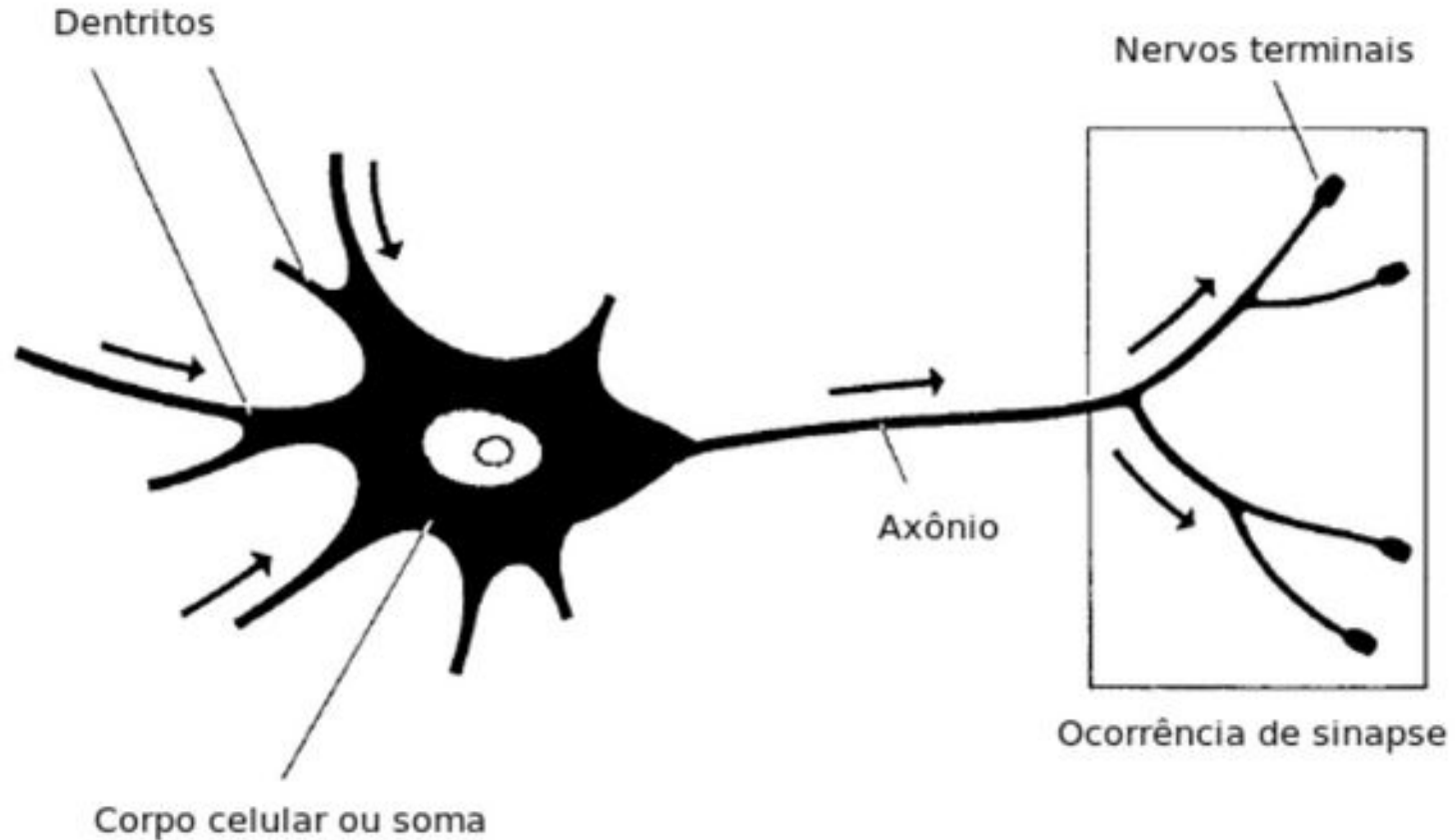


Inteligência Artificial

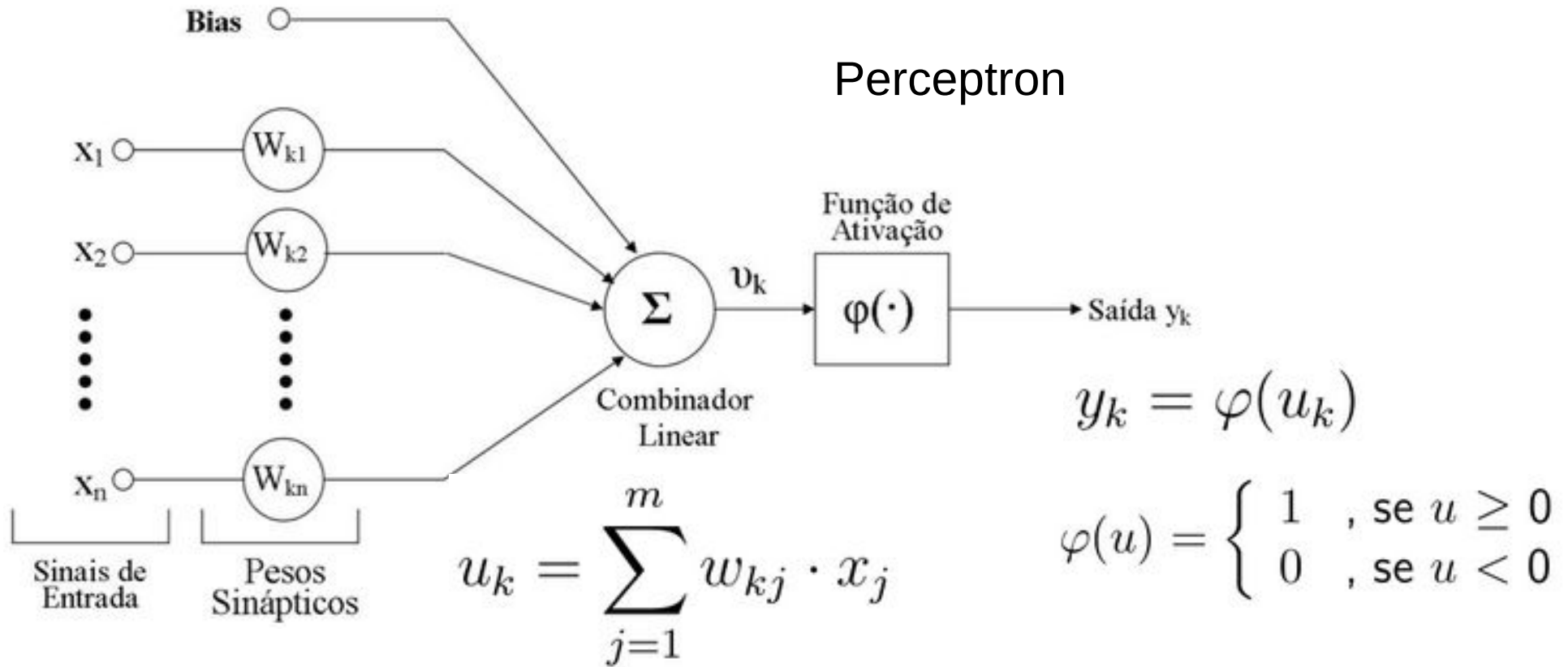
Redes Neurais Artificiais Parte II

Prof. Saulo Popov Zambiasi
saulopz@gmail.com

Inspiração Biológica



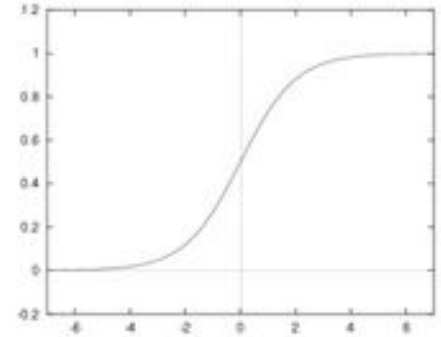
Modelo Matemático McCulloch e Pitts de 1943



