

CI5438. Inteligencia Artificial II

Clase 6: Redes Neuronales

Cap 20.5 Russel & Norvig

Cap 4 Mitchell

Ivette C. Martínez

Universidad Simón Bolívar

30 de enero de 2013

Modelos Conexionistas

Consideremos los seres humanos:

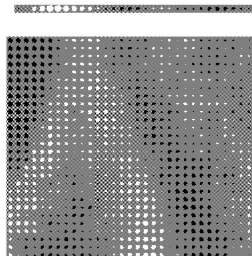
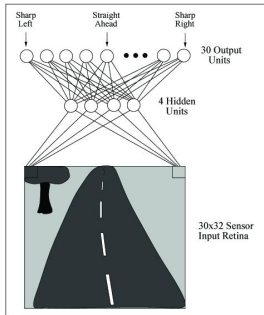
- Tiempo de activación de las neuronas $\approx 0,001$ segundos
- Número de neuronas $\approx 10^{10}$
- Conexiones por neurona $\approx 10^{4-5}$
- Tiempo para reconocer una escena $\approx 0,1$ segundos
- 100 pasos de inferencia no parecen ser suficientes

→ Muchos cálculos paralelos

Propiedades de las redes neurales artificiales:

- Muchas unidades de activación de umbral, como “neuronas”.
- Muchas interconexiones pesadas entre las unidades
- Procesamiento distribuido, altamente paralelo
- Énfasis en la entonación automática de pesos.

ALVINN



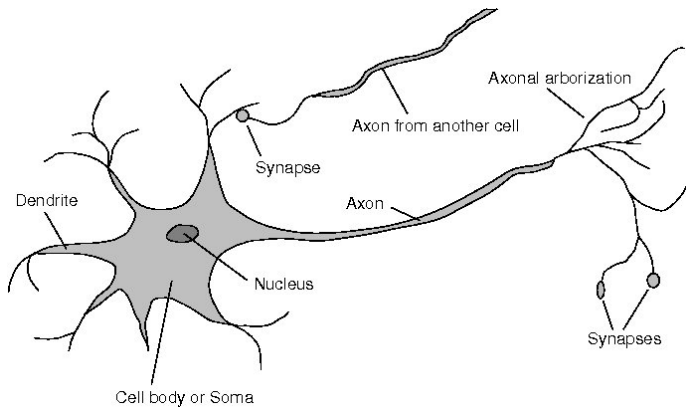
Cuándo considerar Redes Neurales

- La entrada es de muchas dimensiones, de valores discretos o reales (e.j. Entradas de un sensor)
- La salida es de valores discretos o continuos
- La salida es un vector de valores
- Los datos son posiblemente ruidosos
- La forma de la función objetivo es desconocida
- La legibilidad de los resultados por parte de los humanos no tiene importancia

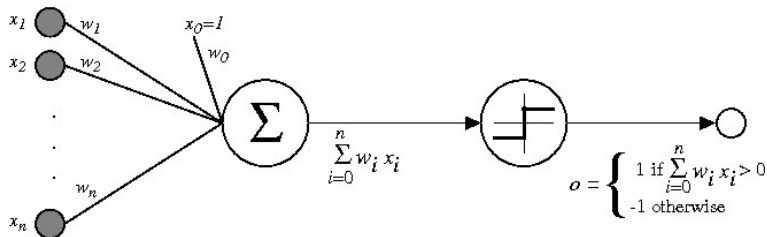
Ejemplos:

- Reconocimiento de fonemas hablados [Waibel]
- Clasificación de imágenes [Kanade, Baluja, Rowley]
- Predicción financiera

Neurona



Perceptrón

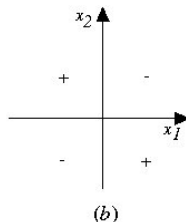
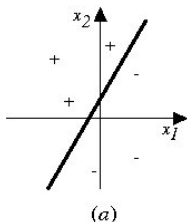


$$o(x_1, \dots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + \dots + w_n x_n > 0 \\ -1 & \text{otherwise} \end{cases}$$

Algunas veces usamos la notación vectorial, que es más simple:

$$o(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x} > 0 \\ -1 & \text{otherwise} \end{cases}$$

Superficies de Decisión de un Perceptrón



Se pueden representar algunas funciones útiles

- Qué pesos representan a

$$g(x_1, x_2) = AND(x_1, x_2)$$

Pero algunas funciones no son representables

- ejem., Funciones que no sean separable linealmente
- Luego, vamos a necesitar redes de estas unidades

