Relazione sul Fine Tuning di BERT per Sentiment Analysis

Nome Autore

10 aprile 2025

1 Introduzione

Il presente documento analizza i risultati ottenuti durante il fine tuning del modello `bert-base-uncased` applicato ad un task di analisi del sentiment su un dataset composto da 9.600 esempi per il training, 2.400 per la validazione e 1.773 per il test. La relazione prende in esame più esecuzioni (dal training "più vecchio" al più recente) riportandone log, metriche di valutazione e commenti tecnici relativi all'addestramento e alla convergenza del modello.

2 Setup Sperimentale

Per ogni training è stato utilizzato:

- **Dataset:**
  - Training: 9.600 esempi
  - Validazione: 2.400 esempi
  - Test: 1.773 esempi
- **Strumenti:**
  - Libreria `transformers`
  - Modello base: `bert-base-uncased`
- **Metodologia:** Diverse esecuzioni sono state realizzate con variazioni nei parametri di training (numero di step, batch size e tempi di esecuzione) e, in alcuni casi, sono stati visualizzati warning relativi alla inizializzazione dei pesi del layer `classifier`.

3 Descrizione delle Esecuzioni

Le esecuzioni riportate sono ordinate cronologicamente, dal training più vecchio al più recente.

3.1 Prima Esecuzione

- **Training:** 1.800 steps per 3 epoche, tempo totale di circa 13:40.
- **Andamento della Loss durante il training:**
  - Step 500: 0.235900
  - Step 1.000: 0.117700
  - Step 1.500: 0.080000
- **Valutazione sul test set:**
  - **eval_loss:** 0.633635
  - **Accuracy:** 0.900169
  - **Precision:** 0.903983
  - **Recall:** 0.900169
  - **F1-score:** 0.899129
  - Tempo di eval: 13.6866 secondi (con 129.543 esempi al secondo)

3.2 Seconda Esecuzione

- **Training:** 900 steps per 3 epoche, tempo totale di circa 04:57.
- **Andamento della Loss durante il training:**
  - Step 500: 0.318500
- **Valutazione sul test set:**
  - **eval_loss:** 0.474599
  - **Accuracy:** 0.899041
  - **Precision:** 0.902639
  - **Recall:** 0.899041
  - **F1-score:** 0.898018
  - Tempo di eval: 3.5089 secondi (505.279 esempi al secondo)

Si osserva come la riduzione del numero di step (o il diverso scheduling del training) abbia condotto a tempi di addestramento e valutazione significativamente inferiori rispetto alla prima esecuzione, con un miglioramento nella loss di valutazione.

3.3 Terza Esecuzione

La terza esecuzione risulta identica alla seconda esecuzione in termini di log, metriche e tempo di esecuzione. Questo conferma la ripetibilità dei risultati con le impostazioni adottate.

3.4 Quarta Esecuzione

- **Warning iniziale:** Viene segnalata l'inizializzazione casuale dei pesi del layer `classifier` (cioè `classifier.bias` e `classifier.weight`). Tale messaggio evidenzia la necessità di effettuare il training sul task downstream.
- **Training:** 900 steps per 3 epoche, tempo totale di circa 04:57. Viene fornito un *training log* dettagliato con i seguenti valori:
  - Step 100: Loss 0.620600, learning rate $\sim 3.92 \times 10^{-6}$
  - Step 200: Loss 0.357300, learning rate $\sim 7.92 \times 10^{-6}$
  - Step 300: Loss 0.216700, learning rate $\sim 1.188 \times 10^{-5}$
  - Step 400: Loss 0.168600, learning rate $\sim 1.588 \times 10^{-5}$
  - Step 500: Loss 0.141000, learning rate $\sim 1.988 \times 10^{-5}$
  - Step 600: Loss 0.146300, learning rate $\sim 1.515 \times 10^{-5}$
  - Step 700: Loss 0.088100, learning rate $\sim 1.015 \times 10^{-5}$
  - Step 800: Loss 0.099200, learning rate $\sim 5.15 \times 10^{-6}$
  - Step 900: Loss 0.075200, learning rate $\sim 1.5 \times 10^{-7}$
- **Valutazione sul test set:**
  - **eval_loss:** 0.501045
  - **Accuracy:** 0.900169
  - **Precision:** 0.903440
  - **Recall:** 0.900169
  - **F1-score:** 0.899212
  - Tempo di eval: 3.5743 secondi (496.039 esempi al secondo)

