



1 Introdução

O objetivo deste trabalho é compartilhar os conhecimentos obtidos com a disciplina IME04737 Redes Neurais. A disciplina teve como base o livro Introduction to the Theory of Neural Computaton[1]. Outra fonte de embasamento foi o Redes Neurais Artificiais: Rede Hopfield e Redes Estocásticas[2]

2 Conhecimentos Obtidos

2.1 McCulloch e Pitts e Relações/limites entre as redes biológicas e as redes neuronais

Primeiramente, é necessário deixar claro que apesar do nome redes neuronais remeterem diretamente ao cérebro, não estudaremos conceitos biológicos de redes de neurônios e sim aspectos teóricos dos elementos de redes que tem seu comportamento inspirado em redes de neurônios.

Como inspiração para as redes abordadas, temos as seguintes características que fazem as redes de neurônios contidas no cérebro:

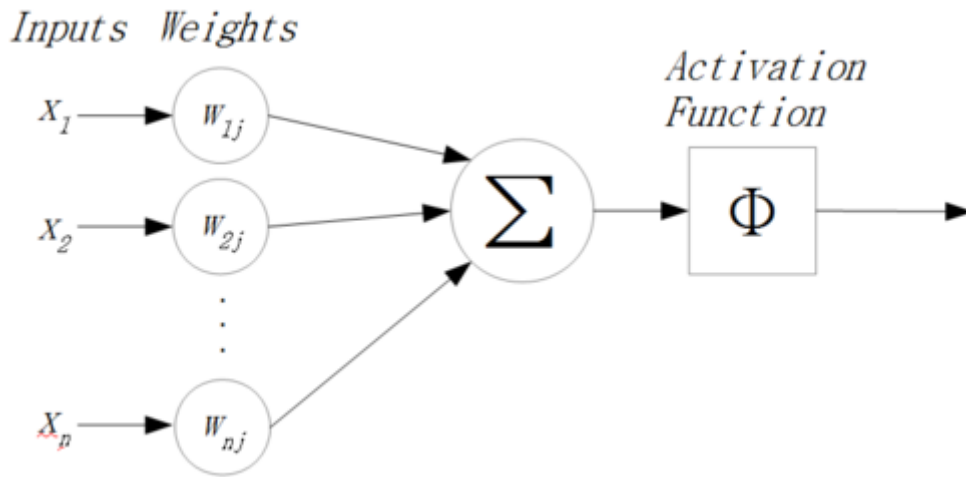
- é tolerante a falhas e a substituição de unidades da rede sem afetar a performance de processamento
- devido a capacidade de aprendizado, consegue se adaptar de forma rápida a mudanças de variáveis ambientais.
- grande capacidade de processamento paralelo
- é hábil para processar informações incompletas, com ruídos, com probabilidades de correteude variáveis.
- é autocontida em espaço pequeno e utilizando pouca energia.

Todas as características acima são altamente desejáveis por qualquer aplicação computacional desde o início do estudo da área e até hoje.

A unidade básica de uma rede neuronal biológica é o neurônio. Neurônios são unidades altamente conectadas entre si, que transmitem sinais (nome dado para designar o envio de uma série de substâncias que causam o aumento ou a diminuição do potencial elétrico) de um neurônio para o outro.

Para que uma célula faça alguma ação, como enviar um sinal, é necessário que ela receba uma quantidade de estímulo (que são enviados pelos neurônios vizinhos). Essa quantidade de estímulo é conhecida como limiar de ativação. Neurônios não fazem nada até receber sinal suficiente da sua vizinhança que seja maior ou igual ao seu limiar de ativação.

McCulloch e Pitts [3] definiram um modelo que define uma representação para o neurônio. No modelo proposto, cada elemento que conecta um neurônio com outro possui um peso, também conhecido como peso sináptico. Peso sináptico é o nome dado para a representação da força de conexão entre os neurônios ligados. No modelo estudado, ele representa um parametro importante para o comportamento do neurônio observado.



O diagrama acima tem os seguintes elementos a serem observados:

- cada entrada significa um sinal enviado por um neurônio conectado ao neurônio estudado.
- os pesos sinápticos, representados por ω
- o somatório, representando o estímulo que o neurônio receberá enviado pelos neurônios com os quais o mesmo está conectado.
- a função de ativação, que determina se o somatório foi superior ou não ao potencial de ativação. O potencial de ativação, ou limite de ativação, é representado por μ

A equação:

$$n_i(t+1) = \Theta\left(\sum_j \omega_{ij} n_j(t) - \mu\right)$$

Representa se um neurônio em um determinado instante de tempo $t+1$ disparará sinal para a sua vizinhança ou se estará inibido, ou seja, nenhum sinal será emitido para os neurônios ao seu redor.

Importante ter em mente que:

- n_i é o neurônio observado
- t é o instante de tempo observado
- ω_{ij} é o peso sináptico que ligam o neurônio destino i com o neurônio origem j , ou seja, a força do sinal enviado.
- n_j o sinal emitido pelo neurônio j
- μ é o limiar de ativação do neurônio i

Destacando a segunda parte da igualdade da equação acima:

$$\Theta\left(\sum_j \omega_{ij} n_j(t) - \mu\right)$$

ela é uma uma função $\Theta(x)$, ou seja, uma função degrau. Uma função degrau é definida por:

$$\Theta(x) = 1, \forall x \geq 0$$

$$\Theta(x) = 0, \forall x < 0$$

Ou seja, os valores 0, 1 representam se o neurônio estará em estado de disparo ou não.

Analisando a equação, podemos afirmar que para que o neurônio dispare num determinado instante t , o somatório do produto dos pesos sinápticos e dos sinais emitidos pelo neurônio j deve ser maior que o limiar de ativação μ do neurônio.

O processo de um neurônio receber impulsos da sua vizinhança e disparar ou não impulsos para os seus vizinhos é chamado de atualização.

