

OPENCOURSEWARE
REDES DE NEURONAS ARTIFICIALES
Inés M. Galván – José M. Valls



Tema 4: Aprendizaje no supervisado

La primera parte de este tema tiene un enfoque más teórico y se hace una introducción al aprendizaje no supervisado para pasar a continuación a estudiar el aprendizaje competitivo. En la segunda parte, se ve un modelo concreto de red de neuronas no supervisada: los mapas autoorganizados de Kohonen. Este modelo se estudia en detalle y con bastantes ejemplos. Finalmente, se explica un algoritmo de clasificación no supervisada muy conocido, k-medias y una versión supervisada del método de Kohonen llamado LVQ.

4.1 Introducción

Cuando se lleva a cabo el aprendizaje supervisado, siempre conocemos la salida esperada o deseada, por lo que al compararla con la salida real que proporciona la red se determinará el error producido que servirá para ajustar los pesos. En el aprendizaje no supervisado no existe una salida esperada o salida objetivo, por lo que no nos podemos guiar por el error producido.

Una red de neuronas que aprenda de forma no supervisada deberá, de forma autónoma, encontrar características, regularidades, correlaciones o categorías en los datos de entrada.

Algunos de los problemas abordables por estas redes son: Agrupamiento o clasificación no supervisada, Prototipado, Extracción de características, etc.

En general, las redes no supervisadas requieren menores tiempos de entrenamiento que las supervisadas. Las arquitecturas suelen ser simples, habitualmente con una sola capa. Una de las redes de neuronas no supervisadas más utilizadas son los mapas autoorganizados de Kohonen (Self Organizing Maps, SOM) que se estudian con detalle en este tema.

4.2 Modelo Básico y aprendizaje competitivo

En los modelos no supervisados existen dos características importantes que no se daban en los modelos supervisados. La primera es la **interacción lateral**, por la cual la activación de una neurona no sólo depende de la entrada de esa neurona sino que además se verá influenciada por las activaciones de otras neuronas que están en la misma capa. Ahora es importante la ubicación de la neurona en su capa para que pueda relacionarse con sus vecinas, es decir, existirá una estructura que llamaremos **vecindario** (neuronas cercanas en la capa). Para entenderlo mejor podemos pensar en una capa oculta de un perceptrón multicapa. No hay una estructura definida dentro de la capa, el lugar que ocupa una neurona oculta dentro de esa capa es irrelevante porque su activación sólo dependerá de las activaciones de las neuronas de la capa

anterior y de las conexiones a esas neuronas. Sin embargo, en una red no supervisada, el lugar que ocupa una neurona en su capa es importante por la interacción lateral. El vecindario es importante porque influirá en las activaciones de cada neurona.

La segunda característica es el **aprendizaje competitivo** donde las neuronas de la misma capa (capa de competición) compiten de forma que una de ellas será la neurona ganadora. En cada iteración del aprendizaje la neurona ganadora refuerza sus conexiones con la capa de entrada.

Estas dos características generales (interacción lateral y aprendizaje competitivo) se dan en el modelo de redes no supervisadas citado anteriormente llamado SOM.

4.3 Mapas Autoorganizados de Kohonen

En 1982 Teuvo Kohonen presentó un modelo de red denominado “Self-Organizing Maps” (SOM) o mapas autoorganizados. Este modelo se basa en aprendizaje competitivo no supervisado. Al ser un aprendizaje no supervisado, no existe una salida esperada que guíe el proceso de aprendizaje, con lo que la red deberá descubrir regularidades o categorías en los datos de entrada. Al ser aprendizaje competitivo, las neuronas compiten entre ellas, de forma que cuando se presenta un patrón de entrada, sólo una de las neuronas (y posiblemente sus vecinas) se activa.