Regla de entrenamiento del Perceptrón

$$w_i \leftarrow w_i + \Delta w_i$$

donde:

$$\Delta w_i = \eta(t - o)x_i$$

Donde:

- $t = c(\vec{x})$ es el valor objetivo
- o es la salida del perceptrón
- η es una constante pequeña (ejem., 0,1) denominada tasa de aprendizaje

Regla de entrenamiento del Perceptrón

Se puede demostrar que converge:

- Si los datos de entrenamiento son linealmente separables
- y η es lo suficientemente pequeña

Descenso del Gradiente

Para entender, consideremos la unidad lineal más simple, donde:

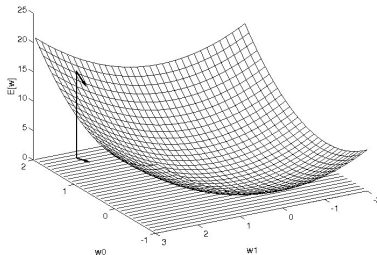
$$o = w_o + w_1x_1 + \dots + w_nx_n$$

Vamos a aprender los w_i que minimicen el error cuadrático medio

$$E[\vec{w}] \equiv \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

Donde D es el conjunto de ejemplos de entrenamiento

Descenso del Gradiente



$$\nabla E[\vec{w}] \equiv \left[\frac{\delta E}{\delta w_0}, \frac{\delta E}{\delta w_1}, \dots, \frac{\delta E}{\delta w_n} \right]$$

Regla de entrenamiento:

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

es decir:

$$\Delta w_i = -\eta \frac{\delta E}{\delta w_i}$$

Descenso del Gradiente

$$\begin{aligned}\frac{\delta E}{\delta w_i} &= \frac{\delta}{\delta w_i} \frac{1}{2} \sum_d (t_d - o_d)^2 \\ &= \frac{1}{2} \sum_d \frac{\delta}{\delta w_i} (t_d - o_d)^2 \\ &= \frac{1}{2} \sum_d 2(t_d - o_d) \frac{\delta}{\delta w_i} (t_d - o_d) \\ &= \sum_d (t_d - o_d) \frac{\delta}{\delta w_i} (t_d - \vec{w} \cdot \vec{x}_d) \\ \frac{\delta E}{\delta w_i} &= \sum_d (t_d - o_d) (-x_{i,d})\end{aligned}$$

Descenso del Gradiente

GRADIENT-DESCENT(*training_examples*, η)

Each training example is a pair of the form $\langle \vec{x}, t \rangle$, where \vec{x} is the vector of input values, and t is the target output value. η is the learning rate (e.g., .05).

- Initialize each w_i to some small random value
- Until the termination condition is met, Do
 - Initialize each Δw_i to zero.
 - For each $\langle \vec{x}, t \rangle$ in *training_examples*, Do
 - * Input the instance \vec{x} to the unit and compute the output o
 - * For each linear unit weight w_i , Do
$$\Delta w_i \leftarrow \Delta w_i + \eta(t - o)x_i$$
 - For each linear unit weight w_i , Do
$$w_i \leftarrow w_i + \Delta w_i$$

La regla de entrenamiento del perceptrón tiene un éxito garantizado si:

- Los ejemplos de entrenamiento son linealmente separables
- La tasa de aprendizaje η es suficientemente pequeña

La regla de entrenamiento de unidades lineales usando descenso del gradiente:

- Garantiza converger a la hipótesis con el menor error cuadrático medio
- Dada una tasa de aprendizaje η es suficientemente pequeña
- Aún cuando los datos de entrenamiento contengan ruido
- Aún cuando los datos de entrenamiento no sean separable por H

Descenso del Gradiente Incremental (Estocástico)

Batch mode Gradient Descent:

Do until satisfied

1. Compute the gradient $\nabla E_D[\vec{w}]$
 2. $\vec{w} \leftarrow \vec{w} - \eta \nabla E_D[\vec{w}]$
-

Incremental mode Gradient Descent:

Do until satisfied

- For each training example d in D
 1. Compute the gradient $\nabla E_d[\vec{w}]$
 2. $\vec{w} \leftarrow \vec{w} - \eta \nabla E_d[\vec{w}]$
-

$$E_D[\vec{w}] \equiv \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

$$E_d[\vec{w}] \equiv \frac{1}{2} (t_d - o_d)^2$$

Incremental Gradient Descent can approximate
Batch Gradient Descent arbitrarily closely if η
made small enough