Reverse Mode Automatic Differentiation

Matteo Brunello

Consideriamo il grafo computazionale della equazione $f = (a + b) \cdot b$.

$$\begin{aligned} y &= v_2 \\ v_2 &= v_1 \cdot v_0 \\ v_1 &= v_{-1} + v_0 \\ v_0 &= b \\ v_{-1} &= a \end{aligned}$$

Vogliamo calcolare la derivata di ogni nodo intermedio rispetto al valore della funzione f, chiamato *adjoint*. Utilizzando semplicemente la regola di derivazione a catena

$$\frac{\partial f}{\partial v_i} = \sum_j \frac{\partial f}{\partial u_j} \cdot \frac{\partial u_j}{\partial v_i}$$

per tutti i nodi intermedi i. A questo punto uno potrebbe chiedersi il perche' di una sommatoria. Di fatto, quando applichiamo la regola di derivazione a catena, dobbiamo scegliere un *"percorso"* all'interno del grafo, scegliendo tutte le quantita' intermedie rispetto a cui derivare per arrivare fino in fondo. Facciamo un esempio e ipotizziamo di voler calcolare $\bar{b} = \bar{v_0}$. Notiamo subito che abbiamo due modi per arrivare fino in fondo al grafo:

$$\frac{\partial f}{\partial v_2}\frac{\partial v_2}{\partial v_1}\frac{\partial v_1}{\partial v_0} \text{ oppure } \frac{\partial f}{\partial v_2}\frac{\partial v_2}{v_0}$$

Scegliendo solo un percorso, non teniamo in conto di fatto il contributo che puo' dare al risultato della funzione *"passando"* per l'altro percorso, per cui risolviamo il problema sommandoli, ottenendo

$$\begin{aligned} \frac{\partial f}{\partial v_0} &= \frac{\partial f}{\partial v_2}\frac{\partial v_2}{\partial v_1}\frac{\partial v_1}{\partial v_0} + \frac{\partial f}{\partial v_2}\frac{\partial v_2}{v_0} \\ &= \bar{v}_2\frac{\partial v_2}{\partial v_1}\frac{\partial v_1}{\partial v_0} + \bar{v}_2\frac{\partial v_2}{v_0} \\ &= 1 \cdot v_0 \cdot 1 + 1 \cdot v_1 \\ &= v_0 + v_1 \end{aligned}$$

Ora che abbiamo giustificato intuitivamente la formula, andiamo avanti e utilizziamola a questo punto per calcolare tutti gli *aggiunti* (*adojoints*). Sta volta pero' partiamo dall'alto verso il basso, in modo che in caso comparissero delle definizioni

di aggiunti calcolati precedentemente possiamo eventualmente sostituirli.

$$
\begin{aligned}
\bar{f} &= 1 \\
\bar{v}_2 &= \bar{f} \cdot \frac{\partial f}{\partial v_2} = 1 \\
\bar{v}_1 &= \bar{v}_2 \cdot \frac{\partial v_2}{\partial v_1} = v_0 \\
\bar{v}_0 &= \bar{v}_1 \cdot \frac{\partial v_1}{v_0} + \bar{v}_2 \cdot \frac{\partial v_2}{\partial v_0} = v_0 + v_1 \\
\bar{v}_{-1} &= \bar{v}_1 \cdot \frac{\partial v_1}{\partial v_{-1}} = v_0
\end{aligned}
$$

Osserviamo che per calcolare un singolo $\bar{v}_i$ gli ingredienti principali sono:

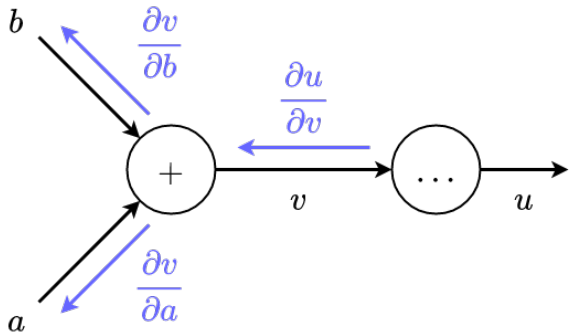
- Gli aggiunti dei valori v_j intermedi in cui compare almeno una volta v_i
- I parziali rispetto agli input del passo successivo (essenzialmente, quanto cambia la prossima funzione in cui compare v_i rispetto a v_i)

E' evidente che ad ogni passo, queste quantita' le posso calcolare solamente al passo successivo (per questo sostituiamo le definizioni). Per questa ragione partiamo dall'output e scendiamo verso gli input, propagando ai propri i vari parziali/aggiunti e moltiplicandoli/aggiungendoli man mano fino ad arrivare ai nodi di input. Il singolo nodo quindi ricevera' gli ingredienti necessari per calcolare il proprio aggiunto, lo calcola e si salva il valore, per poi propagare all'indietro ad ogni input il suo parziale corrispondente.

Ricapitolando, se io ho un nodo, ad esempio $v = a + b$, che a sua volta sara' l'input di un nodo intermedio quasiasi $(\dots)$ (con i puntini intendo che puo' esserci qualsiasi operazione tra $+, -, /, \cdot$ ecc) che produce il risultato u, per calcolare il suo aggiunto $\bar{v}$ necessito:

- L'aggiunto $\bar{u}$
- La derivata $\frac{\partial u}{\partial v}$

Notiamo come l'unico a poter sapere la derivata parziale $\frac{\partial u}{\partial v}$ e' u, per cui essenzialmente lo propaga all'indietro (della serie: *il mio input e' il tuo output*). La situazione e' rappresentata graficamente nella figura seguente.



Si noti inoltre che le derivate rispetto ai propri input sono delle espressioni che sono definite in termini degli input, per cui ogni nodo dovra' salvarsi o i valori di input che riceve, oppure direttamente le derivate rispetto agli input. Questi sono

dettagli dal punto di vista implementativo ma vale la pena menzionarli. Una prima opzione e' appunto salvare gli input nei singoli nodi, per poi utilizzare questi valori per calcolare i parziali da propagare indietro, mentre un'alternativa e' appunto quella di calcolare direttamente i parziali rispetto agli input nel passo di forward in avanti (senza propagarle in avanti) e salvandole opportunamente per poterle propagare all'indietro nel passo di backward.

Questo per il parziale, ma invece per gli aggiunti? Anche qui possiamo fare due cose. La prima consiste nel calcolare l'aggiunto ogni volta che siamo in un nodo per poi propagarlo all'indietro cosi' come facciamo per il parziale (anche moltiplicandolo con esso volendo), oppure creiamo un *tape*, cioe' proprio una lista condivisa che mantiene i valori degli aggiunti mano a mano che scendiamo verso i nodi (variabili) di inputs.

Di seguito sono riportate alcune signatures possibili corrispondenti ai metodi descritti:

- `v.backward(adjoint, dudv)` – Soluzione con accumulo degli aggiunti in ogni nodo
- `v.backward(dudv)` – Soluzione con accumulo in un *tape*