Redes Neurais

Prof. Dr.Mauro Roisenberg
e-mail: mauro@inf.ufsc.br

Introdução ao Estudo das Redes Neurais Artificiais

- Objetivos
  - Oferecer ao aluno uma introdução à abordagem da IA conexionista, descrevendo características de funcionamento, formas de aprendizado e aplicações típicas. Vários modelos de redes serão estudados, seguindo-se o uso de softwares de simulação.

Introdução ao Estudo das Redes Neurais Artificiais

- Súmula
  - Introdução - Inteligência Artificial Simbólica e Conexionista.
  - Histórico das Redes Neurais Artificiais.
  - Nomenclatura Básica, o Neurônio Biológico e Revisão de Álgebra Linear.
  - Tipos de Redes, Arquiteturas, Características e Aplicações.
  - Modelos Básicos: Perceptron, Adaline, Madaline.

Introdução ao Estudo das Redes Neurais Artificiais

- Súmula - continuação
  - 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.

Introdução ao Estudo das Redes Neurais Artificiais

- Bibliografia

  - Freeman, James A. & Skapura, David M.
    Neural Networks: Algorithms, Applications and Programming Techniques. Addison-Wesley Publishing, 1992.
  - Haykin, Simon
    Neural Networks: a comprehensive foundation. IEEE Press, 1994.
  - Barreto, Jorge M.
    - Inteligência Artificial: No limiar do século XXI. ppp Edições, 1997.

Introdução ao Estudo das Redes Neurais Artificiais

- Bibliografia

  - 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.

Origens da Inteligência Artificial

- O que é Inteligência Artificial?
  - “I propose to consider the question, ‘Can machines think?’ This should begin with definitions of the meaning of the terms ‘machine’ and ‘think’.” A. Turing, *Computing Machinery and Intelligence, 1950*
  - “Se queres discutir comigo, define primeiro teus termos.” *Descartes*

O que é Inteligência Artificial?

- O QUE É INTELIGÊNCIA?
  - *Binet*: “Inteligência é julgar bem, compreender bem, raciocinar bem”.
  - *Tearman*: “ A capacidade de conceituar e de compreender o seu significado”.
  - *Helm*: “A atividade inteligente consiste na compreensão do essencial de uma situação e numa resposta reflexa apropriada”.
  - *Piaget*: “Adaptação ao ambiente físico e social”.

O que é Inteligência Artificial?

- A INTELIGÊNCIA É SÓ HUMANA?
  - Em um primeiro momento, a inteligência era geralmente associada a uma característica unicamente humana, de representação de conhecimentos e resolução de problemas, refletindo um ponto de vista altamente antropocêntrico. Mas, ainda assim, nós, humanos, não compreendemos a nós mesmos, como funciona nossa “inteligência” e nem mesmo a origem de nossos pensamentos.

O que é Inteligência Artificial?

- Hoje em dia, para muitos pesquisadores, a idéia de inteligência passou a ser associada com a idéia de sobrevivência.
- *Carne*: “Talvez a característica básica de um organismo inteligente seja sua capacidade de aprender a realizar várias funções em um ambiente dinâmico, tais como sobreviver e prosperar”.
- *Fogel*: “inteligência pode ser definida como a capacidade de um sistema de adaptar seu comportamento para atingir seus objetivos em uma variedade de ambientes”.

O que é Inteligência Artificial?

- Definições de IA
  - É um ramo da ciência da computação ao mesmo tempo recente (oficialmente nasceu em 1956) e muito antigo (lógica de Aristóteles)
  - Até mesmo a origem do termo é cercada de mistério - John McCarthy, criador do termo em 1956 não tem certeza de não haver ouvido o termo anteriormente.

O que é Inteligência Artificial?

- *Elaine Rich*: “IA é o estudo de como fazer os computadores realizarem coisas que, hoje em dia são feitas melhores pelas pessoas”.
- *Winston*: IA é o estudo das idéias que permitem aos computadores serem inteligentes”.
- *Charniak and McDermott*: IA é o estudo das faculdades mentais através da utilização de modelos computacionais”.
- *Bellman*: “IA é o estudo e simulação de atividades que normalmente assumimos que requerem inteligência”.
- *Russell and Norvig*: “IA é o estudo e implementação de agentes racionais”.

(um agente racional é algo que procura atingir seus objetivos através de suas crenças.)

O que é Inteligência Artificial?

- **O objetivo central da IA é simultaneamente teórico - a criação de teorias e modelos para a capacidade cognitiva - e prático - a implementação de sistemas computacionais baseados nestes modelos.**

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

As duas abordagens da IA

- IA Simbólica
  - Um sistema simbólico é capaz de manifestar um comportamento inteligente.
  - O comportamento inteligente global é simulado sem considerar os mecanismos responsáveis por este comportamento.

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

Princípios da IA Simbólica

- **A estratégia fundamental que sustentou boa parte do sucesso inicial da IA Simbólica, se deve à proposta conhecida como "Physical Symbol Systems Hypothesis", de Newell e Simon.**
- **Physical Symbol Systems - Newell & Simon(1976)**

"A physical symbol system consists of a set of entities, called symbols, which are physical patterns that can occur as components of another type of entity called an expression (or symbol structure)...the system also includes a collection of processes that operate on expressions to produce other expressions: processes of creation, modification, reproduction and destruction. A physical symbol system is a machine that produces through time an evolving collection of symbol structures.Such a system exists in a world of objects wider than just these symbolic expressions themselves "

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

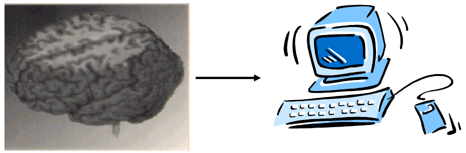
As duas abordagens da IA

- IA Conexionista
  - Se for construído um modelo suficientemente preciso do cérebro, este modelo apresentará um comportamento inteligente. Se apenas uma pequena parte do cérebro for reproduzida, a função exercida por esta parte emergirá do modelo.

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

IA Conexionista

- As origens das redes neurais artificiais remontam no desejo de construir artefatos capazes de exibir comportamento inteligente.



Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

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.**

Computação Baeada em Instruções

- Arquiteturas Von-Neuman, Máquinas Hiper-Cúbicas, Máquinas Sistólicas, Data-flow.
- Se baseiam na execução de instruções para realização do processamento desejado.
- Adotam uma abordagem algorítimica para a solução de problemas.
- CBI - "Computadores Baseados em Instruções".
- **A abordagem algorítmica para solução de problemas pode ser extremamente eficiente desde que se conheça exatamente a seqüência de instruções a serem executadas para resolução do problema.**

Computação "Neural"

- Existem uma série de problemas que os seres vivos, e os seres humanos em particular, parecem resolver de maneira inata.
- O processamento de imagens, o reconhecimento da fala, a recuperação de informações de maneira associativa, a filtragem adaptativa de sinais, o aprendizado de novos fatos e idéias, etc.
- **Se o cérebro dos seres vivos parece ser adequado para resolver os problemas não algorítmicos, deve se buscar uma abordagem que procure se inspirar no funcionamento do cérebro para solução dos problemas.**

IA Simbólica X IA Conexionista

- Conhecimento representado por regras (ou outra estrutura similar) que podem ser facilmente tratadas e analisadas.
- Permite a explicação do processo que levou a uma determinada resposta.
- Fácil inserção de novos conhecimentos obtidos a partir do especialista ou através de métodos automáticos de aquisição de conhecimento.

IA Simbólica X IA Conexionista

- Necessidade de se trabalhar com conhecimentos completos e exatos sobre um determinado problema.
- Dificuldade de explicar todos os conhecimentos relativos ao problema através de regras simbólicas.
- Dificuldade para tratar informações imprecisas ou aproximadas, e valores numéricos (dados quantitativos).

- Exemplo: regular a temperatura da água do banho.

IA Simbólica X IA Conexionista

- Outro Exemplo:

Conhecimento Teórico

```
– AND(A,B) = if A=0
             then AND=0
             else if B=0
                     then AND=0
                     else AND=1
```

Conhecimento Empírico

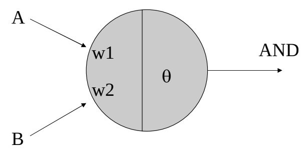
A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

IA Simbólica X IA Conexionista

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

w1=1
w2=1
θ =2

$A.w1+B.w2 \geq \theta$



Interesses em usar Redes Neurais

- Psicólogos
  - Estão vendo possibilidades de construir redes neurais artificiais e ver aparecer comportamentos emergentes tais como o aprendizado de novos conceitos, ajudando a compreensão dos mecanismos do aprendizado.
- Neurofisiologistas
  - Estão interessados em ver as RNAs como metáfora cerebral e, por simulação, melhorar o conhecimento dos mecanismos cerebrais. (estudar a capacidade de memorização, por exemplo).

Interesses em usar Redes Neurais

- Cientistas Cognitivos
  - Se empenham em usar as redes neurais artificiais para um melhor conhecimento dos mecanismos envolvidos no processo cognitivo (qual o melhor método de aprendizado?).
- Engenheiros
  - Olham as RNAs como um caminho para, implementando estas redes em circuitos elétricos, ter computadores realmente paralelos e distribuídos.
  - Muitos encontraram no aprendizado de RNAs um campo para aplicar o que se conhece da teoria da otimização.

Interesses em usar Redes Neurais

- Cientistas de Computação
  - Encontraram um novo paradigma de programação e uma arquitetura distribuída. Explorar este paradigma, analisando suas capacidades e complexidades computacionais, desenvolvendo técnicas de programação, aplicações, etc. é um DESAFIO.

Novo Paradigma de Programação

- Equivalência das RNAs com Máquinas de Turing.

(uma RNA sem periféricos seria um computador com CPU e memória, sem entradas e saídas)

  - Toda RNA pode ser simulada em um CBI.
  - Como toda RNA pode implementar circuitos AND, OR e NOT, é possível construir um CBI com ajuda de uma RNA.

Novo Paradigma de Programação

- Paradigmas de Programação:
  - Imperativa (PASCAL, C, BASIC).
  - Declarativa (PROLOG).
  - Funcional (LISP).
  - Conexionista.
    - Programação por exemplos,
    - Capacidade de generalização,
    - Usa analogia com problemas anteriormente resolvidos,
    - Não necessita de algoritmo explícito,
    - Não necessita de descrição do problema,
    - Baseada na adaptabilidade!

Fundamentos Biológicos

- "Eventually a science of the nervous system based upon direct observation rather than inference will describe the neural states and events which imediately precede instances of behavior. We will know the precise neurological conditions which imediately precede, say, the response: "No, thank you."

B. F. Skinner, Science and human behavior, 1953, pg. 28

Fundamentos Biológicos

- A razão da importância da plausibilidade biológica
  - Redes Neurais Artificiais se inspiram nas redes biológicas e formam um dos paradigmas da Inteligência Artificial - O Conexionismo.
  - Procura imitar a arquitetura do cérebro e espera ver o comportamento inteligente emergir.

Fundamentos Biológicos

- A razão da importância da plausibilidade biológica
  - Em muitos casos, quando o que se deseja é obter um sistema com alto grau de adaptabilidade, uma inspiração biológica tênue pode ser suficiente. Exemplos: reconhecimento de caracteres, previsões financeiras, classificação de padrões, controle de processos.
  - Entretanto, se o objetivo é modelar um processo de ação motora ou processo cognitivo, plausibilidade biológica se torna essencial.

Fundamentos Biológicos

- A Célula
  - A célula é composta basicamente de água, eletrólitos, proteínas, lipídios e carbohidratos. Sua configuração típica é dividida em duas partes: núcleo e citoplasma.
    - Núcleo: controla as reações químicas e a reprodução.
    - Citoplasma: é onde as organelas estão dispersas.
  - Dividindo o meio intracelular do meio extracelular existe a membrana citoplasmática.

Fundamentos Biológicos

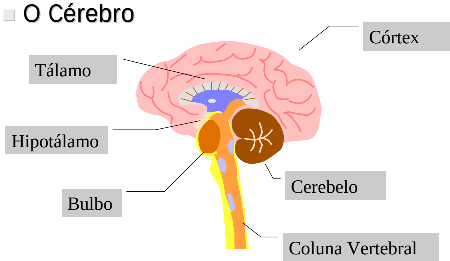
- O Sistema Nervoso
  - Controla as reações rápidas do corpo, como uma contração muscular (função motora), geralmente como resposta a algum estímulo recebido (função sensora).
  - Recebe informações dos sensores, combina estas informações com as informações armazenadas para produzir uma resposta.
  - É organizado hierarquicamente.
    - Medula da coluna vertebral: ato reflexo.
    - Cérebro: Bulbo, Mesencéfalo, Hipotálamo, Tálamo, Cerebelo. Córtex.

Fundamentos Biológicos

- O Cérebro
  - Composto por Neurônios
  (${10}^5$ na mosca da fruta, 5×10^6 no rato e 10^{11} no homem)
- O CérebroHumano
  - Massa: 1-2kg no adulto - 2% do peso
    20% do peso do recém-nascido.
  - Usa 20% do oxigênio, 25% da glucose, 15% do fluxo de sangue.
- O Córtex
  - O tamanho do córtex separa os humanos das outras espécies.
  (5cm^2 no rato, 500 cm^2 no chimpanzé e 2000cm^2 no homem)
  - 3×10^{10} neurônios no córtex humano.
  - 10^3 a 10^4 sinapses por neurônio.

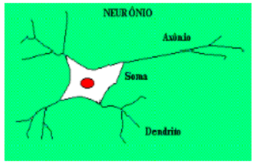
Fundamentos Biológicos

- O Cérebro



Fundamentos Biológicos

- Esquema Simplificado de um Neurônio



Fundamentos Biológicos

- O Neurônio
  - Possui um corpo celular, axônio e diversas ramificações (dendritos).
  - Os dendritos são “dispositivos de entrada” que conduzem sinais das extremidades para o corpo celular.
  - O axônio (geralmente 1) é o “dispositivo de saída” que transmite um sinal do corpo celular para a suas extremidades.
  - As extremidades do axônio são conectadas com dendritos de outros neurônios pelas sinapses.

