Aprendizagem por treinamento de perceptrons

Marco Henrique Terra

Introdução à Inteligência Artificial

Introdução

- Perceptrons podem ser vistos como um tipo especial de rede neural. Um perceptron é a mais simples rede neural.
- Um perceptron possui apenas um neurônio.
- Será visto um exemplo de como um perceptron aprende a identificar digitos mostrados em um mostrador digital de sete segmentos.
- O objetivo deste capítulo é saber como os perceptrons trabalham, entender como o procedimento de convergência do perceptron melhora o desempenho dele. Também, as limitações de um perceptron simples serão ressaltadas.

Perceptrons e aprendizagem perceptron

- Um perceptron é uma representação que é uma rede neural na qual:

  - Existe somente um neurônio.

  - As entradas são binárias: recebem valores 0 ou 1.

  - Caixas lógicas podem ser interpostas entre as entradas dos perceptrons e os pesos dos perceptrons. Cada caixa lógica pode ser vista como uma tabela que produz um valor de saída 0 ou 1 para cada combinação de 0s ou 1s que podem aparecer nas entradas.

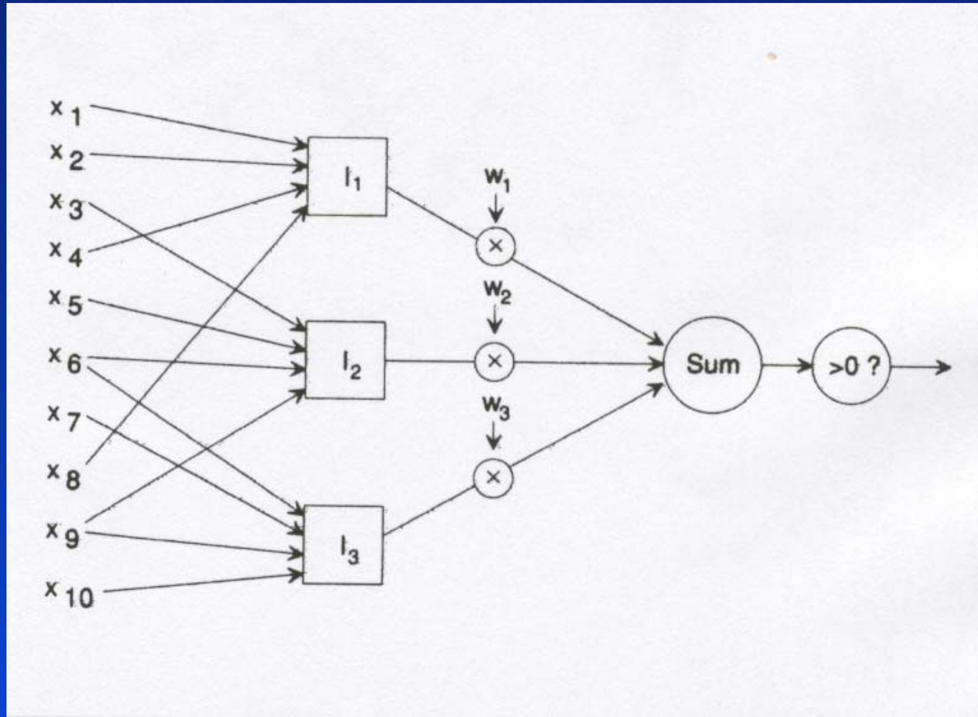
  - A saída do perceptron é 0 ou 1 dependendo se a soma ponderada das saídas das caixas lógicas é maior que o limiar.

- Suponha que a saída da *i*-ésima caixa lógica é l_i suponha também que o *i*-ésimo peso seja w_i e o limiar seja T , a saída do perceptron P é dada pela seguinte fórmula

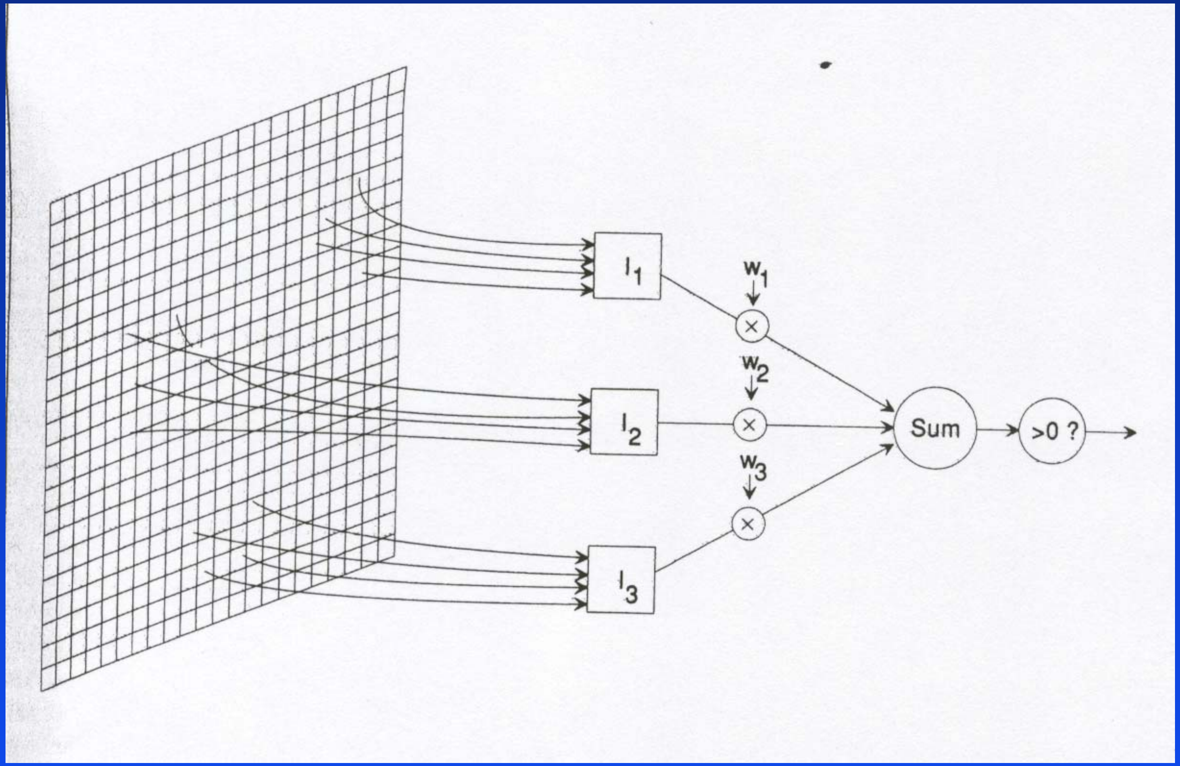
$$P = 1 \text{ se } \sum_{\mathrm{i}} w_i\, l_i > T \qquad \textit{ou}$$

$$P = 0 \text{ em caso contrário.}$$

- Note que cada caixa lógica de um perceptron está conectada somente em um número inferior ao total de entradas isto para evitar uma sobrecarga de combinações a serem consideradas, que poderia ser de 2^n sendo n o número de entradas. Veja as Figuras 1 e 2.

