil, però vaig pensar en la gran quantitat de temps que requereix el treball de recerca i vaig concloure que el treball havia de ser, per força, el que més ganes tingués de fer.

La veritat és, tanmateix, que va ser una decisió força optimista tenint en compte que per dur a terme aquest treball són necessàries unes mínimes nocions de programació per tal de poder confeccionar amb èxit la xarxa neuronal. Unes nocions que, desgraciadament, jo no tenia. Vaig ser una mica agosarat, sí, però amb l'ajuda adequada he aconseguit uns resultats força acceptables i, de fet, no em penedeixo en absolut d'haver-me arriscat.

OBJECTIUS

Com en qualsevol treball amb un cert rigor, cal plantejar-se unes preguntes o fer unes hipòtesis per tal de determinar uns certs objectius. Aquest n'és el resultat:

L'objectiu principal del treball és el de comprovar la validesa de la computació neuronal en diversos problemes reals, de manera que, un cop aquesta sigui aplicada al problema, tingui la capacitat de trobar la solució adequada. De l'objectiu principal, però, se'n deriven objectius secundaris. Aquests són els següents: (1) entrevistar un expert en el camp de la computació neuronal per tal de preguntar-li la seva opinió tant de qüestions ètiques lligades a la intel·ligència artificial com de qüestions teòriques i pràctiques relacionades amb les xarxes neuronals artificials; (2) dissenyar i desenvolupar, mitjançant l'ajuda d'un programa, una xarxa neuronal que classifiqui diferents tipus de teixits tumorals. Aquesta classificació ha de servir per determinar si els tumors són de tipus benigne o maligne, és a dir, si són o no potencialment perjudicials per a l'organisme; i (3) dissenyar i desenvolupar, si és possible i viable, i mitjançant l'ajuda d'un programa, diferents xarxes neuronals que siguin capaces de resoldre problemes que, per si mateixos, no presenten dificultats als éssers humans. En són exemples el reconeixement de caràcters, la classificació de banderes dels diferents països, el reconeixement facial, el reconeixement de veu... La finalitat de resoldre aquests problemes, doncs, no resideix únicament en trobar-ne la solució (ja que no plantegen cap dificultat per a l'ésser humà), sinó la digitalització de la solució. Així doncs, per exemple, aquest procés tindria utilitat en el cas que fos d'interès el disseny d'una pissarra digital capaç d'incorporar la funció de digitalitzar qualsevol caràcter que s'hi escriu (com es pot observar a la Fig. 1.).



Fig. 1. La lletra de doctor escrita a mà passa a ser una lletra més clara gràcies al procés de digitalització.

METODOLOGIA

Per tal de realitzar el cos teòric del treball, s'ha consultat, majoritàriament, enllaços d'internet. Tot i així, en el tema corresponent a les xarxes neuronals biològiques s'ha obtingut la informació principalment de llibres. Cal remarcar que, buscant evitar l'obtenció de continguts erronis, s'ha procedit a consultar més d'un enllaç per a cada tema per comparar si la informació oferta era semblant.

Pel que fa referència al cos pràctic, s'ha recorregut, en gran part, al programa Matlab. Aquest permet desenvolupar, amb un ampli repertori de possibilitats, una xarxa neuronal artificial. El Dr. Jordi Solé, un entès tant en el Matlab com en les xarxes, m'ha ajudat en l'apartat de programació, indicant-me com fer molts dels processos més complexos.

ESTRUCTURA DE LA MEMÒRIA

El cos de la memòria es divideix en dues grans parts, el cos teòric i el cos pràctic. El cos teòric està enfocat a explicar els diferents conceptes relacionats amb les xarxes neuronals artificials, mentre que el cos pràctic està marcadament lligat al disseny i desenvolupament de xarxes neuronals artificials.

Cos teòric:

Així doncs, per definir el concepte de xarxa neuronal artificial, es fa necessari definir, també, els conceptes de xarxa neuronal biològica i intel·ligència artificial. Un cop explicat això, es pot procedir a fer una comparació entre la computació tradicional i la computació neuronal, tot establint-ne els trets diferencials de cadascuna. Acte seguit, és necessari aprofundir en el funcionament de les xarxes i, a continuació, pot ser interessant descriure la història i evolució d'aquestes.

Per acabar, resulta imprescindible exposar els avantatges i desavantatges que presenta la computació neuronal, determinar en quins camps s'aplica i, fins i tot, remarcar que quan les xarxes s'apliquen al reconeixement d'imatges d'una certa manera, poden produir obres de caire artístic. També cal aprofundir en el concepte de perceptró, ja que serà el tipus de xarxa neuronal que s'utilitzarà en els projectes descrits al cos pràctic.

Cos pràctic:

Primerament, resulta necessari explicar què són els programes que permeten dissenyar xarxes neuronals artificials. Un cop fet aquest aclariment, es procedeix a triar un d'aquests programes i a explicar-ne el perquè de la decisió. A continuació es descriu detalladament el programa triat (Matlab), tot fent-ne una descripció i explicant-ne les funcions.

Seguidament, ja es pot començar a descriure els dos projectes assolits, és a dir, la xarxa que permet reconèixer teixits tumorals i la xarxa que permet reconèixer dígitos. És interessant iniciar aquest procés analitzant les dificultats que hi ha hagut en la programació en general i en cada projecte específicament. Després es continua explicant cada projecte detalladament i els passos que s'han hagut de fer per dur-los a terme i, per últim, és important fer un anàlisi dels resultats.

COS TEÒRIC

1.- XARXES NEURONALS BIOLÒGIQUES

Per entendre el concepte de xarxa neuronal artificial, com és lògic, cal recórrer al concepte de xarxa neuronal biològica, ja que és el model en què es basen. És per aquest motiu que cal fer una descripció rigorosa de les xarxes neuronals biològiques i del sistema al que pertanyen, el nerviós.

1.1.- EL PAPER DEL SISTEMA NERVIÓS

El sistema nerviós desenvolupa un rol fonamental en l'organisme dels animals que el posseeixen, degut a la seva rellevant funció de governar i coordinar els diferents processos biològics que es puguin donar. Per una banda, regula la vida vegetativa –és a dir, les funcions vitals com ara la respiració, la circulació o la digestió– i per l'altra, regula la vida de relació, que consisteix en tots aquells processos que involucren l'organisme amb el món exterior, ja sigui de manera concreta (percepció dels estímuls de moviment, per exemple) o bé de forma abstracta (mecanismes elaborats com el pensament, la consciència, l'afecte o la memòria).

L'organització d'aquest sistema es classifica en dues seccions:

- **Secció aferent:** S'ocupa de les funcions sensorials, és a dir, recull totes les informacions possibles sobre els medis exterior i interior de l'organisme. I és precisament a partir d'aquestes informacions –o, més ben dit, d'aquests senyals– que el sistema nerviós executa aquelles reaccions immediates que són necessàries, com el reconeixement del propi organisme i el del món que l'envolta. De fet, sense la possessió de les funcions sensorials, el sistema nerviós és incapaç de realitzar cap mena de procés, des dels reflexos més simples fins a les manifestacions més elaborades de la consciència.
- **Secció eferent:** La seva missió és transmetre i executar les ordres necessàries que permeten la perpetració de diverses funcions en l'organisme. Complementàriament, es poden distingir dos subsistemes que de forma conjunta constitueixen aquesta secció i presenten activitats ben diferenciades. Aquests són el sistema motor i el sistema vegetatiu.
 - El sistema motor, com el seu nom indica, s'ocupa de les funcions motores o de moviment, les quals s'executen mitjançant el control de la contracció dels músculs esquelètics.
 - El sistema vegetatiu, per la seva banda, controla les funcions vegetatives, és a dir, les activitats adreçades a l'interior de l'organisme.

1.2.- LES PARTS DEL SISTEMA NERVIÓS

Per tal de definir les diferents parts del sistema nerviós es sol recórrer a una classificació específica:

Sistema Nerviós Central (SNC): format per l'encèfal i la medul·la espinal, es troba protegit per tres membranes anomenades meninges. També, en el seu interior hi ha un seguit de cavitats (ventricles) pels quals hi circula el líquid cefaloraquídi.

Sistema Nerviós Perifèric (SNP): format per nervis –tant cranials com espinals– que emergeixen del sistema nerviós central i recorren tot el cos.

Mentre que les parts del Sistema Nerviós Perifèric no es diferencien –ja que aquest sistema es compon únicament per nervis– les parts pròpies del Sistema Nerviós Central tenen unes funcions molt específiques i són clarament independents les unes de les altres. Per aquest motiu, és necessari fer-ne una breu descripció (a la Fig. 2. Es poden observar les parts descrites):

- **L'encèfal:** és la part del SNC protegida per els ossos del crani. Està format pel cervell, el cerebel i el tronc encefàlic.
 - **Cervell:** és la part més voluminosa amb diferència. Està dividit en dos hemisferis, dret i esquerra, separats per la cesura interhemisfèrica. La superfície rep el nom d'escorça cerebral i està formada per replegaments. La funció d'aquest òrgan és la de controlar els diferents processos que es donen en l'organisme –un d'aquests processos és, per exemple, la secreció de substàncies químiques-. En el cas dels humans, és l'origen, també, de tots els processos de raonament.
 - **Cerebel:** És una estructura que es situa sota els dos hemisferis cerebrals. La seva funció és la de dirigir l'activitat motora de l'individu, controlant així tant els moviments musculars generals -per exemple, caminar- com els moviments musculars específics -una instància en seria l'acció de posar una clau dins el pany-.
 - **Bulb raquídi:** Presenta una forma de con amb el vèrtex a l'inferior i fa tres centímetres de longitud aproximadament. Les seves funcions inclouen la transmissió d'impulsos de la medul·la espinal a l'encèfal. També regula funcions cardíaques, respiratòries i gastrointestinals.
- **La medul·la espinal:** és una prolongació de l'encèfal, semblant a un cordó que s'estén per l'interior de la columna vertebral. S'encarrega de retransmetre la informació provinent del cervell a altres parts del cos per tal de que aquestes reaccionin i produeixin una resposta. També coordina els reflexes, que són respostes ràpides a estímuls externs que no passen pel cervell.

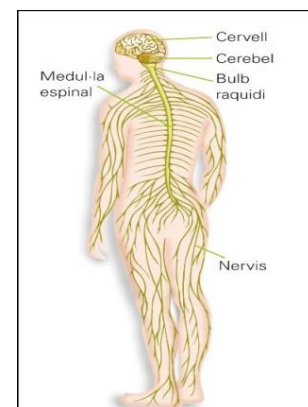


Fig. 2
Representació de
les parts del
sistema nerviós

1.3.- LA NEURONA

1.3.1.- Definició

El sistema nerviós és constituït per cèl·lules altament especialitzades, és a dir, cèl·lules que es caracteritzen per desenvolupar una funció molt específica. Aquestes són les neurones, i són capaces de rebre estímuls i transmetre'ls a distància. Tot i que la comunitat científica és conscient de la importància d'aquestes cèl·lules, motiu que ha portat a l'obertura de noves línies d'investigació en aquest camp, encara no es disposa d'un coneixement complet i detallat sobre qüestions tan discutides com els mecanismes fisiològics i les bases estructurals de la memòria o de la intel·ligència.

L'objectiu últim i fonamental de les neurones és la transmissió a distància de la informació (com es pot observar a la Fig. 3.). Per tal de dur a terme aquesta funció, el cos cel·lular¹ d'aquestes cèl·lules presenta dos tipus de prolongacions:

- **Les dendrites:** En general són ramificacions no gaire llargues i força nombroses que tenen la missió de rebre els senyals que arriben des d'altres neurones per després portar-los conduir-los cap al cos cel·lular de la neurona, al qual estan connectades.
- **L'axó o neurita:** Transporta els senyals des del cos cel·lular cap a l'exterior de la neurona mitjançant les dendrites. Hi ha axons que estan embolcallats per una substància fosfolípídica² anomenada mielina. Aquesta té la funció d'aïllar l'axó i permetre que la transmissió de l'impuls nerviós sigui més ràpida.

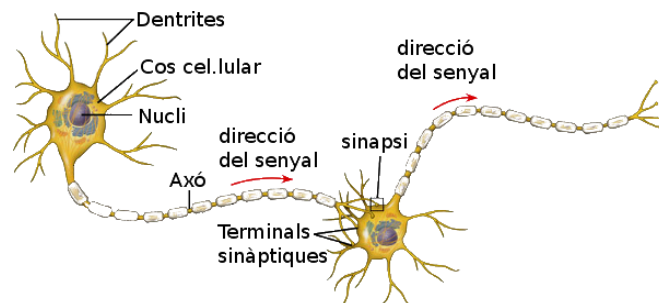


Fig. 3. Representació de neurones i les seves parts corresponents transmetent senyals (d'esquerra a dreta).

1.3.2.- La transmissió de l'impuls nerviós i la sinapsi

La transmissió de l'estímul nerviós presenta una fase elèctrica i una fase química. La primera té lloc a l'interior de la neurona, mentre que la segona es dona entre dues cèl·lules properes, de manera que enllaça l'extrem d'un axó (sortida d'una neurona) i les dendrites (entrades d'una neurona) d'una altra cèl·lula nerviosa.

¹ Regió de la neurona que conté el nucli cel·lular i posseeix una mida considerablement gran en relació al conjunt de la unitat.

² Els fosfolípids són un tipus de biomolècules que com el seu nom indica estan formades per fosfats i per lípids. La funció més rellevant pròpia d'aquesta substància és la de ser un component estructural de la membrana cel·lular.

La fase elèctrica consisteix en la conducció de l'estímul mitjançant canvis de posició entre ions de sodi (Na^+) i ions de potassi (K^+). Aquest impuls es propaga d'un extrem a l'altre de l'axó a una velocitat força elevada que, de fet, pot assolir els 180 km/h. Un cop arriba a l'extrem de l'axó o, el que és el mateix, a la terminació nerviosa, l'impuls elèctric s'atura perquè les membranes de les cèl·lules nervioses mai arriben a establir contacte físic. El conjunt format per aquesta lleugera separació entre les dues membranes s'anomena sinapsi (Fig. 4.).

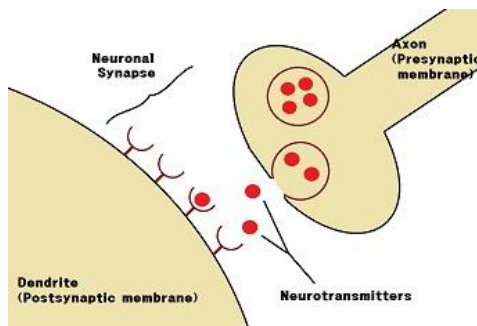


Fig. 4 Neurons transmetent impulsos. Estan separades per la sinapsi.

En la sinapsi s'alliberen, dins de vesícules, tot un conjunt de substàncies químiques (com per exemple la glicina o la serotonina) que reben el nom de neurotransmissors, ja que s'encarreguen de transmetre l'impuls nerviós. Aquests agents químics produeixen un efecte sobre la membrana que pot ser variable; en alguns casos pot causar l'aparició d'un impuls (estímul d'excitació), mentre que en d'altres pot dificultar o impedir el naixement de nous impulsos (estímul d'inhibició).

Tenint en compte la nombrosa quantitat de dendrites que pot posseir cada cèl·lula nerviosa, s'intueix la possibilitat que a una sola neurona hi puguin arribar al mateix temps, procedents d'axons diferents, tant estímuls d'excitació com estímuls d'inhibició. Quan aquest fet es dona, s'estableix una competició entre el conjunt d'estímuls excitadors i el d'inhibidors, que acabarà determinant la transmissió o la no transmissió de nous impulsos. Si els estímuls excitadors superen en nombre els estímuls inhibidors, la neurona s'activa i transmet l'impuls nerviós. En el cas contrari, l'impuls no es transmet degut a l'efecte causat per la inhibició neta dels estímuls.

1.4.- COMPARACIÓ ENTRE CERVELL I ORDINADOR

Sovint, el cervell és comparat amb un ordinador. Resulta interessant, doncs, dur a terme una anàlisi que contempli tant les semblances com les diferències existents entre aquests dos elements, per tal d'establir el grau de validesa d'aquesta comparació.

1.4.1.- Semblances:

Principalment, el cervell es compara amb un computador perquè les unitats perifèriques (corresponents als òrgans sensorials) aporten una gran quantitat d'informació a través dels cables de transmissió (nervis) per tal que una unitat de

processament central (cervell) –que s’encarrega d’emmagatzemar dades (memòria)-l’analitzi i l’executi si és pertinent.

A més, el cervell posseeix dos tipus de vies perifèriques, vies d’entrada i vies de sortida. Dit en altres paraules, necessita una via capaç de rebre les diferents informacions que li arriben a través dels òrgans sensorials (via d’entrada) i d’una altra capaç d’elaborar una resposta en funció de la informació rebuda (via de sortida). De la mateixa manera, un ordinador necessita vies d’entrada (com un teclat o un ratolí) que proporcionin al sistema l’entrada d’informació, la qual apareix acte seguit en una via de sortida (com una pantalla o una impressora) en forma de resposta.



Fig. 5. Perifèrics d’entrada i de sortida propis dels ordinadors

1.4.2.- Diferències:

Per la seva banda, els centres nerviosos tenen més autonomia i més creativitat que qualsevol ordinador actual. Així doncs, mentre que el funcionament d’un ordinador depèn d’una sèrie d’instruccions predeterminades (software), el cervell té tant la capacitat d’aprendre com la de modificar i adaptar el resultat d’aquest aprenentatge.

L’ordinador, en canvi, depèn totalment de les vies d’entrada, ja que el sistema necessita l’ingrés d’informació per generar una resposta –per exemple, s’ha de moure el ratolí (estímul) per tal que el cursor es mogui també (resposta) –. En contrast, al cervell humà no tots els estímuls generen necessàriament una resposta. En moltes situacions, el poder de decisió dels centres nerviosos pot expressar-se mitjançant la voluntat de respondre o no a un estímul i/o adequant el tipus de resposta al context que envolta l’organisme. I no només això; el cervell també pot enviar una ordre sense tenir la necessitat de rebre cap estimulació prèvia procedent de l’exterior.

La majoria de professionals opinen, doncs, que tot i que és curiós fer aquesta comparació en realitat no és tan vàlida, ja que la naturalesa d’un ordinador discrepa en gran mesura amb la del cervell.

2.- INTEL·LIGÈNCIA ARTIFICIAL

2.1.- DEFINICIÓ

La Intel·ligència Artificial (I.A.) és aquella branca de les ciències de la computació que estudia els software i hardware necessaris per poder simular el comportament i la comprensió propis de l'ésser humà. L'objectiu final de la I.A. és, doncs, simular la intel·ligència humana en un sistema o màquina de manera que gaudeixi de consciència i sentiments reals (si no iguals que els dels humans, similars).

El primer pas per conèixer el concepte d'intel·ligència artificial (I.A.) és entendre què significa exactament "intel·ligència". Ara bé, sovint la simple definició d'aquesta paraula no queda exempta de problemes i l'existència de discrepàncies entre pensadors experts en aquest àmbit resulta inevitable, com és el cas de John McCarthy (1927-2011) i Marvin Minsky (1927-1991), que sostenen opinions força diferenciades:

- John McCarthy (creador del terme "intel·ligència artificial"): *"la intel·ligència és la capacitat que té l'ésser humà d'adaptar-se eficaçment al canvi de circumstàncies mitjançant l'ús d'informació sobre aquests canvis"*
- Marvin Minsky (cofundador del laboratori de I.A. del M.I.T.): *"La intel·ligència de la ment humana és el fet d'accionar un conjunt de ments de menor poder que combinen les seves respectives habilitats amb l'objectiu de resoldre problemes"*



Fig. 6. John McCarthy



Fig. 7. Marvin Minsky

La capacitat que posseeix el cervell humà de pensar per si mateix i resoldre de la forma més efectiva possible tant els problemes més simples com els més complexos i embrollats ha generat, sense cap mena de dubte, un gran interès entre filòsofs i científics. Aquests han intentat entendre'n el funcionament i ho han aconseguit, en part. Ara bé, la majoria d'experts opinen que encara s'està molt lluny de comprendre en la seva totalitat aquest, el més misteriós dels òrgans: el cervell.

L'interès creixent per desentranyar les incògnites que encara avui rodegen el concepte de la intel·ligència ha inquietat l'ésser humà fins al punt que l'ha portat a simular artificialment el comportament del cervell. Són aquestes, ni més ni menys, les albers del naixement de la intel·ligència artificial.

Cal assenyalar que el primer contacte amb la I.A es fa a través de les xarxes neuronals artificials. El 1943, dos científics interessats en el cervell humà (McCulloch i Pitts) van proposar un model de neurona artificial pel cervell animal (que es tracta en més profunditat a l'apartat 6.2 d'aquest treball). Aquest model, més aviat abstracte, va proporcionar una representació simbòlica de l'activitat cerebral. McCulloch, molt partidari d'aquesta innovadora ciència, va proposar al seu company Pitts "*el cervell és un solucionador intel·ligent de problemes, per què no imitar-lo?*". I és que imitar el que té èxit a la natura no és un mal guió a seguir si es té en compte que, de fet, un dels vehicles més útils inventats per la humanitat es basa en l'aerodinàmica de les aus.

Desgraciadament, però, l'enorme grau de complexitat que caracteritza el cervell no pot menystenir-se. Sens dubte, alça una barrera que fa que sigui pràcticament impossible simular-lo, almenys detalladament. No cal ni mencionar, a més, que els hardware i software encara poc desenvolupats propis dels anys 40 allunyaven encara més els dos pioners d'un objectiu com el d'aquestes magnituds.

2.2.- EL TEST DE TURING

Si no és possible imitar de manera fiable el cervell, doncs, en quin moment resultaria prudent considerar que una màquina és realment "intel·ligent"? Tot i que la controvèrsia filosòfica està servida, Alan Turing va proposar la seva visió particular.

Alan Turing va ser un matemàtic britànic de molt renom, conegut principalment per ser considerat un dels pares de la computació i haver desenvolupat un paper crucial en la Segona Guerra Mundial en esbrinar el codi utilitzat pels alemanys en la encriptació dels seus missatges. Els historiadors afirmen, de fet, que va aconseguir avançar el final de la guerra uns dos anys aproximadament.

El brillant matemàtic va proposar el 1950 el que en l'actualitat es coneix com "el test de Turing", amb l'objectiu de proporcionar un mecanisme efectiu capaç de discernir entre una màquina intel·ligent i una que no ho és. Turing, però, va ser exageradament optimista en predir que tardarien uns 50 anys -és a dir, a l'any 2000- a desenvolupar una màquina capaç de superar la prova. La descripció detallada d'aquest test està inclosa en l'article publicat per Turing "Computing machinery and intelligence"³ a la revista filosòfica "Mind", i consisteix, fonamentalment, en el següent:

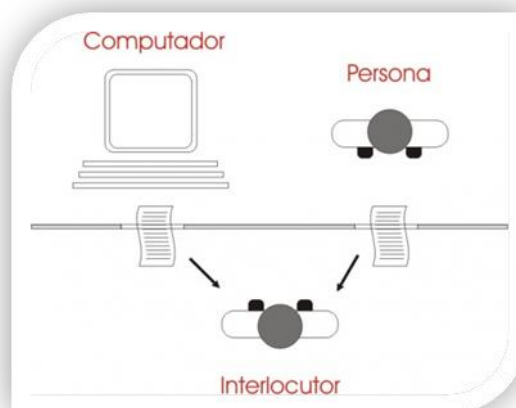


Fig. 8. Test de Turing

Un humà, l'avaluador, fa preguntes a dos subjectes sense poder accedir-hi directament (Fig. 8.). El primer d'aquests subjectes també és de naturalesa humana, però el segon és una computadora dissenyada per generar respostes anàlogues a les

³ En català "Computadores i intel·ligència".

dels humans. Si després d'analitzar la conversa, l'avaluador no pot distingir l'humà de la màquina o considera que la màquina és un humà, aquesta superaria el test de Turing i seria considerada intel·ligent.

