Repaso de Derivadas, Regresión Lineal y Sesgo–Varianza en Aprendizaje Supervisado

Ian Murillo Campos
Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Inteligencia Artificial Gr 2

***Abstract*—This paper reviews key elements of supervised learning. It introduces the use of partial derivatives and gradient descent to optimize the mean squared error function. It also examines issues in linear regression such as nonlinearity and outliers, describing statistical methods to address them. Finally, it outlines dataset partitioning into training, validation, and testing sets, and explains the bias–variance tradeoff as a tool to evaluate model generalization.**

I. Introduction

El aprendizaje supervisado entrena modelos predictivos a partir de ejemplos con etiquetas. Las derivadas parciales permiten calcular la influencia de cada parámetro sobre la función de pérdida y se aplican en el descenso de gradiente.

En la regresión lineal, los problemas comunes incluyen la no linealidad de la relación entre variables y la presencia de outliers, que pueden corregirse con técnicas estadísticas.

La división de datos en entrenamiento, validación y prueba permite medir la capacidad de generalización del modelo. Este análisis se relaciona con el sesgo y la varianza, cuyo equilibrio evita tanto el sobreajuste como el subajuste.

II. Repaso de derivadas

A. Función de pérdida (MSE)

$$L = \frac{1}{N}\sum_{i=1}^{N}\big(f_{w,b}(x_i) - y_i\big)^2, \quad i = 1, \ldots, N$$

Se busca optimizar esta función con valores que aumentan L, siendo L una parabola. Dicho de otro modo, se busca encontrar la pendiente de algún punto de la parabola en el que nos ubiquemos y buscamos descender sobre la función para llegar a su punto mínimo.
Se busca calcular cuanto influye el valor w sobre el valor L, calculando sus derivadas parciales.
Esto deja como resultado lo siguiente:

$$\frac{\partial L}{\partial w} = \frac{1}{N}\sum_{i=1}^{N} 2\big((wx_i + b) - y_i\big)\cdot x_i$$

Tambien se tiene que realizar el procedimiento derivando con base en el bias, dando el siguiente resultado:

$$\frac{\partial L}{\partial b} = \frac{1}{N}\sum_{i=1}^{N} 2\big((wx_i + b) - y_i\big)$$

Posterior a esto se aplica el algoritmo del decenso del gradiente, el cual permite optimizar la posición respecto a w y b representado por la siguiente fórmula:

$$w = w - \alpha\frac{\partial L}{\partial w}$$

$$b = b - \alpha\frac{\partial L}{\partial b}$$

B. Vocabulario

- Epoch: Todas las iteraciones que hacemos sobre todos los samples. Es un hiperparametro que mide todo el recorrido de inicio a fin de todo mi set de entrenamiento
- Batch: Tomar ciertos subconjuntos del epoch. Funciona para la optimización de las pruebas.

III. Potenciales problemas al realizar regresión lineal

Entre los potenciales problemas que nos podemos encontrar están la No linealidad de la relación respuesta predictor, los datos sobresalientes y la colinealidad.
De este último no se va a hablar en la clase, queda como tema de investigación personal.

A. No linealidad de la relación respuesta predictor

Uno de los principales supuestos de la regresión lineal es que existe una relación lineal entre las variables predictoras y la variable respuesta.

1) ¿Qué ocurre si la verdadera relación no es lineal?:

- El modelo lineal no podrá captar adecuadamente la relación.
- Se obtendrán errores sistemáticos en los residuos, definidos como:

$$e_i = y_i - \hat{y}_i$$

donde

$$y_i$$

es el valor real y

$$\hat{y}_i$$

es la predicción del modelo.

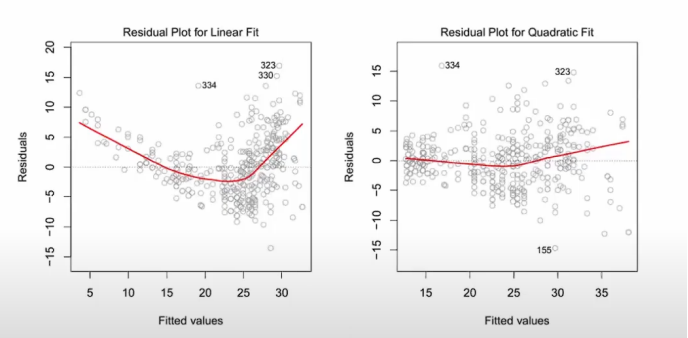


Fig. 1. Ejemplo de gráfica de regresión lineal.

