Primeiras Redes Neurais

Aluizio Fausto Ribeiro Araújo

Universidade Federal de Pernambuco

Centro de Informática

Conteúdo

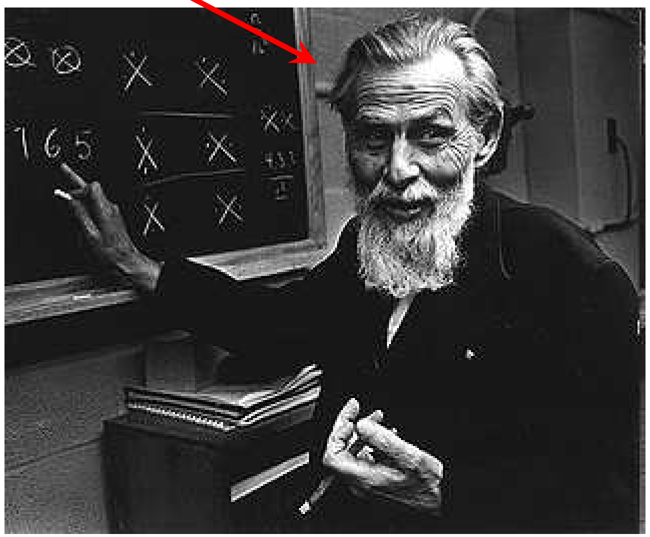
1. Modelo de McCullough and Pitts
2. Teoria de Hebb
3. O Perceptron
4. Exemplos

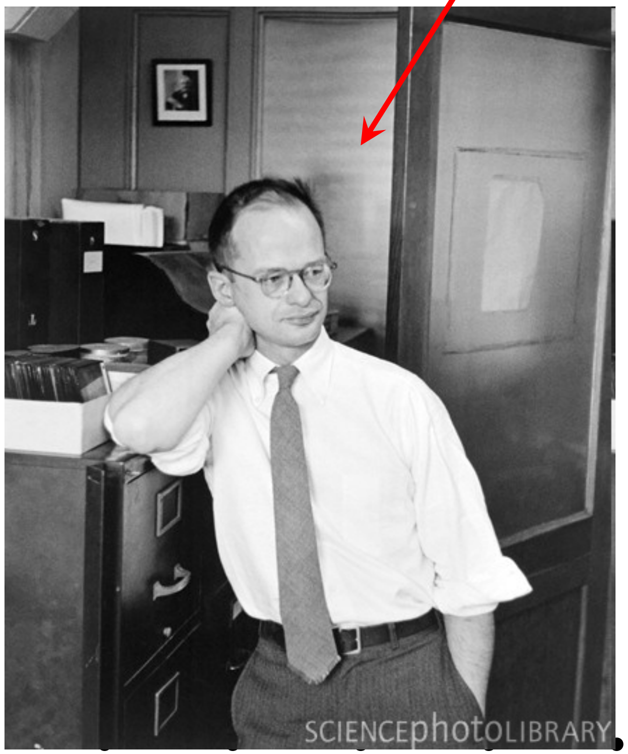
UFPE

Centro de Informática
UFPE

Modelo de McCullough and Pitts

• Modelo proposto pelo neufisiologista americano Warren Sturgis McCulloch (16/11/1898-24/09/1969) e um “logístico” Walter Pitts (23/04/1923-14/05/1969) em 1943 que foi publicado como um modelo eletrônico de como neurônios atuariam.



