

Centro de Ciencias Básicas

Aprendizaje Inteligente
Profesor: Dr. Francisco Javier Luna Rosas

Examen 1er Parcial

Ingeniería en Computación Inteligente
Semestre 6° A

Integrantes:

Alegría Romero Dante Alejandro
Aranda Gonzalez Diego Alberto
Balandrán Félix Andrea Margarita
Moreno Sánchez Diego Emilio

Introducción

Entendamos una red neuronal como un conjunto de entradas y salidas con pesos asociados que, a través de diversos cálculos, comienzan a prever acciones específicas, generando así un modelo inspirado en el sistema nervioso humano y dotando a estas redes artificiales de su propio conocimiento.

A lo largo de este curso, hemos adquirido un considerable conocimiento, tanto a través de ejemplos prácticos como teóricos, sobre cómo funcionan estas redes y cómo se integran de manera adecuada. En el siguiente archivo, se presentarán las redes neuronales Feed-Forward, se detallará la generación de sus cálculos manualmente, y, sobre todo, exploraremos la implementación de una de ellas en Python.

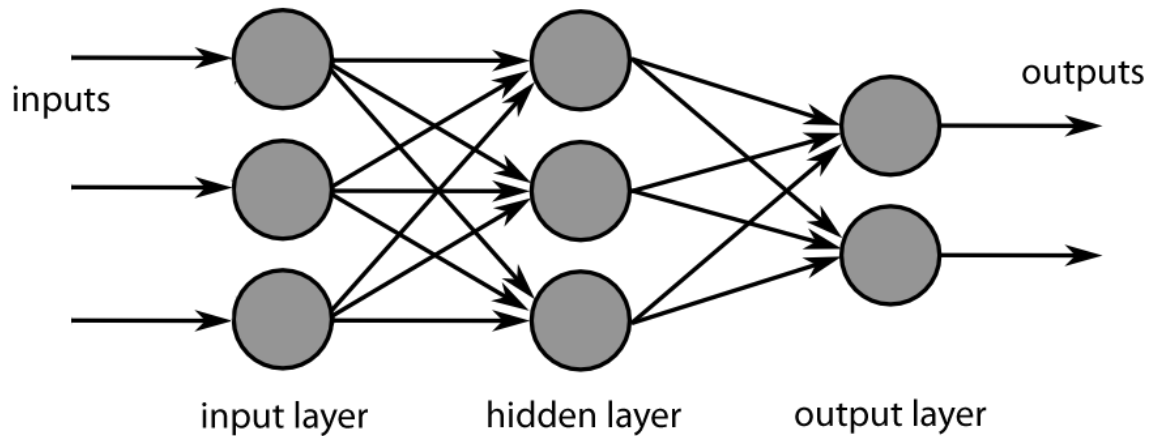
¿Qué es la técnica de propagación hacia atrás (FeedForward)?

Entendamos la propagación hacia atrás como uno de los métodos más clásicos en la implementación de redes neuronales. En términos generales, este método se basa en evaluar la función de la tasa de error obtenida en la época anterior. A través del ajuste de los pesos en cada capa de la red, se busca reducir la tasa de error y obtener resultados más precisos.

Ahora bien, ¿por qué es necesaria la propagación hacia atrás? Como ya se ha mencionado, esta técnica es una de las más utilizadas en el contexto de redes neuronales, y con razón. Es más sencilla y fácil de comprender en comparación con otras técnicas. Además, no requiere ajustes de parámetros adicionales, aparte de los relacionados con los números de entrada. Su flexibilidad es destacable, ya que no demanda conocimientos previos específicos para aplicarla a una red.

En términos generales, las redes feed-forward cuentan con entradas y salidas, como ya se ha mencionado. No obstante, también incorporan capas ocultas que desempeñan un papel crucial. Estas capas permiten que la red aprenda de manera jerárquica los datos y realice cálculos basados en las entradas de la capa anterior y los pesos asociados.

