Treball de Recerca

Intel·ligència Artificial

Aprenentatge Automàtic i Xarxes Neuronals aplicats als entorns virtuals i reals

Stepan Klymonchuk

Tutor: Raül Bercet Institut Narcís Oller 9 de desembre de 2019 A la meva família

Abstract

In computer science, artificial intelligence (AI) is an intelligence presented by machines, unlike the natural intelligence shown by humans and animals. This field of computing is devoted to the development of algorithms that can be implemented on machines so that they make decisions and, therefore, provide them with a certain intelligence to perform cognitive functions that imitate human beings, such as learning and solving problems. In recent years the field of artificial intelligence has evolved a great deal, to such an extent that today many companies have developed intelligent systems capable of reaching points that no one would have thought before. Most of the AI examples that currently exist, from computers that play chess to voice recognition and autonomous vehicles, largely depend on deep learning and the processing of natural language. By using these technologies, computers can be trained to perform certain tasks by processing large amounts of data and recognizing patterns. In this work, an introduction will be made to the world of neuronal networks and deep learning to try to recognize and classify manuscript digits with a high percentage of success, testing different architectures of neural networks and varying their parameters.

Resum

En ciències de la computació, la intel·ligència artificial (IA) és una intel·ligència que presenten les màquines, a diferència de la intel·ligència natural que mostren els humans i els animals. Aquest camp d'informàtica es dedica al desenvolupament d'algoritmes que es puguin implementar en màquines perquè aquestes prenguin decisions i, per tant, dotar-les de certa intel·ligència per realitzar funcions cognitives que imiten a les humanes, com ara aprendre i resoldre problemes. En els últims anys el camp la intel·ligència artificial ha evolucionat molt, fins a tal punt que avui en dia moltes companyies han desenvolupat sistemes intel·ligents capaços d'arribar a punts que anys enrere ningú s'hauria pensat. La majoria dels exemples de la IA que existeixen actualment, des dels ordinadors que juguen als escacs fins al reconeixement de veu i els vehicles autònoms, depenen en gran manera de l'aprenentatge profund i del processament del llenguatge natural. Utilitzant aquestes tecnologies, els ordinadors poden ser entrenats per realitzar determinades tasques processant grans quantitats de dades i reconeixent patrons. En aquest treball es realitzarà una introducció al món de les xarxes neuronals i l'aprenentatge profund per intentar reconèixer i classificar dígits manuscrits amb un alt percentatge d'encert, provant diferents arquitectures de xarxes neuronals i variant els seus paràmetres.

Resumen

En ciencias de la computación, la inteligencia artificial (IA) es una inteligencia que presentan las máquinas, a diferencia de la inteligencia natural que muestran los humanos y los animales. Este campo de informática se dedica al desarrollo de algoritmos que se puedan implementar en máquinas para que estas tomen decisiones y, por tanto, dotarlas de cierta inteligencia para realizar funciones cognitivas que imitan a las humanas, tales como aprender y resolver problemas. En los últimos años el campo la inteligencia artificial ha evolucionado mucho, hasta tal punto que hoy en día muchas compañías han desarrollado sistemas inteligentes capaces de llegar a puntos que años atrás nadie habría pensado. La mayoría de los ejemplos de la IA que existen actualmente, desde los ordenadores que juegan al ajedrez hasta el reconocimiento de voz y los vehículos autónomos, dependen en gran medida del aprendizaje profundo y del procesamiento del lenguaje natural. Utilizando estas tecnologías, los ordenadores pueden ser entrenados para realizar determinadas tareas procesando grandes cantidades de datos y reconociendo patrones. En este trabajo se realizará una introducción al mundo de las redes neuronales y el aprendizaje profundo para intentar reconocer y clasificar dígitos manuscritos con un alto porcentaje de acierto, probando diferentes arquitecturas de redes neuronales y variando sus parámetros.

		•
Т	7	٦
١	v	

La gent que no es creu que les matemàtiques són senzilles, és només que no s'ha adonat de com és de complicada la vida.

J. L. Von Neumann

Índex

\mathbf{A}	bstract	iii
\mathbf{R}	esum	iv
\mathbf{R}	esumen	\mathbf{v}
Ín	ndex	vii
Ín	ndex de figures	x
Ín	ndex de listings	xi
1	Introducció	1
	1.1 Motivació	1
	1.2 Objectius	2
2	$\operatorname{Intel}\cdot\operatorname{lig\`encia}$ artificial	3
	2.1 Història	3
	2.2 Branques de la IA segons les aplicacions	4
	Referències	5
3	Aprenentatge automàtic	7

		viii
	3.1 Aprenentatge supervisat	7
	3.2 Aprenentatge no supervisat	8
	3.3 Aprenentatge per reforç	8
	3.4 Aprenentatge profund	9
	Referències	9
4	Xarxes neuronals artificials	11
	4.1 Perceptró	11
	4.1.1 Interpretació gràfica	13
	4.1.2 Exemple pràctic	13
	4.1.3 Xarxes de perceptrons	15
	4.1.4 Aprenentage del perceptró	15
	4.2 Neurona sigmoide	17
	Referències	18
5	Disseny i programació de xarxes neuronals	20
	5.1 Entorn	22
	5.2 Xarxa neuronal	23
	Referències	28
6	Conclusió	29
\mathbf{A}_{1}	pèndixs	31
\mathbf{A}_{1}	pèndix A Desenvolupament del treball	32
	Referències	33

	ix
Apèndix B Programari utilitzat: LaTEX	35
Referències	36

Índex de figures

3.1	Aprenentatge per reforçament	9
4.1	Funció esglaó de Heaviside	12
4.2	Representació d'un perceptró	12
4.3	La separació de l'espai d'entrada amb un perceptró	13
4.4	Exemple d'un perceptró amb $\mathbf{w}=(6,2,2)$ i $b=-5$	14
4.5	Xarxa de perceptrons	15
4.6	Aprentatge d'una xarxa de perceptrons	16
4.7	Representació d'una neurona sigmoide	17
4.8	Funció sigmoide	18
5.1	Base de dades MNIST	20
5.2	Xarxa bàsica per MNIST $^{[8]}$	22
5.3	Xarxa neuronal	25
5.4	Precisió de la xarxa	26
5.5	Error de la xarxa	26
5.6	Applicació GUI	27
5.7	Applicació GUI	27

Índex de listings

5.1	Importar MNIST i preprocessar les dades	2^{2}
5.2	Model de xarxa neuronal	2!

Capítol 1

Introducció

1.1 Motivació

A l'hora de triar el tema del treball de recerca vaig pensar que m'agradaria fer-lo de programació, però no només duna manera abstracta, sinó en relació amb alguna aplicació concreta que tingués a veure amb els meus interessos. Un dia vaig descobrir què eren les xarxes neuronals i la intel·ligència artificial. D'una banda, em vaig adonar que existeixen molts mites i molta desinformació sobre el tema de la IA: la intel·ligència artificial substituirà tots els llocs de treball, superarà la intel·ligència humana o conduirà a l'esclavitud de la nostra raça per part dels éssers robòtics superiors, com es descriu en diverses pel·lícules i històries de ciència-ficció.

De l'altra banda, em va sorprendre la quantitat d'aplicacions que té la IA actualment (concretament l'aprenentatge automàtic i les xarxes neuronals) i la seva important presència en el nostre dia a dia. D'aquesta manera, després d'haver fet recerca per informar-me del tema, sem van acudir aquestes preguntes:

- Poden els ordinadors aprendre de l'experiència com nosaltres, humans?
- Què són les xarxes neuronals i perquè són tan útils?
- Com ho fan les xarxes neuronals per reconèixer imatges?
- Quins altres mètodes d'aprenentatge automàtic hi ha a part de les xarxes neuronals?

A partir d'aquestes preguntes vaig decidir fer un treball relacionat amb la intel·ligència artificial.

El fet d'haver triat aquest tema també està motivat pel meu interès per les matemàtiques perquè estan relacionades amb la programació. A grans trets el treball consisteix, per

una banda, en entendre què és la IA, l'aprenentatge automàtic i les xarxes neuronals, com funcionen, quines són les seves aplicacions, etc. i, per l'altra, en dissenyar i programar models de xarxes neuronals i aplicar-les als entorns reals i virtuals per resoldre problemes concrets.

