Università degli Studi di Napoli “Federico II”

Scuola Politecnica e delle Scienze di Base
Area Didattica di Scienze Matematiche Fisiche e Naturali

Dipartimento di Fisica “Ettore Pancini”

Laurea triennale in Fisica

Reti neurali convoluzionali per la classificazione di impronte di scarpe sulla scena del crimine

Relatore:
Dott.ssa Autilia Vitiello

Candidato:
Claudia Costantini
Matricola N85001046

A.A. 2020/2021

Indice

Introduzione

Lo sviluppo dell'intelligenza artificiale ha portato all'introduzione di algoritmi in grado di imitare i processi e le funzioni degli organismi viventi, facendo sì che una macchina potesse imitarli al fine di risolvere problemi dotati di una grande quantità di dati in un tempo finito. Esistono diverse tipologie di questi algoritmi, in particolare in questo lavoro di tesi si è approfondito il machine learning ponendo l'attenzione sui neuroni artificiali e le reti neurali artificiali mostrando una loro applicazione pratica nel campo della scienza forense.
Il campo della scienza forense comprende tante branche della scienza, le quali forniscono delle tecniche da utilizzare nell'analisi di tutti gli elementi di una scena del crimine, essi possono essere ad esempio: oggetti materiali oppure tracce biologiche, oppure impronte di scarpe. Queste analisi hanno come fine ultimo quello di ricostruire l'accaduto per poter, poi, arrivare alla soluzione di un qualsiasi tipo di caso.
Lo scopo di questo lavoro di tesi è proprio quello di usare due reti neurali convoluzionali per analizzare le impronte di scarpe su una scena del crimine, nel dettaglio le reti sono in grado di predirre il sesso e la misura del proprietario delle impronte di scarpe, in questo modo gli investigatori sarebbero più facilitati nella ricerca dei sospettati e nell'identificazione del colpevole.
Una delle due reti neurali artificiale è stata utilizzata per affrontare il problema di classificazione binaria (sesso del proprietario delle impronte), mentre l'altra è servita a risolvere il problema di tipo multiclasse (misura delle scarpe del proprietario delle impronte).
La struttura di questo lavoro di tesi è la seguente:

- Nel primo capitolo viene dato spazio al campo di applicazione della rete neurale, ovvero si parla brevemente della scienza forense. In particolare, viene definita la scienza forense e una parte dei suoi settori. Si passa, successivamente, a definire la scena del crimine e ad esporre i metodi per esaminarla; si passano in rassegna le azioni da eseguire su di essa tra cui fotografare la scena e creare un suo schizzo. In seguito, è riportata la definizione di prova e viene fornito anche un suo esempio.

- Nel secondo capitolo si è posta l'attenzione sul machine learning spiegando le sue diverse tipologie. È stato approfondito l'apprendimento supervisionato, in particolare sono state esposte tutte le fasi di questa tipologia di algoritmo per, poi, passare alla trattazione in dettaglio del neurone artificiale, della rete neurale Multilayer Perceptron e della rete neurale convoluzionale.

- Nel terzo e ultimo capitolo si è introdotto il set di dati utilizzato per i due tipi di classificazione effettuati. Dopo sono state esposte le varie operazioni di preprocessing fatte sul dataset, oltre a questo è stato descritto brevemente Tensorflow dato che è stato ampiamente utilizzato. Infine sono stati esposti i risultati ottenuti dalla classificazione delle due reti neurali, specificando che la seconda rete è una semplice modifica della prima.

Capitolo 1

Scienza forense

1.1 Cos'è la Scienza Forense

L'espressione "Scienza forense"indica una disciplina che mette al servizio della legge e della giustizia metodi e tecniche scientifiche appartenenti a qualsiasi campo.
L'obiettivo della scienza forense consiste nel collegare tra loro persone, luoghi e fatti in modo tale da risolvere un crimine di qualsiasi tipo. Essa ha un ruolo cruciale durante la fase di investigazione nella quale è esaminata la scena del crimine e sono raccolte le prove fisiche da esaminare con lo scopo di ricostruire l'accaduto, trovare i sospettati e infine identificare il colpevole.
Lo scienzato forense ha due principali compiti: analizzare le prove e testimoniare durante un processo civile o penale.
In conseguenza di quanto detto in precedenza, la scienza forense si serve di diversi campi scientifici per cui ha bisogno della presenza di figure specializzate nei seguenti campi:

- Patologia forense: essa fornisce la possibilità di conoscere la causa e una stima dell'ora del decesso della vittima, in casi in cui esso avvenga in circostanze misteriose attraverso delle analisi condotte sul corpo da un medico legale.

- L'antropologia forense viene utilizzata nei casi in cui non è possibile identificare una persona; attraverso le analisi di resti l'antropologo forense è in grado di decretare se si tratta di resti umani, il sesso e l'età dell'eventuale persona in questione. In particolare essa viene impiegata nei casi di persone scomparse verficando se le caratteristiche emerse dallo studio dei resti coincidono con quelle della persona scomparsa.

- L'odontologia forense è collegata all'odontoiatria. Spesso in caso di stragi di massa oppure omicidi si riescono a ritrovare denti intatti dato che lo smalto è un materiale molto duraturo; ciò può essere utile per trovare prove contro il presunto colpevole o per ricostruire quello che è accaduto.

- L'ingegneria forense si occupa dello studio dei materiali o degli oggetti che non funzionano come dovrebbero, nel dettaglio si deve trovare la causa di questo malfunzionamento. Viene usata sia nei casi di tipo civile che in quelli di tipo penale.

- La tossicologia si occupa della ricerca di droghe o veleni nel tessuto o nei liquidi corporei.

Data la diversità di ogni indagine, in realtà, potrebbero essere necessarie altre figure specializzate in altri campi non citati precedentemente.

1.2 Analisi della scena del crimine

Attraverso l'analisi di una scena del crimine gli investigatori sono in grando di capire quali degli oggetti ritrovati possono essere considerati delle prove, in quanto esse sono uno strumento necessario per smentire o confermare un'ipotesi avanzata.
Tra le scene del crimine e i siti archeologici c'è molta somiglianza: entrambe conservano tracce di attività umana del passato attraverso il ritrovamento di resti; il compito degli archeologi, analogo a quello degli investigatori, è quello di capire il contesto e le relazioni tra gli oggetti presenti nel sito, per fare ciò si misurano le posizioni degli oggetti rispetto ad un punto di riferimento permanente e poi deve essere trovata la posizione di ogni oggetto rispetto al materiale che li circonda che può essere il suolo, l'acqua o anche un soggiorno di una casa nel caso di una scena del crimine. Quanto descritto in precedenza è simile al processo utilizzato dagli investigatori per individuare tutti gli oggetti sulla scena del crimine che sono considerati delle prove.
Analizzare una scena del crimine oppure un sito archeologico è, dunque, un processo irreversibile poichè quando un reperto viene rimosso oppure una prova viene raccolta viene effettuato sul luogo un cambiamento permanente e non c'è alcun modo per poterlo riportare alla condizione originaria, quindi è un processo di "distruzione" eseguito in maniera molto cauta perchè qualsiasi errore potrebbe compromettere lo studio dello scavo archeologico oppure la risoluzione del caso investigativo.

1.3 Come indagare sulla scena del crimine

In base al tipo di crimine commesso, al luogo in cui esso avviene, all'oggetto usato cambia la scena del crimine ed è per questo motivo che essa è unica nel suo genere. Esistono, quindi, dei protocolli o delle regole che gli investigatori devono seguire ma alcune situazioni particolari richiedono la loro flessibilità e creatività.
Appena l'investigatore arriva sulla scena del crimine deve assolutamente chiudere la zona di interesse per evitare che venga distrutto o alterato il contesto della scena del crimine che, in base a quanto detto nel paragrafo precedente, non può essere più riportato a com'era in precedenza se manomesso in qualche modo. Altri doveri dell'investigatore sono: trattenere i principali sospettati, non contaminare, distruggere o aggiungere altre prove e cercare di fermare chiunque possa fare quanto appena detto.
Oltre ad identificare le zone che fanno parte della scena del crimine, bisogna anche identificare i suoi confini e solo dopo possono partire le indagini e le analisi del caso.
Dopo la delimitazione della scena del crimine bisogna sempre sapere chi entra e chi esce da quest'ultima ed è, inoltre, necessario prendere nota delle condizioni fisiche, ambientali e di tutte le azioni effettuate. Gli appunti devono essere presi sia da un supervisore sia da chi svolge il compito assegnatogli ed entrambi devono indicare l'orario e il compito da svolgere in modo tale che se sorgesse qualche domanda in tribunale sull'attività svolta, essa può essere confermata dal confronto delle note scritte dalle due persone diverse.
Dopo le fasi menzionate in precedenza, inizia l'indagine preliminare che consiste in un lavoro di squadra nel quale vengono scattate delle fotografie della scena del crimine e sono presi appunti su caratteristiche peculiari della scena o di alcune prove. Dovrà essere successivamente eseguita la ricerca delle prove, che non si limita soltanto a quelle più evidenti ma si estende anche a quelle situate nei posti più nascosti che fanno parte della scena del crimine.

1.3.1 Fotografare la scena del crimine

Fotografare la scena del crimine è una delle prime cose da fare poichè permette di documentare ogni stadio dell'investigazione. Tutte le foto scattate vengono poste in un registro insieme alla descrizione del soggetto rappresentato e alla posizione di quest'ultimo nella scena del crimine.
Ai fini di un'accurata ricostruzione devono essere scattate molte foto e con modalità differenti, esse sono: la distanza da cui le foto vengono scattate, le diverse angolazioni di scatto e la scelta di una scala.

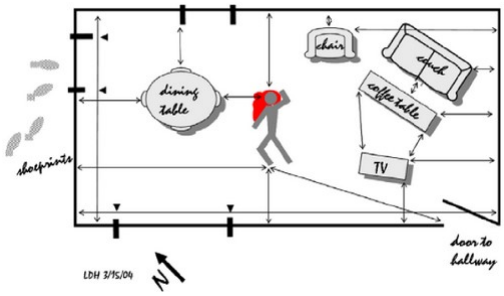


Figura 1.1: Rappresentazione di un bozzetto della scena del crimine. [1]

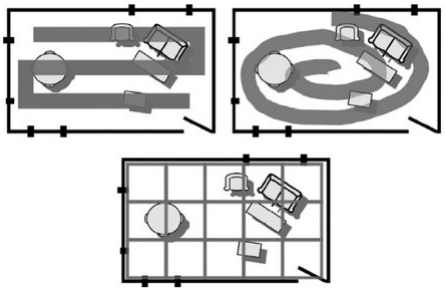


Figura 1.2: Rappresentazione delle possibili modalità schematiche di ricerca delle prove. [1]

1.3.2 Disegnare la scena del crimine

Lo schizzo di una scena del crimine è complementare alle foto. Esso è rappresentato in figura 1.1 ed è un disegno non in scala che contiene le seguenti informazioni: le distanze tra oggetti, la temperatura, la data, la condizione della luce, le dimensioni di stanze, mobili, finestre e una freccia che indica la direzione del polo nord magnetico.

1.3.3 Come raccogliere le prove

Per ricercare le prove gli investigatori non si muovono in maniera casuale sulla scena del crimine ma seguono uno schema ben preciso la cui scelta dipende dalla dimensione o dalla condizione in cui si trova la scena del crimine. La scelta può ricadere su uno schema a griglia, a spirale o a corsia (figura 1.2). Dopo aver trovato le prove bisogna misurare la loro posizione più volte rispetto a diversi oggetti fissi in modo tale da poterle riportare sullo sketch. Le misure prese rispetto a due punti di riferimento diversi servono per la triangolazione che consiste nel calcolare la distanza di un lato del triangolo conoscendo gli angoli e la lunghezza degli altri due lati, nel nostro caso le

lunghezze corrispondono alle misure prese sulla scena del crimine. La triangolazione forense, menzionata in precedenza, serve a localizzare le prove e aiuta a creare una ricostruzione accurata dei fatti.

