Appunti di Analisi Funzionale

Github Repository: Oxke/appunti/AnalFun

Primo semestre, 2025 - 2026, prof. Antonio Edoardo Segatti

0.1 Neural Networks

0.1.1 MLP

Definizione 0.1.1: Multi Level Perceptron

Una Multi-Level Perceptron (**MLP**) è una mappa $\varphi : \mathbb{R}^d \to \mathbb{R}^{n_L}$ tale che $\varphi(x_0) = x_L$ e

$$x_l = \rho_L(A_l x_{l-1} + b_l) \quad \forall l = 1, \dots, L \tag{0.1.1}$$

con $\rho_l : \mathbb{R} \to \mathbb{R}$ componente per componente, funzione di attivazione.

La precedente è una rete "feedforward".

Definizione 0.1.2: ResNet

Prendendo $\rho_l(x) = x_{l-1} + \rho(x)$ in (0.1.1) viene una rete neurale "resuduale" di cui un'implementazione è la rete *ResNet* e per l'ottimizzazione può funzionare meglio, nonostante per l'approssimazione non cambia molto.
Inoltre può essere vista come discretizzazione di

$$\dot{x}(t) = \rho(A(t)x(t) + b(t))$$

Definizione 0.1.3: Recurrent NNs

Dati input $y_1, y_2, \dots \in \mathbb{R}^{n_{in}}$, modifichiamo l'operazione (0.1.1) in

$$x_l = \rho(A_x x_{l-1} + A_y y_l) \quad l = 1, 2, \dots$$

con $x_0 \in \mathbb{R}^N$. Possono essere usate ad esempio per risolvere

$$\begin{cases} \overline{x}(t) = F(x(t), y(t)) \\ x(0) = x_0 \end{cases}$$

Capitolo 1

Errors

Ci sono 3 errori che si possono verificare in ambito di questo tipo di matematica applicata:

1. Approssimazione
2. Generalizzazione
3. Training

Principlamente questo corso si occuperà principalmente di comprendere l'errore dovuto all'approssimazione, parlando meno degli altri due errori.

Siano X, Y due insiemi, con μ una misura su X, $\|\cdot\|_Y$ una norma su Y e $G : X \to Y$ una funzione. Possiamo allora definire una **loss function**

$$\mathcal{L}(\Phi) = \int_X \|G(x) - \Phi(x)\|_Y^2 \, d\mu(x)$$

dove $\Phi \in \mathcal{C}$ una classe di funzioni considerate per l'approssimazione

Non potendo avere misuramenti di G per infiniti valori, si prende in realtà un sample $\{x_i\}_{i=1}^N$ di input e $\{y_i = G(x_i)\}_{i=1}^N$ per cui la loss calcolabile è la **empirical loss**

$$\tilde{\mathcal{L}}(\Phi) = \sum_{i=1}^{n} w_i \|y_i - \Phi(x_i)\|_Y^2$$

con i w_i pesi. Chiamiamo Φ_{min} e $\tilde{\Phi}_{min}$ le funzioni che sarebbero i minimi su $\mathcal{C}$ delle due loss rispettivamente.

In pratica, un'algoritmo di ottimizzazione è usato per minimizzare $\tilde{\mathcal{L}}$ e Φ_{comp} è l'approssimazione calcolata. Allora

$$\|G - \Phi_{comp}\| \leq \underbrace{\|G - \Phi_{min}\|}_{\text{approx error}} + \underbrace{\|\Phi_{min} - \tilde{\Phi}_{min}\|}_{\text{generalization error}} + \underbrace{\|\tilde{\Phi}_{min} - \Phi_{comp}\|}_{\text{training error}}$$

In particolare cosa vogliamo arrivare a dimostrare noi è il *universal approximation theorem*, ossia un teorema che dia le ipotesi per poter avere che l'errore tende a zero.

1.1 Setting

Sia $K \subseteq \mathbb{R}^d$ un compatto. Sia $C(K)$ l'insieme delle funzioni continue $K \to \mathbb{R}$. Sia ρ una funzione di attivazione continua. Sia $\mathcal{M}$ la famiglia

$$\mathcal{M} = \{\mu \text{ misura relativa di Borel su } K \text{ con variazione totale finita }\}$$

ossia il duale topologico di $C(K)$

Teorema 1.1.1: Template of a Universal Approximation Theorem

Under some conditions on ρ,

$$MLP(\rho, d, \sim) \text{ is dense in } C(K)$$

By Riesz Theorem $\forall L \in C(K)'$, $\exists \mu \in \mathcal{M}$ such that $Lf = \int_K f \, d\mu$

Definizione 1.1.1: Discriminant

We say that $f \in C(K)$ is **discriminant** if

$$\int_K f(a^T x + b)\mu(dx) = 0 \quad \forall a \in \mathbb{R}^d, \forall b \in \mathbb{R}$$

We will prove that on condition that makes the theorem true is to have ρ be *discriminant*. An easier constraint on ρ, is have ρ not be polynomial.

