Metodi Matematici per il Machine Learning

Davide Peccioli

a.a. 2024-25

Indice

Parte I

De Rossi

Capitolo 1

Reti Neurali

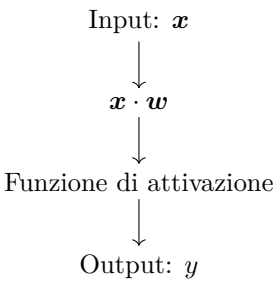
1.1 Neurone Artificiale

Un neurone è una cellula del corpo umano che può essere schematizzata come segue:

> A neuron is a cell which consists of the following parts: dendrites, axon,and body-cell. The synapse is the connection between the axon of one neuron and the dendrite of another. The functions of each part is briefly described below:
>
> - Dendrites are transmission channels that collect information from the axons of other neurons. The signal traveling through an axon reaches its terminal end and produces some chemicals x_i which are liberated in the synaptic gap. These chemicals are acting on the dendrites of the next neuron either in a strong or a weak way. The connection strength is described by the weight system w_i-
>
> - The body-cell collects all signals from dendrites. Here the dendrites activity adds up into a total potential and if a certain threshold is reached, the neuron fires a signal through the axon. The threshold depends on the sensitivity of the neuron and measures how easy is to get the neuron to fire.
>
> - The axon is the channel for signal propagation. The signal consists in the movement of ions from the body-cell towards the end of the axon. The signal is transmitted electrochemically to the dendrites of the next neuron.

Matematicamente, quindi, si considera un neurone come una unità che riceve degli input (un vettore $\boldsymbol{x}$), lo moltiplica per un vettore di pesi $\boldsymbol{w} = (w_0, \ldots, w_n)$, e produce un output processando il

prodotto scalare tramite una funzione di attivazione:

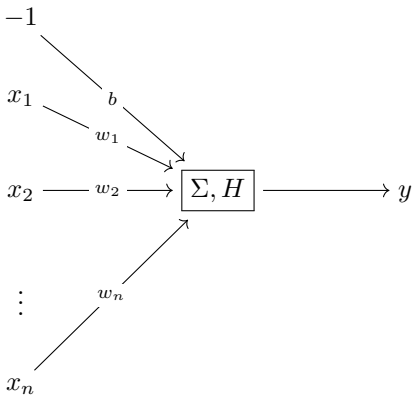


1.1.1 Neurone Sigma-Heaviside

Il modello più semplice è quello che riceve degli input, li somma dopo averli moltiplicati per dei pesi, e:

- restituisce 0 se la somma così ottenuta non supera un treshold b;
- restituisce 1 se la somma così ottenuta è maggiore o uguale a b.

Questo viene schematizzato in questo modo:



e l'output y è dato da[1]

$$y = H\left(\sum_{i=0}^{n} x_i\, w_i\right)$$

dove per convenzione si è posto $w_0 = b$ e $x_0 = -1$.

La convenzione è per semplicità di notazione, infatti

$$H\left(\sum_{i=0}^{n} x_i\, w_i\right) = \begin{cases} 1 & \sum_{i=1}^{n} x_i\, w_i \geq b \\ 0 & \sum_{i=1}^{n} x_i\, w_i < b \end{cases}$$

[1] La funzione $H : \mathbb{R} \to \mathbb{R}$ è la Funzione di Heaviside:

$$H(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

1.1.2 Regressione Lineare

Un altro esempio di neurone è quello che approssima[2] una funzione continua

$$f : K \to \mathbb{R}$$

con $K \subseteq \mathbb{R}^n$ compatto.

L'input del neurone sarà una n-upla $X = (x_1, \dots, x_n) \in K$, mentre l'output sarà la funzione lineare

$$L(X) = b + \sum_{i=1}^{n} a_i \, x_i$$

Per semplicità si considera l'approssimazione vicino allo zero, e si suppone che

$$L(0) = f(0) = 0$$

(a meno di traslazione verticale per $f(0)$).

