Implementazione di una Rete Convoluzionale in CUDA

Michele Valsesia

Nicholas Aspes

Anno accademico 2018/2019

Introduzione
Obiettivi

- Descrivere brevemente l'architettura ed il funzionamento di una *Rete Neurale*

Introduzione

Obiettivi

- Descrivere brevemente l'architettura ed il funzionamento di una *Rete Neurale*
- Motivare le differenti scelte implementative adottate durante lo svolgimento del progetto

Introduzione

Obiettivi

- Descrivere brevemente l'architettura ed il funzionamento di una *Rete Neurale*
- Motivare le differenti scelte implementative adottate durante lo svolgimento del progetto
- Valutare l'accuratezza e lo speed-up della rete rispetto ad una sua implementazione sequenziale

Reti Neurali

Reti Neurali

Scopo

- Le *Reti Neurali* vengono principalmente usate per la classificazione delle immagini

Reti Neurali

Scopo

- Le *Reti Neurali* vengono principalmente usate per la classificazione delle immagini
- Il processo di classificazione consiste nell'associare ad un'immagine un'etichetta che identifica nel miglior modo possibile il suo contenuto semantico

Reti Neurali

Scopo

- Le *Reti Neurali* vengono principalmente usate per la classificazione delle immagini
- Il processo di classificazione consiste nell'associare ad un'immagine un'etichetta che identifica nel miglior modo possibile il suo contenuto semantico
- Una *classe* non è altro che l'etichetta di un'immagine

Reti Neurali

Scopo

- Le *Reti Neurali* vengono principalmente usate per la classificazione delle immagini
- Il processo di classificazione consiste nell'associare ad un'immagine un'etichetta che identifica nel miglior modo possibile il suo contenuto semantico
- Una *classe* non è altro che l'etichetta di un'immagine
- Le reti neurali ricevono in input un'immagine e forniscono in output la relativa classe

Reti Neurali
Apprendimento

- Per poter classificare, una rete neurale deve *imparare* ad associare correttamente le immagini alle varie classi

Reti Neurali
Apprendimento

- Per poter classificare, una rete neurale deve *imparare* ad associare correttamente le immagini alle varie classi

- Il *training set* ed il *test set* sono due insiemi composti da coppie (immagini, etichette) chiamate *esempi*

Reti Neurali

Apprendimento

- Per poter classificare, una rete neurale deve *imparare* ad associare correttamente le immagini alle varie classi

- Il *training set* ed il *test set* sono due insiemi composti da coppie (immagini, etichette) chiamate *esempi*

- Le etichette di ciascun esempio vengono assegnate in maniera soggettiva da personale umano

Reti Neurali
Training Set

- Il training set viene usato durante la fase di apprendimento della rete

Reti Neurali

Training Set

- Il training set viene usato durante la fase di apprendimento della rete
- Per ognuno degli esempi del training set

Reti Neurali

Training Set

- Il training set viene usato durante la fase di apprendimento della rete
- Per ognuno degli esempi del training set
  - La rete riceve in input l'immagine dell'esempio considerato e l'associa ad una delle classi presenti

Reti Neurali

Training Set

- Il training set viene usato durante la fase di apprendimento della rete
- Per ognuno degli esempi del training set
  - La rete riceve in input l'immagine dell'esempio considerato e l'associa ad una delle classi presenti
  - Se la classe di output non corrisponde all'etichetta dell'esempio, la rete corregge i suoi parametri interni e passa all'immagine successiva

Reti Neurali

Test Set

- Il test set verifica che la rete abbia imparato a discriminare correttamente le immagini

Reti Neurali

Test Set

- Il test set verifica che la rete abbia imparato a discriminare correttamente le immagini

- Viene valutata l'*accuratezza* della rete come il rapporto tra il numero di esempi classificati scorrettamente ed il numero totale di esempi

Reti Neurali

Test Set

- Il test set verifica che la rete abbia imparato a discriminare correttamente le immagini
- Viene valutata l'*accuratezza* della rete come il rapporto tra il numero di esempi classificati scorrettamente ed il numero totale di esempi
- Per ognuno degli esempi del test set

