

UNIVERSIDAD SIMON BOLIVAR
Departamento de Cómputo Científico
CO-6612, Introducción a las redes neuronales
Tarea 4: Perceptrón Multicapas

1. Implemente su propio perceptrón multicapas para que tenga una capa oculta con un número n de neuronas en la capa oculta y k neuronas en la capa de salida. n y k son parámetros de su red (función). Usted puede emplear el lenguaje de programación de su preferencia. Entregar el código con una documentación mínima.

En esta tarea usted volverá sobre la clasificación de la base de datos MNIST usada en las tareas anteriores. Especificaciones:

- capa de entrada: 784 unidades.
 - capa de salida: 10 unidades.
 - funciones de activación: sigmoide en todas las capas.
 - respuestas deseadas: 0.9 para la unidad que representa la clase correcta y 0.1 en caso contrario.
 - preprocesamiento: Escalar la entrada a valores en $[0,1]$
 - algoritmo de entrenamiento: descenso de gradiente estocástico con momentum para 50 épocas.
 - parámetros de aprendizaje: $\eta = 0.1$ (tasa de aprendizaje) y $\alpha = 0.9$ (parámetro de momentum)
 - inicialización de pesos: aleatoria en el intervalo $[-0.05, 0.05]$
2. Experimento 1: Realice su entrenamiento variando el número de unidades ocultas con $n = 20, 50, 100$. Incluya el sesgo en cada nodo de la red. Grafique el error de entrenamiento y de validación. Discuta el efecto del número de neuronas en el error de entrenamiento y sobre la validación. Compare los resultados con las tareas anteriores.
 3. Experimento 2: Con $n = 100$ y $\eta = 0.1$, realice entrenamientos para $\alpha = 0, 0.25$, y 0.5 . Compare el entrenamiento con respecto al experimento 1 en relación a los errores, número de épocas para entrenar y cualquier indicio de sobre-entrenamiento.
 4. Experimento 3: Fije n a 100, $\alpha = 0.9$. Entrene dos redes. La primera red se entrenará con $1/4$ del conjunto de entrenamiento y la segunda con la mitad del conjunto de entrenamiento (asegurese en cada caso estar tomando datos de las 10 clases). Comente sus resultados en relación al experimento 1.

Nota: tarea tomada del curso de Machine Learning de la Universidad de Portland