

Centro de Ciencias Básicas

Ciencias de la Computación

Aprendizaje Inteligente

Examen primer parcial.

**Ingeniería en Computación Inteligente semestre 6°A**

**Profesor: Francisco Javier Luna Rosas**

Alumnos:

J.Horacio Cervantes P.

¿Qué es la técnica de propagación hacia atrás (FeedForward)?.Es un método de cálculo del gradiente utilizado en algoritmos de aprendizaje supervisado utilizados para entrenar redes neuronales artificiales. El método emplea un ciclo propagación – adaptación de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas siguientes de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Multicapas de las redes neuronales Feed-Forward.

Cada capa de la red neuronal Feed-Forward se ve compuesta por un Perceptrón multicapa la cual es una red con alimentación hacia delante, compuesta de varias capas de neuronas entre la entrada y la salida de esta, llegando a una salida en común

Topologías de las redes neuronales Feed-Forward.

Cada neurona de la red recibe información a través de las conexiones con las neuronas de la capa anterior, procesa la información y emite un resultado a través de sus conexiones con las neuronas de la capa siguiente, conectándose a través de su sinapsis las cuales son direccionales solo pueden propagarse hacia un solo sentido, principalmente formando una característica de que se forma un cuello de botella, donde hay varias neuronas de capa oculta y después las neuronas se van reduciendo hasta llegar a la salida.

dividiéndose en la capa de entrada, la cual no hace ningún procesamiento solo realiza una propagación de las entradas, las capas ocultas las cuales están entre la capa de entrada y salida, estas son las que realizan los procesamientos y por ultimo la capa de salida la cual proporciona la salida de los datos.

Aplicaciones de las redes neuronales Feed-Forward.

Pueden servir para el reconocimiento de patrones, para el reconocimiento de voz, para el reconocimiento de la escritura, visón computacional.

En que consiste los dataset de entrenamiento y sus valores target

asociados.

El conjunto de datos de entrenamiento es el que se utiliza para entrenar a un Algoritmo, para aprender y producir resultados. Incluye tanto los datos de entrada como los resultados de objetivo o target y este ultimo es el que se busca replicar. En algunos modelos el error se calcula mediante las entradas del dataset.

La razón de aprendizaje de las redes neuronales Feed-Forward.

Las redes neuornales feed-forward aprenden debido al calculo del error entre los datos calculados que se obtienen como salida y los datos de salida objetivos, utilizándolo para crear un ajuste de ponderaciones o valores.

Bueno no sé, si se refería a lo anterior o a la tasa de aprendizaje así que pondere ambas, La tasa de aprendizaje controla la rapidez con la que el modelo se adapta al problema. Las tasas de aprendizaje más pequeñas requieren más épocas de entrenamiento dados los cambios más pequeños realizados en los pesos que cada actualización actualiza, mientras que las tasas de aprendizaje más grandes resultan en cambios rápidos y requieren menos épocas de entrenamiento.

El bias de las redes neuronales Feed-Forward.

Es un parámetro adicional en la red neuronal que se utiliza para ajustar la salida junto con la suma ponderada de las entradas a la neurona.

Los pesos de entrenamiento de las redes neuronales FeedForward.

En un principio los pesos son aleatorios, lo cual nos dará un resultado malo, pero estos se van ajustando en medida que cada época que realice, basándose en el error obtenido, mediante la comparación de la salida de la red y del valor objetivo establecido.

Propagar las entradas hacia adelante en base a la siguiente

formula:



En esta formula calculamos la entrada de la neurona, haciendo la sumatoria de los pesos por la entrada de la neurona, en nuestro caso sucede en nuestras dos neuronas de capa ocultas y al final con la neurona de capa de salida, se le suma el bais para el ajuste de la salida que se calcula en base a la entrada.

Texto

Descripción generada automáticamente

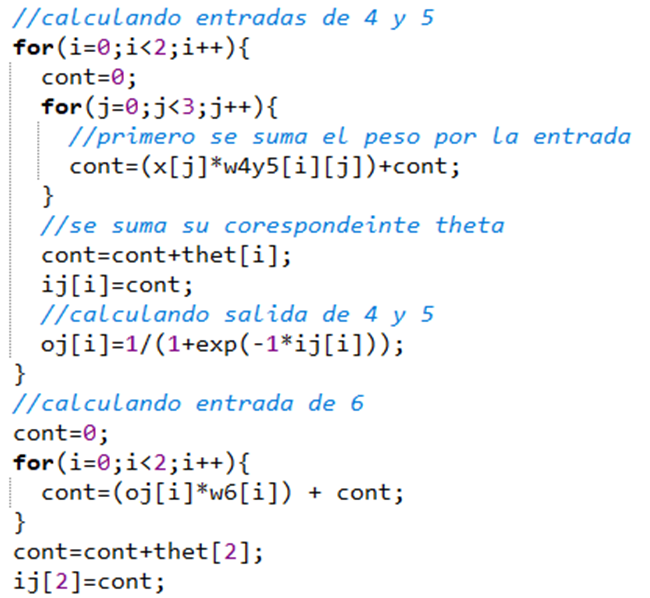
parte del código donde se calcula la entrada.

