

Università degli Studi di Torino
Dipartimento di Economia e Management
Corso di Laurea in Economia Aziendale

DISSERTAZIONE FINALE

Apprendimento profondo applicato alla Sentiment Analysis

Candidato
Tommaso Bassignana
n. matricola: 819060

Relatore:
Prof. Luigi Bollani

a.a. 2017-2018

INDICE

Premessa	5
1. Analisi del sentimento	5
2. Intelligenza artificiale	8
3. L'apprendimento automatico	10
4. L'apprendimento profondo	12
5. Le reti neurali	15
6. Un caso pratico	18
7. Conclusioni	22

PREMESSA

La presente dissertazione esamina i principali ambiti di studio concernenti l'utilizzo dell'apprendimento profondo applicato all'analisi del sentimento, ponendosi l'obiettivo di mettere in luce storia e differenze tra metodologie tradizionali e moderne.

I concetti teorici sono stati utilizzati per l'elaborazione di un modello classificatore¹ della polarità di un esteso campione di recensioni cinematografiche, nel tentativo di sintetizzare un algoritmo capace di aiutare a classificare le risposte aperte fornite nelle indagini di opinione.

1. L'ANALISI DEL SENTIMENTO²

George R.R. Martin scrive "...un lettore vive mille di vite prima di morire, un uomo che non legge mai vive una vita sola". A tanto può arrivare l'impatto della carta stampata, capace di farci immedesimare coi passanti tra le piccole strade dei quartieri popolari di Parigi tratteggiate da Zola, o a farci sentire lordi della polvere di rena rossa cara a Verga.

Nella vita quotidiana, spesso i testi sono strumentali alla ricerca o alla trasmissione di informazioni e numerose sono le mansioni che ne richiedono un utilizzo esteso, risultando tediose per gli individui ed economicamente svantaggiose per le imprese. Per porre rimedio a questa condizione, nell'ultimo ventennio molte

¹ Codice dell'algoritmo nell'appendice A.

² Per la stesura del presente capitolo ci siamo avvalsi dello studio dei seguenti autori:

M. TABOADA, J. BROOKE, *Lexicon-based methods for sentiment analysis*, in "Computational Linguistics", n° 37, 2011;

Ł. AUGUSTYNIAK, P. SZYMAŃSKI, T. KAJDANOWICZ, W. TULIĞŁOWICZ, *Comprehensive Study on Lexicon-based Ensemble Classification Sentiment Analysis*, in "Entropy", n° 18, 2015;

M. THELWALL, K. BUCKLEY, G. PALTOGLOU, D. CAI, A. KAPPAS, *Sentiment strength detection in short informal text*, in "Journal of the American Society for Information Science and Technology", n° 61, 2010.

risorse sono state impiegate per cercare di automatizzare alcune operazioni riguardanti la comprensione o la manipolazione di un testo. Lo spettro di casistiche interessate da questa ricerca è molto ampio, dal parafrasare e riassumere al produrre e correggere, fino ad arrivare all'analisi del significato di un documento. Numerose innovazioni sono state finanziate da enti pubblici ma soprattutto da imprese private, bisognose di strumenti per gestire la crescente mole di dati e il rapporto continuo con i clienti.

L'analisi del sentimento è uno di questi strumenti che, grazie allo studio del linguaggio, fornisce la polarità – anche detta sentimento – rappresentante l'opinione generale sul soggetto della frase o del testo, che si può dedurre dalla struttura grammaticale e logica dello scritto. Generalmente la polarità è classificata come negativa, neutra o positiva. Esemplicando: “Questo gelato è buonissimo” ha una polarità positiva, mentre “Il film non è bello ma neanche brutto” ha una polarità neutra.

Vi sono due grandi famiglie di metodi per svolgere tali analisi: il *lexicon based approach* e l'utilizzo dei metodi statistici. Il *lexicon based approach* consiste nel suddividere l'intero corpo del testo in parti più piccole, dette *token*, ossia segmenti di frasi o singole parole. Successivamente, si effettua la sommatoria del numero di volte in cui un token ricorre e gli si assegna un valore di polarità. Aggettivi come “bellissimo” e “fantastico” avranno un valore alto sulla scala prescelta, le parole neutre si attestano su valori mediani e quelle negative su valori tendenti allo zero. Per elaborare i token, vi sono due strade percorribili, il rilevamento delle *keyword* e lo studio delle affinità lessicali.

Il rilevamento delle *keyword*, utilizzato esclusivamente in casistiche semplici, consente di classificare il testo grazie all'individuazione di parole emotivamente significative non ambigue come “felice” o “triste”. Esse sono emotivamente significative poiché esprimono un giudizio e non ambigue in quanto prive di significati alternativi. Il rilevamento delle *keyword* non ha requisiti particolari in termini di

risorse ma ha il limite di doversi avvalere di un dizionario precostituito e discernere unicamente periodi molto semplici.

Il metodo di affinità lessicale parte dai risultati sopra descritti ma aggiunge, in modo arbitrario, un'affinità probabile tra parole con polarità simile, migliorando in tal modo la selezione delle keyword anche in contesti più complessi. Le risorse necessarie a implementare questa analisi sono minime ma l'arbitrarietà della scelta delle affinità lessicali può diventare un elemento di criticità in quanto dipendente, di volta in volta, dall'esperienza del creatore dell'algoritmo di classificazione.