Reti Neurali

Test Set

- Il test set verifica che la rete abbia imparato a discriminare correttamente le immagini
- Viene valutata l'*accuratezza* della rete come il rapporto tra il numero di esempi classificati scorrettamente ed il numero totale di esempi
- Per ognuno degli esempi del test set
  - La rete riceve in input l'immagine dell'esempio considerato e l'associa ad una delle classi presenti

Reti Neurali

Test Set

- Il test set verifica che la rete abbia imparato a discriminare correttamente le immagini

- Viene valutata l'*accuratezza* della rete come il rapporto tra il numero di esempi classificati scorrettamente ed il numero totale di esempi

- Per ognuno degli esempi del test set
  - La rete riceve in input l'immagine dell'esempio considerato e l'associa ad una delle classi presenti
  - Ogni volta che l'output della rete non corrisponde all'etichetta dell'esempio, viene incrementato un contatore, necessario al calcolo dell'accuratezza

Reti Neurali
Significato Biologico

- Le *Reti Neurali* nascono con lo scopo di modellare una rete neurale biologica

Reti Neurali

Significato Biologico

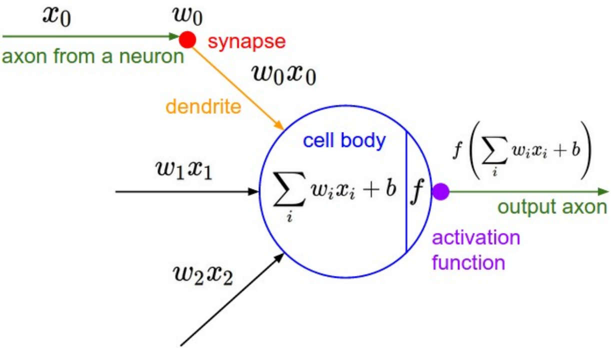
- Le *Reti Neurali* nascono con lo scopo di modellare una rete neurale biologica
- Una rete neurale biologica si compone di unità cellulari di base: i *neuroni*

Reti Neurali
Significato Biologico

- Le *Reti Neurali* nascono con lo scopo di modellare una rete neurale biologica
- Una rete neurale biologica si compone di unità cellulari di base: i *neuroni*
- I neuroni sono collegati tra loro per mezzo di specifiche giunture chiamate *sinapsi*

Reti Neurali

Neurone



Modello matematico di un neurone

Reti Neurali

Funzionamento Neurone

- Attraverso un meccanismo di eccitazione ed inibizione i pesi sinaptici controllano quanto un neurone venga influenzato dagli altri

- Attraverso un meccanismo di eccitazione ed inibizione i pesi sinaptici controllano quanto un neurone venga influenzato dagli altri

- I segnali pesati dalle differenti sinapsi vengono trasportati dai dendriti all'interno del neurone e sommati tra loro

Reti Neurali

Funzionamento Neurone

- Attraverso un meccanismo di eccitazione ed inibizione i pesi sinaptici controllano quanto un neurone venga influenzato dagli altri
- I segnali pesati dalle differenti sinapsi vengono trasportati dai dendriti all'interno del neurone e sommati tra loro
- Se la somma supera una certa soglia, il neurone *spara* un segnale lungo l'assone

Reti Neurali

Funzionamento Neurone

- Attraverso un meccanismo di eccitazione ed inibizione i pesi sinaptici controllano quanto un neurone venga influenzato dagli altri
- I segnali pesati dalle differenti sinapsi vengono trasportati dai dendriti all'interno del neurone e sommati tra loro
- Se la somma supera una certa soglia, il neurone *spara* un segnale lungo l'assone
- La *frequenza di sparo* del neurone viene modellata con una funzione di attivazione f

Definizione

Una *funzione di attivazione* è una funzione matematica non lineare usata per calcolare l'output di un neurone. Riceve come input la somma pesata dei segnali in ingresso al neurone

Definizione

Una *funzione di attivazione* è una funzione matematica non lineare usata per calcolare l'output di un neurone. Riceve come input la somma pesata dei segnali in ingresso al neurone

