

Metodi Matematici per il Machine Learning

Davide Peccioli

a.a. 2024-25

Indice

I	De Rossi	5
1	Reti Neurali	7
1.1	Neurone Artificiale	7
1.1.1	Neurone Sigma-Heaviside	8
1.1.2	Regressione Lineare	9
1.2	Funzioni di attivazione	10
1.2.1	Funzioni Lineari	10
1.2.2	Step Functions	10
1.2.3	Hockeystick Functions	12
1.2.4	Funzioni Sigmoidali	16
1.2.5	Bumped-type Functions	18
1.3	Rete Neurale	20
1.3.1	Hidden Layer di una rete neurale	21
1.4	Funzioni costo (Machine Learning)	21
1.4.1	La Funzione Errore Supremum	21
1.4.2	La Funzione Errore Norma L2	21
1.4.3	Regolarizzazione della Funzione Costo (Machine Learning)	22
1.5	Processo di apprendimento di una rete neurale	23
1.5.1	Errori di Training e di Test	24
1.5.2	Iperparametri di un processo di apprendimento	24
1.5.3	Alcuni esempi di algoritmi di apprendimento	24
2	Cenni di Analisi Matematica	25
2.1	Teoria della misura	25
2.1.1	Funzioni sigmoidali e funzioni discriminatorie	25
2.2	Minimizzazione	26
II	Cordero	27
3	Teoremi di approssimazione	29
3.1	Teoremi di Dini per la convergenza uniforme	29
3.2	Teorema di Ascoli-Arzelà	29
3.3	Teorema di Stone Weierstrass	32

3.3.1	Corollari del teorema	33
3.3.2	Applicazioni alle reti neurali	34
3.4	Teoremi Tauberiani di Wiener	35
3.4.1	Applicazioni dei Teoremi Tauberiani di Wiener al Machine Learning	36

III Sirovich 39

Parte I

De Rossi

Capitolo 1

Reti Neurali

1.1 Neurone Artificiale

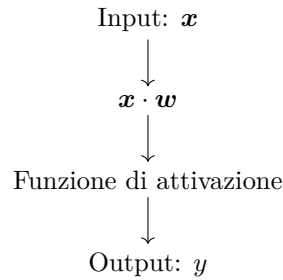
Un neurone è una cellula del corpo umano che può essere schematizzata come segue:

A neuron is a cell which consists of the following parts: dendrites, axon, and body-cell. The synapse is the connection between the axon of one neuron and the dendrite of another. The functions of each part is briefly described below:

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

Matematicamente, quindi, si considera un neurone come una unità che riceve degli input (un vettore \mathbf{x}), lo **moltiplica** per un vettore di pesi $\mathbf{w} = (w_0, \dots, w_n)$, somma un certo bias, e produce un output

processando il prodotto scalare tramite una funzione di attivazione:

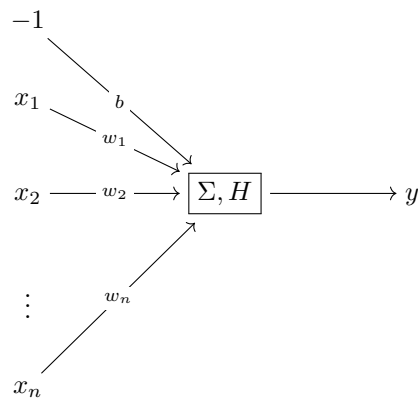


1.1.1 Neurone Sigma-Heaviside

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

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

Questo viene schematizzato in questo modo:



e l'output y è dato da¹

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

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

La convenzione è per semplicità di notazione, infatti

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

¹La funzione $H : \mathbb{R} \rightarrow \mathbb{R}$ è la [Funzione di Heaviside](#):

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

1.1.2 Regressione Lineare

Un altro esempio di neurone è quello che approssima² una [funzione continua](#)

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

con $K \subseteq \mathbb{R}^n$ [compatto](#).

