Unidad 3

By: Mgr. Ruth Chirinos

Índice

PARTE 1:
Bias (Sesgo) y Varianza en Machine Learning

By: Mgr. Ruth Chirinos

Bias (Sesgo) y Varianza en Machine Learning

“En el mundo de Machine Learning, la precisión lo es todo”

- Cuando desarrollamos un modelo nos esforzamos para hacer que sea lo más preciso, ajustando y reajustando los parámetros, pero la realidad es que no se puede construir un modelo 100% preciso ya que nunca pueden estar libres de errores.
- Comprender cómo las diferentes fuentes de error generan bias y varianza te ayudará a mejorar el proceso de ajuste de datos, lo que resulta en modelos más precisos, adicionalmente también evitarás el error de sobreajuste y falta de ajuste.

Bias (Sesgo) y Varianza en Machine Learning

El error de predicción para cualquier algoritmo de Machine Learning se puede dividir en tres partes:

Bias (Sesgo) y Varianza en Machine Learning

Error reductible

El **error irreducible** no se puede reducir, independientemente de qué algoritmo se usa. También se le conoce como ruido y, por lo general, proviene por factores como variables desconocidas que influyen en el mapeo de las variables de entrada a la variable de salida, un conjunto de características incompleto o un problema mal enmarcado.

No importa cuán bueno hagamos nuestro modelo, nuestros datos tendrán cierta cantidad de ruido o un error irreductible que no se puede eliminar.

Bias (Sesgo) y Varianza en Machine Learning

Error de bias o sesgo

Es la diferencia entre la predicción esperada de nuestro modelo y los valores verdaderos.

- **Bajo bias**: sugiere menos suposiciones sobre la forma de la función objetivo. Los algoritmos de Machine Learning con baja bias incluyen: árboles de decisión, k-vecinos más cercanos y máquinas de vectores de soporte.
- **Alto bias**: sugiere más suposiciones sobre la forma de la función objetivo. Los algoritmos con alto bias se incluyen: regresión lineal, análisis discriminante lineal y regresión logística.

Bias (Sesgo) y Varianza en Machine Learning

Error de varianza

Los algoritmos con alto bias se incluyen: regresión lineal, análisis discriminante lineal y regresión logística.

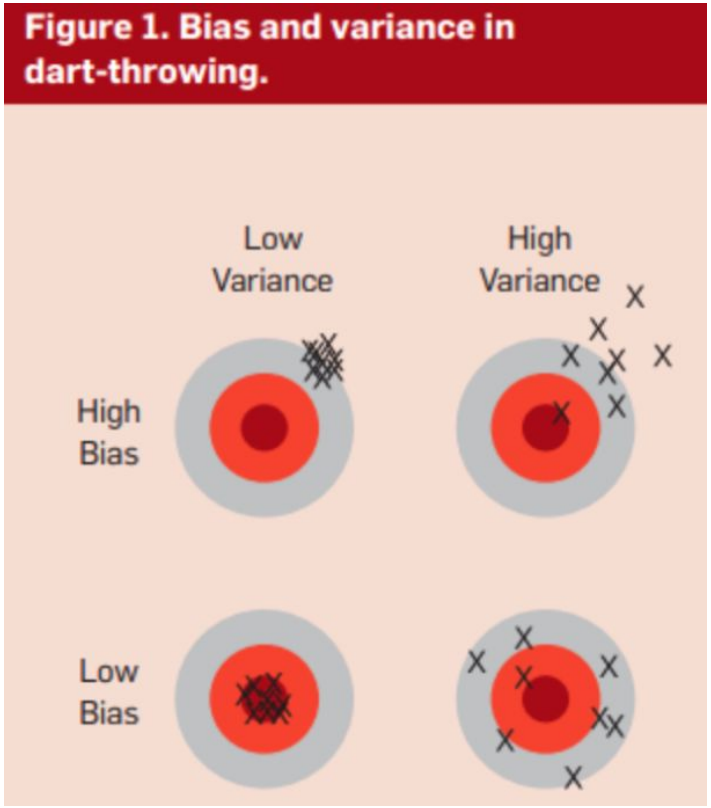
La función objetivo se estima a partir de los datos de entrenamiento mediante un algoritmo de Machine Learning, por lo que deberíamos esperar que el algoritmo tenga alguna variación. Idealmente no debería cambiar demasiado de un conjunto de datos de entrenamiento a otro, lo que significa que el algoritmo es bueno para elegir el mapeo subyacente oculto entre las variables de entrada y de salida.