Fundamentos Biológicos

- O Funcionamento do Neurônio
  - A membrana citoplasmática de uma célula nervosa permite o transporte de eletrólitos que modificam o potencial elétrico entre as partes externas e internas da célula.
  - Esta diferença de potencial provoca um trem de pulsos de freqüência gerado pela célula nervosa (neurônio) através do axônio para os dendritos, que se ligam a outras células.
  - Ante um estímulo de amplitude e duração definida, é codificada a informação que posteriormente é decodificada por dendritos.

Fundamentos Biológicos

- O Funcionamento do Neurônio - A Sinapse
  - Sinapse: é a ligação entre a terminação axônica e os dendritos e que permite a propagação dos impulsos nervosos de uma célula a outra. As sinapses podem ser excitatórias ou inibitórias.
  - As sinapses excitatórias cujos neuro-excitadores são os íons sódio permitem a passagem da informação entre os neurônios e as sinapses inibitórias, cujos neuro-bloqueadores são os íons potássio, bloqueiam a atividade da célula, impedindo ou dificultando a passagem da informação.

Fundamentos Biológicos

- O Funcionamento do Neurônio - O Potencial de Ação
  - Concentrações diferentes de Na^+ e K^- dentro e fora das células provocam diferença de potencial.
  - Estimulação elétrica, química, calor, etc. pode perturbar a membrana do neurônio alterando este potencial.
  - Após um certo tempo as coisas voltam ao normal devido ao mecanismo de transporte ativo. No entanto a onda de variação de tensão se propaga.
  - Na região junto à sinapse o potencial de ação libera neurotransmissores, provocando uma perturbação na membrana do neurônio seguinte, e o fenômeno continua.

Uma Breve Revisão de Álgebra Linear

- Definições Iniciais
  - Estrutura Matricial
    - Matrizes são geralmente representadas como arranjos retangulares de números escalares.
    - A matriz A, m x n, possui m linhas e n colunas. A notação A_{ij} é utilizada para referenciar o número da i-ésima linha, j-ésima coluna de A.

$$A=\begin{bmatrix} 4 & 2 & -5 \\ 1 & 0 & -8 \end{bmatrix} \qquad A_{13}=-5$$

Uma Breve Revisão de Álgebra Linear

- Definições Iniciais
  - Uma matriz n x 1 é geralmente chamada de VETOR.
  - Uma matriz com o mesmo número de linhas e colunas é chamada de MATRIZ QUADRADA.
  - Uma matriz é NULA se possui todos os elementos igual a 0.
  - Uma matriz é DIAGONAL se for quadrada e possuir os elementos da diagonal principal diferentes de 0 e os restantes iguais a 0.
  - Uma matriz diagonal é chamada ESCALAR se todos os elementos da diagonal forem iguais.
  - Uma matriz é chamada IDENTIDADE ou UNIDADE se for escalar com elementos da diagonal igual a 1.

Uma Breve Revisão de Álgebra Linear

- Definições Iniciais

$V=\begin{bmatrix}4\\1\end{bmatrix}$ $E=\begin{bmatrix}5&0\\0&5\end{bmatrix}$

$A=\begin{bmatrix}4&-5\\1&-8\end{bmatrix}$ $I=\begin{bmatrix}1&0\\0&1\end{bmatrix}$

$Z=\begin{bmatrix}0&0&0&0\\0&0&0&0\end{bmatrix}$

$D=\begin{bmatrix}4&0\\0&-8\end{bmatrix}$

Uma Breve Revisão de Álgebra Linear

- Definições Iniciais
  - A TRANSPOSTA de uma matriz A m x n é denotada por A^T. A^T é uma matriz n x m cujas elementos são:

$$(A^T)_{ij} = A_{ji}$$

  - A transposta de um vetor coluna é chamada de VETOR LINHA.

$A=\begin{bmatrix}2&3&4\\1&5&6\end{bmatrix}$ $A^T=\begin{bmatrix}2&1\\3&5\\4&6\end{bmatrix}$

Uma Breve Revisão de Álgebra Linear

- Definições Iniciais
  - Soma de Matrizes
    - Somar duas matrizes A e B resulta em uma matriz cujos elementos são a soma dos correspondentes elementos de A e B: Se C = A + B, então $C_{ij} = A_{ij} + B_{ij}$

$A=\begin{bmatrix}2&3&4\\1&5&6\end{bmatrix}$ $B=\begin{bmatrix}4&-3&0\\2&1&5\end{bmatrix}$

$C=A+B=\begin{bmatrix}6&0&4\\3&6&11\end{bmatrix}$

Uma Breve Revisão de Álgebra Linear

- Definições Iniciais
  - Multiplicação de Matrizes
    - Multiplicar uma matriz A m x n por uma matriz B n x p resulta em uma matriz C m x p cujos elementos são C = AB, então

$$C_{ik} = \sum_{j=1}^{n} A_{ij} B_{jk}$$

$A=\begin{bmatrix}2&3\\4&1\\5&2\end{bmatrix}$ $B=\begin{bmatrix}4&0&1&-2\\1&2&3&5\end{bmatrix}$ $C=\begin{bmatrix}11&6&11&11\\17&2&7&-3\\22&4&11&0\end{bmatrix}$

$C=AB=\begin{bmatrix}2.4+3.1&2.0+3.2&2.1+3.3&2.-1+3.5\\4.4+1.1&4.0+1.2&4.1+1.3&4.-2+1.5\\5.4+2.1&5.0+2.2&5.1+2.3&5.-2+2.5\end{bmatrix}$

Uma Breve Revisão de Álgebra Linear

- Definições Iniciais
  - Multiplicação de Matrizes
    - Dada uma matriz quadrada A n x n, verifica-se a seguinte propriedade: A.I = I.A = A
    - A matriz quadrada A n x n é chamada INVERSÍVEL se existir uma matriz denotada por A^{-1} que satisfaz: $A.A^{-1} = A^{-1}.A = I$
    - Se A^{-1} existir, ela é chamada matriz INVERSA. Se A^{-1} não existir, A é chamada matriz SINGULAR.

Uma Breve Revisão de Álgebra Linear

- Definições Iniciais
  - Determinante de uma Matriz
    - Determinante de uma matriz quadrada é um número associado à matriz dada pela definição recorrente seguinte:

      dada uma matriz quadrada A n x n, chama-se DETERMINANTE de A e indicamos |A| ou det A, ao número dado por:

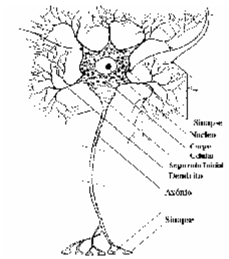
    a) Se n=1 então |A| = a^{11}

    b) Se n>1 então $$|A| = \sum_{j=1}^{n} (-1)^{1+j} a^{1j} . |A^{1,j}|$$

    onde A^{1j} é a submatriz obtida da matriz A eliminando a linha 1 e a coluna j
    - det A = 0 se e somente se A não é inversível.

Redes Neurais Artificiais Origens

- Neurônio
  - Modelo Simplificado
  - Modelo Simulado
  - Características Básicas
    - Adaptação
    - Representação de Conhecimentos baseada em conexões

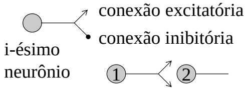


Circuitos Neurais e Computação

- O Modelo de McCulloch & Pitts (1943)
  - O Cérebro como um Sistema Computacional
  - 5 Suposições Básicas
    - A atividade de um neurônio é um processo tudo ou nada.
    - Um certo número fixo (>1) de entradas devem ser excitadas dentro de um período de adição latente para excitar um neurônio.
    - O único atraso significativo é o atraso sináptico.
    - A atividade de qualquer sinapse inibitória previne absolutamente a excitação do neurônio.
    - A estrutura das interconexões não muda com o tempo.

O Modelo de McCulloch& Pitts

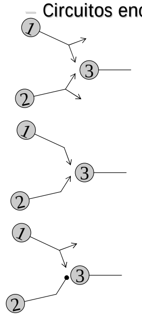
- Comportamento da Rede Neural pode ser expresso por predicados.
- Notação:
  - Ni(t): Asserção que o i-ésimo neurônio dispara no tempo t.
  - ¬ Ni(t): Asserção que o i-ésimo neurônio NÃO DISPARA no tempo t.
- Circuitos encontrados no SNC



N2(t)=N1(t-1)

O Modelo de McCulloch& Pitts

- Circuitos encontrados no SNC



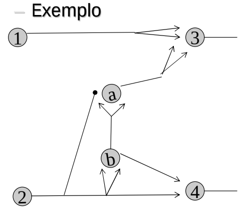
N3(t)=N1(t-1) ou N2(t-1) DISJUNÇÃO

N3(t)=N1(t-1) e N2(t-1) CONJUNÇÃO

N3(t)=N1(t-1) e ¬ N2(t-1)

O Modelo de McCulloch& Pitts

- Exemplo



N3(t)=N1(t-1) ou Na(t-1)

Na(t-1)=Nb(t-2) e ¬ N2(t-2)

Nb(t-2) = N2(t-3)

N3(t)=N1(t-1) ou (N2(t-3) e ¬N2(t-2))

N4(t) = N2(t-1) e Nb(t-1)

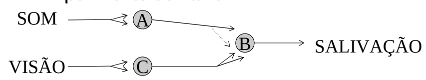
N4(t) = N2(t-1) e N2(t-2)

O Modelo de McCulloch& Pitts

- Conseqüências
  - +
    - Combinação dos neurônios implementa qualquer função lógica - Rede Neural como Computador Digital.
  - -
    - Não explica como são formadas as topologias das Redes Neurais.
    - Não explica como acontece o aprendizado.
    - Rede Neural só funciona corretamente se todos os elementos funcionarem corretamente.

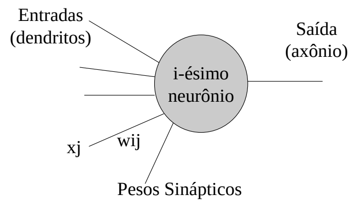
Uma forma de aprendizado no Modelo de McCulloch& Pitts

- O Aprendizado de Hebb (1949)
  - Base para todas as outras regras de aprendizado
    - "Quando um axônio de um neurônio A está próximo o suficiente para excitar um neurônio B, e repetidamente ou persistetemente toma parte do disparo de B, então, ocorre um certo processo de crescimento ou mudança metabólica em uma das 2 células, de forma que a eficiência de A em contribuir para o disparo de B é aumentado ("força do contato sináptico)."
  - Experimento de Pavlov



Modelos de Neurônio Artificiais

- Neurônio Artificial, Nó ou Processing Element-PE
- Um Primeiro Modelo



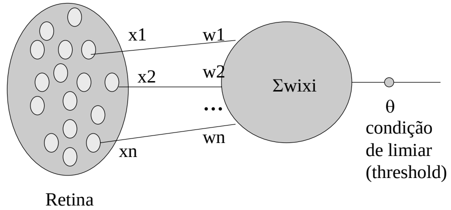
$$net_i = \sum_j w_{ij}.x_j$$

Modelos de Neurônio Artificiais

- O PERCEPTRON : Frank Rosenblatt (1958)
  - "A conectividade desenvolvida nas redes biológicas contém um grande número aleatório de elementos".
  - No início, o Perceptron não é capaz de distinguir padrões e portanto ele é genérico.
  - Pode ser treinado.
  - Com o tempo foi-se notando que a capacidade de separabilidade era dependente de "certas condições de contorno" dos padrões de entrada.

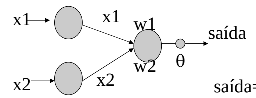
Modelos de Neurônio Artificiais

- O PERCEPTRON : Frank Rosenblatt (1958)



Modelos de Neurônio Artificiais

- O PERCEPTRON : Frank Rosenblatt (1958)



net=x1.w1+x2.w2

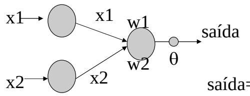
$$\text{saída=f(net)=} \begin{cases} 1 \text{ se net} \geq \theta \\ 0 \text{ se net} < \theta \end{cases}$$

EQUAÇÃO FUNDAMENTAL DO PERCEPTRON

w1.x1+w2.x2=θ ← EQUAÇÃO DE UMA RETA

Modelos de Neurônio Artificiais

- O PERCEPTRON : Exemplos

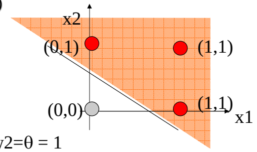


net=x1.w1+x2.w2

saída=f(net)= { 1 se net ≥ θ ; 0 se net < θ }

- Determinar w1, w2 e θ para que o Perceptron aprenda a função OU Lógico (OR)

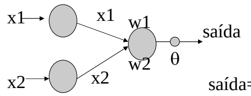
x1	x2	OR
0	0	0
0	1	1
1	0	1
1	1	1



Possível solução: w1=w2=θ = 1

Modelos de Neurônio Artificiais

- O PERCEPTRON : Exemplos

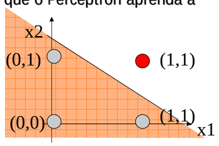


net=x1.w1+x2.w2

saída=f(net)= { 1 se net ≥ θ ; 0 se net < θ }

- Determinar w1, w2 e θ para que o Perceptron aprenda a função E Lógico (AND)

x1	x2	OR
0	0	0
0	1	0
1	0	0
1	1	1

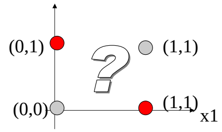


Possível solução: w1=w2=1 θ = 2

Modelos de Neurônio Artificiais

- O PERCEPTRON : Exemplos
  - Determinar w1, w2 e θ para que o Perceptron aprenda a função OU-EXCLUSIVO Lógico (XOR)

x1	x2	OR
0	0	0
0	1	1
1	0	1
1	1	0



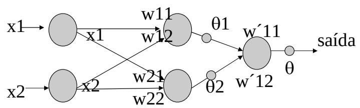
CONCLUSÃO: O PERCEPTRON É CAPAZ DE DISTINGUIR APENAS PADRÕES LINEARMENTE SEPARÁVEIS!!!

