Relazione Perceptron Votato - Gabriele Nannotti

L'esercizio ha lo scopo di riprodurre i dati sperimentali sulla accuratezza della classificazione lineare dei modelli voted e average in confronto al modello di classificazione lineare last emulando in parte quanto descritto nell'articolo (Freund & Schapire 1999).

A tale scopo si utilizza il dataset MNIST i cui esempi consistono in delle immagini (sotto forma di array di pixel) che raffigurano delle cifre.
Si vuole dunque costruire dei classificatori lineari capaci di distinguere le immagini nelle due seguenti classi binarie:

- Classe 1 se l'immagine raffigura uno 0 un 6 un 8 e un 9
- Classe -1 se l'immagine raffigura una qualsiasi altra cifra

Il dataset MNIST utilizzato è composto da 70000 esempi; i primi 60000 sono utilizzati per il training mentre gli ultimi 10000 sono utilizzati per il testing.

I classificatori lineari vengono prodotti utilizzando l'algoritmo voted-perceptron con il quale si produce una lista di classificatori lineari del tipo:

$$h(x) = w\,x + b$$

I passi utilizzati per la costruzione della lista di classificatori sono i seguenti:

- Si parte dal classificatore iniziale $h_0(x)$ con $w = 0$ e $b = 0$
- Si itera su un numero n di entries del dataset eseguendo:
  - Lettura di un esempio (x, y) dal dataset
  - Se $h_i(x)$ classifica bene l'esempio si aumenta di 1 i successi di $h_i(x)$ e si passa all'esempio successivo del dataset mantenendo lo stesso classificatore lineare.
  - Altrimenti si aggiunge un nuovo classificatore $h_{i+1}(x)$ alla lista, copiandone i valori di w e b da $h_i(x)$; si effettua dunque l'aggiornamento dei parametri di $h_{i+1}(x)$, sull'esempio sbagliato (x, y) , con l'aggiornamento (dove R è la massima norma delle istanze negli esempi del dataset):

    $$w = w + y\,x$$

    $$b = b + y\,R$$

    Quindi si continua ad iterare utilizzando $h_{i+1}(x)$

Costruita la lista di classificatori lineari (ottenuta dal training su n entries) la si utilizza per testare l'accuratezza della classificazione lineare con i metodi last, voted e average.

Per ogni entry del testset vengono calcolate 3 predizioni:

- Last: per calcolare la predizione si utilizza l'ultimo piano della lista

  $$h_{last}(x) = sign(w_{last} \cdot x + b_{last})$$

- Voted: per calcolare la predizione si utilizza tutta la lista

  $$h_{voted}(x) = sign\Big(\sum_{i}^{|listlength|} h_i.successes * sign(h_i(x))\Big)$$

- Averaged: per calcolare la predizione si utilizza tutta la lista

$$h_{average}(x) = sign(\sum_{i}^{|listlength|} h_i.successes * h_i(x))$$

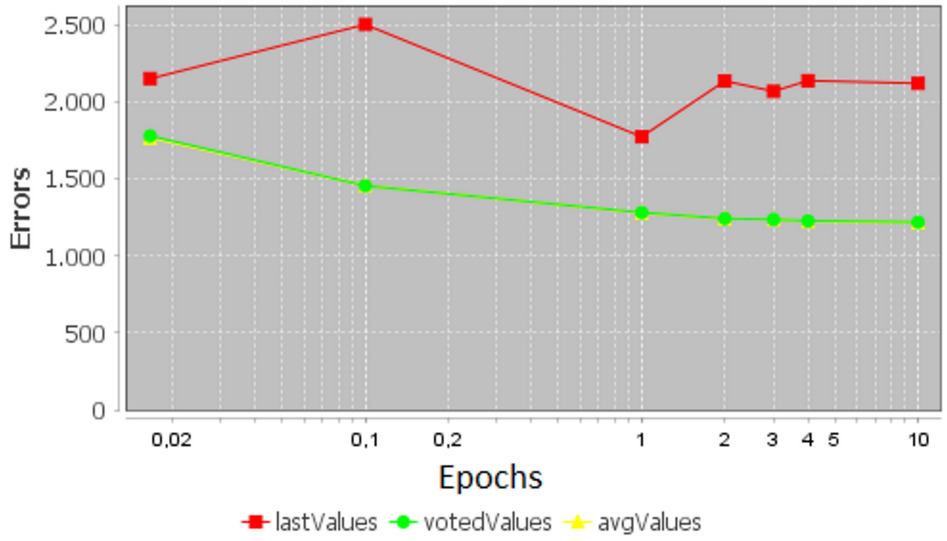
Terminata la fase di test si procede a stampare i dati dell'analisi e viene ripetuto il training della lista su ulteriori n entries del training set, alternando quindi una fase di training ad una di test.