1.2 Objectius

Concretament els objectius d'aquest treball són:

- Investigar en àrees d'IA, informàtica, i programació per aprendre conceptes que poden ser aplicats en diferents àrees com l'anàlisi d'imatges o presa de decisions.
- Explicar el funcionament de diferents mètodes d'aprenentatge automàtic (centrantse principalment en les xarxes neuronals).
- Demostrar que els sistemes d'aprenentatge automàtic són útils i aplicables.
- Comprendre els conceptes, el funcionament i l'abast de les xarxes neuronals i mostrar el procés de disseny i implementació de xarxes neuronals amb TensorFlow i Keras per aconseguir objectius específics.

En el camp de visió artificial (Reconeixement d'imatges):

- Dissenyar i implementar un model predictiu eficaç en la classificació de dígits escrits a mà utilitzant el conjunt de dades MNIST.
- Analitzar el funcionament i el rendiment de diferents arquitectures de xarxes neuronals.

Capítol 2

Intel·ligència artificial

2.1 Història

Tot i que els primers referents històrics es remunten als anys 30 amb Alan Turing, considerat el pare de la intel·ligència artificial, es considera que el punt de partida és l'any 1950, precisament, quan Turing publica un article amb el títol Computing machinery and intelligence a la revista Mind, on es va plantejar la pregunta: poden pensar les màquines? I proposa un mètode per determinar si una màquina pot pensar. Els fonaments teòrics de la IA es troben en l'experiment que es proposa en aquest article i que va passar a anomenar-se test de Turing. Mitjançant la superació d'aquest test per una màquina, és possible que la màquina pugui passar per un ésser humà. El test manté la seva vigència en l'actualitat i és el motiu d'estudis i investigacions contínues.^[1]

No obstant això, nombrosos investigadors i historiadors consideren que el punt de partida de la intel·ligència artificial moderna va ser l'any 1956, quan els pares de la intel·ligència artificial moderna, John McCarty, Marvin Misky i Claude Shannon van definir formalment el terme durant la conferència de Darmouth com la ciència i l'enginy de fer màquines intel·ligents, especialment programes de càlcul intel·ligent.^[2]

El supercomputador Deep Blue d'IBM va guanyar el 1997 al campió mundial d'escacs Gari Kaspàrov, després d'un fracàs previ en 1996 on va guanyar Kaspàrov. L'any 1997 és considerat per alguns historiadors de la IA com el punt d'inflexió on va començar a sentir-se de la intel·ligència artificial fora dels àmbits acadèmics i de recerca.^[1].

Al febrer de 2011, el supercomputador Watson d'IBM, el model de computador cognitiu, com l'anomena el seu creador IBM, guanya al concurs televisiu dels Estats Units *Jeopardy!*, en el qual es realitzen preguntes i qüestions diferents de tota mena, cultura i coneixement, als dos millors concursants del programa, Brad Ruttler i Ken Jennings.^[3]

Una altre fet important va ser la presentació d'Apple de l'assistent virtual Siri integrat al telèfon mòbil iPhone 4S a l'any 2011 i on van començar les primeres experiències d'aprenentatge automàtic i els primers indicis d'aprenentatge profund.^[4]

L'any 2012 és considerat com l'any clau de la segona generació d'intel·ligència artificial, amb el llançament d'assistents virtuals recolzats en IA amb algoritmes d'aprenentatge profund. Al juny de 2012 Google va presentar el seu assistent virtual, Google Now, i a l'abril de 2014 Microsoft va presentar el seu propi assistent virtual, Cortana.^[3]

El 9 de març de 2016, el programari d'intel·ligència artificial Alpha Go de Google De-epMind es va enfrontar al sud-coreà Se-Dol, campió mundial de Go, un joc mil·lenari d'estratègia molt complex, en una partida a cinc jocs. Alpha Go va guanyar els tres primers jocs netament i només en l'últim cinquè joc Se-Dol va guanyar, gràcies a un moviment inicial que va fer i on es va comprovar que la màquina estava poc entrenada per enfrontar-se a situacions inesperades.^[1].

2.2 Branques de la IA segons les aplicacions

En aquesta secció, veurem els diferents camps compatibles amb IA:[3],[5],[6]

- Aprenentatge automàtic. L'aprenentatge automàtic és la ciència d'aconseguir que les màquines interpretin, processin i analitzin dades per tal de resoldre problemes del món real.
- Aprenentatge profund. L'aprenentatge profund és el procés d'implementar les xarxes neuronals en dades d'alta dimensió per obtenir informació i formar solucions. L'aprenentatge profund és un camp avançat d'aprenentatge automàtic que es pot utilitzar per resoldre problemes més avançats.
- Jocs. La IA té un paper crucial en jocs estratègics com escacs, pòquer, tres en ratlla, etc., on la màquina pot pensar en un gran nombre de posicions possibles basades en coneixements heurístics.
- Processament del llenguatge natural. És possible interactuar amb l'ordinador que entén el llenguatge natural que parlen els humans.
- Sistemes experts. Hi ha algunes aplicacions que integren màquina, programari i informació especial per proporcionar raonament i assessorament. Proporcionen explicació i assessorament als usuaris.
- Sistemes de visió. Aquests sistemes comprenen, interpreten i comprenen l'entrada visual a l'ordinador. Per exemple, un avió d'espionatge fa fotografies, que s'utilitzen per conèixer informació espacial o elaborar mapes de les zones. Els metges utilitzen

sistemes experts clínics per diagnosticar els pacients. La policia utilitza programari informàtic que pot reconèixer la cara del criminal amb el retrat elaborat per un artista.

- Reconeixement de veu. Alguns sistemes intel·ligents són capaços d'escoltar i comprendre el llenguatge en termes d'oracions i els seus significats mentre un humà en parla. Pot manejar diferents accents, paraules d'argot, soroll en segon pla, canvis en la pronunciació a causa del fred, etc.
- Reconeixement d'escriptura. El programari de reconeixement d'escriptura manual llegeix el text escrit en paper per un bolígraf. Pot reconèixer les formes de les lletres i convertir-la en text editable.
- Robots intel·ligents. Els robots són capaços de realitzar les tasques donades per un humà. Disposen de sensors per detectar dades físiques del món real com ara llum, calor, temperatura, moviment, so i pressió. Disposen de processadors eficients, múltiples sensors i una enorme memòria per mostrar intel·ligència. A més, són capaços d'aprendre dels seus errors i es poden adaptar al nou entorn.

Referències

- [1] "Breve historia de la inteligencia artificial: el camino hacia la empresa". [En línia]. Disponible en: http://asesoresdepymes.com/breve-historia-la-inteligencia-artificial-camino-hacia-la-empresa/. [Accedit: 09-12-2019]. 3, 4.
- [2] Live Science, "A Brief History of Artificial Intelligence | Live Science". [En línia]. Disponible en: https://www.livescience.com/49007-history-of-artificial-intelligence.html. [Accedit: 09-12-2019]. 3.
- [3] Wikipedia, "Historia de la inteligencia artificial Wikipedia, la enciclopedia libre". [En línia]. Disponible en: https://es.wikipedia.org/wiki/Historia%7B%5C_%7Dde%7B%5C_%7Dla%7B%5C_%7Dinteligencia%7B%5C_%7Dartificial. [Accedit: 09-12-2019]. 3, 4.
- [4] National Geographic, "Breve historia visual de la inteligencia artificial". [En línia]. Disponible en: https://www.nationalgeographic.com.es/ciencia/breve-historia-visual-inteligencia-artificial%7B%5C_%7D14419. [Accedit: 09-12-2019]. 4.

- [5] Tutorials Point, "AI with Python Primer Concept Tutorialspoint". [En línia]. Disponible en: https://www.tutorialspoint.com/artificial%7B%5C_%7Dintelligence%7B%5C_%7Dwith%7B%5C_%7Dpython/artificial%7B%5C_%7Dintelligence%7B%5C_%7Dwith%7B%5C_%7Dpython%7B%5C_%7Dprimer%7B%5C_%7Dconcepts.htm. [Accedit: 09-12-2019]. 4.
- [6] Edureka, What Are The Types Of Artificial Intelligence? | Branches of AI | Edureka. [En línia]. Disponible en: https://www.edureka.co/blog/types-of-artificial-intelligence/ [Accedit: 09-12-2019]. 4.

