



**UNIVERSITA' POLITECNICA DELLE MARCHE**  
**FACOLTA' DI INGEGNERIA**

---

Corso di Laurea triennale **INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE**

**APPRENDIMENTO AUTOMATICO:  
CONFRONTO TRA APPROCCIO SIMBOLICO E NEURALE**

**MACHINE LEARNING:  
COMPARISON BETWEEN SYMBOLIC AND NEURAL NETWORKS**

Relatore: Chiar.mo/a

Prof. **Aldo Franco Dragoni**

Tesi di Laurea di:

**Manila Cuicchi**

**A.A. 2019 / 2020**



# Indice

<b>Introduzione:</b> .....	5
<b>1. Nozioni generali di Intelligenza artificiale</b> .....	8
<b>1.1 Cos'è l'Intelligenza artificiale?</b> .....	8
<b>1.2 Machine Learning</b> .....	9
<b>1.2.1 Tipologie di Apprendimento</b> .....	10
<b>1.2.2 Apprendimento Supervisionato</b> .....	12
<b>1.2.3 Condizione di realizzazione di un apprendimento supervisionato</b> .....	13
<b>1.3 Deep Learning</b> .....	14
<b>1.4 Cardini del funzionamento di una AI</b> .....	15
<b>2. Approccio Simbolico: Alberi di decisione</b> .....	16
<b>2.1 Cos'è un algoritmo di classificazione?</b> .....	16
<b>2.2 Struttura di un albero di decisione</b> .....	17
<b>2.3 Accuratezza dell'albero di decisione</b> .....	21
<b>2.4 Criterio di scelta dell'attributo da testare: Entropia.</b> .....	21
<b>3. Approccio Neurale: Convolutional Networks</b> .....	23
<b>3.1 Cosa sono le Convolutional Networks?</b> .....	23
<b>3.2 Principi delle Convolutional Networks</b> .....	25
<b>3.3 Pooling Layer</b> .....	28
<b>3.4 Struttura degli output e tipi di dato</b> .....	29
<b>3.5 Riduzione dei costi di una rete neurale convoluzionale.</b> .....	30
<b>3.6 Principi Neuroscientifici delle reti convoluzionali</b> .....	30
<b>Conclusioni</b> .....	33
<b>Fonti Bibliografiche e Sitografia</b> .....	37



## **Introduzione:**

L'intelligenza artificiale è una delle discipline di studio per cui negli ultimi settant'anni si è incentrata maggiormente la ricerca e lo sviluppo di nuovi algoritmi sempre più efficienti.

La domanda che sorge spontanea è “perché negli ultimi anni si sente molto parlare di AI?”

La ragione principale è per migliorare il nostro stile di vita ed il mondo in generale, in diversi settori, come quello imprenditoriale (ad esempio la creazione di macchine per fabbricare oggetti oppure agenti autonomi come le semplici autovetture ma con più funzionalità di come già le conosciamo) e quello medico (ad esempio “macchine” con software specializzati per far fronte ad eventuali interventi medici con lo scopo di migliorarne sempre di più la qualità e il tempo di prestazione).

L'intelligenza umana rappresenta indubbiamente una grande risorsa, che nei secoli ha portato a grandi ed importanti scoperte fondamentali per l'evoluzione della nostra specie, ma con un grande limite quello di non supportare infinite informazioni tutto nello stesso momento. L'obiettivo principale non è soltanto quello di ampliare le conoscenze, la ricerca, ma anche quello di renderci capaci attraverso nuove “macchine” o meglio, agenti intelligenti, di contenere quante più informazioni possibili che non saremmo in grado altrimenti da soli di accogliere. Alla luce di questo proposito entra in gioco l'intelligenza artificiale, la cui grande risorsa è quella di avere una grande capacità di calcolo, parliamo di potenzialità di calcolo in crescita esponenziale e ciò permette a queste “macchine” di contenere una quantità infinita di dati.

In secondo luogo andiamo a capire che tipo di software questa disciplina ci permette di creare e quale siano le necessità legate a questa creazione.

Un altro aspetto che andremo ad affrontare fondamentalmente sono gli approcci che possono essere impiegati proprio per mettere al mondo questi agenti intelligenti. Si parlerà di un approccio simbolico e un approccio neurale, differenti tra loro ma ognuno con vantaggi e svantaggi che li rendono ancora in grande competizione tra loro. La sfida più grande della scienza è ora quella di provvedere allo sviluppo di un metodo efficiente che possa rendere un agente in grado di elaborare quante più informazioni possibili, e in questo ci sta già riuscendo, ma con la garanzia che la diagnosi finale possa essere più convincente ed affidabile possibile.

Durante questa analisi è possibile comprendere nello specifico le diversità con le quali operano i due approcci trattati e soprattutto, comprendere le fasi che porteranno ad una conclusione finale stabilendo quale delle due applicazioni sarà più promettente per il futuro.



## Capitolo 1

### 1. Nozioni generali di Intelligenza artificiale

#### 1.1 Cos'è l'Intelligenza artificiale?

L'intelligenza artificiale può essere definita come la capacità di rendere un sistema informatico capace di risolvere problemi, eseguire compiti ed in sostanza di agire simulando l'abilità di un essere umano.

Dal punto di vista informatico, è definibile come la disciplina in grado di creare macchine, come software o hardware, capaci di operare autonomamente.

Negli ultimi anni sentiamo molto parlare di Intelligenza Artificiale, specialmente nel web, pronunciando sempre quale possa essere il suo effetto nella società del futuro, cioè come cambierà la nostra vita quotidiana.

Sarebbe opportuno cercare di comprendere i veri aspetti di questa nuova materia oggetto di studio che proprio in questo capitolo introduttivo andremo a discutere più in profondità, in altre parole com'è realmente organizzata.

I primi studi sono risalenti alla seconda metà del ventesimo secolo, quando due scienziati cercarono di riproporre la struttura di un vero e proprio neurone umano, questo al fine di dimostrare che attraverso la tecnologia ci potesse essere la possibilità di creare macchine in grado di apprendere l'arte di elaborare informazioni ed essere di supporto all'intelligenza umana.

Per noi umani la mente è una grande risorsa, che ci ha permesso nel corso della nostra evoluzione di raggiungere risultati sorprendenti che hanno segnato la storia e migliorato decisamente il nostro stile di vita, ma con un limite, che in futuro la quantità di informazioni che ci ritroviamo ad elaborare è sempre maggiore e questo per noi potrebbe rappresentare un problema molto grande.

Perciò l'intelligenza artificiale entra soprattutto in gioco in questo frangente, già oggi ci sono software di dimensioni sempre più piccole con elevatissime capacità di calcolo, e questo permette proprio di alimentare la crescita dei dati contenibili in essi, il cui risultato è quello di una grande rivoluzione.



Dare una definizione precisa di Intelligenza Artificiale è un compito davvero arduo, ma possiamo intendere questa grande disciplina sotto due punti di vista differenti che ne caratterizzano l'evoluzione, si tratta di intelligenza artificiale debole e forte.

Per darne una definizione, **l'intelligenza artificiale debole** descrive un sistema tecnologico in grado di simulare delle abilità cognitive tipiche dell'uomo, ad esempio riguarda la realizzazione di un programma matematico in grado di risolvere un dato problema o di un programma che consente alla macchina di poter prendere decisioni.

L'**intelligenza artificiale forte** rappresenta dei veri e propri sistemi definiti "sapienti" capaci di sviluppare da sé una propria intelligenza, senza la necessità di emulare un pensiero umano come nel caso precedente.

Questa distinzione è indispensabile per apprendere al meglio le due classificazioni di cui andremo a discutere in seguito, rispettivamente: Machine Learning e Deep Learning.