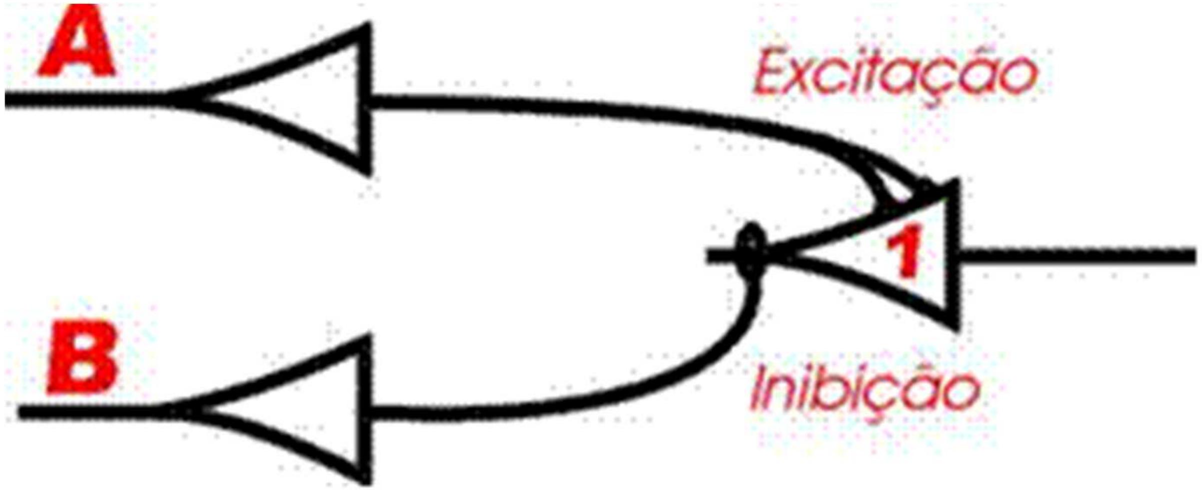


Modelo de McCullough and Pitts

- Hipóteses do Modelo:
  - O neurônio é bi-estável (saída 0 ou 1);
  - Há um número fixo de sinapses excitatórias que precisam receber estímulos para ativar o neurônio;
  - O atraso devido à sinapse é o único significativo;
  - Ativação de uma sinapse inibitória impede (inibe) ativação de um neurônio;
  - A estrutura do neurônio não muda com o tempo.

Modelo de McCullough and Pitts

- Proposta de cálculo lógico para descrever neurônios e redes, onde:
  - Todas as sinapses excitatórias têm o mesmo peso.
  - Todo neurônio é ativado por número fixo de sinapses.
  - Todo neurônio computa função lógica da entrada (função limiar).
  - A rede pode ser construída para computar qualquer função arbitrária.



Teoria de Hebb

- Modelo teórico de como os neurônios atuam foi porposto no livro de Hebb (1949), *The Organization of Behavior*.

Donald Olding Hebb
1904-1985

- Crescimento das Sinapses: mudanças nos valores das conexões.

"Quando o axônio de uma célula A está próximo o suficiente para excitar uma célula B e repetida e insistentemente toma parte na emissão de sinal elétrico da célula B, algum processo de crescimento ou mudança metabólica acontece em ambas células tal que a eficiência de A, para fazer a célula B disparar, é aumentada".

U F P E

Centro de Informática UFPE

O Perceptron

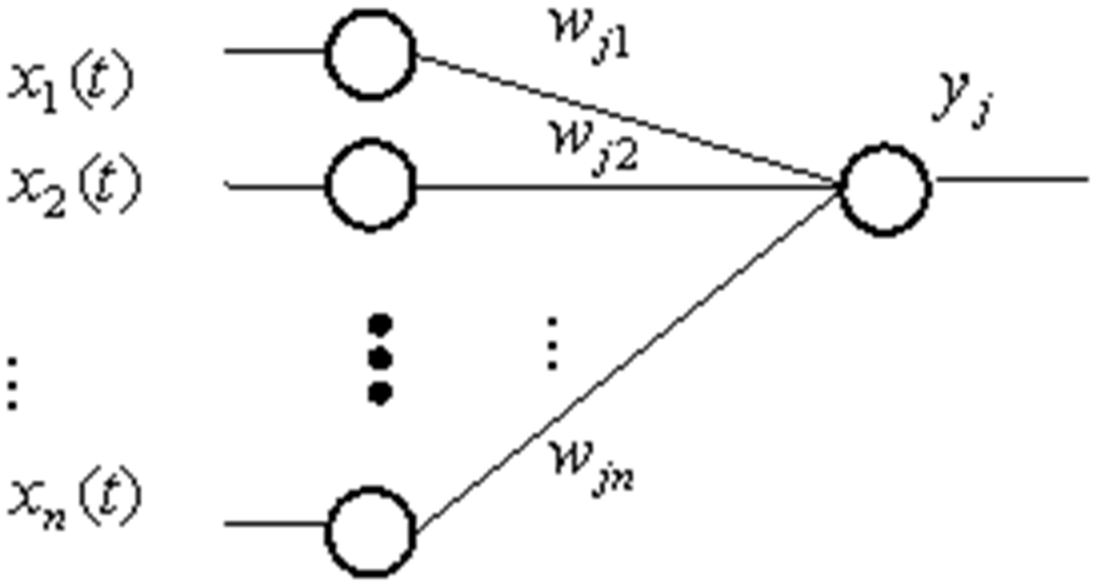
• Frank Rosenblatt (11/07/1928-1971), um neuro-cientista americano que estava vinculado à Cornell quando pesquisava sobre a operação do olho de uma mosca que realiza a maior parte do processamento que determina para onde a mosca deve fugir.

