Università degli Studi di Salerno

Corso di Fondamenti di intelligenza artificiale

Bollicine

Relazione progetto

## Introduzione

La capacità di dare un nome, di riconoscere e di classificare oggetti in tipologie o categorie che si conformano a modelli teorici (pattern) è una delle caratteristiche principali dell'intelligenza umana.

La branca dell'intelligenza artificiale che si occupa dello studio e dello sviluppo di metodi e sistemi per il riconoscimento automatico di oggetti attraverso l'identificazione di regolarità significative in un sistema complesso e in presenza di rumore è nota come ''pattern recognition".

L'identificazione avviene attraverso un processo nel quale i dati vengono analizzati con l'obiettivo di individuare delle classi o categorie in maniera tale che il grado di similarità o di associazione sia alto fra gli elementi o individui della stessa categoria e basso fra gli elementi di categorie distinte.

L’occhio umano riconosce e interpreta un oggetto appartenente ad un’immagine in maniera immediata, con poco sforzo, nonostante l’immagine possa variare in angolazione, formato, scala e rotazione. Questo processo di identificazione da parte di un calcolatore è molto più elaborato ed è oggi una grande sfida in Computer Vision.

Per ogni oggetto in un’immagine, esistono caratteristiche interessanti (features) che lo contraddistinguono da altri oggetti. Queste possono essere estratte in modo da fornire una descrizione caratteristica e distintiva dell’oggetto.

Il Riconoscimento di immagini è generalmente suddivisa, dai ricercatori in ambito di Computer Vision, in due tipi: il riconoscimento di un oggetto specifico e il riconoscimento generico.

Nel caso di riconoscimento specifico, la scopo `e quello di identificare un istanza di un particolare oggetto, persona, luogo.

* 1. Obiettivo

Nel nostro caso l’obiettivo è quello di distinguere in due categorie, vino bianco e vino rosso, le immagini di bottiglie di vino tramite una rete neurale convoluzionale.

* 1. Implementazione

Dal punto di vista implementativo abbiamo utilizzato:

* Jupyter noetbook
* Pyton
* TesorFlow
* Numpy
* pyplot
  1. PEAS

Formulazione del problema in termini di **Performance**, **Environment**, **Actuators e Sensors**

|  |  |  |  |
| --- | --- | --- | --- |
| PERFORMANCE | ENVIRONMENT | ACTUATORS | SENSORS |
| % img correttamente classificate | Collezione di immagini | Etichettatore della bottiglia nell’immagine | Array di pixel |

## Organizzazione dei dati

Per il nostro progetto, per via della mancanza di un dataset esistente, abbiamo scaricato complessivamente 200 immagini di bottiglie di vino.

I dati sono stati organizzati in cartelle divise tra bianchi e rossi (classi utilizzate per la classificazione).

* 1. Dati di training

Abbiamo suddiviso la cartella di “train” in due sottocartelle: “bianco” e “rosso” contenenti rispettivamente 100 immagini di bottiglie di vini appartenenti alle categorie descritte dal nome della cartella.

* 1. Dati di validazione

I dati di validazione sono presenti nella cartella “validation”, la quale è strutturata in maniera analoga alla cartella di “train” ovvero, contiene due sottocartelle: “bianco” e “rosso” contenenti rispettivamente 20 immagini di bottiglie di vini appartenenti alle categorie descritte dal nome della cartella.

* 1. Dati di test

I dati utilizzati per testare l’effettivo funzionamento dell’algoritmo sono presenti nella cartella “test”.

Questi dati vengono reperiti su internet cercando di utilizzare immagini che non sono presenti nei dati di training.

## Approccio teorico

* 1. Introduzione CNN

Una **rete neurale convoluzionale** (CNN o ConvNet) è un’architettura di rete per il **deep learning** che apprende direttamente dai dati, eliminando la necessità di estrarre manualmente le **feature**(caratteristiche).

Le CNN sono particolarmente utili per individuare pattern nelle immagini per il riconoscimento di oggetti, volti e scene. Inoltre, possono essere efficaci per la classificazione di dati non immagine come dati audio, serie storiche e segnali.