## 1.2 Machine Learning

Machine Learning, o apprendimento automatico, si definisce attraverso una serie di metodi che la macchina (software o hardware) deve apprendere per saper in maniera autonoma simulare/svolgere un compito o un'attività.

La macchina che fisicamente rappresenta la nostra intelligenza può essere chiamata **agente intelligente**, il cui obiettivo è quello di apprendere tramite l'osservazione e dall'esperienza. Questo significa che dovrà essere capace di far migliorare le proprie prestazioni analizzando il mondo che lo circonda e percependo i risultati delle proprie azioni. Altro aspetto non da meno, è interiorizzare le regole che governano il mondo in cui si ritroverà ad agire soltanto osservandolo e successivamente tradurle nel linguaggio a lui conosciuto, attraverso le formule.

Le ragioni che spingono alla creazione di un software in grado di apprendere non sono affatto banali, anzi sono semplici ma fondamentali.

Il programmatore potrebbe non conoscere tutte le situazioni in cui il software implementato si troverà ad agire in uno scenario futuro, non può affatto prevedere quali siano i cambiamenti che riguardano il mondo esterno all'agente ma soprattutto non può avere in ogni circostanza la certezza di avere sempre le conoscenze adatte per saper risolvere un certo problema.

L'insieme delle forme di apprendimento che caratterizzano l'apprendimento automatico dell'intelligenza artificiale son ben quattro, e se ne distingue una classificazione dei relativi algoritmi:

- Apprendimento Supervisionato;
- Apprendimento Non Supervisionato;
- Apprendimento con Rinforzo;
- Semi-Apprendimento Supervisionato.

### **1.2.1 Tipologie di Apprendimento**

In questo paragrafo vengono illustrate in generale, quali sono le caratteristiche delle tipologie di apprendimento esistenti:

#### **1. Apprendimento Supervisionato:**

Questo apprendimento è determinato da una serie in di input ed output che vengono forniti all'AI, la quale deve essere capace di apprendere come comportarsi.

Possiamo immaginare come la presenza di un "maestro" che fornisce una serie di informazioni relative a certe situazioni. Così l'intelligenza (AI) da questo apprendimento deve essere in grado di intuire quali sono le azioni giuste da dover compiere e quelle scorrette, da dover evitare, che in passato hanno condotto a soluzioni errate (un esempio di questa rappresentazione di apprendimento si vedrà nel capitolo successivo).

#### **2. Apprendimento Non Supervisionato:**

In questo caso l'intelligenza artificiale, non viene implementata attraverso algoritmi che le dicono come dover apprendere, bensì il vero e proprio apprendimento, avviene dall'analisi dei risultati ottenuti come output.

In altre parole, il software capisce come agire in base ai risultati e di conseguenza si adatta anche il suo metodo di apprendimento.

Un esempio molto vicino a noi, nel campo di applicazione, è quello dei motori di ricerca, che utilizziamo ormai quotidianamente.

In pratica quest'ultimi, attraverso una serie di parole chiave digitate riescono a fornire come risultato una serie di link o collegamenti, che secondo l'algoritmo sono più

attinenti alla ricerca eseguita.

La loro efficienza viene valutata in base alla ricerca, ovvero in funzione di quanto le informazioni e dati relativi ai link forniti dall'algoritmo, risultano effettivamente utili.

### 3. **Apprendimento con Rinforzo:**

Questo metodo si distingue dai precedenti in quanto emula ciò che accade ad un agente intelligente che si abitua ad associare le risposte corrette agli stimoli, ed assimilando questo procedimento riesce a migliorare ed affinare le associazioni che hanno portato alla formulazione di risposte corrette, e sfavorire quelle che hanno portato a risposte sbagliate.

Un esempio pratico che rappresenta questa tipologia di apprendimento, è il “multi-agent hide and seek” di OpenAI, ossia un'intelligenza artificiale multi-agente in cui gli agenti apprendono come giocare a nascondino.

Si è visto che questi agenti nella fase di addestramento riescono ad apprendere nel mentre molte altre abilità collaborando tra loro, perché si trovano in un ambiente competitivo. Quindi ciò comporta che proprio gli agenti sviluppino comportamenti anomali che non si erano programmati. Questi agenti nascondino non hanno fatto altro che evolversi in modo organico grazie ai loro stessi progressi.

### 4. **Apprendimento Automatico Semi-Supervisionato:**

Facciamo riferimento ad un esempio di agente intelligente, che è costituito da più di uno di questi metodi di apprendimento che cooperano insieme.

In altre parole, possiamo dire che questa formula rientra tra apprendimento supervisionato e non supervisionato, ovvero approcci ibridi.

Ad esempio, etichettare ogni singola informazione presente nei messaggi che vengono scambiati nei social networks risulta molto dispendioso, perciò a questo punto entra in gioco la rappresentazione di un apprendimento semi supervisionato.

Avendo una porzione di informazioni etichettate, utilizzando le tecniche supervisionate, sfruttando questi piccoli insiemi di dati, si riesce più facilmente ad arrivare a conoscere il resto del contenuto di un eventuale messaggio.

## 1.2.2 Apprendimento Supervisionato

Come già accennato si tratta di una forma di apprendimento che si supponga sia guidata da un “maestro”, il quale attraverso degli stimoli, cioè coppie di input ed output, cerca di far capire all’agente quali sono le regole nascoste (non fornite esplicitamente) da dover individuare affinché sia capace di elaborare la giusta soluzione.

Esso è caratterizzato principalmente da coppie di input/output, in particolare,  $N$  coppie che costituiscono un “training set”, ovvero un set di allenamento:

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

Queste coppie nascono da una funzione generica ed ignota,  $y = f(x)$ , nota al “maestro” per associare un certo input ad un certo output che inizialmente è sconosciuta al nostro agente. Infatti, è proprio quest’ultimo a scoprire una certa funzione  $h(x)$ , la quale appartiene ad un certo spazio di ipotesi  $H$ , le quali vanno ad approssimare la funzione  $f(x)$ .

Questo significa che l’agente nel momento in cui individua una funzione  $h(x)$  elabora nel suo processo mentale un’ipotesi, corrispondente alla funzione  $f(x)$  che ha prodotto le varie coppie input/output conosciuta dal “maestro”, e che successivamente il nostro agente deve approssimare.

Nella fase successiva si va ad individuare tra le coppie del “training set” un altro insieme di coppie, distinte dal precedente, che formano il **test set**, ovvero degli input (variabili  $x$ ) per testare la funzione  $h(x)$ .

L’agente deve essere in grado di dare il valore  $y$  corrispondente alla funzione  $h(x)$  che gli è stata data, ed in base a quest’ultima, andare a vedere se il valore  $y$  corrisponde o meno, a quello della funzione ignota.

In questo modo sarà possibile verificare l’accuratezza dell’ipotesi.

In generale, la funzione ignota in questione,  $f(x)$ , può assumere tre tipologie differenti:

- Funzione ***f* stocastica**: corrisponde ad una distribuzione di probabilità ignota, in cui la funzione  $h$  dovrebbe individuare quale sia la distribuzione che ha generato gli esempi;
- Funzione ***f* enumerativa**: il cui apprendimento prende il nome di **classificazione**;

- Funzione  $f$  **numerica**: il cui apprendimento viene detto **regressione**.

### 1.2.3 Condizione di realizzazione di un apprendimento supervisionato

*“Un apprendimento è realizzabile se la funzione  $f$  appartiene allo spazio delle ipotesi ( $f \in H$ ), ma non è sempre dato conoscere tale funzione  $f$ .”<sup>1</sup>*

Questo avviene perché si possono presentare due situazioni: una in presenza di un “maestro” con la conoscenza di tale funzione, nell’altra, in assenza di un “maestro” dove la ricerca di tale funzione avviene attraverso l’esperienza e quindi, non la si conosce.