No obstant, el test ha estat força qüestionat i com a conseqüència, n'han sorgit crítiques que en desacrediten el mètode. Les dues contraposicions principals són les que segueixen:

- La primera crítica és molt simple, i de caire pràctic; el mètode no és aplicable. Actualment el test de Turing no es pot utilitzar en el camp de la intel·ligència artificial, ja que aquesta àrea de recerca està molt lluny de produir sistemes que puguin actuar com un humà. Com s'ha mencionat anteriorment, és un assumpte d'una complexitat desmesurada i no falten opinions que n'afirmen l'impossibilitat.
- Des del punt de vista filosòfic, John Searle –professor de filosofia nascut el 1931 especialitzat en la relació entre el llenguatge i la ment– va plantejar “L’habitació xinesa” com a resposta al test de Turing. Es tracta d’una prova semblant a la de Turing que va ser popularitzada pel físic i matemàtic Roger Penrose (nascut el 2005). Aquesta té com a objectiu argumentar el perquè cap màquina pot superar el test de Turing, o el que és el mateix, ser intel·ligent.

Essencialment és igual en la forma, però es duu a terme tancant dues persones en una mateixa habitació amb la condició que cap d’elles conegui l’idioma en el què es realitza la conversa.

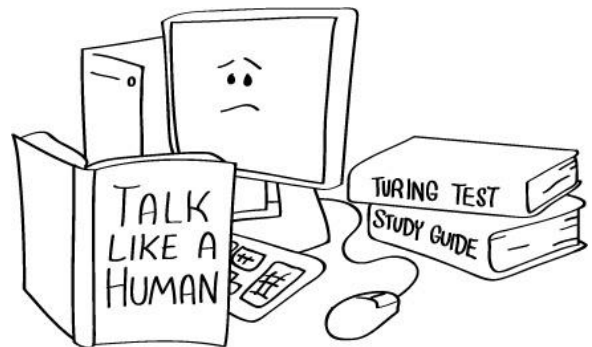


Fig. 9. Sàtira del Test de Turing

Es posa per cas que l’idioma desconegut pels subjectes sigui el xinès. Doncs bé, resulta que als dos participants se’ls permet fer ús de diccionaris. Com es pot deduir, amb una mica de temps i paciència, la conversa transcourrà sense problemes, ja que amb l’ajuda del diccionari, els subjectes traduiran les paraules que vulguin expressar al xinès amb l’objectiu d’emetre informació i faran el procés invers –traduir del xinès a l’idioma que els sigui propi- per tal de rebre i entendre informació.

Aquesta prova demostra, per tant, que per molt que una persona sigui capaç d’enviar una cadena de símbols en xinès en relació a una altra cadena de símbols escrits també en xinès, no vol dir que domini aquest idioma, sinó que sap aplicar un conjunt de regles que li indiquen el què ha de fer. Així doncs, en el cas del test de Turing, el programador és el que adoptaria el paper de diccionari entre màquina i humà.

Searle afirma, tot arribant a una conclusió, que no s’ha de confondre una màquina pensant amb una que sembla que pensa.

2.3.- LES POSTURES ENVERS LA POSSIBILITAT DE SIMULAR LA CONSCIÈNCIA A TRAVÉS DE LA I.A.

No és d'estranyar que un tema generador de tanta polèmica com ho és la I.A. doni lloc a diferents maneres de concebre'l. Davant la possibilitat de simular la consciència a través d'un agent no humà, doncs, es defineixen les següents postures:

1. **I.A. Forta:** Els partidaris d'aquesta postura pensen que tota l'activitat mental de l'ésser humà és de tipus computacional, sentiments i consciència inclosos. Per tant, és possible l'obtenció d'aquests elements emprant el mètode de la computació.
2. **I.A. Dèbil:** Es tracta de la concepció de la consciència com a una característica pròpia del cervell; mentre que tota propietat física es pot simular computacionalment, la consciència s'escapa d'aquests paràmetres, sent així inimitable amb l'ús d'un mètode d'aquest tipus. John Searle n'era un dels partidaris.
3. **Física nova:** Aquesta postura defensa que es necessita una nova física per explicar la ment humana i que potser en un futur aquesta es podrà simular, tot i que no a través de mètodes computacionals. En altres paraules, és necessari que en un futur es descobreixin nous mètodes científics que actualment són desconeguts per assolir una simulació precisa/adequada/que faci justícia a la intel·ligència humana.
4. **Mística:** Sosté que la consciència no es pot explicar ni físicament ni computacionalment, ni per qualsevol altre mitjà de caire científic. És un element que pertany al món espiritual i no pot ser estudiada a través de la raó, ja que s'escapa del coneixement racional.

Els que es consideren partidaris, respectivament, de les dues primeres postures estan convençuts que la consciència és un procés físic que sorgeix del cervell i que, per tant, és explicable a través de la ciència actual. La principal diferència entre les dues postures rau en el fet que, mentre que la I.A. forta opina que la consciència es pot escriure en llenguatge informàtic –és a dir, que és computable–, la I.A. dèbil afirma el contrari.

Procedint en la tercera postura, aquesta es basa fonamentalment en l'ombra de la ignorància actual que plana sobre aquest tema. Diposita tota esperança en la convicció que en un futur es doni un enfoc totalment diferent en la qüestió de la simulació de la intel·ligència, que permeti fer un pas endavant i definitiu cap a la simulació de la consciència.

La quarta postura, sens dubte, és la que més s'apropa al teisme. Això es deu al fet que és perfectament coherent amb el dualisme entre matèria i esperit. Ara bé, cal remarcar que el teisme és compatible, també, amb qualsevol de les quatre postures; la ciència podrà o no respondre la pregunta que apel·la al funcionament de la ment, però possiblement mai donarà resposta a la pregunta que qüestiona l'existència d'aquesta o la seva finalitat.

3.- XARXES NEURONALS ARTIFICIALS

3.1.- DEFINICIÓ

Les xarxes neuronals artificials (XNA) –també anomenades, per simple comoditat, xarxes neuronals– s’inspiren en les xarxes neuronals biològiques del cervell humà. Estan constituïdes per elements que es comporten de forma similar a la neurona biològica en les seves funcions més bàsiques i que, en el seu conjunt, també presenten una organització semblant a la del cervell humà.

Les XNA, a banda d’imitar el cervell en relació a la seva estructura neuronal, també posseeixen característiques pròpies d’aquest òrgan. Per exemple, aprenen⁴ a través de l’experiència, generalitzen⁵ d’exemples previs a exemples nous i abstreuen⁶ les característiques principals d’una sèrie de dades.

3.2.- FUNCIONAMENT BÀSIC DE LES XNA

En aquest tipus de xarxes, la unitat anàloga a la neurona biològica és l’anomenada neurona artificial o element processador, que té diverses entrades i una sola sortida, fet que permet la transmissió i el processament de la informació. El seu mecanisme de processament és el següent:

Les entrades són normalment combinades amb una suma bàsica. A la suma d’aquestes entrades se li aplica una funció, de manera que s’obté un valor que és rebut per la sortida de l’element processador. És a dir, les entrades de la neurona artificial tenen una determinada informació, aquesta és processada a través d’una funció i la sortida de la neurona és la que rep la informació ja processada.

Com no és d’estranyar, la sortida de la neurona artificial es pot connectar a entrades d’altres neurones per formar una xarxa de neurones interconnectades, elevant, així, el grau de complexitat del sistema.

⁴ Aprendre: Adquirir coneixement d’una cosa per mitjà d’estudi, exercici o experiència. Les XNA poden canviar el seu comportament en funció de l’entorn. Se’ls mostra un conjunt d’entrades i elles mateixes s’ajusten per produir unes sortides consistents.

⁵ Generalitzar: Expandir o ampliar una cosa. Les XNA generalitzen automàticament degut a la seva pròpia estructura i naturalesa. Aquestes xarxes poden oferir, dins d’un marge, respostes correctes a entrades que presenten petites variacions degut a efectes distorsionadors, com per exemple ho seria el soroll en una xarxa especialitzada en el reconeixement de veu.

⁶ Abstreure: Aïllar mentalment o separar les qualitats d’un objecte. Algunes XNA són capaces d’abstreure l’essència d’un conjunt d’entrades que aparentment no presenten aspectes comuns o relatius.

Quan una xarxa neuronal opera (fig.10.), el que fa és processar el conjunt de dades proporcionades per les múltiples neurones que formen el sistema, i el resultat d'aquest processament és un valor, el qual rep el nom de valor de sortida. Aquest valor de sortida, doncs, dependrà de les informacions o valors de les respectives entrades, del tipus de neurona, de l'arquitectura de la xarxa i de l'entrenament que hagi rebut aquesta. S'aprofundirà més en el funcionament en l'apartat 5.

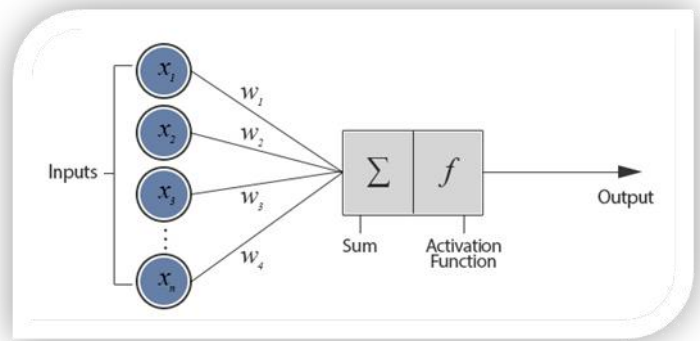


Fig. 10. Esquema sobre el funcionament d'una xarxa neuronal.

3.3.- ENTRENAMENT I ARQUITECTURA

Les xarxes són dispositius entrenables, cosa que significa que a partir d'exemples i seguint uns algorismes⁷ determinats, són capaces de fer canvis i d'adaptar la informació per tal de donar el resultat esperat. De fet, tota la informació que permet a les xarxes neuronals artificials resoldre problemes de forma correcta és obtinguda a través d'exemples, de l'experiència i de l'abstracció de conceptes. Per aquesta raó, es tracta d'un tipus de computació molt adaptativa.

L'arquitectura de la xarxa –o, el que és el mateix, la forma en què les neurones es connecten entre elles– desenvolupa un paper essencial en la mateixa xarxa, ja que el tipus de problemes que es puguin resoldre, l'eficàcia i les limitacions del sistema depenen directament de l'arquitectura.

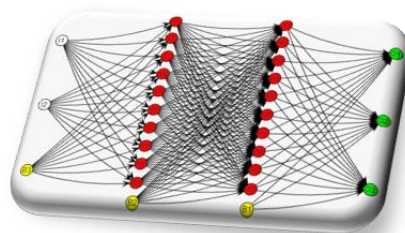


Fig. 11. Arquitectura d'una xarxa complexa.

4.- COMPUTACIÓ TRADICIONAL I COMPUTACIÓ NEURONAL

⁷ Algorisme: seqüència d'instruccions que indica a un sistema com operar per tal d'aconseguir una determinada finalitat, sent la més habitual la resolució de problemes.

És interessant comparar la computació tradicional i la computació neuronal per així analitzar-ne els seus trets diferencials i deduir en quins problemes serà més útil fer l'ús d'una o de l'altra.

4.1.- LA RESOLUCIÓ DE PROBLEMES

Les tècniques tradicionals de programació utilitzades de forma freqüent en l'actualitat requereixen la creació d'un algoritme o seqüència d'instruccions.

El disseny d'aquesta seqüència d'instruccions és relativament senzill i constitueix el primer pas cap a la resolució d'un problema, posem per cas, de comptabilitat. Ara bé, en el món real hi ha una gran quantitat de problemes que no es poden resoldre fent un simple algoritme. Un d'aquests problemes és, per exemple, el reconeixement facial d'una persona; existeix un enorme ventall de variacions en la seva imatge: un dia pot somriure, l'altre pot estar seriosa, haver-se deixat barba o aplicat maquillatge... Tots aquests elements són obstacles que s'han de tenir en compte a l'hora de dissenyar un algoritme, de manera que o bé s'aconsegueixen algoritmes que contemplin tota la infinitat de paràmetres possibles o bé s'accepta que la computació tradicional resulta poc eficaç per solucionar determinats problemes.

4.2.- LA MEMÒRIA DE LES XARXES NEURONALS ARTIFICIALS

Una altra característica de les xarxes neuronals artificials és la memòria distribuïda. En un ordinador, la memòria ocupa un lloc determinat; en canvi, a les XNA la informació es troba repartida per totes les connexions. Això fa que el sistema sigui tolerant als errors, és a dir, pot suportar la pèrdua d'una neurona sense que es produeixi un error que al seu torn provoqui la fallada del sistema. A nivell biològic passa, de fet, exactament el mateix: el cervell pot perdre neurones diàriament sense que el funcionament d'aquest es vegi perjudicat, almenys no fins passats molts anys.

4.3.- ARQUITECTURA TRADICIONAL I ARQUITECTURA NEURONAL

Les XNA presenten una arquitectura totalment diferent a la dels ordinadors tradicionals d'un únic processador. Les màquines basades en el model de Von Neumann –és a dir, les màquines de computació tradicional– tenen un únic element processador, la CPU (Control Process Unit), que realitza càlculs executant totes les instruccions de la seqüència programada en l'algoritme. En canvi, una xarxa neuronal utilitza molts processadors simples que actuen paral·lelament –o el que és el mateix, de manera simultània– i que estan interconnectats entre si, tot autoorganitzant-se i aprenent d'exemples.

Aquest processament en paral·lel de les XNA és clau. Un ordinador convencional processa instruccions, una rere l'altre. Si el volum de dades a processar és molt gran, el temps de resposta també és llarg. Els ordinadors actuals de gamma o bé mitjana o bé alta tenen diversos nuclis (els processadors o "cervells" dels ordinadors), ja que si

s'augmenta el número de processadors, aquests es reparteixen les tasques i el resultat es tradueix en un augment de la velocitat de resposta. Tanmateix, quan aquesta velocitat no és suficient, sorgeix un problema que només es pot solucionar recorrent a supercomputadores d'alta complexitat i d'elevat cost. Ara bé, en una xarxa neuronal hi pot haver milers de neurones operant al mateix temps. Això fa que en molts casos les XNA puguin oferir respostes en temps real, mentre que no és possible dur-ho a terme en un ordinador convencional.

4.4.- TAULA COMPARATIVA:

S'ha considerat oportú crear una taula resumint les característiques que més defineixen el tipus de computació, possibilitant així una fàcil comparació.

	Computació tradicional	Computació neuronal
Especialitat	Problemes de números i símbols	Problemes de percepció
Memòria	Separada del processador Localitzada	Integrada al processador Distribuïda
Processament	Seqüencial Algoritmes de programació	Paral·lel Auto-aprenentatge
Fiabilitat	Vulnerable	Consistent
Processador	Complex D'alta velocitat Un o uns quants	Simple Velocitat en temps real Un gran nombre

Fig. 12. Taula que permet comparar la computació tradicional amb la computació neuronal.

5.- EL FUNCIONAMENT DE LES XARXES NEURONALS ARTIFICIALS

Les XNA són models que intenten reproduir el comportament del cervell. Ara bé, com ja s'ha dit anteriorment, el cervell és extremament complex. Resulta que aquest òrgan només es pot descriure amb una quantitat d'informació que és excessiva i redundant a parts iguals. Això fa que per garantir l'èxit en la construcció d'una xarxa neuronal artificial, s'hagi de recórrer a una selecció d'aquells elements que desenvolupen un paper realment rellevant en el funcionament del cervell. Aquests elements, traduïts al llenguatge computacional, són els que compondran l'estructura bàsica del sistema i intervindran de manera directa en el seu funcionament. Són els següents:

- Les neurona artificial
- El processament d'informació: regles i funcions matemàtiques
- Classificació de l'arquitectura
- L'entrenament
- Els algoritmes d'aprenentatge

5.1.- LA NEURONA ARTIFICIAL

La neurona artificial actua com un processador simple que a través d'informació d'entrada, procedent de l'exterior o d'altres neurones, dona lloc a una sortida o resposta. El model general de neurona artificial està compost per:

- **Entrades:** Les entrades es caracteritzen per tenir la capacitat de captar senyals de l'exterior o d'altres neurones. La informació rebuda es manifesta normalment en forma discreta⁸, sent la més habitual la binària (0 o 1). Ara bé, tot i que és més inusual, hi ha xarxes que utilitzen valors continus⁹.
- **Sortides:** Les sortides es caracteritzen per rebre la informació ja processada i donar una resposta. Aquesta resposta és un valor que s'acostuma a expressar, de la mateixa manera que les entrades, de forma discreta -així, la informació de la sortida d'una neurona pot passar a ser la informació d'entrada d'una altra neurona i permetre una estructura de xarxa-.
- **Pesos sinàptics:** Per a les connexions o sinapsis que uneixen les neurones, la informació tan sols pot dirigir-se en un únic sentit en cada connexió. Degut a això, es parla de neurones presinàptiques i postsinàptiques. Respectivament,

⁸ Variable discreta: es tracta d'una variable que només pot tenir valors dins d'un conjunt numerable, és a dir, no accepta qualsevol valor, només aquells que pertanyen al conjunt. Per exemple, agafant el conjunt de nombres naturals de l'1 al 10 hi ha l'1, el 2, el 3... i el 10.

⁹ Variable contínua: es tracta d'una variable que agafa tots els valors limitats per un interval. Aquests valors, doncs, són infinits. Per exemple, agafant l'interval que va de l'1 al 10 hi ha l'1, l'1,1, l'1,11, el 2,034...

unes envien la informació, mentre que les altres la reben. Ara bé, tot i que per a cada connexió només és permès un sentit, entre dues neurones hi pot haver més d'una connexió, de manera que la retroalimentació és possible (Fig. 13.). Els pesos sinàptics, doncs, són els valors que tenen aquestes connexions entre neurones i que acaben influint en l'entrada de la neurona postsinàptica. L'objectiu dels pesos és representar la intensitat d'interacció entre la neurona presinàptica i la neurona postsinàptica, factor que també es dona en les neurones biològiques. D'aquesta manera, els pesos són un factor clau per determinar si una neurona ha d'estar activa o inactiva. Si el pes és positiu, és a dir, excitador, fomentarà l'activació de la neurona. En canvi, si pel contrari, el pes és negatiu, aquest en fomentarà la inhibició.

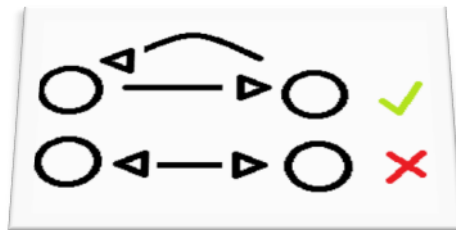


Fig. 13. La retroalimentació és possible, però que els pesos sinàptics circulin en dos sentits no (ja que cada neurona només pot donar o rebre informació a la vegada).

5.2.- PROCESSAMENT DE LA INFORMACIÓ: REGLES I FUNCIONS MATEMÀTIQUES

Per tal de possibilitar el processament de la informació –i per tant garantir el correcte funcionament de les xarxes neuronals–, aquestes necessàriament són dependents d'unes determinades regles i funcions matemàtiques.

5.2.1.- Regla de Propagació

La regla de propagació és una funció que processa el potencial postsinàptic o, en altres paraules, la informació total que arriba a la neurona. Els factors que componen aquest potencial són els pesos de les connexions i les entrades de la neurona.

La funció més habitual i més utilitzada en la regla de propagació consisteix en la suma de les diferents entrades de la neurona multiplicades pels seus respectius pesos sinàptics (Fig. 14).

$$h(t) = \sum w_{ij}x_j$$

Fig. 14 Expressió matemàtica de la funció que reflecteix la regla de propagació, on w_{ij} és el pes sinàptic entre la neurona presinàptica (i) i la neurona postsinàptica (j) i x_j és el conjunt d'entrades de la neurona postsinàptica (j). La lletra (h) representa el valor que té el potencial postsinàptic en un temps (t) determinat.

5.2.2.- Funció d'activació

La funció d'activació dóna com a resultat l'estat d'activació actual de la neurona a partir del potencial postsinàptic i del seu estat d'activació anterior (tot i que només té rellevància quan es té en compte el factor temps).

Pel que fa a l'estat d'activació de la neurona (j), existeixen tres possibles solucions en funció del valor del pes sinàptic w_{ij} . (Fig. 14.)

- Si és positiu, hi haurà un efecte excitador, de manera que la neurona (j) passarà a un estat actiu.
- Si és negatiu, hi haurà un efecte inhibitor, de manera que la neurona (j) passarà a un estat inactiu.
- Si el valor és 0, significarà que no hi ha cap connexió entre la neurona (i) i la neurona (j).

5.2.3.- Funció de transferència o de sortida

Aquesta funció s'aplica al valor que s'obté de la funció d'activació, proporcionant així un resultat final. S'utilitza per acotar la sortida d'una neurona, pel que el tipus de funció d'activació dependrà de la interpretació que es vulgui donar a les sortides. Algunes de les més utilitzades són la funció sigmoide i la funció esgraó, ja que ambdues permeten obtenir valors discrets i binaris com es pot observar a la taula (Fig. 15.)

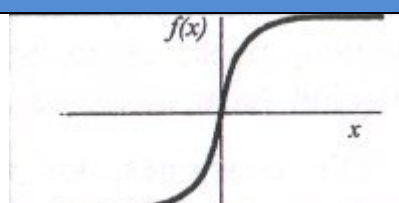
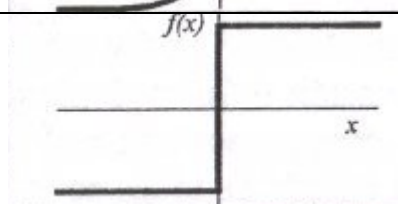
Nom	Funció	Recorregut (interval)	Gràfic
Sigmoide	$y = \frac{1}{1 + e^{-x}}$ $y = \operatorname{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Esgraó	$y = \operatorname{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	

Fig. 15. Taula sobre les dues funcions de transferència més comunes; la sigmoide i l'esgraó. Es poden aplicar variants de la funció per tal d'obtenir un recorregut diferent.

A la figura 16 es pot veure, de manera completa, el funcionament d'una neurona artificial. Unes entrades tenen uns determinats pesos que són modificats a partir de la regla de propagació. El potencial postsinàptic resultant passa per una funció d'activació, que acabarà sent el valor de la sortida de la neurona j .

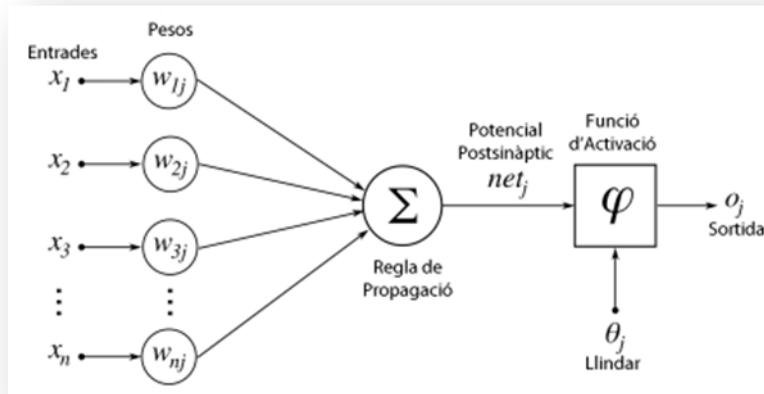


Fig. 16. Expressió gràfica del funcionament d'una neurona artificial.

5.3.- CLASSIFICACIÓ DE L'ARQUITECTURA

L'arquitectura d'una xarxa és l'estructura o patró segons el qual es connecten les neurones d'aquesta. L'estructura de les connexions determinarà el comportament de la xarxa, és a dir, si aquesta canvia, les respostes del conjunt de la xarxa també variaran. Existeixen diferents maneres de classificar l'arquitectura de les xarxes, depenent de quin sigui l'element que més interessa observar.

5.3.1.- Segons el número de capes

Convé recordar que les neurones s'agrupen en diferents capes. Alhora, aquestes capes estableixen connexions entre elles, fent del sistema neuronal una xarxa.