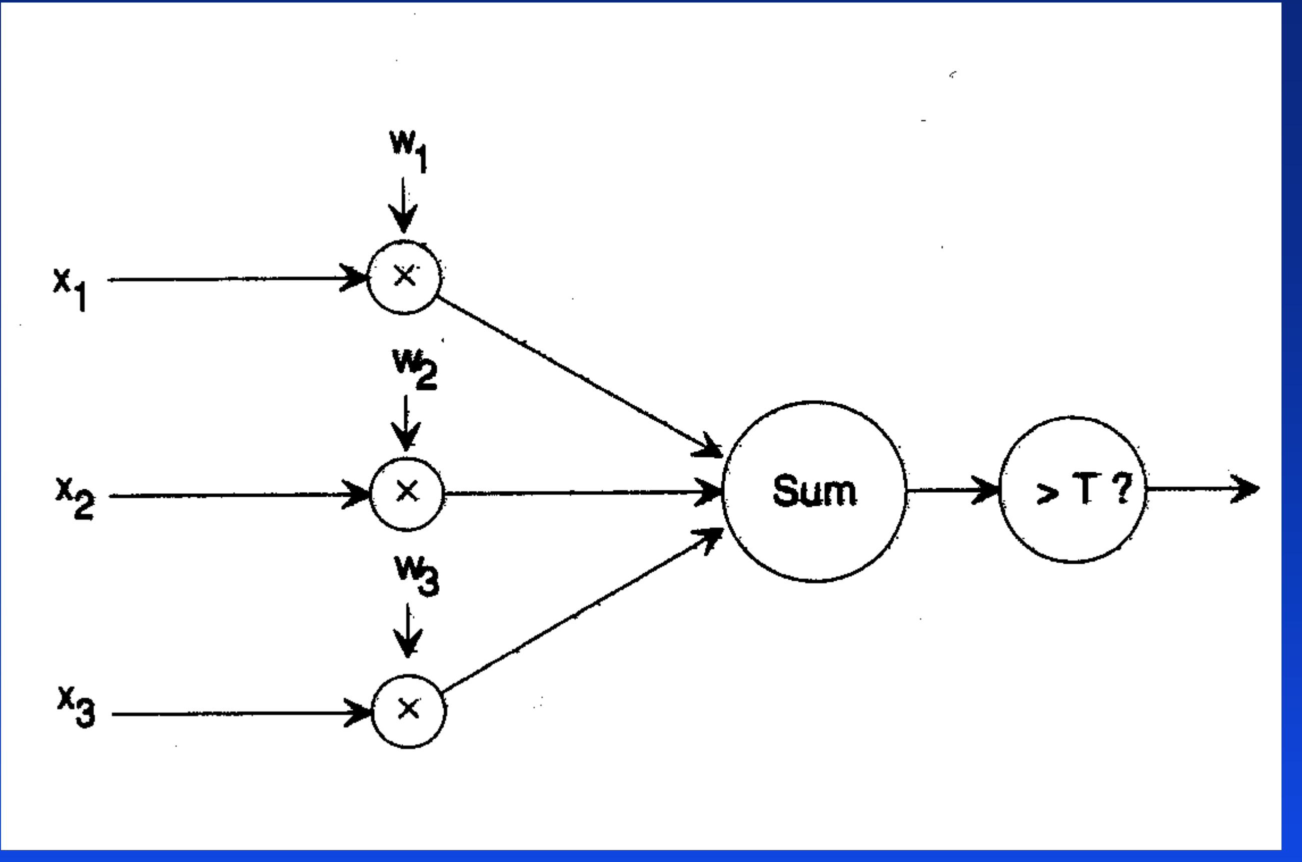


- Figura 1. Descrição de um perceptron. As entradas assumem valores 0 ou 1, também as saídas das caixas lógicas assumem valores 0 ou 1. Se a soma das saídas ponderadas das caixas lógicas é maior que 0, a saída do perceptron é 1 e o perceptron diz sim, ou seja, reconhece uma classe. Caso contrário o perceptron diz não, ou seja, não reconhece uma classe.



- Figura 2: Outro exemplo de perceptron. Neste caso as entradas estão organizadas de maneira retangular mapeando um padrão de duas dimensões.

- Na Figura 2, as entradas do perceptron são organizadas em uma matriz retangular sugestivamente chamada de retina com a finalidade de reconhecer um padrão de duas dimensões como por exemplo um caracter alfanumérico.
- Duas maneiras de se organizar um perceptron são definidas a seguir:
- Perceptron de diâmetro limitado *d*, as entradas são organizadas de maneira retangular, matriz do tipo retina, e todas as entradas para qualquer caixa lógica particular deve estar dentro de um círculo de diâmetro *d* e
- Perceptron direto, cada caixa lógica possui somente uma entrada.
- De maneira alternativa um perceptron direto pode ser visto como um perceptron sem caixas lógicas, veja Figura 3. Os exemplos a seguir consideram perceptrons do tipo direto porque os exemplos de aprendizagem são mais fáceis de serem gerados.