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

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

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

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

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

Si vuole quindi minimizzare

$$C(a_1, \dots, a_n) = \frac{1}{2} \|f - L\|_{L^2}^2 = \frac{1}{2} \int_K \left(\sum_{i=1}^n a_i x_i - f(X) \right)^2 dx_1 \cdots dx_n$$

calcolandone il [gradiente](#)

$$\begin{aligned} \frac{\partial C}{\partial a_k} &= \int_K x_k \left(\sum_{i=1}^n a_i x_i - f(X) \right) dx_1 \cdots dx_n \\ &= \sum_{i=1}^n a_i \int_K x_i x_k dx_1 \cdots dx_n - \int_K x_k f(X) dx_1 \cdots dx_n \end{aligned}$$

e dunque, posti

$$\rho_{ij} := \int_K x_i x_j dx_1 \cdots dx_n, \quad m_k := \int_K x_k f(X) dx_1 \cdots dx_n$$

si ha che

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

ovvero, in forma matriciale, posta³ $\rho = (\rho_{ij})$, $\mathbf{a} = (a_1, \dots, a_n)$ e $\mathbf{m} = (m_1, \dots, m_n)$:

$$\nabla C = \rho \mathbf{a} - \mathbf{m}$$

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

³Vedi:

- [Spazio delle matrici](#)
- [Matrice Trasposta](#)
- [Gradiente di una funzione](#)

Dunque, posto che ρ sia [invertibile](#), si ottiene che i valori ottimali per L siano

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

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

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 $A \subseteq \mathbb{R} \rightarrow B \subseteq \mathbb{R}$.

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. 1.1)

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

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

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

Bipolar step function

La funzione segno: (vedi Fig. 1.2)

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

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

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

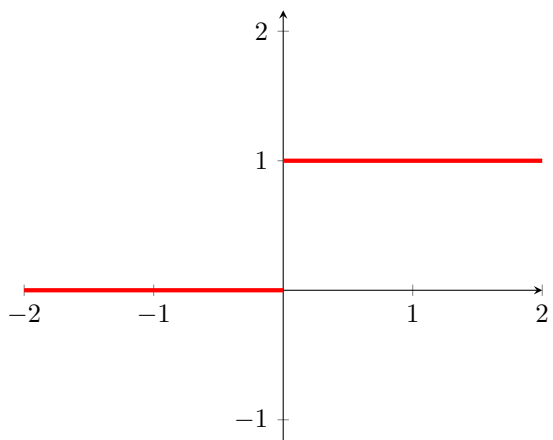


Figura 1.1: La funzione di Heaviside

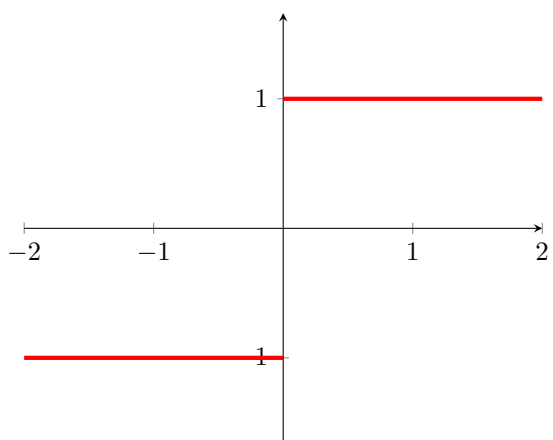


Figura 1.2: La funzione segno

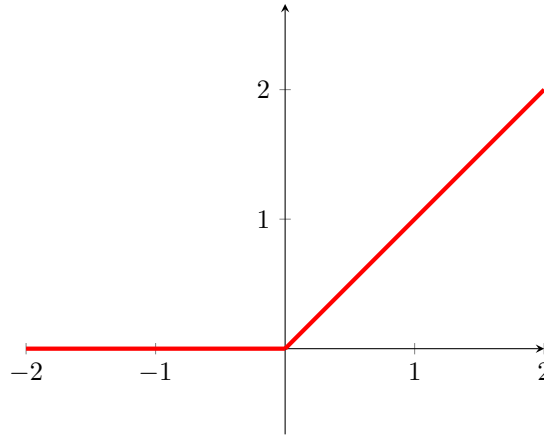


Figura 1.3: La funzione $\text{ReLU}(x)$

1.2.3 Hockeystick Functions

ReLU

La *Rectified Linear Unit* (ReLU) è (vedi Fig. 1.3)

