

Belief Networks

Con la backpropagation rischiamo di bloccarci in ottimi locali, che spesso danno buoni risultati. Nelle reti profonde però, gli ottimi locali nei quali si potrebbe incappare inizializzando i pesi a piccoli valori casuali, possono trovarsi anche molto lontano dall' ottimo globale.

La soluzione è quindi quella di usare la discesa del gradiente in combinazione con l'apprendimento non supervisionato, quindi per creare una rappresentazione interna dell' input e non per creare una correlazione tra un input ed un output.

Modello generativo: modello che calcola il valore da assegnare a determinate variabili in base al valore osservato di altre.

Più tecnicamente si cerca di adattare i pesi per massimizzare la probabilità che il modello generativo (la rete) riesca a rigenerare l' input.

Il modello generativo necessario sarà un ibrido tra quello generativo e quello a energia globale.

Reti neurali con modelli generativi:

Modelli basati sull' energia: esempio migliore sono le Restricted Boltzmann Machine, in cui però dobbiamo limitarci un singolo strato di neuroni nascosti per rendere possibile l' apprendimento.

Modelli causali: si utilizzano grafi diretti ed aciclici, un esempio sono le Sigmoid Belief Net (introdotte da Neal nel 1992.)

Belief Nets: Modelli grafici di dipendenze tra variabili

Belief Nets: sono grafi sparsi (con pochi nodi), diretti, aciclici e connessi che rappresentano le dipendenze tra nodi, i quali sono composti da variabili di tipo stocastico. Rendono possibile un efficiente calcolo probabilistico dei valori di nodi non direttamente osservabili sulla base del valore di altre variabili dipendenti. (L' algoritmo diventa inefficiente su grafi densi)

Nell' immagine di esempio immaginiamo di poter osservare solo le variabili rosse e dover:

- Ipotizzare il valore delle variabili non osservate/osservabili (quelle blu, ed è più semplice se sono indipendenti)
- Stabilire il peso dei collegamenti in modo tale da ricreare il valore dell' input proposto alla rete (stabilire quale nodo interno viene manipolato da quale nodo esposto ed in che misura)

Sigmoid Belief Net: sono delle Belief Net in cui le variabili corrispondono a neuroni binari stocastici. Si è dimostrato che sono più facilmente addestrabili delle Boltzmann Machine (non Restricted). Il processo di infrenza procede sequenzialmente dall' alto verso il basso. I neuroni più in alto, in base al loro stato (che dipende dal bias) generano stocasticamente quello del livello inferiore, fino ai neuroni visibili (che non hanno bias) che ridanno forma al vettore dati usato come campione di apprendimento.

Algoritmo di apprendimento delle Belief Network

Appunti grezzi:

- Osservano il fenomeno (dato del training set)

- Danno delle interpretazioni blande (stima basata su presupposti errati del valore degli stati dei neuroni hidden)
- Usano queste interpretazioni per creare un modello che rappresenti la realtà
- Il modello interpretativo viene modificato in base ai nuovi fenomeni

Introduzione teorica:

Probabilità a Priori/Posteriori

Probabilità a Posteriori: è la probabilità che un evento accada a seguito di prove eseguite e informazioni raccolte.

Probabilità a Priori: è la probabilità che un evento accada, prima di aver raccolto nuove prove a riguardo.

La probabilità a posteriori è un miglioramento di quella a priori attraverso nuove prove (chiamate **likelihood**).

Esempio1: sappiamo che la probabilità di attivare un neurone è di 0.3 (prob. a priori). Sottoponiamo il vettore ad un dato che lo triggera (likelihood) e otteniamo una prob a posteriori di 0.5

Esempio2: sappiamo che il 50% degli studenti non si laureano in tempo (prob. a priori). Facciamo nuovi sondaggi (likelihood) e scopriamo che la vera percentuale è del 80% (prob. a posteriori).

L'idea di base dell' algoritmo è quella di approssimare la probabilità a posteriori e di massimizzare poi il likelihood. Durante l' apprendimento i pesi vengono mossi da due principi:

- L' adattamento ai dati, fornire un buon modello di rappresentazione e quindi avvicinarsi ad essi e riuscire a replicarli
- L'adattamento della probabilità a posteriori stimata inizialmente con la vera probabilità a posteriori (in realtà fa in modo di manipolare la reale probabilità a posteriori).

Quindi crea un modello che rappresenta i dati e contemporaneamente si aggiusta i parametri in modo tale che i risultati prodotti dal modello siano quelli osservati appena prima.

E' complicato ottenere una reale distribuzione di priorità a posteriori dei neuroni nascosti anche dopo aver fornito il vettore dati. Quindi assumiamo (SBAGLIANDO) di poter calcolarla/approssimarla, per ogni stato, moltiplicando le probabilità dei neuroni hidden tra loro (l' errore sta nel fatto che, a differenza delle RBM, in questo caso i neuroni hidden non sono indipendenti tra loro).

Esempio pratico:

Immaginiamo di avere 3 neuroni con prob: (0.3 , 0.6, 0.8).

La stima di ottenere probabilità a posteriori pari a (1, 0, 1) = $0.3 * (1 - 0.6) * (0.8)$

Probabilità di ottenere attivare tutti i neuroni tranne quello centrale = $0.096 = 9\%$

The wake-sleep algorithm (Hinton nel 1995)

Ci sono due differenti pesi,

- Il **peso generativi**: servono per generare la distribuzione di probabilità del vettore dati
- I **pesi di riconoscimento**: servono per approssimare la distribuzione di probabilità dei neuroni hidden.

Wake phase: mostri i dati allo strato visibile della rete (in basso) usando i **pesi di riconoscimento** e per ogni livello aggiorni stocasticamente e in modo indipendente tra loro, gli stati dei neuroni nascosti che incontri.

A questo punto tratti questi stati come se fosse generati correttamente dal vettore dati (non lo sono perché il calcolo stocastico fa assunzioni errate, è una semplificazione del reale calcolo) e li usi per applicare l' algoritmo di maximum likelihood learning per trovare i **pesi generativi** del modello.

Sleep phase: Si fa l'opposto, si generano i **pesi di riconoscimento** utilizzando i **pesi generativi**. Si inizia inizializzando casualmente il livello nascosto più alto della rete e generando mano mano gli stati dei neuroni che incontri durante la discesa fino a generare il campione del vettore dati nel livello in fondo. A quel punto usi i livelli precedenti per costruire i **pesi di riconoscimento** che li legano ai livelli superiori.

Calcolo probabilità di attivazione di un neurone visibile, conoscendo lo stato dei neuroni nascosti associati ed il loro peso: