Alma Mater Studiorum - Università di Bologna

Relazione per il corso di Data Science

Liam Cavini
Semestre Invernale 2024/2025

7° Foglio, Reti Neurali
15/01/2025

Risorse

Il codice utilizzato, insieme al file .tex di questo documento, possono essere trovati nella seguente repository github: https://github.com/LazyLagrangian/data_science.

Esercizio 1 - Somme di numeri con le reti neurali

L'esercizio consiste nell'allenare una rete neurale capace di sommare due numeri interi compresi tra 0 e 999. I dataset di training e testing, composti da una coppia di numeri (gli addendi), sono stati generati casualmente. Complessivamente i due dataset contengono 10000 coppie, 2000 per il testing e 8000 per il training.

La rete neurale allenata è costituita da due layer nascosti densi formati da 64 neuroni, con attivazione ReLU, e un layer di output denso composto da un singolo neurone.

La rete è stata allenata per 100 epoche, ottenendo un'accuratezza maggiore del 99% sul dataset di training, con tolleranza 0.5. Dunque più del 99% degli output nel testing hanno come intero più vicino il risultato della somma.

L'evoluzione del loss durante il training è mostrato in figura 1.

La tabella 1 mostra alcuni esempi di input e i corrispondenti output forniti dalla rete neurale.

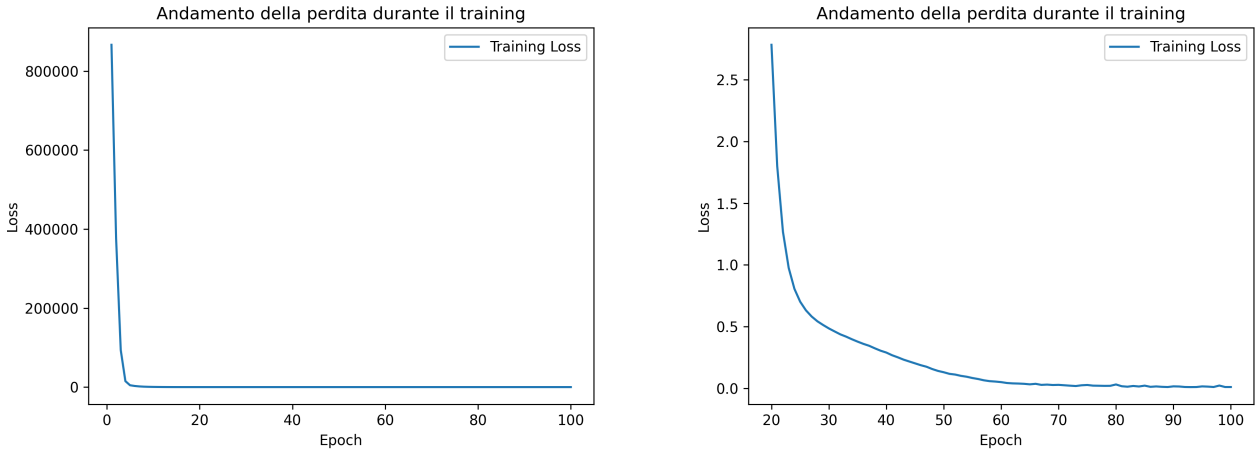

Figura 1: *L'immagine a sinistra mostra il training loss in funzione dell'epoch. L'immagine a destra è una versione 'zoomed' del grafico del loss che parte alla 20-esima epoch.*

Input	**Output ideale**	**Output effettivo**
$(5, 7)$	12	11.99
$(100, 200)$	300	299.98
$(400, 200)$	600	599.92
$(999, 999)$	1998	1997.74

Tabella 1: *Esempi di somme compiute dalla rete neurale. Nella prima colonna gli addendi, nella seconda la somma, e nella terza l'output della rete neurale.*

Esercizio 2 - Segno di una somma di numeri con le reti neurali

L'esercizio consiste nel modificare la rete neurale di regressione dell'esercizio precedente in una di classificazione. La rete prende in input una coppia di numeri, e la classifica in base al segno della loro somma, che può essere negativo, positivo o nullo.

Sono quindi stati modificati i label, rendendoli idonei alla classificazione. Per fare ciò potrebbe sembrare che sia sufficiente scambiare [1]:

```
labels = inputdata[:,0] + inputdata[:,1]
```

con

```
labels = np.sign(inputdata[:,0] + inputdata[:,1])
```

Ma il modo corretto di modificare i labels è invece di usare un one-hot encoding. Nel codice si è usato $[1, 0, 0]$ per la classe dei valori negativi, $[0, 1, 0]$ per la classe dei valori nulli, $[0, 0, 1]$ per la classe dei valori positivi.

La rete è stata prima allenata sullo stesso dataset dell'esercizio 1, ottenendo una accuratezza del 100%, attesa dato che il dataset è composto da solo numeri positivi [2].

Successivamente è stato modificato il dataset in modo da essere composto da coppie di interi compresi tra -999 e 999. Si è ottenuta una accuratezza maggiore del 99%.

L'evoluzione del training loss per entrambi i casi è mostrata in figura 2.

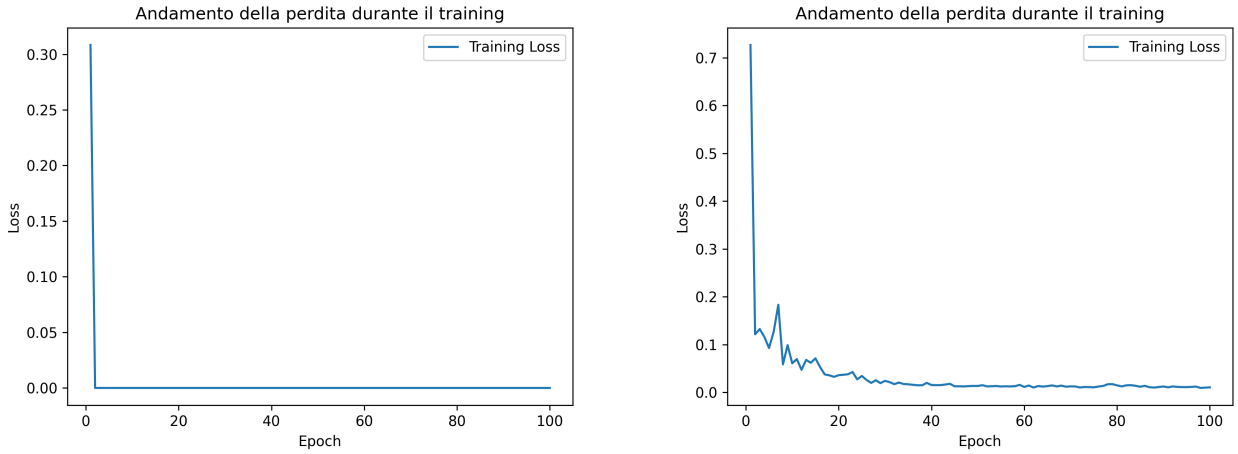

Figura 2: *L'immagine a sinistra mostra il training loss in funzione dell'epoch per il training svolto sul dataset dell'esercizio precedente, quella a destra per il dataset modificato.*

[1] Qua seguo la convenzione della consegna, le variabili nel codice effettivo hanno nomi diversi.
[2] Si potrebbero anche ottenere valori nulli nel dataset, ma la probabilità è di $\sim 1\%$.