Apuntes Martes 3/18/2025

Luis Carlos N. Todd c.2022212158

***Abstract*—En este documento se exponen las respuestas del quiz 2 y se detalla el proceso de actualización de parámetros mediante descenso del gradiente, incluyendo sus variantes: Batch Gradient Descent, Stochastic Gradient Descent y MiniBatch Gradient Descent. Se discuten estrategias para evitar el overfitting, como la técnica de Early Stopping y la correcta partición del dataset en training, validation y testing. Finalmente, se analiza el compromiso entre sesgo y varianza (Bias-Variance Tradeoff), destacando su impacto en la capacidad del modelo.**

***Index Terms*—función de pérdida, descenso del gradiente, overfitting, underfitting, sesgo, varianza, partición del dataset, Early Stopping, Bias-Variance Tradeoff.**

I. Respuestas del Quiz

A. Ejercicio 1: Diferencia de mínimo local y mínimo global

Mínimo local: Punto más bajo en un intervalo especifico de la function.

Mínimo global: Punto más bajo de toda la function

B. Ejercicio 2: Describa la diferencia entre un problema de clasificacion y uno de regresión

Clasificacion: Predecir una etiqueta especifica (set de imágenes limitado).

Regresion: Predecir un target o valor especifico (set infinito, función)

C. Ejercicio 3: Describa paso a paso el algoritmo de KNN

Tengo un sample nuevo.

Calculo las distancias L2 con el dataset que ya tengo.

Tomo los K elementos con distancias más cortas del set de entrenamiento.

Clasifico según las etiquetas de esos k.

D. Ejercicio 4

$$F(x,y) = x^3 + 2xy + y^2 + 5x + 3y + 7 \tag{1}$$

$$\frac{\partial F}{\partial x} = (x^3)' + (2xy)' + (y^2)' + (5x)' + (3y)' + (7)' \tag{2}$$

$$\frac{\partial F}{\partial x} = 3x^2 + 2y + 5 \tag{3}$$

II. Repaso función de pérdida, actualización de parámetros y descenso del gradiente

Calculando derivadas de L con respecto a w.

$$\frac{\partial L}{\partial w} = \frac{1}{N}\sum(((wx_i + b) - y_i)^2)' \tag{4}$$

$$... \tag{5}$$

$$\frac{\partial L}{\partial w} = \frac{1}{N}\sum 2(\bar{y} - y_i) \tag{6}$$

Calculando derivadas de L con respecto a b.

$$\frac{\partial L}{\partial b} = \frac{1}{N}\sum(((wx + b) - y)^2)' \tag{7}$$

$$... \tag{8}$$

$$\frac{\partial L}{\partial w} = \frac{1}{N}\sum 2(\bar{y} - y_i)x_i \tag{9}$$

A. Actualización de parámetros

$$w := w - \alpha\frac{\partial L}{\partial w} \tag{10}$$

$$b := b - \alpha\frac{\partial L}{\partial b} \tag{11}$$

Small alpha: Va lento pero es estable

Large alpha: Es más inestable y puede no converger

Fix alpha: Ir ajustando alpha (Haciendolo menor conforme se va avanzando)

- Para progras: Empezar con alpha = 0.001 hasta 0.009

Epoch: Iteraciones sobre todos los datos de entrenamiento.

Batch: Tomar un subconjunto del total de datos de entrenamiento para calcular el gradiente.

B. Tres algoritmos para aplicar el decenso del gradiente.

Batch Gradient Descent: Vanilla, calcula el error por cada sample, funciona bien pero es muy lento. Requiere el dataset en memoria. Tarda en converger. Es estable.

Stochastic gradient Descent: Modifica los parámetros para cada sample del training set. Es computacionalmente complejo hacer actuailización de parámetros por cada sample. Señales de gradiente ruidosas.

MiniBatch Gradient Descent: Combina ambas estrategias, utiliza un batch específico dependiendo de la aplicación, Evita mínimos locales y agrega robustez. Computacionalmente eficiente en comparación a SGD. Require un hiperparámetro más para el entrenamiento.