- *Sigmoide*

Reti Neurali

Funzioni di Attivazione

Definizione

Una *funzione di attivazione* è una funzione matematica non lineare usata per calcolare l'output di un neurone. Riceve come input la somma pesata dei segnali in ingresso al neurone

- *Sigmoide*
- *Tangente Iperbolica*

Definizione

Una *funzione di attivazione* è una funzione matematica non lineare usata per calcolare l'output di un neurone. Riceve come input la somma pesata dei segnali in ingresso al neurone

- *Sigmoide*
- *Tangente Iperbolica*
- *Softplus*

Reti Neurali

Sigmoide

Definizione

La *Sigmoide* $\sigma : \mathbb{R} \to [0, 1]$ è definita come $\sigma(x) = \frac{1}{(1+e^{-x})}$

Definizione

La *Sigmoide* $\sigma : \mathbb{R} \to [0,1]$ è definita come $\sigma(x) = \frac{1}{(1+e^{-x})}$

- Per elevati valori negativi di input la sigmoide restituisce 0: il neurone non spara affatto

Reti Neurali

Sigmoide

Definizione

La *Sigmoide* $\sigma : \mathbb{R} \rightarrow [0, 1]$ è definita come $\sigma(x) = \frac{1}{(1+e^{-x})}$

- Per elevati valori negativi di input la sigmoide restituisce 0: il neurone non spara affatto
- Per elevati valori positivi di input la sigmoide restituisce 1: il neurone satura e spara con una frequenza di sparo pari a 1

Reti Neurali

Sigmoide

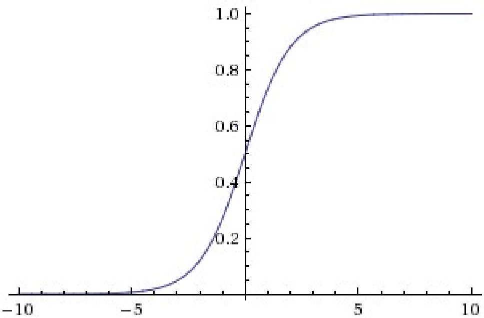
Definizione

La *Sigmoide* $\sigma : \mathbb{R} \to [0, 1]$ è definita come $\sigma(x) = \frac{1}{(1+e^{-x})}$

- Per elevati valori negativi di input la sigmoide restituisce 0: il neurone non spara affatto
- Per elevati valori positivi di input la sigmoide restituisce 1: il neurone satura e spara con una frequenza di sparo pari a 1
- La sua derivata è uguale a $\sigma'(x) = 1 - \sigma(x)$

Reti Neurali

Sigmoide



Rappresentazione grafica Sigmoide

Reti Neurali

Tangente Iperbolica

Definizione

La *Tangente Iperbolica* $\tanh : \mathbb{R} \rightarrow [-1, 1]$ è definita come $\tanh(x) = 2\sigma(2x) - 1$

Reti Neurali

Tangente Iperbolica

Definizione

La *Tangente Iperbolica* $\tanh : \mathbb{R} \rightarrow [-1, 1]$ è definita come $\tanh(x) = 2\sigma(2x) - 1$

- La tangente iperbolica è una sigmoide scalata

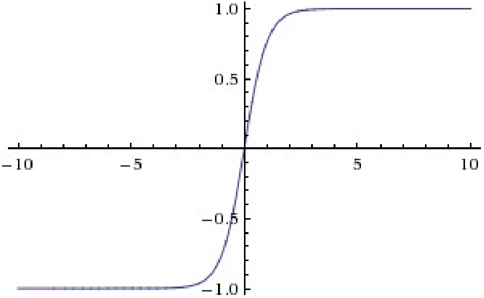
Definizione

La *Tangente Iperbolica* $\tanh : \mathbb{R} \to [-1, 1]$ è definita come $\tanh(x) = 2\sigma(2x) - 1$

- La tangente iperbolica è una sigmoide scalata
- La sua derivata è uguale a $\tanh'(x) = 1 - \tanh^2(x)$

Reti Neurali

Tangente Iperbolica



Rappresentazione grafica Tangente Iperbolica

Reti Neurali

Softplus

Definizione

La *Softplus* $s : \mathbb{R} \rightarrow [0, +\infty]$ è definita come $s(x) = \log(1 + e^x)$