Si vuole quindi minimizzare

$$C(a_1, \dots, a_n) = \frac{1}{2} \left\| f - L \right\|_{L^2} = \frac{1}{2} \int_K \left(\sum_{i=1}^{n} a_i x_i - f(X) \right)^2 \mathrm{d}x_1 \cdots \mathrm{d}x_n$$

calcolandone il gradiente

$$\begin{aligned} \frac{\partial C}{\partial a_k} &= \int_K x_k \left(\sum_{i=1}^{n} a_i x_i - f(X) \right) \mathrm{d}x_1 \cdots \mathrm{d}x_n \\ &= \sum_{i=1}^{n} a_i \int_K x_i x_k \, \mathrm{d}x_1 \cdots \mathrm{d}x_n - \int_K x_k f(X) \, \mathrm{d}x_1 \cdots \mathrm{d}x_n \end{aligned}$$

e dunque, posti

$$\rho_{ij} \coloneqq \int_K x_i x_j \, \mathrm{d}x_1 \cdots \mathrm{d}x_n, \qquad m_k \coloneqq \int_K x_k f(X) \, \mathrm{d}x_1 \cdots \mathrm{d}x_n$$

si ha che

$$\frac{\partial C}{\partial a_k} = \sum_{i=1}^{n} a_i \, \rho_{ik} - m_k$$

ovvero, in forma matriciale, posta[3] $\rho = (\rho_{ij})$, $\boldsymbol{a} = {}^T(a_1, \dots, a_n)$ e $\boldsymbol{m} = {}^T(m_1, \dots, m_n)$:

$$\nabla C = \rho \, \boldsymbol{a} - \boldsymbol{m}$$

[2] Approssima in senso L^2 (ovvero minimizza la norma L^2 della differenza).

[3] Vedi:

- Spazio delle matrici
- Matrice Trasposta
- Gradiente di una funzione

Dunque, posto che ρ sia invertibile, si ottiene che i valori ottimali per L siano

$$\boldsymbol{a} = \rho^{-1}\boldsymbol{m}.$$

Nel caso di funzioni a valori in $\mathbb{R}^m$ il problema si scompone nelle diverse coordinate.

1.2 Rete Neurale

Una rete neurale è [...]

Questa riceve degli input e produce un output, in base a certi parametri $\boldsymbol{w}$, per simulare la FUNZIONE TARGET; quest'ultima è l'obiettivo finale della Rete Neurale (ovvero si vuole far sì che l'output della rete neurale sia il più vicino possibile al risultato della funzione target).

Per misurare la distanza tra l'output di una rete e la funzione target si utilizza una funzione errore (o funzione costo), che deve essere scelta in base all'applicazione specifica.

Il processo di apprendimento è quello che, partendo dai parametri $\boldsymbol{w}$, li modifica (iterativamente), fino a dei parametri $\boldsymbol{w}^*$, che sono ottimali, nel senso che minimizzano la funzione errore. Dunque il processo di apprendimento comporta la minimizzazione della funzione costo.

1.3 Funzioni costo (Machine Learning)

Una funzione costo è una funzione che misura, dati certi parametri $\boldsymbol{w}$ di una rete neurale, quanto la rete neurale si discosta dalla funzione target.

1.3.1 La Funzione Errore Supremum

Una rete neurale prende input $x \in [0,1]$ e deve imparare una data funzione continua $\phi : [0,1] \to [0,1]$.

La funzione della rete neurale, dipendete dai parametri $\boldsymbol{w}, b$, è $f_{\boldsymbol{w},b}(x)$.

La funzione costo Supremum è

$$C(\boldsymbol{w}, b) \coloneqq \sup_{x \in [0,1]} |f_{\boldsymbol{w},b}(x) - \phi(x)|.$$

Se la funzione ϕ è conosciuta solo per N valori $x_1, \ldots, x_N$, allora la funzione costo diventa

$$C(\boldsymbol{w}, b) \coloneqq \max_{i=1,\ldots,N} |f_{\boldsymbol{w},b}(x_i) - \phi(x_i)|.$$

1.3.2 La Funzione Errore Norma L2

Una rete neurale prende input $x \in [0,1]$ e deve imparare una data funzione $\phi : [0,1] \to \mathbb{R}$ tale che

$$\int_0^1 (\phi(x))^2 \, \mathrm{d}x < \infty$$

La funzione della rete neurale, dipendete dai parametri $\boldsymbol{w}, \boldsymbol{b}$, è $f_{\boldsymbol{w},\boldsymbol{b}}(x)$. La funzione costo associata a questo tipo di problema è quella che misura la distanza nella norma L^2:

$$C(\boldsymbol{w}, \boldsymbol{b}) \coloneqq \int_{[0,1]} (f_{\boldsymbol{w},\boldsymbol{b}}(x) - \phi(x))^2 \, \mathrm{d}x.$$