Capítol 3

Aprenentatge automàtic

En el sentit més bàsic, l'aprenentatge automàtic és una manera d'implementar la intel·ligència artificial. De forma similar a l'IA, l'aprenentatge automàtic és una branca de la informàtica en la qual s'estudia el disseny d'algoritmes que puguin aprendre. En l'aprenentatge automàtic s'utilitzen algoritmes que proporcionen a l'ordinador la capacitat d'aprendre i comprendre automàticament sense haver de ser programat una vegada darrere l'altra.

Com aprendrà l'ordinador de forma automàtica? Amb les dades. S'introdueixen les dades amb diferents atributs o característiques a partir de les quals els algoritmes han d'entendre i fer prediccions en funció de les dades proporcionades. Una vegada que l'algoritme ha après i interpretat les dades, és a dir, que s'ha format per si mateix, es posar l'algoritme a prova i sense programar-les explícitament, introduir dades de prova i obtenir resultats.

3.1 Aprenentatge supervisat

L'aprenentatge supervisat, com el seu nom indica, és una tècnica d'aprenentatge on es supervisa o es controla tot el procés d'aprenentatge. L'objectiu principal d'aquests algoritmes d'aprenentatge és predir el resultat d'un conjunt de mostres d'entrenament juntament amb les etiquetes (resultats desitjats) de mostres. Durant l'entrenament l'algoritme sap el que hauria de predir, per això s'anomena aprenentatge supervisat.^[1]

Suposem que tenim 10000 imatges, 5000 imatges de gats i 5000 de gossos, i cada imatge té una etiqueta: 0 per als gats i 1 per als gossos. L'objectiu d'aquesta tècnica és trobar patrons a les dades donades segons les etiquetes. L'algoritme d'aprenentatge supervisat intentarà trobar un límit entre les 10000 imatges que les divideix en dues parts segons l'etiqueta. De manera que en un moment de prova quan una nova imatge, per exemple una imatge d'un gat es presenta com a entrada sense cap etiqueta, l'algoritme li posarà

l'etiqueta 0, cosa que significa que l'algoritme pot predir o classificar aquesta imatge com a imatge d'un gat.^[2]

3.2 Aprenentatge no supervisat

A diferència de l'aprenentatge supervisat, els algoritmes d'aprenentatge no supervisat prenen un conjunt de dades que conté només les entrades, sense les etiquetes. Per tant, els algorismes aprenen de les dades que no han estat etiquetades ni classificades.^[3]

Els algoritmes s'elaboren de tal manera que puguin trobar l'estructura i els patrons adequats en les dades. Una vegada aquests patrons consistents es fan evidents, els punts similars es poden agrupar junts, i diferents punts estaran en diferents grups. Una aplicació de l'aprenentatge no supervisat es troba en el camp de l'estimació de la densitat en l'àmbit d'estadística amb finalitat de visualització o anàlisi. [1]

3.3 Aprenentatge per reforç

L'aprenentatge per reforç és un tipus d'aprenentatge automàtic que té un agent (per exemple, un robot) que aprèn a comportar-se en un entorn fent accions i quantificant els resultats. Si l'agent realitza una resposta correcta, obtindrà un punt de recompensa, fet que augmenta la confiança de l'agent per dur a terme més accions d'aquest tipus. El funcionament d'aquests algoritmes es basa en el procés de decisió de Markov. [3], [4]

Per exemple, a la fig. 3.1, es pot veure que l'agent (robot) actua en l'entorn (un laberint), interpretant amb els ulls, obté una recompensa per cada moviment correcte i en funció del seu estat actual efectua la següent acció.

Algunes aplicacions de l'aprenentatge per reforç són jugar a jocs (Alpha Go, Escacs, Mario), robòtica, sistemes de control de semàfors, etc.

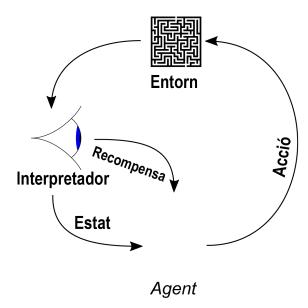


Figura 3.1: Aprenentatge per reforçament

3.4 Aprenentate profund

L'aprenentatge profund és una subcategoria d'aprenentatge automàtic. De forma similar a l'aprenentatge automàtic, l'aprenentatge profund també té l'aprenentatge supervisat, no supervisat i per reforç. Com s'ha comentat anteriorment, la idea de la IA es va inspirar en el cervell humà. Per tant, l'aprenentatge profund es va inspirar en xarxes neuronals artificials i les xarxes neuronals artificials es van inspirar en xarxes neuronals biològiques humanes. L'aprenentatge profund és una de les maneres d'executar l'aprenentatge automàtic. [1],[5]

L'aprenentatge profund utilitza xarxes neuronals artificials amb diverses capes ocultes (fig. 4.5) per extreure progressivament característiques o atributs de nivell superior a partir l'entrada. Per exemple, en el processament d'imatges, les capes inferiors poden identificar les vores, mentre que les capes superiors poden identificar els conceptes rellevants per a un humà com ara dígits, lletres o cares.^[6]

Referències

[1] DataCamp, "Machine Learning & Deep Learning". [En línia]. Disponible en: https://www.datacamp.com/community/tutorials/machine-deep-learning. 7-9.

- [2] Edureka, What Are The Types Of Artificial Intelligence? | Branches of AI | Edureka. [En línia]. Disponible en: https://www.edureka.co/blog/types-of-artificial-intelligence/ [Accedit: 09-12-2019]. 8.
- [3] Wikipedia, "Machine learning Wikipedia". [En línia]. Disponible en: https://en.wikipedia.org/wiki/Machine%7B%5C_%7Dlearning. [Accedit: 09-12-2019]. 8.
- [4] Towards Data Science, "Reinforcement Learning Demystified: Markov Decision Processes (Part 1)". [En línia]. Disponible en: https://towardsdatascience.com/reinforcement-learning-demystified-markov-decision-processes-part-1-bf00dda41690. [Accedit: 09-12-2019]. 8.
- [5] Deep Learning, *Deep Learning*. [En línia]. Disponible en: http://deeplearning.net/ [Accedit: 09-12-2019]. 9.
- [6] Wikipedia, *Deep learning Wikipedia*. [En línia]. Disponible en: https://en.wikipedia.org/wiki/Deep%7B%5C_%7Dlearning [Accedit: 09-12-2019]. 9.

Capítol 4

Xarxes neuronals artificials

El cervell és, sens dubte, l'òrgan més complex del nostre cos. El cervell humà és una xarxa neuronal biològica: una xarxa interconnectada de neurones que transmeten senyals elèctrics. Les neurones o cèl·lules nervioses són les unitats fonamentals del cervell i el sistema nerviós. Per les dendrites (ramificacions nervioses), les neurones reben senyals d'entrada a través dels neurotransmissors que alliberen els axons d'altres cèl·lules nervioses i, quan els senyals rebuts són prou forts (superen un cert llindar), les neurones s'activen i emeten un senyal de sortida a través d'un axó. L'impuls elèctric viatja al llarg de l'axó i provoca l'alliberament de neurotransmissors en la sinapsi, que és el punt on es produeix aquest alliberament i la recepció del missatge per una altra neurona. [1] D'aquesta manera les neurones es poden comunicar entre elles. Tanmateix, el funcionament del cervell humà és un misteri molt més complex, que no s'explicarà en aquest treball.

Les xarxes neuronals artificials (en endavant, xarxes neuronals) són un model d'aprenentatge automàtic inspirat en xarxes neuronals biològiques que formen els cervells dels humans i animals. Una xarxa neuronal està formada per un conjunt d'unitats o nodes connectats anomenats neurones artificials (en endavant, neurones) que simulen el funcionament de les neurones biològiques. A través de les connexions (sinapsis en les neurones biològiques) les neurones poden transmetre i rebre senyals d'altres neurones. Cada neurona rep una sèrie d'entrades, processa aquestes entrades i emet una sortida. [2]

4.1 Perceptró

Per començar explicaré què és un perceptró. El perceptró és la xarxa neuronal més simple possible: un model computacional duna sola neurona. Els perceptrons van ser inventats i desenvolupats en els anys 1950 i 1960 per Frank Rosenblatt.^[3] Tanmateix, el principal model de neurones més utilitzat actualment és la neurona sigmoide.

