1 Storia

Uno dei primi esempi di rete neurale fu il percettrone di Rosenbatt (1958). Di seguito fu creata ADELINE (Widrow & Hoff, 1960), composta da un singolo layer con un singolo neurone e utilizzava il LMS (least mean square) insieme ad un algoritmo di discesa stocastica del gradiente per la configurazione dei pesi. Ne descriviamo adesso le caratteristiche. Siano $x \in R^N$ gli input e $w \in R^N$ i pesi; dal modello abbiamo:

$$\widehat{y} = x \cdot w. \tag{1}$$

Dato l'output desiderato $y \in R$ possiamo definire la funzione errore e e la $loss\ function$ L. L'obiettivo é quello di trovare il valore dei pesi che minimizzi L.

$$e = y - \widehat{y} \tag{2}$$
$$L = e^2 \tag{3}$$
$$w \leftarrow arg\,min(L). \tag{4}$$

L'algoritmo LMS ottimizza L usando il gradiente discendente:

$$\nabla L_w := -2ex \tag{5}$$
$$w \leftarrow w + \alpha ex \tag{6}$$

dove α é il tasso di apprendimento.

Widrow, McCool and Ball (1975) hanno esteso l'algoritmo LMS al dominio complesso fornendo la derivazione delle parti reali e immaginarie. Brandwood (1983) generalizzó la teoria applicando il gradiente al numero complesso, senza separarlo in parte reale e parte immaginaria attraverso il gradiente di Wirtinger (1927).

Aggiorniamo quindi il problema nel dominio complesso:

$$L = e\overline{e} \tag{7}$$
$$\nabla L_w := -2e\overline{x} \tag{8}$$
$$w \leftarrow w + \alpha e\overline{x}. \tag{9}$$

Nonostante i risultati di Brandwood, fino a non molto tempo fa la letteratura non applicava il calcolo di Wirtinger, a favore della derivazione separata della parte reale da quella immaginaria.

2 Funzione d'attivazione

Definiamo l'input $x \in C^M$ e i pesi $\mathbf{W} \in C^{N \times M}$, con M e N rispettivamente le dimensioni di input e output; di conseguenza l'uotput $y \in C^N$ di un qualsiasi layer é determinato da:

$$z = \mathbf{W}x \tag{10}$$
$$y = f(z) \tag{11}$$

dove f solitamente é una funzione di attivazione non lineare.

3 Loss function

La maggior parte della letteratura attuale utilizza come funzione di costo la $mean\,squarred\,error$. Dato un target y e l'output ottenuto $\widehat{y}$, entrambi in C^N e l'errore

$$e := y - \widehat{y} \tag{12}$$

la $complex\,mean\,squared\,loss\,function$ é definita come segue:

$$L(e) = \sum_{i=0}^{N-1} |e_i|^2 \tag{13}$$

$$= \sum_{i=0}^{N-1} e_i \overline{e_i}. \tag{14}$$

La (**??**) é una funzione a valori reali scalari non negativi, che tende a zero insieme al modulo dell'errore. Savitha, Suresh e Sundararajan proposero di sostituire l'errore (**??**) con:

$$e := log\,\widehat{y} - log\,y. \tag{15}$$

La $loss\,function$ diventerebbe quindi:

$$L(e) = (log\,|\widehat{y_i}| - log\,|y_i|)^2 + (arg\,\widehat{y_i} - arg\,y_i)^2 \tag{16}$$

L'equazione (**??**) ha la proprietà di rappresentare esplicitamente l'ampiezza e la fase.
Le equazioni (**??**) e (**??**) possono essere $error\,function$ appropriate per reti neurali complesse.

4 Ottimizzazione

4.1 Il gradiente complesso ed il calcolo di Wirtinger

Brandwood e Van den Bos formularono le prime derivazioni del gradiente complesso. Wirtinger (1927) forní un formalismo equivalente che rese il calcolo della derivata di funzioni con valori complessi meno oneroso rispetto a funzioni olomorfe e non analitiche , agendo interamente nel campo complesso. Nonostante tale apparente comoditá, solo recentemente si ricominció ad utilizzare il calcolo di Wirtinger per la *backpropagation* di reti neurali complesse.

Definiamo:

$$f(z) : = f\left(z, \overline{z}\right) \tag{17}$$

$$= g\left(x, y\right) \tag{18}$$

$$= u\left(x, y\right) + iv\left(x, y\right) \tag{19}$$

con $z \in C$, $x, y \in R$ e $z = x + iy$.