Funções de Ativação Mais Comuns

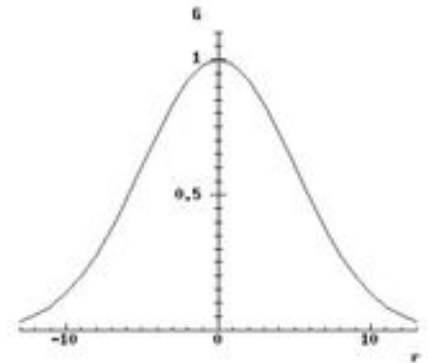
– Sigmoidal:

$$y = f\left(h = w_0 \cdot 1 + \sum_{i=1}^n w_i \cdot x_i; p\right) = \frac{1}{1 + e^{-h/p}}$$



– Radial (Gaussian):

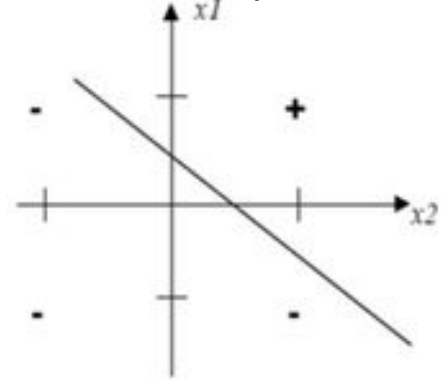
$$y = f\left(h = \sum_{i=1}^n (x_i \cdot w_i)^2; \sigma = w_0\right) = \frac{1}{2\pi\sigma} e^{-\frac{h^2}{2\sigma^2}}$$



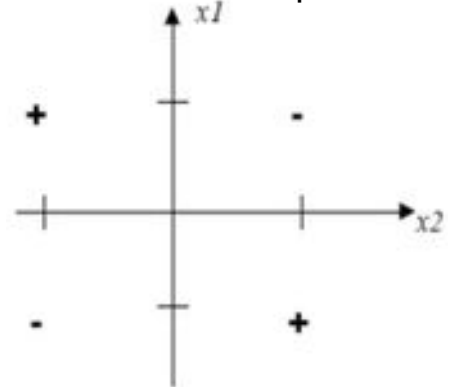
Limitações do Perceptron

- Só consegue resolver problemas linearmente separáveis
- Contudo, a maioria dos problemas existentes são não linearmente separáveis.

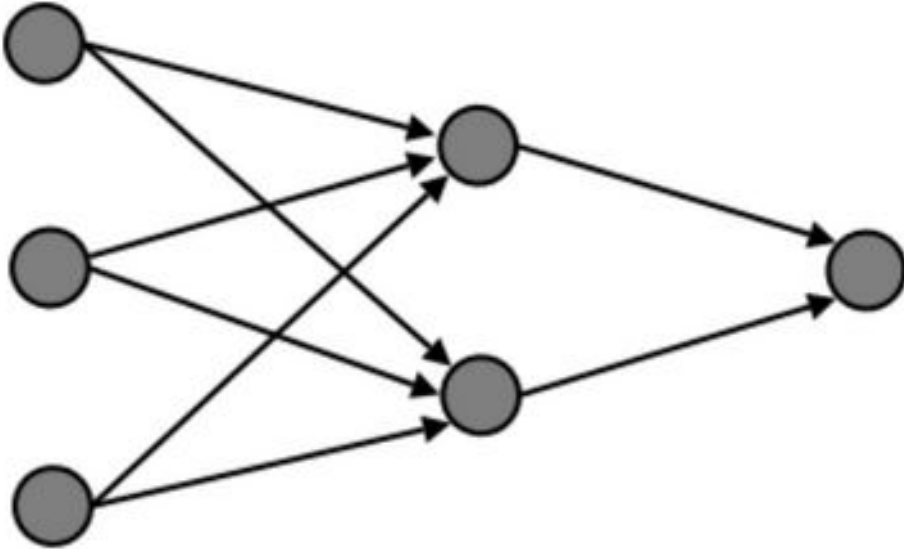
Porta lógica AND
Linearmente separável



Porta lógica XOR
Não-linearmente separável



Redes Neurais Artificiais

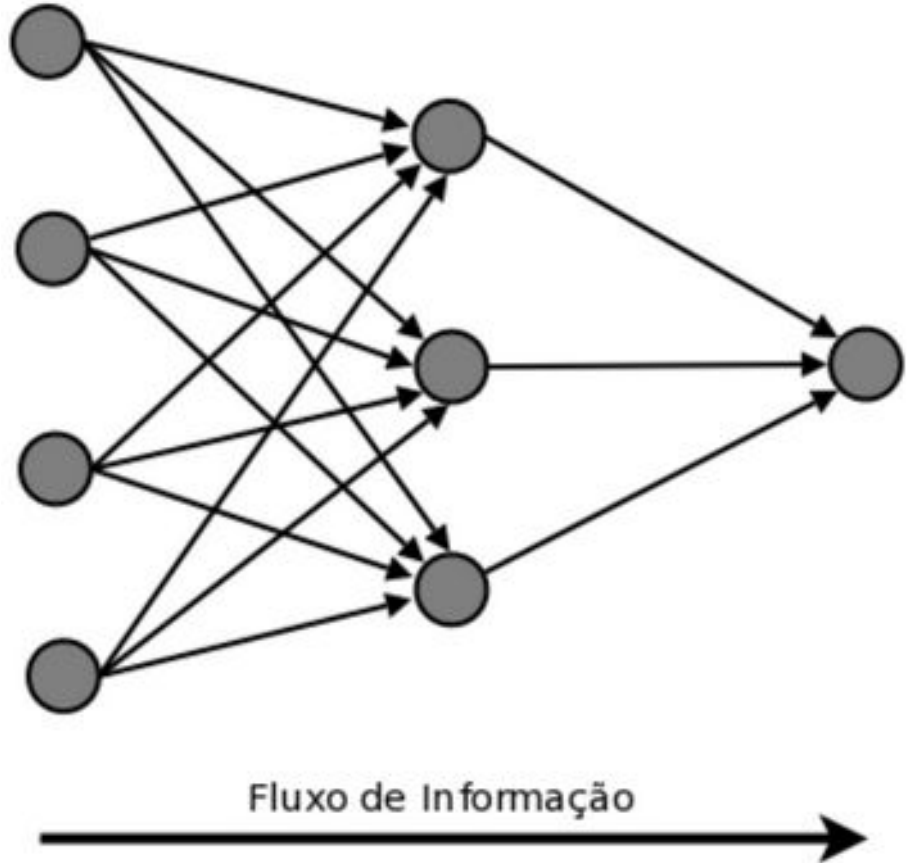


- Resolvem problemas não-linearmente separáveis
- Neurônios interligados, influenciando uns aos outros
- Formam um sistema maior que armazena conhecimento adquirido por meio de exemplos apresentados
- Podem realizar inferências sobre novos exemplos (situações desconhecidas)
- Geralmente um grafo orientado
 - Os vértices são os neurônios
 - As arestas as sinapses
 - A direção das arestas informa como os neurônios são alimentados

Topologia das Redes Neurais

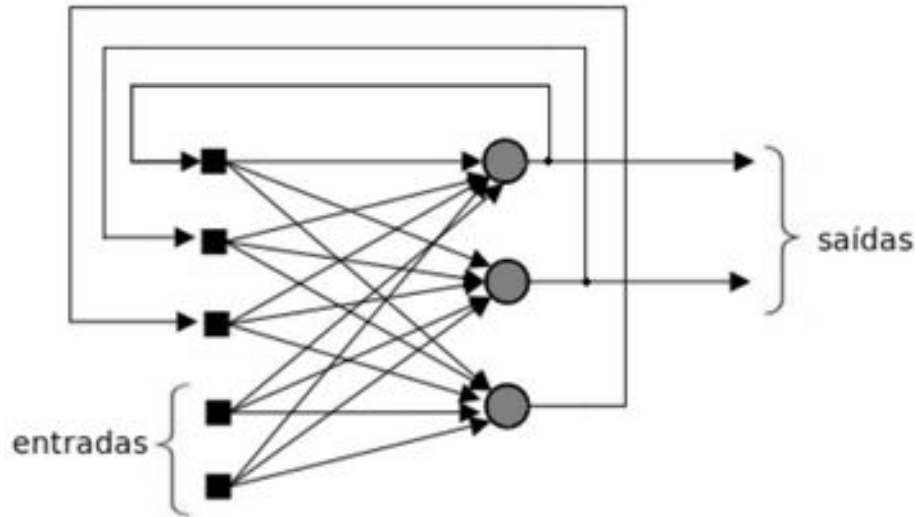
- Se refere à disposição dos neurônios na rede, como são estruturados.
- A topologia da rede está diretamente ligada ao tipo de algoritmo de aprendizagem utilizado.
- Classes de Topologias:
 - Redes alimentadas adiante (*Feed-forward*)
 - Redes Recorrentes (*Feed-backward*)
 - Redes Competitivas

