Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação

Máquinas Estocásticas e suas Aproximações Baseadas na Mecânica Estatística

Redes Neurais 2006

Prof. Dr. Zhao Liang

Grupo:

Fabrício Breve
João R. Bertini Jr.
Renato Fernandes

Sumário

Máquinas Estocásticas e suas Aproximações Baseadas na Mecânica Estatística

1. Introdução

O tema *mecânica estatística* abrange o estudo formal das propriedades macroscópicas do equilíbrio de grandes sistemas de elementos que estão sujeitos às leis microscópicas da mecânica. São usados métodos probabilísticos devido ao alto número de graus de liberdade nesses sistemas.

O interesse na utilização da mecânica estatística como base para o estudo de redes neurais se iniciou com a *máquina de Boltzmann*, a primeira máquina de aprendizagem em múltiplas camadas inspirada pela mecânica estatística. Basicamente, a máquina de Boltzmann é um dispositivo para modelar a distribuição de densidade de probabilidade de um determinado conjunto de dados, de modo que possam ser derivadas distribuições condicionais que possam ser usadas em tarefas como complementação e classificação de padrões.

2. Mecânica Estatística

Dado um sistema físico com muitos graus de liberdade, podendo residir em qualquer estado de um grande número de estados possíveis, temos que a probabilidade de ocorrência de um estado *i* é dado por:

$$p_i \geq 0 \qquad \text{para todo } i$$

e

$$\sum_i p_i = 1$$

Considerando que E_i representa a *energia* do sistema quando está no estado *i*. a probabilidade de ocorrência de um estado é dada por

$$p_i = \frac{1}{Z}\exp\left(-\frac{E_i}{T}\right)$$

onde *T* é uma temperatura que controla as flutuações térmicas que representam o efeito de *ruído sináptico* de um neurônio. Também temos uma quantidade normalizadora *Z*, chamada de *soma dos estados* ou *função de partição*, dada por:

$$Z = \sum_i \exp\left(-\frac{E_i}{T}\right)$$

Essa distribuição de probabilidade é chamada *distribuição de Gibbs* e tem como características:

- Estados de baixa energia têm maior probabilidade de ocorrência que estados de alta energia
- Conforme a temperatura *T* é reduzida, a probabilidade é concentrada em um conjunto menor de estados de baixa energia

2.1 Energia Livre e Entropia

A *energia livre* de um sistema físico, representada por *F*, é dada por:

$$F = -T \log Z$$

A *energia média* do sistema é dada por:

$$< E >= \sum_i p_i E_i$$

Portanto a energia livre é dada por:

$$< E > -F = T \sum_i p_i \log p_i$$

O lado direito da equação, com execeção de *T*, é a entropia do sistema, dada por *H* como segue:

$$H = -\sum_i p_i \log p_i$$

Assim podemos reescrever da seguinte forma:

$$F =< E > -TH$$

A energia livre do sistema, *F*, tende a diminuir e se tornar mínima na situação de equilíbrio térmico do sistema. O mínimo de energia de um sistema estocástico em relação às variáveis do sistema é alcançado no equilíbrio térmico, onde então o sistema é governado pela distribuição de Gibbs.

2.2 Neurônios Estocásticos

Os neurônios estocásticos, diferentemente dos tradicionais, mudam seus estados de maneira probabilística, em vez de apenas determinística. Podem estar ligados (+1) ou desligados (-1). A vantagem é que a rede nunca fica parada em um estado estável, pois os neurônios estarão sempre mudando, mesmo que a entrada não mude. Com a rede rodando livremente podemos gravar os estados por quais ela passa e construir uma distribuição de probabilidade destes estados.

Para exemplificar o funcionamento de um neurônio estocástico considere uma rede de Hopfield tradicional, onde cada neurônio muda para o estado cujo energia for menor, conforme a regra abaixo:

$$\Delta E_i = E_i(-1) - E_i(+1) = \sum_j w_{ij} s_j$$

Se $\Delta E < 0$ o neurônio fica inativo

Se $\Delta E > 0$ neurônio fica ativo

Se modificarmos a regra de atualização para fazer desse neurônio um neurônio estocástico, ela ficaria assim:

$$p_i(+1) = \frac{1}{1 + e^{\Delta Ei/T}}$$

onde *T* é uma "temperatura" da rede, seguindo a nomenclatura do modelo físico em que foi inspirada.

Dessa forma o neurônio normalmente irá para o estado que reduz a energia do sistema, mas algumas vezes ele irá para o "estado errado" (da mesma forma que um sistema físico às vezes vai para estados de maior energia). Quanto maior for a temperatura *T* maior é a probabilidade de que o sistema possa ir para um estado de maior energia. Com temperatura zero o comportamento será igual ao neurônio determinístico.

Deixando a rede rodar por tempo suficiente é possível obter a distribuição de probabilidade dos estados que ela visita, porém é necessário esperar o equilíbrio térmico para fazer essa medida, ou seja, quando a média ativação do i-ésimo neurônio <Si> não estiver mais mudando com o tempo. A distribuição de probabilidade no equílibro térmico pode ser similar ao mundo real se mudarmos os pesos de conexões na rede de maneira correta.

