Metodi Matematici per il Machine Learning

Davide Peccioli

a.a. 2024-25

Indice

Parte I

De Rossi

Capitolo 1

Introduzione alle Reti Neurali

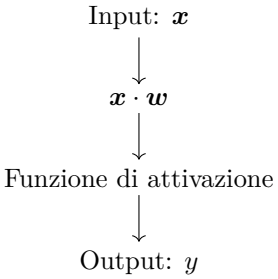
1.1 Neurone Artificiale

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 Σ-neurone come una unità che riceve degli input (un vettore $\boldsymbol{x}$), lo moltiplica per un vettore di pesi $\boldsymbol{w} = (w_0, \ldots, w_n)$, somma un certo bias, e produce

un output processando il prodotto scalare tramite una funzione di attivazione:



Altri tipi di neuroni, invece, detti Π-neuroni, moltiplicano gli input.

Definizione 1.1.1. *Un neurone astratto è una quadrupla* $\langle \boldsymbol{x}, \boldsymbol{w}, \varphi, y \rangle$ *dove* $\boldsymbol{x} = (x_0, x_1, \ldots, x_n)$ *è il vettore degli input,* $\boldsymbol{w} = \{w_0, w_1, \ldots, w_n\}$ *è il vettore dei pesi, con* $x_0 = -1$ *e* $w_0 = b$ *il bias, e* φ *è una funzione di attivazione.* y *è la funzione di output, che nel caso di un* Σ*-neurone è*

$$y = \varphi(\boldsymbol{w} \cdot \boldsymbol{x}).$$

Alcuni esempi di neuroni:

- Perceptron
- Neurone Sigmoidale

1.1.1 Input type di un neurone

Gli input del neurone possono essere di diversi tipi:

- *binary*: $x_i \in \{0, 1\}$;
- *signed*: $x_i \in \{-1, 1\}$;
- *digital*: $x_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- *real*: $x_i \in \mathbb{R}$;
- *interval*: $x_i \in [a, b] \subseteq \mathbb{R}$.

1.1.2 Input efficiency

Sia n il numero degli input, e sia β il numero di stati possibili (ovvero quanti valori ciascun input può assumere).

Il costo computazionale dell'implementazione di un neurone F è

$$F \propto \beta \cdot n, \quad F = k\beta n.$$

Si consideri F fissato. Si vuole massimizzare la quantità di dati trasmessi, ovvero la quantità di stati diversi che la n-upla $(x_1, \ldots, x_n)$ può assumere: β^n.

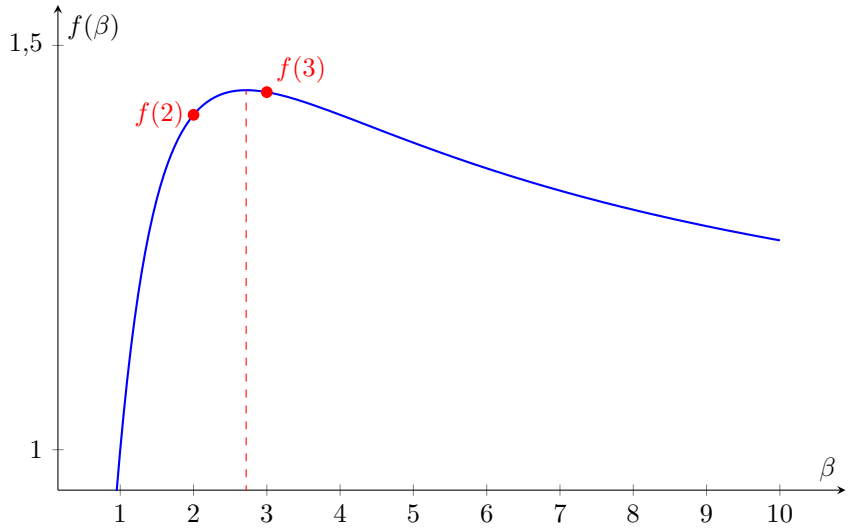


Figura 1.1: La funzione costo $f(\beta) = \beta^{1/\beta}$

Sia quindi $f(\beta)$ la quantità da massimizzare:

$$f(\beta) = \beta^n = \beta^{F/(k\beta)}$$

Per farlo, si considera

$$\begin{aligned}\frac{\partial}{\partial\beta}\ln f(\beta) &= \frac{\partial}{\partial\beta}\left(\frac{F}{k\beta}\ln\beta\right)\\ &= \frac{\partial}{\partial\beta}\left(\frac{F}{k\beta}\right)\cdot\ln\beta + \frac{F}{k\beta}\cdot\frac{\partial}{\partial\beta}(\ln\beta)\\ &= -\frac{F}{k\beta^2}\ln\beta + \frac{F}{k\beta^2}\\ &= \frac{F}{k\beta^2}(1-\ln\beta).\end{aligned}$$

Dunque $f(\beta)$ ha un massimo per $\beta = e$. Graficamente (vedi Fig. 1.1) è possibile vedere che, restringendosi a $\beta \in \mathbb{N}$, il valore più efficiente è $\beta = 3$.

1.1.3 Approssimazione di una funzione continua

Tramite un neurone è possibile approssimare[1] una funzione continua

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

con $K \subseteq \mathbb{R}^n$ compatto, utilizzando una regressione lineare.

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

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

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

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[2] $\rho = (\rho_{ij})$, $\boldsymbol{a} = {}^T(a_1, \dots, a_n)$ e $\boldsymbol{m} = {}^T(m_1, \dots, m_n)$:

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

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.

[2] Vedi:

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

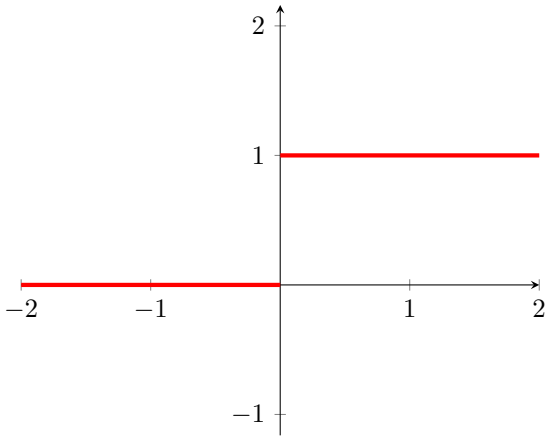


Figura 1.2: La funzione di Heaviside

1.2 Funzioni di attivazione

Nel Machine Learning le funzioni che agiscono nei neuroni vengono dette funzioni di attivazione. Se ne presentano alcuni esempi, con i loro nomi specifici.

Sono tutte funzioni $A \subseteq \mathbb{R} \to B \subseteq \mathbb{R}$.

1.2.1 Funzioni Lineari

Tra le funzioni di attivazione utilizzate vi sono le seguenti funzioni lineari:

- $f(x) = kx$, per $k > 0$ costante;
- la funzione identità $x \mapsto x$.

1.2.2 Step Functions

Threshold step function

La funzione di Heaviside (vedi Fig. 1.2)

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

la cui derivata (nel senso delle distribuzioni) è una Delta di Dirac: $H'(x) = \delta(x)$:

$$\delta(x) = \begin{cases} 0 & x \neq 0 \\ +\infty & x = 0 \end{cases}$$

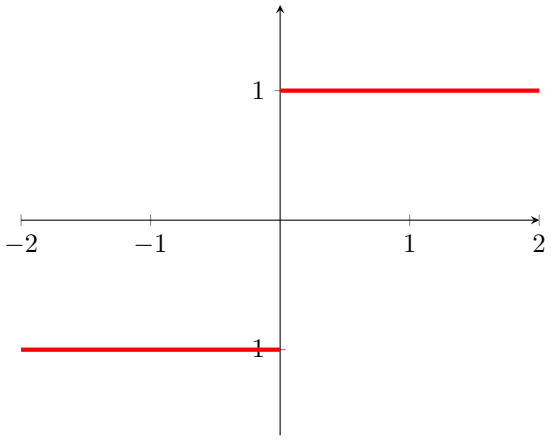


Figura 1.3: La funzione segno

Bipolar step function

La funzione segno: (vedi Fig. 1.3)

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

per cui vale: $S(x) = 2H(x) - 1$. Pertanto la sua derivata è

$$S'(x) = 2H'(x) = 2\delta(x)$$

1.2.3 Hockeystick Functions

Funzione di attivazione ReLU

La *Rectified Linear Unit* (ReLU) è (vedi Fig. 1.4)

$$\text{ReLU}(x) = xH(x) = \max\{0, x\}$$

e la sua derivata $\text{ReLU}'(x) = H(x)$.

PReLU

La *Parametric Rectivied Linear Unit* (PReLU) è (vedi Fig 1.5), per $\alpha > 0$

$$\text{PReLU}(\alpha; x) = \text{PReLU}_\alpha(x) = \begin{cases} \alpha x & x < 0 \\ x & x \geq 0 \end{cases}$$

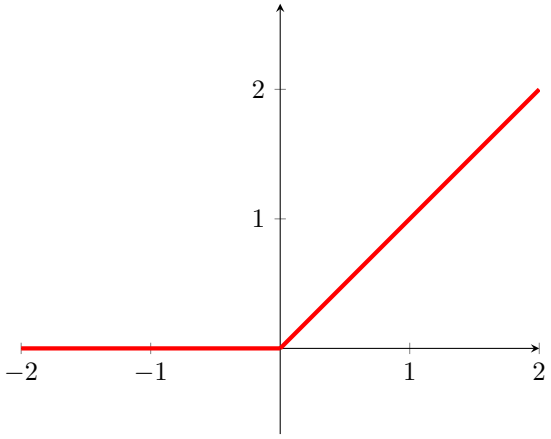


Figura 1.4: La funzione RELU(x)

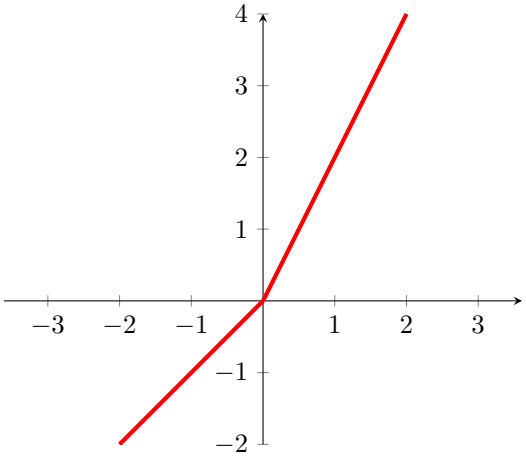


Figura 1.5: La funzione $\text{PRELU}_2(x)$

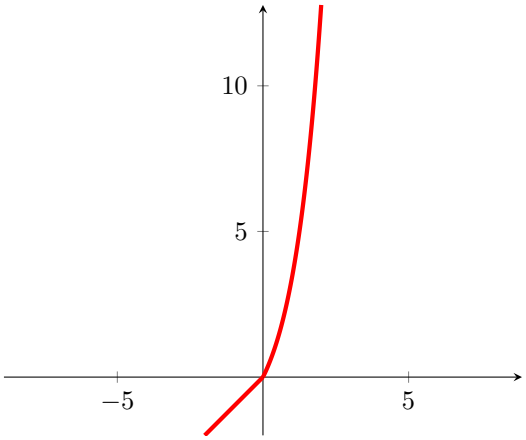


Figura 1.6: La funzione $\mathrm{ELU}(\alpha, x)$

ELU

La *Exponential Linear Units* (ELU) è (vedi Fig. 1.6):

$$\mathrm{ELU}(\alpha, x) = \begin{cases} x & x > 0 \\ \alpha(e^x - 1) & x \leq 0 \end{cases}$$

SELU

La *Scaled Exponential Linear Units* (SELU) è (vedi Fig. 1.7)

$$\mathrm{SELU}(\alpha, \lambda, x) = \lambda\, \mathrm{ELU}(\alpha, x) = \begin{cases} \lambda\, x & x > 0 \\ \alpha\lambda(e^x - 1) & x \leq 0. \end{cases}$$

SLU

La *Sigmoid Linear Units* (SLU) è (vedi Fig. 1.8)

$$\phi(x) = \frac{x}{1 + e^{-x}}.$$

Questa non è una funzione monotona, ma ha un minimo in $x_0 \approx -1,27$.

Spesso si usa anche la versione parametrica:

$$\phi_c(x) = \frac{x}{1 + e^{-cx}}.$$

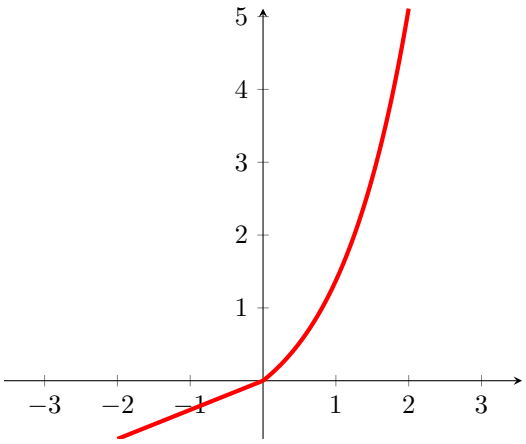


Figura 1.7: La funzione SELU$(0.4, 2, x)$

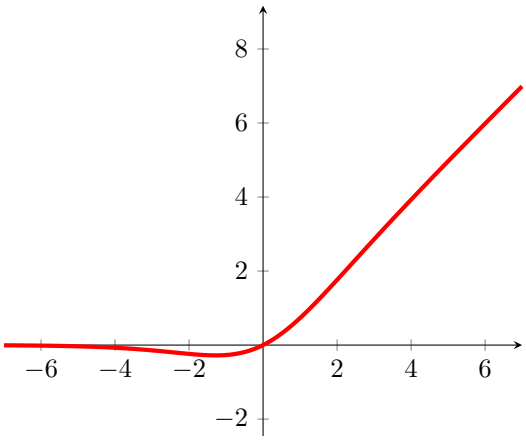


Figura 1.8: La funzione SLU(x)

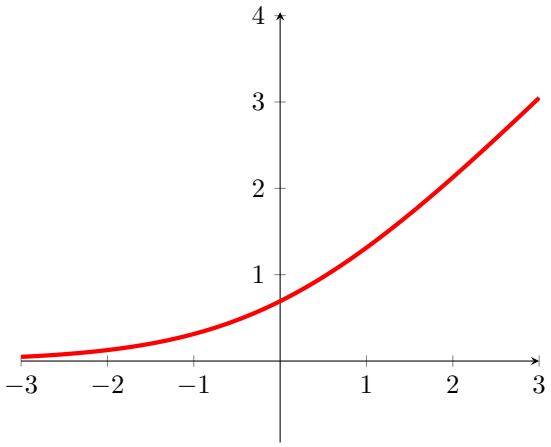


Figura 1.9: La funzione $\mathrm{sp}(x)$

Funzione Softplus

Questa è una funzione positiva crescente, con range $(0, +\infty)$: (vedi Fig. 1.9)

$$\mathrm{sp}(x) = \ln(1 + e^x).$$

Inoltre

$$\begin{aligned}\mathrm{sp}(x) - \mathrm{sp}(-x) &= \ln(1 + e^x) - \ln(1 + e^{-x}) \\ &= \ln\left(\frac{1 + e^x}{1 + e^{-x}}\right) = \ln\left(\frac{1 + e^x}{e^{-x}(1 + e^x)}\right) \\ &= \ln e^x = x.\end{aligned}$$

La sua derivata è

$$\mathrm{sp}'(x) = \frac{1}{1 + e^{-x}} > 0.$$

1.2.4 Funzioni Sigmoidali

Funzione Logistica

La funzione logistica è (vedi Fig. 1.10)

$$\sigma_c(x) = \sigma(c; x) = \frac{1}{1 + e^{-cx}}.$$

La famiglia di funzioni $(\sigma_c(x))_{c\in(0,+\infty)}$ approssima la funzione di Heaviside $H(x)$, in quanto

$$\lim_{c\to\infty} e^{-cx} = \begin{cases} 0 & x > 0 \\ 1 & x = 0 \\ +\infty & x < 0 \end{cases}$$

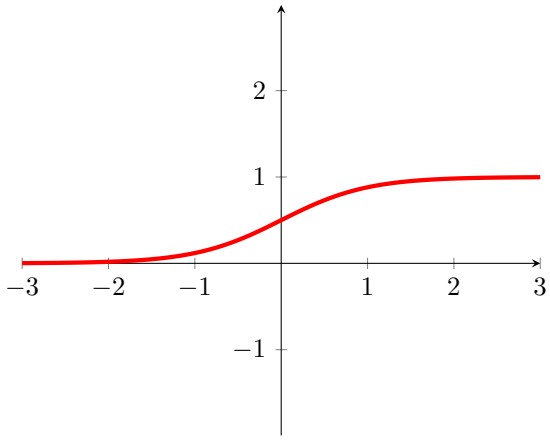


Figura 1.10: La funzione $\sigma_2(x)$

e pertanto

$$\lim_{c\to+\infty} \sigma_c(x) = \begin{cases} 1 & x > 0 \\ \frac{1}{2} & x = 0 \\ 0 & x < 0 \end{cases}$$

e pertanto, per ogni $x \neq 0$: $H(x) = \lim_{c\to+\infty} \sigma_c(x)$.