In particolare i test vengono effettuati quando il numero di esempi su cui è stato fatto il training raggiunge i seguenti valori:
100, 6000, 60000, 120000, 180000, 240000, 600000

I valori considerati corrispondono, considerando che il trainset utilizzato contiene 60000 esempi, alle epoche:
0.002, 0.1, 1, 2, 3, 4, 10.

Per riprodurre più fedelmente possibile il metodo utilizzato nell'articolo (Freund & Schapire 1999) si esegue la procedura sopra delineata per 3 liste di piani distinte (modelli) costruite utilizzando diverse permutazioni del training set; i dati sperimentali finali sono ottenuti dalla media dei dati sperimentali ottenuti da ciascun modello.

I dati prodotti sono riassunti nel seguente grafico (i dati numerici sono infondo alla relazione):



Epoch: numero di epoche di addestramento dei modelli
Errors: numero di errori commessi sui 10000 esempi del testset in relazione alle epoche di addestramento.

Non essendo il dataset linearmente separabile si osserva che in effetti l'errore non converge a 0 per nessuna tipologia di classificatore.
Tuttavia è decisamente più netta la convergenza dell'accuratezza dei metodi average e voted rispetto al metodo last; essendo il testset composto da 10000 esempi si osserva dunque che l'accuratezza di voted e average si aggira attorno al 14% per convergere fino al 12% dopo circa 5 epoche di addestramento; il metodo last invece non presenta nessun netto miglioramento legato ad un addestramento più prolungato (mantenendo un'accuratezza attorno al 20% senza una convergenza netta).

Si può dunque dedurre sperimentalmente che i metodi average e voted sono una alternativa più accurata del metodo di classificazione last con lo svantaggio di un maggior costo in termini di tempo e memoria per essere eseguiti (dato che è necessario memorizzare e utilizzare tutti i classificatori lineari prodotti durante la fase di training, a differenza del metodo last che considera solo l'ultimo).

Descrizione del lavoro:
Per produrre l'analisi descritta è stato sviluppato un programma in java dove sono stati implementati le logiche di funzionamento dell'algoritmo, utilizzando programmazione a oggetti ed il supporto della libreria jfreechart e jcommon per la rappresentazione grafica.

Tutti i parametri di configurazione dell'algoritmo (tra cui le entries ogni quanto effettuare il test) sono stati inseriti in un file di configurazione del programma, in modo da permettere la modifica dei dati in input senza dover ricompilare l'intero programma.

Si fa riferimento al documento README per la specifica delle implementazioni effettuate nel programma sviluppato.