Modelos de Neurônio Artificiais

- O PERCEPTRON : Minsky e Papert (1969)
  - **CONCLUSÃO: O PERCEPTRON É CAPAZ DE DISTINGUIR APENAS PADRÕES LINEARMENTE SEPARÁVEIS!!!**
  - **Isto causou um "trauma" na comunidade científica e levou ao corte de verbas para as pesquisas em Redes Neurais Artificiais.**
  - **"Se colocarmos mais uma camada de neurônios podemos resolver esta limitação. Mas como achar os pesos?"**
  - **Na época não se sabia como.**

Modelos de Neurônio Artificiais

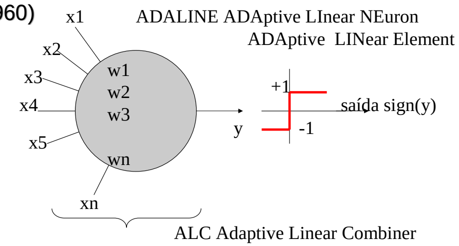
- O PERCEPTRON : Minsky e Papert (1969)



w11=w12=w21=w22=1
θ1=0,4
θ2=1,2
θ = 0.5

Modelos de Neurônio Artificiais

- O ALGORITMO DE APRENDIZADO DO PERCEPTRON : O ADALINE - Widrow e Hoff (1960)



Modelos de Neurônio Artificiais

- O ADALINE - Widrow e Hoff (1960)
  - Formulação Vetorial

$$W = \begin{pmatrix} w0 \\ w1 \\ w2 \\ \dots \\ wn \end{pmatrix} \quad X = \begin{pmatrix} x0 \\ x1 \\ x2 \\ \dots \\ xn \end{pmatrix} \quad y = W^T.X$$

Suponha que temos um conjunto de vetores de entrada $\{X_1, X_2, ..., X_L\}$, cada um com seu valor de saída correto (desejado) $\{d_1, d_2, ..., d_L\}$.
O Método de Aprendizado procura achar os pesos de forma a minimizar a diferença entre a saída desejada e a saída obtida com o vetor de entrada.

Modelos de Neurônio Artificiais

- O ADALINE - Widrow e Hoff (1960)
- A Regra de Hebb
  - Idealizada por Hebb, a idéia básica é que se duas unidades são ativadas simultaneamente, suas interconexões tendem a se fortalecer. Se **i** recebe o sinal de saída de **j**, o peso **Wij** é modificado de acordo com :

$$\Delta W_{ij} = \lambda a_i a_j$$

  - onde λ (lambda) é uma constante de proporcionalidade representando a taxa de aprendizado e **ai** e **aj** são ativações (ou saídas) das unidades **i** e **j** respectivamente. Alguns autores representam alternativamente a matriz **Wij** por **Wji**.

Modelos de Neurônio Artificiais

- O ADALINE - Widrow e Hoff (1960)
- A Regra Delta - Erro Médio Quadrático Mínimo (LMS) - Widrow-Hoff
  - É uma variante da Regra de Hebb, introduzida por Widrow-Hoff. A diferença quanto a de Hebb é que possui uma saída desejada dj. Assim, o peso será proporcional à saída.
  - Sendo aj e ai os níveis de ativações das unidades j e i respectivamente, a variação dos pesos é:

$$\Delta W_{ij} = (d_j - a_j) a_i$$

Modelos de Neurônio Artificiais

- Como Minimizar o Erro Médio Quadrático
  - Exemplo
    - Função OR: k=4

x1	x2	OR
0	0	0
0	1	1
1	0	1
1	1	1

$\underbrace{x1 \quad x2}_{X_k}$ OR: d_k

$$X_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} d_1 = 0$$

$$X_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} d_2 = 1$$

$$X_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} d_1 = 1$$

$$X_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} d_4 = 1$$

Modelos de Neurônio Artificiais

- Como Minimizar o Erro Médio Quadrático
  - Erro Médio Quadrático
    - Erro para o k-ésimo vetor de entrada

$E_k = d_k - y_k$ desejado - obtido
    - Erro Médio Quadrático

$$< E_k^2 > = \frac{1}{L} \sum_{k=1}^{L} E_k^2$$

então $< E_k^2 > = < (d_k - y_k)^2 >$

mas $y_k = W^T X_k$

logo $< E_k^2 > = < (d_k - W^T X_k)^2 > =$

$< d_k^2 > + W^T < X_k X_k^T > W - 2 < d_k X_k^T > W$

Modelos de Neurônio Artificiais

- Como Minimizar o Erro Médio Quadrático
  - Erro Médio Quadrático

$$< E_k^2 > = < d_k^2 > + W^T < X_k X_k^T > W - 2 < d_k X_k^T > W$$

Equação de uma Parábola

Minimização do Error Médio Quadrático significa encontrar o FUNDO DA PARÁBOLA.

$$\frac{\partial < E_k^2 >}{\partial W} = 0$$

$$2 < X_k X_k^T > W - 2 < d_k X_k > = 0$$

Modelos de Neurônio Artificiais

- A Regra Delta - Erro Médio Quadrático Mínimo (LMS) - Widrow-Hoff
  - Determinação dos Conjunto de Pesos W, pelo Método da Descida Mais Íngreme (Steepest Descente)
    - Método Iterativo
    - Escreve-se W como uma função do "tempo".
      - Vetor de pesos iniciais W(0)
      - Vetor de pesos no "passo" ou "tempo" t W(t)

Modelos de Neurônio Artificiais

- Determinação dos Conjunto de Pesos W, pelo Método da Descida Mais Íngreme (Steepest Descent)

1-Começar especificando valores aleatórios para os pesos.

2-Aplicar um vetor de entrada X_k.

3-Determinar o erro $E_k(t)$ utilizando W(t).

$$E_k^2(t) = (d_k - W^T(t)X_k)^2$$

4-Supor que $E_k^2(t)$ é uma aproximação razoável para o Error Médio Quadrático $<E_k^2(t)>$

Modelos de Neurônio Artificiais

- Determinação dos Conjunto de Pesos W, pelo Método da Descida Mais Íngreme (Steepest Descent)

5-Calcular o gradiente do erro, isto é a "direção" em que derivada é maior.

$$E_k^2(t) = (d_k - W^T(t)X_k)^2$$

$$\nabla E_k^2(t) = 2(d_k - W^T(t)X_k)\frac{\partial E_k}{\partial W} =$$

$$2E_k(t).(-X_k) = -2E_k X_k$$

Modelos de Neurônio Artificiais

- Determinação dos Conjunto de Pesos W, pelo Método da Descida Mais Íngreme (Steepest Descent)

6-Atualizar o Vetor de Pesos.

$$W(t+1) = W(t) - \mu \nabla E_k^2(t)$$

$$W(t+1) = W(t) + 2\mu E_k X_k$$

com

$$E_k = d_k - W^T(t)X_k$$

Modelos de Neurônio Artificiais

- Determinação dos Conjunto de Pesos W, pelo Método da Descida Mais Íngreme (Steepest Descent)

7-Repetir os passos de 2 a 6 até o erro alcançar um valor suficientemente pequeno.

O Período "Negro"

- Minsky e Papert - 1969 - Livro "Perceptrons". O Perceptron (ou o Adaline) é incapaz de <u>classificar</u> corretamente padrões não linearmente separáveis.
- A **maioria** dos problemas são **<u>não linearmente separáveis</u>**.
- Apesar do descrédito gerado sobre a área da neurocomputação, entre 1969 e 1982 os estudos neste campo continuaram, ainda que englobadas em outras linhas de pesquisa, como processamento adaptativo de sinais, reconhecimento de padrões, modelamento biológico, etc. Este trabalho, ainda que silencioso, construiu as bases necessárias para que o desenvolvimento das redes neurais pudesse continuar de forma consistente.

O Renascimento

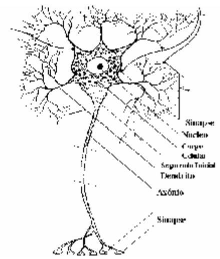
- Em 1974, Paul Werbos conseguiu o maior progresso em termos de redes neurais desde o perceptron de Rosenblatt: ele lançou as bases do algoritmo de retro-propagação ("backpropagation"), que permitiu que redes neurais com múltiplas camadas apresentassem capacidade de aprendizado. Em 1982, David Parker desenvolveu um método similar, de forma aparentemente independente. Contudo, a potencialidade deste método tardou a ser reconhecida.

O Renascimento

- Os primeiros resultados da retomada do desenvolvimento sobre redes neurais foram publicados em 1986 e 1987, através dos trabalhos do grupo PDP (Parallel and Distributed Processing), onde ficou consagrada a técnica de treinamento por backpropagation.

Redes Neurais Artificiais Os Tempos Modernos

- Elementos Básicos de um Neurônio Artificial
  - Uma Abordagem Unificada.
  - Terminologia Básica.



Elementos Básicos de um Neurônio Artificial

- A evolução das pesquisas no campo da abordagem conexionista levou ao desenvolvimento de uma infinidade de modelos de neurônios artificiais, de topologias de interconexão destes neurônios e algorítmos para aprendizado.
- Um trabalho que procurasse apresentar de maneira extensiva todos os modelos de neurônios, de topologias e de algorítmos de aprendizado, certamente ocuparia vários volumes e milhares de páginas.
- Abordagem Unificada - Modelo Formal

Elementos Básicos de um Neurônio Artificial

- **Definições de Sistemas Dinâmicos Aplicados ao Neurônio Artificial**
- *A partir da definição de Sistemas Dinâmicos, podemos definir um Neurônio Artificial (NA) como um sendo um Sistema Dinâmico onde:*

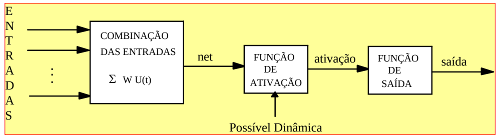
$T \subset \Re;$ ⮜Conjunto Tempo
$\Omega = \{\omega\} \subset f(\Re \rightarrow \Re^n);$ ⮜Conjunto dos Valores de Entrada
$U \subset \Re^n;$ ⮜Conjunto das Funções de Entrada
$Y \subset \Re;$ ⮜Conjunto dos Valores de Saída
$\Gamma = \{\gamma\} \subset f(\Re \rightarrow \Re);$ ⮜Conjunto das Funções de Saída
$X \subset \Re;$ ⮜Conjunto de Estados
$\phi : T \times T \times X \times \Omega \rightarrow X;$ ⮜Função de Transição de Estados
$\lambda : T \times X \times U \rightarrow Y.$ ⮜Função de Saída

Elementos Básicos de um Neurônio Artificial

- **Definições de Sistemas Dinâmicos Aplicados ao Neurônio Artificial**
  - A partir da definição formal para a representação de um neurônio artificial é possível descrever o funcionamento de diversos modelos de neurônios, bastando particularizar os parâmetros que definem o sistema. Particular atenção será dada à escolha da função de transição de estados ϕ e na função de saída λ e na maneira de combinar os valores de entrada dos neurônios.

Elementos Básicos de um Neurônio Artificial

- **Modelo de Neurônio Artificial**



Elementos Básicos de um Neurônio Artificial

- **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.

$$net_i(t) = \sum_{j=1}^{n} w_{ij} u_j(t)$$

wij é um número real que representa a conexão sináptica da entrada do i^{ésimo} neurônio com a saída do j^{ésimo} neurônio.

A conexão sináptica é conhecida como excitatória se $w_{ij}>0$ ou inibitória caso $w_{ij}<0$

Elementos Básicos de um Neurônio Artificial

- **Modelo de Neurônio Artificial**
  - Após a determinação do net_i, o valor da ativação do neurônio é atualizado através da função de ativação ϕ e finalmente, o valor de saída do neurônio é produzido através da função de saída λ.
  - A Função de Ativação ϕ $\quad x(t+1) = \phi(x(t), net(t))$
    - Os estados futuros de um neurônio são afetados pelo estado atual do neurônio e pelo valor do *net* de entrada. Este tipo de neurônio, que possui "memória" é conhecido como "neurônio dinâmico".
    - Por outro lado, se considerarmos a função como constante, teremos neurônios que não possuem "memória", ou seja, o estado atual é igual aos estados anteriores e portanto o neurônio é conhecido como "neurônio estático".

Elementos Básicos de um Neurônio Artificial

- **Modelo de Neurônio Artificial**
  - A Função de Saída λ
    - Essencialmente, qualquer função contínua e monotonicamente crescente tal que $x \in \Re$ e $y(x) \in [-1,1]$ pode 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

Elementos Básicos de um Neurônio Artificial

- **Modelo de Neurônio Artificial**
  - A Função de Saída λ
    - 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.

Elementos Básicos de um Neurônio Artificial

- **Modelo de Neurônio Artificial**
  - A Função de Saída λ
    - 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.

  - Confusão na Nomenclatura - CUIDADO!!!
    - **Na literatura muitas vezes só é apresentado o neurônio estático e portanto muitas vezes se confunde a função de ativação com a função de saída. Também é comum encontrarmos o termo "função de transferência".**

Elementos Básicos de um Neurônio Artificial

- **Modelo de Neurônio Artificial**
  - A Função de Saída λ
    - 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.

  - Confusão na Nomenclatura - CUIDADO!!!
    - **Na literatura muitas vezes só é apresentado o neurônio estático e portanto muitas vezes se confunde a função de ativação com a função de saída. Também é comum encontrarmos o termo “função de transferência”.**

Elementos Básicos de um Neurônio Artificial

- **Algumas Observações sobre a Dinâmica do Neurônio**
  - **Decaimento Passivo de Primeira Ordem.**
  - **Taxa de Decaimento.**
  - **Nível Residual.**

Topologias Das Redes Neurais Como os Neurônio se Conectam

- **Depende da forma como os Neurônios se conectam para formar uma “Rede” de neurônios.**
- **Redes Diretas - “Feedforward”**
- **Redes Recorrentes - “Feedback”**

Topologia das Redes Neurais

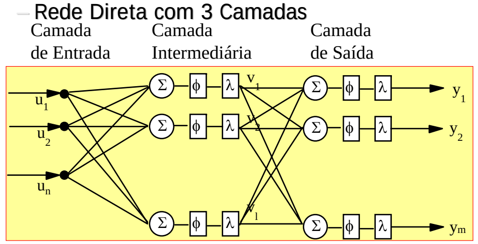
- Redes Neurais Diretas - Feedforward
  - As redes diretas são aquelas cujo grafo não tem ciclos.
  - Freqüentemente é comum representar estas redes em camadas e neste caso são chamadas redes de camadas.