Le funzioni logistiche sono soluzioni della seguente equazione differenziale:

$$\sigma_c' = c\sigma_c\,(1 - \sigma_c)$$

infatti:

$$\begin{aligned} \sigma_c'(x) &= -\frac{1}{(1+e^{-cx})^2} \cdot (-c\,e^{-cx}) \\ &= c \cdot \frac{1}{1+e^{-cx}} \cdot \frac{e^{-cx}}{1+e^{-cx}} \\ &= c \cdot \frac{1}{1+e^{-cx}} \cdot \left(\frac{e^{-cx}+1-1}{1+e^{-cx}}\right) \\ &= c \cdot \frac{1}{1+e^{-cx}} \cdot \left(1 + \frac{-1}{1+e^{-cx}}\right) \\ &= c \cdot \sigma_c(x) \cdot (1 - \sigma_c(x)). \end{aligned}$$

Spesso ci si rifesce a $\sigma := \sigma_1$ come alla funzione logistica o funzione sigmoide (in quanto è l'archetipo della Funzione sigmoidale).

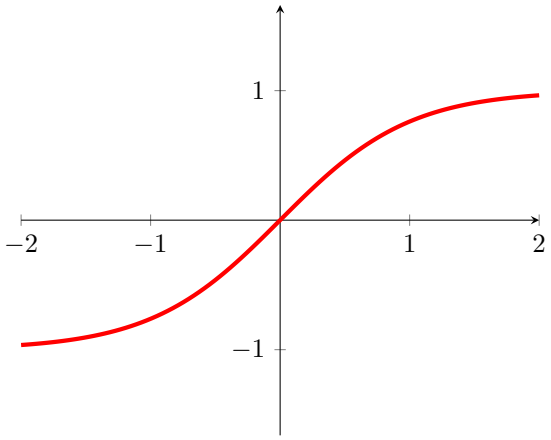


Figura 1.11: La funzione $\tanh(x)$

Tangente Iperbolica

La tangente iperbolica $\tanh(x)$ è (vedi Fig. 1.11)

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma_2(x) - 1$$

Inoltre si ha che $\tanh'(x) = 1 - (\tanh(x))^2$:

$$\begin{aligned}\tanh'(x) &= 2\sigma_2'(x) = 2 \cdot 2\sigma_2(x) \cdot (1 - \sigma_2(x)) \\ &= 2\sigma_2(x) \cdot (2 - 2\sigma_2(x)) \\ &= (\tanh(x) + 1) \cdot (1 - 2\sigma_2(x) + 1) \\ &= (\tanh(x) + 1)(-\tanh(x) + 1) = 1 - (\tanh(x))^2\end{aligned}$$

Arcotangente

È spesso utilizzata la seguente arcotangente (vedi Fig. 1.12):

$$h(x) = \frac{2}{\pi}\arctan(x) \qquad x \in \mathbb{R}.$$

Softsign

La seguente funzione differenziabile è la funzione *softsign*: (vedi Fig. 1.13)

$$\mathrm{so}(x) = \frac{x}{1 + |x|}, \qquad x \in \mathbb{R}$$

che ha raange $(-1, 1)$.

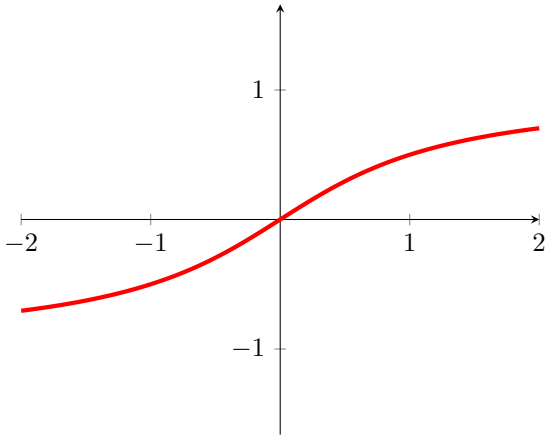


Figura 1.12: La funzione $h(x)$

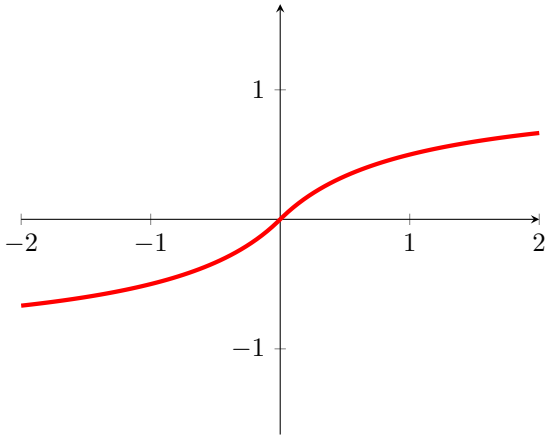


Figura 1.13: La funzione so(x)

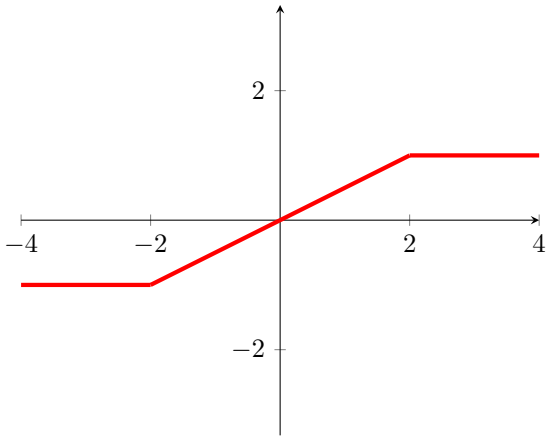


Figura 1.14: La funzione $f_2(x)$

Piecewise Linear

Dato un parametro $\alpha > 0$ (vedi Fig. 1.14)

$$f_\alpha(x) = f(\alpha, x) = \begin{cases} -1 & x \leq -\alpha \\ x/\alpha & -\alpha < x < \alpha \\ 1 & x \geq \alpha. \end{cases}$$

1.2.5 Bumped-type Functions

Gaussiana

La funzione gaussiana mappa $\mathbb{R}$ nell'intervallo $(0, 1]$: (vedi Fig. 1.15)

$$g(x) = e^{-x^2}, \qquad x \in \mathbb{R}.$$

Doppio esponenziale

Mappa la retta reale nell'intervallo $(0, 1]$ ed è definita: (vedi Fig. 1.16)

$$f(x) = e^{-\lambda |x|}, \qquad x \in \mathbb{R}, \lambda > 0.$$

1.3 Rete Neurale

Una rete neurale è [...]

Questa riceve degli input e produce un output (dando luogo ad una funzione input-output), in base a certi parametri $\boldsymbol{w}$, per simulare la FUNZIONE TARGET; quest'ultima è l'obiettivo finale della

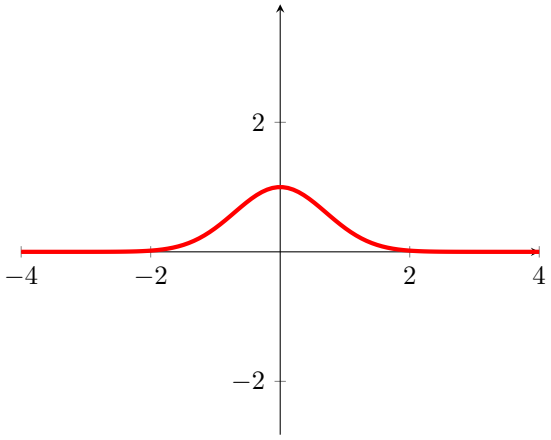


Figura 1.15: La funzione $g(x)$

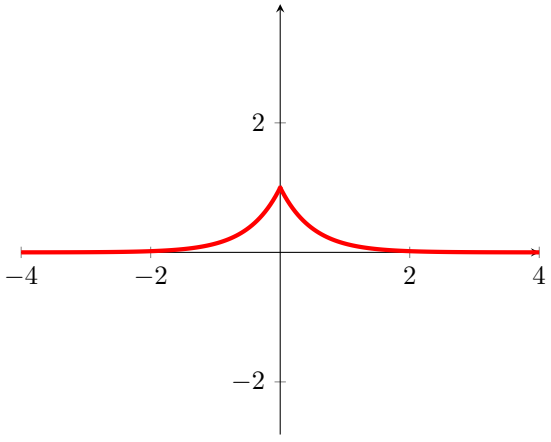


Figura 1.16: La funzione $f(x)$ con parametro $\lambda = 2$

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.

1.3.1 Hidden Layer di una rete neurale

1.3.2 Rete Neurale Feedforward

Definizione 1.3.1. *Una rete neurale ReLU-feedforward con input $\boldsymbol{x} \in \mathbb{R}^n$ e output $y \in \mathbb{R}^m$, illustrata in Fig. 1.17, è data da:*

- $L-1$ *numeri naturali* $\ell_1, \dots, \ell_{L-1}$*;*
- L *funzioni affini* $A_1, \dots, A_L$ *tali che*

$$\begin{aligned} A_1 &: \mathbb{R}^n \to \mathbb{R}^{\ell_1} \\ A_i &: \mathbb{R}^{\ell_{i-1}} \to \mathbb{R}^{\ell_i} \\ A_L &: \mathbb{R}^{\ell_{L-1}} \to \mathbb{R}^m \end{aligned}$$

L'output della rete è

$$A_L \circ \mathrm{ReLU} \circ A_{L-1} \circ \cdots \circ \mathrm{ReLU} \circ A_1,$$

dove ReLU *è la versione multidimensionale della funzione di attivazione ReLU, ovvero*

$$\mathrm{ReLU}(x_1, \dots, x_r) = \big(\mathrm{ReLU}(x_1), \dots, \mathrm{ReLU}(x_r)\big).$$

Si utilizza la seguente nomenclatura:

- L è la profondità della rete neurale;
- la larghezza della rete neurale è

  $$w := \max\{\ell_1, \dots, \ell_{L-1}\}$$

  mentre la larghezza di un layer è ℓ_i;
- la dimensione della rete neurale è

  $$s = \ell_1 + \cdots + \ell_{L-1}$$

  ovvero il numero di neuroni nascosti.

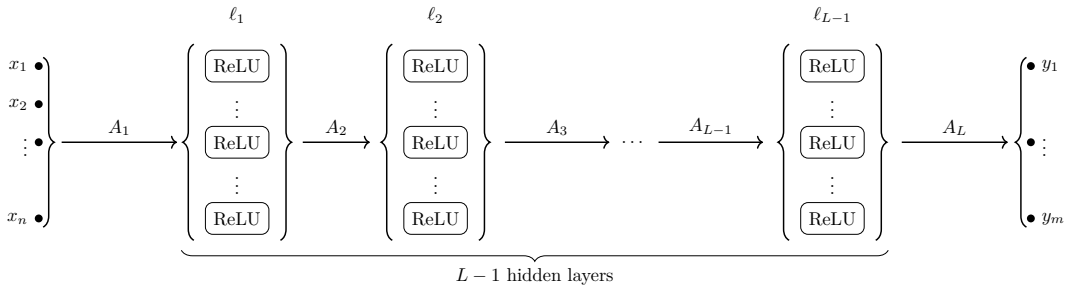


Figura 1.17: Una rete neurale ReLU-feedforward

1.4 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.

1.4.1 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

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

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

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

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[3]

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

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

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

Queste sono le equazioni normali, che operativamente diventano

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

1.4.3 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.

[3] Vedi: "Derivata parziale"

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.

1.5 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 $\mathscr{T}$ (c.a. 70% dei dati);
- test set 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 $C_{\mathscr{T}}(\boldsymbol{w})$: 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.

1.5.1 Errori di Training e di Test

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

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

Ci sono tre possibili scenari, a questo punto:

- sia $C_{\mathscr{T}}(\boldsymbol{w}^*)$ che $C_{\mathrm{T}}(\boldsymbol{w}^*)$ sono piccoli: questo è lo scenario desiderato;
- $C_{\mathscr{T}}(\boldsymbol{w}^*)$ 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 $C_{\mathscr{T}}(\boldsymbol{w}^*)$ 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.

1.5.2 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.

1.5.3 Alcuni esempi di algoritmi di apprendimento

Altri algoritmi sono:

- Perceptron Learning Algorithm.

Algoritmo di Regressione Lineare (Machine Learning)

Algoritmo di Gradient Descent

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]

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

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

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. *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.*

Proposizione 2.2.2. Sia $A \subseteq \mathbb{R}^n$ aperto e sia $f \in C^2(A)$[2]. Sia c un punto critico per f. Sia $H_f(c)$ l'Hessiana di f calcolata nel punto c.

- Se $H_f(c)$ è definita positiva, allora c è un punto di minimo locale forte.
- Se $H_f(c)$ è definita negativa, allora c è un punto di massimo locale forte.
- Se $H_f(c)$ è indefinita, allora c è un punto di sella.

Dimostrazione. Si dimostra che se $H_f(c)$ è definita positiva, allora c è un punto di minimo locale.

Sia $\gamma : [-\varepsilon, \varepsilon] \to A$ una curva di classe C^2 tale che $\gamma(0) = c$ e $\left\|\gamma'(0)\right\| = 1$. Sia $\boldsymbol{v} \coloneqq \gamma'(0)$. Sia $g(t) \coloneqq f \circ \gamma(t)$.