- *Ver Notebook de RegresiónLineal.ipynb*

Técnica Stop Loss o Early Stop. Detener entrenamiento cuando en varias epocas no mejora el algoritmo. Ahorrar epochs.

Sesgo y Varianza

Dataset Cortamos el dataset en Training set y Testing set. 80% y 20% respectivamente.

Training Set: Utilizado para ajustar el modelo. Ajusta los parámetros de acuerdo a las muestras disponibles. El modelo identifica patrones basado en estos datos. Debería representar la diversidad de escenarios que se espera encontrar. Permitirá al modelo entrenado predecir datos nunca vistos

antes. Encuentra patrones entre las entradas y salidas. Por lo tanto, establece las relaciones entre las varialbes y los pesos/parámetros del modelo. Suficientemente grande para que sea significativo pero que no cause **Overfitting***.

Overfitting*: Sobreajuste a los datos de entrenamiento que tengo. Aprendo muy bien los datos de entrenamiento y no sabe generalizar.

Testing set Utilizado para evaluar el modelo con ejemplos que no se utilizaron en el entrenamiento. Debe ser independiente del set de entrenamiento. Se calculan métricas para estimar el rendimiento del modelo. Los valores de Loss de training es insignificante. Ejemplos Accuracy, Loss, etc

Objetivos: Medir el modelo de forma realista.

Caso overfitting: Inicio entrenamiento de modelo y capturo métricas del training set. Buen accuracy. Función de loss disminuyendo. Tardo 3 días entrenando mi modelo. Ejecuto el testing set: Overfitting!!

Utilizar otro split más al dataset: Normalmente: 80 training y 20 testing. El nuevo split: Training y Validation 80 (70 y 10) y 20 testing. Uso el set de validation durante el training para asegurarse que el training no está haciendo overfitting.

Validation set: Conjunto de datos que me sirven para valorar la capacidad de generalización de mi modelo a datos nunca vistos. Escencial para el ajuste de parámetros.

Técnicas para subdividir el dataset:

Random Sampling: Aleatorio datos que están en el training set y random en testing set. Útil solo para datos con clases balanceadas. No agrega ningún sesgo al momento de hacer la división. Datos imbalanceados pueden producir validation o testing set con menos datos o ningunos de las clases menos representadas.

Stratified sampling: Usado para datos imbalanceados. Asegura una representación de toas las clases en cada split. Mantiene la misma distribución de datos para cada clase en cada subconjunto. Modelo más robusto.

K-Fold Cross-Validation: Se divide el subconjunto en K partes. Se entrena el modelo con K-1 partes. Se reserva uno para validación. Se continua este proceso rotando los subconjuntos usados para el entrenamiento y validación. Toma el promedio de datos del rendimiento del modelo.

Posibles escenarios:

- Bajo error en training, bajo error en testing (Escenario ideal). Mi modelo evita el ruido existente en los datos. Puede generalizar correctamente.
- Bajo error en training, alto error en testing. Overfitting!! No es capaz de generalizar. Alta varianza.
- Alto error en training, alto error en testing. Underfitting!! El modelo no está aprendiendo nada de los datos. Modelo muy simple. Alto sesgo.

Bias-Variance Tradeoff

Necesito un modelo que tenga baja varianza y bajo sesgo. Mientras más complejo es mi modelo el sesgo baja y la varianza sube. Mientras más simple es mi modelo el sesgo sube y la varianza baja.

Alto Bias El modelo comete muchos errores en el training set. Underfitting. El modelo asume mucho del training set. No usa todos los features del modelo. El modelo es simple. Cómo evitar un bias alto? Utilizar un modelo mmás complejo. Features del training set no son adecuados para el problema.

Alta varianza El modelo se ajusta mucho a los datos de entrenamiento. Overfitting. No es capaz de generalizar. Es sensible a las variaciones de los datos. Suele suceer con datos de alta dimensionalidad y pocos ejemplos.

¿Cómo evitar una varianza alta? Intentar con un modelo más simple. Reducir dimensionalidad Obtener más ejemplos para el training set (No siempre es posible). Aplicar técnias de regularización. Hacer el modelo más simple. Apagar neuronas es otra técnica.