Neurônios biológicos não possuem homogeneidade em relação ao tempo que levam no disparo de sinais para a sua vizinhança. Na equação inicial, a generalização $t \rightarrow t + 1$ é perceptivamente síncrona - ou seja, todos os neurônios se atualizam em $t + 1$ unidades de tempo. Para se assemelhar mais com o neurônio biológico, existe também a atualização assíncrona. Os tipos de atualização serão abordados mais adiante neste trabalho.

Além da atualização, existem várias propriedades dos neurônios reais que não foram incluídos na equação de McCulloch-Pitts. Conseguimos tornar a equação mais próxima do neurônio biológico com algumas modificações:

$$n_i = g\left(\sum_j \omega_{ij}n_j - \mu\right)$$

Sendo:

- n_i o estado do neurônio (ou unidade) i .
- $g(x)$ substituindo $\Theta(x)$. $g(x)$ é chamada de função de ativação. A grande diferença da função de ativação é que ela não depende do tempo, ela depende do evento de atualização de um neurônio na vizinhança, não importa quando o mesmo ocorrer. Essa atualização é conhecida como assíncrona.

Em termos de computação paralela, podemos pensar no cérebro como um sistema paralelo com 10^{11} núcleos de processamento.

Utilizando o modelo descrito por McCulloch-Pitts e a equação modificada, com cada processador computando a soma das entradas de dados enviados por outros processadores e transformando-a em um único valor numérico.

Da mesma forma que este neurônio recebeu entradas de outros processadores, ele enviará o valor que ele calculou da soma dos outros processado para os outros processadores.

O modelo que estamos estudando permite que os pesos sinápticos sejam modificados e adaptados para diferentes propósitos. Ainda assim, da forma que a computação baseada no modelo de von Neumann, mesmo com a velocidade do processamento baseado em silício sendo mais veloz, a capacidade do cérebro biológico se adaptar a falhas demonstra um caminho que as redes neuronais baseadas no modelo de McCulloch-Pitts e suas variancias ainda precisam percorrer: serem cada vez mais tolerantes a falhas e a ruídos.

Além da questão da tolerância a falhas, as redes neuronais tem outros desafios a enfrentar para simular o paralelismo do cérebro biológico:

- Escolha de arquitetura - quantas conexões cada núcleo terá e como os elementos da rede neuronal serão organizados? Quantos núcleos a rede terá? Como será o modelo de atualização: síncrono ou assíncrono? Será uma rede determinística ou estocástica?
- Escolha de como será o aprendizado da rede.
- Qual a velocidade que a rede terá? Como ela lidará com informação incorreta e falhas?
- É possível montar a rede neuronal pretendida em algum tipo de hardware ou um conjunto de hardware?

Apesar da parte de hardware ainda ter um caminho para avançar, o lado positivo das redes neuronais é que o poder de generalização para novas situações. Elas são capazes de gerar um novo conhecimento através dos exemplos que a mesma foi treinada.

2.2 Como a memória cria associação?

Segundo o texto Associative Learning and the Hippocampus [4], a memória associativa é definida pela habilidade de aprender e lembrar as relações entre elementos não relacionados.

Outra definição é que memórias associativas (AMs, associative memories) são modelos matemáticos inspirados na habilidade do cérebro humano de armazenar e recordar informações. Tais como as redes neurais biológicas, uma AM deve ser capaz de recuperar uma informação memorizada mesmo diante da apresentação de um item incompleto ou distorcido. Aplicações de AMs incluem classificação, previsão e reconhecimento de padrões [5]

Em redes neuronais, memória associativa é um problema definido por: "Guarde um conjunto p de padrões ξ_i^μ de forma que quando um novo padrão ζ_i for apresentado para a rede, a rede consiga responder com o padrão armazenado mais próximo ao padrão ζ_i apresentado".

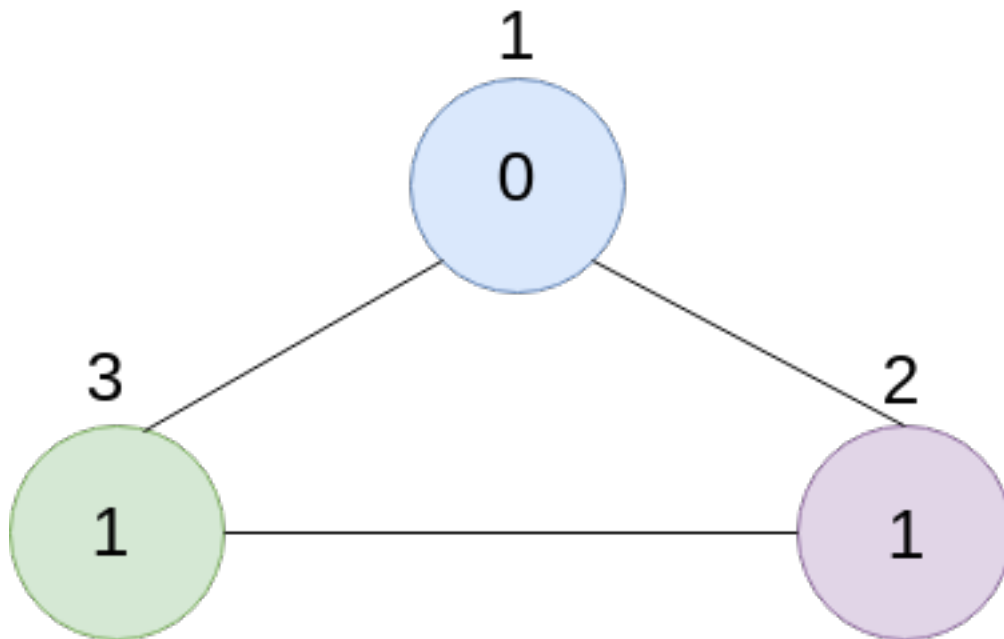
O "padrão armazenado mais próximo ao padrão apresentado", ou seja, dentro do conjunto de padrões p , qual é o mais próximo de ζ_i é obtido calculando a distancia de Hamming:

$$\sum [\xi_i^\mu (1 - \zeta_i) + (1 - \xi_i^\mu) \zeta_i]$$

A distancia de Hamming é calculada entre o padrão ζ_i e cada padrão pertencente ao conjunto p . Quem tiver a menor distância é o padrão mais semelhante ao padrão ζ .