Los algoritmos de Machine Learning que tienen una gran varianza están fuertemente influenciados por los detalles de los datos de entrenamiento, esto significa que los detalles de la capacitación influyen en el número y los tipos de parámetros utilizados para caracterizar la función de mapeo.

Bias (Sesgo) y Varianza en Machine Learning

La compensación
Bias-Varianza o Trade-off

El objetivo de cualquier algoritmo supervisado de Machine Learning es lograr un bias bajo y una baja varianza, a su vez, el algoritmo debe lograr un buen rendimiento de predicción.

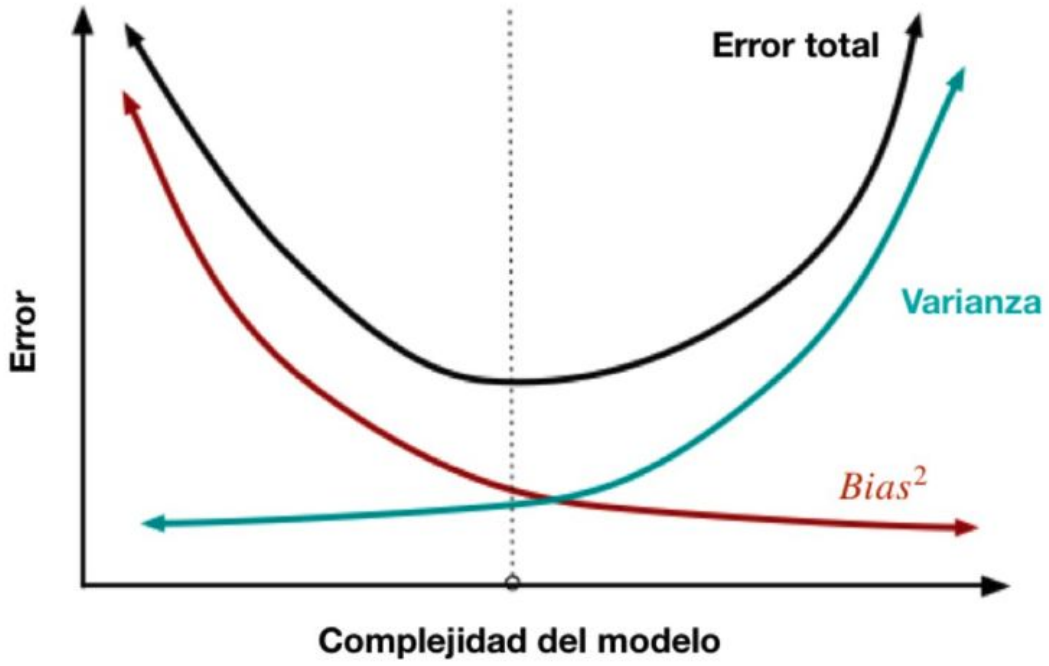


Bias (Sesgo) y Varianza en Machine Learning

Error Total

Comprender el bias y la varianza es fundamental para comprender el comportamiento de los modelos de predicción, pero en general lo que realmente importa es el error general, no la descomposición específica. El punto ideal para cualquier modelo es el nivel de complejidad en el que el aumento en el bias es equivalente a la reducción en la varianza.

Para construir un buen modelo, necesitamos encontrar un buen equilibrio entre el bias y la varianza de manera que minimice el error total.

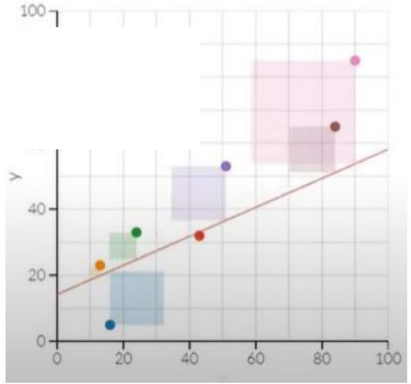


”Para construir un buen modelo, necesitamos encontrar un buen equilibrio entre el bias y la varianza de manera que minimice el error total.”