- $g'(0) = 0$. Infatti, applicando la chain rule:[3]

$$\begin{aligned} g'(0) &= \nabla f(\gamma(0)) \cdot \gamma'(0) \\ &= \nabla f(c) \cdot \boldsymbol{v} = 0 \cdot \boldsymbol{v} = 0. \end{aligned}$$

- $g''(0) > 0$. Infatti, si noti che[4]

$$g''(0) = D_{\boldsymbol{v}} \left(D_{\boldsymbol{v}} f\right)(c)$$

  Inoltre, siccome f è differenziabile, si ha che $D_{\boldsymbol{v}} f(x) = \nabla f(x) \cdot \boldsymbol{v}$.

  Anche $D_{\boldsymbol{v}} f$ è differenziabile, e pertanto $<$

$$D_{\boldsymbol{v}} \left(D_{\boldsymbol{v}} f\right)(x) = \nabla (D_{\boldsymbol{w}} f)(x) \cdot \boldsymbol{v}$$

  Ma

$$\nabla(\nabla f(x) \cdot \boldsymbol{v}) = H_f(x)\boldsymbol{v}$$

  e pertanto $g''(0) = H_f(c)\boldsymbol{v} \cdot \boldsymbol{v} > 0$. ■

Proposizione 2.2.3. 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.

[2] Vedi "Funzione di classe Ck"
[3] Vedi "Gradiente di una funzione"
[4] Vedi "Derivata direzionale"

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 2.2.4. 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 2.2.5. 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 2.2.6. 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

Sirovich

Capitolo 3

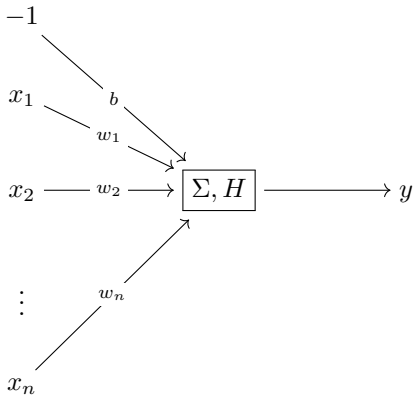
Neuroni artificiali

3.1 Percettrone

Il modello di neurone 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]

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

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

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

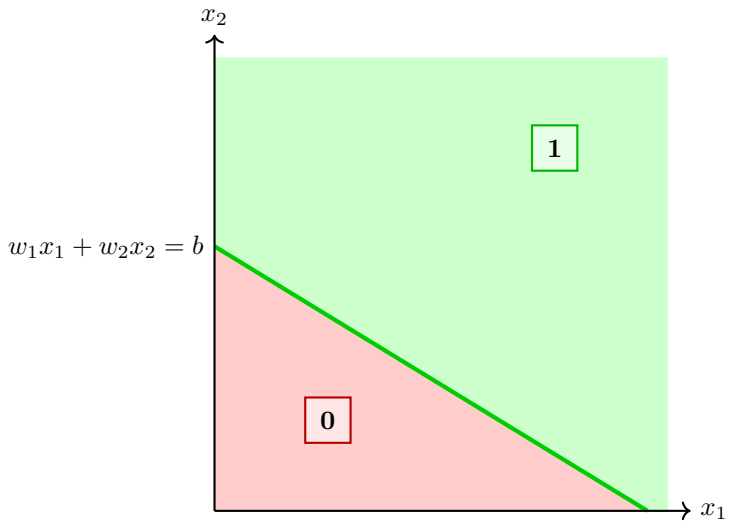


Figura 3.1: Output di un Percettrone

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

La convenzione è per semplicità di notazione, infatti

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

Definizione 3.1.1. *Il percettrone è il neurone con funzione di attivazione H.*

3.1.1 Interpretazione Geometrica

Consideriamo il neurone con dimensione di input 2, ovvero $\boldsymbol{x} = (x_1, x_2)$. Al variare di w_1, w_2, b, l'output del neurone è quello mostrato in Figura 3.1

Inoltre:

- il percettrone sa implementare la funzione AND

x_1	x_2	AND
0	0	0
0	1	0
1	0	0
1	1	1

  con parametri $w_1 = w_2 = 1$ e $b = 1.5$, come in figura 3.2

- il percettrone sa implementare la funzione OR

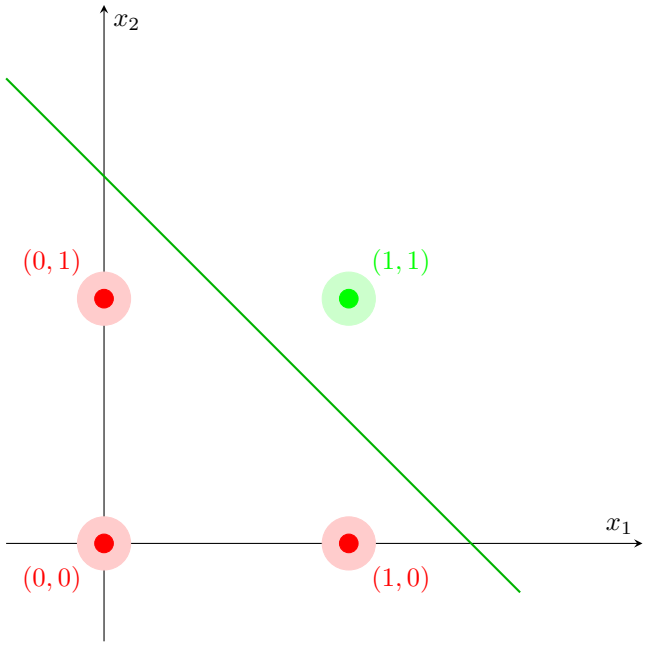


Figura 3.2: Funzione AND

x_1	x_2	OR
0	0	0
0	1	1
1	0	1
1	1	1

con parametri $w_1 = w_2 = 1$ e $b = 0.5$, come in figura 3.2

- il percettrone non sa implementare la funzione XOR, mostrata in figura 3.4

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Infatti, supponiamo che esistano $w_1, w_2, b \in \mathbb{R}$ tale che la retta $w_1 x_1 + w_2 x_2$ è

$$\begin{aligned} \{(0,1),(1,0)\} &\geq b \\ \{(0,0),(1,0)\} &< b \end{aligned}$$

allora $w_2 \geq b$ e $w_1 \geq b$ per la prima espressione, ma $w_1 < b$ per la seconda. Assurdo.

3.1.2 Generalizzazione: classificazione binaria

Si consideri un percettrone con input bidimensionale, $n = 2$, e $x, y \in [0, 1]$.

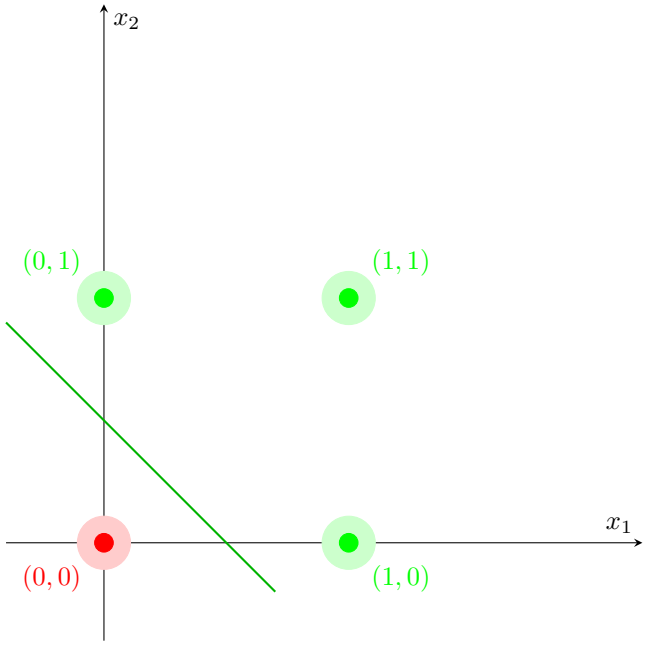


Figura 3.3: Funzione OR

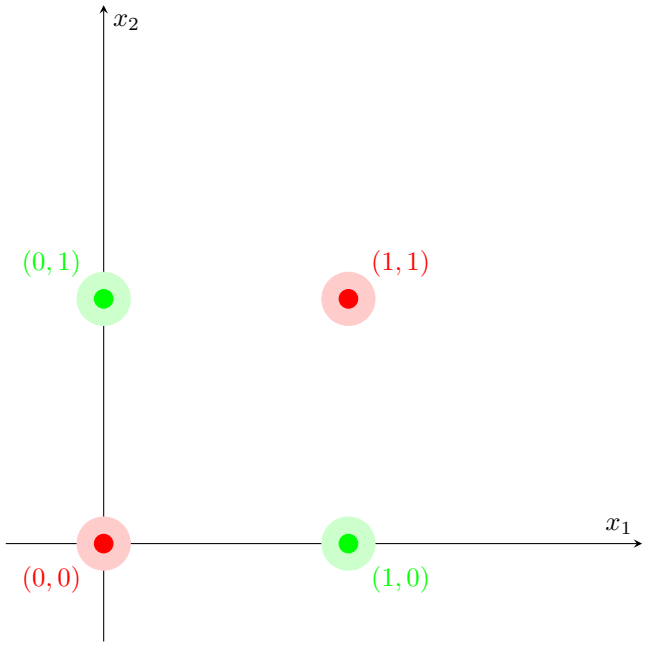


Figura 3.4: Funzione XOR

Lo spazio dei possibili input è rappresentato da $[0,1] \times [0,1]$. Si supponga che sia diviso in due: $\mathcal{G}_1, \mathcal{G}_\in$, e che vi sia una funzione $z(x,y) \in \{0,1\}$ che indica l'appartenenza ad uno dei due gruppi.

Si vuole studiare se il percettrone è in grado di distinguere gli elementi del gruppo 1 da quelli del gruppo 2. Sia $f_{\boldsymbol{w},b}(x,y)$ l'output del percettrone.

Si conoscesse esattamente $z(x,y)$ sarebbe possibile calcolare i pesi $\boldsymbol{w} = (w_1, w_2)$ e b minimizzando la norma L^2:[2]

$$C(\boldsymbol{w},b) = \int_{[0,1]^2} \big(f_{\boldsymbol{w},b}(x,y) - z(x,y)\big)^2 \,\mathrm{d}x\,\mathrm{d}y = \mathbb{E}\left[(f_{\boldsymbol{w},b}(x,y) - z(x,y))^2\right]$$

Questo è quindi il metodo del Mean Square Error (MSE)

Senza conoscere esattamente $\mathcal{G}_1$ e $\mathcal{G}_2$, ma conoscendo soltanto la classificazione di un numero finito di punti

$$\big\{\langle (x_i,y_i), z_i = z(x_i,y_i)\rangle \mid i = 1,\dots,N\big\}$$

è possibile soltanto calcolare la versione empirica $\widetilde{C}(\boldsymbol{w},b)$, utilizzando la legge dei grandi numeri:

$$\widetilde{C}(\boldsymbol{w},b) = \frac{1}{N}\sum_{i=1}^{N} \big(f_{\boldsymbol{w},b}(x_i,y_i) - z_i\big)^2 \quad \xrightarrow[N\to\infty]{} \quad C(\boldsymbol{w},b)$$

Questo è il metodo dell'Empirical MSE.

3.1.3 Perceptron Learning Algorithm

Sia $D = \big\{(\boldsymbol{x}_i, y_i) \mid i = 1,\dots,N\big\}$, con $z_i \in \{0,1\}$ l'insieme su cui si esegue l'allenamento, e sia $f_{\boldsymbol{w}}(\boldsymbol{x}) = \varphi(\boldsymbol{w}\cdot\boldsymbol{x})$ la funzione di output di un neurone perceptron.

Si definiscono

$$P \coloneqq \big\{\boldsymbol{x} \mid (\boldsymbol{x},1) \in D\big\}$$
$$N \coloneqq \big\{\boldsymbol{x} \mid (\boldsymbol{x},0) \in D\big\}$$

Si vuole trovare $\boldsymbol{w}^*$ tale per cui

$$\forall \boldsymbol{x} \in P: \quad f_{\boldsymbol{w}^*}(\boldsymbol{x}) = 1$$
$$\forall \boldsymbol{x} \in N: \quad f_{\boldsymbol{w}^*}(\boldsymbol{x}) = 0$$

L'algoritmo di apprendimento per questo task è il seguente: si scegla $\boldsymbol{w}^{(0)}$ casualmente. Il seguente ciclo iterativo si esegue finché non c'è convergenza: al passo $k+1$, si scelga casualmente $\boldsymbol{x} \in P \cup N$;

- se $\boldsymbol{x} \in P$ e $\boldsymbol{w}^{(k)} \cdot \boldsymbol{x} < 0$ allora si pone $\boldsymbol{w}^{(k+1)} \coloneqq \boldsymbol{w}^{(k)} + \boldsymbol{x}$;
- se $\boldsymbol{x} \in N$ e $\boldsymbol{w}^{(k)} \cdot \boldsymbol{x} \geq 0$ allora si pone $\boldsymbol{w}^{(k+1)} \coloneqq \boldsymbol{w}^{(k)} - \boldsymbol{x}$;
- altrimenti si pone $\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)}$.

[2] Vedi "Valore atteso"

In sostanza l'algoritmo è riassumibile come segue: fissato $\boldsymbol{w}^{(0)}$ e una successione di punti $(\boldsymbol{x}_i)_{i\in\mathbb{N}} \subseteq P \cup N$ catalogati da y_i[3], si ha che

$$\boldsymbol{w}^{(k+1)} := \boldsymbol{w}^{(k)} + \Big(y_k - \varphi(\boldsymbol{w}^{(k)} \cdot \boldsymbol{x}_k)\Big) \cdot \boldsymbol{x}_k.$$

Teorema 3.1.2. Se il dataset è linearmente separabile allora il percettrone trova un iperpiano di separazione in un numero finito di passi, altrimenti cicla infinite volte.

Dimostrazione. 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. ■

3.2 Neurone Sigmoidale

Il neurone sigmoidale è $\langle \boldsymbol{x}, \boldsymbol{y}, \sigma, y\rangle$ dove

- σ è una funzione di attivazione sigmoidale, o direttamente la sigmoide

$$\sigma(t) = \frac{1}{1+e^{-t}}$$

- $y = \sigma(\boldsymbol{w}\cdot\boldsymbol{x}) = \sigma(\boldsymbol{w}\cdot\boldsymbol{x} - b)$.

Il vantaggio dei neuroni sigmoidali rispetto al percettrone è la continuità di σ (infatti la Heaviside non è continua): a piccole variazioni di input e pesi corrispondono piccole variazioni dell'output; nello specifico:

$$\begin{aligned}
\mathrm{d}y &= \sum_{i=1}^{n} \frac{\partial y}{\partial w_i}\,\mathrm{d}w_i + \frac{\partial y}{\partial b}\,\mathrm{d}b \\
&= \sum_{i=1}^{n} \sigma'(\boldsymbol{w}\cdot\boldsymbol{x} - b)x_i\,\mathrm{d}w_i + \sigma'(\boldsymbol{w}\cdot\boldsymbol{x} - b)(-1)\,\mathrm{d}b \\
&= \sigma'(\boldsymbol{w}\cdot\boldsymbol{x} - b)\cdot\left(\sum_{i=1}^{n} x_i\,\mathrm{d}w_i - \mathrm{d}b\right)
\end{aligned}$$

Si noti inoltre che la derivata della sigmoide è maggiorata da 1.