Un perceptró rep diverses entrades, normalment binàries, en forma de vector $\mathbf{x} = (x_1, x_2, \dots, x_{n-1}, x_n)$, i produeix una única sortida binària y. La seva funció d'activació és la funció esglaó de Heaviside (fig. 4.1).

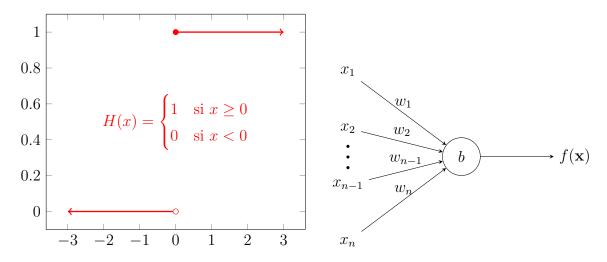


Figura 4.1: Funció esglaó de Heaviside Fig

Figura 4.2: Representació d'un perceptró

En la fig. 4.2 es mostra un perceptró amb n entrades. A l'hora de calcular la sortida, el vector de pesos $\mathbf{w} = (w_1, w_2, \dots, w_{n-1}, w_n)$ expressa la importància de les respectives entrades per a la sortida. la qual depèn de si la suma ponderada $\sum_{i=1}^{n} w_i x_i$ és major o menor que el llindar, un paràmetre real del perceptró, [4] com es pot veure en l'eq. (4.1):

$$y = f(\mathbf{x}) = \begin{cases} 1 & \text{si } \sum_{i=1}^{n} w_i x_i \ge \text{llindar} \\ 0 & \text{altrament} \end{cases}$$
 (4.1)

Per simplificar podem reescriure $\sum_{i=1}^{n} w_i x_i$ com un producte escalar de vectors, $\mathbf{w} \cdot \mathbf{x} \equiv \sum_{i=1}^{n} w_i x_i$. El llindar pot ser substituït pel *biaix*, $b \equiv -\text{llindar}$, que indica la facilitat del perceptró per activar-se (emetre un 1).^[5] L'eq. (4.1) es pot expressar com:

$$y = f(\mathbf{x}) = \begin{cases} 1 & \text{si } \mathbf{w} \cdot \mathbf{x} + b \ge 0 \\ 0 & \text{altrament} \end{cases}$$
 (4.2)

4.1.1 Interpretació gràfica

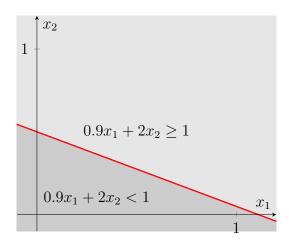


Figura 4.3: La separació de l'espai d'entrada amb un perceptró

Un perceptró separa l'espai d'entrada en dos semiespais. Per als punts que pertanyen a un semiespai el resultat del càlcul és 0 i per als que pertanyen a l'altre, és 1. La fig. 4.3 mostra la separació del pla en dos semiplans en el cas d'un perceptró de dues entrades, x_1 i x_2 , amb llindar = 1, i pesos $w_1 = 0.9$ i $w_2 = 2$. És possible generar separacions arbitràries d'espai d'entrada ajustant els paràmetres d'aquest exemple. [6]

4.1.2 Exemple pràctic

En resum, un perceptró és un dispositiu que pren decisions sospesant l'evidència. [4],[7] Intentem reproduir aquest procés de presa de decisions al món real. Suposem que vull veure un partit de futbol. Abans de prendre la decisió de reservar les entrades, tindria en compte diversos factors:

- 1. Hi juga el meu equip preferit?
- 2. El preu de l'entrada és barat?
- 3. El temps és bo?

Aquestes són els quatre factors (variables) a partir dels quals prendré la decisió. Són les entrades binàries x_1, x_2, x_3 per al perceptró, respectivament. Per exemple, tenim $x_1 = 1$ si m'agraden els equips que juguen, i $x_1 = 0$ si no m'agraden. De la mateixa manera, $x_2 = 1$ si l'entrada és barata, i $x_2 = 0$ si no ho és. I de manera similar per a x_3 .

Ara s'ha d'establir els pesos w_1, w_2, w_3 . Els pesos relacionen cada entrada amb la de sortida corresponent. Suposem que adoro el futbol, de manera que estic encantat d'anar

al festival, encara que faci mal temps i el preu de les entrades sigui car. Però sóc molt exigent amb els equips que hi juguen.

Podem triar un pes $w_1 = 6$ per als equips, i $w_2 = 2$ i $w_3 = 2$ per als altres factors (fig. 4.4). El valor més gran de w_1 i indica que els equips són molt importants per a mi, molt més que el temps i el preu. També establim un llindar mitjà de 5, llindar = 5, o un biaix de -5, b = -5.

Ara avaluem els dos casos següents:

• El meu equip favorit juga, $x_1 = 1$; el preu de l'entrada és molt alt, $x_2 = 0$; el temps és molt dolent, $x_3 = 0$. Podem veure que el valor avaluat del model és 1, superior a 0 (eq. (4.3)). Per tant, la sortida del perceptró serà 1 (eq. (4.1)), que vol dir que aniré al partit (fig. 4.4a).

$$\mathbf{w} \cdot \mathbf{x} + b = w_1 x_1 + w_2 x_2 + w_3 x_3 + b = 6 \cdot 1 + 2 \cdot 0 + 2 \cdot 0 - 5 = 1 > 0$$
 (4.3)

• El meu equip favorit no juga, $x_1 = 0$; el preu de l'entrada és molt barat, $x_2 = 1$; el temps és bo, $x_3 = 1$. Podem veure que el valor avaluat del model és -1, inferior a 0 (eq. (4.4)). Per tant, la sortida del perceptró serà 0 (eq. (4.1)), que vol dir que aniré al partit (fig. 4.4b).

$$\mathbf{w} \cdot \mathbf{x} + b = w_1 x_1 + w_2 x_2 + w_3 x_3 + b = 6 \cdot 0 + 2 \cdot 1 + 2 \cdot 1 - 5 = -1 < 0 \tag{4.4}$$

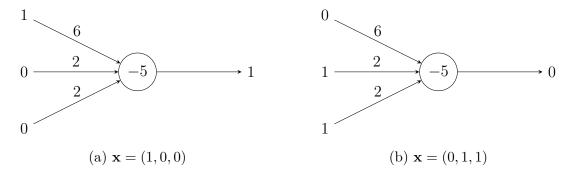


Figura 4.4: Exemple d'un perceptró amb $\mathbf{w} = (6, 2, 2)$ i b = -5

Com podem veure, amb aquests paràmetres el perceptró implementa un model de presa de decisions, que produeix un 1 a la sortida sempre que hi jugui el meu equip preferit, i un 0 sempre que no hi jugui. El preu de les entrades i el temps no influeixen en el resultat.

Variant els pesos i el llindar, podem obtenir diferents models de presa de decisions. Per exemple, suposem que escollim un llindar de 3. Aleshores, el perceptró decidirà que aniré a veure el partit quan hi jugui el meu equip preferit o quan tant el preu de l'entrada sigui barat i el temps sigui bo. És a dir, seria un model diferent de presa de decisions. Un llindar més petit significa que estic més disposat a anar al partit.^[4]

4.1.3 Xarxes de perceptrons

Evidentment, el perceptró no és un model complet de presa de decisions humanes, però el que ens mostra l'exemple anterior és la capacitat d'un perceptró de prendre decisions simples. Per tant, sembla plausible que una xarxa complexa de perceptrons (fig. 4.5) pugui prendre decisions molt subtils.^[7]

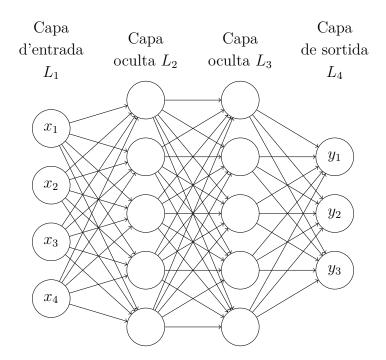


Figura 4.5: Xarxa de perceptrons

Les entrades x_1, x_2, \ldots es poden representar com a variables flotant a l'esquerra de la xarxa de perceptrons (fig. 4.2), però és convencional dibuixar una capa addicional de perceptors (la capa d'entrada L_1 , fig. 4.5) per codificar les entrades. Els perceptrons d'entrada realment no són perceptrons, sinó unitats especials que simplement emeten els valors desitjats x_1, x_2, \ldots

4.1.4 Aprenentage del perceptró

Ara ens podem fer dues preguntes senzilles:

- Com seleccionar els valors dels pesos? Als exemples anteriors, aquests valors s'introdueixen manualment, però les xarxes neuronals no funcionen d'aquesta manera.
- Com seleccionar el valor del llindar o del biaix?