Uma das principais vantagens dessa versão estocástica é que o estado inicial é irrelevante, pois enquanto a temperatura for relativamente alta sempre será possível "escapar" dos mínimos locais. Enquanto que usando neurônios determinísticos tradicionais o estado inicial é crucial para definir se a solução encontrada será ótima ou apenas um mínimo local.

3. Algoritmo de Metropolis

O algoritmo de Metropolis foi apresentado inicialmente por Metropolis em [Metropolis *et al.* 1953] e generalizado por Hastings [Hasting 1970] resultando no algoritmo de Metropolis-Hastings. Esse método é usado geralmente quando é difícil gerar amostras da condicional completa a posteriori . Neste caso, gera-se valores do parâmetro a partir de uma distribuição proposta e esse é aceito ou não com uma certa probabilidade de aceitação.

Suponha que a variável randômica Xn representa uma cadeia de Markov arbitraria que está no estado xi no tempo n. Será gerado um novo estado xj, representando a realização de outra variável randômica Yn. É assumido que a geração deste nova estado satisfaz a condição simétrica:

$P(Y_n = x_j \mid X_n = x_i) = P(Y_n = x_i \mid X_n = x_j)$

Seja ΔE a diferença de energia resultante da transição do sistema do estado $X_n = x_i$ para o estado $Y_n = x_j$:

- Se a diferença é negativa, então a transição esta levando para um estado com menor energia e a transição é aceita.
- Se a diferença é positiva, então o algoritmo procede em uma maneira probabilista para este ponto:
    - Primeiro, seleciona um numero randômico ξ uniformemente distribuído entre [0,1]
    - Se $\xi < \exp(-\Delta E/T)$, a transição é aceita e Xn+1 = Yn
    - Se não a transição é rejeitada e Xn+1 = Xn (que é a configuração antiga será utilizada na próxima iteração)

Escolhendo a probabilidade de transição

Seja uma cadeia de Markov que tem uma probabilidade de transição inicial de τ_{ij} que satisfaz três condições:

1) Não-negatividade

$\tau_{ij} \geq 0$ para todo (i,j)

2) Normalização

$\sum_j \tau_{ij} = 1$ para todo i

3) Simetria

$\tau_{ij} = \tau_{ji}$ todo (i,j)

steady state

Seja π_i a probabilidade que a cadeia de Markov esta no estado x_i, i=1,2,...,K. Nos podemos usar a simétrica τ_{ij}s e a taxa de distribuição de probabilidade π_j / π_i para formular o desejado conjunto de probabilidade de transição como:

$$p_{ij} = \begin{cases} \tau_{ij}\left(\frac{\pi_j}{\pi_i}\right) para \frac{\pi_j}{\pi_i} < 1 \\ \tau_{ij} \ldots para \frac{\pi_j}{\pi_i} \geq 1 \end{cases}$$

Na transição temos:

$$p_{ij} = 1 - \sum_{j \neq i} \alpha_{ij} \tau_{ij}$$

onde

$$\alpha_{ij} = \min\left(1, \frac{\pi_j}{\pi_i}\right)$$ e a probabilidade de movimento

Agora é necessário escolher a taxa $\frac{\pi_j}{\pi_i}$. Caso escolhemos a distribuição de probabilidade desejada para convergir como sendo uma distribuição de Gibbs, então:

$$\frac{\pi_j}{\pi_i} = \exp\left(-\frac{\Delta E}{T}\right)$$

A probabilidade de transição é o modelo probabilístico do “passo randômico” do algoritmo de metropolis. O “passo” é seguido de uma decisão randômica. A probabilidade de transição p_{ij} definida em termos da probabilidade de transição a priori τ_{ij} e da probabilidade de estado fixo π_j são de fato as escolhas que vão fazer o algoritmo Metropolis ter sucesso ou não.

4. Simulated Annealing

Em uma rede neural, o objetivo é muitas vezes de minimizar a função custo definida como a energia global de uma rede. O numero de neurônios contidos na rede é muito grande. Assim, encontrar a solução de energia mínima de uma rede não é diferente do que encontrar o estado de temperatura mínima de um sistema físico.

Algoritmo determinísticos usado para a minimização de energia sofre de uma falha fundamental do procedimento de gradiente descente, onde o algoritmo pode ficar "parado" em um "mínimo local" que não é ótimo globalmente. Este problema aparece principalmente quando alguns dos neurônios da rede são externamente forçados a "segurar" algum padrão de entrada e é necessário encontrar uma solução de energia mínima compatível com aquele padrão de entrada particular. Neste caso, a rede precisa ser capaz de escapar do mínimo local, para alcançar a configuração que representa o mínimo global, para conseguir chegar no padrão de entrada desejado.

