

Neural Networks and deep learning (Mod.B)

aa. 2023/2024 docente: R. Prevete

Progetto finale di esame

(Linguaggi di Programmazione possibili: Matlab, Python , C, C++,)

PARTE A (Comune a tutti).

- Progettazione ed implementazione di una libreria di funzioni per:
 - simulare la propagazione in avanti di una rete neurale multi-strato full-connected. Con tale libreria deve essere possibile implementare reti con più di uno strato di nodi interni e con qualsiasi funzione di attivazione per ciascun strato.
 - la realizzazione della back-propagation per reti neurali multi-strato, per qualunque scelta della funzione di attivazione dei nodi della rete e la possibilità di usare almeno la somma dei quadrati o la cross-entropy con e senza soft-max come funzione di errore.

PARTE B (Da scegliere).

Dato il dataset “minist” di immagini di cifre scritte a mano (<http://yann.lecun.com/exdb/mnist/>), scegliere uno dei seguenti punti

1. (*Difficoltà bassa*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training e test set (considerare almeno 10000 elementi per il training set e 2500 per il test set). Si fissi la classica discesa del gradiente come algoritmo di aggiornamento dei pesi. Si studi l'apprendimento di una rete neurale (ad esempio epoche necessarie per l'apprendimento, andamento dell'errore su training e validation set, accuratezza sul test) con uno solo strato di nodi interni **al variare della modalità di apprendimento: online, batch e mini-batch. Si faccia tale studio** per almeno 3 dimensioni diverse (numeri di nodi) per lo strato interno. Scegliere e mantenere invariati le funzioni di attivazione. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`)
2. (*Difficoltà bassa*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training e test set (considerare almeno 10000 elementi per il training set e 2500 per il test set). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si studi l'apprendimento di una rete neurale (ad esempio epoche necessarie per l'apprendimento,

andamento dell'errore su training e validation set, accuratezza sul test) con uno solo strato di nodi interni **al variare del numero di nodi interni** (scegliendo almeno 5 dimensioni diverse) e con funzione di errore cross-entropy più soft-max. Scegliere e mantenere invariati tutti gli altri “iper-parametri” come, ad esempio, le funzioni di attivazione ed i parametri della Rprop. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`)

3. (*Difficoltà bassa*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training e test set (Considerare almeno 10000 elementi per il training set e 2500 per il test set). Si fissi la discesa del gradiente con momento come algoritmo di aggiornamento dei pesi, si studi l'apprendimento di una rete neurale (ad esempio epoche necessarie per l'apprendimento, andamento dell'errore su training e validation set, accuratezza sul test) con un solo strato interno di neuroni al variare di eta (learning rate) e del momento per almeno 5 diverse dimensioni (numero di nodi) dello strato interno. Scegliere e mantenere invariati tutti gli altri “parametri” come, ad esempio, le funzioni di output. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`)
4. (*Difficoltà media*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie (considerare almeno 10000 elementi). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi, ed una rete neurale con un unico strato di nodi interni. Si scelgano gli iper-parametri del modello, cioè i parametri della Rprop (eta-positivo ed eta-negativo) ed il numero di nodi interni, sulla base di un approccio di cross-validation k-fold (ad esempio $k=10$). Scegliere e mantenere invariati tutti gli altri “parametri” come, ad esempio, le funzioni di attivazione. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`).
5. (*Difficoltà media*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training e test set (considerare almeno 10000 elementi per il training set e 2500 per il test set). Si fissi la resilient backpropagation