Il passaggio successivo è quello di cercare una funzione  $h$  che massimizzi la probabilità che tale funzione ( $h$ ) ricercata dall’agente nello spazio delle ipotesi ( $H$ ), sia quella giusta alla luce dei dati che hanno origine dal **training set**. Tale condizione dal teorema delle probabilità (o anche detto teorema di Bayes) vediamo essere uguale in termini più specifici a quest’espressione:

$$h^* = \underset{h \in H}{\operatorname{argmax}} P(h|\text{data}) = \underset{h \in H}{\operatorname{argmax}} P(\text{data}|h)P(h)$$

La particolarità in questa ricerca da parte dell’agente della relativa funzione  $h$  è la volontà che la probabilità delle ipotesi, ovvero  $P(h)$ , sia più alta per le ipotesi più semplici. Questo aspetto è importante da considerare per il semplice fatto, che più lineare ed omogenea è l’ipotesi più ci sarà la possibilità che quella sia l’ipotesi più accessibile e, corrispondente a quella cercata.

Inoltre, è utile considerare anche un vantaggio in quanto la semplicità di un’ ipotesi, riconducibile alla funzione da ricercare, ha un notevole impatto sull’ implementazione dell’algoritmo che la rappresenta. Significa che per un programmatore se l’ipotesi è semplice diminuiscono le difficoltà durante la costruzione dell’algoritmo, e ciò lo renderà più funzionale ed efficace.

---

<sup>1</sup> Definizione tratta dalle slides del corso di Intelligenza Artificiale a cura di Aldo Franco Dragoni

### 1.3 Deep Learning

Deep Learning, ovvero “apprendimento in profondità”, ha come obiettivo quello di simulare il funzionamento della mente umana, con l’abilità di saper apprendere in maniera del tutto autonoma per poi fornire una propria soluzione logica rispetto ad un certo problema.

La classifichiamo come una particolare branca che opera nel Machine Learning, che corrisponde ad un metodo di apprendimento non supervisionato in quanto alla macchina vengono soltanto forniti degli input e sarà lei stessa tramite diversi tentativi pian piano a stabilire quali siano i classificatori opportuni da utilizzare per distinguere le informazioni ed imparare a prendere decisioni, riconoscere certi oggetti piuttosto che altri.

Per classificatori intendiamo modelli, che caratterizzano alcuni sistemi tecnologici e cercano di emularne l’apprendimento, la cui particolarità è quella di non essere determinati da un ricercatore ma vengono ricercati attraverso un’opportuna metodologia dal software stesso.

Uno degli esempi più conosciuti che corrisponde al metodo di apprendimento appena descritto sono le **reti neurali** (*Deep Artificial Neural Networks*).

La loro struttura è basata su una serie di livelli di profondità piuttosto elevata, la cui scelta dei classificatori, si basa su una quantità di informazioni davvero elevate, per questo entra in gioco la necessità di avere a disposizione grande capacità computazionale volta a far fronte a questo genere di gestione ed analisi dell’informazione.

Tutto questo per emulare ciò che accade realmente con le connessioni neurali del nostro cervello, ovvero l’obiettivo finale.

## 1.4 Cardini del funzionamento di una AI

Fino ad ora abbiamo fatto chiarezza su quali “linee” si estende questa nuova disciplina che negli ultimi decenni si sta sviluppando molto velocemente e con grandi risultati.

L’intelligenza artificiale si basa fundamentalmente su alcune definizioni intellettuali che ne caratterizzano in primo luogo le fondamenta su cui si erge poi l’obiettivo finale al quale aspira.

Stiamo parlando di:

1. **Comprensione:** rispecchia l’abilità di manifestare capacità cognitive come quelle del nostro cervello legate a dati ed eventi, che migliorano le abilità dell’intelligenza in questione, nel riconoscere informazioni, immagini, luoghi;
2. **Ragionamento:** capacità di elaborare un procedimento logico, ovvero costruzione di un pensiero plausibile, di senso compiuto, in grado di soddisfare una teoria valida al fine di poter collegare tra loro le informazioni;
3. **Apprendimento:** si tratta di un metodo che prevede l’analisi e la gestione delle informazioni ricevute come input e responsabile anche delle informazioni corrette da restituire in output come risultato valido;
4. **Interazione:** rendere un’intelligenza capace di interagire, scambiare informazioni con l’uomo, in grado di collaborare al fine di potenziare i risultati in uscita.

Questi cardini possono essere visti sotto due punti di vista, quello informativo o quello pratico.

Dal punto di vista informativo si limitano a diffondere il significato su cui si basa la costruzione di un’intelligenza artificiale e quindi parliamo della divulgazione di un concetto generale.

Inoltre, dal punto di vista pratico svolgono un vero e proprio lavoro perché sono una guida da cui far partire il reale progetto di realizzazione e stabiliscono le linee fondamentali su cui basare il proprio progetto di realizzazione.

## Capitolo 2

### 2. Approccio Simbolico: Alberi di decisione

In questo capitolo, in particolare si affronta una rappresentazione dell'apprendimento appartenente all'approccio simbolico, o meglio Machine Learning. La rappresentazione considerata viene detta *albero di decisione*.

#### 2.1 Cos'è un algoritmo di classificazione?

Una premessa importante è che l'albero di decisione è un algoritmo di classificazione, perciò è importante avere presente quale sia il loro funzionamento.

Per cominciare l'obiettivo di un algoritmo di classificazione è quello di individuare schemi o caratteristiche che appartengono ad una determinata classe che identifica un certo oggetto.

Al principio si considera l'insieme dei dati in input, detto training set, costituito da record ognuno dei quali viene caratterizzato da attributi necessari a definire la classe di appartenenza di un certo oggetto.

Infine, compare una voce "Classe" che va considerata come una sorta di etichetta di classe.

	Attributo 1	Attributo 2	.....	Attributo n	Classe
Record 1					
Record 2					
Record 3					
Record 4					
.....					
.....					
Record n					

Figura 1- tabella che rappresenta la struttura di un training set

Il primo passo è quello di analizzare questi dati in input, utilizzando le peculiarità (ovvero gli attributi) dei dati, per essere in grado di creare un'identificazione e modello che descrive al meglio la classe a cui appartiene un dato oggetto.

Questo avviene considerando inizialmente degli insiemi di classi già definiti ed esistenti, con il fine di sviluppare delle similarità che appartengono alle varie classi.



Nel training, come ad esempio gli attributi servono per definire le classi di oggetti, e quindi i vari record anche se inizialmente non se ne conosce la classe di appartenenza.

Ciò per cercare di rafforzare e fortificare lo sviluppo di una migliore conoscenza di ciascuna classe nel dataset.

## 2.2 Struttura di un albero di decisione

Gli alberi di decisione rappresentano una delle modalità più semplici per classificare gli “oggetti” del mondo che ci circondano, attraverso un numero finito di classi.

In altre parole, è una modalità con cui poter trattare un problema di classificazione, dove l’obiettivo è quello di imparare una volta note le regole di classificazione di una certa classe che identifica un oggetto come applicarle, con quale ordine. Queste regole sono conosciute dal “maestro”, citato nel precedente capitolo per spiegare cosa fosse un apprendimento supervisionato.

Un albero di decisione è così chiamato perché la sua suddivisione produce una gerarchia detta ad albero. Questo significa che la classificazione degli oggetti, viene fatta attraverso una suddivisione dei record (dati iniziali) in sottoinsiemi simili rispetto alla variabile finale (per quest’ultima intendiamo l’attributo classe del dataset).

In questa struttura ad albero i sottoinsiemi dei record costituiscono i *nodi*, e quelli finali vengono chiamati *foglie*.

