

PRÁCTICA 1 DE NEUROCOMPUTACIÓN

Curso 2013-2014

Introducción a las Redes Neuronales Artificiales

4 de Febrero de 2013

Fecha de Entrega: 28 de Febrero

1. Introducción

El sistema nervioso es un sistema de procesamiento de información en paralelo, capaz de realizar una gran cantidad de tareas de forma rápida y eficiente bajo diversas condiciones de error. Las características que presenta son deseables para otros muchos sistemas en diversos campos de la ciencia, lo que hace del sistema nervioso un buen ejemplo a imitar. La Neurocomputación Artificial intenta proporcionar a los sistemas artificiales alguna de estas habilidades inspirándose en la forma en que los sistemas biológicos procesan la información [1,2].

Las redes neuronales artificiales surgieron a mediados del siglo XX [3,4], heredando dos de las características más significativas del sistema nervioso: la capacidad de detectar patrones de comportamiento y la de aprender a partir de ejemplos sin la necesidad de formalizar el conocimiento a tratar. Estas características son la base de dos de sus mayores ventajas respecto a otros paradigmas de razonamiento artificial: el aprendizaje adaptativo y la auto-organización.

El objetivo principal de esta práctica es diseñar e implementar distintos modelos neuronales artificiales “tradicionales” aplicables a la detección de distintos patrones de comportamiento.

2. Neuronas de McCulloch-Pitts

Warren McCulloch y Walter Pitts diseñaron en 1943 uno de los primeros modelos de neuronas artificiales. En una neurona típica de McCulloch–Pitts la activación es binaria (su salida es 0 o 1), las conexiones excitadoras que llegan a una misma neurona tienen el mismo peso, el umbral se escoge de forma que una sola conexión inhibidora fuerce a que la salida de la neurona sea 0 en ese paso y las salidas tardan un intervalo de tiempo en llegar a las neuronas receptoras. Todas estas propiedades hacen que las neuronas de McCulloch-Pitts sean útiles para modelar fenómenos que requieran funciones lógicas y retrasos temporales.

2.1. Objetivos

Diseñar una red con neuronas de McCulloch-Pitts que sea capaz de distinguir entre estímulos presentados en dos direcciones diferentes. Los estímulos tendrán la forma:

Arriba-Abajo:

	t	$t+1$	$t+2$	$t+3$
x_1	1	0	0	1
x_2	0	1	0	0
x_3	0	0	1	0

Abajo-Arriba:

	t	$t+1$	$t+2$	$t+3$
x_1	0	0	1	0
x_2	0	1	0	0
x_3	1	0	0	1

Como puede verse en las tablas anteriores, la dirección se conserva de forma recurrente. La red deberá tener tres neuronas de entrada y dos de salida. La salida de la red en t será (0 1) si el estímulo que se presenta en el instante $t-1$ está orientado hacia abajo respecto al estímulo en el instante $t-2$. Si el estímulo se presenta hacia arriba la salida será (1 0). Si el estímulo está quieto (se repite en instantes sucesivos) la salida será (0 0).

Programa la red de forma que se escriba en un fichero la activación de todas las neuronas en cada paso de tiempo. El programa debe leer los estímulos sucesivos desde un fichero en cada paso de simulación. Este será un fichero de texto con tantas líneas como tiempos diferentes. En cada línea estará la activación de cada neurona de entrada separada por espacios. Por ejemplo, el fichero correspondiente al ejemplo arriba-abajo dado anteriormente será:

1 0 0

0 1 0

0 0 1

1 0 0

El fichero podrá contener un número de líneas (tiempos) arbitrarias.

2.2. Material a entregar

La memoria contendrá el diseño de la red con los valores de los pesos utilizados, discutiendo el por qué del diseño implementado e incluyendo referencias a ejemplos en los que se muestre el valor de las activaciones en cada paso para todas las unidades de la red. El código deberá ir acompañado de un fichero makefile con al menos los siguientes objetivos:

- help: para explicar el uso del fichero.

- compile: para compilar todos los programas que se entreguen.
- Un objetivo para ejecutar cada uno de los programas y simulaciones que se discutan.

3. Perceptrón y Adaline

Una de las características más destacables de las redes neuronales es su capacidad de aprendizaje y generalización. El objetivo de esta práctica es el estudio de las reglas de aprendizaje que utilizan el Perceptrón y el Adaline y su aplicación a la clasificación de patrones.

3.1. Objetivos

1. Diseñar un Perceptrón con una codificación bipolar tanto en la entrada como en la salida que solucione el problema de predicción dado en la base de datos "problema_real1.txt". El programa deberá:
 - leer la base de datos
 - realizar una partición aleatoria entrenamiento-test (utilizando por ejemplo unos porcentajes 2/3 – 1/3).
 - entrenar el modelo usando los datos de entrenamiento
 - una vez entrenado, el modelo clasificará los datos de test.
 - El programa deberá dar al usuario la siguiente información:
 - Número de iteraciones de entrenamiento
 - Error en entrenamiento y test
 - Clase real y clase predicha para los datos de test
2. Convertir la red anterior en un Adaline y repetir los experimentos realizados con el Perceptrón. Se seguirá utilizando una codificación bipolar.
3. Explorar la influencia de los distintos parámetros de la red en los errores de entrenamiento y test y, en caso de no encontrar una configuración óptima para la clasificación, indicar las razones que hacen que los patrones no puedan ser clasificados con el modelo de red neuronal artificial correspondiente.
4. Ampliar los programas para que admitan, como caso especial de porcentajes entrenamiento-test, los valores 1 - 1. En este caso, tanto el conjunto de entrenamiento como el de test son iguales a la base de datos leída. Este será el único caso donde el conjunto de entrenamiento tiene patrones comunes a test. Repetir los análisis anteriores para las bases de datos nand.txt, nor.txt y xor.txt. ¿Se pueden resolver estos problemas completamente? ¿Por qué?
5. Aplicar ahora el Perceptrón y el Adaline al problema de clasificación definido en la base de datos problema_real2.txt. Finalmente, generar los ficheros predicciones_perceptron.txt y predicciones_adaline.txt con las predicciones realizadas por dichos modelos para los datos no etiquetados problema_real2_no_etiquetados.txt

6. Repetir los apartados 4 y 5 añadiendo como entradas adicionales, combinaciones no lineales de los atributos originales. ¿Mejora la predicción? ¿por qué?

3.2. Material a entregar

La memoria contendrá la descripción del diseño definitivo del Perceptrón y de Adaline, así como la de los experimentos realizados tanto para entrenar como para probar ambas redes. Se entregará además una discusión con la comparación entre los resultados obtenidos con ambas.

Se deberá entregar los ficheros generados predicciones_perceptron.txt y predicciones_adaline.txt con las predicciones realizadas para los datos no etiquetados del problema real 2.

El código fuente a entregar deberá ir acompañado de un fichero makefile con como mínimo los siguientes ficheros:

- help para explicar el uso del fichero.
- compile para compilar todos los programas que se entreguen.
- Un objetivo para ejecutar cada uno de los programas y simulaciones que se discutan.

Referencias

- [1] L. V. Fausett. Fundamentals of Neural Networks. Prentice-Hall Inc., 1994.
- [2] S. Haykin. Neural Networks: A Comprehensive Foundation. Prentice-Hall Inc., Englewood Cliffs, NJ, 1998.
- [3] D. O. Hebb. The organization of Behavior: A Neuropsychological Theory. Wiley. New York, 1949.
- [4] W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5(1-2):115–133, 1943.