INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

METODOS CUANTITATIVOS PARA LA TOMA DE DECISIONES

Abril de 2021.

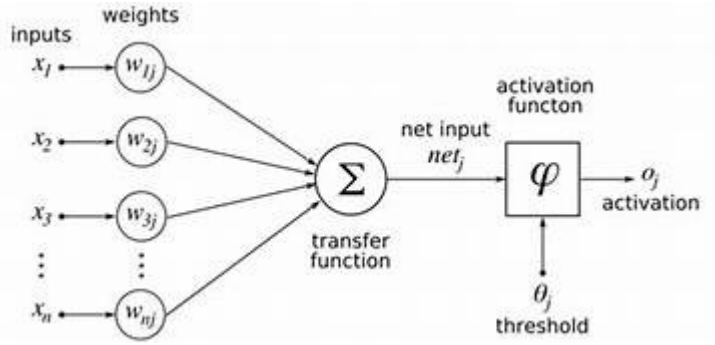
Nombres: _____

Instrucciones: Resuelva en equipos de 3 integrantes según corresponda. Entregue su hoja de respuestas.

Una RNA red neuronal artificial es un modelo computacional inspirado en el comportamiento observado en su homólogo biológico. Consiste en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal produciendo unos valores de salida.

Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso. Estos pesos en los enlaces pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Del mismo modo, a la salida de la neurona, puede existir una función limitadora o umbral, que modifica el valor resultado o impone un límite que no se debe sobrepasar antes de propagarse a otra neurona. Esta función se conoce como función de activación. (Sigmoide, Tanh, escalón, etc.)

Estos sistemas aprenden y se forman a sí mismos, en lugar de ser programados de forma explícita, y sobresalen en áreas donde la detección de soluciones o características es difícil de expresar con la programación convencional.



Existen muchas técnicas sobre la forma de entrenamiento. La más común es backpropagation (Modifica los pesos analizando el error de la red analizándola de final a inicio). El determinar el número de neuronas de la capa oculta de igual forma tiene muchos criterios de diseño, generalmente se aplica $\sqrt{entradas * salidas}$.

El dataset generalmente se emplea con la relación 80/20. 80% de vectores de entrenamiento y 20% de vectores para prueba.

INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
METODOS CUANTITATIVOS PARA LA TOMA DE DECISIONES

Encuentre los pesos sinápticos de la siguiente Red Neuronal Artificial (RNA), mediante el algoritmo de templado simulado y mediante Solver u otro método conocido. La RNA está asignada a la clasificación de números, mediante la identificación de aquellos valores menores e iguales a 5. La tabla de vectores de entrenamiento se encuentra a continuación.

Id Vector	Valor	Salida
1	**1**	**1**
2	**5**	**1**
3	**6**	**0**
4	**10**	**0**

Dado que es un clasificador de números, notará que la salida requerida es uno para afirmar la clasificación y cero para discriminar valores que no cumplen con este criterio.

Emplee el archivo "**rna_mctd_2020.xlsx**"

Anote sus resultados de salida obtenidos (w0 y w1) en la siguiente tabla y evalúe su modelo matemático con los vectores de entrada señalados:

Vector Entrada	**Método Backpropagation** **w0=** **w1=**	**Semilla aleatoria (sigmoide antes de redondeo)** **w0=** **w1=**	**GRG NonLinear** **w0=** **w1=**	**Templado simulado** **w0=** **w1=**
-1				
3				
5.4				
5.5				
5.6				
6				
8				
100				

Para nuestro caso el número de neuronas que pueden emplearse son:

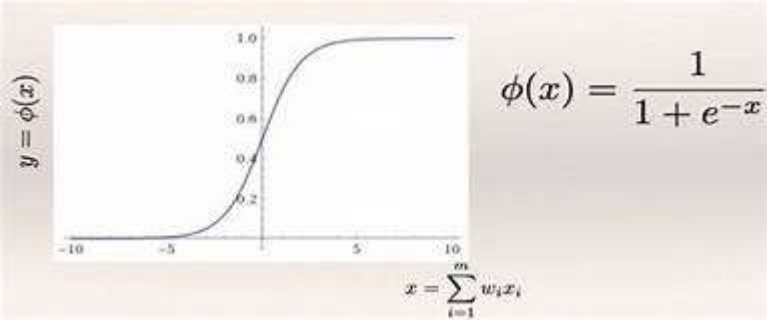
$$\sqrt{entradas * salidas} = \sqrt{1_{entrada} * 1_{salida}} = 1 \cong 1\ neurona\ al\ menos$$

El archivo está integrado por 4 hojas.
Se calculan 2 pesos (w0, w1), aunque sea una sola entrada (x1), ya que requerimos la señal de activación(x0), w0 para x0 y w1 para x1.

