Realizzazione di una Rete Neurale per la Classificazione di Articoli di Giornale

Advanced Topic in Computer Science
Università degli Studi di Roma Tre

Jhonattan Loza, Anton Shanya, and Nicholas Tucci

CONTENTS

I. INTRODUZIONE

Il progetto che presentiamo in questo articolo, mostra la realizzazione di una Rete Neurale per la classificazione di articoli di giornali italiani, nello specifico una volta che è stata allenata sarà in grado di classificare un articolo di giornale in input in una delle relative categorie (i.e. sport, politica, economia...) con una accuracy che si attesta intorno al 90%.
Per poterla realizzare ci siamo basati su Keras, una libreria che si poggia su Tensorflow o Theano, che permette di implementare una rete neurale ad alto livello; questo senza scendere a livelli implementativi troppo dettagliati, evitando di andare a specificare i singoli elementi della rete, ma interagendo in modo più generale con essa, andando a definire ad esempio il numero di strati e la relativa tipologia della rete neurale. Inoltre è stato necessario adattare i dati di input (le parole che compongono gli articoli di giornale) alla rete neurale che abbiamo implementato.

II. DATASET

Come input per allenare la rete neurale, ci è stato richiesto di utilizzare articoli di giornale scritti in lingua italiana; non essendo disponibili grandi dataset con queste caratteristiche, abbiamo fatto dello *Scraping* da alcuni dei siti di giornali più importanti italiani (Messaggero, Ansa, Repubblica), scrivendo per ogni sito uno script Python che tramite delle operazioni GET del protocollo HTTP li scaricasse automaticamente.
In particolare ci siamo basati sugli RSS che questi siti mettono a disposizione per ogni tipologia di articolo di giornale e li abbiamo raggruppati in 7 categorie (Cronaca, Economia, Mondo, Politica, Scienza e Tecnologia, Società, Sport). Certi siti forniscono anche categorie più specifiche (e.g. calcio), ma per ottenere una accuracy migliore abbiamo scelto queste categorie più generiche. Così facendo siamo riusciti ad ottenere un dataset di più di 2000 articoli di giornale.
In particolare il dataset è stato così strutturato: una cartella per ogni categoria, dove all'interno ci sono i relativi articoli di giornale sotto forma di file testuali.

III. ADATTAMENTO DEL DATASET ALLA RETE NEURALE

A. Data Cleaning

Il dataset a disposizione è composto quindi da file testuali (che contengono rispettivamente ognuno un articolo di giornale) messi all'interno di cartelle che raggruppano gli articoli per argomento; questo dataset dovrà essere usato per allenare la rete neurale, ma prima è necessario fare del *Data Cleaning*. Abbiamo fatto prima una pulizia grezza con delle espressioni regolari, per rimuovere i simboli di punteggiatura e portare tutti i caratteri al *lower case*.
Abbiamo proseguito applicando delle *stop words*: basandoci su una libreria esterna, abbiamo ottenuto questa collezione di parole della lingua italiana che possono essere considerate irrilevanti dal punto di vista informativo (ad esempio articoli, preposizioni, congiunzioni) e che quindi per migliorare l'efficienza e l'efficacia della futura fase di *training* della rete neurale, abbiamo rimosso.
Infine abbiamo applicato l'operazione di `stemming`, per poter ricondurre tutte le parole con una radice comune ad un'unica parola. Si faccia l'esempio della parola "mangiare" e tutte le sue declinazioni (mangio, mangerò, mangiammo...), ai fini di *training* ha poco senso trattare distintamente le varie forme quando esprimono sempre lo stesso concetto.

