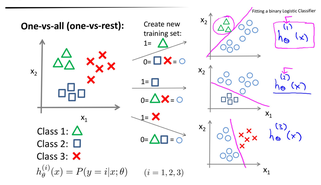
MULTI-CLASS CLASSIFICATION



ONE VS ALL: FAI IL TRAINING DEL CLASSIFICATORE BINARIO $h_\theta^{(k)}(x)$ PER OGNI CLASSE k

AL FINE DI PREDIRE LA PROBABILITA' $P(y=k|x;\theta)$

QUANDO SI DEVE CLASSIFICARE UN NUOVO ESEMPIO x^{NEW} BISOGNERA' PRENDERE LA CLASSE CORRISPONDENTE AL CLASSIFICATORE $h_\theta^{(k)}$ CHE HA PROBABILITA' MAGGIORE

$$\hat{k} = \max_k h_\theta^{(k)}(x^{NEW})$$

ESEMPIO

$h_\theta^{(1)}(x^{NEW}) = 0.3$

$h_\theta^{(2)}(x^{NEW}) = 0.8 \Rightarrow$ CLASSE 2

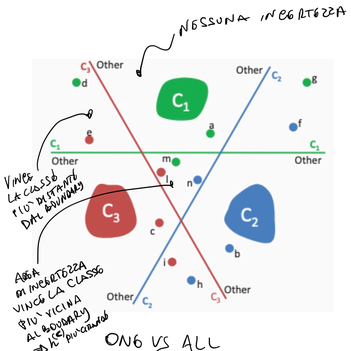
$h_\theta^{(3)}(x^{NEW}) = 0.1$

N.B.: $\sum_c h_\theta^{(c)}(x^{NEW})$ PUO' ASSUMERE VALORI MAGGIORI DI 1 (I 3 OUTPUT NON FORMANO UNA DISTRIBUZIONE DI PROBABILITA')

E se tutti e tre i classificatori avessero un valore di output minore di 0,5?

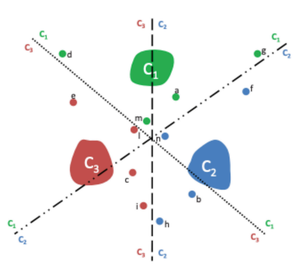
E' corretto classificare prendendo in considerazione il classificatore che ha il valore massimo sebbene minore di 0,5? (es. se il classificatore 1 ha il valore massimo tra i tre classificatori e il suo valore è pari a 0.4, valore che per quanto detto indica che è più probabile che il pattern non sia di classe 1 ma di una delle altre due classi, è corretto classificarlo come di classe 1?)

NESSUNA INCERTEZZA



VINCE LA CLASSE PIU' DISTANTE DAL BOUNDARY

AREA DI INCERTEZZA VINCE LA CLASSE PIU' VICINA AL BOUNDARY $\Rightarrow h^{(2)}$ MAGGIORE



CON K CLASSI $\frac{K(K-1)}{2}$ CLASSIFICATORI

ONE VS ALL

DISTANZA DAL BOUNDARY

$h^{(1)}(m) < 0.5$

$h^{(3)}(m) < 0.5$

$h^{(2)}(m) < 0.5$

$h^{(2)}(m) < h^{(3)}(m) < h^{(1)}(m)$

$\Rightarrow$ CLASSE C_1

ESEMPIO

m ●

l ●

n ●

ONE VS ONE

C_1 VS C_2 VOTA PER C_1

C_3 V C_2 VOTA PER C_3

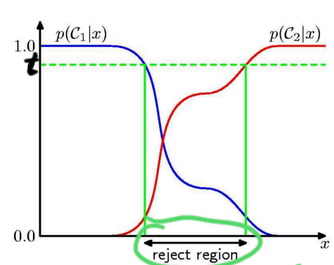
C_1 V C_3 VOTA PER C_1

C_1	C_2	C_3
2	0	1

2 VOTI PER $C_1 \Rightarrow$ CLASSE C_1

ANCHE IN QUESTO CASO SI PUO' RAGIONARE IN TERMINI DI DISTANZA DAL BOUNDARY PERO' TORNANO IN CONSIDERAZIONE LE CLASSI C_1 & C_3

Binary Classification Rejection Option



VALORI DI INPUT PER CUI NON DECIDIAMO

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{d}\theta_j^2$$

NELL'ALGORITMO DI DISCESA DEL GRADIENTE SI AGGIUNGE LA DERIVATA DEL TERMINE DI REGOLARIZZAZIONE PER TUTTI I θ_j $\forall j = 1, \dots, d$

REPEAT {

$$\theta_0^{new} \leftarrow \theta_0^{old} - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_0$$

$$\theta_j^{new} \leftarrow \theta_j^{old} - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_j + \frac{\lambda}{m}\theta_j\right] \quad \forall j = 1, \dots, n$$