I nodi rappresentano i nomi degli attributi, mentre gli archi, ovvero i rami dell’albero che collegano vari nomi vengono etichettati con i valori degli attributi. Infine, le foglie vengono mostrate con le diverse forme dell’attributo “classe”.

La classificazione degli oggetti avviene attraverso l’individuazione di un percorso che parte dalla radice per arrivare ad una foglia, ed i vari percorsi sono determinati dalle varie regole rappresentate dai rami dell’albero. Queste regole vengono determinate attraverso una sequenza di test che forniscono una decisione<sup>2</sup> finale, dato un singolo valore in output.

Generalmente, gli alberi di decisioni possono avere attributi interni di due tipologie.

---

<sup>2</sup>Ricordiamo che per “decisione” si intende l’output finale, scelto a partire da un vettore di valori di attributi dato inizialmente.

La prima riguarda variabili di tipo discreto più comunemente usati e facili da gestire in quanto sono semplici alberi di classificazione che ripongono i record in una certa categoria o classe, come nell'esempio a cui faremo riferimento a breve. La seconda tipologia fa riferimento ad alberi di decisione che intuiscono variabili di tipo continuo ed in letteratura prendono il nome di alberi di regressione (che in questa tesi citiamo soltanto).

Quest'ultimi sono ardui da gestire in quanto costituiti da elementi infiniti, in cui non è semplice individuare una categoria o classe precisa, come nella tipologia precedente.

Qui di seguito riporto un esempio per intuire più chiaramente quale sia il funzionamento di un albero di decisione.

L'esempio che sto per andare a proporre è tratto dal testo "Programmazione in prolog per l'intelligenza artificiale" dell'autore I. Bratko, in cui la situazione proposta è quella di una coppia padre e figlio solita ad andare molto spesso a cena nei ristoranti.

Il padre decide di elaborare un metodo tale per cui riesce a prevedere quale sia il luogo più conveniente in cui decidere di fermarsi a mangiare ed aspettare che si liberi un tavolo. Sarà possibile elaborare una risposta soddisfacente dopo che il padre riesca a compiere una serie di test con esito soddisfacente.

Quest'ultimi non sono noti al figlio, il quale cerca di intuire quali sono i criteri assunti dal padre per elaborare la migliore decisione, attraverso la sua esperienza nell'andare a cena con il padre ed osservare ad esempio quante volte il padre si è fermato a mangiare in un determinato luogo piuttosto che le volte in cui non si è recato in un altro ristorante.

Consideriamo una variabile booleana (vero o falso) ossia variabile obiettivo che chiamiamo "*Willwait*" ed attributi di input che possono essere di tipo booleano o enumerativo.

attributi di input:

- 1) *Alternate*: c'è un'alternativa valida in un ristorante vicino? (Y/N)
- 2) *Bar*: il ristorante ha un'area bar comoda dove aspettare? (Y/N)
- 3) *Fri/Sat*: è un WE? (Y/N)
- 4) *Hungry*: si è affamati? (Y/N)
- 5) *Patrons*: affluenza [none, someone, full]
- 6) *Price*: categoria del ristorante [\*, \*\*, \*\*\*]
- 7) *Raining*: piove? (Y/N)
- 8) *Reservation*: abbiamo una prenotazione? (Y/N)
- 9) *Type*: genere di cucina [french, italian, thai, burger]
- 10) *WaitEstimate*: stima tempo d'attesa [0-10, 10-30, 30-60, >60]

Figura 2- lista degli attributi, ossia caratteristiche iniziali per la creazione dell'albero di decisione relativo (immagine tratta dalle slides del professor Aldo Franco Dragoni del corso di Intelligenza artificiale, fonte anche delle successive immagini che troviamo in questo paragrafo)

In seguito, vediamo la possibile struttura di un albero di decisione corrispondente al supervisore relativo all'esempio che stiamo considerando.

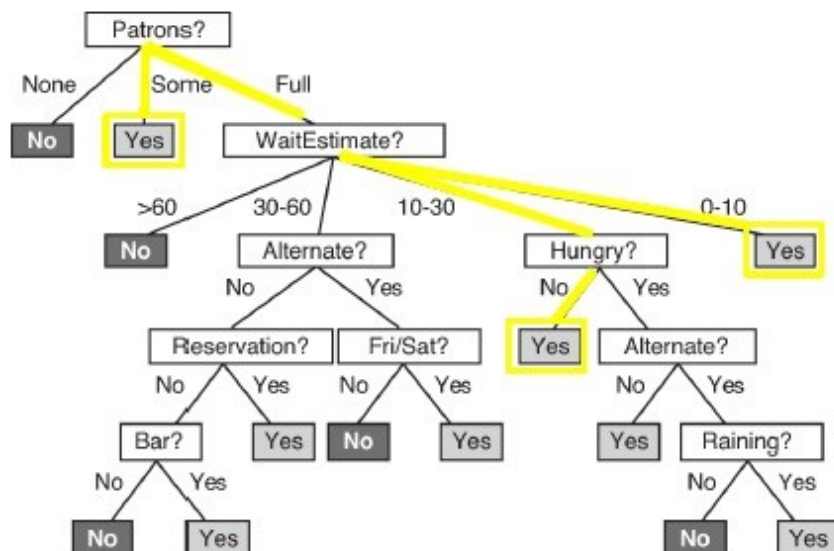


Figura 3-rappresentazione del possibile albero di decisione.

Ciò che si nota dalla figura (Figura 3) “Patrons?” sta ad indicare la radice di questo albero, per cui si indica l’affluenza di persone in un determinato ristorante che può essere: nessuna (None),

qualcuno (Some) e pieno (Full).

Infatti, da questa prima situazione si ottiene una foglia, che può essere positiva o negativa mi fornisce quindi una risposta. Nel caso in cui la risposta fosse “Full” si genera un’ulteriore diramazione dove si va a stimare il tempo possibile di attesa da cui deriva un’altra funzione di verità, ottenendo un’ulteriore risposta e così via.

Perciò è evidente che le funzioni di verità che appaiono sono molte, perciò è necessario individuare attraverso delle regole, criteri quale possa essere una stima dell’albero di decisione simile che porti ad una decisione finale che soddisfi il più possibile, a partire da un insieme di esempi che vengono elaborati dal figlio e utilizzati come test nella fase di apprendimento.

	Alt	Bar	WE	Hun	Pat	Pri	Rain	Res	Type	Est	WillWait
$x_1$	Y	N	N	Y	Some	***	N	Y	Fren	0-10	$y_1=Yes$
$x_2$	Y	N	N	Y	Full	*	N	N	Thai	30-60	$y_2=No$
$x_3$	N	Y	N	N	Some	*	N	N	Burg	0-10	$y_3=Yes$
$x_4$	Y	N	Y	Y	Full	*	Y	N	Yhai	10-30	$y_4=Yes$
$x_5$	Y	N	Y	N	Full	***	N	Y	Fren	>60	$y_5=No$
$x_6$	N	Y	N	Y	Some	**	Y	Y	Ital	0-10	$y_6=Yes$
$x_7$	N	Y	N	N	None	*	Y	N	Burg	0-10	$y_7=No$
$x_8$	N	N	N	Y	Some	**	Y	Y	Thai	0-10	$y_8=Yes$
$x_9$	N	Y	Y	N	Full	*	Y	N	Burg	>60	$y_9=No$
$x_{10}$	Y	Y	Y	Y	Full	***	N	Y	Ital	10-30	$y_{10}=No$
$x_{11}$	N	N	N	N	None	*	N	N	Thai	0-10	$y_{11}=No$
$x_{12}$	Y	Y	Y	Y	Full	*	N	N	Burg	30-60	$y_{12}=Yes$

Figura 4- esempi considerati durante la fase di apprendimento di un albero di decisione

### 2.3 Accuratezza dell'albero di decisione

Il concetto di accuratezza è necessario per stabilire le capacità di classificazione di un albero di decisione.