• Em 1957, o Perceptron, foi proposto durante esta pesquisa e foi implementado em hardware, tornando-se o primeiro modelo de rede neural artificial.

• Um Perceptron de camada única foi proposto como classificador de conjunto de padrões com valores contínuos em uma de duas classes.

O Perceptron

• A arquitetura de mapeamento de padrões chamada Perceptron objetiva aprender classificações de padrões através de treinamento supervisionado.



O Perceptron

- As entradas x_i , i = 1, 2, ..., n são binárias;
- Os pesos w_{ji} podem ser positivos ou negativos;
- Regra de propagação:

$$net_j = \sum_{i=1}^{n} w_{ji} . x_i$$

- A saída binária é determinada pela regra de ativação:

$$y_j = \begin{cases} 1 & \text{se } net_j \geq T_j \\ 0 & \text{se } net_j < T_j \end{cases}$$

UFPE

Centro de Informática UFPE

O Perceptron

- Nesta parte discute-se como treinar a rede. Isto é, discute-se como construir um mecanismo que vai absorver o conhecimento dentro da rede. Duas são as considerações básicas:
  - Em temos cognitivos existe uma tendência de se aprender o comportamento recompensado e se esquecer o comportamento penalizado.
  - Em termos “microscópios” ou de “microcognição” é necessário incluir o conceito de aprendizagem no mecanismo da rede.

O Perceptron

- O paradigma de aprendizagem pode ser descrito da seguinte maneira:
  - Considere valores de pesos e limiares (*thresholds*) iniciais;
  - Apresente uma entrada;
  - Calcule o efeito da entrada na saída;
  - Altere pesos para saídas indesejáveis;
- O Teorema da Convergência dos Perceptrons (Rosenblatt, 1958; Block, 1962; Novikoff, 1963) limita o número de erros que o algoritmo do perceptron pode cometer:

"Seja $(x_1,y_1),\ldots,(x_n,y_n)$ uma seqüência de exemplos rotulados com $x_i \in \Re^n, \|x_i\| \leq R$ e $y_i \in \{-1,1\}, \forall i.$ Seja $u \in \Re^n, \varepsilon > 0$, tal que $y_i(u \cdot x_i) \geq \varepsilon, \forall i.$ Então o perceptron comete no máximo $\left(R^2 \|u\|^2\right)/\varepsilon^2$ erros nesta sequência de exemplos.

O Perceptron

- Algoritmo de aprendizagem:

Inicialize pesos e limiares:

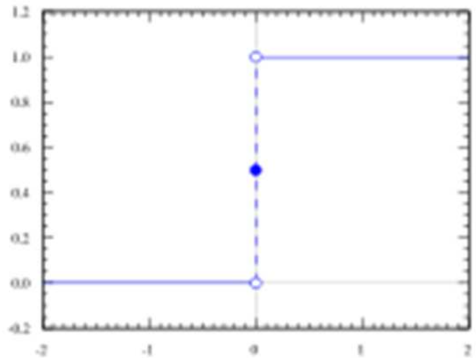
Atribua valores aleatórios para ω_{ji}, ($0 \leq i \leq n$) e $T_{j;}$ Como $j = 1$, índice some.