Se la funzione ϕ è conosciuta soltanto in N punti

$$z_1 = \phi(x_1), \quad z_2 = \phi(x_2), \qquad , z_N = \phi(x_N)$$

allora, posti $\boldsymbol{z} = (z_1, \ldots, z_N)$ e $\boldsymbol{x} = (x_1, \ldots, x_N)$, la funzione costo diventa la distanza in $\mathbb{R}^N$ tra $\boldsymbol{z}$ e $f_{\boldsymbol{w},\boldsymbol{b}}(\boldsymbol{x}) \coloneqq \big(f_{\boldsymbol{w},\boldsymbol{b}}(z_1), \ldots, f_{\boldsymbol{w},\boldsymbol{b}}(z_N)\big)$:

$$C(\boldsymbol{w}, \boldsymbol{b}) = \big\| \boldsymbol{z} - f_{\boldsymbol{w},\boldsymbol{b}}(\boldsymbol{x}) \big\|^2 = \sum_{i=1}^{N} |z_i - f_{\boldsymbol{w},\boldsymbol{b}}(x_i)|^2$$

Interpretazione Geometrica

Fissati $\boldsymbol{x}$ e $\boldsymbol{z}$, la mappa $(\boldsymbol{w}, \boldsymbol{b}) \mapsto f_{\boldsymbol{w},\boldsymbol{b}}(\boldsymbol{x})$ rappresenta una ipersuperficie in $\mathbb{R}^N$:

$$\Phi(\boldsymbol{w}, \boldsymbol{b}) = \begin{pmatrix} \Phi_1(\boldsymbol{w}, \boldsymbol{b}) \\ \vdots \\ \Phi_N(\boldsymbol{w}, \boldsymbol{b}) \end{pmatrix} = \begin{pmatrix} f_{\boldsymbol{w},\boldsymbol{b}}(x_1) \\ \vdots \\ f_{\boldsymbol{w},\boldsymbol{b}}(x_N) \end{pmatrix}$$

e la funzione costo $C(\boldsymbol{w}, \boldsymbol{b})$ è la distanza euclidea in $\mathbb{R}^N$ di un punto sulla ipersuperficie dal punto $\boldsymbol{z}$. Si suppongano appropriate ipotesi di differenziabilità della ipersuperficie.

Il costo è minimizzato in $(\boldsymbol{w}^*, \boldsymbol{b}^*)$ quando la distanza è minima, ovvero quando $\Phi(\boldsymbol{w}^*, \boldsymbol{b}^*)$ è la proiezione ortogonale di $\boldsymbol{z}$ sulla ipersuperficie: questo significa che il vettore $\Phi(\boldsymbol{w}^*, \boldsymbol{b}^*) - \boldsymbol{z}$ è ortogonale al piano tangente alla ipersuperficie in $\Phi(\boldsymbol{w}^*, \boldsymbol{b}^*)$: quest'ultimo è generato dai vettori[4]

$$\partial_{w_k} \Phi(\boldsymbol{w}, \boldsymbol{b})|_{(\boldsymbol{w}^*, \boldsymbol{b}^*)}; \qquad \partial_{b_j} \Phi(\boldsymbol{w}, \boldsymbol{b})|_{(\boldsymbol{w}^*, \boldsymbol{b}^*)}$$

Richiedere l'ortogonalità, quindi, significa richiedere che i prodotti scalari:

$$\begin{aligned} \Big(\partial_{w_k} \Phi(\boldsymbol{w}, \boldsymbol{b})|_{(\boldsymbol{w}^*, \boldsymbol{b}^*)}\Big) \cdot (\Phi(\boldsymbol{w}^*, \boldsymbol{b}^*) - \boldsymbol{z}) = 0 \\ \Big(\partial_{b_j} \Phi(\boldsymbol{w}, \boldsymbol{b})|_{(\boldsymbol{w}^*, \boldsymbol{b}^*)}\Big) \cdot (\Phi(\boldsymbol{w}^*, \boldsymbol{b}^*) - \boldsymbol{z}) = 0. \end{aligned}$$

