INE5377 – Top. Esp em Soft. Aplicativo II

INTELIGÊNCIA ARTIFICIAL II

Ementa:
Visão detalhada e comparativa das abordagens "não simbólicas" de Inteligência Artificial, também conhecidas como Inteligência Computacional, envolvendo a abordagem conexionista, a evolutiva e a lógica "fuzzy".

Prof. Mauro Roisenberg

Tópicos

- Revisão de Conceitos Básicos de Redes Neurais
- Histórico das Redes Neurais Artificiais.
- Nomenclatura Básica, o Neurônio Biológico.
- Modelos Básicos: Multi-layer Perceptron e Regra de Aprendizado Back-Propagation.
- Redes "Counterpropagation".
- Memórias Associativas: Hopfield e BAM.
- Redes de Kohonen.
- Redes ART
- Redes Recorrentes.

Tópicos

- Tópicos de Lógica Nebulosos.
- Tópicos de Teoria dos Conjuntos Nebulosos.
- Sistemas Especialistas Nebulosos.
- Tópicos de Computação Evolucionária
- Algoritmos Genéticos
- Programação Evolucionária.

Bibliografia

- Kasabov, Nikola K. Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. MIT Press, 1996.
- Freeman, James A. & Skapura, David M. Neural Networks: Algorithms, Applications and Programming Techniques. Addison-Wesley Publishing, 1992.
- Haykin, Simon. Redes Neurais: princípios e prática. Bookman, 2a. Ed., 2001.
- Barreto, Jorge M. Inteligência Artificial: No limiar do século XXI. Duplic Edições, 1997.
- Kovács, Zsolt Laszlo. Redes Neurais Artificiais: fundamentos e aplicações. Collegium Cognitio, 1997.
- Rumelhart, D.; Hinton, G. & Williams, R. Learning Internal representation by Error Propagation. In: Parallel Distributed Processing: explorations in the microstructure of cognition - Vol 1. MIT Press, 1986.
- Arbib, Michael A. (Ed) The handbook of Brain Theory and Neural Networks. MIT Press, 1995.
- Mehrotra, K.; Mohan, C. K. & Ranka, S. Elements of Artificial Neural Networks. MIT Press, 2000.

Revisão de Conceitos Básicos de Redes Neurais

- Histórico
- Estrutura e Funcionamento de um Único Neurônio
  - Neurônios Biológicos
  - Modelos de Neurônios Artificiais
- Arquiteturas de Redes Neurais
  - Redes Totalmente Conectadas
  - Redes em Camadas
    - Redes Acíclicas
    - Redes Ciclicas
    - Redes Feedforward
    - Redes Modulares

Revisão de Conceitos Básicos de Redes Neurais

- Aprendizado em Redes Neurais
  - Aprendizado Supervisionado
  - Pesos Pré-determinados
  - Aprendizado por Correlação
  - Aprendizado por Realimentação
- Aprendizado Não-Supervisionado
  - Aprendizado Competitivo

Revisão de Conceitos Básicos de Redes Neurais

- Aplicações de Redes Neurais
  - Classificação
  - Clusterização
  - Quantização Vetorial (compressão)
  - Associação de Padrões
  - Aproximação de funções
  - Previsões
  - Aplicações em Controle
  - Otimização
  - Busca

Revisão de Conceitos Básicos de Redes Neurais

- Avaliação de Redes
  - Avaliação da capacidade de aprendizado de um ser humano
  - Avaliação do desempenho das redes neurais
    - Qualidade de Resultados
    - Capacidade de Generalização
    - Recursos Computacionais

Fases Históricas

- **Época Pré-Histórica (até 1875 quando Camillo Golgi visualizou o neurônio)**
  - **Objetivo:**
    - **Criar seres e mecanismos apresentando comportamento inteligente.**
  - **Metodologia e Conquistas:**
    - **Mecanismos usando mecânica de precisão desenvolvida nos autômatos, mecanismos baseados em teares, etc.**
  - **Limitações:**
    - **Complexidade dos mecanismos, dificuldades de construção.**

Fases Históricas

- **Época Antiga (1875-1943 - Neurônio de McCulloch & Pitts)**
  - **Objetivo:**
    - **Entender a Inteligência Humana.**
  - **Metodologia e Conquistas:**
    - **Estudos de psicologia e neurofisiologia. Nascimento da psicanálise.**
  - **Limitações:**
    - **Grande distância entre as conquistas da psicologia e da neurofisiologia.**

Fases Históricas

- **Época Romântica (1943-1956 - Reunião no Darthmouth College)**
  - **Objetivo:**
    - **Simular a Inteligência Humana .**
  - **Metodologia e Conquistas:**
    - **Inspiração na Natureza, Nascimento da Cibernética. Primeiros mecanismos imitando o funcionamento de redes de neurônios. Primeiros programas imitando comportamento inteligente.**
  - **Limitações:**
    - **Limitações das capacidades computacionais.**

Fases Históricas

- **Época Barroca (1956-1969 - Livro Perceptrons)**
  - **Objetivo:**
    - **Expandir ao Máximo as aplicações da IA tanto usando a abordagem simbólica quanto a conexionista.**
  - **Metodologia e Conquistas:**
    - **Perceptron. Primeiros sistemas especialistas usando a abordagem simbólica.**
  - **Limitações:**
    - **Dificuldades em técnicas de aprendizado de redes complexas. Subestimação da complexidade computacional dos problemas.**

Fases Históricas

- **Época das Trevas (1969-1981 - Anuncio dos Computadores de Quinta Geração)**
  - **Objetivo:**
    - **Encontrar para a IA aplicações práticas. Simular a Inteligência Humana em situações pré-determinadas.**
  - **Metodologia e Conquistas:**
    - **Sistemas Especialistas. Formalismos de representação de conhecimento adaptados ao tipo de problema.**
  - **Limitações:**
    - **Subestimação da quantidade de conhecimento necessária para tratar mesmo o mais banal problema de senso comum.**

Fases Históricas

- **Renascimento (1981-1987 - Primeira Conferência Internacional em Redes Neurais)**
  - **Objetivo:**
    - **Renascimento da IA Simbólica e Conexionista.**
  - **Metodologia e Conquistas:**
    - **Sistemas de regras, representação da incerteza, popularização do Prolog. Alguns pesquisadores criando condições para a fase seguinte no que diz respeito às Redes Neurais.**
  - **Limitações:**
    - **IA Simbólica e Conexionista evoluindo separadamente.**

Fases Históricas

- **Época Contemporânea (1987-...)**
  - **Objetivo:**
    - **Alargamento das aplicações das Redes Neurais Artificiais (RNAs).**
  - **Metodologia e Conquistas:**
    - **Redes Diretas como aproximador universal. Bons resultados em problemas mal-definidos.**
  - **Limitações:**
    - **Falta de um formalismo e de uma profunda análise matemática sobre as capacidades das Redes Neurais. Falta de estudos sobre computabilidade e complexidades neurais.**

Estrutura e Funcionamento de um único neurônio

- Um neurônio típico é composto por um corpo celular, um axônio tubular e várias ramificações arbóreas conhecidas como dendritos. Os dendritos formam uma malha de filamentos finíssimos ao redor do neurônio. O axônio é essencialmente um tubo longo e fino que ao final se divide em ramos que terminam em pequenos bulbos que quase tocam os dendritos dos outros neurônios. O pequeno espaço entre o fim do bulbo e o dendrito é conhecido como sinapse, através da qual as informações se propagam.
- O número de sinapses recebidas por cada neurônio variam de 100 a 100000. As sinapses podem ser excitatórias ou inibitórias.

Estrutura e Funcionamento de um único neurônio

- Tipos de Sinapses
  - Excitatórias (contém neurotransmisores como por exemplo: glutamato ou aspartato)
  - Inibitórias (contém neuro-bloqueadores como por exemplo: ácido gamma-amino butírico)
- Funcionamento
  - Uma diferença de potencial eletrostático é mantido entre o interior e o exterior da célula ao longo da membrana, com o interior sendo negativamente carregado. Íons se difundem através da membrana para manter a diferença de potencial.