Stabilire quale sia l'accuratezza è possibile a partire da un numero di dati che si possiedono inizialmente, di cui una parte si decide di utilizzare per effettuare la fase di apprendimento e un'altra parte da impiegare nella fase di test per verificare praticamente quale sia la percentuale di volte che l'albero indotto ha stabilito praticamente la scelta corretta rispetto agli esempi utilizzati nella fase di test set.

### 2.4 Criterio di scelta dell'attributo da testare: Entropia.

L'Entropia è un concetto necessario da esporre legato alla costruzione dell'albero di decisione, che ci guida verso la scelta di quello migliore, ovvero ne verifica la bontà con un metodo probabilistico.

Se consideriamo due classi del nostro problema di classificazione, positiva (+) e negativa (-) e  $V$  la variabile aleatoria di valori  $v_k$ , ognuno dei quali con una probabilità  $P(v_k)$ , l'entropia introdotta da Shannon e Weaver (nel 1949) si esprime nel seguente modo:

$$H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = - \sum_k P(v_k) \log_2 P(v_k) \quad (1)$$

Adesso si consideri una variabile booleana  $B$ , con una certa probabilità  $q$ :

$$H(B) \stackrel{\text{def}}{=} B(q) = -(q \log_2 q + (1 - q) \log_2 (1 - q)) \quad (2)$$

Da specificare che la scelta dell'attributo si ha tra due estremi:

- *Attributo perfetto*: genera sottoinsiemi appartenenti alle sole classi positiva o negativa;
- *Attributo inutile*: genera sottoinsiemi con identico numero di classi di positivi e negativi.

Nella formula (2) sono espresse tra parentesi la probabilità ( $q$ ) di esempi classificati come positivi e la probabilità ( $q$ ) degli altri classificati come negativi.

Di seguito in figura riportiamo l'andamento della variazione di entropia.

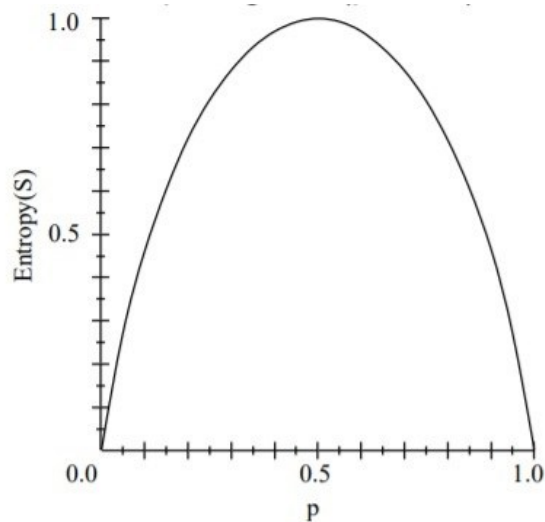


Figura 5-grafico che mi rappresenta i valori che può assumere l'entropia

Dalla figura 5, si ha che nel caso in cui la probabilità ( $q$ ) è pari a 0 e 1 allora l'entropia  $H(B)=0$  mentre l'entropia  $H(B)=1$  cioè massima, se la probabilità è pari a  $\frac{1}{2}$ .

Se l'entropia assume il valore massimo significa che si genera un ordine caotico, ossia si fa una certa difficoltà ad assegnare ogni record ad una classe di identificazione in base agli attributi, perciò si ha un'informazione meno precisa.

Da questa situazione l'obiettivo prossimo è quello di ridurre questo disordine che è stato prodotto e per fare ciò si ricorre ad un'ulteriore partizione dei record, che avviene attraverso un preciso attributo  $A$ , che mi conduce ad un altro valore dell'entropia che posso chiamare  $H'(V)$  in maniera tale da porre la condizione  $H'(V) \leq H(V)$ . Si nota che in questa modalità si ottiene una diminuzione del valore dell'entropia e al contempo massimizza **information gain** ( $G$ ).

Quest'ultima è la quantità che si ottiene dalla sottrazione tra la quantità iniziale di entropia  $H(V)$  e quella ottenuta dopo il partizionamento rispetto all'attributo  $A$ .

Ciò determina la scelta dei nodi nel problema di un albero di decisione che mi permette di scegliere di volta in volta, l'attributo  $A$  che mi garantisce al contempo la diminuzione del valore dell'entropia e la massimizzazione dell'information gain.

## Capitolo 3

### 3. Approccio Neurale: Convolutional Networks.

In questo capitolo, si pone attenzione su un'altra tipologia o rappresentazione di apprendimento che riguarda l'approccio neurale o meglio, Deep Learning (Apprendimento in profondità).

Vedremo cosa sono, qual è la loro struttura, perché vengono applicate e qual è in particolare il loro ruolo e finalità.

#### 3.1 Cosa sono le Convolutional Networks?

*Convolutional Networks*, sono anche conosciute come *Convolutional Neural Networks*, prendono questo nome in quanto sono delle tipologie di reti neurali che implementano un'operazione matematica detta *convoluzione* ed hanno una struttura a "griglia", cioè articolata in più settori.

L'operazione di *convoluzione* è data da una duplice funzione che ha come argomento un valore reale.

Consideriamo ad esempio un sensore laser che fornisce un singolo output che indichiamo con  $x(t)$ , ad un certo tempo  $t$  ( $x$  e  $t$  sono valori reali). Ora vogliamo che il sensore laser sia meno rumoroso perciò si vuole ottenere una stima meno rumorosa grazie ad una media di misurazioni.

Perciò è necessario eseguire una media ponderata di una serie di misurazioni, per fare questo abbiamo bisogno di una funzione di ponderazione indicata con  $w(a)$ , dove "a" sta ad indicare l'età di una misurazione. Applicando la funzione di ponderazione ad ogni istante  $t$  si ottiene un'altra funzione che indichiamo con  $s$ , fornendo così una stima regolare del valore che si vuole ottenere dal principio.

La nuova funzione si esprime:

$$s(t) = \int x(a)w(t - a)da$$

Questa prende il nome di *convoluzione*.

Essa è denotata da un asterisco, vale a dire:

$$s(t) = (x * w)(t)$$

nella quale, le condizioni necessarie sono che la funzione  $w$  deve essere:

- una valida funzione di densità di probabilità affinché l'output sia una media ponderata;
- sia zero per tutti gli argomenti negativi, altrimenti è come prevedere qualcosa di futuro e ciò non è nelle nostre capacità.

Nelle reti di convoluzione, il primo argomento della convoluzione (nel nostro caso si fa riferimento alla funzione  $x$ ) è spesso un **input**, il secondo argomento è come il **kernel** mentre l'output si riferisce ad una “**mappa delle caratteristiche**” (dall'inglese **feature map**).

Il parametro  $t$ , con cui indichiamo il tempo, è un indice che prende solo valori interi, e quindi ponendo che le funzioni  $x$  e  $w$  sono definite solo con valori positivi si definisce una convoluzione discreta:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t - a)$$

Nelle applicazioni degli apprendimenti automatici, abbiamo che l'input spesso è un array multidimensionale di dati e il kernel è un vettore multidimensionale di parametri che sono immagazzinati dall'algoritmo di apprendimento. Si fa riferimento a questo tipo di vettori perché ogni elemento dell'input e kernel deve essere esplicitamente memorizzato separatamente.

Quindi, ciò che accade è che solitamente queste funzioni sono zero ovunque tranne che nell'insieme finito di punti per il quale si memorizzano i valori.

Tutto ciò significa che si implementa una somma infinita come la somma su un numero finito di elementi.