B. Dictionary

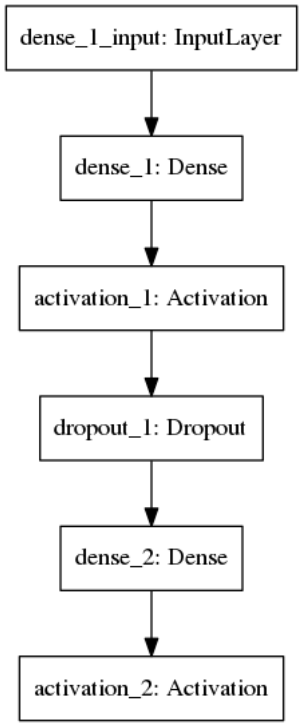
Una volta terminata la fase di *Data Cleaning*, abbiamo dovuto adattare i dati a disposizione all'*input* richiesto dalla libreria `Keras`.
Inizialmente andiamo a leggere da *File System* (con la struttura a cartelle precedentemente illustrata) i vari file (articoli) all'interno di cartelle (il cui nome rappresenta la categoria); già da questa fase di lettura andiamo a discriminare il *training set* dal *test set*, scegliendo in modo randomico di assegnare l'80% dei file al *training set* e il restante 20% al *test set*, come suggerisce la letteratura delle reti neurali. In particolare nella collezione di *training set* (o *test set*), per ogni file passiamo un array composto dalle parole che contenute nell'articolo (su cui è effettuata l'operazione di *Data Cleaning*) più il nome della cartella (la categoria) in ultima posizione.
A questo punto ci andiamo a costruire un *Dictionary* (della libreria `Gensim`) di parole, contenente appunto tutte le parole che sono state lette da tutti gli articoli nel dataset (compresi i nomi degli argomenti) che ci sarà utile in seguito.
Per come è strutturato keras, richiede una struttura dati composta da 4 insiemi: (x_train, y_train), (x_test, y_test) dove x indica l'elemento con cui allenare (e testare) la rete neurale, e y la classe predetta; ovviamente abbiamo dovuto far attenzione che ad ogni parola dell'articolo nell'insieme x corrispondesse la categoria y da predire; perciò abbiamo adattato il nostro dataset in questa maniera. In particolare era richiesto un valore numerico piuttosto che una stringa, pertanto grazie al Dictionary ed alla sua funzione `token2id`, abbiamo associato un id numerico univoco ad ogni parola del nostro dataset, in modo tale che se all'interno di esso vi fosse nuovamente una parola già memorizzata, questa venisse identificata con lo stesso valore.
Perciò una volta separati i valori x e y come descritto, sono pronti per essere passati in input a Keras.

IV. DEFINIZIONE DELLA RETE NEURALE

Una volta che si ha a disposizione il dataset adattato come richiesto dalla libreria di `keras`, basta richiamare le funzioni mostrate nella rispettiva documentazione per passargli l'input corretto. In particolare è stata utilizzata una rete neurale sequenziale, in cui abbiamo aggiunto un primo strato di nodi, aventi una *ReLU* come funzione di attivazione, seguita da un secondo strato di nodi con una *softmax* come funzione di attivazione; grazie ad un tool grafico messo a disposizione da `Keras`, è stato possibile anche tracciare un *workflow* grafico della rete neurale implementata, qui riportato{fig. 1}.

In maniera sperimentale abbiamo poi settato il numero di epoche per allenare la rete pari a 50 e il *batch size* (la dimensione di parole alla volta che gli viene passata alla rete) pari a 256. A questo punto viene stabilito un input da predire per usare la rete neurale appena allenata; questo input sarà un articolo di giornale di cui si vuole conoscere la categoria di appartenenza. Verrà effettuato lo stesso procedimento di adattamento che era stato fatto per il dataset, verrà quindi inserito nello stesso dictionary del dataset e infine gli verrà fatta attraversare la rete neurale. Il risultato sarà la categorizzazione dell'articolo di giornale passato in input.
Per evitare di allenare la rete neurale ogni volta che si vuole predire un articolo (essendo anche un processo oneroso e che richiede molte risorse computazionali), alla fine del training facciamo salvare la rete neurale sul file system come un file .h5: se è già presente questo file, non sarà necessario allenare nuovamente la rete neurale.

FIG. 1. Rete Neurale



V. CONCLUSIONI E SVILUPPI FUTURI

La rete neurale così definita, alla fine del training ha raggiunto una accuracy dell'86% e testando con altri articoli presi dagli stessi siti giornalistici che abbiamo utilizzato per costruire il dataset, la prediction risulta corretta tranne in alcuni casi ambigui. Usando invece altri articoli provenienti da altri siti giornalistici, da test sperimentali si è notato che la corretta prediction avveniva sotto il 50% dei casi; il problema risiede nel fatto che la rete neurale non conosce questa tipologia di scrittura propria di altre testate giornalistiche e pertanto l'accuracy cala. Uno sviluppo futuro potrebbe quindi essere proprio quello di arricchire il dataset di articoli provenienti da più giornali, rendendolo così robusto a input sconosciuti.