

---

## OCR-Weka: Redes neuronales artificiales - MLP

---

**NOMBRE:**

**Fecha:**

---

En la carpeta *classifiers/functions* de la pestaña *Classify* de Weka puedes encontrar el clasificador *MultilayerPerceptron*. Su implementación sigue las mismas pautas que el sistema explicado en el curso.

(a) Explica los siguientes parámetros: *hiddenLayers*, *learningRate*, *momentum* y *training time*. ¿Qué valores toman por defecto en Weka?

\* *hiddenLayers*: el numero de capas intermedias de la red neuronal artificial. Puede tomar los valores 1,2,3 por ejemplo para especificar una red con 3 capas intermedias.

\* *learningRate*: el valor de la tasa de aprendizaje.

\* *momentum*: un valor aplicado al update de los pesos.

\* *training time*: el numero de pasos durante cuales la red neuronal artificial esta entrenada.

(b) Prueba el ejemplo de OCR con diferentes valores de los parámetros anteriores e indica las tasas de acierto que obtienes para el conjunto de test. Recuerda entrenar cada clasificador con el conjunto de datos de entrenamiento.

\*Parametros default: Tasa de acierto 78%

\**learningRate* = 0.1, *momentum*=0 => tasa de acierto 76%

\**learningRate* = 0.1, *momentum*=0, *hiddenLayers*0,1,2,3 => tasa de acierto 10%

(c) Dado que en el proceso de aprendizaje, cálculo de errores cometidos, se calculan distancias entre valores obtenidos y valores reales, es conveniente normalizar los atributos de entrada y los valores de las clases a valores numéricos en el intervalo  $[-1,1]$ . Por ello, verás que se utilizan dos parámetros que por defecto realizan dicha normalización: *normalizeAttributes* y *normalizeNumericClass*. Prueba qué sucede si no se realiza esta normalización. ¿Qué tasa de acierto se obtiene para el conjunto de test? Para el resto de parámetros utiliza los valores por defecto de Weka.

Si no se realiza la normalización la tasa de acierto baja a 10% para el conjunto de test.