Explique la topología (arquitectura) de las redes neuronales Feed-Forward



1- Pesos

Cada uno de los nodos contiene un peso que se le asigna y que no ayudará para realizar distintos cálculos

2- Tasa De Aprendizaje

valor positivo que determina la magnitud de los ajustes aplicados a los pesos de la red durante el entrenamiento

3- Bias

Parámetros de adicionales para ajustar la salida de cada neurona

4- Capas de entrada

Contienen los predictores

5- Capas ocultas

compuesta por nodos (neuronas) que realizan cálculos basados en los pesos y activaciones de la capa anterior.

6- Capas de salida

Contienen las respuestas, donde cada unidad de salida es una función de las capas ocultas y puede representar la salida final del modelo

7- cálculos de error

Se calcula la diferencia entre las predicciones y los valores reales.

8- Ajustes de pesos

Hacer estos ajustes, ayudan a la red a ser más precisa en los siguientes cálculos

9- Función de activación

Esta ayuda a que no existan linealidades, esto hace que la red genere relaciones más complejas.

Aplicaciones de las redes neuronales Feed-Forward

Hoy en día, las redes neuronales representan una solución integral para abordar problemáticas que han afectado al ser humano a lo largo de los siglos. Su capacidad para mejorar la eficiencia y rapidez en la resolución de problemas que antes podrían haber generado complicaciones es evidente en diversas aplicaciones. A continuación, se presentan algunos ejemplos de estas aplicaciones:

- Obtención de diagnósticos rápidos para la detección eficiente de fallos en motores de vehículos a gasolina, optimizando así el uso de recursos.
- Implementación de diagnósticos para la detección de fallas en motores de encendido provocado mediante la aplicación de lógica difusa.
- Reconocimiento de patrones en una amplia variedad de contextos
- Evaluación de riesgos crediticios, facilitando la toma de decisiones en el ámbito financiero.
- Análisis de imágenes médicas para la detección temprana de enfermedades
- Predicción de enfermedades patológicas mediante el análisis de datos clínicos
- Control eficiente de procesos industriales para aumentar la productividad
- Reconocimiento por audio para aplicaciones como la identificación de comandos de voz y sistemas de asistentes virtuales.
- Identificación y análisis de grandes conjuntos de datos para extraer patrones y realizar descubrimientos valiosos.

Sin duda alguna, esta es una herramienta con la que tenemos contacto día a día y que no se ve que tenga fin, puesto que solo conocemos un poco de todo el potencial que nos puede ofrecer.