Queste sono le equazioni normali, che operativamente diventano

$$\begin{aligned} \sum_{i=1}^{N} (f_{\boldsymbol{w},\boldsymbol{b}}(x_i) - z_i) \cdot \partial_{w_k} f_{\boldsymbol{w},\boldsymbol{b}}(x_i)|_{(\boldsymbol{w},\boldsymbol{b})=(\boldsymbol{w}^*, \boldsymbol{b}^*)} = 0 \\ \sum_{i=1}^{N} (f_{\boldsymbol{w},\boldsymbol{b}}(x_i) - z_i) \cdot \partial_{b_j} f_{\boldsymbol{w},\boldsymbol{b}}(x_i)|_{(\boldsymbol{w},\boldsymbol{b})=(\boldsymbol{w}^*, \boldsymbol{b}^*)} = 0 \end{aligned}$$

[4] Vedi: "Derivata parziale"

1.3.3 Regolarizzazione della Funzione Costo (Machine Learning)

Per evitare il fenomeno dell'overfitting, è bene mantenere i parametri piccoli. Pertanto, data una funzione costo $C(\boldsymbol{w})$, la si regolarizza, utilizzando una funzione costo $G(\boldsymbol{w})$, data da $C(\boldsymbol{w})$ più un termine di regolarizzazione.

Regolarizzazione L^2. Si aggiunge alla funzione $C(\boldsymbol{w})$ la 2-norma in $\mathbb{R}^n$ dei parametri:

$$G(\boldsymbol{w}) \coloneqq C(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_2^2, \qquad \text{dove } \|\boldsymbol{w}\|_2^2 = \sum_{i=1}^{n} (w_i)^2.$$

Il valore $\lambda > 0$ è un moltiplicatore di Lagrange; questo parametro deve essere scelto in maniera da minimizzare l'*overfitting*.

Regolarizzazione L^1. Si aggiunge alla funzione $C(\boldsymbol{w})$ la 1-norma in $\mathbb{R}^n$ dei parametri:

$$G(\boldsymbol{w}) \coloneqq C(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_2^2, \qquad \text{dove } \|\boldsymbol{w}\|_1 = \sum_{i=1}^{n} |w_i|.$$

Il valore $\lambda > 0$ è un moltiplicatore di Lagrange. Questo metodo, non differenziabile nell'origine, potrebbe dare dei problemi nella ricerca dei minimi tramite il gradiente.

Potential Regolation. Sia $U : \mathbb{R}^n \to \mathbb{R}^+$ tale che:

1. $U(x) = 0$ se e solo se $x = 0$;
2. U ha un minimo assoluto in $x = 0$.

La funzione costo regolarizzata diventa:

$$G(\boldsymbol{w}) = C(\boldsymbol{w}) + \lambda U(\boldsymbol{w}), \qquad \lambda > 0$$

Il potenziale deve essere scelto in maniera tale che l'errore, utilizzando G, sia minore che utilizzando C.

1.4 Processo di apprendimento di una rete neurale

L'apprendimento di una rete neurale è il processo di ricerca dei parametri ottimali per approssimare la funzione target. Questo è un processo iterativo algoritmico, che genera una sequenza (w_t) di parametri. Poiché si parla di numeri immensi di elementi in questa sequenza, spesso ci si riferisce a t come una sorta di variabile temporale continua.

Si vuole allenare un modello per replicare una funzione target di cui si conoscono N valori: $\{(x_i, z_i)\}$, minimizzando la funzione costo $C(\boldsymbol{w})$.

Questo insieme è diviso in tre parti:

- training set $\mathcal{T}$ (c.a. 70% dei dati);
- test set $\mathbb{T}$ (c.a. 20% dei dati);
- validation set $\mathcal{V}$ (c.a. 10% dei dati).

Si suppone che siano identicamente distribuiti, e che siano indipendenti. Si ottengono quindi tre errori:

- errore di training $C_{\mathcal{T}}(\boldsymbol{w})$: è il valore della funzione costo utilizzando i valori della funzione target presi dal training set;
- errore di test $C_{\mathrm{T}}(\boldsymbol{w})$: è il valore della funzione costo utilizzando i valori della funzione target presi dal test set;
- validation error $C_{\mathcal{V}}(\boldsymbol{w})$: è il valore della funzione costo utilizzando i valori della funzione target presi dal validation set.