[3] Ovvero per ogni $i \in \mathbb{N}$, $(\boldsymbol{x}_i, y_i) \in D$.

Capitolo 4

Bias-Variance tradeoff

4.1 Bias-Variance tradeoff nel Machine Learning

Si consideri un dataset $D = \{(x_i, z_i) \mid i = 1, \dots, N\}$, si consideri una rete neurale con output $f_{\boldsymbol{w},b}(t)$ dipendente dai parametri $\boldsymbol{w}, b$, e se ne consideri il processo di addestramento tramite minimizzazione della funzione costo[1]

$$\frac{1}{N}\sum_{i=1}^{N}(z_i - f_{\boldsymbol{w},b}(x_i))^2.$$

Si divide quindi il dataset D negli insiemi di training $\mathscr{T}$ e di test T. Ciascuna possibile scelta di $\mathscr{T}$ produce, tramite minimizzazione, una funzione $f_{\mathscr{T}} \coloneqq f_{\boldsymbol{w}^*,b^*}$.

Fissato dunque $(x_0, z_0) \in D$, si vuole calcolare in che modo varia l'errore commesso in quel punto in relazione alla scelta di $\mathscr{T}$:

$$\mathbb{E}_{\boldsymbol{w},b}\left[\left(z_0 - f_{\boldsymbol{w},b}(x_0)\right)^2\right].$$

dove la media è calcolata al variare dei parametri $\boldsymbol{w}, b$ (il variare dei parametri è equivalente al variare di $\mathscr{T}$).

Per semplicità si omettono le variabili e i parametri. Si noti che l'unica cosa che dipende dai

[1] Vedi Empirical MSE

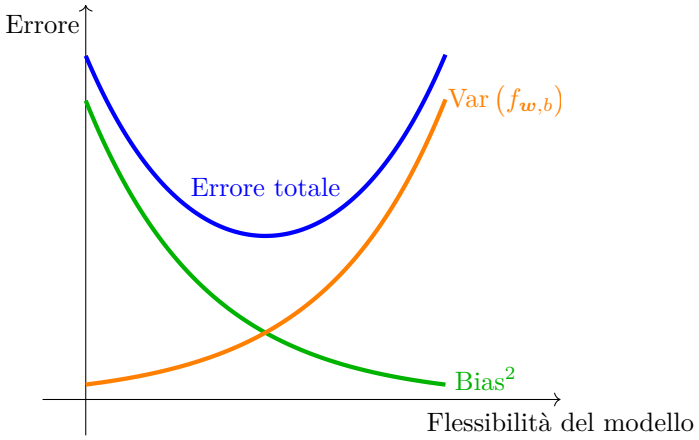


Figura 4.1: Errore nell'approssimazione di un dataset

parametri è $f_{\boldsymbol{w},b}$.

$$\begin{aligned}
\mathbb{E}\left[(z_0 - f)^2\right] &= \mathbb{E}\left[(z_0 - \mathbb{E}[f] + \mathbb{E}[f] - f)^2\right] \\
&= \mathbb{E}\left[(z_0 - \mathbb{E}[f])^2 + (f - \mathbb{E}[f])^2 + 2(z_0 - \mathbb{E}[f])(\mathbb{E}[f] - f)\right] \\
&= \mathbb{E}\left[(z_0 - \mathbb{E}[f])^2\right] + \underbrace{\mathbb{E}\left[(f - \mathbb{E}[f])^2\right]}_{\mathrm{Var}(f)} + \mathbb{E}\left[2(z_0 - \mathbb{E}[f])(\mathbb{E}[f] - f)\right] \\
&= (z_0 - \mathbb{E}[f])^2 + \mathrm{Var}(f) + 2(z_0 - \mathbb{E}[f])\mathbb{E}\left[\mathbb{E}[f] - f\right] \\
&= (z_0 - \mathbb{E}[f])^2 + \mathrm{Var}(f) + 2(z_0 - \mathbb{E}[f])\cancel{(\mathbb{E}[f] - \mathbb{E}[f])} \\
&= (z_0 - \mathbb{E}[f])^2 + \mathrm{Var}(f) \\
&= \mathrm{Var}\left(f_{\boldsymbol{w},b}\right) + \left(z_0 - \mathbb{E}[f_{\boldsymbol{w},b}]\right)^2
\end{aligned}$$

dove $\left(z_0 - \mathbb{E}[f_{\boldsymbol{w},b}]\right)$ è il bias.

Se il modello è molto flessibile (ovvero si può adattare molto bene ai dati), allora

$$z_0 \approx \mathbb{E}[f_{\boldsymbol{w},b}].$$

Allo stesso tempo la varianza è molto alta, in quanto basta modificare di poco il training set affinché la funzione di output (adattandosi molto ai dati) cambi tanto.

Viceversa, se il modello è poco flessibile, allora difficilmente $\left(z_0 - \mathbb{E}[f_{\boldsymbol{w},b}]\right)$ è piccolo; allo stesso tempo, cambiando di poco il training set, la funzione di output non cambia molto, e dunque la varianza è bassa.

Pertanto le due parti si comportano come mostrato in figura 4.1

Parte III

Cordero

Capitolo 5

Teoremi di approssimazione

5.1 Teoremi di Dini per la convergenza uniforme

Teorema 5.1.1. Sia $f_n : [a,b] \to \mathbb{R}$ una successione di funzioni continue.

1. Se per ogni $n \in \mathbb{N}$: $f_{n+1} \leq f_n$ e
$$\forall x \in [a,b] \ f_n(x) \to 0$$
allora f_n converge uniformemente a 0 su $[a,b]$.
2. Sia $g : [a,b] \to \mathbb{R}$ continua. Se per ogni $n \in \mathbb{N}$: $f_{n+1} \leq f_n$ e
$$\forall x \in [a,b] \ f_n(x) \to g(x)$$
allora f_n converge uniformemente a g su $[a,b]$.

5.2 Teorema di Ascoli-Arzelà

Definizione 5.2.1. *Una famiglia di funzioni $\mathcal{F}$ da un insieme A a valori reali si dice uniformemente limitata (*uniformly bounded*) se esiste $M > 0$ tale che*

$$\forall x \in A, \ \forall f \in \mathcal{F} \ |f(x)| \leq M.$$

Esempio 5.2.2. Sia $\mathcal{F} \coloneqq \{\cos(ax+b); a,b \in \mathbb{R}\}$. Allora la famiglia $\mathcal{F}$ è uniformemente limitata su $\mathbb{R}$, per $M = 1$.

Esempio 5.2.3. Si consideri

$$\mathcal{F} \coloneqq \left\{ \sum_{j=1}^{N} \alpha_j \sigma(w_j x + b_j); \alpha_j, w_j, b_j \in \mathbb{R} \mid \sum_{j=1}^{N} \alpha_j^2 \leq 1 \right\}$$

dove $\sigma(x)$ è una funzione sigmoidale fissata.

Questa famiglia è uniformemente limitata su $\mathbb{R}$, per $M = \sqrt{N}$, poiché, per C-S

$$\left| \sum_{j=1}^{N} \alpha_j \sigma(w_j x + b_j) \right|^2 \leq \left(\sum_{j=1}^{N} \alpha_j \right) \cdot \left(\sum_{j=1}^{N} \sigma(w_j x + b_j) \right) \leq 1 \cdot N = N.$$

Definizione 5.2.4. *Una famiglia di funzioni $\mathcal{F}$ da un insieme $A \subseteq \mathbb{R}$ a valori reali si dice equicontinua se per ogni $\varepsilon > 0$ esiste $\delta > 0$ tale che*

$$\forall f \in \mathcal{F} \ \forall x, y \in A \ \left(|x - y| < \delta \implies |f(x) - f(y)| < \varepsilon \right).$$

Equivalentemente, le funzioni di $\mathcal{F}$ sono uniformemente continua per gli stessi ε e δ.

Esempio 5.2.5. Si consideri $\mathcal{F} \subseteq C^1\left([a,b]\right)$[1] tale che esista $L > 0$:

$$\forall f \in \mathcal{F} \quad \sup_{x \in [a,b]} |f'(x)| < L.$$

Allora $\mathcal{F}$ è equicontinua.

Infatti, per il Teorema di Lagrange:

$$|f(x) - f(y)| \leq \left(\sup_{c \in [a,b]} |f'(c)| \right) \ |x - y| \leq L \ |x - y|$$

e dunque, scegliendo $\delta = \varepsilon / L$ si ha la tesi.

Esempio 5.2.6. Sia σ una funzione sigmoidale tale che esista $\lambda > 0$:

$$\forall x \in \mathbb{R} : \ |\sigma'(x)| < \lambda < 1$$

e sia $\mathcal{F}$ la famiglia definita su $[a, b] \subseteq \mathbb{R}$ come segue:

$$\mathcal{F} := \left\{ \sum_{j=1}^{N} \alpha_j \sigma(w_j x + b_j); \alpha_j, w_j, b_j \in \mathbb{R} \mid \sum_{j=1}^{N} (\alpha_j^2 + w_j^2) \leq 1 \right\}.$$

Allora $\mathcal{F}$ è equicontinua. Infatti:

- $\mathcal{F} \subseteq C^1\left([a,b]\right)$;
- Sia $f \in \mathcal{F}$: per C-S:

$$\begin{aligned} |f'(x)| = \left| \sum_{j=1}^{N} \alpha_j w_j \sigma'(w_j x + b_j) \right| &\leq \sum_{j=1}^{N} |\alpha_j| \, |w_j| \, \lambda \\ &\leq \lambda \left(\sum_{j=1}^{N} \alpha_j^2 \right)^{1/2} \cdot \left(\sum_{j=1}^{N} w_j^2 \right)^{1/2} \leq \lambda. \end{aligned}$$

[1] Vedi "Classe C di una funzione"

Per l'esempio precedente, si ottieene la tesi.

Si noti, inoltre, che se σ è la sigmoide logistica, allora la derivata massima è ottenuto in $x = 0$ e vale

$$\sigma'(0) = \sigma(0)\,(1 - \sigma(0)) = \frac{1}{4} < 1.$$

Teorema 5.2.7. Sia $\mathcal{F}$ una famiglia di funzioni continue su $[a, b]$ a valori reali. Sono fatti equivalenti:

1. Ogni successione $(f_n)_{n\in\mathbb{N}} \subseteq \mathcal{F}$ contiene una sottosuccessione $(f_{n_k})_{k\in\mathbb{N}}$ che converge uniformemente.
2. La famiglia $\mathcal{F}$ è equicontinua e uniformemente limitata.

Ecco alcuni semplici corollari del Teorema di Ascoli-Arzelà.

Teorema 5.2.8. Sia $N \geq 1$ un intero fissato, e si consideri una rete neurale con un layer nascosto tale che:

1. l'input della rete sia una variabile reale $x \in [a, b]$;
2. l'output della rete sia un neurone unidimensionale con funzione di attivazione lineare e zero bias;
3. ci sono N neuroni nel layer nascosti con una funzione di attivazione differenziabile tale che $|\sigma'| < \lambda < 1$;
4. i pesi soddisfano la condizione di regolarizzazione:

$$\sum_{j=1}^{N} (\alpha_j^2 + w_j^2) \leq 1$$

dove i w_j sono i pesi per l'input al layer nascosto, e gli a_j sono i pesi dal layer nascosto all'output.

Allora esiste una funzione continua $g : [a, b] \to \mathbb{R}$ che può essere approssimata dalla rete.

Dimostrazione. È sufficiente mostrare che la famiglia di funzioni output del sistema sia composta da funzioni continue, e che sia equicontinua e uniformemente limitata.

Con una banale applicazione del Teorema di Ascoli-Arzelà, si ottiene che ogni successione di funzioni output ammette una sottosuccessione convergente uniformemente ad una funzione continua; quest'ultima, data la uniforme convergenza, viene approssimata dalla rete con un grado di accuratezza arbitrario. ■

Proposizione 5.2.9. Anche per una rete neurale composta da un unico neurone, con output

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \sigma(\boldsymbol{w} \cdot \boldsymbol{x} + b)$$

dove σ è la funzione logistica, con pesi $\boldsymbol{w} \in \mathbb{R}^n, b \in \mathbb{R}$ e input $\boldsymbol{x} \in I_n \coloneqq [0, 1]^n$, esiste una funzione continua $g : I^n \to \mathbb{R}$ che può essere approssimata. Si supponga che $\|\boldsymbol{w}\| \leq 1$.

Dimostrazione. Infatti, la famiglia di funzioni continue

$$\mathcal{F} \coloneqq \left\{ f_{\boldsymbol{w},b} \mid \|\boldsymbol{w}\| \leq 1, b \in \mathbb{R} \right\}$$

è uniformemente limitata poiché $|f_{\boldsymbol{w},b}| < 1$, ed inoltre è equicontinua, in quanto, per il Teorema di Lagrange

$$\begin{aligned} |f_{\boldsymbol{w},b}(\boldsymbol{x}) - f_{\boldsymbol{w},b}(\boldsymbol{y})| &= \left|\sigma(\boldsymbol{w}\cdot\boldsymbol{x}+b) - \sigma(\boldsymbol{w}\cdot\boldsymbol{y}+b)\right| \\ &\leq \max|\sigma'| \ |\boldsymbol{w}\cdot\boldsymbol{x}+b-\boldsymbol{w}\cdot\boldsymbol{y}-b| \\ &= \frac{1}{4}|\boldsymbol{w}\cdot(\boldsymbol{x}-\boldsymbol{y})| \leq \frac{1}{4}\|\boldsymbol{w}\|\cdot\|\boldsymbol{x}-\boldsymbol{y}\| \\ &\leq \frac{1}{4}\|\boldsymbol{x}-\boldsymbol{y}\|. \end{aligned}$$

Con una banale applicazione del Teorema di Ascoli-Arzelà, si ottiene che ogni successione di funzioni output ammette una sottosuccessione convergente uniformemente ad una funzione continua; quest'ultima, data la uniforme convergenza, viene approssimata dalla rete con un grado di accuratezza arbitrario. ■

5.3 Teorema di Stone Weierstrass

Definizione 5.3.1. *Sia $\mathcal{F}$ un insieme di funzioni con lo stesso dominio, a valori in $\mathbb{R}$. $\mathcal{F}$ si dice un'algebra di funzioni reali se, per ogni $f, g \in \mathcal{F}$ e $c \in \mathbb{R}$:*

1. $f + g \in \mathcal{F}$;
2. $c\,f \in \mathcal{F}$;
3. $f\,g \in \mathcal{F}$.

Questa è la definizione di $\mathbb{R}$-algebra, specializzata in questo ambito.

Esempio 5.3.2. Sia $\mathcal{A}$ l'insieme di tutte le serie di Fourier finite su $[0, 2\pi]$:

$$\mathcal{A} \coloneqq \left\{ f(x) = c_0 + \sum_{k=1}^{N} (a_k \cos kx + b_k \operatorname{sen} kx) \mid c_0, a_k, b_k \in \mathbb{R}, N \in \mathbb{N} \right\}.$$

Ovviamente $\mathcal{A}$ è chiuso per combinazioni lineari. Utilizzando il fatto che

$$\cos(mx)\cos(nx) = \frac{1}{2}\left[\cos((m+n)x) + \cos((m-n)x)\right]$$

segue che $\mathcal{A}$ è anche chiuso per prodotti. Pertanto è una $\mathbb{R}$-algebra.

