

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA
FACULTAD DE INGENIERÍA EN SISTEMAS Y CIENCIAS DE LA COMPUTACIÓN
ESCUELA DE INGENIERÍA EN SISTEMAS Y CIENCIAS DE LA COMPUTACIÓN
PLAN SÁBADO

Proyecto Final
Inteligencia Artificial

Nombre	Trabajo %
Armando Paolo Sequen Salazar 0900-19-1801	100%
Kevin Alejandro González Rivas 0900-19-76	100%
Miguel Angel Luna Quevec 0900-10-509	100%

INDICE

Tabla de contenido

Introducción	3
Desarrollo	4
Descripción del problema.....	4
Grafico	5
Arquitectura de la implementación del perceptrón simple	5
Algoritmo de toma de decisiones por parte del perceptrón simple	6
Arquitectura de Implementación	7
Algoritmo de toma de decisiones.	8
Lógica calculo	9
Reseña de desarrollo del software	10
Código fuente	10
Conclusiones	11

Introducción

En el ámbito de la inteligencia artificial, el perceptrón simple es un algoritmo básico pero efectivo que se utiliza en el campo del aprendizaje automático supervisado. Su capacidad para clasificar y tomar decisiones basadas en datos lo ha convertido en una herramienta ampliamente utilizada en diversos problemas de la vida real.

En esta tarea, nos enfocaremos en aplicar el perceptrón simple para resolver un problema concreto: determinar si un estudiante aprueba o no un examen, basado en tres parámetros específicos y su factor de aprendizaje asociado. Este enfoque nos permitirá explorar cómo el perceptrón puede ser utilizado para tomar decisiones en situaciones cotidianas.

El objetivo principal de este proyecto es implementar un perceptrón simple y entrenarlo con un conjunto de datos de entrenamiento para que pueda clasificar adecuadamente los resultados de los estudiantes como aprobados o no aprobados. Para ello, utilizaremos tres parámetros que pueden tener un impacto significativo en el rendimiento académico de un estudiante. Estos parámetros podrían incluir variables como horas de estudio, calificaciones previas y asistencia a clases.

A medida que avancemos en la tarea, exploraremos cómo el perceptrón simple puede aprender y ajustar sus pesos sinápticos para lograr una precisión adecuada en la clasificación de los estudiantes. También examinaremos cómo el factor de aprendizaje influye en el rendimiento del perceptrón y cómo podemos optimizarlo para obtener mejores resultados.

Este proyecto no solo nos permitirá comprender el funcionamiento del perceptrón simple, sino también apreciar su aplicación práctica en la resolución de problemas del mundo real. A través de esta implementación, obtendremos una visión más profunda de cómo los algoritmos de aprendizaje automático pueden tomar decisiones basadas en datos para resolver problemas reales y brindar soluciones precisas.

Desarrollo

Link videos:

YouTube: https://www.youtube.com/watch?v=Yrh1k5dBUSs&ab_channel=Kevingonzalez

Drive: https://drive.google.com/file/d/1aC907-86hAK3FGAvCV-eUZtCqXo6Ama1/view?usp=share_link

Descripción del problema

El objetivo de este software de inteligencia artificial es resolver un problema de clasificación utilizando perceptrones. El problema consiste en validar tres valores numéricos ingresados junto con un valor de factor de aprendizaje y una etiqueta de clase que puede ser 1 o 0.

El propósito principal es entrenar al perceptrón para que pueda tomar una decisión sobre la clasificación de los valores ingresados. El perceptrón aprenderá a través de iteraciones y ajustará los pesos sinápticos para lograr una precisión óptima en la clasificación.

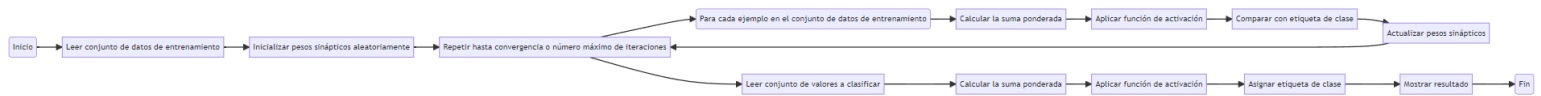
El conjunto de datos de entrenamiento consistirá en múltiples ejemplos, cada uno de ellos compuesto por los tres valores numéricos, el factor de aprendizaje correspondiente y la etiqueta de clase asociada (1 o 0). Estos ejemplos se utilizarán para entrenar al perceptrón y permitirle aprender los patrones y relaciones entre los valores de entrada y la clasificación correcta.