Esercizio 1.1.1

1. Argue that the thesis of universal approx theorem can't be true if ρ is a polynomial
2. Conclude that a polynomial can't be discriminant
3. Show 2. with the definition

Nel caso di funzioni di attivazioni *regolari* $\max_{x\in[-k,k]} |\Phi(x) - x| \leq \frac{k}{\lambda}$, dove i pesi di Φ sono contenuti (in valore assoluto) in $[\frac{1}{\lambda}, c\lambda]$ per qualche $c > 0$. Ne consegue che

$$\text{errore } \leq \exp\left(-\underbrace{\max |\log(\text{weight}(\Phi))|}_{\text{bit di informazione nei pesi}}\right)$$

Ci importa approssimare la mappa $z \mapsto z^2$, poiché questo implica poter approssimare la moltiplicazione $x, y \mapsto xy$. Infatti

$$\left(\frac{x+y}{2}\right)^2 - \left(\frac{x-y}{2}\right)^2 = xy$$

dall'approssimazione della moltiplicazione segue che è possibile approssimare i polinomi.

Nel caso di una rete shallow monodimensionale, questa è uguale a

$$\sum_{j=1}^{N} c_j \text{ReLU}(a_j x + b_j) + c_0$$

che è una funzione affine a tratti. Ne consegue che l'errore è proporzionale a $\frac{1}{N}$, con N il numero di neuroni, infatti divide gli intervalli in intervallini di diametro $h = \frac{1}{N}$.

Vogliamo mostrare che con reti profonde l'errore diminuisce molto più velocemente. Definiamo ora

$$F_1(x) = \begin{cases} 2x & x \in [0, \frac{1}{2}] \\ 2(1-x) & x \in [\frac{1}{2}, 1] \end{cases}$$

che ha $h = \frac{1}{2}$ e $\texttt{depth} = 2$ Se ora compongo F_1 con se stessa, otteniamo

$$F_2 = F_1 \circ F_1 = \begin{cases} 4x\%1 & x \in [0, \frac{1}{4}] \cup [\frac{1}{2}, \frac{3}{4}] \\ 4(1-x)\%1 & x \in [\frac{1}{4}, \frac{1}{2}] \cup [\frac{3}{4}, 1] \end{cases}$$

che ha $h = \frac{1}{4}$ e $\texttt{depth} = 3$. Similmente osserviamo che

$$h_N = \frac{1}{2}^{\texttt{depth}(F_N)-1}$$

e la rete ha dimensione $\texttt{size}(F_N) = 6 + 3^2(N-1)$.

Ora però dobbiamo mostrare che è possibile approssimare x^2, e non solo che usando la composizione (rete profonda) è possibile superare la barriera inevitabile per le reti *narrow*, ossia $h \sim \frac{1}{N}$.

Proposizione 1.1.2.

$$\left| x^2 - x + \underbrace{\sum_{n=1}^{N} \frac{F_n(x)}{2^{2n}}}_{\text{interpolante di } x^2 \text{ nei punti } \{k2^{-N}\}_{k=0}^{2^N}} \right| \leq 2^{-2N-2}$$

Definizione 1.1.2: Concatenazione Sparsa

Siano $L_1, L_2 \in \mathbb{N}$ due profondità, Φ_1, Φ_2 due reti neurali a parametri rispettivamente $A_{k_1}^{(1)}, b_{k_1}^{(1)}$ e $A_{k_2}^{(2)}, b_{k_2}^{(2)}$ per $k_j \in 1 \dots L_j$ e $j \in 1, 2$.
Si supponga che $\dim \texttt{out}\Phi_2 = \dim \texttt{in}\Phi_1$ allora la *concatenazione sparsa di* Φ_1 *e* Φ_2 è definita come

$$\Phi_1 \odot \Phi_2 = \Phi_1 \cdot \Phi^{\mathrm{Id},2} \cdot \Phi_2$$

e vale che $\texttt{depth}(\Phi_1 \odot \Phi_2) = L_1 + L_2$ e $\texttt{size}(\Phi_1 \odot \Phi_2) \leq 2(\texttt{size}(\Phi_1) + \texttt{size}(\Phi_2))$