Estrutura e Funcionamento de um único neurônio

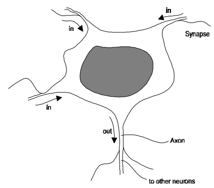
- Funcionamento
  - Sinais excitatórios ou inibitórios de outros neurônios chegam ao neurônio pela sinapse dos dendritos. A magnitude do sinal recebido por um neurônio depende da eficiência da transmissão sináptica, e pode ser pensada como uma força de contato entre os neurônios.
  - A membrana celular se torna eletricamente ativa quando suficientemente excitada pelas sinapse que chegam ao neurônios.

Estrutura e Funcionamento de um único neurônio

- Funcionamento
  - Um neurônio dispara, i.e. manda um impulso de saída de aproximadamente 100 mV através de seu axônio, se excitações suficiente aparecem em seus dendritos em um curto período de tempo, chamada período de adição latente. O neurônio dispara se o somatório das excitações superar o somatório das inibições por uma quantidade crítica conhecida como limiar ou "threshold".
  - O disparo é seguido por um período conhecido como refratário, durante o qual o neurônio fica inativo.

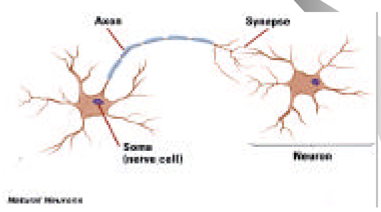
Estrutura e Funcionamento de um único neurônio

- Funcionamento



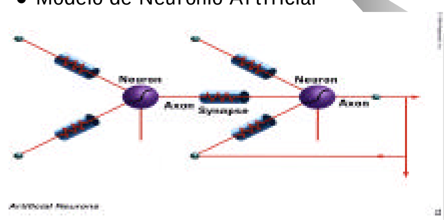
Estrutura e Funcionamento de um único neurônio

- Funcionamento



Estrutura e Funcionamento de um único neurônio

- Neurônios Artificiais
  - Um neurônio artificial é um modelo grosseiro do funcionamento de um neurônio biológico.
- Modelo de Neurônio Artificial



Estrutura e Funcionamento de um único neurônio

- Neurônios Artificiais
- Um neurônio artificial é um modelo grosseiro do funcionamento de um neurônio biológico.
- Modelo de Neurônio Artificial
  - As Entradas
    - As entradas de um neurônio podem ser as saídas de outros neurônios, entradas externas, um bias ou qualquer combinação destes elementos.
  - A Combinação das Entradas - O "Net"
    - O somatório de todas estas entradas, multiplicadas por suas respectivas forças de conexão sináptica (os pesos), dá origem ao chamado "net" de um neurônio.
    - wij é um número real que representa a conexão sináptica da entrada do i-ésimo neurônio com a saída do j-ésimo neurônio.

Estrutura e Funcionamento de um único neurônio

  - A Combinação das Entradas - O "Net"
    - A conexão sináptica é conhecida como excitatória se wij>0 ou inibitória caso wij<0
- A Função de Saída
  - Essencialmente, qualquer função contínua e monotonicamente crescente tal quepode ser utilizada como função de saída na modelagem neural. Existem, no entanto, uma série de funções mais comumente utilizadas como funções de saída em neurônios. Estas funções são:
    - A Função Linear
    - A Função Sigmoidal ou Logística
    - A Função Tangente Hiperbólica

Estrutura e Funcionamento de um único neurônio

- A Função de Saída
  - A Função Linear

$$y(x) = ax$$

  - A Função Sigmoidal ou Logística - Função UNIPOLAR mais utilizada.

$$y(x) = \frac{1}{1+e^{-kx}}$$

onde k é um fator de escala positivo.

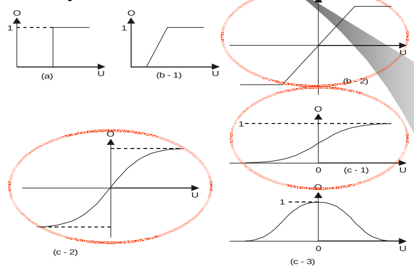
  - A Função Tangente Hiperbólica - A Função BIPOLAR mais utilizada.

$$y(x) = \tanh(kx) = \frac{1-e^{-kx}}{1+e^{-kx}}$$

  - onde k é um fator de escala positivo.

Estrutura e Funcionamento de um único neurônio

- **Funções de Saída mais Utilizadas**



Topologias Das Redes Neurais Como os Neurônio se Conectam

- Um único neurônio ou elemento processador é insuficiente para a maioria dos problemas práticos, e, redes com grande número de neurônios interconectados são freqüentemente utilizadas.
- A maneira como os neurônios são interconectados determina como as capacidades computacionais da rede e se constitui em uma das primeiras e mais importantes decisões no desenvolvimento de uma rede neural.
- Inspiração Biológica:
  - Diferentes partes do sistema nervoso central estão estruturadas de maneiras diversas; portanto é incorreto afirmar que uma única arquitetura é capaz de modelar todo o processamento neural.
  - O córtex cerebral, onde se acredita que ocorra a maior parte do processamento, consiste de cinco a sete camadas de neurônios com cada camada fornecendo as entradas para a camada seguinte.

Topologias Das Redes Neurais Como os Neurônio se Conectam

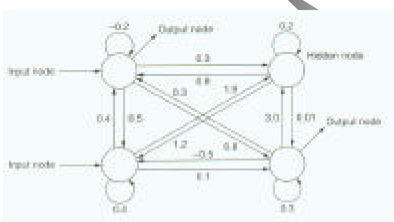
- Inspiração Biológica:
  - Entretanto as fronteiras entre as camadas não são estritas e não são raros os casos de conexões que atravessam camadas.
  - Conexões para trás (feedback) também são conhecidas, como por exemplo, entre o córtex visual e o núcleo geniculado lateral. Cada neurônio está conectado a vários, mas não a todos os neurônios vizinhos de uma mesma camada. Muitas destas conexões são excitatórias, mas algumas também são inibitórias.
  - Existem neurônios capazes de "vetar", isto é, inibir totalmente o efeito de um grande número de conexões excitatórias de um neurônio.
  - Uma certa quantidade de auto-excitação indireta também é encontrada, isto é, um neurônio excita o seu vizinho, que por sua vez excita o primeiro neurônio novamente.

Topologias Das Redes Neurais Como os Neurônio se Conectam

- Redes Totalmente Conectadas
  - Nesta arquitetura de rede neural cada neurônio está conectado a todos os outros neurônios e estas conexões podem ser tanto excitatórias, irrelevantes (peso zero), como inibitórias.
  - É a mais geral das arquiteturas e todas as outras podem ser encaradas como casos particulares dela onde alguns pesos são setados para zero (0).
  - O peso entre um neurônio e outro pode ser diferente daquele entre o segundo e o primeiro neurônio, formando uma rede totalmente conectada assimétrica.
  - Este tipo de rede assimétrica raramente é utilizada, e além disso, é biologicamente implausível pois os neurônios raramente se conectam a neurônios geograficamente muito distantes.

Topologias Das Redes Neurais Como os Neurônio se Conectam

- Redes Totalmente Conectadas



Topologias Das Redes Neurais Como os Neurônio se Conectam

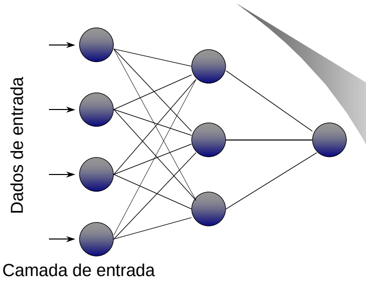
- **Redes em Camadas**
- Existem redes em que os neurônios são particionados em subconjuntos chamados camadas.
- Depende da forma como os Neurônios se conectam para formar uma "Rede" de neurônios.
- Redes Diretas - "Feedforward" –
  - Não existem ciclos.
  - Conexões para a "frente".
- Redes Recorrentes - "Feedback"
  - Existem ciclos.
  - Conexões para trás e/ou para os lados.

