

Introducción al reconocimiento de patrones:

Trabajo Práctico 7, Redes Neuronales

M. Sc. Saúl Calderón Ramírez
Instituto Tecnológico de Costa Rica,
Escuela de Computación, bachillerato en Ingeniería en Computación,
PAttern Recongition and MACHine Learning Group (PARMA-Group)

17 de octubre de 2018

Fecha de entrega: Martes 13 de Noviembre.

Entrega: Un archivo .zip con el código fuente LaTeX o Lyx, el pdf, y un script en MATLAB o python, debidamente documentado, con una función definida por ejercicio. A través del TEC-digital.

Modo de trabajo: Grupos de 3 personas.

En el presente trabajo práctico se introducirá el concepto de redes neuronales.

1. Implementación de una red neuronal realimentada (Perceptrón de tres capas)

1. Construya una red perceptrón multicapa, usando los siguientes lineamientos.

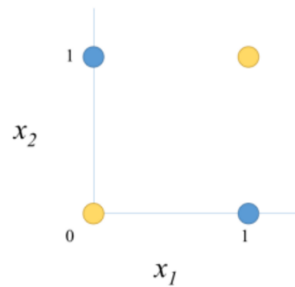
a) Construya una función *inicializarRed(neuronasPorCapa, alpha, maxPesosRand)* la cual reciba:

- 1) *neuronasPorCapa*: Un arreglo con las neuronas por cada capa, de la forma $[D, M, K]$ con D las neuronas de entrada, M las neuronas en la capa oculta, y K las neuronas en la capa de salida.
- 2) *alpha*: El coeficiente de aprendizaje α .
- 3) *maxPesosRand*: El valor máximo que puede tomar cualquiera de los pesos en las dos capas.

Se recomienda utilizar las facilidades de MATLAB u Octave para crear estructuras de datos dinámicas, de modo que la función retorne una estructura de datos con el nombre *red*, y las propiedades como por ejemplo el coeficiente de aprendizaje *alpha* sea definido de la forma *red.alpha*. Además, por ejemplo, la matriz de pesos por cada

capa puede ser definido en dos celdas del atributo $red.W$, de modo que para acceder a la matriz de pesos entre la capa de entrada y la oculta se haga $red.W\{1\}$ y para los pesos entre la capa oculta y de salida $red.W\{2\}$. También se pueden fijar como atributos los valores de los pesos netos por cada neurona en cada capa, de la salida en la función y de los deltas δ , y los valores en las neuronas de entrada.

- b) Construya una función *calcularPasadaAdelante(red)* que reciba una red y actualice los pesos netos y salidas de cada capa, usando el algoritmo de propagación hacia adelante descrito en el material del curso.
 - c) Defina la función *calcularPasadaAtras(red)* que también reciba una red y actualice los valores de los deltas en cada capa según el material visto en clase.
 - d) Escriba también, la función *actualizarPesosSegunDeltas(red)* la cual actualice los pesos de ambas capas según los deltas de la red.
 - e) Defina las funciones *asignarEntrada(x, red)* la cual reciba una muestra \vec{x} , además de la red, para asignar los valores en las salidas de las neuronas de entrada, y *asignarSalidaDeseada(t, red)* la cual reciba los valores a asignar en las neuronas de salida según el arreglo \vec{t} .
 - f) A partir de todas las funciones anteriores, construya la función *entrenarRed(numIteraciones, X, T, red)*, la cual recibe:
 - 1) *numIteraciones*: el número de iteraciones para el cual se va a ajustar los pesos en todo el conjunto de muestras.
 - 2) *X*: El conjunto de muestras de entrenamiento.
 - 3) *T*: El conjunto correspondiente a las etiquetas del conjunto de entrenamiento *X*.
 - g) Finalmente, defina las funciones *evaluarMuestra(x, red)*, la cual obtenga la salida de la red *y* al evaluar la muestra \vec{x} , y a partir de ella, construya la función *evaluarClasificacionesErroneas(X, T, red)*, la cual evalúe el error total para todas las muestras. Por cada iteración de la función *entrenarRed*, ejecute la función *evaluarClasificacionesErroneas* de forma que se pueda construir un gráfico del error en función del número de iteraciones.
2. Construya una red que resuelva la clasificación de muestras con componentes booleanos $\vec{x} \in \mathbb{R}^2$, según la tabla de verdad del X OR, de modo que el conjunto *T* agrupe las salidas esperadas del X OR para las entradas en *X*. Como se observa en la Figura 1, el problema de la clasificación para muestras de una función X OR es un problema sin solución para un clasificador lineal.
- a) Entrene la red, probando distintos parámetros, y documente los valores para tales parámetros los cuales logran un error de cero para las 4 muestras de entrada.



Cuadro 1: X OR como un problema de clasificación no lineal.

- b) Documente la cantidad de iteraciones necesarias para la convergencia de la red y muestre un gráfico con el error en función del número de iteraciones.
3. Utilice la función provista *generarDatosR2*, para generar los conjuntos de datos X y T , representadas matricialmente y realice lo siguiente:
 - a) Muestre la gráfica de los datos, e identifique si son linealmente separables.
 - b) Construya una red neuronal para su clasificación, con $D = 2$ neuronas de entrada, $K = 1$ neuronas en la capa de salida y:
 - 1) $M = 20$ neuronas en la capa oculta, y $\alpha = 0,01$.
 - 2) $M = 20$ neuronas en la capa oculta, y $\alpha = 0,1$.
 - 3) $M = 5$ neuronas en la capa oculta, y $\alpha = 0,01$.
 - 4) $M = 5$ neuronas en la capa oculta, y $\alpha = 0,1$.
 - 5) $M = 2$ neuronas en la capa oculta, y $\alpha = 0,01$.

Por cada red, realice lo siguiente:

- 1) Ejecute el entrenamiento con todo el conjunto de muestras X , evalúe el error para todas esas muestras, y grafique la función de las iteraciones vs. el error.
- 2) Ejecute el entrenamiento con el 80 % de las muestras en X y el restante 20 % de las muestras empleelo para la validación (evaluación del error), y grafique la función de iteraciones vs. el error.
- c) Compare los resultados según las gráficas, y explique, como se puede ajustar el coeficiente de aprendizaje α en función de las neuronas en la capa oculta? Qué relación hay entre la cantidad de neuronas en la capa oculta y el error en el conjunto de entrenamiento? Cómo se puede ajustar la cantidad de neuronas en la capa oculta en función de las neuronas en la capa de entrada?

4. Cargue las imágenes de dígitos manuscritos o muestras X de la base de datos MNIST, y las correspondientes etiquetas en el conjunto T y realice lo siguiente:
- a) Defina la cantidad de neuronas en la capa de entrada D y la cantidad de neuronas en la capa de salida T
 - b) Según las heurísticas comentadas en clase y deducidas en los apartados anteriores, calibre los valores de M , α y la cantidad de iteraciones P , documente tales valores, y grafique el error usando el 100 % de las muestras para entrenamiento y validación, y 80 % de las muestras.
 - c) Dibuje manualmente en un editor de imágenes digital (como *paint*) 5 muestras nuevas, y documente el resultado de utilizar la red obtenida en el paso anterior, mostrando por cada muestra si su clasificación fue correcta o incorrecta.