Facultad: Ingeniería
Escuela: Computación
Asignatura: Sistemas Expertos e Inteligencia Artificial

Tema: Redes Neuronales. Perceptrón.

Objetivos Específicos

- Comprender los fundamentos biológicos y las generalidades de las redes neuronales.
- Identificar los elementos que constituyen una red neuronal.
- Comprender la importancia del Perceptrón en el diseño de agentes inteligentes.
- Implementar redes neuronales del tipo Perceptrón en Microsoft Visual C#.

Materiales y Equipo

- Guía Número 8.
- Computadora con programa Microsoft Visual C#.

Introducción Teórica

La capacidad del cerebro y aparato neurológico humano de pensar, recordar, reaccionar y resolver problemas ha inspirado a muchos científicos a realizar modelos computacionales del funcionamiento de las neuronas humanas. El resultado de este interés ha dado lugar a un área de estudio subordinada de la Inteligencia Artificial llamada Redes Neuronales Artificiales.
Las Redes Neuronales son un campo muy importante dentro de la Inteligencia Artificial. Inspirándose en el comportamiento conocido del cerebro humano (principalmente el referido a las neuronas y sus conexiones), trata de crear modelos artificiales que solucionen problemas difíciles de resolver mediante técnicas algorítmicas convencionales.

Desde la década de los 40, en la que nació y comenzó a desarrollarse la informática, el modelo neuronal la ha acompañado. De hecho, la aparición de los computadores digitales y el desarrollo de las teorías modernas acerca del aprendizaje y del procesamiento neuronal se produjeron aproximadamente al mismo tiempo, a finales de los años cuarenta.

Desde entonces hasta nuestros días, la investigación neurofisiológica y el estudio de sistemas neuronales artificiales (ANS, Artificial Neural Systems) han ido de la mano. Sin embargo, los modelos de ANS no se centran en la investigación neurológica, si no que toma conceptos e ideas del campo de las ciencias naturales para aplicarlos a la resolución de problemas pertenecientes a otras ramas de las ciencias y la ingeniería.

Podemos decir que la tecnología ANS incluye modelos inspirados por nuestra comprensión del cerebro, pero que no tienen por qué ajustarse exactamente a los modelos derivados de dicho entendimiento. Los primeros ejemplos de estos sistemas aparecen al final de la década de los cincuenta. La referencia histórica más corriente es la que alude al trabajo realizado por Frank Rosenblatt en un dispositivo denominado ***Perceptrón***. Hay otros ejemplos, tales como el desarrollo del ***Adaline*** por el profesor Bernard Widrow.

Durante todos estos años, la tecnología ANS no siempre ha tenido la misma consideración en las ramas de la ingeniería y las ciencias de la computación, más ansiosas de resultados que las ciencias neuronales. A partir de 1969, el pesimismo debido a las limitadas capacidades del Perceptrón hizo languidecer este tipo de investigación.

A principios de los 80, por un lado Hopfield y sus conferencias acerca de la memoria autoasociativa y por otro lado la aparición del libro Parallel Distributed Processing (PDP), escrito por Rumelhart y McClelland reactivaron la investigación en el campo de las redes neuronales. Hubo grandes avances que propiciaron el uso comercial en campos tan variados como el diagnóstico de enfermedades, la aproximación de funciones o el reconocimiento de imágenes.

Las redes neuronales artificiales son más que una simple forma de emular ciertas características propias de los humanos, como la capacidad de memorizar y de asociar hechos. Todos los problemas que no pueden expresarse mediante un simple algoritmo tienen la característica común de necesitar de la experiencia. El ser humano es capaz de resolver estas situaciones usando su experiencia acumulada.

De esa forma, es claro que la forma correcta de aproximarse a estos tipos de problemas sea mediante la construcción de sistemas que sean capaces de reproducir esta característica eminentemente humana. En ese sentido, las redes neuronales no son más que un modelo artificial y simplificado del funcionamiento del cerebro humano, el cual es el mejor ejemplo de un sistema capaz de adquirir su conocimiento a través de la experiencia.

Red neuronal.

Es un sistema para el tratamiento de la información cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona.

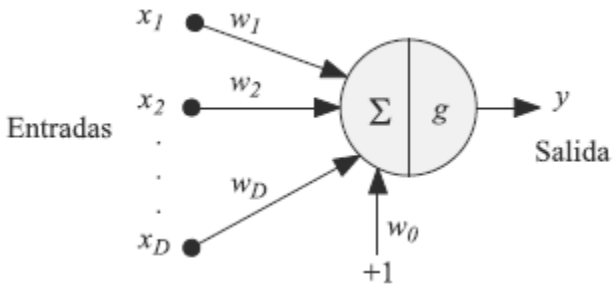
Un sistema de computación compuesto por un gran número de elementos simples, elementos de procesos muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas.