Topologia das Redes Neurais

- Redes Neurais Diretas - Feedforward
  - Neurônios que recebem sinais de excitação 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.

Topologia das Redes Neurais

- Redes Neurais Diretas - Feedforward
  - Rede Direta com 3 Camadas

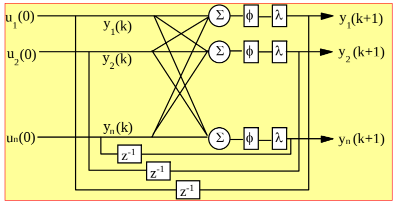


Topologia das Redes Neurais

- Redes Neurais Recorrentes - com Realimentação - Feedback
  - Redes com "feedback" são aquelas cujo grafo de conectividade contém pelo menos um ciclo.
  - Por esta razão McCulloch chamou-as de "networks with cycles", ou redes com ciclos.
  - Duas destas redes tem particular importância: as redes propostas por Hopfield e as redes bi-direcionais.

Topologia das Redes Neurais

- Redes Neurais Recorrentes - com Realimentação - Feedback



Topologia das Redes Neurais

- Rede Direta com Neurônios Estáticos
  - É Topologia de Rede Neural mais **popular** atualmente.
  - Também chamada de
    - Rede FeedForward
    - Multi Layer Perceptron
    - Rede Back Propagation (erradamente)

Aprendizado de Redes Neurais

- Um neurônio é considerado ser um elemento adaptativo. Seus pesos sinápticos são modificáveis dependendo do algoritmo de aprendizado.

Aprendizado de Redes Neurais

- 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. Um neurônio é considerado ser um elemento adaptativo. Seus pesos sinápticos são modificáveis dependendo do algoritmo de aprendizado.

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 (feedforward).

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.
    - Até o início dos anos 70 nenhum algoritmo de aprendizado para estas redes multi-camadas havia sido desenvolvido.
    - 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
  - Foi desenvolvido de maneira independente por vários pesquisadores
  - Em 1974, Werbos descobriu o algoritmo enquanto desenvolvia sua tese de doutorado em estatística e o chamou de "Algoritmo de Realimentação Dinâmica".
  - Parker em 1982 redescobriu o algoritmo e chamou-o de "Algoritmo de Aprendizado Lógico".
  - Foi com o trabalho de Rumelhart, Hinton e Williams do grupo PDP ("Parallel Distributed Processing") do MIT, que em 1986 divulgou e popularizou o uso do Backpropagation para o aprendizado em redes neurais.

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.

Aprendizado de Redes Neurais

- O Aprendizado Backpropagation
  - 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.

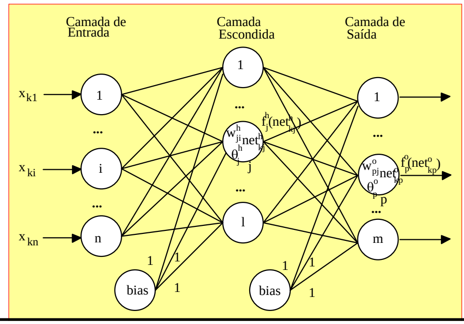
Aprendizado de Redes Neurais

- 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.

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.

Algoritmo Backpropagation



Algoritmo Backpropagation

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

$$Y = \theta(X) \quad : \quad X \in \Re^n, Y \in \Re^m$$

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

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

Algoritmo Backpropagation

- Etapas
  - 1. Aplicar um vetor de entrada do conjunto de treinamento e propagar até a saída
    - Um vetor de entrada $\mathbf{X}_k = [x_{k1}\ x_{k2}\ \dots\ x_{kn}]^T$ 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:

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

    - 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, assumimos que ao valor da função de ativação seja igual ao net, então, 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:

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

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

Algoritmo Backpropagation

- Etapas
  - 2. Calcular o error 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}

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

    - Podemos escrever a derivada de f_p^o como:

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

    - e o último termo da equação como:

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

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:

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

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

    - 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:

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

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:

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

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.

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

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

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

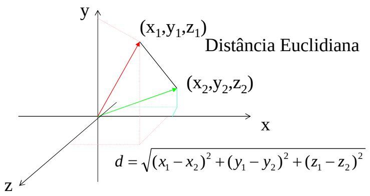
- Algumas dicas práticas
  - Valor do parâmetro de momento?
    - Valores "grandes" entre 0.8 e 0.9.
  - O que acontece se usarmos taxas de aprendizado muito grandes?
    - Converge rápido, mas pode não chegar no valor de erro mínimo, pois fica "saltando" de um lado para outro na superfície de erros.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

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"

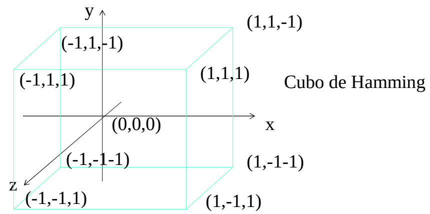
Redes Neurais Artificiais
Memórias Associativas

- 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



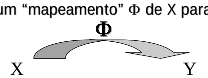
Redes Neurais Artificiais
Memórias Associativas

- 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: $d = \sqrt{4(\# \text{"desencontros"})}$
    - 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.



    - 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

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

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

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

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

$\Phi\ (X_2) = Y_2$

Redes Neurais Artificiais
Memórias Associativas

- Implementação por Redes Neurais (Elementos Processadores Distribuídos)

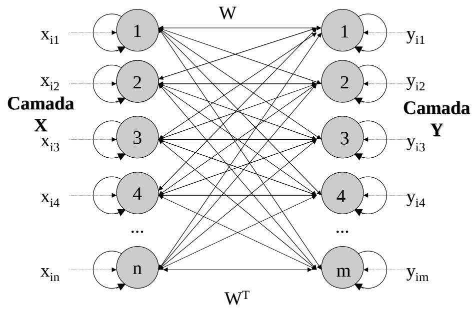
A MEMÓRIA BAM

(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
    - Aplicar um par de vetores iniciais (X_0,Y_0).
    - Propagar a informação de X para Y (multiplicar X_0 pela matriz W) e atualizar Y.
    - Propagar a informação atualizada de Y para X (multiplicar Y pela matriz W^T) e atualizar X.
    - 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

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

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

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

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
Memória BAM

- A Equação de Energia da BAM
  - Uma “conversa” a respeito de ESTABILIDADE e CONVERGÊNCIA.
    - O processamento de sistemas neurais artificiais é governado tipicamente por duas áreas da matemática: a ESTABILIDADE GLOBAL e a CONVERGÊNCIA.
    - ESTABILIDADE GLOBAL é a eventual estabilização de todas as ativações dos neurônios a partir de uma entrada inicial.
    - CONVERGÊNCIA é a eventual minimização do erro entre as saídas calculadas e desejadas.

Redes Neurais Artificiais
Memória BAM

- Uma “conversa” a respeito de ESTABILIDADE e CONVERGÊNCIA.
    - Durante o processo de treinamento, nas redes diretas com neurônios estáticos, os pesos formavam um sistema dinâmico. Isto é, os pesos se alteravam em função do tempo, e estas alterações podiam ser representadas por um conjunto de equações diferenciais.
    - Para a BAM, uma situação diferente ocorre. Os pesos são calculados anteriormente e não fazem parte do sistema dinâmico. Por outro lado, a rede pode levar vários “passos” até se estabilizar. Neste caso os vetores X e Y mudam em função do tempo e formam um sistema dinâmico.

Redes Neurais Artificiais
Memória BAM

- Uma “conversa” a respeito de ESTABILIDADE e CONVERGÊNCIA.
    - A questão então para os sistemas dinâmicos representados por redes neurais artificiais é se o sistema vai <u>CONVERGIR</u> para uma SOLUÇÃO <u>ESTÁVEL</u>.
    - Infelizmente muitos modelos de redes neurais não possuem provas para a convergência.
    - Quanto a estabilidade, na teoria de sistemas dinâmicos, um teorema pode ser provado no que diz respeito a existência de estados estáveis.
    - Este teorema utiliza o conceito de uma função chamada FUNÇÃO DE LYAPUNOV.

Redes Neurais Artificiais
Memória BAM

- Uma “conversa” a respeito de ESTABILIDADE e CONVERGÊNCIA.
  - FUNÇÃO DE LYAPUNOV
    - Definição: um ponto fixo χ é estável no sentido de Lyapunov se para todo $\varepsilon > 0$ existir um $\delta > 0$ tal que se $||x(t_0)- \chi || < \delta$, então $||x(t)- \chi || < \varepsilon$ para todo $t >= t_0$. Se além disso toda vizinhança de $x(t_0)$ tende para χ quando t vai a $\propto$, dizemos que χ é assintoticamente estável.

Redes Neurais Artificiais
Memória BAM

- FUNÇÃO DE LYAPUNOV
    - Se pudermos encontrar uma função limitada das variáveis de estado de um sistema dinâmico, de tal modo que toda mudança de estado resulte em uma diminuição do valor da função, então o sistema possui uma solução estável.
    - Esta função é chamada de Função de Lyapunov ou Função de Energia.

Redes Neurais Artificiais
Memória BAM

- A Equação de Energia da BAM
  - No caso da BAM, a Função de Lyapunov existe e é chamada <u>função de energia da BAM</u> e possui a forma:

    $E(X,Y) = -Y^T.W.X$
  - ou em termos dos seus componentes:

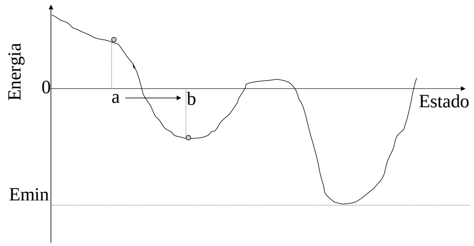
$$E = -\sum_{i=1}^{m}\sum_{j=1}^{n} y_i wij x_j$$

Redes Neurais Artificiais
Memória BAM

- A Equação de Energia da BAM
  - Teorema de Cohen-Grossberg de Estabilidade da BAM
    - 1. Qualquer mudança em X ou Y durante o processamento da BAM resulta em uma diminuição de E.
    - 2. E é limitada inferiormente por $E_{min} = -\Sigma_{ij}|w_{ij}|$.
    - 3. Quando E muda, ela deve mudar de valores finitos.
  - Itens 1 e 2 provam que E é uma função de Lyapunov. Item 3 restringe a possibilidade que as mudanças em E sejam infinitesimais, o que levaria a um tempo de convergência infinito.

Redes Neurais Artificiais
Memória BAM

- A Equação de Energia da BAM



Redes Neurais Artificiais
Memória BAM

- A Equação de Energia da BAM
  - Exemplo
    - Para o exemplo 1 visto anteriormente:
    - $E_{min} = -\Sigma_{ij}|w_{ij}| = -64$
    - para $X_0=[-1,-1,-1,1,-1,1,1,-1,-1,1]^T$ e $Y_0=[1,1,1,1,-1,-1]^T$

      $E = -Y_0^T W X_0 = -40$
    - para $X_0=[-1,-1,-1,1,-1,1,1,-1,-1,1]^T$ e $Y_{new}=[1,-1,-1,-1,-1,1]^T$

      $E = -Y^T_{new} W X_0 = -56$
    - para $X_{new}=[1,-1,-1,1,-1,1,1,-1,-1,1]^T$ e $Y_{new}=[1,-1,-1,-1,-1,1]^T$

      $E = -Y^T_{new} W X_{new} = -64$

Redes Neurais Artificiais
Memórias Associativas
A Memória de Hopfield

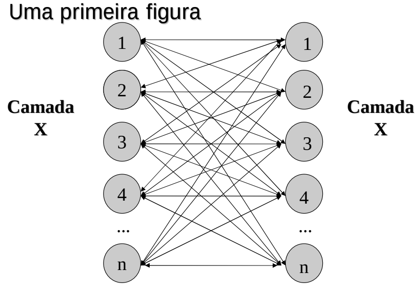
- Conceitos Básicos
  - Quem é John Hopfield?
    - É um professor de biologia e química do CalTech - California Institute of Technology.
  - O que ele fez?
    - Em 1982 Hopfield publicou um artigo que influenciou vários pesquisadores, chamando a atenção para as propriedades associativas de um classe 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.

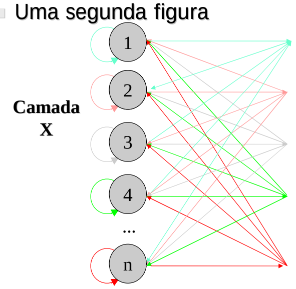
Redes Neurais Artificiais Memória de Hopfield

- Uma primeira figura



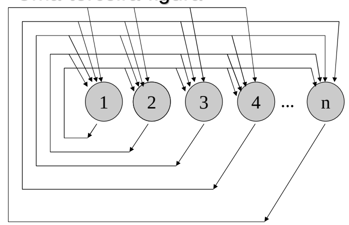
Redes Neurais Artificiais Memória de Hopfield

- 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.

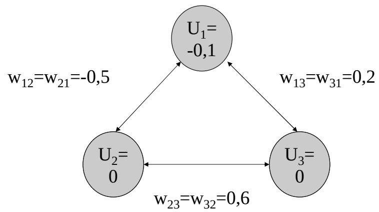
$$x_i(t+1) = \begin{cases} +1 & \text{se} \quad net_i^x > U_i \\ x_i(t) & \text{se} \quad net_i^x = U_i \\ -1 & \text{se} \quad net_i^x < U_i \end{cases}$$

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 $x_i=1$).

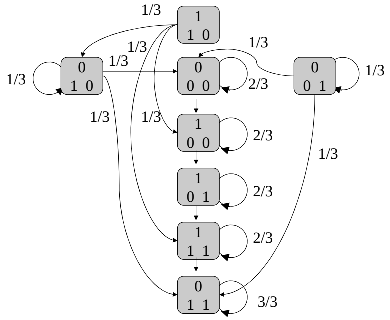
Redes Neurais Artificiais Memória de Hopfield

- 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:

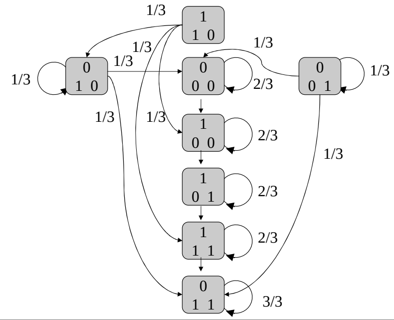
      $x_3(t+1)=0.(0,2)+0.(0,6)=0 < 0 \ (U_3)$

      $x_3(t+1)=0$

    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 $x_1,x_2,x_3=011$

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

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

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

      $E=-0,3$

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

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.