Le applicazioni che richiedono il riconoscimento di oggetti e la visione artificiale, come i veicoli a guida autonoma e le applicazioni di riconoscimento facciale, si basano ampiamente sulle CNN.

Le CNN vengono utilizzate principalmente per tre motivi:

* Le CNN eliminano la necessità di estrarre manualmente le feature in quanto queste vengono apprese direttamente dalla CNN.
* Le CNN producono risultati di riconoscimento ad alta precisione
* Le CNN possono essere addestrate nuovamente per nuove attività di riconoscimento, consentendo agli utenti di basarsi sulle reti preesistenti.

* 1. Funzionamento generale

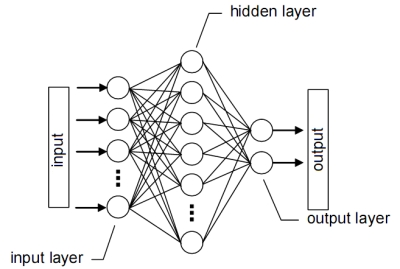
Una rete neurale convoluzionale può avere decine o centinaia di layer, ciascuno dei quali apprende feature diverse di un’immagine.

A ciascuna immagine di addestramento vengono applicati dei filtri a diverse risoluzioni e l’output di ciascuna immagine convoluta viene utilizzato come input per il layer successivo.

I filtri possono essere inizialmente feature molto semplici, ad esempio la luminosità o i bordi, e diventare sempre più complessi fino a includere feature che definiscono in modo univoco l’oggetto.

* 1. Apprendimento delle feature, layer e classificazione

Una CNN è costituita da un layer di input, un layer di output e tanti layer intermedi nascosti.



Il livello di input è costituito da una sequenza di neuroni in grado di ricevere le informazioni dell’immagine da trattare. A questo livello, infatti, verrà passato il vettore di dati che rappresentano i pixel dell’immagine di ingresso. Nel caso ad esempio di una immagine a colori di 32 x 32 pixel il vettore di ingresso dovrà avere una lunghezza di 32 x 32 x 3; in pratica per ogni pixel dell’immagine di dimensione 32 X 32 avremo 3 valori che rappresentano i tre colori dell’immagine in formato RGB (Red, Green e Blue) .

Nel nostro caso utilizzeremo un vettore 200 x 200 x 3.

Questi layer eseguono operazioni che alterano i dati al fine di apprendere le feature specifiche dei dati stessi. Tre dei layer più diffusi sono: la convoluzione, l’attivazione o ReLU e il pooling.

* La convoluzione sottopone le immagini di input a una serie di filtri convoluzionali, ciascuno dei quali attiva determinate feature dalle immagini.
* L’unità lineare rettificata (ReLU) consente di eseguire un addestramento più rapido ed efficace mappando i valori negativi a zero e mantenendo quelli positivi. Questa operazione è talvolta definita attivazione, dal momento che solo le feature attivate vengono trasmesse al layer successivo.
* Il pooling semplifica l’output mediante l’esecuzione di un downsampling (ridurre la dimensionalità delle caratteristiche) non lineare, riducendo in tal modo il numero di parametri che la rete deve apprendere.

Queste operazioni vengono reiterate su decine o centinaia di layer e ciascun layer impara ad identificare feature diverse.

* 1. Bias e pesi condivisi

Peso: peso della connessione usato dal neurone per dare più importanza (peso) alle caratteristiche distintive per quella classe

Bias: Il valore del bias va scelto opportunamente in modo che agisca come soglia effettiva che porta l’output della rete nello spazio di rappresentazione della classe A oppure B

Analogamente a una rete neurale tradizionale, una CNN possiede neuroni con pesi e bias. Il modello apprende questi valori durante l’addestramento e li aggiorna costantemente con ogni nuovo esempio di addestramento. Tuttavia, nel caso delle CNN, i valori dei pesi e dei bias sono gli stessi per tutti i neuroni nascosti in un determinato layer.