L'utilizzo di metodi statistici è la strada più complessa ma più accurata in quanto si fa spesso uso di metodologie di apprendimento profondo. Questa opzione richiede un grande fabbisogno di risorse, sia in termini computazionali sia in termini di dati necessari per la corretta costituzione del modello classificatore ma presenta un'accuratezza maggiore dovuta all'abbandono del dizionario precostituito, sostituito dalla rappresentazione astratta delle frasi stesse prodotta dall'algoritmo. Questo processo riesce a superare alcuni ostacoli tipici delle analisi sui corpi di testo come la presenza di sarcasmo, periodi lunghi o rimandi a parti del testo separate tra loro. I metodi tradizionali classificherebbero la frase "La macchina mi ha lasciato a piedi, fantastico!" tra quelle positive, mentre un modello statistico correttamente costruito ha un'elevata probabilità di individuare casi limite come il sarcasmo o l'ironia. La comprensione di questi metodi è vincolata ad alcune considerazioni riguardo ambiti di studio più ampi, che verranno esaminati nei prossimi capitoli.

2. INTELLIGENZA ARTIFICIALE³

Intelligenza artificiale è un termine ampio che necessita in questa trattazione di un chiarimento sulla sua accezione. Per intelligenza artificiale si intende l'insieme dei metodi matematico-statistici attualmente utilizzati per risolvere problemi in vari ambiti tra cui knowledge reasoning, planning, natural language processing, computer vision e robotics. Con intelligenza artificiale generale, invece, si indica la possibilità di creare un algoritmo capace di emulare fedelmente il pensiero umano.

Nel diciassettesimo secolo Leibniz, Hobbes e Descartes ipotizzarono che i pensieri umani fossero riproducibili con la stessa chiarezza con cui, per esempio, vengono rappresentate le formule matematiche, dando vita al primo sforzo concettuale per la realizzazione di un'idea verso la quale l'uomo è stato sempre attratto: attribuire caratteristiche antropomorfe a oggetti che non dovrebbero, per loro natura, essere animati o godere di coscienza; si pensi al Golem presente nella tradizione ebraica o al più recente Frankenstein di Mary Shelley.

All'inizio degli anni Cinquanta del secolo scorso, alcuni tra scienziati e matematici sostennero che gli avanzamenti della scienza avrebbero permesso di approssimare una rappresentazione informatica del cervello capace di agire specularmente al suo corrispettivo biologico. Questa dunque è l'idea di intelligenza artificiale generale: una struttura in grado di riflettere le capacità di pensiero e ragionamento umano.

Durante i primi tentativi però ci si è resi conto che per arrivare a tale scopo fosse necessario creare programmi per risolvere problemi semplici in un contesto

³ Per la stesura del presente capitolo ci siamo avvalsi dello studio dei seguenti autori:
R. NEAPOLITAN, X. JIANG, *Artificial Intelligence: With an Introduction to Machine Learning*, London, Chapman & Hall/CRC, 2018;
D. POOLE, A. MACKWORTH, *Artificial Intelligence: Foundations of Computational Agents* (2nd ed.), New York, Cambridge University Press, 2017;
D. CREAVIER, *AI: The Tumultuous Search for Artificial Intelligence*, New York, BasicBooks, 1996.

limitato, per esempio svolgere compiti di classificazione o elaborare strategie in situazioni a scelta multipla come nel gioco degli scacchi.

Possiamo paragonare i vari algoritmi a macchinari industriali, creati per compiere determinate operazioni con prestazioni migliori rispetto alla forza lavoro umana. Così come i macchinari sono solamente in grado di svolgere un numero limitatissimo di operazioni, anche gli algoritmi di intelligenza artificiale, fino ad ora, si limitano alla risoluzione di problemi molto specifici, come l'analisi del testo o il riconoscimento vocale.

Non vi è la certezza, ad oggi, di riuscire a produrre un algoritmo che riesca a mimare totalmente i processi cognitivi umani. Dalle ricerche in corso emergono molti elementi che fanno supporre come sia necessario un lungo lasso di tempo per disporre della tecnologia utile a creare un'intelligenza generale simile a quella ipotizzata dalla futurologia, branca delle scienze sociali che studia applicazioni e implicazioni di tecnologie non ancora sviluppate.

Gli ambiti compresi sotto l'ombrello dell'intelligenza artificiale sono destinati a una rapida evoluzione, in quanto i processi matematico-statistici che li governano consentono di sfruttare trasversalmente le innovazioni nei vari campi di ricerca. Per esempio gli avanzamenti nell'ambito della computer vision hanno portato a progressi anche nella robotica grazie all'elaborazione innovativa dei segnali ricevuti come input dalla macchina.

Dei diversi aspetti riguardanti l'intelligenza artificiale, si passa ora a esaminare unicamente l'apprendimento automatico in quanto funzionale allo svolgimento dell'analisi del sentimento oggetto di questa dissertazione.

3. L'APPRENDIMENTO AUTOMATICO⁴

L'apprendimento automatico, o più comunemente *machine learning* è un sottoinsieme dell'intelligenza artificiale.⁵

Con il termine apprendimento si indica la capacità, attraverso l'utilizzo di tecniche statistiche, di migliorare progressivamente le performance in una data operazione.

In questo ambito si studia la creazione di algoritmi che sono in grado di modificarsi in base all'esattezza delle loro previsioni. Attraverso la creazione di modelli generati dai dati di input, si lascia all'algoritmo l'estrazione di classi di modalità significative generando un modello in grado di migliorare l'operazione di classificazione mano a mano che vengono introdotti nuovi dati. Essi pertanto superano l'importante limitazione di dover prevedere tutte le modalità possibili degli attributi presi in esame, operazione questa molto complessa, se non impossibile.