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

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.

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

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, $w_{23}-U_3<0$

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

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, $w_{23}+w_{13}-U_3>0$

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

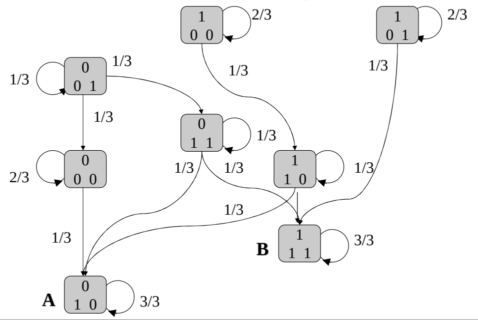
Redes Neurais Artificiais Memória de Hopfield

- 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



Redes Neurais Artificiais
A Rede “Counterpropagation”

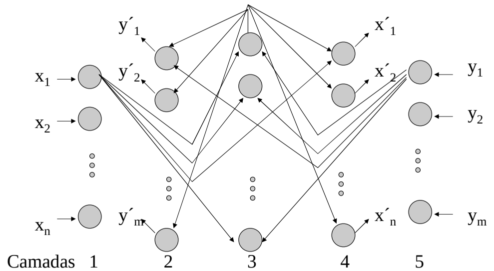
- O Que é uma Rede Counterpropagation (CounterPropagation Network - CPN)?
  - Basicamente é um modelo de rede neural desenvolvido por Hecht-Nielsen em 1987, formado através da “combinação” de outras duas arquiteturas já existentes de redes neurais: *A Rede Competitiva* e a estrutura *Outstar* de Grossberg.
  - Dado um conjunto de pares de vetores $(X_1,Y_1),(X_2,Y_2),\ldots,(X_L,Y_L)$, a rede pode aprender a associar um vetor X na camada de entrada com um vetor Y na camada de saída.
  - Não usa o aprendizado backpropagation.

Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Que é uma Rede Counterpropagation (CPN)?
  - Se a relação entre X e Y puder ser descrita através de uma função Φ contínua, tal que Y= Φ(X), a Rede CPN aprenderá a aproximar este mapeamento para qualquer valor de X no domínio especificado pelo conjunto de treinamento.
  - No modelo original, a rede seria também capaz de aprender o mapeamento da função inversa se ela existir.
  - Como para muitos casos práticos a função inversa não existe, a discussão da CPN pode ser simplificada apresentando apenas o mapeamento direto.

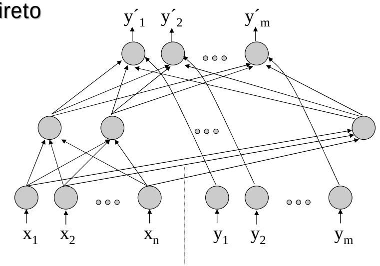
Redes Neurais Artificiais
A Rede “Counterpropagation”

- Desenho de uma Rede CPN Completa



Redes Neurais Artificiais
A Rede “Counterpropagation”

- Desenho de uma Rede CPN Mapeamento Direto



Redes Neurais Artificiais
A Rede "Counterpropagation"

- Como funciona a CPN para mapeamento direto?
    - Ela se parece com uma arquitetura de três camadas de uma rede Multi-Layer Perceptron, mas o funcionamento é diferente.
    - Um vetor de entrada é aplicado na camada de entrada, pré-processado e propagado para a camada intermediária.
    - Na camada intermediária, camada neurônio calcula o seu net e compete com os outros neurônios da masma camada para ver quem tem o maior valor de net.
    - Apenas a unidade vencedora é que manda o seu sinal para os neurônios da camada de saída.

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

Redes Neurais Artificiais
A Rede "Counterpropagation"

- Características interessantes da CPN.
  - +
    - Lançou a idéia de poder combinar vários tipos de redes neurais para formar uma nova arquitetura.
    - Usa um algoritmo de aprendizado diferente em cada camada.
    - Aprendizado bastante rápido.
    - Treinamento: Aplica-se (X,Y).
    - Recuperação: Aplica-se (X,ø), recupera-se Y´.
  - -
    - Não é muito "precisa".

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

Redes Neurais Artificiais
A Rede "Counterpropagation"

- Componentes da CPN.
  - Uma camada de entrada
    - Realiza um "pré-processamento" nos dados de entrada.
  - Um neurônio chamado "instar"
  - Uma camada de instars, chamada "camada competitiva"
  - Uma estrutura conhecida como "outstar"

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

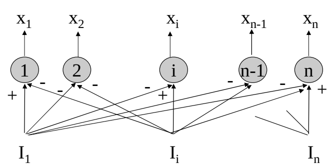
Redes Neurais Artificiais
A Rede "Counterpropagation"

- A Camada de Entrada.
  - Realiza a normalização dos dados de entrada.
    - Em outros modelos normalmente consideramos a camada de entrada como "buffer", que faz a distribuição dos valores de entrada para os neurônios da próxima camada.
    - Já fornecemos os valores de entrada escalonados ou normalizados, para evitar "overflow".
    - Isto não funciona assim na natureza. Sistemas biológicos devem possuir um mecanismo interno que previna a saturação por valores de entrada muito grandes. (ajuste de contraste)

Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

Redes Neurais Artificiais
A Rede "Counterpropagation"

- A Camada de Entrada.



Prof. Dr. Mauro Roisenberg - CPGCC - INE - UFSC

Redes Neurais Artificiais
A Rede "Counterpropagation"

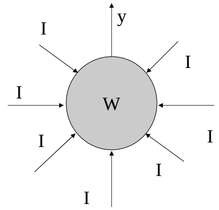
- A Camada de Entrada.
  - Padrão de Entrada
    - $I=(I_1,I_2,\ldots,I_i,\ldots,I_{n-1},I_n)^T$
  - Intensidade do padrão de entrada
    - $I=\Sigma I_i$
  - Padrão de Refletância (normalizado)
    - $\theta=(\theta_1, \theta_2,\ldots, \theta_i,\ldots, \theta_{n-1}, \theta_n)^T$
    - com $\theta_i = I_i / \Sigma I_i$
    - $\Sigma \theta_i=1$
  - Valor de saída dos neurônios da camada de entrada
    - $x_i = I_i / (\Sigma I_i^2)^{1/2}$

Redes Neurais Artificiais
A Rede “Counterpropagation”

- A Camada de Entrada.
  - O Padrão de Refletância é independente da intensidade total do padrão de entrada correspondente.
  - O Padrão de Refletância da face de uma pessoa é a mesma independentemente se a pessoa está no sol ou na sombra.

Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Instar.
  - A camada intermediária da CPN possui uma série de elementos conhecidos como INSTARS.

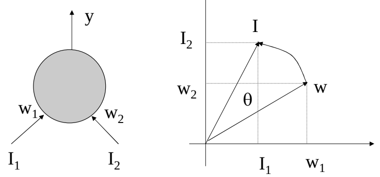


Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Instar.
  - Um INSTAR representa um “cluster” ou uma “classe” de padrões.
  - O INSTAR calcula o net ou produto escalar de um vetor de entrada pelo vetor de pesos.
  - Tanto o vetor de entrada como o vetor de pesos são normalizados.
  - net=I.W = ||I||.||W||.cos θ.
  - como os vetores estão normalizados, net=cos θ.
  - Se desejarmos que o Instar responda ao máximo a um vetor de entrada particular, devemos providenciar para que o vetor de pesos seja idêntico ao vetor de entrada. (cos 0º=1)

Redes Neurais Artificiais
A Rede “Counterpropagation”

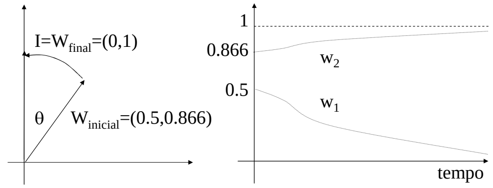
- O Instar.



O Instar “aprende” um vetor de entrada, rotando seu vetor de pesos na direção do vetor de entrada de modo que os dois fiquem alinhados.

Redes Neurais Artificiais
A Rede “Counterpropagation”

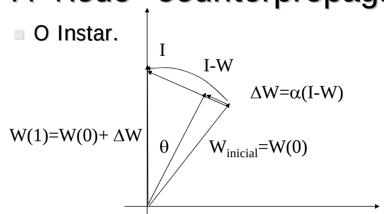
- O Instar.



Dado um vetor I=(0,1) e um vetor inicial w(0)=(0.5,0.866), os componentes w1 e w2 evoluem com o tempo procurando se alinhar com o vetor I.

Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Instar.



A quantidade (I-W) é um vetor que aponta de W em direção a I. W move-se em passos discretos na direção de I segundo a equação: $W(t+1)=W(t)+\alpha(I-W(t))$

Redes Neurais Artificiais
A Rede “Counterpropagation”

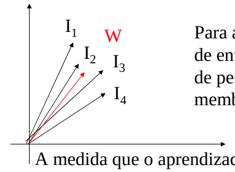
- O Instar aprendendo a se alinhar com a média de um cluster.
  - Um único Instar aprendendo um único vetor de entrada não é de grande interesse.
  - Considere a situação em que temos vários vetores de entrada, de tal forma próximos entre si de modo a que formem um “cluster” (classe).
  - Gostariamos que o Instar aprendesse a representar o cluster, se alinhando com a média dos vetores da classe.

Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Instar aprendendo a se alinhar com a média de um cluster.
  - Procedimento para aprendizado do Instar:
    - 1.Selecionar aleatoriamente um vetor de entrada I_i, de acordo com a distribuição de probabilidade do cluster.
    - 2.Calcular α(I-W) e atualizar o vetor de pesos.
    - 3. Repetir os passos 1 e 2 por um número de vezes igual ao número de vetores de entrada do cluster.
    - 4. Repetir o passo 3 várias vezes.

Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Instar aprendendo a se alinhar com a média de um cluster.



Para aprender um cluster de vetores de entrada, selecionamos um valor de peso inicial igual a um dos membros do cluster.

A medida que o aprendizado se desenrola, o vetor de peso acaba se estabilizando se uma região que representa a média dos vetores de entrada.

Redes Neurais Artificiais
A Rede “Counterpropagation”

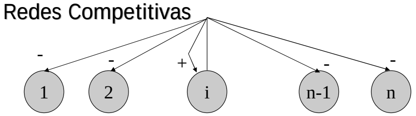
- Redes Competitivas
  - Um Instar aprende a responder a um grupo de vetores de entrada clusterizados juntos em uma região do espaço.
  - Vários Instars, agrupados um uma camada, cada um respondendo maximamente a um grupo de vetores em diferentes regiões do espaço.
  - Podemos dizer que esta camada de Instars classifica qualquer vetor de entrada, por que aquele Instar que tiver a maior saída, identifica a região do espaço correspondente ao vetor de entrada.

Redes Neurais Artificiais
A Rede “Counterpropagation”

- Redes Competitivas
  - Podemos determinar externamente qual o Instar com maior valor de saída (mas isto não existe na natureza).
  - Podemos também deixar que os Instars compitam entre si para ver quem é o vencedor.
  - O vencedor leva tudo (winner takes all)

Redes Neurais Artificiais
A Rede “Counterpropagation”

- Redes Competitivas

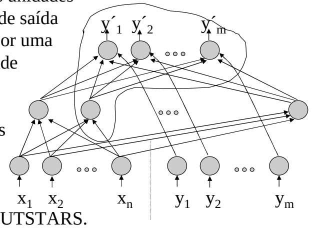


Um sistema com EXCITAÇÃO CENTRAL e INIBIÇÕES LATERAIS para implementar a competição entre um grupo de INSTARS. Cada unidade envia um sinal de realimentação positivo para si mesmo e um sinal inibitório para os outros neurônios. A unidade cujo vetor de peso melhor representa o vetor de entrada manda o sinal inibitório mais forte para as outras unidades e recebe a maior realimentação de si mesmo.

Redes Neurais Artificiais
A Rede “Counterpropagation”

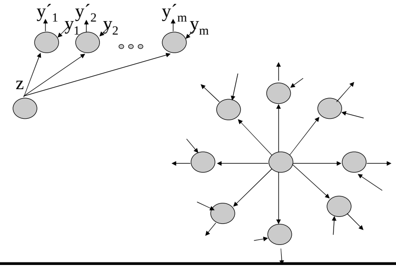
- O Outstar (Grossberg).

Um OUTSTAR é composto por todas as unidades da camada de saída do CPN e por uma única unidade da camada escondida. As unidades da camada de saída participam de vários OUTSTARS.



Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Outstar.



Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Outstar.
  - Grossberg argumenta que o Outstar é a mínima estrutura neural capaz de se condicionada. (condicionamento clássico)
  - Durante o período de treinamento, o vencedor da competição na camada escondida fornece um ESTÍMULO CONDICIONADO (o som da campainha) para os neurônios da camada de saída.
  - O ESTÍMULO INCONDICIONADO (a visão da carne), é fornecido pelo vetor Y da camada de entrada.

Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Outstar.
  - Como desejamos que a rede aprenda o vetor Y, a saída ou RESPOSTA NÃO CONDICIONADA (a salivação) será a mesma que o vetor Y.
  - Uma vez que o treinamento esteja completo, durante a operação da rede, a saída Y´ irá aparecer na saída da rede, mesmo que Y seja zero.

Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Outstar.
  - Como apenas um Instar é o vencedor a cada vez, o saída de cada neurônio da camada de saída é dada por w_i.

$$w_i(t+1)=w_i(t)+\beta(y_i-w_i(t))$$

  - $Y'=(y'_1,y'_2,\ldots,y'_m)^T$
  - $Y'=(w_1, w_2, \ldots,w_m)^T$

Redes Neurais Artificiais
A Rede “Counterpropagation”

- O Processamento da CPN.
  - Podemos agora combinar as estruturas componentes vistas anteriormente em uma CPN.
  - Considera-se uma simulação digital.