Red Neuronal Artificial (RNA).

Redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico.

Una RNA es un paradigma de procesamiento de información inicialmente inspirado en el modo en el que lo hace el cerebro. El elemento clave de este paradigma es su estructura. Las RNA están compuestas por un cierto número de elementos de procesamiento o ***neuronas*** que trabajan al unísono para resolver un problema específico.

Las redes neuronales actuales se basan en el modelo matemático de neurona propuesto por McCulloch y Pitts en 1943. En dicho modelo, cada neurona recibe un conjunto de entradas {$X_1, X_2,..., X_D$} y devuelve una única salida *y,* tal como se observa en la figura siguiente:



Además, dentro de una RNA existen numerosas conexiones entre las distintas neuronas que la forman. Estas conexiones simulan las conexiones interneuronales del cerebro y, al igual que éstas, pueden establecerse con mayor o menor intensidad.

En el caso de las RNA esta intensidad la determinan los ***pesos sinápticos*** (o simplemente pesos). De este modo, cada entrada X_i de una neurona se encuentra afectada por un peso W_i.

La actividad que una unidad de procesamiento o neurona artificial realiza es simple. Normalmente, consiste en sumar los valores de las entradas (***inputs***) que recibe de otras

unidades conectadas a ella, comparar esta cantidad con el valor ***umbral*** y, si lo iguala o supera, enviar activación o salida (***output***) a las unidades a las que esté conectada.

Tanto las entradas que la unidad recibe como las salidas que envía dependen a su vez del peso o fuerza de las conexiones por las cuales se realizan dichas operaciones.

Una RNA se compone básicamente de tres partes:

a. Entradas.
b. Núcleo.
c. Salidas.

Entradas.

Reciben los datos o parámetros que le permiten decidir a la neurona si estará activa o no, normalmente se representan como X_1, X_2,…, X_n.

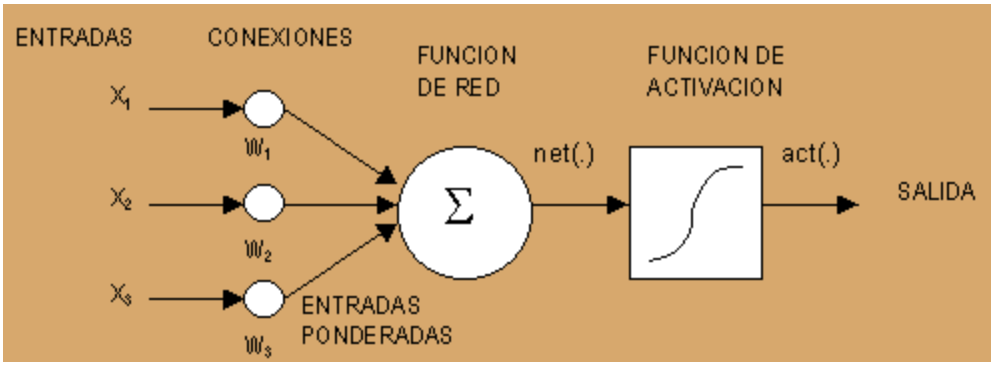
Entre la entrada y el núcleo aparecen los pesos (W_1, W_2,…, W_n), que representan la memoria de la red.

Núcleo.

Aquí se realizan todas las operaciones necesarias para determinar la salida de la neurona. El proceso que se realiza en el núcleo varía dependiendo de la red neuronal que se esté trabajando. En el núcleo se utilizan las siguientes tres funciones:

Función de propagación: también llamada ***función de excitación*** que por lo general consiste en la sumatoria de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina ***excitatoria***; si es negativo, se denomina ***inhibitoria***.

Esta función calcula el valor de base o entrada total a la unidad, generalmente como simple suma ponderada de todas las entradas recibidas, es decir, de las entradas multiplicadas por el peso o valor de las conexiones. Equivale a la combinación de las señales excitatorias e inhibitorias de las neuronas biológicas, como se observa en la siguiente figura:



Esta función permite obtener, a partir de las entradas y los pesos el valor del potencial post-sináptico hi de la neurona. La función más habitual es de tipo lineal, y se basa en la suma ponderada de las entradas con los pesos sinápticos

$$h_i\,(t) = \sum_j w_{ij} x$$

que formalmente también puede interpretarse como producto escalar de los vectores de entrada y los pesos.

