

# Trabajo Práctico 1: Redes Neuronales

Ph. D. Saúl Calderón Ramírez  
Instituto Tecnológico de Costa Rica,  
Escuela de Computación,  
PAttern Recongition and MACHine Learning Group (PARMA-Group)

16 de abril de 2024

**Fecha de entrega:** Martes 23 de Abril.

**Entrega:** Un archivo .zip con el código fuente LaTeX o Lyx, el pdf, y un script en jupyter, debidamente documentado. A través del TEC-digital.

**Modo de trabajo:** Grupos de 3 personas.

En el presente trabajo práctico se introducirá el concepto de redes neuronales. Para la documentación externa, describa el funcionamiento de cada función en términos generales y presente al menos dos pruebas unitarias que aseguren su correcto funcionamiento.

## 1. (30 puntos) Implementación de una red neuronal realimentada (Perceptrón de tres capas)

1. Construya una clase `Multilayer_perceptron`, con los siguientes metodos y atributos.

a) **(5 puntos)** Construya una función `__init__(neurons_per_layer, alpha, max_weights)` la cual reciba:

- 1) `neurons_per_layer`: 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  $\alpha$ .
- 3) `max_weights`: El valor máximo que puede tomar cualquiera de los pesos en las dos capas.

Debe utilizar la representacion matricial de los pesos vista en clase.

- 1) Documente al menos una prueba unitaria diseñada por usted donde verifique su correcto funcionamiento.
- b) 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  $\delta$ , y los valores en las neuronas de entrada.

c) **(5 puntos)** Construya una función *forward(X)* que actualice los pesos netos y salidas de cada capa, usando el algoritmo de propagación hacia adelante descrito en el material del curso. Se recibe como entrada la matriz  $X$  con  $n$  observaciones de entrada. Use la función tangente hiperbolico como función de activación para ambas capas.

1) Documente al menos una prueba unitaria diseñada por usted donde verifique su correcto funcionamiento.

d) **(10 puntos)** A partir de todas las funciones definidas abajo, construya la función *train\_mlp(num\_epochs, X, T, alpha, gamma)* la cual recibe:

1) *num\_epochs*: 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  $X$ .

El algoritmo de entrenamiento a implementar se basará en el descenso del gradiente. Use la función de tangente hiperbolico como función de activación en ambas capas. Detalle las ecuaciones de actualización a utilizar. Identifique y explique qué modificaciones deben realizarse al algoritmo de propagación hacia atrás visto en clase.

1) Tome un pantallazo de una prueba unitaria diseñada por usted donde verifique su correcto funcionamiento.

2) Defina la función *backpropagate\_deltas(X, T)* actualice los valores de los deltas en cada capa según el material visto en clase.

*a'* Tome un pantallazo de una prueba unitaria diseñada por usted donde verifique su correcto funcionamiento.

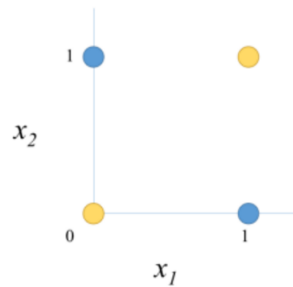
3) Escriba también, la función *update\_weights(X, T)* la cual actualice los pesos de ambas capas según los deltas de la red.

*a'* Tome un pantallazo de una prueba unitaria diseñada por usted donde verifique su correcto funcionamiento.

e) **(5 puntos)** Finalmente, construya la función *get\_error(X, T)*, la cual evalúe el error total para todas las muestras. Por cada iteración de la función *entrenar\_red*, ejecute la función *evaluar\_error* de forma que se pueda construir un gráfico del error en función del número de iteraciones. Implemente el error MSE.