Es poden dissenyar algoritmes d'aprenentatge que puguin ajustar automàticament els pesos i els biaixos d'una xarxa de neurones artificials. Aquest ajustament es produeix en resposta als estímuls externs, sense intervenció directa d'un programador. Els algoritmes d'aprenentatge permeten a les xarxes neuronals aprendre a resoldre problemes. Però, com podem elaborar aquests algoritmes?

Suposem que tenim una xarxa de perceptors que volem utilitzar per aprendre a resoldre algun problema. Per exemple, les entrades a la xarxa poden ser els píxels d'una imatge manuscrita d'un dígit. I volem que la xarxa aprengui els pesos i els biaixos per obtenir a la seva sortida una classificació correcta del dígit. Per veure com pot funcionar l'aprenentatge, suposem que fem un petit canvi en algun pes (o biaix) a la xarxa. El que volem és que aquest petit canvi de pes causi només un petit canvi corresponent en la sortida de la xarxa (fig. 4.6). Aquesta propietat farà possible l'aprenentatge.

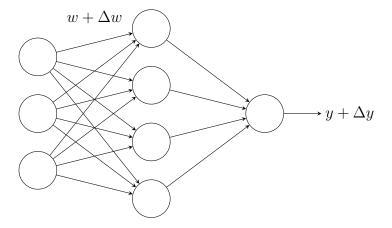


Figura 4.6: Aprentatge d'una xarxa de perceptrons

Si és cert que un petit canvi en un pes (o biaix) provoca només un petit canvi en la sortida, podem utilitzar aquest fet per modificar els pesos i els biaixos per aconseguir que la xarxa es comport de la manera que desitgem. Per exemple, suposem que la xarxa classificava erròniament una imatge com a 8 quan hauria de ser un 9. Podríem esbrinar com es pot fer un petit canvi en els pesos i els biaixos, de manera que la xarxa s'acosti una mica més a la correcta classificació de la imatge. I després repetiríem això, canviant els pesos i els biaixos una vegada darrere l'altra per aconseguir cada cop millors resultats. La xarxa estaria aprenent.

El problema és que això no passa amb perceptrons. Així doncs, un petit canvi en els pesos o biaixos d'un sol perceptró d'una xarxa fa que la sortida d'aquesta passi de 0 a 1 i viceversa, canviant el comportament de la xarxa de forma molt aleatòria, ja que el

perceptró replica una funció esglaó (fig. 4.1). Això fa que sigui difícil veure com modificar gradualment els pesos i els biaixos de manera que la xarxa s'acosti al comportament desitjat.

Podem superar aquest problema introduint un nou tipus de neurona artificial anomenada neurona sigmoide.

4.2 Neurona sigmoide

Les neurones sigmoides són similars als perceptrons, però es modificades de tal manera que els petits canvis en els seus pesos i biaixos causen només un petit canvi en la seva sortida. Aquest fet permetrà aprendre a una xarxa de neurones sigmoides.^[8]

Les neurones sigmoides es representen de la mateixa manera que els perceptrons (fig. 4.7).

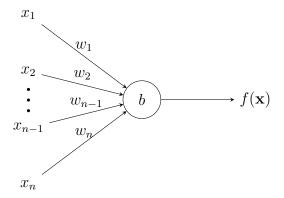


Figura 4.7: Representació d'una neurona sigmoide

Igual que un perceptró, la neurona sigmoide té diverses entrades reals, x_1, x_2, \ldots , pesos per a cada entrada, w_1, w_2, \ldots i un biaix, b. Però la seva sortida no és binària (0 o 1). En canvi, la sortida d'una neurona sigmoide és $\sigma(\mathbf{w} \cdot \mathbf{x} + b)$, on σ s'anomena funció sigmoide (eq. (4.5), fig. 4.8). La funció σ també es diu funció logística i les neurones es diuen neurones logístiques.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.5}$$

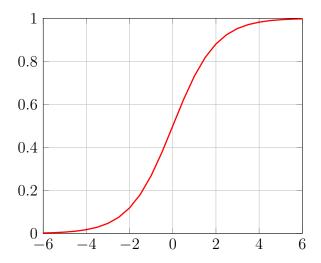


Figura 4.8: Funció sigmoide

En aquest cas, la funció sigmoide és la funció d'activació o funció de transferència de la neurona sigmoide. En el cas del perceptró, la seva funció d'activació és la funció esglaó H(x) i la sortida $f(\mathbf{x}) = H(\mathbf{w} \cdot \mathbf{x} + b)$. Per a qualsevol altra funció d'activació g(x) la sortida de la neurona serà $f(\mathbf{x}) = g(\mathbf{w} \cdot \mathbf{x} + b)$.

Per tant, la sortida d'una neurona sigmoide amb entrades $\mathbf{x} = (x_1, x_2, \ldots)$, pesos $\mathbf{w} = (w_1, w_2, \ldots)$ i biaix b és:

$$f(\mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w} \cdot \mathbf{x} + b)}}$$
(4.6)

La forma de la funció sigmoide (fig. 4.8) és una versió suavitzada de la funció esglaó (fig. 4.1). La suavitat de σ significa que petits canvis Δw_i en els pesos i Δb en el biaix produiran un petit canvi Δy en la sortida de la neurona, propietat necessària perquè la xarxa pugui aprendre.

Referències

- [1] C. Gershenson, "Artificial Neural Networks for Beginners", ag. de 2003. arXiv: cs/0308031. [En línia]. Disponible en: http://arxiv.org/abs/cs/0308031. 11.
- [2] I. Basheer i M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application", Journal of Microbiological Methods, vol. 43, núm. 1, pàg. 3-31, des. de 2000, ISSN: 01677012. DOI: 10.1016/S0167-7012(00)00201-3. [En línia]. Disponible en: http://ethosun2.unige.ch/etho5.10/pdf/basheer.hajmeer.2000.fundamentals.design.and.application.of.neural.networks.review.pdf. 11.

- [3] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms", Cornell Aeronautical Lab Inc Buffalo NY, inf. tèc., 1961. [En línia]. Disponible en: https://apps.dtic.mil/dtic/tr/fulltext/u2/256582.pdf. 11.
- [4] M. A. Nielsen, "Neural Networks and Deep Learning", 2015. [En línia]. Disponible en: http://neuralnetworksanddeeplearning.com/. [Accedit: 25-11-2019]. 12, 13, 15.
- [5] Towards Data Science, "Perceptron: The Artificial Neuron (An Essential Upgrade To The McCulloch-Pitts Neuron)", 2018. [En línia]. Disponible en: https://towardsdatascience.com/perceptron-the-artificial-neuron-4d8c70d5cc8d. [Accedit: 25-11-2019]. 12.
- [6] R. Rojas, Neural Networks A Systematic Introduction. Springer-Verlag, 1996. [En línia]. Disponible en: https://page.mi.fu-berlin.de/rojas/neural/neuron.pdf. 13.
- [7] Medium, "Build your own Neural Network The story of Perceptron & Sigmoid". [En línia]. Disponible en: https://medium.com/explore-artificial-intelligence/build-your-own-neural-network-1-the-story-of-perceptron-sigmoid-c9d1a9afdf95. [Accedit: 03-12-2019]. 13, 15.
- [8] Towards Data Science, "Sigmoid Neuron Building Block of Deep Neural Networks". [En línia]. Disponible en: https://towardsdatascience.com/sigmoid-neuron-deep-neural-networks-a4cd35b629d7. [Accedit: 08-12-2019]. 17.