DATI SPERIMENTALI NUMERICI:
Descrizione:

TRAINING ERRORS: numero di errori commessi durante il training per ciascun modello e media di tali errori
LAST MODELS TEST ERRORS: numero di errori commessi durante il test su 10000 entries con classificatore last
VOTED MODELS TEST ERRORS: numero di errori commessi durante il test su 10000 entries con classificatore voted
AVERAGE MODELS TEST ERRORS: numero di errori commessi durante il test su 10000 entries con classificatore average

Tabella valori numerici:

Results after training over 1000 entries

TRAINING ERRORS
Model 0: 294 Model 1: 285 Model 2: 287
Average: 288.66666

LAST MODELS TEST ERRORS:
Model 0: 2125 Model 1: 2415 Model 2: 1907
Average: 2149.0

VOTED MODELS TEST ERRORS:
Model 0: 1710 Model 1: 1701 Model 2: 1921
Average: 1777.3334

AVERAGE MODELS TEST ERRORS:
Model 0: 1688 Model 1: 1706 Model 2: 1900
Average: 1764.6666

Results after training over 6000 entries

TRAINING ERRORS
Model 0: 1447 Model 1: 1403 Model 2: 1444
Average: 1431.3334

LAST MODELS TEST ERRORS:
Model 0: 2695 Model 1: 2980 Model 2: 1827
Average: 2500.6667

VOTED MODELS TEST ERRORS:
Model 0: 1448 Model 1: 1463 Model 2: 1451
Average: 1454.0

AVERAGE MODELS TEST ERRORS:
Model 0: 1450 Model 1: 1468 Model 2: 1450
Average: 1456.0

Results after training over 60000 entries

TRAINING ERRORS
Model 0: 12724 Model 1: 12599 Model 2: 12658
Average: 12660.333

LAST MODELS TEST ERRORS:
Model 0: 1896 Model 1: 1757 Model 2: 1663
Average: 1772.0

VOTED MODELS TEST ERRORS:
Model 0: 1290 Model 1: 1271 Model 2: 1279
Average: 1280.0

AVERAGE MODELS TEST ERRORS:
Model 0: 1284 Model 1: 1270 Model 2: 1279
Average: 1277.6666

Results after training over 120000 entries

TRAINING ERRORS
Model 0: 24700 Model 1: 24502 Model 2: 24565
Average: 24589.0

LAST MODELS TEST ERRORS:
Model 0: 2523 Model 1: 2170 Model 2: 1709
Average: 2134.0

VOTED MODELS TEST ERRORS:
Model 0: 1244 Model 1: 1241 Model 2: 1238
Average: 1241.0

AVERAGE MODELS TEST ERRORS:
Model 0: 1246 Model 1: 1237 Model 2: 1239
Average: 1240.6666

Results after training over 180000 entries

TRAINING ERRORS
Model 0: 36444 Model 1: 36325 Model 2: 36321
Average: 36363.332

LAST MODELS TEST ERRORS:
Model 0: 1722 Model 1: 2543 Model 2: 1936
Average: 2067.0

VOTED MODELS TEST ERRORS:
Model 0: 1235 Model 1: 1233 Model 2: 1237
Average: 1235.0

AVERAGE MODELS TEST ERRORS:
Model 0: 1235 Model 1: 1232 Model 2: 1240
Average: 1235.6666

Results after training over 240000 entries

TRAINING ERRORS
Model 0: 48095 Model 1: 47931 Model 2: 48022
Average: 48016.0

LAST MODELS TEST ERRORS:
Model 0: 2088 Model 1: 2515 Model 2: 1805
Average: 2136.0

VOTED MODELS TEST ERRORS:
Model 0: 1225 Model 1: 1230 Model 2: 1223
Average: 1226.0

AVERAGE MODELS TEST ERRORS:
Model 0: 1222 Model 1: 1222 Model 2: 1225
Average: 1223.0

Results after training over 600000 entries

TRAINING ERRORS
Model 0: 118029 Model 1: 117259 Model 2: 117735
Average: 117674.336

LAST MODELS TEST ERRORS:
Model 0: 2179 Model 1: 2460 Model 2: 1717
Average: 2118.6667

VOTED MODELS TEST ERRORS:
Model 0: 1217 Model 1: 1212 Model 2: 1224
Average: 1217.6666

AVERAGE MODELS TEST ERRORS:
Model 0: 1213 Model 1: 1212 Model 2: 1221
Average: 1215.3334