- **Xarxes neuronals monocapes:** es correspon amb el tipus de xarxa neuronal més senzilla, ja que només hi ha una capa de neurones que projecten les entrades a una capa de neurones de sortida, on es fan diferents càlculs. El nom de monocapa sorgeix del pensament que, com que la capa d'entrada no realitza cap tipus de càlcul, no pot ser considerada una capa funcional.

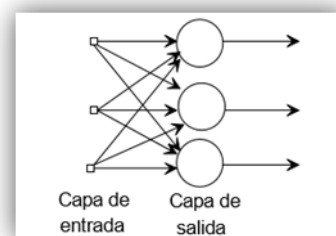


Fig. 17. Xarxa monocapa. El sentit de la informació va de la capa d'entrada a la capa de sortida.

- **Xarxes neuronals multicapa:** es tracta d'una generalització de l'exemple anterior amb la única diferència que hi ha un conjunt de capes situades entre la

capa d'entrada i la de sortida. Aquestes capes reben el nom de capes ocultes. Aquest tipus de xarxa pot estar totalment o parcialment connectada

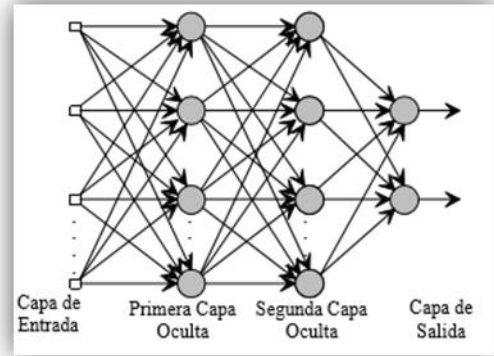


Fig. 18. Xarxa multicapa. El sentit de la informació va de la capa d'entrada a la capa de sortida, passant per les capes ocultes.

5.3.2.- Segons el tipus de connexions

- **Xarxes neuronals no recurrents:** En aquesta xarxa la propagació dels senyals només es produeix en un sol sentit, sense que existeixi la possibilitat de retroalimentació (les figures 17 i 18 en són exemples).
- **Xarxes neuronals recurrents:** Aquesta xarxa es caracteritza pels seus llaços de retroalimentació, que es poden establir entre neurones de diferents capes, de la mateixa capa, o encara més simple, entre la mateixa neurona. El tipus d'estructura recurrent fa que la xarxa tingui una gran capacitat per estudiar la dinàmica de sistemes no lineals.

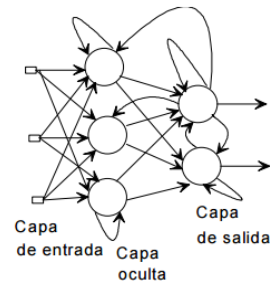


Fig. 19. Capa recurrent. La informació no circula en un únic sentit ja que es recorre a la retroalimentació.

5.3.3.- Segons el grau de connexió

- **Xarxes neuronals totalment connectades:** En aquest cas totes les neurones d'una capa estan connectades amb les de la capa següent (propi de les xarxes no recurrents) o amb les de l'anterior (propi de les xarxes recurrents).
- **Xarxes parcialment connectades:** En aquest cas no existeix una connexió total entre neurones de diferents capes, ja que la connexió es perd en alguna de les capes ocultes –el fet que hi hagi capes ocultes significa que només es pot donar en xarxes multicapa–.

5.4.- ENTRENAMENT

Les xarxes neuronals artificials, a diferència dels mètodes emprats en la computació tradicional –que es basen en instruccions prèviament programades–, han de ser prèviament entrenades, com ja s'ha esmentat a l'apartat 3.3. L'entrenament és el procés pel qual s'ajusten els paràmetres de la xarxa neuronal per tal que aquesta realitzi un processament determinat fent ús d'exemples. Això significa que es mostren uns exemples a la capa d'entrada de la xarxa i posteriorment ella mateixa s'ajusta en funció d'alguna regla d'aprenentatge.

Habitualment els paràmetres a ajustar són els pesos de les connexions, tot i que durant l'entrenament també es poden crear o destruir neurones.

Per fer un sistema neuronal s'ha de triar un model de neurona i el tipus d'arquitectura de la xarxa. Els pesos de les connexions, tanmateix, poden ser nuls o aleatoris en un principi, i és l'entrenament el que s'encarregarà de donar-los els valors adequats. Com és de suposar, doncs, una xarxa sense entrenar no té cap tipus d'utilitat i és per això que aquest procés adquireix tanta rellevància.

En el cervell humà, el coneixement es troba en la sinapsi, la connexió entre neurones. Quan una xarxa s'entrena i aprèn, igual que els humans, obté coneixement. Aquesta informació obtinguda s'ha de manifestar d'alguna manera, i ho fa a través dels pesos. Es pot determinar, doncs, que en el cas de les xarxes neuronals el coneixement es troba en els pesos, l'element equivalent a la sinapsi. És per això que tot procés d'aprenentatge implica un cert nombre de canvis o modificacions en els valors dels pesos.

De fet, el coneixement de la xarxa no es guarda en un lloc concret com sí que passa en el cas dels ordinadors convencionals, sinó que aquest coneixement es reparteix entre tots els pesos i com a resultat, queda distribuït per tota la xarxa. Passa exactament el mateix amb els humans; no hi ha una zona determinada del cervell on es guardi la informació referent a l'any en què va començar la Revolució Francesa, el dia en què un amic celebra l'aniversari, etc.

L'entrenament es fa utilitzant el que s'anomena una regla o algoritme d'aprenentatge, que va modificant de manera contínua els pesos reduint així l'error. Tanmateix, perquè això sigui possible, s'han de fer iteracions, és a dir, s'han de presentar exemples a la xarxa de manera que la regla d'aprenentatge pugui modificar el pes de cada connexió. Cal fer iteracions fins que es pugui considerar que la xarxa ha "après" dels exemples donats. Si, a més, la xarxa gaudeix d'una bona capacitat de generalització, al final de l'entrenament el sistema neuronal no només donarà respostes vàlides pels exemples que se li han proporcionat, sinó que també ho farà per a altres situacions. Lògicament, els errors que pugui fer la xarxa dependran de manera directa de la quantitat i el tipus d'exemples que se l'hi ha donat.

5.5.- ALGORITMES D'APRENTATGE

Els algoritmes d'aprenentatge són el mètode que s'aplica durant l'entrenament perquè les xarxes puguin aprendre dels exemples que se'ls proporciona. Es poden classificar en dos grups:

- Aprenentatge supervisat
- Aprenentatge no supervisat

5.5.1.- Aprenentatge supervisat:

Els algoritmes d'aquest tipus d'aprenentatge requereixen que cada informació d'entrada sigui emparellada amb el seva informació de sortida. Això significa que l'entrenament consistirà en presentar un exemple a la xarxa, calcular una resposta, i comparar aquesta resposta amb la que hauria d'haver donat o el que és el mateix, la sortida desitjada. L'error o diferència entre sortida real i sortida desitjada s'utilitza per retroalimentar la xarxa i canviar els pesos d'acord amb un algoritme que tendeix a minimitzar l'error. Aquest procés es repeteix fins que l'error equivalgui a un valor petit i acceptable.

Existeixen tres maneres d'enfocar l'aprenentatge supervisat:

- **Aprenentatge per correcció d'error:** Consisteix en ajustar els pesos de les connexions de la xarxa en funció de la diferència entre els valors desitjats i els valors obtinguts de la sortida de la xarxa. Cada error que comet la xarxa significa una modificació o correcció en els pesos.
- **Aprenentatge per reforç:** Es tracta d'un aprenentatge més lent, degut a que aquest model no corregeix els errors. Si la xarxa envia una sortida que s'ajusta a la que es desitja, s'envia un senyal de reforç que fomenta el correcte funcionament de la xarxa.
- **Aprenentatge estocàstic:** Consisteix en canviar aleatòriament els valors dels pesos de les connexions de la xarxa i avaluar l'efecte que produeix tenint en compte la sortida que es desitja.

5.5.2.- Aprenentatge no supervisat:

Els sistemes neuronals amb aprenentatge supervisat han tingut èxit en moltes aplicacions. No obstant, reben moltes crítiques perquè des d'un punt de vista biològic no tenen cap mena de lògica. És difícil creure que hi ha un mecanisme en el cervell que compara les sortides desitjades amb les reals; el cervell no sap quina és la sortida desitjada, simplement té les sortides que té i no hi ha cap mecanisme que les reguli. Per tant, en aquest tipus d'aprenentatge només es proporciona informació d'entrada, obviant la informació de la sortida desitjada.

El procés d'entrenament es basa en organitzar les diferents entrades: es localitzen regularitats, s'extreuen característiques comunes o s'organitzen les entrades segons algun altre tipus de similitud. Aquest mètode s'utilitza normalment per organitzar grans quantitats de dades, com podria ser per exemple algun cas d'anàlisi estadístic.

Hi ha dos tipus d'aprenentatge no supervisat:

- **Aprenentatge hebbià:** Consisteix en l'ajustament dels pesos de les connexions d'acord amb la correlació positiu-negatiu (+1 i -1) dels valors d'activació (sortides) de les neurones connectades. Quan dues accions són

actives (resultat positiu) la connexió és reforçada, però si una és activa i l'altra passiva (resultat negatiu) la connexió es debilita.

- **Aprentatge competitiu i cooperatiu:** Aquesta “competició” es dona entre totes les capes de la xarxa. En l'aprenentatge competitiu les neurones tenen connexions recurrents d'excitació amb elles mateixes i connexions d'inhibició amb les neurones veïnes.

En canvi, en l'aprenentatge cooperatiu totes les connexions, tant les que estableix una neurona amb ella mateixa com les que estableix amb les neurones veïnes, són d'excitació. L'objectiu d'aquest aprenentatge és que les informacions similars formin part de la mateixa categoria i per tant, activin la mateixa neurona de sortida. Les categories són creades per la mateixa xarxa, característica pròpia de l'aprenentatge no supervisat.

6.- HISTÒRIA DE LES XARXES NEURONALS ARTIFICIALS

6.1.- EL PREDECESSOR DE LES XNA:

La ciència de les Xarxes Neuronals Artificials va fer la seva primera aparició significativa als anys 40. L'inici d'aquesta es va donar quan un grup d'investigadors especialitzats en neurobiologia van intentar reproduir les funcions del cervell humà, desenvolupant, així, uns models físics¹⁰ amb l'únic objectiu d'ampliar els escassos coneixements que es tenien aleshores sobre el sistema neuronal humà. Ara bé, amb el pas del temps, la complexitat sobre l'objecte d'estudi es va anar fent cada vegada més evident, i aquests primers models van ser considerats aproximacions rudimentàries.

6.2.- MCCULLOCH I PITTS

Si bé és veritat que la simplicitat dels models físics del cervell va esdevenir un problema pel que fa als estudis de caire biològic, aquesta mateixa simplicitat va resultar ser un factor clau que va permetre a dos visionaris, Warren McCulloch i Walter Pitts, donar un enfocament de tipus computacional a aquests models. Mentre que McCulloch tenia estudis de filosofia i psicologia, Pitts era un matemàtic que destacava en l'àmbit de la lògica. El 1942 van decidir treballar junts per escriure l'article "*A Logical Calculus of Ideas Immanent in Nervous Activity*"¹¹, que va ser el resultat de qüestionar si el sistema nerviós pot ser considerat una màquina de computació universal. Més tard, el 1951, van aconseguir entrar a l'Institut de Tecnologia de Massachusetts (MIT) per desenvolupar les seves idees i així convertir-se en els pioners de les xarxes neuronals artificials, tot perfeccionant el primer model matemàtic sobre les neurones, conegut avui pel nom de "La Neurona McCulloch-Pitts".

6.2.1.- La neurona de McCulloch-Pitts

Tot i que l'essència d'aquest model va ser proposada el 1943, els seus autors van aconseguir posar-lo a prova i millorar-lo després d'anys treballant entorn aquesta primera concepció de les xarxes neuronals artificials. Sens dubte, és la seva forma més bàsica, però és indiscutible que va establir un fonament necessari i suficientment influent com perquè amb el pas del temps, experts interessats en aquest camp tinguessin la oportunitat de fer-hi les seves pròpies aportacions.

No es pot negar que la idea d'implementar una rèplica del sistema nerviós animal a la computació té un enorme potencial, però alhora tampoc es pot posar en dubte que és un projecte ambiciós. Així doncs, la realitat és que un model bàsic equival a un model que resol problemes encara més bàsics.

¹⁰ Els models físics són mètodes utilitzats per investigar detalls particulars d'un sistema que no és conegut amb precisió.

¹¹ El títol d'aquest article, es tradueix al català com a "Un càlcul lògic de les idees propi de l'activitat nerviosa".

6.3.- APRENTATGE HEBBIÀ

Cal recordar que un dels principals reptes a l'hora de dissenyar una xarxa neuronal és el fet de triar, calcular o buscar el valor apropiat pels pesos de les entrades en funció del problema que es vol resoldre. Com s'ha descrit anteriorment, aquesta tasca rep el nom d'aprenentatge o entrenament de la xarxa.

Des d'un punt de vista històric, s'ha de ressaltar l'aprenentatge hebbià per ser un dels més utilitzats i estudiats. A partir de les seves investigacions, el neuropsicòleg D.O. Hebb va escriure el 1949 el llibre "L'organització del comportament", en el qual estableix una connexió entre psicologia i fisiologia. Va ser un dels primers en explicar el procés d'aprenentatge -que resulta ser un element bàsic de la intel·ligència humana- des d'un punt de vista psicològic. La teoria de Hebb es basa en el següent: "quan un axó presinàptic causa l'activació d'una neurona postsinàptica, l'eficàcia de la sinapsis que les relaciona es reforça". Traslladat al camp de les xarxes neuronals artificials, això significa que els pesos es modifiquen en proporció al producte entre l'entrada d'una neurona i la sortida d'una altra.

Tot i que aquest és un aprenentatge simple i que es realitza a nivell de cada neurona, s'ha pres com a punt de partida per a molts altres algoritmes d'aprenentatge. A més, treballs experimentals posteriors confirmen la validesa de les teories de Hebb, afirmant, així, que aquest tipus d'aprenentatge es dona en les neurones biològiques juntament amb altres mecanismes.

6.4.- EL PERCEPTRÓ

El 1956, un grup d'investigadors de la Intel·ligència Artificial (Minsky, McCarthy, Rochester i Shanon) van organitzar una conferència sobre les xarxes neuronals. Aquest esdeveniment va tenir lloc a la localitat anglesa de Darmouth, i adquireix certa rellevància ja que es fa referència a aquest moment com a la primera vegada en què les xarxes neuronals artificials van ser considerades una opció viable que cal fonamentar.

L'impacte de la conferència de Darmouth no va tardar a donar resultats. El 1957, Frank Roseblatt va publicar el treball d'investigació més important en aquella època pel que fa a la computació neuronal, que consisteix en el desenvolupament d'un element anomenat "perceptró".

El perceptró és un sistema classificador que pot identificar patrons geomètrics i abstractes. El primer perceptró ja presentava una de les característiques més interessants de les XNA, la tolerància als errors. Això significa que el seu comportament només varia si resulten perjudicats els principals components del sistema. Ara bé, aquest innovador element presentava algunes limitacions degut al fet que encara estava en vies de desenvolupament. Els que les van posar en manifest van ser Minsky i Papert, anys després que el perceptró fos presentat.

6.5.- ADALINE I MADALINE

El 1959, Bernard Widrow va desenvolupar a Stanford la primera xarxa neuronal aplicada a un problema real. Aquesta va rebre el nom de “Adaline” (ADAPtative LINear Elements) i van anomenar la seva versió en dues capes “Madalaine”. Va ser destinada sobretot a la creació de filtres adaptatius per eliminar els ecos de les línies telefòniques, sistema que es va aplicar durant varies dècades. A part del reconeixement de veu, aquesta xarxa també va ser enfocada al reconeixement de caràcters i a la predicció del temps, sense tant d'èxit.

6.6.- MINSKY I PAPERT: “TOY PROBLEMS”

L'eufòria inicial dels anys 60 va ser substituïda per desil·lusió quan Minsky i Papert, de l'Institut Tecnològic de Massachussets (MIT), van publicar el 1969 el llibre “Perceptrons”. Van demostrar matemàticament que el perceptró no era capaç de resoldre problemes relativament fàcils, com per els que requereixen l'aprenentatge d'una funció no-lineal. Això va demostrar que en realitat el perceptró era molt dèbil, ja que les funcions no-lineals són utilitzades molt freqüentment en la computació convencional.

En el llibre, s'esmenta sense embuts que les XNA no poden resoldre “problemes de juguina”. Aquestes dures crítiques van indicar d'una manera tan rotunda que la computació neuronal es tractava en realitat d'una pèrdua de temps, que hi va haver una disminució dràstica en les inversions i, com a conseqüència, es van frenar en gran mesura les investigacions que aleshores estaven en procés.

Tot semblava apuntar que, irònicament, el que va començar al MIT amb McCulloch i Pitts, acabaria al MIT amb Minsky i Papert.

6.7.- A CONTRACORRENT: ANDERSON, KOHONEN I GROSSBERG

Un dels escassos investigadors que va continuar amb el seu treball en la computació neuronal després de la publicació del llibre “Perceptrons” va ser James Anderson. El seu treball (1972) consisteix en la creació d'un model lineal i associatiu que es basa en el principi o teoria de Hebb –és a dir que les connexions d'aquest model són reforçades (incrementen el seu valor) amb la contínua activació de les neurones–.

Teuvo Kohonen, de La universitat de Hèlsinki, va ser juntament amb Anderson un dels impulsors de la computació neuronal més destacats de la dècada dels 70. Tot i que Kohonen era enginyer elèctric i Anderson neurofisiòleg, el treball que va realitzar Kohonen s'assemblava força al d'Anderson, ja que els dos models es basen en la memòria associativa simultània que utilitza (de manera adaptada) la regla de Hebb.

Un altre investigador que va continuar treballant en aquest tema després de les crítiques de Minsky i Papert va ser Stephen Grossberg. Es va interessar especialment en fer servir dades d'estudis neurològics per construir models de computació neuronal. Això va comportar que la majoria de les seves regles i postulats derivessin d'estudis de

caire fisiològic. Un dels models més rellevants que va crear el 1977 va ser l'anomenat "Adaptative Resonance Theory" (ART). Aquesta xarxa té una arquitectura que es diferencia de totes les que s'havien inventat en aquella època, ja que imita algunes habilitats del cervell com la memòria a llarg i a curt termini.

6.8.- JOHN HOPFIELD

John Hopfield va representar, metafòricament, un oasis per la computació neuronal després d'una dècada de caminar pel desert. El 1982 realitza un treball clau pel ressorgiment de les xarxes neuronals. Gran part de l'impacte no va ser conseqüència del treball en si mateix, sinó de la fama de Hopfield com a distingit físic teòric. Així doncs, aquest conegut personatge va presentar un model que consistia en elements processadors (neurones) interconnectats que tendien al mínim consum d'energia, aconseguint així una optimització de recursos.

Un fet curiós és que Grossberg va considerar aquest mecanisme, però mai el va arribar a implementar.

6.9.- RUMELHART, HINTON I LA RETROPROPAGACIÓ (BACKPROPAGATION)

El 1986 apareix un treball que resulta decisiu per acabar de ressuscitar aquest interès perdut per les xarxes neuronals. Rumelhart i Hinton desenvolupen l'algoritme d'aprenentatge anomenat retropropagació per XNA multicapa. A partir d'aquell any, el número de treballs sobre xarxes neuronals ha anat augmentant exponencialment i s'han realitzat moltes aportacions tant pel que fa als mètodes d'aprenentatge com en les arquitectures o les aplicacions d'aquestes xarxes. Una mostra de l'èxit que tenen és la implementació d'aquest mètode per part de l'empresa Google.

7.- ENTREVISTA AMB UN EXPERT:

Es va considerar oportú entrevistar a un expert en el camp de la computació neuronal, el Dr. Jordi Solé (color blau). En l'entrevista es tracten qüestions ètiques que fan referència a la intel·ligència artificial i també qüestions tan teòriques com pràctiques sobre les xarxes neuronals artificials.

- **Opines que la intel·ligència artificial podrà igualar, o com a mínim apropar-se a imitar el cervell (i per tant la intel·ligència humana) en un futur?**

El cervell humà és d'una complexitat enorme i mai es podrà replicar exactament de forma artificial, igual que tampoc es poden replicar altres òrgans del cos. El que sí que es pot fer són òrgans artificials que imitin però no igualin els òrgans originals.

L'objectiu de la intel·ligència artificial és el d'ajudar a fer tasques de manera independent a través d'agents no humans (diguem-ne màquines) que prendran les seves decisions a partir de les condicions del seu entorn. Aquestes decisions seran fruit d'un aprenentatge fet al llarg del temps. Així doncs, per a mi, aquest procés d'aprenentatge és el factor més rellevant i el que defineix la intel·ligència artificial.

- **El test de Turing és una eina vàlida per determinar si una màquina és intel·ligent?**

Determinar si una màquina és intel·ligent o no depèn de què s'entengui exactament per intel·ligència. Si el que es vol és saber si una acció l'ha fet una màquina o un home, actualment hi ha testos com el "Captcha¹²" que permeten veure-ho. De totes maneres, en un futur els sistemes artificials probablement seran prou capaços d'extreure el text distorsionat de la imatge i per tant es necessitaran altres sistemes per determinar si l'usuari es tracta d'una màquina o un ésser humà.

Per concloure, diré que una cosa és ser intel·ligent i l'altra és semblar-ho... i una màquina podria semblar-ho de forma bastant senzilla.

Saps algun cas concret en què s'hagin hagut de fer servir les xarxes neuronals pel fet de que resultaven més pràctiques que un altre mètode? Si és així, podries explicar breument els casos i quines són les característiques de les XNA que han influenciat més la decisió?

¹² Captcha: és l'acrònim de «Completely Automated Public Turing test to tell Computers and Humans Apart» (en anglès, "test de Turing públic i automàtic per a diferenciar màquines d'humans").

D'exemples n'hi ha moltíssims. El tema és fer servir les xarxes neuronals en els casos en què realment poden funcionar millor que altres sistemes. Les característiques que es busquen en aquests casos són, essencialment, la seva capacitat de generalitzar, la seva tolerància a fallades, la seva capacitat d'aprenentatge i la capacitat que tenen de processar en paral·lel, fent que es puguin resoldre els problemes en menys temps.

- **Saps alguns exemples quotidians en què es fan servir xarxes neuronals artificials? Per exemple, s'utilitza Google diàriament però la majoria d'usuaris no saben que aquesta companyia (en part) utilitzen les xarxes neuronals per a millorar el seu sistema de cerca.**

N'hi ha molts i en molts camps diversos. Per exemple, en determinar el millor programa per a un rentavaixelles, o per a la detecció de cares en una càmera de fotografiar, per al reconeixement òptic de caràcters (OCR), per al control de tràfic, etc. Al nostre grup de recerca (Grup de Tractament de Dades i Senyals, Universitat de Vic – Universitat Central de Catalunya) les hem utilitzat per al reconeixement biomètric de persones (cares), per desenvolupar sistemes d'ajuda al diagnòstic precoç de la malaltia de l'Alzheimer o fins i tot per aproximar funcions.

- **Creus que les XNA són un mètode molt efectiu per a uns determinats problemes? Estan en el seu estat d'esplendor o es poden millorar?**

Les XNA són efectives en casos on altres sistemes no funcionen prou bé. Per exemple, reconèixer patrons en una imatge és una cosa difícil de formular en termes d'algorisme: què hem de buscar exactament per saber si hi ha una cara en una foto? Com sabem diferenciar què és un ull d'una boca? Com ens adonem que un subjecte porta ulleres o no?...

Fixa't que totes aquestes coses l'humà les fa de manera natural, sense pensar i sense confondre's en cap cas. En canvi, intentar formular regles per fer això de forma general és realment difícil. En aquest cas, una XNA pot funcionar molt bé aprenent a partir d'exemples de cares de diferents tipus.