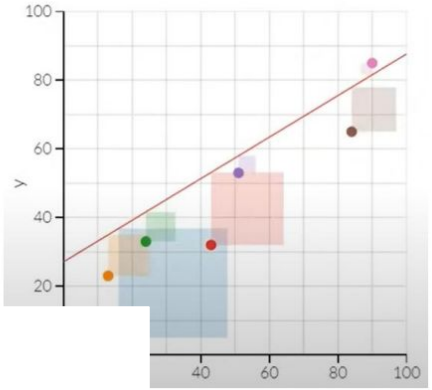
PARTE 2:
Descenso del Gradiente

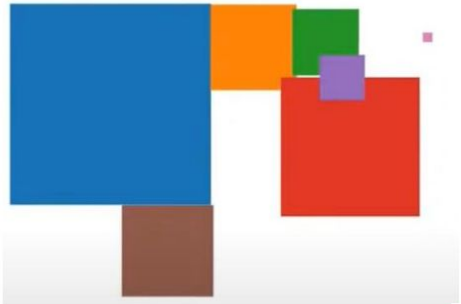
By: Mgr. Ruth Chirinos

Descenso del Gradiente



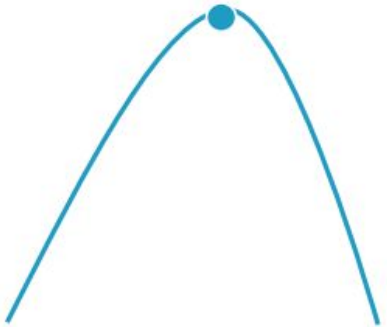
Error = 175





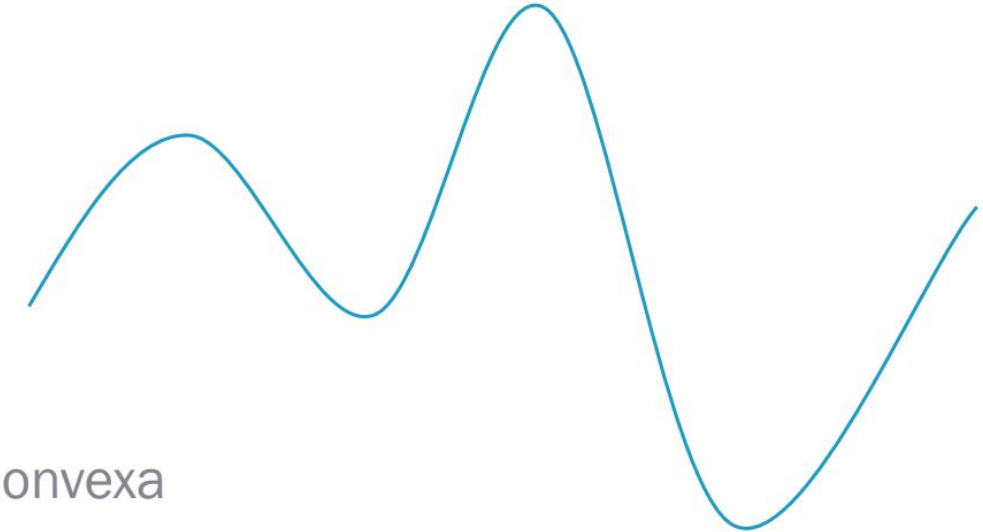
Error = 352

Descenso del Gradiente



- ✓ f(x) convexa
- ✓ Punto Mínimo
- ✓ f(x) cóncava
- ✓ 1 – f(x) convexa 

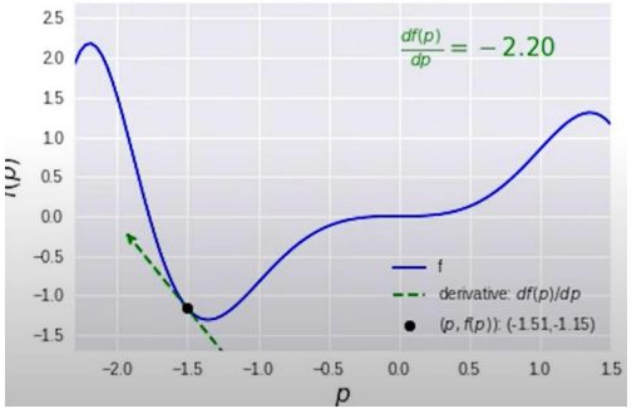
Descenso del Gradiente

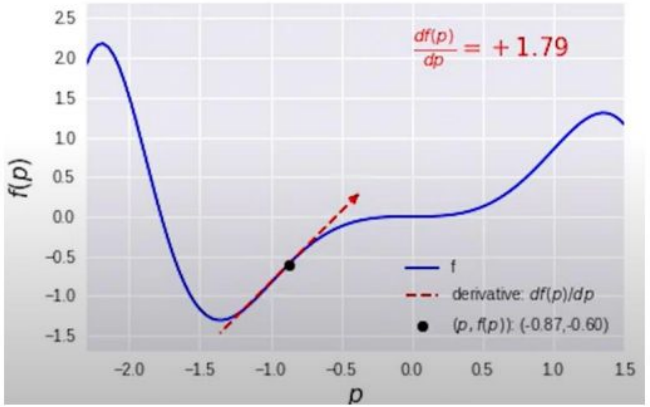


- ✓ f(x) no convexa
- ✓ Es posible encontrar un punto mínimo que no sea el mínimo global
- ✓ Esto es un problema

Descenso del Gradiente

- ✓ Si **derivamos un función** podemos saber la pendiente de la función en ese punto.

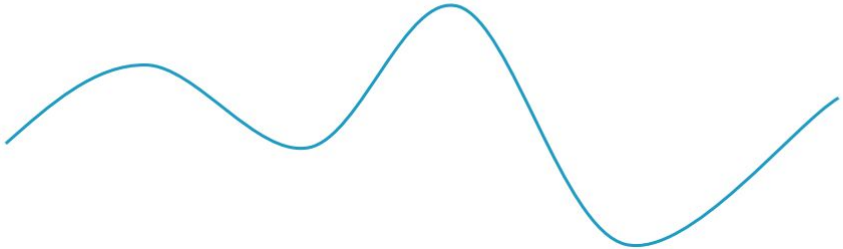




