Introducción:

En esta práctica vamos a ver cómo una computadora puede aprender a reconocer números escritos a mano utilizando un método llamado AdaBoost. Este es un desafío común en el campo de la inteligencia artificial, donde queremos que las máquinas interpreten y entiendan los datos como lo haría un ser humano.

Este proyecto es un ejemplo de cómo los métodos de inteligencia artificial pueden aprender tareas complejas a partir de datos simples. La capacidad de una máquina para reconocer dígitos manuscritos tiene muchas aplicaciones, como leer cheques en un banco o interpretar escritura a mano en tablets y smartphones.

Conceptos Básicos:

* Aprendizaje Automático: Es como enseñar a una computadora a reconocer patrones y tomar decisiones basadas en ejemplos, en lugar de seguir instrucciones específicas.
* Clasificación: Es el proceso de decidir a qué categoría pertenece una nueva imagen basándose en ejemplos que ya conoce.
* AdaBoost: Es una técnica que combina muchas reglas simples de decisión para formar una regla más compleja y precisa.

AdaBoost y Decision Stump: Entendiendo el Proceso:

Imagina que tienes un grupo de amigos que intentan decidir si una imagen muestra el número 9 o no. Cada amigo tiene una regla simple para decidir, como "si la parte superior del número es redonda, entonces podría ser un 9". AdaBoost toma todas estas reglas simples y las combina para hacer una decisión grupal más informada.

El Conjunto de Datos MNIST:

Para entrenar nuestro sistema, utilizamos un conjunto de miles de imágenes de números escritos a mano llamado MNIST.

Implementación: Explica paso a paso cómo cada amigo (clasificador débil) hace su juicio y cómo estos juicios se combinan para llegar a una decisión final.

Entrenamiento del Clasificador AdaBoost:

El proceso de entrenamiento es como una serie de rondas de práctica donde cada amigo ajusta su regla para ser mejor en la predicción. Con cada ronda, el grupo como un todo se vuelve más preciso en identificar el número 9.

Evaluación y Resultados:

Al final del entrenamiento, probamos a nuestros amigos con imágenes nuevas para ver cuántas pueden clasificar correctamente. Esta tasa de acierto nos dice qué tan bien ha aprendido el grupo.

Conclusión:

En nuestro proyecto, encontramos que después de 20 rondas de práctica, nuestros amigos pudieron identificar correctamente si una imagen era el número 9 con una precisión mayor al 90%. Esto muestra la promesa de AdaBoost como una herramienta para enseñar a las computadoras a reconocer imágenes.

Importaciones:

* numpy es una biblioteca que se utiliza para operaciones matemáticas y manipulación de arrays.
* keras.datasets contiene el conjunto de datos MNIST, que vamos a utilizar para entrenar y probar nuestro clasificador.
* accuracy\_score de sklearn.metrics se utiliza para calcular qué porcentaje de predicciones son correctas.
* time se utiliza para medir cuánto tiempo toma entrenar el clasificador.

DecisiónStump()

Constructor de un clasificador muy simple.

Selecciona una característica de manera aleatoria y establece un umbral y una polaridad (dirección) también de manera aleatoria.

Constructor Predict()

Hace una predicción basada en la característica seleccionada y el umbral. Si la polaridad es 1, predice -1 para todos los valores de la característica que están por debajo del umbral y 1 para los que están por encima. Si la polaridad es -1, es al revés.

**Adaboost**

Adaboost es un algoritmo de "ensemble learning" o aprendizaje ensamblado, que combina múltiples modelos simples para crear un modelo final más potente y preciso. Imagina que en lugar de confiar en el juicio de una sola persona para tomar una decisión, consultas a un grupo de personas y tomas la decisión basada en la mayoría de sus opiniones. Cada persona en este grupo podría no ser experta, pero juntas, sus opiniones combinadas son más confiables que la de cualquier individuo por sí solo.

Concepto de Clasificador Débil

Adaboost comienza con lo que se llama clasificadores débiles. Un clasificador débil es un modelo que hace predicciones que son apenas mejores que adivinar al azar. En el contexto del reconocimiento de dígitos, un clasificador débil podría ser una regla muy simple, como "si la parte superior del número es redonda, entonces podría ser un 9".

Entrenamiento de Adaboost

El entrenamiento de Adaboost consiste en construir una serie de estos clasificadores débiles, cada uno enfocado en los errores del anterior. Cada vez que Adaboost agrega un nuevo clasificador débil, aumenta la importancia (o el peso) de los ejemplos que fueron mal clasificados previamente. Esto significa que el siguiente clasificador débil se enfoca más en los ejemplos difíciles, intentando corregir los errores pasados.

Pesos y Votos

Cada clasificador débil tiene un voto para decidir la clasificación final de un ejemplo, y este voto es ponderado por su precisión. Los clasificadores que lo hacen mejor (es decir, tienen menos errores) tienen más peso en la decisión final.

Proceso Iterativo

Adaboost repite este proceso un número predefinido de veces, agregando clasificadores débiles y ajustando sus votos. Con cada iteración, el modelo se vuelve más refinado y mejor en la predicción.

Predicción Final

Para hacer una predicción, Adaboost suma los votos ponderados de todos sus clasificadores débiles. Si la suma es positiva, el modelo predice una clase (por ejemplo, "esto es un 9"); si es negativa, predice la otra clase ("esto no es un 9").