Apresente as entradas e as saídas desejadas:

Represente binariamente os vetores de entrada e saída;

Apresente a entrada (x_0, x_1, ..., x_n) e a saída alvo [$\boldsymbol{t}(t)$]

Calcule a saída pela Função de Heaviside em t:



$$y(t) = f_h\left[\sum_{i=1}^{n} w_i x_i(t)\right]$$

Recalcule os Pesos:

$$y(t) = 0, t(t) = 1 \Rightarrow w_{ji} = w_{ji} + x_i$$

$$y(t) = 1, t(t) = 0 \Rightarrow w_{ji} = w_{ji} - x_i$$

$$y(t) = t(t) \Rightarrow w_{ji} = w_{ji}$$

Centro de Informática UFPE

U F P E

Adaptive Linear Neurons

• A primeira modificação consiste de atenuar as modificações nos pesos no período de treinamento. Isto é conseguida através da introdução de fator multiplicativo da variação do peso.

• Substitua o passo de ajustar pesos:

$$y(t) = 0, t(t) = 1 \Rightarrow w_{ji} = w_{ji} + \eta x_i$$

$$y(t) = 1, t(t) = 0 \Rightarrow w_{ji} = w_{ji} - \eta x_i$$

$$y(t) = t(t) \Rightarrow w_{ji} = w_{ji}, \quad 0 < \eta \leq 1.$$

• Widrow e Hoff (1960) modificaram a regra acima de maneira que as variações nos pesos fossem proporcionais às diferenças entre a saída real e a desejada. Os pesquisadores propuseram se calcular a diferença entre as saídas mencionadas acima e chamá-la de ERRO.

Adaptive Linear Neurons

• Em 1959, Bernard Widrow (24/12/1929) e Marcian Edward "Ted" Hoff, Jr. (28/10/1937), de Stanford, desenvolveram modelos chamados ADALINE e MADALINE que receberam seus nomes devido ao uso de elementos lineares e adaptativos múltiplos (*Multiple ADAptive LINear Elements*). MADALINE foi a primeira RN usada em um problema do mundo real: filtro adaptativo para eliminar ecos em linhas telefônicas.

Adaptive Linear Neurons

- Adaline foi uma versão modificada do Perceptron, cuja regra de propagação de Adaline é:

$$net_j = \sum_{i=1}^{n} w_i x_i$$

- A regra de ativação (para uma representação binária) é:

$$y(t) = f_h\left[\sum_{i=0}^{n} w_i x_i(t)\right]$$

- Algoritmo de treinamento proposto por Widrow-Hoff atualiza os pesos com base em um erro entre a saída obtida e a desejada.

Adaptive Linear Neurons

- Algoritmo de treinamento proposto por Widrow-Hoff:
- Seja x_p um padrão com saída desejada e obtida t_p e y_p. Define-se o erro:

$$\delta_p = t_p - y_p$$

- Os pesos são ajustados para minimizar o erro $E_p = \frac{1}{2}\left\|\delta_p\right\|^2$

- Este erro varia com relação a cada um dos pesos:

$$\frac{\partial E_p}{\partial w_i} = \frac{\partial}{\partial w_i}(y_p - t_p) = \delta_i . x_i$$

- A regra de Widrow-Hoff ou regra Delta ou regra Least-Means-Square (LMS):

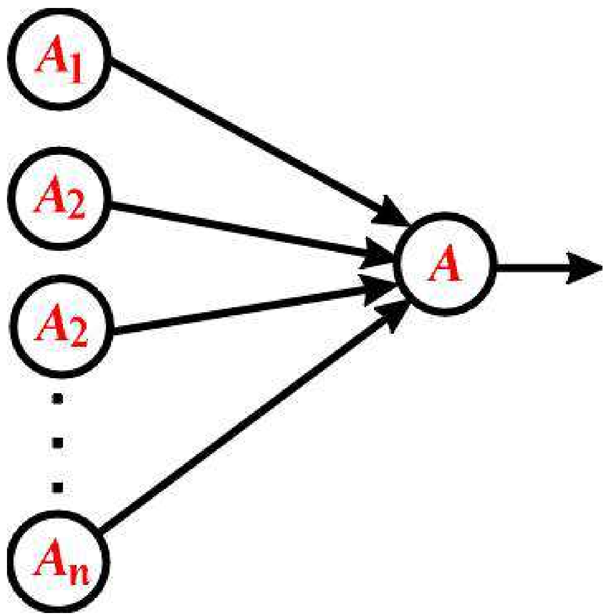
$$w_i = w_i + \eta \frac{\partial E_p(t)}{\partial \omega_i(t)} \therefore w_i = w_i + \eta\, \delta_i\, x_i$$

Centro de Informática UFPE

Adaptive Linear Neurons

- Variações do modelo ADALINE:
  - HARDWARE: implementada no computador analógico.
  - SOFTWARE: simulações num IBM 1620 até 1000 pesos.