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

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

PReLU

La *Parametric Rectified Linear Unit* (PReLU) è (vedi Fig 1.4), per $\alpha > 0$

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

ELU

La *Exponential Linear Units* (ELU) è (vedi Fig. 1.5):

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

SELU

La *Scaled Exponential Linear Units* (SELU) è (vedi Fig. 1.6)

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

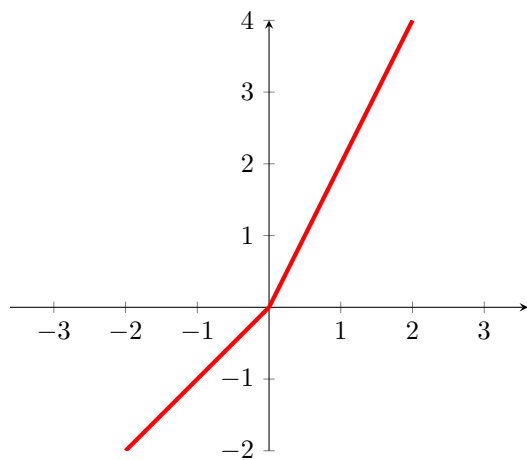


Figura 1.4: La funzione $\text{PRELU}_2(x)$

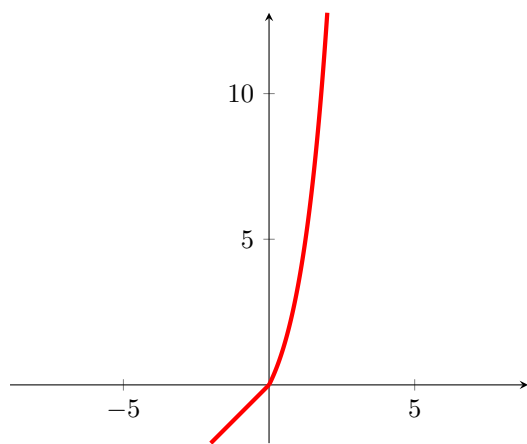


Figura 1.5: La funzione $\text{ELU}(\alpha, x)$

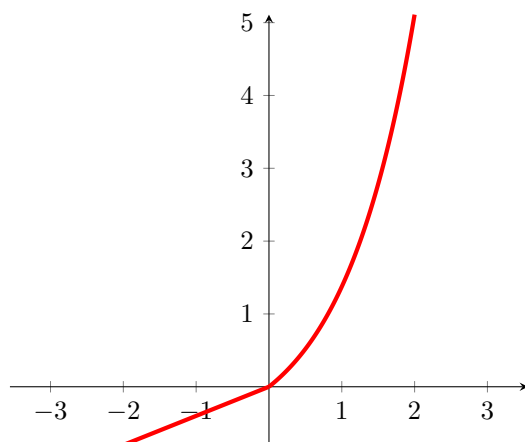


Figura 1.6: La funzione $\text{SELU}(0.4, 2, x)$

SLU

La *Sigmoid Linear Units* (SLU) è (vedi Fig. 1.7)

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

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

Spesso si usa anche la versione parametrica:

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

Softplus

Questa è una funzione positiva crescente, con **range** $(0, +\infty)$: (vedi Fig. 1.8)

