Bitácora de Desarrollo - Implementación del Algoritmo Adam

Nombre: Abel Albuez Sanchez

Materia: Aprendizaje de Máquina

Profesor: Leonardo Florez Valencia

Repositorio del Proyecto: GitHub - AbelAlbuez/machine-learning-assignments

Paso 1 - Comprensión del problema y exploración del dataset

Para abordar el problema, primero revisé el dataset disponible en Kaggle: Red Wine Quality Dataset. Este dataset contiene datos fisicoquímicos del vino y una etiqueta de calidad.

- **Objetivo:** Implementar un modelo de clasificación binaria para determinar si un vino tiene buena calidad (1) o no (0).
- **Estrategia:** Convertir la variable de calidad en un formato binario donde valores mayores a 6.5 sean considerados "buenos" (1) y los demás "regulares" (0).

Paso 2 - Comprender el Algoritmo Adam y otros optimizadores

Antes de la implementación, analicé las diferencias entre los métodos de optimización:

- **SGD (Stochastic Gradient Descent):** Realiza actualizaciones basadas en gradientes simples.
- **Momentum:** Suaviza el gradiente utilizando la información previa.
- **RMSProp:** Ajusta la tasa de aprendizaje para cada peso según la magnitud del gradiente.
- **Adam:** Combina Momentum y RMSProp, ajustando la tasa de aprendizaje y acumulando momentos del gradiente.

Decidí comparar **Adam vs. SGD** en el entrenamiento para visualizar las diferencias.

Paso 3 - Definición del modelo y parámetros iniciales

1. **Seleccionar las características:** Eliminé la columna de calidad original y normalicé las variables.
2. **Inicialización de pesos:** Se asignaron pesos aleatorios pequeños cercanos a 0.
3. **Uso de la función sigmoide:** Para convertir las predicciones en valores entre 0 y 1.

Paso 4 - Entrenamiento del modelo

- Implementé la función de entrenamiento donde Adam y SGD ajustan los pesos en cada iteración.
- Evalué la pérdida y la precisión en cada época.

Paso 5 - Comparación con y sin regularización

- Primero entrené el modelo **sin regularización**.
- Luego probé con **L1 (Lasso)** y **L2 (Ridge)** para reducir el sobreajuste.

Paso 6 - Visualización del ajuste de pesos

- Comparé cómo Adam y SGD actualizan los pesos en cada época.
- Grafiqué la pérdida y la precisión para observar la convergencia.

Paso 7 - Comprender el uso de Binary Cross-Entropy

Usé la **Entropía Cruzada Binaria** como función de pérdida, ya que:

- Penaliza más los errores en clasificación binaria.
- Su formulación mejora el ajuste del modelo al reducir la incertidumbre.

La fórmula utilizada fue: $L = -\frac{1}{n}\sum[y\log(\hat{y}) + (1-y)\log(1-\hat{y})]$ Donde:

- y es la etiqueta real.
- $\hat{y}$ es la predicción del modelo.

Conclusiones y Aprendizajes

- Adam convergió más rápido y con menos oscilaciones que SGD.
- La regularización mejoró la generalización, reduciendo el sobreajuste.
- Binary Cross-Entropy ayudó a mejorar la interpretabilidad de la pérdida.

📌 **Próximos pasos:**

- Optimizar la tasa de aprendizaje para mejorar la convergencia.
- Implementar un enfoque multi-clase para clasificar en más de dos categorías de calidad del vino.