1.3.4 Ispezione finale della scena del crimine

Questa fase finale è l'ultimo sopralluogo effettuato sulla scena del crimine in modo tale da verificare se agli investigatori sia sfuggito qualche dettaglio. Vengono controllati documenti, appunti presi, si portano via tutte le prove raccolte e si scattano foto della condizione finale della scena del crimine.
Se nessun'altro specialista ha bisogno di fare ulteriori analisi sulla scena del crimine quest'ultima viene aperta, terminando ufficialmente il lavoro di ispezione.

1.4 Cos'è una prova

Nella risoluzione di un caso investigativo le prove raccolte giocano un ruolo fondamentale poichè hanno un duplice impiego:

- Ricostruire il contesto della scena del crimine grazie alla presenza delle prove fisiche.
- Dimostrare la veridicità di un'affermazione.

Le prove sono rappresentate da i resti ottenuti da un transferimento di informazioni avvenuto tramite il contatto tra due oggetti, tra due persone o tra un oggetto e una persona, quindi per capire come nasce una prova bisogna porre l'attenzione sulle rimanenze causate dal trasferimento dell'informazione e non sul transferimento in sè. Questa teoria costituisce una delle linee guida della scienza forense e fu sviluppata all'inizio del ventesimo secolo da Edmund Locard, scienzato forense francese.
Esistono due tipi di trasferimento di informazione: diretto e indiretto.
Per comprendere quanto affermato in precedenza è utile fare un esempio: si ponga il caso che un cane e il suo padrone giochino insieme ogni giorno prima che lui vadi al lavoro come conseguenza dell'interazione si origina una prova, infatti i resti dell'interazione potrebbero essere dei peli di cane che si posano sui pantaloni del proprietario e questo trasferimento dell'informazione è di tipo diretto. Successivamente, quando il padrone del cane va al lavoro può sicuramente trasferire i peli del suo cane sulla sedia su cui è seduto e se qualcun'altro si sedesse su quella sedia avrebbe la possibilità di trovarsi dei peli di cane sui pantaloni; in questo caso il tipo di trasferimento è indiretto. I

trasferimenti indiretti sono molto più difficili da interpretare rispetto a quelli diretti e potrebbero portare anche a dei fraintendimneti.
In seguito all'individuazione della scena del crimine ulteriori trasferimenti di informazioni tra oggetti e persone o tra oggetti stessi potrebbero contaminare le prove e ciò potrebbe sviare le indagini.
L'altra parte del processo di transferimento è la persistenza, ovvero la capacità di una prova di essere presente in un determinato punto fino a quando non viene raccolta o fino a quando non si consuma.
Dopo aver capito la teoria secondo cui si generano le prove, si può passare a parlare delle loro diverse tipologie.
Se nella ricerca delle prove emergono capelli, impronte digitali, impronte di scarpe oppure sangue, allora queste prove sono dette fisiche. Esistono anche delle prove chiamate dimostrative non trovate sul luogo del crimine ma generate successivamente con lo scopo di spiegare il significato di una specifica prova fisica, ad esempio: un diagramma sulle caratteristiche dei capelli può aiutare a capire testimonianze più complicate durante il processo.

1.4.1 Un esempio comune: le impronte di scarpe

Tramite le impronte di scarpe gli investigatori sono in grado di capire quante persone erano presenti sulla scena del crimine.
In alcuni casi, esse costituiscono delle prove cruciali poichè collegano direttamente il sospettato con l'incidente.
Esistono sia le impronte di scarpe bidimensionali sia quelle tridimensionali.
Il primo tipo di impronta è generato dalla pressione di una scarpa su un suolo piatto e duro. Se viene trasferito del materiale sul suolo calpestato l'impronta viene chiamata positiva, nel caso contrario in cui il materiale viene portato via, ad esempio su un suolo in cui è presente polvere, l'impronta è chiamata negativa.
Nella maggior parte dei casi le impronte si deteriorano a tal punto da non essere visibili ad occhio nudo, per questo motivo vengono usate delle tecniche per renderle più evidenti. Possono essere usate delle luci per illuminare tutte le impronte nascoste in modo che esse vengano fotografate.
Le impronte tridimensionali sono generate quando una scarpa calpesta un suolo morbido come la neve o la sabbia. Così come quelle bidimensionali anche queste impronte devono essere subito fotografate e inoltre per poterle, ulteriormente, analizzare si può creare un calco a partire dalla loro forma impressa nel suolo.
Questo esempio di prova non è stato citato a caso, perchè le impronte di scarpe saranno utilizzate nello studio effettuato in questo lavoro di tesi.

Capitolo 2

Algoritimi di machine learning

2.1 Machine learning

Il machine learning è un settore dell'intelligenza artificiale che utilizza degli algoritmi in grado di apprendere delle conoscenze dai dati fornitegli, al fine di eseguire delle predizioni future. Il suo sviluppo è l'evidente conseguenza dell'enorme quantità di dati presenti al giorno d'oggi; per questo motivo gli esseri umani non sono più in grado di analizzarli manualmente ma affidano questo compito ad un computer che tramite un algoritmo di autoapprendimento riesce ad analizzare una mole gigante di dati, strutturati e non strutturati, dandogli un significato e apprendendo la loro struttura nascosta.
Nel machine learning sono presenti tre tipi di apprendimento:

- Apprendimento supervisionato. In questa tecnica si utilizzano dati etichettati in modo da tale da avere un riscontro diretto per poter fare predizioni corrette in futuro.
- Apprendimento non supervisionato. In questo caso i dati sono senza etichette e bisogna trovare la struttura presente in essi senza avere a disposizione alcun tipo di riscontro.
- Apprendimento per rinforzo. Si utilizza un sistema decisionale basato sulla ricompensa che si ottiene dopo aver effettuato un'azione in un determinato ambiente, quindi si impara da queste ultime.

2.1.1 Apprendimento supervisionato

I dati forniti agli algoritimi di apprendimento supervisionato sono chiamati dati di addestramento e la loro caratteristica fondamentale, come anticipato in precedenza, è quella di essere dotati di etichette le quali rendono possibile

la determinazione dell'appartenenza di un dato ad una classe. L'obiettivo di questi algoritimi riguarda la ricerca di un modello che è in grado di classificare i dati di addestramento e di fare predizioni corrette su nuovi dati, come mostrato sottoforma di schema in figura 2.1.

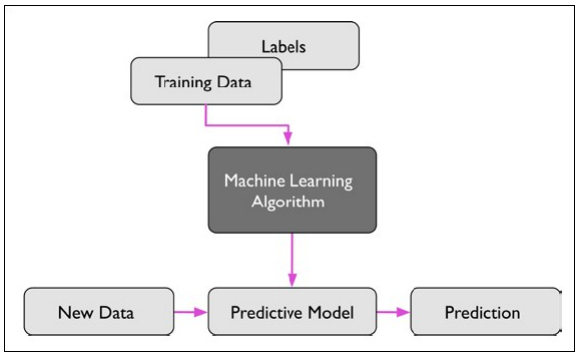


Figura 2.1: Sono rappresentati schematicamente tutti gli elementi necessari per l'apprendimento supervisionato.[4]

Una sottocategoria degli algoritimi di apprendimento supervisionato è costituita dagli algoritmi di classficazione, il loro compito consiste nel determinare l'appartenenza di una nuova istanza ad una classe basandosi sulle classificazioni effettuate sui dati di addestramento. Esistono due casi differenti: la classificazione binaria e quella multiclasse.
Nel primo caso i dati di addestramento sono etichettati in modo da appartenere a due classi opposte, come ad esempio in un algoritmo che deve classificare le e-mail in spam e non-spam. All'algoritmo dell'esempio vengono forniti dei dati e ogni campione è rappresentato attraverso i valori x_1 e x_2 nel grafico in figura 2.2, dove si può notare la presenza di una linea tratteggiata che rappresenta sia il confine di separazione tra le due classi sia la regola di classificazione che l'algoritmo ha appreso dai dati stessi. In questo modo l'algoritmo sarà in grado di classificare nuovi dati attraverso le loro caratteristiche rappresentate dai valori x_1 e x_2.
Riferendosi sempre alla figura 2.2 le e-mail appartenenti alla classe spam sono indicate col simbolo "-" mentre quelle appartenenti alla classe non-spam vengono denotate dal simbolo "+" . Si può notare che qualche punto denotato dal simbolo "-" oppure dal "+" è più spostato verso la linea di confine rispetto agli altri, questo vuol dire che le caratteristiche dei dati suggerivano sia la classificazione in non-spam che in spam quindi la classe di appartenenza viene decisa in base alla caratteristica predominante. Se ci fosse stata la presenza di un punto "+" o "-" sulla linea di confine allora

l'algoritmo non sarebbe stato in grado di svolgere in modo giusto il compito di classificazione.

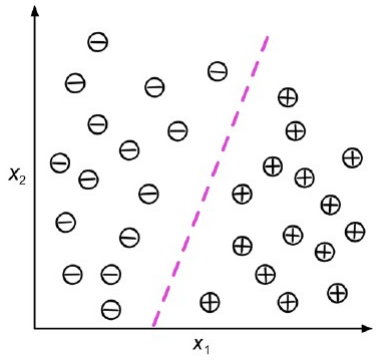


Figura 2.2: Nel grafico, x_1 e x_2 rappresentano i valori delle caratteristiche dei dati.[4]

A volte si possono presentare delle situazioni in cui la classificazione non è più binaria ma esistono diverse etichette, in questo caso il problema che l'algoritmo deve risolvere viene detto multiclasse. Un tipico esempio di problema multiclasse consiste in un algortimo che è in grado di classificare le cifre numeriche scritte a mano.
Un'altra sottocategoria di algoritmi di apprendimento supervisionato consiste nell'analizzare la regressione. Le predizioni che vengono effettuate con questa tecnica sono di tipo continuo a differenza della classificazione, dove le etichette da assegnare alle nuove istanze erano rappresentate da dei valori discreti.
L'analisi della regressione consiste nel capire la relazione che c'è tra le variabili misurate e le variabili di risposta.

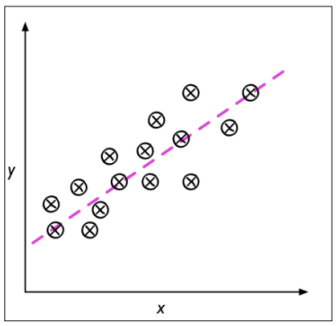


Figura 2.3: Nel grafico viene mostrata la retta di fit per i dati di addestramento misurati indicati con x e la loro risposta indicata con y.[4]

Nel grafico della figura 2.3 viene tracciata una retta di fit per minimizzare la distanza media al quadrato tra i punti che rappresentano i dati e la retta di fit stessa, l'intercetta e la pendenza ricavata dai dati verrà usata per la predizione sui nuovi dati.

2.1.2 Apprendimento per rinforzo

Nell'apprendimento per rinforzo un sistema agisce sull'ambiente esterno cambiando il proprio stato e queste interazioni servono a migliorare le sue prestazioni. L'apprendimento per rinforzo può essere considerato parte dell'apprendimento supervisionato in quanto l'interazione tra il sistema e l'ambiente è rappresentata da un segnale di reward che costituisce un feedback, esso non corrisponde più al valore corretto dell'etichetta ma ad una misura di quanto l'azione sia corretta.
In questo tipo di algoritmi l'obiettivo consiste nel rendere massimo il segnale di reward attraverso tutte le interazioni avute tra il sistema e l'ambiente.