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

Inoltre

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

La sua derivata è

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

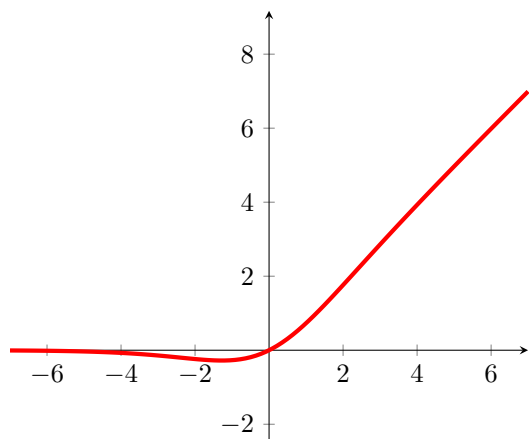


Figura 1.7: La funzione $SLU(x)$

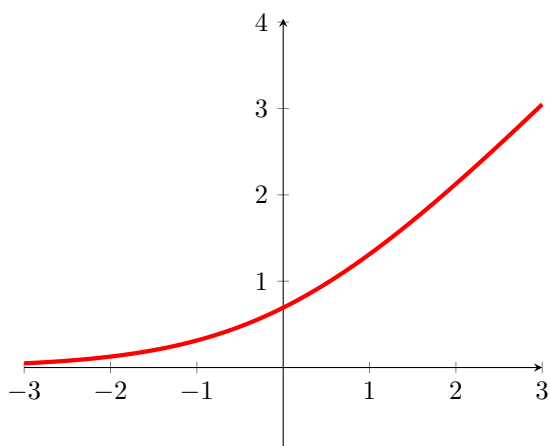


Figura 1.8: La funzione $sp(x)$

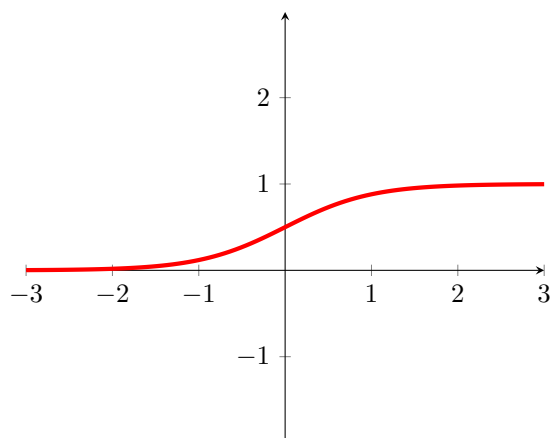


Figura 1.9: La funzione $\sigma_2(x)$

1.2.4 Funzioni Sigmoidali

Funzione Logistica

La funzione logistica è (vedi Fig. 1.9)

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

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

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

e pertanto

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

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

Le funzioni logistiche sono soluzioni della seguente equazione differenziale:

$$\sigma'_c = c\sigma_c(1 - \sigma_c)$$

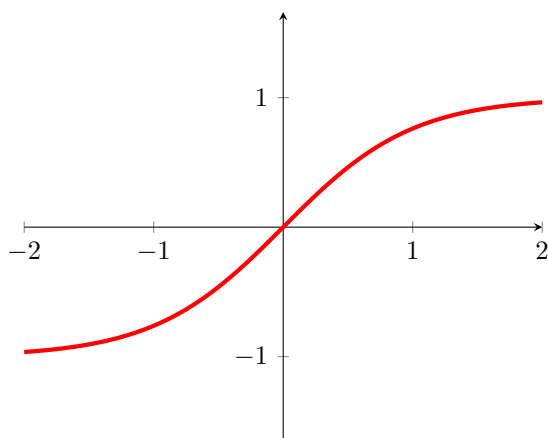


Figura 1.10: La funzione $\tanh(x)$

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 riferisce a $\sigma := \sigma_1$ come alla funzione logistica-

Tangente Iperbolica

La tangente iperbolica $\tanh(x)$ è (vedi Fig. 1.10)

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

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

$$\begin{aligned}
 \tanh'(x) &= 2\sigma'_2(x) = 2 \cdot 2\sigma_2(x) \cdot (1 - \sigma_2(x)) \\
 &= 2\sigma_2(x) \cdot (2 - 2\sigma_2(x)) \\
 &= (\tanh(x) + 1) \cdot (1 - 2\sigma_2(x) + 1) \\
 &= (\tanh(x) + 1)(-\tanh(x) + 1) = 1 - (\tanh(x))^2
 \end{aligned}$$