És important tenir clara aquesta diferència: mentre que els sistemes de computació convencionals aprenen a partir de regles, les XNA aprenen a partir d'exemples. Aquestes resolen problemes, comproven la validesa del resultat obtingut i a partir d'aquí adapten el seu comportament.

- **La computació tradicional o de Von-Neumann és la més utilitzada actualment. En algun moment la balança es podria inclinar a favor de la computació neuronal? O és que potser no s'ha de parlar tant de competència entre els dos tipus de computacions sinó més aviat de convivència?**

No es tracta de saber quin sistema és millor sinó quin sistema cal aplicar en cada cas. Evidentment, si es saben exactament els passos a seguir per resoldre un problema, el que s'ha de fer és programar un sistema per realitzar aquests passos. Ara bé, si no es sap com s'ha de resoldre el problema, però es

tenen diversos exemples de relacions entrada-sortida, llavors un sistema com les xarxes neuronals pot resoldre el problema aprenent a partir dels exemples. Es podria fer una analogia amb la cuina: si es vol fer un plat nou i es posseeix la recepta, aquesta es segueix acuradament i així es soluciona el problema. Però si no es té la recepta, s'ha de fer alguna altra cosa, com ara mirar exemples d'altres plats similars i fer alguna cosa semblant amb els ingredients que es tinguin. Cada sistema té el seu camp d'aplicació i per tant, com comentes, no és tant una competència sinó una convivència.

- **Són necessàries més inversions en el camp de la computació neuronal? És una ciència que actualment està estancada? Si és així, què n'opinaríes de les inversions en cercar nous algorismes que puguin tenir un efecte semblant al del back-propagation (que va permetre la posada en funcionament del perceptró multicapa)?**

Sí, sempre es necessiten inversions en recerca, no només en el camp que comentes sinó en tots els camps. I no només en àmbits de tecnologia computacional o enginyeria, sinó també en àmbits de ciències de la vida, de ciències socials o d'història, per posar alguns exemples.

A mesura que evolucionen els sistemes de càlcul, van evolucionant els llenguatges de programació i les possibilitats de fer coses. També evolucionen els algorismes i les matemàtiques que intrínsecament contenen. Per tant, cal anar treballant en el desenvolupament de nous algorismes i noves estructures de xarxes neuronals que permetin millorar els resultats que es tenen actualment per així aconseguir resoldre nous reptes o resoldre problemes ja resolts però d'una manera més eficaç.

- **Deixant de banda la resolució de problemes, les xarxes neuronals ens poden sorprendre en altres aspectes. Navegant per internet m'he trobat amb impactants imatges creades per XNA (molt abstractes i creatives) i també amb un vídeo en què es podia sentir una cançó creada igualment per XNA (que no acabava de sonar malament). Creus que es pot arribar a l'extrem en què les màquines puguin participar en una activitat tan humana com ho és l'art?**

Sí, poden ser un complement molt interessant en diferents aspectes. De fet, ja es fan representacions que combinen un contingut tecnològic amb un d'humà. També es poden fer coses exclusivament tecnològiques, amb sistemes artificials que executin una obra (musical, pictòrica...).

8.- AVANTATGES I DESAVANTATGES

AVANTATGES

A l'apartat 3.1, ja s'ha fet una menció prèvia a les semblances que les XNA, per la seva constitució i fonament, presenten amb el cervell. Per exemple, són capaces d'aprendre de l'experiència, generalitzar de casos anteriors a nous casos, abstraure característiques essencials a partir d'entrades que representen informació irrellevant...

Entre els avantatges s'inclouen els que segueixen:

- **Aprenentatge adaptatiu:** consisteix en la capacitat d'aprendre a realitzar tasques basades en un entrenament inicial. D'aquesta manera, la màquina pot aprendre a dur a terme certes tasques mitjançant l'entrenament a partir d'exemples il·lustratius, motiu pel qual no és necessari l'elaboració de models.
- **Autoorganització:** les XNA permeten organitzar per si mateixes el que s'ha après. Mentre que l'aprenentatge és la modificació de cada element processador, l'autoorganització consisteix en la modificació de la xarxa neuronal completa per dur a terme un objectiu específic.
- **Tolerància als errors:** els primers models computacionals ja van gaudir d'aquesta característica; quan es produeixen errors en un número de neurones que no és exageradament gran, encara que el comportament del sistema sí que es vegi influenciat, la repercussió no és gaire negativa.
- **Operacions en temps real:** de tots els mètodes existents, les XNA són les més indicades per al reconeixement de patrons en temps real, degut al fet que treballen en paral·lel (processen múltiples dades simultàniament). Això ocasiona que els resultats s'actualitzin i s'obtinguin de manera simultània, motiu pel qual el temps de processament és molt més ràpid.
- **Fàcil inserció en la tecnologia existent:** és relativament senzill obtenir xips especialitzats per les xarxes neuronals (aquests es destinen a millorar la seva capacitat en certs treballs). Això facilita la integració de la computació neuronal en sistemes ja existents.

DESAVANTATGES

Les xarxes neuronals artificials no constitueixen un sistema perfecte. Encara que les seves característiques les dotin d'avantatges molt diversos, no es poden passar per alt, tampoc, els desavantatges que presenten. La majoria d'aquests inconvenients, però, semblen ser ocasionats per una manca del hardware i del software adequats.

- **Absència d'esquemes:** Un dels principals problemes que existeix en aquest tipus de computació és la falta de regles definitòries o d'esquemes a seguir.

Aquests últims tindrien la funció d'ajudar a construir una xarxa per a un determinat problema, fet molt necessari degut a la gran quantitat de factors que cal tenir en compte: l'algoritme d'aprenentatge, l'arquitectura, el número de neurones per capa, el número de capes, la representació de les dades...

- **El problema de “la caixa negra”:** La major part de l'esforç dedicat en la investigació de les xarxes neuronals s'ha centrat en el desenvolupament de nous algoritmes d'aprenentatge, l'exploració de noves arquitectures i l'expansió de l'aplicació de les XNA cap a nous camps. Ara bé, s'ha invertit poca atenció en desenvolupar procediments que permetin comprendre aquestes xarxes. Es podria dir que una XNA es presenta a l'usuari com una espècie de “caixa negra” que, d'alguna manera desconeguda, transforma les entrades en les sortides desitjades. Expressat en diferents paraules, no es pot saber immediatament com els pesos de la xarxa o els valors d'activació de les neurones estan relacionats amb el conjunt de dades. Així doncs, a diferència dels models estadístics clàssics, en les xarxes neuronals no es pot predir l'efecte que tindrà la informació de les variables d'entrada sobre la informació de les variables de sortida.

Tot i així, aquesta percepció sobre les XNA com una “caixa negra” no és del tot certa. De fet, des de finals dels anys 80, ja van dur-se a terme diversos intents que portaven per objectiu el desenvolupament d'una metodologia que permetés interpretar tot allò que és après per la xarxa, però com s'ha dit abans, els estudis sobre aquest tema són molt escassos. Aquests esforços dedicats a entendre les xarxes neuronals, a més, no només intenten determinar quines variables tenen més importància, sinó que també pretenen detectar les variables redundants o irrellevants, cosa que permetria obtenir un error de generalització més baix. Lamentablement, però, l'èxit de tals investigacions és totalment inexistent, no tant perquè l'objectiu que es proposen sigui inassolible, sinó perquè els mètodes utilitzats són força qüestionables i haurien de ser descartats per a futures indagacions sobre el tema.

9.- APLICACIONES

Les aplicacions de les xarxes neuronals són múltiples, ja que les seves característiques inherents, tant funcionals com estructurals, faciliten la seva adaptació i funcionament a situacions ben diverses. A continuació s'ofereix un petit tast d'algunes de les aplicacions més rellevants, juntament amb una breu definició i algun exemple d'aplicació real:

Conversió de text a veu: la conversió text-veu consisteix en canviar els símbols gràfics d'un text a un llenguatge parlat. Un exemple n'és el sistema de computació neuronal anomenat NetTalk, que és capaç de convertir text en fonemes amb l'ajuda d'un sintetitzador de veu.

Compressió d'imatges: la compressió d'imatges és la transformació de les dades d'una imatge a una representació diferent que requereix menys memòria. Cottrel, Munro i Zisper, enginyers de la Universitat de San Diego, van dissenyar un sistema de compressió d'imatges utilitzant una xarxa neuronal amb un factor de compressió de 8:1. Van aconseguir, doncs, comprimir una imatge fins a obtenir-ne una de vuit vegades més petita.

Reconeixement de caràcters: és el procés d'interpretació visual i classificació de símbols. Un exemple n'és el projecte dels investigadors de l'empresa Nestor Inc. (van desenvolupar un sistema de computació neuronal que, després d'un entrenament amb un tot conjunt de caràcters, és capaç de reconèixer un tipus de lletra que no hagi vist anteriorment.

Problemes de combinatòria: en aquest tipus de problemes, la solució requereix un temps de procés que és exponencial al nombre d'entrades. Un exemple és el problema del venedor; l'objectiu és triar el camí més curt possible que ha de realitzar el venedor per cobrir un número limitat de ciutats en una àrea geogràfica específica. Aquest problema va ser completat amb èxit per Hopfield (apartat ..) a través d'una xarxa neuronal.

Processat del senyal en el filtrat de soroll: les XNA poden ser utilitzades per eliminar el soroll d'un senyal, i aquest tipus de xarxes es caracteritzen per l'èxit en la seva implementació en casos reals. Un exemple és l'Adaline, explicat en l'apartat ...

Models econòmics i financers: una de les aplicacions més destacables és la creació de pronòstics econòmics pel que fa als preus de les existències, la producció de les collites, el volum de les ventes, etc. D'aquesta manera es pot fer un estudi sobre l'evolució d'una determinada empresa, analitzant-ne la solvència, la liquiditat i la rendibilitat. Per exemple, l'empresa Nestor va aplicar una xarxa neuronal per assignar assegurances de manera automàtica a la concessió de préstec hipotecaris amb un 84% d'èxit aproximadament.

10.- L'ART PROVINENT DEL REONEIXEMENT D'IMATGES

Com es podrà observar a continuació, les xarxes neuronals dedicades a resoldre problemes de reconeixement d'imatges tenen una curiosa peculiaritat. Són capaces de crear, amb les ordres adequades, imatges xocants que s'apropen molt al concepte d'art.

10.1.- Els “somnis” de les xarxes neuronals de Google

L'arxiconeguda empresa Google fa ús de les xarxes neuronals artificials per millorar el seu sistema de cerca i de classificació. Especialment les fa servir pels temes en què aquests sistemes destaquen i mostren el seu potencial, com el reconeixement de veu o el reconeixement d'imatges. Així doncs, gran part de les persones que a diari fa servir el motor de cerca de Google, de ben segur que no s'imagina que en realitat està utilitzant un sistema que imita el seu cervell.

El tipus de xarxes que seran introduïdes a continuació estan dissenyades per treballar amb el reconeixement d'imatges. Així doncs, amb la finalitat d'entrenar-les, els enginyers de Google els mostren imatges en una primera fase d'aprenentatge, de tal manera que si l'exemple mostrat es tracta d'un pont, aquestes en detectaran els detalls i les formes fins que acabin “comprent” el que és el concepte de “pont” i així, en puguin identificar d'altres.

Segons afirmen els professionals que treballen amb aquests mecanismes, existeix, tanmateix, un gran problema: resulta que alguns algoritmes d'aprenentatge donen millors resultats que d'altres, però no s'ha aconseguit, malgrat tot, esbrinar-ne el motiu. Així doncs, per evitar malgastar el temps en fer proves i més proves fins a trobar aquell mètode que proporcionés millors resultats, els enginyers van decidir estudiar el funcionament -sobretot enfocant-se en l'aprenentatge- de les xarxes neuronals.

Gràcies a aquest estudi, va sorgir la brillant idea de revertir el procés; en comptes de fer que les xarxes detectessin imatges, van fer-los mostrar el que havien après. Així, les imatges obtingudes com a resultat van ser tan abstractes, belles i impressionants que els treballadors de Google van decidir anomenar-les “somnis”. El fet que les màquines somniïn pertany a l'àmbit de la ciència ficció, però és innegable que les xarxes neuronals han aconseguit crear, a partir del que han “vist”, un contingut extraordinari que resulta impossible d'obtenir a través de la computació tradicional.



Fig. 20. Imatges creades per les xarxes neuronals de Google.

10.2.- Instapainting

Sense guardar cap relació amb l'empresa "Google", la pàgina web instapainting (<https://www.instapainting.com/ai-painter>) permet combinar, mitjançant xarxes neuronals, dues imatges obtenint així resultats que tenen una certa similitud amb les imatges vistes anteriorment. Aquesta pàgina em va ser recomanada per el Dr. Jordi Solé, i, se'ns dubte, em va captivar la manera com les xarxes neuronals extreuen característiques de dues imatges i les combinen en una sola, donant a lloc a, pel que fa a la meua opinió personal, art.

A partir d'una primera fotografia, s'han creat diferents imatges molt abstractes i variades. Aquestes són el resultat de combinar la primera fotografia amb diverses obres artístiques força famoses.

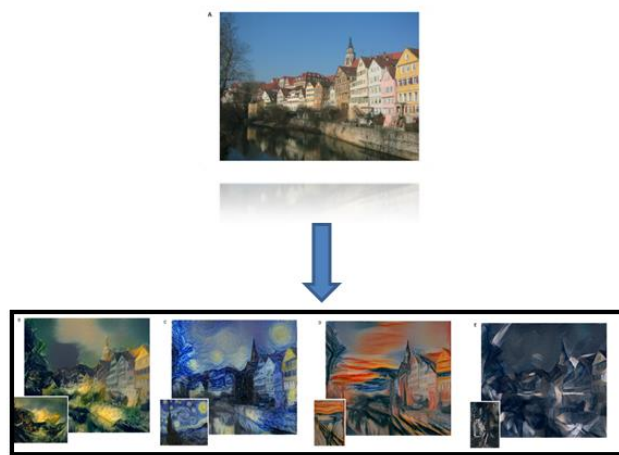


Fig. 21. Resultat de combinar una imatge quotidiana amb diverses obres pictòriques.

Per part meua, la curiositat em va poder i vaig decidir combinar també dues imatges. Com que el resultat em va agradar, ho vaig tornar a provar –de fet, la foto de la dreta correspon a la foto utilitzada per a la portada–.

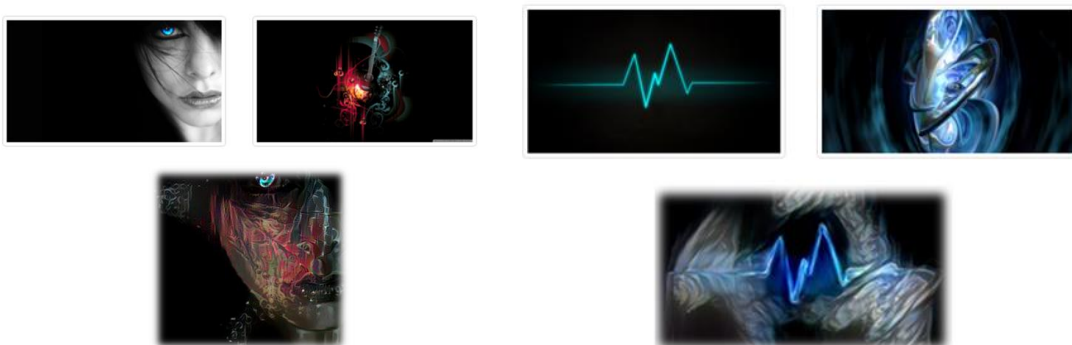


Fig. 22. Resultat de combinar diversos fons d'escriptori.

Es pot concloure, doncs, que les xarxes neuronals artificials tenen un cert potencial per desenvolupar imatges abstractes, inversemblants i sorprenents en certa manera. Si es pot o no considerar art, això ho decideix un mateix.

11.- EL PERCEPTRÓ

Anteriorment s'ha descrit tant el concepte com el funcionament de xarxa neuronal emprant una forma generalitzada i sense aprofundir en detalls tècnics. Ara bé, és important indagar més en la xarxa anomenada "perceptró", ja que serà la que constituirà els dos exemples de XNA que posteriorment s'analitzaran a l'apartat de procediment 14.2. En referència al contingut d'aquest apartat, es tractarà tant la definició com les característiques dels diferents tipus de perceptró i la seva història.

11.1.- EL PERCEPTRÓ SIMPLE

Un perceptró és un tipus de xarxa neuronal, sent més concretament un sistema classificador de patrons. Així doncs, s'especialitza en identificar patrons tant geomètrics com abstractes, i es caracteritza per ser una de les primeres xarxes anomenades "robustes", és a dir, tolerant als errors del sistema.

La xarxa tipus Perceptró va ser inventada pel psicòleg Frank Rosenblatt l'any 1957. La seva intenció era il·lustrar algunes propietats fonamentals dels sistemes intel·ligents en general, sense entrar en detalls específics (o desconeguts) d'organismes biològics concrets. Rosenblatt va dissenyar el primer perceptró de tal manera que fos capaç de realitzar un reconeixement òptic de patrons, motiu pel qual va rebre el nom de "fotoperceptró". Aquest primer model constava de fins a 400 fotocèl·lules o neurones artificials que pretenien imitar a les neurones fotosensibles pròpies dels ulls.

11.1.1.- Arquitectura

El perceptró està format per dues capes, una d'entrada i una altra de sortida. Cal recalcar que la capa d'entrada no realitza cap càlcul, ja que la seva utilitat es limita al simple propòsit de permetre l'entrada d'informació a la xarxa. Les neurones de cada capa s'actualitzen simultàniament i les capes s'executen en l'ordre intuïtiu, és a dir, de la capa d'entrada a la capa de sortida (que s'encarrega de processar la informació provinent de la capa d'entrada).

Les neurones d'una mateixa capa, ja sigui la d'entrada o la de sortida, no es connecten entre elles, però sí que es connecten amb les neurones de l'altra capa. Dit d'una altra manera, la capa d'entrada està totalment connectada amb la capa de sortida i viceversa (Fig. 23.).

El perceptró accepta únicament entrades i sortides binàries, que depenent del tipus de funció utilitzada poden ser les parelles 0 i 1 o bé -1 i 1.

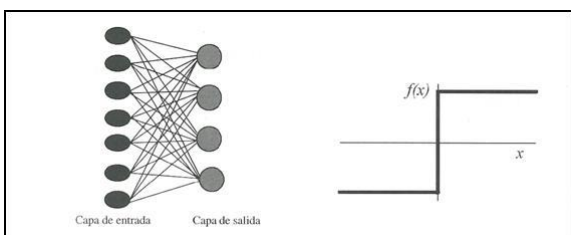


Fig. 23. Arquitectura (esquerra) i funció de transferència tipus esgraó (dreta) d'un perceptró simple.

11.1.2.- Funcionament

El perceptró es posa en funcionament quan les neurones de la segona capa realitzen la suma de totes les entrades de la xarxa i són multiplicades pel seu pes. Al valor obtingut se li resta el llindar, un valor lliure que es calibra durant l'entrenament de la xarxa.

$$y_i = f \left(\sum_{j=1}^n w_{ij}x_j - \theta_i \right)$$

On y_i és la sortida de la neurona i , w_{ij} és el pes de la connexió entre la neurona i i la neurona j , x_j és l'entrada de la neurona j i θ_i és el valor anomenat llindar.

Aquest procés es fa amb l'objectiu de complir regla de decisió, la qual dicta que el perceptró ha de classificar les diferents informacions que se li donen en dues classes. És per això que les funcions només poden donar resultats discrets binaris.

11.1.3.- Algoritme d'aprenentatge del perceptró simple

L'algoritme ideat per Rosenblatt permet determinar automàticament els pesos sinàptics i els llindars que classifiquen tot un conjunt de patrons. El perceptró té un aprenentatge supervisat, motiu pel qual no s'ha de proporcionar únicament el conjunt de patrons a la xarxa, sinó que també se li ha d'indicar les sortides esperades. És per aquesta raó que el perceptró aprèn a base d'anar corregint les equivocacions de la xarxa (comparant la sortida que dona davant d'un determinat patró amb la sortida que hauria de donar), modificant, així, els pesos i els llindars en funció de l'error.

Per a l'entrenament del perceptró, és necessària una xarxa amb tot de pesos i llindars aleatoris i un conjunt n de patrons x_p amb les sortides desitjades t_p on $p = (1, \dots, n)$.

El procés d'aprenentatge consisteix en entrar un patró x_p a la xarxa, de manera que si la sortida de la xarxa t_i no coincideix amb t_p -és a dir, si la sortida donada no és la sortida desitjada- s'actualitzen els pesos de la xarxa w_{ij} seguint la regla

$$\Delta w_{ij} = \varepsilon \cdot (t_i - y_i)x_j$$

següent:

Expressió matemàtica de l'algoritme d'aprenentatge del perceptró que indica com corregir els valors dels pesos, on w_{ij} és el valor del pes de la connexió entre la neurona anomenada i i la neurona j , t_i és la sortida de la neurona i , y_i és l'entrada de la neurona i i x_j és l'entrada de la neurona j .

En la figura anterior, ε és un valor arbitrari que representa el ritme d'aprenentatge. En el disseny de l'entrenament cal considerar quin valor pot ser el més adient, tenint en compte que un valor massa baix alenteix l'aprenentatge i un valor massa gran fa que els pesos emmagatzemin la informació poc detalladament.

11.1.4.- Limitacions del perceptró simple

Com van asserir Minsky i Papert el 1969 (apartat 6.6) el perceptró té una enorme limitació. L'inconvenient més destacable és, sens dubte, la impossibilitat d'adaptar els pesos de capes posteriors en funció de l'error comès en capes anteriors. Això fa que només pugui resoldre problemes separables linealment, és a dir, només pot resoldre problemes on existeix la possibilitat de dividir tot el mapa d'entrades en un hiperplà¹³. D'aquesta manera se separen dues regions de decisió, una en què el perceptró respon, posem per cas, 0 i l'altra en la qual el perceptró respon 1.

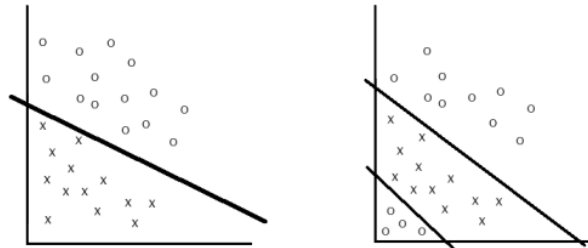


Fig. 24. A la dreta regions de decisió del pla separades per un hiperplà, a l'esquerra un problema no separable linealment i per tant irresoluble per a un perceptró simple

Així doncs, el perceptró simple pot resoldre amb èxit qualsevol problema separable linealment, però li serà impossible no fracassar en els problemes que no són d'aquesta naturalesa.

A la pràctica, quan s'intenta fer que un perceptró resolgui un problema no separable linealment, el sistema no indica directament que no pot resoldre'l, sinó que intenta trobar la solució del problema durant un temps indefinit. Aquest últim fet, doncs, seria un indicador –no teòric (com els que van trobar Minsky i Pappert), sinó experimental– de la incapacitat de resoldre problemes no separables linealment per part del perceptró.

11.2.- ADALINE

Un altre model de xarxa neuronal artificial clàssic és l'Adaline. Se n'ha parlat breument en l'apartat (6.5), però és necessari explicar-lo d'una altra manera per tal de compararlo amb el perceptró.

Va ser introduït per Widrow, un científic nord-americà de la Universitat d'Standford, l'any 1959. El seu nom prové d'ADAdaptive Linear NEuron. En l'Adaline les entrades poden ser contínues i s'utilitza un tipus de neurona similar a la del perceptró simple.

11.2.1.- LMS o regla de mínims quadrats, comparació amb l'algoritme d'aprenentatge del perceptró

Una diferència fonamental que existeix entre l'Adaline i el perceptró radica en l'algoritme d'aprenentatge; en aquest primer sistema s'utilitza la regla de Widrow-Hoff

¹³ Hiperplà: dit d'una línia recta en dues dimensions.