Topologia das Redes Neurais

- Redes Neurais Diretas - Feedforward
  - As redes diretas são aquelas cujo grafo não tem ciclos.
  - Freqüentemente é comum representar estas redes em camadas e neste caso são chamadas redes de camadas.

Topologia das Redes Neurais

- Redes Neurais Diretas - Feedforward

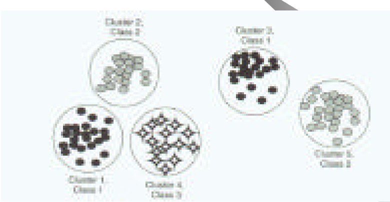


Topologia das Redes Neurais

- Redes Neurais Diretas
  - Neurônios que recebem sinais de excitação do ambiente são chamados de **camada de entrada** ou primeira camada.
  - Neurônios que tem sua saída como saída da rede pertencem a **camada de saída** ou última camada.
  - Neurônios que recebem sinais apenas de outros neurônios e enviam suas saídas para outros neurônios, pertencem a **camadas intermediárias** ou escondidas (hidden).

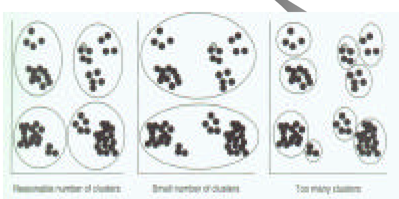
Aplicações das Redes Neurais

- Classificação



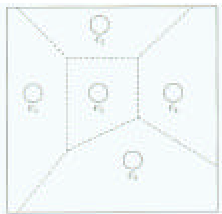
Aplicações das Redes Neurais

- Agrupamento (Clusterização)



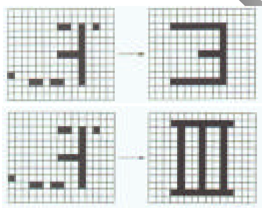
Aplicações das Redes Neurais

– Quantização Vetorial (Compressão)



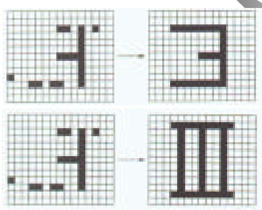
Aplicações das Redes Neurais

– Associação de Padrões



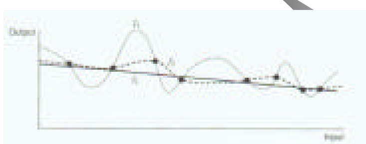
Aplicações das Redes Neurais

– Associação de Padrões



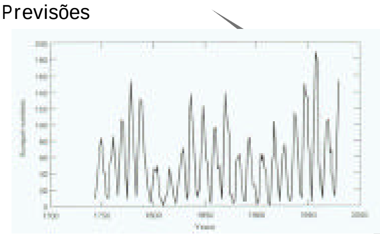
Aplicações das Redes Neurais

– Aproximação de Funções



Aplicações das Redes Neurais

– Previsões



Aplicações das Redes Neurais

– Aplicações em Controle

Aprendizado de Redes Neurais

- Um neurônio é considerado ser um elemento adaptativo. Seus pesos sinápticos são modificáveis dependendo do algoritmo de aprendizado.
- Por exemplo, alguns, dependendo do sinal de entrada que recebem, tem seus valores de saída associados a uma resposta diante de um aprendizado supervisionado por uma espécie de "professor".
- Em alguns casos o sinal do "professor" não está disponível e não há informação de erro que possa ser utilizada para correção dos pesos sinápticos, assim o neurônio modificará seus pesos baseado somente no sinal de entrada e/ou saída, sendo o caso do aprendizado não-supervisionado.

Aprendizado de Redes Neurais

- Aprendizado Supervisionado
  - Neste caso, o "professor" indica explicitamente um comportamento bom ou ruim.
  - Por exemplo, seja o caso de reconhecimento de caracteres e para simplificar seja reconhecer entre um A e um X.
  - Escolhe-se uma rede direta, com dois neurônios na camada de saída, uma ou várias camadas internas e um conjunto de neurônios na camada de entrada capaz de representar com a precisão desejada a letra em questão.
  - Apresentam-se estas letras sucessivamente a uma retina artificial constituída de uma matriz de elementos foto-sensíveis, cada um ligado a um neurônio da rede neural artificial direta.

Aprendizado de Redes Neurais

- Aprendizado Supervisionado
  - Observa-se qual dos dois neurônios de saída está mais excitado. Se for o que se convencionou representar a letra que for apresentada nada deve ser corrigido, caso contrário modifica-se os valores das conexões sinápticas no sentido de fazer a saída se aproximar da desejada.
  - Foi exatamente isto que Rosenblatt fez com o seu Perceptron. Como a cada exemplo apresentado uma correção é introduzida depois de observar a saída da rede, este é um caso de aprendizado supervisionado.

Aprendizado de Redes Neurais

- Aprendizado Não-Supervisionado
  - É aquele que para fazer modificações nos valores das conexões sinápticas não usa as informações sobre a resposta da rede, isto é se a resposta está correta ou não.
  - Usa-se por outro lado um esquema, tal que, para exemplos de coisas semelhantes, a rede responda de modo semelhante.

Aprendizado de Redes Neurais

- O Aprendizado Backpropagation
  - Estrutura da Rede:
    - Rede Direta Multi-camada com Neurônios Estáticos.
  - Modo de Treinamento:
    - Supervisionado.
  - Solução para superar o problema do aprendizado da classificação de padrões não-linearmente separáveis:
    - Utilização de uma camada intermediária de neurônios, chamada Camada Intermediária (ou Escondida - "Hidden Layer"), de modo a poder implementar superfícies de decisão mais complexas.

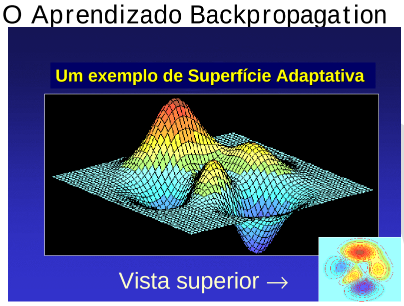
Aprendizado de Redes Neurais

- O Aprendizado Backpropagation
  - Desvantagem em utilizar esta camada escondida:
    - O aprendizado se torna muito mais difícil.

      A característica principal da camada escondida é que seus elementos se organizam de tal forma que cada elemento aprenda a reconhecer características diferentes do espaço de entrada, assim, o algoritmo de treinamento deve decidir que características devem ser extraídas do conjunto de treinamento.
    - Nos anos 80, um algoritmo chamado Retro-propagação ou Backpropagation, veio fazer renascer o interesse geral pelas redes neurais.

Aprendizado de Redes Neurais

- O Aprendizado Backpropagation

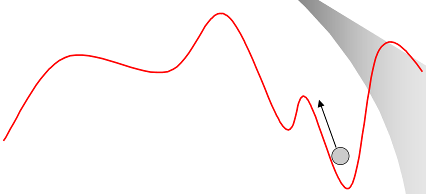


Aprendizado de Redes Neurais

- O Aprendizado Backpropagation
  - Utiliza o mesmo princípio da Regra Delta
    - a minimização de uma função custo, no caso, a soma dos erros médios quadráticos sobre um conjunto de treinamento, utilizando a técnica de busca do gradiente-descendente.
  - Também é chamado muitas vezes de Regra Delta Generalizada ("Generalized Delta-Rule").
    - A modificação principal em relação a Regra Delta foi a utilização de funções contínuas e suaves como função de saída dos neurônios ao invés da função de limiar lógico.
    - Como as funções de saída passaram a ser deriváveis, isto permitiu a utilização da busca do gradiente-descendente também para os elementos das camadas intermediárias.