- (RProp) come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si studi l'apprendimento di una rete neurale (ad esempio epoche necessarie per l'apprendimento, andamento dell'errore su training e validation set, accuratezza sul test) facendo variare il numero di strati interni da 1 a 5 confrontando il caso in cui si utilizza come funzione di attivazione dei nodi la tangente iperbolica con i casi in cui si usa come funzione di attivazione dei nodi interni la ReLU e la leaky ReLU. Provare diverse scelte del numero dei nodi per gli strati interni. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`).
6. (*Difficoltà medio-alta*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training e test set (considerare almeno 10000 elementi per il training set e 2500 per il test set). Seguendo l'articolo "Empirical evaluation of the improved Rprop learning algorithms, Christian Igel, Michael Husken, neurocomputing, 2003" si confronti la classica resilient backpropagation (RProp) con almeno 2 varianti proposte nell'articolo, come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si fissi la funzione di attivazione ed il numero di nodi interni (almeno tre diverse dimensioni) e si confrontino i risultati ottenuti con i diversi algoritmi di apprendimento. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`).
 7. (*Difficoltà medio-alta*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie (almeno $N=10000$). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi, ed una rete neurale con un unico strato di nodi interni. Si scelgano gli iper-parametri del modello, cioè i parametri della Rprop (eta-positivo ed eta-negativo) ed il numero di nodi interni, sulla base di un approccio di cross-validation k-fold (ad esempio $k=10$). **Per la ricerca degli iper-parametri confrontare l'approccio classico "a griglia" con quello "random"** (*J Bergstra, Y Bengio, Random search for hyper-parameter optimization, 2012*). Scegliere e mantenere invariati tutti gli altri "parametri" come, ad esempio, le funzioni di attivazione. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`).
 8. (*Difficoltà medio-alta*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un

dataset globale di N coppie, e lo si divida opportunamente in training e test set (considerare almeno 10000 elementi per il training set e 2500 per il test set). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si studi l'apprendimento di una rete neurale (ad esempio epoche necessarie per l'apprendimento, andamento dell'errore su training e validation set, accuratezza sul test) con un solo strato di nodi interni e con la sigmoide come funzione di attivazione dei nodi interni **al variare del criterio di early-stopping**. Facendo riferimento all'articolo "Early Stopping | but when? Lutz Prechelt, 1999", si considerino i due algoritmi di early-stopping GL e PQ con differenti valori del parametro *alfa*. Considerare reti con un diverso numero di nodi interni (**almeno 5 diverse dimensioni**). Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione *imresize*).

9. (*Difficoltà medio-alta*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training e test set (considerare almeno 10000 elementi per il training set e 2500 per il test set). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si consideri, inoltre, di costruire k autoencoder costituiti da un unico strato interno composto da m_h nodi, con $h=1,2,\dots,k$ e $k=5$. In questo modo abbiamo k encoder E_h , con $h=1,2,\dots,k$, che ci permettono di proiettare i dati \mathbf{x} in k rappresentazioni diverse. Allora per ciascun encoder E_h , si studi l'apprendimento di una rete neurale (ad esempio epoche necessarie per l'apprendimento, accuratezza sul test) con un solo strato di nodi interni e con la sigmoide come funzione di attivazione dei nodi interni **che riceve come input l'output del encoder E_h (\mathbf{x})**. Si confronti l'apprendimento di tali reti con quello di una rete che riceve, invece, i dati di input originali \mathbf{x} . Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione *imresize*).
10. (*Difficoltà alta*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training e test set (considerare almeno 10000 elementi per il training set e 2500 per il test set). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si modifichino, se è necessario, le funzioni realizzate nella parte A), in modo tale da poter implementare una rete neurale convoluzionale con almeno uno strato convolutivo come una rete feed-forward (seguire l'articolo *An Equivalence of Fully Connected Layer and*

Convolutional Layer, Wei Ma, Jun Lu, arXiv, 2017). Testare tale rete sul dataset mnist riportandone le accuratezze ottenute al variare nel numero di filtri considerati ed, eventualmente, del numero di strati convolutivi.

- **Per ogni traccia della parte B si possono formare gruppi di massimo 3 persone**
- **Ad ogni traccia della parte B è possibile associare al più 2 diversi gruppi.**

Linguaggio di programmazione consigliati: matlab, python, ma è possibile utilizzare altri linguaggi come C, C++.