2.1.3 Apprendimento non supervisionato

Questo tipo di apprendimento viene utilizzato quando non si hanno a disposizione né dati etichettati né funzioni di reward da massimizzare.
Lo scopo è estrarre delle informazioni utili e dei possbili pattern nascosti tramite l'esplorazione dei dati stessi, per fare ciò si può utilizzare o il clustering o la riduzione della dimensionalità dei dati.
Il clustering ci permette di analizzare i dati e di suddivederli in sottogruppi, detti cluster comuni. Un dato che appartiene ad un cluster avrà delle caratteristiche somiglianti con gli altri dati appartenenti allo stesso cluster. Una volta creati i cluster i loro dati possono essere, ad esempio, etichettati.
La riduzione della dimensionalità serve a ridurre le dimensioni dei dati escludendo le caratteristiche non rilevanti per lo specifico problema da risolvere. Inoltre, questa tecnica è utile in quanto con dati che hanno un minor numero di dimensioni ci si può preoccupare di meno dello spazio di archiviazione limitato che ha un computer e viene resa più facile la visualizzazione dei dati stessi.

2.2 Fasi di un algoritmo di machine learning

La struttura dei dati è una matrice le cui colonne corrispondono alle caratteristiche dell'oggetto da classificare ad eccezione dell'ultima dove è indica-

ta l'etichetta della classe di appartenenza mentre le righe corrispondono al numero di campioni che sono stati presi in considerazione per le misure.
Il preprocessing è una procedura che viene eseguita sui dati grezzi per renderli nella giusta forma in modo tale che l'algoritmo di machine learning possa funzionare al meglio. Attraverso questa procedura dai dati grezzi si possono estrarre le caratteristiche significative per il problema specifico affrontato, ad esempio: nel dataset iris le caratteristiche relative alla lunghezza e alla larghezza dello stelo o del petalo del fiore potrebbero essere state estratte da una serie di immagini dei fiori.
Le caratteristiche devono, poi, essere riscalate in un intervallo di valori compresi tra [0,1] oppure normalizzate con media zero e varianza unitaria.
In seguito, per migliorare la performance dell'algoritmo i dati ottenuti dal preprocessing devono essere divisi in modo casuale in almeno due gruppi: il dataset di training e il dataset di test; anche se nella maggior parte dei casi per avere un riscontro della performance dell'algortimo mentre si esegue si usa un terzo dataset detto di validation.
Vi sono altre due fasi di un algoritmo di machine learning: l'addestramento e la valutazione del modello.
L'addestramento è la fase in cui viene scelto il tipo di algoritmo da utilizzare per creare il modello più adatto alla risoluzione del problema in esame. Un modo per capire quale sia il miglior modello è scegliere una metrica che rifletta le performance dell'algoritmo impiegato, a tal proposito un parametro molto comune è l'accuratezza della classificazione definita come il numero di predizioni corrette sul numero di campioni totali. Dato che il test set viene utilizzato nella fase di valutazione non è possibile sapere se durante l'addestramento l'algoritmo è in grado di generalizzare, quindi è adoperato un terzo dataset che è quello di validation , già citato in precedenza, che permette di avere un feedback sulla generalizzazione.
Un punto importante della fase di addestramento è la regolazione dei valori degli iperparametri, presenti negli algoritmi di apprendimento implementati nelle librerie. Ogni problema ha bisogno di una regolazione diversa di questi parametri in modo tale da migliorare al massimo le performance del modello scelto, per questo motivo il loro valore non è dedotto durante l'addestramento ma esistono delle tecniche per conoscere la migliore regolazione dei loro valori.
L'altra fase riguarda la valutazione del modello selezionato attraverso l'utilizzo del dataset di test.
I dati di test devono essere resi nella stessa forma di quelli di training per cui l'algoritmo prima di entrare nella fase di apprendimento eseguirà tutte le istruzioni della fase di preprocessing.
Se l'algoritmo ha delle buone performance allora il modello scelto sarà quello giusto per fare predizioni su dati futuri.

2.3 Introduzione al neurone artificiale

Il machine learning, di cui si è parlato in precedenza, è una delle tecniche dell'intelligenza computazionale capace di simulare la funzione di apprendimento di un organismo vivente. L'imitazione dei processi umani richiede la conoscenza della struttura del cervello umano in modo tale da porterla riprodurre con un modello matematico, da quest'esigenza nascono diversi modelli di neurone artificiale che hanno portato allo sviluppo delle reti neurali.
Dal punto di vista biologico il neurone è l'unità fondamentale del cervello umano, in figura 2.4 si può vedere che le sue componenti principali sono: i dendriti, il soma, gli assoni, le sinapsi e i bottoni sinaptici.

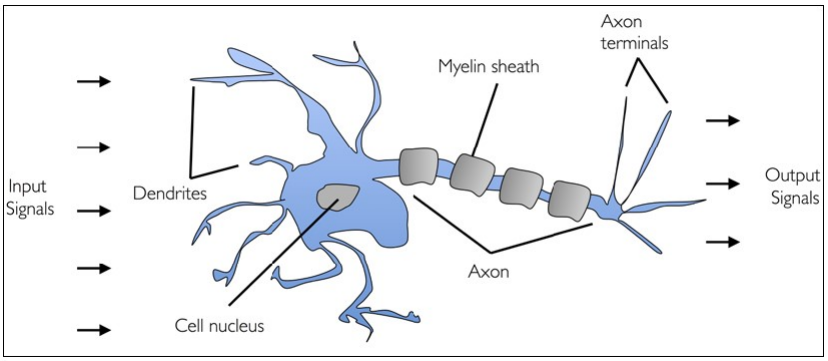


Figura 2.4: In quest'immagine è mostrato il meccanismo di passaggio del segnale elettrico attraverso le varie componenti del neurone.[4]

Nel cervello tutti i neuroni sono interconnessi tra loro tramite le sinapsi, protuberanze posizionate dopo l'assone che si incastrano perfettamente con le sporgenze dei dendriti di altri neuroni.
Il segnale elettrico attraverso cui si propaga l'informazione viene ricevuto dal neurone tramite i suoi dendriti e arriva nel soma o corpo cellulare dove viene elaborato. Nei bottoni sinaptici il segnale elettrico viene trasformato in neurotrasmettitore (sostanza chimica) e passa in questo modo ai dendriti dell'altro neurone dove viene riconvertito in segnale elettrico. Tutti gli impulsi arrivati nel soma vengono accumulati e se la differenza di potenziale tra interno ed esterno della membrana cellulare è superiore ad una certa soglia, allora solo in questo caso il neurone genera un segnale di output.
Nel cervello umano sono presenti 100 miliardi di neuroni interconnessi tra loro, questo tipo di struttura viene chiamata rete neurale multilivello ed è proprio la struttura che il machine learning si propone di ricreare in modo artificiale poichè grazie ad essa emergono i processi cognitivi, tra cui proprio l'apprendimento.

2.3.1 Modello di neurone artificiale

McCullock e Pitts descrissero il neurone come una porta logica che prende in input il segnale elettrico e lo trasforma in un output binario, dipendente dal superamento o meno di una certa soglia. Successivamente Rosenblatt introdusse una regola di apprendimento migliorando il percettrone di MCP, ma nonostante questo il percettrone di Rosenblatt rimane comunque un classificatore di tipo binario. Esso riceve in ingresso una combinazione lineare del vettore di input $\mathbf{x} = (x_1, ..., x_m)$ con il vettore dei pesi $\mathbf{w} = (w_1, ..., w_m)$, quindi l'input del percettrone sarà:

$$z = w_1 x_1 + ... + w_m x_m \tag{2.1}$$

La combinazione z diventa l'argomento di una funzione ϕ, detta funzione di decisione, il cui output verrà deciso in base al fatto che z sia maggiore o minore di una certa soglia chiamata θ. Per comodità si rinomina $-\theta = w_0$ (chiamato anche bias) e si passa θ a sinistra dell'uguale della combinazione 2.1, per cui essa diventerà:

$$z = w_0 x_0 + w_1 x_1 + ... + w_m x_m = \mathbf{w}^T \mathbf{x}$$

Quindi l'output sarà dato da:

$$\phi(z) = \begin{cases} 1 & \text{se } z \geq 0 \\ -1 & \text{se } z < 0 \end{cases}$$

I valori "1" e "-1", assunti dalla funzione $\phi(z)$, rappresentano le due classi distinte.
L'apprendimento consiste nel variare i pesi delle connessioni in modo da poter classificare in modo corretto i dati di input nelle due classi; l'algoritmo, introdotto da Rosenblatt, che permette di apprendere minimizzando l'errore di classificazione consta di due fasi: la prima riguarda l'inizializzazione dei pesi al valore 0 e la seconda riguarda il loro aggiornamento attraverso il calcolo dell'errore di classificazione. I pesi sono aggiornati nel seguente modo:

$$w_j = w_j + \Delta w_j$$

$$\Delta w_j = \eta(y^i - \tilde{y}^i)x_j^i \tag{2.2}$$

Nell'equazione 2.2 il valore η è chiamato learning rate e rappresenta la velocità con cui il percettrone sta imparando a classificare, mentre con $(y^i - \tilde{y}^i)$ è stata indicata la differenza tra l'output corretto e l'output trovato col classificatore. L'aggiornamento dei pesi è una procedura che viene ripetuta fino a

quando non si arriva ad un buon livello di classificazione da parte del percettrone. Bisogna, inoltre notare che questo classificatore arriverà a convergenza solo nei casi in cui le due classi sono linearmente separabili e il learning rate è basso. Nel caso di classi non linearmente separabili il percettrone aggiornerà i suoi pesi in continuazione, quindi per fermarlo si può decidere di eseguire questa procedura per un numero finito di volte oppure si fissa una soglia massima di classificazioni sbagliate. Per visualizzare il caso di classi linearmente separabili si può fare riferimento al grafico della figura 2.2, invece il caso contrario è mostrato nella figura 2.5 in cui non è possibile trovare una retta che divide perfettamente i dati nelle due classi opposte.

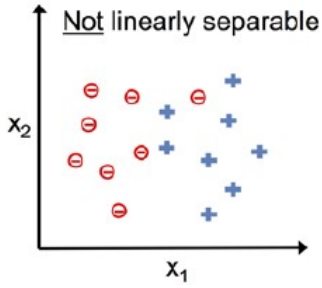


Figura 2.5: Nel grafico si può notare che i valori x_1 e x_2 non permettono una separazione lineare dei dati classificati.[4]

2.4 Reti neurali artificiali

Le reti neurali artificiali sono formate da neuroni artificiali, detti anche unità elaborative, interconnessi tra loro in analogia con la struttura del cervello umano. Un'ulteriore analogia col cervello umano riguarda il parallelismo con cui vengono eseguiti i vari processi.
Il metodo attraverso il quale la rete neurale apprende dai dati di input consiste nell'aggiornare i valori dei pesi delle interconnessioni tra le varie unità elaborative, questo procedimento avviene durante la fase di addestramento della rete. Successivamente la rete neurale sarà pronta per elaborare nuovi dati cercando di classificarli nel modo giusto.
È importante precisare che la presenza di più unità elaborative rende la rete neurale uno strumento idoneo per la risoluzione di problemi di classificazione multiclasse.
In generale, le componenti di una rete neurale sono:

- Insieme di unità elaborative.

- Funzione di attivazione. Essa è una funzione che prende tutti i valori provenienti dalle unità elaborative e restituisce altri valori che diventeranno a loro volta input delle unità elaborative successive collegate alle precedenti. Esistono diversi tipi di funzioni di attivazione.
- Connessioni fra le unità.
- Regola di propagazione dei valori di output. Essa fornisce il metodo per poter propagare gli output da un unità elaborativa alla successiva.
- Regola di apprendimento per modificare il peso delle connessioni.
- Funzione di output. Essa prende i valori in uscita dalle funzioni di attivazione e li trasforma nel segnale di output.