Redes Alimentadas adiante



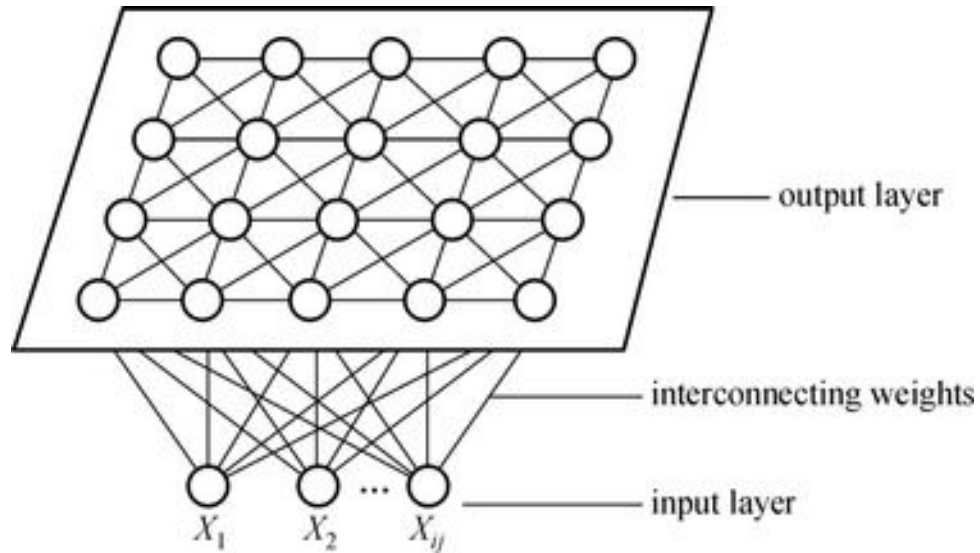
- Neurônios organizados em camadas
- Camada inicial é a de entrada
- A camada final contém as saídas
- As camadas intermediárias são chamadas de Camadas Ocultas
- Cada neurônio se conecta com todos da camada seguinte
- Neurônios da mesma camada não se conectam

Redes Recorrentes



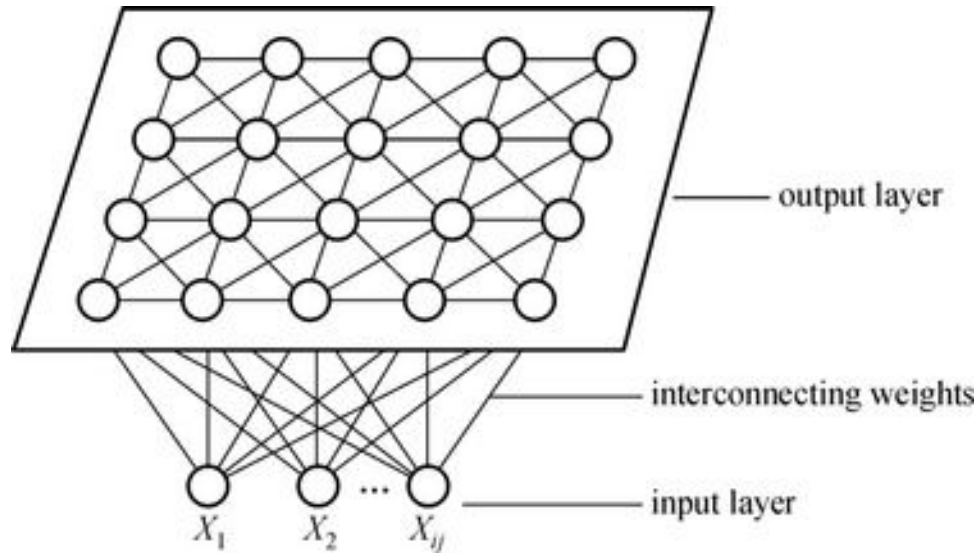
- Existe a ocorrência de realimentação
- A saída de um neurônio é aplicada como entrada no próprio neurônio e/ou em outros neurônios de camadas anteriores
- Ou seja, ocorrem ciclos no grafo.

Redes Competitivas



- Os neurônios são divididos em duas camadas
 - Camada de entrada, ou nós fontes
 - Camada de saída, conhecida como “grade”
- Os neurônios da grade são competem entre si, com base no nível de similaridade entre o padrão de entrada e a grade de neurônios
- Somente o neurônio vencedor é disparado (ativado) a cada iteração
- Utilizam algoritmo de aprendizagem competitivo.
- A rede mais conhecida desta classe é a rede de Kohonen, ou Mapa Auto- Organizável (SOM)

Redes Competitivas



- Os neurônios são divididos em duas camadas
 - Camada de entrada, ou nós fontes
 - Camada de saída, conhecida como “grade”
- Os neurônios da grade são competem entre si, com base no nível de similaridade entre o padrão de entrada e a grade de neurônios
- Somente o neurônio vencedor é disparado (ativado) a cada iteração
- Utilizam algoritmo de aprendizagem competitivo.
- A rede mais conhecida desta classe é a rede de Kohonen, ou Mapa Auto- Organizável (SOM)

Processo de Aprendizagem

- A capacidade da rede de aprender é a principal característica
- É um processo de atualização da representação interna do sistema em resposta a um estímulo externo, para resolver uma tarefa específica
- A atualização da representação é feita pela modificação da arquitetura, ajuste dos pesos das conexões entre os neurônios e ativação regras de neurônios individuais
- As regras de aprendizagem definem como a rede deve ajustar os pesos sinápticos
- 4 Tipos básicos