Matematicamente, ξ é um vetor contendo μ padrões e i posições.

Por exemplo, dado:



$$\vec{\xi}^1 = (0, 1, 1)$$

0 é o padrão armazenado na unidade 1, ou seja ξ_1^1

1 é o padrão armazenado na unidade 2, ou seja ξ_2^1

1 é o padrão armazenado na unidade 3, ou seja ξ_3^1

$\mu = 1$, pois o array ξ só tem 1 dimensão de padrões armazenados.

Se apresentarmos para esta rede $\zeta_1 = (0, 0, 1)$, podemos calcular a distancia de Hamming:

$$\begin{aligned} \sum \xi_i^\mu (1 - \zeta_i) + (1 - \xi_i^\mu) \zeta_i &= ((\xi_1^1 (1 - \zeta_1) + (1 - \xi_1^1) * \zeta_1) \\ &\quad + ((\xi_2^1 (1 - \zeta_2) + (1 - \xi_2^1) * \zeta_2) \\ &\quad + ((\xi_3^1 (1 - \zeta_3) + (1 - \xi_3^1) * \zeta_3)) \end{aligned}$$

Substituindo os valores, temos

$$\begin{aligned} \sum \xi_i^\mu (1 - \zeta_i) + (1 - \xi_i^\mu) \zeta_i &= ((0(1 - 0) + (1 - 0) * 0) \\ &+ (1(1 - 0) + (1 - 1) * 0) \\ &+ (1(1 - 1) + (1 - 1) * 1)) = 1 \end{aligned}$$

Ou seja, existe apenas uma unidade com padrão armazenado diferente do padrão apresentado.

Se a distancia de Hamming for calculada entre dois números binários, a distância é o número de bits que são diferentes entre os dois números.

Unificando o modelo de McCulloch-Pitts e a distancia de Hamming, se a configuração do modelo começar com $n_i = \zeta_i$, gostaríamos de saber se algum dos pesos sinápticos w_{ij} fará com que a rede chegue ao estado com $n_i = \xi_i^{\mu_0}$ e qual μ_0 terá a menor distancia de Hamming. Além disso, desejamos que a memória tenha conteúdo endereçável e releve erros pequenos dos padrões apresentados. Com a equação acima conseguimos encontrar a primeira parte do objetivo desejado.

Uma das aplicações mais comuns de memória associativa é o reconhecimento e a reconstrução de imagens. As imagens reconhecidas estão parcialmente completas e com a associação entre o padrão que temos armazenado, ou seja, a imagem completa e os fragmentos apresentados, conseguimos reconstruir a imagem apresentada para que ela fique igual ao padrão armazenado. Este processo ocorre diversas vezes tanto em redes neuronais biológicas quanto artificiais.



Figura 1: exemplo de como a memória reconhece e/ou completa informações se parte estiver armazenada na rede

Na imagem acima, fica exemplificado o processo de reconhecimento e reconstrução de imagens. Apesar das imagens estarem incompletas ou corrompidas, como os padrões já estão armazenados, é possível reconstruir as imagens. A coluna da esquerda demonstra o que foi apresentado para a rede neural. Na central, o processo intermediário. Na da direita, a imagem reconstruída, conforme os padrões previamente armazenados na rede.

Reconhecimento e reconstrução não somente serve para imagens, mas também para reconhecimento parcial de texto, por exemplo.

Memória de conteúdo endereçável, citado acima, é também conhecida como associação dinâmica. Tentaremos demonstrar a construção da mesma no decorrer do trabalho - e obter a segunda parte do objetivo desejado descrito acima.

Uma rede neural tem estados que dependem do tempo e dos sinais enviados pelas outras unidades da rede. Analizando todas as combinações de sinais emitidos de todas as unidades e as suas respostas, obtemos um conjunto chamado de espaço de configuração. Esses estados são úteis para armazenarmos os padrões pertencentes a nossa rede. Dentro do espaço de configuração, os padrões $\vec{\xi}^i$ recebem outro nome: atratores.

Como o sistema é dinâmico, é como se os pontos (os sinais) fossem "carregados" pelo sistema para os atratores. Se plotarmos um gráfico em que cada ponto representa o estado do neurônio durante o tempo, percebemos que os atratores representam pontos de mínimos locais. O nome atrator chega a ser literal, devido ao fato dos estados "caírem" na região do atrator. Essa região é chamada de bacia de atração (também pode ser chamado de vale de atração, mas bacia de atratores é mais comum para basins of attraction). Uma rede tem várias bacias de atratores.