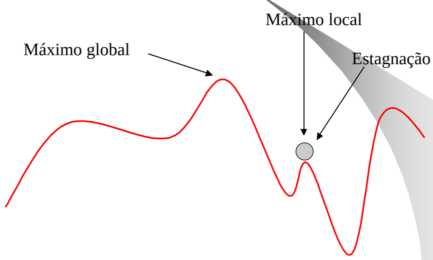
Aprendizado de Redes Neurais

- O Aprendizado Backpropagation



Aprendizado de Redes Neurais

- O Aprendizado Backpropagation



Aprendizado de Redes Neurais

- O Aprendizado Backpropagation
  - O algoritmo Backpropagation é hoje em dia a técnica de aprendizado supervisionado mais utilizada para redes neurais unidirecionais multi-camadas com neurônios estáticos.
  - Basicamente, a rede aprende um conjunto pré-definido de pares de exemplos de entrada/saída em ciclos de propagação/adaptação.
  - Depois que um padrão de entrada foi aplicado como um estímulo aos elementos da primeira camada da rede, ele é propagado por cada uma das outras camadas até que a saída seja gerada. Este padrão de saída é então comparado com a saída desejada e um sinal de erro é calculado para cada elemento de saída.

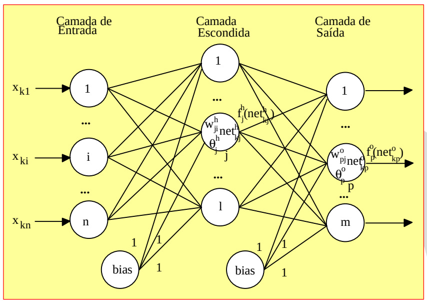
Aprendizado de Redes Neurais

- O Aprendizado Backpropagation
  - O sinal de erro é então retro-propagado da camada de saída para cada elemento da camada intermediária anterior que contribui diretamente para a formação da saída.
  - Cada elemento da camada intermediária recebe apenas uma porção do sinal de erro total, proporcional apenas à contribuição relativa de cada elemento na formação da saída original.

Aprendizado de Redes Neurais

- O Aprendizado Backpropagation
  - Este processo se repete, camada por camada, até que cada elemento da rede receba um sinal de erro que descreva sua contribuição relativa para o erro total.
  - Baseado no sinal de erro recebido, os pesos das conexões são então atualizados para cada elemento de modo a fazer a rede convergir para um estado que permita a codificação de todos os padrões do conjunto de treinamento.

Algoritmo Backpropagation



Algoritmo Backpropagation

- Princípios Básicos
  - Suponhamos que tenhamos um conjunto de P pares de vetores (**X**$_1$,**Y**$_1$), (**X**$_2$,**Y**$_2$), ..., (**X**$_P$, **Y**$_P$), no nosso conjunto de treinamento e que são exemplos de um mapeamento funcional definido como:

$$Y = q(X) \;:\; X \in \Re^n, Y \in \Re^m$$

  - Desejamos treinar a rede de modo que ela consiga aprender uma aproximação da forma:

$$O = Y' = q'(X) \;:\; X \in \Re^n, Y' \in \Re^m$$

Algoritmo Backpropagation

- Etapas
  - 1. Aplicar um vetor de entrada do conjunto de treinamento e propagar até a saída
    - Um vetor de entrada $\mathbf{X}_k=[x_{k1}\ x_{k2}\ \ldots\ x_{kn}]^T$ do conjunto de treinamento é apresentado à camada de entrada da rede. Os elementos de entrada distribuem os valores para os elementos da camada escondida. O valor do net para o j^{ésimo} elemento da camada escondida vale:

$$net_{kj}^h = \sum_{i=1}^{n} w_{ji}^h x_{ki} + q_j^h$$

    - onde w_{ji} é o peso da conexão entre o i^{ésimo} elemento da camada de entrada e o j^{ésimo} elemento da camada escondida *h*.

Algoritmo Backpropagation

- Etapas
  - 1. Aplicar um vetor de entrada do conjunto de treinamento e propagar até a saída
    - Como os neurônios são estáticos, o valor de saída para um neurônio da cada escondida vale:

$$i_{kj} = f_j^h(net_{kj}^h)$$

Algoritmo Backpropagation

- Etapas
  - 1. Aplicar um vetor de entrada do conjunto de treinamento e propagar até a saída
    - Do mesmo modo, as equações para os neurônios da camada de saída são:

$$net_{kp}^o = \sum_{j=1}^{l} w_{pj}^o i_{kj} + q_p^o$$

$$o_{kp} = f_p^o(net_{kp}^o)$$

Algoritmo Backpropagation

- Etapas
  - 2. Calcular o erro entre a saída calculada pela rede e a saída desejada no conjunto de treinamento.
    - Definimos o erro para um único neurônio p na camada de saída para um vetor de entrada k como sendo:

$$E_{kp} = (y_{kp} - o_{kp})$$

    - E o erro a ser minimizado pelo algoritmo para todos os neurônios da camada de saída como:

$$E_k = \frac{1}{2}\sum_{p=1}^{m} E_{kp}^2$$

$$E_k = \frac{1}{2}\sum_{p=1}^{m} (y_{kp} - o_{kp})^2$$

Algoritmo Backpropagation

- Etapas
  - 3. Determinar em que direção a mudança de peso deverá ocorrer.
    - Para determinar a direção da modificação dos pesos, calculamos o negativo do gradiente de E_k,∇E_k, com relação aos pesos w_{pj}
    - Podemos escrever a derivada de f_p^o como:

$$\frac{\partial E_k}{\partial w_{pj}^o} = -(y_{kp} - o_{kp})\frac{\partial f_p^o}{\partial (net_{kp}^o)}\frac{\partial (net_{kp}^o)}{\partial w_{pj}^o}$$

    - e o último termo da equação como:

$$f_p^{o'}(net_{kp}^o)$$

$$\frac{\partial (net_{kp}^o)}{\partial w_{pj}^o} = \left(\frac{\partial}{\partial w_{pj}^o}\sum_{j=1}^{l} w_{pj}^o i_{kj} + q_p^o\right) = i_{kj}$$

Algoritmo Backpropagation

- Etapas
  - 3. Determinar em que direção a mudança de peso deverá ocorrer.
    - Combinando as equações, temos para o negativo do gradiente:

$$-\frac{\partial E_k}{\partial w_{pj}^o} = (y_{kp} - o_{kp})\, f_p^{o'}(net_{kp}^o)\, i_{kj}$$

  - 4. Determinar o valor da mudança de cada peso.
    - A atualização dos pesos dos neurônios da camada de saída se faz por:

$$w_{pj}^o(t+1) = w_{pj}^o(t) + \Delta_k w_{pj}^o(t)$$

$$\Delta_k w_{pj}^o = h\,(y_{kp} - o_{kp})\, f_p^{o'}(net_{kp}^o)\, i_{kj}$$

    - onde η é a TAXA DE APRENDIZADO.

Algoritmo Backpropagation

- Etapas
  - 4. Determinar o valor da mudança de cada peso.
    - AS FUNÇÕES DE SAÍDA -para que possamos implementar a busca do gradiente-descendente, é necessário que f_p^o seja diferenciável.
    - as funções usualmente utilizadas são:
    - a função linear:

$$f_p^o(net_{jp}^o) = net_{jp}^o$$

    - a função logística ou sigmoidal:

$$f_p^o(net_{jp}^o) = \frac{1}{1+e^{-net_{jp}^o}}$$

    - a função tangente hiperbólica:

$$f_p^o(net_{jp}^o) = \tanh(net_{jp}^o) = \frac{1-e^{-net_{jp}^o}}{\;\;{}^{-net_{jp}^o}}$$

Algoritmo Backpropagation

- Etapas
  - 4. Determinar o valor da mudança de cada peso.
    - AS FUNÇÕES DE SAÍDA
      - Estas funções são bastante populares pois as suas derivadas podem ser calculadas de maneira simples, sem a necessidade de cálculos complexos.
      - para a função linear:

$$f_p^{o'} = 1$$

      - para a função logística ou sigmoidal:

$$f_p^{o'} = f_p^o * (1 - f_p^o)$$

      - para a função tangente hiperbólica:

$$f_p^{o'} = 1 * (1 - f_p^{o\,2})$$

Algoritmo Backpropagation

- Etapas
  - 5. Repetir os procedimentos para os pesos da Camada Intermediária.
    - Desejamos repetir para a camada escondida os mesmos tipos de cálculos que realizamos para a camada de saída.
    - O problema aparece quando tentamos determinar uma medida para o erro das saídas dos neurônios da camada escondida.
    - Sabemos qual é a saída destes neurônios calculada pela rede, porém não sabemos a priori qual deveria ser a saída correta para estes elementos. Intuitivamente, o erro total, E_k, deve de alguma forma estar relacionado com o valor de saída dos neurônios da camada escondida.