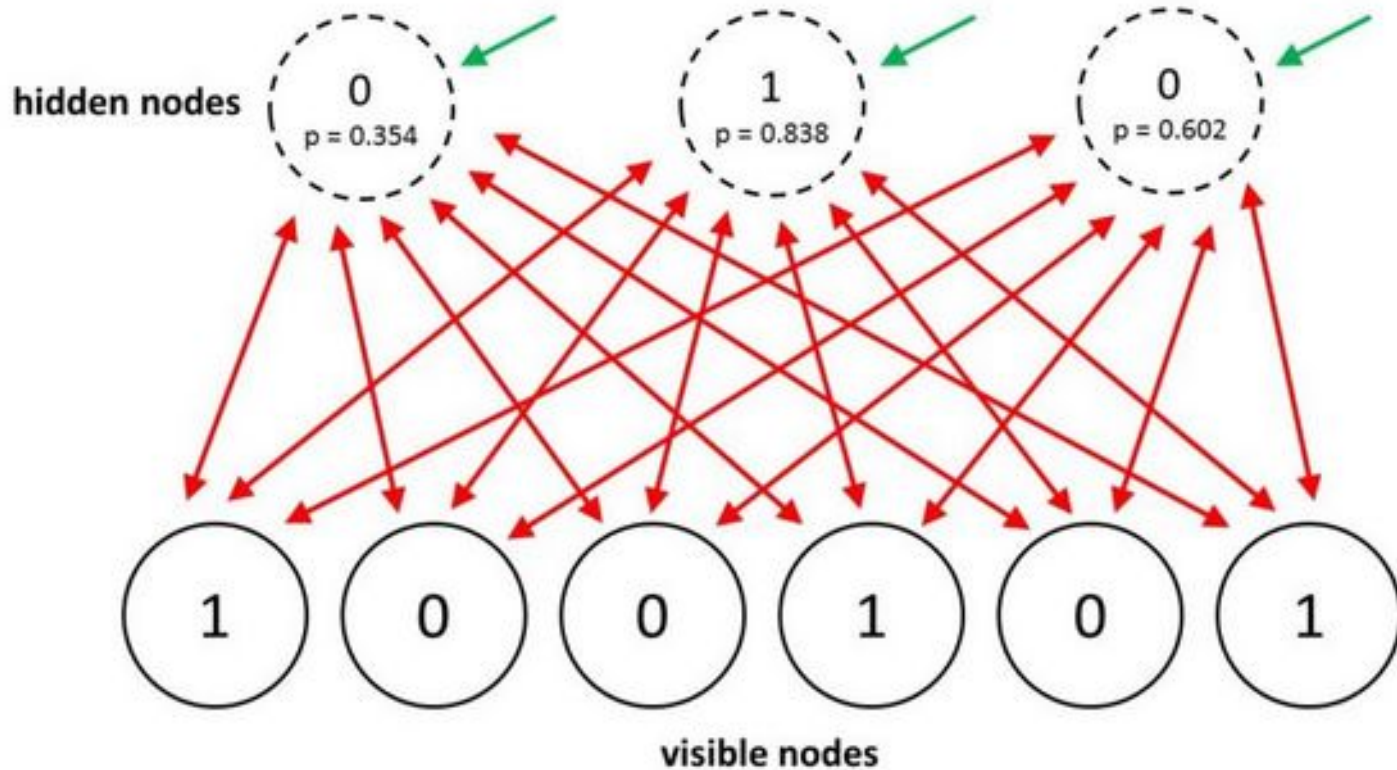
Aprendizagem por Correção de Erro

- Treinamento supervisionado
- Ajusta os pesos sinápticos por meio do erro, que é obtido através da diferença entre o valor de saída da rede e o valor esperado em um ciclo de treinamento.
- Diminui gradualmente o erro geral da rede

Aprendizagem Hebbiana

- Baseado no postulado de aprendizagem de Hebb (1949)
 - “Se dois neurônios em ambos os lados de uma sinapse são ativados sincronamente e simultaneamente, então a força daquela sinapse é seletivamente aumentada”
- Processo de treinamento local, ajustando o peso das conexões baseado nas atividades dos neurônios.

Aprendizagem de Boltzmann



Aprendizagem de Boltzmann

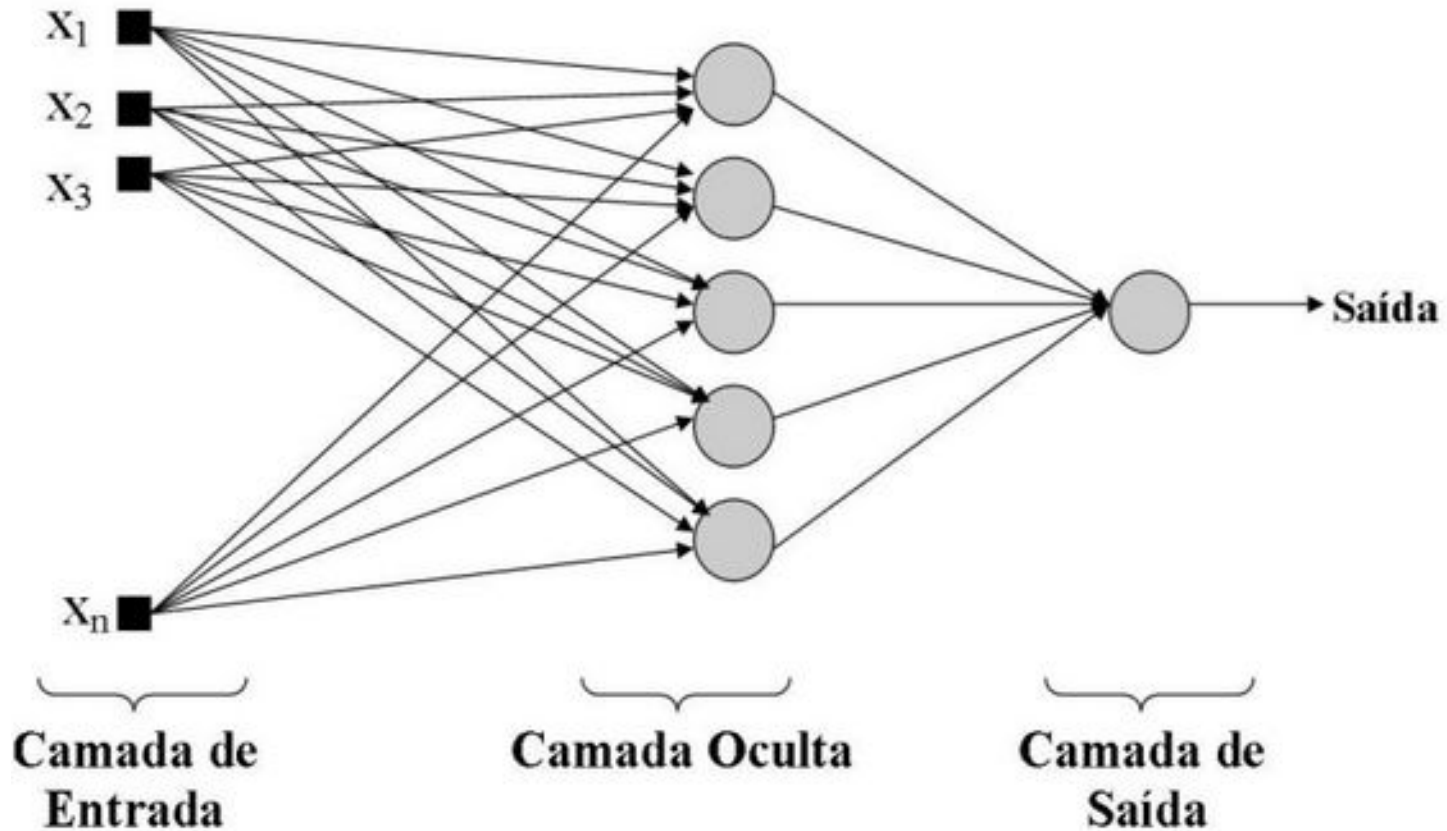
- Método de aprendizagem estocástico derivado das ideias da mecânica estatística.
- A rede é dita ser uma MB!
- Neste modelo os neurônios são estocásticos, podendo residir em dois estados possíveis, ligado (+1) e desligado (-1)
- Os neurônios são divididos em dois grupos funcionais, presos e livres, sendo responsáveis pela interação com o ambiente e pela explicação das restrições subjacentes dos padrões de entrada do ambiente, respectivamente.
- Na MB, os neurônios possuem conexões bidirecionais
- Aprendizagem não-supervisionado para modelar uma distribuição de probabilidade, especificada pelos padrões presos aos neurônios visíveis com probabilidades apropriadas.

Aprendizagem Competitiva

- Os neurônios competem entre si e, a cada iteração, somente o vencedor será ativado
- Todos os pesos dos neurônios próximos ao neurônio vencedor têm seus valores ajustados.

Rede de Multicamadas

Multilayer Percetron - MLP



Redes Multilayer Perceptron

- São redes de alimentação adiante que possuem uma ou mais camadas de neurônios entre as camadas de entrada e saída, chamada de camada oculta
- A camada oculta adiciona um poder maior em relação às redes Perceptron de camada única, que classifica apenas padrões linearmente separáveis
- Os neurônios ocultos responsáveis por capturar a não-linearidade dos dados.
- Todos os neurônios são ligados aos neurônios da camada subsequente
- Não possuem ligação com os neurônios da mesma camada não tem realimentação.
- A aprendizagem é um processo iterativo, conhecido como aprendizagem por experiência
 - Padrões de treinamento (exemplos) são apresentados a rede
 - com base nos erros obtidos, são realizados ajustes nos pesos sinápticos, com o intuito de diminuir o erro nas próximas iterações.