La 1ª hoja es el algoritmo de **Backpropagation.** Usted puede configurar el coeficiente de aprendizaje, se recomienda [0,1], el predeterminado está en 0.5, celda L2.
Se comprueba que la red está entrenada cuando la salida deseada es igual a la obtenida. Columnas H e I, y es la obtenida y Sal_des es la deseada.

Ariel López Rojas, 2021

Como función de transferencia se está utilizando la función sigmoide.



Haga varias pruebas cambiando el coeficiente de aprendizaje y determine los pesos que corresponden al entrenamiento correcto, anotándolos en la tabla anterior, sección color amarillo.

En la hoja de Excel verá que existe un recuadro que dice prueba. Es para que evalúe los datos solicitados en la tabla anterior y verifique que está clasificando correctamente.

Lo mismo hará para las otras tres hojas restantes: **semilla aleatoria, GRG no lineal, y templado simulado.**

Todas incluyen el recuadro de prueba. Para GRG no lineal y templado simulado, ya se incluye la función objetivo y las restricciones, que están minimizando el error obtenido de la salida obtenida y la salida deseada.

El de semilla aleatoria, es más simple, porque se totalizan los errores de la salida obtenida con la salida deseada, considerando 3 funciones de transferencia. Cuando la suma de errores es cero, se considera la red entrenada que corresponde a los pesos (w0y w1) obtenidos.

Para todos los casos, hay que hacer la prueba con los pesos calculados para los vectores indicados en la tabla:

- -1
- 3
- 5.4
- 5.5
- 5.6
- 6
- 8
- 100

Recuerde que la prueba la puede simular calculando:

$$Salida\ obtenida = \frac{1}{1 + e^{-(x0w0+x1w1)}}$$

Lo ideal es que Salida obtenida tienda a cero para vectores que sean mayores a 5, y que tienda a uno cuando sea 5 ó menos.

$$x0 = 1$$

$$x1 = entrada\ (vectores\ indicados\ en\ la\ tabla)$$

$$w0\ y\ w1\ son\ los\ valores\ de\ los\ pesos\ calculados$$

NOTA "Templado Simulado":

Recuerde que el algoritmo de templado simulado es un tipo de metaheurística que permite al proceso de búsqueda escapar de un óptimo local.

Z_c = Valor de la función objetivo de la solución de prueba *actual*,
Z_n = Valor de la función objetivo del candidato actual a ser la siguiente solución de prueba,
T = Parámetro que mide la tendencia a aceptar el candidato actual para ser la próxima solución de prueba si este candidato no es una mejora sobre la solución de prueba actual.

Si se supone que el objetivo es la *maximización* de la función objetivo, acepte o rechace este candidato para ser la próxima solución de prueba como sigue:
Si $Z_n >= Z_c$, siempre acepte este candidato.
Si $Z_n < Z_c$, acepte el candidato con la siguiente probabilidad:

$$Prob\{aceptación\} = e^x\ , donde\ x = \frac{Z_n - Z_c}{T}$$

Si el objetivo es de ***minimización***, se deben invertir Z_n y Z_c en la expresión anterior.
Si el candidato es rechazado repita el proceso con un vecino inmediato de la solución de prueba actual seleccionado de manera aleatoria.
Si ya no existen vecinos inmediatos restantes, el algoritmo termina.

$T_1 = 0.2Z_c$ cuando Z_c es el valor de la función objetivo de la solución de prueba inicial,
$T_2 = 0.5T_1$,
$T_3 = 0.5T_2$,
$T_4 = 0.5T_3$,
$T_5 = 0.5T_4$.