Práctica 4 - Introducción a los Modelos de Computación Conexionistas

Perceptrón/Adaline

Profesor responsable: Patricio García Báez
Fecha tope de corrección: 19 de noviembre del 2002
Objetivos: Trabajo con redes monocapa

Se pretende **hacer uso** de los modelos de redes neuronales monocapas de **Perceptrón simple y Adaline** desde el **SNNS**, así como **probarlos** haciendo uso de **patrones ya utilizados** anteriormente el la práctica 1.

El SNNS no implementa de modo exclusivo los modelos de Perceptrón y Adaline, pero **haciendo uso** de las funciones de activación **Act_StepFunc** y **Act_Identity** junto a la función de aprendizaje **Std_Backpropagation** dichos modelos pueden ser simulados casi con entera exactitud[1]. Además, si es necesario, una vez finalizado el aprendizaje del modelo Adaline es posible acoplarle la función de salida **Out_Threshold05**.

`StepFunc`	$a_j(t) = \begin{cases} 1 & \text{if } net_j(t) > 0 \\ 0 & \text{if } net_j(t) \leq 0 \end{cases}$
`Identity`	$a_j(t) = net_j(t)$
`Threshold_0.5`	$o_j(t) = \begin{cases} 0 & \text{if } a_j(t) \leq 0.5 \\ 1 & \text{if } a_j(t) > 0.5 \end{cases}$

Los **patrones** con los que se tratará de **probar** la práctica serán aquellos empleados en la **práctica 1,** así como los correspondientes a ***letters.pat*** que se incluyen con el SNNS (*ftp://ftp.csi.ull.es/pub/asignas/IMCC/prac4_02/letters.pat*).

Ha de estudiarse los resultados de aplicar ambos modelos ante los patrones anteriores, tomando nota de si **convergen o no**, la **influencia de los ratios** de aprendizaje en el **tiempo de convergencia**, los **errores obtenidos**.

Con los resultados de los entrenamientos anteriores se confeccionará una **tabla de resultados** en la que se indique tipo de patrón, variante de RN empleada, rango de ratios de aprendizaje útiles, número de ciclos empleado para el aprendizaje y mejor error obtenido, tanto su valor SSE como en porcentaje de aciertos así como las observaciones que se quieran hacer constar. Dicha tabla será la que finalmente se entregue como justificación de la práctica.

1 Para ello es necesario pasar del modelo de neurona tipo 1 (con umbral), que implementa el SNNS, al modelo 3 (con *bias*), para ello habría que añadir a cada patrón una entrada adicional con valor de activación a 1. El peso de la conexión de esa entrada con las neuronas hará las veces de *bias*, que el algoritmo Std_Backpropagation se encargará de actualizar.