–també coneguda com a LMS (Least Mean Square) o regla de mínims quadrats–. Aquesta regla permet que es donin actualitzacions en els pesos en proporció als errors comesos per la xarxa.

La regla d'actualització de pesos LMS, doncs, es fonamenta en la consideració de l'aprenentatge dels pesos d'una neurona com un problema d'optimització d'una determinada funció. Aquesta funció mesurarà el rendiment actual de la xarxa i dependrà dels pesos sinàptics de la mateixa.

Cal ressaltar, a més, que mentre que la regla d'aprenentatge del perceptró simple implica actualitzacions discretes en els pesos, la regla LMS produeix en l'Adaline actualitzacions de tipus continu, de manera que un error més gran produeix una actualització més freqüent. També, degut a la diferència del tipus d'entrades entre aquests dos sistemes, mentre que el perceptró té un número finit d'iteracions (si és capaç de classificar correctament tots els patrons, és a dir, trobar una solució al problema), la regla LMS s'apropa asimptòticament a una solució, ja que en cada iteració els valors dels pesos tenen un increment més petit, de manera que el sistema s'aproparà cada vegada més al valor correcte, però mai l'assolirà.

Tant és així que s'ha anat fent ús de l'Adaline des de principis de la dècada dels seixanta com a filtre adaptatiu, per exemple en aplicacions relacionades amb la reducció del soroll en la transmissió de senyals. D'aquesta manera, i des de fa anys, milions de mòdems de tot el món inclouen o bé una Adaline o bé un successor d'aquest sistema.

11.3.- EL PERCEPTRÓ MULTICAPA O MLP (MULTI-LAYER PERCEPTRON)

Com és lògic, les xarxes més grans i més complexes ofereixen una millor qualitat en el càlcul computacional que les xarxes més simples. Una manera de solucionar els problemes ocasionats per les limitacions del perceptró simple, doncs, és a través de l'addició de capes ocultes, obtenint, així, una xarxa neuronal anomenada perceptró multicapa. És per aquest motiu que la construcció d'una xarxa amb més d'una capa és quelcom que sempre ha originat un gran interès en els experts de la computació neuronal.

En els anys en què va desenvolupar-se el perceptró, els investigadors no van ser capaços de dissenyar un algoritme capaç de propagar les correccions dels pesos a través de xarxes multicapa. Afortunadament per a la computació neuronal, però, van sorgir noves regles d'aprenentatge per a xarxes amb múltiples capes, així com també nous algoritmes d'aprenentatge, el més popular dels quals és l'anomenat "Backpropagation", gràcies a la qual es va aconseguir resoldre els problemes originats per la classificació de patrons no separables linealment.

Tot i que les configuracions de les xarxes construïdes després de la primera implementació del perceptró multicapa presenten aspectes molt diferents, en tenen un de comú: l'organització de les neurones en diferents capes o nivells, imitant així l'estructura que presenta el cervell en algunes parts.

La gran popularitat que rep l'arquitectura del perceptró multicapa amb una capa de sortida i una única capa oculta es deu al fet que aquesta permet l'aproximació de qualsevol funció contínua al nivell desitjat dins d'un cert interval, qüestió demostrada per Funahashi (1989). Aquest fet va proporcionar una base molt sòlida al camp de la computació neuronal i explica el ressorgiment de l'interès quan van aparèixer aquest tipus de xarxes.

Ara bé, el nombre de neurones necessàries de què ha de disposar la capa oculta per dur a terme l'aproximació varia segons el problema, i moltes vegades aquest fet dificulta en gran mesura la implementació del perceptró multicapa. No es pot negar, tanmateix, que passar de la impossibilitat d'aplicació que presenta, en alguns casos, el perceptró simple a la mera dificultat d'aplicació ja és un gran pas.

11.3.1.- Funcionament

El funcionament d'un perceptró multicapa és extremament similar al d'un perceptró simple. Es realitzen les mateixes operacions i aquestes es van repetint per a cada capa. Les capes intermèdies o ocultes i la capa de sortida utilitzen com a entrada les sortides de la capa anterior. D'aquesta manera, un perceptró pot disposar d'un nombre infinit de capes. Cal recordar, no obstant, que el problema per a la creació d'aquest tipus de sistemes no radica en la simple connexió entre capes que s'acaba de descriure, sinó en la manca d'algoritmes d'aprenentatge que permetin que els valors obtinguts en la resolució del problema siguin els encertats.

11.3.2.- La regla d'aprenentatge del perceptró multicapa: l'algoritme de retropropagació (Back-Propagation)

La regla d'aprenentatge del Perceptró de Rosenblatt i l'algoritme LMS de Widrow van ser dissenyats per entrenar xarxes d'una sola capa. Aquestes xarxes tenen el desavantatge que només poden resoldre problemes linealment separables, cosa que va portar al naixement de les xarxes multicapa per sobrepassar aquesta dificultat.

El primer algoritme d'entrenament per a xarxes multicapa va ser desenvolupat per Paul Werbos el 1974. Aquest va desenvolupar-lo dins d'un marc teòric general, constituint-ne, així, les xarxes neuronals artificials una aplicació específica que va passar desapercebuda. Per aquesta raó, l'algoritme no va ser acceptat de forma immediata per la comunitat de desenvolupadors de xarxes neuronals. De fet, no va ser fins a mitjans dels anys 80 que l'algoritme "Back-Propagation" o algoritme de retropropagació va redescobrir-se al mateix temps per diversos investigadors. Aquests van ser David Rumelhart, Geoffrey Hinton, Ronal Williams, David Parker i Yann Le Cun, tots ells experts en el camp de la computació neuronal.

Ara bé, l'algoritme va popularitzar-se per mèrit de Rumelhart, quan va incloure aquest concepte en la seva obra "Parallel Distributed Processing Group¹⁴". La publicació d'aquest llibre va suposar un auge per les investigacions de les xarxes neuronals,

¹⁴ Traduït al català: "Conjunt de processament distribuït en paral·lel"

convertint l'algoritme de retropropagació en un dels mètodes més utilitzats en aquest camp a partir d'aleshores. I és que, de fet, actualment se'n continua fent ús de forma habitual –els dos projectes desenvolupats en aquest treball l'utilitzen–.

Una de les grans millores que introdueix l'algoritme de retropropagació consisteix en l'aprofitament de la naturalesa de processament en paral·lel pròpia de les XNA per així reduir el temps de classificació de patrons.

La majoria dels sistemes actuals de computació tradicional s'han dissenyat per dur a terme funcions matemàtiques lògiques a una velocitat que resulta increïblement alta si es compara amb la que és característica de l'ésser humà. Ara bé, l'habilitat de càlcul no és allò que es necessita per solucionar problemes de reconeixement de patrons en entorns sorollosos, per exemple. El problema de la computació tradicional radica, principalment, en la naturalesa seqüencial que li és pròpia.

Cal definir que n entorn sorollós és un entorn on els senyals que el sistema capta per obtenir informació contenen interferències. Aquestes dificulten el procés de classificació, causant, així, conseqüències com ara un major temps de processament o una taxa més elevada d'errors. El terme “soroll” prové del cas concret del reconeixement de veu, on el que interessa és la veu del subjecte parlant i no els sorolls que puguin existir a l'entorn.

COS PRÀCTIC:

12.- ELS DIFERENTS TIPUS DE PROGRAMES

Existeix una gran quantitat de programes que serveixen per treballar amb xarxes neuronals artificials. Principalment, pretenen oferir dos grans serveis:

- Proporcionar a l'usuari la base en què s'establirà la xarxa, és a dir, un llenguatge de programació i una estructura determinada.
- Proporcionar a l'usuari una sèrie d'eines que es relacionaran directament amb el disseny de la xarxa neuronal, intervenint, així, en qualsevol aspecte d'aquesta. Alguns exemples d'aquests aspectes són el tipus de xarxa, l'arquitectura de la xarxa, l'algoritme d'aprenentatge, les dades a analitzar, etc.

Tanmateix, els diferents programes relacionats amb la computació poden disposar o no d'accés gratuït, havent-hi, així, avantatges i desavantatges en els dos casos.

El principal avantatge d'utilitzar un programa d'ús gratuït és que el seu programari acostuma a ser de tipus lliure -de manera que l'usuari pot modificar-ne el codi si aquest té nocions avançades de programació-. Aquests programes es caracteritzen, a més, per la seva fàcil obtenció a través d'internet. Malgrat tot, i aquí és on rau el principal desavantatge, les eines que proporcionen són més limitades i ofereixen menys possibilitats pel que fa a la programació.

Exemples: Pythia, Python i Teruel.

Si, pel contrari, els programes no són gratuïts, és imprescindible el pagament d'una llicència. No obstant, és cert que solen ser programes més complets i com a conseqüència, més versàtils, característica imprescindible per assolir l'objectiu proposat en aquest treball (com es veurà més endavant).

Exemples: Matlab i Sensivity Neural Network (SNN).

13.- MATLAB:

Donada la decisió d'utilitzar el Matlab, considero oportú exposar les raons d'aquesta tria. També serà necessari explicar què és exactament el Matlab, quin és el seu funcionament bàsic i quins seran els dos subprogrames que intervindran en el desenvolupament dels dos projectes assolits (sent aquests programes el "Neural Network Toolbox i el "GUI")

13.1.- PER QUÈ EL MATLAB?

En un principi cal saber que, a diferència d'altres programes, el Matlab no treballa únicament amb xarxes neuronals artificials. Així doncs, es podria arribar a la conclusió que seria millor escollir un programa enfocat plenament a aquest tipus de sistemes.

Ara bé, una conclusió precipitada no sempre és la més encertada, ja que precisament és la versatilitat que ofereix aquest programa la que ha decantat la balança a favor del Matlab.

Resulta que es caracteritza per ser el que es denomina un “macroprograma”, és a dir, un programa que engloba altres subprogrames. Un d'aquests subprogrames (“Neural Network Toolbox”) és el que permet desenvolupar una xarxa neuronal. Combinant-lo, doncs, amb el subprograma “GUI”, apareix la possibilitat de generar una interfície gràfica que permet relacionar l'usuari amb la xarxa d'una manera directa.

Així doncs, per triar el programa adequat, s'ha de tenir en compte allò que es vol fer amb aquest. El subprograma que permet generar una interfície gràfica resulta ser la millor eina per a una interacció usuari-xarxa (com en el cas, per exemple, del problema de reconeixement de caràcters). Això és degut al fet que aquesta interfície és la que farà finalment d'intermediari entre l'usuari que dibuixa el caràcter – proporcionant, així, un exemple nou– i la xarxa neuronal. D'aquesta manera, sabent que per dur a terme una xarxa neuronal interactiva la interfície gràfica és un element totalment imprescindible, el programa Matlab sembla una molt bona opció.

Ara bé, tot i que la combinació entre xarxa neuronal i interfície gràfica constitueix l'argument de més pes a l'hora de triar el Matlab, també es van tenir en compte altres factors en la presa d'aquesta decisió. D'aquests factors secundaris, el principal n'és la certesa que es tracta d'un programa de molt bona reputació, que, a més, gaudeix de nombroses recomanacions per part dels usuaris que ja l'han provat. Addicionalment, hi ha una gran quantitat d'informació a internet que descriu el seu funcionament, fet que em va acabar de convèncer.

Com a argument de contrapartida hi ha, desafortunadament, el preu. La llicència de Matlab per estudiants té un cost actual de 35 euros. Tot i que no era el preu que m'imaginava, un cop definits els avantatges que presenta aquest programa vaig considerar que es tractava d'una xifra prou raonable. Així doncs, vaig adquirir el producte, i no va resultar gens decebedor

13.2.- QUÈ ÉS EL MATLAB?

El programa Matlab¹⁵ constitueix un entorn de computació orientat al desenvolupament d'aplicacions que permet dur a terme projectes on es necessiten, per una banda, complexos càlculs matemàtics i, per l'altra, la visualització gràfica d'aquests. Així doncs, enfocades sobretot en un àmbit matemàtic i un entorn de programació, les funcions del Matlab es limiten a l'anàlisi numèric, el càlcul matricial i el processat de senyal i visualització gràfica.

El programa neix com a solució a la necessitat d'eines de càlcul més potents. Aquestes permeten la resolució de problemes de càlcul complexos en els que s'han de calcular estrictament grans quantitats de dades. Va ser desenvolupat a finals del 1970 per tal de facilitar els projectes de software matricial LINPACK i EISPACK, que eren

¹⁵ El nom “Matlab” prové de “matrix laboratory” (laboratori matricial).

els més punters de l'època. Aviat es va expandir a l'ambient universitari, i el 1984 es va fundar la companyia MathWorks per així iniciar-ne la comercialització.

Un dels punts forts del Matlab és la seva arquitectura oberta. Això significa que permet l'acoblament d'una gran varietat de subprogrames de suport especialitzats, els anomenats "toolboxes" o, traduït literalment, "caixes d'eines". Aquests augmenten significativament el número de funcions pròpies del programa principal i cobreixen gairebé totes les àrees principals del món de la enginyeria i la simulació, com ara el processat d'imatges i senyals, les matemàtiques simbòliques, la lògica difusa, la identificació de sistemes o les xarxes neuronals.

Com que es tracta d'un programa complet i eficient, se'n recomana el seu ús en la comprensió de coneixements relacionats amb tota mena de tecnologies, i és per aquest motiu que és utilitzat de forma habitual en les carreres de tipus tecnològic. No obstant, també té aplicacions en el disseny de sistemes de control, l'enginyeria financera i la investigació mèdica.

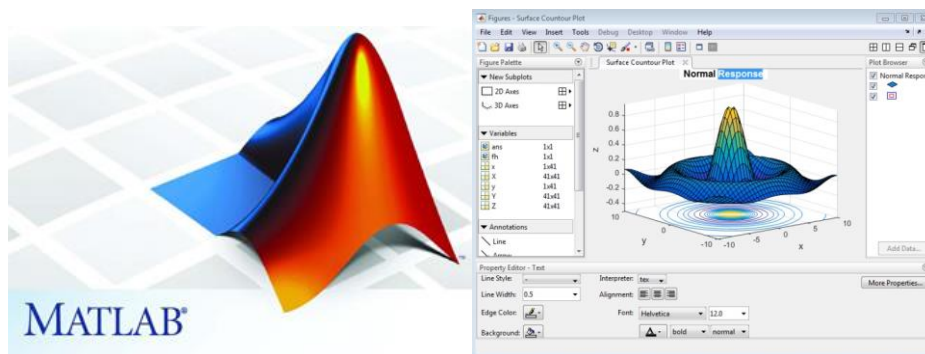


Fig. 25. Icona característica del Matlab i una imatge que permet observar com es visualitza, d'una manera aproximada, el programa.

13.3.- FUNCIONAMENT BÀSIC

El símbol `>>` a la "finestra d'ordres" indica l'inici d'una afirmació escrita en el llenguatge de programació de Matlab. Aquestes afirmacions poden ser expressions matemàtiques com per exemple la que es presenta a continuació:

```
>> 1 + 1
```

la qual, poc sorprenentment, serà resposta amb un:

```
Ans = 2
```

Traduït al nostre llenguatge, el programa indica que l'operació $1 + 1$ dóna com a resultat el valor 2.

També es poden definir variables o funcions, com ara la que segueix:

```
>> A (0,7) = cos (0,7)
```

```
A= 0,7648
```

En aquest cas, el programa proporciona la resposta corresponent al valor del cosinus d'un nombre determinat (a l'exemple és 0,7) en funció de la variable A.

13.4.- CAIXA D'EINES DE XARXES NEURONALS (NEURAL NETWORK TOOLBOX)

Aquest "toolbox" permet el disseny, inicialització, simulació i entrenament dels models neuronals d'ús més estès en l'actualitat. Alguns exemples en són el perceptró, les xarxes lineals, les xarxes de base radial... També proporciona diferents opcions a l'hora de triar el tipus d'aprenentatge, incorporant, així, l'aprenentatge competitiu i l'aprenentatge cooperatiu entre d'altres.

Mitjançant un ampli ventall de funcions i mètodes predeterminats, l'usuari pot efectuar el disseny d'arquitectures complexes. No obstant, això no suposa cap mena d'obstacle per a l'usuari a l'hora de definir les seves pròpies funcions i els paràmetres típics de les xarxes neuronals, com ara el tipus de xarxa, el nombre de neurones presents a la capa oculta o l'algoritme d'aprenentatge.

El "toolbox" es complementa amb les facilitats i prestacions gràfiques pròpies del Matlab, amb l'objectiu d'estudiar el comportament de les xarxes. Aquests gràfics resulten molt útils a l'hora d'estudiar l'eficàcia dels diferents algoritmes d'aprenentatge, ja que representen fidelment el conjunt d'errors que poden produir-se al llarg de l'entrenament.

Adicionalment, inclou un manual d'instruccions referent al camp de les xarxes neuronals, juntament amb una col·lecció de demostracions i aplicacions molt didàctiques, útils per a l'estudi i l'aprofundiment en les qüestions fonamentals referents a les xarxes neuronals.

Els usuaris poc familiaritzats amb la programació poden fer ús d'un assistent de desenvolupament de xarxes neuronals, que permet dissenyar una xarxa dins d'un entorn limitat. En altres paraules, es proporcionen diferents paràmetres propis de les xarxes neuronals que l'usuari pot anar modificant fins que obtingui una xarxa personalitzada. D'aquesta manera, es diferencien dos mètodes per desenvolupar xarxes neuronals: el mètode convencional –en el que són necessaris uns coneixements bàsics de programació– i el mètode "nnstart"¹⁶ –en el que es fa ús de l'assistent de xarxes neuronals ja descrit-. Els projectes d'aquest treball permetran observar els dos mètodes.

Aquest subprograma, tanmateix, no s'orienta als problemes on la terminologia, fonaments matemàtics i procediments de disseny estan ja fermament establerts i s'han aplicat durant anys. Pel contrari, es pretén que sigui utilitzat per a la valoració i disseny de diferents xarxes neuronals aplicades en la indústria i sobretot en educació i investigació.

Dins de les aplicacions bàsiques del "toolbox", cal destacar aquelles que s'enfoquen cap als següents àmbits:

¹⁶ "nnstart" és l'ordre que s'ha d'introduir al Matlab per tal d'iniciar l'assistent de xarxes neuronals.

- **Administració de dades:** reconeixement de patrons, processament de senyals, identificació d'imatges, extracció de característiques i compressió de dades.
- **Electrònica:** control de processos, anàlisi d'errors, modelatge no lineal, síntesi de veu i visió per ordinador.
- **Economia:** anàlisi financer i anàlisi predictiu.
- **Indústria general:** control de processos, identificació en temps real i sistemes d'inspecció.
- **Indústria aeroespacial i automoció:** simulació, sistemes de control i autopilotaje.
- **Robòtica:** control de trajectòries i sistemes de visió.
- **Telecomunicacions:** control de dades i imatges, serveis d'informació automatitzada i traducció del llenguatge parlat en temps real.

13.5.- GUI

13.5.1.- Descripció

La interfície gràfica d'usuari del Matlab (GUI) permet desenvolupar finestres, gràfics, botons, menús, etc. que tenen diferents funcions i permeten visualitzar petits programes desenvolupats en aquest entorn.

L'elaboració de finestres gràfiques es pot dur a terme de dues maneres:

- La primera -utilitzada en aquest treball- és la implementació d'un seguit d'ordres de programació (script) que defineixi els diferents elements i la funció d'aquests.
- La segona és l'ús d'una eina de disseny de GUIs que proveeix Matlab anomenada GUIDE.

Una finestra gràfica consta de diversos elements. Aquests són el menú d'interfície, els diferents controls que ens permeten executar i controlar accions en el programa i els eixos que permeten desplegar gràfiques i imatges.

El flux del programa en una GUI és diferent al de qualsevol funció que es crea en Matlab. En un script o funció definida en Matlab les ordres s'executen seguint unes determinades condicions lògiques fins que s'arriba al final del programa, podent obtenir o no un resultat final a la sortida. No obstant això, en una GUI les ordres no s'executen en una seqüència preestablerta, sinó que és l'usuari el qui controla el flux del programa interactuant amb els controls de la GUI. Aquest cicle continua fins que l'usuari decideix tancar el programa.

3.5.2.- Objectes gràfics en el GUI

Hi ha una determinada estructura jeràrquica (Fig. 26) en els objectes gràfics del Matlab. La jerarquia no s'estableix en el sentit que hi ha elements més importants que d'altres, sinó en el sentit que hi ha elements que no poden existir sense altres elements¹⁷.

- **Pantalla:** l'objecte més general en Matlab és la pantalla. Només n'hi pot haver una. En aquesta pantalla (*screen*) pot haver-hi una o diverses finestres (*figures*).
- **Finestra:** en una finestra trobem un o més eixos de coordenades (*AXES*), uns controls que ens permeten interactuar amb el programa (*uicontrols*) i els menús (*uimenu*).
- **Eixos:** finalment, els eixos poden contenir cinc objectes gràfics; línies (*line*), polígons (*patch*), superfícies (*surfaces*), imatges tipus bitmat (*image*) i text (*text*).

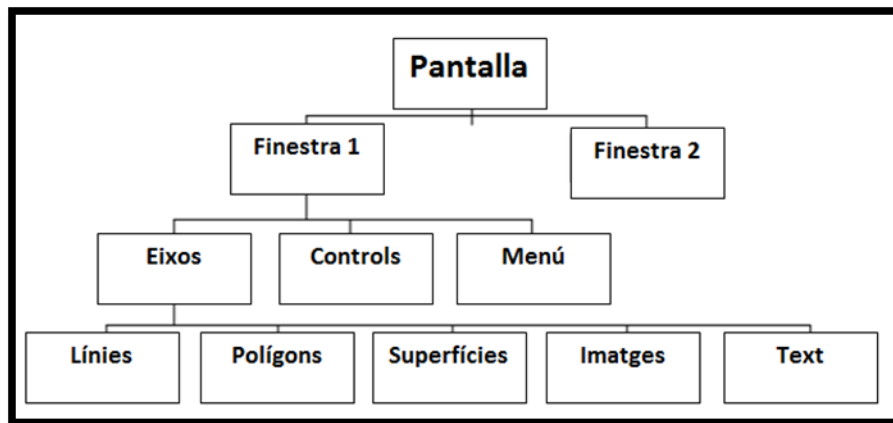


Fig. 26. Estructura jeràrquica dels objectes gràfics del GUI.

¹⁷ La denominació oficial en anglès d'aquests elements serà escrita en cursiva.

14.- PROJECTES ASSOLITS

En aquest apartat es descriuran les diverses dificultats a les quals s'ha hagut de fer front, primerament, de forma general i acte seguit, de manera específica –és a dir, dificultats concretes per als dos projectes assolits–.

Seguidament, es farà una comparació entre aquests dos projectes o problemes, establint les característiques que tenen en comú, tot fent, així, un repàs dels conceptes que s'escaiguin sobre les xarxes neuronals.

Finalment, s'explicarà com s'ha realitzat cada projecte, intentant descriure d'una manera divulgativa cada pas –tot i que a vegades és difícil no fer ús d'alguns tecnicismes- fins a arribar a l'estat actual dels projectes. També es procedirà a un anàlisi dels resultats obtinguts.

14.1.- DIFICULTATS

14.1.1.- Dificultats generals

Per raons purament pràctiques, els objectius plantejats difereixen en cert grau amb els projectes assolits. En un principi, hom pot pensar que, fent ús d'un programa que ja permet el desenvolupament de xarxes neuronals, l'únic esforç que caldrà fer consisteix fonamentalment en definir correctament els diferents paràmetres propis d'una XNA, per tal que aquesta gaudeixi del mínim percentatge d'errors.

Ara bé, la realitat no és aquesta. Resulta que cal construir l'estructura de la xarxa en si i, per a aquest propòsit, és necessària l'aplicació d'un determinat conjunt d'ordres o instruccions. Amb això vull dir, simplement, que s'ha de programar i, per tant, són imprescindibles unes nocions bàsiques de programació.