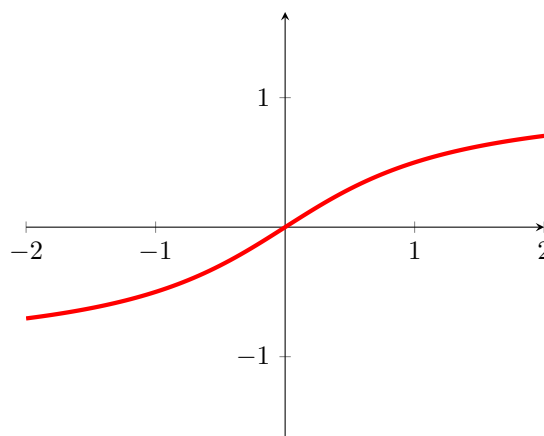


Figura 1.11: La funzione $h(x)$

Arcotangente

È spesso utilizzata la seguente [arcotangente](#) (vedi Fig. 1.11):

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

Softsign

La seguente funzione differenziabile è la funzione *softsign*: (vedi Fig. 1.12)

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

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

Piecewise Linear

Dato un parametro $\alpha > 0$ (vedi Fig. 1.13)

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

1.2.5 Bumped-type Functions

Gaussiana

La funzione gaussiana mappa \mathbb{R} nell'intervallo $(0, 1]$: (vedi Fig. 1.14)