Definizione

La *Softplus* $s : \mathbb{R} \to [0, +\infty]$ è definita come $s(x) = \log(1 + e^x)$

- La softplus è un approssimazione della *Rectifier Linear Unit (ReLU)*

Reti Neurali

Softplus

Definizione

La *Softplus* $s : \mathbb{R} \to [0, +\infty]$ è definita come $s(x) = \log(1 + e^x)$

- La softplus è un approssimazione della *Rectifier Linear Unit (ReLU)*
- Viene usata per sostituire la ReLU che presenta un punto di discontinuità in 0

Reti Neurali
Softplus

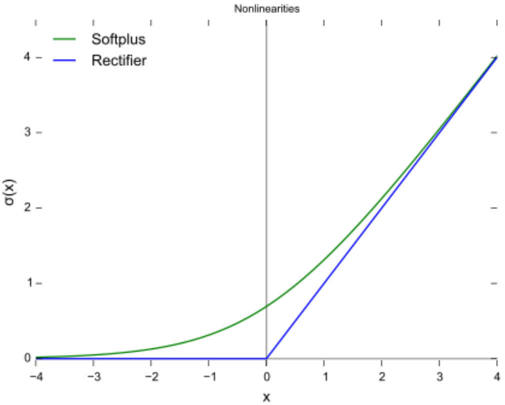
Definizione

La *Softplus* $s : \mathbb{R} \to [0, +\infty]$ è definita come $s(x) = \log(1 + e^x)$

- La softplus è un approssimazione della *Rectifier Linear Unit (ReLU)*
- Viene usata per sostituire la ReLU che presenta un punto di discontinuità in 0
- La sua derivata è uguale a $s'(x) = \frac{1}{(1+e^{-x})}$

Reti Neurali

Softplus



Confronto grafico tra ReLU e Softplus

Reti Neurali

Rete Neurale

Definizione

Una *Rete Neurale* è composta da un insieme di neuroni connessi tra loro in un grafo aciclico

Definizione

Una *Rete Neurale* è composta da un insieme di neuroni connessi tra loro in un grafo aciclico

- I neuroni sono organizzati in insiemi distinti chiamati *livelli* o *layer*

Reti Neurali

Rete Neurale

Definizione

Una *Rete Neurale* è composta da un insieme di neuroni connessi tra loro in un grafo aciclico

- I neuroni sono organizzati in insiemi distinti chiamati *livelli* o *layer*
- I livelli vengono posti uno di seguito all'altro in modo da formare una sequenza

Reti Neurali

Rete Neurale

Definizione

Una *Rete Neurale* è composta da un insieme di neuroni connessi tra loro in un grafo aciclico

- I neuroni sono organizzati in insiemi distinti chiamati *livelli* o *layer*
- I livelli vengono posti uno di seguito all'altro in modo da formare una sequenza
- I livelli intermedi prendono il nome di *hidden*

Reti Neurali

Rete Neurale

Definizione

Una *Rete Neurale* è composta da un insieme di neuroni connessi tra loro in un grafo aciclico

- I neuroni sono organizzati in insiemi distinti chiamati *livelli* o *layer*
- I livelli vengono posti uno di seguito all'altro in modo da formare una sequenza
- I livelli intermedi prendono il nome di *hidden*
- L'output dei neuroni di un livello diventano l'input dei neuroni del livello successivo

- Quando si effettua il conteggio dei livelli di una rete non si considera il livello di input

- Quando si effettua il conteggio dei livelli di una rete non si considera il livello di input

- Una rete a *singolo livello* non presenta livelli hidden

Reti Neurali

Rete Neurale

- Quando si effettua il conteggio dei livelli di una rete non si considera il livello di input
- Una rete a *singolo livello* non presenta livelli hidden
- Per determinare la grandezza di una rete ci si concentra sul numero di neuroni e sui relativi pesi ad essi associati

Reti Neurali

Livello Fully-Connected

Definizione

Un livello è di tipo *Fully-Connected* quando i neuroni appartenenti a due livelli adiacenti sono completamente connessi tra loro mentre i neuroni associati ad un singolo livello non condividono nessuna connessione