Consta de una capa de entrada y una capa de competición. Las neuronas de la capa de competición (también llamada “mapa”) están organizadas de forma unidimensional o bidimensional (ver transparencia 13). Cada neurona de la capa de competición tiene asociado un vector de pesos. Una neurona genérica j tendría asociado el vector $\mu_j = \{ \mu_{1j}, \mu_{2j}, \dots, \mu_{nj} \}$ que corresponde al conjunto de pesos de las conexiones entre las neuronas de entrada y dicha neurona j . Por tanto, este vector tendrá la misma dimensionalidad que los datos de entrada y se puede representar gráficamente como un punto en el espacio de entrada, junto a los datos de entrada (ver transparencia 15 y 16).

Para calcular la activación de las neuronas de la capa de competición cuando se presenta un dato, se utiliza la distancia euclídea entre el vector asociado a la neurona y el dato correspondiente. Cada neurona producirá una activación (su distancia a dicho dato) y siendo un aprendizaje competitivo, habrá una neurona ganadora que será la que produzca la salida más pequeña o, dicho de otra forma, la neurona “más similar” al dato presentado, es decir, la neurona más cercana. La neurona ganadora reforzará sus conexiones (su vector asociado) siguiendo una ley de aprendizaje que comentaremos a continuación. También reforzarán sus pesos las neuronas vecinas a la neurona ganadora. El concepto de vecindad se aplica a las neuronas en el mapa o capa de competición, no a la situación de las neuronas en el espacio de entrada según sus vectores asociados. Por ejemplo, en la transparencia 15 vemos que las neuronas C1 y C2 son vecinas en el mapa, pero están lejos en el espacio de entrada, donde están representados sus vectores asociados. Es muy importante no confundir la situación de una neurona en el mapa, que normalmente tendrá 1 o 2 dimensiones y que determina el vecindario o conjunto de neuronas cercanas a la neurona ganadora y que verán reforzadas sus conexiones, con la situación de una neurona en el espacio de entrada, que normalmente tendrá una dimensión mucho más alta y que depende del vector de pesos asociado.

Aprendizaje

Vamos a considerar dos situaciones. En la primera, **no se tendrá en cuenta el vecindario**. El incremento del peso que va desde la entrada i a la neurona j viene dado por:

$$\Delta\mu_{ij} = \alpha(t)\delta_j(t)(e_i(t) - \mu_{ij}(t))$$

Donde $\alpha(t)$ es la razón de aprendizaje que va disminuyendo con el tiempo. Comienza con un valor prefijado como parámetro del método, y en cada ciclo va disminuyendo de forma que al finalizar el número de ciclos previsto, la razón de aprendizaje vale cero o un valor muy pequeño.

δ_j valdrá 1 si la neurona j es la ganadora y 0 en caso contrario. Es decir, sólo se modificarán los pesos de la neurona ganadora.

e_i es la coordenada i del dato de entrada. μ_{ij} es el peso actual que se quiere modificar.

El efecto de aplicar esta ley de aprendizaje es que la neurona se acercará al dato presentado en el espacio de entrada.

A continuación, vemos la ley de aprendizaje **teniendo en cuenta el vecindario**. El incremento será

$$\Delta\mu_{ik} = \frac{\alpha(t)}{\sigma(c_j, c_k)}(e_i(t) - \mu_{ik}(t)), \text{ si } \sigma(c_j, c_k) < \Theta. \text{ En otro caso, } \Delta\mu_{ik} = 0$$

Siendo j la neurona ganadora y k la neurona vecina cuyos pesos queremos modificar. $\sigma(c_j, c_k)$ es la **distancia de vecindario** entre k y j . Cuando la neurona cuyos pesos se van a reforzar coincide con la neurona ganadora se considera que la distancia de vecindario es 1. Θ es el límite de vecindario. Sólo se consideran vecinas las neuronas del mapa que están a una distancia menor que Θ de la neurona ganadora y sólo estas neuronas serán reforzadas. El límite de vecindario suele ser dinámico y decrece con el tiempo (como la razón de aprendizaje). Cuanto más alejadas estén las neuronas vecinas, menor será el factor $\frac{\alpha(t)}{\sigma(c_j, c_k)}$ que multiplica a la diferencia de coordenadas del dato de entrada y de la neurona correspondiente.