Descenso del Gradiente

- ✓ Si la derivada de la función la igualamos a cero $f'(x) = 0$, es decir, la pendiente es nula, hemos encontrado el punto mínimo.
- ✓ En funciones convexas solo hay un punto mínimo.

- ✓ En funciones no convexas nos encontramos muchos puntos mínimos y muchas ecuaciones que resolver.

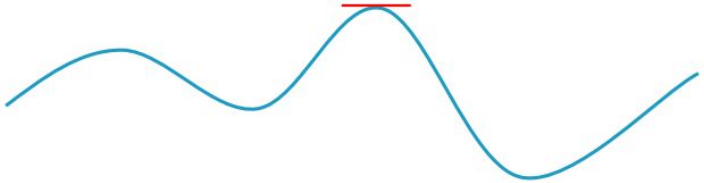


Descenso del Gradiente > Problemas de las Funciones No Convexas

- ✓ Mínimos locales.

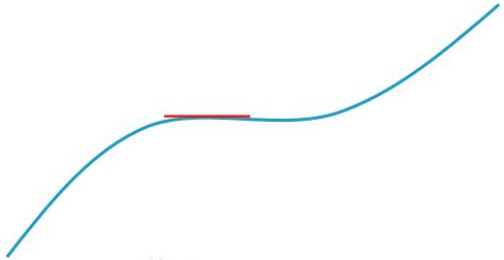
  Muchos puntos mínimos. Muchas funciones a resolver.

- ✓ Pendiente Nula en Máximos Locales

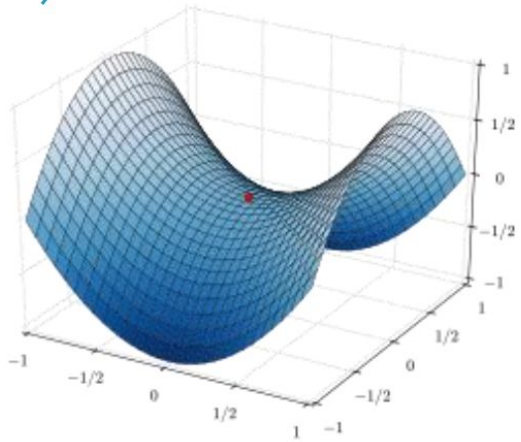


Descenso del Gradiente > Problemas de las Funciones No Convexas

- ✓ Pendiente Nula en Puntos de Inflexión.



- ✓ Pendiente Nula en Puntos de silla.



