Report Progetto Machine Learning

Alessandro Arrighi

Settembre 2024

Laboratorio di Ottimizzazione, Intelligenza Artificiale e Machine Learning
Corso di Tecnologie dei Sistemi Informatici
Alma Mater Studiorum - Università di Bologna

Contents

1 Introduzione

Il seguente documento specifica l'implementazione di un modello di rete neurale per il sentitment analysis. Questo tipo di analisi vede come obbiettivo quello di identificare il linguaggio naturale secondo il tono emotivo utilizzato. I dati devono essere classificati secondo una percezione dell'essere umano. Possono categorizzarsi come sentiment analysis: la percezione di bello o brutto, positivo o negativo, buono o cattivo ecc. Il modello deve essere addestrato in modo da comprendere il contesto, e ottenere le informazioni necessarie all'analisi. Risulta essere estremamente complicato comprendere tutte le sfaccettature del linguaggio naturale. Il sarcasmo, l'ironia, l'interpretazione di caratteri speciali come le emoji, rendono molto difficoltoso l'apprendimento.

2 Dataset

Una volta compreso il problema, è importante cominciare con la valutazione e analisi del dataset. I dati devono passare una serie di processi prima di poterli utilizzare, in modo da estrapolarne le informazioni necessarie. Il dataset utilizzato è costituito da una serie di frasi associate ad una etichetta binaria: positivo o negativo.

2.1 Analisi del Dataset

Come primo passaggio è stato doveroso accertarsi che il dataset fosse idoneo all'utilizzo. Sono stati effettuati controlli sulla dimensione generica del dataset, la lunghezza delle frasi, il bilanciamento tra i dati e il numero di parole utilizzate.

2.2 Manipolazione dei dati

I dati sono stati successivamente manipolati, così da poterli rendere usufruibili alla macchina per il suo addestramento. È stato necessario rimuovere eventuali valori duplicati per evitare ridondanza nel dataset. La "pulizia" dei dati, è un termine utilizzato per definire la rimozione di caratteri speciali non rilevanti all'addestramento del modello. Questo passaggio permette di rimuovere rumore dal dataset così da favorire un migliore utilizzo. In aggiunta è buona pratica effettuare anche una normalizzazione sui dati. Questo può essere fatto andando a convertire tutti i caratteri in minuscolo, così da garantire una maggiore coerenza nei dati e una minore ridondanza. Ora il dataset è pronto per essere diviso in token, ovvero piccole unità di testo tutte della stessa lunghezza. I token permettono quindi, in fase di caricamento del dataset, di ottenere dei tensori di dimensione fissa. Con i dati puliti si crea una bag of words, ovvero un vocabolario contenente, in modo univoco, tutte le parole presenti nel dataset. Ad ogni parola viene associato un valore numerico, anch'esso univoco, in modo tale da poter codificare in caratteristiche numeriche tutti i dati.

Un'aggiunta a questa fase è la rimozione delle stop words. Con questo termine si indicano tutte quelle parole non significative nel contesto della frase (il,

e, con...), ma che aumentano il rumore dei dati. Un ulteriore step da applicare è lo stemming, processo che prevede la riduzione delle parole alla loro radice. In questo modo è possibile snellire il dataset mantenendo il significato semantico.

3 Architettura

L'architettura adotta per questo problema è una rete neurale ricorrente. Con una rete di questo tipo, è possibile creare persistenza dei dati lungo la rete, creando così una sua memoria. Il modello è quindi in grado di elaborare dati sequenziali mantenendo le informazioni precedentemente elaborate. In questo modo il modello riesce ad interpretare le frasi mantenendo memorizzate le informazioni passate. Nello specifico, i tre layer principali che costituiscono il modello sono: un layer di embedding, un layer LSTM e un layer sigmoid.

3.1 Embedding Layer

Il layer di embedding ha lo scopo di trasformare, i token di parole precedentemente codificati in interi, in moduli vettoriali densi. Così si è in grado di rappresentare i dati in uno spazio più ridotto. Essendo che i token codificati soffrono del problema di alta dimensionalità, il layer di embedding è in grado di risolverlo. Inoltre, questo layer permette di gestire l'input estrapolandone la semantica, così da raggruppare i dati più simili tra loro. L'output del layer è un insieme di vettori densi, ognuno dei quali rappresenta una parola ottenuta in input.