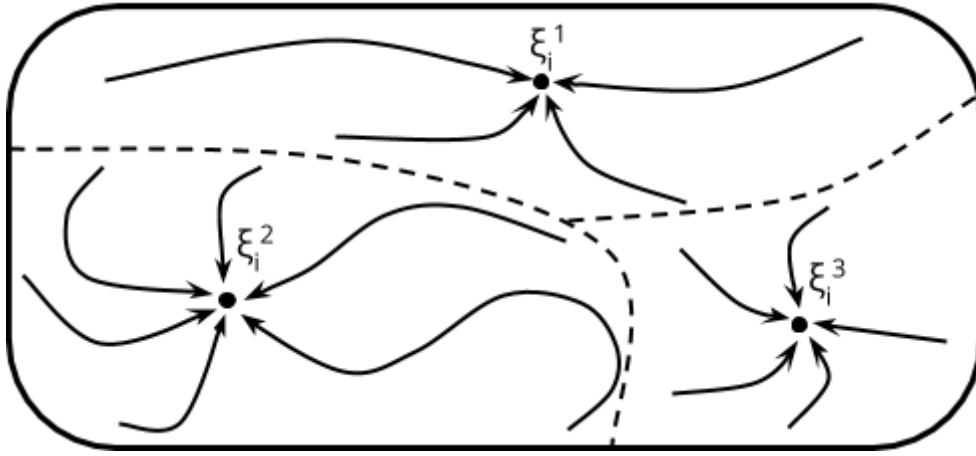


Figura 2: Espaço de configuração de atratores

2.3 Modelo de Hopfield

John Hopfield foi o pesquisador que criou um modelo que une os elementos explicitados acima: [rever definição modelo hopfield] modelo com memórias com conteúdos associativos.

O Modelo de Hopfield tem como características importantes:

- os estados dos nós representam a unidade disparando ou não.
- a garantia de convergência para os mínimos locais/bacias de atração, ou seja, garante que converja para algum atrator com padrão armazenado mais próximo do que procuramos.
- apesar da garantir descrita acima, existem casos de convergência ocorrer para um falso atrator, denotado como $-\vec{\xi}^i$

Para fins de clareza matemática, nós trabalharemos com novos valores para representar o estado de disparo ou não disparo: +1 para disparo e -1 para não disparo (no lugar original do 0). Outra substituição que é feita é S_i no lugar de n_i , visto que conseguimos relacionar S_i e n_i através da equação:

$$S_i = 2n_i - 1$$

E com isso, as equações de McCulloch-Pitts que determinavam se um neurônio estava disparando ou não (também conhecida como a dinâmica da rede) podem ser reescritas da seguinte forma:

$$S_i = \text{sgn}\left(\sum_j \omega_{ij} S_j - \theta_i\right)$$

E o resultado da função sinal será lido como: $\text{sgn}(x) = 1$, se $\sum_j \omega_{ij} S_j - \theta_i \geq 0$ ou $\text{sgn}(x) = -1$, se $\sum_j \omega_{ij} S_j - \theta_i < 0$

E finalizando as correlações: θ_i é equivalente a μ_i . A equação que relaciona θ e μ é:

$$\theta_i = 2\mu_i - \sum_j \omega_{ij}$$

Considerando, por fins adaptação aos padrões que temos por objetivo observar, $\theta_i = 0$, obtemos a equação:

$$S_i = \text{sgn}\left(\sum_j \omega_{ij} S_j\right)$$

- equação muito importante, também conhecida como equação do movimento neural/dinâmica de Hopfield.

Observe que a inclusão da função $\text{sgn}(x)$ torna o resultado da equação acima discreto.

A atualização dos estados das unidades é essencial numa rede neural. Ela pode ser ocorrer de 2 formas: síncrona e assíncrona.

A atualização síncrona atualiza todas as unidades num determinado momento de tempo. A sincronização é de todas, sempre contando com o estado (disparo ou não disparo das unidades no momento anterior ao momento calculado).

Já na atualização assíncrona, cada unidade é atualizada em cada instante de tempo, levando em consideração a atualização realizada imediatamente anterior.

A grande diferença entre as atualizações é que a síncrona sempre considera o estado anterior ($t - 1$) das unidades para calcular a atualização no momento de tempo observado, enquanto a assíncrona leva em consideração o estado atual de todas as unidades no cálculo da atualização de uma unidade.

A atualização assíncrona é mais próxima ao cérebro biológico.

Para a atualização síncrona ocorrer, é necessário um controle central de tempo para controlar os momentos de atualizações. Como as unidades são atualizadas de uma única vez, não pode começar uma atualização antes de terminar a anterior. Para tal é necessário controlar o tempo de início e fim das atualizações.

Já na atualização assíncrona, podemos escolher entre as alternativas:

- seleccionar uma unidade randomicamente a cada momento de tempo
- deixar cada unidade se atualizar sozinha (aplicando a equação $(S_i = \text{sgn}(\sum_j \omega_{ij} S_j))$ de tempos em tempos. (confirmar se está é a expressão mais fácil de entender).

A primeira opção é mais indicada para simulações. A segunda para hardwares autônomos. O que devemos ter cuidado é que existe uma pequena possibilidade de 2 unidades acabarem atualizando ao mesmo tempo, para tal, deve-se tomar as precauções/decisões para evitar esta situação.

Durante o planejamento de qual será a forma de atualização das unidades de uma rede neural, é necessário definir quantas atualizações serão realizadas para que sejam encontrado o padrão armazenado buscado. Além disso, devemos ter em mente como vamos distribuir os nós e a informação armazenada de forma randômica - aleatoriedade é relevante para a capacidade de armazenamento da rede.

Para efeitos de estudo, vamos dividir as redes pela quantidade de padrões que desejamos armazenar/-memorizar, começando com 1 único padrão ξ_i :

Para este caso, a condição necessária é chegar a um estado estável, ou seja:

$$sgn(\sum_j \omega_{ij} \xi_j) = \xi_i \forall 0 < i < N$$

Já que a equação do movimento neural garante que um estado estável não produz mudanças nos estados das unidades da rede.

Para calcularmos os pesos semânticos, temos a equação: $\omega_{ij} = (1/N)\xi_i\xi_j$, sendo N o número de neurônios da rede.

A soma dos sinais que as unidades de rede enviam para uma unidade i é calculada: $h_i = \sum_j \omega_{ij} S_j$. Se a distancia de Hamming entre os bits fornecidos por $sgn(h_i)$ e ξ_i for maior que a metade dos bits do padrão apresentado, o padrão ξ_i se torna um atrator. Lembrando que no modelo de Hopfield, temos o ξ_i e o $-\xi_i$. Os 2 padrões tem a primeira metade com os mesmos bits e a segunda metade é o inverso dos bits dos seu oposto.