Le reti neurali vengono dette feedforward quando la loro architettura è formata da livelli di unità elaborative, inoltre una rete feedforward è di tipo fully-connected se ogni unità elaborativa di un livello è connessa con tutte le unità elaborative del livello successivo. Tra il primo livello di input di una rete e quello di output possono esserci più di un livello nascosto come mostrato in figura 2.6.

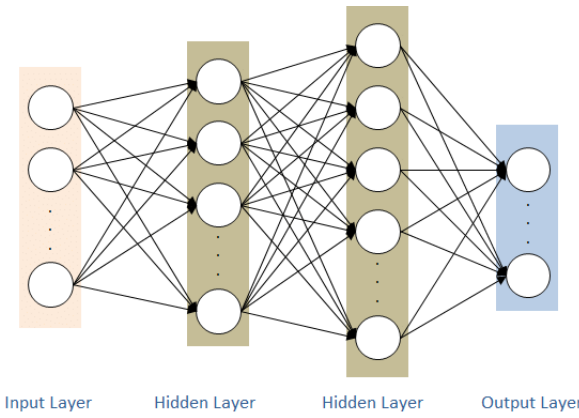


Figura 2.6: Esempio di una struttura di una rete neurale. [5]

Le due tipologie di reti neurali su cui ci si soffermerà nei prossimi paragrafi sono: la Multilayer Perceptron e la rete neurale convoluzionale.

2.4.1 Multilayer Perceptron

Una Multilayer Perceptron (MLP) è una rete neurale di tipo feedforward e fully-connected, nonostante nel suo nome sia presente la parola perceptron

le sue unità elaborative non sono dei percettroni poichè a differenza di questi ultimi l'informazione viene passata tra un livello e il successivo tramite le funzioni di attivazione delle singole unità elaborative.
L'architettura più semplice di una MLP è formata dal livello di input, un unico livello nascosto e un livello di output contenente tante unità elaborative quante ne sono le classi da assegnare ad ogni istanza di un dataset; essa è la struttura di riferimento per tutte le considerazioni che verranno fatte in questo sottoparagrafo.
Il primo passo che la MLP compie è la propagazione feedforward del pattern dei dati di training al fine di generare un output. La figura 2.7 è riportata in modo tale da rendere visibile sia la sua strutttura che la notazione che verrà chiarita in seguito.

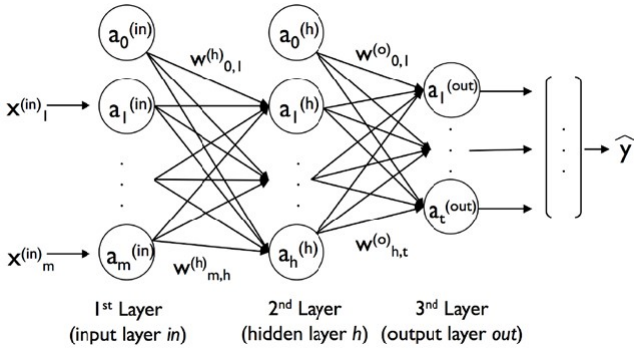


Figura 2.7: In figura è mostrata la struttura più semplice che può avere una rete neurale. [4]

Nella MLP in figura 2.7 si devono calcolare le unità di attivazione del livello nascosto e di quelle del livello di output considerando ogni singola unità elaborativa del livello precedente. Esse sono indicate con $a_k^{(h)}$ per il livello nascosto e con $a_k^{(out)}$ per quello di output, con il pedice k è stata indicata una qualsiasi unità elaborativa di quel livello. Partendo dal livello nascosto, per una singola unità elaborativa si ha:

$$z_1^{(h)} = a_0^{(in)} w_{0,1}^{(h)} + a_1^{(in)} w_{1,1}^{(h)} + ... + a_m^{(in)} w_{m,1}^{(h)} \tag{2.3}$$

Nella combinazione lineare 2.3 vanno chiarite le notazioni utilizzate: con w è indicato il peso della connessione, i suoi pedici corrispondono rispettivamente ad un'unità elaborativa del livello $(L-1)$ e ad una del livello L mentre l'apice corrisponde a livello L; $a^{(in)}$ denota l'attivazione nel caso del livello di input che coincide proprio col vettore di caratteristiche includendo anche il bias $(x_0^{(in)} = 1, x_1^{(in)}, ..., x_m^{(in)})$. Riprendendo la 2.3, $a_1^{(h)}$ è:

$$a_1^{(h)} = \phi(z_1^{(h)})$$

Il tipo di funzione ϕ (funzione di attivazione) è scelto in base al tipo di problema da risolvere.
Con la seguente notazione più compatta si indica l'attivazione calcolata per tutte le unita elaborative del livello nascosto e per tutte le istanze del dataset, quindi si avrà:

$$Z^{(h)} = A^{(in)}W^{(h)} \tag{2.4}$$
$$A^{(h)} = \phi(Z^{(h)})$$

Nella relazione 2.4 i termini $A^{(in)}$ e $W^{(h)}$ sono matrici di dimensioni nxm (n è riferito al numero di istanze presenti nel dataset e m alle unità di input incluso il bias) e mxd (d è riferito al numero di unità nascoste), moltiplicando queste due matrici risulterà che la matrice $Z^{(h)}$, dunque anche $A^{(h)}$, avrà dimensioni nxd.
Per calcolare l'attivazione nel livello di output basterà ripetere il ragionamento precedente, per cui i risultati indicati in notazione compatta sono:

$$Z^{(out)} = A^{(h)}W^{(out)}$$
$$A^{(out)} = \phi(Z^{(out)})$$

In questo caso moltiplicando la matrice W^{out} di dimensioni dxt (t è il numero di unità di output) per la matrice $A^{(h)}$ di dimensioni nxd si otterrà la matrice $Z^{(out)}$ di dimensioni nxt. Le dimensioni di $Z^{(out)}$ sono in accordo con la rappresentazione onehot degli output, nella quale ad ogni classe è associato un vettore, nel nostro caso di dimensione t, composto da tutti zero e un solo uno ogni volta in una posizione differente.
Il secondo step effettuato dalla MLP è la minimizzazione dell'errore di classificazione utilizzando l'algoritmo di backpropagation. Quest'algortimo impiega una funzione di costo per calcolare l'errore di classificazione, quest'ultimo sarà propagato in ogni livello partendo da destra verso sinistra (verso opposto rispetto alla propagazione feedforward) con lo scopo di aggiornare i pesi e i bias in modo tale da minimizzare proprio la funzione di costo. La forma della funzione di costo è:

$$C_0 = \sum_{j=0}^{n_L-1} (a_j^L - y_j)^2 \tag{2.5}$$

Nella 2.5 l'indice j della sommatoria scorre sulle unità elaborative del livello che viene indicato con la lettera L, inoltre il pedice 0 indica il fatto che questa funzione deve essere calcolata per ogni livello. Dato che la funzione di costo dipende dai pesi e dai bias, essa sarà influenzata dal loro cambiamento, infatti:

$$\frac{\partial C}{\partial w_{j,k}^L} = a_k^{L-1}\phi'(z_j^L)\frac{\partial C}{\partial a_j^L} \tag{2.6}$$

$$\frac{\partial C}{\partial b_j^L} = \phi^{'}(z_j^L)\frac{\partial C}{\partial a_j^L}$$

Nel caso della 2.6 gli indici j e k sono riferiti all'unità del livello L connessa a quella del livello $L-1$. L'ultimo termine della 2.6 può essere scritto come:

$$\frac{\partial C}{\partial a_j^L} = \sum_{j=0}^{n_{L+1}-1} w_{j,k}^{L+1}\phi^{'}(z_j^{L+1})\frac{\partial C}{\partial a_j^{L+1}}$$

Tutte le equazioni appena ricavate sono ottenute applicando la regola di derivazione delle funzioni composte, inoltre in quest'ultima relazione si può notare che nei termini a destra dell'uguale sono presenti elementi appartenti al livello successivo rispetto a quello dell'elemento a sinistra dell'uguale. Quanto appena detto mette in evidenza la propagazione all'indietro dell'errore e il conseguente cambiamento dei pesi e bias. Dopo aver calcolato tutte le derivate parziali del costo si otterrà il suo gradiente:

$$\nabla C = (\frac{\partial C}{\partial w^1}, \frac{\partial C}{\partial b^1}, ..., \frac{\partial C}{\partial w^L}, \frac{\partial C}{\partial b^L})$$

Infine, l'aggiornamento dei pesi sarà dato da:

$$W = W - \eta\nabla C$$

La propagazione feedforward e la backpropagation vengono effettuate dalla rete neurale sui dati di addestramento per un numero di epoche stabilito dal programmatore; dopo la fine dell'addestramento vengono generati gli output grazie ai quali si potranno comprendere le performance della MLP.

2.4.2 Rete neurale convoluzionale

La rete neurale convoluzionale, in breve CNN, è ispirata al funzionamento della corteccia visiva cerebrale.
In presenza di dataset composti da immagini questa rete costituisce il mezzo più efficace per la classificazione, in quanto è capace di estrarre le caratteristiche salienti che garantiscono la buona performance dell'algoritmo.
La struttura di una CNN contiene dei livelli convoluzionali e infine dei livelli fully connected.
Dall'immagine di input la CNN è in grado di calcolare le feature maps, ovvero delle caratteristiche fondamentali ricavate da delle parti di pixel dell'immagine stessa chiamate campi ricettivi locali.
Nella CNN, a differenza della MLP, ogni feature map è associata ad un unico pezzo di pixel e poi i pesi sono condivisi poichè sono usati per diversi parti

di pixel. Di conseguenza, sono ridotti sia il numero di interconnessioni che il numero di pesi ciò comporta una minore complessità di calcolo e il vantaggio di comprendere le caratteristiche chiave per un funzionamento migliore dell'algoritmo.
Dal punto di vista matematico la CNN esegue un'operazione chiamata convoluzione. Date due matrici X e W di dimensioni rispettivamente $n_1 x n_2$ e $m_1 x m_2$ con $n_1 \geq m_1$ e $n_2 \geq m_2$, la convoluzione è definita come segue:

$$Y = X * W \longrightarrow Y[i,j] = \sum_{k_1=0}^{k_1=m_1-1} \sum_{k_2=0}^{k_2=m_2-1} X[i-k_1, j-k_2] W[k_1, k_2] \quad (2.7)$$

Le dimensioni della matrice di output sono date da:

$$o_{1(2)} = [\frac{n_{1(2)} + 2p_{1(2)} - m_{1,(2)}}{s_{1(2)}}] + 1$$

Esse dipendono da due parametri applicati sulle due dimensioni: il padding e lo stride. Il padding consiste nell'aggiunta controllata di zeri lungo le due dimensioni della matrice X, ne esistono di tre tipi diversi:

- Full padding. In questo caso i parametri di padding p_1 e p_2 sono posti a $m_1 - 1$ e a $m_2 - 1$, quindi le dimensioni dell'output Y saranno più grandi di quelle di input X.
- Same padding. Viene utilizzato quando l'output deve avere le stesse dimensioni di input, dunque i parametri p_1 e p_2 sono decisi in base alle dimensioni della matrice W chiamata anche filtro.
- Valid padding. In questo caso si ha: $p_1 = 0$ e $p_2 = 0$.

Ritornando alla convoluzione, oltre la definizione 2.7 si può andare più nel dettaglio per una sua migliore comprensione attraverso un esempio fornito dalla figura 2.8.