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

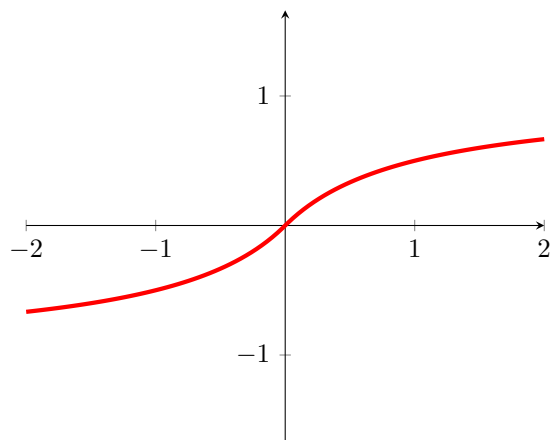


Figura 1.12: La funzione $\text{so}(x)$

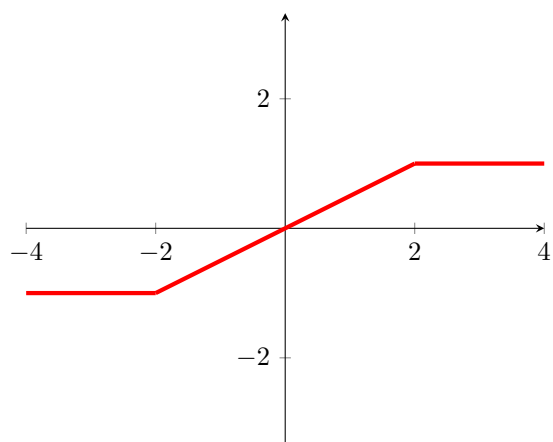


Figura 1.13: La funzione $f_2(x)$

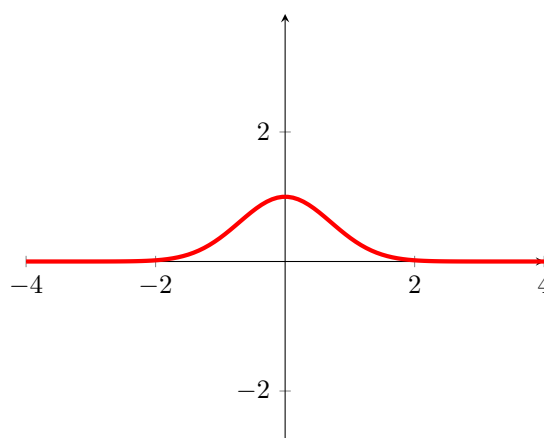


Figura 1.14: La funzione $g(x)$

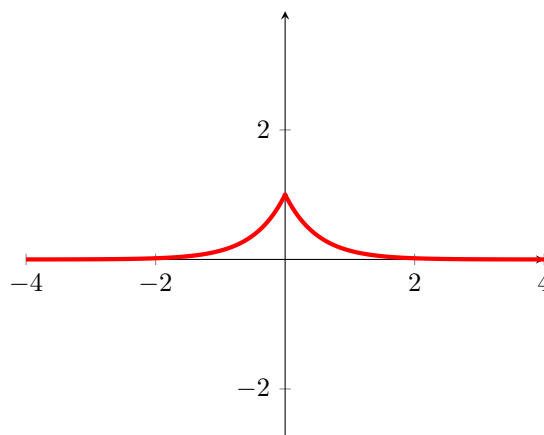


Figura 1.15: La funzione $f(x)$ con parametro $\lambda = 2$

Doppio esponenziale

Mappa la retta reale nell'intervallo $(0, 1]$ ed è definita: (vedi Fig. 1.15)

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

1.3 Rete Neurale

Una rete neurale è [...]

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

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

Il processo di apprendimento è quello che, partendo dai parametri \mathbf{w} , li modifica (iterativamente), fino a dei parametri \mathbf{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

1.4 Funzioni costo (Machine Learning)

Una funzione costo è una funzione che misura, dati certi parametri \mathbf{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 $\phi : [0, 1] \rightarrow [0, 1]$.

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

La funzione costo Supremum è

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

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

$$C(\mathbf{w}, b) := \max_{i=1, \dots, N} |f_{\mathbf{w},b}(x_i) - \phi(x_i)|.$$

1.4.2 La Funzione Errore Norma L2

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

$$\int_0^1 (\phi(x))^2 dx < \infty$$

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

$$C(\mathbf{w}, b) := \int_{[0,1]} (f_{\mathbf{w},b}(x) - \phi(x))^2 dx.$$

Se la funzione ϕ è conosciuta soltanto in N punti

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

allora, posti $\mathbf{z} = (z_1, \dots, z_N)$ e $\mathbf{x} = (x_1, \dots, x_N)$, la funzione costo diventa la [distanza](#) in \mathbb{R}^N tra \mathbf{z} e $f_{\mathbf{w},b}(\mathbf{x}) := (f_{\mathbf{w},b}(x_1), \dots, f_{\mathbf{w},b}(x_N))$:

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

Interpretazione Geometrica

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

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

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

Il costo è minimizzato in $(\mathbf{w}^*, \mathbf{b}^*)$ quando la distanza è minima, ovvero quando $\Phi(\mathbf{w}^*, \mathbf{b}^*)$ è la proiezione ortogonale di \mathbf{z} sulla ipersuperficie: questo significa che il vettore $\Phi(\mathbf{w}^*, \mathbf{b}^*) - \mathbf{z}$ è ortogonale al [piano tangente](#) alla ipersuperficie in $\Phi(\mathbf{w}^*, \mathbf{b}^*)$: quest'ultimo è generato dai vettori⁴

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

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

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

Queste sono le equazioni normali, che operativamente diventano

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

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 $C(\mathbf{w})$, la si regolarizza, utilizzando una funzione costo $G(\mathbf{w})$, data da $C(\mathbf{w})$ più un termine di regolarizzazione.

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

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

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

⁴Vedi: "[Derivata parziale](#)"

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

$$G(\mathbf{w}) := C(\mathbf{w}) + \lambda \|\mathbf{w}\|_1^2, \quad \text{dove } \|\mathbf{w}\|_1 = \sum_{i=1}^n |w_i|.$$

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 Regulation. Sia $U : \mathbb{R}^n \rightarrow \mathbb{R}^+$ 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:

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

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 $C(\mathbf{w})$.

Questo insieme è diviso in tre parti:

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

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

- errore di training $C_{\mathcal{T}}(\mathbf{w})$: è il valore della funzione costo utilizzando i valori della funzione target presi dal training set;
- errore di test $C_{\mathcal{T}}(\mathbf{w})$: è il valore della funzione costo utilizzando i valori della funzione target presi dal test set;
- validation error $C_{\mathcal{V}}(\mathbf{w})$: 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 \mathbf{w}^* che minimizza $C_{\mathcal{T}}$. Successivamente, si calcola $C_{\mathcal{T}}(\mathbf{w}^*)$, e generalmente vale:

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

Ci sono tre possibili scenari, a questo punto:

- sia $C_{\mathcal{T}}(\mathbf{w}^*)$ che $C_{\mathcal{T}}(\mathbf{w}^*)$ sono piccoli: questo è lo scenario desiderato;
- $C_{\mathcal{T}}(\mathbf{w}^*)$ è piccolo, ma $C_{\mathcal{T}}(\mathbf{w}^*)$ è grande: questo è un fenomeno di overfitting; questo significa che la rete neurale sta “memorizzando” il training set, e non riesce a generalizzare bene; probabilmente bisogna rivedere l’architettura della rete neurale, probabilmente diminuendo i parametri;
- sia $C_{\mathcal{T}}(\mathbf{w}^*)$ che $C_{\mathcal{T}}(\mathbf{w}^*)$ 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 minimizzazione del validation error proprio per regolare gli iperparametri.

1.5.3 Alcuni esempi di algoritmi di apprendimento

Regressione Lineare

Capitolo 2

Cenni di Analisi Matematica

2.1 Teoria della misura

2.1.1 Funzioni sigmoidali e funzioni discriminatorie

Definizione 2.1.1. Una funzione $\sigma : \mathbb{R} \rightarrow [0, 1]$ si dice sigmoidale se¹

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

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

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

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

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

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

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

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

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

¹Vedi “*Limite (Analisi Matematica)*”

2.2 Minimizzazione

Definizione 2.2.1. *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur 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. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

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

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

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

Parte II

Cordero

Capitolo 3

Teoremi di approssimazione

3.1 Teoremi di Dini per la convergenza uniforme

Teorema 3.1.1. Sia $f_n : [a, b] \rightarrow \mathbb{R}$ una [successione di funzioni continue](#).

1. Se per ogni $n \in \mathbb{N}$: $f_{n+1} \leq f_n$ e

$$\forall x \in [a, b] \quad f_n(x) \rightarrow 0$$

allora f_n [converge uniformemente](#) a 0 su $[a, b]$.

2. Sia $g : [a, b] \rightarrow \mathbb{R}$ continua. Se per ogni $n \in \mathbb{N}$: $f_{n+1} \leq f_n$ e

$$\forall x \in [a, b] \quad f_n(x) \rightarrow g(x)$$

allora f_n [converge uniformemente](#) a g su $[a, b]$.

3.2 Teorema di Ascoli-Arzelà

Definizione 3.2.1. 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

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

Esempio 3.2.2. Sia $\mathcal{F} := \{\cos(ax + b); a, b \in \mathbb{R}\}$. Allora la famiglia \mathcal{F} è uniformemente limitata su \mathbb{R} , per $M = 1$.

Esempio 3.2.3. Si consideri

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

dove $\sigma(x)$ è una [funzione sigmoideale](#) 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 3.2.4. 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

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

Equivalentemente, le funzioni di \mathcal{F} sono [uniformemente continue](#) per gli stessi ε e δ .

Esempio 3.2.5. Si consideri $\mathcal{F} \subseteq C^1([a, b])^1$ 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 3.2.6. Sia σ una funzione sigmoideale 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:

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

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

¹Vedi "[Classe C di una funzione](#)"

Per l'esempio precedente, si ottiene la tesi.

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

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

Teorema 3.2.7. 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 3.2.8. 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](#) $g : [a, b] \rightarrow \mathbb{R}$ 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 3.2.9. Anche per una rete neurale composta da un unico neurone, con output

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

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

Dimostrazione. Infatti, la famiglia di funzioni continue

$$\mathcal{F} := \{f_{\mathbf{w},b} \mid \|\mathbf{w}\| \leq 1, b \in \mathbb{R}\}$$

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

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

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. ■

3.3 Teorema di Stone Weierstrass

Definizione 3.3.1. 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 **R-algebra**, specializzata in questo ambito.