3.2 LSTM Layer

I valori di embedding ottenuti sono in seguito, l'input per il layer LSTM. Questo è una versione migliorata delle RNN che permettono così di apprendere dipendenze a lungo termine nei dati sequenziali. Nel caso del sentiment analysis, il lavoro che la rete deve compiere è su una lunga serie di dati, questo rende l'LSTM una buona soluzione al problema. L'archiettura è costituita da una cell state che rappresenta una memoria a lungo termine e da un hidden state che rappresenta una memoria a breve termine. Con questi elementi l'archietettura è in grado di comprendere lunghe sequenze di dati ed estrapolarne le loro informazioni.

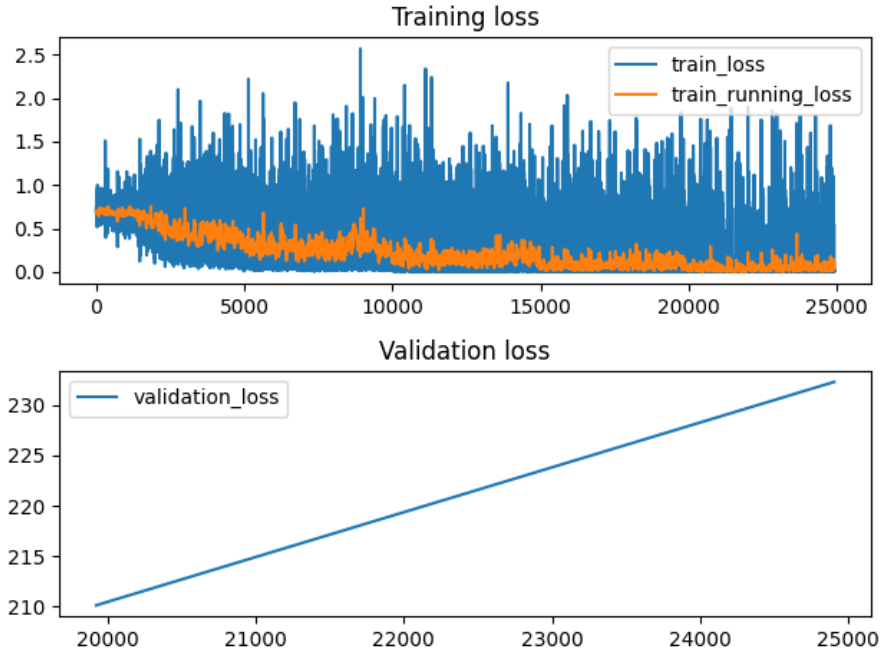
3.3 Sigmoid Layer

L'output dal layer LSTM deve passare infine come input di un layer sigmoid che elaborerà i dati normalizzati in un intervallo compreso tra 0 e 1. Il suo output non verrà preso in toto bensì una sola parte di questo. Solo l'ultimo elemento di ogni vettore viene selezionato ottenendo un vettore monodimensionale. Questo avviene poiché nelle reti LSTM, l'ultimo elemento risulta essere quello più significativo, essendo l'ultimo che viene prodotto in ordine temporale. I valori precedenti invece, potrebbe contenere informazioni parziali.