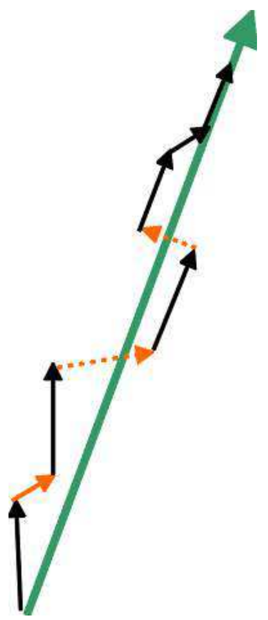
- MADALINE: Conjunto de ADALINES que lançam suas respostas em uma ADALINE fixa.
  - A ADALINE fixa atua com voto de maioria: Se mais que a metade das saídas das ADALINEs são +1 a saída da MADALINE também o é.
  - Pode resolver problemas complexos, mas não se provou convergência.



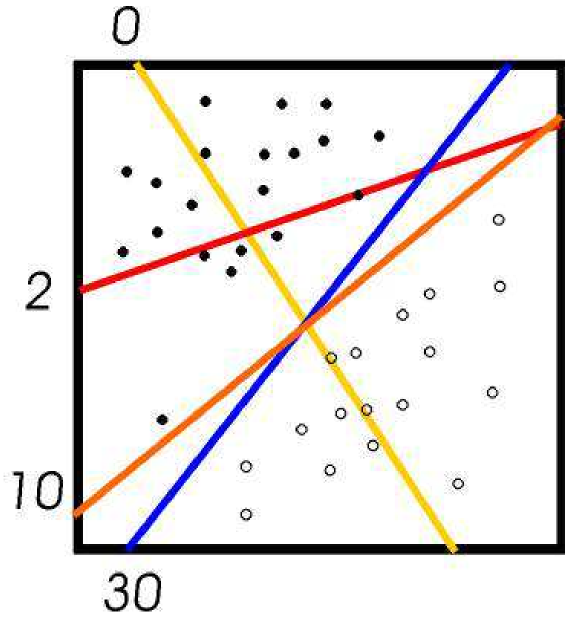
Centro de Informática UFPE

Perceptron

- Pode-se entender o procedimento de aprendizagem do Perceptron observando a evolução do vetor peso no tempo.



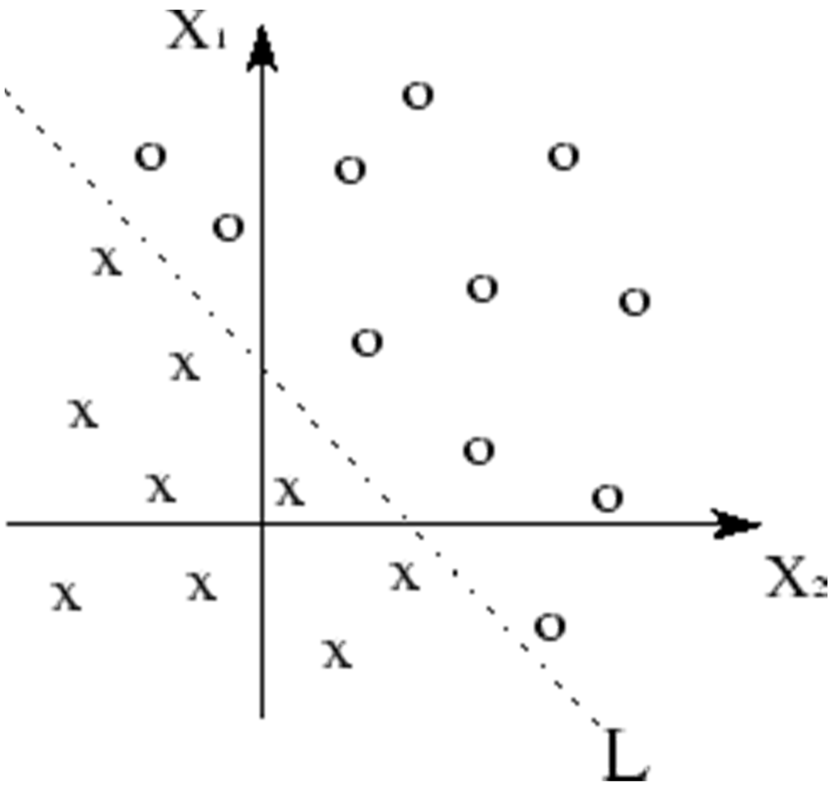
Comportamento do vetor de pesos no espaço de padrões