Definizione

Un livello è di tipo *Fully-Connected* quando i neuroni appartenenti a due livelli adiacenti sono completamente connessi tra loro mentre i neuroni associati ad un singolo livello non condividono nessuna connessione

- I pesi dei neuroni di un livello vengono salvati all'interno di matrici

Reti Neurali

Livello Fully-Connected

Definizione

Un livello è di tipo *Fully-Connected* quando i neuroni appartenenti a due livelli adiacenti sono completamente connessi tra loro mentre i neuroni associati ad un singolo livello non condividono nessuna connessione

- I pesi dei neuroni di un livello vengono salvati all'interno di matrici
- Le righe della matrice identificano i neuroni del livello mentre le colonne rappresentano i pesi di ciascun neurone

Reti Neurali

Livello Fully-Connected

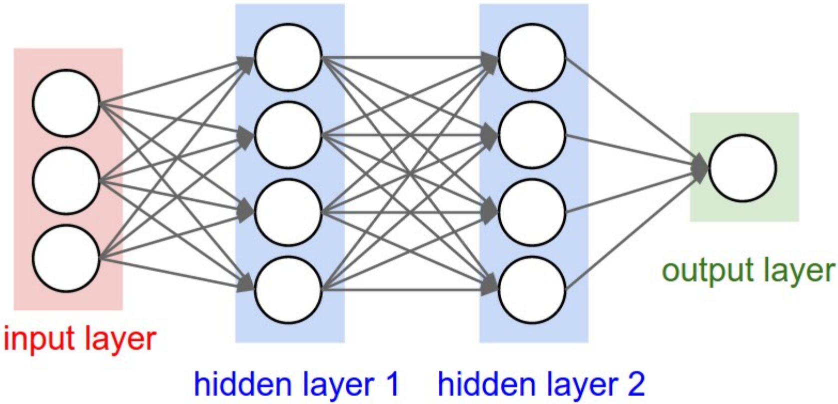
Definizione

Un livello è di tipo *Fully-Connected* quando i neuroni appartenenti a due livelli adiacenti sono completamente connessi tra loro mentre i neuroni associati ad un singolo livello non condividono nessuna connessione

- I pesi dei neuroni di un livello vengono salvati all'interno di matrici
- Le righe della matrice identificano i neuroni del livello mentre le colonne rappresentano i pesi di ciascun neurone
- La struttura a livelli di una rete neurale consente di facilitare le varie operazioni sfruttando il calcolo matriciale

Reti Neurali

Livello Fully-Connected



Una rete neurale a 3 livelli

Reti Neurali

Funzionamento

Il processo di apprendimento di una rete neurale è suddiviso in quattro fasi distinte

Il processo di apprendimento di una rete neurale è suddiviso in quattro fasi distinte

- *Inizializzazione dei pesi*

Reti Neurali
Funzionamento

Il processo di apprendimento di una rete neurale è suddiviso in quattro fasi distinte

- *Inizializzazione dei pesi*
- *Forward Propagation*

Reti Neurali
Funzionamento

Il processo di apprendimento di una rete neurale è suddiviso in quattro fasi distinte

- *Inizializzazione dei pesi*
- *Forward Propagation*
- *Funzione di perdita*

Reti Neurali

Funzionamento

Il processo di apprendimento di una rete neurale è suddiviso in quattro fasi distinte

- *Inizializzazione dei pesi*
- *Forward Propagation*
- *Funzione di perdita*
- *Back Propagation*

Reti Neurali

Inizializzazione dei pesi

- Al momento della nascita gli esseri umani non sono in grado di discriminare nessun tipo di oggetto a causa del mancato addestramento della loro rete neurale biologica

Reti Neurali

Inizializzazione dei pesi

- Al momento della nascita gli esseri umani non sono in grado di discriminare nessun tipo di oggetto a causa del mancato addestramento della loro rete neurale biologica
- Per riprodurre questo comportamento, all'inizio della fase di training, i pesi sinaptici w_i di ciascun livello vengono inizializzati in maniera casuale

Reti Neurali

Forward Propagation

Definizione

La *Forward Propagation* è il meccanismo utilizzato da una rete neurale per associare un'immagine ad una determinata classe