Simulated Annealing é uma classe de meta-heurística proposta originalmente por Kirkpatrick [Kirkpatrick *et al.* 1983], sendo uma técnica de busca local probabilística, que se fundamenta em uma analogia com o procedimento em sistemas físicos com muitos graus de liberdade em equilíbrio térmico, operação conhecida como recozimento. Neste processo físico, um sólido (por exemplo na fabricação de pecas de cristal) é aquecido a uma temperatura alta (ate um valor Máximo) onde todas as partículas do sólido são auto arranjadas randomicamente na fase liquida. Então a temperatura é diminuída lentamente, permitindo todas as partículas se arranje de forma a encontrar o melhor arranjo. Se a temperatura descer bruscamente o sólido pode apresentar falhas (no caso o cristal não terá uma ordem cristalina, ficando um objeto com defeitos).

4.1 Descrição do Algoritmo

Esta técnica começa sua busca a partir de uma solução inicial qualquer. O procedimento principal consiste em um loop que gera aleatoriamente, em cada iteração, um único vizinho $\boldsymbol{s}'$ da solução corrente $\boldsymbol{s}$.

A cada geração de um vizinho s´ de s, é testada a variação Δ do valor da função objetivo (f), isto é: $\Delta = f(s') - f(s)$.

Se $\Delta < 0$, o método aceita a solução e s´ passa a ser a nova solução corrente.

Se $\Delta \geq 0$, a solução vizinha candidata também poderá ser aceita, mas neste caso, com uma probabilidade exp(-Δ/T), onde T é um parâmetro do método, chamado de temperatura e que regula a probabilidade de aceitação de soluções com custo pior.

A temperatura T assume, inicialmente, um valor elevado T_0. Após um numero fixo de iterações (o qual representa o numero de iterações necessárias para o sistema atingir o equilíbrio térmico em uma dada temperatura), a temperatura é gradativamente diminuída por uma razão de resfriamento α, tal que :

$T_n \leftarrow \alpha * T_{n-1}$, sendo $0 < \alpha < 1$.

Com este procedimento , dá-se, no inicio uma chance maior para escapar de mínimos locais e, à medida que T aproxima-se de zero, o algoritmo comporta-se como o método de descida, uma vez que diminui a probabilidade de se aceitar movimentos de piora

(para $T \rightarrow 0$ entao $\exp(-\Delta/T), \rightarrow 0$.

O procedimento pára quando a temperatura chega a um valor próximo de zero e nenhuma solução que piore o valor da melhor solução é mais aceita, isto é, quando o sistema esta estável. A solução obtida quando o sistema encontra-se nesta situação evidencia o encontro de um mínimo local.

Os parâmetros de controle do procedimento são a razão de resfriamento α, o numero de iterações para cada temperatura e a temperatura inicial T_o.

5. Amostragem de Gibbs

Considere um vetor aleatório X de dimensionalidade *K* constituído das componentes $X_1, X_2, ..., X_K$. Suponha que tenhamos conhecimento da distribuição condicional de X_k, dado os valores de todas as outras componentes de X para $k = 1,2,...,K$. O amostrador de Gibbs atua gerando um valor para a distribuição condicional para componente do vetor aleatório $\mathbf{X}$, dados os valores de todas as outras componentes de X. Por exemplo, partindo de uma configuração qualquer $\{x_1(0), x_2(0), ..., x_K(0)\}$, a primeira iteração da amostragem de Gibbs é dada por:

$x_1(1)$ é retirado da distribuição de X_1, dados $x_2(0), x_3(0),..., x_K(0)$
$x_2(1)$ é retirado da distribuição de X_2, dados $x_1(1), x_3(0),...x_K(0)$
⋮
$x_k(1)$ é retirado da distribuição de X_k, dados $x_1(1),..., x_{k-1}(1), x_{k+1}(0),...,...x_K(0)$
⋮
$x_K(1)$ é retirado da distribuição de X_K, dados $x_1(1), x_2(1),..., x_{k-1}(1)$

Procedemos da mesma maneira na segunda e demais iterações. Dois pontos devem ser considerados:

- Cada componente do vetor aleatório X é "visitada" na ordem natural, com o resultado de *K* novas variantes geradas a cada iteração
- O novo valor da componente *XK*-1 é usado imediatamente quando um novo valor de *XK* é retirado para *k* = 2, 3, ..., *K*

Portanto dizemos que o amostrador de Gibbs é um esquema *adaptativo iterativo*. Após *n* iterações de seu uso, chegamos a *K* variantes: $X_1(n), X_2(n),..., X_K(n)$.

Sob condições suaves são válidos três teoremas para a amostragem de Gibbs:

1. *Teorema de Convergência*: a variável aleatória $X_k(n)$ converge em distribuição para as distribuições verdadeiras de X_k para $k = 1,2,..., K$ quando *n* se aproxima do infinito; ou seja,

   $$\lim_{n\to\infty} P\left(X_k^{(n)} \le x \mid x_k(0)\right) = F_{X_k}(x) \quad \text{para } k = 1,2,..., K$$

   onde $F_{X_k}(x)$ é a função distribuição de probabilidade marginal de X_K.

2. *Teorema da taxa de convergência*: A distribuição de probabilidade conjunta das variáveis aleatórias $X_1(n), X_2(n),..., X_K(n)$ converge para a distribuição de probabilidade conjunta verdadeira de $X_1, X_2,,,,, X_K$ em uma taxa geométrica de *n*.

