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

Tema 2: Primeros modelos computacionales

En este tema se estudian los más importantes de los primeros modelos computacionales que aparecieron a mediados del siglo XX: Perceptron simple y Adaline. El primero es un modelo simple de neurona con dos estados posibles de salida y una regla de aprendizaje basada en la corrección del error. Este modelo permite clasificar patrones en dos clases diferentes. El segundo, Adaline, se diferencia del anterior en que la salida es continua y depende linealmente de la entrada. Las leyes de aprendizaje en ambos modelos son diferentes: en el Perceptron sólo se tiene en cuenta si se ha equivocado la red o no, mientras que en Adaline se tiene en cuenta cuánto se ha equivocado la red. Además, en este tema se tratarán los problemas de clasificación y de regresión lineal y los problemas de clasificación que no son linealmente separables.

A continuación se presenta un resumen del contenido de las transparencias.

2.1 Perceptron Simple

Este modelo fue propuesto por Rosenblatt en 1958 y se concibió como un sistema capaz de realizar tareas de clasificación de forma automática. El sistema debía determinar las ecuaciones de las superficies que hacían de frontera entre las clases a partir de un conjunto de ejemplos de las diferentes clases. El sistema, al final del proceso, sería capaz de determinar a qué clase pertenecía cualquier ejemplo nuevo.

La arquitectura del Perceptron simple (ver transparencia 4) consiste en una red monocapa con varias neuronas de entrada y una neurona de salida. Las neuronas de entrada recogen los valores de los atributos de entrada ($x_1,\ldots, x_n$), y se conectan a la neurona de salida. Cada conexión tiene asociado un número real llamado peso ($w_1, \ldots, w_n$). Además la neurona de salida tiene un umbral o bias (Θ) que puede considerarse como un peso adicional conectado a una entrada de valor 1.

La salida de la red y tendrá un valor binario +1 o -1 dependiendo del valor de la suma de las entradas multiplicadas por sus pesos y sumándole el umbral, $\boldsymbol{w_1x_1+\ldots+w_nx_n+\theta}$, como puede verse en la transparencia 4. Si este valor es mayor que 0, la salida será 1. Si el valor es menor o igual que 0 la salida será -1.

$w_1x_1+\ldots+w_nx_n+\theta = 0$ es una ecuación lineal que representa a un hiperplano de dimensión n-1. Este hiperplano será la superficie discriminante que separará las dos clases. Si dado un patrón de entrada ($x_1,\ldots, x_n$), la salida del Perceptron es +1, ese patrón pertenecerá a una de las clases. Desde un punto de vista geométrico, el patrón corresponderá a un punto en un espacio de n dimensiones que estará situado a un lado del hiperplano. Si la salida es -1, el patrón pertenecerá a la clase contraria, es decir, el punto estará situado al otro lado del hiperplano.

En la transparencia 6 podemos ver el caso de n=2, donde los datos o patrones serán puntos de un espacio bidimensional. La ecuación del hiperplano discriminante será $w_1x_1+w_2x_2+\theta = 0$, que es la ecuación de una recta (hiperplano de dimensión 1).

Aprendizaje

En las transparencias 7 a 9 se explica el proceso de aprendizaje del percetron simple. A partir de un conjunto de ejemplos etiquetados con la clase a la que pertenecen (conjunto de entrenamiento), se determinan de forma automática los pesos y el umbral del Perceptron que determinarán la ecuación del hiperplano discriminante que separará a los ejemplos pertenecientes a clases distintas. En la tr. 9 se detalla el algoritmo de aprendizaje. Básicamente consiste en ir introduciendo cada uno de los ejemplos del conjunto y si la salida no coincide con la clase etiquetada para ese ejemplo se modificarán los pesos y el umbral siguiendo una ley de aprendizaje. Se seguirá haciendo esto presentando una y otra vez el conjunto de entrenamiento hasta cumplir con el criterio de parada.

2.2 Adaline

Fue desarrollado en 1960 por Widrow y Hoff. Su estructura es muy similar al Perceptron pero hay algunas diferencias fundamentales. La salida es directamente la suma de las entradas por sus pesos más el umbral, $\boldsymbol{y= w_1x_1+...+w_nx_n+ \theta}$. Es decir, corresponde a una función lineal, continua, no a una función escalón o umbral como el Perceptron. En la tr. 16 puede verse la estructura y la ecuación de la salida. Otra diferencia fundamental tiene que ver con el aprendizaje donde se utiliza la diferencia entre el valor de salida esperado y la salida de la red, al ser ambos números reales, esta diferencia indica cuánto se separa la salida del valor esperado mientras que en el Perceptron sólo se sabía si la red se equivocaba o no.