Reti Neurali

Forward Propagation

Definizione

La *Forward Propagation* è il meccanismo utilizzato da una rete neurale per associare un'immagine ad una determinata classe

- L'output dei neuroni del livello i viene moltiplicato per la matrice dei pesi del livello $i + 1$ ottenendo il vettore v

Reti Neurali

Forward Propagation

Definizione

La *Forward Propagation* è il meccanismo utilizzato da una rete neurale per associare un'immagine ad una determinata classe

- L'output dei neuroni del livello i viene moltiplicato per la matrice dei pesi del livello $i+1$ ottenendo il vettore v
- Al vettore v viene aggiunto il vettore dei bias del livello $i+1$

Reti Neurali

Forward Propagation

Definizione

La *Forward Propagation* è il meccanismo utilizzato da una rete neurale per associare un'immagine ad una determinata classe

- L'output dei neuroni del livello i viene moltiplicato per la matrice dei pesi del livello $i+1$ ottenendo il vettore v
- Al vettore v viene aggiunto il vettore dei bias del livello $i+1$
- L'output del livello $i+1$ si ottiene applicando la funzione di attivazione f ad ogni entry del vettore v

Reti Neurali

Forward Propagation

Definizione

La *Forward Propagation* è il meccanismo utilizzato da una rete neurale per associare un'immagine ad una determinata classe

- L'output dei neuroni del livello i viene moltiplicato per la matrice dei pesi del livello $i+1$ ottenendo il vettore v
- Al vettore v viene aggiunto il vettore dei bias del livello $i+1$
- L'output del livello $i+1$ si ottiene applicando la funzione di attivazione f ad ogni entry del vettore v
- Le operazioni precedenti sono svolte per tutti i livelli ad eccezione dell'ultimo

Reti Neurali

Funzione di perdita

Definizione

Una *funzione di perdita* L viene utilizzata per determinare l'errore di classificazione di una rete neurale

Reti Neurali

Funzione di perdita

Definizione

Una *funzione di perdita* L viene utilizzata per determinare l'errore di classificazione di una rete neurale

- La funzione di perdita più usata è la *Mean Squared Error (MSE)* $L = \frac{1}{2}\sum(y - o)^2$

Reti Neurali

Funzione di perdita

Definizione

Una *funzione di perdita* L viene utilizzata per determinare l'errore di classificazione di una rete neurale

- La funzione di perdita più usata è la *Mean Squared Error (MSE)* $L = \frac{1}{2}\sum(y-o)^2$
- y identifica l'output della rete mentre o l'etichetta dell'esempio considerato

Reti Neurali

Funzione di perdita

Definizione

Una *funzione di perdita* L viene utilizzata per determinare l'errore di classificazione di una rete neurale

- La funzione di perdita più usata è la *Mean Squared Error (MSE)* $L = \frac{1}{2}\sum(y - o)^2$
- y identifica l'output della rete mentre o l'etichetta dell'esempio considerato
- Minimizzando la funzione di perdita L si riduce l'errore di una rete neurale

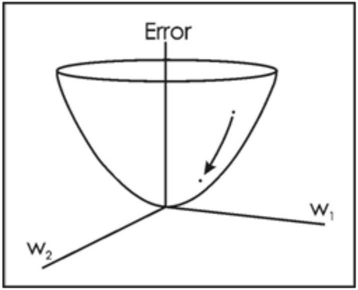
Reti Neurali

Funzione di perdita

Definizione

Una *funzione di perdita* L viene utilizzata per determinare l'errore di classificazione di una rete neurale

- La funzione di perdita più usata è la *Mean Squared Error (MSE)* $L = \frac{1}{2}\sum(y - o)^2$
- y identifica l'output della rete mentre o l'etichetta dell'esempio considerato
- Minimizzando la funzione di perdita L si riduce l'errore di una rete neurale
- Zalcolando la derivata di L in funzione dei pesi w_i si individua il minimo globale della funzione di perdita



Mean Squared Error (MSE). I pesi w_1 e w_2 sono le variabili indipendenti. La funzione di perdita L è la variabile dipendente