3. *Teorema Ergódico*: Para qualquer função mensurável *g*, por exemplo, das variáveis aleatórias $X_1, X_2,,,,, X_K$ cujo valor esperado exista, temos

   $$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} g(X_1(i), X_2(i),..., X_k(i)) \to E[g(X_1, X_2,..., X_K)]$$

   com probabilidade 1 (quase certeza)

6. A Máquina de Boltzmann

A máquina é composta por neurônios estocásticos conectados por ligações bidirecionais. Um neurônios está sempre um de dois estados, ativado (1) ou desativado (-1), e esses estados é dado por uma função probabilística de acordo com a equação abaixo.

$$p_i(+1) = \frac{1}{1 + \exp(\Delta E_i / T)}$$

Onde T é o parâmetro temperatura. Os pesos podem tomar valores reais positivos ou negativos. O peso em uma ligação representa uma restrição fraca entre duas hipóteses. Um peso positivo indica que as duas hipóteses tendem a suportar um outro; se um for aceito atualmente, aceitar o outro deve ser mais provável. De maneira análoga, um peso negativo sugere, outras coisas que são iguais, que as duas hipóteses não devem ser aceitas. Os pesos das ligações são simétricos, tendo a mesma força em ambos os sentidos [Hinton & Sejnowski, 1983]. Um modelo geral da máquina de Boltzmann é mostrado na Figura 1.

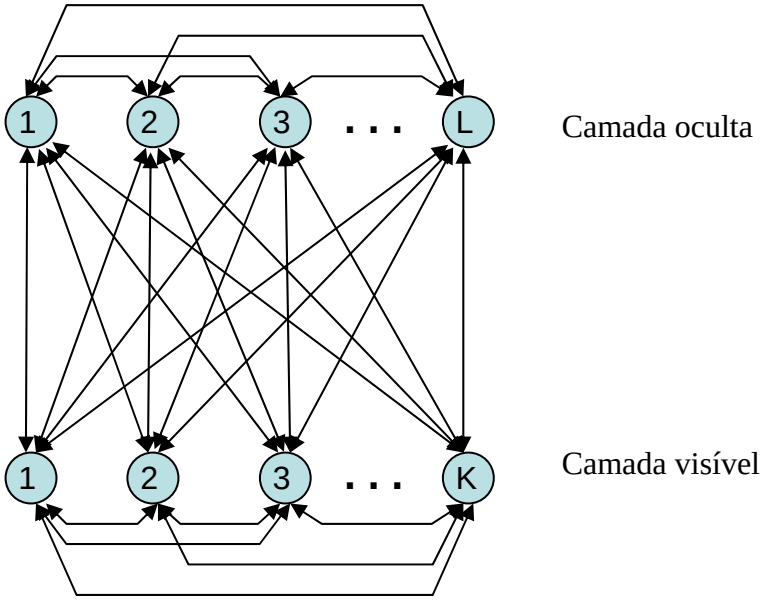

Figura 1 – Arquitetura geral de uma máquina de Boltzmann

Considere um conjunto de padrões α e a distribuição real $P\alpha$ sobre esses padrões. Para cada atributo nestes vetores de padrão, é criado um neurônio visível na máquina de Boltzmann cuja atividade é associada ao valor deste neurônio. A camada oculta da máquina é criada com um número maior de neurônios usados na camada visível, na verdade para que a

representação de todos os padrões do conjunto seja armazenada corretamente o número de neurônios na camada oculta é dado por uma função exponencial da entrada. Todas as unidades em uma máquina de Boltzmann computam uma diferença de energia (*energy gap*):

$$\Delta E_i = E_{-1} - E_{+1} = \sum_j w_{ij} S_j$$

Se a rede executar por tempo suficiente, o sistema alcançará um ponto de baixa temperatura chamado ponto de equilíbrio térmico. Neste ponto a probabilidade da rede estar em qualquer estado depende somente de sua energia dividido por sua temperatura. Pode-se estimar a distribuição de probabilidade sobre as unidades visíveis nesta fase em que a rede executa livremente calculando as atividades médias < S > de todas as unidades visíveis. Seja esta medida de distribuição denotada por Pβ. Deseja-se que esta distribuição esteja próxima à distribuição desejada Pα. Para medir o quão próximo está às distribuições de probabilidades e consequentemente o desempenho da máquina, usa-se a distância de Kullback-Leibler [Kullback & Leibler 1951] entre as a distribuições Pα e Pβ:

$$G = \sum_\alpha P_\alpha \ln\left(\frac{P_\alpha}{P_\beta}\right)$$

Intuitivamente para aproximar as distribuições de probabilidades Pα e Pβ , e assim melhorar a representação do modelo em questão, basta minimizar a função distancia G como mostrado na equação a seguir, onde $\langle s_i s_j \rangle^+$ é probabilidade do neurônio i e do neurônio j estarem ativos na fase positiva e $\langle s_i s_j \rangle^-$ é o análogo na fase negativa.