Algoritmo Backpropagation

- Etapas
  - 5. Repetir os procedimentos para os pesos da Camada Intermediária.
    - Voltando a equação do Erro temos:

$$E_k = \frac{1}{2}\sum_p (y_{kp} - o_{kp})^2$$
$$= \frac{1}{2}\sum_p (y_{kp} - f_p^o(net_{kp}^o))^2$$
$$= \frac{1}{2}\sum_p (y_{kp} - f_p^o(\sum_j w_{pj}^o i_{kj} + \theta_p^o))^2$$

Algoritmo Backpropagation

- Etapas
  - 5. Repetir os procedimentos para os pesos da Camada Intermediária.
    - Sabendo que i_{pj} depende dos pesos da camada escondida, podemos utilizar este fato para calcular o gradiente de E_k com respeito aos pesos da camada escondida.

$$\frac{\partial E_k}{\partial w_{ji}^h} = \frac{1}{2}\sum_p \frac{\partial}{\partial w_{ji}^h}(y_{kp} - o_{kp})^2$$
$$= -\sum_p (y_{kp} - o_{kp}) \frac{\partial o_{kp}}{\partial (net_{kp}^o)} \frac{\partial (net_{kp}^o)}{\partial i_{kj}} \frac{\partial i_{kj}}{\partial (net_{kj}^h)} \frac{\partial (net_{kj}^h)}{\partial w_{ji}^h}$$

Algoritmo Backpropagation

- Etapas
  - 5. Repetir os procedimentos para os pesos da Camada Intermediária.
    - Cada um dos fatores da equação pode ser calculado explicitamente das equações anteriores, assim como foi feito para o gradiente da camada de saída.
    - O resultado fica:

$$\frac{\partial E_k}{\partial w_{ji}^h} = -\sum_p (y_{kp} - o_{kp}) f_p^{o'}(net_{kp}^o) w_{pj}^o f_j^{h'}(net_{kj}^h) x_{ki}$$

Algoritmo Backpropagation

- Etapas
  - 5. Repetir os procedimentos para os pesos da Camada Intermediária.
    - Por fim, assim como no caso da camada de saída, atualizamos os pesos da camada escondida proporcionalmente ao valor negativo da equação.

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \Delta_k w_{ji}^h(t)$$

- onde:

$$\Delta_k w_{ji}^h = \eta f_j^{h'}(net_{kj}^h) x_{ki} \sum_p (y_{kp} - o_{kp}) f_p^{o'}(net_{kp}^o) w_{pj}^o$$

- η novamente é a taxa de aprendizado.

Algoritmo Backpropagation

- Etapas
  - 6. Voltar ao passo 1, escolhendo um novo vetor de entrada do conjunto de treinamento e repetir os passos de 1 a 5, somando o erro.
  - 7. Após todos os vetores do entrada do conjunto de treinamento terem sido apresentados (uma "época"), calcular o erro médio quadrático.

    Se for aceitável parar,

    senão voltar ao passo 1.

Algoritmo Backpropagation

- O que são mínimos locais?
  - São pequenos "buracos" na superfície de erro, mas não são na realidade a "soluçao" (fundo do poço) do problema.
- Como escapar de mínimos locais?
  - Podemos escapar de mínimos locais usando na atualização dos pesos um termo proporcional a última direção de alteração do peso. (Alteração do Peso no passo anterior do algoritmo Backpropagation) - idéia de inércia ou um "empurrão" para sair de buracos.

$$w_{pj}^o(t+1) = w_{pj}^o(t) + \Delta_k w_{pj}^o(t) + \alpha \Delta_k w_{pj}^o(t-1)$$

  - onde α é o parâmetro conhecido como "momento".

Algoritmo Backpropagation

- Algumas dicas práticas
  - Inicialização dos valores dos pesos
    - valores aleatórios entre -1 e +1.
  - Valor da taxa de aprendizado η
    - valores pequenos 0.01 e 0.1
  - Se a função de saída for sigmoidal, escalar os valores de saída.
    - As saídas nunca atingem exatamente 0 ou 1. Usar valores como 0.1 e 0.9 para representar o menor e o maior valor de saída.
  - Se a função de saída for tangente hiperbólica, escalar os valores de saída.
    - As saídas nunca atingem exatamente -1 ou 1. Usar valores como -0.9 e 0.9 para representar o menor e o maior valor de saída.

Algoritmo Backpropagation

- Algumas dicas práticas
  - Valor do parâmetro de momento?
    - Valores "grandes" entre 0.8 e 0.9.
  - O que acontece se usarmos taxas de aprendizado muito grandes?
    - Converge rápido, mas pode não chegar no valor de erro mínimo, pois fica "saltando" de um lado para outro na superfície de erros.
  - Quantos neurônios na camada intermediária?
  - Quantas camadas intermediárias?

Avaliação do Aprendizado

- A capacidade de aprendizado de um ser humano pode ser avaliada dos seguintes modos:
  - Quão bem o aprendiz responde aos dados para os quais o aprendiz já foi treinado, isto é, qual é a diferença entre os resultados esperados e os resultados fornecidos pelo aprendiz?
  - Quão bem o aprendiz responde a novos dados não vistos durante o treinamento?
  - Quais os recursos computacionais (tempo, espaço, esforço) requeridos pelo aprendiz?

Avaliação do Aprendizado

- O desempenho das redes neurais pode ser avaliado usando os mesmos critérios.
  - Qualidade de resultados – O desempenho de uma rede neural é freqüentemente associado a medidas de erro.
    - A medida de erro mais freqüentemente utilizada é a distância euclidiana onde di é o i-ésimo elemento do vetor de saída desejado e oi é o i-ésimo elemento do vetor de saída calculado pela rede.
    - Entre outras formas de medida, , a distância Manhattan ou distância de Hamming, é freqüentemente usada, especialmente em tarefas de associação de padrões. Esta medida de distância dá igual ênfase para todas as dimensões dos dados de entrada.

Avaliação do Aprendizado

- Algumas vezes é mais significativo utilizar medidas de distância ponderadas que associam diferentes graus de importância para diferentes dimensões.
- Supondo um vetor de saída bidimensional cujos componentes são (altura, peso), qual possui a menor distância: (1.5, 60) de (1.5, 58) ou (1.5, 60) de (1.0, 60)?
- Uma melhor medida de distância neste caso seria:

  $$\|x^1 - x^2\| = \sqrt{a^2 (x^{1,1} - x^{1,2})^2 + b^2 (x^{2,1} - x^{2,2})^2}$$

  onde a escolha de a e b reflete a quantidade de variação dos dois componentes. (a=1 e b=0.03 para o exemplo anterior).

Avaliação do Aprendizado

- A natureza do problema muitas vezes direciona a escolha da medida de erro. Em problemas de classificação, além da distância Euclidiana, outra medida de erro possível é a fração de exemplos mal classificados.
- E = Número de casos mal-classificados/Número total de casos

Avaliação do Aprendizado

- **Capacidade de Generalização** – Não basta que um sistema apresente um bom desempenho para responder aos dados que ele viu durante o aprendizado. Uma boa capacidade de generalização também é necessária, isto é, o sistema deve responder adequadamente a dados de teste, distintos dos dados de treinamento.
  - Considere uma criança que é ensinada os resultados das adições 3+4 e 33+44. Dados estes números, a criança dará a resposta correto. No entanto, será que a criança seria capaz de dar a resposta para a adição 333+444?
  - Nas redes neurais, geralmente, os dados disponíveis são separados em dois conjuntos. Um dos conjuntos é utilizado para treinamento e o segundo conjunto para teste.