Redes Neurais Artificiais
A Rede "Counterpropagation"

- O Processamento da CPN.
  - Propagação pela CPN.
  - Considera-se que a seja já esteja treinada.
    - 0. Apresenta-se um vetor de entrada I.
    - 1. Normaliza-se o vetor de entrada, $x_i = I_i / (\Sigma_n I_n^2)^{1/2}$
    - 2. Aplica-se este vetor normalizado na porção X da camada de entrada. Aplica-se zero (vetor nulo) na porção Y desta mesma camada
    - 3. Uma vez que o vetor de entrada está normalizado, a camada de entrada apenas distribui este valor para os neurônios da camada intermediária.

Redes Neurais Artificiais
A Rede "Counterpropagation"

- O Processamento da CPN.
  - Propagação pela CPN.
    - 4. A camada intermediária é uma camada competitiva do tipo o vencedor leva tudo. Aquela unidade cujo vetor de peso melhor representa o vetor de entrada "vence" e tem sua saída setada para 1. Todas as unidades restantes tem sua saída zerada.
    - 5. A unidade vencedora da camada intermediária excita o Outstar e o valor de saída da rede é calculado. O valor da saída será igual ao valor do peso das conexões.

Redes Neurais Artificiais
A Rede "Counterpropagation"

- O Processamento da CPN.
  - Treinamento da CPN - Camada Competitiva.
  - Considera-se os vetores de entrada pré-normalizados.
  - Existe um juiz externo para determinar o vencedor da rede competitiva.
    - 1. Selecionar um vetor de entrada.
    - 2. Normalizar o vetor de entrada e aplicá-lo à Rede Competitiva.
    - 3. Determinar o vencedor (neurônio com maior net)
    - 4. Calcular α(X-W) apenas para a unidade vencedora e atualiazr o vetor de peso desta unidade

$$W(t+1)=W(t)+\alpha(X-W(t))$$

Redes Neurais Artificiais
A Rede "Counterpropagation"

- O Processamento da CPN.
  - Treinamento da CPN - Camada Competitiva.
    - 5. Repetir os passos 1 a 4, até todos os vetores de treinamento terem sido processados 1 vez.
    - 6. Repetir a etapa 5 até todos os vetores de entrada serem classificados corretamente.
    - 7. Testar.

Redes Neurais Artificiais
A Rede "Counterpropagation"

- O Processamento da CPN.
  - Treinamento da CPN - Camada de Saída.
  - Caso 1. Suponha que cada cluster represente uma classe, e todos os vetores em um cluster mapeiem um único vetor de saída.
    - Não é necessário treinamento iterativo.
    - Se a i-ésima unidade da camada competitiva vence para todos os vetores de uma classe cuja saída desejada é um vetor A, então setamos $w_{ki}=A_k$, onde w_{ki} é o peso da conexão da i-ésima da camada escondida para a k-ésima unidade de saída.

Redes Neurais Artificiais
A Rede "Counterpropagation"

- O Processamento da CPN.
  - Treinamento da CPN - Camada de Saída.
  - Caso 2. Se cada vetor de entrada em um cluster mapear para um vetor de saída diferente, então o processo de aprendizadom deve levar o outstar a reproduzir a média dos vetores de saída quando algum membro da classe for apresentado na entrada da CPN.
  - Se a média dos valores de saída não for conhecida, então um processo iterativo deve ser utilizado.

Redes Neurais Artificiais
A Rede "Counterpropagation"

- O Processamento da CPN.
  - Treinamento da CPN - Camada de Saída.
  - Caso 2.
    - 1. Aplica-se um vetor de entrada normalizado Xk, e seu correspondente vetor de saída Yk, às entradas X e Y da camada de entrada da rede.
    - 2. Determina-se a unidade vencedora da camada competitiva.

Redes Neurais Artificiais
A Rede "Counterpropagation"

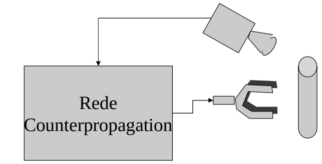
- O Processamento da CPN.
  - Treinamento da CPN - Camada de Saída.
  - Caso 2.
    - 3. Atualiza-se os pesos das conexões da unidade vencedora da camada competitiva com as unidades de saída.

$$w_i(t+1)=w_i(t)+\beta(y_{ki}-w_i(t))$$

    - 4. Repete-se os passos 1 a 3 até que todos os vetores de todas as classes produzam saídas satisfatórias.

Redes Neurais Artificiais
A Rede "Counterpropagation"

- Exemplo de Aplicação da CPN.
  - Ensinar para um robô, o ângulo em que se encontra um determinado objeto.

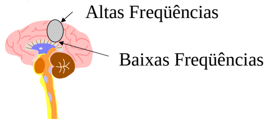


Redes Neurais Artificiais
Redes Auto-Organizáveis

- Metáfora Biológica - O Córtex
  - Características do Córtex
    - **É uma "FOLHA" larga (1~2 metros) e fina (2~4 milímetros).**
    - **Possui em média 6 camadas de neurônios de vários tipos e densidades.**
    - **"Dobrado e amassado" para caber na caixa craniana.**
    - **Centros especializados em diversas áreas, tais como fala, visão, audição, sensorial, motora, etc. estão localizados em regiões bem definidas e próximas umas das outras.**

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Metáfora Biológica - O Córtex
  - Características do Córtex - Regiões Especializadas
    - **Áreas individuais apresentam um ordenamento lógico consistente com a sua função.**
    - **Exemplo: MAPA TONOTÓPICO DA AUDIÇÃO**
      - Neurônios "vizinhos" respondem de maneira similar a sons de mesma freqüência, em uma seqüência ordenada das freqüências mais altas para as mais baixas.



Redes Neurais Artificiais
Redes Auto-Organizáveis

- Metáfora Biológica - O Córtex
  - Características do Córtex - Regiões Especializadas
    - **Exemplo: MAPA SOMATOTÓPICO**
      - **Existe uma estreita relação entre regiões da área somatotópica do cérebro e as partes do corpo que elas controlam.**
      - **A estrutura básica do corpo é refletida na organização do córtex nesta região.**

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Princípio de Funcionamento
  - Voltando às Redes Competitivas
    - **O Aprendizado se dá por um processo de auto-organização - APRENDIZADO NÃO-SUPERVISIONADO.**
    - **Cada neurônio da Rede Competitiva aprende a responder maximamente a diferentes padrões dos vetores de entrada.**
    - **A localização física dos neurônios da Rede Competitiva parece não refletir nenhuma relação entre as diferentes classes de dados que estão sendo aprendidas.**
    - **Existe um mapeamento aleatório entre as classes de dados e os neurônio competitivos.**

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Princípio de Funcionamento
  - Uma simples “extensão” do algoritmo de aprendizado competitivo resulta em um mapeamento com ***preservação-da-topologia*** dos dados de entrada nos neurônios da camada competitiva.
  - Para que a topologia se preserve, neurônios localizados fisicamente próximos uns dos outros devem responder de maneira similar a classes de vetores de entrada que também estejam próximos uns dos outros (sejam parecidos).

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Princípio de Funcionamento
  - Embora seja fácil visualizar neurônios vizinhos em uma matriz bidimensional, não é tão fácil determinar que classes de vetores estão próximas umas das outras em um espaço multi-dimensional.
  - Vetores de entrada multi-dimensionais são de certa forma “projetados” em uma superfície bidimensional, de modo que a ordem natural dos vetores de entrada se mantenha.
  - Isto permite que se visualize relações importantes entre os dados que de outra forma poderiam passar despercebidas.

Redes Neurais Artificiais
Redes Auto-Organizáveis

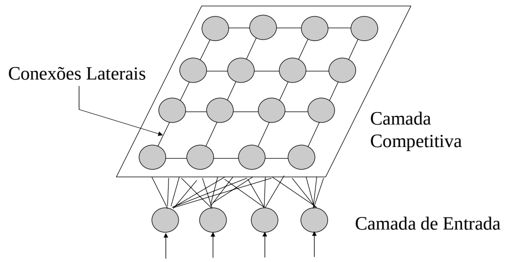
- Princípio de Funcionamento
  - O pioneiro no desenvolvimento da teoria das Redes Competitivas foi Teuvo Kohonen, e por esta razão os neurônios de uma rede competitiva são muitas vezes chamados de neurônios de Kohonen. Do mesmo modo, as rede auto-organizáveis são também conhecidas como Redes de Kohonen ou Mapas Topológicos de Kohonen.

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Princípio de Funcionamento
  - A Rede de Kohonen é uma estrutura de duas camadas de neurônios.
  - A primeira camada é a Camada de Entrada, e seus neurônios estão completamente interconectados aos neurônios da segunda camada.
  - A segunda camada é a Camada Competitiva. Normalmente esta camada está organizada como uma grade bidimensional, com cada neurônio conectado a todos os neurônios em sua vizinhança.

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Princípio de Funcionamento



Redes Neurais Artificiais
Redes Auto-Organizáveis

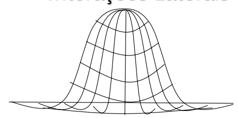
- Processamento da Rede de Kohonen
  - Na Rede CPN, a camada competitiva era organizada segundo o esquema de “excitação central - inibição lateral”, isto é, os neurônios excitavam a si mesmos e procuravam inibir os outros neurônios para que, no equilíbrio, aquele neurônio que possuisse a maior saída, vencesse a competição e a saída dos outros neurônios caisse a zero.
  - Na Rede de Kohonen, durante o processo de treinamento, a realimentação positiva atinge não só o próprio neurônio, como também a uma vizinhaça finita em torno do neurônio.

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Processamento da Rede de Kohonen
  - Na Rede CPN, apenas ao neurônio vencedor, isto é, aquele cujos pesos melhor representasse o vetor de entrada é que era dada a chance de “aprender”, ou seja, de modificar os seus pesos na mesma direção do vetor de entrada.
  - Na Rede de Kohonen, durante o processo de treinamento, todos os neurônios que recebem sinais excitatórios do neurônio vencedor também tem seus pesos modificados, participando assim do processo de “aprendizado”.
  - Estes sinais excitatórios são propagados através das conexões laterais.

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Interações Laterais da Rede de Kohonen

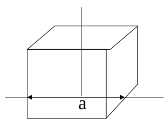


Curva característica das interações laterais entre certos neurônios encontrados no córtex, é chamada de CHAPÉU MEXICANO.

• Um neurônio no centro, mais fortemente excitado, excita lateralmente uma pequena vizinhança com realimentações positivas.
• A medida que a distância lateral do neurônio central aumenta, a excitação decresce até que vira uma inibição.

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Interações Laterais da Rede de Kohonen



Uma aproximação prática para a função CHAPÉU MEXICANO.

• A distância “a” define uma vizinhança de neurônios em torno de um neurônio central, que vão participar do aprendizado, junto com o neurônio central.

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Aprendizado na Rede de Kohonen
  - Durante o período de treinamento, cada neurônio dentro da vizinhança do neurônio vencedor participa do processo de treinamento.
  - 1. Inicializa-se os pesos aleatoriamente.
  - 2. Aplica-se um vetor de entrada e determina-se o neurônio vencedor.
    - Para um vetor de entrada X, o neurônio vencedor é $||X-W_c|| = \min\{||X-W_i||\}$
    onde o índice c se refere ao neurônio vencedor.

  O neurônio vencedor é aquele “mais próximo” do vetor de entrada.

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Aprendizado na Rede de Kohonen
  - 3. Atualiza-se não só os pesos do neurônio vencedor, mas também os pesos dos neurônios que estão na vizinhança do neurônio vencedor
    - $Wi(t+1)=Wi(t) + \alpha(t).(X-Wi(t))$

      se o neurônio for o neurônio vencedor ou estiver na vizinhança do vencedor.
  - Cada vetor de peso que participa do processo de aprendizado “rota” um pouco na direção do vetor de entrada X.

Redes Neurais Artificiais
Redes Auto-Organizáveis

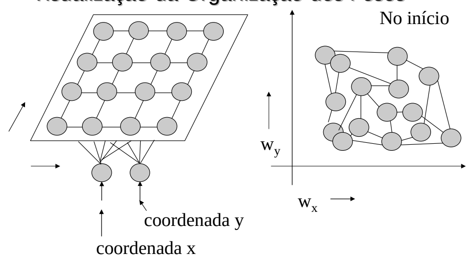
- Aprendizado na Rede de Kohonen
  - A vizinhança começa com um valor grande para que um grande número de neurônios participem do processo de aprendizado.
  - A medida que o treinamento vai prosseguindo, o tamanho da vizinhança vai diminuindo até englobar apenas o próprio neurônio vencedor.

Redes Neurais Artificiais
Redes Auto-Organizáveis

- Visualização da Organização dos Pesos
  - Kohonen desenvolveu um maneira interessante para ilustrar como se desenvolve o processo de treinamento.
  - Suponha que desejemos ensinar a uma Rede de Kohonen a reconhecer pontos de um sub-espaço bidimensional.

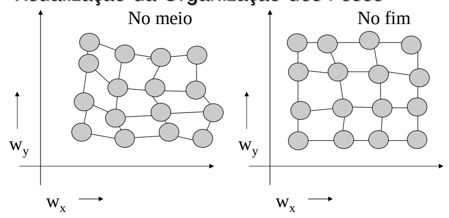
Redes Neurais Artificiais
Redes Auto-Organizáveis

- Visualização da Organização dos Pesos



Redes Neurais Artificiais
Redes Auto-Organizáveis

- Visualização da Organização dos Pesos



Redes Neurais Artificiais
Redes Auto-Organizáveis

- Observações Importantes
  - A grande utilidade da Rede de Kohonen é tentar projetar em uma superfície bidimensional características de vetores multidimensionais.
  - É mais fácil visualizar neurônios que estão próximos uns dos outros em uma superfície bidimensional, do que determinar que classes de vetores estão próximas entre si em um espaço multidimensional
  - Esta redução dimensional, preservando a ordem natural dos vetores de entrada, permite visualizar relações importantes entre os dados, que, de outro modo, poderiam passar desapercebidas.