Función de Activación: Es talvez la característica principal o definitoria de las neuronas, la que mejor define el comportamiento de la misma. Proporciona el estado de activación de la neurona en función del estado anterior y del valor post-sináptico.
Se usan diferente tipos de funciones, desde simples funciones de umbral a funciones no lineales. Se encarga de calcular el nivel o estado de activación de la neurona en función de la entrada total. Esta función es la que modifica a la anterior. Puede existir o no existir, siendo en este caso la salida, la misma función de propagación.

Función de Transferencia: que se aplica al valor devuelto por la función de activación. Se le conoce también como "función de salida". Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas. Algunas de las más utilizadas son la **función sigmoidea**, que son para obtener valores en el intervalo [0, 1] y la tangente hiperbólica en el intervalo [-1, 1].

Salidas.
Devuelven la respuesta de la neurona, es decir si está activa o no, representadas comúnmente como Y_1, Y_2,…, Y_n.

Aprendizaje en las redes neuronales.
La parte más importante de una red neuronal es el aprendizaje. El esquema de aprendizaje de una red es lo que determina el tipo de problemas que será capaz de resolver. Las redes neuronales son sistemas de aprendizaje basados en ejemplos.
La capacidad de una red para resolver un problema estará ligada de forma fundamental al tipo de ejemplos de que se dispone en el proceso de aprendizaje.
Desde el punto de vista de los ejemplos, el conjunto de aprendizaje debe poseer las siguientes características:

- *Ser significativo*. Debe haber un número suficiente de ejemplos. Si el conjunto de aprendizaje es reducido, la red no será capaz de adaptar sus pesos de forma eficaz.
- *Ser representativo*. Los componentes del conjunto de aprendizaje deberán ser diversos. Si un conjunto de aprendizaje tiene muchos más ejemplos de un tipo que del resto, la red se especializará en dicho subconjunto de datos y no será de aplicación general. Es importante que todas las regiones significativas del espacio de estados estén suficientemente representadas en el conjunto de aprendizaje.

El aprendizaje en una RNA consiste en la determinación de los valores precisos de los pesos para todas sus conexiones, que la capacite para la resolución eficiente de un problema.
El proceso general de aprendizaje consiste en ir introduciendo paulatinamente todos los ejemplos del conjunto de aprendizaje, y modificar los pesos de las conexiones siguiendo un determinado esquema de aprendizaje. Una vez introducidos todos los ejemplos se comprueba si se ha cumplido cierto criterio de convergencia; de no ser así se repite el proceso y todos los ejemplos del conjunto vuelven a ser introducidos. La modificación de los pesos puede hacerse después de la introducción de cada ejemplo del conjunto, o una vez introducidos todos ellos.

El Perceptrón.

Es la red de neuronas artificiales más sencilla. Está compuesta únicamente por una capa de neuronas de entrada y otra capa de neuronas de salida.
Si bien este tipo de red apenas se emplea en la actualidad por las limitaciones que presenta, su estudio es obligado dado que es la base de los métodos actuales.
Una neurona sola y aislada carece de razón de ser, su labor especializada se torna valiosa en la medida en que se asocia a otras neuronas, formando una Red.
El modelo de operación Neuronal presentado en 1943 por Warren McCulloch y Walter Pitts fue el primero, este planteaba que una red neuronal podía estudiarse mediante la lógica proposicional y que cada neurona tenía un comportamiento biestable, al producir salidas todo-nada, asemejándose al comportamiento también biestático de los conmutadores eléctricos, abierto-cerrado.
La característica principal del modelo Neuronal de Warren-Pitts era que la suma ponderada de las señales de entrada, fuera comparada con un umbral para determinar la salida de la neurona, de manera que cuando la suma fuera mayor o igual al umbral, la salida sería igual a 1, y cuando la suma fuera menor que el umbral, la salida seria 0.