- Figura 3: Perceptron do tipo direto.

- O procedimento de convergência do perceptron garante sucesso se o sucesso for possível

- Para treinar um perceptron proceda da seguinte maneira:
  - Até que o perceptron gere um resultado correto para cada amostra de treinamento, para cada amostra verifique se
    - o perceptron fornece uma resposta errada;
      - se o perceptron diz não quando ele deveria dizer sim, some o vetor de saída da caixa lógica para o vetor de peso
      - caso contrário, subtraia o vetor de saída da caixa lógica do vetor peso
    - o perceptron fornce uma resposta certa, então não faça nada.

- Diz-se que um perceptron diz não se a saída do perceptron assumir valor 0 e o perceptron diz sim quando a saída valer 1.
- Considere por exemplo o perceptron da Figura 4 e suponha que se quer treiná-lo para ele funcionar como uma porta lógica Ou.
- Em virtude deste perceptron ser do tipo direto , o vetor de saída da caixa lógica, (l_1 l_2 l_3), é o mesmo que o vetor de entrada (x_1 x_2 x_3). Assim, a amostras das entradas e as saídas correspondentes das caixas lógicas são as seguintes para a lógica Ou:

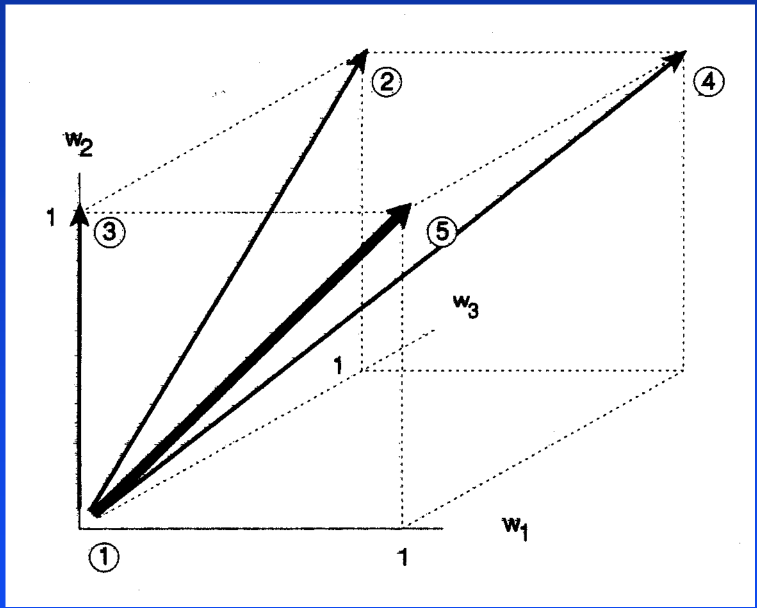
Amostra	x1 = I1	x2 = I2	x3= I3= 1	saída des.
1	0	0	1	0
2	0	1	1	1
3	1	0	1	1
4	1	1	1	1

- Tabela 1. Dados para o treinamento de uma função lógica Ou.

- Através das informações da Tabela 1, o perceptron eventualmente aprende a função lógica ou através de quatro mudanças.
- A primeira mudança ocorre seguindo um erro na segunda amostra durante o primeiro passo. Inicia-se o processo com um vetor peso nulo (0 0 0), desta forma a saída é 0, quando ela deveria ser 1.
- Então, seguindo o procedimento indicado, quando o vetor de entrada é (0 1 1), ele é somado ao vetor peso, o novo vetor peso passa a ser (0 1 1).
- As próximas duas entradas encontradas durante o primeiro passo produzem 1 na saída, como de fato deveriam ser, então não se faz mais mudanças durante o primeiro passo.
-

- Durante o segundo passo, a primeira amostra considerada (0 1 1) produz valor 1, mas ela deveria ser 0. Então, o vetor de entrada, (0 0 1) é subtraído do vetor de pesos, (0 1 1), produzindo um novo vetor (0 1 0). Com esta mudança, a terceira amostra a ser considerada produz um 0 quando ela deveria produzir um valor 1. Então o vetor de entrada (1 0 1) é somado ao vetor peso (0 1 0) , produzindo um novo vetor (1 1 1).
- Durante o terceiro passo (quando a amostra (1 0 1) está sendo considerada), a primeria amostra produz um erro novamente. O resultado deveria ser 0 mas ela produz um valor 1. Subtraindo o vetor de entrada, (0 0 1), do vetor de pesos, (1 1 1), tem-se o vetor (1 1 0), que funcionará para todas as amostras.