Capítol 5

Disseny i programació de xarxes neuronals

La finalitat del treball és l'aplicació dels conceptes de xarxes neuronals. Es provaran xarxes neuronals simples amb diferents funcions d'activació per veure el seu comportament i la seva precisió. Com a base del projecte, s'ha utilitzat la base de dades MNIST, fig. 5.1, un conjunt d'imatges de dígits escrits a mà que són emprats per a l'entrenament de sistemes de classificació automàtica d'imatges. És l'evolució de la base de dades NIST. MNIST consta d'un conjunt de 70000 dígits manuscrits: 60000 imatges per entrenar la xarxa neuronal i 10000 imatges per mesurar la seva eficiència. [1]

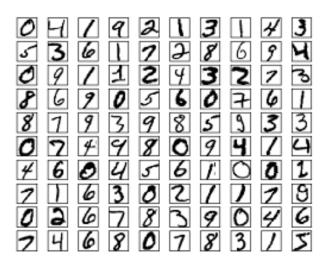


Figura 5.1: Base de dades MNIST

Pel que fa a les tecnologies i *frameworks* d'aprenentatge profund, hi ha diverses eleccions a considerar: **TensorFlow**, **Keras**, **PyTorch**:

• TensorFlow és una biblioteca de programari de codi obert per a l'aprenentatge profund i automàtic desenvolupada per l'equip Google Brain per satisfer les seves

necessitats de crear i entrenar xarxes neuronals per detectar i desxifrar patrons i correlacions. S'utilitza tant per a la investigació com per a la producció a Google i està sota la llicència Apache 2.0.^[2]

En aquest projecte s'utilitzarà TensorFlow per la seva capacitat per realitzar un bon rendiment a càlculs de nivell baix i alt. També, a causa de la naturalesa de processament d'imatges del projecte i la seva capacitat d'integració fàcil amb altres biblioteques Python com NumPy^[3] o PIL.^[4]

- Keras és una biblioteca de xarxes neuronals de codi obert escrita en llenguatge Python. Keras es pot executar damunt de TensorFlow, Microsoft Cognitive Toolkit (CNTK) o Theano. Va ser desenvolupat per l'experimentació en xarxes neuronals. Els seus advantages són la facilitat d'ús, modularitat i extensibilitat.^[5]
- **PyTorch** és una biblioteca de programari de codi obert per a l'aprenentatge automàtic, basada en la biblioteca Torch i escrita en Python, C++ i CUDA. És desenvolupada principalment pel grup de recerca d'intel·ligència artificial de Facebook. ^[6]

Com ja s'ha indicat anteriorment, un dels objectius del treball és aconseguir un valor de precisió el més proper al 100% possible. Això s'aconsegueix mitjançant la configuració ideal de les variables com ara els pesos o els biaixos.

L'activitat consisteix en programar una xarxa neuronal en la qual la sortida generada sigui el valor del número que la xarxa ha obtingut com a entrada però en format d'imatge. Ès a dir, la xarxa ha d'endevinar quin es el número escrit a mà. Com més encerti, obtindrà un valor de precisió o més elevat.

Per aconseguir aquests valors, es diferencien dues fases de modelització

• Fase d'entrenament. En aquesta fase s'utilitza una part del conjunt de dades (MNIST en aquest cas) per determinar els pesos que defineixen el model de la xarxa neuronal. Aquests pesos es calculen de manera iterativa, amb l'objectiu de minimitzar l'error obtingut entre la sortida resultant de la xarxa i el valor de sortida desitjat.

Un altre concepte que s'utilitza en l'entrenament consisteix en la retropropagació o backpropagation. [7] És un metode de càlcul de pesos que s'utilitza en algoritmes d'aprenentatge i que ofereix molt bons resultats gràcies al reajustament dels pesos de les neurones en direcció contrària després d'haver recorregut la xarxa des del principi fins al final.

Durant aquesta fase, el model pot patir un *sobreentrenament*, de manera que s'ajusta massa a les particularitats del patró d'entrenament. En aquest cas es perdria l'habilitat de generar el seu aprenentatge en nous casos amb altres dades.

• Fase de prova. Per evitar el problema del sobreentrenament, s'utilitza un segon grup de dades diferents de les d'entrenament per permetre controlar el procés d'aprenentatge.

Aquest grup de dades s'utilitza per veure com reacciona el model davant dades que no ha analitzat anteriorment i es pot saber quin és el seu grau de precisió i el seu grau d'error.

El primer pas en el desenvolupament del projecte consisteix en crear una xarxa neuronal bàsica d'una sola capa. Aquesta vindria representada per 784 unitats d'entrada, que corresponen al nombre de píxels que conté la imatge del model de dades (24x24 píxels), i 10 sortides, que representen els deu possibles dígits [0-9] de la imatge d'entrada.

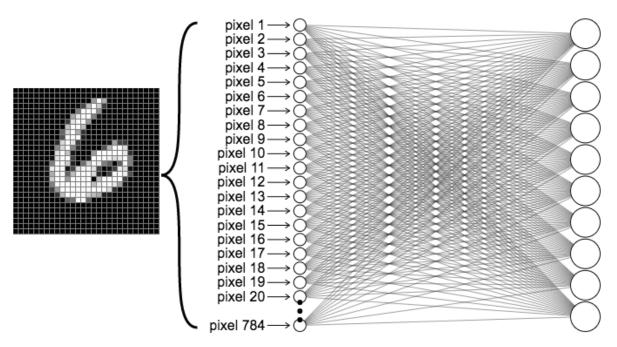


Figura 5.2: Xarxa bàsica per MNIST^[8]

5.1 Entorn

Primer de tot, cal destacar l'entorn dels experiments, tant el hardware com el software:

• CPU: Intel Core i7-8750H

• GPU: NVIDIA GeForce GTX 1060

• RAM: 16 GB DDR4

• OS: Windows 10

- Python. És un dels llenguatges de programació més utilitzats per experiments amb IA. És un llenguatge multiplataforma, orientat a objectes i fàcil d'aprendre gràcies a la seva sintaxi. Ha estat escollit per la gran quantitat de biblioteques, tipus de dades i funcions incorporades en el mateix llenguatge, que ajuden a realitzar moltes tasques sense necessitat de programar-les de zero.^[9]
- **TensorFlow**. Un dels *frameworks* més utilitzats per realitzar la gestió de xarxes neuronals.^[10]
- **Keras**. S'executa per damunt de TensorFlow. Biblioteca de Python que proporciona, d'una manera simple i eficient, la creació de models de xarxes neuronals.^{[5],[10]}
- Anaconda. És una distribució de codi obert dels llenguatges de programació Pvthon. [11]
- Jupyter Notebook. És un entorn computacional interactiu basat en la web per crear documents de quaderns Jupyter (notebooks) en llenguatge Python.^[12]

5.2 Xarxa neuronal

El primer pas realitzat consisteix en carregar el model de dades MNIST explicat anteriorment per a poder tractar-lo. Keras ja incorpora el conjunt de dades MNIST i només cal importar-lo i guardar-lo en les variables d'entrenament i test. Un cop fet, només cal preprocessar correctament les dades carregades d'aquest conjunt, com es mostra en el listing 5.1.

En aquest punt, tenim 60000 mostres de dígits del 0 al 9, escrits a mà, en una variable llesta per ser tractada per la xarxa que es crea a continuació.

```
import tensorflow as tf
   from tensorflow.keras.models import Sequential
   from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
   from tensorflow.python.keras.utils import to_categorical
10
   from tensorflow.python.keras.datasets import mnist
11
   (X train, y train), (X test, y test) = mnist.load data()
12
   X_train = X_train reshape(X_train shape[0], 784)
   X_test = X_test.reshape(X_test.shape[0], 784)
   X_train = X_train.astype('float32')
17
   X test = X test.astype('float32')
19
   X train /= 255
20
   X_test /= 255
^{21}
22
   Y_train = to_categorical(y_train, 10)
23
   Y test = to categorical(y test, 10)
24
```

Listing 5.1: Importar MNIST i preprocessar les dades

El següent pas consisteix en la creació del model de la xarxa (listing 5.2). Primerament, s'ha creat una xarxa amb estructura bàsica on les entrades es connecten directament amb les 10 neurones de sortida (una per cada dígit del 0 al 9, fig. 5.3). Seguidament es fa la compilació i lentrenament de la xarxa.