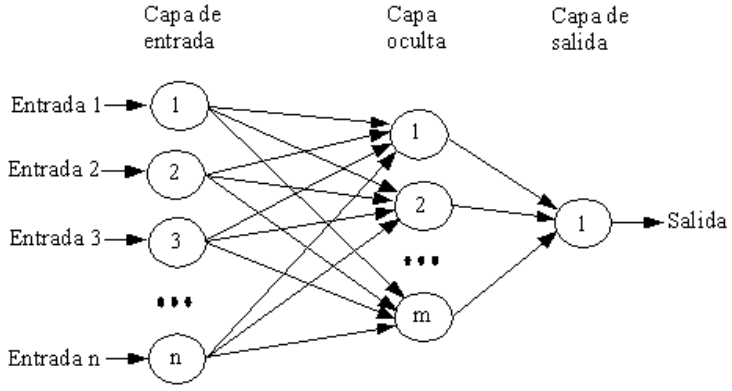
Este modelo tenía limitaciones, pero a pesar de esto fue tomado como punto de partida para el desarrollo de muchos de los modelos neuronales actuales y como punto de referencia para evaluar el comportamiento de otros modelos.

Con el paso de tiempo se agregaron características adicionales, y una de ellas fue propuesta por Frank Rosenblatt, psicólogo notable en el campo de la Inteligencia Artificial, quien rechazó el uso que McCulloch y Pitts hicieron de la lógica simbólica aplicada a las redes, y defendió métodos probabilísticos, de manera que llamó Perceptrones a unas redes McCulloch-Pitts capaces de modificar los pesos de sus conexiones, si las respuestas de la red no eran las correctas, y demostró que estas redes se podían entrenar para clasificar ciertos patrones en iguales o distintos.

Básicamente la característica que él le agrego, era ***la capacidad de aprender***, bajo una regla de aprendizaje simple y que permitía que automáticamente la red aprendiera de sus errores; esta regla expresaba lo siguiente: "*La regla de aprendizaje simple convergirá a los pesos correctos de la Red si es que existen los pesos que solucionan dicho problema*".

Por lo tanto, una red con todas las entradas conectadas directamente a las salidas se denomina Red Neuronal de una sola capa, o Red Perceptrón.

En esta, las entradas se pasan primero a través de algunos pre-procesadores, que comúnmente se llaman ***unidades de asociación***, y quienes son las encargadas de detectar la presencia de ciertas características en las entradas, además, cada unidad de salida es independiente de las otras y cada peso afecta solo a una de las salidas; como observamos en la siguiente figura:



El Perceptrón usa una matriz para representar las redes neuronales, y esta es un discriminador terciario que traza su entrada "X", siendo este un vector binario, hacia un único valor de salida (x), es decir, un solo valor binario, a través de dicha matriz.

$$f(x) = \begin{cases} 1 & \text{si } w \cdot x - u > 0 \\ 0 & \text{en otro caso} \end{cases}$$

Donde "*w"* es un vector de pesos reales y "*w.x*" es el producto escalar, que calcula una suma ponderada, "*u*" es el umbral, el cual representa el grado de inhibición de la neurona, y es un término constante que no depende del valor que tome la entrada.

Funcionamiento del Perceptrón.

El Perceptrón es una red capaz de aprender. En su configuración inicial a los pesos de las conexiones se les da valores arbitrarios, por lo que ante la presencia de estímulos, la red genera respuestas arbitrarias, respuestas que no coinciden con las deseadas, y no es hasta que los pesos se han ajustado de tal modo que la respuesta que emite es la deseada, que se considera que la red ha conseguido aprender.

El procedimiento propuesto por Rosenblatt para este entrenamiento era sencillo: se le presenta a la red un patrón cuya señal se transmite hasta la capa de salida, provocando la activación de alguna de sus unidades, si se activan las unidades de respuesta correcta, no se hace ningún ajuste de sus pesos, pero si la respuesta es incorrecta se procede de la manera siguiente: si la unidad debía estar activada y no lo está, aumentar todos los pesos de sus conexiones; si la unidad debía estar desactivada y está activada, disminuir los pesos de sus conexiones. Se repite este procedimiento con todos los patrones deseados de estímulo-respuesta.

Puntualmente, Rosenblatt creyó que era posible hacer que los pesos convergieran en un conjunto de valores, a partir de los cuales le era posible a la red computar cada uno de los patrones de entrada para producir los correspondientes patrones de salida.

El algoritmo de aprendizaje es de tipo supervisado, lo que requiere que sus resultados sean evaluados y se realicen oportunas modificaciones de los pesos si fuera necesario.

Los pasos necesarios para el entrenamiento del Perceptrón son:

1. *Inicialización de los pesos y del umbral.*
   Inicialmente se asignan valores aleatorios a cada uno de los pesos Wi, con i = 1, 2, 3,…, n de las conexiones y al umbral W_{n+1}.