Existem algumas implementações da rede de hopfield. Na referência Redes Neurais Artificiais: Rede Hopfield e Redes Estocásticas[2] encontramos os seguintes algoritmos: Algoritmo Seq_Hopfiel, que descreve a atualização síncrona das unidades e o Algoritmo Dist_Hopfiel que descreve a atualização assíncrona das unidades.

Abaixo, algoritmo referente a atualização síncrona:

Algoritmo 1 Algoritmo Seq Hopfield

```

1: para  $i \leftarrow 1$  até  $n$  faça
2:    $v_i \leftarrow v_i^0$ 
3: fim para
4: faça
5:   para  $i \leftarrow 1$  até  $n$  faça
6:      $v_i \leftarrow v_i$ 
7:      $v_i = sgn(\sum \omega_{ij} v_j)$ 
8:   fim para
9: enquanto  $v_i v \forall n_i \in N$ 

```

O algoritmo de atualização assíncrona pode ser resumido a 3 funções (no texto citado, são referenciadas como procedimentos):

O algoritmo paralelo possui os procedimentos $Inicialize_i$, $Update_i(MSG_i)$, e $Notify_i$ para cada processador p_i . As mensagens trocadas entres os processadores são os estados dos neurônios que são utilizadas na atualização[2]

Algoritmo 2 Algoritmo Paralelo Hopfield - $Inicialize_i$

```

 $v_i = v_i^0$ 
Envie  $v_i$  para  $p_0$ 
Envie  $v_i \forall$  os vizinhos  $p_i$  de  $p_j$ , com arestas saindo de  $p_i$  para  $p_j$ 

```

Algoritmo 3 Algoritmo Paralelo Hopfield - $Update_i(MSG_i)$

```

Seja  $v_j \in MSG_i$  o estado de  $n_j \forall n_j \in V(n_i)$ 
 $v_i = step(\sum_j \omega_{ij} v_j)$ 

```

Algoritmo 4 Algoritmo Paralelo Hopfield - Notify_iEnvie v_i para p_0 Envie $v_i \forall n_j \mid n_j \text{ in } V(n_i)$

Se tomarmos como exemplo:

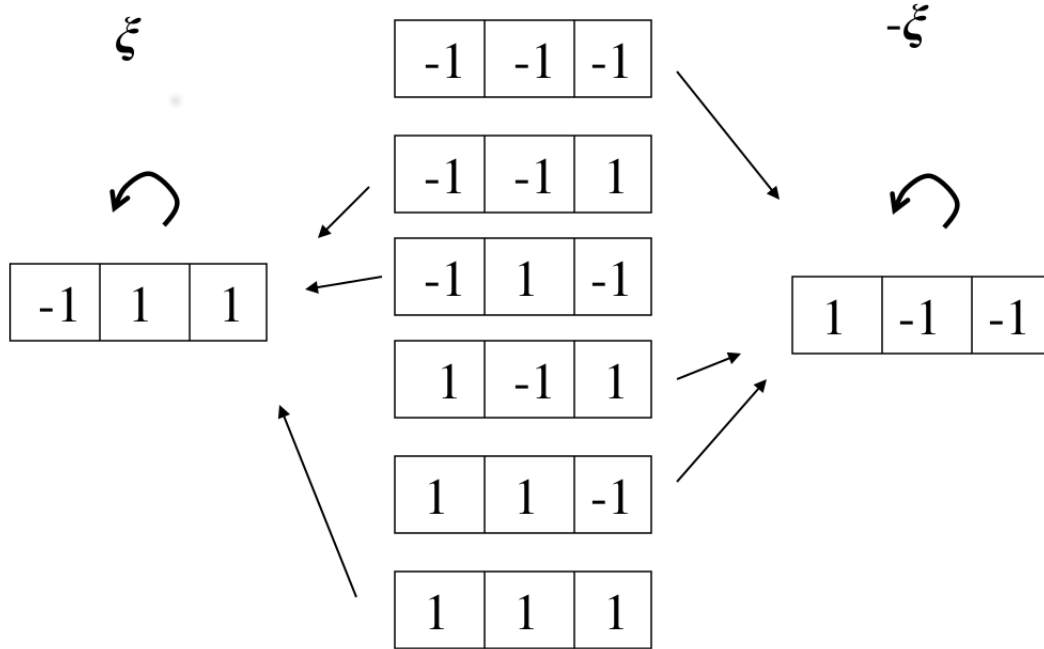


Figura 3: padrões apresentados convergem para padrões armazenados se eles existirem

A rede apresentada é de um único padrão (one pattern), temos o ξ que é o atrator da rede e a sua versão espúria, o $-\xi$.

$$\xi = (-1, 1, 1)$$

$$-\xi = (1, -1, -1)$$

Sabemos se uma rede converge ou não para um atrator através da equação da dinâmica de Hopfield:

$$S_i = \text{sgn}\left(\sum_j \omega_{ij} S_j\right)$$

O primeiro passo do cálculo para saber a convergência da rede dada acima é calcular os pesos sinápticos, ou seja ω_{ij} , tendo em mente que $\omega_{ij} = \omega_{ji}$

$$\omega_{ji} = 1/N \xi_i \xi_j$$

$$w_{12} = w_{21} = 1/3(-1)(1) = -1/3$$

$$w_{13} = w_{31} = 1/3(-1)(1) = -1/3$$

$$w_{23} = w_{32} = 1/3(1)(1) = 1/3$$

Calculando a atualização síncrona:

$$S = (-1, -1, -1)$$

$$T = 1$$

$$S_1 = \text{sgn}(\omega_{12} S_2(0) + \omega_{13} S_3(0))$$

$$S_1 = \text{sgn}(-1/3(-1) - 1/3(-1))$$

$$S_1 = \text{sgn}(-2/3) = -1$$

$$S_2 = \text{sgn}(\omega_{21}S_1(0) + \omega_{23}S_3(0))$$

$$S_2 = \text{sgn}(-1/3(-1) + 1/3(-1))$$

$$S_2 = \text{sgn}(0) = 1$$

$$S_3 = \text{sgn}(\omega_{31}S_1(0) + \omega_{32}S_2(0))$$

$$S_3 = \text{sgn}(-1/3(-1) + 1/3(-1))$$

$$S_3 = \text{sgn}(0) = 1$$

Para o instante $T=1$, para $-\xi$ Isso significa que $(-1, -1, -1)$ com 1 atualização síncrona converge para $-\xi$.

Calculando a atualização assíncrona para $(-1, -1, -1)$:

$$S_1 = \text{sgn}(\omega_{12}S_2 + \omega_{13}S_3)$$

$$S_1 = \text{sgn}(-1/3(-1) - 1/3(-1))$$

$$S_1 = \text{sgn}(2/3) = 1$$

Aqui está a grande diferença entre a atualização síncrona e assíncrona. O resultado que calculamos para S_1 já é atualizado e levado em consideração para calcular S_2

$$S = (1, -1, -1)$$

$$S_2 = \text{sgn}(\omega_{21}S_1 + \omega_{23}S_3)$$

$$S_2 = \text{sgn}(-1/3(1) + 1/3(-1))$$

$$S_2 = \text{sgn}(-2/3) = -1$$

Agora temos $S = (1, -1, -1)$, que também será utilizado para calcular S_3 .

$$S_3 = \text{sgn}(\omega_{31}S_1 + \omega_{32}S_2)$$

$$S_3 = \text{sgn}(-1/3(1) + 1/3(-1))$$

$$S_3 = \text{sgn}(-2/3) = -1$$

Com a atualização do último índice de S , obtemos $S=(1, -1, -1)$ que converge para $-\xi$. Com apenas um estágio da atualização, conseguimos encontrar a convergência.

No lugar de esperar todos atualizarem os índices atualizarem para considerar a propagação nos cálculos, na atualização assíncrona cada atualização já é propagada e utilizada no cálculo da atualização do próximo índice de S .

Após entendermos como é armazenado um padrão, vamos observar como um padrão numa rede com múltiplos padrões é armazenado/reconhecido:

Para calcularmos os pesos sinápticos, precisamos levar os padrões armazenados em consideração

$$\omega_{ij} = (1/N) \sum_{\mu=1}^p \xi_i^{\mu} \xi_j^{\mu}$$

Onde P é o número total de padrões armazenados e μ é o índice de cada padrão.

A equação acima é conhecida como Regra de Hebb. A Regra de Hebb é baseada numa hipótese de que a força dos pesos sinápticos mudam de de acordo com o estado, ou seja, a força sináptica do neurônio antes do neurônio atirar é diferente da força sináptica pós disparo. Esta hipótese é provada no caso das redes neurais:

- configuração dos padrões ξ_i^{μ} durante a fase de treinamento
- ajuste as forças sináptica ω_{ij} antes e depois do disparo, chegamos na equação da regra de Hebb.

A aplicação do modelo de Hopfield ocorre quando temos um modelo de memória associativa que é atualizada de forma assíncrona e o cálculo dos pesos sinápticos segundo a regra de Hebb para todos os pares i e j da rede.

Para calcularmos a condição de estabilidade, temos a seguinte equação:

$$\text{sgn}(h_i^\nu = \xi_i^\nu \forall i$$

sendo i o neurônio da rede e ν o padrão, calculamos h_i^ν :

$$h_i^\nu \equiv \sum_j \omega_{ij} \xi_j^\nu = (1/N) \sum_j \sum_\mu \xi_i^\mu \xi_j^\mu \xi_j^\nu$$

E quando $\mu = \nu$:

$$h_i^\nu = \xi_i^\nu + (1/N) \sum_j \sum_{\mu \neq \nu} \xi_i^\mu \xi_j^\mu \xi_j^\nu$$

Se a segunda parte da equação (o somatório) for zero, sabemos que o padrão ν é estável. O somatório acima recebe o nome de crosstalk term. Se o crosstalk term for bem pequeno para todo i e ν , podemos considerar que os padrões armazenados estão num estado estável.

Tão importante quanto saber como inserir e reconhecer padrões é saber quantas unidades a sua rede suporta. Para tal, vamos calcular a capacidade de armazenamento de uma rede, sendo que C_i^ν equivale a quantidade:

$$C_i^\nu \equiv -\xi(1/N) \sum_j \sum_{\mu \neq \nu} \xi_i^\mu \xi_j^\mu \xi_j^\nu$$

Se C_i^ν for negativo, o crosstalk term então é pequeno, logo o ξ_i^ν pode ser inserido sem problemas. Se C_i^ν for positivo e maior que 1, isso significa que o a unidade i do padrão ν está instável.

A resultado de C_i^ν depende dos padrões ξ_i^ν que tentaremos inserir na rede.

A fins de estudo, observaremos a tentativa de inclusão de padroes com probabilidade de serem 50% para $\xi_i^\nu = +1$ e 50% para $\xi_i^\nu = -1$. Desta forma, podemos afirmar:

$$P_{error} = \text{Prob}(C_i^\nu > 1)$$

Como o que fazemos é uma tentativa, existe uma taxa e erro envolvida na inclusão de cada padrão na rede. Considerando o erro de

$$P_{error} < 0.01$$

, podemos achar a capacidade máxima de unidades que mantem essa taxa de erro. Temos que ter em mente que quanto mais padrões adicionamos à rede, maior a chance de erro, portando obedecer os limites impostos pela taxa de erro requerida é importante para a segurança das conclusões obtidas.