- Em virtude deste exemplo apresentar apenas três amostras, duas para as entradas e uma para o limiar, cada combinação de peso pode ser vista como um vetor tri-dimensional, como mostrado na Figura 4. Em geral o vetor de pesos aumenta e diminui na medida em que a aprendizagem evolui.



- Figura 4. Evolução do treinamento do perceptron, o vetor peso inicia com todos os componentes nulos. Durante o treinamento, o tamanho do vetor de pesos aumenta e diminui.

- A álgebra de vetores auxilia na demonstração da convergência do perceptron

- Seja **w** o vetor de pesos e **l** o vetor das saídas das caixas lógicas.
- Considera-se que exista um vetor $\mathbf{w}^*$ tal que

$$\mathbf{w}^* \, \mathbf{l} > \delta$$

se **l** é produzido pelas entradas que supostamente deveriam fazer com que o perceptron dissesse sim (saída 1), e

$$\mathbf{w}^* \, \mathbf{l} < \delta$$

em caso contrário.

- O coseno do ângulo entre dois vetores $\mathbf{w}^*$ e $\mathbf{w}_n$ é definido como segue:

$$\cos\theta = \frac{\mathbf{w}^* \cdot \mathbf{w}_n}{\|\mathbf{w}^*\| \|\mathbf{w}_n\|}.$$

- Suponha que o erro que causou a mudança fosse um erro para o qual o perceptron tivesse dito não (0) quando ele deveria ter dito sim (1). Este erro acontece quando

$$\mathbf{w}_{n-1} \cdot \mathbf{l} \leq 0,$$

- que resulta na seguinte mudança para o vetor peso

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{l}.$$

- O numerador da função coseno fica da seguinte forma

$$\mathbf{w}^* \cdot \mathbf{w}_n = \mathbf{w}^* \cdot (\mathbf{w}_{n-1} + \mathbf{l}) = \mathbf{w}^* \cdot \mathbf{w}_{n-1} + \mathbf{w}^* \cdot \mathbf{l}.$$

- Em virtude de que $\mathbf{w}^*$ sempre produz resultados corretos, sabe-se que $\mathbf{w}^*\ \mathbf{l} > \delta$, dado que o perceptron deve dizer sim. Assim o numerador da expressão coseno possui a seguinte restrição:

$$\mathbf{w}^* \cdot \mathbf{w}_n > \mathbf{w}^* \cdot \mathbf{w}_{n-1} + \delta.$$

- e consequentemente

$$\mathbf{w}^* \cdot \mathbf{w}_n > n\delta.$$

- Vamos verificar agora em que região o denominador do coseno está restrito

$$\|\mathbf{w}^*\|\|\mathbf{w}_n\|.$$

- Rescrevendo a norma de $\mathbf{w}_n$ ao quadrado tem-se que

$$\begin{aligned}\|\mathbf{w}_n\|^2 &= \mathbf{w}_n \cdot \mathbf{w}_n \\ &= (\mathbf{w}_{n-1} + \mathbf{l}) \cdot (\mathbf{w}_{n-1} + \mathbf{l}) \\ &= \|\mathbf{w}_{n-1}\|^2 + 2\mathbf{w}_{n-1} \cdot \mathbf{l} + \|\mathbf{l}\|^2.\end{aligned}$$

