L'algoritmo di Levenberg-Marquardt e il Deep Learning

Madon Federica - Matricola 825628
Relatore: Prof. Beirao Da Veiga Lourenco

Università degli Studi di Milano-Bicocca

25 novembre 2021

Il Deep Learning

Il Deep Learning è un sottoinsieme del *Machine Learning* (apprendimento automatico) e indica quella branca dell'Intelligenza Artificiale che fa riferimento agli algoritmi, ispirati alla struttura e alla funzione del cervello, chiamati **reti neurali artificiali**.

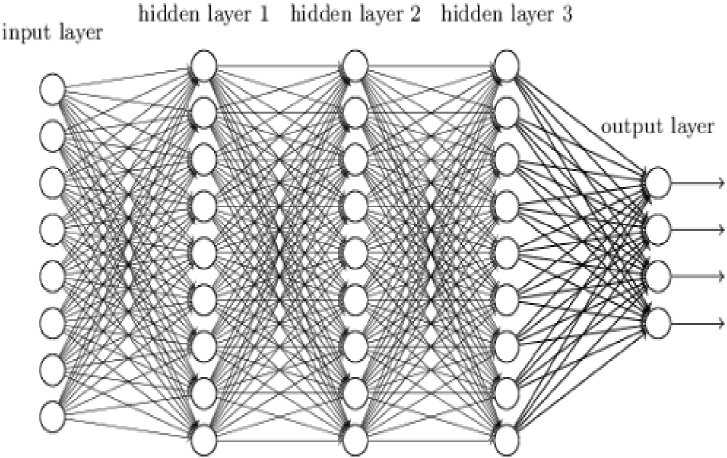
Applicando il Deep Learning, si avrà quindi una "macchina" che consente di approssimare, in uno specifico contesto applicativo, la corrispondenza tra un dato di ingresso e uno di uscita.

Le reti neurali artificiali

Le **reti neurali artificiali** sono modelli di calcolo matematico-informatici composti da neuroni artificiali.

I neuroni ricevono in ingresso una combinazione dei segnali provenienti dall'esterno o dalle altre unità e ne effettuano una trasformazione tramite una funzione, tipicamente non-lineare, detta **funzione di attivazione**. L'uscita di ciascun neurone viene poi inviata agli altri nodi oppure direttamente all'uscita della rete, attraverso connessioni orientate e pesate.

Le reti neurali artificiali



Addestrare una rete neurale

L'addestramento di una rete neurale artificiale si basa sulla risoluzione di un problema di ottimizzazione consistente nella scelta di pesi e costanti che minimizzino la seguente **funzione costo**, di forma di tipo quadratico non-lineare:

$$Cost = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\|y(x^{\{i\}}) - a^{[L]}(x^{\{i\}})\|_2^2.$$

- La coppia di valori $\{x^{\{i\}}\}_{i=1}^{N}$ e $\{y(x^{\{i\}})\}_{i=1}^{N}$ formano il **training set**, $x^{\{i\}} \in \mathbb{R}^{n_1}$ e $y(x^{\{i\}}) \in \mathbb{R}^{n_L}$;
- $a_j^{[l]}$ denota l'output dal neurone j dello strato l, perciò $a^{[l]} = f(W^{[l]}a^{[l-1]} + b^{[l]})$, dove f è la funzione di attivazione e $a^{[1]} = x^{\{i\}}$;
- $W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$ denota la matrice dei pesi al livello l;
- $b^{[l]} \in \mathbb{R}^{n_l}$ è il vettore delle costanti per il layer l.

L'algoritmo di Levenberg-Marquardt

L'algoritmo di **Levenberg-Marquardt** (**LM**) è un algoritmo iterativo di ottimizzazione usato per la soluzione di problemi in forma di minimi quadrati non-lineari.

Fornisce un metodo, che dipende in modo non-lineare da un parametro γ, per calcolare la correzione al punto corrente.

Questo algoritmo risulta molto efficiente per l'addestramento di reti neurali artificiali di piccole dimensioni.