Usando la prima definizione avremo le derivate in z e $\overline{z}$ date da:

$$\left.\frac{\partial f}{\partial z}\right|_{\overline{z}\ costante} \tag{20}$$

$$\left.\frac{\partial f}{\partial \overline{z}}\right|_{z\ costante} \tag{21}$$

le quali, espresse in funzioe di x e y, diventano:

$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - i\frac{\partial f}{\partial y}\right) \tag{22}$$

$$\frac{\partial f}{\partial \overline{z}} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + i\frac{\partial f}{\partial y}\right). \tag{23}$$

Si osservi che la derivata parziale rispetto a $\overline{z}$ é nulla per ogni funzione olomorfa. Richiamiamo le condizioni di esistenza di Cauchy-Riemann per la derivata complessa della funzione $f(z, \overline{z}$ presa in considerazione:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \tag{24}$$

$$\frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}. \tag{25}$$

Se applichiamo le condizioni di Cauchy-Riemann alla derivata di f in $\overline{z}$ (**??**) notiamo che effettivamente essa si annulla. Le funzioni olomorfe quindi non dipendono esplicitamente da $\overline{z}$. Brandwood dimostró che l'annullarsi di (**??**) o (**??**) per una generica $f : C \rightarrow R$ é condizione sufficiente e necessaria affinché f abbia un punto stazionario. Per estensione se $f : C^N \rightarrow R$ é una funzione a valori reali di un certo vettore $z = [z_0,\ z_1,\ \cdots\ ,\ z_{N-1}]^T \in C^N$ e definiamo il cogradiente e il gradiente coniugato come:

$$\frac{\partial}{\partial z} := \left[\frac{\partial}{\partial z_0}, \frac{\partial}{\partial z_1}, \cdots, \frac{\partial}{\partial z_{N-1}}\right] \tag{26}$$

$$\frac{\partial}{\partial z} := \left[\frac{\partial}{\partial \overline{z_0}}, \frac{\partial}{\partial \overline{z_1}}, \cdots, \frac{\partial}{\partial \overline{z_{N-1}}}\right] \tag{27}$$

allora $\frac{\partial f}{\partial z} = 0$ o $\frac{\partial f}{\partial \overline{z}} = 0$ sono condizioni sufficienti e necessarie per determinare un punto di stazionarietá.

Se f é una funzione di un vettore complesso z, la sua derivata totale é:

$$df = \frac{\partial f}{\partial z}dz + \frac{\partial f}{\partial \overline{z}}d\overline{z}. \tag{28}$$

Se f é reale allora avremmo:

$$df = 2Re\left\{\frac{\partial f}{\partial z}dz\right\}. \tag{29}$$

Definendo ora l'operatore gradiente come:

$$\nabla_z := \left(\frac{\partial}{\partial \overline{z}}\right)^T \tag{30}$$

$$= \left(\frac{\partial}{\partial z}\right)^* \tag{31}$$

puó essere dimostrato, usando la disuguaglianza di Cauchy-Scharz, che f ha il maggior tasso di cambiamento lungo i gradiente. Grazie a queste definizioni possiamo costruire una $cost\,function$ reale con argomenti complessi anche se alcuni elementi della funzione non sono olomorfi. In generale, date le funzioni arbitrarie fe g, combinando gli jacobiani come segue

$$J_{f\circ g} = J_f J_g + J_f^{(c)} \overline{\left(J_g^{(c)}\right)} \tag{32}$$

$$J_{f\circ g}^{(c)} = J_f J_g^{(c)} + J_f^{(c)} \overline{(J_g)}. \tag{33}$$

Supponiamo di avere una funzione composta $(f \circ g \circ h)(z, \overline{z})$, con f la *cost function* reale, g una funzione complessa non olomorfa e h una funzione olomorfa. tenendo a mente che f é una funzione a valori reali di variabile complessa e che h é olomorfa ($\frac{\partial h}{\partial \overline{z}=0}$), applichiamo la *chain rule*:

$$J_f = \frac{\partial f}{\partial g} \tag{34}$$

$$J_f^{(c)} = \frac{\partial f}{\partial \overline{g}} \tag{35}$$

$$J_{f\circ g} = J_f \frac{\partial g}{\partial h} + J_f^{(c)} \overline{\left(\frac{\partial g}{\partial \overline{h}}\right)} \tag{36}$$

$$J_{f\circ g\circ h} = J_{f\circ g} \frac{\partial h}{\partial z} \tag{37}$$

$$\nabla_z f = \left(J_{f\circ g\circ h}\right)^* \tag{38}$$

Il Calcolo di Wirtinger rende un po' piú semplice costruire un grafico computazionale per reti complesse aventi composizioni miste di operazioni olomorfe e non olomorfe.

4.2 Inputs

La funzione che la rete neurale deve apprendere ha come argomenti i vettori reali α e φ che rappresentano rispettivamente le componenti di ampiezza e fase. Di seguito mostriamo alcuno modalità per rappresentare l'input:

Ampiezza-Fase I parametri di ampiezza e fase vengono concatenati nel vettore reale:

$$x^{(ap)} := [\cdots,\ \alpha_i,\ \varphi_i,\ \cdots]^T \in R^{2K} \tag{39}$$

Complessa I parametri vengono rappresentati all'interno del vettore complesso:

$$x^{(c)} := \left[\cdots,\ \alpha_i e^{i\varphi_i},\ \cdots\right]^T \in C^K \tag{40}$$