Existe um desenvolvimento matemático relevante sobre o cálculo de erro de inclusão de padrões por número de unidades na rede, mas antes de discutirmos sobre isso, devemos observar a tabela abaixo. Ela relaciona a taxa de erro requerida pela proporção de padrões máximos inseridos por unidade da rede:

Tabela 1: Capacidade da Rede

P_{error}	p_{max}/N
0.001	0.105
0.0036	0.138
0.01	0.185
0.05	0.37
0.1	0.61

Observando a formula acima de forma matemática:

$$\begin{aligned}
P_{error} &= \frac{1}{2} \left[\frac{e^{\sqrt{N/2p}}}{\sqrt{\pi} \sqrt{N/2p}} \right] \\
\log(p_{error}) &\equiv \log_2 \left(\frac{e^{\sqrt{N/2p}}}{\sqrt{\pi} \sqrt{N/2p}} \right) = \frac{1}{2} \log(\pi) - \frac{1}{2} \log \left(\frac{N}{2p} \right) \\
\log \frac{1}{2} \left[\frac{e^{(\sqrt{N/2p})^2}}{\sqrt{\pi} \sqrt{N/2p}} \right] &= \log \frac{1}{2} \log \left[\frac{e^{(\sqrt{N/2p})^2}}{\sqrt{\pi} \sqrt{N/2p}} \right] \\
&= (\log 1 - \log 2) + (\log e^{-\sqrt{N/2p}}) - (\log \sqrt{\pi} \sqrt{N/2p}) \\
&= -\log 2 + (-N/2p) \log e - \log \sqrt{\pi} - \log \sqrt{N/2p} \\
&= -\log 2 - N/2p \log e - \log \pi^{1/2} - \log N/2p^{1/2} \\
&= -\log 2 - N/2p \log e - \frac{1}{2} \log \pi - \frac{1}{2} \log N/2p \\
\text{Considerando: } \log &= \ln \\
&= -\log 2 - N/P - 1/2 \log \pi - 1/2 \log(N/p)
\end{aligned}$$

Os estados dos neurônios mudam de tempos em tempos, assim como suas configurações. O conjunto das configurações dos estados S_i é chamado de função energia. Função energia é tem relevância devido ao fato de demonstrar como a energia sempre decresce perto dos atratores. Sabemos pela "equação do movimento" que o sistema converge para os atratores com o decorrer do tempo, ou seja, os padrões armazenados. O nome função de enegia é devido a analogia com o comportamento de sistemas magnéticos.

A função energia é calculada através da seguinte formula:

$$H = -1/1 \sum_{ij} \omega_{ij} S_i S_j$$

A condição de existencia da função energia é que as conexões entre as unidades da rede neural sejam simétricas, ou seja, o peso sináptico $\omega_{ij} = \omega_{ji}$.

Dada a condição de existencia, podemos reescrever o cálculo da função energia:

$$H = C - \sum_{(ij)} \omega_{ij} S_i S_j$$

Observar que (ij) são todos os pares distintos de conexões. O termo C representa os termos (ii).
[pulei iniciando com funcao energia]

No modelo de hopfield, sao produzidos 2 tipos de atratores: ξ_i^μ e $-\xi_i^\mu$, chamados de estados reversos, possuem a mesma energia do seu estado primário.

Além dos estados reversos, existem também os estados mistos ξ_i^{mix} . Estados mistos são diferentes de todos os estados, pois sao obtidos da combinação linear de um número randomico de padrões armazenados.

Calculamos os padrões mistos com a fórmula:

$$\xi_i^{mix} = \text{sgn}(\pm \xi_i^{\mu 1} \pm \xi_i^{\mu 2} \pm \xi_i^{\mu 3})$$

Separando os casos de sinais positivos dos negativos, e analisado a proximidade dos estados mistos dos reais atratores, percebemos que as distancias sao equidistantes (calculando a distancia de Hamming). Com isso, podemos afirmar:

$$h_i^{mix} = 1/N \sum_{i\mu} \xi_i^\mu \xi_j^\mu \xi_j^{mix} = 1/2 \xi_i^{\mu 1} + 1/2 \xi_i^{\mu 2} + 1/2 \xi_i^{\mu 3} + \text{cross - termos}$$

Para os termos negativos, a formula e analoga, apenas trocando o sinal positivo para o negativo.

2.4 Redes Estocásticas

Existe uma correspondência entre o modelo de Hopfield e os modelos de materiais magnéticos. É muito útil observar esta proximidade para entender melhor as redes neuronais com unidades estocásticas, ou seja, unidades que são influenciadas por outros fatores variáveis pertencentes a rede, além das sinapses da sua vizinhança. O fator que iremos adicionar à nossa rede é a temperatura.

Modelos magnéticos são uma representação de cristal - os átomos arrumados com a mesma distância entre si - de materiais magnéticos. A representação de cristal é uma característica importante destes modelos e a nomenclatura dos átomos magnéticos é spin. Um spin pode estar ligado com n outros spins.

Um modelo que nos ajuda neste processo de entendimento é o modelo de Ising, que permite o estado S_i apenas os valores ± 1 e conexões com apenas 2 outros spins. Quando $S_i = 1$, o spin aponta para cima. Quando $S_i = -1$, o spin aponta para baixo.

Um conjunto de spins magnéticos são influenciados pelo campo externo gerado pela presença dos próprios spins. É relevante saber a intensidade deste campo magnético, representado por h^{ext} - e a intensidade deste campo magnético pode ser considerado o elemento estocástico deste modelo.