Esempio 3.3.2. Sia \mathcal{A} l'insieme di tutte le serie di Fourier finite su $[0, 2\pi]$:

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

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

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

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

Definizione 3.3.3. Sia \mathcal{A} una **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 3.3.4. Sia \mathcal{A} l'insieme dei polinomi definiti su $[a, b]$:

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

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

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

Teorema 3.3.5. 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 metrica⁴:

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

3.3.1 Corollari del teorema

Proposizione 3.3.6. Sia $[a, b] \subseteq \mathbb{R}$, e si consideri $C([a, b])^3$ munito della topologia indotta dalla metrica⁴:

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

Sia:

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

Allora \mathcal{A} è denso in $C([a, b])$, ovvero: per ogni $f : [a, b] \rightarrow \mathbb{R}$ 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 3.3.7. Sia $f : [a, b] \times [c, d] \rightarrow \mathbb{R}$ 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

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

²Vedi “Funzione identità”

³Vedi “Classe C di una funzione”

⁴Tale massimo esiste per il Teorema di Weierstrass

Dimostrazione. Si consideri

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

Si osserva facilmente che \mathcal{A} è chiuso per somme, prodotti e prodotti per scalari, ovvero è una **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**. ■

3.3.2 Applicazioni alle reti neurali

Proposizione 3.3.8. Per ogni **funzione periodica** $F : \mathbb{R} \rightarrow \mathbb{R}$ 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

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

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 **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

$$G(x) := a_0 + \sum_{j=1}^N a_j \cos\left(\frac{2\pi}{T} jx\right) + c_j \sin\left(\frac{2\pi}{T} jx\right) \in \mathcal{A}$$

$$\max_{x \in [0, T]} |G(x) - F(x)| = \max_{x \in \mathbb{R}} |G(x) - F(x)| < \varepsilon$$

Si consideri una rete neurale fatta come segue:

⁵Vedi “**Classe C di una funzione**”

⁶Questo 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. 3.1 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

$$w_j := \frac{2\pi}{T}$$

e α_j, b_j tali che

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

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

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

■

3.4 Teoremi Tauberiani di Wiener

Si consideri l'operatore funzionale

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

Teorema 3.4.1. 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](#):

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

⁷Vedi ["Spazi Lp"](#)

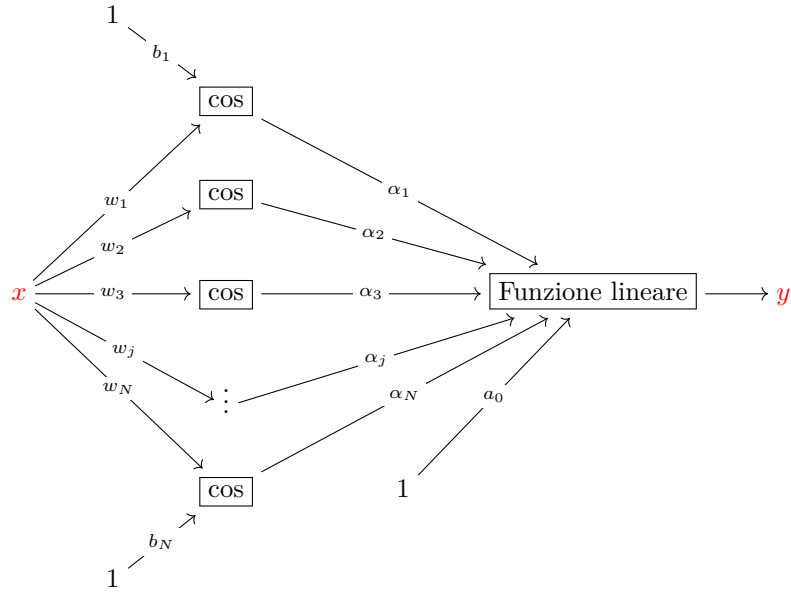


Figura 3.1: La rete neurale considerata

Teorema 3.4.2. 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 Lebesgue** μ :

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

3.4.1 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

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

(ovvero f soddisfa le ipotesi del Teorema 3.4.1)

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

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

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

⁸Vedi “**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

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

(ovvero f soddisfa le ipotesi del Teorema 3.4.2)

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

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

$$\int_{\mathbb{R}} (g(x) - G(x))^2 dx < \varepsilon.$$

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 3.4.1 e 3.4.2.

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

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

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

Parte III

Sirovich