$$\frac{\partial G}{\partial w_{ij}} = -\frac{1}{T}\left(\langle s_i s_j \rangle^+ - \langle s_i s_j \rangle^-\right)$$

6.1 Aprendizado na máquina de Boltzmann

Na máquina de Boltzmann toda a informação sobre como uma mudança particular do peso altera a energia G do sistema está disponível localmente. O procedimento de

aprendizagem para a máquina de Boltzmann tem duas fases, positiva e negativa. Na fase positiva, as unidades visíveis têm seu estado definido de acordo com o valor de um padrão de treinamento particular, e a rede é permitida a alcançar o estado térmico equilíbrio térmico. Então incrementa-se os pesos das conexões entre quaisquer dois neurônios que estejam ambos ativos, como na aprendizagem Hebbiana [Hebb 1949]. Esta fase é repetida um grande número vezes, para cada padrão de treinamento sendo apresentado de acordo com a distribuição de probabilidade que a rede deve aprender.

Já na fase negativa, a rede funciona livremente (nenhum neurônio de entrada tem seu estado imposto) e os estados dos neurônios são coletados até que a rede atinja o equilíbrio térmico.

Se as fases positiva e negativa forem alternadas com freqüência aproximadamente igual (na verdade a fase positiva deve ocorrer mais frequentemente), então o procedimento de aprendizado reduzirá, em média, a entropia cruzada entre a fase positiva e negativa da rede. Dessa forma os pesos da rede são atualizados como segue, onde η é a taxa de aprendizado, similar à usada no backpropagation.

$$\Delta w_{ij} = \eta\left(\left\langle s_i s_j \right\rangle^{+} - \left\langle s_i s_j \right\rangle^{-}\right)$$

7. A Máquina de Helmholtz

A máquina de Helmholtz [Dayan *et al.* 1995] e [Hinton *et al.* 1995] fornece uma estrutura em múltiplas camadas para representar e aprender as relações estatísticas de ordem mais alta entre as entradas sensoriais de interesse de uma maneira não-supervisionada, sem usar a amostragem de Gibbs.

Ela utiliza dois conjuntos totalmente diferentes de conexões sinápticas, conforme ilustrado na Figura 2, onde é mostrada uma rede com duas camadas de neurônios estocásticos e binários.

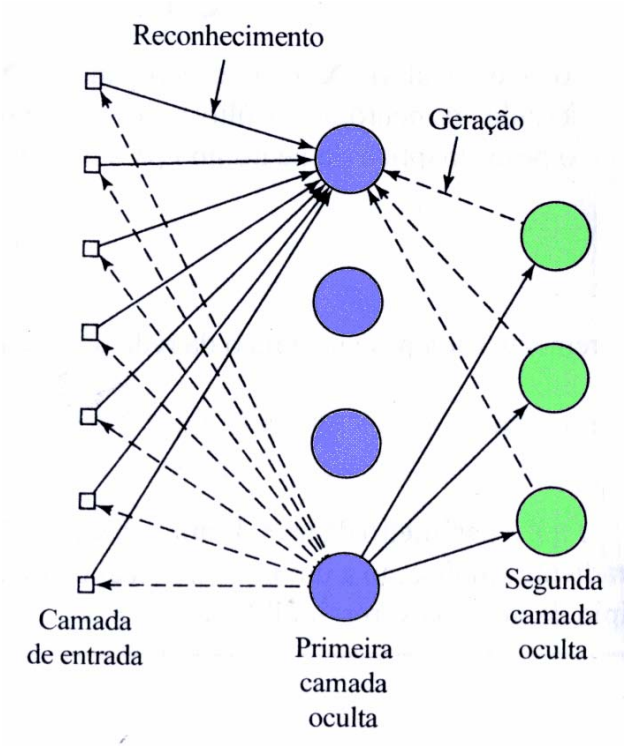

Figura 2. Grafo arquitetural da máquina de Helmholtz consistindo de neurônios conectados com conexões de reconhecimento (linhas sólidas) e de geração (linhas tracejadas)

As conexões para frente constituem o modelo de reconhecimento e inferem uma distribuição de probabilidade relacionada com as causas do vetor de entrada. As conexões de realimentação constituem o modelo de geração, e reconstroem uma aproximação do vetor de entrada original a partir das representações subjacentes capturadas pelas camadas ocultas da rede. Ambos os modelos de reconhecimento e geração trabalham exclusivamente com alimentação para frente (sem realimentação) interagindo apenas através do processo de aprendizagem.

O aprendizado se dá em duas fases, uma fase *acordada* e uma fase *adormecida*. Na fase acordada, a rede é acionada para frente pelos pesos de reconhecimento produzindo uma representação do vetor de peso na primeira camada oculta da rede. A segunda camada oculta produz uma segunda representação da primeira representação, e assim por diante. O conjunto dessas representações fornece uma representação global do vetor de entrada pela rede. Embora os neurônios sejam ajustados pelo peso de reconhecimento, apenas os pesos de geração são ajustados nessa fase usando a informação disponível localmente.