Caratteristiche principali

L'algoritmo ha l'obiettivo di minimizzare la somma di quadrati

$$F(x) = \|f(x)\|^2 = f^T(x)f(x),$$

dove f è un vettore n-dimensionale, di funzioni regolari f_i, delle variabili indipendenti $x_1, x_2, \ldots, x_p$ con $p < n$.
I vettori di discesa $h_i(\gamma)$, $i = 1, 2, \ldots,$ sono scelti risolvendo, ad ogni passo, il problema dei minimi quadrati lineari, che minimizza $\|r_i(\gamma)\|^2$, dove

$$\begin{bmatrix} f(x_i) \\ 0 \end{bmatrix} + \begin{bmatrix} A_i \\ \gamma B_i \end{bmatrix} h_i(\gamma) = r_i(\gamma),$$

e $A_i = \nabla_x f(x_i)$, B_i è una matrice $p \times p$ di rango massimo e $\gamma \geq 0$ è un parametro che regola la grandezza e la direzione della h_i.

Risultati teorici

Teorema

Sia $\{\gamma_i\}$ *una successione limitata tale che:* $0 < \sigma \leq \psi(x_i, \gamma_i)$,

$$x_{i+1} = x_i + h_i(\gamma_i), \qquad \psi(x_i, \gamma_i) = \frac{F(x_i) - F(x_{i+1})}{2(F(x_i) - \|r_i\|^2)}.$$

$\sigma > 0$ *è una costante fissata. Allora la successione* $\{F(x_i)\}$ *è convergente e i punti limite della successione* $\{x_i\}$ *sono punti stazionari di* $F(x)$.

La disuguaglianza $0 < \sigma \leq \psi(x_i, \gamma_i)$ fornisce un metodo per scegliere un γ_i adatto. Fissato σ, se tale disequazione non è soddisfatta, si incrementa il γ_i e si ricalcola h_i; mentre, se è soddisfatta subito, γ_{i+1} sarà diminuito rispetto a γ_i.

Risultati teorici

Teorema

Per ogni $\sigma < 1$ e x_i esiste un γ che soddisfa la condizione $0 < \sigma \leq \psi(x_i, \gamma_i)$.

Teorema

Si supponga che la successione $\{\gamma_i\}$, trovata applicando l'algoritmo di Levenberg-Marquardt, sia illimitata. Allora la norma della matrice Hessiana $\nabla^2 F$ è anch'essa illimitata.

Corollario

Sia $\|\nabla^2 F\|$ finito sull'insieme di livello $\mathcal{L}_0 = \{x \in \mathbb{R}^p$ tali che $f(x) \leq f(x_0)\}$, allora ogni punto limite finito di $\{x_i\}$ è un punto stazionario di $F(x)$.

Test numerici

Per completare questo lavoro sono stati eseguiti alcuni test numerici (usando il linguaggio **MATLAB**).

Si considera una generica funzione continua $g : [0,1] \to \mathbb{R}$. Siano $x_1, x_2, \ldots, x_d$, con $x_1 = 0$, $x_d = 1$ e $d = 20$, i d nodi di una partizione uniforme dell'intervallo $[0,1]$. Il problema modello di regressione non-lineare da risolvere consiste nella ricerca del vettore $w = [w_1, w_2, w_3] \in \mathbb{R}^3$ tale che la funzione di tipo esponenziale

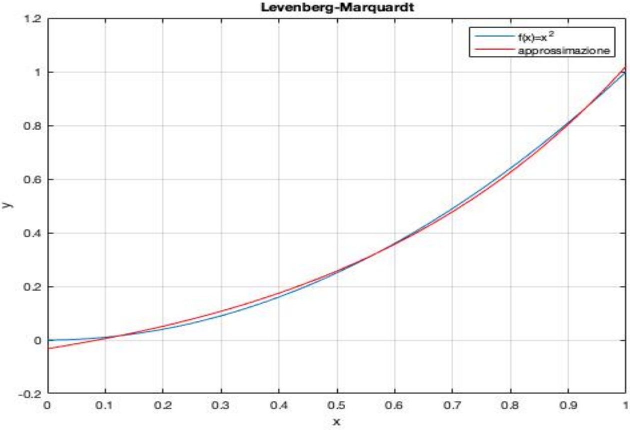
$$w_1 + w_2 e^{(w_3 x)}, \quad x \in [0,1],$$

è quella che meglio approssima la funzione g, nel senso che minimizza il funzionale

$$F(w) = \sum_{i=1}^{d} |g(x_i) - w_1 - w_2 exp(w_3 x_i)|^2.$$

Test numerici

Nella figura seguente il grafico blu rappresenta la funzione $g(x) = x^2$ da approssimare e il grafico rosso l'approssimazione ottenuta, variando il vettore iniziale fornito all'algoritmo.



Grazie per l'attenzione.