

## Práctica Multi Layer Perceptron (MLP)

El alcance de este trabajo es elaborar un MLP solamente con  $R=1$ , una o dos capas ocultas donde  $S^1 \leq 16$ ,  $S^2 \leq 14$  y la capa de salida  $S^3=1$  para los polinomios 1 a 3, y podrá agregar una tercera capa oculta con  $S^3 \leq 10$  para los polinomios 4 y 5.

Las funciones  $g(p)$  las proporcionará el profesor y son las únicas que deben ser usadas, no se puede usar otras funciones.

El objetivo de la práctica **no es** que el alumno obtenga un ajuste perfecto a la función  $g(p)$ , una aproximación aceptable es suficiente. Se reitera, no se requiere que el ajuste a las funciones  $g(p)$  sea muy precisa, de hecho por eso se limita el número de neuronas de las capas. Así, el objetivo es que el alumno explore el funcionamiento del MLP de una manera sencilla, práctica.

Los pasos a seguir son:

1. Se solicita al usuario que indique cuál de los 5 polinomios se va a aproximar.
  - a. Cada uno de los datos del polinomio a aproximar se deberá separar en tres subconjuntos:  
80% entrenamiento - 10% validación - 10% prueba  
**Nota:** La separación de los datos debe ser homogénea y cubrir lo mejor posible el rango de señal, para ello se debe programar una función que reciba al vector original de datos y regrese un vector que cumpla estas características.
  - b. El usuario tiene que indicar la arquitectura del MLP se usará para ajustar el polinomio. Por ejemplo:  
v1: [**1**  $S^1$  **1**] ó [**1**  $S^1$   $S^2$  **1**]  
v2: [2 **1**] ó [3 2 **1**] **Nota:** Los valores en rojo se quedan fijos.  
Solo para los polinomios 4 y 5, se puede usar:  
v1: [**1**  $S^1$   $S^2$   $S^3$  **1**] con  $S^3 \leq 10$   
**Nota:** Siempre se debe buscar aproximar al polinomio con MLP más pequeño posible.

### Entrenamiento por bloques:

- Se usa pocas épocas de aprendizaje y observar cómo va el ajuste de la señal.
- El aumento de las épocas se debe hacer de manera escalonada (Ejemplo: Deseo hacer 3000 épocas, entonces ejecuto el programa 10 veces y en cada ejecución hago 300 épocas. La primera vez usó valores aleatorios y en las 9 veces restantes, cargo el archivo con los valores finales de la anterior ejecución, así se podrá hacer muchas épocas de aprendizaje, pero en pasos pequeños.) y de preferencia **no ejecutar un aprendizaje con más de 5000 épocas.**
- Si el tiempo de ejecución pasa de las 2 horas, ya no realizar más épocas.

c. El usuario nos da el valor del factor de aprendizaje ( $\alpha$ ). Se recomienda usarlo en el siguiente rango de  $1 \times 10^{-2}$  a  $1 \times 10^{-5}$ .

**Queda prohibido usar valores más grandes o más pequeños.**

d. El usuario debe indicar los valores de los criterios de finalización:

i. Número de épocas máximo: **epochmax**

ii. Valor objetivo al que debe llegar el error de aprendizaje:

**error\_train**  $1 \times 10^{-4}$  ó  $1 \times 10^{-5} \simeq$

iii. El valor de cada cuantas épocas durante el aprendizaje se llevará a cabo una época de validación: Se recomienda **epochval** igual al 10% de **epochmax**.

iv. El número de incrementos consecutivos del error de validación **numval**. Este valor debe guardar una proporción correcta para que el programa pueda activar el Early Stopping.

Ejemplo: **epochmax=300**, con **epochval=10** y **numval=3**, así le damos hasta 10 oportunidades al Early Stopping de activarse, con **numval=5** serían seis oportunidades. En cambio si ponemos **numval=30** da 300 épocas por lo que el programa terminaría al mismo tiempo que esta primera validación, en otras palabras **cancelamos** al Early Stopping y

no será posible determinar si hubo sobre **entrenamiento**.