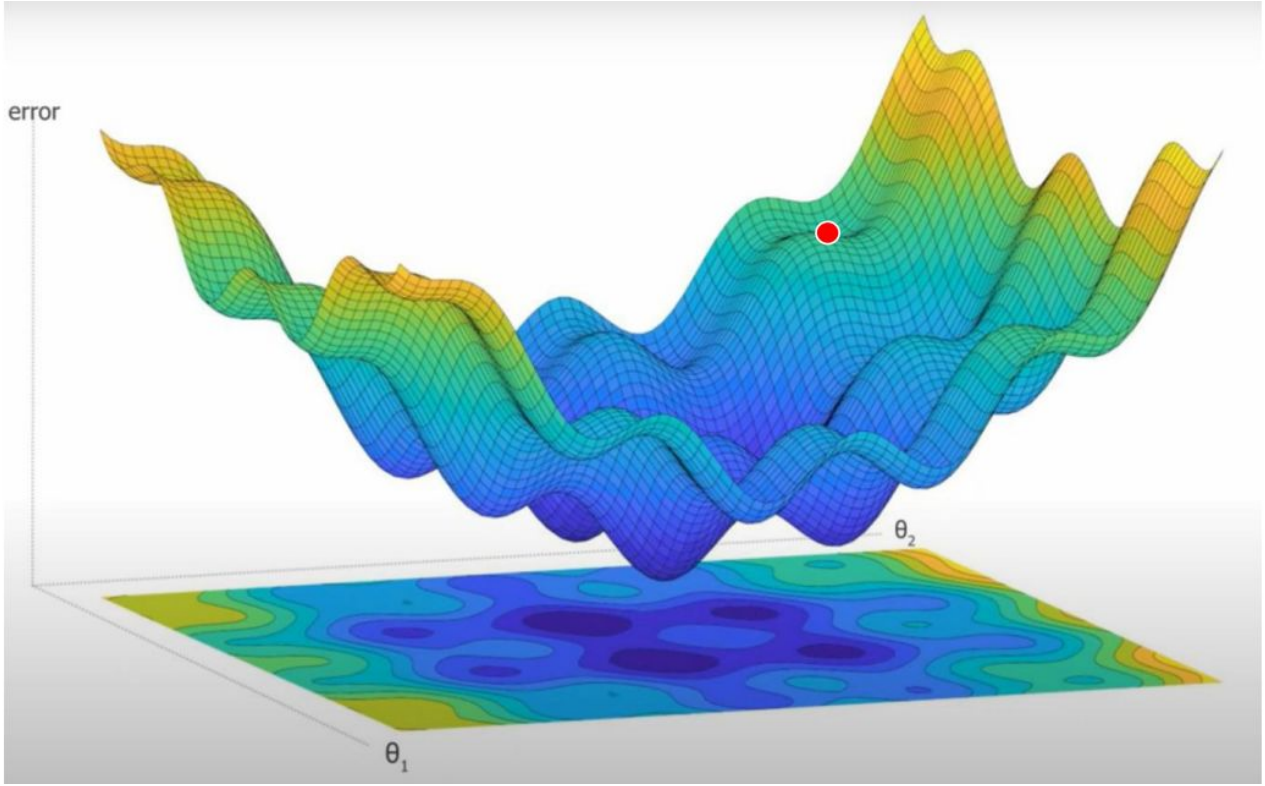
Descenso del Gradiente > Problemas de las Funciones No Convexas

- ✓ Sistema de Ecuaciones Enorme e Ineficiente.
- ✓ En Regresión Lineal, podemos usar el error cuadrático medio como función de coste al poder obtener una función convexa. En redes neuronales no es útil, no lo podemos usar.
- ✓ Pero... Hay que buscar una solución para funciones no convexas.