- $\mathbf{w}_{n-1} \cdot \mathbf{l}$ deve ser 0, ou menor que zero em virtude de que o evento que iniciou a mudança no peso foi o perceptron dizendo não quando ele deveria dizer sim, assim tem-se a seguinte restrição

$$\|\mathbf{w}_n\|^2 \le \|\mathbf{w}_{n-1}\|^2 + \|\mathbf{l}\|^2.$$

- E consequentemente

$$\|\mathbf{w}_n\|^2 \le n\|\mathbf{l}\|^2.$$

- Em virtude dos elementos de **l** serem 0 ou 1, $||l||^2$ não pode ser maior que o número de portas lógicas, #**l**. A expressão do denominador da função coseno está restrita à seguinte expressão

$$\|\mathbf{w}^*\|\|\mathbf{w}_n\| \le \|\mathbf{w}^*\|\sqrt{n}\sqrt{\#\mathbf{l}}.$$

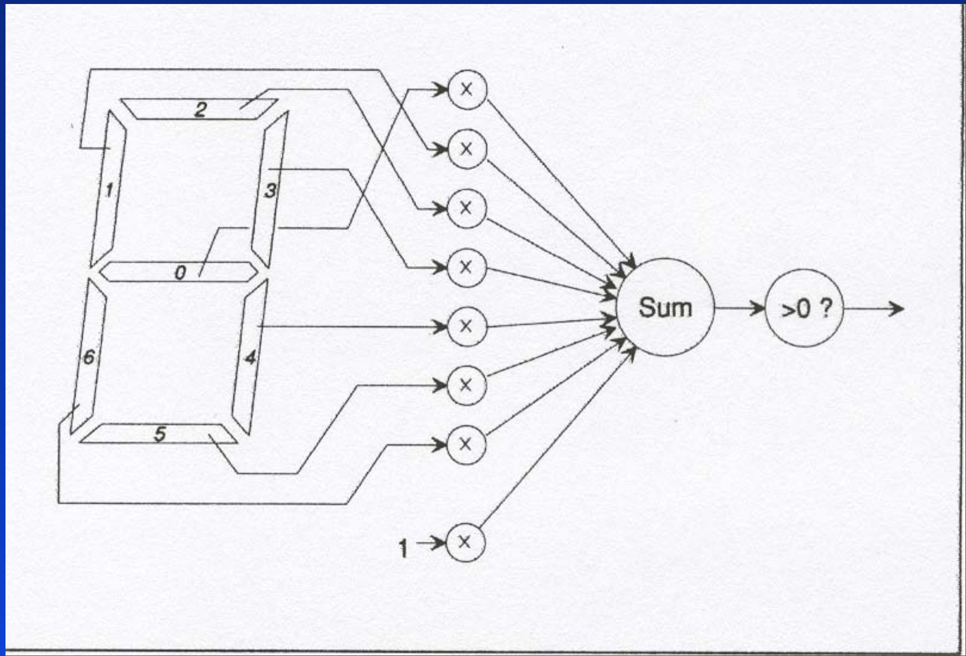
- Substituindo estas restrições, o expressão do coseno fica definida da seguinte maneira

$$\cos\theta > \frac{\sqrt{n}\delta}{\|\mathbf{w}^*\|\sqrt{\#I}}.$$

- Nota-se que o limite inferior da função coseno do ângulo entre $\mathbf{w}^*$ e $\mathbf{w}_n$ deve aumentar em cada mudança, e que o limite inferior é proporcional a $\sqrt{n}$.

- Um perceptron direto pode aprender a indentificar dígitos

- Suponha que um sistema de visão hipotético está apto para identificar qual dos sete segmentos em um mostrador está ligado, com isto ele fornece as entradas para o perceptron direto mostrado na Figura 5.



- Figura 5. Um perceptron para reconhecer dígitos. Sete das entradas conectadas aos sete segmentos dos dígito, a oitava está conectada a uma entrada que está sempre em 1, que substitui um limiar.