Redes Multilayer Perceptron - Backpropagation

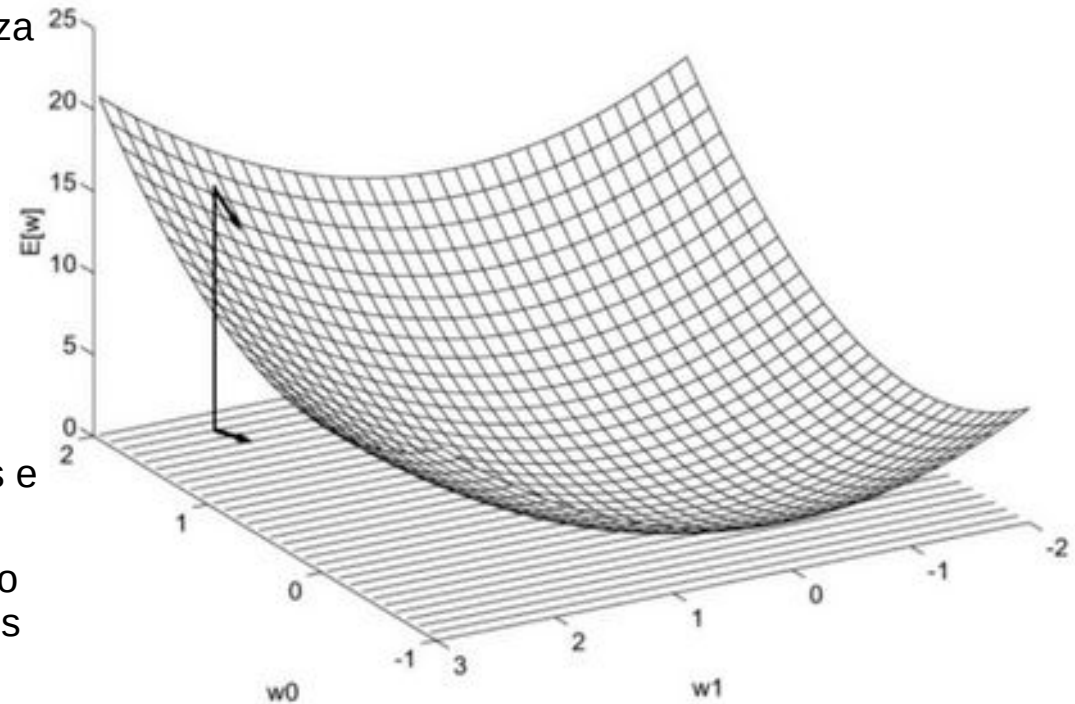
- O principal algoritmo de treinamento é o algoritmo de retropropagação de erro, mais conhecido como ***backpropagation***
- baseada na regra de aprendizagem por correção de erro (dois passos, um para frente e outro para trás):
 - **Passo para frente** (propagação): os valores provindos dos neurônios de entrada são aplicados aos neurônios ocultos e posteriormente suas saídas são aplicadas como entradas aos neurônios da camada final, obtendo a resposta da rede. Durante este passo os pesos sinápticos da rede são todos fixos.
 - **Passo para trás** (retropropagação): Ajusta os pesos sinápticos, por meio do cálculo do erro realizado na camada de saída. Os pesos sinápticos entre as camadas antecessoras são ajustados de acordo com uma regra de correção de erro.

Backpropagation

- **Aprende os pesos para uma rede multicamadas**, dada uma rede com um número fixo de unidades e interconexões.
- O algoritmo emprega a **descida do gradiente** para minimizar o erro quadrático entre a saída da rede e os valores alvos para estas saídas.

Backpropagation – Descida do Gradiente

- Busca determinar um vetor de pesos que minimiza o erro.
- Inicia com um vetor inicial de pesos arbitrário e modificando-o repetidamente em pequenos passos.
- A cada passo, o vetor de pesos é alterado na direção que produz a maior queda ao longo da superfície de erro.
- Aprende os pesos para uma rede multicamadas, dada uma rede com um número fixo de unidades e interconexões.
- Emprega a descida do gradiente para minimizar o erro quadrático entre a saída da rede e os valores alvos para estas saídas.



Backpropagation

Algoritmo 1: Algoritmo de treinamento *backpropagation*

```
1 begin  
2   Atribuição de valores iniciais aos pesos sinápticos;  
3   repeat  
4     Apresentação à rede dos padrões de entrada e as saídas desejadas;  
5     Cálculo dos valores de saída dos neurônios ocultos;  
6     Cálculo dos valores de saída dos neurônios de saída (resposta real da rede);  
7     Cálculo do erro (diferença entre resposta da rede e valor esperado);  
8     Ajuste dos pesos sinápticos;  
9   until Condição de parada não satisfeita ;  
10 end
```

Backpropagation

- **Linha 2:** Atribuição de valores iniciais dos pesos sinápticos, normalmente aleatória no intervalo $\{0,1\}$
- **Linha 4:** Apresentação dos padrões de entrada e saída conhecida previamente (treinamento supervisionado)

Backpropagation – Linhas 5 e 6

- O cálculo dos valores de saída são realizados pela aplicação do campo local induzido (v) à uma função de ativação. De uma maneira matemática, o campo local induzido para o neurônio j na camada l na iteração n é obtido por:

$$v_j^{(l)}(n) = \sum_{i=1}^{r+1} w_{ji}^{(l)}(n) y_i^{(l-1)}(n)$$

onde $y_i^{(l-1)}(n)$ é o sinal de saída do neurônio i na camada anterior $l - 1$, na iteração n , $w_{ji}^{(l)}$ é o peso sináptico do neurônio j da camada l , que é alimentado pelo neurônio i da camada $l - 1$ e r o número de neurônios na camada anterior ($l - 1$). Observe que ao invés de r é utilizado $r + 1$, isto se deve ao fato da inclusão de um *bias*, um neurônio adicional com valor $+1$. O bias equivale a dizer: $y_{r+1}^{(l-1)}(n) = +1$.

Denotaremos por $\varphi_j(v_j)$, sendo $\varphi(\cdot)$ a função de ativação e v_j o campo local induzido do neurônio j , o valor de saída de um neurônio i da camada l . Assim o valor de saída do neurônio

$$y_j^{(l)} = \varphi(v_j(n))$$

Backpropagation

- **Linha 7:** O sinal de erro da saída da rede, na iteração n é calculada por $e_j(n) = d_j(n) - y_j(n)$, onde d_j é a j -ésima resposta desejada e y_j é a j -ésima resposta da rede;
- **Linha 8:** O ajuste dos pesos é o núcleo deste algoritmo, sendo realizado da camada oculta (l) para a camada de entrada. Em qualquer camada l os valores dos pesos $w_{ji}^{(l)}$ na iteração n serão ajustados da iteração anterior ($n - 1$) de acordo com:

Backpropagation – Linha 8

$$w_{ji}^{(l)}(n) = w_{ji}^{(l)}(n-1) + \Delta w_{ji}^{(l)}(n)$$

onde $\Delta w_{ji}^{(l)}(n)$ é correção aplicada ao peso da conexão. A correção é determinada pela *regra delta modificada*, definida como segue:

$$\Delta w_{ji}^{(l)}(n+1) = \eta \delta_j^{(l)} y_i^{(l-1)} + \mu \Delta w_{ji}^{(l)}(n)$$

onde η é a taxa de aprendizagem, $\delta_j^{(l)}$ é o gradiente local, μ é a constante de momento e $y_i^{(l-1)}$ é o sinal de saída do neurônio i na camada anterior $l-1$.

A taxa de aprendizagem define o tamanho do passo de atualização e a constante de momento é utilizado para que o método possa fugir do mínimo local na superfície de erro, objetivando o mínimo global.

Backpropagation – Linha 8

O gradiente local é calculado de maneira distinta entre neurônios de saída e ocultos. Sendo L a camada de saída, o gradiente do neurônio j , na iteração n da camada de saída é calculado por:

$$\delta_j^{(L)}(n) = e_j^{(L)}(n) \varphi'(v_j^{(L)})$$

onde $\varphi'(\cdot)$ é derivada da função de ativação. Para os neurônios ocultos a correção é obtida por

$$\delta_j^{(l)}(n) = \varphi'(v_j^{(l)}) \left(\sum_{i=1}^{r+1} \delta_i^{(l+1)} w_{ij}^{(l+1)} \right)$$

sendo l uma camada oculta qualquer.

