

2. APLICACION DE TECNICAS DE INTELIGENCIA ARTIFICIAL PARA EL MODELAMIENTO MATEMATICO DE FENOMENOS

2.1. REDES NEURONALES:

En esencia una Red Neuronal es un modelo matemático acerca del funcionamiento del cerebro en donde los sencillos elementos de cálculo aritmético equivalen a las neuronas (células que procesan información en el cerebro) y la red en general equivale a un conjunto de neuronas conectadas entre sí. [11], [55]

Además de sus útiles propiedades de cómputo, las redes neuronales son idóneas para el estudio de diversos fenómenos complejos que requieren simular las capacidades funcionales del cerebro.

Se ha demostrado matemáticamente, Funahashi (1989, 1990), que una red neuronal multicapa puede aproximar cualquier función no lineal o mapa no lineal multivariable [12].

$$f(x) : R^n \rightarrow R$$

Ecuación 5 Aproximación de una función no lineal

Donde:

$F(x)$ representa la función no lineal

R^n representa la dimensión del conjunto a la que pertenece la función.

R representa la nueva dimensión de la función. La función ha sido aproximada.

Por lo que puede servir para generar un modelo matemático que pueda precalcular daños por desastres naturales que afectan a comunidades vulnerables.

2.2. APLICACIONES DE LAS REDES NEURONALES:

Jhon Denker afirma que “las redes neuronales es la segunda mejor opción para hacer cualquier cosa “, aunque podría parecer una exageración, es un hecho que el desempeño de las redes neuronales en muchas tareas que resultarían difíciles de realizar mediante otras técnicas de programación puede considerarse como aceptable

La aplicación de las Redes Neuronales ha sido muy eficiente en el reconocimiento de patrones, el análisis de series de tiempo, el reconocimiento de caracteres escritos, estudios financieros, verificación de firmas, procesamiento de imágenes y control de pequeños robots. [11], [12].

A continuación se describen algunas aplicaciones:

El programa NETtalk (Sejnowski y Rosenberg, 1987) es una Red Neuronal que aprende a pronunciar un texto escrito. La entrada es una secuencia de caracteres que se presenta en una ventana que va deslizándose a través del texto. La capacitación consistió de un texto de 10024 palabras cuyos rasgos fonémicos se transcribieron en forma manual. NETtalk aprende a realizar su tarea con un 95% de precisión en el escenario de la capacitación luego de revisar 50 veces los datos.

ALVINN (Autonomous Land Vehicle In a Neural Network, vehículo terrestre autónomo en una red neuronal) (Pomerleau, 1993) es una red neuronal cuyo desempeño ha sido bastante satisfactorio en dominios donde otros métodos han fracasado. Es capaz de aprender a conducir un vehículo en un solo carril basándose en la actuación de un conductor humano. Los resultados obtenidos con la capacitación de la Red son impresionantes. ALVINN ha conducido a velocidades superiores a los 105 kilómetros por hora, a través de más de 135 km, en carreteras públicas cercanas a Pittsburgh. También ha conducido a velocidades normales en caminos de terracería.

de un solo carril, caminos pavimentados para bicicletas y calles de dos carriles de los suburbios.

Se entrenó una Red Neuronal para aprender a predecir la pluviosidad en la Sabana de Bogotá. La serie está expresada en “días de lluvia/mes” y se extiende desde 1972 hasta 1990 (19 años), los datos se normalizan dividiendo por 30. Después de múltiples análisis la estructura seleccionada fue una Red Multicapa de 40 entradas, 12 neuronas en la capa intermedia y 12 neuronas en la capa de salida, con un error de generalización promedio de 10.38 días/mes.

La compañía IBM en Escocia ha desarrollado un sistema neurona de inspección para detectar imperfecciones muy pequeñas para el ojo humano en circuitos impresos. Este sistema desarrollado en el Reino Unido se introducirá en Japón y en Estados Unidos.

La calidad de la soldadura es crítica en el ensamblaje de circuitos impresos para la industria de computadoras. La tarjeta principal de una computadora personal puede tener hasta 4000 puntos de soldadura y cuando se producen 5000 tarjetas diarias, el control de calidad es un problema importante.

La industria de alimentos utiliza bombas para transportar productos líquidos entre equipos de proceso. Aunque el proceso es robusto, cuando ocurren bloqueos, el producto que pasa por la bomba se puede dañar. La Universidad de Brunel y Neural Technologies Limited han desarrollado un sistema para detección temprana de fallas en bombas utilizando una red neuronal. El sistema contiene una Red Neuronal de 256

entradas, 10 neuronas en la capa intermedia y una salida y un micrófono montado cerca a la bomba, la salida de la Red activa una luz de alarma cuando se detecta un posible bloqueo. El sonido del micrófono se procesa antes de entregarlo a la Red Neuronal, la señal análoga del sonido se convierte a digital y luego se calcula el nivel de sonido en 256 bandas de frecuencia (espectro de frecuencia).

Varias organizaciones financieras han encontrado donde aplicar el cómputo neuronal efectivamente. El fondo de pensiones Deere y Company ha estado manejando un portafolio de US 100 millones desde Diciembre de 1993 usando Redes Neuronales.