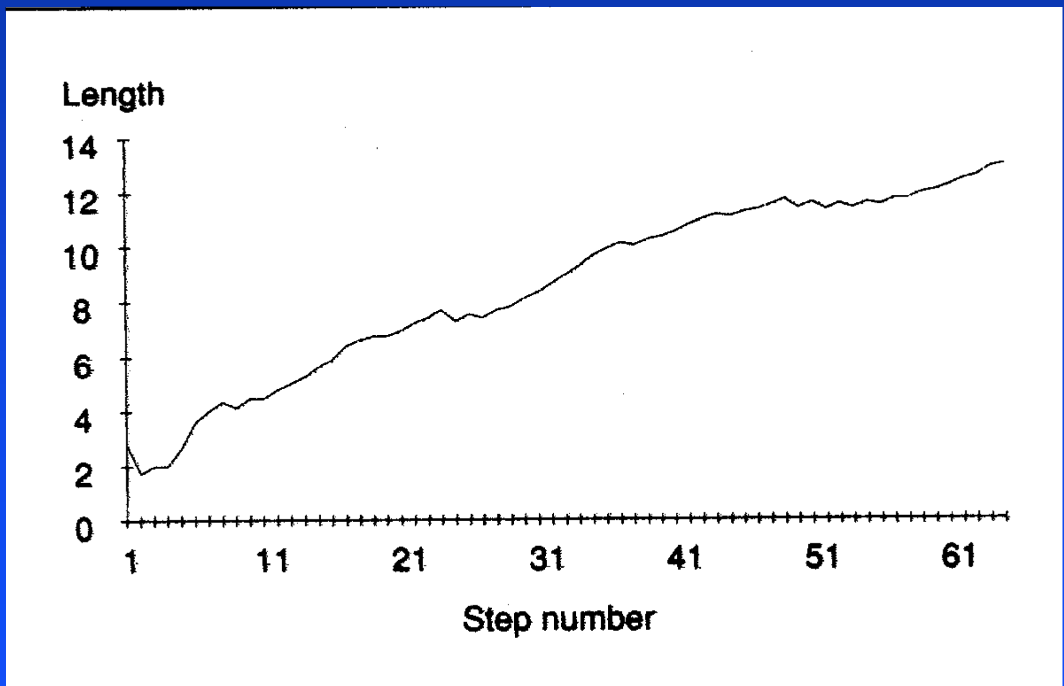
- As entradas dos sete segmentos, mais a entrada 1, forncem oito entradas com os correspondentes oito pesos que podem ser alterados. As amostras estão definidas na tabela 2 a seguir.

Digit	x_0	x_1	x_2	x_3	x_4	x_5	x_6
0	0	1	1	1	1	1	1
9	1	1	1	1	1	1	0
8	1	1	1	1	1	1	1
7	0	0	1	1	1	1	0
6	1	1	1	0	1	1	1
5	1	1	1	0	1	1	0
4	1	1	0	1	1	1	0
3	1	0	1	1	1	1	0
2	1	0	1	1	0	1	1
1	0	0	0	1	1	0	0

- Para treinar um perceptron para reconhecimento de um dígito, para identificar o número 0 deve-se fazer com que o perceptron produza uma saída 1 apenas para a primeira linha da tabela acima.

- Para treiná-lo para identificar 1, deve-se informar ao perceptron que apenas para a última linha da tabela o valor da saída deverá ser 1.
- Quando o perceptron aprende a identificar 0, a convergência é rápida: somente duas mudanças são necessárias. Primeiro, porque o perceptron imediatamente falha para identificar o 0, a primeira amostra do primeiro passo, (0 1 1 1 1 1 1 1) é somada com todos os pesos iniciais nulos. Em seguida, porque o perceptron de maneira errada identifica o número 9, a segunda amostra neste primeiro passo, (1 1 1 1 1 1 0 1) é subtraída. O resultado é o vetor peso (-1 0 0 0 0 0 1 0), que é satisfatório, o 0 é identificado e todos os outros dígitos são rejeitados.
- Quando o perceptron tenta identificar o dígito 8 a convergência é a mais lenta de todas, veja a resposta do programa.

- A Figura 6 mostra o tamanho das mudanças do vetor de pesos com a aprendizagem do perceptron para aprender o dígito 8. Note que o tamanho diminui e aumenta até alcançar o peso apropriado.



- A Figura 7 mostra a evolução do ângulo entre o vetor de pesos e as mudanças dos valores finais dele com a evolução do aprendizado para reconhecer o dígito 8. Note que o valor do ângulo aumenta e diminui até chegar no valor desejado.