Una proprietà della convoluzione è quella di essere commutativa, perciò se consideriamo un'immagine bidimensionale  $I$  come input, utilizziamo presumibilmente anche un kernel bidimensionale  $K$ , espresso come:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n).$$

Essendo commutativa può anche essere scritta nel seguente modo:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n).$$



Si nota da quest'ultima formula che la proprietà commutativa nasce perché è stato capovolto il kernel relativo all'input, ovvero  $m$  cresce e l'indice nell'input cresce, ma l'indice relativo al kernel decresce.

Una curiosità è che alcune reti neurali hanno librerie che implementano una funzione simile alla convoluzione ma senza capovolgere il kernel, si chiama **cross convolution**, e si scrive:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n).$$

Nella maggior parte dei casi la convoluzione nel machine learning viene sempre applicata con altre funzioni mentre le convoluzioni discrete sono mostrate come una moltiplicazione di matrici.

### 3.2 Principi delle Convolutional Networks

I principi su cui si basano e si sviluppano le *reti neurali convoluzionali* sono ben tre:

- **interazioni sparse;**
- **condivisione di parametri;**
- **rappresentazioni equivarianti.**

Esse sono caratterizzate, il più delle volte, da **interazioni sparse**, ciò significa che avendo livelli che operano usando moltiplicazioni di matrici con una matrice di parametri con un parametro separato che descrive l'interazione tra ogni unità di input e ogni unità di output, il kernel deve essere minore rispetto all'input.

Questo ci permette di ridurre i requisiti di memoria del modello considerato e migliorare l'efficienza, usufruendo di minori parametri. Ad esempio se gli input sono  $m$  e gli output sono  $n$ , allora le condizioni della matrice di moltiplicazione sono  $m \times n$  parametri e gli algoritmi processati hanno  $O(m \times n)$  runtime.

Nelle immagini seguenti (tratte dal libro "Deep Learning" di I. Goodfellow, Yoshua Bengio, Aaron, Cap.9), è possibile vedere la struttura delle interazioni sparse.

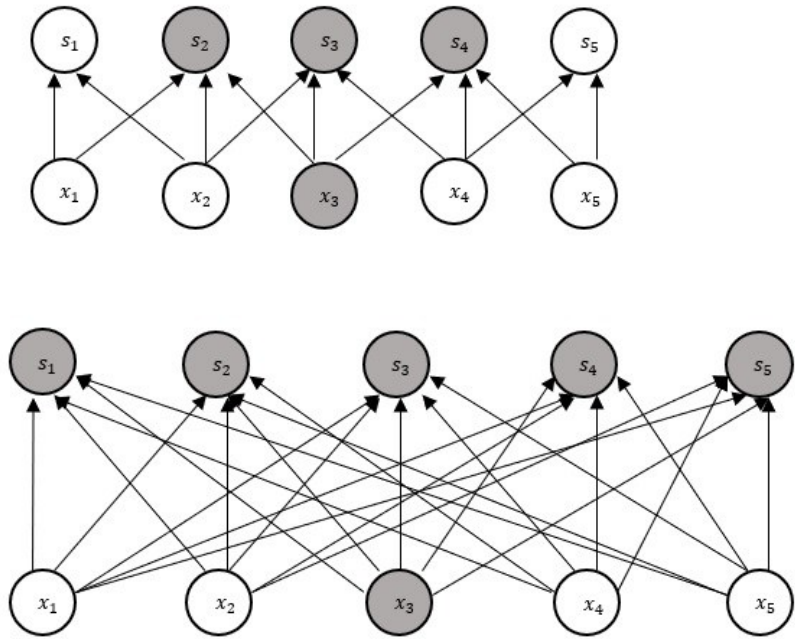


Figura 6 - Struttura di una connessione sparsa, con prospettiva dal basso, vediamo i tre output indicati con  $s$ , che sono legati all'unità di input  $x_3$ .

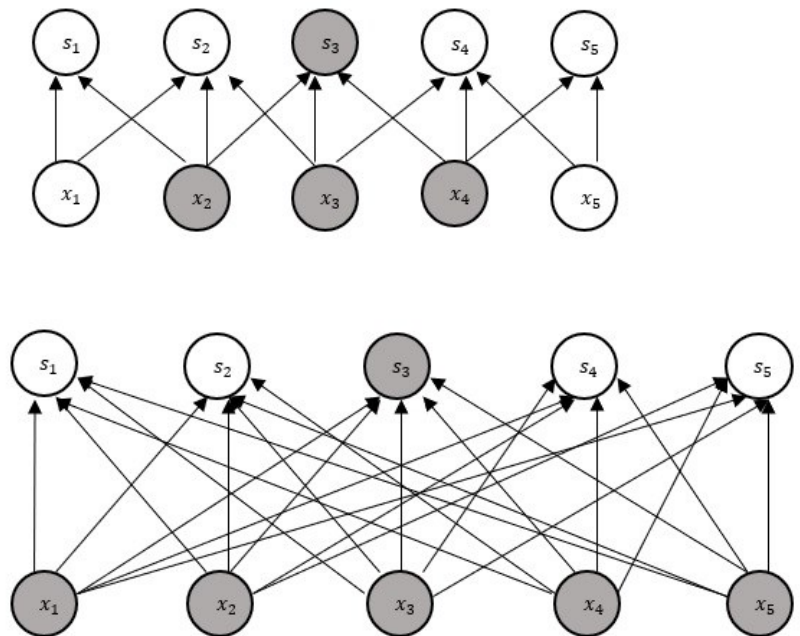


Figura 7 - Si tratta di una connettività sparsa con prospettiva dall'alto, si nota una sola unità di output indicato con  $s_3$  e le relative tre unità di input (indicate con  $x$ ) che influenzano l'unità di output.

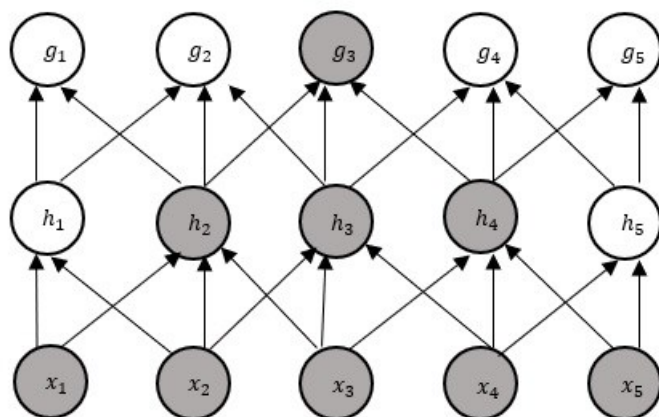


Figura 8 - Esempio di rete neurale convoluzionale più profonda, in cui si può evidenziare che le unità negli strati più profondi possono interagire indirettamente con una maggiore quantità di input.

La condivisione di parametri significa che in un modello si cerca di usare lo stesso parametro per più di una funzione. In particolare, ogni membro del nucleo viene usato in ogni posizione dell'input e il parametro di condivisione impiegato dall'operazione di convoluzione che invece di apprendere un insieme separato di parametri per ogni posizione (come avviene in una rete neurale tradizionale), viene appreso un unico set di parametri.

Questa peculiare forma di parametro di condivisione altera lo strato per ottenere una proprietà chiamata *equivalenza* alla traduzione.

Una funzione si dice equivalente, se quando varia l'input varia alla stessa maniera anche l'output.

E si può dire, che data una funzione  $f(x)$  è equivalente alla funzione  $g$ , espressa nel seguente modo:

$$f(g(x)) = g(f(x)).$$

In pratica, andando ad elaborare dei dati di una serie temporale, la convoluzione non fa altro che rappresentare una linea temporale delle funzionalità che caratterizzano l'input, ciò significa che se nell'input spostiamo un evento avanti nel tempo, questo verrà mostrato esattamente nella rappresentazione dell'output poco dopo nel tempo.