Listing 5.2: Model de xarxa neuronal

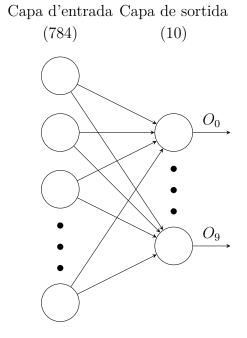


Figura 5.3: Xarxa neuronal

El model creat representa un model sequencial amb 784 entrades per una capa de 10 sortides i una funció d'activació sigmoide presentada anteriorment.

Fins ara, hem aconseguit construir una xarxa neuronal capaç d'identificar dígits del 0 al 9 amb un percentatge d'encert prou alt. La figs. 5.4 and 5.5 mostra els resultats obtinguts mitjançant aquesta xarxa. Es poden observar dos gràfics que representen els valors accuracy (precisió) i loss (error) respecte les epochs, és a dir, el percentatge d'encert i el valor d'error respecte a cada iteració del model.

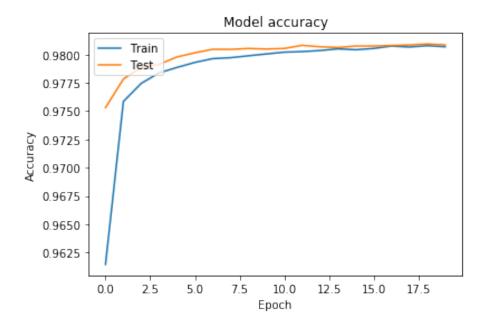


Figura 5.4: Precisió de la xarxa

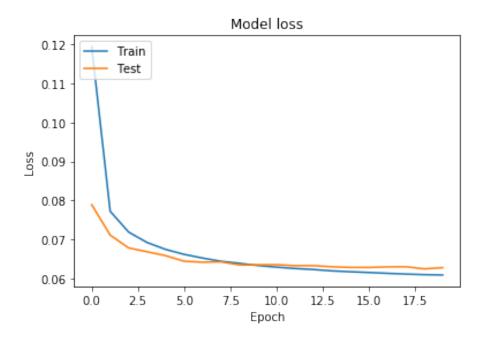


Figura 5.5: Error de la xarxa

Les modificacions posteriors de la xarxa neuronal amb la finalitat d'aconseguir més eficiència i l'applicació amb GUI (figs. 5.6 and 5.7) es poden trobar a [13].

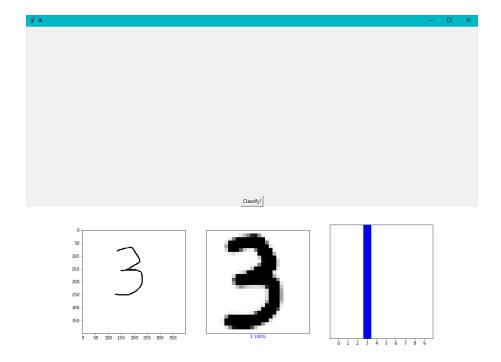


Figura 5.6: Applicació GUI

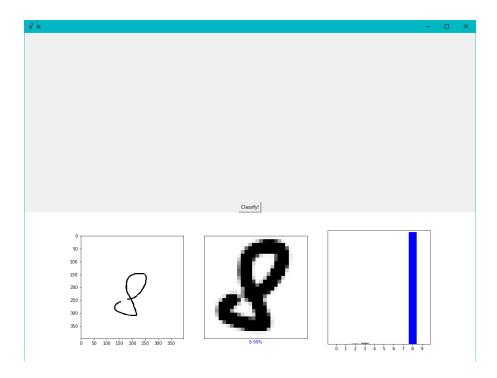


Figura 5.7: Applicació GUI

Referències

- [1] C. B. Yann LeCun, Corinna Cortes, "MNIST handwritten digit database". [En línia]. Disponible en: http://yann.lecun.com/exdb/mnist/. 20.
- [2] TensorFlow, "TensorFlow". [En línia]. Disponible en: https://www.tensorflow.org/. 21.
- [3] NumPy, "NumPy NumPy". [En línia]. Disponible en: https://numpy.org/. [Accedit: 08-12-2019]. 21.
- [4] Pillow, "Pillow Pillow (PIL Fork) 6.2.1 documentation". [En línia]. Disponible en: https://pillow.readthedocs.io/en/stable/. [Accedit: 08-12-2019]. 21.
- [5] Keras, "Keras Documentation". [En línia]. Disponible en: https://keras.io/. 21,23.
- [6] PyTorch, "PyTorch". [En línia]. Disponible en: https://pytorch.org/. 21.
- [7] Medium, "Neural networks and back-propagation explained in a simple way". [En línia]. Disponible en: https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e. [Accedit: 08-12-2019]. 21.
- [8] Ml4a, "Looking inside neural nets". [En línia]. Disponible en: https://ml4a.github.io/ml4a/looking%7B%5C_%7Dinside%7B%5C_%7Dneural%7B%5C_%7Dnets/. [Accedit: 08-12-2019]. 22.
- [9] Python, "Welcome to Python.org". [En línia]. Disponible en: https://www.python.org/. [Accedit: 08-12-2019]. 23.
- [10] Towards Data Science, "Top 10 Best Deep Learning Frameworks in 2019 Towards Data Science". [En línia]. Disponible en: https://towardsdatascience.com/top-10-best-deep-learning-frameworks-in-2019-5ccb90ea6de. [Accedit: 08-12-2019]. 23.
- [11] Anaconda, "Anaconda | The World's Most Popular Data Science Platform". [En línia]. Disponible en: https://www.anaconda.com/. [Accedit: 08-12-2019]. 23.
- [12] Jupyter, "Project Jupyter | Home". [En línia]. Disponible en: https://jupyter.org/. [Accedit: 08-12-2019]. 23.
- [13] Stepiii, "Stepiii/ai". [En línia]. Disponible en: https://github.com/Stepiii/ai. 26.

Capítol 6

Conclusió

Com hem pogut observar, les xarxes neuronals proporcionen un gran potencial davant la resolució de problemes de predicció i classificació. En aquest cas, s'ha construït un model capaç de resoldre un problema relativament senzill per un ésser humà, però assolint amb un percentatge d'encert molt elevat.

Tenint en compte que el problema resolt es considera senzill i introductori, la seva resolució il·lumina la idea sobre la diversitat dels problemes que es poden arribar a solucionar mitjançant l'aplicació de les xarxes neuronals en qualsevol context.

Tot i això, les aplicacions que es poden extrapolar del model generat podrien arribar a ser molt útils i amb una bona acceptació comercial.

Pel que fa als objectius fixats al començament del projecte, s'assoleix completament el propòsit principal de dissenyar i programar un model basat en xarxes neuronals capaç de reconèixer dígits escrits a mà amb una eficiència elevada.

Aquest treball ha complert definitivament els seus objectius, no només cap a la familiarització amb conceptes de la intel·ligència artificial i aprenentatge automàtic aplicats al processament d'imatges, que és un camp que evoluciona constantment, sinó també s'ha après a dissenyar models de xarxes neuronals per al processament i reconeixement d'imatges.

Afrontar problemes comuns en aquest àmbit com ara triar el framework més adequat, triar el mètode d'optimització més eficaç, tractar les dades abans i després del processament, intentar esbrinar el millor mètode d'anàlisi per als resultats, solucionar problemes d'execució derivats de mal funcionament o optimitzar el model, són habilitats útils i de difícil adquisició.

• Hem aprendre conceptes que poden ser aplicats en diferents àrees com l'anàlisi

d'imatges o presa de decisions.

- Explicar el funcionament de diferents mètodes d'aprenentatge automàtic (centrantse principalment en les xarxes neuronals).
- Demostrar que els sistemes d'aprenentatge automàtic són útils i aplicables.
- Comprendre els conceptes, el funcionament i l'abast de les xarxes neuronals i mostrar el procés de disseny i implementació de xarxes neuronals amb TensorFlow i Keras per aconseguir objectius específics.