$\hookrightarrow \delta(\theta^T x^{(i)})$ }

FORMA USATA X LA CLASSIFICAZIONE

$J(\theta) \leftarrow$ FUNZIONE OTTIMIZZAZIONE PARAMETRI DE GOLARIZZATA (CALCOLATA SUL TRAINING)

$J_{TRAIN} \Rightarrow$ UGUALE A $J(\theta)$ MA NON INCLUDE TERMINE DI REGOLARIZZAZIONE (CALCOLATA SUL TRAINING SET)

$J_{VAL} \Rightarrow$ UGUALE A $J(\theta)$ MA NON INCLUDE TERMINE DI REGOLARIZZAZIONE (CALCOLATA SUL VALIDATION SET)

$J_{TEST} \Rightarrow$ CALCOLATA SUL TEST SET. UGUALE A $J(\theta)$ MA NON CONSIDERA TERMINE DI REGOLARIZZAZIONE

$$J_{ERROR} = \frac{1}{m}\sum_{i=1}^{m_{test}} ERROR(h_\theta(x^{(i)}), y^{(i)})$$

MISCLASSIFICATION ERROR $\equiv$ FRAZIONE DI PATTERN ERRONEAMENTE CLASSIFICATI

$$ERROR(h_\theta(x^{(i)}), y^{(i)}) = \begin{cases} 1 & \text{IF } \left((h_\theta(x^{(i)}) \geq 0.5) \text{ AND } (y^{(i)} = 0)\right) \text{ OR } \left((h_\theta(x^{(i)}) < 0.5) \text{ AND } (y^{(i)} = 1)\right) \\ 0 & \text{IF } \left((h_\theta(x^{(i)}) \geq 0.5) \text{ AND } (y^{(i)} = 1)\right) \text{ OR } \left((h_\theta(x^{(i)}) < 0.5) \text{ AND } (y^{(i)} = 0)\right) \end{cases}$$

STOCHASTIC GRADIENT DESCENT (SGD)

OSSERVAZIONE: LA DISCESA DEL GRADIENTE COPRIVA $J(\theta)$ SU TUTTI I CAMPIONI DEL TRAINING SET.

LA SOMMA DEI RESIDUI CONTENEVA m TERMINI AD OGNI ITERAZIONE DELLA DISCESA DEL GRADIENTE CHE VENIVA MOLTIPLICATA d VOLTE PER I VALORI DEGLI COEFFICIENTI

AL CRESCERE DI m E DI d IL PROBLEMA ESCONDE COMPUTAZIONALMENTE

1000 CAMPIONI = m > $m \times d$ OPERAZIONI AD OGNI ITERAZIONE

100 FEATURE = d

$\Downarrow$

1000000 PER OGNI ITERAZIONE

m ITERAZIONI GRADIENT DESCENT

$m \times 1000000$

SE m VALE ANCHE SOLO 1000

$\Downarrow$

1000000000 OPERAZIONI

$\Rightarrow$ STOCASTIC IS SUPER HERO!

STOCASTIC = RANDOM

1) CAMPIONA RANDOM UN SET DI PCCM PUNTI DAL TRAINING (UN MINI BATCH)

2) ESEGUI L'ITERAZIONE DEL GRADIENT DESCENT DI DISCESA DEL GRADIENTE SUI PUNTI CAMPIONATI

3) TORNA AL PUNTO 1 SE NON SEI ARRIVATO A CONVERGENZA

MOMENTUM

ALCUNE FUNZIONI POSSONO ESSERE DIFFICILI DA OTTIMIZZARE PER CUI SGD HA DIFFICOLTA' A TROVARE IL MINIMO LINK https://distill.pub/2017/momentum/

AGGIORNAMENTO CON MOMENTUM

$$\Delta\theta^{new} \leftarrow \theta^{old} - \alpha\, \Delta\theta^{old}$$

CON $\Delta\theta^{new} \leftarrow \beta\, \Delta\theta^{old} + \frac{\partial J(\theta^{old})}{\partial \theta^{old}}$

$\hookrightarrow$ MOMENTUM PARAMETER

$\beta = 0 \Rightarrow$ CLASSIC GRADIENT DESCENT

$\Rightarrow$ ONE MORE PARAMETER TO SET!