Cioè la modifica effettuata nell'input è visibile chiaramente nella serie temporale ottenuta in uscita.

### 3.3 Pooling Layer

Nelle reti neurali convoluzionali uno strato tipico prende il nome di **pooling**, il quale si compone di tre step fondamentali:

1. lo strato riproduce diverse convoluzioni in parallelo al fine di realizzare un insieme di attivazioni lineari;
2. Ognuna di queste attivazioni lineari è eseguita con una funzione di attivazione non lineare, questa fase prende il nome di rilevatore (dall'inglese “*detector* stage”);
3. Nell'ultimo e terzo stadio, viene impiegata una funzione pooling per modificare di nuovo l'uscita dello strato.

Qui di seguito è riportata l'immagine della struttura del pooling layer, che può essere descritta sotto due comuni punti di vista, nella struttura di sinistra si nota che la rete è rappresentata come un piccolo numero di strati relativamente complessi, mentre nella struttura di destra viene mostrata come un ampio numero di semplici strati.

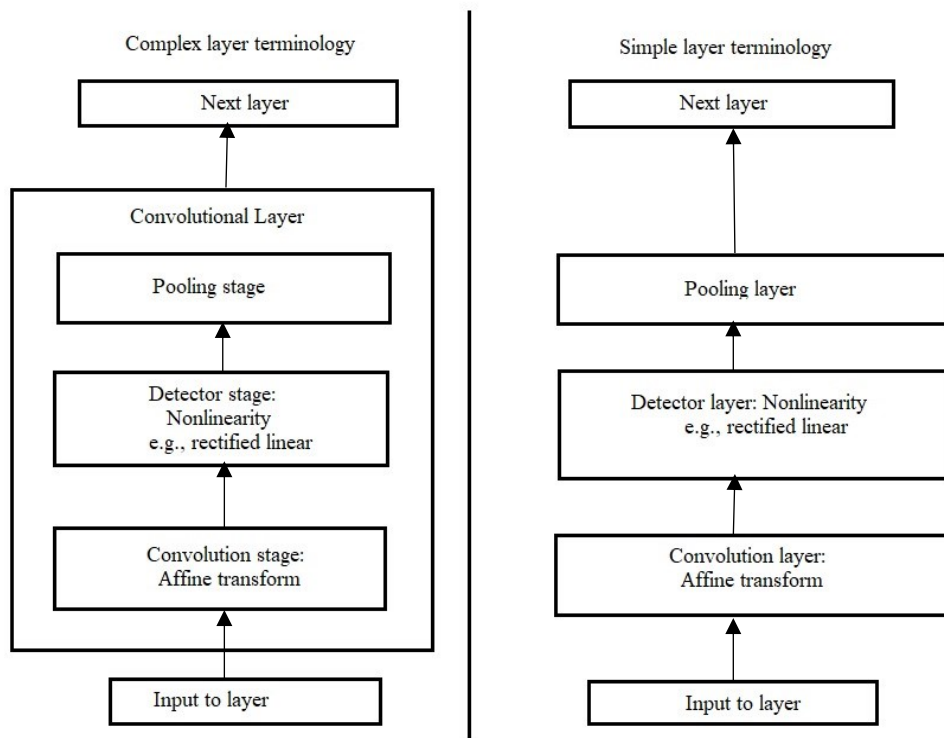


Figura 9 - Struttura delle componenti del tipico strato di una rete convoluzionale caratterizzato dalla funzione di pooling (immagine tratta dal libro “Deep Learning” di I. Goodfellow, Y. Bengio, A. Courville)

La funzione di pooling ci permette di sostituire l'output della rete considerata con un riassunto della statistica degli output vicini.

Se vogliamo citare un esempio noto di funzione di pooling, potrebbe essere una media ponderata basata sulla distanza da un pixel centrale in un'immagine.

Questa funzione è davvero importante per migliorare la costruzione della rappresentazione facendola diventare in maniera approssimativa invariante per traduzioni molto piccole degli input.

Quando questa funzione è corretta migliora notevolmente la statistica di efficienza della rete alla quale viene implementata.

### **3.4 Struttura degli output e tipi di dato.**

Il compito di una rete neurale convoluzionale (CNN), è quello di essere impiegate nell'analisi di immagini visive, a tal scopo in questo paragrafo illustriamo un'idea generale, senza soffermarci nel dettaglio, di quelle che sono: la struttura degli output in uscita e delle caratteristiche principali dei tipi di dato utilizzati.

La struttura di un output di una rete neurale convoluzionale ci permette di realizzare un formato alto, un'oggetto strutturato, in alternativa ad una semplice etichetta di classe per identificare una certa attività di classificazione. Si tratta di elaborare un tensore  $\mathcal{S}$ , che espresso come  $S_{i,j,k}$ , indica la probabilità che il pixel  $(j,k)$  dell'input nella rete appartenga alla classe  $i$ . Ciò permette a tale modello di etichettare ciascun pixel appartenente ad un immagine e disegnare maschere precise che riproducono i contorni dei singoli oggetti presenti nella figura, che dovranno essere riconosciuti dall'intelligenza artificiale.

Nel riconoscere oggetti dalle immagini analizzate, questa tipologia di reti neurali vengono costruite sulla base di un insieme di dati specifici, ad esempio dataset di animali prevedono immagini di animali, come dataset di riconoscimento includono immagini di volti. Questo al fine di dare la possibilità alla macchina di riconoscere tramite le caratteristiche degli oggetti presenti in quelle immagini (come larghezza, altezza, ampiezza) il soggetto da identificare.

Ad esempio non potremo mai ottenere una risposta positiva da un macchina se facciamo

analizzare immagini di animali ad una rete neurale convoluzionale che identifica soltanto le immagini contenenti volti umani, questo perché non riconosce le forme, dimensioni e oggetti rappresentati in esse.

### **3.5 Riduzione dei costi di una rete neurale convoluzionale.**

Uno svantaggio delle reti neurali convoluzionali è che risultano molto dispendiose nella fase di apprendimento delle caratteristiche. Al contrario lo strato di output mi garantisce una maggiore gestibilità (cioè risulta meno costoso) perché presenta un minor numero delle funzionalità date come input in seguito all'aver attraversato alcuni livelli di pooling.

Risulta possibile diminuire i costi di tale tipologia di rete neurale utilizzando durante la fase di addestramento (in cui vengono apprese le caratteristiche per riconoscere una certa tipologia di immagini) funzionalità che non sono state addestrate in modo supervisionato. Ci sono tre modalità da tenere presenti:

1. Il primo è quello di inizializzare i kernel della convoluzione casualmente;
2. Un altro è quello di implementarli a mano e quindi, far in modo di impostare ogni kernel per trovare ad esempio i bordi su una certa scala;
3. Infine, si possono progettare i kernel con un criterio senza supervisione, ciò significa che si può far uso di un apprendimento non supervisionato per creare una rete neurale convoluzionale senza far mai utilizzo della convoluzione durante l'addestramento dei processi.

### **3.6 Principi Neuroscientifici delle reti convoluzionali.**

Le reti neurali convoluzionali o Convolutional Neural Networks (CNN), sono la tipologia di rete neurale che sta avendo più successo e che prende spunto da alcuni principi delle neuroscienze.

Non si vuole entrare nel dettaglio, in quanto nel capitolo cerchiamo di vedere quali sono gli aspetti importanti generali sulla loro struttura, funzionamento ed obiettivo, ma questo aspetto fornisce sicuramente un'ulteriore visione e punto di vista rispetto alle reti neurali.

Si cerca di riprodurre sostanzialmente come i neuroni del cervello riescono a rispondere a certi stimoli, in particolare vedendo certe immagini, ed analizzare quale sia la risposta.