Entradas de la red

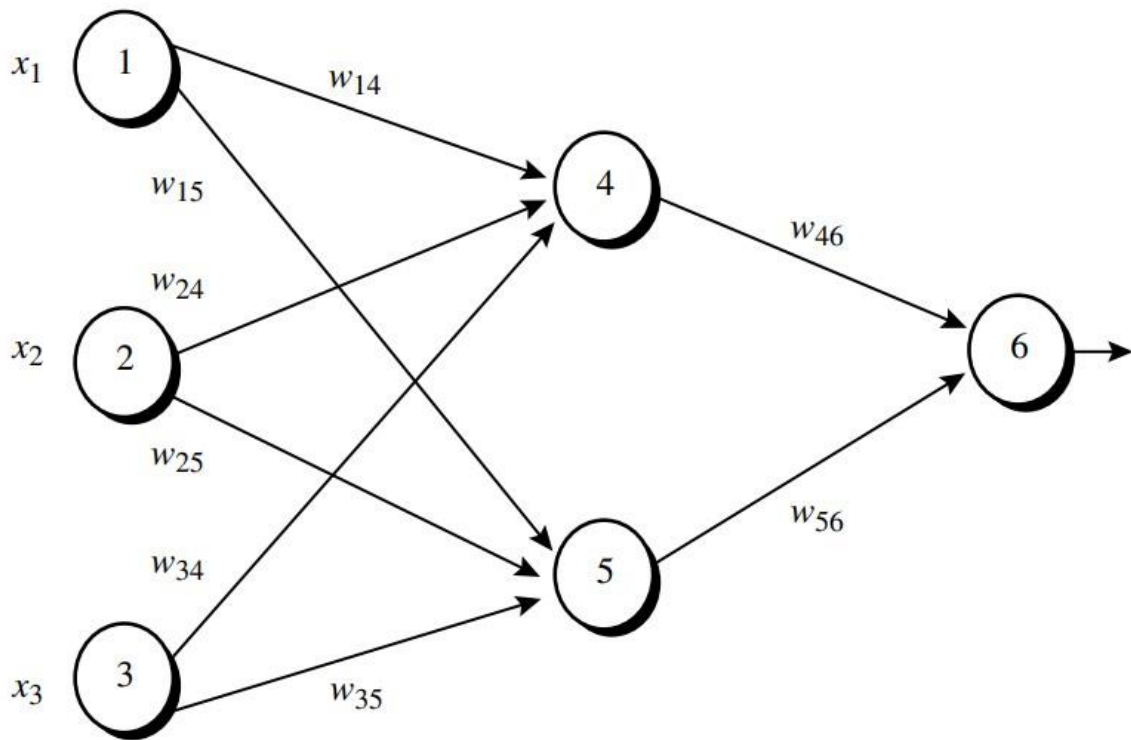
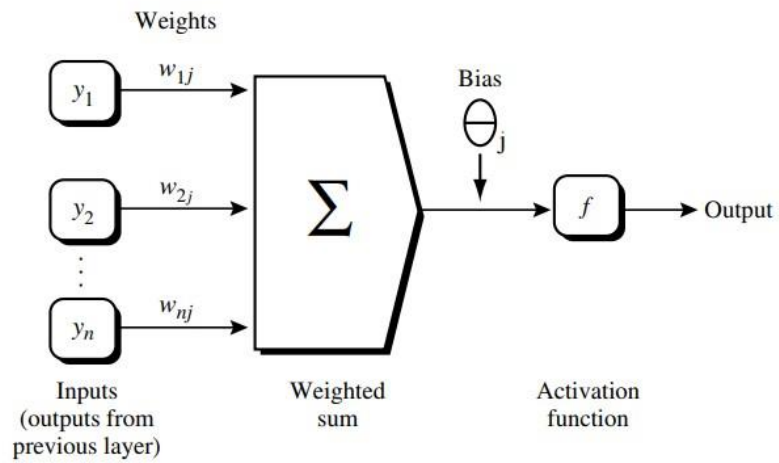
Dataset de entrenamiento, son los datos que usamos para entrenar un modelo. La calidad de nuestro modelo de aprendizaje automático va a ser directamente proporcional a la calidad de los datos, junto con los valores target que asociamos a ellos podemos entrenar el modelo para que los dataset reservados puedan llegar al target principal en el modelo ya entrenado.

El objetivo principal de aprendizaje de una red neuronal feed – forward es ajustar los pesos sinápticos para minimizar el error entre las respuestas generadas por la red y las respuestas esperada. Esto se logra mediante técnicas como la retro propagación, que actualiza los pesos sinápticos.

El bias de las redes neuronales Feed – Forward nos permite ajustar la salida de la red independientemente de las entradas. Cada neurona tiene un valor de bias. Durante el proceso de aprendizaje, tanto los pesos de las conexiones entre las neuronas como los valores de bias se ajustan para minimizar el error de la red.

Los pesos de entrenamiento de las redes neuronales Feed – Forward, son valores que determinan la importancia relativa de cada entrada en el cálculo de la salida de una neurona. Durante el entrenamiento de la red neuronal, estos pesos se ajustan para minimizar el error en la predicción de la salida.

Dada la figura 1 y 2



x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	θ_4	θ_5	θ_6
1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2	-0.4	0.2	0.1

Tenemos que:

Las fórmulas a utilizar son...

