

Relazione progetto di Intelligenza Artificiale

Informazioni sul progetto

Redatto	Francesco Corti - 1142525
	Giovanni Sorice - 1144558
Referente	Giovanni Sorice - 1144558 giovanni.sorice@studenti.unipd.it

Link al sito

<http://tecweb1819.studenti.math.unipd.it/gsorice/>

Descrizione

Documento riportante le informazioni relative al progetto di intelligenza artificiale.



Indice

1	Introduzione	2
1.1	Il problema dello spam	2
2	Panoramica	3
3	Metodologie utilizzate	3
3.1	Reti neurali	3
3.1.1	Configurazione	3
3.2	Logistic Regression	4
4	Realizzazione	5
5	Validazione	5
6	Comparazioni	5
7	Conclusioni	5
	Appendice	5
A	Tabelle riassuntive	5



1 Introduzione

1.1 Il problema dello spam

Il problema dello spam affligge da una decina di anni tutte le persone che dispongano di una casella di posta elettronica oppure di uno smartphone.

In passato è stato constatato come l'eliminazione manuale dei messaggi spam, data la quantità di messaggi inviati, presentasse costi di tempo insostenibili.

Questo ha portato ad uno sviluppo di tecniche algoritmiche che permettessero di classificare automaticamente un messaggio ricevuto, come spam o **ham**.

Si è però scoperto che un approccio di tipo *offline learning*, presentava dei problemi.

Gli spammer, persone o bot che spediscono messaggi spam, riuscivano a modificare i messaggi in modo da renderli classificati come ham dai sistemi anti-spam presenti.

Questo era possibile in quanto gli algoritmi non evolvevano nel tempo, cambiando quindi la struttura del messaggio di spam questo veniva erroneamente identificato come un messaggio non spam.

Si è quindi passati a un approccio di tipo **online** che si è visto essere quello più ottimale.

Gli algoritmi in questo modo non smettono di imparare una volta terminato l'input dei dati ma evolvono nel corso del tempo imparando a classificare nuove tipologie di messaggi come spam.



2 Panoramica

3 Metodologie utilizzate

Per il progetto didattico abbiamo deciso di utilizzare i seguenti approcci: l'algoritmo di *Logistic Regression* e le *Neural Networks*.

Tuttavia sono possibili altri approcci che, per rimanere all'interno dei tempi stabiliti, non siamo riusciti ad affrontare.

3.1 Reti neurali

Il primo approccio che abbiamo scelto di utilizzare è stato quello delle reti neurali. Per farlo ci siamo appoggiati alla libreria *TensorFlow* la quale, dalla versione 2.0.0, integra *Keras* al suo interno.

Maggiori informazioni riguardanti l'integrazione di Keras sono presenti al seguente link tf.keras.

Questo permette di utilizzare TensorFlow come ecosistema e definisce la rete tramite delle classi messe a disposizione nelle API di Keras.

3.1.1 Configurazione

È stata scelto di utilizzare il tipo di rete *Sequential*, il problema infatti non prevedeva l'utilizzo di particolari tipi di layer.

In tal caso sarebbe stato opportuno utilizzare una rete di tipo *Functional* che permette una maggiore libertà per la definizione della struttura della rete. Tramite il metodo *compile()* presente in Keras è possibile stabilire la:

1. **Loss functions:** *binary_crossentropy* nel nostro caso, maggiori info al seguente link [loss functions](#);
2. **Optimizer:** l'ottimizzatore che verrà utilizzato per l'aggiustamento dei pesi e per minimizzare la loss function. Nel nostro caso *Adam*.
3. **Metrics:** la lista di metriche che verranno valutate dal modello durante la fase di training e testing. Nel nostro caso è stata scelta la metrica di *binart accuracy*.

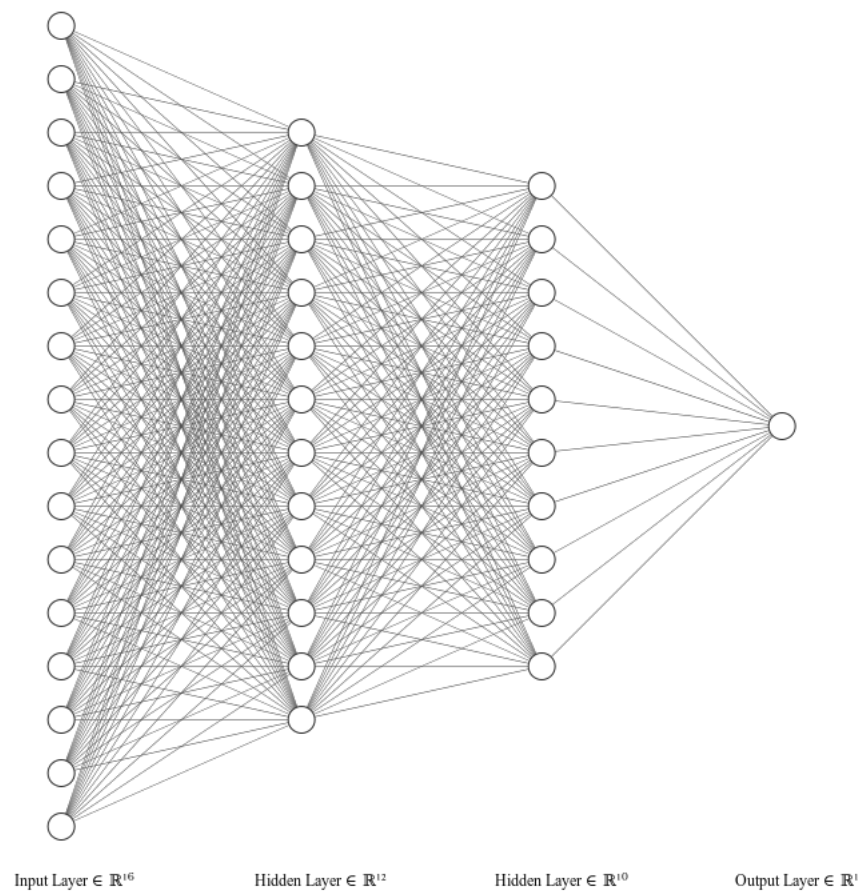


Figura 1: Esempio di rete neurale

3.2 Logistic Regression

S



4 Realizzazione

5 Validazione

6 Comparazioni

7 Conclusioni

A Tabelle riassuntive

Riferimenti