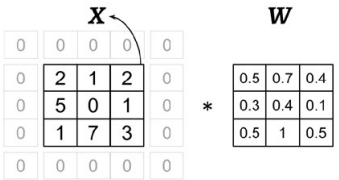


Figura 2.8: Esempio numerico di convoluzione.[4]

Per eseguire la convoluzione della figura 2.8 bisogna prima ruotare il filtro W, nel caso dell'esempio diventerà: $W^r = \begin{bmatrix} 0.5 & 1 & 0.5 \\ 0.1 & 0.4 & 0.3 \\ 0.4 & 0.7 & 0.5 \end{bmatrix}$

Adesso devono essere eseguiti i prodotti di Hadamard fra la matrice X e la matrice W facendo scorrere quest'ultima sulla prima, su entrambi le dimensioni, di un numero di passi indicati dal parametro stride. Successivamente la somma degli elementi delle matrici ottenute saranno gli elementi della matrice risultante dalla convoluzione, la figura 2.9 è posizionata a scopo esplicativo.

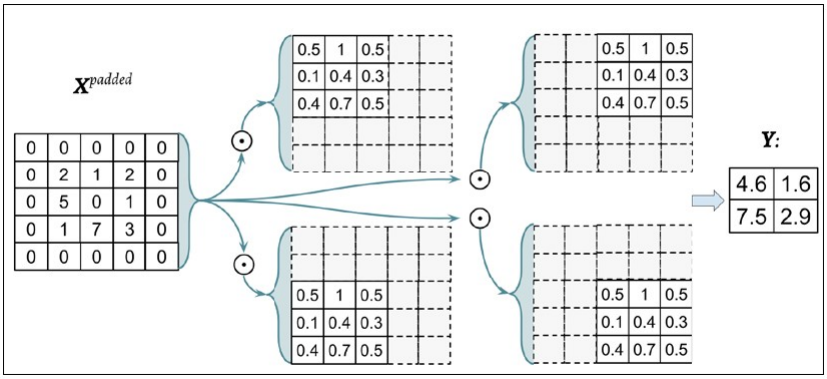


Figura 2.9: In questo caso lo stride è impostato a $(2, 2)$ mentre il padding a $(1, 1)$, vengono ricavate quattro matrici che portano alla matrice $Y_{2,2}$.[4]

Dopo che la rete neurale ha calcolato Y per ogni instanza del dataset di immagini rappresentate da X_{n_1,n_2}, il livello convoluzionale avrà una preattivazione pari a $A = Y + b$ (b indica il bias) e la feature map verrà estratta in questo modo:

$$H = \phi(A)$$

Questo tipo di rete neurale artificiale sarà lo strumento utilizzato per i problemi di classificazione affrotati nello studio effettuato in questo lavoro di tesi.

Capitolo 3

Esperimenti e risultati

In quest'ultimo capitolo si conciliano i due argomenti trattati nei capitoli precedenti usando due CNN in grado di analizzare le impronte di scarpe che fanno parte dei possibili elementi presenti in una scena del crimine.

3.1 Dataset

Il dataset[1] utilizzato è stato realizzato da dei ricercatori del Center for Statistic and Applications in Forensic Evidence (CSAFE) all'univesità dell'Iowa [3]. Per la costruzione del dataset 150 persone hanno dovuto scannerizzare le proprie impronte cinque volte sia per la scarpa destra che per la sinistra, dunque in totale sono presenti 1500 immagini. Lo scanner utilizzato rileva la distribuzione di peso di chi lo calpesta, inoltre per poter rilevare i piccoli dettagli dovuti all'usura ogni persona ha dovuto spostare il suo peso dalla punta al tacco per ognuna delle due scarpe. Le immagini fornite offrono un'ampia gamma di design delle suole utilizzando sia scarpe dello stesso brand ma di modelli diversi, sia scarpe di differenti brand.
Altre informazioni riportate sono: il sesso di chi indossa le scarpe, la misura e il modello di scarpe, il brand ed un numero identificativo della persona che le indossa, poichè può capitare che due persone abbiano lo stesso paio di scarpe.

3.1.1 Preprocessing

Per iniziare il preprocessing le immagini del dateset sono state importate nel codice, esse avevano dimensioni diverse tra loro così sono state riscalate tutte

[1]Link al dataset:(`https://data.csafe.iastate.edu/2DFootwearOutsoleStudy/`)

alle stesse dimensioni e i valori dei pixel sono stati anch'essi tutti riscalati in modo tale da essere compresi tra 0 e 1.
L'operazione di preprocessing svolta successivamente ha riguardato l'estrazione delle etichette relative alle immagini, in particolare sono stati creati due vettori di etichette: uno per la distinzione tra uomo e donna e l'altro contenente le misure di scarpe corrispondenti a quelle delle immagini.
Viene, poi, eseguito uno shuffle casuale dei dati del dataset ottenuto in modo tale da evitare che le immagini relative allo stesso paio di scarpe siano sottoposte consecutivamente all'algortimo che si occupa della classificazione, di cui si parlerà nelle pagine successive.
Il dataset ottenuto è costituito da 1500 immagini riscalate, di cui 800 sono state usate per formare il dataset di training mentre per il validation set e il test set ne sono state usate rispettivamente 400 e 300. L'ultima operazione di preprocessing è la normalizzazione di tutti e tre i tipi di dataset, ottenuta calcolando il valor medio su tutte le caratteristiche del dataset di training e sottraendolo ad ogni dataset e successivamente dividendo ogni volta il risultato ottenuto per la loro deviazione standard.
La figura 3.1 riporta alcuni esempi di immagini di scarpe presenti nel dataset.

(a) (b) (c) (d)

Figura 3.1: Quattro esempi di impronte di scarpe presenti nel dataset.

3.2 Strumenti utilizzati

Per l'implementazione dell'algoritmo di machine learning è stato scelto Python come linguaggio di programmazione, poichè è dotato di molte librerie in cui sono presenti funzioni molto utili per il campo dell'intelligenza computazionale.
La parte di preprocessing è stata svolta utilizzando moduli molto famosi, tra cui: NumPy, Matplotlib, Os, Cv2 e Pandas. Il modulo impiegato per la parte dell'algoritmo che si occupa di classificare le istanze del dataset è Tensorflow.

Nella maggior parte dei casi i dati con cui si lavora sono dei tensori, essi sono enti matematici di più dimensioni indicate dalla parola rango, per esempio: un tensore di rango 0 è proprio uno scalare. Tensorflow dà la possibilità agli utenti di eseguire operazioni complesse fra tensori grazie ai suoi metodi già implementati.
Per eseguire delle operazioni bisogna istanziare un grafico computazionale vuoto e riempirlo con i nodi, i quali rappresentano le operazioni da svolgere, dopodichè bisogna eseguire il grafico creato.
In un grafico computazionale bisogna definire le variabili che rappresentano i tensori utilizzati nelle operazioni ed è importante però notare che non hanno nessun valore, fino a quando non sono inizializzate durante la fase di esecuzione del grafico.
Un altro strumento utile di tensorflow sono i placeholder, essi sono dei tensori con dimensioni specifiche che non contenogno dati e sono usati proprio per avere strutture a disposizone nel codice da "riempire" ogni volta con dati diversi durante l'esecuzione del grafico.
Nella figura 3.2 è mostrato uno schema di un grafico computazionale.

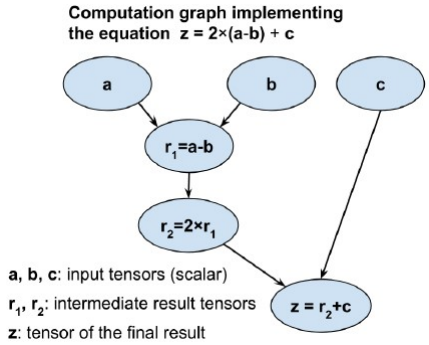


Figura 3.2: Un grafico computazionale può essere rappresentato in questo modo collegando tutti i nodi fra loro.[4]

Di seguito, è riportato un esempio di codice [4] che rappresenta l'implementazione del grafico della figura 3.2 utilizzando anche dei placeholder.

```
import tensorflow as tf

g = tf.Graph()
with g.as_default():
    tf_a = tf.placeholder(tf.int32, shape[], name='tf_a')
    tf_b = tf.placeholder(tf.int32, shape[], name='tf_b')
    tf_c = tf.placeholder(tf.int32, shape[], name='tf_c')
```

```
r1 = t_a - tf_b
r2 = 2*r1
z = r2 + tf_c
```

In quest'ultima parte di esempio si vede come lanciare il grafico fornendo i valori ai placeholder [4]:

```
with tf.Session(graph=g) as sess:
     feed = {tf_a: 1, tf_b: 2, tf_c: 3}

     print('z:', sess.run(z, feed_dict=feed))

output -> z: 1
```

In conclusione, gli esempi di codice sono stati riportati nel paragrafo in modo tale da rendere chiaro quali metodi e quali oggetti principali di Tensorflow sono stati utilizzati per l'algoritmo di machine learning di questo lavoro di tesi.

3.3 Rete neurale convoluzionale per la classificazione

La scelta dell'algoritmo di machine learning sia per la classificazione binaria che per quella multiclasse è ricaduta su una rete neurale convoluzionale. Per quanto detto nel capitolo precedente, le CNN costituiscono un approccio utilizzato sempre più spesso in caso di dataset con immagini rispetto a quello fornito dalle MLP. Il codice per implementare le CNN è stato leggermente modificato nell'architettura delle reti ma, sostanzialmente, è stato preso dal seguente libro: Rashka R., Mirjalili V., *Python Machine Learning, 2nd edition*, Packt, UK, 2017, chapter 15. Il codice appena citato è stato usato per risolvere il problema della classificazione delle cifre scritte a mano, dunque le leggere modifiche menzionate in precedenza servono ad adattare le CNN ai problemi affrontati in questo lavoro di tesi .
Il primo caso che verrà analizzato sarà il problema binario, ovvero la classificazione tra uomo e donna delle impronte di scarpe mentre nel secondo caso si analizzerà il problema multiclasse, ovvero la CNN dovrà essere in grado di predirre la misura delle scarpe corrispondente ad una determinata impronta.

3.3.1 Classificazione binaria delle impronte di scarpe

In questo caso è stato effettuato uno studio preliminare scegliendo di riscalare le immagini del dataset ad un formato di $16x16$ pixel, la figura 3.3 mostra il risultato di questo riscalamento.