Definizione

La *Back Propagation* è il meccanismo utilizzato da una rete neurale per correggere gli errori di classificazione. Si individuano i pesi w_i che hanno influenzato maggiormente l'errore commesso e si aggiusta il loro valore in modo da ridurre la funzione di perdita

Definizione

La *Back Propagation* è il meccanismo utilizzato da una rete neurale per correggere gli errori di classificazione. Si individuano i pesi w_i che hanno influenzato maggiormente l'errore commesso e si aggiusta il loro valore in modo da ridurre la funzione di perdita

- Per calcolare la derivata della funzione L in funzione dei pesi w_i viene usata la *regola della catena (chain rule)*

Definizione

La *Back Propagation* è il meccanismo utilizzato da una rete neurale per correggere gli errori di classificazione. Si individuano i pesi w_i che hanno influenzato maggiormente l'errore commesso e si aggiusta il loro valore in modo da ridurre la funzione di perdita

- Per calcolare la derivata della funzione L in funzione dei pesi w_i viene usata la *regola della catena (chain rule)*
- Questa regola viene usata per trovare la derivata di una funzione composta

Reti Neurali

Aggiornamento dei Pesi e Learning Rate

- Il nuovo valore del peso w_i è dato da $w_i = w_i - \eta \frac{\partial L}{\partial w_i} = w_i + \Delta w_i$

Reti Neurali

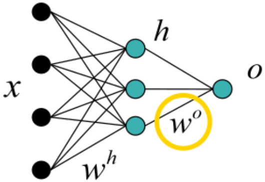
Aggiornamento dei Pesi e Learning Rate

- Il nuovo valore del peso w_i è dato da $w_i = w_i - \eta \frac{\partial L}{\partial w_i} = w_i + \Delta w_i$

- Il *learning rate* η è un parametro usato per controllare la velocità di convergenza della rete al valore minimo. Un learning rate alto comporta step elevati nell'aggiornamento dei pesi, un tempo di esecuzione più basso, ma una maggiore probabilità di terminare in un minimo locale

Reti Neurali

Esempio Back Propagation



$\boldsymbol{x} \in \mathbb{R}^{n,1} \quad \boldsymbol{w^h} \in \mathbb{R}^{n,m}$

$\boldsymbol{h} \in \mathbb{R}^{m,1} \quad \boldsymbol{w^o} \in \mathbb{R}^{1,m}$

$$z_j^h = \sum_{i=0}^{n} w_{ij}^h x_i \qquad z^o = \sum_{j=0}^{m} w_j^o h_j$$

$$h_j = f(z_j^h) \qquad o = f(z^o)$$

Reti Neurali

Esempio Back Propagation

- Dato il neurone o appartenente all'ultimo livello della rete neurale
$$\frac{\partial L}{\partial w_j^o} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial z^o} \cdot \frac{\partial z^o}{\partial w_j}$$

Reti Neurali

Esempio Back Propagation

- Dato il neurone o appartenente all'ultimo livello della rete neurale

$$\frac{\partial L}{\partial w_j^o} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial z^o} \cdot \frac{\partial z^o}{\partial w_j}$$

  - $\frac{\partial L}{\partial o} = \frac{\partial}{\partial o}[\frac{1}{2}(y - o)^2] = -(y - o)$

Reti Neurali

Esempio Back Propagation

- Dato il neurone o appartenente all'ultimo livello della rete neurale

$$\frac{\partial L}{\partial w_j^o} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial z^o} \cdot \frac{\partial z^o}{\partial w_j}$$

  - $\frac{\partial L}{\partial o} = \frac{\partial}{\partial o}[\frac{1}{2}(y - o)^2] = -(y - o)$
  - $\frac{\partial o}{\partial z^o} = f'(z^o)$

Reti Neurali

Esempio Back Propagation

- Dato il neurone o appartenente all'ultimo livello della rete neurale
$$\frac{\partial L}{\partial w_j^o} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial z^o} \cdot \frac{\partial z^o}{\partial w_j}$$
  - $\frac{\partial L}{\partial o} = \frac{\partial}{\partial o}[\frac{1}{2}(y - o)^2] = -(y - o)$
  - $\frac{\partial o}{\partial z^o} = f'(z^o)$
  - $\frac{\partial z^o}{\partial w_j^o} = h_j$