En la figura 1 se ve a la izquierda un plot residual que muestra la fiderencia entre los puntos que tenia que predecir y cuanto se alejan entre si, el modelo de plot residual más correcto es el que esté más cercano al cero.
En la figura de la derecha es el arreglo a los datos, utilizando técnicas para que una función cuadratica como la de la izquierda se comporte más como una función lineal.
La forma de solucinarlo es extender el modelo lineal incorporando transformaciones pólinomicas del predictor, con eso:

- aunque la relación es no lineal en los datos, el modelo sigue siendo lineal en los parámetros.
- Se puede resolver con regresión lineal estándar.

B. Outliers

Son datos que se salen de la distribución que se trata de predecir.
Al tratar de entrenar el modelo de IA, este se va a centrar

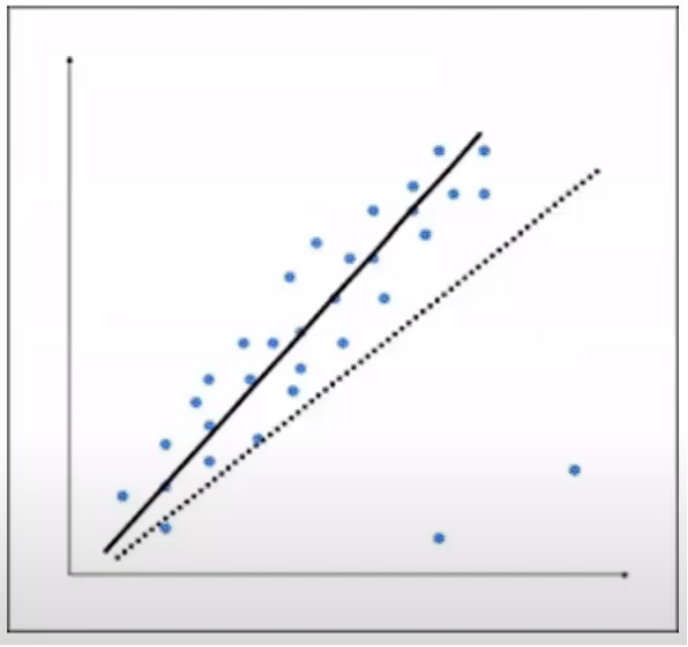


Fig. 2. Ejemplo de Outlier.

tanto en los modelos cercanos a la linea como a los que están más alejados, provocando que el modelo busque hacer un "trade off" entre los datos. Esto no está bien ya que el modelo quedaría sesgado por los outliers y no se utilizarían correctamente los datos que si quiero buscar predecir.
Estos datos en su mayoria son ocasionados por errores de captura, y la forma de corregirlos puede ser eliminarlos directamente del dataset.
Otras técnicas pueden ser:

- Standardized Residuals: Escalar el residuo crufo por una desviación estándar global de los errores.
  Donde:
  - $e_i = y_i - \hat{y}_i$ es el residuo crudo
  - $n =$ número de observaciones
  - $p =$ número de parámetros estimados en el modelo (incluye el i

  La razón de utilizar una desviación estandar es que teniendo todo estandarizado, se puede saber que a ciertas desviaciones estandar de la media se encuentra un porcentaje de los datos. Con esto se puede definir un umbral donde los datos son sobresalientes.
- Regla del rango intercuartílico: Utilizar directamente los datos en lugar del modelo, tomamos el rango intercuantilico que existe entre todos los datos, por definición se ve de la siguiente forma:

$$IQR = Q_3 - Q_1$$

  La regla para detectar los outliers es la siguiente:

$$\left[Q_1 - 1.5 \cdot IQR,\ Q_3 + 1.5 \cdot IQR \right]$$

  - Valores $< Q_1 - 1.5 \cdot IQR \Rightarrow$ outliers inferiores.
  - Valores $> Q_3 + 1.5 \cdot IQR \Rightarrow$ outliers superiores.