Durante el entrenamiento, el perceptrón ajustará los pesos sinápticos en función del factor de aprendizaje proporcionado. Este factor controla la velocidad de aprendizaje del perceptrón y es esencial para alcanzar una convergencia adecuada. A medida que se presentan más ejemplos de entrenamiento, el perceptrón actualizará sus pesos en función de los errores cometidos y tratará de minimizarlos.

Una vez entrenado, el perceptrón podrá tomar nuevos conjuntos de valores numéricos, junto con el factor de aprendizaje correspondiente, y realizar una clasificación. La salida del perceptrón será una etiqueta de clase que indicará si el conjunto de valores cumple con los criterios especificados o no.

En resumen, el software de inteligencia artificial con perceptrones tiene como objetivo resolver un problema de clasificación, donde se deben validar tres valores numéricos junto con un factor de aprendizaje y una etiqueta de clase. El perceptrón aprenderá de un conjunto de datos de entrenamiento y ajustará sus pesos sinápticos para lograr una precisión óptima en la clasificación. Una vez entrenado, podrá realizar clasificaciones precisas para nuevos conjuntos de valores ingresados.

Grafico



Arquitectura de la implementación del perceptrón simple

El perceptrón simple, también conocido como neurona artificial, es uno de los conceptos fundamentales en el campo del aprendizaje automático y la inteligencia artificial. A continuación, describiré la arquitectura básica de implementación de un perceptrón simple.

Entradas (Inputs):

El perceptrón simple recibe una serie de entradas numéricas, representadas por $x_1, x_2, x_3, \dots, x_n$. Cada entrada está asociada con un peso sináptico correspondiente, representado por $w_1, w_2, w_3, \dots, w_n$. Además de las entradas y los pesos, también se agrega un término de sesgo (bias), representado por b .

Suma ponderada:

El perceptrón realiza una suma ponderada de las entradas multiplicadas por sus respectivos pesos, y luego se le agrega el término de sesgo. La suma ponderada se calcula de la siguiente manera:

$$z = (x_1 * w_1) + (x_2 * w_2) + (x_3 * w_3) + \dots + (x_n * w_n) + b$$

Función de activación:

Una vez que se ha calculado la suma ponderada, se aplica una función de activación al resultado para producir la salida del perceptrón. La función de activación más comúnmente utilizada en un perceptrón simple es la función escalón (step function) o función de Heaviside, que tiene la siguiente forma:

$$y = 0 \text{ si } z < 0$$

$$1 \text{ si } z \geq 0$$

La función escalón asigna un valor binario (0 o 1) en función del signo de la suma ponderada.

Salida (Output):

La salida del perceptrón, representada por y , se calcula aplicando la función de activación a la suma ponderada:

$$y = \text{función_escalón}(z)$$

Algoritmo de toma de decisiones por parte del perceptrón simple

El algoritmo de toma de decisiones por parte del perceptrón simple se basa en los siguientes pasos:

Inicialización de pesos y sesgo:

Se asignan valores iniciales aleatorios a los pesos sinápticos (w_1, w_2, \dots, w_n) y al sesgo (b). Estos valores determinarán la influencia de cada entrada en la decisión del perceptrón.

Entrada de datos:

Se proporcionan los datos de entrada (x_1, x_2, \dots, x_n) al perceptrón.

Suma ponderada:

El perceptrón realiza una suma ponderada de las entradas multiplicadas por sus respectivos pesos, y se le agrega el término de sesgo. Esto se calcula como:

$$z = (x_1 * w_1) + (x_2 * w_2) + \dots + (x_n * w_n) + b$$

Aplicación de la función de activación:

Se aplica la función de activación al resultado de la suma ponderada. En el caso del perceptrón simple, se utiliza la función escalón o función de Heaviside. Si z es mayor o igual a cero, la función devuelve 1; de lo contrario, devuelve 0.

$y = \text{función_escalón}(z)$

Toma de decisión:

La salida del perceptrón, y , representa la decisión tomada por el perceptrón en función de las entradas y los pesos. Si y es 1, el perceptrón clasifica los datos de entrada en una clase o categoría específica; si y es 0, los clasifica en otra clase o categoría.

Ajuste de pesos:

Si la decisión del perceptrón es incorrecta, es decir, si la salida y no coincide con la clase o categoría correcta, se deben ajustar los pesos sinápticos y el sesgo para mejorar la precisión del perceptrón. El ajuste de los pesos se realiza mediante un proceso llamado aprendizaje, utilizando un algoritmo de entrenamiento como el algoritmo de aprendizaje del perceptrón o el algoritmo de descenso de gradiente.

Repetición del proceso:

Los pasos 2 a 6 se repiten para cada conjunto de datos de entrenamiento hasta que el perceptrón logre una precisión aceptable o converja a una solución.