Idea del Descenso del Gradiente

Desierto de Almería

Descenso del Gradiente

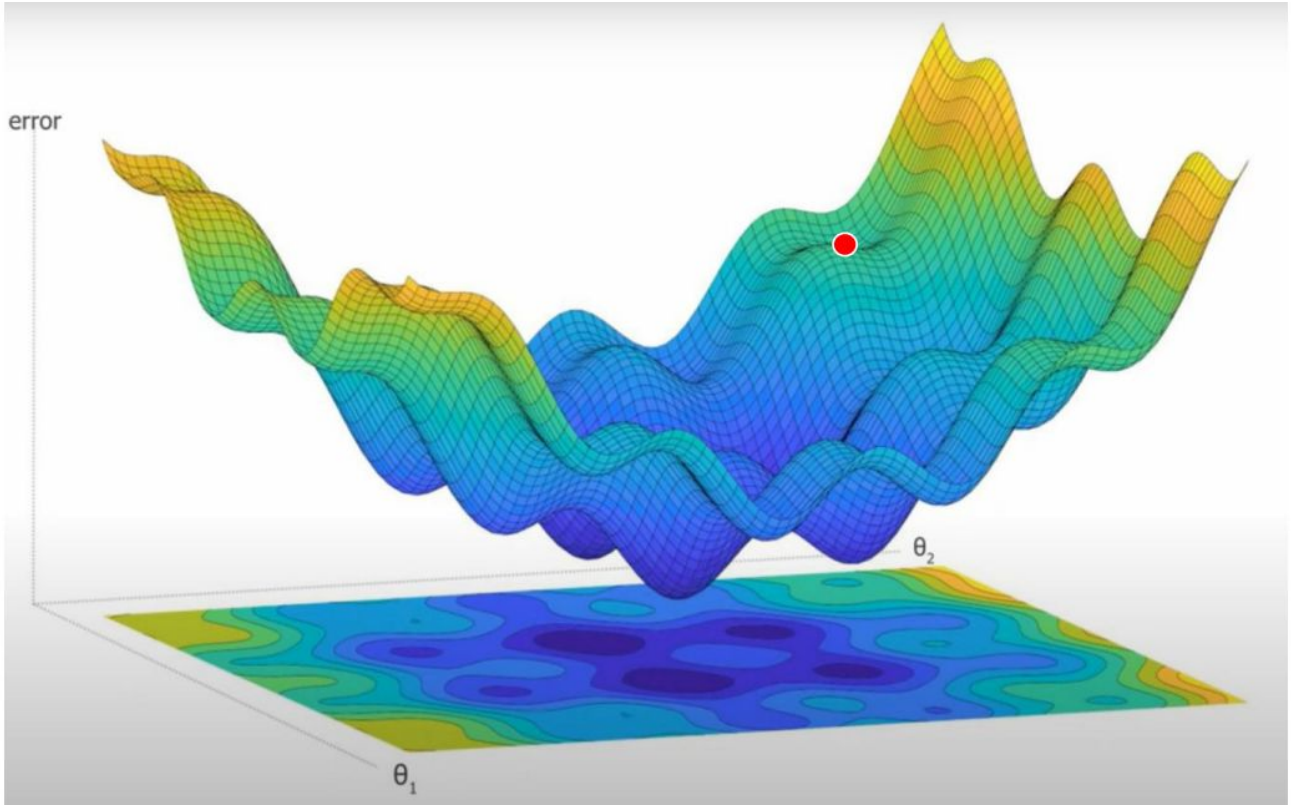


Ejes X, Y → parámetros

Eje Z → Error

Los parámetros se inicializan con un valor aleatorio (cualquier punto del terreno)