Redes Neurais Artificiais
Redes ART - Adaptive Resonance Theory

- Metáfora Biológica - Memória Humana
  - Pode adicionar nova informação sem "esquecer" aquela já armazenada.
- Característica Não Presente na Maioria dos Modelos de Redes Neurais.
  - Normalmente uma RNA codifica informação na forma de pesos durante a fase de treinamento.
  - Após o treinamento, a adição de uma nova informação implica retreinamento da rede para esta informação e para TODA a informação anterior.

Redes Neurais Artificiais
Redes ART

- Dilema da Estabilidade-Plasticidade (Grossberg)
  - Como um sistema adaptativo pode permanecer plástico em resposta a uma entrada significativa e ainda se manter estável para entradas irrelevantes?
  - Quando chavear entre modos adaptativos (plásticos) e modos estáveis? Em geral, a rede não sabe que ela não conhece o padrão de entrada.
  - Como reter informação aprendida previamente e continuar aprendendo novas informações?

Redes Neurais Artificiais
Redes ART

- Idéia Básica
  - Usar uma extensão do aprendizado competitivo visto anteriormente.
  - Novidade:

    Utilizar um mecanismo de realimentação entre a camada competitiva e a camada de entrada.

Redes Neurais Artificiais
Redes ART

- Idéia Básica
  - Este mecanismo de realimentação permite:
    - o aprendizado de novas informações sem a destruição da informação anterior,
    - o chaveamento automáticos entre os modos plástico e estável, e
    - a estabilização da codificação das classes feita pelos neurônios.

Redes Neurais Artificiais
Redes ART

- Idéia Básica
  - O nome ART veio da forma como o aprendizado e a recuperação da informação ocorrem na rede.
  - Existem vários modelos de redes ART:
    - ART1 - ENTRADAS BINÁRIAS
    - ART2- ENTRADAS ANALÓGICAS
    - FUZZY-ART, ARTMAP, FUZZY-ARTMAP
  - RESSONÂNCIA:
    - uma vibração de pequena amplitude em um freqüência apropriada, causa uma vibração de grande amplitude.

Redes Neurais Artificiais
Redes ART

- Idéia Básica
  - Em uma rede ART, a informação, na forma de saída dos neurônios reverbera para frente e para trás entre as camadas.
  - Se um padrão apropriado se desenvolve no interior do sistema, aparece uma oscilação estável, o que é o equivalente ao conceito de ressonância para a rede neural.
  - Durante este período ressonante, o aprendizado ou a adaptação podem ocorrer.

Redes Neurais Artificiais
Redes ART

- Como obter um estado ressonante
  - Um estado ressonante pode ser alcançado de duas maneiras:
    - Se a rede já havia aprendido anteriormente a reconhecer um vetor de entrada, então o estado ressonante será rapidamente atingido quando o vetor for apresentado. Durante a ressonância, o processo de adaptação irá reforçar a memorização do padrão armazenado.

Redes Neurais Artificiais
Redes ART

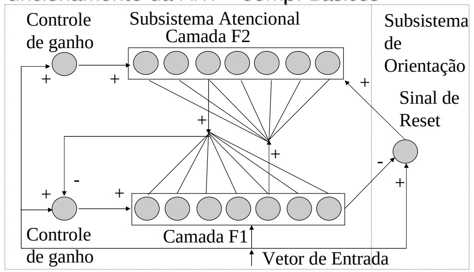
- Como obter um estado ressonante
  - Se o vetor de entrada não for imediatamente reconhecido, a rede irá rapidamente procurar entre os padrões já armazenados por um "match". Se não ocorrer um "match" a rede irá entrar num estado ressonante durante o qual o novo padrão será armazenado pela primeira vez.
  - Deste modo a rede responde rapidamente a padrões já memorizados, enquanto permanece capaz de aprender quando um novo padrão é apresentado.

Redes Neurais Artificiais
Redes ART

- Funcionamento da ART
  - As equações matemáticas que governam o funcionamento da ART são bastante complicadas.
  - "É fácil perder a visão da floresta como um todo quando visualizamos de perto a uma folha".
  - Apresentaremos primeiramente uma descrição qualitativa do processamento da rede ART.

Redes Neurais Artificiais
Redes ART

- Funcionamento da ART - Comp. Básicos

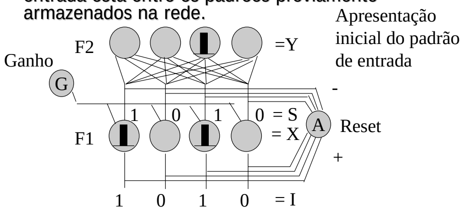


Redes Neurais Artificiais
Redes ART

- Funcionamento da ART
  - Padrões da atividade que se desenvolve entre os neurônios das duas camadas do subsistema atencional é chamada de Memória de Curta Duração - Short Term Memory (STM), poiseles existem apenas enquanto o vetor de entrada está sendo aplicado na rede.
  - Os pesos associados com as conexões de cima-par-baixo e vice-versa entre F1 e F2 são chamados de Memória de Longa Duração - Long Term Memory (LTM), pois eles codificam uma informação que ficará armazenada na rede por um longo período de tempo.

Redes Neurais Artificiais
Redes ART

- Funcionamento da ART
  - Descrição de uma seqüência hipotética de eventos que devem ocorrer em uma rede ART, quando esta tenta determinar se um padrão de entrada está entre os padrões previamente armazenados na rede.



Redes Neurais Artificiais
Redes ART

- Funcionamento da ART
  - Um padrão de entrada I, é apresentado para os neurônios em F1.
  - Um padrão de ativação X é produzido ao longo de F1, de modo similar ao funcionamento da camada de entrada de uma rede CPN.
  - O mesmo padrão de entrada excita tanto o subsistema de orientação A como o controle de ganho G.

Redes Neurais Artificiais
Redes ART

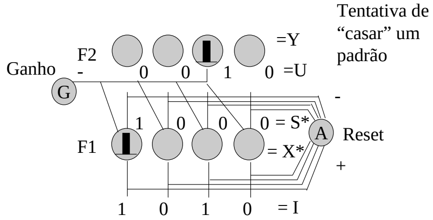
- Funcionamento da ART
  - O padrão de saída produz um sinal de saída inibitório que é enviado também para A.
    - **A Rede é estruturada de tal maneira que este sinal inibitório cancela exatamente o sinal excitatório que I manda para A, de modo que A permanece inativo,**
    - **Repare que G manda um sinal excitatório para F1. Como o mesmo sinal é aplicado a todos os neurônios da camada, neste caso ele é chamado de SINAL NÃO ESPECÍFICO.**
  - O aparecimento de X resulta em um padrão de saída S que é enviado através das conexões para F2.

Redes Neurais Artificiais
Redes ART

- Funcionamento da ART
  - Os neurônios de F2 calculam o net da maneira usual, de modo que um padrão de atividade Y se desenvolve ao longo dos neurônios de F2.
  - F2 é uma camada competitiva como na rede CPN, de modo que o vencedor leva tudo.

Redes Neurais Artificiais
Redes ART

- Funcionamento da ART



Redes Neurais Artificiais
Redes ART

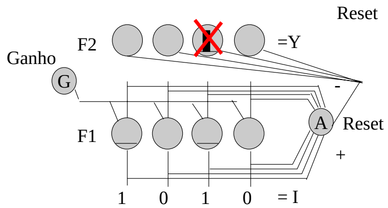
- Funcionamento da ART
  - O padrão de atividade Y, determina que um neurônio ganhe a competição na camada F2, este neurônio vencedor produz um padrão de saída U de F2.
  - Esta saída é enviada como uma conexão inibitória para o sistema de controle de ganho.. Este sistema é configurado de tal maneira que, se ele receber qualquer sinal inibitório de F2, ele cessa a sua atividade.
  - U se torna um um segundo padrão para os neurônios de F1. U é ponderada pelos pesos das conexões de F2 e é transformada num padrão V.

Redes Neurais Artificiais
Redes ART

- Funcionamento da ART
  - Os neurônios de F1 recebem agora sinais de entrada I e V. O padrão que se desenvolve em F1 neste instante é $I \cap V$, a interseção de I e V.
  - Assim, como os padrões I e V não são iguais, um novo padrão $X^* = I \cap V$ se desenvolve em F1.
  - Como o novo padrão de saída S*, é diferente do padrão original S, o sinal inibitório para A não mais cancela a excitação vinda de I.
  - A se torna ativo, como resposta ao não "casamento" de padrões em F1.

Redes Neurais Artificiais
Redes ART

- Funcionamento da ART

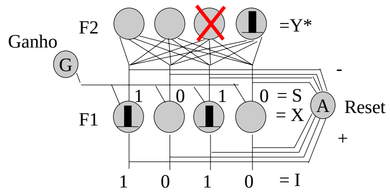


Redes Neurais Artificiais
Redes ART

- Funcionamento da ART
  - Como A se tornou ativo devido ao “não casamento” de padrões em F1, A manda um sinal não específico de RESET para todos os neurônios de F2.
  - Os neurônios de F2 respondem de acorso com o seu estado atual.
    - Se o neurônio estava ativo, ele se torna inativo e permanece neste estado por um longo período de tempo, de modo que ele não participa da competição entre os neurônios de F2 no próximo ciclo.
  - Y desaparece e portanto o sinal para F1 e o sinal inibitório para o ganho também desaparecem.

Redes Neurais Artificiais
Redes ART

- Funcionamento da ART



Redes Neurais Artificiais
Redes ART

- Funcionamento da ART
  - O padrão original X reaparece em F1 e um novo ciclo de tentativa de “casamento” recomeça.
  - Agora um novo padrão Y* aparece em F2 e o ciclo continua.
  - Este processo continua até que um “casamento” seja alcançado ou até que todos os neurônios de F2 tenham sido experimentados como “representantes” do padrão de entrada I.
  - Se não ocorrer o casamento, a rede escolherá um “novo” neurônio de F2 para representar o padrão e irá aprender o padrão.

Redes Neurais Artificiais
Redes ART

- Funcionamento da ART
  - O aprendizado ocorre pela modificação de pesos.
  - Quando um “casamento” ocorre, a rede entra em um estado ressonante, durante o qual os pesos são “reforçados”, isto é, se “aproximam” do padrão vencedor.

Redes Neurais Artificiais
Redes ART

- Algoritmo de Processamento da ART
  - 0. Determina-se o tamanho das camadas F1 e F2. Seja M o número de neurônios em F1 e N o número de neurônios em F2.
  - 1. Inicialização dos pesos.
    - $w^{F1->F2}_{ij}(0)=1/1+M$
    - $w^{F2->F1}_{ij}(0)=1$
  - 2. Aplicar um vetor de entrada I.
  - 3. Calcular os valores de ativação dos neurônios em F2.
    - $Y_i=\Sigma w^{F1->F2}_{ij}(t_0).I_j$
  - 4. Selecionar o neurônio vencedor k em F2.

Redes Neurais Artificiais
Redes ART

- Algoritmo de Processamento da ART
  - 5.Teste de vigilância para saber se houve “casamento” ou não.
    - Escolhe-se previamente para a rede um limiar para o teste de vigilância (ρ).
    - ρ indica quão perto do padrão de entrada deve estar um exemplar para ser considerado um casamento.
    - Se $\dfrac{(W^{F2->F1})^T_k(t).I}{I^T.I} > \rho$
      - então vá para o passo 7 (houve “casamento”)
      - senão vá para o passo 6 (não houve “casamento”).

Redes Neurais Artificiais
Redes ART

- Algoritmo de Processamento da ART
  - 6. Não houve o "casamento", portanto o neurônio k da camada F2 não representa o padrão I e deve ser desconsiderado.

    Desativar o neurônio k. A saída é zerada e o neurônio não participa nas próximas seleções do vencedor.

    Ir para o passo 3.

Redes Neurais Artificiais
Redes ART

- Algoritmo de Processamento da ART
  - 7. Houve o "casamento", e portanto deve atualizar os pesos para que na próxima vez este neurônio k represente ainda melhor este vetor I.

    Para cada neurônio r da camada F1.

$$w_{rk}^{F2->F1}(t+1) = w_{rk}^{F2->F1}(t).I_r$$

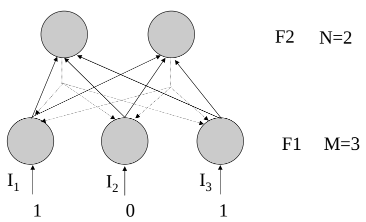
$$w_{kr}^{F1->F2}(t+1) = \frac{w_{kr}^{F1->F2}(t).I_r}{\frac{1}{2} + \sum_{j=1}^{M} w_{jk}^{F2->F1}(t).I_j}$$

Redes Neurais Artificiais
Redes ART

- Algoritmo de Processamento da ART
  - 8. Reativar todos os neurônios de F2 e ir para o passo 2.

Redes Neurais Artificiais
Redes ART

- Exemplo de Processamento da ART
  - 0. M=3 e N=2



Redes Neurais Artificiais
Redes ART

- Exemplo de Processamento da ART
  - 1. Inicialização dos pesos.

$w^{F2->F1}_{11}(0)=1$ $w^{F2->F1}_{12}(0)=1$: neurônio 1 de F1
$w^{F2->F1}_{21}(0)=1$ $w^{F2->F1}_{22}(0)=1$: neurônio 2 de F1
$w^{F2->F1}_{31}(0)=1$ $w^{F2->F1}_{32}(0)=1$: neurônio 3 de F1

$w^{F1->F2}_{11}(0)=1/1+M=1/4=0.25$
$w^{F1->F2}_{12}(0)=0.25$: neurônio 1 de F2
$w^{F1->F2}_{13}(0)=0.25$
$w^{F1->F2}_{21}(0)=1/1+M=1/4=0.25$
$w^{F1->F2}_{22}(0)=0.25$: neurônio 2 de F2
$w^{F1->F2}_{23}(0)=0.25$

Redes Neurais Artificiais
Redes ART

- Exemplo de Processamento da ART
  - 2. Aplicar I=(1 0 1)
  - 3. Calcular Y_i em F2

    Y_1= net_1 = 0.25.1+0.25.0+0.25.1 = 0.5
    Y_2= net_2 = 0.25.1+0.25.0+0.25.1 = 0.5
  - 4. Como os valores de ativação dos dois neurônios de F2 deram o mesmo resultado, tanto faz qual dos dois é o vencedor.