Evolução da "Linha de Classificação" ()

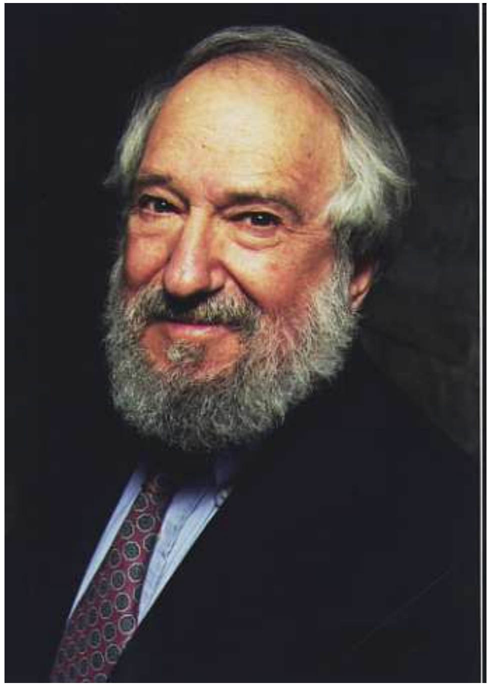
Perceptron

- Separabilidade linear: Separação linear de dois conjuntos de padrões pertencentes a classe diferentes.
- Limitação do perceptron com respeito à separabilidade linear.



Perceptron

Marvin Lee *Minsky*
(09/08/1927)

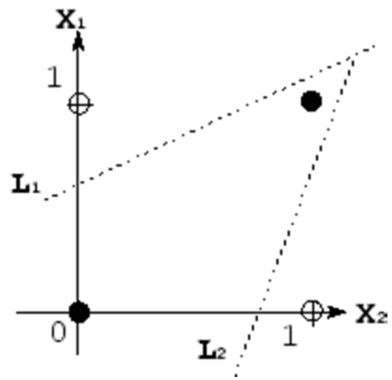


Seymour Papert
(29/02/1928)

- O perceptron não pode aprender exemplos que não sejam linearmente separáveis tais como a porta XOR.

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

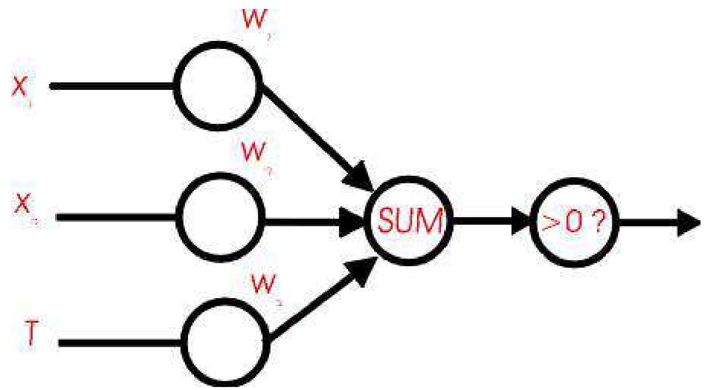
$Y = X_1 \oplus X_2$



Exemplos

- EXEMPLO 1:
  - Um Perceptron deve ser treinado para reconhecer a porta lógica **OR**.
  - As condições iniciais de treinamento são: $w_1 = 0; w_2 = 0; w_3 = 0; T = 1$
- As amostras e as saídas da porta lógica OR são