Na fase adormecida os pesos de reconhecimento são desligados. A rede é acionada na direção contrária pelos pesos de geração, iniciando na camada oculta mais externa e terminando na camada visível. Pelo fato de os neurônios serem estocásticos, a repetição desse processo provocaria tipicamente o surgimento de vetores “fantasiosos” na camada de entrada.

Estas fantasias forneceriam uma amostra sem viés do modelo de geração da rede sobre o mundo. Uma vez produzida a fantasia, os pesos de reconhecimento são ajustados por uma regra delta simples de modo a maximizar o logaritmo da probabilidade de recuperar as atividades ocultas que realmente causaram a fantasia. Nessa fase também é usada apenas informação disponível localmente.

A regra de aprendizagem para os pesos de geração também utiliza a regra delta simples, mas em vez de seguir o gradiente da função logaritmo da verossimilhança, segue o gradiente de uma função logaritmo da verossimilhança penalizada. O termo de punição é a divergência de Kullback-Leibler entre a distribuição a posteriori verdadeira e a distribuição real produzida pelo modelo de reconhecimento. O processo de aprendizagem tenta ajustar os pesos de geração para trazer a distribuição a posteriori real tão perto quanto possível da distribuição realmente calculada pelo modelo de reconhecimento. Aprender os pesos de reconhecimento não corresponde precisamente a função de verossimilhança penalizada, assim não é garantido que o procedimento de aprendizagem acordado-adormecido funcione em todas as situações práticas, ele falha algumas vezes.

8. Sigmoid Belief Network

Inicialmente sigmoid belief network (SBN) [Neal 1992] foi desenvolvida para compartilhar com a máquina de Boltzmann (BM) a capacidade de aprender a distribuição de probabilidade arbitrária sobre vetores binários, sem a necessidade da fase negativa no procedimento de aprendizado da BM.

Desta forma alterando as conexões simétricas da BM com conexões diretas que formam um gráfico acíclico (GAD). Portanto SBN tem uma arquitetura multi-camadas com neurônios estocásticos binários. A natureza acíclica a faz fácil de realizar cálculos probabilísticos. A Figura 3 ilustra a arquitetura geral de uma rede SBN.

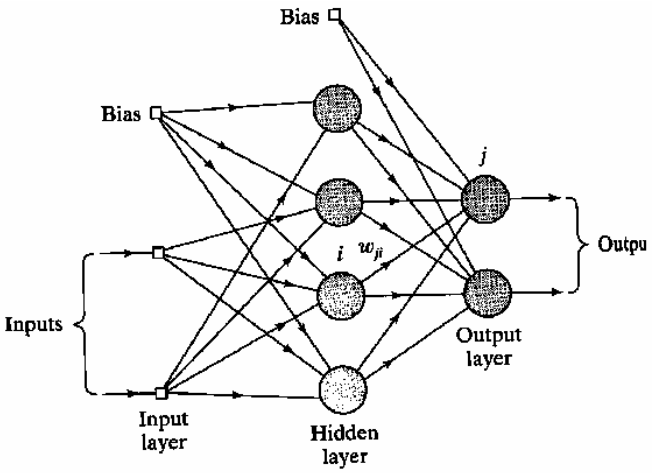

Figura 3 – Arquitetura geral de uma rede SBN

8.1 Motivações

- Treinamento supervisionado de modelos complexos (por exemplo, com redes de muitas camadas) é dificilmente realizado (problemas de otimização)
- Modelos superficiais (Support vector machines, redes com uma camada, boosting, etc...) são candidatos a aprender alto nível de abstração necessário para inteligência artificial.
- Aprendizado não supervisionado pode fazer um "aprendizado-local", onde cada módulo tenta seu melhor para modela o que ele vê.
- Inferência (+aprendizado) é intratável nos modelos gráficos diretos com muitas variáveis escondidas.

Métodos de aprendizados atuais não estendem o aprendizado facilmente para os múltiplos níveis da representação

8.2 Propriedades Fundamentais

O projeto de uma SBN é altamente dependente da forma que os neurônios visíveis e escondidos são arranjados, portanto, diferentes arranjos resultam em diferentes configurações. Seja o vetor X, consistindo de variáveis randômicas de 2 valores X_1,X_2,....X_N, composto de N neurônios estocásticos. Os "pais" de X_j em X é:

$$pa(X_j) \subseteq \{X_1, X_2, ..., X_{j-1}\}$$

Que é um conjunto Menor que X, então

$$P(X_j = x_j \mid X_1 = x_1, ..., X_{j-1} = x_{j-1}) = P(X_j = x_j \mid pa(X_j))$$

Em particular a probabilidade que o j-ésimo neurônio estar ativado é definido pela função sigmoid

$$P(X_j = x_j \mid pa(X_j)) = \varphi(\frac{x_j}{T} \sum_{i<j} w_{ji}\, x_i) \text{ (mesma do BM)}$$

A probabilidade condicional P(Xj = xj |pa(Xj)) depende somente de pa(Xj) através de uma soma de pesos de entrada. Então, as seguintes propriedades são importantes:

1) $w_{ji} = 0$ para todo X_i não pertence ao $pa(X_j)$ → segue da definição dos "pais"
2) $w_{ji} = 0$ para todo $i \geq j$ → segue da definição do GAD

Diferente de BM, a SBN precisa somente uma fase de aprendizado. A razão para esta simplificação é que a normalização da distribuição de probabilidade sobre os vetores de estado é completado pelo nível local de cada neurônio via a função sigmoid, ao invés de globalmente via a função de partição Z que envolve todas as possíveis configurações de estado.