El Fondo supervisa 1000 acciones en los Estados Unidos semanalmente, para cada una de las acciones hay una red neuronal que modela el futuro de la acción como función de 40 factores técnicos y producen un estimado del cambio del precio semanal. La compañía selecciona luego un portafolio de las 1000 mejores acciones y distribuye los dineros del fondo proporcionalmente El programa de Deere y Co. Está soportado por 4 personas, empezó negocios en Diciembre de 1993 y sus ganancias son del 30% lo cual es un buen margen según el estándar financiero.

2.3. RED MULTICAPA:

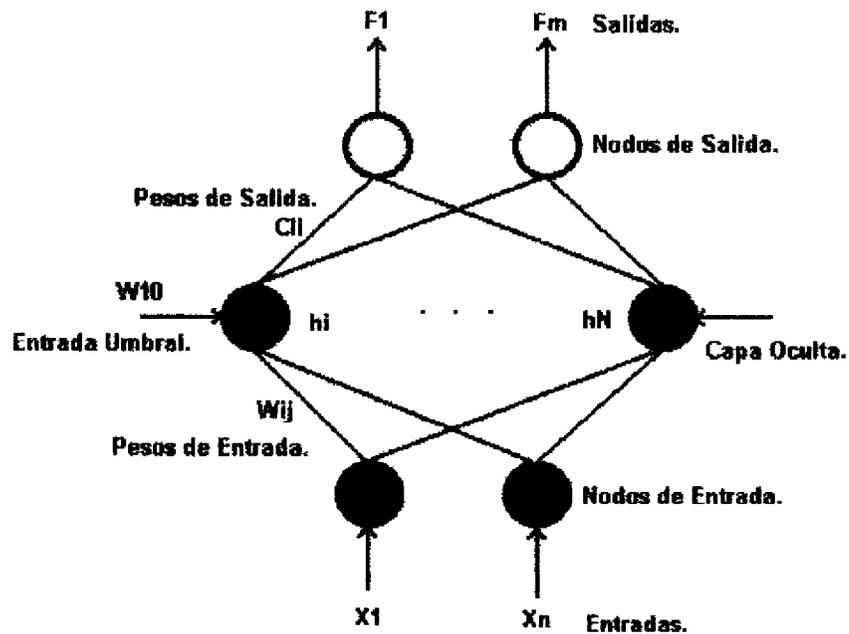


FIGURA 8. Esquema de una Red Neuronal Multicapa

Una red neuronal es una estructura paralela que procesa información en forma distribuida y está constituida por elementos de procesamiento, llamados neuronas, interconectados con canales unidireccionales. Cada elemento de procesamiento tiene una conexión de salida, la cual se ramifica en tantas conexiones colaterales como se requiera (cada una lleva la misma señal, la salida del elemento procesador) [12],[13] [14],[15].

Aquí se muestra una red neuronal multicapa, la estructura es n-N-m. Hay n nodos de entrada, N neuronas ocultas y m neuronas de salida. Los nodos de entrada reciben los valores a procesar x_j ($j=1\dots n$), las neuronas ocultas reciben las entradas x_j ponderadas por los pesos w_{ij} y producen las salidas h_i ($i=1\dots N$), finalmente los nodos de salida hacen la suma ponderada de las salidas de las neuronas ocultas

La red neuronal multicapa puede resolver problemas de clasificación de patrones no separables linealmente como la función lógica XOR

Para la red mostrada, la salida de las neuronas ocultas está dada por [12], [13], [14],:

$$h_i = \sigma(s_i) = \sigma\left(\sum_{j=1}^n w_{ij} \cdot x_j + w_{i0}\right)$$

Ecuación 6. Salida de las neuronas ocultas

Donde.

x_j representa las condiciones de vulnerabilidad de la comunidad

w_{ij} representa el valor de los pesos de entrada

w_{i0} representa el valor de los pesos umbrales de entrada.

σ representa la función Tanh, que es la función de activación en la primera capa de la Red Neuronal.

Las salidas de la red son:

$$f_l = \sum_{i=1}^N c_{li} \cdot h_i$$

Ecuación 7 Salidas de la Red Neuronal.

Donde:

F_L representa las consecuencias y los daños producidos por la acción del desastres.

h , representa el valor de las salidas de la primer capa de la Red Neuronal.

C_{Lj} representa el valor de los pesos de salida.

2.4. ENTRENAMIENTO CON EL ALGORITMO DE PROPAGACION INVERSA RED MULTICAPA:

Para entrenar una Red Neuronal se asume que las neuronas tienen una función de activación continua y se utiliza el gradiente de la función de error para actualizar el vector de pesos.

Si consideramos la red multicapa 3-3-2 se utilizará la ley general de aprendizaje, basada en gradiente, [12], [13], [15], [16], [17]:

$$w_{k+1} = w_k + \alpha \cdot (-\nabla_k)$$

Ecuación 8. Ley general del aprendizaje basada en el gradiente.