Estos dos valores se usarán para implementar el algoritmo Early Stopping que es una manera de evitar el sobre aprendizaje (que el error de validación diverge, en vez de converger a cero).

## 2. Se inicia el aprendizaje del MLP.

a. Se inicializan todos los valores de pesos y bias de la RNA con valores aleatorios entre -1 y +1

b. Se inicia la fase del aprendizaje

i. Dado un valor del contador de época, se verifica si es igual o múltiplo del valor epochval, si es así se realiza una época de validación, si no hace una época de entrenamiento.

ii. **Época de entrenamiento:** Se propaga hacia adelante y se aplican las reglas de aprendizaje a cada uno de los datos del conjunto de entrenamiento. Al final se calcula el error de época del entrenamiento (**error\_train**).

iii. **Época de validación:** Aquí se aplica el algoritmo de **Early Stopping**

★ Se inicia un contador en cero **cont\_val**=0.

★ Se propaga hacia adelante todos los datos del conjunto de validación usando los W y b de la última época de entrenamiento. NO se aplican las reglas de aprendizaje y se calcula el error de validación (**error\_val**).

★ A partir del segundo **error\_val**, se verifica si este valor es mayor al anterior **error\_val**, si es así se aumenta en uno al **cont\_val**

★ Se verifica si **cont\_val** = **numval** (esto representa que hubo **numval** incrementos consecutivos del **error\_val**), si es así se termina prematuramente el aprendizaje y se usan los valores de W y b cuando inicio en incremento consecutivo de **cont\_val**, si no se continúa con la siguiente época de entrenamiento.

- ★ Si el **error\_val** actual es menor al anterior y **cont\_val** es diferente de cero, este contador se hace cero.
- ★ Si el criterio de finalización del aprendizaje fue Early stopping, se recomienda iniciar un nuevo experimento, cambiando la arquitectura del MLP y/o el valor del factor de aprendizaje por uno más pequeño.

3. Se verifica si se cumple algún criterio de finalización como epochmax o que el error de entrenamiento sea menor al dado por el usuario. Si se cumple alguno termina el aprendizaje. Si no se cumple ninguno de los criterios continúa el aprendizaje. Una vez finalizado, se pasa a la prueba.
4. Prueba (generalización del conocimiento): Usando los valores finales del aprendizaje ( $W$  y  $b$ ) se propaga hacia adelante el conjunto de prueba y se calcula el error de prueba (**error\_test**). Para que el aprendizaje se considere exitoso el **error\_test** debe estar en el rango de  $1 \times 10^{-3}$  a  $1 \times 10^{-4}$

#### **Nota importante:**

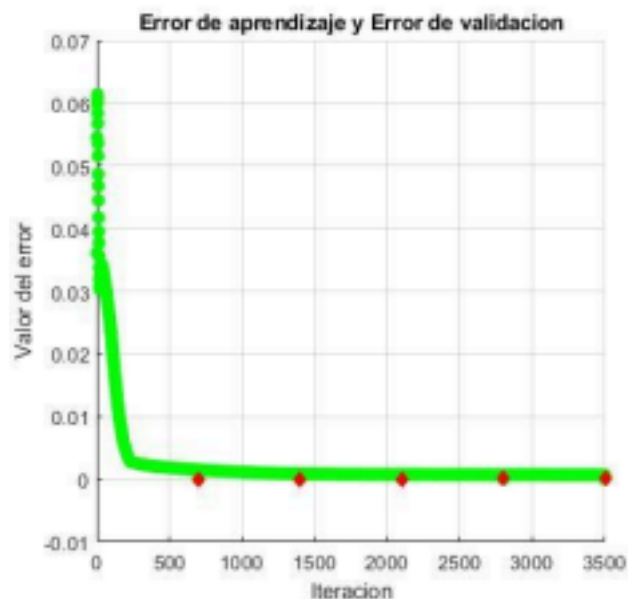
Se recomienda realizar un aprendizaje por bloques. Esto consiste en poner un número pequeño (**epochmax=300**) de épocas y ver el resultado usando la etapa de despliegue de resultados, si las gráficas indican que el error de aprendizaje está convergiendo a cero, tomen los valores finales de ese bloque y usen con valores iniciales del siguiente bloque de aprendizaje. Sigán usando estos bloques siempre y cuando el error siga bajando a cero y se acerquen a el error de  $1 \times 10^{-4}$  ó  $1 \times 10^{-5}$