Backpropagation – Linha 9

- Basheer e Hajmeer (2000) identificam alguns critérios de parada, dentre eles (i) erro de treinamento ($e \leq \epsilon$), (ii) gradiente do erro menor que um σ ou (iii) utilizando técnica de validação cruzada.
- O critério de parada (iii) é geralmente utilizado através da análise gráfica do comportamento do erro, com base na técnica de validação cruzada

MLP – Dificuldades de Implementação

- Necessário fazer uma boa configuração para que uma MLP possa obter bons resultados na aplicação em problemas reais
- Deve ser realizada de maneira distinta dependendo de várias características do problema, como
 - tipo dos dados de entrada (inteiro, booleanos, híbrido)
 - número de padrões disponíveis para treinamento e teste
 - dimensionalidade dos dados de entrada, entre outros.
- Para isso, valores ideais dos vários parâmetros da rede MLP devem ser utilizados, porém estes valores não são de fácil obtenção.

MLP – Dificuldades de Implementação

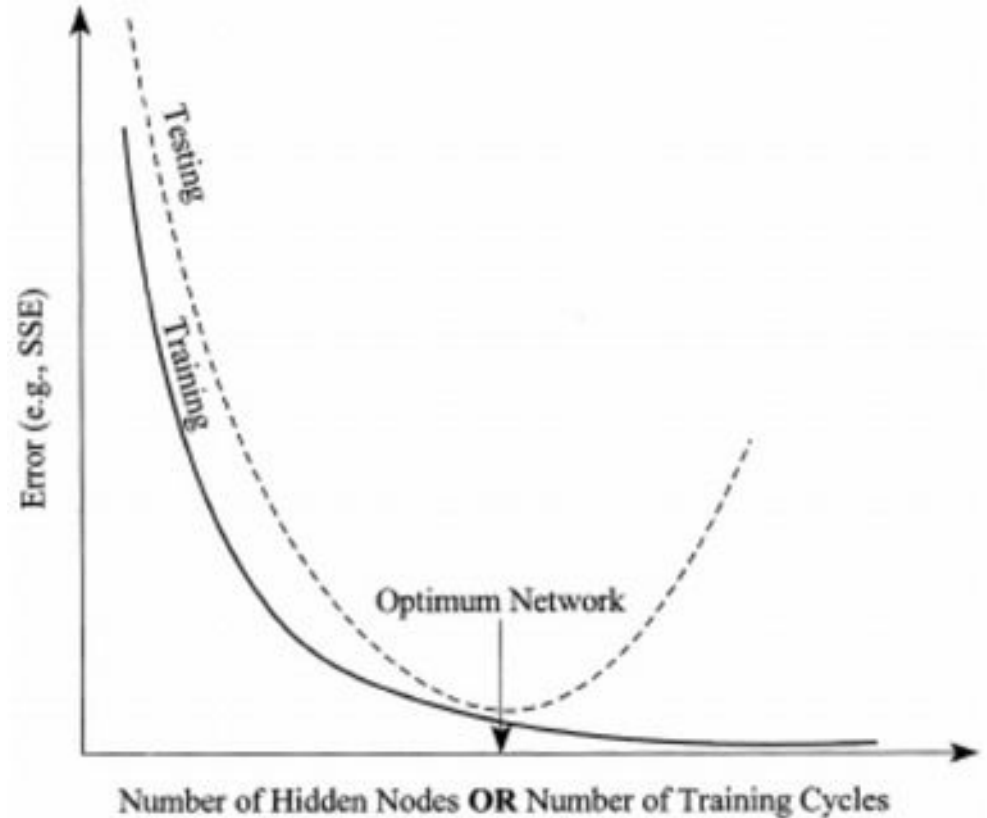
- Alguns parâmetros são determinados por tentativa e erro
 - são atribuídos vários valores distintos aos parâmetros e analisando os resultados obtidos, a melhor configuração é escolhida.
 - Dentre os parâmetros, os que possuem maior dificuldade para ajuste são:
 - taxa de aprendizagem
 - constante de momento
 - número de camadas ocultas
 - Número de neurônios nas camadas ocultas

MLP – Dificuldades de Implementação

- Algumas soluções para contornar este problema foram sugeridas, como aplicação de outras técnicas de aprendizagem de máquina, as quais são conhecidas pela alta aplicabilidade em problemas de otimização, como:
 - Algoritmos Genéticos
 - Otimização por Enxame de Partículas
 - Otimização por Colônias de Formigas

MLP – Dificuldades de Implementação

- Outra dificuldade é a determinação do número ideal de ciclos de treinamento da rede
 - Determinado por tentativa e erro.
 - Se um número muito grande de ciclos de treinamento for aplicado, a rede entra em um processo de "memorização" dos padrões a ela apresentados, chamado de supertreinamento (*overtraining*), perdendo a capacidade de generalização.
 - Se um número muito pequeno for aplicado, a rede torna-se incapaz de representar os dados.



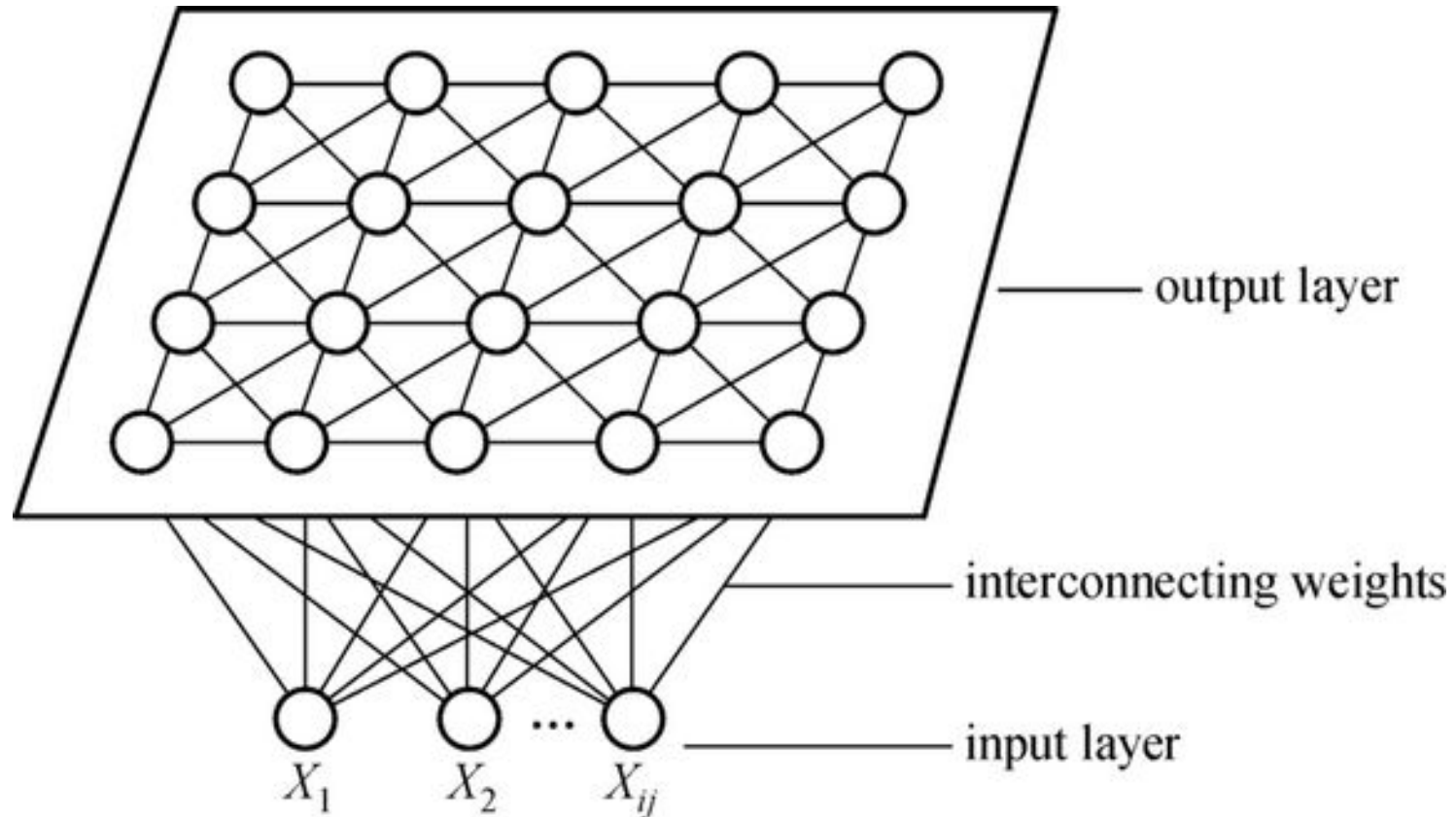
Redes de Função de Base Radial – RBF!