Reconeixement d'imatges (visió artificial):

- Dissenyar i implementar un model predictiu eficaç en la classificació de dígits escrits a mà utilitzant el conjunt de dades MNIST.
- Analitzar el funcionament i el rendiment de diferents arquitectures de xarxes neuronals.

Apèndixs

Apèndix A

Desenvolupament del treball

En el primer bloc de la memòria s'ha explicat la part més teòrica del treball que consta de 3 apartats: Intel·ligència artificial, Aprenentatge automàtic i Xarxes neuronals artificials. Aquesta part del treball inclou la recerca d'informació i, quan sigui possible, alguns exemples pràctics per complementar la teoria. En la recerca bibliogràfica s'han utilitzat diverses fonts:

- Com és natural, la font d'informació que més he utilitzat, la que predominarà en la part teòrica del treball i la que m'ha clarificat més conceptes ha estat la recerca amb els buscadors d'Internet. Per aquesta font he aconseguit molts treballs relacionats amb la intel·ligència artificial i l'aprenentatge automàtic. Entre els diferents treballs, destaquen els treballs de fi de grau (TFG) i els projectes de final de carrera (PFC) en l'àmbit de les Tecnologies de la Informació i les Comunicacions, sobretot l'Enginyeria Informàtica. Principalment, tota la informació d'aquesta font és en idioma anglès i extreta dels repositoris o dipòsits digitals de les universitats, com ara la UPF^[1], la UAB^[2] o la UPC.^[3]
- A parts dels repositoris, he fet servir bases de dades d'autors i d'articles publicats a revistes científiques internacionals, amb l'ajuda del motor de cerca Google Acadèmic. M'agradaria destacar algunes pàgines web que m'han resultat molt útils:
 - Google Acadèmic és un motor de cerca de Google que indexa el text complet o les metadades de literatura científico-acadèmica de gran quantitat de formats i disciplines.^[4]
 - Microsoft Academic és un motor de cerca web públic i gratuït per a publicacions acadèmiques, desenvolupat per Microsoft Research.
 - ResearchGate és una xarxa social i una eina de col·laboració per a científics i investigadors per compartir documents, fer i respondre preguntes, i trobar col·laboradors.^[6]

- arXiv és un repositori científic per a la publicació d'articles científics en format digital en els camps de les matemàtiques, física, informàtica i biologia que poden ser obtinguts lliurement a través d'Internet.^[7]
- Gran part de la informació per dur a terme el disseny i la programació dels diferents models de xarxes neuronals en la part pràctica del treball són de fonts lliures. Això vol dir que aquesta informació s'ha obtingut de fòrums, blocs i pàgines web amb tutorials on els autors exposen, expliquen i mostren com fer determinades coses, sense un ànim de lucre. Comparteixen els seus mètodes i les seves conclusions de forma desinteressada. Això implica provar i verificar molts dels mètodes per assegurar el seu funcionament. Per tant, són referències a pàgines web actives i que poden modificar-se i desaparèixer en el temps. Concretament he utilitzat:
 - GitHub és un servei de hosting de repositoris Git, el qual ofereix tota la funcionalitat de Git de control de revisió distribuït i administració de codi de la font (SCM) així com afegint les seves característiques pròpies.^[8]
 - Stack Overflow és una pàgina web de preguntes i respostes per a programadors professionals i entusiastes que forma part de la Xarxa Stack Exchange.
 - Tutorials Point és una pàgina web que ofereix una gran quantitat de tutorials, cursos i articles gratuïts relacionats amb temes de llenguatges de programació i altres tecnologies.^[10]
 - Medium és un servei de publicació de blogs.^[11]

Referències

- [1] Universitat Pompeu Fabra, "e-Repositori UPF". [En línia]. Disponible en: https://repositori.upf.edu/. 32.
- [2] Universitat Autònoma de Barcelona, "Dipòsit Digital de Documents de la UAB". [En línia]. Disponible en: https://ddd.uab.cat/. 32.
- [3] Universitat Politècnica de Cataluya, "Pàgina inicial de UPCommons". [En línia]. Disponible en: https://upcommons.upc.edu/. 32.
- [4] Google, "Google Scholar". [En línia]. Disponible en: https://scholar.google.com/. 32.
- [5] Microsoft, "Home | Microsoft Academic". [En línia]. Disponible en: https://academic.microsoft.com/home. 32.
- [6] ResearchGate, ResearchGate | Find and share research. [En línia]. Disponible en: https://www.researchgate.net/. 32.

- [7] ArXiv.org, "arXiv.org e-Print archive". [En línia]. Disponible en: https://arxiv.org/. 33.
- [8] GitHub, "The world's leading software development platform · GitHub". [En línia]. Disponible en: https://github.com/. 33.
- [9] Stack Overflow, "Stack Overflow Where Developers Learn, Share, & Build Careers". [En línia]. Disponible en: https://stackoverflow.com/. 33.
- [10] Tutorials Point, "Tutorials Point The Biggest Online Tutorials Library". [En línia]. Disponible en: https://www.tutorialspoint.com/index.htm. 33.
- [11] Medium, "Medium Get smarter about what matters to you." [En línia]. Disponible en: https://medium.com/. 33.

Apèndix B

Programari utilitzat: LATEX

Tot el treball està escrit utilitzant LATEX com a processador de textos. Es la primera vegada que utilitzo aquest programa i ha requerit un esforç inicial extra per aprendre'l. En general, crec que ha valgut la pena fer aquest esforç, ja que la presentació final queda molt millor que si estigués fet amb els programes habituals per processar textos, com ara el Word. A part, aquest esforç es veurà recompensat, ja que en un futur podré utilitzar el LATEX per altres treballs, ja que presenta molts avantatges, en especial pel que fa al format de fórmules i la gestió de tot el contingut.

El LATEXés programari lliure i és molt utilitzat en l'àmbit acadèmic, especialment a les universitats en els àmbits relacionats amb les ciències (matemàtiques, informàtica, física, ...). Va ser elaborat per L. Lamport a partir del llenguatge TeX inventat per D. Knuth.^[1]

Dins dels avantatges d'aquest programari vull destacar els següents.

- Format general. Permet centrar-se exclusivament en el contingut sense preocuparse en els detalls de format, que es gestionen de manera general i es poden modificar en qualsevol moment de manera rapida.
- Estructura general. Permet estructurar fàcilment el document en capítols, seccions, bibliografia, índex..., amb comptadors i numeració automàtica.
- Projecte-arbre. Es pot crear un fitxer general que crida a altres fitxers, de manera que crea un arbre amb l'estructura dels diferents capítols. Així es poden organitzar els fitxers per capítols, i només et cal obrir el capítol en el qual estàs treballant. Això va bé perquè no cal compilar cada cop tot el treball.
- Automatització de l'índex general i de l'índex de figures.
- Fórmules. Aquest programa ja est'a pensat per tal de poder-hi introduir fórmules matemàtiques. Això és un gran avantatge, ja que es poden escriure tota mena de fórmules per ser reproduïdes amb una tipografia professional.

• Taules i figures. Permet crear taules i afileraments amb molts paràmetres possibles. També permet incloure figures de diferents formats. Tant les taules com les figures poden ser definits com a elements flotants per facilitar la seva distribució a les pàgines, la seva enumeració automàtica, el text de peu de figura associat i el llistat final de l'índex de figures.

Tot i que com es veu és un programa amb un gran potencial, té també alguns inconvenients. El principal és que cal programar instruccions i, per tant, requereix un temps i un esforç d'aprenentatge. En l'elaboració concreta, les dificultats amb què m'he trobat són les següents:

- Tot i que està preparat a escala mundial, he tingut alguns problemes amb els accents.
- La inclusió d'imatges no és tan directa com en altres programes, i ha requerit instal·lar algun paquet addicional.
- Cal anar compilant per veure com queda, ja que no és WYSIWYG (What You See Is What You Get)
- Com que és un programa, la sintaxi ha de ser estrictament correcta, sovint d'ona errors i no es pot compilar. No sempre és fàcil veure l'origen de l'error per poder corregir-lo.

Referències

[1] LaTeX, "LaTeX - A document preparation system". [En línia]. Disponible en: https://www.latex-project.org/. 35.