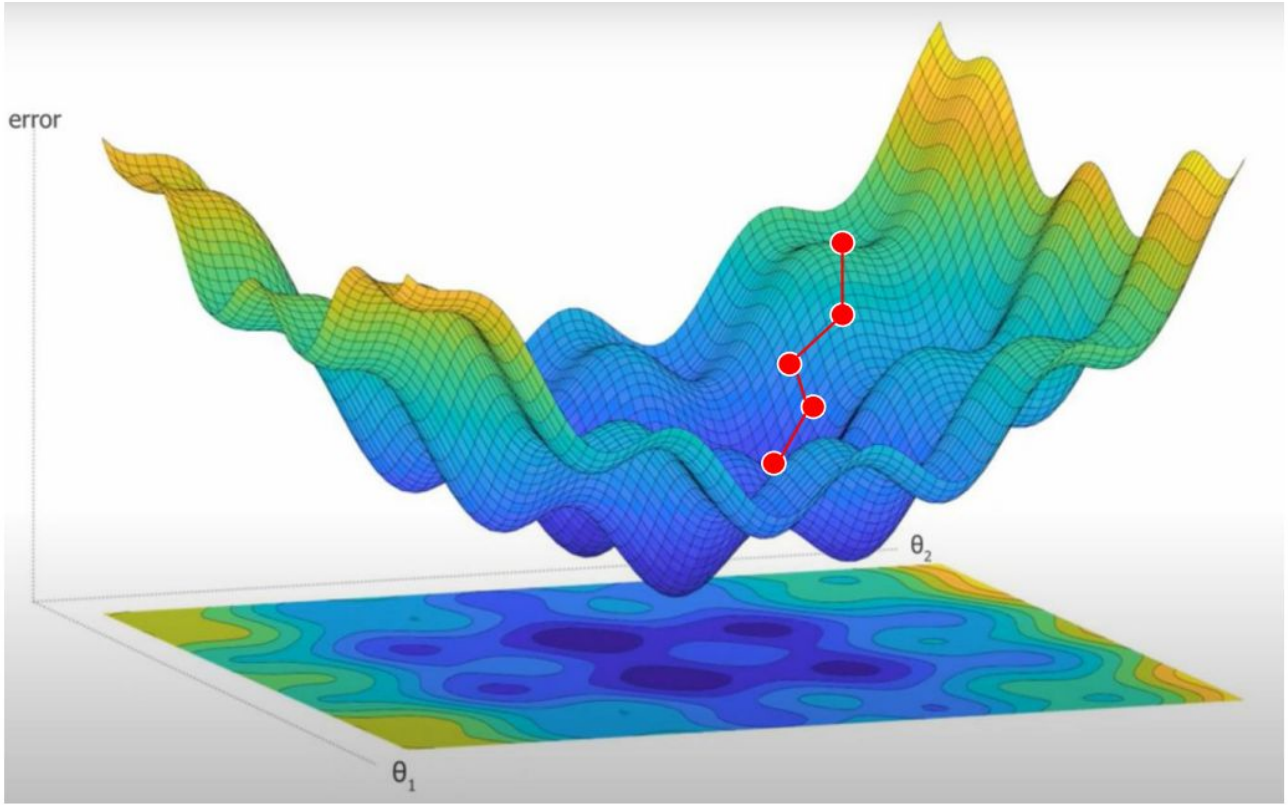
Descenso del Gradiente



$$\theta = (\theta_1, \theta_2)$$

$$\begin{bmatrix} \frac{\partial error}{\partial \theta_1} \\ \\ \frac{\partial error}{\partial \theta_2} \end{bmatrix} = \nabla f$$

Descenso del Gradiente



$$\theta = (\theta_1, \theta_2)$$

$$\begin{bmatrix} \frac{\partial error}{\partial \theta_1} \\ \\ \frac{\partial error}{\partial \theta_2} \end{bmatrix} = \nabla f$$

$$\theta = \theta - \nabla f$$

Descenso del Gradiente > Resumiendo consiste en...

1. Empezamos en un punto (θ_1, θ_2) al azar.
2. Encontramos la dirección de máxima pendiente hacia abajo
3. Damos un paso en esa dirección
4. Repetimos el proceso desde el nuevo punto obtenido hasta llegar a un mínimo

Descenso del Gradiente > Resumiendo consiste en...