Influencia del vecindario

¿Qué diferencia podemos observar entre un mapa entrenado con vecindario y otro sin él?

Cuando se utiliza el vecindario, al presentar un dato se acerca a él no sólo la neurona ganadora sino también sus vecinas. De esta forma, el mapa se conserva la topología del espacio de entrada que generalmente es altamente dimensional. Dicho de otra forma, neuronas vecinas en el mapa serán vecinas también en el espacio de entrada. Podemos considerar que se produce una proyección del espacio de muchas dimensiones en el espacio de dos dimensiones del mapa.

Si en el aprendizaje no se utiliza vecindario, cada neurona se moverá independientemente de sus vecinas con lo que no se conservará esta relación entre el espacio de entrada y el espacio del mapa.

4.4 K-Medias

Es un algoritmo de agrupación o clasificación no supervisada, propuesto 1967. Divide el espacio de entrada en K clusters. El número de clusters debe establecerse a priori. Dado un conjunto de patrones o datos se pretende encontrar K centros (de la misma dimensión que los datos) con el objetivo de minimizar la suma de distancias euclídeas entre cada patrón y el centro más cercano (llamaremos J a la función que calcula esta suma de distancias). El algoritmo básicamente consiste en lo siguiente: 1: se inicializan aleatoriamente los K centros. 2: Se asigna cada patrón al centro más cercano. 3: Para cada centro, se calcula el centro de masas de todos los patrones asignados a él. 4: Estos centros de masas pasan a ser los nuevos centros. 5: Vuelven a repetirse

los pasos 2,3,4 y 5 hasta que los centros no modifiquen su posición o lo hagan en una magnitud menor a un valor dado muy pequeño.

Este algoritmo es muy sencillo de implementar y de usar. Además, es muy eficiente y converge en pocas iteraciones a un mínimo local de la función J. La calidad de la solución obtenida depende de los valores iniciales de los centros. En K-medias no existe el concepto de vecindario. Sería muy similar, en cuanto a los resultados obtenidos, a un mapa de Kohonen (tantas neuronas como centros) donde no se utiliza el vecindario, aunque el algoritmo de aprendizaje es diferente.

4.5 LVQ

LVQ (Learning Vector Quantization) es un algoritmo de clasificación supervisada propuesto por T. Kohonen. Aunque es un método supervisado, se incluye en este tema por su similitud con SOM ya que es una versión supervisada del método de Kohonen. Se basa en la determinación de un conjunto de prototipos para representar a cada una de las clases. Estos prototipos serán vectores de la misma dimensión que los datos de entrada (equivalen a los vectores de pesos asociados a las neuronas del SOM). Puede haber varios prototipos por cada clase. Cada prototipo tiene asociada una clase e inicialmente se distribuyen de forma aleatoria en el espacio de entrada. Mediante la regla de aprendizaje, los prototipos se van posicionando de forma que representen a las clases. Una vez construido el modelo, es decir, situados los prototipos basándose en los datos de entrenamiento, para clasificar un nuevo dato (patrón de test) se le asigna la clase asociada al prototipo más cercano.

Aprendizaje

1: Se generan un determinado número de prototipos por cada clase y se distribuyen aleatoriamente por el espacio de entrada. 2: Se presentan los datos de entrada sucesivamente y para cada uno, se modifica el prototipo más cercano según la siguiente regla de aprendizaje (muy similar a SOM cuando no se utilizaba vecindario): $\Delta\mu_{ij} = \alpha(t)\delta_j(t)(e_i(t) - \mu_{ij}(t))$

En este caso, $\delta_j(t) = 0$ cuando el prototipo C_j no es el más cercano. $\delta_j(t) = 1$ cuando el prototipo C_j es el más cercano y tiene asignada la misma clase que el dato. $\delta_j(t) = -1$ cuando el prototipo C_j es el más cercano y tiene una clase diferente a la del dato. Por tanto, cuando el prototipo más cercano es de la misma clase, se acerca al dato. En caso contrario, se aleja del dato. La razón de aprendizaje $\alpha(t)$ también se va decrementando con el tiempo.