Aquest fet suposa una dificultat força destacable, tenint en compte que abans de fer aquest treball les meves experiències en programació "seriosa" eren escasses, per no dir nul·les. Fa uns anys, l'optativa d'informàtica que cursava em va donar l'oportunitat de desenvolupar un videojoc fent ús del programa "Scratch". Així doncs, quan vaig decidir el tema del treball ja vaig intuir que hauria de programar, però confiava en les meves experiències prèvies. Ara puc afirmar, amb total rotunditat, que el concepte de programació dista molt de tenir similituds amb l'"Scratch", que es tracta d'un entorn amigable, gairebé infantil, en el qual la majoria d'elements ja estan programats i la feina que té l'usuari és simplement la de lligar-los o relacionar-los. Consegüentment, quan vaig tenir el Matlab al davant és quan vaig adonar-me que no sabia programar.

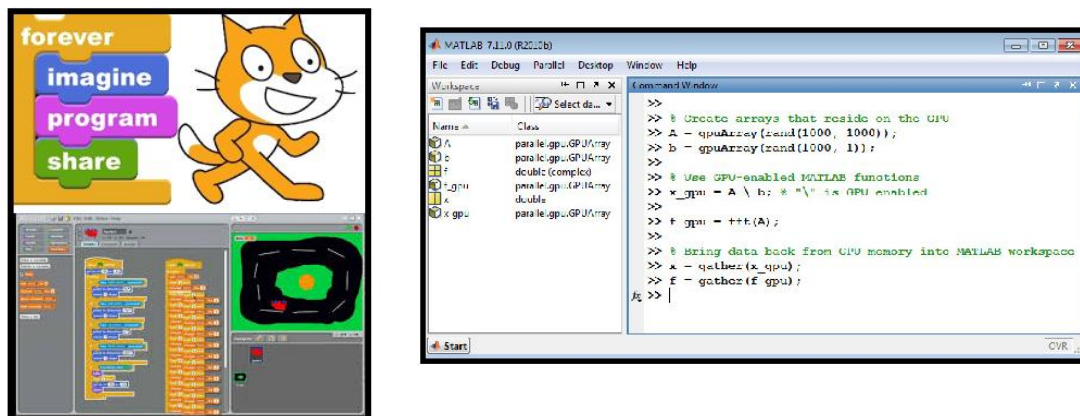


Fig. 27. Comparació entre el programa Scratch (esquerra) i el programa Matlab (dreta).

Així, vaig decidir que reduiria el nombre de xarxes neuronals desenvolupades a dues, ja que cada projecte suposa molta dedicació. Un petit parèntesi; el fet que siguin dues xarxes no es deu únicament a la raó que existeixin dos mètodes per a desenvolupar-les, però vaig considerar oportú ensenyar-los tots dos. Tornant al tema que ens ocupa, els problemes que em cridaven més l'atenció eren els del reconeixement de teixits tumorals i reconeixement de caràcters, així que em vaig concentrar en aquestes dues opcions.

El meu tutor, Joan Mercader, quan li vaig exposar les dificultats que plantejaven els meus objectius, em va recomanar la posada en contacte amb el Dr. Jordi Solé. Es tracta d'una persona fortament relacionada amb la Universitat de Vic, líder de diversos grups de recerca enfocats principalment a l'estudi del processat de senyal (DSP) i a la recerca experimental científica i tecnològica (UScience Tech). Vam creure - encertadament, tot sigui dit- que la seva experiència tant en les xarxes neuronals biològiques com en les xarxes neuronals artificials ens podria servir d'ajuda. Per últim, però no per això menys rellevant, cal mencionar que el Dr. Solé ¹⁸disposa de coneixements avançats sobre el Matlab, els quals, sens dubte, han resultat d'una immensa utilitat.

14.1.2.- Dificultats específiques de cada problema

14.1.2.1.- Reconeixement de caràcters

En un principi pretenia fer un programa relacionat amb el reconeixement dels caràcters propis de l'abecedari, és a dir, les lletres. Ara bé, a mesura que anava trobant informació sobre les xarxes neuronals, em vaig adonar que hi havia una sortida semblant, però molt més viable. Aquesta era el reconeixement de dígit. La raó per triar aquesta opció no és gens complexa: mentre que en el primer cas existeixen 27 possibilitats o lletres, en el segon cas només hi ha 10 possibilitats o dígit. I, de fet, que existeixin poques possibilitats és un factor que determina el correcte funcionament

¹⁸ Per a una biografia més acurada caldria consultar https://www.researchgate.net/profile/Jordi_Sole-Casals

de la xarxa, disminuint potencialment l'error, ja que com menys opcions tingui la xarxa per fallar, menys fallarà.

També vaig tenir en compte una opció menys elegant, que segurament hauria triat en l'hipotètic cas que només gaudís de la possibilitat de fer la xarxa sobre el reconeixement de lletres. Aquesta opció consisteix, ni més ni menys, en el reconeixement per part de la xarxa d'un determinat conjunt de lletres –posem per cas, les primeres 10 lletres de l'abecedari–. Fent-ho així, el procediment esdevé igual de viable que el dels dígit i, fins i tot, es podria fer encara més viable reduint el nombre de lletres a reconèixer. No obstant, la meua intenció (i il·lusió) des d'un principi era aconseguir que el projecte que desenvolupés tingués una mínima utilitat en la vida real, i no puc evitar posar en dubte que existeixi algú que no doni per inútil una xarxa neuronal que només reconeix les primeres 10 lletres de l'abecedari. Així doncs, pels arguments donats, vaig decidir elaborar una xarxa enfocada al reconeixement de dígit, opció que em convenia molt més.

14.1.2.2.- Reconeixement de teixits tumorals

Com s'ha exposat anteriorment, una de les dificultats principals a l'hora de dissenyar una xarxa neuronal amb el Matlab és la programació. S'han de disposar d'uns coneixements previs que no tothom té i, per a un problema tan complex com el reconeixement de teixits tumorals, la dificultat augmenta encara més. És per aquest motiu que vaig decidir fer servir l'assistent de desenvolupament de xarxes neuronals proporcionat pel "Neural Network Toolbox" o, el que és el mateix, el mètode "nnstart".

D'entrada cal mencionar que el mètode "nnstart" té certes limitacions. Moltes d'aquestes restriccions estan relacionades amb els diferents paràmetres de les xarxes neuronals (només se'n poden utilitzar alguns), però la més molesta és aquella que impossibilita la connexió de la xarxa neuronal amb una interfície gràfica (GUI). Això genera un problema; tot i que la xarxa serà completament operativa, no podrà ser interactiva.

Conseqüentment, l'usuari no podrà subministrar exemples nous a la xarxa amb la finalitat que aquesta els classifiqui de la mateixa manera que en el cas anterior, que és, definitivament, un mètode molt més eficient i ràpid. Per tal de donar exemples nous a la xarxa, doncs, s'haurà de seguir un procediment relativament pesat, que serà descrit a l'apartat 14.4.4.

14.2.- CARACTERÍSTIQUES COMUNES ENTRE ELS DOS PROBLEMES

Els tres aspectes descrits a continuació estan presents tant en el problema de reconeixement de dígit com en el problema de reconeixement de teixits tumorals. Cal assenyalar que les dues primeres característiques exposen de manera resumida conceptes ja tractats, els quals són força rellevants. És per això que és necessari posar-hi un cert èmfasi.

1. Les dues xarxes són de tipus perceptró multicapa i solucionen problemes basats en el reconeixement de patrons:

Cal recordar que el perceptró és aquell tipus de xarxa caracteritzada per gaudir d'una especialització exclusiva en el reconeixement de patrons.

En els problemes de reconeixement de patrons, l'objectiu de la xarxa neuronal és classificar una sèrie d'entrades en diferents categories, donant, així, una determinada sortida. Un cop s'ha proporcionat a la xarxa tot un conjunt d'exemples i s'ha especificat el resultat esperat, es procedeix a l'entrenament. Durant aquesta fase, els pesos de les neurones, que comencen tenint uns valors aleatoris, varien en funció del resultat desitjat. Finalment, la xarxa és capaç de classificar patrons que no ha "vist" anteriorment, és a dir, classificar informació nova.

2. Les dues xarxes utilitzen l'algoritme de retropropagació (Back-Propagation) per al seu aprenentatge:

Aquest algoritme permet el correcte funcionament dels perceptrons amb més d'una capa (perceptró multicapa), i es caracteritza tant per permetre que la xarxa resolgui problemes més enllà dels que són linealment separables com per optimitzar la velocitat en què es classifiquen els patrons (aprofitant el processament en paral·lel propi de les xarxes neuronals).

3. Les dues xarxes poden ser analitzades a partir d'una matriu de confusió:

Aquesta es tracta d'una eina molt útil per visualitzar gràficament el rendiment i el percentatge d'errors comès per la xarxa.

14.3.- RECONeixEMENT DE DÍGITS

14.3.1.- Descripció

L'objectiu d'aquesta xarxa neuronal consisteix en reconèixer correctament –és a dir, amb un error inferior al 5%–, i a partir d'imatges, els següents dígit: 0,1,2,3,4,5,6,7,8,9. Addicionalment, la xarxa ha de ser capaç d'admetre i classificar exemples nous proporcionats per l'usuari, mitjançant una interfície gràfica generada a través del subprograma GUI.

Els exemples que s'utilitzaran per entrenar la xarxa provenen d'una enorme base de dades, la USPS. La tria d'aquesta base de dades es deu principalment a les característiques del seu format, totalment compatible amb el Matlab. Conté 1100 exemples de cada dígit, estant aquests en una escala de colors grisos de 8 bits (per facilitar el procés de binarització blanc-negre). Les imatges presentades a continuació corresponen als exemples de tres dígit seleccionats aleatòriament: el 0, el 5 i el 9.

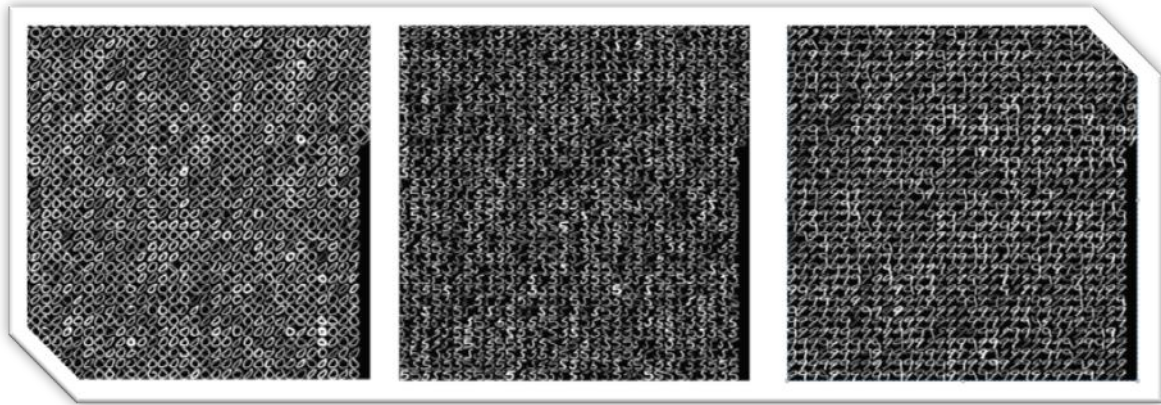


Fig. 28. Exemples dels dígitos 0, 5 i 9 respectivament

El mètode utilitzat per a la resolució d'aquest problema és el de la programació tradicional, i és el mètode que se sol fer servir normalment, ja que dóna un ampli grau de llibertat a l'usuari. Per aquesta raó, el resultat del procés és un "script" –és a dir, un text format per unes instruccions donades per l'usuari–, el qual una vegada és executat, s'inicia la xarxa neuronal.

Es tracta, doncs, d'un fitxer que té la finalitat de manipular matrius per aconseguir els objectius exposats prèviament. Aquestes matrius corresponen al valor numèric d'unes imatges que serviran tant d'exemples per entrenar la xarxa com d'imatges per testar el funcionament de la xarxa i observar-ne el percentatge d'errors.

A banda de llegir les dades ja disponibles, però, cal recordar que també ha d'existir la possibilitat d'introduir dades noves a partir d'una interfície gràfica mentre l'usuari fa ús de la pantalla i el ratolí.

Així doncs, tot el contingut de la xarxa es redueix a l'script en si, i aquest motiu pel qual es considera oportú designar els passos necessaris per a la construcció d'un script propi al problema del reconeixement de dígitos (*Veure Annex I: Script complet de la "Xarxa neuronal de reconeixement de dígitos"*).

14.3.2.- Script

En primer lloc, cal mencionar que el document digital que posseeix tota la informació que és donada d'entrada a la xarxa és el document anomenat 'usps_all' –usps fa referència a la procedència de la base de dades–. Les dades en si, però, estan a la variable 'data', que té dimensions 256x1100x10.

Tenint en compte que la imatge de cada exemple està composta per una quadrícula de 16x16 franges –o si es prefereix, píxels–, s'entén que la primera dimensió de la variable fa referència a la matriu de dades 16x16 vectoritzada, és a dir, totes les petites franges posades en fila (cal considerar que $16 \times 16 = 256$). Procedint en l'explicació, doncs, la segona dimensió constitueix el número d'exemples que són presents de cada dígit, és a dir, 1100 (els 11000 exemples totals dividits entre els 10 dígitos). Per últim, la tercera dimensió fa referència als diferents tipus de dígitos que hi ha en total, és a dir, 10. Els diferents tipus de dígitos, a partir d'ara, seran entesos com a "etiquetes", de manera que hi haurà en total 10 etiquetes.

Un cop fet aquest aclariment, ja es pot continuar amb la descripció dels passos que s'han seguit al llarg del desenvolupament de la XNA. Aquests es poden classificar en quatre blocs amb l'objectiu de diferenciar els diferents processos que es donen.

14.3.2.1.- Bloc 1

El primer que cal fer és carregar la base de dades (figura 30) i traduir les imatges que serviran com a exemples a un llenguatge que el Matlab pugui processar. Aquestes, com s'ha dit anteriorment, estan formades per una quadrícula de 16x16 que cal vectoritzar per obtenir les 256 franges. No obstant, per donar a l'usuari una idea de com són els exemples, és convenient fer ús de la instrucció que permet desvectoritzar la informació, com es pot veure a la figura 29.

```
load('usps_all');
```

Fig. 30. Càrrega de dades.

```
subplot(2,5,i)  
imagesc(reshape(squeeze(data(:,1,i)),16,16))
```

Fig. 29. Desvectorització de la informació (reshape and squeeze data).

Per fer-ho, tanmateix, s'han de definir les variables. Com es pot observar a la figura (..), s'imposa que la variable 'i' correspongui a un dígit determinat, de l'1 al 10. Ara bé, el que es pretén és que els dígit vagin del 0 al 9, però per raons que no s'esmentaran, el Matlab no pot acceptar el 0 com a possible dígit. El cas és que s'haurà d'indicar, d'alguna manera, que el 10 es correspon en realitat amb el 0.

```
figure  
for i = 1:10  
    if i ~=10  
        titol = ['Imatge del dígit ', num2str(i)];  
    else  
        titol = ['Imatge del dígit 0'];  
    end  
    title(titol)  
end  
colormap gray  
disp('Bloc 1...')  
pause
```

Fig. 31. Definició de variables i indicació que la posició 10 es correspon amb el dígit 0.

Executant el Bloc 1, doncs, s'obté com a resultat la càrrega de dades i l'aparició de la següent imatge. En aquesta es poden visualitzar els primers exemples de cada nombre, estant aquests en una escala de grisos (no és únicament blanc i negre). A sobre de cada dígit, a més, s'hi pot trobar la frase "imatge del dígit ...", però desafortunadament les diferents frases es superposen per la falta d'espai i no acaba de ser estètic.

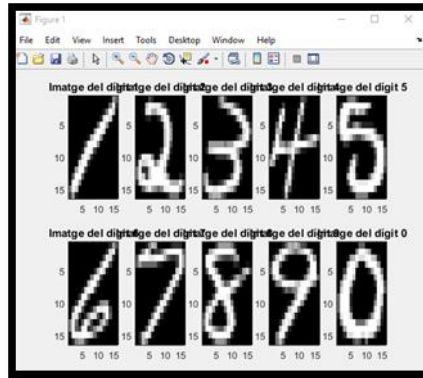


Fig. 32. Els 10 primers exemples de cada dígit.

14.3.2.2.- Bloc 2

Es procedeix a reorganitzar les dades posant cada exemple en forma de vector o de fila (vectorització). D'aquesta manera, el Matlab podrà processar la informació que que li és proporcionada.

Es creen les etiquetes de cada classe (ylabel), tot determinant la tercera dimensió mencionada a l'apartat anterior. Concretament, es fa un vector de 1100 exemples per a cada etiqueta havent-hi, així, un total de 11000 exemples (en total hi ha 10 etiquetes).

```
X = double(reshape(data,256,11000)');
ylabel = [1:9 0];
Y = reshape(repmat(ylabel,1100,1),11000,1);
disp('Bloc 2...')
pause
```

Fig. 33. Creació de les etiquetes i vectorització.

14.3.2.3.- Bloc 3

Seguidament, per tal de garantir el correcte funcionament de la xarxa neuronal, és necessari binaritzar els diferents colors de la imatge, la qual, cal recordar, està en una escala de grisos. D'aquesta manera, només seran possibles dos valors per a cadascuna de les 256 franges (Figura 34): o bé blanc (valor 1) o bé negre (valor 0).

```
X_BN = double(im2bw(X/max(max(X))));
```

Fig. 34. Binarització; aplicació del valor màxim pel color blanc o pel color negre.

Aquest fet és força rellevant per a la xarxa. I així ha de ser, ja que la informació provinent dels exemples serà entesa en clau d'uns i zeros, i no en clau dels matisos propis de les imatges (com entendria el cervell humà). Així doncs, aquest procés converteix la informació provinent d'exemples reals en informació computable, fet que el dota d'una suma importància. Tanmateix, per tal d'executar-lo, resulta necessari definir abans tota una sèrie de paràmetres que es relacionaran posteriorment amb el desenvolupament del que és la xarxa neuronal en si.

```
% definició de paràmetres:
num_class = 10;      % número de classes
num_exemp = 1100;   % número d'exemples per classe
dimX = 16;          % dimensions en x
dimY = 16;          % dimensions en y
```

Fig. 35. Definició de paràmetres per tal d'executar el procés de binarització.

Finalment, es pot recórrer a un procés força similar a aquell que és descrit al Bloc 1, que proporciona a l'usuari la possibilitat de veure el primer exemple de cada dígit, ara binaritzat. Altra vegada, és necessari definir que la desena posició equivalgui al dígit 0.

```
% visualitzem el primer exemple de cada classe, ara amb X_BN
figure
for i = 1:10
    subplot(2,5,i)
    imagesc(reshape((X_BN(1+num_exemp*(i-1),:)),16,16))
    if i ~=10
        titol = ['Imatge del dígit ', num2str(i)];
    else
        titol = ['Imatge del dígit 0'];
    end
    title(titol)
end

disp('Bloc 3...')
pause
```

Fig. 36. Definició de variables i indicació que la posició 10 es correspon amb el dígit 0.

Executant el Bloc 3, doncs, s'obté com a resultat la visualització del primer exemple referent als diferents dígits, únicament amb els colors blanc i negre. És oportú comparar la imatge obtinguda al Bloc 3 amb la del Bloc 1, per tal d'observar-ne el canvi resultant (figura 37).

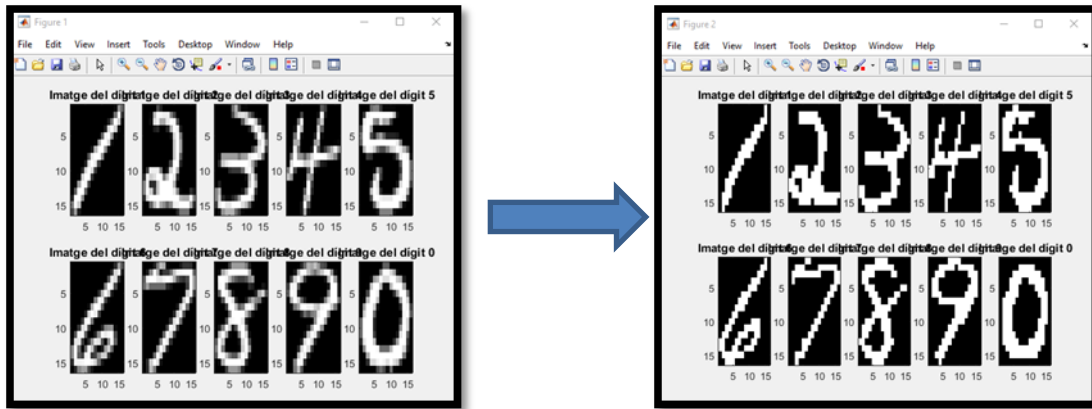


Fig. 37. Representació gràfica del procés de binarització

14.3.2.4- Bloc 4

El Bloc 4 possiblement és aquell que gaudeix de més rellevància, ja que permetrà el desenvolupament de la xarxa neuronal artificial en si mateixa.

Malgrat tot, per tal de d'assolir aquest objectiu, es requereix l'ús de recursos addicionals a la pròpia programació que ofereix el Matlab. Concretament, aquests recursos constitueixen arxius descarregables que o bé defineixen variables que desenvolupen un paper clau en el funcionament de la xarxa o bé proporcionen eines per analitzar gràficament els resultats del problema.

Un cop fet aquest aclariment, resulta interessant remarcar l'organització de les dades de la xarxa. Es va decidir, per qüestions d'eficiència en la programació, que la meitat dels exemples extrets de la base de dades gaudirien d'un paper essencial en l'entrenament de la xarxa neuronal, mentre que l'altra meitat complirien la funció de testar la xarxa.

En primer lloc és necessari definir alguns paràmetres. El '.5' fa referència al tant per u d'exemples que s'utilitzen per entrenar la xarxa. En aquest cas particular, s'ha agafat la meitat d'exemples respecte del total, de manera que el tant per u és de 0,5 i l'equivalència en tant per cent és de 50 ($0,5 \cdot 100 = 50\%$).

```
cv = cvpartition(Y, 'holdout', .5);

Xtrain = X_BN(cv.training, :);
Ytrain = Y(cv.training, 1);
Xtest = X_BN(cv.test, :);
Ytest = Y(cv.test, 1);

net = mlp(dimX*dimY, 100, 10, 'softmax');
```

Fig. 38. Definició de paràmetres per tal d'executar la xarxa neuronal.

A continuació es procedeix a la posada en funcionament de la xarxa. Per a aquest propòsit cal definir, en primer lloc, la partició dels exemples que seran utilitzats per testar la xarxa i en segon lloc, el valor numèric de la sortida.

Per tal d'explicar el valor numèric de la sortida s'ha de recórrer al concepte dels pesos sinàptics. Ara bé, potser seria convenient explicar-ho amb altres paraules per situar-lo en aquest cas concret.

De forma simplificada, podria dir-se que per a cadascun del 10 dígit existeixen 10 etiquetes. Si la xarxa creu amb seguretat que la imatge proporcionada correspon al dígit "2" –deducció que realitza gràcies al procés d'aprenentatge donat en l'entrenament–, l'etiqueta corresponent al dígit "2" tindrà un valor numèric de 1, mentre que les etiquetes pertanyents a la resta de nombres tindran un valor numèric de 0.

Ara bé, aquesta certesa tan contundent només pot donar-se en el cas que l'exemple que s'està fent servir per testar la xarxa sigui el mateix que s'ha fet servir per entrenar-la. Així, si els exemples utilitzats per provar la xarxa són diferents, que és el que interessa, la xarxa assignarà diferents valors numèrics a les etiquetes en funció del dígit que cregui que és.

Per exemple, la xarxa pot assignar un valor numèric de 0,7 a l'etiqueta del 2, de 0,2 a l'etiqueta del 5 i de 0,1 a l'etiqueta del 7. Això significa que, tot i que la xarxa afirmarà que es tracta del dígit "2" (0,7 és un valor força proper a 1), també existeix una petita probabilitat que es tracti del dígit "5" i una probabilitat encara més reduïda que es tracti del dígit "7" –tenint en compte que 0,1 constitueix un valor inferior a 0,2–.