AMOSTRA	**X_1**	**X_2**	**T**	**SAÍDA DESEJADA**
1	0	0	1	0
2	0	1	1	1
3	1	0	1	1
4	1	1	1	1



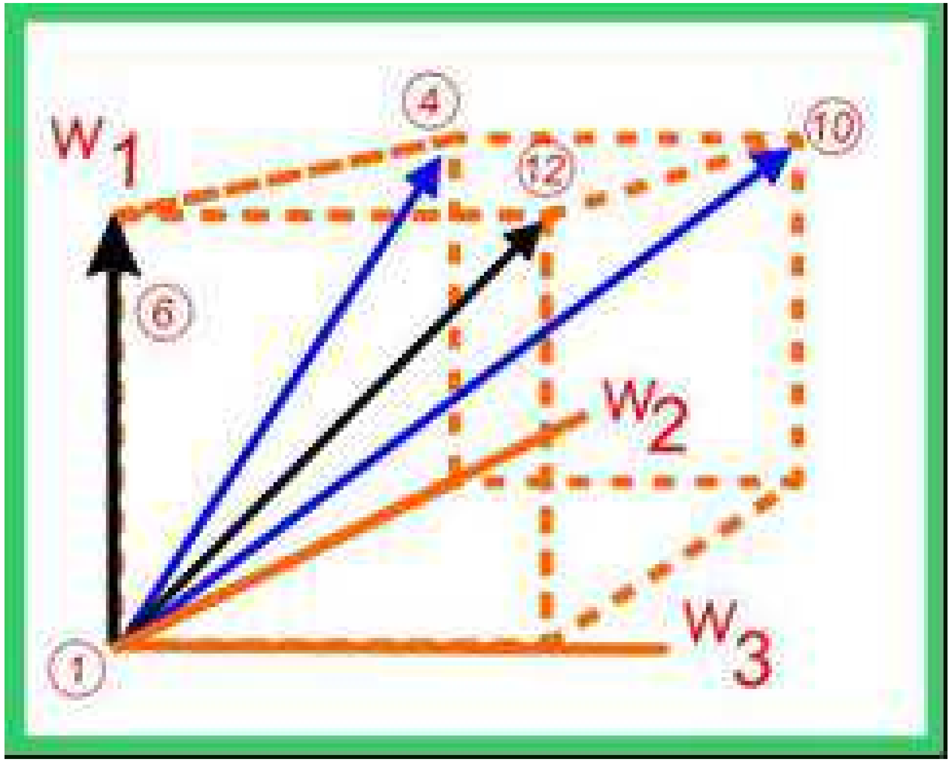
Exemplos

- Seqüência de treinamento:

ESTÁGIO	INST. DE TEMPO	ENTRADAS (X_1, X_2, T)	PESOS (W_1, W_2, W_3)	SUM	SAÍDAS REAL	SAÍDAS DES.
I	1 2 3	(0 0 1) (0 1 1) (0 1 1)	(0 0 0) (0 0 0) (0 0 0) + (0 1 1)	0 0 2	0 0 1	0 1 1
II	4 5	(0 0 1) (0 0 1)	(0 1 1) (0 1 1) – (0 0 1)	1 0	1 0	0 0
III	6 7 8 9	(0 0 1) (0 1 1) (1 0 1) (1 0 1)	(0 1 0) (0 1 0) (0 1 0) (0 1 0) + (1 0 1)	0 1 0 2	0 1 0 1	0 1 1 1
IV	10 11 12 13 14	(0 0 1) (0 0 1) (0 1 1) (1 0 1) (1 1 1)	(1 1 1) (1 1 1) - (0 0 1) (1 1 0) (1 1 0) (1 1 0)	1 0 1 1 2	1 0 1 1 1	0 0 1 1 1