- Apresentam a rede neural como um problema de ajuste de curvas (aproximação) em um espaço de alta dimensionalidade
- A aprendizagem de uma rede RBF! É um processo de busca em uma superfície multidimensional, que forneça melhores ajustes para os dados de treinamento, sendo os “melhores ajustes” uma medida no sentido estocástico e a generalização equivale ao uso da superfície multidimensional para interpolação dos dados de teste.
- Se diferem das redes MLP! por três características principais:
 - Sempre apresenta uma única camada intermediária (oculta);
 - Neurônios da saída são sempre lineares;
 - Neurônios da camada intermediária têm funções de base radial como função de ativação, ao invés de funções sigmoidais ou outras.

Redes de Função de Base Radial – RBF!

- Em suma, é uma rede constituída por três camadas, com papéis distintos.
 - A **camada inicial**, ou nós fonte, alimentada com os dados de entrada
 - Uma única **camada oculta** responsável pela transformação não-linear do espaço de entrada para o espaço oculto
 - A **camada de saída**, linear, fornece uma resposta ao estímulo gerado pela aplicação de um padrão (dados de entrada), pela camada de entrada.

Mapas Auto-organizáveis – SOM!



Mapas Auto-organizáveis – SOM!

- Aprendizagem não-supervisionada.
- Conhecida como Rede de Kohonen (1982).
- Base biológica
 - a forma como os neurônios se organizam muitas vezes refletem algumas características físicas sentidas pelos estímulos externos.
 - O cérebro humano é organizado em várias áreas, de modo que entradas sensoriais diferentes são representadas por mapas computacionais ordenados topologicamente
- Possui duas camadas
 - camada de entrada
 - grade pós-sináptica (ou mapa de características), bidimensional.

Mapas Auto-organizáveis – SOM!

- A grade pós-sináptica os neurônios são interligados aos neurônios mais próximos e os neurônios de entrada são ligados com todos os neurônios da grade.
- Usa uma aprendizagem híbrida composta por aprendizagem Hebbiana e competitiva
 - os neurônios da grade pós-sináptica competem entre si, com base em uma medida de similaridade com o sinal de entrada
 - o neurônio mais similar é dito ser o vencedor
 - o neurônio vencedor excita os neurônios próximos a ele.
- A distância euclidiana é geralmente utilizada como medida de similaridade.
- O algoritmo responsável pela formação do mapa começa pela atribuição de valores iniciais dos pesos sinápticos da grade, que deve ser feito atribuindo valores pequenos gerados aleatoriamente.

Mapas Auto-organizáveis – SOM!

- **Competição:** Para cada padrão de entrada, os neurônios da grade competem entre si, calculando a distância euclidiana entre seus pesos sinápticos e os valores do padrão de entrada, sendo o neurônio com a menor distância o vencedor (aprendizagem competitiva)
- **Cooperação:** O neurônio vencedor determina uma área, vizinhança topológica, na qual os neurônios cooperarão entre si (aprendizagem Hebbiana)
- **Adaptação Sináptica:** Determinada a vizinhança, os neurônios cooperam de modo a atualizarem seus pesos sinápticos, sendo que os mais próximos do neurônio vencedor sofrem modificações mais significativas do que os neurônios mais distantes.

Mapas Auto-organizáveis – SOM!

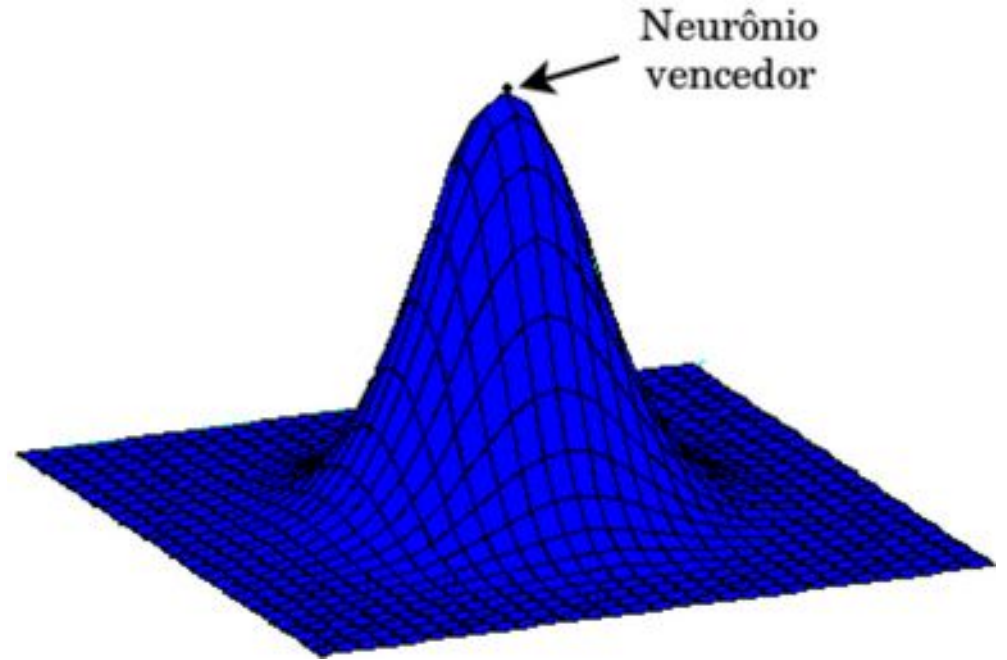
A função Gaussiana é utilizada para definir como será feita a atualização dos pesos dos neurônios na vizinhança, neste caso o neurônio vencedor está localizado no centro da vizinhança. A Equação mostra o cálculo da excitação de um neurônio j , sendo d a distância entre o neurônio j e o neurônio vencedor i . A distância euclidiana, definida por $D(x, y) = \sqrt{x^2 + y^2}$, é comumente utilizada.

$$h_{ji} = \exp\left(-\frac{d_{ji}^2}{2\sigma^2}\right)$$

onde σ é a "largura efetiva" da vizinhança topológica. Ela mede o grau com o qual neurônios excitados na vizinhança participam do processo de aprendizagem

Mapas Auto-organizáveis – SOM!

- A vizinhança topológica gaussiana é mais biologicamente apropriada do que uma vizinhança retangular.



Mapas Auto-organizáveis – SOM!

- Com o decorrer da execução do algoritmo, os padrões de entrada similares são mapeados em neurônios topologicamente próximos, formando áreas ou *clusters*
- O SOM! é aplicado à clusterização de dados, ou seja, agrupamento de dados intrinsecamente semelhantes, utilizado quando nada é conhecido sobre os dados.
- Os mapas de Kohonen podem ser aplicados em compressão de dados, uma vez que dados de alta dimensionalidade são mapeados em um espaço de baixa dimensão, preservando seu conteúdo.