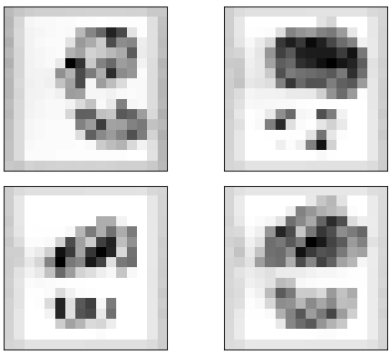
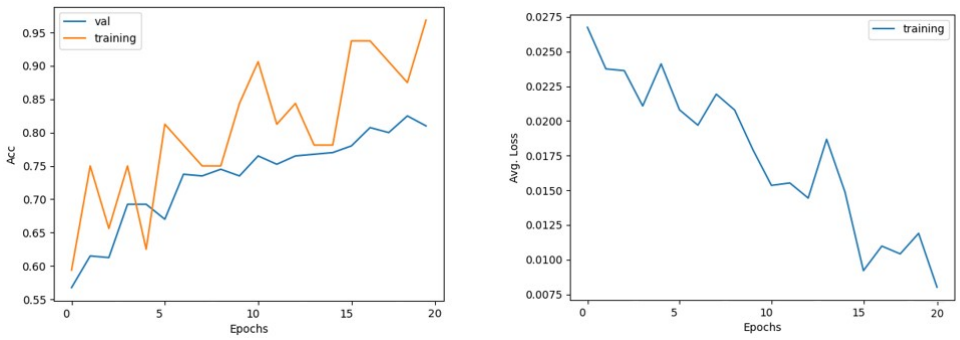


Figura 3.3: Effetto del riscalamento delle immagini del dataset riportate in figura 3.1

La CNN ha la seguente struttura: due livelli convoluzionali intervallati da due livelli di max-pooling più altri due livelli del tipo fully connected. Il funzionamento dei livelli convoluzionali e di quelli fully connected è stato già spiegato nel secondo capitolo, il livello di max pooling, invece, serve per ridurre le dimensioni del tensore con cui si sta lavorando. In input al primo livello convoluzionale viene fornito il tensore con le seguenti dimensioni: ($batchsize x16x16x1$), il parametro *batch-size* permette di eseguire la fase di addestramento della CNN su delle porzioni del dataset, in questo caso il suo valore è stato impostato a 32. La quarta dimensione del tensore indica il fatto che le immagini sono in bianco e nero e quindi c'è un solo canale.
Viene dunque effettuata la convoluzione tra il tensore di input e il filtro di dimensione ($1x5x5x1$) e si ottiene in output un tensore dalle seguenti dimensioni: ($batchsize x12x12x32$), in particolare l'ultima dimensione indica che sono state estratte 32 feature-map da ogni immagine.
Successivamente c'è il livello di max pooling che permette di ridurre le dimensioni del suo input, infatti riesce a creare un tensore di output che ha dimensioni: ($batchsize x6x6x32$). I parametri che servono al max pooling riguardano le dimensioni delle sottomatrici da cui devono essere estratti solo gli elementi più grandi numericamente, in questo caso le dimensioni scelte sono: ($1x2x2x1$). Si deve, inoltre, specificare che sia le dimensioni del filtro sia quelle del max pooling sono sempre uguali, anche il padding e lo stride non cambiano: in particolare il primo è nella modalità Valid oppure Same, a

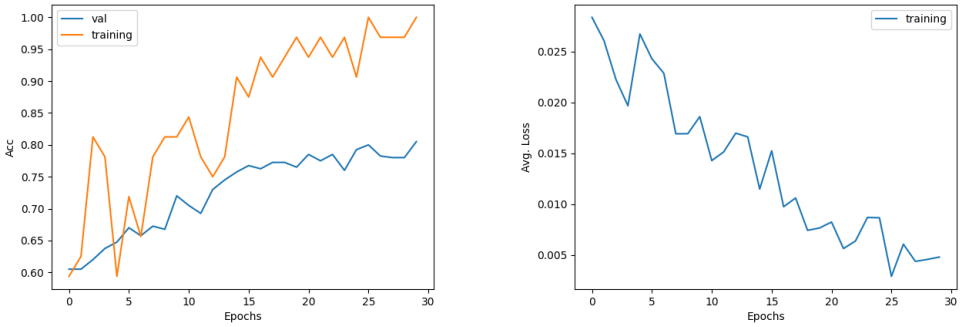
seconda che il livello sia convoluzionale o di max pooling, mentre il secondo ha dimensioni $(1x2x2x1)$.
La presenza di un secondo livello convoluzionale permette l'estrazione di 64 feature-map con il conseguente tensore $(batchsizex2x2x64)$ di output, il quale verrà sottoposto ad un altro livello di max pooling che ridurrà le sue dimensioni a $(batchsizex1x1x64)$.
Infine sono presenti due livelli di tipo fully connected, nel primo scegliamo di collegare le 64 feature-map con 1024 unità elaborative mentre nel secondo devono perforza essere presenti soltanto due unità elaborative, in quanto la classificazione deve avvenire tra la classe uomo o la classe donna.
Alla fine di ogni livello della rete ogni tensore generato in output prima di andare in input al livello successivo viene, in realtà, passato ad una funzione di attivazione.
Tutte le operazioni menzionate vengono eseguite durante la fase di addestramento per tutti i *batch* di dati. Inoltre, viene calcolata l'accuratezza sia sul training dataset che sul validation dataset.
Adesso si possono valutare le performance della CNN andando a vedere i grafici dell'accuratezza e del costo in funzione delle epoche di addestramento, essi sono mostrati nella figura 3.4.



(a) (b)

Figura 3.4: Nel grafico (a) l'andamento dell'accuratezza sia del training set che del validation set ha un andamento che cresce con le epoche, nonostante ci siano delle fluttazioni. Il grafico (b) della funzione di costo decresce con le epoche anche se fluttua.

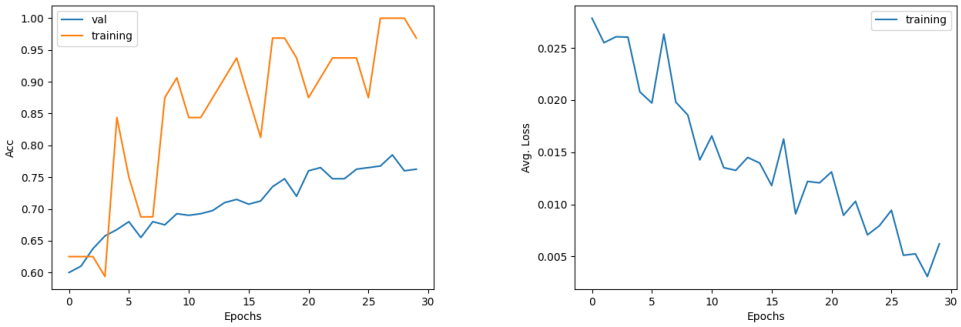
Il valore massimo dell'accuratezza del training set è del 97% mentre quella del validation set è dell'82%, dunque la rete costruita non riesce a generalizzare nel modo giusto infatti l'accuratezza sul test set è dell'81%. Un altro tentativo è stato eseguito alzando le epoche di addestramento, passandole a 30 (figura 3.5).



(a) (b)

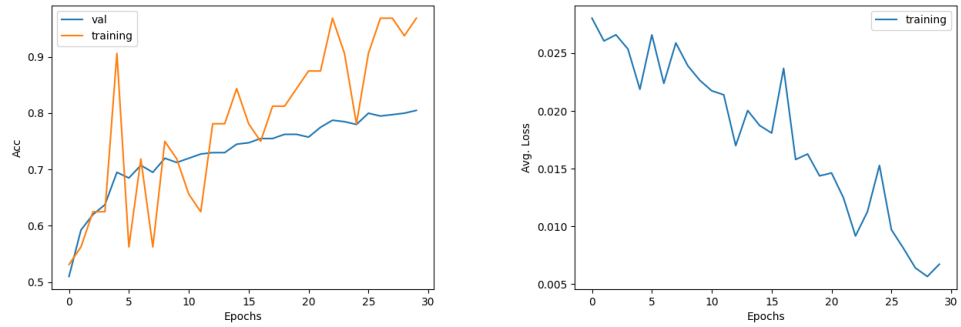
Figura 3.5: Il comportamento dei grafici (a) e (b) è simile a quello dei grafici della figura 3.4

In questo caso si nota che l'accuratezza del training set riesce a raggiungere il 100% ma quella del validation set arriva al massimo all'80%, mentre sul test set il risultato è pari a 82.667% dunque leggermente migliorata rispetto al caso precedente. Un metodo adottato per tentare di regolarizzare la CNN con lo scopo di migliorare le sue performance è il *dropout*. Esso consiste nello spegnimento casuale di una parte delle unità elaborative soltanto nella fase di addestramento, secondo un coefficiente di probabilità che deve essere fornito dal programmatore; inoltre fornisce la possibilità di addestrare più modelli contemporaneamente cercando di aiutare la rete a generalizzare. I valori di *dropout* scelti sono: 0.5 e 0.7, nelle figure 3.6 e 3.7 sono riportati i grafici relativi a questi tentativi.
Si può notare che anche nel caso della figura 3.6a viene raggiunto da parte del training set il 100% di accuratezza invece sul validation set si è riusciti ad arrivare ad un massimo di 78.5%, quindi c'è ancora una forte discrepanza tra le performance sui due dataset e infatti ciò lo si può anche evincere guardando semplicemente il grafico 3.6a; le performance sul test set sono pari all'81%.
Nella figura 3.7a si può notare, a differenza della figura 3.6a, che l'accuratezza sul training set non arriva al 100% ma ad un valore vicino. L'accuratezza sul validation set cresce fino al 79% e infine sul test set è pari al 79.667%, quindi non c'è stato nessun miglioramento sulle performance della CNN sia in presenza di dropout pari a 0.5 che in assenza di dropout.



(a) (b)

Figura 3.6: Il grafico (a) e il grafico (b) sono il risultato dell'addestramento della CNN con un valore di dropout pari a 0.5, ciò vuol dire che sono state spente in modo casuale metà delle unità elaborative.

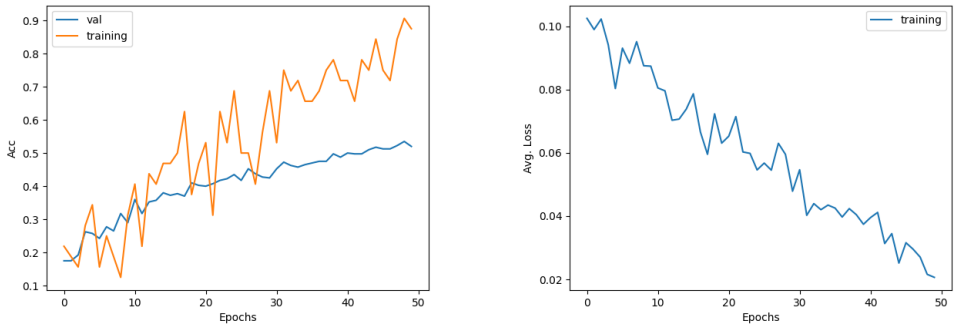


(a) (b)

Figura 3.7: A differenza della figura 3.6, questi grafici sono ottenuti ponendo il valore di dropout a 0.7. Si può notare che aumentando sempre di più il dropout sia il grafico (a) che quello (b) sono soggetti a fluttuazioni più grandi.

3.3.2 Classificazione multiclasse delle impronte di scarpe

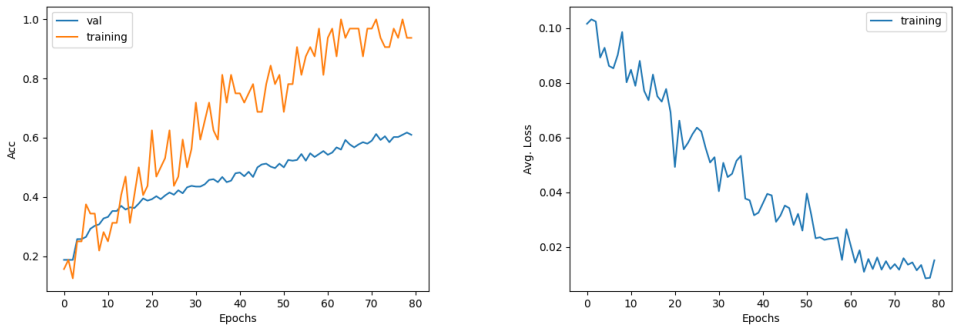
Questo caso riguarda la classificazione delle impronte di scarpe secondo la loro misura, in particolare nel dataset c'erano 15 misure di scarpe differenti dunque sono presenti 15 classi distinte.
La CNN usata per questo problema è la stessa che è stata implementata anche nel caso di classificazione binaria, l'unica differenza sta nel fatto che l'ultimo livello fully connected ha 15 unità elaborative in quanto gli output che la rete deve predirre sono appunto 15.
Inizialmente la rete è stata addestrata per 50 epoche e senza usare dropout, la figura 3.8 mostra i grafici relativi alle accuratezze su training e validation dataset e alla funzione di costo in funzione delle epoche.