2. *Presentación de nuevo par (Entrada, Salida esperada).*
   Presentar un nuevo patrón de entrada $Xp = \{X_1, X_2, \ldots, X_n\}$ junto con la salida esperada.

3. *Cálculo de la salida actual.*

$$y(t) = f\ \left[\sum(w_i x_i - \theta)\right]$$

Donde f es la función de transferencia escalón, es decir:

$$f\ (neta) = \begin{cases} 0, & f(t) < 0 \\ 1, & f(t) \geq 0 \end{cases}$$

4. *Adaptación de los pesos.*

   Es necesaria solo si la salida no corresponde a lo que se espera.

$$w_i(t+1) = w_i(t) + \alpha\,[d(t) - y(t)]\,x_i\,(t)$$

   Donde:

   W_i (t): pesos actuales

   d (t): salida deseada

   y (t): salida actual

   X_i (t): entrada del Perceptrón

   α: factor de ganancia (velocidad de aprendizaje), puede variar su valor entre 0.0 y 1.0

5. Si no hay convergencia, los pesos no cambian, volver al paso 2. Repetir hasta que la red genere una salida válida para todos los patrones que se quieren enseñar.
6. *Fin*

Procedimiento

Ejemplo 1. Se desea entrenar un Perceptrón con el algoritmo descrito anteriormente. Considerar las siguientes condiciones iniciales:

Pesos elegidos aleatoriamente: $W_0 = 1.5$, $W_1 = 0.5$, $W_2 = 1.5$, $\alpha = 1$, X_0 (umbral) = 1

Se utilizará la función OR para la salida deseada “d (t)”.

X_1	X_2	d (t) = X_1 or X_2
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Debemos considerar uno a uno los cuatro patrones de entrada y aplicar el algoritmo anteriormente descrito.

Entrenamiento para patrón $X_1 = 0$ y $X_2 = 0$

$y(t) = W_1 X_1 + W_2 X_2 + W_0 X_0$

$y(1) = 0.5(0) + 1.5(0) + 1.5(1) = 1.5$

Aplicando la función de salida y (t) tenemos que: $y(t) \geq 0$ por lo que la salida de la neurona es 1, como no es la salida deseada (de acuerdo a la tabla OR: 0, error = 0 - 1 = -1), modificamos los pesos, así:

$W_0(t+1) = W_0 + \alpha [d(t) - y(t)] X_0(t)$

$W_0(1+1) = W_0 + \alpha [d(1) - y(1)] X_0(1)$

$W_0(2) = 1.5 + 1[0 -1] * 1 =$ **0.5**

$W_1(t+1) = W_1 + \alpha [d(t) - y(t)] X_1(t)$

$W_1(1+1) = W_1 + \alpha [d(1) - y(1)] X_1(1)$

$W_1(2) = 0.5 + 1[0 -1] * 0 =$ **0.5**

$W_2(t+1) = W_2 + \alpha [d(t) - y(t)] X_2(t)$

$W_2(1+1) = W_2 + \alpha [d(1) - y(1)] X_2(1)$

$W_2(2) = 1.5 + 1[0 -1] * 0 =$ **1.5**

Con los nuevos pesos procedemos a calcular de nuevo la salida para el patrón **$X_1 = 0$ y $X_2 = 0$**

$y(t) = W_1 X_1 + W_2 X_2 + W_0 X_0$

$y(1) = 0.5(0) + 1.5(0) + 0.5(1) = 0.5$

Aplicando la función de salida y (t) tenemos que: $y(t) \geq 0$ por lo que la salida de la neurona es 1, como no es la salida deseada (de acuerdo a la tabla OR: 0, error = 0 - 1 = -1), modificamos los pesos, así:

$W_0(t+1) = W_0 + \alpha [d(t) - y(t)] X_0(t)$

$W_0(1+1) = W_0 + \alpha [d(1) - y(1)] X_0(1)$

$W_0(2) = 0.5 + 1[0 -1] * 1 =$ **-0.5**

$W_1(t+1) = W_1 + \alpha [d(t) - y(t)] X_1(t)$

$W_1 (1 +1) = W_1 + \alpha [d (1) – y (1)] X_1 (1)$
$W_1 (2) = 0.5 + 1 [0 -1] * 0 =$ **0.5**

$W_2 (t + 1) = W_2 + \alpha [d (t) – y (t)] X_2 (t)$
$W_2 (1 +1) = W_2 + \alpha [d (1) – y (1)] X_2 (1)$
$W_2 (2) = 1.5 + 1 [0 -1] * 0 =$ **1.5**

Con los nuevos pesos procedemos a calcular de nuevo la salida para el patrón **$X_1 = 0$ y $X_2 = 0$**
$y (t) = W_1 X_1 + W_2 X_2 + W_0 X_0$
$y (1) = 0.5 (0) + 1.5 (0) + -0.5 (1) = - 0.5$

Aplicando la función de salida y (t) tenemos que: y (t) < 0 por lo que la salida de la neurona es 0, como es la salida deseada (de acuerdo a la tabla OR: 0, error = 0 - 0 = 0), no modificamos los pesos y continuamos con el siguiente patrón.