Vi sono due grandi categorie di metodi di apprendimento automatico in funzione della presenza o meno di feedback che coadiuvano l'algoritmo. La prima macro-area è chiamata *supervised learning* poiché gli esempi di input forniti al calcolatore sono associati all'output atteso.

In ambienti dinamici, come per esempio quello delle vetture a guida autonoma, i dati di input consistono in premi o punizioni in base alla performance dell'algoritmo e si parla di *reinforcement learning*.

Quando i feedback sono parziali parliamo di *semi-supervised learning*, mentre parliamo di *active learning* quando l'algoritmo deve anche ottimizzare una risorsa scarsa.

La seconda macro area viene denominata *unsupervised learning*: in questo caso nessun feedback è fornito all'algoritmo. Questa metodologia può essere

⁴ Per la stesura del presente capitolo ci siamo avvalsi dello studio dei seguenti autori:
A. ETHEM, *Introduction to Machine Learning*, Cambridge, MIT Press, 2004;
T. HASTIE, R. TIBSHIRANI, J. FRIEDMAN, *The Elements of Statistical Learning*, Berlin, Springer, 2001.

⁵ Appendice B, immagine 1, organizzazione degli ambiti di studio.

utilizzata per scoprire correlazioni nascoste tra i dati o per imparare features aggiuntive.

Dalle definizioni si può desumere che i compiti di classificazione sono generalmente risolti tramite una metodologia di supervised learning: gli input sono divisi in una o più classi e l'algoritmo deve essere in grado di generare un modello capace di assegnare nuovi input a una o più delle classi fornite.

Nei problemi di clustering, viceversa, gli input devono essere divisi in gruppi originariamente sconosciuti, risultando tipicamente un'operazione di unsupervised learning.

All'interno dell'ambito dell'apprendimento automatico, negli ultimi anni gli algoritmi di apprendimento profondo si sono rivelati l'approccio migliore a un discreto numero di problemi come l'analisi del sentimento.

4. L'APPRENDIMENTO PROFONDO⁶

I modelli di apprendimento profondo sono algoritmi che presentano la caratteristica peculiare di usare una sequenza di *layer* per l'estrazione delle modalità, durante questa operazione essi generano rappresentazioni che corrispondono a diversi livelli di astrazione.

Il termine *deep learning* è stato introdotto da Rina Dechter nel 1986, proposto sulla base degli studi di Alexey Ivakhnenko datati 1965. Durante gli anni Ottanta gli algoritmi di deep learning vennero applicati a vari ambiti tra cui la computer vision. Verso la fine del decennio Yann LeCun implementò l'algoritmo di *backpropagation* nel processo di ottimizzazione dell'apprendimento profondo. La retropropagazione dell'errore, utilizzata ancora oggi, era già diffusa dagli inizi degli anni Settanta in ambiti estranei a quelli trattati in questa dissertazione.

Tra la fine degli anni Ottanta e l'inizio degli anni Novanta, grazie all'apprendimento profondo, fu possibile la creazione di software per il riconoscimento di numeri scritti a mano e di oggetti in uno spazio caotico. Inizialmente questi strumenti di analisi erano utilizzati solamente da grandi aziende o enti come le poste americane, prima organizzazione governativa a utilizzare questo tipo di tecnologia per leggere lo ZIP code sulle lettere.

Arrivati agli inizi degli anni Duemila i costi elevati e la difficoltà di creare algoritmi capaci di generalizzare le loro previsioni su nuovi dati fecero propendere gli attori economici all'utilizzo di programmi creati e adattati a ogni compito specifico. In questo periodo la divisione tra i vari metodi di risoluzione dei problemi di classificazione era massima, in quanto da un lato vi erano le esigenze delle imprese – vincolate a ottimizzare le risorse e a ottenere risultati immediati – e dall'altro

⁶ Per la stesura del presente capitolo ci siamo avvalsi dello studio dei seguenti autori:
Y. LECUN et al., *Backpropagation Applied to Handwritten Zip Code Recognition*, in "Neural Computation", 1989, pp. 541-551;
J. SCHMIDHUBER, *Deep Learning in Neural Networks: An Overview*, in "Neural Networks", n° 69, 2015, pp. 85-117.

erano collocati i ricercatori che puntavano allo sviluppo di algoritmi capaci di adattarsi a una moltitudine di situazioni diverse.

Tale conflitto spiega la genesi delle difficoltà odierne da parte delle imprese di implementare questa tecnologia. Manca una chiara distinzione tra l'ingegnere informatico, che utilizza conoscenze pregresse adattandole alle esigenze della singola azienda, e il ricercatore, che punta a migliorare gli algoritmi esistenti o a crearne di nuovi nel tentativo di generalizzare il più possibile. Nella quasi totalità dei casi, un algoritmo che raggiunga livelli di accuratezza superiori al 99% è sufficiente alle esigenze di un'impresa che assumerà uno o più ingegneri non tanto per la creazione dell'algoritmo – già elaborato dai ricercatori – ma per la creazione di un software in grado di interfacciarsi con la realtà aziendale, fornendo input di qualità e output facilmente interpretabili dai decision maker aziendali.

Vi sono però casi in cui un'accuratezza del 99% è estremamente bassa, basti pensare ai tentativi di creare veicoli a guida totalmente autonoma: se per ogni centinaio di semafori anche solo uno venisse erroneamente interpretato, il risultato sarebbe disastroso. In questa circostanza intervengono i ricercatori, che hanno il compito di produrre incrementi marginali nell'accuratezza degli algoritmi.