Definizione 5.3.3. *Sia $\mathcal{A}$ una $\mathbb{R}$-algebra di funzioni di dominio $K \subseteq \mathbb{R}^n$ a valori in $\mathbb{R}$.*

Si dice che $\mathcal{A}$ separa i punti se per ogni $x, y \in K$ esistono $f, g \in \mathcal{A}$ tali che

$$f(x) \neq f(y).$$

Esempio 5.3.4. Sia $\mathcal{A}$ l'insieme dei polinomi definiti su $[a,b]$:

$$\mathcal{A} \coloneqq \left\{ f|_{[a,b]} \mid f(x) = \sum_{k=0}^{n} c_k x^k \mid c_k \in \mathbb{R}, n \in \mathbb{N} \right\}.$$

Ovviamente $\mathcal{A}$ è una $\mathbb{R}$-algebra di funzioni, e inoltre separa i punti, poiché $f(x) = \mathrm{Id}_{[a,b]} \in \mathcal{A}$[2].

Inoltre $1 \in \mathcal{A}$, e pertanto $\mathcal{A}$ contiene tutte le funzioni costanti.

Teorema 5.3.5. Sia $K \subseteq \mathbb{R}^n$ compatto, e sia $\mathcal{A} \subseteq C(K)$[3] una $\mathbb{R}$-algebra. Se

1. $\mathcal{A}$ separa i punti di K;
2. $\mathcal{A}$ contiene le funzioni costanti;

allora $\mathcal{A}$ è un sottoinsieme denso di $C(K)$ munito della topologia indotta dalla metrica[4]:

$$\forall f, g \in C(K): \quad d(f,g) \coloneqq \max_{x \in K} |f(x) - g(x)|.$$

5.3.1 Corollari del teorema

Proposizione 5.3.6. Sia $[a,b] \subseteq \mathbb{R}$, e si consideri $C([a,b])$[3] munito della topologia indotta dalla metrica[4]:

$$\forall f, g \in C(K): \quad d(f,g) \coloneqq \max_{x \in [a,b]} |f(x) - g(x)|.$$

Sia:

$$\mathcal{A} \coloneqq \left\{ f|_{[a,b]} \mid f(x) = \sum_{k=0}^{n} a_k x^k; a_k \in \mathbb{R}, n \in \mathbb{N} \right\}.$$

Allora $\mathcal{A}$ è denso in $C([a,b])$, ovvero: per ogni $f : [a,b] \to \mathbb{R}$ continua e per ogni $\varepsilon > 0$ esiste $g \in \mathcal{A}$ tale che

$$\max_{x \in [a,b]} |f(x) - g(x)| < \varepsilon$$

Dimostrazione. $\mathcal{A}$ è una $\mathbb{R}$-algebra che separa i punti e contiene le funzioni costanti. ■

Proposizione 5.3.7. Sia $f : [a,b] \times [c,d] \to \mathbb{R}$ una funzione continua. Allora, per ogni $\varepsilon > 0$ esiste $N \geq 1$ ed esistono, per ogni $i = 1, \dots, N$, delle $g_i \in C([a,b])$[3] e $h_i \in C([c,d])$ tali che

$$\max_{\substack{x \in [a,b] \\ y \in [c,d]}} \left| f(x,y) - \sum_{i=1}^{N} g_i(x) h_i(y) \right| < \varepsilon$$

[2] Vedi "Funzione identità"
[3] Vedi "Classe C di una funzione"
[4] Tale massimo esiste per il Teorema di Weierstrass

Dimostrazione. Si consideri

$$\mathcal{A} \coloneqq \left\{ G(x,y) = \sum_{i=1}^{N} g_i(x)h_i(y) \mid g_i \in C([a,b]), h_i \in C([c,d]), N = 1,2,\dots \right\}$$

Si osserva facilmente che $\mathcal{A}$ è chiuso per somme, prodotti e prodotti per scalari, ovvero è una $\mathbb{R}$-algebra. Inoltre contiene le funzioni costanti.

Se $(x_0, y_0) \neq (x_1, y_1)$:

- se $x_0 \neq x_1$ allora $G(x,y) = x \cdot 1 \in \mathcal{A}$ separa i due punti;
- se $(y_0 \neq y_1)$ allora $G(x,y) = 1 \cdot y \in \mathcal{A}$ separa i due punti. ■

5.3.2 Applicazioni alle reti neurali

Proposizione 5.3.8. Per ogni funzione periodica $F : \mathbb{R} \to \mathbb{R}$ esiste una rete neurale che la approssima.

Dimostrazione. Sia T il periodo di F, ovvero $T \in \mathbb{R}$ tale che

$$\forall t \in \mathbb{R} \quad F(t+T) = F(t).$$

Si consideri

$$\mathcal{A} \coloneqq \left\{ f|_{[0,T]} \mid f(x) = a_0 + \sum_{j=1}^{n} a_j \cos\left(\frac{2\pi}{T} jx\right) + c_j \operatorname{sen}\left(\frac{2\pi}{T} jx\right); a_i, c_i \in \mathbb{R}, n = 1,2,\dots \right\}$$

Si noti che per ogni $f \in \mathcal{A}$

$$f(0) = f(T).$$

L'insieme $\mathcal{A} \subseteq C([0,T])$[5], ed inoltre è una $\mathbb{R}$-algebra[6]. Contiene tutte le funzioni costanti (basta porre $a_j = c_j = 0$ per ogni $j > 0$), e separa i punti, in quanto

$$g(x) = \cos\left(\frac{\pi}{T} x\right) \in \mathcal{A}$$

è una biiezione tra $[0,T]$ e $[0,1]$.

Dunque per il Teorema di Stone-Weierstrass $\mathcal{A}$ è denso in $C([0,T])$ ed in particolare, siccome $F \in C([0,T])$, per ogni $\varepsilon > 0$, esiste $N \in \mathbb{N}$ tale che

$$G(x) \coloneqq a_0 + \sum_{j=1}^{N} a_j \cos\left(\frac{2\pi}{T} jx\right) + c_j \operatorname{sen}\left(\frac{2\pi}{T} jx\right) \in \mathcal{A}$$
$$\max_{x\in[0,T]} \left|G(x) - F(x)\right| = \max_{x\in\mathbb{R}} \left|G(x) - F(x)\right| < \varepsilon$$

Si consideri una rete neurale fatta come segue:

[5] Vedi "Classe C di una funzione"
[6] Questo si vede facilmente seguendo l'Esempio di "Algebra di funzioni reali"

- input: $x \in \mathbb{R}$;
- un layer nascosto con N neuroni con funzione di attivazione cos, con peso dall'input w_j e bias b_j;
- il neurone di output con funzione di attivazione lineare, bias a_0 e pesi dall'hidden layer α_j.

Questa rete è rappresentata in Fig. 5.1 e produce output

$$y = a_0 + \sum_{j=1}^{N} \alpha_j \cos(w_j x + b_j).$$

Si dimostra che tale rete neurale è in grado di produrre come output $G(x)$, ovvero che, fissato un livello di precisione ε, esiste una rete neurale che produce come output una funzione periodica che approssima F con quel livello di precisione.

Infatti, ponendo

$$w_j \coloneqq \frac{2\pi}{T}$$

e α_j, b_j tali che

$$\begin{cases} a_j = \alpha_j \cos b_j \\ c_j = -\alpha_j \operatorname{sen} b_j \end{cases}$$

si ottiene che, detto per semplicità di notazione $\nu \coloneqq 2\pi/T$

$$\begin{aligned} y &= a_0 + \sum_{j=1}^{N} \alpha_j \cos(w_j x + b_j) \\ &= a_0 + \sum_{j=1}^{N} \alpha_j \cos(\nu j x) \cos b_j - \alpha_j \operatorname{sen}(\nu j x) \operatorname{sen} b_j \\ &= a_0 + \sum_{j=1}^{N} a_j \cos(\nu j x) + c_j \operatorname{sen}(\nu j x) \\ &= a_0 + \sum_{j=1}^{N} a_j \cos\left(\frac{2\pi}{T} j x\right) + c_j \operatorname{sen}\left(\frac{2\pi}{T} j x\right) = G(x). \end{aligned}$$

■

5.4 Teoremi Tauberiani di Wiener

Si consideri l'operatore funzionale

$$T_\theta : f(x) \mapsto f(x - \theta).$$

Teorema 5.4.1. Sia $f \in L^1(\mathbb{R})$[7]. Lo span di $\{T_\theta f \mid \theta \in \mathbb{R}\}$ è denso in $L^1(\mathbb{R})$ se e solo se la trasformata di Fourier:

$$\forall \xi \in \mathbb{R} : \qquad \hat{f}(\xi) \neq 0$$

[7] Vedi "Spazi Lp"

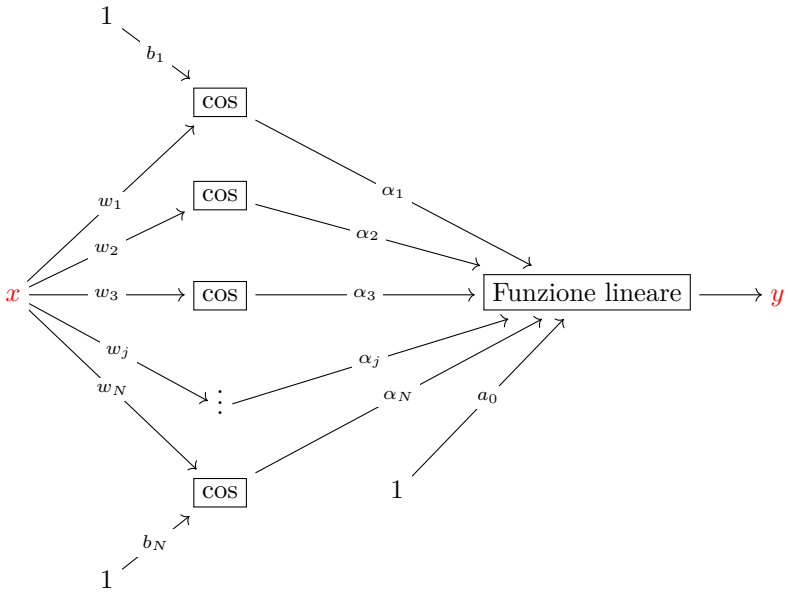


Figura 5.1: La rete neurale considerata

Teorema 5.4.2. Sia $f \in L^2(\mathbb{R})$[7]. Lo span di $\{T_\theta f \mid \theta \in \mathbb{R}\}$ è denso in $L^2(\mathbb{R})$ se e solo se gli zeri della trasformata di Fourier[8]

$$\hat{f}(\xi) \neq 0 \text{ q.o. } \xi \in \mathbb{R}$$

ovvero per la misura di Lebsegue μ:

$$\mu\left(\left\{\xi \in \mathbb{R} \mid \hat{f}(\xi) = 0\right\}\right) = 0$$

5.4.1 Applicazioni dei Teoremi Tauberiani di Wiener al Machine Learning

1. Sia $g \in L^1(\mathbb{R})$, e sia $f \in L^1(\mathbb{R})$ tale che

   $$\forall \xi \in \mathbb{R}: \qquad \hat{f}(\xi) \neq 0$$

   (ovvero f soddisfa le ipotesi del Teorema 5.4.1)

   Allora, per ogni $\varepsilon > 0$ esiste $N \in \mathbb{N}$ ed esistono, per $j = 1, \ldots, N$, degli $\alpha_j, \theta_j \in \mathbb{R}$ tali che

   $$G(x) \coloneqq \sum_{j=1}^{N} \alpha_j f(x + \theta_j)$$

   $$\int_{\mathbb{R}} |g(x) - G(x)| \, \mathrm{d}x < \varepsilon.$$

[8] Vedi "Proprietà vera quasi ovunque"

La funzione $G(x)$ è l'output di una rete neurale con un layer nascosto di N neuroni (e neurone di output lineare), dove i neuroni del layer nascosto hanno funzione di attivazione f.

2. Sia $g \in L^2(\mathbb{R})$, e sia $f \in L^2(\mathbb{R})$ tale che

$$\mu\left(\{\xi \in \mathbb{R} \mid f(\xi) = 0\}\right) = 0$$

(ovvero f soddisfa le ipotesi del Teorema 5.4.2)

Allora, per ogni $\varepsilon > 0$ esiste $N \in \mathbb{N}$ ed esistono, per $j = 1, \dots, N$, degli $\alpha_j, \theta_j \in \mathbb{R}$ tali che

$$G(x) \coloneqq \sum_{j=1}^{N} \alpha_j f(x + \theta_j)$$
$$\int_{\mathbb{R}} (g(x) - G(x))^2 \, \mathrm{d}x < \varepsilon.$$

La funzione $G(x)$ è l'output di una rete neurale con un layer nascosto di N neuroni (e neurone di output lineare), dove i neuroni del layer nascosto hanno funzione di attivazione f.

Si noti che le seguenti funzioni di attivazione soddisfano le ipotesi dei Teoremi 5.4.1 e 5.4.2.

Doppio esponenziale. Sia $f(x) = e^{-\lambda|x|}$, con $\lambda > 0$.

$$\int_{\mathbb{R}} |f(x)| \, \mathrm{d}x = 2 \int_0^{+\infty} e^{-\lambda x} \, \mathrm{d}x = -\frac{2}{\lambda}[e^{-\lambda x}]_0^{+\infty} = \frac{2}{\lambda} < \infty$$

e dunque $f \in L^1(\mathbb{R})$. Inoltre

$$\int_{\mathbb{R}} (f(x))^2 \, \mathrm{d}x = \int_{\mathbb{R}} e^{-2\lambda|x|} \, \mathrm{d}x = \frac{1}{\lambda} < \infty$$

e pertanto $f \in L^2(\mathbb{R})$.

Si calcola la trasformata di Laplace:

$$\begin{aligned}
\hat{f}(\xi) &= \int_{\mathbb{R}} e^{-2\pi i \xi x} e^{-\lambda|x|} \, \mathrm{d}x \\
&= \int_{-\infty}^{0} e^{-2\pi i \xi x} e^{-\lambda|x|} \, \mathrm{d}x + \int_0^{+\infty} e^{-2\pi i \xi x} e^{-\lambda|x|} \, \mathrm{d}x \\
&= \int_{-\infty}^{0} e^{-2\pi i \xi x} e^{\lambda x} \, \mathrm{d}x + \int_0^{+\infty} e^{-2\pi i \xi x} e^{-\lambda x} \, \mathrm{d}x \\
&= \int_0^{+\infty} e^{2\pi i \xi x} e^{-\lambda x} \, \mathrm{d}x + \int_0^{+\infty} e^{-2\pi i \xi x} e^{-\lambda x} \, \mathrm{d}x \\
&= \int_0^{+\infty} e^{-(-2\pi i \xi + \lambda)x} \, \mathrm{d}x + \int_0^{+\infty} e^{-(2\pi i \xi + \lambda)x} \, \mathrm{d}x \\
&= \frac{1}{-2\pi i \xi + \lambda} + \frac{1}{2\pi i \xi + \lambda} = \frac{(2\pi i \xi + \lambda) + (-2\pi i \xi + \lambda)}{(2\pi i \xi + \lambda) \cdot (-2\pi i \xi + \lambda)} = \frac{2\lambda}{4\pi^2 \xi^2 + \lambda^2}
\end{aligned}$$

e quindi per ogni $\xi \in \mathbb{R}$ si ha che $\hat{f}(\xi) \neq \emptyset$.