Vantagens em relação ao BM

1) SBN é capaz de aprender de modelos de distribuição não trivial
2) Ela pode aprender mais rápido que a BM
3) Eliminação da Fase Negativa da BM do procedimento de aprendizado.

8.3 Procedimento de Aprendizado

- Inicializar a rede ajustando os valores dos pesos wji para valores randômicos distribuídos uniformemente.
- Dado um conjunto de treinamento T, segurar os neurônios visíveis da rede para xα
- Para cada xα realizar uma simulação de amostragem Gibbs da rede para uma temperatura de operação T e observar o estado de x (de toda a rede). A simulação deve ser longa o suficiente para garantir que os valores de x para os diferentes casos do

conjunto de treinamento deverão vir da distribuição condicional do correspondente vetor X.

- Calcular a média ρji,
- Incrementar cada peso sináptico wji através da equação:

$$\Delta w_{ji} = \eta \rho_{ji}$$

- Este ajuste nos pesos wji deverá mover os pesos da rede para um Máximo local da função L(w).

9. Teoria Mean - Field

Na Maquina de Boltzmann o aprendizado é feito usando duas fases. Primeiro a rede é rodada com tanto as entradas quanto as saídas fixadas. As probabilidades são medidas após o sistema ter realizado o mínimo global de energia. Este procedimento é então repetido com apenas as unidades de entrada fixada. Os pesos são então ajustados de acordo com o gradiente-descendente. Para realizar um mínimo de energia global em cada fase deste processo, é necessário usar o método de Simulated-Annealing que consome muito tempo. Também o fato de que em maquinas estocásticas é desejável saber o estado de todos os neurônios da rede a todo tempo, mas que em uma rede com um grande numero de neurônio o estado neural contem muito mais informação do que é necessário na pratica. A teoria de Mean-Field é uma forma de derivar a aproximação determinística para acelerar o processo de aprendizado em maquinas estocásticas.

Existem 2 métodos podem ser seguidos:

1) Correlações são substituídas por suas médias; (aplicado a BM)
2) Um modelo intratável é substituído por um modelo tratável através de um principio variacional – aplicável a SBN e Helmholtz

Método 1) Em [Peterson & Anderson 1987] demonstrou que o processo de Simulate-Annealing estocástico nas Maquinas de Boltzmann pode ser substituído por um conjunto de equações determinísticas chamada de aproximação da teoria de Mean-Field. Em uma rede com um grande numero de neurônios os estados neurais contem mais informação do que é necessário na pratica. Na verdade, a informação dos valores médios do estado do neurônio já

é suficiente. Este algoritmo de aprendizado da teoria Mean-Field (MFT) tipicamente provém um substancial aumento de velocidade sobre a Maquina de Boltzmann.

Em um neurônio estocástico o mecanismo de gatilho é descrito por uma regra estocástica.

$$X_j = \begin{cases} +1 & com\ probabilidade\ P(v_j) \\ -1 & com\ probabilidade\ 1 - P(v_j) \end{cases}$$

onde,

$$P(v_j) = \frac{1}{1 + \exp(\frac{-v_j}{T})}$$

Podemos expressar a media $<X_j>$ para um especifico valor do campo local induzido v_j da forma:

$<X_j> = P(x_j = +1)\ (+1) + P(x_j = -1)\ (-1)$

$<X_j> = (+1)\ P(v_j) + (-1)\ [1 - P(v_j)]$

$<X_j> = 2^*\ P(v_j) - 1$

$<X_j> = \text{Tanh}(v_j / 2T)$ onde j=1,2,...n (1)

Onde a media “Mean” de X_j ($<X_j>$) é também considerada a media “Térmica”, desde que o ruído sináptico é geralmente modelado em termos de flutuações térmicas.

A figura abaixo mostra o gráfico do <Xj> pelo Vj, onde para valores T→0 tem-se

$$< x_j > \rightarrow \text{sgn}(v_j) \quad para \quad T \rightarrow 0$$

Isto para um neurônio estocástico. Para o caso de uma maquina estocástica composta de muitos neurônios, a dificuldade deste método ocorre devido aos seguintes fatores:

1) a probabilidade $P(v_j)$ que o neurônio j esta “on” é uma função não-linear do campo local induzido v_j.
2) V_j é uma variável randômica, sendo influenciada pela ação estocástica de outros neurônios conectados a entrada do neurônio j.