Si representamos los datos como puntos $(x_1,\dots, x_n, d)$, siendo d la salida esperada para ese patrón, podemos representar también en ese mismo espacio la función correspondiente a Adaline, $\boldsymbol{y=w_1x_1+...+w_nx_n+ \theta}$, que corresponderá a un hiperplano. El objetivo es que la salida de la red (y) se aproxime lo más posible a la salida esperada para cada patrón (d), o dicho de otra forma, que el hiperplano se ajuste lo más posible a la nube de puntos que representa a los datos. En la tr. 18 puede verse una representación de un conjunto de datos en un espacio tridimensional (2 atributos de entrada más la salida) junto al plano que representa a Adaline y que se ajusta a la nube de puntos.

Aprendizaje

Como hemos visto, el objetivo del aprendizaje de Adaline es que la salida de la red se aproxime lo más posible a la salida esperada, teniendo en cuenta todos los patrones de entrenamiento. O lo que es equivalente, que el error sea el mínimo posible. Hay que elegir una medida para dicho error y una medida muy adecuada es el error cuadrático, E, que es la suma de los cuadrados de la diferencia entre la salida esperada y la salida de la red para cada patrón (tr. 19). Por lo tanto, la función a minimizar, E, que depende de los pesos, es una función continua y derivable. Además, esta función es cuadrática y tiene un mínimo global. La regla delta nos da el conjunto de pesos que minimiza esta función de error. Mediante un proceso iterativo se van presentando los patrones uno a uno y se van modificando los pesos siguiendo la siguiente **ley de aprendizaje**: El incremento de un peso genérico j será igual a la tasa de aprendizaje multiplicada por la diferencia entre la salida esperada y la salida de la red y la entrada correspondiente a ese peso x_j (ver tr. 22). Como el umbral se puede considerar como un peso más conectado a una entrada de valor 1, el incremento que debe aplicarse al umbral será igual a la tasa de aprendizaje por la diferencia entre la salida esperada y la salida de la red. La tasa o razón de aprendizaje es un parámetro que gradúa la magnitud de los incrementos.

En la tr. 23 pueden verse detallados todos los pasos a seguir en el proceso de aprendizaje de Adaline, incluyendo la ley de aprendizaje descrita anteriormente.

2.3 Problemas de clasificación lineal

En la tr. 28 se describen los problemas de clasificación lineal. En un problema de n dimensiones, tratan de encontrar un **hiperplano** de dimensión n-1 (función lineal) que separe los datos de diferentes clases.

En la transparencia puede verse representado gráficamente un problema con dos atributos de entrada (2 dimensiones) con los puntos verdes representando a los datos de una clase (A) y los puntos rojos a los datos de la clase contraria (B). El hiperplano separador es la recta (1 dimensión) que separa los puntos verdes de los rojos. El **Perceptron simple** resuelve estos problemas de clasificación lineal.

2.4 Problemas de regresión lineal

En la tr. 29 se describen los problemas de regresión lineal, donde se trata de encontrar una función lineal o hiperplano que se ajuste a un conjunto de puntos x= $(x_1, x_2, \ldots x_n, y)$, siendo y la salida esperada para el punto x, de forma que la función se aproxime lo más posible a las salidas esperadas y. El modelo Adaline es capaz de resolver estos problemas de regresión lineal.

2.5 Problemas no linealmente separables

Hay problemas de clasificación que no son linealmente separables, es decir, no existe un hiperplano que sea capaz de separar los puntos de clases diferentes de forma que deje a un lado todos los puntos de una clase y al otro lado todos los puntos de la clase contraria. En la tr. 30 puede verse un ejemplo sencillo con la función XOR. Este problema no puede resolverse con un perceptron simple pero sí con la combinación de varios perceptrones y encontrando los pesos adecuados. El problema de este enfoque es que la ley de aprendizaje que habíamos visto para el perceptron simple no es aplicable y los pesos tendrían que ser determinados con un proceso manual.