Durante este procedimiento, deben guardar todos los valores de pesos, bias, error de aprendizaje y error de validación para ser usados en la etapa de despliegue de resultados.

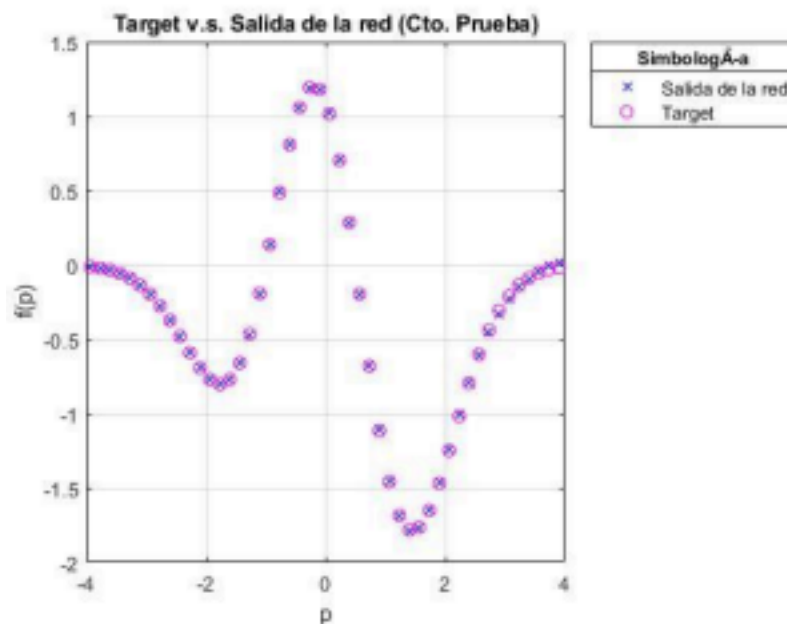
#### **Etapas de despliegue de resultados:**

Una vez concluida la fase del aprendizaje por alguna de las condiciones de finalización, es decir que no necesariamente se obtuvo un aprendizaje exitoso, se deben realizar lo siguiente:

1. Gráfica de evolución del error de aprendizaje y error de validación.



2. Graficar sólo el conjunto de prueba (usar círculos sin relleno) versus salida del MLP (usar cruz). Se grafican sólo los puntos, no dibujar líneas.



3. Gráfica de evolución de cada peso y bias del MLP por **época**. 4. Tabla con las distintas arquitecturas de MLP (poner el valor de los vectores V1 y V2) y el valor final de los tres errores: entrenamiento, validación y prueba. En la

última fila debe aparecer la arquitectura con la que se obtuvo el mejor resultado. Ejemplo:

	Arquitectura	No. de épocas	Alfa	Error final del entrenamiento	Error final de validación	Error de prueba	No. de g(p)
<b>1</b>	v1 = [1 2 1];	10000	0.001	6.00E-01	5.14E-01	5.49E-01	<b>1</b>
	v2 = [2 1];						
<b>2</b>	v1 = [1 2 1];	10000	0.001	5.84E-01	5.71E-01	6.91E-01	<b>1</b>
	v2 = [2 1];						
<b>3</b>	v1 = [1 3 1];	10000	0.001	5.84E-01	5.37E-01	5.91E-01	<b>1</b>
	v2 = [2 1];						

5. Guardar en el archivo **pesos\_y\_bias\_finales.txt**, el valor de pesos y bias con el que se obtuvo el mejor resultado y presentar estos valores en el reporte.