Rede de Hopfield

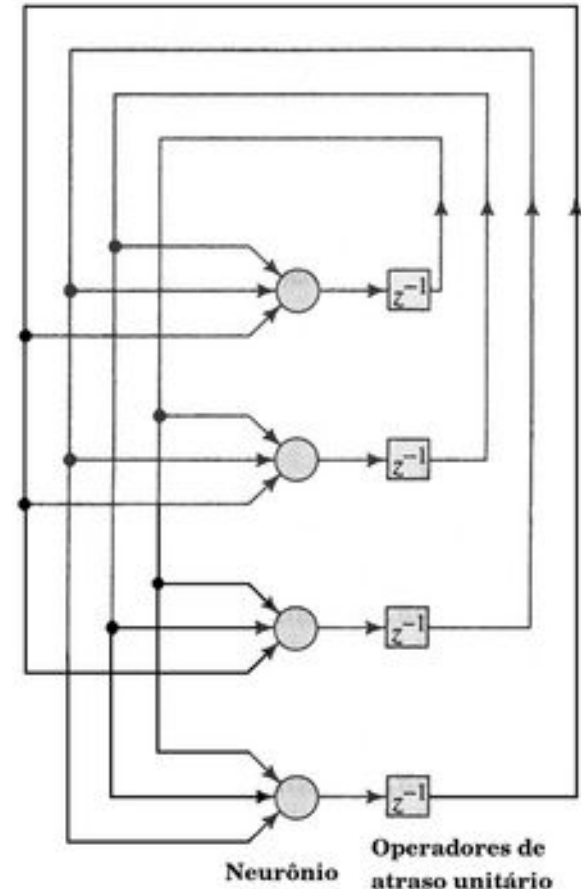
- Rede totalmente conectada agindo como uma **memória associativa**, capaz de armazenar padrões
- Uma memória associativa serve para armazenar um conjunto de vetores, de tal forma que se for endereçada com um vetor arbitrário y , retorna como saída aquele vetor mais próximo em algum sentido pré-definido.
- Geralmente é utilizada a **distância de Hamming**, como forma de mensurar a proximidade entre os vetores.
- A distância de Hamming entre dois vetores (strings) de mesmo tamanho é definido pelo número de posições que os símbolos correspondentes são diferentes.
- Uma memória associativa pode ser interpretada como um classificador de padrões, onde as classes são representadas pelos vetores armazenados.

Rede de Hopfield

- É a implementação de uma memória associativa por uma rede recorrente, porém sem auto-relimentação.
- Neste modelo os vetores de entrada são normalmente binários.
- A rede é iniciada por um vetor $x(0)$ e gera $y(0)$ como saída, sendo $y(0)$ reaplicado como entrada da rede no ciclo posterior.
- Esse processo (dinâmica da rede) é realizado até que a rede se estabilize, ou seja:

$$y(k + 1) = \sim y(k)$$

- O estado no qual a rede se estabilizou é dito ser um mínimo local do problema.



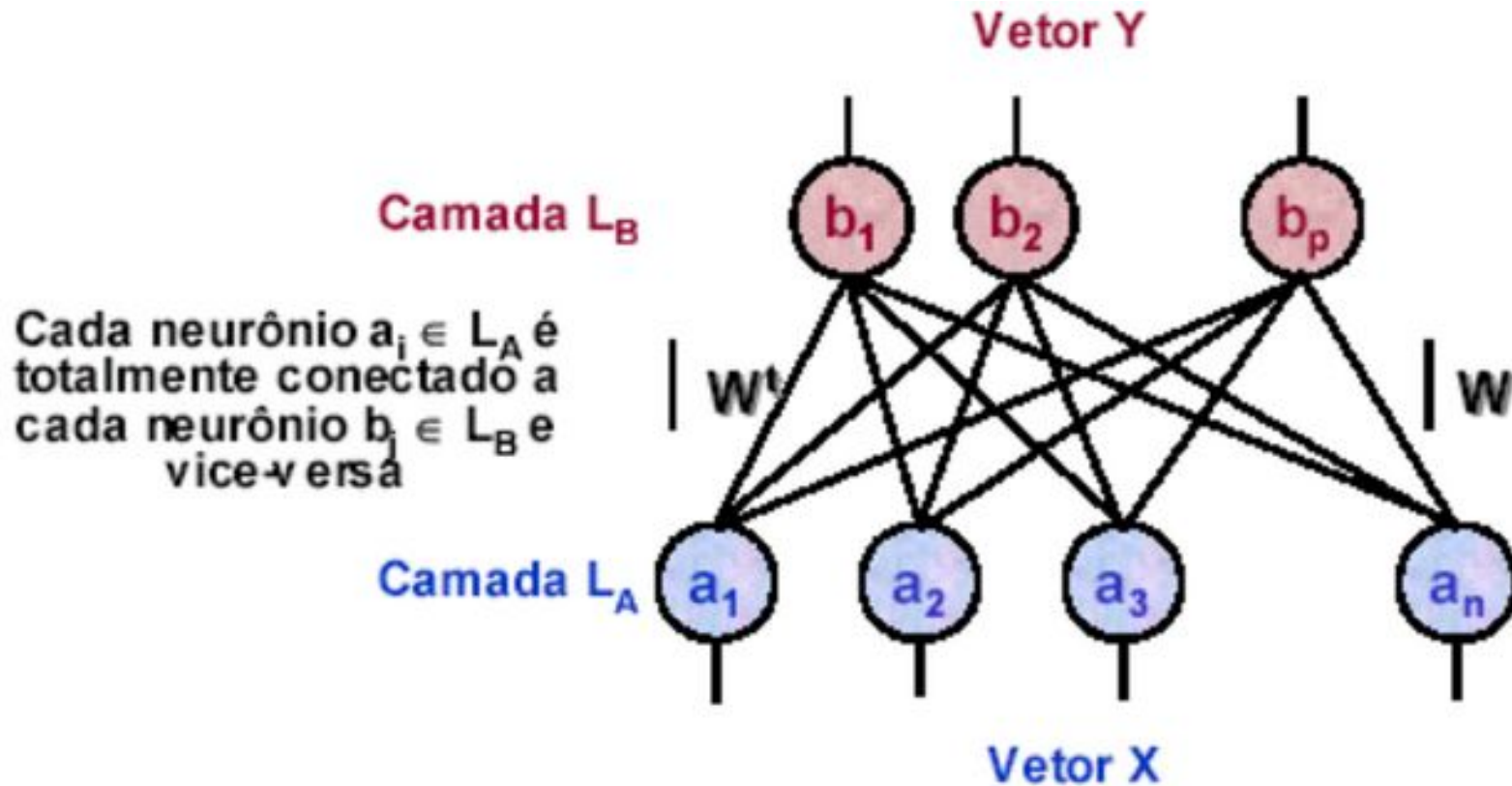
Rede de Hopfield

- Uma função de energia (função de Lyapunov) é utilizada para análise do processo de evolução da rede.
- A função de energia da rede é submetida a sua própria dinâmica e só pode decrescer ou permanecer constante, ou seja, no decorrer da execução a rede tenta ir para estados melhores ou no mínimo iguais ao inicial.
- A rede de Hopfield é mais apropriada quando uma exata representação binária é possível e menos apropriada quando os valores de entrada são contínuos.

Memória Associativa – BAM

- Modelos de memória associativa que apresentam ligações recorrentes em sua estrutura.
- A estruturada BAM é baseada na interligação de elementos processadores, as quais possuem pesos associados.
- São redes decorrentes de 2 camadas com conexões bidirecionais.
- São memórias associativas do tipo heteroassociativa.
- A rede aceita um mapeamento entre vetores diferentes.
- Processo dinâmico converge para o padrão mais próximo.

Memória Associativa – BAM



Memória Associativa – BAM

- Todos os seus neurônios se interconectam, permitindo uma associação dinâmica,
 - ou seja existe uma conversão para o padrão mais próximo.
- Conforme figura anterior, a rede aceita o mapeamento entre vetores diferentes demonstrando assim sua dinamicidade.
- Relacionamento de vetores conforme funcionamento da BAM
 - **Para uma associação $X \Rightarrow Y$:** O vetor X é apresentado à camada L_a , multiplicado pela matriz W , apresentando a saída Y na camada L_b . Essa saída é então aplicada à camada L_b , multiplicada pela matriz W^t , produzindo X na camada L_a .

Memória Associativa – BAM

- Quando X é associado com Y, um conjunto de peso entre os nodos na rede são trocados por associação dos componentes.
- Sob a apresentação de cada vetor X ou Y, a rede é geralmente capaz de lembrar os vetores desaparecidos.
- A BAM é bastante interessante quando a falha é lembrada.
 - Isto acontece quando uma substring do vetor é identificada por outro vetor previamente treinado.
- Uma rede BAM pode estabilizar aprendendo somente uma fração do potencial da capacidade do sistema, quando procurar na capacidade de apresentação do vetor de pesos.

Inteligência Artificial

Redes Neurais Artificiais Parte II

Prof. Saulo Popov Zambiasi
saulopz@gmail.com