Para resolver este caso é usada uma “aproximação de mean-field” que muitas vezes traz bons resultados. A idéia nesta aproximação é substituir a atual flutuação do campo local induzido v_j por cada neurônio j pela sua media $<v_j>$ da seguinte forma:

$$v_j \overset{aprox}{=} < v_j >= \left\langle \sum_i w_{ji} x_i \right\rangle = \sum_i w_{ji} < x_i > \quad (2)$$

Então substituindo (2) em (1) tem-se:

$$< x_j >= \tanh(\frac{1}{2T} v_j) \overset{aprox}{=} \tanh(\frac{1}{2T} < v_j >) = \tanh(\frac{1}{2T} \sum_i w_{ji} < x_i >) \quad \text{para j=1,2,..,n (3)}$$

O sistema agora é determinístico e é aproximado pelo n medias equações representadas pela equação (3). É importante notar que a equação (3) é significante apenas quando a rede esta no equilíbrio térmico, que significa que toda a quantidade $<X_j>$ converge (ou seja,, torna-se invariante no tempo).

10. Maquina de Boltzmann Determinística

Peterson e Anderson [Peterson & Anderson 1987] propuseram um método para acerar o processo de aprendizado da Maquina de Boltzmann baseado na Mean-Field Theory. Este método consiste em substituir as correlações na regra de aprendizado da máquina de Boltzmann por uma aproximação da Mean-Field como é mostrado abaixo:

$$\Delta w_{ji} = \eta(\rho_{ji}^{+} - \rho_{ji}^{-}) \quad onde \quad \rho_{ji}^{+} =< x_j x_i >^{+} , \rho_{ji}^{-} =< x_j x_i >^{-} \text{ Regra BM}$$

$$< x_j x_i >= < x_j >< x_i > \quad (i, j) = 1,2,...., K$$

Então, a regra de aprendizado da Maquina de Boltzmann Determinística fica da forma:

$$\Delta w_{ji} = \eta(U_j^{+} U_i^{+} - U_j^{-} U_i^{-})$$

onde U_j^{+} *e* U_j^{-} *são as médias das saidas do neuronio visivel j* nas condições de “clamped” e “free-running” para um determinado padrão.

Este método se aplica apenas em casos supervisionados, que é, quando alguns dos neurônios visíveis são assinalados as regras dos neurônios de saída. No aprendizado não supervisionado não funciona para todo mundo o regime de “Media” porque o estado médio tem uma representação muito pobre na fase do aprendizado “free-running”.

No aprendizado supervisionado, o uso da Maquina de Boltzmann determinística esta restrita a redes com uma camada escondida. Apesar de na teoria não existir nenhuma razão para carregar múltiplas camadas escondidas, na pratica mostrou ter problemas quando coloca-se mais de 1 camada escondida.

Devido a sua formula simples, a maquina de Boltzmann determinística tem uma gama de aplicações em larga escala em hardware.

Referencias Bibliográficas

[Bilbro & Snyder 1991] Bilbro, G.L., Snyder, W.E., Mean-field approximation minimizes relative entropy, *J. Opt.Soc.Am*, Vol. 8, No.2, 1991.

[Dayan *et al.* 1995] Dayan, P., Hinton, G.E, Neal, R.M., Zemel, R.S. The Helmholtz machine, *Neural Computation*, Vol 7, pp. 889-904, 1995.

[Hasting 1970] Hasting, W. K., Monte Carlo sampling methods using Markov chains and their applications, Biometrika, Vol. 87, pp. 177-221, 1970.

[Haykin 1999] Haykin, S., *Neural Networks A comprehensive Foundation* – Second Edition. Prentice Hall, 1999.

[Hebb 1949] Hebb, D.O., *The organization of behavior: A Neuropsychological Theory*, New York, Wiley, 1949.

[Hinton & Sejnowski, 1983] Hinton, G.E., Sejnowski, T.J., Optimal perceptual inference, In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 448-453, Washington, D.C., 1983.

[Hinton *et al.* 1995] Hinton, G.E, Dayan, P., Frey, B.J., Neal, R.M. The 'wake-sleep' algorithm for unsupervised neural networks, *Science*, Vol. 268, pp. 1158-1161, 1995.

[Kirkpatrick *et al.* 1983] Kirkpatrick, S., Gellat, C.D., Vecchi, M.P., Optimization by simulated annealing, *Science*, Vol. 220, pp. 671-680, 1983.

[Kullback & Leibler 1951] Kullback, S., Leibler, R. A., On information and sufficiency, *Annals of Mathematical Statistics*, Vol. 22, pp. 79-86, 1951

[Metropolis *et al.* 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H.; Teller, E, Equations of state calculations by fast computing machines, *J. Chem. Phys* Vol. 21, 1087-1092. 1953.

[Neal 1992] Neal, R.M., 1992, "Connectionist learning of belief network", Artificial Intelligence, vol.56, pp.71-113, 1992.

[Peterson & Anderson 1987] Peterson, C., Hartman, E., 1989 – "Explorations of the Mean Field Theory Learning Algorithm", Neural Networks, vol 2, pp. 475-494, 1987.

[Roweis 1995] Roweis, S., Boltzmann Machines. Lecture Notes, 1995