Tornando alla metà degli anni Ottanta, dal punto di vista tecnico, le difficoltà erano di due tipi: la diminuzione del gradiente e la debole connessione temporale tra input successivi. Questi inconvenienti portavano a un guadagno di accuratezza decrescente all'aumentare delle iterazioni e a una scarsa capacità di mettere in relazione tra loro due input reciprocamente "distanti".

Miglioramenti sono stati fatti mediante l'introduzione di un tipo di rete neurale ricorrente chiamato LSTM (*long short-term memory*) per lo studio del riconoscimento vocale tra gli anni 2003 e 2005. Dieci anni dopo Google introdusse sul mercato il suo riconoscimento vocale basato su questa tecnologia che riscontrò un miglioramento del 49% rispetto ai metodi precedentemente utilizzati.

L'apprendimento profondo risulta oggi la tecnologia dominante nei settori del riconoscimento vocale, di immagini e computer vision.

Per quanto riguarda l'hardware, l'innovazione più rilevante è avvenuta durante il 2009, quando Nvidia sviluppò delle GPU's (*graphics processing units*) ottimizzate per le operazioni con vettori e matrici coinvolte nell'apprendimento profondo, riducendo i tempi di “allenamento” dell'algoritmo da settimane a giorni. A oggi le alternative per chi vuole utilizzare l'apprendimento profondo sono due: affittare server ottimizzati messi a disposizione da aziende come Google e Amazon o assemblare dispositivi dedicati.

5. LE RETI NEURALI⁷

«Cogito ergo sum». Con queste parole il filosofo e matematico francese Cartesio ha riassunto la certezza indubitabile che l'uomo ha di se stesso in quanto individuo pensante. La nostra specie ha fatto del pensiero, del ragionamento, il suo vantaggio competitivo, lungo tutta la traiettoria della propria esistenza. Dalla lotta per la sopravvivenza alle mansioni più triviali l'intelletto è in grado di adattare alcune proprietà fondamentali a una serie smisurata di compiti.

L'utilizzo dell'astrazione – passando per la gestione di modelli e simboli attraverso l'iterazione e la ricorsione, fino ad arrivare al dialogo con altre menti pensanti – sono stati gli strumenti utilizzati per modellare la nostra realtà.

L'uomo ha sempre creato oggetti in grado di superare limiti fisici e cognitivi: si pensi ai sistemi di leve e carrucole o all'invenzione della scrittura. Percorrendo le tappe della storia si è giunti, tra gli anni Quaranta e l'inizio degli anni Cinquanta del secolo scorso, a gettare le fondamenta della creazione per eccellenza: riprodurre l'oggetto stesso che produce innovazione. Tra gli approcci tentati, abbandonati e riscoperti negli ultimi anni, troviamo le reti neurali artificiali.

Una rete neurale artificiale presenta due componenti fondamentali: i layer o livelli, e i neuroni contenuti al loro interno. È sempre presente negli algoritmi il layer di input, che ha il compito di fornire i dati al layer successivo e il layer di output, avente lo scopo di contenere i risultati previsti.

Ogni neurone⁸ riceve come input dei valori \mathbf{x} e genera dei valori previsti $\hat{\mathbf{y}}$. I valori \mathbf{x} rappresentano i caratteri che descrivono un'unità statistica e ogni neurone presenta due variabili che vengono ottimizzate durante il processo di apprendimento: \mathbf{w} rappresenta il *weight* e \mathbf{b} determina il *bias* di attivazione. Durante ogni

⁷ Per la stesura del presente capitolo ci siamo avvalsi dello studio dei seguenti autori:

H. ANTON, *Elementary Linear Algebra* (5th ed.), New York, 1987;

S. HAYKIN, *Neural networks: a comprehensive foundation*, Prentice Hall, 1997;

P. WERBOS, *The Roots of Backpropagation. From Ordered Derivatives to Neural Networks and Political Forecasting*, New York, John Wiley & Sons, 1994.

⁸ Appendice B, immagine 2, struttura del neurone.

iterazione il neurone calcola la media pesata tra il vettore di input \mathbf{x} e il vettore di pesi \mathbf{w} alla quale viene aggiunto il valore del bias per arrivare al risultato \mathbf{z} . Una funzione di attivazione non lineare prende come argomento \mathbf{z} e restituisce come output $\hat{\mathbf{y}}$ che è anche l'output del neurone. Indichiamo con il pedice i la posizione del singolo neurone all'interno del layer e con l'apice l il livello stesso.

$$z_i^{[l]} = \mathbf{w}_i^T \mathbf{a}^{[l-1]} + b_i \quad \mathbf{a}_i^{[l]} = g^{[l]}(z_i^{[l]})$$

Passando all'analisi dei livelli, sostituiamo i singoli valori \mathbf{x} e $\hat{\mathbf{y}}$ con il vettore \mathbf{a} , che rappresenta la funzione di attivazione del corrispondente livello.