  En la figura 3 se puede ver de forma gráfica la regla del rango intercuartílico.
  NOTA: El 1.5 * IQR es aproximadamente equivalente a

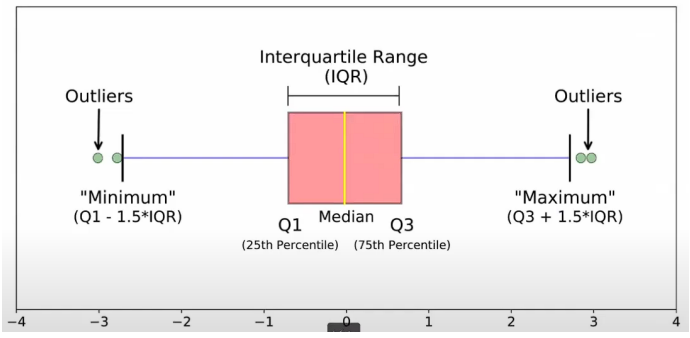


Fig. 3. Ejemplo de Outlier.

  2-2.7 desviaciones estándar de la media (depende de la forma de la distribución).
- Winsorización: Técnica que reemplaza los valores extremos por percentiles límite, en lugar de eliminarlos.
  El procedimiento es el siguiente:
  - Elegir percentiles de corte (Ej. 5% y 95%).

- Valores menores al percentil 5 se reemplazan por el valor del percentil 5.
- Valores mayores al percentil 95 se reemplazan por el valor del percentil 95.

Dentro de sus ventajas se encuentra que:

- Conserva el tamaño de la muestra.
- Reduce la influencia de valores extremos.

IV. Sesgo y Varianza

A. Dataset

Los datos se dividen entre datos de entrenamiento y pruebas, como se ve en la figura 4, los datos de entrenamiento son con los que se optimiza el modelo de IA, mientras que los de pruebas son para verificación, una división de 80% y 20% es lo más común.

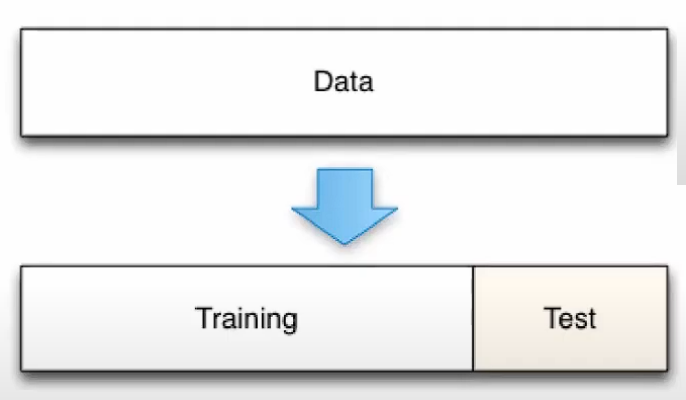


Fig. 4. Ejemplo de Dataset.

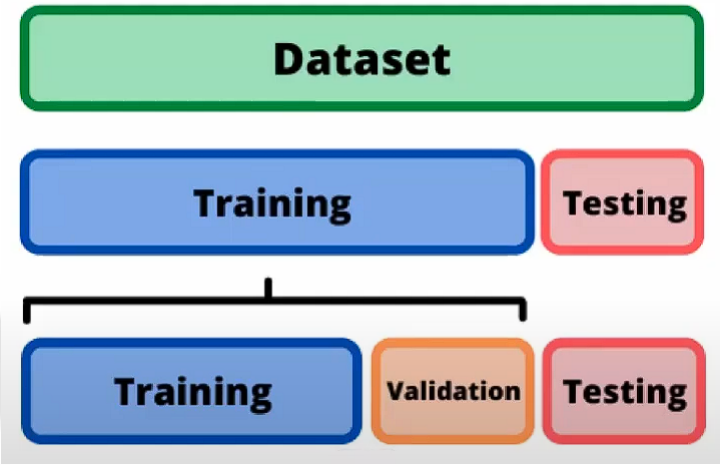
B. Training set

Se utiliza para ajustar el modelo ajustando los parámetros de acuerdo a las muestras disponibles.
El modelo identifica patrones basado en estos datos, ya que estos deberían representar la diversidad de escenarios que se espera encontrar. De esta forma permitirá al modelo entrenado predecir datos nunca vistos antes y encontrar patrones entre las entradas y salidas. Por lo tanto, sirve para establecer relaciones entre las variables y los pesos o parámetros del modelo.
Este debe ser duficientemente grande para que sea significativo, pero sin causar Overfitting. El overfitting ocurre cuando los datos son muy especializados y adaptados al conjunto de entrenamiento por lo que el modelo se vuelve incapaz de generalizar adecuadamente.