Dunque f soddisfa le condizioni dei Teoremi 5.4.1 e 5.4.2.

Potenziale di Laplace. Sia $f(x) = \frac{1}{a^2+x^2}$ con $a > 0$.

$$\int_{\mathbb{R}} \left| \frac{1}{a^2+x^2} \right| \mathrm{d}x = \int_{\mathbb{R}} \frac{1}{a^2+x^2}\,\mathrm{d}x = \left. \frac{\arctan(x/a)}{x} \right|_{-\infty}^{\infty} = \frac{\pi}{a} < \infty$$

e dunque $f \in L^1(\mathbb{R})$. Inoltre $f \in L^2(\mathbb{R})$.

Si ha la seguente trasformata di Laplace:

$$\hat{f}(\xi) = \int_{\mathbb{R}} e^{-2\pi i \xi x} \cdot \frac{1}{a^2+x^2}\,\mathrm{d}x = \frac{\pi}{a} e^{-2\pi a|\xi|}$$

e quindi per ogni $\xi \in \mathbb{R}$ si ha che $\hat{f}(\xi) \neq \emptyset$.

Dunque f soddisfa le condizioni dei Teoremi 5.4.1 e 5.4.2.

Gaussiana. Sia $f(x) = e^{-ax^2}$, con $a > 0$.

$$\begin{aligned}\int_{\mathbb{R}} |e^{-ax^2}|\,\mathrm{d}x = \int_{\mathbb{R}} e^{-ax^2}\,\mathrm{d}x &= \frac{1}{\sqrt{a}} \int_{\mathbb{R}} e^{-(\sqrt{a}x)^2} \sqrt{a}\,\mathrm{d}x \\ &= \frac{1}{\sqrt{a}} \int_{\mathbb{R}} e^{-u^2}\,\mathrm{d}u = \frac{\sqrt{\pi}}{\sqrt{a}} < \infty\end{aligned}$$

e pertanto $f \in L^1(\mathbb{R})$ e $f \in L_2^{\mathbb{R}}$.

Si calcola la trasformata di Laplace:

$$\begin{aligned}\hat{f}(\xi) &= \int_{\mathbb{R}} e^{-2\pi i x \xi} e^{-ax^2}\,\mathrm{d}x = \int_{\mathbb{R}} \exp\left(-(2\pi i \xi)x + -ax^2\right)\mathrm{d}x \\ &= \int_{\mathbb{R}} \exp\left(-\frac{\pi^2\xi^2}{a} + \frac{\pi^2\xi^2}{a} + -(2\pi i\xi)x + -ax^2\right)\mathrm{d}x \\ &= \exp\left(-\frac{\pi^2\xi^2}{a}\right) \int_{\mathbb{R}} \exp\left[-\left(-\frac{\pi^2\xi^2}{a} + 2\pi i\xi x + ax^2\right)\right]\mathrm{d}x \\ &= \exp\left(-\frac{\pi^2\xi^2}{a}\right) \int_{\mathbb{R}} \exp\left[-\left(\frac{i\pi\xi}{\sqrt{a}} + x\sqrt{a}\right)^2\right]\mathrm{d}x \\ &= \exp\left(-\frac{\pi^2\xi^2}{a}\right) \frac{\sqrt{\pi}}{\sqrt{a}}\end{aligned}$$

e quindi per ogni $\xi \in \mathbb{R}$ si ha che $\hat{f}(\xi) \neq \emptyset$.

Dunque f soddisfa le condizioni dei Teoremi 5.4.1 e 5.4.2.

Capitolo 6

Apprendimento con input unidimensionale

6.1 Risultati preliminari

Proposizione 6.1.1. Sia $0 = x_0 < x_1 < \cdots < x_N = 1$ una partizione di $[0,1]$, e sia[1]

$$c(x) = \sum_{i=0}^{N-1} \alpha_i \chi_{[x_i, x_{i+1})}$$

Allora $c(x)$ può essere scritto come combinazione lineare di Funzioni di Heaviside:

$$\begin{aligned} \chi_{[x_i, x_{i+1})} &= H(x - x_i) - H(x - x_{i+1}); \\ c(x) &= \sum_{i=0}^{N-1} \big(\alpha_i H(x - x_i) - \alpha_i H(x - x_{i+1})\big) \\ &= \sum_{i=0}^{N} c_i H(x - x_i). \end{aligned}$$

Proposizione 6.1.2. Se $c(x)$ è come sopra, segue che la derivata distribuzionale di c, $c'(x)$, è[2]

$$c'(x) = \sum_{i=0}^{N} c_i H'(x - x_i) = \sum_{i=0}^{N} c_i \delta(x - x_i) = \sum_{i=0}^{N} c_i \delta_{x_i}(x)$$

Proposizione 6.1.3. Sia $\varphi : \mathbb{R} \to \mathbb{R}$ tale che

1. φ sia crescente;

[1] Con χ_A si intende la funzione caratteristica di A.
[2] Con $\delta(x)$ si intende la Delta di Dirac, e con $\delta_a(x)$ si intende la Delta di Dirac centrata in a.

2. $(\lim_{x\to+\infty}\varphi(x)) - (\lim_{x\to-\infty}\varphi(x)) = 1$;
3. φ sia derivabile, con $|\varphi'(x)|$ limitata.

Sia $\varphi_\varepsilon(x) \coloneqq \varphi(x/\varepsilon)$ e sia μ_ε la misura con densità φ'_ε:

$$\mathrm{d}\mu_\varepsilon(x) = \varphi'_\varepsilon(x)\,\mathrm{d}x.$$

Allora[3] $\varphi_\varepsilon \to \delta$ in senso debole, per $\varepsilon \to 0$, ovvero, per ogni $g \in C^\infty(\mathbb{R})$[4] a supporto compatto:

$$\lim_{\varepsilon\to 0}\int_{\mathbb{R}} g(x)\,\mathrm{d}\mu_\varepsilon(x) = \int_{\mathbb{R}} g(x)\delta(x)\,\mathrm{d}x.$$

Dimostrazione. Sia $g \in C^\infty(\mathbb{R})$ a supporto compatto.

$$\begin{aligned}\int_{\mathbb{R}} g(x)\,\mathrm{d}\mu_\varepsilon(x) &= \int_{\mathbb{R}} g(x)\,\varphi'_\varepsilon(x)\,\mathrm{d}x \\ &= \int_{\mathbb{R}} g(x)\,\varphi'\left(\frac{x}{\varepsilon}\right)\frac{1}{\varepsilon}\,\mathrm{d}x \\ &= \int_{\mathbb{R}} g(\varepsilon y)\varphi'(y)\,\mathrm{d}y \qquad \text{ponendo } y = \frac{x}{\varepsilon}\end{aligned}$$

Dunque, passando al limite:

$$\begin{aligned}\lim_{\varepsilon\to 0}\int_{\mathbb{R}} g(x)\,\mathrm{d}\mu_\varepsilon(x) &= \lim_{\varepsilon\to 0}\int_{\mathbb{R}} g(\varepsilon y)\varphi'(y)\,\mathrm{d}y \\ &= \int_{\mathbb{R}} g(0)\varphi'(y)\,\mathrm{d}y \qquad \text{per Teorema di Convergenza Dominata} \\ &= g(0)\int_{\mathbb{R}}\varphi'(y)\,\mathrm{d}y = g(0)\cdot 1 \qquad \text{per l'ipotesi 2.}\end{aligned}$$

Siccome $g(0) = \int_{\mathbb{R}} g(x)\delta(x)\,\mathrm{d}x$, si ha la tesi.

È necessario ancora dimostrare che sia possibile applicare il teorema di convergenza dominata (questo si può fare solo perché g ha supporto compatto, e quindi l'integrale viene svolto su un dominio limitato):

$$|g(\varepsilon y)\,\varphi'(y)| \le \|g\|_\infty\,|\varphi'(y)| < M \in \mathbb{R}$$

poiché g ha supporto compatto ed è C^∞, mentre $|\varphi'(y)|$ è limitato per ipotesi. ■

Teorema 6.1.4. Siano $(M,d),(N,\partial)$ due spazi metrici, e sia $f : M \to N$ continua.

Se M è compatto allora f è uniformemente continua.

Corollario 6.1.5. In particolare, se $g : [a,b] \to \mathbb{R}$ è continua, allora è uniformemente continua[5].

[3] δ è la Delta di Dirac
[4] Vedi Classe C di una funzione
[5] Perché $[a,b]$ è compatto per il Teorema di Heine-Borel.

6.2 Funzioni continue

6.2.1 One Hidden Layer Perceptron Network

Teorema 6.2.1. Per ogni funzione $g \in C[0,1]$[6] e per ogni $\varepsilon > 0$ esistono $0 = x_0 < x_1 < \cdots < x_N = 1$ ed esistono $c_i \in \mathbb{R}$ tali che, posta[7]

$$c(x) = \sum_{i=0}^{N-1} c_i H(x - x_i)$$

si ha che

$$\forall x \in [0,1] : \quad |g(x) - c(x)| < \varepsilon$$

Dimostrazione. Sia $\varepsilon > 0$ fissato. Siccome $[0,1]$ è compatto, allora g è uniformemente continua. Sia $\delta > 0$ che soddisfi la condizione di uniforme continuità.

Sia $0 = x_0 < \cdots < x_N = 1$ la equipartizione di $[0,1]$ tale che $|x_{i+1} - x_i| < \delta$, e siano i $c_0, \ldots, c_{N-1}$ tali che

$$\begin{aligned}
g(x_0) &= c_0 & c_0 &= g(x_0) \\
g(x_1) &= c_0 + c_1 & c_1 &= g(x_1) - g(x_0) \\
&\vdots \\
g(x_i) &= c_0 + c_1 + \cdots + c_i & c_i &= g(x_i) - g(x_{i-1}) \\
&\vdots \\
g(x_{N-1}) &= c_0 + c_1 + \cdots + c_{N-1} & c_{N-1} &= g(x_{N-1}) - g(x_{N-2}).
\end{aligned}$$

Questo definisce la funzione $c(x)$.

Sia ora $u \in [0,1]$. Allora esiste $k < N$ tale che $u \in [x_k, x_{k+1})$ e tale che $|u - x_k| < \delta$. Si osservi quindi che

$$H(u - x_j) = \begin{cases} 1 & j \leq k \\ 0 & j > k \end{cases}$$

e pertanto $c(u)$ vale:

$$c(u) = \sum_{i=0}^{N-1} c_i \, H(u - x_i) = \sum_{i=0}^{k} c_i = g(x_k).$$

È possibile ora calcolare la distanza da g:

$$\begin{aligned}
|g(u) - c(u)| &= |g(u) - g(x_k) + g(x_k) - c(u)| \\
&\leq |g(u) - g(x_k)| + \cancel{|g(x_k) - c(u)|} = |g(u) - g(x_k)| < \varepsilon
\end{aligned}$$

dove l'ultima condizione deriva dall'uniforme continuità di g. Per l'arbitrarietà di u questo dimostra la tesi. ■

[6] Vedi "Classe C di una funzione"
[7] Questa è una Funzione costante a tratti

Osservazione. La funzione $c(x)$ di cui sopra è la funzione output di una rete neurale con un layer nascosto:

- i pesi dall'input ai neuroni nascosti sono $w_i = 1$;
- ci sono N neuroni nascosti con funzione di attivazione $H(x)$ e bias $\theta_i = -x_i$;
- i pesi dai neuroni nascosti al neurone di output sono i c_i;
- il neurone di output è lineare.

6.2.2 One Hidden Layer Sigmoidal Network

Teorema 6.2.2. Sia $\varphi : \mathbb{R} \to \mathbb{R}$ tale che

1. φ sia crescente;
2. $(\lim_{x\to+\infty} \varphi(x)) - (\lim_{x\to-\infty} \varphi(x)) = 1$;
3. φ sia derivabile, con $|\varphi'(x)|$ limitata.

Sia $g \in C[0,1]$[8]. Allora per ogni $\varepsilon > 0$ esiste $N \in \mathbb{N}$ ed esistono, per ogni $i = 0, \dots, N$ degli $c_i, \theta_i, w \in \mathbb{R}$ tali che

$$\forall x \in [0,1] \qquad \left| g(x) - \sum_{i=1}^{N} c_i \varphi(w x_i + \theta_i) \right| < \varepsilon$$

Osservazione. In particolare, tutte le funzioni di attivazione sigmoidali φ rispettano le ipotesi.

Dimostrazione. Si divide la dimostrazione in fasi:

1. Approssimazione di $g(x)$ con una funzione costante a tratti $c(x)$ (di coefficienti arbitrariamente piccoli);
2. Costruzione di una versione "smoother" $c_\alpha(x)$ di $c(x)$ tramite la convoluzione;
3. Aprrossimazione di $c(x)$ con $c_\alpha(x)$ (usando l'Approssimazione della misura di Dirac);
4. Approssimazione di $g(x)$ con $c_\alpha(x)$. (banale disuguaglianza triangolare)

Vedi Theorem 8.3.1 di [**calinDeepLearningArchitectures2020**] ■

6.2.3 One Hidden Layer ReLU Network

Proposizione 6.2.3. Sia $g : [a,b] \to \mathbb{R}$ una funzione continua. Allora, per ogni $\varepsilon > 0$ esiste una partizione equidistante di $[a,b]$:

$$a = x_0 < x_1 < \cdots < x_N = b$$

tale che la funzione lineare a tratti $g_\varepsilon : [a,b] \to \mathbb{R}$ che passa per i punti $\big(x_i, g(x_i)\big)$ per $i = 0, \dots, N$, soddisfa

$$\forall x \in [a,b] : \quad |g(x) - g_\varepsilon(x)| < \varepsilon.$$

[8] Vedi "Classe C di una funzione"

Dimostrazione.

Parte 1. Sia $\varepsilon > 0$ fissato. Siccome $[a, b]$ è compatto, allora g è uniformemente continua. Sia $\delta > 0$ che soddisfi la condizione di uniforme continuità per $\varepsilon' = \varepsilon/2$.