Le equazioni di un generico layer $l+1$ saranno dunque:

$$z_1^{[2]} = \mathbf{w}_1^T \mathbf{a}^{[1]} + b_1 \quad \mathbf{a}_1^{[2]} = g^{[2]}(z_1^{[2]})$$

$$z_2^{[2]} = \mathbf{w}_2^T \mathbf{a}^{[1]} + b_2 \quad \mathbf{a}_2^{[2]} = g^{[2]}(z_2^{[2]})$$

$$z_3^{[2]} = \mathbf{w}_3^T \mathbf{a}^{[1]} + b_3 \quad \mathbf{a}_3^{[2]} = g^{[2]}(z_3^{[2]})$$

$$z_4^{[2]} = \mathbf{w}_4^T \mathbf{a}^{[1]} + b_4 \quad \mathbf{a}_4^{[2]} = g^{[2]}(z_4^{[2]})$$

$$z_5^{[2]} = \mathbf{w}_5^T \mathbf{a}^{[1]} + b_5 \quad \mathbf{a}_5^{[2]} = g^{[2]}(z_5^{[2]})$$

$$z_6^{[2]} = \mathbf{w}_6^T \mathbf{a}^{[1]} + b_6 \quad \mathbf{a}_6^{[2]} = g^{[2]}(z_6^{[2]})$$

Per rendere efficienti i calcoli si vettorizzano le suddette operazioni creando una matrice \mathbf{W} contenente l'insieme dei vettori rappresentanti il weight tra un neurone del livello l e tutti i neuroni del livello $l+1$, un vettore \mathbf{b} contenente i bias e i vettori \mathbf{z} e \mathbf{a} .

Il flusso dei dati tra i layer è il medesimo per tutte le reti: un layer prende come input l'output del layer precedente e fornisce l'input al layer successivo.

Parte del motivo dell'abbandono di questa struttura, nel periodo chiamato “inverno” dell'intelligenza artificiale, consiste nella sua scarsa capacità di gestire dati sequenziali di cospicua dimensione. Per superare tale ostacolo si utilizza un tipo particolare di rete neurale, detta ricorrente.

Le reti neurali ricorrenti, o RNN (*recurrent neural network*), sono state create nella prima metà degli anni Ottanta e hanno iniziato ad ottenere una grande no-

torietà in quest'ultimo decennio grazie al miglioramento nella loro architettura e alle prestazioni migliori dei processori. Sono indicate nell'elaborazione di dati sequenziali essendo in grado di mantenere all'interno di ogni neurone informazioni relative all'input del neurone precedente. Dati una serie di neuroni, il secondo manterrà le informazioni rilevanti del primo, il terzo quelle del secondo e così via. Teoricamente, l'ultimo neurone dovrebbe memorizzare tutto l'array di informazioni, per esempio una frase dall'ultima parola alla prima. Accade però che con l'aumentare degli input forniti la rete tenderà a giudicare più significativi gli ultimi rispetto ai primi. Per ovviare a questo problema si utilizzano le unità LSTM all'interno del network mitigando la perdita di informazioni. Queste unità di memoria permettono alla rete di essere più accurata e sono uno dei motivi della recente notorietà di questa tecnologia.

6. UN CASO PRATICO⁹

Per esplicitare il funzionamento di questa procedura si è deciso di elaborare un algoritmo che ricade nella categoria del supervised learning, illustrando in questo modo l'uso di una rete neurale ricorrente per classificare il sentimento di un campione di recensioni cinematografiche. Il dataset utilizzato è stato scelto per la sufficiente quantità e qualità dei dati al suo interno e permette un'applicazione in contesti diversificati che travalicano il mero scopo illustrativo. Il campione è composto di un totale di 22.500 recensioni scritte da utenti reali, formate da una o più frasi e viene diviso in due set di dati: il primo detto *training dataset*, viene utilizzato dall'algoritmo per generare il modello classificatore (su queste unità statistiche avverrà l'ottimizzazione di weight e bias della rete neurale). Il secondo, chiamato *testing dataset*, viene utilizzato per testare l'efficacia dell'algoritmo che dovrà operare la classificazione su input mai ricevuti prima.

I due dataset, che derivano dallo stesso campione, avranno un'identica distribuzione di probabilità: ciò rende possibile verificare l'accuratezza dell'algoritmo e controllare la presenza di *overfitting*. Riscontrare un'accuratezza maggiore sul training dataset rispetto al testing dataset, implica la creazione di relazioni non esistenti nella totalità dei dati ma soltanto nel sottoinsieme utilizzato per la creazione del modello. Questo fenomeno è abbastanza comune e sono numerosi i modi per mitigare questo effetto; è più raro invece riscontrare il fenomeno opposto, denominato *underfitting*.

Di ogni unità statistica si utilizzano due variabili: il testo della recensione e il sentimento effettivo delle frasi, appurato da un addetto. La polarità fungerà da feedback nel processo di apprendimento.

⁹ Per la stesura del presente capitolo ci siamo avvalsi dello studio dei seguenti autori:
L. DEY, S.K. MIRAJUL HAQUE, *Opinion Mining from Noisy Text Data*, in "Proceedings of the second workshop on Analytics for noisy unstructured text data", 2008, pp.83-90;
L.RAMALHO, *Fluent Python: Clear, Concise, and Effective Programming*, Boston, O'Reilly Media, 2015;
B. CHRISTIAN, T. GRIFFITHS, *Chapter 7: Overfitting* in "Algorithms to live by: the computer science of human decisions", 6 April 2017, pp. 149–168;
S. HOCHREITER, J.SCHMIDHUBER, *Long short-term memory*, in "Neural Computation", n° 7, 1997, pp.1735-1780.

Una volta diviso il campione si procederà alla fase di *data preprocessing* durante la quale si trasformeranno i dati affinché siano utilizzabili dall'algoritmo. A questo proposito sono state fatte due operazioni: nella prima il feedback positivo o negativo è stato convertito in due classi numeriche, 0 e 1. Seguendo la logica convenzione, 0 è stato posto equivalente a un giudizio negativo e 1 a un giudizio positivo.