## 2. (20 puntos) Clasificación de función XOR

1. Construya una red que resuelva la clasificación de muestras con componentes booleanos  $\vec{x} \in \mathbb{R}^2$ , según la tabla de verdad del XOR, de modo que el conjunto  $T$  agrupe las salidas esperadas del XOR para las entradas en  $X$ . Como se observa en la Figura 1, el problema de la clasificación



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

para muestras de una función X OR es un problema sin solución para un clasificador lineal.

- a) Documente la arquitectura definida para resolver tal problema (cantidad de neuronas de entrada, ocultas y salida), incluyendo un diagrama de la misma.
- b) Entrene la red, probando distintos parámetros, y documente los valores para tales parámetros los cuales logran un error de cero para las 4 observaciones de entrada.
  - 1) Muestre las curvas de aprendizaje para los hiper-parametros probados.
- c) 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. (30 puntos) Clasificador de datos en $\mathbb{R}^2$

1. (10 puntos) Utilice la función provista *generar\_datos\_R2*, para generar los conjuntos de datos  $X$  y  $T$ , representadas matricialmente y realice lo siguiente:
  - a) Genere dos conjuntos de datos, unos linealmente separables y otros no linealmente separables. Muestre la gráfica de los datos, y documente los parametros seleccionados para generar los datos.
2. (20 puntos) Construya una red neuronal para su clasificación, con  $D = 2$  neuronas de entrada,  $K = 1$  neuronas en la capa de salida y:
  - a)  $M = 25$  neuronas en la capa oculta,  $\alpha_1$ .
  - b)  $M = 25$  neuronas en la capa oculta,  $\alpha_1$ .
  - c)  $M = 4$  neuronas en la capa oculta,  $\alpha_2$ .

- d)  $M = 4$  neuronas en la capa oculta,  $\alpha_2$ . Con  $\alpha_1$  y  $\alpha_2$  escogidos por ustedes.

Por cada red, realice lo siguiente, por  $P = 50$  iteraciones:

- a) Ejecute el entrenamiento con todo el conjunto de muestras para los dos escenarios (linealmente separables y no linealmente separables)  $X$ ,
- 1) Use 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. Use sklearn para hacer la partición de los datos.
    - a' Construya una tabla de graficas de la función de las iteraciones vs. el error.
    - b' Construya una tabla donde documente para cada configuración probada el error final de convergencia.
  - b) Compare los resultados según las gráficas, y explique, como se puede ajustar el coeficiente de aprendizaje  $\alpha$  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? Investigue, citando las referencias externas consultadas (artículos científicos), puede ajustar la cantidad de neuronas en la capa oculta en función de las neuronas en la capa de entrada?

#### 4. (20 puntos) Clasificador de ataques usando datos estructurados

1. (10 puntos) Cargue los registros de mensajes disponibles en el dataset UNSW-NB15: *A Comprehensive Data set for Network Intrusion Detection systems* y realice lo siguiente:
  - a) Utilice una codificación *one-hot vector* para los atributos categoricos.
  - b) Defina la cantidad de neuronas en la capa de entrada  $D$  y la cantidad de neuronas en la capa de salida  $T$
  - c) Según las heurísticas comentadas en clase y deducidas en los apartados anteriores, calibre los valores de  $M$ ,  $\alpha$  y la cantidad de iteraciones  $P$ , documente tales valores, y grafique el error usando el 80 % de las observaciones para entrenamiento.
    - 1) Calibre manualmente  $M$ ,  $\alpha$  con  $P = 50$ , y documente las curvas de aprendizaje.
  - 2) (10 puntos extra) Calibre los 3 parametros anteriores con optuna o weights and biases.

2. **(10 puntos)** Divida el conjunto de datos en 80 % de entrenamiento y 20 % de prueba en 30 corridas y compare el utilizar la entrada sin normalización, y el normalizar la entrada, usando el siguiente procedimiento: Normalice la entrada de 0 a 1, usando el valor máximo de la imagen (normalización por observación  $\ell_\infty$ ).
- a) Documente de forma externa al menos dos pruebas de la normalización.
  - b) Comente los resultados de la comparativa.