C. Testing set

Se utiliza para evaluar el modelo con ejemplos que NO se utilizaron en el entrenamiento.
Debe ser independiente del set de entrenamiento.
Simula la aplicación de un examen a nuestro modelo y con el se calculan métricas como : Accuracy, Loss, Etc...
El objetivo de este set es crear un modelo que generalice adecuadamente todos los escenarios.
Midiendo el rendimiento del modelo realísticamente simulando datos que nunca ha vistohaciendo posible la comparación.

1) Caso Overfitting: Una solución es dividir ptra parte de los datos en datos de validación, que se ejecuten como tests cada cierto tiempo durante la etapa de entrenamiento y que esos datos aseguren que no se da un overfitting.



D. Validation set

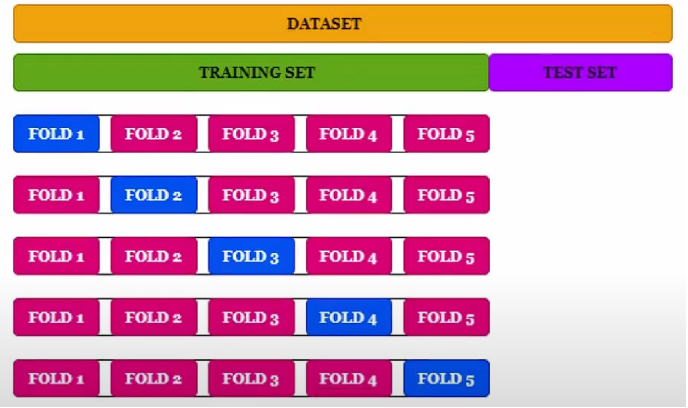
Son un conjunto de datos que sirven para valorar la capacidad de generalizaciónde mi modelo a datos nunca vistos, este set de datos brinda resultados con los que puedo tomar decisiones sobre el proceso de entrenamiento y es un set esencial para el ajuste de hiperparámetros.

E. Técnicas para subdividir el dataset

1) Random sampling: Se divide aleatoriamente el dataset, es útil para datos con clases balanceados ya que no se agrega ningún sesgo al momento de hacer la división.
Los datos imbalanceados pueden producir validation o testing sets con menos datos o ninguno, de las clases menos representadas.

2) Stratified sampling: Se utiliza para datos imbalanceados ya que asegura una representación de todas las clases en cada Split.
Mantiene la misma distribución de datos para cada clase en cada subconjunto, lo que da un modelo más robusto.

3) K-fold Cross-Validation: Se divide el subconjunto en K partes y el modelo se entrena con K-1 partes ya que una se reserva para validación.
Se continua este proceso rotando los subconjuntos usados para el entrenamiento y validación. Permite tomar el promedio del rendimiento del modelo y es útil cuando tenemos pocos datos y deseamos validar nuestro modelo. Se puede ver de forma más gráfica en la siguiente imagen:

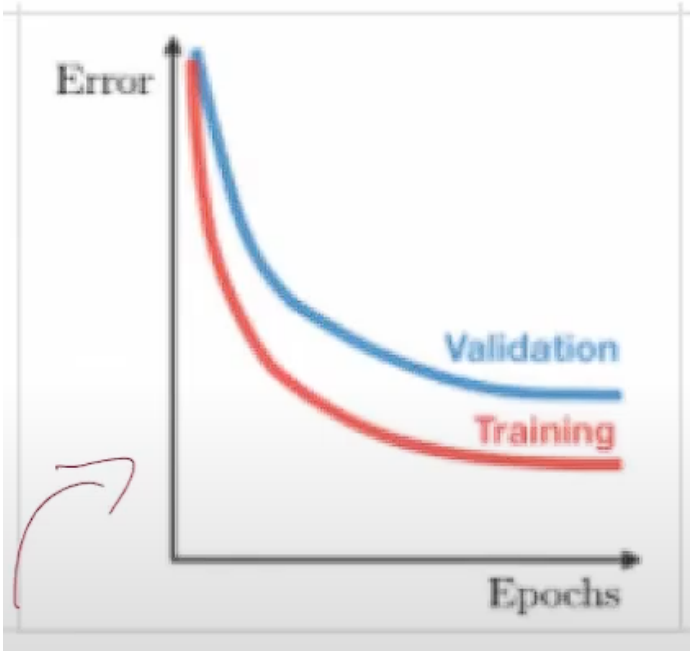


F. Posibles escenarios

Dentro de los posibles escenarios de estos métodos se encueentran:

- Bajo error en training, bajo error en testing.
- Escenario ideal.
- Modelo evita el ruido existente en los datos.
- Puede generalizar correctamente.

Visualmente se puede ver de la siguiente forma:



Otro escenario es cuando se tiene lo siguiente:

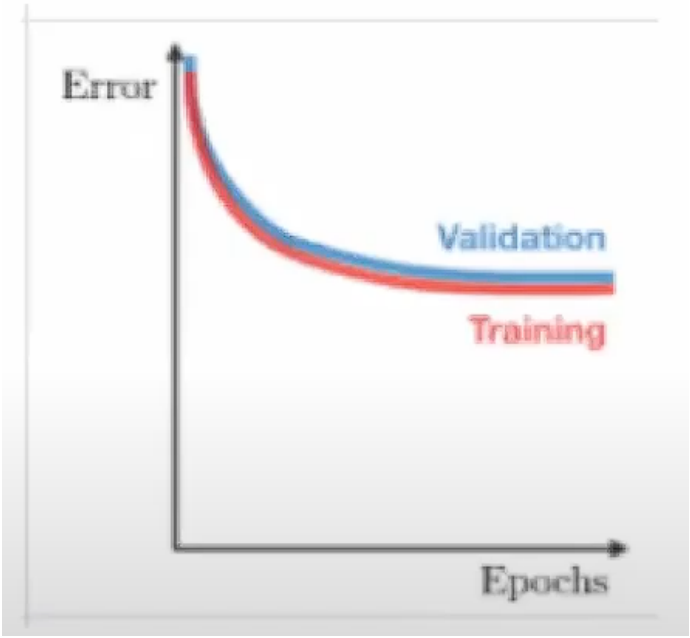
- Bajo error en training, alto error en testing.
- Overfitting.
- No es capaz de generalizar.
- Alta varianza.

Visualmente se ve de la siguiente forma:

Otro escenario es el siguiente:

- Alto error en training, alto error en testing.
- Underfitting.
- El modelo no está aprendiendo nada de los datos.
- Modelo muy simple.
- Alto sesgo

Visualmente se ve de la siguiente forma:

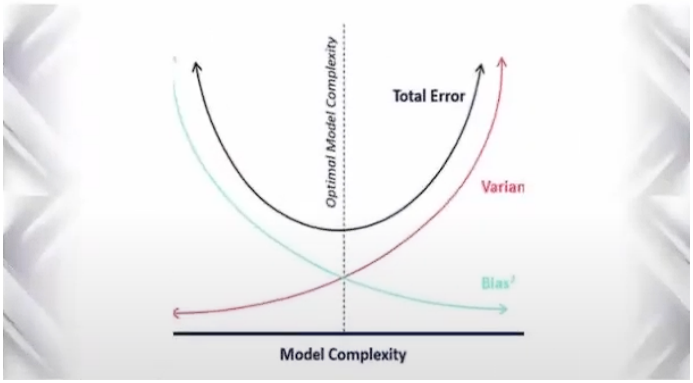


Para solucionar este último caso se utiliza un Bias-Variance tradeoff.

V. Bias-Variance tradeoff

Se busca un modelo que tenga baja varianza y bajo sesgo, para eso se editan valores en las pruebas,

visualmente se ve un arreglo de la siguiente forma:



REFERENCES

[1] S. Pacheco, “Repaso de Matemática: Álgebra Lineal,” Presentación, Instituto Tecnológico de Costa Rica, 2025.
[2] S. Pacheco, “Sesgo y Varianza,” Presentación, Instituto Tecnológico de Costa Rica, 2025.