Avaliação do Aprendizado

- Pode-se observar muitas vezes que o treinamento excessivo no conjunto de treinamento, reduz o desempenho no conjunto de teste, o que é chamado de sobre-treinamento ou "overtraining".
- Uma maneira de evitar que isto ocorra é verificar o desempenho do sistema sobre o conjunto de teste a medida que o treinamento se desenrola. Após um certo número de iterações de aprendizado com o conjunto de treinamento, no qual verifica-se que o desempenho do sistema melhora, testa-se se o desempenho do sistema sobre o conjunto de teste também melhora.
- Caso isto não ocorra depois de algumas ocasiões, considera-se que o sobre-treinamento está ocorrendo e que o processo de treinamento deve ser interrompido.

Avaliação do Aprendizado

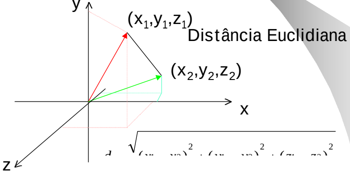
- **Recursos Computacionais** – Uma vez que o treinamento esteja completo, a maioria das redes neurais dispende um tempo mínimo na sua execução ou aplicação a um problema específico. Entretanto, o treinamento de redes neurais pode ser extremamente demorado, requerendo muitas horas ou mesmo dias.
  - O tempo de treinamento aumenta rapidamente com o tamanho da rede e a complexidade do problema.
  - As capacidades de uma rede neural estão relacionadas com o seu tamanho. Além disso, o uso de redes neurais muito grandes aumenta o tempo de treinamento e reduz a capacidade de generalização.

Redes Neurais Artificiais Memórias Associativas

- Conceitos Básicos - Memória Associativa
  - É um conceito "intuitivo".
    - Parece ser uma das funções primárias do cérebro.
    - Facilmente ASSOCIAMOS a face de um amigo com seu nome, ou um nome a um número de telefone.
    - Também serve para reconstituir padrões "corrompidos" ou incompletos.
    - Se olharmos uma foto com os lábios da Natasha Kinsky, logo recompomos todo o seu rosto.
    - Se vemos um amigo que normalmente não usa óculos, com eles, ainda assim, reconhecemos a face como sendo da pessoa em questão.
  - Recuperação de informação pelo "conteúdo"

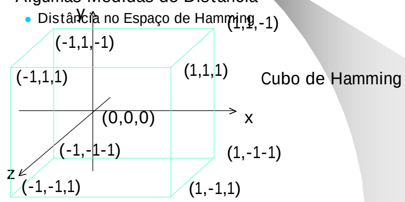
Redes Neurais Artificiais Memórias Associativas

- Definições Iniciais
  - Algumas Medidas de Distância
    - Distância no Espaço Euclidiano



Redes Neurais Artificiais Memórias Associativas

- Definições Iniciais
  - Algumas Medidas de Distância
    - Distância no Espaço de Hamming



Redes Neurais Artificiais
Memórias Associativas

- Definições Iniciais
  - Distância de Hamming X Distância Euclidiana
    - Dados dois pontos (xi,yi) $\in$ {-1,+1}
    - Distância Euclidiana
      - $(x1-x2)^2 \in \{0,4\}$
      - Distância Euclidiana: d = $\sqrt{4}$(# "desencontros")
    - Distância de Hamming
      - Distância de Hamming: h = # "desencontros"

d = $2\sqrt{h}$

Redes Neurais Artificiais
Memórias Associativas

- Memórias Associativas - Definição Formal
  - Associadores Lineares
    - Suponha que tenhamos L pares de vetores, $\{(X_1,Y_1),(X_2,Y_2),\ldots,(X_L,Y_L)\}$, com $X_i \in \Re^n$, e $Y_i \in \Re^m$.
    - Chamamos a estes vetores **exemplares**.
    - O que desejamos com os Associadores Lineares é fazer um "mapeamento" Φ de X para Y.

$$X \xrightarrow{F} Y$$

- Podemos distinguir então 3 tipos de **MEMÓRIAS ASSOCIATIVAS**.

Redes Neurais Artificiais
Memórias Associativas

- Memórias Heteroassociativas
  - Implementa um mapeamento Φ de X para Y tal que $\Phi(X_i) = Y_i$. Se X for o mais próximo (menor distância de Hamming) de X_i do que qualquer X_j, j=1,2,...,L, então $\Phi(X) = Y_i$.
- Memórias Associativas Interpolativas
  - Implementa um mapeamento Φ de X para Y tal que $\Phi(X_i) = Y_i$. Mas se o vetor de entrada X diferir de um exemplar X_i por um vetor D, de forma que $X=X_i+D$, então a saída da memória também difere dos exemplares de saída por um vetor E, ou seja:
    $\Phi(X) = \Phi(X_i+D) = Y_i + E$.

Redes Neurais Artificiais
Memórias Associativas

- Memórias Autoassociativas
  - Assume que $X_i = Y_i$ e implementa um mapeamento Φ de X para X tal que $\Phi(X_i) = X_i$. Se X for o mais próximo (menor distância de Hamming) de X_i do que qualquer X_j, j=1,2,...,L, então $\Phi(X) = X_i$.

Redes Neurais Artificiais
Memórias Associativas

- Implementação Matemática
  - Construir uma memória associativa não é difícil se introduzirmos a restrição de que os vetores exemplares devam ser "ortonormais" entre si (vetores dos vértices de um Cubo de Hamming).
  - $X_i^T.X_j=\delta_{ij}$, onde $\delta_{ij}=1$ se i=j, e $\delta_{ij}=0$ se i$\neq$j.
    $\Phi(X) = (Y_1X_1^T + Y_2X_2^T + \ldots + Y_LX_L^T).X$
  - Exemplo

$\Phi(X_2) = (Y_1X_1^T + Y_2X_2^T + \ldots + Y_LX_L^T).X_2$

$\Phi(X_2) = Y_1X_1^T X_2 + Y_2X_2^T X_2 + \ldots + Y_LX_L^TX_2$

$\Phi(X_2) = Y_1\delta_{12} + Y_2\,\delta_{22} + \ldots + Y_L\,\delta_{L2}$

$\delta ij=0 \quad i\neq j, \quad \delta ij=1 \quad i=j$

$\Phi(X_2) = Y_2$

Redes Neurais Artificiais
Memórias Associativas

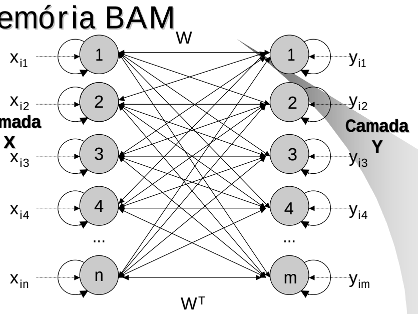
- Implementação por Redes Neurais (Elementos Processadores Distribuídos)

A MEMÓRIA <u>BAM</u>

(Bidirectional Associative Memory)

- Consiste de duas camadas de neurônios que estão completamente conectados entre as camadas. Cada neurônio está conectados a si mesmo.
- O peso das conexões é determinado a-priori, baseado nos vetores de "treinamento".

Redes Neurais Artificiais
Memória BAM



Redes Neurais Artificiais
Memória BAM

- "Treinamento" da BAM
  - A matriz de pesos W é construída utilizando o modelo de Associador Linear.
    - Dados L pares de vetores que constituem o conjunto de exemplares que desejamos armazenar,
      $W = Y_1X_1^T + Y_2X_2^T + Y_3X_3^T + \ldots + Y_LX_L^T$
      fornece os pesos das conexões da camada X para a camada Y.
      W^T
      fornece os pesos das conexões da camada Y para a camada X.