Reale-Immaginaria Possiamo scomporre il vettore (**??**) in parte reale e parte immaginaria e ottenere un input nella forma:

$$x^{(ri)} : = \left[Re\left\{x^{(c)}\right\}^T,\ Im\left\{x^{(c)}\right\}\right]^T \tag{41}$$

$$= [\cdots\ \alpha_i cos\varphi_i\ \cdots\ \alpha_i sin\varphi_i\ \cdots]^T \tag{42}$$

Complesso aumentata L'input é un vettore complesso contenente sia (**??**) che il suo coniugato:

$$x^{(ca)} : = \left[\left(x^{(c)}\right)^T,\ \left(x^{(c)}\right)^*\right]^T \tag{43}$$

$$= \left[\cdots,\ \alpha_i e^{i\varphi_i}\ \cdots\ \alpha_i e^{-i\varphi_i}\ \cdots\right]^T \tag{44}$$

Il *mapping* definito dal complesso coniugato é antilineare e di conseguenza non puó essere calcolato attraverso la moltiplicazione di matrici complesse. ció potrebbe richiedere alla rete l'apprendimento di un ulteriore parametro. In effetti. l'utilizzo del vettore complesso aumentato fornisce un *path* piú semplice per la modellizzazione della distribuzione completa al secondo ordine dei dati. Gli inut aumentati sono utilizzati in una stima lineare.

4.3 Architettura

Esaminiamo un modello $feedforward$ avente un unico hidden layer; abbiamo di conseguenza:

$$h = f\left(W^{(h)}x + b^{(h)}\right) \tag{45}$$

$$\widehat{y} = W^{(o)}x + b^{(o)} \tag{46}$$

con $\theta = \left\{W^{(h)},\ W^{(o)},\ b^{(h)},\ b^{(o)}\right\}$ sono i parametri del modello. Definendo M il numero dei nodi dell'input e N il numero dei nodi dell'output, avremo:

$$\begin{array}{ccc} W^{(h)} \in R^{M\times M} & or & W^{(o)} \in C^{M\times M} \\ b^{(h)} \in R^{M} & or & b^{(o)} \in C^{M} \\ W^{(o)} \in R^{N\times M} & or & W^{(o)} \in C^{N\times M} \\ b^{(o)} \in R^{N} & or & b^{(o)} \in C^{N} \end{array} \tag{47}$$

In tutti i casi gli inut, i pesi e gli output appartengono allo stesso dominio numerico.

4.4 Funzioni di attivazione

Identitá Permette una modellizzazione lineare

$$f^{(-)}\left(x\right) := x \tag{48}$$

Tangente Iperbolica Eúna funzione sigmoidale e differenziabile. Permette una non linearitá ampiamente utilizzata per l'apprendimento delle reti neurali

$$f^{(\sigma)} := tanh\left(x\right) \tag{49}$$

Split reale-immaginario Applica la tangente iperbolica separatamente alla parte reale e alla parte immaginaria:

$$f^{(ri)}\left(x\right) := tanh\left(Re\ x\right) + i\ tanh\left(Im\ x\right) \tag{50}$$

Split ampiezza-fase Applica la tangente iperbolica al modulo del numero complesso, senza modificarne la fase (questa funzione non é differenziabile nel campo complesso)

$$f^{(ap)}\left(x\right) := tanh\left(|x|\right) e^{i\, arg\, x} \tag{51}$$

ModReLU Arjosky nel 2015 propose una variazione alla classica ReLU utilizzata nelle reti neurali reali, definita come segue:

$$ModReLU\left(z\right) = ReLU\left(\left|z\right| + b\right)e^{i\theta_z} = \begin{cases} \left(\left|z\right| + b\right)\frac{z}{\left|z\right|} & \text{se } \left|z\right| + b \geq 0 \\ 0 & \text{altrimenti} \end{cases} \tag{52}$$

dove θ_z é la fase di z e b il bias, apprendibile dalla rete neurale, e necessario per creare una *dead zone* di raggio b attorno all'origine dove il neurone é inattivo. Tale funzione non soddisfa le equazioni di Cauchy-Riemann e quindi non é olomorfa.

CReLU Consiste in una applicazione della funzione ReLu individualmente alla parte reale e alla parte immaginaria:

$$CReLU\left(z\right) = ReLU\left(Re\left(z\right)\right) + i\,ReLU\left(Im\left(z\right)\right). \tag{53}$$

Questa soddisfa le condizioni di Cauchy-Riemann solo se sia la parte reale che la parte immaginaria sono contemporaneamente strettamente positive o strettamente negative, quindi nell'intervallo $\theta_z \in \left(0, \frac{\pi}{2}\right)$ oppure $\theta_z \in \left(\pi, \frac{3\pi}{2}\right)$.

zReLU Proposta nel 2016 da Guberman e basata anch'essa sulla ReLu, é definita come:

$$zReLU\left(z\right) = \begin{cases} z & \text{se}\theta_z \in \left[0, \frac{\pi}{2}\right] \\ 0 & \text{altrimenti} \end{cases} \tag{54}$$