Dada a influência do campo externo e a influência dos outros spins, podemos calcular o estado do spin com a seguinte equação:

$$h_i = \sum_j \omega_{ij} S_j + h^{ext}$$

Onde:

- h_i é o estado do spin i (ou campo magnético) obtido com a influência do campo interno e do campo externo.
- ω_{ij} é a intensidade da influência (conexão) de um spin i e j . Em interações magnéticas, $\omega_{ij} = \omega_{ji}$
- S_i é a direção da influência do spin i
- h^{ext} representa o campo externo.

Com a equação acima, podemos já correlacionar com o modelo de Hopfield. O atributo h_i corresponde ao atributo S_i descrito no modelo de Hopfield. A estabilidade do modelo de spins magnéticos ocorre quando estes se alinham com h_i , ou seja, o sinal de h_i é igual ao sinal de S_i .

O campo magnético h_i define as interações de S_i .

Quanto menor a temperatura, mais fácil é obter a estabilidade.

Quando é desejado observar a energia potencial de cada interação, podemos calcular com a equação abaixo:

$$H = 1/2 \sum_{ij} \omega_{ij} S_i S_j - h^{ext} \sum_i S_i$$

Finalizando a correlação entre as equações do estado magnético com o modelo de Hopfield, temos como função energia a energia emitida pelo magneto, quando $h^{ext} = 0$. Nestas condições, temos a equivalência com a equação de McCulloch-Pitts.

Nem sempre temos a temperatura suficientemente baixa para garantir a estabilidade do modelo magnético e aparecem as flutuações térmicas, que mudam o alinhamento dos spins em relação ao campo magnético. A influência do campo magnético e das flutuações térmicas no caso de temperaturas não desprezíveis sempre ocorrerão.

Para os casos de temperatura finita, nós podemos calcular o efeito das flutuações térmicas sobre o modelo de Ising utilizando a dinâmica de Glauber:

$$S_i = +1 \text{ com probabilidade } g(h_i)$$

$$S_i = -1 \text{ com probabilidade } 1 - g(h_i)$$

Onde $g(h_i)$ é a função:

$$g(h) = f_\beta(h) \equiv 1/(1 + \exp(-2\beta h))$$

E β - relacionado com a temperatura absoluta é encontrado por:

$$\beta = 1/k_B T$$

Finalizando com k_B representando a constante de Boltzmann, ou seja, $k_B = 1.38 \times 10^{-16} \text{ erg/K}$

Percebemos a relação $1 - f_{\beta}(h) = f_{\beta}(-h)$

Reescrevendo a equação $g(h)$, temos:

$$\text{Prob}(S_i = \pm 1) = f_{\beta}(\pm h_i) = 1/(1 + \exp(\pm 2\beta h_i))$$

Com isso, conseguimos limitar os resultados de $f_{\beta}(h)$ variando entre 0 e 1, com h variante entre ∞ e $-\infty$

Observaremos a questão do equilíbrio para apenas um único spin do campo magnético. Como só temos um spin, consideramos $h = h^{ext}$ e $S \pm 1$, nós chegamos a uma equação interessante, conhecida como magnetização média:

$$\begin{aligned} \langle S \rangle &\equiv \text{Prob}(+1) \cdot (+1) + \text{Prob}(-1) \cdot (-1) \\ &= 1/(1 + e^{-2\beta h}) - 1/(1 + e^{2\beta h}) = e^{\beta h} / (e^{\beta h} + e^{-\beta h}) + e^{-\beta h} / (e^{\beta h} + e^{-\beta h}) \\ &= \tanh \beta h \end{aligned}$$

Devemos notar que $\tanh(x)$ é do mesmo tipo de $f_{\beta}(h)$, diferenciando que $\tanh(x)$ varia de -1 a 1 .

Apesar do desenvolvimento acima ser apenas para 1 único spin, se temos um conjunto N de spins que estão sujeitos ao mesmo campo magnético externo e não influenciam uns aos outros, podemos considerar as equações acima para calcular seu equilíbrio. Essa correlação entre o sistema e um único spin é conhecida como paramagnetismo. Quanto mais quente estiver o sistema, menor a magnetização do mesmo.

Quando temos múltiplos spins interagindo, temos que levar em consideração a evolução do estado de cada spin e seu campo magnético gerado. Para contornar este problema, temos uma aproximação média conhecida como teoria do campo médio:

$$\langle h_i \rangle = \sum_j \omega_{ij} \langle S_j \rangle + h^{ext}$$

Lembrando que:

$$\langle S \rangle = \tanh(\beta \langle h \rangle) = \tanh(\beta \sum_j \omega_{ij} \langle S_j \rangle + \beta h^{ext})$$

Com a generalização acima, conseguimos remover variáveis estocásticas.

Mesmo com o campo médio, devemos ter em mente os infinitos intervalos de interação entre os spins uns com os outros. Ainda assim, a equação acima consegue mensurar bem o fenômeno observado do campo médio gerado.

Referências

- [1] Richard G. Palmer John Hertz, Anders Krogh. *Introduction to the theory of neural computation*. Addison-Wesley, 1991.
- [2] Juliana Mendes N. Silva. Redes neurais artificiais: Rede hopfield e redes estocásticas. http://www2.ic.uff.br/~jsilva/monografia_RNA.pdf.

- [3] Prof. Antônio Roque. Introdução à neurociência computacional (graduação)–prof. antônio roque história 1 –mcculloch-pitts e hebb. <http://sisne.org/Disiplinas/Grad/IntrodNeurocComput/McCulloch-Pitts%20e%20Hebb.pdf>.
- [4] PhD Wendy A. Suzuki. Associative learning and the hippocampus. <http://www.apa.org/science/about/psa/2005/02/suzuki.aspx>.
- [5] Marcos Eduardo Ribeiro do Valle Mesquita. Algumas generalizações das memórias associativas recorrentes por correlação. <https://www.ime.unicamp.br/~valle/Researches/FAPESP/Fapesp2013.pdf>.