

Machine Learning - Challenge 3

Abdula Kalus

Svolgimento challenge 3, corso di Intelligenza Artificiale, Università degli Studi di Trieste

Abstract

In questa challenge si vuole creare un modello capace di classificare correttamente dei caratteri giapponesi scritti a mano, il dataset Kuzushiji-MNIST (KMIST) verrà usato per il training ed il testing del modello il quale contiene 70.000 immagini in scala di grigi 28x28 che coprono 10 classi. Il dataset inoltre è perfettamente bilanciato come il dataset MNIST originale (6k/1k train/test per ogni classe), il dataset prima di essere usato per il training è stato normalizzato con media 0 e varianza 1.

Modelli proposti

I modelli proposti per la classificazione sono 3 modelli di Fully Connected Neural Network (FCNN) ed un modello di Convolutional Neural Network (CNN), per ciascun modello è stato fatto il tuning degli iperparametri inoltre durante il training sono stati testati diversi algoritmi di ottimizzazione, i modelli proposti sono i seguenti:

FCNN1 modello con un unico hidden layer lineare fully connected e funzione di attivazione ReLU (output del layer 64)

FCNN2 modello con un 2 hidden layer lineari fully connected e funzione di attivazione ReLU per entrambi (output l1 = 128 ed output l2 = 64)

FCNN3 modello con un 3 hidden layer lineare fully connected e funzione di attivazione ReLU per tutti i hidden layer (output l1 = 256, output l2 = 128 ed output l3 = 64)

CNN modello convolutivo con 3 hidden layer

1. layer convolutivo 2d con 1 in channel e 8 out channel
2. layer di max pooling (W/2 ed H/2) quindi dimensione dimezzata
3. layer convolutivo con 8 in channel e 16 out channel

anche in questo modello la funzione di attivazione per gli hidden layer è la ReLU

tutti i modelli come output layer hanno un layer lineare completamente connesso, al quale non viene applicata nessuna funzione di attivazione dato che tutti i modelli usano come criterio di valutazione della bontà la **Cross Entropy Loss**, la quale applica direttamente **SoftMax**. Per quando riguarda gli algoritmi di ottimizzazione per il training sono stati usati

Adam e **SGD**, con applicando anche lo scheduler in alcuni casi ed il momentum per **SGD**.

Nella tabella 1 sono riassunti i risultati dei modelli

Modello	Optim	Accuracy
FCNN1	Adam	85.69
FCNN2	Adam	89.03
FCNN3	Adam	90.32
FCNN3	Adam + scheduler	91.32
CNN	Adam	89.07
CNN	Adam + scheduler	91.96
CNN	SGD	90.57

Table 1. Tabella riassuntiva accuracy

escludendo il modello con un layer FCNN1 possiamo osservare che lo score degli altri modelli è pressoché lo stesso e che nessuno di loro supera il 92% di accuracy, questo può essere dovuto a un problema del test set, infatti potrebbero essere presenti delle immagini con molto rumore le quali non sono classificabili correttamente. Un altro punto su cui porre attenzione è che i training con scheduler hanno portato un miglioramento all'accuracy, anche se piccolo, questo può significare che diminuendo il learning rate si è riusciti ad avvicinarsi di più al minimo della funzione, ovvero una volta arrivati ad un punto molto al minimo e diminuendo il learning rate si riesce a rimanere e andare verso il minimo senza saltarlo.

Train vs Test loss di FCNN3

In figura 1 è riportato l'andamento del training (blu) del modello per singolo batch contro il testing (arancione) per singolo batch.

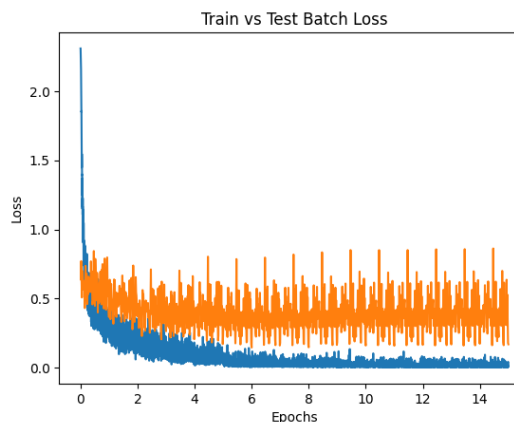


Fig. 1. Train vs Test loss

Andando nel dettaglio e osservando in figura 2 dove sono riportati i grafici che mostrano l'accuracy che cresce man mano che la loss diminuisce per training e testing, notiamo che nel training l'andamento è deciso e regolare e porta l'accuracy al 100% circa e la loss quasi a 0, invece nel testing si osserva che viene seguito un andamento irregolare in entrambi i grafici ovvero la loss nel testing è soggetto a molte oscillazioni la quale si ripercuote anche sull'accuracy, queste oscillazioni possono essere dovuto al problema nominato precedentemente, ovvero la presenza di qualche immagine che provoca dei valori anomali, cioè la sua classificarla in modo corretta è molto difficile. Infatti vediamo che in uno dei batch del test set la loss assume valori molto alti, quindi potrebbe essere che è presente un dato anomalo in quel batch.

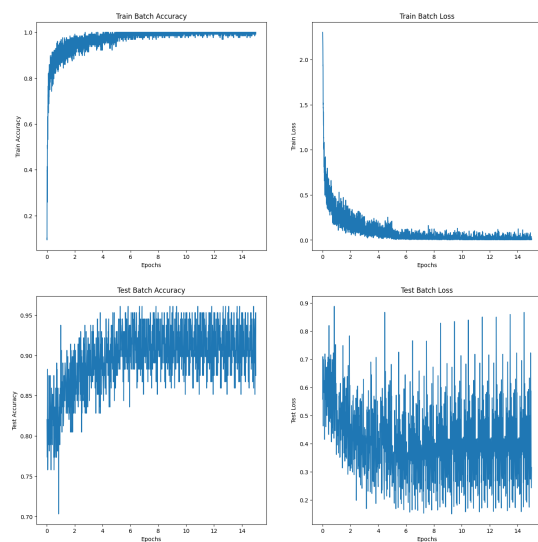


Fig. 2. Train vs Test