    Escolhemos o 1 como vencedor.
  - 5.Então calculamos o valor de saída de F2, como (1 0) e propagamos de volta para F1, e fazemos o Teste de Vigilância.

Redes Neurais Artificiais
Redes ART

- Exemplo de Processamento da ART
  - 5. Teste de Vigilância
    - $\frac{(W^{F2->F1})^T_k(t).I}{I^T.I} > \rho$

    (1 1 1).(1 0 1)T = 2

    (1 0 1).(1 0 1)T = 2

    2/2 = 1 > ρ
  - 7. Houve o “casamento”. Devemos agora atualizar os pesos.

Redes Neurais Artificiais
Redes ART

- Exemplo de Processamento da ART
  - 7. Atualização dos pesos.

$$w_{rk}^{F2->F1}(t+1) = w_{rk}^{F2->F1}(t).I_r$$

  - $w^{F2->F1}_{11}(1)=1.1=1$
  - $w^{F2->F1}_{21}(1)=1.0=0$
  - $w^{F2->F1}_{31}(1)=1.1=1$

Redes Neurais Artificiais
Redes ART

- Exemplo de Processamento da ART
  - 7. Atualização dos pesos.

$$w_{kr}^{F1->F2}(t+1) = \frac{w_{kr}^{F1->F2}(t).I_r}{\frac{1}{2}+\sum_{j=1}^{M} w_{jk}^{F2->F1}(t).I_j}$$

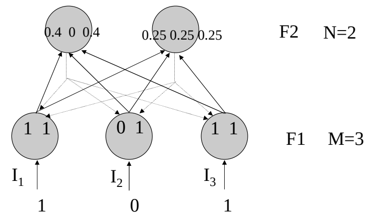
$$w_{11}^{F1->F2}(1) = \frac{1.1}{\frac{1}{2}+1.1+1.0+1.1} = \frac{1}{2.5} = 0.4$$

$$w_{12}^{F1->F2}(1) = \frac{1.0}{\frac{1}{2}+1.1+1.0+1.1} = \frac{0}{2.5} = 0$$

$$w_{13}^{F1->F2}(1) = \frac{1.1}{\frac{1}{2}+1.1+1.0+1.1} = \frac{1}{2.5} = 0.4$$

Redes Neurais Artificiais
Redes ART

- Exemplo de Processamento da ART
  - 7. Atualização dos pesos.



Redes Neurais Artificiais
Redes ART

- Exemplo de Processamento da ART
  - 8. Voltar para o passo 2. (Apresentar outro padrão de entrada - ou o mesmo)

Redes Neurais Artificiais
Redes Recorrentes

- Introdução

  Redes Neurais Recorrentes são aquelas que apresentam “ciclos” nas suas conexões, isto é, a saída de neurônios de uma camada i são entradas de neurônios de uma camada i-j, com j>=0;

  Redes BAM, de Hopfield e Competitivas em geral são de certo modo redes recorrentes.

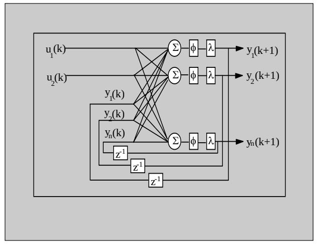
  Entretanto 2 modelos são de maior interesse neste caso:
  - Redes de Elman
  - Redes de Jordan

Redes Neurais Artificiais Redes Recorrentes

- Introdução
  - Importância das Redes Neurais Recorrentes
    - Capacidade de Modelagem
      - Redes Neurais Feedforward com neurônios estáticos são capazes de modelar apenas sistemas estáticos.
      - Redes Neurais Dinâmicas são capazes de modelar sistemas dinâmicos.
      - A dinâmica de uma rede por ser obtida utilizando-se neurônios dinâmicos ou através de uma dinâmica externa (uma realimentção externa munida de um retardo, por exemplo).

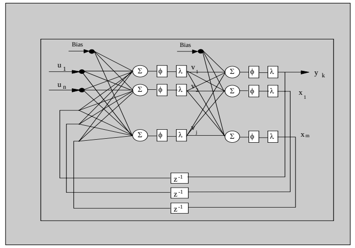
Redes Neurais Artificiais Redes Recorrentes

- Modelos com Dinâmica Externa
  - Modelo “Clássico”



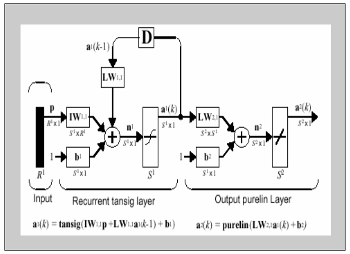
Redes Neurais Artificiais Redes Recorrentes

- Modelos com Dinâmica Externa
  - Modelo “com Camada Intermediária”



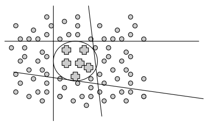
Redes Neurais Artificiais Redes Recorrentes

- Modelos com Dinâmica Externa
  - Redes de Elman



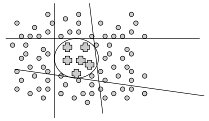
Redes RBF (Radial Basis Function)

- Introdução
  - A maioria das funções de saída dos neurônios vistos até agora eram funções monotonicamente crescentes dos valores de “net” dos neurônios.
  - Esta pode não ser a melhor escolha para alguns problemas encontrados na prática.



Redes RBF (Radial Basis Function)

- Introdução
  - Para o exemplo abaixo, uma rede neural direta convencional precisaria de 4 a 5 neurônios na camada escondida.
  - Um neurônio seria suficiente para discriminar as duas classes se sua função de saída se aproximasse de um círculo.



Redes RBF (Radial Basis Function)

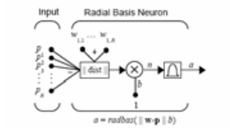
- Definição
  - Uma função é radialmente simétrica (uma RBF) se sua saída depende da distância entre o dado de entrada (vetor) e outro vetor armazenado.
  - As funções radiais simétricas comumente utilizadas (ρ) são funções não-crescentes de uma medida de distância u que é seu único argumento. Com $\rho(u_1)>\rho(u_2)$ sempre que $u_1<u_2$.
  - Normalmente u é a distância euclidiana entre o "centro" ou vetor armazenado μ e o vetor de entrada *i*. u=|| μ- *i* ||

Redes RBF (Radial Basis Function)

- Definição
  - Em redes RBF, a gaussiana é descrita pela equação:

$$\rho_g(u)\ \alpha\ e^{-(u/c)^2}$$

Modelo de neurônio de base radial:



Redes RBF (Radial Basis Function)

- Princípio de Funcionamento
  - Em muitas problemas de aproximação de funções a abordagem clássica envolve interpolação linear:

$$f(x_0) = f(x_1) + (f(x_2) - f(x_1))(x_0 - x_1)/(x_2 - x_1)$$

que simplifica para:

$$\frac{(D_1^{-1} f(x_1) + D_2^{-1} f(x_2))}{(D_1^{-1} + D_2^{-1})}$$

Onde D_1 e D_2 são as distâncias entre x_0 e x_1 e x_2 respectivamente.

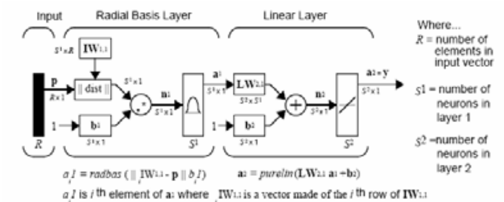
Redes RBF (Radial Basis Function)

- Princípio de Funcionamento
  - Em geral segmentos de hiperplanos conectam pontos próximos, de forma que o valor de saída correspondente a um novo vetor de entrada n-dimensional é determinado inteiramente pelos P0 exemplos de entrada que o "circundam".

$$\frac{(D_1^{-1} f(x_1) + D_2^{-1} f(x_2) + \Lambda\ + D_{P_0}^{-1} f(x_{P_0}))}{(D_1^{-1} + D_2^{-1} + \Lambda\ + D_{P_0}^{-1})}$$

Redes RBF (Radial Basis Function)

- Descrição



Redes RBF (Radial Basis Function)

- Descrição
  - Se apresentarmos um vetor de entrada para esta rede, cada neurônio na camada de base radial irá produzir uma saída inversamente proporcional a distância deste vetor de entrada com cada vetor de peso de cada neurônio.
  - Deste modo, neurônios de base radial com vetores de pesos muito diferentes do vetor de entrada, terão a sua saída praticamente igual a zero.
  - Esta saída pequena terá um efeito desprezível na saída dos neurônios da camada linear.

Redes RBF (Radial Basis Function)

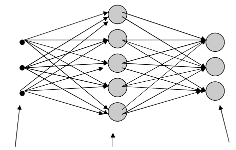
- Descrição
  - Por outro lado, um neurônio de base radial com vetor de pesos perto do vetor de entrada produzirá uma saída próxima de um.
  - Se um neurônio tem uma saída em 1, seus pesos de saída serão passados para os neurônios de saída da segunda camada (linear).

Redes Learning Vector Quantization (LVQ)

- Combina o aprendizado Supervisionado e Não-Supervisionado (competitivo).
  - Aprendizado Não-Supervisionado e Agrupamento podem ser muito úteis na etapa de pré-processamento em problemas de CLASSIFICAÇÃO.
  - A rede LVQ é uma aplicação de redes competitivas do tipo “o vencedor leva tudo”, juntamente com uma camada linear para tarefas em que se conhece a classe a que pertence cada padrão entrada do conjunto de treinamento.

Redes Learning Vector Quantization (LVQ)

- Combina o aprendizado Supervisionado e Não-Supervisionado (competitivo).



Redes Learning Vector Quantization (LVQ)

- A camada competitiva aprende a classificar os vetores de entrada, no que chamamos de sub-classes.
  - Nesta camada, o número de neurônios para cada classe é aproximadamente proporcional ao número de padrões de treinamento disponíveis para cada classe.

  (cada cluster tem aproximadamente o mesmo número de pontos)
- A camada linear transforma as sub-classes da camada competitiva na classificação final definida pelo usuário, chamadas de classes alvo (target classes).

Redes Learning Vector Quantization (LVQ)

- Camada Competitiva
  - A regra de atualização de pesos na camada competitiva de uma LVQ é um pouco diferente da regra de atualização de uma rede competitiva simples.
  - *“Quando um padrão **i** da classe C(**i**) é apresentado a rede, seja o neurônio vencedor um neurônio **j** que representa a classe C(**j**). Os pesos do neurônio **j** são movidos <u>na direção</u> do padrão **i** se C(**i**)=C(**j**) e na <u>direção inversa</u> caso contrário.*

Redes Learning Vector Quantization (LVQ)

- Camada Competitiva – Algoritmo LVQ1
  - Inicialize todos os pesos entre [0, 1]
  - Repita
    - *Ajuste a taxa de aprendizagem η(t)*
    - *Para cada padrão de entrada ik faça*
      - *Ache o neurônio vencedor (aquele cujos pesos forem mais próximos da entrada).*
      - *Para cada peso l do neurônio j faça*
        - *Se a rótulo da classe do neurônio j é igual a classe desejada para o padrão de entrada ik*
        - *Então ∆wj= η(t)(ik-wj),*
        - *Senão ∆wj= -η(t)(ik-wj)*
      - *Fim faça*
    - *Fim faça*
  - *Até que a rede convirja ou depois de um determinado tempo.*

Redes Learning Vector Quantization (LVQ)

- Camada Competitiva – Algoritmo LVQ1
  - A regra de atualização de pesos usa uma taxa de aprendizagem que é variável e diminui ao longo do tempo.
    - $\eta(t)=1/t$ ou
    - $\eta(t)=a[1-(t/A)]$ onde a e A são constantes positivas e A>1.
  - Isto permite que a rede convirja para um estado em que os vetores de pesos se tornam "estáveis" e variam pouco com a apresentação dos padrões de entrada.

Redes Learning Vector Quantization (LVQ)

- Camada Linear
  - Combina as saídas da camada competitiva em classes pré-determinadas pelo usuário.
  - Determinação dos pesos da camada linear:

$$W=\begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \updownarrow \text{Classes = n. de neurônios}$$

$\longleftrightarrow$ Neurônios da camada competitiva

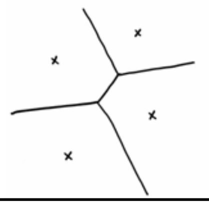
Apenas um 1 por coluna

Redes Learning Vector Quantization (LVQ)

- Camada Linear
  - A coluna da matriz indica que agrupamento da camada competitiva (sub-classe) pertence a que classe alvo (target class).
  - A matriz é fixa a calculada a priori.

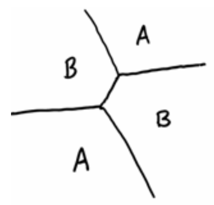
Redes Learning Vector Quantization (LVQ)

- Os pesos da primeira camada implicitamente dividem o espaço de entrada em "células de Voronoi".
- Cada célula contém os pontos mais próximos do seu vetor de pesos representativo.



Redes Learning Vector Quantization (LVQ)

- Na segunda camada os pesos agrupam várias regiões de modo a formar as classes desejadas.



Redes Learning Vector Quantization (LVQ)

- Mapas Auto-Organizáveis X LVQ
  - No treinamento das redes auto-organizáveis, o neurônio vencedor (o mais próximo do padrão de entrada) e sua vizinhança aprende movendo-se na direção do padrão de entrada.
  - Na rede LVQ, o neurônio mais próximo pode classificar correta ou incorretamente um padrão de entrada. Se classificação correta, ele se move na direção do padrão de entrada, se aproximando deste. Caso contrário, ele se move na direção contrária, se afastando dele.

Redes Learning Vector Quantization (LVQ)

- LVQ2
  - É uma variação do algoritmo de aprendizado LVQ1 e deve ser utilizado depois de um aprendizado LVQ1.
  - Se, entre dois neurônios com pesos W1 e W2, os mais próximos de um padrão de entrada (i), apenas um deles pertence a classe desejada, e ambos os pesos estão a distâncias comparáveis de i, então um deles (o que foi corretamente classificado) se aproxima de i, enquanto o outro se afasta de i.