Apprendimento di Reti di Perceptron

Abbiamo visto che un singolo Perceptron non riesce ad apprendere tutte le funzioni booleane (es. XOR)

Però una rete di Perceptron può implementare una qualunque funzione booleana (tramite AND, OR, NOT).

Problema: come effettuare l'apprendimento di una rete di Perceptron ?

Non si sa come assegnare "credito" o "colpe" alle unità nascoste:

PROBLEMA DELL'ASSEGNAMENTO DEL CREDITO

Una possibile soluzione è quella di rendere il singolo neurone derivabile e sfruttare la tecnica di Discesa del Gradiente per apprendere i pesi "giusti".

Vediamo, quindi, come la Discesa del Gradiente si applica ad un Perceptron "semplifi cato".

Discesa di Gradiente

Consideriamo un Perceptron SENZA la hard-threshold:

$$out(\vec{x}) = \sum_{i=0}^{n} w_i x_i = \vec{w} \cdot \vec{x}$$

e defi niamo una misura dell'errore commesso da un particolare vettore dei pesi:

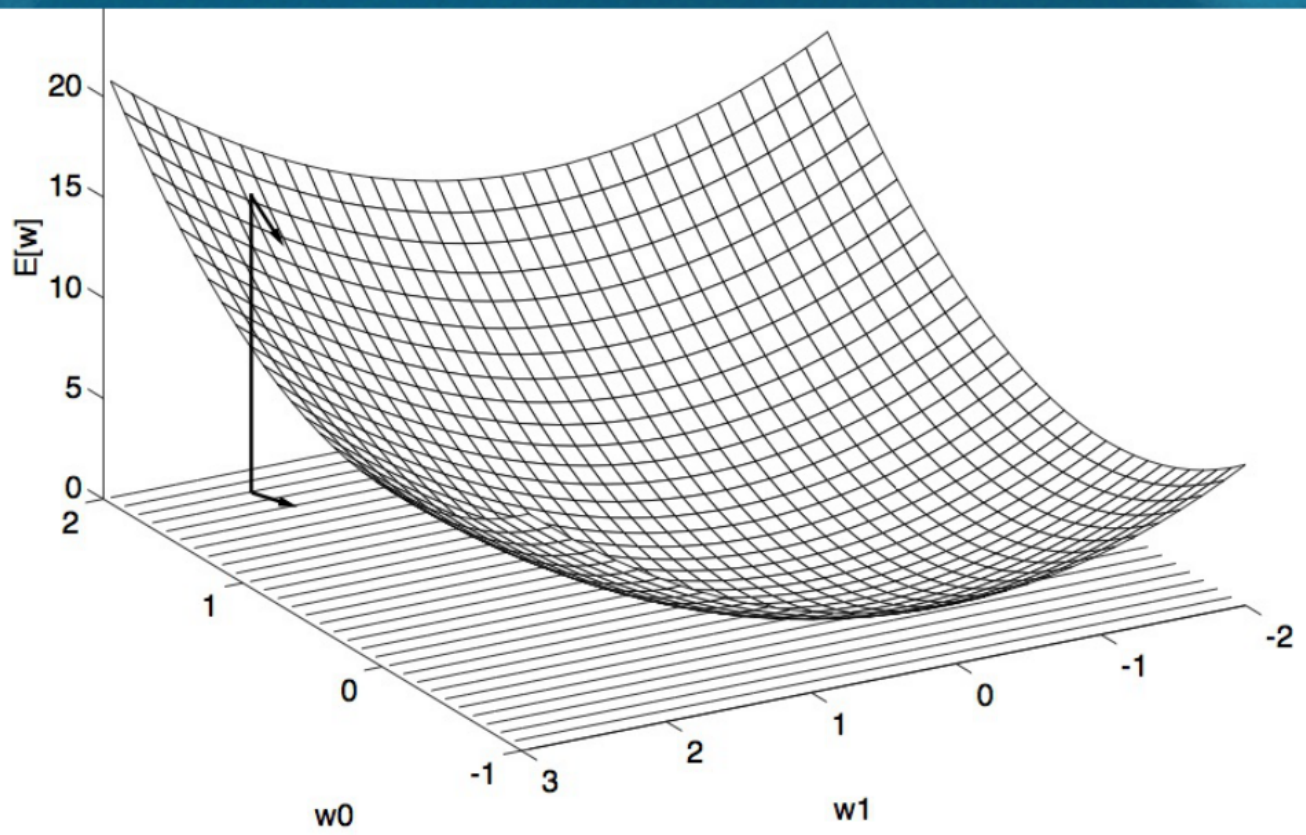
Funzione Errore: $E[\vec{w}] = \frac{1}{2N_{Tr}} \sum_{(\vec{x}^{(i)}, t^{(i)}) \in Tr} \left(t^{(i)} - out(\vec{x}^{(i)}) \right)^2$

dove N_{Tr} è la cardinalità dell'insieme di apprendimento Tr.

La funzione errore di sopra misura lo scarto quadratico medio (diviso 2) del valore target da quello predetto dal neurone (out).