VALUTAZIONE CLASSIFICAZIONE BINARIA

MATRICE DI CONFUSIONE

		ETICHETTE PREDETTE	
		POSITIVE	NEGATIVE
ETICHETTE VERE	POSITIVE	a	c
	NEGATIVE	d	b

a = NUMERO DEI CAMPIONI DI CLASSE POSITIVA CLASSIFICATI COME POSITIVI DAL NOSTRO ALGORITMO $\Rightarrow$ VERI POSITIVI $\Rightarrow TP = \frac{a}{a+c}$

b = NUMERO DI CAMPIONI DI CLASSE NEGATIVA CLASSIFICATI CORRETTAMENTE COME NEGATIVI DALL'ALGORITMO $\Rightarrow$ VERI NEGATIVI $\Rightarrow TN = \frac{b}{b+d}$

c = NUMERO DI CAMPIONI POSITIVI CLASSIFICATI ERRONEAMENTE COME NEGATIVI DAL MODELLO $\Rightarrow$ FALSI NEGATIVI $\Rightarrow FN = \frac{c}{c+a}$

d = NUMERO DI CAMPIONI NEGATIVI CLASSIFICATI ERRONEAMENTE COME POSITIVI DAL MODELLO $\Rightarrow$ FALSI POSITIVI $\Rightarrow FP = \frac{d}{d+b}$

IN PERCENTUALE

$a + c$ = TOTALE DEI CAMPIONI POSITIVI NELLA BANCA DATI DI TEST

$d + b$ = TOTALE DEI CAMPIONI NEGATIVI NELLA BANCA DATI DI TEST

$TP + FN = 1$

$TN + FP = 1$

ACCURACY

$\frac{a+b}{a+b+c+d}$ PERCENTUALE DEI CAMPIONI CORRETTAMENTE CLASSIFICATI

ATTENZIONE AD AVERE SET DI TEST BILANCIATI NEL NUMERO DI CAMPIONI DELLE DIVERSE CLASSI (NON E' SEMPRE POSSIBILE, MA BISOGNA TENERNE CONTO IN FASE DI VALUTAZIONE) MA PERCHE'?

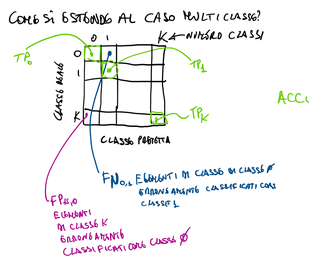
PRECISION

$\frac{TRUE\ POSITIVE\ a}{TRUE\ POSITIVE\ a + FALSE\ POSITIVE\ d}$ DI TUTTI I CASI CHE SONO STATI PREDETTI COME POSITIVI, QUALE FRAZIONE SONO EFFETTIVAMENTE POSITIVI? WE WANT LOW FALSE POSITIVE

RECALL

$\frac{TRUE\ POSITIVES\ a}{TRUE\ POSITIVES\ a + FALSE\ NEGATIVE\ c}$ DI TUTTI I CASI POSITIVI QUANTE FRAZIONE ABBIAMO CORRETTAMENTE ETICHETTATO COME POSITIVI? WE WANT LOW FALSE NEGATIVE

COME SI ESTENDE AL CASO MULTI CLASSE?



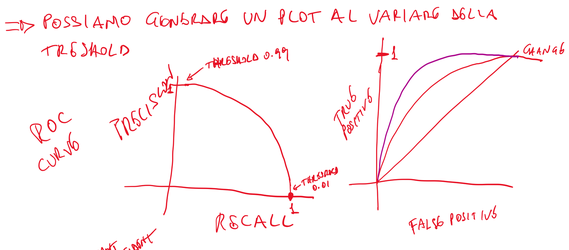
ACCURACY $\frac{\sum_i M_{ii}}{\sum_i \sum_j M_{ij}}$

NB: WE USE $h_\theta(x) \geq 0.5$ TO CLASSIFY

$\hookrightarrow$ THRESHOLD $\Rightarrow$ PRECISION AND RECALL WHEN WE USE THRESHOLD 0.5

SE CAMBIAMO LA TRESHOLD OTTENIAMO ALTRI VALORI DI PRECISION E RECALL

$\Rightarrow$ POSSIAMO GENERARE UN PLOT AL VARIARE DELLA TRESHOLD



WE WANT TO BE CONFIDENT $\Rightarrow$ HIGHER PRECISION, LOWER RECALL

WE WANT TO AVOID MANY CASES OF CANCER (AVOID FALSE NEGATIVE) $\Rightarrow$ HIGHER RECALL, LOWER PRECISION

IN RECALL THE DENOMINATOR DOES NOT DEPEND FROM TRESHOLD. LOWERING THRESHOLD INCREASE RECALL (INCREASE TP) LOWERING THE THRESHOLD MAY INCREASE THE DENOMINATOR OF PRECISION (INCREASE THE NUMBER OF POSITIVE a AND FALSE POSITIVE d)