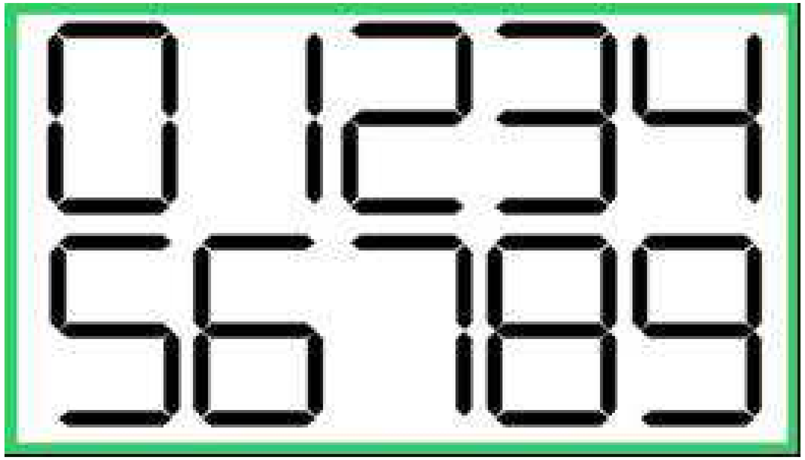
Exemplos

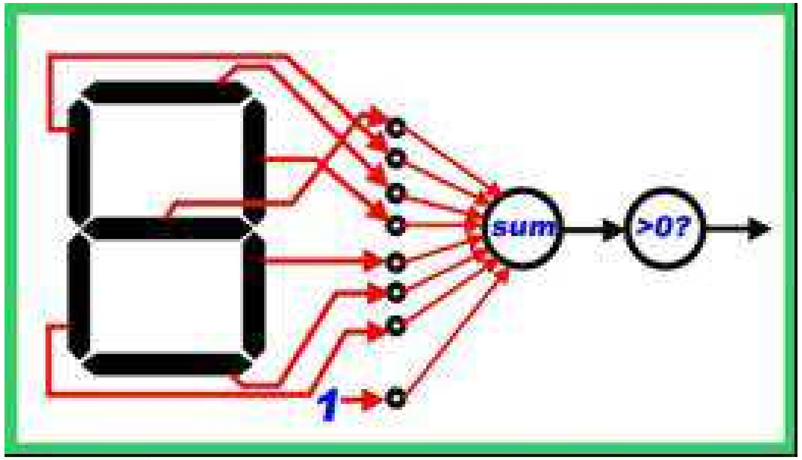
- EXEMPLO 1:
  - Variação dos pesos durante o treinamento:



Exemplos

- EXEMPLO 2:
  - Reconhecimento de dígitos comumente usados em displays digitais. Tais dígitos são resultado de uma combinação apropriada de segmentos como em sete possibilidades como mostrada na figura ao lado.
- Um sistema de visualização identifica os estados de ativação dos segmentos e estes estados são entradas para um perceptron como na figura ao lado.





Exemplos

➢ Representação das entradas:

X_0	X_1	X_2	X_3	X_4	X_5	X_6	Dígito
0	1	1	1	1	1	1	0
1	1	1	1	1	0	1	9
1	1	1	1	1	1	1	8
0	0	1	1	1	0	0	7
1	1	1	0	1	1	1	6
1	1	1	0	1	1	0	5
1	1	0	1	1	1	0	4
1	0	1	1	1	1	0	3
1	0	1	1	0	1	1	2
0	0	0	1	1	0	0	1

Exemplos

- EXEMPLO 2:
  - Cada um dos dígitos é reconhecido pela “rede”. Logo, para treinar um Perceptron reconhecedor de dígitos 1, só a última linha produziria saída 1, enquanto todas as demais produziriam saída 0.
  - Para identificar o número 0 (zero) só duas mudanças são necessárias.

Entrada	Pesos	Som	Saída	Resposta Correta
(0 1 1 1 1 1 1 1)	(0 0 0 0 0 0 0 0)	0	0	1
(0 1 1 1 1 1 1 1)	(0 1 1 1 1 1 1 1)	7	1	1
(1 1 1 1 1 0 1 1)	(0 1 1 1 1 1 1 1)	7	1	0
(0 1 1 1 1 1 1 1)	(-1 0 0 0 0 0 1 0)	1	1	1

  - Com dois seguimentos (0 – inativo e 6 – ativo) o zero é identificado.
  - São observadas 65 mudanças para identificar o dígito 8, resultando nos pesos (3 3 0 6 –1 –7 4 –7).

UFPE

Centro de Informática UFPE