Cálculo de entradas y salidas

$$I_j = \sum_i w_{ij}(O_{ij}) + \theta_j$$

Progapación de las entradas

$$O_i = \frac{1}{1 + e^{I_j}}$$

Ajuste de pesos

$$\Delta w_{ij} = l(err_j)(O_j)$$

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

Error capas ocultas

$$err_j = O_j(I - O_j) \sum_k err_k w_{jk}$$

Calcula error de salida

$$err_j = O_j(I - O_j)(T_j - O_j)$$

Ajuste de bias

$$\Delta \theta_j = l(err_j)$$

$$\theta_j = \theta_j + \Delta \theta_j$$

Los datos a utilizar son...

$$\begin{aligned}x_1 &= 1, x_2 = 0, x_3 = 0, \\w_{14} &= 0.2, w_{15} = -0.3, w_{24} = 0.4, w_{25} = 0.1, \\w_{34} &= -0.5, w_{35} = 0.2, w_{46} = -0.3, w_{56} = -0.2, \\ \theta_4 &= -0.4, \theta_5 = 0.2, \theta_6 = 0.1\end{aligned}$$

Iniciemos calculando las salidas 4, 5, 6

$$\begin{aligned}I_4 &= (0.2)(1) + (0.4)(0) - (0.5)(1) - 0.4 = -0.7 \\I_5 &= (-0.3)(1) + (0.1)(0) + (0.2)(1) + 0.2 = 0.1 \\I_6 &= (-0.3)(0.332) + (0.2)(0.524) + 0.1 = -0.105\end{aligned}$$

Calculando el error de la salida de la red

$$err_6 = 0.474(1 - 0.474)(1 - 0.474) = 0.1311$$

Calculando los errores 4 y 5

$$\begin{aligned}err_5 &= 0.524(1 - 0.524)(0.1311)(-0.2) = -0.0065 \\err_4 &= 0.332(1 - 0.332)(0.1311)(-0.3) = -0.0087\end{aligned}$$

Ajustando pesos

$$\begin{aligned}w_{14} &= 0.2 + (0.5)(-0.0087)(1) = 0.195 \\w_{15} &= -0.3 + (0.5)(-0.0063)(1) = -0.303 \\w_{24} &= 0.4 + (0.5)(-0.0087)(0) = 0.4\end{aligned}$$

$$\begin{aligned}
 w_{25} &= 0.1 + (0.5)(-0.0063)(0) = 0.1 \\
 w_{34} &= -0.5 + (0.5)(-0.0087)(1) = -0.504 \\
 w_{35} &= 0.2 + (0.5)(-0.0063)(1) = 0.196 \\
 w_{56} &= -0.2 + (0.5)(0.1311)(0.524) = -0.165 \\
 w_{46} &= -0.3 + (0.5)(0.1311)(0.332) = -0.278
 \end{aligned}$$

Ajustamos bias

$$\begin{aligned}
 \theta_6 &= 0.1 + 0.5(0.1311) = 0.165 \\
 \theta_5 &= 0.2 + 0.5(-0.0063) = 0.196 \\
 \theta_4 &= -0.4 + 0.5(-0.0087) = -0.404
 \end{aligned}$$

Conclusiones

Aprender la técnica de feed – forward es un gran inicio para entender como funciona una red neuronal y como esta se ajusta (se entrena) para tener la respuesta que uno estaba solicitando. El uso de esta técnica para entrenar una red neuronal es una muestra del aprendizaje supervisado y de lo que es capaz de hacer y a su vez muestra una gran cantidad de aplicaciones donde podemos implementar esta tecnica.

Referencias Bibliográficas

Redacción. (2023, 5 junio). *Arquitecturas de redes neuronales: Feedforward y Recurrente -*

Inteligencia Artificial. Inteligencia Artificial.

<https://inteligenciaartificial.science/arquitecturas-de-redes-neuronales-feedforward-y-recurrente/>