Ciò significa che tutti i neuroni nascosti rilevano la stessa feature, come bordi o macchie, in diverse aree dell’immagine. Ciò rende la rete tollerante alla traslazione di oggetti in un’immagine.

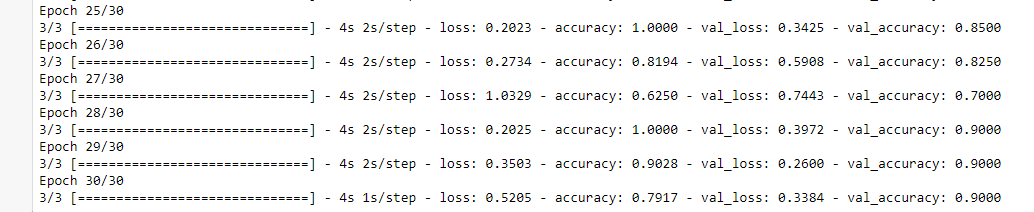
* 1. Layer di classificazione

Dopo aver appreso le feature in numerosi layer, l’architettura di una CNN passa alla classificazione.

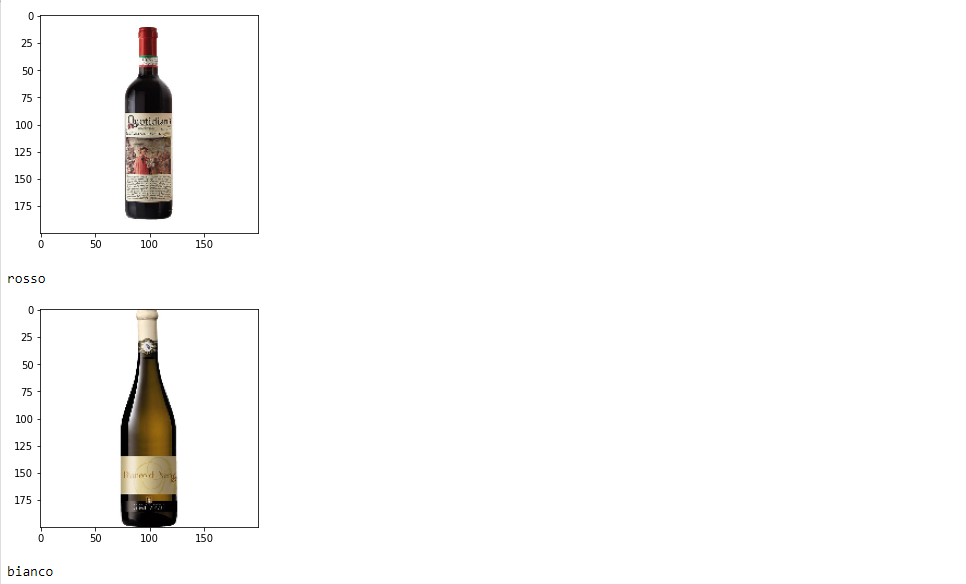
Il penultimo layer è un layer completamente connesso che emette un vettore di dimensioni K dove K è il numero di classi che la rete sarà in grado di prevedere. Questo vettore contiene le probabilità per ciascuna classe di qualsiasi immagine classificata.

L’ultimo layer dell’architettura CNN utilizza un layer di classificazione per fornire l’output della classificazione.

## Risultati



Risultati dell’esecuzione del corretto funzionamento



## Conclusioni

L’obiettivo prefissato inizialmente per il progetto “Bollicine” è stato raggiunto.

Da un’analisi preliminare circa i potenziali competitors, siamo giunti a conclusione che in quest’ambito mancava un progetto che andasse a classificare bottiglie di vino rosso e bottiglie di vino bianco.

A tal proposito, la sfida è stata quella di creare un dataset di immagini che potesse rappresentare al meglio le due categorie.

In definitiva, siamo soddisfatti del lavoro portato a termine in quanto abbiamo creato una repository Github che potrebbe rivelarsi utile anche ad altri utenti.