

Redes de Neuronas no Supervisadas

Aprendizaje Automático

Ingeniería Informática

Fernando Fernández Rebollo y Daniel Borrajo Millán

Grupo de Planificación y Aprendizaje (PLG)
Departamento de Informática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

27 de febrero de 2009

En Esta Sección:

- 8 Aprendizaje no Supervisado
 - Introducción
 - Métodos Paramétricos
 - Métodos No Paramétricos

- 9 Mapas Auto-organizativos
 - Introducción
 - Mapas Auto-organizativos
 - Ejemplos y Aplicaciones

Aprendizaje No Supervisado

- Problemas para los que sólo están disponibles los patrones de entrada: no existe salida deseada
- Ejemplos:
 - Agrupación o *clustering*: obtención de grupos (clases) de entre los datos
 - Extracción de características: obtención de características a partir de los datos de entrada
 - Reducción de dimensionalidad: los datos de entrada son agrupados en subespacios de una dimensión más baja que la inicial

Aprendizaje Competitivo

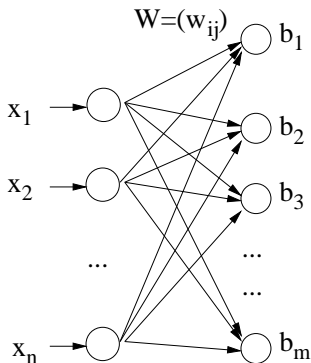
- La adaptación de los pesos, generalmente, se basa en la competición entre las neuronas
 - Neurona ganadora: aquella neurona que reciba más activación, es decir, esté más “cercana” al patrón de entrada
 - Adaptación de los pesos de la neurona ganadora: acercar dichos pesos hacia el patrón de entrada

Introducción

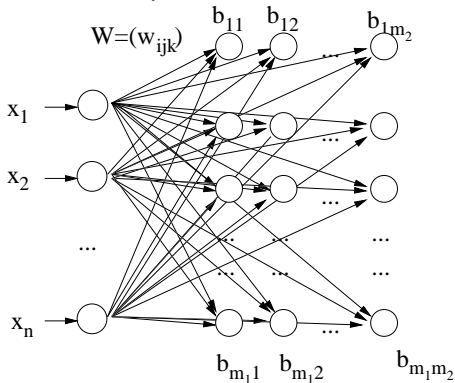
- Desarrollados por Teuvo Kohonen [?]
- Red de dos únicas capas:
 - Capa de entrada: recibe los patrones $\vec{x} = (x_1, \dots, x_n)$
 - Capa de salida: cada neurona se activa en función de la entrada, proporcionando una salida
- Número de neuronas en capas de entrada: depende de la dimensión de los datos (n)
- Número de neuronas en capa de salida (cada neurona representando un grupo o *cluster*):
 - Mapa lineal: m neuronas
 - Mapa bidimensional: $(m_1 \times m_2)$ neuronas

Arquitectura

Mapa Lineal

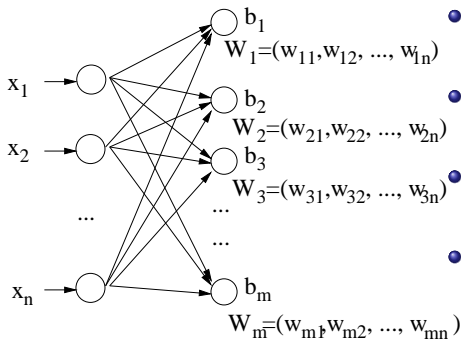


Mapa Bidimensional



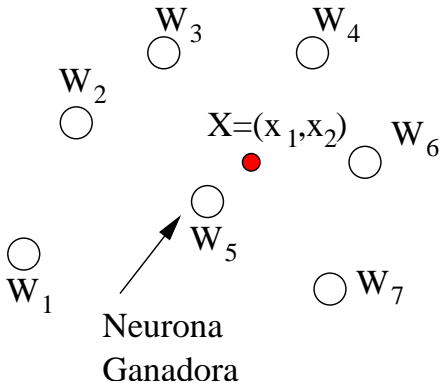
Pesos y Conexiones

- Los pesos se miran desde el punto de vista de las neuronas de la capa oculta
- Neurona i -ésima, pesos $\vec{W}_i = (w_{i1}, \dots, w_{in})$



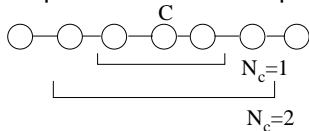
- Entrada de la red: X , puntos en espacio de \mathcal{R}^n
- Pesos de las conexiones: W_i , puntos en espacio de \mathcal{R}^n
- Activación de las neuronas: $b_i = d(X, W_i) = \sum_{j=1}^n \|x_j - w_{ij}\|$
- Neurona ganadora: aquella que minimice la distancia al patrón: $\arg_i \text{mín } d(X, W_i)$