Calcular las entradas/salidas de las neuronas en base a la

siguiente formula:



A través de la entrada: obtenida de la formula anterior de calcula la salida, de la capa correspondiente, de la capa oculta tambien se calcula a través de la misma fórmula.



Calcular el error de salida de la red en base a la siguiente formula:



Calcular el error de las capas oculta en base a la siguiente formula:



La primera formula marca la salida por la salida de la capa por menos uno, después se vuelve a multiplicar por el resultado objetivo o target de la red menos la salida de la red, así obtendríamos el error de la salida de la red, para el error de las salidas de las capas ocultas seria bastante similar solo que ahora en ves de multiplicar por el target menos la salida de la capa, ahora es la sumatoria del error de las capas (en nuestro caso sería el mismo de la salida) por el peso de las capas ocultas.

Parte del código que implementa esto.

Texto, Carta

Descripción generada automáticamente

Ajustes de pesos en base a la siguiente formula:

Texto

Descripción generada automáticamente

La forma en la que se actualiza el error es a través de la multiplicación del error de la capa, multiplicando por la tasa de aprendizaje la cual en nuestro caso es de .9 y finalmente por la salida de la capa, en el caso de los pesos de la capa de entrada, serian en base a las entradas, ya en el caso de los peso de las capas ocultas seria la salida de dicha neurona. Finalmente, se le sumaria el peso anterior.

Parte del código donde se realiza

Imagen de la pantalla de un celular de un mensaje en letras blancas

Descripción generada automáticamente con confianza baja

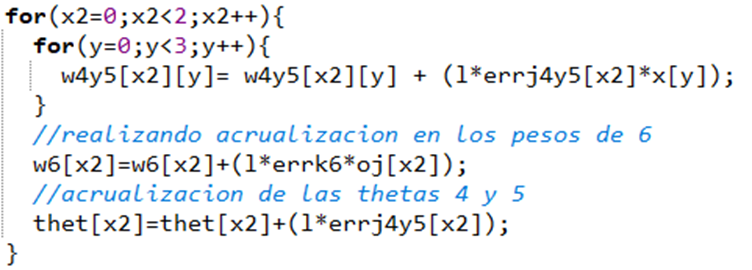
Ajustes del Bias en base a la siguiente formula:

Texto

Descripción generada automáticamente

Al igual que el peso el bias se actualiza multiplicando la tasa de aprendizaje y el error de la capa, después sele suma el bias anterior.

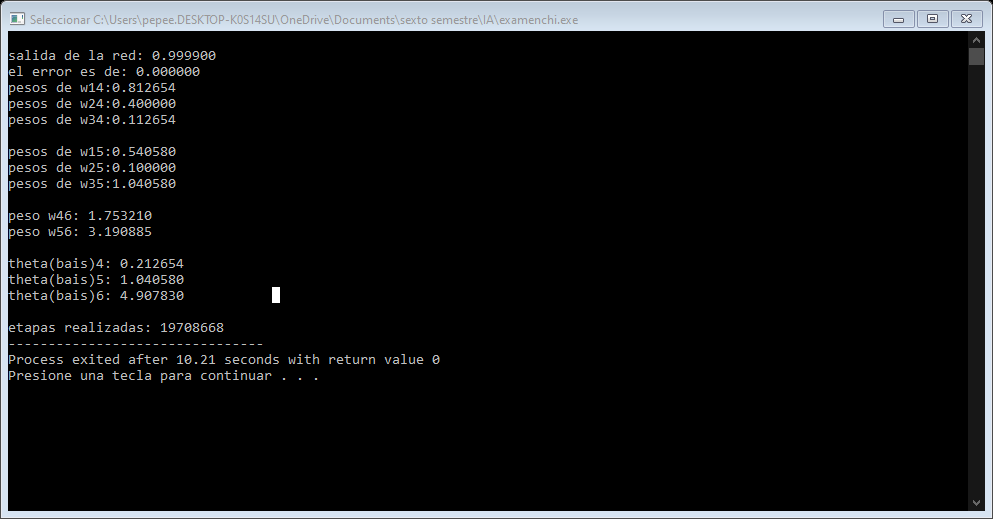




**Implementación:**

Bueno la implementación nos dio los siguientes resultados se usaron los valores previamente establecidos, como se menciono anteriormente la tasa de aprendizaje es de .9.

Ubo un fallo de ortografia y no son estapas, serian epocas.



No se alcanzo el target, tardaba demasiado tiempo posiblemente debido a la tasa de aprendizaje se quedo atrapado en un subóptimo, pero alcanzo un buen valor de salida prácticamente alcanzo el valor objetivo que era uno.