Redes Neurais Artificiais
Memória BAM

- Processamento da BAM
  - Cálculo das Matrizes de Pesos
  - Recuperação da Informação
    1. Aplicar um par de vetores iniciais (X_0,Y_0).
    2. Propagar a informação de X para Y (multiplicar X_0 pela matriz W) e atualizar Y.
    3. Propagar a informação atualizada de Y para X (multiplicar Y pela matriz W^T) e atualizar X.
    4. Repetir os passos 2 e 3 até não haver mudança nos valores dos neurônios.
  - Dado X_0 o resultado será X_i com a menor distância de Hamming de X_0.

Redes Neurais Artificiais
Memória BAM

- Algumas Considerações
  - É este algoritmo que fornece à BAM suas características bi-direcionais.
  - Os termos "entrada" e "saída" dependem da direção atual de propagação.
  - Após algumas iterações a rede irá estabilizar em um estado estável.
  - Não sobrecarregar demais a memória com muita informação, ou ela irá estabilizar em um "estado espúrio" (crosstalk).
  - O Crosstalk ocorre quando os padrões exemplares estão muito "próximos" uns dos outros.

Redes Neurais Artificiais
Memória BAM

- Matemática da BAM
  - Os neurônios calculam o net, como neurônios "normais":

cálculo do net para o neurônio i da camada Y.

$$net_i^Y = \sum_{j=1}^{n} w_{ij} x_j$$

cálculo do net para a camada Y.

$$net^Y = W.X$$

Redes Neurais Artificiais
Memória BAM

- Matemática da BAM
  - Os neurônios calculam o net, como neurônios "normais":

cálculo do net para o neurônio j da camada X.

$$net_j^X = \sum_{i=1}^{m} w_{ji} y_i$$

cálculo do net para a camada X.

$$net^X = W^T.Y$$

Redes Neurais Artificiais
Memória BAM

- Matemática da BAM
  - O valor da saída para cada neurônio depende do valor do net e do valor da saída "anterior".

Novo valor para a saída y no instante $t+1$
está relacionada com o valor de y no instante t por

$$y_i(t+1) = \begin{cases} +1 & \text{se} \quad net_i^y > 0 \\ y_i(t) & \text{se} \quad net_i^y = 0 \\ -1 & \text{se} \quad net_i^y < 0 \end{cases}$$

Novo valor para a saída x no instante $t+1$
está relacionada com o valor de x no instante t por

$$x_i(t+1) = \begin{cases} +1 & \text{se} \quad net_i^x > 0 \\ x_i(t) & \text{se} \quad net_i^x = 0 \\ -1 & \text{se} \quad net_i^x < 0 \end{cases}$$

Redes Neurais Artificiais
Memória BAM

- Alguns exemplos

$$X_1 = [1,-1,-1,1,-1,1,1,-1,-1,1]^T \text{ e } Y_1 = [1,-1,-1,-1,-1,1]^T$$

$$X_2 = [1,1,1,-1,-1,-1,1,1,-1,-1]^T \text{ e } Y_2 = [1,1,1,1,-1,-1]^T$$

$$W = Y_1 X_1^T + Y_2 X_2^T$$

$$W = \begin{bmatrix} 2 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & -2 & 0 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ -2 & 0 & 0 & 0 & 2 & 0 & -2 & 0 & 2 & 0 \\ 0 & -2 & -2 & 2 & 0 & 2 & 0 & -2 & 0 & 2 \end{bmatrix}$$

Redes Neurais Artificiais
Memória BAM

- Exemplo - continuação
  - Tomemos X_0=[-1,-1,-1,1,-1,1,1,-1,-1,1]^T
  - h(X_0,X_1)=1 e h(X_0,X_2)=7 (distância de Hamming)
  - Fazemos Y_0 igual a um dos vetores exemplares (ou usamos um vetor bipolar aleatório) Y_0=Y_2=[1,1,1,1,-1,-1]^T.
  - Propagamos de X para Y, já que a "chave" é X_0.
  - net^Y=W.X_0=[4,-12,-12,-12,4,12]^T
  - Calculamos o novo vetor Y, Y_{new}=[1,-1,-1,-1,1,1]^T
  - Propagamos agora de Y para X.
  - net^X=[4,-8,-8,8,-4,8,4,-8,-4,8]^T
  - Calculamos o novo vetor X, X_{new}=[1,-1,-1,1,-1,1,1,-1,-1,1]^T
  - Novas propagações não alterarão o resultado e portanto consideramos a memória estabilizada e com o padrão X_1 recuperado.

Redes Neurais Artificiais
Memória BAM

- Exemplo 2
  - Tomemos X_0=[-1,1,1,-1,1,1,1,-1,1,-1]^T e Y_0=[-1,1,-1,1,-1,-1]^T
  - h(X_0,X_1)=7 e h(X_0,X_2)=5 (distância de Hamming)
  - h(Y_0,Y_1)=4 e h(Y_0,Y_2)=2
  - Propagamos de X para Y, já que a "chave" é X_0.
  - net^Y=W.X_0=[-4,4,4,,4,-4]^T
  - Calculamos o novo vetor Y, Y_{new}=[-1,1,1,1,1,-1]^T
  - Propagamos agora de Y para X.
  - net^X=[-4,8,8,-8,4,-8,-4,8,4,-8]^T
  - Calculamos o novo vetor X, X_{new}=[-1,1,1,-1,1,-1,-1,1,1,-1]^T
  - Novas propagações não alterarão o resultado e portanto consideramos a memória estabilizada.
  - **Qual foi o padrão recuperado?**

Redes Neurais Artificiais
Memória BAM

- Exemplo 2 - continuação
  - O padrão recuperado foi o "complemento" de X_1.
  - Propriedade básica da BAM: Se armazenamos um padrão (X,Y), automaticamente armazenamos o complemento do padrão.
  - Não é comum uma criança dizer "apaga" a luz quando na verdade quer acender a luz?

Redes Neurais Artificiais
Memórias Associativas
A Memória de Hopfield

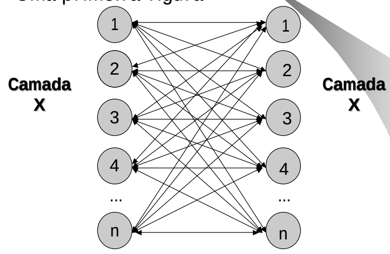
- Conceitos Básicos
  - Quem é John Hopfield?
    - É um professor de biologia e química do CalTech - California Institute of Technology.
  - O que ele fez?
    - Em 1982 Hopfield publicou um artigo que influenciou vários pesquisadores, chamando a atenção para as propriedades associativas de um classe de Redes Neurais.
    - A análise é baseada na definição de "energia" da rede e uma prova de que a rede opera minimizando esta energia quando evolui para padrões estáveis de operação.

Redes Neurais Artificiais
Memória de Hopfield

- Definições Iniciais
  - É uma Memória Autoassociativa.
  - Suas entradas são valores binários {0,1}.
  - Possui uma natureza de operação assíncrona.
    - Isto é a cada instante de tempo, cada neurônio tem se estado de ativação "avaliado" de maneira independente dos outros neurônios.

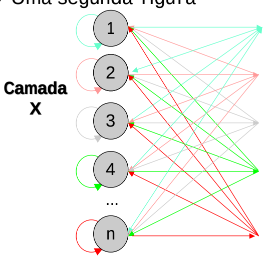
Redes Neurais Artificiais
Memória de Hopfield

- Uma primeira figura



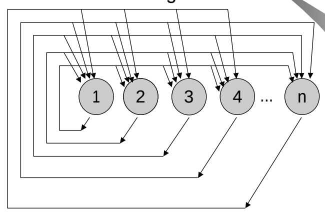
Redes Neurais Artificiais
Memória de Hopfield

- Uma segunda figura



Redes Neurais Artificiais
Memória de Hopfield

- Uma terceira figura



Redes Neurais Artificiais
Memória de Hopfield

- Uma quarta figura e um exemplo
  - Cálculo do net para o neurônio *i*

$$net_i = \sum_{j \neq i}^{n} w_{ij} x_j$$

  - O valor da saída para cada neurônio depende do valor do net e do valor da saída "anterior".
    - Cálculo do valor de saída x_i para o neurônio i no instante de tempo t+1.