1.4.1 Errori di Training e di Test

Con un qualche algoritmo si trova il valore $\boldsymbol{w}^*$ che minimizza $C_{\mathcal{T}}$. Successivamente, si calcola $C_{\mathrm{T}}(\boldsymbol{w}^*)$, e generalmente vale:

$$C_{\mathcal{T}}(\boldsymbol{w}^*) \leq C_{\mathrm{T}}(\boldsymbol{w}^*)$$

Ci sono tre possibili scenari, a questo punto:

- sia $C_{\mathcal{T}}(\boldsymbol{w}^*)$ che $C_{\mathrm{T}}(\boldsymbol{w}^*)$ sono piccoli: questo è lo scenario desiderato;
- $C_{\mathcal{T}}(\boldsymbol{w}^*)$ è piccolo, ma $C_{\mathrm{T}}(\boldsymbol{w}^*)$ è grande: questo è un fenomeno di overfitting; questo significa che la rete neurale sta "memorizzando" il training set, e non riesce a generalizzare bene; probabilmente bisogna rivedere l'architettura della rete neurale, probabilmente diminuendo i parametri;
- sia $C_{\mathcal{T}}(\boldsymbol{w}^*)$ che $C_{\mathrm{T}}(\boldsymbol{w}^*)$ sono grandi: questo è un fenomeno di underfitting; bisogna rivedere l'architettura della rete neurale, probabilmente aumentando i parametri.

Pertanto si utilizza il test set per verificare che i valori dei parametri trovati sul training set siano sufficientemente generalizzabili.

1.4.2 Iperparametri di un processo di apprendimento

L'algoritmo di apprendimento dipende da un insieme di parametri diversi da quelli della rete neurale. Questi sono detti iperparametri.

Si utilizza la mimimizzazione del validation error proprio per regolare gli iperparametri.

1.4.3 Alcuni esempi di algoritmi di apprendimento

Regressione Lineare

Capitolo 2

Cenni di Analisi Matematica

2.1 Teoria della misura

2.1.1 Funzioni sigmoidali e funzioni discriminatorie

Definizione 2.1.1. *Una funzione $\sigma : \mathbb{R} \to [0,1]$ si dice sigmoidale se*[1]

$$\lim_{x \to -\infty} \sigma(x) = -1, \qquad \lim_{x \to +\infty} \sigma(x) = +1.$$

Definizione 2.1.2. *Sia $\mathcal{M}$ la famiglia delle misure di Baire per $\mathbb{R}^n$ sul cubo $I^n := [0,1]^n \subseteq \mathbb{R}$, finite, con segno e regolari.*

Una funzione $f : \mathbb{R} \to \mathbb{R}$ si dice discriminatoria per $\mathcal{M}$ se per ogni $\mu \in \mathcal{M}$:

$$\left(\forall \boldsymbol{w} \in \mathbb{R}^n, \forall \theta \in \mathbb{R} \quad \int_{I^n} f(\boldsymbol{w} \cdot \boldsymbol{x}) \, \mathrm{d}\mu(\boldsymbol{x}) = 0 \right) \implies \mu = 0$$

Proposizione 2.1.3. Ogni funzione sigmoidale $\sigma : \mathbb{R} \to [0,1]$ è discriminatoria per $\mathcal{M}$, dove $\mathcal{M}$ è l'insieme misure di Baire per $\mathbb{R}^n$ sul cubo $I^n := [0,1]^n \subseteq \mathbb{R}$, finite, con segno e regolare.

Ovvero, se $\sigma : \mathbb{R} \to [0,1]$ è tale che

$$\lim_{x \to -\infty} \sigma(t) = 0; \qquad \lim_{x \to +\infty} \sigma(t) = 1$$

allora, per ogni $\mu \in \mathcal{M}$,

$$\left(\forall \boldsymbol{w} \in \mathbb{R}^n, \forall \theta \in \mathbb{R} \quad \int_{I^n} f(\boldsymbol{w} \cdot \boldsymbol{x}) \, \mathrm{d}\mu(\boldsymbol{x}) = 0 \right) \implies \mu = 0$$

[1] Vedi "Limite (Analisi Matematica)"

2.2 Minimizzazione

Definizione 2.2.1.

Proposizione 2.2.2. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lemma 2.2.3. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Lemma 2.2.4. Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Teorema 2.2.5. Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Parte II

Cordero

Parte III

Sirovich