

Práctica 3 - Introducción a los Modelos de Computación Conexionista

REDES MONOCAPA

Objetivo: Trabajo con RN monocapa

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

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 exactitud¹. 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 los **seis primeros** empleados en la **práctica 1**, así como los correspondientes a **letters.pat** que se incluyen con el SNNS (**letters.pat**).

Ha de tratarse de generar redes neuronales resuelvan lo mejor posible los problemas, y ratios de aprendizaje que la hagan converger en el número mínimo de ciclos posibles. Con los resultados de los entrenamientos anteriores se confeccionará una **tabla de resultados** en la que se indique **tipo de patrón**, **modelo de RN empleado**, **mejor error** obtenido (tanto su valor **SSE** como en **porcentaje de errores**), valor del **ratio de aprendizaje óptimo** y **número de ciclos** empleado para el aprendizaje, así como las **observaciones** que se quieran hacer constar. Dicha tabla (**formulario_practica3.odt**) será la que finalmente se envíe 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*), lo que supone 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.