Ovviamente, se $\forall (\vec{x}^{(i)}, t^{(i)}) \in Tr$ si ha $out(\vec{x}^{(i)}) = t^{(i)} \rightarrow \ E[\vec{w}] = 0$

Quindi bisogna MINIMIZZARE $E[\vec{w}]$ rispetto a $\vec{w}$



Idea base: partire da un $\vec{w}$ random e modifi carlo nella direzione contraria al gradiente (che indica la direzione di crescita di $E[\vec{w}]$)

$$\underbrace{\nabla E[\vec{w}]}_{\text{gradiente}} \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_n}\right], \qquad \Delta\vec{w} = -\eta \nabla E[\vec{w}], \qquad \Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Calcolo del gradiente

$$
\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i}\frac{1}{2N_{Tr}}\sum_{d\in Tr}(t^{(d)} - out^{(d)})^2 \\
&= \frac{1}{2N_{Tr}}\sum_{d\in Tr}\frac{\partial}{\partial w_i}(t^{(d)} - out^{(d)})^2 \\
&= \frac{1}{2N_{Tr}}\sum_{d\in Tr}2(t^{(d)} - out^{(d)})\frac{\partial}{\partial w_i}(t^{(d)} - out^{(d)}) \\
&= \frac{1}{N_{Tr}}\sum_{d\in Tr}(t^{(d)} - out^{(d)})\frac{\partial}{\partial w_i}(t^{(d)} - \vec{w}\cdot\vec{x}^{(d)}) \\
\frac{\partial E}{\partial w_i} &= \frac{1}{N_{Tr}}\sum_{d\in Tr}(t^{(d)} - out^{(d)})(-x_i^{(d)}) \\
&= -\frac{1}{N_{Tr}}\sum_{d\in Tr}(t^{(d)} - out^{(d)})(x_i^{(d)})
\end{aligned}
$$

Discesa di gradiente

Gradient-Descent(Tr, η)

ogni esempio di apprendimento è una coppia $(\vec{x}, t)$*, dove* $\vec{x}$ *è il vettore di valori in input, e* t *è il valore desiderato in output (target).* η *è il coefficiente di apprendimento (che ingloba* $\frac{1}{N_{Tr}}$*).*

- Assegna a w_i valori piccoli random
- Finché la condizione di terminazione non è verifi cata, fai
  - $\Delta w_i \leftarrow 0$
  - Per ogni $(\vec{x}, t)$ in Tr, fai
    - Presenta $\vec{x}$ al neurone e calcola l'output out
    - Per ogni w_i, fai

$$\Delta w_i \leftarrow \Delta w_i + \eta(t - out)x_i$$

  - Per ogni w_i, fai

$$w_i \leftarrow w_i + \Delta w_i$$

Discesa di gradiente con sigmoide

Consideriamo un Perceptron con funzione sigmoidale:

$$out(\vec{x}) = \sigma(\sum_{i=0}^{n} w_i x_i) = \sigma(\vec{w} \cdot \vec{x})$$

dove si ricorda che $\sigma(net) = \frac{1}{1+e^{-net}}$

Si noti che per $\sigma()$ vale la seguente relazione

$$\sigma'(net) = \frac{d\,\sigma(net)}{d\,net} = \sigma(net)(1 - \sigma(net))$$

e ricordiamo che (derivata di funzioni composte)

$$\frac{d\,f(g(x))}{d\,x} = \frac{d\,f(g(x))}{d\,g(x)} \frac{d\,g(x)}{d\,x}$$

$$
\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i}\frac{1}{2N_{Tr}}\sum_{d\in Tr}(t^{(d)}-out^{(d)})^2 \\
&= \frac{1}{2N_{Tr}}\sum_{d\in Tr}\frac{\partial}{\partial w_i}(t^{(d)}-out^{(d)})^2 \\
&= \frac{1}{2N_{Tr}}\sum_{d\in Tr}2(t^{(d)}-out^{(d)})\frac{\partial}{\partial w_i}(t^{(d)}-out^{(d)}) \\
&= \frac{1}{N_{Tr}}\sum_{d\in Tr}(t^{(d)}-out^{(d)})\frac{\partial}{\partial w_i}(t^{(d)}-\sigma(\vec{w}\cdot\vec{x}^{(d)})) \\
\frac{\partial E}{\partial w_i} &= \frac{1}{N_{Tr}}\sum_{d\in Tr}(t^{(d)}-out^{(d)})(-\frac{\partial\sigma(\vec{w}\cdot\vec{x}^{(d)})}{\partial\vec{w}\cdot\vec{x}^{(d)}}\frac{\partial\vec{w}\cdot\vec{x}^{(d)}}{\partial w_i}) \\
&= -\frac{1}{N_{Tr}}\sum_{d\in Tr}(t^{(d)}-out^{(d)})\sigma(\vec{w}\cdot\vec{x}^{(d)})(1-\sigma(\vec{w}\cdot\vec{x}^{(d)}))x_i^{(d)}
\end{aligned}
$$