Conseqüentment, per a cada exemple a testar, la xarxa assignarà un valor numèric superior a 0,5 a una determinada etiqueta i s'aventurarà a afirmar, mitjançant l'assignació esmentada, de quin número es tracta (a partir de la seva experiència prèvia en l'entrenament). D'aquesta manera, com més proper a 1 sigui aquest valor, més probabilitats hi haurà que la resposta proporcionada per la xarxa sigui l'encertada.

En la figura 39 es pot observar com s'emmarquen dos dels conceptes anteriors, essent el primer concepte el fet que el valor numèric màxim correspongui al valor '1' i essent el segon concepte el fet que la xarxa classifiqui els exemples de dígit en una determinada etiqueta quan aquesta té un valor superior a '0,5'.

```
YtestM = zeros(5500,10);
for t = 1:10
    YtestM(1+550*(t-1):550*t,t) = 1;
end

% Train the Network
options = foptions;
[net, options] = netopt(net, options, Xtrain, YtestM, 'scg');

% Test the Network
ytest = mlpfwd(net,Xtest);
ytestB = ytest>0.5;
[ma,in] = max(ytestB, [], 2);
Ytest(end-550:end)=10;
```

Fig. 39. Xarxa neuronal artificial, procés d'entrenament i de test.

Finalment, un cop s'ha executat aquest conjunt d'ordres –i per tant, la xarxa ja ha estat entrenada i testada–, es pot observar el percentatge d'errors que ha comès. A més a més, també es pot analitzar aquesta dada a través d'una matriu de confusió.

```

% calculem quants errors s'han comès...
err = (in-Ytest)~=0;
Confmat_NN = confusionmat(in,Ytest);

figure,
heatmap(Confmat_NN, 0:9, 0:9, 1,'Colormap','red','ShowAllTicks',1,'UseLogColorMap',true,'Colorbar',true);
title('Confusion Matrix: Single Classification Tree')
cvErr = (sum(err)/length(in))*100;
text = ['El percentatge d'encerts és del ', num2str(100-cvErr), ' %'];
disp(text)

```

Fig. 40. Càlcul dels errors comesos per la xarxa.

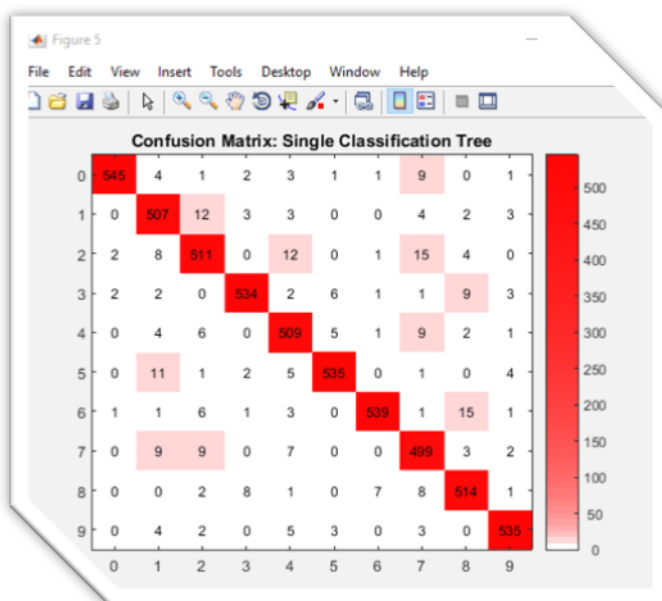
Seguint les instruccions de la figura 40, el Matlab informa del que segueix:

- La finalització del procés, que es dóna passat un nombre determinat d'iteracions.
- El percentatge d'encerts (figura 41), el qual, tot i que per poc, es troba per sota del 5%.

Warning: Maximum number of iterations has been exceeded
 El percentatge d'encerts és del 95.0545 %

Fig. 41. A dalt s'indica la finalització del procés. A baix, el percentatge d'encerts.

14.2.3.5- Anàlisi dels resultats: Matriu de confusió



Cada columna es correspon a l'etiqueta que pertany als diferents dígit (del 0 al 9, on 0 fa referència al número 10), les quals poden observar-se a l'eix d'abscisses. D'aquesta manera, la primera columna fa referència a l'etiqueta del 0, la segona, a la de l'1 i, així, successivament. A l'eix d'ordenades, en canvi, es representa el dígit que ha esbrinat la xarxa en cada exemple.

Cal recordar que hi ha 11000 exemples en total per a cadascun dels 10 dígit. Ara bé, en la matriu de confusió només es representen 5500 exemples –és a dir, 550 per cada dígit-, que són els utilitzats per testar (coincideixen amb la meitat dels 11000 exemples totals).

Per tant, si es posa per cas l'etiqueta del 0 i s'observa la columna que li correspon (la primera), es veurà clarament que la suma en vertical dels números de la columna és de 550. No és cap coincidència, són les 550 imatges que tenien el dígit 0 dibuixat. En aquest cas concret, la xarxa s'ha equivocat dues vegades confonent el 0 per un 2, dues vegades confonent-lo per un 3 i una vegada confonent-lo per un 6. Així doncs, en aquest cas la xarxa no ha donat el resultat desitjat en 5 de les 550 imatges proposades, de manera que és possible asserir que el percentatge d'encerts pel que fa al dígit 0 és força elevat (d'un 99% aproximadament). Es podria aprofundir encara més en aquest resultat, de fet, aventurant la conclusió que el 0 té una forma molt característica, i per tant, és fàcil per a la xarxa reconèixer-lo, cosa que justifica l'elevat percentatge d'encerts que, tanmateix, no s'observa, per exemple, en dígitos com l'1, el 7 o el 8.

Així doncs, la matriu de confusió és una eina molt útil per determinar el rendiment d'una xarxa, ja que mostra d'una manera directa i ràpida el seu comportament. Si, per exemple, no es gaudeix del temps suficient per analitzar detingudament els resultats, hom sempre pot fiar-se del fet que si la diagonal de la matriu és roja, la xarxa tindrà, com a conseqüència, un bon rendiment.

14.3.2.6- La interfície gràfica (GUI)

Aquest bloc constitueix el punt àlgid del projecte de reconeixement de dígit, ja que és on realment es posa a prova el sistema. Primerament cal idear, mitjançant el subprograma GUI, una interfície gràfica que permeti a l'usuari dibuixar un dígit.

Per a aquests propòsits, cal que la imatge proporcionada tingui les mateixes dimensions 16x16 que les dels exemples amb els quals s'ha entrenat la xarxa. Així doncs, s'estableix d'entrada una quadrícula on totes les franges són negres –és a dir, tenen valor 0 (Veure bloc 3, on és descrita la binarització)–. Seguidament, s'ha d'indicar al programa el fet que, quan l'usuari cliqui amb el ratolí una determinada franja, se li sumarà el valor 1, és a dir, canviarà el color de negre a blanc (de 0 a 1). Així doncs, a mesura que l'usuari procedeix en el clic de les diferents franges, pot convertir-les al color blanc per acabar dibuixant, així, el dígit en qüestió.

Per pura comoditat, també s'estableix la instrucció que, si una franja té valor 1 i aquesta és clicada, es restableixi al valor 0. D'aquesta manera, l'usuari pot rectificar els seus moviments i convertir una franja blanca altra vegada en una franja negra.

```
b = zeros(16,16);
figure
imagesc(b)

t=1;
while t
    [x,y] = ginput(1);
    cx=round(x); if cx==0, cx=cx+1; end
    cy=round(y); if cy==0, cy=cy+1; end
    b(cy,cx)=not(b(cy,cx));
    imagesc(b)
    colormap gray
    t = not(waitforbuttonpress);
end
```

Fig. 42. Interfície gràfica que permet dibuixar dígitos.

Una vegada la interfície gràfica ha estat elaborada, cal relacionar-la amb la xarxa neuronal. A la figura 43 es pot observar com el procés de test de la xarxa és similar a l'utilitzat al bloc 4.

```
valors = [1:9,0];

% Test the Network
ytest_b = mlpfwd(net,b(:)');
ytestB_b = ytest_b>0.5;
[ma_b,in_b] = max(ytestB_b,[],2);
text = ['El número dibuixat és el ', num2str(valors(in_b))];
disp(text)
```

Fig. 43. Procés que relaciona la interfície gràfica amb la xarxa, testant aquesta última a través dels exemples obtinguts amb l'interfície.

Finalment, només falta provar si la xarxa neuronal funciona de forma correcta. Per tal de determinar la bondat del seu funcionament, es dibuixarà el dígit que ha tingut més èxit en la fase de test anterior, el dígit 0, i acte seguit, el Matlab informará sobre la classificació que ha fet la xarxa neuronal (Figura 44).

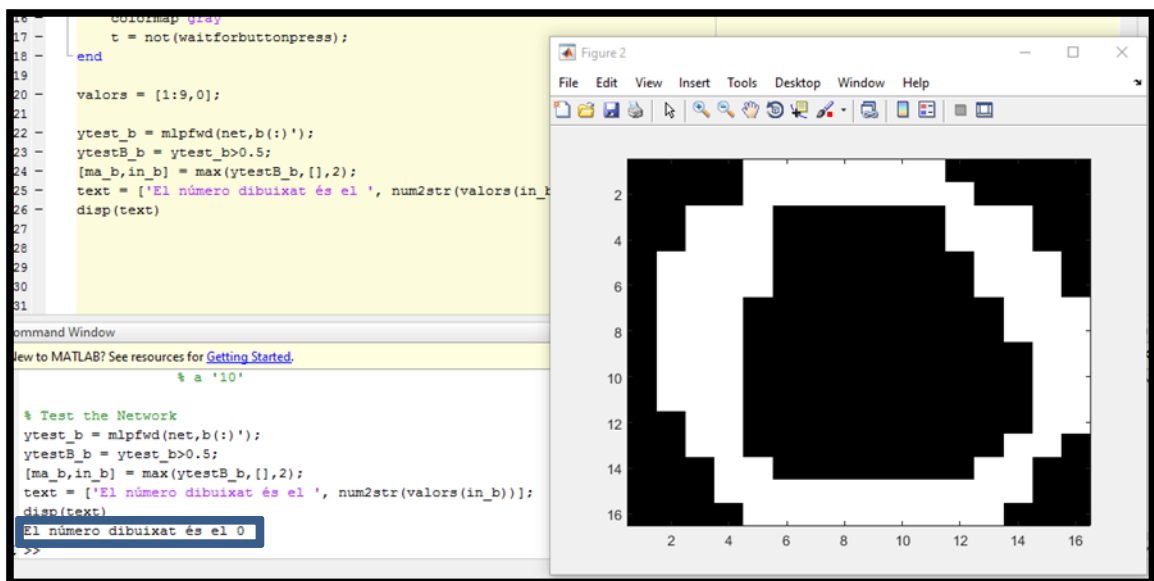


Fig. 44. Comprovació del correcte funcionament de la xarxa. Remarcat en blau el Matlab informa "el número dibuixat és el 0".

Com es pot observar, la xarxa funciona correctament ja que el requadre blau indica que el número dibuixat és el 0, i és tracta de la única resposta vàlida. Cal esmentar que el percentatge d'encerts en el cas del 0 és d'un 99% aproximadament, de manera que, en base a aquestes dades, és possible deduir que en 1 de cada 100 exemples dibuixats, l'afirmació de la xarxa no es correspondrà amb la realitat.

14.4.- RECONeixEMENT DE TEIXITS TUMORALS (CÀNCER DE PULMÓ)

14.4.1- Descripció

L'objectiu d'aquest projecte és dissenyar una xarxa neuronal artificial que sigui capaç de classificar diferents teixits cancerosos, indicant si aquests presenten un tumor benigne o maligne –és a dir, indicant si el pacient presenta la malaltia del càncer o no-.

La metodologia seguida en aquest cas consisteix en triar un conjunt reduït d'atributs (concretament, 9), que puguin utilitzar-se com a indicadors fiables que el tipus de tumor que presenta un teixit determinat és maligne. Aquests atributs s'expressaran de forma numèrica, de manera que estaran fortament relacionats amb mesures i quantitats, ja que han de constituir atributs quantificables.

El tipus de càncer triat és el de pulmó, degut a que és un dels càncers més freqüents i, per tant, un dels quals tenen més bases de dades.

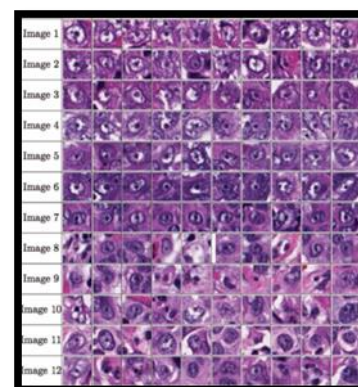
14.4.2- La base de dades

En primer lloc, cal dir que aquest problema utilitza els exemples d'una base de dades recomanada pel Matlab, provinent de la pàgina web de la UCI. La UCI és un magatzem d'informació digital caracteritzada per proveir, de manera gratuïta, una gran quantitat de bases de dades molt diverses entre elles.

En el cas particular que s'està tractant, la base de dades conté exemples dels teixits cancerosos propis del càncer de pulmó (breast cancer). Aquesta s'anomena 'cancer_dataset.mat' i es diferencia en dues parts:

- **Exemples d'entrada o 'cancerInputs'**: es tracta d'una matriu 9x699 que defineix nou atributs en sis-cents noranta nou biòpsies¹⁹ diferents. D'entre els 9 atributs, els més destacables són la uniformitat de la mida de les cèl·lules, la uniformitat de la forma de les cèl·lules i la detecció del procés de mitosi²⁰.
- **Exemples "diana" o 'cancerTargets'**: es tracta d'una matriu 2x699 que defineix dos tipus de tumors (benignes o malignes) en 699 biòpsies.

Aquesta diferenciació és necessària pel tipus de problema en qüestió, ja que d'alguna manera s'ha de relacionar els nou atributs dels teixits analitzats per la xarxa amb el tipus de tumor que presenten els teixits. Així doncs, com que no es poden lligar en una mateixa matriu els dos factors anteriors, s'han de diferenciar en exemples d'entrada i exemples "diana". A la figura 45 es poden observar alguns d'aquests exemples.



¹⁹ Biòpsia: extracció i examen d'un òrgan o fragment de teixit d'un cos viu.

²⁰ Mitosi: procés de divisió de les cèl·lules que consisteix, fonamentalment, en una divisió dels cromosomes i del citoplasma, donant lloc a dues cèl·lules filles amb igual dotació cromosòmica que la cèl·lula mare.

Fig. 45. Exemples de teixits tumorals donats a la xarxa

14.4.3- Explicació del mètode

Aquest projecte s'ha realitzat amb l'ajuda de l'assistent de xarxes neuronals del Matlab, és a dir, fent servir el mètode "nnstart". Tot i que té unes certes limitacions, és molt útil per solucionar problemes que presenten una complexitat més elevada (com el que s'està tractant).

La primera finestra que s'obre en iniciar l'assistent permet triar el tipus de xarxa que es vol desenvolupar. En aquest cas, és convenient triar la que correspon al reconeixement de patrons (Pattern Recognition app), ja que l'objectiu és reconèixer teixits cancerosos.

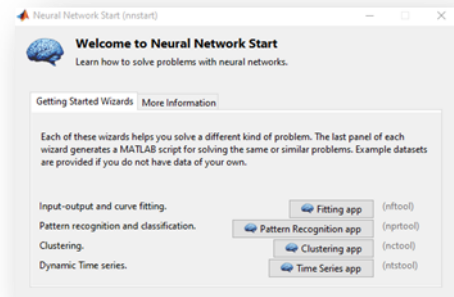


Fig. 46. Inici de l'assistent de xarxes neuronals de Matlab

La finestra que ve a continuació dóna un seguit d'indicacions referents a les característiques de la xarxa neuronal (Figura 47). S'hi pot observar l'esquema de l'arquitectura (entrada, capa oculta, capa de sortida i sortida) i també dos factors força determinants:

- La funció de transferència serà de tipus sigmoide (de fet, és la més habitual).
- L'algoritme utilitzat per entrenar la xarxa serà el de retropropagació (back-propagation).

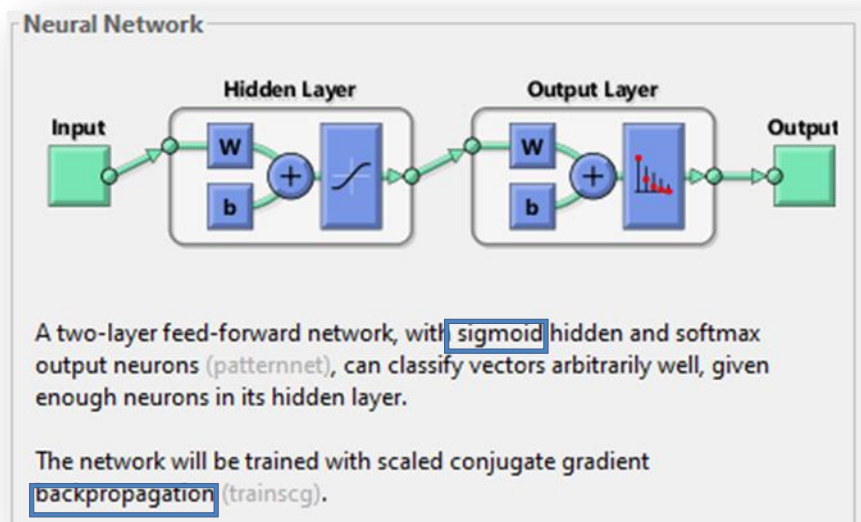


Fig. 47. Esquema de l'arquitectura de la xarxa.

Seguidament, es procedeix a carregar la base de dades, és a dir, a proporcionar la informació que serà donada a la xarxa tant per entrenar-la com per testar-la (figura 48).

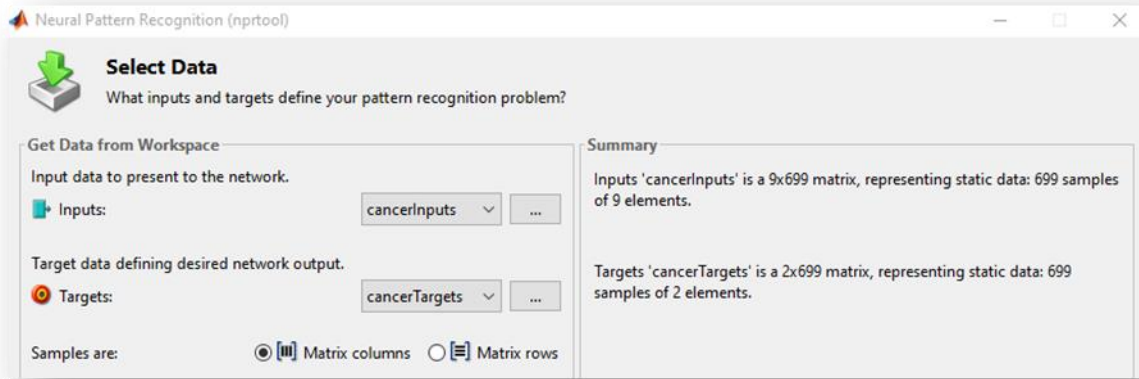


Fig. 48. Menú d'entrada de dades.

Després d'haver proveït la xarxa d'informació, l'assistent demana com s'ha d'organitzar aquesta informació. L'organització de les dades pot classificar-se en tres processos:

- **Entrenament:** Durant aquest procés, les imatges es presenten a la xarxa i aquesta ajusta els seus pesos en funció de l'error que es doni.
- **Validació:** Durant aquest procés, les imatges s'utilitzen per mesurar la capacitat de generalització de la xarxa. Quan la generalització deixa de millorar, el procés d'entrenament s'atura (la validació i l'entrenament s'inicien i s'aturen simultàniament).
- **Test:** Aquest procés no té efectes sobre l'entrenament i proporciona una mesura independent del rendiment de la xarxa.

L'organització d'informació que sol ser més eficaç obeeix a la repartició següent: un **70%** dels exemples són conferits a l'entrenament, mentre que un **15%** s'adjudica als processos tant de validació com de test (figura 49).

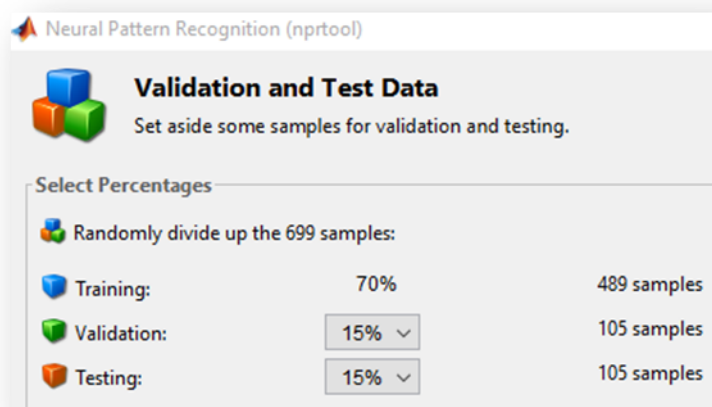


Fig. 49. Classificació de les dades en entrenament (70%) validació (15%) o test (15%).

A continuació, es procedeix a la tria de l'arquitectura que posseirà la xarxa. L'únic paràmetre que permet modificar l'assistent és el nombre de neurones de la capa oculta. Cal recordar que aquesta és la capa encarregada del processament, de manera que un elevat nombre de neurones en aquesta capa millorarà el rendiment de la xarxa fins a un cert límit. Després de fer algunes proves, s'ha determinat que el número de neurones en la capa oculta que dona un millor rendiment és el de 55 neurones (Fig. 50.)

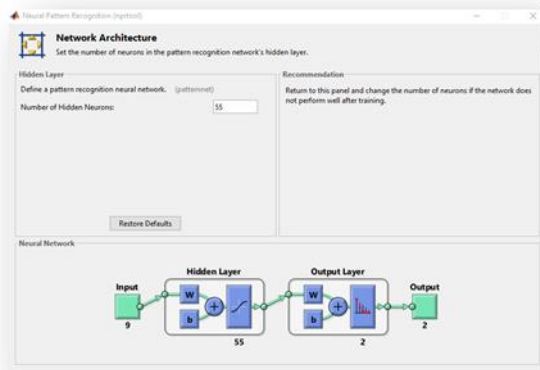


Fig. 50. Arquitectura de la xarxa i tria de neurones en la capa oculta.

Un cop definida l'arquitectura ja es pot entrenar la xarxa. L'entrenament consistirà en anar fent iteracions fins que la generalització (indicada pel procés de validació) deixi de millorar. Com ja s'ha dit amb anterioritat, l'algoritme utilitzat per entrenar la xarxa és el de retropropagació (Fig. 51.).

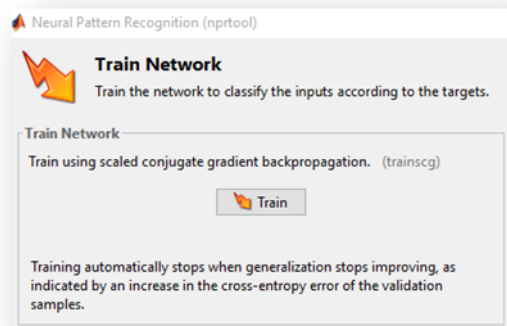
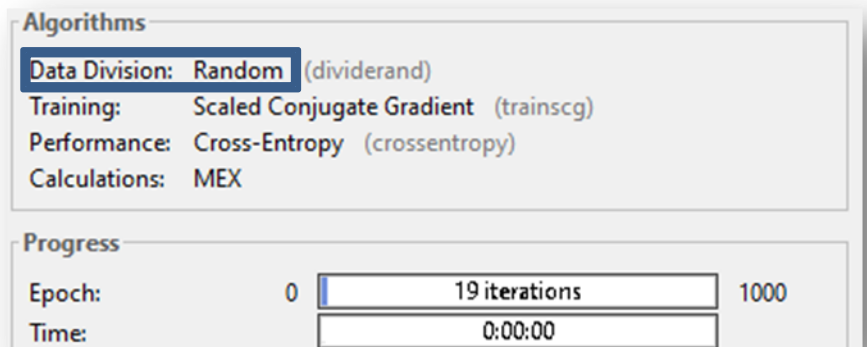


Fig. 51. Entrenament de la xarxa. S'indica que s'utilitza l'algoritme de retropropagació.