4 Analisi dei dati ottenuti

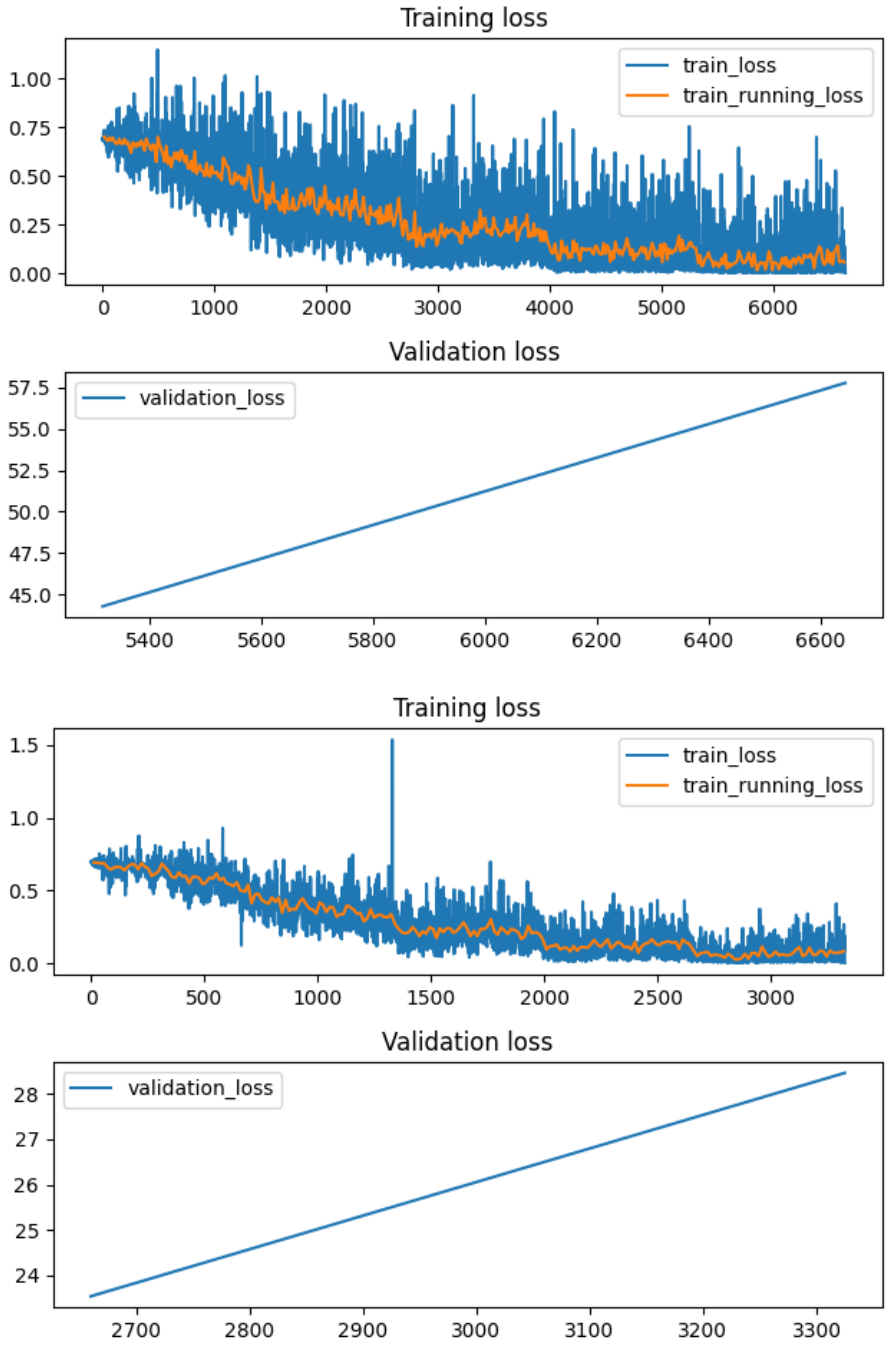
Completata tutta l'architettura non rimane altro che eseguire il modello modificando gli hyper parametri così da poter gestire il comportamento del modello. In questo modo è possibile ottenere una configurazione che permetta al modello di convergere.

4.1 Test 1



Il primo test che è stato eseguito proponeva come batch size un valore di 4, per il learning rate 0.001 ed infine 5 epoche per l'addestramento. Come si può visionare dal grafico, la loss di training, con il proseguire dell'addestramento diminuisce di valore, stando ad indicare che il modello risponde positivamente. Per quanto riguarda la loss di validazione, il grafico un valore molto alto che, in modo lineare, aumenta con l'aumentare delle epoche nell'addestramento. Questo è un fenomeno che può essere riconducibile all'overfitting: condizione in cui il modello fornisce previsioni accurate solo in fase di addestramento. Il modello quindi si focalizza troppo sulle caratteristiche del dataset di training, per questo è opportuno provare ad aumentare il batch size. Infatti, l'accuracy arriva a toccare il 99% con il dataset di training, mentre per la validazione e la fase di test, questa si ferma circa a metà addestramento intorno ad 81% senza miglioramenti.

