

Neural Network FrameWork

14 marzo 2005

1 L'Idea

Il neural network framework (NNF) consente di poter costruire una rete neurale artificiale sulla base di elementi di costruzione (Building-blocks) ed attraverso regole di composizione dei building-blocks.

La rete neurale è, quindi, il risultato della combinazione dei vari elementi.

Il tutto è pensato sia per dare enorme flessibilità ed estensibilità delle architetture possibili, sia per mantenere al massimo l'efficienza nella simulazione della rete neurale.

1.1 Building-Blocks

Gli elementi di costruzione si suddividono in tre classi:

- **Cluster:** è un oggetto che contiene più neuroni al suo interno. Definisce dei neuroni d'ingresso e dei neuroni d'uscita. Questi due insiemi non sono necessariamente identici; alcuni neuroni possono essere solo d'ingresso o solo d'uscita o nessuno dei due (nascosti all'interno del cluster). Vedere l'InnerLinker per capire come usare i neuroni nascosti del cluster.
- **Linker:** è l'oggetto che permette di connettere due o più cluster insieme. In generale, il linker prende gli output dei neuroni di uno o più cluster (denominati *cluster d'ingresso*) e li inserisce come input dei neuroni di un'altro cluster (denominato *d'uscita*). Possono esistere varie tipologie di linker a seconda del tipo di connessione che si vuole realizzare, alcuni esempi sono:
 - Connessione completa fra i neuroni dei cluster in ingresso con quello di uscita;
 - Connessione sparsa con probabilità P tra i neuroni del cluster in ingresso con quello d'uscita;

- Connessioni che simulano sinapsi dinamiche;
- Connessioni con *time-delay*
- **InnerLinker**: questo oggetto è importante per le connessioni all'interno dei neuroni di un cluster. Questo consente di connettere i neuroni che sono soltanto d'ingresso di un cluster con quelli che sono solo d'uscita o nascosti. Un esempio dell'utilità dell'InnerLinker è la mappa di Kohonen o le reti di Hopfield. Infatti, un InnerLinker può connettere i neuroni di un cluster a formare una rete di Hopfield ed attraverso i Linker usare solo alcuni neuroni come ingresso e altri come output.

1.2 Regole di Costruzione

Semplicemente si creano i vari cluster e li si connettono come desiderato.

Ad ogni cluster inserito all'interno della rete verrà associato un identificativo numerico, così come ai Linker e InnerLinker.

1.3 La Rete Neurale

La rete neurale è un oggetto che contiene i vari cluster e linker connessi tra loro.

Ad ogni Cluster è associato un livello che ne determina la gerarchia strutturale. Il livello 0 è pensato come il livello contenente i neuroni d'ingresso della rete neurale, e l'ultimo livello come l'output della rete neurale (anche se questo non è vincolante). Almeno il livello 0 deve esistere !!! La numerazione dei livelli non ammette 'buchi'.

La rete neurale contiene un orologio *temporale* che tiene 'il tempo' della rete neurale o degli *steps* eseguiti. Questo consente di avere sia una dinamica discreta che continua all'interno della rete neurale. Infatti, le varie regole di aggiornamento dei linker potrebbero basarsi sia sul tempo che sugli steps della rete neurale

Non ho ben chiaro se la rete neurale deve avere i metodi per manipolare l'architettura, oppure solamente contenere i cluster e i linker e delegare la creazione della struttura ad un *ArchitectureManager* che offre i metodi necessari a manipolare i cluster e i linker.

1.4 Spreading della rete

Lo spreading della rete si effettua aggiornando i Cluster, i Linker e gli InnerLinker della rete neurale costruita.

Ogni Cluster, sulla base della funzione d'attivazione dei neuroni, aggiorna gli output dei neuroni sulla base dell'input attuale.

Ogni Linker preleva gli output dei neuroni dei cluster d'ingresso, effettua i calcoli necessari e registra gli input dei neuroni del cluster d'uscita.

Ogni InnerLinker aggiorna gli input dei neuroni del cluster sulla base degli input al tempo di calcolo. Gli InnerLinker sono intrinsecamente ricorrenti, e quindi terranno conto di questo.

Il coordinamento fra queste operazioni dipende dalla particolare strategia scelta. Ad esempio, nella strategia *feedforward* si partirà aggiornando i cluster di livello 0 e successivamente i Linker, e così via fino all'ultimo livello.

2 Come Mantenere l'efficienza

Per consentire che questo meccanismo non perda d'efficienza durante lo spreading della rete, i Linker e gli InnerLinker dovranno avere accesso alla struttura interna che contiene gli output e gli input dei neuroni del cluster. Quindi, andranno fatte delle assunzioni forti all'interno della classe Cluster che tutte le classi derivate dovranno rispettare.

Durante l'aggiornamento di un Linker, questo accede direttamente alla struttura interna così da evitare la chiamata a funzione per ottenere il vettore degli output e dell'input dei neuroni dei cluster.

Non ho ancora idea di come far condividere le strutture di memoria senza al tempo stesso farle dipendere da una implementazione specifica !!! :(Forse i template mi potrebbero aiutare... o forse no !!

In ogni caso, forse in prima istanza un semplice array statico del C++ creato attraverso l'uso di un template potrebbe risolvere la questione. Ad esempio, un cluster di un certo numero di neuroni potrebbe essere una classe template che accetta il numero di neuroni che deve contenere. Si andrebbe a realizzare, così, un cluster statico, non ridimensionabile... ma che sarebbe veloce !!!

O forse in prima istanza basta una classe che alloca memoria dal costruttore e poi non permette la modifica a run-time. O forse si ?!?! Mah :-)

3 Si possono realizzare tutte le reti immaginabili ???

Bè, a prima vista direi di sì !!

- E' possibile creare strutture ibride a piacimento; Anche mischiare continue e discrete insieme... forse in futuro chissà !! :-)
- Reti con sinapsi dinamiche o ritardate sia discrete che continue
- Gli spiking neuron anche ?!?! Ummm... ci devo pensare un pò

- Le reti tipo la GasNet, che richiedono una posizione spaziale dei neuroni... potrebbe essere implementata attraverso un cluster che contiene un solo neurone che definisce una posizione spaziale del cluster.... O meglio ancora con un cluster con una sua dimensione spaziale interna e che organizza i neuroni all'interno nello spazio del cluster (bi- o tri-dimensionale) ed usare un InnerLinker derivato o direttamente l'update del cluster stesso per gestire le attivazioni dipendenti dalla posizione spaziale come nelle GasNet.
- Le Ricorrenze... quante e quali più vi piacciono !! Nelle reti in cui vengono copiati gli output di certi neuroni come input di altri neuroni possono essere realizzate attraverso dei Linker che connettono uno-a-uno che semplicemente eseguono una copia dal cluster d'ingresso a quello d'uscita.