INTRODUZIONE

È UN AMPIA GAMMA DI TECNICHE PER L'APPRENDIMENTO AUTOMATICO IN CUI LE IPOTESI ASSUMONO LA FORMA DI COMPLESSI CIRCUITI ALGEBRICI.
DEEP XCHÉ I CIRCUITI SONO GENERALMENTE STRUTTURATI IN MOLTEPLICI STRATI.

LE RETI ADDESTRATE TRAMITE METODI DI DEEP LEARNING VENGONO CHIAMATE RETI NEURALI.

OGNI NEURONE RICEVE INPUT DA ALTRI NEURONI O DALL'ESTERNO.
OGNI INPUT x_i È ASSOCIATO AD UN PESO w_i. OGNI NEURONE HA POI UN VALORE CHIAMATO BIAS.
OGNI NEURONE CALCOLA:

$$\left(\sum_{i=1}^{m} w_i x_i\right) + b$$

IL VALORE CALCOLATO VIENE POI PASSATO AD UNA FUNZIONE DI ATTIVAZIONE.
IL VALORE FINALE DEL NEURONE PASSA ALLO STRATO SUCCESSIVO.
INIZIALMENTE TUTTI I PESI E BIAS SONO INIZIALIZZATI RANDOMICAMENTE.

BACK PROPAGATION

ALGORITMO UTILIZZATO PER ADDESTRARE LE RETI NEURALI CONSENTENDO DI OTTIMIZZARE I PESI W E I BIAS b.
VISTO CHE W e b INIZIALMENTE SONO INIZIALIZZATI RANDOMICAMENTE:

INIZIALMENTE L'OUTPUT SARÀ SBAGLIATO.
UTILIZZO UNA FUNZIONE DI COSTO PER CALCOLARE QUANTO IL MODELLO SI DISCOSTA DALL'OUTPUT ATTESO.
VOGLIO MINIMIZZARE QUESTA FUNZIONE.

VOGLIO MINIMIZZARE QUESTA DISTANZA

LA SOMMA SARÀ MINORE QUANDO CLASSIFICHIAMO CORRETTAMENTE GLI OUTPUT, POICHÉ AVREMO DEI VALORI PIÙ PICCOLI NELLE DIFFERENZE *

UTILIZZANDO L'ALGORITMO DELLA DISCESA DEL GRADIENTE VOGLIO QUINDI TROVARE DEI PESI & BIAS $\begin{bmatrix} 0.16 \\ 0.72 \\ \vdots \\ +0.53 \\ 4.64 \end{bmatrix}$ CHE MINIMIZZINO LA DISTANZA SOPRACITATA.

UNA VOLTA FORNITO L'OUTPUT VIENE CALCOLATO IL GRADIENTE DELL'ERRORE PER LO STRATO FINALE.
PER OGNI STRATO NASCOSTO L'ERRORE VIENE CALCOLATO IN BASE ALL'ERRORE DELLO STRATO SUCCESSIVO, PROPAGATO INDIETRO ATTRAVERSO I PESI DELLA RETE.
CALCOLATI I GRADIENTI PER OGNI STRATO VENGONO AGGIORNATI I PESI & BIAS USANDO LA FORMULA DELLA DISCESA DEL GRADIENTE.

EPOCH

UN EPOCH CONSISTE IN UN PASSAGGIO IN AVANTI E UNO IN BACKWARD DI TUTTO IL DATASET.

PER OGNI EPOCH I PESI & BIAS VENGONO AGGIUSTATI.

SE AD ESEMPIO ABBIAMO EPOCH = 5, TUTTI I NOSTRI ESEMPI VENGONO VISTI PER 5 VOLTE.

IMPORTANTE: DOPO OGNI EPOCH IL MODELLO VIENE VALUTATO SU UN SET DI VALIDAZIONE.

BATCH & BATCH SIZE

QUANDO ABBIAMO UN DATASET TROPPO GRANDE DA FAR ELABORARE AL NOSTRO MODELLO IN UNA SOLA VOLTA POSSIAMO SUDDIVIDERLO IN BATCH.
IL BATCH SIZE È UN IPER PARAMETRO CHE DEFINISCE IL NUMERO DI ESEMPI DA ELABORARE PRIMA DI AGGIORNARE I SUOI PARAMETRI INTERNI (W & b).
CON UN BATCH SIZE DI 500 E UN DATASET DI 1000 ESEMPI AVREMO 2 BATCH.

$$\text{NUMERO DI BATCH PER EPOCH} = \frac{\text{\# DI ESEMPI}}{\text{BATCH SIZE}} = \frac{1000}{500} = 2$$

IN OGNI EPOCH AVREMO 2 BATCH DA 500 ESEMPI CIASCUNO.

UN ITERATION RAPPRESENTA UN SINGOLO PASSAGGIO IN CUI IL MODELLO ELABORA UN BATCH DI DATI E AGGIORNA I SUOI PESI.

LEARNING RATE

IL LEARNING RATE (TASSO DI APPRENDIMENTO) È UN IPERPARAMETRO CHE CONTROLLA LA VELOCITÀ CON CUI IL MODELLO AGGIORNA I SUOI PESI DURANTE IL TRAINING.

COME ABBIAMO GIA VISTO DURANTE L'ADDESTRAMENTO IL MODELLO USA UN ALGORITMO (es. DISCESA DEL GRAD.) PER MINIMIZZARE LA FUNZIONE DI PERDITA CHE MISURA QUANTO LE PREDIZIONI SONO LONTANE DALLE ETICHETTE CORRETTE.
IL LEARNING RATE STABILISCE DI QUANTO I PESI VENGONO AGGIORNATI OGNI VOLTA CHE VIENE CALCOLATO UN GRADIENTE.

FORMULA PER L'AGGIORNAMENTO DEI PESI:

$$w = w - \eta \times \nabla L(w)$$

- w È IL PESO
- η È IL LEARNING RATE
- $\nabla L(w)$ GRADIENTE DELLA FUNZ. DI PERDITA DI w.

LEARNING RATE TROPPO ALTO ↑ ⟶ OGNI VOLTA CHE CALCOLIAMO L'ERRORE E AGGIORNIAMO I PESI, FACCIAMO UN PASSO REALMENTE GRANDE, QUESTO POTREBBE FARCI SALTARE OLTRE IL MINIMO DELLA LOSS FUNCTION.

LEARNING RATE TROPPO BASSO ↓ ⟶ OGNI AGG. DEI PESI SARÀ PIÙ PRECISO MA MOLTO LENTO. IL MODELLO AVRÀ BISOGNO DI PIÙ AGGIORNAMENTI (ITERAZIONI) PER TROVARE LA SOLUZIONE OTTIMALE.