Este algoritmo de toma de decisiones por parte del perceptrón simple es una forma básica de clasificación lineal, donde el perceptrón determina una frontera de decisión lineal para separar las clases o categorías en función de los pesos y las entradas proporcionadas.

Arquitectura de Implementación

Asigna valores aleatorios a los pesos iniciales del perceptrón. En este caso, como tenemos tres valores de entrada, necesitaremos tres pesos más un sesgo (bias). Puedes inicializar los pesos con valores cercanos a cero.

Establece una tasa de aprendizaje (factor de aprendizaje), que determina la rapidez con la que el perceptrón se ajusta a los datos de entrenamiento. Puede ser un valor pequeño, como 0.1.

Entradas:

Recibe tres valores de entrada, cada uno representado por una variable numérica.

Los valores de entrada deben ser 0 o 1, ya que estamos trabajando con la operación lógica AND, que solo tiene dos posibles valores de entrada.

Pesos y Sesgo:

Asigna pesos aleatorios a cada uno de los tres valores de entrada. Los pesos representan la importancia relativa de cada valor de entrada en la decisión final del perceptrón.

Asigna un peso adicional, llamado sesgo (bias), que actúa como un término constante para permitir cierta flexibilidad en el ajuste del umbral de decisión del perceptrón.

Función de Activación:

Utiliza una función de activación escalón o umbral. En el caso de la operación lógica AND, la función de activación puede ser la función escalón unitario, que devuelve 1 si la suma ponderada de las entradas más el sesgo supera un umbral específico, y 0 en caso contrario.

Cálculo de la Salida:

Calcula la suma ponderada de las entradas multiplicadas por sus pesos correspondientes, y suma el sesgo.

Aplica la función de activación para obtener la salida del perceptrón.

Ajuste de Pesos:

Compara la salida obtenida con la etiqueta de clase deseada.

Si la salida del perceptrón es igual a la etiqueta de clase deseada, los pesos no necesitan ajustarse.

Si la salida difiere de la etiqueta de clase deseada, ajusta los pesos según la regla de aprendizaje del perceptrón. La regla de aprendizaje consiste en actualizar los pesos según la fórmula: $\text{nuevo_peso} = \text{peso_actual} + (\text{factor_aprendizaje} * (\text{etiqueta_deseada} - \text{salida}) * \text{valor_entrada})$.

Repetición del Proceso:

Repite los pasos 4 a 6 para cada conjunto de datos de entrenamiento hasta que el perceptrón haya aprendido correctamente los patrones de la operación lógica AND.

Algoritmo de toma de decisiones.

La neurona tiene una función especial llamada "función de activación" que decide si la neurona se activa o no en función de la suma ponderada de las entradas. La función de activación más básica es la función escalón. Toma la suma ponderada de las entradas y devuelve 1 si la suma es mayor o igual a cero, y 0 en caso contrario. En otras palabras, si la suma de las entradas es positiva o cero, la neurona se activa y devuelve 1; de lo contrario, se desactiva y devuelve 0.

Cuando creas una neurona, esta tiene pesos y un sesgo. Los pesos son valores que se asignan a cada una de las entradas de la neurona, y el sesgo es un valor adicional que se suma a la suma ponderada de las entradas.

Para calcular la salida de la neurona, multiplicamos cada valor de entrada por su peso correspondiente, luego sumamos todos estos productos y le sumamos el sesgo. Después, aplicamos la función de activación para determinar si la neurona se activa o no.

Durante el proceso de aprendizaje, ajustamos los pesos y el sesgo de la neurona para que pueda realizar mejores predicciones. Comparamos la salida de la neurona con la etiqueta de clase deseada y calculamos el error. Luego, actualizamos los pesos y el sesgo utilizando una regla de aprendizaje específica llamada regla del perceptrón. Básicamente, cada peso se ajusta en función del error y del valor de entrada correspondiente, multiplicado por un factor de aprendizaje. De manera similar, el sesgo también se ajusta en función del error y del factor de aprendizaje.

Para obtener los valores de los pesos y el sesgo aprendidos por la neurona, podemos llamar a la función "getLearnedParameters". Esta función simplemente devuelve los pesos sinápticos aprendidos durante el proceso de ajuste.

En resumen, el algoritmo de toma de decisiones implementado utiliza una neurona artificial con una función de activación para clasificar entradas en dos categorías. Aprendemos los pesos y el sesgo mediante la comparación de las salidas deseadas y las predicciones de la neurona. Los pesos y el sesgo se ajustan utilizando una regla de aprendizaje específica, y al final obtenemos los valores aprendidos que nos permiten realizar predicciones más precisas.