Nella seconda operazione si è provveduto a trasformare ogni parola in un vettore di uno spazio multidimensionale al cui interno vengono raggruppati vicini i vettori rappresentanti parole semanticamente simili. I vettori sono stati poi resi di lunghezza uniforme per favorire le operazioni di calcolo successive, mediante la tecnica di *padding* che consiste nell'aggiunta di caratteri neutri, in questo caso una serie di zeri, fino a raggiungere la lunghezza del vettore desiderata.

Successivamente sono stati scelti i layer della rete neurale.

Il primo layer prende il nome di *input layer* e ha come scopo di introdurre ogni parola di una recensione, sotto forma di vettore, in un neurone differente. Le recensioni vengono introdotte singolarmente.

Il secondo layer prende il nome di *embedding layer* e permette di creare uno spazio vettoriale continuo rappresentante la vicinanza semantica delle parole all'interno delle frasi.

Il terzo consiste nel *recurrent neural network*, del tipo Long Short Term Memory, e consente di memorizzare informazioni senza perdite dovute alla lunghezza delle frasi. In questo layer si esegue un'operazione definita *dropout* che aiuta a ridurre l'overfitting attraverso lo "spegnimento" casuale del collegamento tra alcuni neuroni della rete.

Il quarto layer è denominato *fully connected* e presenta la proprietà di avere ogni suo neurone collegato ad ogni singolo neurone del layer precedente. Dai precedenti livelli provengono una serie di *learned feature vectors* e aggiungere un livello fully connected è un modo computazionalmente economico di generare

combinazioni non lineari tra essi. In questo livello, una funzione di attivazione trasformerà valori numerici su scala reale in valori di probabilità in una scala da 0 a 1. Questi valori sono equivalenti alle due classi scelte inizialmente per classificare le recensioni. Zero o valori tendenti ad esso, indicano un sentimento negativo e uno o valori tendenti a esso, indicano un sentimento positivo. La funzione di attivazione ha una duplice importanza in quanto introduce un elemento di non linearità nel flusso dei dati che altrimenti darebbero lo stesso output di una funzione di regressione e “collabora” insieme al bias nel decidere l’attivazione o meno del neurone successivo. Ogni layer ha una sua funzione di attivazione e diversi layer possono avere diverse funzioni di attivazione. Nel nostro caso si utilizza solamente la sigmoide per la semplicità degli output che fornisce e per la sua particolare derivata, una curva di distribuzione Normale, che rende agevole i calcoli per il cambiamento dei weight e dei bias durante la fase di backpropagation. Gli output generati dalla funzione di attivazione sono quelli usati per il calcolo dell’errore rispetto all’output atteso, fornendo i valori confluenti nella funzione di errore, essa indica lo scostamento della classificazione dell’algoritmo rispetto all’output atteso.

Questi livelli compongono l’algoritmo classificatore che avrà due stadi di funzionamento, *forward propagation* e *backpropagation*. I dati di input entrano nel primo livello della rete, subiscono tutte le operazioni sopra elencate nei vari layer per giungere alla previsione dell’output da parte della rete neurale utilizzata per il calcolo dell’errore. Con la determinazione della funzione d’errore termina la fase di forward propagation che si concretizza nel processo atto a creare la funzione di errore partendo dai dati in entrata. La fase di backpropagation, comunemente detta fase di apprendimento, consiste nel trovare la derivata della funzione di errore rispetto ad ogni singolo weight e bias e sottrarre i valori trovati ai weight e bias presenti nei neuroni per arrivare a valori di **W** e **b** tali che la funzione di errore venga minimizzata. Per trovare il minimo della funzione generalmente si

ricorre alla discesa del gradiente. I parametri della rete neurale vengono regolati dalle seguenti formule,

$$W^{[l]} = W^{[l]} - \alpha dW^{[l]}$$

$$b^{[l]} = b^{[l]} - \alpha db^{[l]}$$

dove α rappresenta il tasso di apprendimento, un iperparametro fondamentale arbitrariamente scelto. Un learning rate α troppo basso renderebbe lento il processo di apprendimento, mentre uno eccessivamente alto non permetterebbe di trovare il minimo della funzione a causa dell'*overshooting* nella ricerca del minimo.

Aggiornati tutti i weight e bias una nuova fase di forward propagation inizia, in un ciclo che continua fino a quando non si trova il punto di minimo della funzione di errore.¹⁰

Nel network creato, dopo un centinaio cicli di allenamento l'algoritmo presenta un'accuratezza che si attesta intorno al 96%.

Può essere definito ciclo di allenamento un singolo passaggio di tutto il training dataset attraverso l'algoritmo.

¹⁰ Appendice B, immagine 3, ciclo di forward e backpropagation.

7. CONCLUSIONI¹¹

Questo tipo di analisi viene applicato, con i dovuti adattamenti tecnici, a vari ambiti. Enti e imprese usano questa tecnologia per rendere più snello il processo di customer service, riuscendo a filtrare automaticamente molte richieste con la diminuzione del carico di lavoro del personale di assistenza.