(a) (b)

Figura 3.8: Nel grafico (a) sono presenti sia l'andamento dell'accuratezza sul training set sia quello dell'accuratezza sul validation, si nota che nel primo ci sono molte fluttuazioni grandi mentre nel secondo continuano ad esserne tante ma sono molto meno grandi. Il grafico (b), invece, mostra il comportamento decrescente della funzione di costo.

Per quanto riguarda le performance si può evincere che non sono buone, infatti mentre l'accuratezza sul training set riesce a salire a più del 90% quella sul validation arriva ad un massimo di 53.5% e sul test è pari al 51.667%. Prima di provare ad usare il dropout si è provato semplicemente ad aumentare il numero di epoche ad 80, questo ha portato ad un leggero miglioramento rispetto al caso con 50 epoche in quanto l'accuratezza sul training set è arrivata al 100%, sul validation set è arrivata al 61.8% e infine sul test set è pari al 60.333%. In figura 3.9 sono mostrati i grafici relativi a questo tentativo che però non ha comunque dato delle buone performance.



(a) (b)

Figura 3.9: I seguenti grafici sono relativi all'accuratezza e alla funzione di costo in funzione delle 80 epoche.

Successivamente sono stati eseguiti dei tentativi con i valori 0.5 e 0.7 di dropout, nella figura 3.10 sono riportati i grafici ottenuti. Sia nel grafico 3.10a che in quello 3.10c l'accuratezza sul training set tocca il 90% e nel primo supera anche questo valore. Sul validation set l'accuratezza arriva al 61% per il dropout pari a 0.5, invece in caso di dropout pari a 0.7 arriva al 55.8%; sul test set l'accuratezza nei due casi è pari al 60% e al 56%.

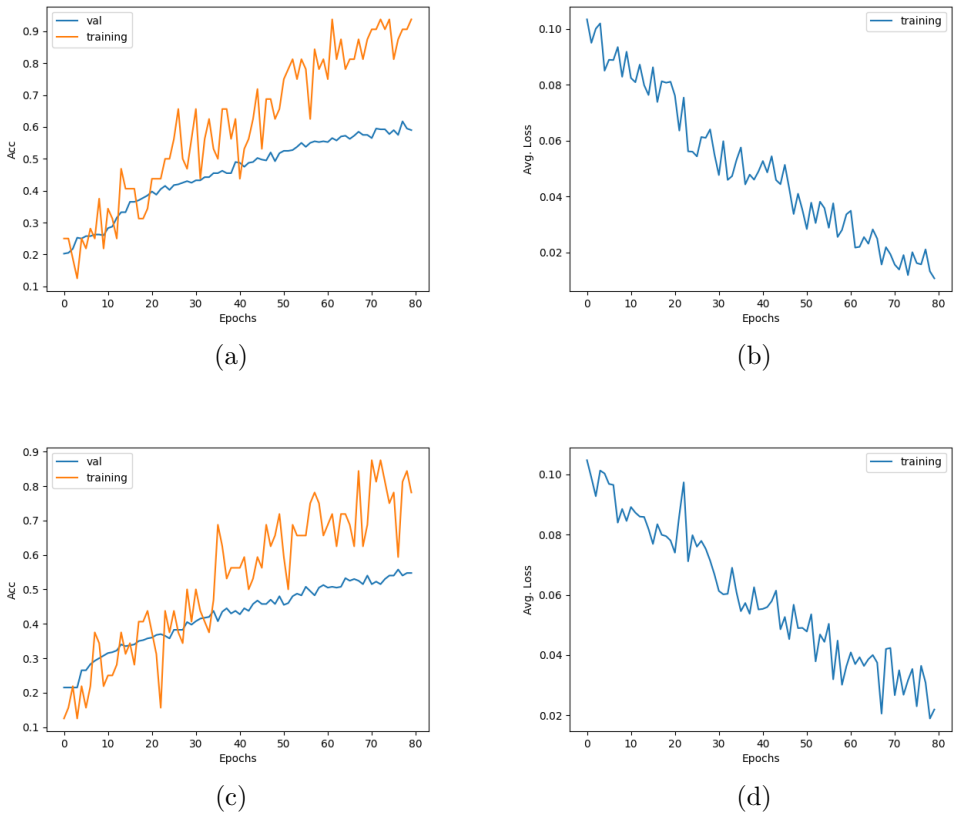


Figura 3.10: I grafici (a) e (b) sono relativi al caso di dropout pari a 0.5, mentre quelli (c) e (d) sono stati ottenuti impostando il dropout a 0.7.

In conclusione, la rete neurale implementata non riesce a classificare bene le istanze del dataset di test sia nel caso multiclasse che in quello binario del sottoparagrafo precedente. La causa di questo malfunzionamento riguarda sia il riscalamento eccessivo delle immagini del dataset, sia l'estrazione di troppe poche feature-map. Nel sottoparagrafo successivo vedremo come le performance della rete migliorano modificando le cause del malfunzionamento.

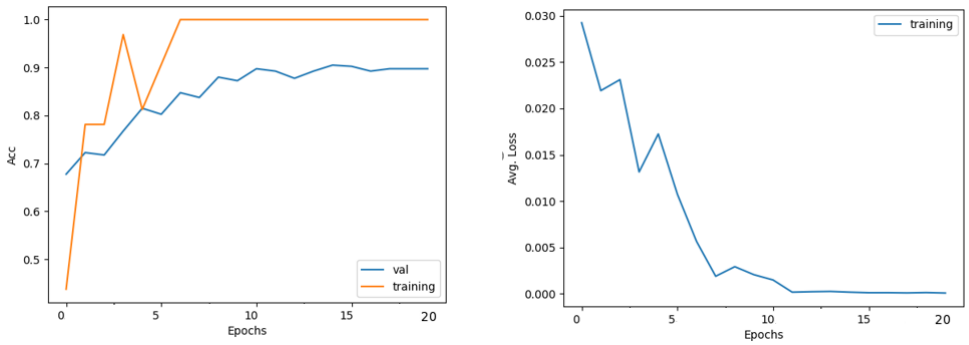
3.3.3 Classificazione binaria attraverso una CNN modificata

La CNN implementata in precedenza è stata leggermente cambiata in modo tale che essa riesca ad avere delle performance migliori rispetto alle precedenti.

Il numero dei livelli convoluzionali passa a tre in modo tale da poter estrarre 128 feature map, essi sono intervallati da due livelli di max pooling con stride impostato a $(1x4x4x1)$. I due livelli finali fully connected rimangono invariati, così come tutto ciò che non è stato menzionato.
L'altro cambiamento riguarda il riscalamento delle immagini a $256x256$ pixel. Le dimensioni del tensore attraverso i vari livelli sono cambiate nel seguente modo:

- In input il tensore avrà dimensioni pari a $(batchsizex256x256x1)$.
- Dopo la convoluzione avrà dimensioni: $(batchsizex252x252x32)$.
- Applicando il max pooling le dimensioni sono $(batchsizex63x63x32)$.
- Dalla seconda convoluzione le dimensioni sono $(batchsizex59x59x64)$.
- Dopo il secondo max pooling le dimensioni diventano $(batchsizex15x15x64)$.
- La terza convoluzione restituisce un tensore di dimensioni $(batchsizex11x11x128)$, successivamente ci sono i due livelli fully connected che hanno rispettivamente 3200 e 2 unità elaborative.

Adesso vengono mostrati i risultati ottenuti senza applicare il dropout; nelle figure 3.11(a) e 3.11(b) sono riportati i grafici di accuratezza e del costo in funzione delle 20 epoche di addestramento.



(a) (b)

Figura 3.11: Nei grafici (a) e (b) sono mostrati gli andamenti delle accuratezze e delle funzioni di costo.

L'accuratezza sul validation set, in questo caso, arriva ad un valore massimo pari al 90.5%.

Per cercare di migliorare le performance della CNN è stato eseguito uno studio cambiando il dropout tra i seguenti valori: 0.2, 0.4, 0.5, 0.7, 0.8. Di seguito sono riportati tutti i grafici dell'accuratezze e del costo per tutti i valori di dropout.
Nei casi di dropout pari a 0.2 e 0.4 (figura 3.12), le accuratezze sul validation set corrispondono rispettivamente all' 89.7% e al 90.8%.

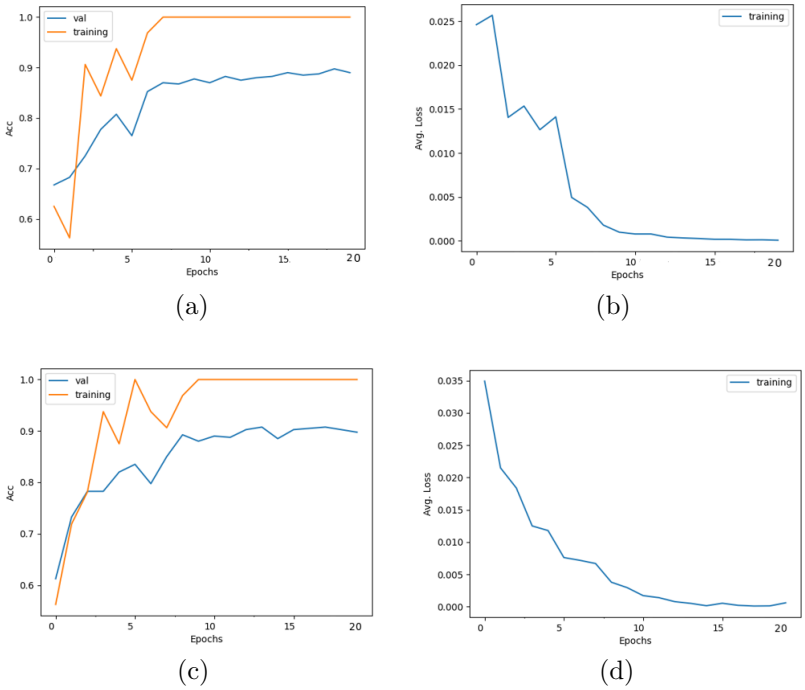
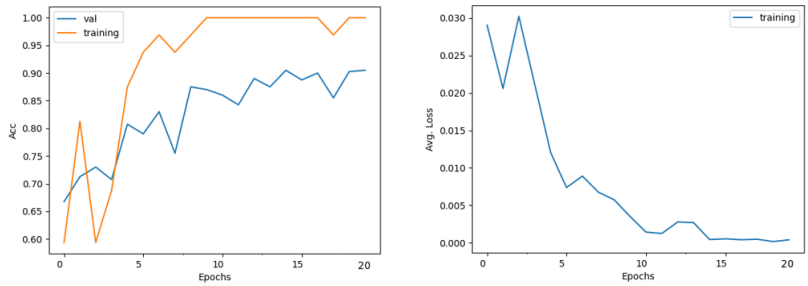
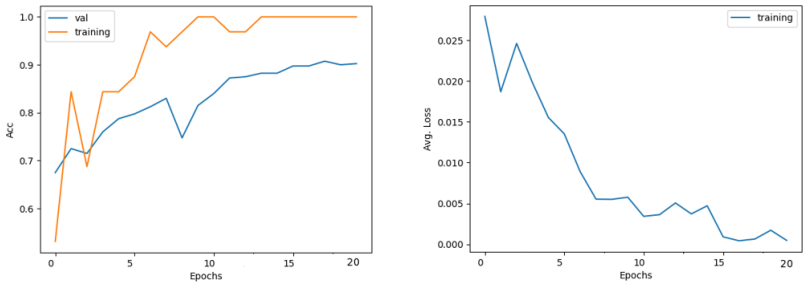


Figura 3.12: I seguenti grafici sono stati ottenuti ponendo il dropout a 0.2 e a 0.4.