IF THRESHOLD IS TOO HIGH RESULT MAY BE ALL TRUE POSITIVE WHICH INCREASE PRECISION

F1 SCORE

$$F_1 = 2\frac{PRECISION \cdot RECALL}{PRECISION + RECALL}$$

FUNZIONA RACCOLTO DI CONSIDERARE MEDIA DI PRECISION E RECALL

	PRECISION	RECALL	AVERAGE	F_1
MODEL 1	0.5	0.4	0.45	0.004
MODEL 2	0.7	0.1	0.4	0.175
MODEL 3	0.02	1	0.51	0.039

Predice sempre 1

$P = 0$ OPPURE $R = 0$

$\Downarrow$

$F_1 = 0$

$P = 1$ E $R = 1$

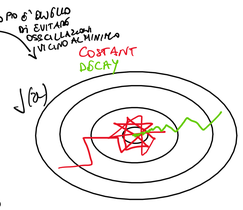
$\Downarrow$

$F_1 = 1$

LEARNING RATE DECAY

- DURANTE LE FASI DI TRAINING E' SPESSO UTILE RIDURRE IL LEARNING RATE

LO SCOPO E' QUELLO DI EVITARE OSCILLAZIONI VICINO AL MINIMO



- "LEARNING RATE SCHEDULES": AGGIUSTA IL LEARNING RATE DURANTE LA FASE DI TRAINING TENENDO IN CONSIDERAZIONE UNO SCHEMA PRE-DEFINITO:

- MINIBATCH "GRANDI" PERMETTONO DI STIMARE MEGLIO IL GRADIENTE (PIU' E' GRANDE E PIU' SI AVVICINA AL GRADIENTE DI $J(\theta)$ SULL'INTERO SET DI DATI)
- PIU' GRANDE E' IL BATCH PIU' GRANDE E' DIFFICILE POI SCEGLIE LO STEP DA FARE NEL GRADIENTE (OVVERO α)

$b * K \rightarrow \alpha * K$ LINEAR SCALING

$b * K \rightarrow \alpha * \sqrt{K}$ ALTERNATIVE PER LA SCALING

$\frac{\alpha}{b}$ PIU' GRANDE E' QUESTO RAPPORTO E MIGLIORE LA CONDIZIONE PER RAGGIUNGERE MINIMO

OSSERVAZIONI

b = # CAMPIONI NEL MINI BATCH (BATCH SIZE)

α = LEARNING RATE

e = # EPOCHE (ITERAZIONI SGD) AD OGNI EPOCA AGGIORNIAMO I PESI

m = # CAMPIONI

$\frac{m}{b}$ = NUMERO DI BATCH DA CONSIDERARE PER OGNI EPOCA (AGGIORNO $\frac{m}{b}$ VOLTE I PESI) $\Rightarrow \frac{m}{b}$ ITERAZIONI PER EPOCA

$\Downarrow$

$e \cdot \frac{m}{b}$ ITERAZIONI

- "STEP-DECAY": IL LEARNING RATE SI RIDUCE DI UN FATTORE COSTANTE OGNI K EPOCHE

$$\alpha^{new} = \alpha^0 \cdot \frac{1}{2^d} \quad \text{DOVE } d = \left\lfloor \frac{EPOCH}{K} \right\rfloor$$

EPOCHE = COMPLETO PASSAGGIO DEI DATI IN UNA FASE DI TRAINING

SOLITAMENTE SI SCEGLIE DI RIDURRE IL LEARNING RATE DELLA META' OGNI $K = 10$ EPOCHE

- EXPONENTIAL DECAY

$$\alpha^t = \alpha^0 \cdot e^{-Kt}$$

K = PARAMETRO; t = NUMERO DI ITERAZIONI DI TRAINING

- ADAPTIVE LEARNING RATE METHODS

IL PROBLEMA PRINCIPALE UTILIZZANDO UNO SCHEDULER PER IL LEARNING RATE E' LA DEFINIZIONE DEGLI IPERPARAMETRI DELLO SCHEDULER. COINVOLTI NELLA FORMULAZIONE INFATTI QUESTI DIPENDONO DAL PROBLEMA DA RISOLVERE, DAI DATI E DAL MODELLO.

ESISTONO ALGORITMI DETTI DI "ADAPTIVE GRADIENT DESCENT" CHE POSSONO AIUTARE A RISOLVERE QUESTO PROBLEMA

- ADAGRAD: GRANDI UPDATE PER PARAMETRI SPARSI E PICCOLI UPDATE PER PARAMETRI NON SPARSI (SPARSI = VALORI IN INPUT DELLA COMPONENTE PARI A 0 MOLTO SPESSO)
- ADAM: AGGIUSTA ADAGRAD (MEDIA AGGIORNAMENTI), USA MOMENTUM E LEARNING RATE DIFFERENTE PER OGNUNO DEI PESI

- MORE SOPHISTICATED APPROACH EXIST FOR A BETTER TRAINING OF MORE COMPLEX MODELS (E.G. DEEP LEARNING MODELS). ONE OF THEM IS LARS (LAYER-WISE ADAPTIVE RATE SCALING) WHICH USE SEPARATE LEARNING RATE FOR EACH LAYER