Donde:

w_{k+1} representa el nuevo valor del peso

w_k representa el valor del peso sin actualizar

α representa la rata de aprendizaje del sistema.

$-\nabla_k$ representa el valor del gradiente.

Las entradas a la red son x_1 , x_2 y x_3 , y las correspondientes salidas deseadas son d_1 , d_2 , un patrón entradas salidas es el vector $(x_1, x_2, x_3, d_1, d_2)$. Las salidas de la red son y_1, y_2 , por lo tanto los errores entre salidas deseadas y salidas actuales son: $e_1 = d_1 - y_1$, $e_2 = d_2 - y_2$.

Durante la presentación de un patrón, el error cuadrático de la salida está dado por:

$$e^2 = (d_1 - y_1)^2 + (d_2 - y_2)^2$$

Ecuaciones 9. Error cuadrático de la salida.

Donde,

$$y_1 = c_{11} \cdot h_1 + c_{12} h_2 + c_{13} \cdot h_3$$

$$y_2 = c_{21} \cdot h_1 + c_{22} h_2 + c_{23} \cdot h_3$$

e representa el valor del error.

d_1 representa el valor deseado de la salida en 1.

d_2 representa el valor deseado de la salida en 2.

Y_1 y Y_2 es el valor de salida real de la Red Neuronal.

h_1, h_2, h_3 representan los valores de las salidas de la primer capa de la Red Neuronal.

C representa los valores de los pesos de salida.

Estas ecuaciones representan las entradas, las salidas y el error cuadrático para una Red Neuronal de dos salidas y una capa intermedia de tres neuronas.

El error es cero si los errores para cada salida son cero. Calculando el gradiente del error cuadrático respecto de los pesos de salida de la red neuronal tenemos,

$$\frac{\partial e^2}{\partial c_{li}} = -2.e_l.h_i$$

Ecuación 10. Gradiente del error cuadrático.

Donde:

e_l representa el valor del error.

h_i representan los valores de las salidas de la primer capa de la Red Neuronal.

C representa los valores de los pesos de salida.

Entonces la regla para actualizar los pesos de salida es,

$$c_{li}^{k+1} = c_{li}^k + 2.\alpha.e_l.h_i$$

Ecuación 11. Regla para actualizar los pesos de salida.

Donde:

e_l representa el valor del error.

h_i representan los valores de las salidas de la primer capa de la Red Neuronal.

C representa los valores de los pesos de salida.

α representa la tasa de aprendizaje del sistema.

$l=1 \dots m$.

$i=1 \dots N$.

La salida de las neuronas ocultas está dada por,

$$h_1 = \sigma(s_1) = \sigma(w_{11} \cdot x_1 + w_{12} \cdot x_2 + w_{13} \cdot x_3 + w_{10}),$$

$$h_2 = \sigma(s_2) = \sigma(w_{21} \cdot x_1 + w_{22} \cdot x_2 + w_{23} \cdot x_3 + w_{20}),$$

$$h_3 = \sigma(s_3) = \sigma(w_{31} \cdot x_1 + w_{32} \cdot x_2 + w_{33} \cdot x_3 + w_{30}),$$

Ecuaciones 12. Cálculo de las salidas de las neuronas ocultas

Donde.

h_i representa el valor de las salidas de la primer capa de la Red Neuronal

X_j representa las condiciones de vulnerabilidad de la comunidad.

W_{ij} representa el valor de los pesos de entrada.

σ representa la función Tanh, que es la función de activación en la primera capa de la Red Neuronal.

El gradiente del error cuadrático respecto de los pesos de la capa oculta de la red neuronal es,

$$\frac{\partial e^2}{\partial w_{ij}} = -2 \cdot \delta_i \cdot \sigma'(s_i) \cdot x_j,$$

Ecuación 13. Cálculo del gradiente del error cuadrático respecto a los pesos de la capa oculta de la red neuronal.

Donde el error especial es:

$$\delta_i = \sum_{l=1}^m c_{li} e_l = \sum_{l=1}^m c_{li} \cdot (d_l - y_l),$$

Ecuación 14. Error especial.

Donde:

δ representa el valor del error especial.

e representa el valor del error

d_l representa el valor deseado de la salida en l .

Y_l es el valor de salida real de la Red Neuronal.

C representa los valores de los pesos de salida.

Entonces la regla para actualizar los pesos de entrada es:

$$w_{ij}^{k+1} = w_{ij}^k + 2 \cdot \alpha \cdot \delta_i \cdot \sigma'(s_i) \cdot x_j,$$

Ecuación 15. Regla para actualizar los pesos de entrada.

Donde:

h_i representa el valor de las salidas de la primer capa de la Red Neuronal.

X_j representa las condiciones de vulnerabilidad de la comunidad.

W_{ij} representa el valor de los pesos de entrada.

e representa el valor del error.

h_i representan los valores de las salidas de la primer capa de la Red Neuronal.

C representa los valores de los pesos de salida.

α representa la rata de aprendizaje del sistema.

δ representa el valor del error especial.

$l=1 \dots m$.

$i=1 \dots N$.

σ representa la función Tanh, que es la función de activación en la primera capa de la Red Neuronal.