- Derivata della funzione L in funzione del peso w_j^o

$$\frac{\partial L}{\partial w_j^o} = -(y - o) \cdot f'(z^o) \cdot h_j = -\delta_j^o h_j$$

Reti Neurali

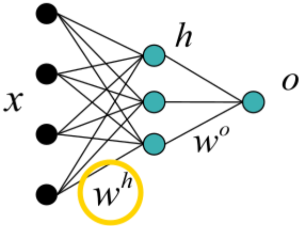
Esempio Back Propagation

- Derivata della funzione L in funzione del peso w_j^o
$$\frac{\partial L}{\partial w_j^o} = -(y - o) \cdot f'(z^o) \cdot h_j = -\delta_j^o h_j$$

- Aggiornamento del peso w_j^o
$$\Delta w_j^o = \eta \delta_j^o h_j$$

Reti Neurali

Esempio Back Propagation



$$\frac{\partial L}{\partial w_{ij}^h} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial z^o} \cdot \frac{\partial z^o}{\partial h_j} \cdot \frac{\partial h_j}{\partial z_j^h} \cdot \frac{\partial z_j^h}{\partial w_{ij}^h}$$

Reti Neurali

Esempio Back Propagation

$$\frac{\partial L}{\partial o} = \frac{\partial}{\partial o}[\frac{1}{2}(y-o)^2] = -(y-o)$$

Reti Neurali

Esempio Back Propagation

$$\frac{\partial L}{\partial o} = \frac{\partial}{\partial o}[\frac{1}{2}(y - o)^2] = -(y - o)$$

$$\frac{\partial o}{\partial z^o} = f'(z^o)$$

Reti Neurali

Esempio Back Propagation

$$\frac{\partial L}{\partial o} = \frac{\partial}{\partial o}[\frac{1}{2}(y-o)^2] = -(y-o)$$

$$\frac{\partial o}{\partial z^o} = f'(z^o)$$

$$\frac{\partial z^o}{\partial h_j} = w_j^o$$

Reti Neurali

Esempio Back Propagation

$$\frac{\partial L}{\partial o} = \frac{\partial}{\partial o}[\frac{1}{2}(y - o)^2] = -(y - o)$$

$$\frac{\partial o}{\partial z^o} = f'(z^o)$$

$$\frac{\partial z^o}{\partial h_j} = w_j^o$$

$$\frac{\partial h_j}{\partial z_j^h} = f'(z_j^h)$$

Reti Neurali

Esempio Back Propagation

$$\frac{\partial L}{\partial o} = \frac{\partial}{\partial o}[\frac{1}{2}(y - o)^2] = -(y - o)$$

$$\frac{\partial o}{\partial z^o} = f'(z^o)$$

$$\frac{\partial z^o}{\partial h_j} = w_j^o$$

$$\frac{\partial h_j}{\partial z_j^h} = f'(z_j^h)$$

$$\frac{\partial z_j^h}{\partial w_{ij}^h} = x_i$$

Reti Neurali

Esempio Back Propagation

- Derivata della funzione L in funzione del peso w_{ij}^h

$$\frac{\partial L}{\partial w_{ij}^h} = -(y - o) \cdot f'(z^o) \cdot w_j^o \cdot f'(z_j^h) \cdot x_i = -\delta_j^h x_i$$

Reti Neurali

Esempio Back Propagation

- Derivata della funzione L in funzione del peso w_{ij}^h

$$\frac{\partial L}{\partial w_{ij}^h} = -(y - o) \cdot f'(z^o) \cdot w_j^o \cdot f'(z_j^h) \cdot x_i = -\delta_j^h x_i$$

- Aggiornamento del peso w_{ij}^h

$$\Delta w_{ij}^h = \eta \delta_j^h x_i$$

Reti Neurali

Rete Neurale Convoluzionale

Una *Rete Neurale Convoluzionale* si differenzia da una più classica in quanto assume che l'input della rete sia un'immagine

Implementazione della Rete

Analisi dei Risultati