Immagine pubblica e brand reputation sono rispettivamente due aspetti della vita di persone e aziende il cui monitoraggio viene reso molto più facile grazie all'utilizzo di questo tipo di algoritmo. Si possono infatti discernere le opinioni riguardo una persona analizzando le frasi scritte dagli utenti in rete. Nel settore pubblico di alcuni Paesi esteri sono stati fatti passi per migliorare il rapporto con i cittadini attraverso l'implementazione di software che rendono più immediato e veloce il rapporto istituzioni-individui. Il programma creato durante lo svolgimento di questa dissertazione, per sua naturale costituzione, potrebbe essere facilmente adattato all'automazione parziale della lettura di alcuni tipi di risposte a domande aperte contenute in sondaggi demoscopici o indagini similari, facilitando il compito di categorizzare le risposte e fornendo uno strumento per gestire una grande mole di dati.

Non è sempre consigliato tuttavia l'utilizzo delle reti neurali in quanto, oltre alle risorse necessarie per un corretto funzionamento, questo approccio presenta alcuni svantaggi da tenere in considerazione. Facilmente intuibile è la complessità che caratterizza le reti neurali e le decisioni riguardanti la loro architettura devono essere compiute da individui con esperienza in vari campi, risultando dispendiose per enti e imprese. Può capitare che personale specializzato debba lavorare per superare numerose complicazioni durante le fasi di costruzione e apprendimento del modello e, nonostante una varietà di metodi per aumentare l'accuratezza delle reti neurali, rimane difficile individuare la radice delle imprecisioni nelle

¹¹ Per la stesura del presente capitolo ci siamo avvalsi dello studio dei seguenti autori: L.DENG, D.YU, *Deep Learning: Methods and Applications*, in "Foundations and Trends in Signal Processing", n° 7, pp. 1-199.

previsioni. La fase di generazione è spesso vincolata all'affitto di server dotati della capacità di gestire la grande quantità di operazioni necessarie alla creazione del modello classificatore. Ciò implica che per ogni cambiamento effettuato bisognerà nuovamente allenare l'algoritmo per far sì che generi un nuovo modello, operazione che provoca l'impossibilità di continuare a lavorare sul dataset fino al completamento che può richiedere anche diversi giorni. I dati di output devono essere rappresentati in modo chiaro per mettere in condizione i decision maker di usufruire di tutte le informazioni estrapolate durante l'indagine.

In conclusione, è preferibile usare la metodologia risolutiva più semplice per ogni problema e non cadere nella trappola di affidarsi a procedimenti innovativi dove non necessari ma, nelle casistiche non affrontabili da strumenti tradizionali, l'apprendimento profondo può essere una valida alternativa e le performance degne di nota giustificano le risorse necessarie per l'implementazione.

BIBLIOGRAFIA

cap. 1

M. TABOADA, J. BROOKE, *Lexicon-based methods for sentiment analysis* in “*Computational Linguistics*”, n° 37, 2011.

P. STONE, P. DUNPHY, M. SMITH, M. OGLIVE, *The General Inquirer: A Computer Approach to Content Analysis*, Cambridge, MIT Press, 1966.

L.A. GOTTSCHALK, G.C. GLEESER, *The measurement of psychological states through the content analysis of verbal behavior*, California, California Press, 1969.

Ł. AUGUSTYNIAK, P. SZYMAŃSKI, T. KAJDANOWICZ, W. TULIGŁOWICZ, *Comprehensive Study on Lexicon-based Ensemble Classification Sentiment Analysis*, in “*Entropy*”, n° 18, 2015.

M. THELWALL, K. BUCKLEY, G. PALTOGLOU, D. CAI, A. KAPPAS, *Sentiment strength detection in short informal text* in “*Journal of the American Society for Information Science and Technology*”, n° 61, 2010.

P. SOUJANYA et al., *Aspect extraction for opinion mining with a deep convolutional neural network*, in “*Knowledge-Based Systems*”, n° 108, 2016, pp. 42-49.

B. LIU, *Sentiment Analysis and Subjectivity*, in N. INDURKHYA, F. DAMERAU, “*Handbook of Natural Language Processing*”, (Second ed.), 2010.

cap. 2

M. HUTTER, *Universal Artificial Intelligence*, Berlin, Springer, 2005.

P. JACKSON, *Introduction to Artificial Intelligence* (2nd ed.), New York, Dover, 1985.

R. NEAPOLITAN, X. JIANG, *Artificial Intelligence: With an Introduction to Machine Learning*, London, Chapman & Hall/CRC, 2018.

D. POOLE, A. MACKWORTH, *Artificial Intelligence: Foundations of Computational Agents* (2nd ed.), New York, Cambridge University Press, 2017.

D. CREAVIER, *AI: The Tumultuous Search for Artificial Intelligence*, New York, Basic-Books, 1996.

N. NILLSON, *The Quest for Artificial Intelligence: A History of Ideas and Achievements*, New York, Cambridge University Press, 2009.

cap. 3

T. HASTIE, R. TIBSHIRANI J. FRIEDMAN, *The Elements of Statistical Learning*, Berlin, Springer, 2001.

P. DOMINGOS, *The Master Algorithm*, New York, Basic Books, 2015.

I. WITTEN, E. FRANK, *Data Mining: Practical machine learning tools and techniques*, Burlington, Morgan Kaufmann Books, 2011.

A. ETHEM, *Introduction to Machine Learning*, Cambridge, MIT Press, 2004.

cap. 4

J. SCHMIDHUBER, *Deep Learning in Neural Networks: An Overview*, in “*Neural Networks*”, n° 69, 2015, pp. 85-117.

Y. BENGIO, Y. LECUN, G. HINTON, *Deep Learning*, in “*Nature*”, n° 521, 2015, pp. 436-444.

S. HAYKIN, *Neural Networks: A Comprehensive Foundation*, New Jersey, Prentice Hall, 1994.

Y. LECUN et al., *Backpropagation Applied to Handwritten Zip Code Recognition*, in “*Neural Computation*”, 1989, pp. 541–551.