Sia N sufficientemente grande affinché $\frac{b-a}{N} < \delta$, e si definisca la partizione equidistante

$$x_j = a + \frac{b-a}{N}\, j$$

e la funzione lineare a tratti, per $i = 1, \ldots,$:

$$g_\varepsilon(x) = g(x_{i-1}) + \frac{g(x_i) - g(x_{i-1})}{x_i - x_{i-1}}(x - x_{i-1}), \qquad \forall x \in [x_{i-1}, x_i]$$

Parte 2. Sia $x \in [a, b]$ fissato. Allora $x \in [x_{k-1}, x_k]$ per qualche k.

$$\begin{aligned} |g(x) - g_\varepsilon(x)| &\leq |g(x) - g(x_{k-1})| + |g(x_{k-1}) - g_\varepsilon(x)| \\ &\leq \frac{\varepsilon}{2} + |g(x_{k-1}) - g_\varepsilon(x)| \\ &= \frac{\varepsilon}{2} + |g_\varepsilon(x_{k-1}) - g_\varepsilon(x)| \\ &\leq \frac{\varepsilon}{2} + |g_\varepsilon(x_{k-1}) - g_\varepsilon(x_k)| \\ &= \frac{\varepsilon}{2} + |g(x_{k-1}) - g(x_k)| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon \end{aligned}$$

■

Teorema 6.2.4. Si consideri la funzione di attivazione $\mathrm{ReLU}(x) = x\,H(x)$[9]. Allora per ogni $g \in C[0, 1]$[10] esiste $N \in \mathbb{N}$ ed esistono, per $i = 0, \ldots, N-1$ degli $\alpha_i, \theta_i, \beta \in \mathbb{R}$ tali che, detta

$$G(x) \coloneqq \beta + \sum_{i=0}^{N-1} \alpha_i \,\mathrm{ReLU}(x + \theta_i)$$

si ha che

$$\forall x \in [0, 1] \qquad |g(x) - G(x)| < \varepsilon.$$

6.2.4 One Hidden Layer softplus Network

Lemma 6.2.5. La funzione di attivazione *softplus* $\mathrm{sp}(x)$ è data dalla convoluzione:[11]

$$\mathrm{sp}(x) = (\mathrm{ReLU} * K)(x) = \int_{-\infty}^{+\infty} \mathrm{ReLU}(\tau) K(x - \tau)\, \mathrm{d}\tau$$

dove $K(x) = \frac{1}{(1+e^x)(1+e^{-x})} = \sigma'(x)$[12].

[9] Dove $H(x)$ è la Funzione di Heaviside
[10] Vedi "Classe C di una funzione"
[11] Vedi la funzione ReLU
[12] Dove $\sigma(x)$ è la Funzione Logistica.

Lemma 6.2.6. Sia $K(x) \coloneqq \frac{1}{(1+e^x)(1+e^{-x})}$. Allora

1. $K(x)$ è una densità di probabilità simmetrica;
2. Sia $K_\alpha \coloneqq \frac{1}{\alpha}K(x/\alpha)$ e si consideri la misura μ_α tale che
$$\mathrm{d}\mu_\alpha \coloneqq K_\alpha(x)\,\mathrm{d}x.$$
Allora $\int_{-\infty}^{+\infty} K_\alpha(x)\,\mathrm{d}x = 1$ e $\mu_\alpha \to \delta$[13] in senso debole.

Lemma 6.2.7. Si definisca ora la funzione softplus scalata:
$$\varphi_\alpha(x) \coloneqq \alpha\,\mathrm{sp}\left(\frac{x}{\alpha}\right) = \alpha\ln(1+e^{x/\alpha})$$
Si ha che
$$\varphi''_\alpha(x) = \frac{1}{\alpha}\,\mathrm{sp}''(x/\alpha) = \frac{1}{\alpha}K(x/\alpha) = K_\alpha(x)$$
poiché $\mathrm{sp}'(x) = \sigma(x)$ e $\sigma'(x) = K(x)$.

Lemma 6.2.8. Si consideri la convoluzione $G_\alpha \coloneqq G * K_\alpha$, dove
$$G(x) = \sum_{j=0}^{N-1} \alpha_j\,\mathrm{ReLU}(x - x_j) + \beta$$
per qualche $N \in \mathbb{N}$, $\alpha_j, x_j, \beta \in \mathbb{R}$.

Allora esistono $c_j, w, \theta_j \in \mathbb{R}$, che dipendono dagli α_j, x_j tali che
$$G_\alpha(x) = \sum_{j=0}^{N-1} c_j\,\mathrm{sp}(wx - \theta_j) + \beta$$

Lemma 6.2.9. Per ogni $\varepsilon > 0$ esiste $\eta > 0$ tale per cui, se $\alpha < \eta$ allora
$$\forall x \in [0,1] \qquad |G(x) - G_\alpha(x)| < \varepsilon$$
ovvero G_α converge uniformemente a G su $[0,1]$.

Dimostrazione. Banale applicazione del Teorema del Dini dopo aver notato che
$$\varphi_\alpha(x) \to \mathrm{ReLU}(x).$$
puntualmente, e $\varphi_\alpha < \varphi_\beta$ se $\alpha > \beta$. ■

Teorema 6.2.10. Sia $g \in C[0,1]$[14]. Allora per ogni $\varepsilon > 0$ esiste $N \in \mathbb{N}$ ed esistono, per ogni $i = 0, \dots, N-1$, degli $c_j, w, \theta_j, \beta \in \mathbb{R}$ tali che
$$\forall x \in [0,1] \qquad \left| g(x) - \sum_{j=0}^{N-1} c_j\,\mathrm{sp}(wx - \theta_j) - \beta \right| < \varepsilon.$$

[13] δ è la Delta di Dirac
[14] Vedi "Classe C di una funzione"

Capitolo 7

Universal Approximation

Definizione 7.0.1. *Sia $(\mathcal{S}, d)$ uno spazio metrico di funzioni. Una rete neurale è un approssimatore universale per $\mathcal{S}$ se lo spazio $\mathcal{U}$ delle funzioni output della r.n. è denso in $\mathcal{S}$.*

Lo spazio $\mathcal{U}$ si dice spazio di approssimazione dello spazio target $\mathcal{S}$.

7.1 Teoremi di approssimazione universale

Corollario 7.1.1. Sia $(\mathcal{X}, \|\cdot\|)$ uno spazio vettoriale normato, e sia $\mathcal{U} \subseteq \mathcal{X}$ un sottospazio vettoriale non denso.

Allora esiste un funzionale lineare limitato

$$L : \mathcal{X} \to \mathbb{R}$$

tale che $L \neq 0$ e la restrizione $L|_{\mathcal{U}} = 0$.

7.1.1 Funzioni continue

Definizione 7.1.2. *Sia $\mathcal{M}$ la famiglia delle misure di Baire 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 se per ogni $\mu \in \mathcal{M}$:

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

Lemma 7.1.3. Sia $I_n := [0,1]^n$, e si consideri $C(I_n)$[1] dotato della topologia indotta dalla metrica

$$d(f,g) := \sup_{x \in I_n} |f(x) - g(x)| = \max_{x \in I_n} |f(x) - g(x)|.$$

[1] Vedi "Classe C di una funzione"

Sia $\mathcal{M}$ la famiglia delle misure di Baire per $\mathbb{R}^n$ sul cubo $I^n\mathbb{R}$, finite, con segno e regolari.

Sia $\mathcal{U} \subseteq C(I_n)$ un sottospazio vettoriale non denso. Allora esiste una misura $\mu \in \mathcal{M}$ tale che

$$\forall h \in \mathcal{U} : \qquad \int_{I_n} h(x)\,\mathrm{d}\mu = 0$$

Proposizione 7.1.4. Sia σ una funzione discriminatoria e continua. Allora l'insieme

$$\left\{ G(x) = \sum_{i=1}^{N} \alpha_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x} + \theta_j) \mid N \in \mathbb{N}, \boldsymbol{w}_j \in \mathbb{R}^n, \alpha_j, \theta_j \in \mathbb{R} \right\}$$

è denso in $C(I_n)$[1] dotato della topologia indotta dalla metrica

$$d(f,g) \coloneqq \sup_{x \in I_n} |f(x) - g(x)| = \max_{x \in I_n} |f(x) - g(x)|.$$

Teorema 7.1.5. Sia $\sigma : \mathbb{R} \to [0,1]$ una funzione sigmoidale. Allora l'insieme

$$\left\{ G(x) = \sum_{i=1}^{N} \alpha_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x} + \theta_j) \mid N \in \mathbb{N}, \boldsymbol{w}_j \in \mathbb{R}^n, \alpha_j, \theta_j \in \mathbb{R} \right\}$$

è denso in $C(I_n)$[1] dotato della topologia indotta dalla metrica

$$d(f,g) \coloneqq \sup_{x \in I_n} |f(x) - g(x)| = \max_{x \in I_n} |f(x) - g(x)|.$$

Il teorema continua a valere anche in questa forma:

Teorema 7.1.6. Sia $\sigma : \mathbb{R} \to [0,1]$ una funzione sigmoidale. Allora l'insieme

$$\left\{ G(x) = \sum_{i=1}^{N} \alpha_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x} + \theta_j) \mid N \in \mathbb{N}, \boldsymbol{w}_j \in \mathbb{R}^n, \alpha_j, \theta_j \in \mathbb{R} \right\}$$

è denso in $C(K)$ dotato della topologia indotta dalla metrica

$$d(f,g) \coloneqq \sup_{x \in K} |f(x) - g(x)| = \max_{x \in K} |f(x) - g(x)|,$$

per ogni K compatto di $\mathbb{R}^n$.

Una rete $\Sigma\Pi$ contiene sia neuroni Σ che neuroni Π.

Teorema 7.1.7. Sia $I_n \coloneqq [0,1]^n$, e si consideri $C(I_n)$[2] dotato della topologia indotta dalla metrica

$$d(f,g) \coloneqq \sup_{x \in I_n} |f(x) - g(x)| = \max_{x \in I_n} |f(x) - g(x)|.$$

[2] Vedi "Classe C di una funzione"

Sia $\varphi : \mathbb{R} \to \mathbb{R}$ una qualsiasi funzione continua non costante. Allora le funzioni nella forma

$$G(x) = \sum_{k=1}^{M} \beta_k \prod_{j=1}^{N_k} \varphi(\boldsymbol{w}_{jk} \cdot \boldsymbol{x} + \theta_{jk}), \qquad G : I_n \to \mathbb{R}$$

con $\boldsymbol{w}_{jk} \in \mathbb{R}^n$, $\beta_j, \theta_{jk} \in \mathbb{R}$, $M, N_k \in \mathbb{N}$, sono dense in $C(I_n)$

7.1.2 Funzioni L^2

Definizione 7.1.8. *Una funzione $\sigma : \mathbb{R} \to \mathbb{R}$ si dice discriminatoria in senso L^2 se:*

1. $\forall x \in \mathbb{R}\ 0 \leq \sigma(x) \leq 1$;
2. *se $g \in L^2(I_n)$[3] e*

$$\forall \boldsymbol{w} \in \mathbb{R}^n\ \forall \theta \in \mathbb{R} : \qquad \int_{I_n} \sigma(\boldsymbol{w} \cdot \boldsymbol{x} + \theta) g(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = 0$$

allora $g = 0$

Esempio 7.1.9. La funzione logistica $\sigma(x) = \dfrac{1}{1 + e^{-x}}$ è discriminatoria in senso L^2.

Proposizione 7.1.10. Sia $\sigma : \mathbb{R} \to \mathbb{R}$ una funzione discriminatoria in senso L^2. Allora l'insieme

$$\left\{ G(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x} + \theta_j) \mid N \in \mathbb{N}, \boldsymbol{w}_j \in \mathbb{R}^n, \alpha_j, \theta_j \in \mathbb{R} \right\}$$

è denso in $L^2(I_n)$[4].

Lemma 7.1.11. Sia $g \in L^1(I_n)$[5]. Se per ogni $\boldsymbol{w} \in \mathbb{R}^n$ e per ogni $\theta \in \mathbb{R}$, detto $\mathcal{H}_{\boldsymbol{w},\theta} := \left\{ x \in I_n \mid \boldsymbol{w} \cdot \boldsymbol{x} + \theta > 0 \right\}$ il semispazio, si ha che

$$\int_{\mathcal{H}_{\boldsymbol{w},\theta}} g(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = 0$$

allora $g = 0$ q.o.

---

[3] Con I_n si indica il cubo

$$I_n = [0,1] \times \cdots \times [0,1] = [0,1]^n.$$

Si vedano "Spazi Lp"

[4] Con I_n si indica il cubo

$$I_n = [0,1] \times \cdots \times [0,1] = [0,1]^n.$$

Si vedano "Spazi Lp"

[5] Con I_n si indica il cubo

$$I_n = [0,1] \times \cdots \times [0,1] = [0,1]^n.$$

Si vedano "Spazi Lp"

7.1.3 Funzioni L^1

Definizione 7.1.12. *Una funzione $\sigma : \mathbb{R} \to \mathbb{R}$ si dice discriminatoria in senso L^1 se:*

1. *σ è misurabile e limitata;*
2. *σ è sigmoidale;*
3. *se $g \in L^\infty(I_n)$[3] e*

$$\forall \boldsymbol{w} \in \mathbb{R}^n \ \forall \theta \in \mathbb{R} : \qquad \int_{I_n} \sigma(\boldsymbol{w} \cdot \boldsymbol{x} + \theta) g(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = 0$$

allora $g = 0$

Proposizione 7.1.13. Sia $\sigma : \mathbb{R} \to \mathbb{R}$ una funzione discriminatoria in senso L^1. Allora l'insieme

$$\left\{ G(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x} + \theta_j) \mid N \in \mathbb{N}, \boldsymbol{w}_j \in \mathbb{R}^n, \alpha_j, \theta_j \in \mathbb{R} \right\}$$

è denso in $L^1(I_n)$[6].

Siccome ogni funzione sigmoidale, misurabile e limitata è discriminatoria in senso L^1, segue il corollario.

Corollario 7.1.14. Sia $\sigma : \mathbb{R} \to \mathbb{R}$ una funzione sigmoidale, misurabile e limitata. Allora l'insieme

$$\left\{ G(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x} + \theta_j) \mid N \in \mathbb{N}, \boldsymbol{w}_j \in \mathbb{R}^n, \alpha_j, \theta_j \in \mathbb{R} \right\}$$

è denso in $L^1(I_n)$[6].

7.1.4 Funzioni Misurabili

Sia $\mathcal{M}^n$ lo spazio delle funzioni Borel-misurabili:

$$\mathcal{M}^n \coloneqq \left\{ f : \mathbb{R}^n \to \mathbb{R} \mid f \text{ Borel misurabile} \right\}.$$

Se $\mathbf{Bor}(\mathbb{R}^n)$ e $\mathbf{Bor}(\mathbb{R})$ denotano le algebre di Borel di $\mathbb{R}^n$ e di $\mathbb{R}$, rispettivamente, allora

$$f \in \mathcal{M}^n \quad \iff \quad \forall A \in \mathbf{Bor}(\mathbb{R}) : \ f^{-1}(A) \in \mathbf{Bor}(\mathbb{R}^n).$$

Per definire una distanza d su $\mathcal{M}^n$ si deve scegliere $\mu : \mathbf{Bor}(\mathbb{R}^n) \to [0, 1]$ misura di probabilità su $\left(\mathbb{R}^n, \mathbf{Bor}(\mathbb{R}^n)\right)$, con $\mu(\mathbb{R}^n) = 1$.

---

[6] Con I_n si indica il cubo

$$I_n = [0, 1] \times \cdots \times [0, 1] = [0, 1]^n.$$

Si vedano "Spazi Lp"

Quindi $(\mathcal{M}^n, d_\mu)$ è uno spazio metrico, dove si è definita la distanza:

$$d_\mu(f,g) = \inf\left\{\varepsilon > 0 \mid \mu\left(\{x \in \mathbb{R}^n : |f(x) - g(x)| > \varepsilon\}\right) < \varepsilon\right\}.$$

Pertanto $\mathcal{M}^n$ è dotato di una topologia indotta dalla metrica. Si ricorda che lo spazio delle funzioni misurabili $\mathcal{M}^n$ è dotato di una distanza d_μ per ogni misura di probabilità μ su $\mathbb{R}^n$[7].