A la figura 52 es pot observar que s'han fet un total de 19 iteracions en un temps d'ordre inferior a les centèsimes de segon. Cal recordar que les xarxes neuronals processen en paral·lel i aquesta característica fa que el seu processament sigui molt més ràpid que en els mètodes de computació tradicionals.

Adicionalment, també es pot observar que els pesos de les neurones s'inicien amb uns valors totalment aleatoris (data division: random). A mesura que l'entrenament va avançant, però, els pesos es van ajustant.



14.4.4.- Anàlisi dels resultats: Matriu de confusió

En aquest cas, també resulta interessant analitzar els resultats de l'entrenament a través d'una matriu de confusió de les dades testades (figura 52).

Si es recorda, les dades testades equivalien a un 15% del total de 699 exemples. Això significa que aquesta matriu només té en compte 105 dels 699 exemples –ja que 104 exemples s'han fet servir pel procés de validació (15%) i 490 exemples, en el procés d'entrenament (70%)–.

Un cop fet l'aclariment, és possible entendre perquè la suma dels números que pertanyen als requadres verds i vermells dona exactament un valor de 105. La primera fila fa referència al teixit de tipus 1 –és a dir, un teixit que presenta un tipus de tumor benigne–, i es pot observar que la xarxa ha encertat 67 dels 69 teixits pertanyents a aquest tipus. De la mateixa manera, observant la segona fila, es pot deduir que la xarxa ha encertat 35 dels 36 teixits de tipus 2 –és a dir, que presenten tumors malignes–.

Així doncs, mentre que el percentatge total (requadre en blau) d'encerts és del 97,1%, el percentatge d'errors correspon a un 2,9%. Cal dir que són uns resultats força acceptables, fruit de posar un nombre elevat de neurones a la capa oculta de la xarxa.

En un hipotètic cas en què els resultats no fossin acceptables, existeix la possibilitat de reentrenar la xarxa. No obstant, aquest pas només és recomanable si el percentatge d'error obtingut és molt elevat, ja que reentrenar la xarxa quan aquesta ja funciona correctament no té un efecte positiu en el seu rendiment, més aviat al contrari.

Finalment, quan la xarxa ja està creada i ja s'han dut a terme tots els processos corresponents, es poden afegir de manera opcional més imatges. Així doncs, es pot donar informació nova a la xarxa i aquesta és capaç de classificar-la. El problema principal, però, rau en el fet que, a diferència del projecte de reconeixement de dígit, aquest constitueix un procés molt més pesat.

Output Class	1	2	
1	67 63.8%	2 1.9%	97.1% 2.9%
2	1 1.0%	35 33.3%	97.2% 2.8%
	98.5% 1.5%	94.6% 5.4%	97.1% 2.9%
	1	2	
	Target Class		

Fig. 52. Matriu de confusió dels exemples utilitzats per testar la xarxa.

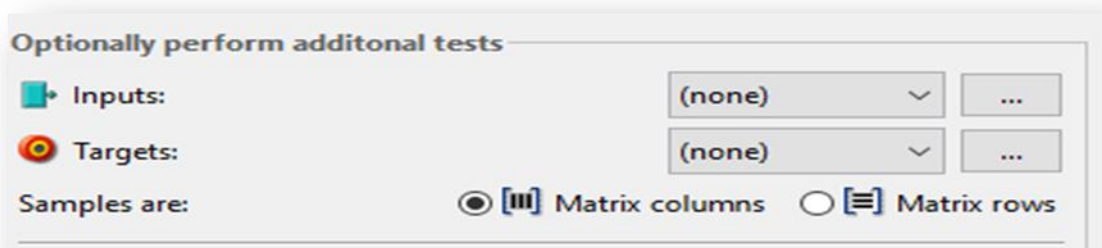


Fig. 53. Menú per afegir dades noves a la xarxa neuronal.

Així doncs, aquesta xarxa podria tenir una certa utilitat en un hospital on es duguessin a terme les proves específiques per obtenir la imatge d'un teixit cancerós. Un cop aplicada la xarxa neuronal a la imatge, aquesta seria correctament classificada una gran majoria de vegades. Tot i que el personal dels hospitals de ben segur està capacitat per fer la mateixa deducció que la xarxa, si s'hagués d'analitzar per exemple un teixit molt extens (i aquest fos dividit en moltes imatges), l'aplicació de la xarxa podria suposar un estalvi de temps molt gran.

No obstant, potser cal dir que per tal que aquesta xarxa arribés a tenir una implementació en un problema real s'hauria d'intentar reduir més el percentatge d'errors. El cas és que, en mans d'un expert en el tema i amb els recursos suficients, reduir el percentatge d'errors no resultaria un procés difícil.

15.- CONCLUSIÓ

Les xarxes neuronals artificials estan inspirades en el funcionament del cervell humà, intentant, per tant, emular aquelles característiques pròpies del seu comportament. Aquest és el principal nexa d'unió entre la biologia i les xarxes neuronals, però és a partir d'aquest punt que ambdues disciplines es separen. El coneixement del funcionament global del cervell és tan limitat que el dissenyador de sistemes no disposa de les dades suficients per copiar les complexes operacions que duu a terme el centre d'operacions humà. Per tant, l'enginyer ha d'anar més enllà dels coneixements biològics, tot buscant estructures i algorismes útils i eficients per resoldre els problemes amb els que es va trobant. En la majoria dels casos, aquesta cerca dona com a resultat un allunyament de la realitat biològica, on el cervell acaba sent una metàfora. Tanmateix, tot i que l'analogia entre la biologia i les xarxes neuronals artificials és tènue en aquest sentit, els resultats d'aquestes últimes evoquen contínuament comparacions amb el funcionament del cervell, ja que, al cap i a la fi, estan basades en aquest.

El camp de les xarxes neuronals artificials és un camp força madur, ja que la computació neuronal és un mètode que s'utilitza en nombroses aplicacions amb finalitats no experimentals. Per altra banda, però, no és un camp tan madur perquè certament mostra un potencial molt gran que no s'acaba de dur a terme, de manera que són necessàries inversions per afavorir-ne el desenvolupament.

El fet que una xarxa sigui capaç d'aprendre d'exemples és molt més del que pot semblar quan s'observen els algorismes d'aprenentatge, on només es veuen modificacions de pesos i errors. Aquestes modificacions signifiquen ni més ni menys que el concepte d'aprenentatge traduït en llenguatge matemàtic. Es pot afirmar, doncs, que en certa manera una xarxa pot aprendre conceptes i extrapolar-ne d'altres. Això es pot considerar tant un comportament intel·ligent com un comportament que sembla que sigui intel·ligent, però el que realment és interessant és que aquest està lligat d'alguna manera a la intel·ligència artificial i que això permet la resolució de problemes.

De fet, tant en el projecte de reconeixement de dígit com en el projecte de reconeixement de teixits cancerosos, l'error relatiu comès per la xarxa neuronal és inferior al 5%, que recordem, és l'error que s'havia posat com a límit. Així doncs, els dos projectes realitzats en el treball podrien implementar-se en problemes reals, provant, així, la validesa i utilitat de les xarxes neuronals artificials.

Per exemple, la xarxa de reconeixement de dígit podria implementar-se en una pissarra digital que tingués la capacitat de digitalitzar els dígit que són dibuixats a mà. També podria implementar-se en aparells com tablets o mòbils en els que es volgués disposar de la mateixa funció.

En el cas de la xarxa de reconeixement de teixits cancerosos, la única possibilitat d'aplicació útil és en el camp de la medicina. Seria força útil per examinar d'una manera ràpida i eficaç un determinat teixit i detectar si aquest presenta tumors

benignes o malignes. Això podria facilitar en certa part la feina del personal sanitari i, especulant potser més del que toca, la nova rapidesa guanyada es podria traduir en menys cues d'espera als hospitals. Tot i que aquesta última afirmació sembla més aviat un intent de treure suc d'on no n'hi ha, en realitat el que es vol fer deduir al lector és que si les xarxes neuronals s'apliquessin no només al reconeixement de càncer sinó també en altres tasques, llavors sí que es podria finalment reduir les cues d'espera en els hospitals.

Actualment, les XNA es consideren una alternativa per totes aquelles tasques on la computació tradicional no arriba a obtenir resultats satisfactoris. S'especula que, en un futur pròxim, les xarxes neuronals podrien arribar a estar al mateix nivell que la computació tradicional si els investigadors proporcionen els coneixements suficients per al desenvolupament d'aquestes, ja que, ara mateix, els fonaments teòrics no són prou robustos per justificar aquestes prediccions. Així, concloent, donat el potencial que presenten les XNA, si s'inverteix en el desenvolupament del camp de la computació neuronal, és probable que la humanitat pugui recórrer a les màquines per solucionar problemes que ara semblen irresolubles, com el pilotatge automàtic de cotxes i avions.

FONTS DOCUMENTALS CONSULTADES:

Per tal de dur a terme aquest treball s'han consultat nombroses fonts documentals amb l'objectiu d'obtenir informació el més fiable possible. Així doncs, aquestes són les fonts documentals que han contribuït més bé en la definició de conceptes necessaris pel treball.

BIBLIOGRAFIA

FOLCH I GUILLÈN, Ramon. *El medi natural*. Barcelona. Enciclopèdia catalana, 1997.

En el capítol 23, apartat 4 (pàgs. 151.155) s'ha utilitzat tota la informació necessària per definir el sistema nerviós.

WEBGRAFIA:

- www.viquipedia.org (diverses consultes)
https://es.wikipedia.org/wiki/Sistema_nervioso Parts del sistema nerviós
<https://es.wikipedia.org/wiki/Cerebro> Cerebellum
<https://ca.wikipedia.org/wiki/Fosfol%C3%ADpid> Fosfolípids
https://ca.wikipedia.org/wiki/Test_de_Turing Test de Turing
http://es.wikipedia.org/wiki/Red_neuronal_artificial Xarxa neuronal artificial
https://es.wikipedia.org/wiki/Variable_discreta_y_variable_continua (variable discreta i contínua)
https://es.wikipedia.org/wiki/Funci%C3%B3n_de_transferencia Funció de transferència
https://es.wikipedia.org/wiki/Neurona_de_McCulloch-Pitts Neurona de McCulloch-Pitts
https://en.wikipedia.org/wiki/Roger_Penrose Roger Penrose
https://ca.wikipedia.org/wiki/Marvin_Minsky Marvin Minsky
https://ca.wikipedia.org/wiki/Test_de_CAPTCHA Captcha
- El cerebell i les seves parts.
<http://www.iaveriana.edu.co/Facultades/Ciencias/neurobioquimica/libros/neurobioquimica/CEREBELO.htm>
- La medulla espinal.
<http://comofuncionaque.com/funciones-de-la-medula-espinal/>
- ANTONIO, José. Intel·ligència artificial i consciència.
http://www3.uah.es/benito_fraile/ponencias/inteligencia-artificial.pdf
- ALEJANDRO, Andrés. Definició d'algoritme
<http://informaticafrida.blogspot.com.es/2009/03/algoritmo.html> Def algoritme

- JULIÁN, Guillermo. Article divulgatiu que explica el concepte de xarxa neuronal i el potencial que tenen.
<http://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-estan-volviendo>
- BASOGAIN OLABE, Xabier. Funcionament de les xarxes neuronals artificials.
http://www.ciberesquina.una.edu.ve:8080/2014_2/350_E.pdf
- Funcions pròpies de les neurones artificials (esgraó i sigmoide)
http://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales-html/x38.html
- Aprenentatge de les xarxes neuronals artificials.
<http://es.slideshare.net/mentelibre/redes-neuronales-de-aprendizaje-no-supervisado-hebb>
- JIMÉNEZ, Eimy. Warren McCulloch i Walter Pitts.
<https://prezi.com/m7rewyt5gkjl/warren-mcculloch-y-walter-pitts/>
- Tutorial de com fer xarxes neuronals especialitzades en el reconeixement de patrons (enllaç proveït pel Matlab).
<http://es.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.html>
- ZADNIK, Ziga. Treball basat en el reconeixement de caràcters. Útil per comprendre com elaborar el projecte enfocat al reconeixement de dígit. http://lab.fs.uni-lj.si/lasin/wp/IMIT_files/neural/doc/seminar5.pdf
- LUERSEN, Martin. Funcionament bàsic del Matlab i explicació del procediment per desenvolupar xarxes neuronals.
<http://members.optusnet.com.au/~luerssen/papers/ANNtutorial.pdf>
- Informació diversa sobre el Matlab
<http://www.monografias.com/trabajos5/matlab/matlab.shtml>
- MORENO, Alfonso. Descripció del GUI.
<http://e-archivo.uc3m.es/bitstream/handle/10016/8488/Proyecto%20Redes%20neuronales%2GUI.pdf?sequence=1>
- Bases de dades de dígit del 0 al 9 escrits a mà.
<http://www.cs.nyu.edu/~roweis/data.html>
- Base de dades de teixits tumorals.
<http://mlearn.ics.uci.edu/MLRepository.html>

- BOFILL, Pau. Qüestionament de que la biologia formi part de les xarxes neuronals artificials, degut a la seva naturalesa abstracta (conclusió).
<http://publicacions.iec.cat/repository/pdf/00000120%5C00000074.pdf>
- ZARZAVILLA, Heidy. Blog que aconsegueix resumir les xarxes neuronals.
http://juanitajuanabernal.blogspot.com.es/p/ejemplo_18.html
- Article en el que es basa la pàgina instapainting per crear les seves imatges mitjançant xarxes neuronals (n'he extret imatges)
<http://arxiv.org/pdf/1508.06576v2.pdf>
<https://www.instapainting.com/ai-painter>
- Article que explica les xarxes neuronals de Google i el seu potencial artístic.
<http://hipertextual.com/2015/07/redes-neuronales>

ANNEX

S'ha considerat oportú incloure els diferents scripts que he obtingut a través del programa Matlab i que defineixen els dos projectes realitzats al llarg del treball.

ANNEX I

Script complet de la “Xarxa neuronal de reconeixement de dígit”:

El que ve a continuació és el seguit d'ordres que s'han d'indicar per tal de executar la xarxa neuronal capaç de reconèixer dígit:

```
clear all
close all
clc

load('usps_all');

figure
for i = 1:10
    subplot(2,5,i)
    imagesc(reshape(squeeze(data(:,1,i)),16,16))
    if i ~=10
        titol = ['Imatge del dígit ', num2str(i)];
    else
        titol = ['Imatge del dígit 0'];
    end
    title(titol)
end
colormap gray

disp('Bloc 1...')
pause
X = double(reshape(data,256,11000)');
ylabel = [1:9 0];
Y = reshape(repmat(ylabel,1100,1),11000,1);
clear data

disp('Bloc 2...')
pause

% binaritzem els valors de X per tenir només blanc (1) i negre (0)
X_BN = double(im2bw(X/max(max(X))));
% definim alguns paràmetres
num_class = 10;      % número de classes
num_exemp = 1100;    % número d'exemples per classe
dimX = 16;           % dimensions en x
dimY = 16;           % dimensions en y

% visualitzem el primer exemple de cada classe, ara amb X_BN
figure
for i = 1:10
    subplot(2,5,i)
    imagesc(reshape((X_BN(1+num_exemp*(i-1),:)),16,16))
```

```

        if i ~=10
            titol = ['Imatge del digit ', num2str(i)];
        else
            titol = ['Imatge del digit 0'];
        end
        title(titol)
    end
    colormap gray

disp('Bloc 3...')
pause

% Create a Pattern Recognition Network
cv = cvpartition(Y, 'holdout', .5);

Xtrain = X_BN(cv.training,:);
Ytrain = Y(cv.training,1);
Xtest = X_BN(cv.test,:);
Ytest = Y(cv.test,1);

% network
net = mlp(dimX*dimY, 100, 10,'softmax');

YtestM = zeros(5500,10);
for t = 1:10
    YtestM(1+550*(t-1):550*t,t) = 1;
end

% Train the Network
options = foptions;
[net, options] = netopt(net, options, Xtrain, YtestM, 'scg');

% Test the Network
ytest = mlpfwd(net,Xtest);
ytestB = ytest>0.5;
[ma,in] = max(ytestB,[],2);

Ytest(end-550:end)=10;

% calculem quants errors s'han fet...
err = (in-Ytest)~=0;
Confmat_NN = confusionmat(in,Ytest);

figure,
heatmap(Confmat_NN, 0:9, 0:9,
1, 'Colormap', 'red', 'ShowAllTicks',1, 'UseLogColorMap',true, 'Colorbar',t
rue);
title('Confusion Matrix: Single Classification Tree')
cvErr = (sum(err)/length(in))*100;
text = ['El percentatge d'encerts és del ', num2str(100-cvErr), '
%'];
disp(text)

% provem amb dades noves entrades manualment...

```



```

% entrem exemples manualment...
% cal fer dos clics per marcar un requadre i tocar una tecla per
% sortir del dibuix...

b = zeros(16,16);
figure
imagesc(b)

t=1;
while t
    [x,y] = ginput(1);
    cx=round(x); if cx==0, cx=cx+1; end
    cy=round(y); if cy==0, cy=cy+1; end
    b(cy,cx)=not(b(cy,cx));
    imagesc(b)
    colormap gray
    t = not(waitforbuttonpress);
end

valors = [1:9,0];

% Test the Network
ytest_b = mlpfwd(net,b(:)');
ytestB_b = ytest_b>0.5;
[ma_b,in_b] = max(ytestB_b,[],2);
text = ['El número dibuixat és el ', num2str(valors(in_b))];
disp(text)

```

ANNEX II

Script complet de la “Xarxa neuronal de reconeixement de dígit”:

Aquest script es genera de forma automàtica una vegada s’ha completat la xarxa neuronal a través de l’assistent que ofereix Matlab. Si s’executa, s’iniciarà la xarxa capaç de reconèixer teixits tumorals.

```

unction [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 10-Dec-
2015 21:31:04.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = 9xQ matrix, input #1
% and returns:
%   y = 2xQ matrix, output #1
% where Q is the number of samples.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

```

```

% Input 1
x1_step1_xoffset = [0.1;0.1;0.1;0.1;0.1;0.1;0.1;0.1;0.1];
x1_step1_gain =
[2.222222222222222;2.222222222222222;2.222222222222222;2.222222222222222;2.
.222222222222222;2.222222222222222;2.222222222222222;2.222222222222222;2.2
222222222222222];
x1_step1_ymin = -1;

% Layer 1
b1 = [-2.2855632471835245;-2.0636182871319546;2.0079113552417622;-
1.9516273893830436;1.8591719072559469;1.6608822290159078;1.65776357109
21987;1.5159279039589588;1.4067050129235643;1.4312284240632653;1.25960
78655426834;1.3726306891075946;1.1014716580339581;1.0891330381704747;-
0.93496930684624102;-0.83491575206873636;-
0.71539245666202067;0.68183397981480409;0.50134942577619757;0.47626351
037200282;-0.3740735770076819;-0.30421712500408449;-
0.21977011195763449;0.13130343289063612;0.033388470126894193;0.0799986
72732651516;0.14895111670207209;-0.29022682088950147;-
0.44518548455043327;-0.44185854826290843;-
0.47449212966501081;1.0758989362775293;-
0.64164781698810003;0.56008235086290059;-
0.84352512327686879;1.0160004669178935;-1.0154164892293847;-
1.1444523666343285;-
1.1790655751897301;1.3200181149545593;1.142235584672612;-
1.5221334604534971;1.5592801599973503;-
1.7480586112625318;1.7340024433828471;1.8051025709497175;1.83993414937
61319;-1.9606780800105752;-2.0738966267616274;-2.1457678430382572];
IW1_1 = [0.35137927703777622 -0.19600954982435126 -0.84744647657295247
-0.65462703115451837 -0.41348743782434455 -0.74113053248704464
0.74428441308300031 -0.42964552480048435
1.2383488752527199;0.084730938092837757 -0.35311952127933099
0.99964183863831746 -1.104362839018225 0.38541486498969751
0.94865001142533678 0.5844536887584113 -0.841568886532295 -
0.51996715648010616;-0.86933719878783866 0.10916037754670087
0.67399093869748616 0.45069769487243772 -0.60107294205701889 -
0.94144524779739736 1.1003641135415314 -0.37104366986116044
0.68400146127053862;0.41292609897935378 0.66432996116592291
0.26275073597676391 -0.31180091872222493 -0.39513488385079837 -
0.95586160566136358 -0.97514698634308239 -0.79271134025883816 1.

% Layer 2
b2 = [0.30477717313009367;0.024320728390096039];
LW2_1 = [0.82376983122990965 0.25948565752759217 -0.71037546235065507
-0.23943215544589008 -0.21780677272086818 0.60410688364239284
0.25963295027683936 0.68520311033234726 0.27217004212272244
0.11026896604778404 0.7879422145132291 -0.90293703451507024 -
0.13066116440892339 -0.6898010646697299 -0.32703059246770133 -
0.34387577075713843 -0.091076155982247595 -0.38949353901984562 -
0.0014803783620919754 0.89053935816219465 -0.12462984413895857 -
0.66365537457065527 -0.81874837775156006 -0.31833661099445465 -
1.0643135755475597 -0.80392048658641346 0.23587250122220413
0.95198335274206336 0.82335280925404075 -0.10015652623671806 -
0.22473912249814687 -1.0769461387246353 -0.68015154392011035
0.54688855995458108 0.35189149317387769 0.61854714962979873 -
0.67368019636080245 0.55576180672916953 0.2420438036647444
0.36069386995109826 -1.0511270643445267 0.083897510381192544
0.33913054641068879 0.48605461724458737 -0.079167085595042902 -
0.59944871748678874 0.40356480537468958 0.7864408908967262 -
0.12264138620895712 -0.084805317438384681;-0.48889678037967094
0.41552612421302487 -0.22935770373666753 0.22334553935555362

```

```

0.57697324967773156 -0.17089607338727733 0.12578332099652148
0.20279427121899257 -0.47061137856432383 0.23017140212197873
0.42257921767262363 0.28971551822441399 -0.11668082878551088
0.55323808271820962 -0.53080230180199772 -0.21931350697712568
0.30802405068011918 -0.022600244669480837 0.34727590097705868
0.45598806702271982 0.58853201352325402 -0.94673439809227156 -
0.6849608809597042 -0.42976999400815347 0.16748574550748946 -
0.71230441527348043 0.74505073566527669 0.13813676467973035
0.27743076688692192 -0.20756243210209596 0.068355226492429311
0.99449873761597285 -0.31821136251525622 -0.1045601782507303
0.51233974040106478 1.0161461834698555 -0.42758600478340864
0.42669328372364662 -0.0087444622999302592 0.34192895349850483
0.36306418880243346 0.70402982848314721 0.29175946569226563 -
0.35193501263340105 0.22653719066986674 0.036551246838427309 -
0.16104715573730261 0.34887408277976806 -0.2488717859127022
% ===== SIMULATION =====

% Dimensions
Q = size(x1,2); % samples

% Input 1
xp1 =
mapminmax_apply(x1,x1_step1_gain,x1_step1_xoffset,x1_step1_ymin);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = softmax_apply(repmat(b2,1,Q) + LW2_1*a1);

% Output 1
y1 = a2;
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y =
mapminmax_apply(x,settings_gain,settings_xoffset,settings_ymin)
y = bsxfun(@minus,x,settings_xoffset);
y = bsxfun(@times,y,settings_gain);
y = bsxfun(@plus,y,settings_ymin);
end

% Competitive Soft Transfer Function
function a = softmax_apply(n)
nmax = max(n,[],1);
n = bsxfun(@minus,n,nmax);
numer = exp(n);
denom = sum(numer,1);
denom(denom == 0) = 1;
a = bsxfun(@rdivide,numer,denom);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n)
a = 2 ./ (1 + exp(-2*n)) - 1;

```