Pesos y Conexiones: ejemplo

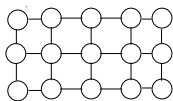


Vecindario

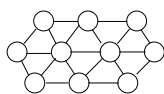
- Las neuronas no son elementos aislados, sino que se considera que están relacionadas con las neuronas de alrededor mediante el concepto de vecindario
- El vecindario (N_c) suele ser elegido como una región (cuadrada, hexagonal) alrededor de cada una de las neuronas, o como una función ($N_c(k)$), por ejemplo, gaussiana
- Es aplicable tanto a mapas lineales como bidimensionales



Mapa lineal



Topología Rectangular



Topología Hexagonal

Mapa bidimensional

Aprendizaje

- Mecanismo de cálculo de los pesos de la red
- Equivale a asociar las neuronas con posiciones del espacio adecuadas, donde dicha posición viene dada por los pesos de las neuronas
- Algoritmo:
 - Repetir k veces:
 - Para todos los patrones de entrada, X
 - 1 Calcular $d(X, W_j), \forall W_j$
 - 2 Obtener la neurona ganadora: $c = \arg_j \text{mín } d(X, W_j)$
 - 3 Adaptar los pesos de las neuronas:
 $W_j(t+1) = W_j(t) + a(k)(x - w_j(k))$, si j en vecindad de c
 $W_j(t+1) = W_j(t)$ en otro caso
- Tendencia del proceso de aprendizaje: cada neurona tiende a colocarse en el centroide de aquellos grupos de ejemplos para los cuales es la neurona ganadora.

Fases del Aprendizaje

- Se suelen definir dos fases:
 - Fase de ordenación: Se desea ordenar el mapa, Se plantean valores de $a(k)$ altos, que pueden ir reduciéndose en el proceso de aprendizaje. También se plantean valores altos para el radio del vecindario, que podría incluso contener a todas las neuronas.
 - Fase de convergencia: se ajustan las posiciones de las neuronas, asumiendo que el mapa está ordenado. Requiere bajos valores de $a(k)$ y del radio del vecindario

Ejemplos

- Som_Pak (<http://www.cis.hut.fi/research/som-research/nnrc-programs.shtml>)
- Herramienta para visualizar proceso de aprendizaje: xsom (<http://fuzzy.cs.uni-magdeburg.de/~borgelt/doc/somd/somd.html>)
- Agrupación de documentos y reducción de dimensionalidad: WEBSOM - Self-Organizing Maps for Internet Exploration (<http://websom.hut.fi/websom/>)

Bibliografía

- Pedro Isasi and Inés Galván. Redes de neuronas artificiales : un enfoque práctico. Pearson Prentice Hall, 2004.
- Teuvo Kohonen. Self-Organizing Maps. Springer, 1995

Parte IV

Aprendizaje por Refuerzo

Procesos de Decisión de Markov

Aprendizaje Automático

Ingeniería Informática

Fernando Fernández Rebollo y Daniel Borrajo Millán

Grupo de Planificación y Aprendizaje (PLG)
Departamento de Informática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

27 de febrero de 2009

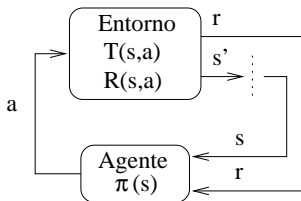
En Esta Sección:

- 10 Procesos de Decisión de Markov
 - Definición de Aprendizaje por Refuerzo
 - Procesos de Decisión de Markov
 - Definición de un MDP
 - Políticas y Optimalidad
 - Programación Dinámica

- 11 Aprendizaje por Refuerzo
 - Aprendizaje Por Refuerzo
 - Aproximaciones Libres de Modelo
 - Métodos Basados en el Modelo
 - Representación de la función Q
 - Generalización en Aprendizaje por Refuerzo
 - Discretización del Espacio de Estados
 - Aproximación de Funciones
 - Ejemplos de Aplicación

Aprendizaje por Refuerzo

- El aprendizaje por refuerzo consiste en aprender a decidir, ante una situación determinada, qué acción es la más adecuada para lograr un objetivo.
- Elementos principales:
 - Proceso iterativo de prueba y error
 - Aprendizaje a través de señales de refuerzo



Métodos de Resolución

- Si se conoce el modelo (funciones de transición de estado y refuerzo): Programación Dinámica
- Si no se conoce el modelo:
 - Aprender el modelo: métodos basados en el modelo o *Programación Dinámica*
 - Aprender las funciones de valor directamente: métodos libres de modelo o Aprendizaje por Refuerzo