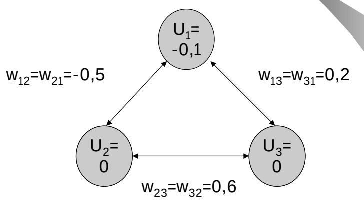
$$x_i(t+1) = \begin{cases} +1 & \text{se } net_i^x > U_i \\ x_i(t) & \text{se } net_i^x = U_i \\ 1 & \text{se } net_i^x < U_i \end{cases}$$

Redes Neurais Artificiais
Memória de Hopfield

- Uma quarta figura e um exemplo
  - O valor de saída de um neurônio é x_i=0 se ele não está disparando e x_i=1 se ele está disparando.
  - O neurônio i recebe entradas de um neurônio j com uma "força" definida como w_{ij}.
  - Como as conexões são bidirecionais, $w_{ij}=w_{ji}$.
  - U_i é o valor de limiar (threshold) acima do qual o neurônio dispara (saída x_i=1).

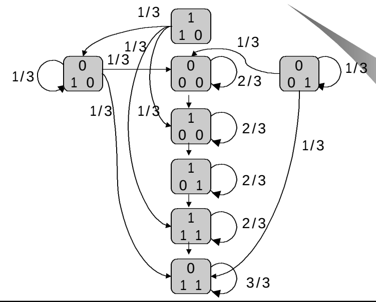
Redes Neurais Artificiais
Memória de Hopfield

- Uma quarta figura e um exemplo



Memória de Hopfield

- Uma quarta figura e um exemplo



Memória de Hopfield

- Uma quarta figura e um exemplo
  - Dado um estado qualquer, digamos x_1,x_2,x_3=000.
  - Podemos calcular o que vai acontecer quando "avaliarmos" cada neurônio.
    - Se o neurônio 1 for o primeiro a ser avaliado:
      $x_1(t+1)=0.(-0,5)+0.(0,2)=0 > -0,1$ (U_1)
      $x_1(t+1)=1$

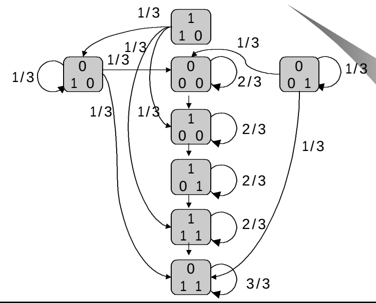
    Logo o novo estado é x_1,x_2,x_3=100.
    - Se o neurônio 2 for o primeiro a ser avaliado:
      $x_2(t+1)=0.(-0,5)+0.(062)=0 < 0$ (U_2)
      $x_2(t+1)=0$

    Logo o novo estado é x_1,x_2,x_3=000.
    - Se o neurônio 3 for o primeiro a ser avaliado:
      $x_3(t+1)=0.(0,2)+0.(0,6)=0 < 0$ (U_3)
      $x_3(t+1)=0$

    Logo o novo estado é x_1,x_2,x_3=000.

Memória de Hopfield

- Uma quarta figura e um exemplo



Redes Neurais Artificiais
Memória de Hopfield

- A Equação de Energia da Memória de Hopfield
  - A maneira como os estados estão organizados no slide anterior não é acidental.
  - Eles estão arranjados de tal maneira que uma mudança de estado ou permanece namesma "altura" ou vai para "baixo".
  - Cada estado está associado a um valor de energia que tende a diminuir cada vez que um neurônio altera seu estado.

Redes Neurais Artificiais
Memória de Hopfield

- A Equação de Energia da Memória de Hopfield

$$E = -\frac{1}{2}\sum_i \sum_{j \neq i} x_i w_{ij} x_j + \sum_i U_i x_i$$

Redes Neurais Artificiais Memória de Hopfield

- A Equação de Energia da Memória de Hopfield
  - Voltando ao Exemplo
    - Para o estado $x_1,x_2,x_3=111$

    $E=-1/2(w_{12}.x_1.x_2+w_{13}x_1x_3+w_{23}x_2x_3)+U_1.x_1+U_2.x_2+U_3.x_3$

    $E=-1/2((-0,5).1.1+0,2.1.1+0,6.1.1)+(-0,1).1+0.1+0.1$

    $E=-1/2(-0,5+0,2+0,6)+(-0,1)$

    $E=-1/2(0,3)-0,1$

    $E=-0,25$

Redes Neurais Artificiais Memória de Hopfield

- A Equação de Energia da Memória de Hopfield
  - Voltando ao Exemplo - continuação
    - Para o estado $x_1,x_2,x_3=011$

    $E=-1/2(w_{12}.x_1.x_2+w_{13}x_1x_3+w_{23}x_2x_3)+U_1.x_1+U_2.x_2+U_3.x_3$

    $E=-1/2((-0,5).0.1+0,2.0.1+0,6.1.1)+(-0,1).0+0.1+0.1$

    $E=-1/2(0,6)$

    $E=-0,3$

Redes Neurais Artificiais Memória de Hopfield

- Como criar Estados Estáveis na Memória de Hopfield
  - O comportamento da rede pode ser útil se os estados estáveis puderem ser selecionados ou criados pelo usuário.
  - Estados Estáveis podem ser criados
    - Através do cálculo prévio dos valores dos pesos das conexão - como na BAM.
    - Através de treinamento.

Redes Neurais Artificiais Memória de Hopfield

- Criando estados estáveis através do cálculo prévio das conexões.
- Exemplo
  - Suponha que tenhamos uma rede de Hopfield com três neurônios e desejamos dois estados estáveis: x1,x2,x3=010 e 111
    - chamemos X1,x2,x3=010 de estado padrão A,
    - chamemos X1,x2,x3=111 de estado padrão B.

Redes Neurais Artificiais Memória de Hopfield

- Criando estados estáveis através do cálculo prévio das conexões.
- Exemplo
  - para o estado padrão A
    - $x_1=0$.
    - Logo, $w_{12}x_2+w_{13}x_3-U_1<0$.
    - Mas $x_2=1$ e $x_3=0$.
    - Assim, $w_{12}-U_1<0$.
    - Do mesmo modo para x_2, $U_2<0$.
    - E, para x_3, $w_{23}-U_3<0$

Redes Neurais Artificiais Memória de Hopfield

- Criando estados estáveis através do cálculo prévio das conexões.
- Exemplo
  - para o estado padrão B
    - $x_1=1$.
    - Logo, $w_{12}x_2+w_{13}x_3-U_1>0$.
    - Mas $x_2=1$ e $x_3=1$.
    - Assim, $w_{12}+w_{13}-U_1>0$.
    - Do mesmo modo para x_2, $w_{12}+w_{23}-U_2>0$.
    - E, para x_3, $w_{23}+w_{13}-U_3>0$

Redes Neurais Artificiais
Memória de Hopfield

- Criando estados estáveis através do cálculo prévio das conexões.
- Exemplo
  - o conjunto de inequações agora pode ser resolvido.
    - Arbitremos para w_{12} o valor 0,5.
    - Então para satisfazer a primeira inequação $0,5<U_1<1$. Arbitremos o valor 0,7.
    - Da quarta inequação $0,2<w_{13}<1$. Arbitremos o valor 0,4.
    - A segunda inequação requer $U_2<0$. Arbitremos 0,2.
    - E assim por diante.

Redes Neurais Artificiais
Memória de Hopfield

- Criando estados estáveis através do cálculo prévio das conexões.
- Exemplo
  - Finalmente
    - w_{12} = 0,5.
    - w_{13} = 0,4.
    - w_{23} = 0,1.
    - U_1=0,7.
    - U_2=-0,2.
    - U_3=0,4.
  - Podemos então construir o diagrama de transição de estados.

Redes Neurais Artificiais
Memória de Hopfield