Entrenamiento para patrón $X_1 = 0$ y $X_2 = 1$
$y (t) = W_1 X_1 + W_2 X_2 + W_0 X_0$
$y (2) = 0.5 (0) + 1.5 (1) + -0.5 (1) = 1$

Aplicando la función de salida y (t) tenemos que: y (t) ≥ 0 por lo que la salida de la neurona es 1, como es la salida deseada (de acuerdo a la tabla OR: 1, error = 1 - 1 = 0), no modificamos los pesos y continuamos con el siguiente patrón.

Entrenamiento para patrón $X_1 = 1$ y $X_2 = 0$
$y (t) = W_1 X_1 + W_2 X_2 + W_0 X_0$
$y (3) = 0.5 (1) + 1.5 (0) + -0.5 (1) = 0$

Aplicando la función de salida y (t) tenemos que: y (t) ≥ 0 por lo que la salida de la neurona es 1, como es la salida deseada (de acuerdo a la tabla OR: 1, error = 1 - 1 = 0), no modificamos los pesos y continuamos con el siguiente patrón.

Entrenamiento para patrón $X_1 = 1$ y $X_2 = 1$
$y (t) = W_1 X_1 + W_2 X_2 + W_0 X_0$
$y (4) = 0.5 (1) + 1.5 (1) + -0.5 (1) = 1.5$

Aplicando la función de salida y (t) tenemos que: y (t) $\geq$ 0 por lo que la salida de la neurona es 1, como es la salida deseada (de acuerdo a la tabla OR: 1, error = 1 - 1 = 0), no modificamos los pesos y continuamos con el siguiente patrón.

Como ya no hay patrones, tenemos que los pesos correctos después del entrenamiento son:

$$W_0 = -0.5,\ W_1 = 0.5,\ W_2 = 1.5$$

Con estos nuevos pesos los patrones de entrada coinciden con las salidas, ya no se comete ningún error y por lo tanto la etapa de aprendizaje concluye.

Análisis de resultados

Tomando como referencia la información presentada en el ejemplo No.1, implementar un simulador (en entorno de Windows Forms) para el Perceptrón, que simule el entrenamiento del mismo.

Considerar la siguiente funcionalidad para el simulador:

- Permitir que el usuario proporcione los datos iniciales: los pesos W_0, W_1 y W_2, el factor de ganancia "α" y el umbral (X_0)
- El aprendizaje del Perceptrón debe realizarse utilizando el algoritmo de Rosenblatt (estudiado en la guía) y la función OR para la salida deseada "d (t)".
- Diseñar la interfaz considerando que debe mostrarse todos los pasos del entrenamiento realizado por la neurona Perceptrón.

Investigación Complementaria

Para la siguiente semana:

Investigar sobre el funcionamiento del Perceptrón multinivel (investigación bibliográfica) y realizar su implementación a través de un simulador en entorno de Windows Forms, que simule el entrenamiento del mismo. Considerar todas las validaciones necesarias. La funcionalidad deberá implementarse tomando en cuenta las mismas consideraciones del Perceptrón simple.

Guía 8: Redes Neuronales. Perceptrón.

Hoja de cotejo: 8

Alumno:

Máquina No:

Docente:

GL:

Fecha:

EVALUACIÓN					
	%	1-4	5-7	8-10	Nota
CONOCIMIENTO	Del 20 al 30%	Conocimiento deficiente de los fundamentos teóricos	Conocimiento y explicación incompleta de los fundamentos teóricos	Conocimiento completo y explicación clara de los fundamentos teóricos	
APLICACIÓN DEL CONOCIMIENTO	Del 40% al 60%				
ACTITUD	Del 15% al 30%	No tiene actitud proactiva.	Actitud propositiva y con propuestas no aplicables al contenido de la guía.	Tiene actitud proactiva y sus propuestas son concretas.	
TOTAL	100%				