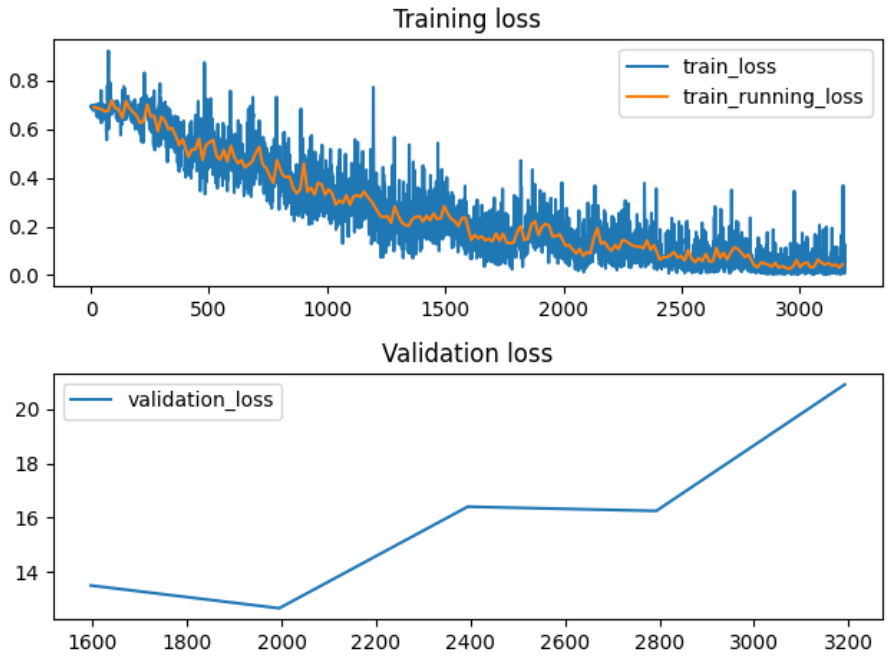
4.2 Test 2



In questi due test il batch size è stato impostato a 15 e 30. L'andamento della

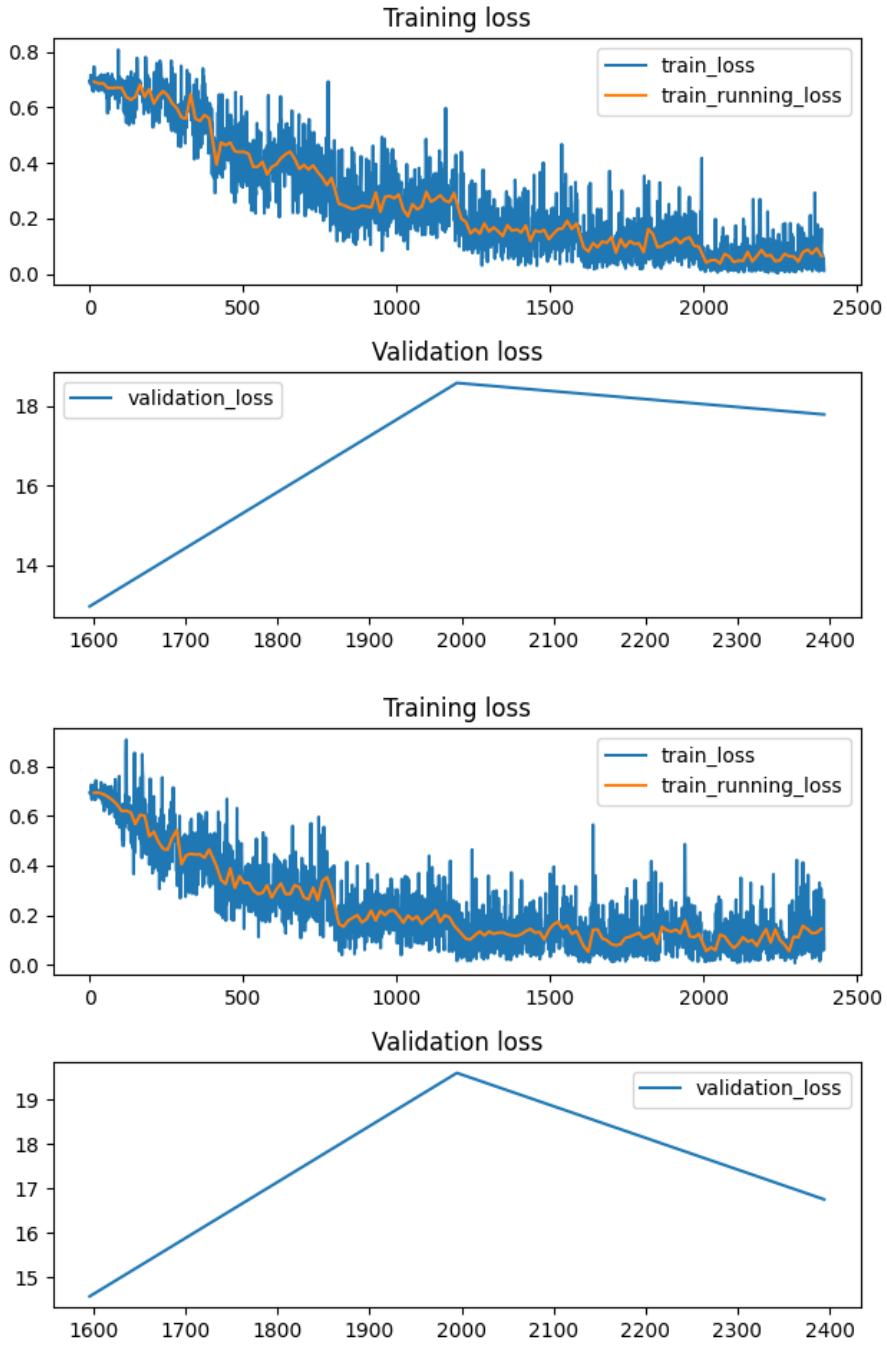
loss di training in entrambi gli addestramenti è similare. È possibile visionare invece come la loss di validazione, abbia sempre un andamento lineare crescente. È però importante notare che i valori della loss diminuiscano al crescere della batch size. Questo è importante per comprendere che il fenomeno dell'overfitting è in calo. I risultati dell'accuracy di validazione e test sono similari, circa l'80%, però dalla seconda epoca in poi, questa si stabilizza ed in alcuni casi diminuisce leggermente.

4.3 Test 3



In questo test è stato aumentato nuovamente il batch size a 50 e il numero di epoche a 8. Qui è possibile vedere un miglioramento della loss di validazione. Infatti, questa si trova in un range di valori ancora più inferiore, ma soprattutto la funzione ha un andamento per alcune fasi costante. Il valore dell'accuracy di validazione e test rimane stabile intorno all'80% con una piccola perdita di qualche percentuale verso fine addestramento.

4.4 Test 4



Gli ultimi due test sono stati effettuati aumentando il learning rate, questo

perché permette al modello di apprendere più velocemente. Infatti, questa è una tecnica che si utilizza per diminuire la funzione di loss. Nel primo grafico è stato assegnato 0.002 al learning rate mentre nel secondo 0.008. È possibile visionare come la loss di training sia leggermente aumentata mentre la loss di validazione risulti crescente fino a metà addestramento e poi comincia a diminuire.

5 Conclusioni finali

In conclusione, il modello arriva ad un'accuratezza massima pari all'80% in quasi tutte le casistiche testate. La differenza che si può notare è nell'andamento della loss di validazione. Per risolvere in parte il problema dell'overfitting è stata adottata come strategia l'aumento della batch size in modo da rendere l'addestramento del modello più generico. In un secondo, ottenuto un valore di batch size idoneo, è stato aumentato il learning rate. Così facendo si è cercato di evitare che il modello non si adattasse eccessivamente al set di training. Per rendere questi test più significativi, si dovrebbe aumentare il quantitativo di dati. In questo modo è possibile migliorare anche l'accuratezza che in tutte le casistiche, si è sempre bloccata all'80% con alcune perdite.