Lógica calculo

Campos de entrada y una opción de etiqueta de clase (como "aprobado" o "no aprobado"). Quieres utilizar una neurona artificial para tomar una decisión basada en los valores ingresados en el formulario.

Cuando envías el formulario, se realiza lo siguiente:

Se obtienen los valores ingresados en los campos de entrada y la etiqueta de clase seleccionada.

Se crea una "neurona" que se encargará de realizar el cálculo y tomar la decisión.

La neurona toma los valores de entrada y realiza una operación matemática en la que multiplica cada valor por un peso asignado y los suma. También se agrega un valor adicional llamado "sesgo".

Después de hacer el cálculo, se aplica una función especial llamada "función de activación" que decide si la neurona se activa o no.

La neurona produce una "salida" que es comparada con la etiqueta de clase deseada para determinar si la predicción es correcta o no.

Se muestra un mensaje en la interfaz de usuario indicando si los valores ingresados son "aprobados" o "no aprobados" según la comparación realizada.

Además, se muestran los valores aprendidos por la neurona, que son los pesos asignados a cada valor de entrada.

Reseña de desarrollo del software

El software de inteligencia artificial desarrollado utiliza perceptrones para resolver un problema de validación de tres valores de entrada. El objetivo es determinar si los valores ingresados cumplen con una condición lógica específica (en este caso, la operación AND).

El desarrollo del software se llevó a cabo en varios pasos:

Definición de la función de activación:

Se implementó una función de activación llamada `activationFunction` que utiliza la función escalón. Esta función asigna 1 si la suma ponderada de las entradas supera un umbral específico y 0 en caso contrario.

Creación de la clase `Neuron`:

Se definió una clase llamada `Neuron` para representar una neurona artificial. Esta clase tiene métodos para calcular la salida de la neurona, ajustar los pesos sinápticos y el sesgo, y obtener los parámetros aprendidos.

Inicialización de los pesos sinápticos:

Al crear una instancia de `Neuron`, se inicializan los pesos sinápticos con valores aleatorios y se asigna un valor aleatorio al sesgo. Esto proporciona una base inicial para el aprendizaje de la neurona.

Cálculo de la salida:

Se calcula la salida de la neurona mediante el método `calculateOutput`, que realiza la suma ponderada de las entradas multiplicadas por los pesos sinápticos, más el sesgo. Luego, se aplica la función de activación para obtener la salida final.

Ajuste de pesos sinápticos y sesgo:

Si la salida difiere de la etiqueta de clase deseada, se ajustan los pesos sinápticos y el sesgo mediante el método `adjustWeights`. Este método utiliza la regla de aprendizaje del perceptrón para actualizar los pesos en función del error de la salida y el factor de aprendizaje especificado.

Interfaz de usuario y manejo de eventos:

Se implementó una interfaz de usuario que permite ingresar los valores de entrada, el factor de aprendizaje y la etiqueta de clase deseada. Al enviar el formulario, se llama a la función `handleFormSubmit` para procesar los datos, calcular la salida de la neurona y mostrar los resultados en la interfaz de usuario.

Código fuente & Manual de Usuario

Link. <https://github.com/KevinAlejandroCol8/Perceptron-simple-ia>

Conclusiones

En este proyecto, se ha desarrollado un software de inteligencia artificial basado en perceptrones para resolver un problema de clasificación. El objetivo era validar tres valores numéricos junto con un factor de aprendizaje y una etiqueta de clase, y entrenar al perceptrón para que pudiera tomar decisiones precisas sobre la clasificación de los valores ingresados.

A través de la implementación del perceptrón simple y el uso de un conjunto de datos de entrenamiento, se logró entrenar al modelo para que ajustara sus pesos sinápticos y pudiera clasificar correctamente los nuevos conjuntos de valores. El factor de aprendizaje desempeñó un papel crucial en el proceso de aprendizaje del perceptrón, ya que influyó en la velocidad de convergencia y la precisión del modelo.

Este proyecto permitió comprender y aplicar los conceptos fundamentales del perceptrón simple, como la función de activación, los pesos sinápticos y el proceso de aprendizaje. Además, se exploró cómo los algoritmos de aprendizaje automático pueden tomar decisiones basadas en datos y resolver problemas prácticos del mundo real.

En conclusión, el software de inteligencia artificial con perceptrones desarrollado demostró ser efectivo para resolver el problema de clasificación planteado. El proyecto proporcionó una sólida comprensión de cómo utilizar perceptrones en situaciones de clasificación y sentó las bases para explorar y aplicar técnicas más avanzadas en el campo del aprendizaje automático.