Infatti, si è posta una particolare attenzione ad una parte del nostro cervello chiamata **corteccia visiva primaria** o anche indicata con V1. Questa è la prima area del nostro cervello in grado di analizzare ed elaborare in maniera significativa le informazioni percepite con l'impatto visivo (ossia come input).

La corteccia visiva primaria ha delle caratteristiche fondamentali che devono essere acquisite ed elaborate da una rete neurale convoluzionale, la quale è strutturata proprio per eseguire questo compito.

In primo luogo, essendo organizzata come una **mappa spaziale**, una struttura bidimensionale capace di riprodurre in maniera simile la struttura di un'immagine presente nella retina, e la rete convoluzionale riesce ad acquisire queste caratteristiche proprio perché composta come una mappa bidimensionale.

Inoltre, è formata da **celle semplici**, che possono essere costituite da una funzione lineare in un campo ricettivo di piccole dimensioni e circoscritto.

La fase in cui è presente il **rilevatore**, citata in uno dei paragrafi precedenti, è composto da unità in grado di emulare le proprietà di celle semplici presenti nella V1.

Altra proprietà di quest'area (V1), è quella di contenere anche delle celle complesse, che pur avendo caratteristiche che si avvicinano a quelle delle celle semplici rimangono invariante a piccoli spostamenti relativi alla posizione della funzione.

Questo è fondamentale per capire chiaramente come le reti neurali in generale cercano di riprodurre lo stesso funzionamento dei neuroni presenti nel cervello umano perciò richiedono un continuo perfezionamento, e nello specifico abbiamo visto quello che accade per le reti neurali convoluzionali.

Nell'intero capitolo, si è deciso prendere in considerazione proprio quest'ultime, in quanto sono il modello più richiesto al momento e con il quale si sono ottenuti ottimi traguardi nel campo del **deep learning**.





## Conclusioni

L'approccio simbolico e quello neurale, come si è potuto notare anche nei capitoli precedenti di questa tesi, erano volti a dare un'idea generale sulla struttura di queste due tipologie di apprendimento che si è deciso di mettere a confronto. Sono entrambe molto articolate ed operano in maniera differente, ma nonostante questo hanno ciascuna un obiettivo principale cioè quello di offrirci una soluzione valida e corretta che può aiutare l'essere umano a superare i propri orizzonti e limiti (di cui abbiamo accennato inizialmente).

Ora possiamo vederne gli aspetti positivi e negativi, cercare di elaborare una conclusione valida sui dati e informazioni appresi.

Nell'apprendimento simbolico, la cui rappresentazione analizzata è quella degli alberi di decisione, cioè un apprendimento supervisionato che attraverso specifici passi, riesce ad identificare una possibile soluzione appresa dai vari stimoli che gli sono stati forniti. La sua struttura gli permette di fornire una diagnosi, o soluzione, logica ed affidabile quanto più possibile, verificando quale sia la percentuale di volte che l'algoritmo ha effettivamente prodotto la risposta corretta. E questo ci dà una garanzia sulla forma di apprendimento utilizzata per elaborarla e renderla praticamente coerente.

In altre parole, sono dei metodi di apprendimento che ci permettono di esaminare i criteri utilizzati ed i passi effettuati per arrivare ad una certa decisione piuttosto che un'altra. Non si può di certo pensare di dare per scontata una decisione solo perché lo ha stabilito la macchina, ma entrano in gioco altri fattori umani che ci danno il diritto di avere una spiegazione logica rispetto alla soluzione ottenuta. E questo genere di apprendimento ce lo può garantire a differenza delle reti neurali, che sono in fase di perfezionamento, e ciò vale prevalentemente per ambiti come la medicina.

Tutto questo però non significa che le reti neurali siano da escludere a priori, anzi rappresentano un campo dell'apprendimento automatico molto vasto, che in un certo senso rappresenta l'evolversi del futuro e della nostra quotidianità, ma questo non deve di certo spaventarci.

Non a caso, si sono trattate le reti neurali convoluzionali (Convolutional Neural Networks) che sono una tipologia di rete neurale utilizzata per l'acquisizione di oggetti presenti nelle immagini, divise ovviamente per tipologie (come già citato nello scorso capitolo), e che negli ultimi anni hanno avuto molto successo. La loro struttura è particolarmente complessa, perché a differenza di un apprendimento simbolico, basato su passaggi logici che la macchina deduce attraverso diversi tentativi, qui si tratta di far apprendere alla macchina la struttura di certi oggetti, imparando a riconoscerli solamente attraverso immagini.

E tutto ciò avviene senza passi logici, ma la stessa macchina deve essere in grado di sviluppare una propria "coscienza" e, per questa ragione diventa più difficile affidarsi alla sua risposta, o diagnosi finale. Si può intuire quindi, che se questa tipologia di apprendimento in futuro riesca a fornirci delle risposte corrette e specifiche possono essere una grande risorsa sotto molti aspetti. Perciò la ricerca e lo sviluppo, oggi giorno stanno puntando molto in questa direzione con lo scopo di ottenere e raggiungere criteri, algoritmi sempre più precisi e affidabili.

A fortificare queste considerazioni, è anche un articolo molto interessante chiamato "The Bitter Lesson" di Rich Sutton (pubblicato il 13 marzo 2019), nel quale vengono specificati alcuni aspetti davvero avvincenti.

Ciò che ha portato ad un notevole cambiamento nei decenni rispetto a questa disciplina, quale l'Intelligenza artificiale, è stato proprio l'aver posto più attenzione ad una migliore implementazione e ricerca dei software ed hardware per migliorare le prestazioni di calcolo. Fondamentale l'argomento che viene affrontato in questo articolo è che, la struttura di un algoritmo basato sulla conoscenza umana inizialmente aveva garantito ottimi successi, in diverse occasioni nel confronto tra più macchine. Con l'avvento della ricerca e apprendimento che pone la sua attenzione sulle potenzialità e capacità di calcolo, le precedenti macchine sono risultate meno efficienti, o meglio in caso di successo, non erano affatto paragonabili a quelle implementate con criteri statistici. Questo perché agenti implementati rispettando algoritmi che riproducessero il ragionamento umano risultavano più complessi e non garantivano la semplicità in un algoritmo, e ciò si ripercuote anche sull'aspetto economico delle fonti messe a disposizione per la loro costruzione, ed inoltre, anche sul tempo impiegato per realizzarle.

Anche se tuttora non si è arrivati a conoscere in maniera specifica i precisi meccanismi matematici che guidano l'apprendimento profondo, non vuol dire che si stia sperimentando lungo una direzione sbagliata. Anzi, è bene approfondire e conoscere al meglio gli errori che si compiono tuttora, per poterli gestire, migliorare e far in modo in futuro di poterli evitare.

L'aspetto da considerare che comporta una notevole differenza nell'efficienza di un algoritmo e la morale che si trae dagli errori commessi (posto che se ne avranno altri in futuro), ma da cui trarre vantaggio, è che le nostre menti devono fornire i metodi e strumenti affinché gli agenti intelligenti in modalità autonoma (cioè attraverso l'applicazione di metodi statistici) siano capaci di sviluppare un pensiero. Bisogna evitare che la ricerca delle informazioni avvenga con il nostro stesso modo di pensare, perché non si farà altro che generare complicazioni ed errori ulteriori.



## **Fonti Bibliografiche e Sitografia**

Bengio Y., Courville A., Goodfellow I., *Deep Learning*, 2017.

Bratko I., *Programmare in prolog per l'intelligenza artificiale*, 1988.

Dragoni A. F., *Slides del corso "Ingegneria Artificiale"*, 2017.

Sutton R., *The Bitter Lesson*, 13 marzo 2019.

[www.treccani.it](http://www.treccani.it), consultati per vocaboli e sinonimi.

[www.deepmind.com](http://www.deepmind.com), consultato per fornire alcuni esempi.