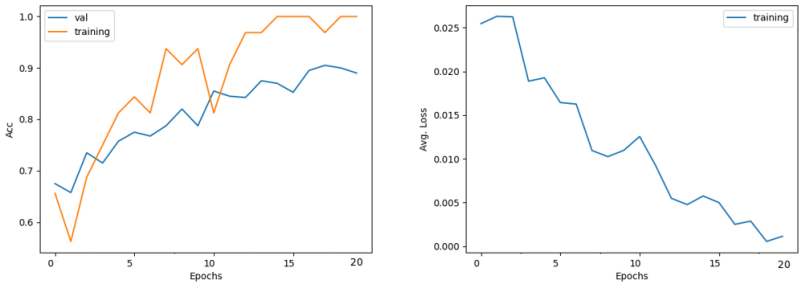


(a) (b)



(c) (d)

Figura 3.13: I seguenti grafici sono stati ottenuti ponendo il dropout a 0.5 e a 0.7



(a) (b)

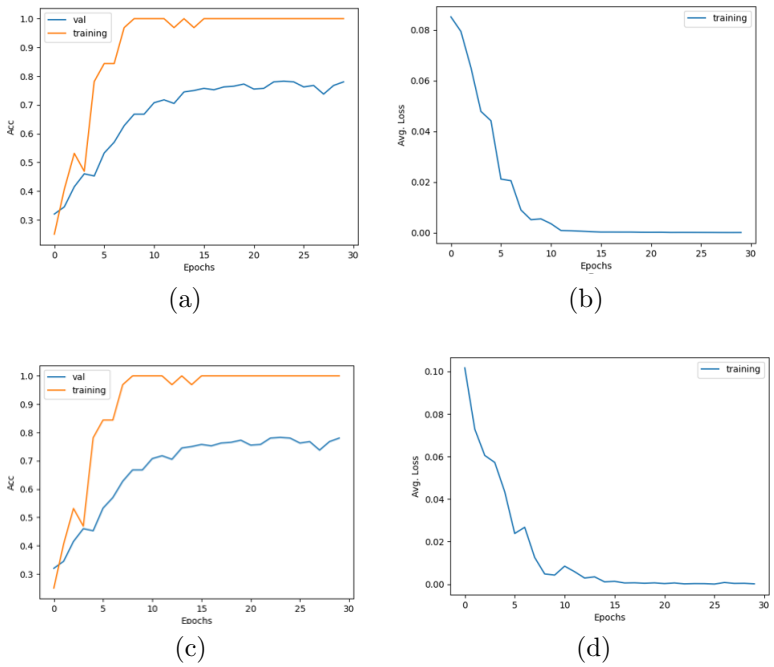
Figura 3.14: Questi ultimi due grafici sono relativi al dropout pari a 0.8

Nella figura 3.13, vi sono altri due casi dove il dropout è uguale a 0.5 e 0.7, a questi valori corrispondono le seguenti accuratezze sul validation set: 90.5% e 90.8%.

L'ultimo caso è relativo alla figura 3.14 per il quale l'accuratezza sul validation set è pari al 87%.
Nel caso dei parametri di dropout pari a 0.4 e 0.7 le due accuratezze sul validation set sono uguali in valore e sono le più alte rispetto a quelle ottenute tramite gli altri parametri; la decisione di quale parametro adottare è ricaduta su 0.7 dopo aver effettuato una valutazione sulle prestazioni della CNN con dropout impostato prima a 0.4 e poi a 0.7. La prestazione migliore sul validation set è stata ottenuta col parametro 0.7 (figura 3.13(c)). Dopo l'addestramento la rete è stata in grado di classificare i dati del test set con un accuratezza pari all'89.667%. Utilizzando questa nuova struttura della CNN e aumentando il numero di pixel delle immagini, la rete neurale ha migliorato le sue performance rispetto al caso precedente che aveva fonito, invece, un'accuratezza sul test set pari all'81%.

3.3.4 Classficazione multiclasse attraverso una CNN modificata

La struttura della CNN impiegata per classificare la misura delle scarpe è uguale a quella descritta nel sottoparagrafo precedente.
Anche in questo caso la rete è stata addestrata per 30 epoche senza usare il dropout. Successivamente sono stati eseguiti diversi tentativi usando i valori di dropout citati nel sottoparagrafo precedente. Sono riportati nelle seguenti figure i grafici di accuratezza e della funzione di costo per ogni tentativo.



(a) (b)

(c) (d)

Figura 3.15: I grafici (a) e (b) sono relativi al tentativo senza dropout, mentre in quelli (c) e (d) il dropout è impostato a 0.2

Nei casi di dropout pari a 0 e 0.2 (figura 3.15), l'accuratezze sul validation set corrispondono rispettivamente all' 73.3% e all' 78.3%. Nella figura 3.16, vi sono altri due casi dove il dropout è uguale a 0.4 e 0.5, a questi valori corrispondono le seguenti accuratezze sul test set: 78.5% e 71.5%.

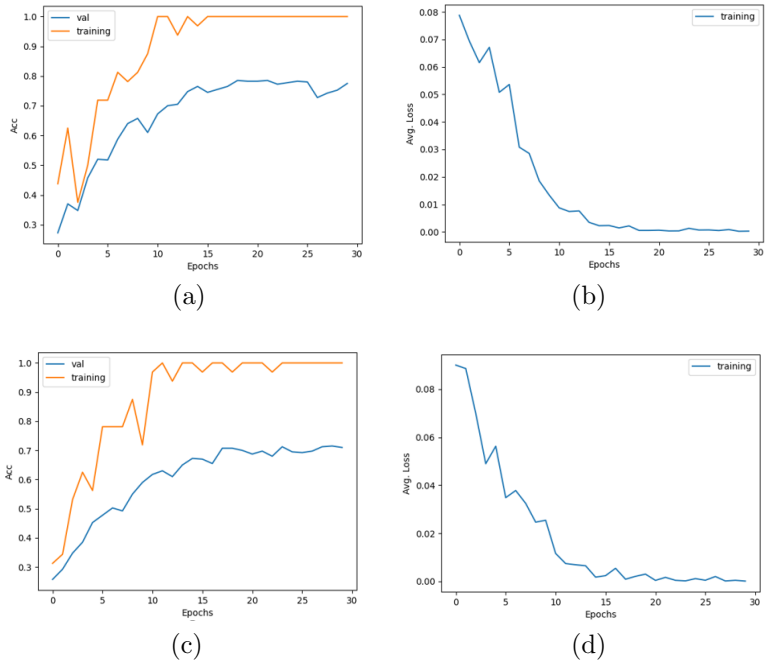


Figura 3.16: I grafici (a) e (b) sono relativi al tentativo con dropout a 0.4, mentre in quelli (c) e (d) il dropout è impostato a 0.5

Infine, nella figura 3.17 l'accuratezza sul test set risulta pari al 75.7% in caso di dropout uguale a 0.7, mentre è uguale al 76.7% per il valore 0.8 di dropout.

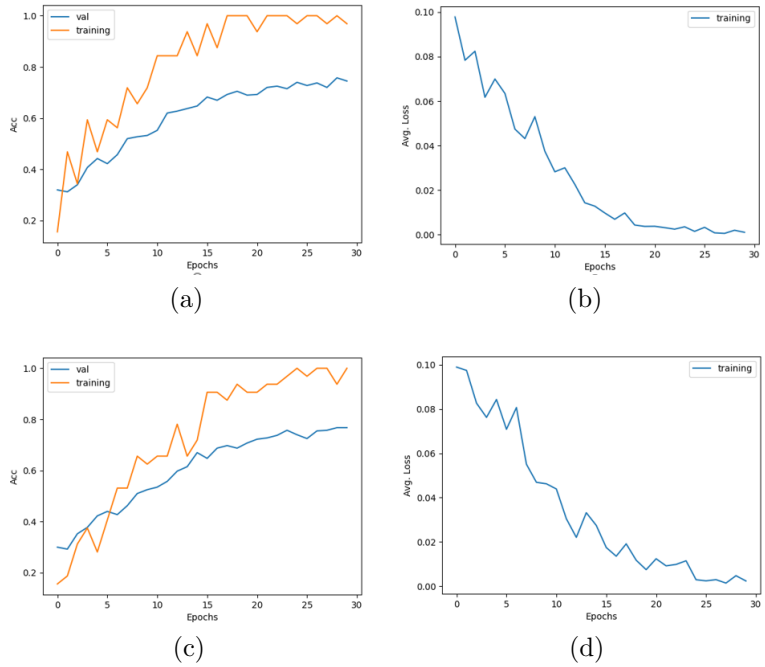


Figura 3.17: I grafici (a) e (b) sono relativi al tentativo con dropout a 0.7, mentre in quelli (c) e (d) il dropout è impostato a 0.8

La scelta del valore di dropout è ricaduta su 0.4, poiché l'accuratezza sul validation set è risultata la più alta rispetto a quella ottenuta utilizzando gli altri valori, inoltre anche in questo caso si può notare un miglioramento nella performance della CNN in quanto con questo valore di dropout l'accuratezza sul test set è pari al 78%, che è un valore più alto rispetto al 63.333% ottenuto nel caso di immagini riscalate a $16x16$ pixel ed estraendo solo 64 feature-map.

Conclusioni

In questo lavoro di tesi è stato affrontato uno studio atto a risolvere sia la classificazione del sesso del proprietario delle impronte di scarpe che la sua misura di scarpe. Inizialmente le strutture delle CNN scelte per la risoluzione dei due problemi, non hanno fornito dei risultati positivi neanche svolgendo una piccola regolazione dei suoi paramentri. Basti notare che in entrambi i tipi di classificazione, il tentativo con il valore dell'accuratezza sul validation set più alto ha portato ad ottenere delle accuratezze sul test set pari all' 81% in caso binario e pari all' 63% nel caso multiclasse.
Successivamente, attraverso un cambiamento della struttura dei dati e delle CNN sono state ottenute delle performance migliori, infatti i migliori tentativi hanno portato ad avere le seguenti accuratezze sul test set: 89.7% e 78%, in particolare il primo valore è riferito al caso binario mentre il secondo al caso multiclasse.
Nel futuro, questo studio potrebbe essere completato confrontando i risultati ottenuti in questa tesi con quelli ottenuti usando differenti approcci di machine learning come il Multilayer Perceptron, per ulteriomente investigare i vantaggi del modello proposto.
In definitiva, l'obiettivo pratico di questo studio è fornire uno strumento in grado di aiutare gli investigatori nell'analisi delle impronte di scarpe trovate su una scena del crimine. Pertanto, come sviluppo futuro, il modello di rete neurale convoluzionale implementato e valutato in questo lavoro di tesi, potrebbe essere incluso nell'architettura di un'applicazione per dispositivi mobili come uno smartphone per essere utilizzata direttamente sulla scena del crimine.

Bibliografia

[1] Max M Houck and Jay A Siegel. *Fundamentals of forensic science*. Academic Press, 2009.

[2] Andrew RW Jackson and Julie M Jackson. *Forensic science*. Pearson Education, 2008.

[3] Soyoung Park and Alicia Carriquiry. A database of two-dimensional images of footwear outsole impressions. *Data in brief*, 2020.

[4] Sebastian Raschka and Vahid Mirjalili. *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow*. Packt Publishing Ltd, 2017.

[5] Maria Scanlon. *Semantic Annotation of Aerial Images using Deep Learning, Transfer Learning, and Synthetic Training Data*. PhD thesis, 09 2018.