F. KOLEN, U. KREMER, C. STEFAN, *A Field Guide to Dynamical Recurrent Networks*, Hoboken, John Wiley & Sons, 2001.

cap. 5

K. GURNEY, *An introduction to neural networks*, London, UCL Press, 1997.

S. HAYKIN, *Neural networks: a comprehensive foundation*, Prentice Hall, 1997.

P. WERBOS, *The Roots of Backpropagation. From Ordered Derivatives to Neural Networks and Political Forecasting*, New York, John Wiley & Sons, Inc., 1994.

Y. LI, Y. FU, H. LI, S. ZHANG, *The Improved Training Algorithm of Back Propagation Neural Network with Self-adaptive Learning Rate*, in “*International Conference on Computational Intelligence and Natural Computing*”, pp. 73-76.

H. ANTON, *Elementary Linear Algebra* (5th ed.), New York, Wiley and Sons, 1987.

cap. 6

E. CAMBRIA, B. SCHULLER, Y. XIA, C. HAVASI, *New Avenues in Opinion Mining and Sentiment Analysis*, in “*IEEE Intelligent Systems*”, vol. 28, n° 2, 2013, pp. 15-21.

L. DEY, S.K. MIRAJUL HAQUE, *Opinion Mining from Noisy Text Data*, in “*Proceedings of the second workshop on Analytics for noisy unstructured text data*”, 2008, pp.83-90.

L.RAMALHO, *Fluent Python: Clear, Concise, and Effective Programming*, Boston, O'Reilly Media, 2015.

B. CHRISTIAN, T. GRIFFITHS, *Chapter 7: Overfitting* in “*Algorithms to live by: the computer science of human decisions*”, 2017, pp. 149–168.

S. HOCHREITER, J.SCHMIDHUBER, *Long short-term memory*, in “*Neural Computation*”, n° 7, 1997, pp.1735-1780.

cap. 7

L. DENG, D.YU, *Deep Learning: Methods and Applications*, in “*Foundations and Trends in Signal Processing*”, n° 7, pp. 1-199.

SITOGRAFIA

<https://it.wikipedia.org/>

<https://medium.com/>

<https://machinelearningmastery.com/>

<https://towardsdatascience.com/>

<http://www.wildml.com/>

APPENDICE A

Codice algoritmo in Python 2.7

```
from __future__ import division, print_function, absolute_import
import tflearn
from tflearn.data_utils import to_categorical, pad_sequences
from tflearn.datasets import imdb
```

IMDB Dataset loading

```
train, test, _ = imdb.load_data(path='imdb.pkl', n_words=10000,
                                valid_portion=0.1)
trainX, trainY = train
testX, testY = test
```

Data preprocessing

Sequence padding

```
trainX = pad_sequences(trainX, maxlen=100, value=0.)
testX = pad_sequences(testX, maxlen=100, value=0.)
```

Converting labels to binary vectors

```
trainY = to_categorical(trainY, nb_classes=2)
testY = to_categorical(testY, nb_classes=2)
```

Network building

```
net = tflearn.input_data([None, 100])
net = tflearn.embedding(net, input_dim=10000, output_dim=128)
net = tflearn.lstm(net, 128, dropout=0.8)
net = tflearn.fully_connected(net, 2, activation='softmax')
net = tflearn.regression(net, optimizer='adam', learning_rate=0.001,
                           loss='categorical_crossentropy')
```

Training

```
model = tflearn.DNN(net, tensorboard_verbose=0)
model.fit(trainX, trainY, validation_set=(testX, testY), show_metric=True,
          batch_size=32)
```

APPENDICE B

Immagine 1. *Organizzazione degli ambiti di studio.*

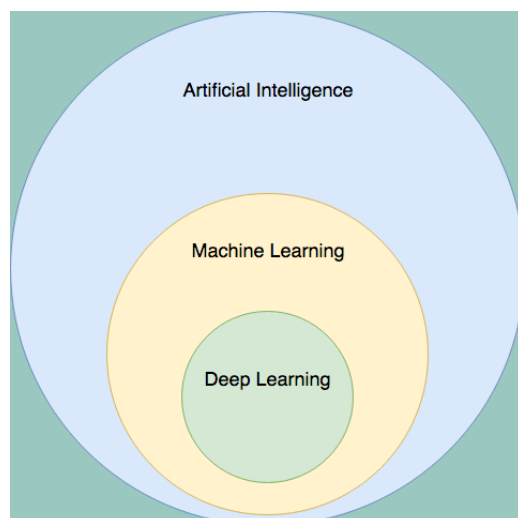


Immagine 2. *Struttura del neurone.*

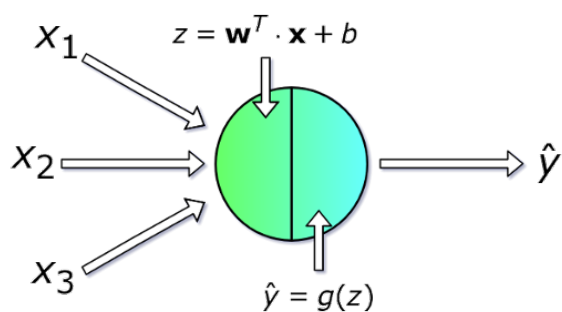


Immagine 3. *Ciclo di forward e backpropagation.*