In aggiunta, viene fornita una tabella riassuntiva che riprende le metriche finali, evidenziando la robustezza del modello in termini di accuratezza ed F1-score.

Metric	Valore
Eval Loss	0.501045
Accuracy	0.900169
Precision	0.903440
Recall	0.900169
F1-score	0.899212

Tabella 1: Risultati di valutazione della quarta esecuzione

4 Analisi Comparativa e Commenti Tecnici

4.1 Convergenza e Andamento della Loss

Dagli output si osserva che:

- La prima esecuzione, pur avendo un numero di step significativamente maggiore (1.800 step rispetto ai 900 delle altre esecuzioni), porta a una *eval_loss* relativamente maggiore (0.633635).
- Nelle esecuzioni successive (seconda e terza) la *eval_loss* risulta ridotta (circa 0.474599), suggerendo una migliore convergenza del modello.
- La quarta esecuzione, pur partendo con un warning di inizializzazione, mostra un andamento della loss consistente con i precedenti training (loss che scende progressivamente fino a 0.075200), con un *eval_loss* finale di 0.501045.

Questo evidenzia come, con un opportuno scheduling del learning rate e adeguate impostazioni del training, BERT riesca a convergere in modo stabile, producendo metriche di performance simili in termini di accuratezza (intorno al 90%) e F1-score (circa 0.899).

4.2 Tempi di Addestramento e Velocità di Evaluazione

Si noti una netta differenza nei tempi di esecuzione:

- La prima esecuzione ha richiesto circa 13 minuti e 40 secondi per l'addestramento e un tempo di valutazione di quasi 14 secondi, con una velocità di campioni per secondo significativamente inferiore.
- Le esecuzioni successive (la seconda, terza e quarta) hanno richiesto circa 5 minuti per il training, con una velocità di valutazione molto più alta (circa 500 esempi al secondo).

Questi dati indicano che l'ottimizzazione dei parametri di training (come batch size, learning rate scheduler, e numero di step) ha avuto un impatto rilevante sia sulle performance in termini di convergenza sia sui tempi di esecuzione.

4.3 Inizializzazione dei Pesi e Avvertimenti

Nella quarta esecuzione si evidenzia il seguente warning:

```
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at bert-base-uncased and are newly initialized:
['classifier.bias', 'classifier.weight']
```

Ciò è normale nel caso in cui si stia utilizzando un modello pre-addestrato per il task specifico (in questo caso la classificazione) in cui il layer finale non è presente nel checkpoint originale. L'avvertimento non impatta negativamente il risultato, ma conferma la necessità di addestrare il modello sul task di destinazione.

5 Conclusioni

Dall'analisi dei risultati ottenuti si può concludere quanto segue:

- I risultati in termini di accuratezza, precisione, recall e F1-score risultano stabili attorno al 90% e 0.9, evidenziando un buon rendimento del modello per il task di analisi del sentiment.
- Le variazioni nei tempi di training ed evaluation suggeriscono che la configurazione del training (numero di step, batch size e scheduler del learning rate) incide significativamente sulle prestazioni in termini di efficienza.
- Il warning relativo all'inizializzazione dei pesi del classifier è atteso e non comporta particolari criticità nel contesto del fine tuning.

I dati complessivi indicano una convergenza efficace del modello, pur manifestando alcune differenze nei tempi di esecuzione e nella *eval_loss* che possono essere oggetto di ulteriori analisi per una migliore comprensione dei trade-off tra efficienza e performance.