Lemma 7.1.15. Per ogni $j \in \mathbb{N}$ sia $f_j \in \mathcal{M}^n$, e sia $g \in \mathcal{M}^n$. Sono fatti equivalenti:

1. $d_\mu(f_j, g) \to 0$ per $j \to \infty$;
2. per ogni $\varepsilon > 0$,
$$\mu\left(\{x \in \mathbb{R}^n \mid |f_j(x) - g(x)| > \varepsilon\}\right) \to 0 \qquad j \to \infty;$$
3. $\displaystyle\int_{\mathbb{R}^n} \min\left(|f_j(x) - g(x)|, 1\right) \mathrm{d}x \to 0$ per $j \to \infty$.

Si ricorda che lo spazio delle funzioni misurabili $\mathcal{M}^n$ è dotato di una distanza d_μ per ogni misura μ su $\mathbb{R}^n$[8].

Proposizione 7.1.16. Sia $(f_j)_j \subseteq \mathcal{M}^n$ una successione di funzioni in $\mathcal{M}^n$ che converge uniformemente sui compatti a f, ovvero tale che, per ogni $K \subseteq \mathbb{R}^n$ compatto

$$\sup_{x \in K} |f_j(x) - f(x)| \to 0, \quad j \to \infty.$$

Allora $d_\mu(f_j, f) \to 0$ per $j \to \infty$.

Teorema 7.1.17. Sia $\sigma : \mathbb{R} \to \mathbb{R}$ una funzione continua e sigmoidale. Allora l'insieme

$$\mathcal{G} = \left\{ G(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x} + \theta_j) \mid N \in \mathbb{N}, \boldsymbol{w}_j \in \mathbb{R}^n, \alpha_j, \theta_j \in \mathbb{R} \right\}$$

è uniformemente denso sui compatti in $C(\mathbb{R}^n)$[9], ovvero per ogni $f \in C(\mathbb{R}^n)$ esiste una successione di funzioni $(G_m)_m \subseteq \mathcal{G}$ tale che $G_m \to f$ uniformemente su ogni compatto $K \subseteq \mathbb{R}^n$, per $m \to \infty$.

Dimostrazione. 1. Si applica il Teorema di Cybenko, che vale per ogni K compatto di $\mathbb{R}^n$, costruendo per ogni $f \in C(K)$ una successione

$$G_j^{(K)} \to f \text{ uniformemente su } K$$

tale che $\sup_{x \in K} |G_j^{(K)}(x) - f(x)| < 1/j$.

2. Si costruisce una successione crescente di compatti $K_j \to \mathbb{R}^n$.

[7] Vedi "Spazio delle funzioni misurabili come spazio metrico"
[8] Vedi "Spazio delle funzioni misurabili come spazio metrico"
[9] Vedi "Classe C di una funzione". Si considera $C(\mathbb{R}^n)$ dotato della topologia indotta dalla metrica

$$d(f,g) \coloneqq \sup_{x \in \mathbb{R}^n} |f(x) - g(x)|.$$

3. Per ogni $f \in C(\mathbb{R}^n) \subseteq C(K_i)$ si ottengono applicando 1. ω successioni $G_j^{(i)} \to f$ uniformemente per $j \to \infty$ su K_i.

   La successione $G_m \coloneqq G_m^{(m)}$ è quella cercata.

■

Teorema 7.1.18. Sia $\sigma : \mathbb{R} \to \mathbb{R}$ una funzione continua e sigmoidale. Allora l'insieme

$$\mathcal{G} = \left\{ G(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x} + \theta_j) \mid N \in \mathbb{N}, \boldsymbol{w}_j \in \mathbb{R}^n, \alpha_j, \theta_j \in \mathbb{R} \right\}$$

è, per ogni misura di probabilità μ su $\mathbb{R}^n$, denso nello spazio delle funzioni misurabili $\mathcal{M}^n$[10].

Questo segue banalmente dal fatto che $C(\mathbb{R}^n)$ sia denso in $\mathcal{M}^n$.

7.1.5 Funzioni L^q

Proposizione 7.1.19. La famiglia delle funzioni continue a supporto compatto

$$\left\{ f : \mathbb{R}^n \to \mathbb{R} \mid f \text{ continua a supporto compatto} \right\}$$

è densa in $L^q(\mathbb{R}^n)$ per ogni $q \in [1, +\infty)$[11].

Proposizione 7.1.20. La famiglia delle funzioni continue e lineari a tratti $\mathbb{R}^n \to \mathbb{R}$ è densa nella famiglia delle funzioni continue a supporto compatto con la topologia indotta dalla distanza

$$d(f, g) = \sup_{x \in \mathbb{R}^n} |f(x) - g(x)|$$

Teorema 7.1.21. Ogni funzione in $L^q(\mathbb{R}^n)$[12] può essere approssimata arbitrariamente bene nella norma $\|\cdot\|_q$ da una rete neurale feedforward ReLU.

7.1.6 Accuratezza dell'approssimazione

Si consideri una rete neurale con un layer nascosto contenente N neuroni, che produce un output $G(x)$, che vuole approssimare f in uno spazio funzionale quale[13]

$$C(I_n), \quad L_1(I_n), \quad L_2(I_n)$$

[10] Si considera $\mathcal{M}^n$ dotato della topologia indotta dalla metrica indotta da μ.
[11] Vedi "Spazi Lp"
[12] Vedi "Spazi Lp"
[13] Sia $I_n \coloneqq [0, 1]^n$. Vedi

- "Classe C di una funzione" e "One Hidden Layer Discriminatory Network impara funzioni continue sul cubo".
- Si consideri $C(I_n)$ dotato della topologia indotta dalla metrica

  $$d(f, g) \coloneqq \sup_{x \in I_n} |f(x) - g(x)| = \max_{x \in I_n} |f(x) - g(x)|.$$

- "Spazi Lp" e
  - "One Hidden Layer Discriminatory Network impara funzioni L1 sul cubo"
  - "One Hidden Layer Discriminatory Network impara funzioni L2 sul cubo"

In che modo N influenza l'accuratezza dell'approssimazione?

Proposizione 7.1.22. Sia $f \in L^2(\mathbb{R}^n)$ tale che, detta $\|\boldsymbol{w}\|_1 \coloneqq |\boldsymbol{w}\cdot\boldsymbol{w}|^{1/2}$ e detta $\hat{f}(\boldsymbol{w})$ la trasformata di Fourier di f, si abbia

$$C_f \coloneqq \int_{\mathbb{R}^n} \|\boldsymbol{w}\|_1 \, \hat{f}(\boldsymbol{w}) \, \mathrm{d}\boldsymbol{w} < \infty.$$

Allora esiste una combinazione lineare

$$G(\boldsymbol{x}) = b_0 + \sum_{k=1}^{N} \alpha_k \sigma(\boldsymbol{w}_k \cdot \boldsymbol{x} + \theta_k)$$

tale che, per ogni $r \in \mathbb{R}^{>0}$ e per ogni misura di probabilità μ_r sulla palla chiusa $B_r \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}\| \leq r\}$:

$$\|G - f\|^2_{L^2(B_r,\mu_r)} = \int_{B_r} |G(\boldsymbol{x}) - f(\boldsymbol{x})|^2 \, \mathrm{d}\mu_r(\boldsymbol{x}) < \frac{(2r\, C_f)^2}{N}.$$

Capitolo 8

Exact Approximation

8.1 Exact Learning (Machine Learning)

By exact learning we mean the expressibility of a network to reproduce exactly the desired target function, i.e. the output of the network is exactly the target function. For an exact learning the network weights do not need tuning; their values can be found exactly. Si parla quindi di rappresentazione esatta, e non di apprendimento.

Alcuni casi in cui questo può succedere:

- One Hidden Layer Perceptron Network rappresenta esattamente funzioni a supporto finito
- Rete Neurale ReLU-feedforward rappresenta esattamente la funzione massimo tra N input

8.1.1 Exact Learning non è sempre possibile

Si consideri a titolo esplicativo $C([0,1])$[1], e si consideri una rete neurale con un layer nascosto con funzione di attivazione logistica σ. La funzione di input-output della rn è

$$G(x) = \sum_{j=1}^{N} \alpha_j \sigma(w_j x + \theta_j).$$

Siccome σ è analitica, allora $G(x)$ è analitica, mentre esistono funzioni continue ma non analitiche, come

$$f(x) = \begin{cases} 0 & x \in \left[0, \frac{1}{2}\right] \\ x - \frac{1}{2} & x \in \left[\frac{1}{2}, 1\right]. \end{cases}$$

[1] Vedi "Classe C di una funzione"

8.2 Funzioni a supporto finito

Proposizione 8.2.1. Sia $g : \mathbb{R}^r \to \mathbb{R}$ una funzione qualsiasi e sia $S = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subseteq \mathbb{R}^r$. Allora esiste

$$G(\boldsymbol{x}) = \sum_{j=1}^{n} \alpha_j \, H(\boldsymbol{w}_j \cdot \boldsymbol{x} - \theta_i),$$

dove H è la funzione di Heaviside, $\alpha_j, \theta_j \in \mathbb{R}$ e $\boldsymbol{w}_j \in \mathbb{R}^r$, tale che

$$\forall i = 1, \dots, n : \qquad G(x_i) = g(x_i).$$

Corollario 8.2.2. Dati $\{(x_i, y_i) \mid i = 1, \dots, n\} \subseteq \mathbb{R}^r \times \mathbb{R}$ esiste

$$G(\boldsymbol{x}) = \sum_{j=1}^{n} \alpha_j \, H(\boldsymbol{w}_j \cdot \boldsymbol{x} - \theta_i),$$

dove H è la funzione di Heaviside, $\alpha_j, \theta_j \in \mathbb{R}$ e $\boldsymbol{w}_j \in \mathbb{R}^r$, tale che

$$\forall i = 1, \dots, n : \qquad G(x_i) = y_i.$$

8.3 Funzione massimo

Proposizione 8.3.1. La funzione $\max\{x_1, x_2\}$ è rappresentata esattamente da una rete neurale con un layer nascosto con 4 neuroni ReLU, ovvero

$$\max\{x_1, x_2\} = \sum_{j=1}^{4} \lambda_j \, \mathrm{ReLU}(w_{1j} x_1 + w_{2j} x_2)$$

per qualche $\lambda_j, w_{ij} \in \mathbb{R}$.

Proposizione 8.3.2. La funzione $f(x_1, \dots, f_{x_n}) = \max\{x_1, \dots, x_n\}$ è rappresentata esattamente da una rete neurale ReLU-feedforward con

$$L = \begin{cases} 2k & \text{se } n = 2^k \\ 2(\lfloor \log_2 n \rfloor + 1) & \text{altrimenti} \end{cases}$$

Proposizione 8.3.3. Esiste una rete neurale feedforward con input di dimensione 2, un layer nascosto di larghezza 2, output di dimensione 1 con funzione di attivazione ReLU in ogni neurone che rappresenta esattamente la funzione

$$f(x_1, x_2) = \max\{x_1, x_2\} \quad \forall x_1 \in \mathbb{R}, x_2 \in \mathbb{R}^{\geq 0}.$$

come in Fig. 8.1.

In realtà, però, per ogni $x_1, x_2 \in \mathbb{R}$

$$\max\{x_1, x_2\} = \mathrm{ReLU}(x_1 - x_2) + x_2.$$

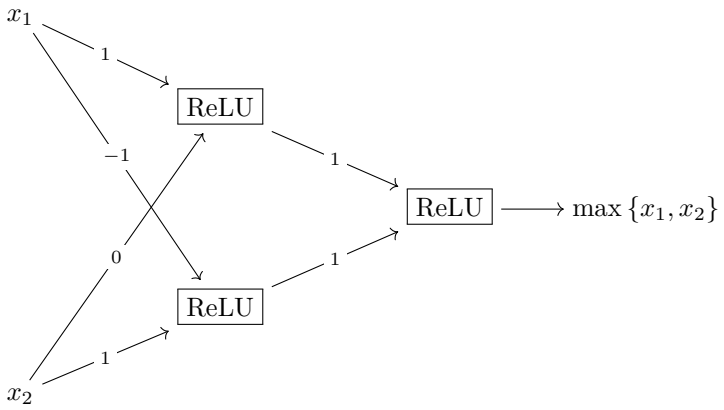


Figura 8.1: Rete Neurale che impara $\max\{x_1, x_2\}$.

8.4 Width vs Depth

Proposizione 8.4.1. Sia $\mathcal{N}$ una rete neurale ReLU-feedforward, con

- input d-dimensionale
- 1 layer nascosto di larghezza n
- output 1-dimensionale.

Allora esiste un'altra rete neurale ReLU-feedforward $\widetilde{\mathcal{N}}$, con

- input d-dimensionale di input d
- profondità $n+2$, dove ciascun layer nascosto ha larghezza $d+2$ (quindi $n+1$ layer nascosti),
- output 1-dimensionale,

tale che $\mathcal{N}$ e $\widetilde{\mathcal{N}}$ producono la stessa funzione per input in $[0,1]^d$.

8.5 ReLU feedforward network

Lemma 8.5.1. La funzione input-output di una rete neurale ReLU-feedforward è una funzione continua e lineare a tratti.

Proposizione 8.5.2. Sia $f : \mathbb{R}^n \to \mathbb{R}$ una funzione continua e lineare a tratti. Allora esiste una rete neurale feedforward, con neuroni ReLU e lineari, che rappresenta esattamente f. Inoltre ha L layers, con

$$L = \begin{cases} 2(k+1) & \text{se } n+1 = 2^k \\ 2(\lfloor \log_2(n+1) \rfloor + 2) & \text{altrimenti} \end{cases}$$

8.6 Kolmogorov-Arnold-Sprecher

Teorema 8.6.1. Ogni funzione continua $f : [0,1]^n \to \mathbb{R}$, per $n \geq 2$, può essere scritta come

$$f(x_1, \ldots, x_n) = \sum_{j=1}^{2n+1} \chi_j \left(\sum_{i=1}^{n} \psi_{ij}(x_i) \right)$$

dove χ_j e ψ_{ij} sono funzioni continue in una variabile, e ψ_{ij} sono funzioni monotone che non dipendono da f.

Teorema 8.6.2. Per ogni $n \geq 2$ esiste una funzione crescente $\psi : [0,1] \to \mathbb{R}$ tale che $\psi\left([0,1]\right) = [0,1]$, dipendente da n, con la seguente proprietà: per ogni $\delta > 0$ esiste $\varepsilon \in \mathbb{Q}$, con $0 < \varepsilon < \delta$, tale che ogni $f : [0,1]^n \to \mathbb{R}$ può essere scritta come

$$f(x_1, \ldots, x_n) = \sum_{j=1}^{2n+1} \chi \left(\sum_{i=1}^{n} \lambda^i \psi(x_i + \varepsilon(j-1)) + j - 1 \right)$$

dove $\chi : \mathbb{R} \to \mathbb{R}$ è continua e $\lambda \in \mathbb{R}$ è indipendente da f.

8.6.1 Applicazione alle reti neurali

This result is important to the field of neural networks because it states that any continuous function on $[0,1]^n$ can be represented exactly by a neural network with two hidden layers. The activation function for the first hidden layer is ψ and for the second hidden layer is χ.