- ✓ El Descenso del Gradiente consiste en Repetir

$$\theta = \theta - \nabla f$$

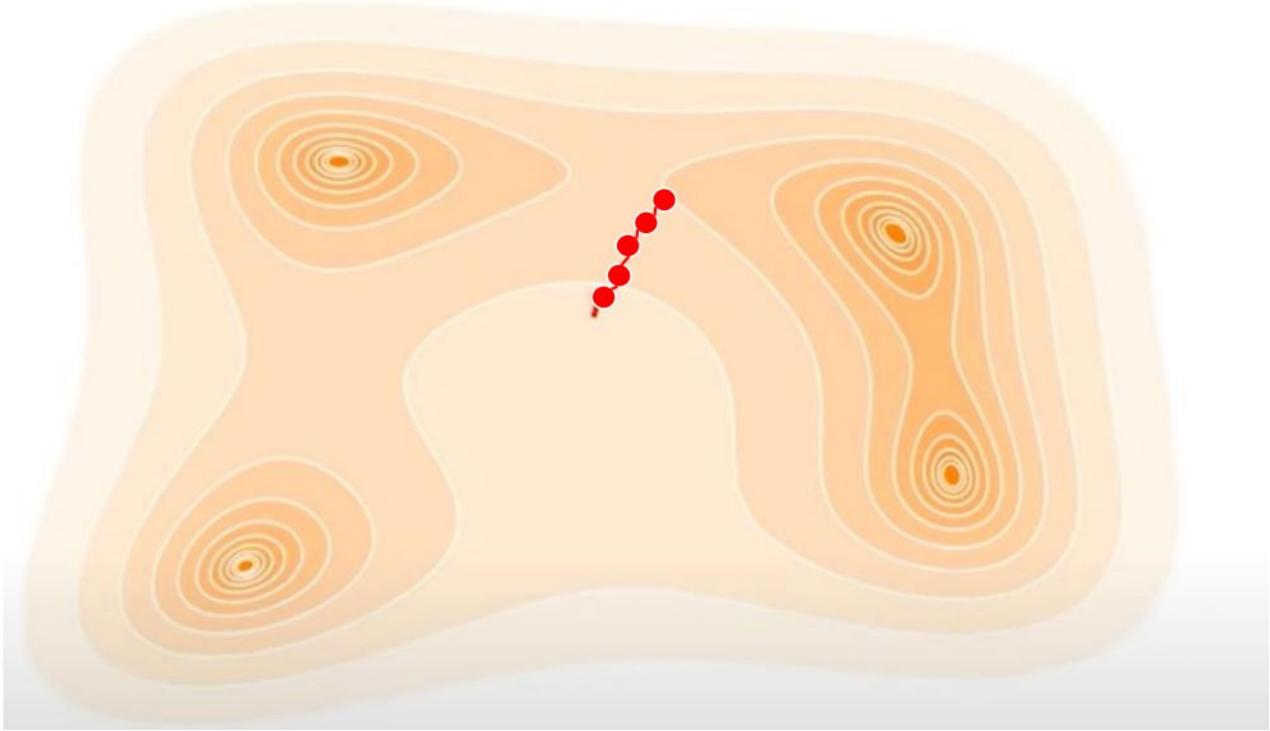
... Hasta Converger

- ✓ PERO

- ✓ ¿Qué distancia recorro cada vez que me acerco a mi objetivo?

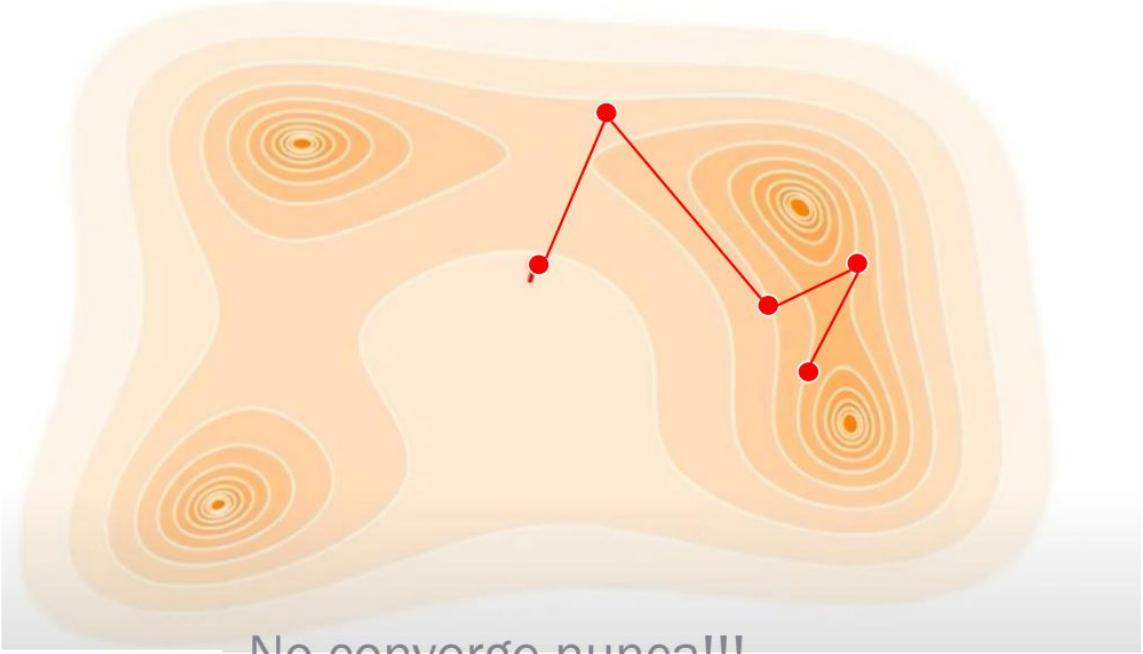
Descenso del Gradiente

- ✓ Me acerco poco a poco... para no equivocarme



Descenso del Gradiente

- ✓ Vamos más deprisa... para llegar antes



No converge nunca!!!

Descenso del Gradiente

- ✓ Necesitamos determinar la distancia a la que sea aproxima a un mínimo en cada paso

$$\theta = \theta - \alpha \nabla f$$

- ✓ α es el learning rate

- ✓ Hay diferentes técnicas para determinar el valor óptimo de α (pero no las vamos a ver ahora)

Descenso del Gradiente > Recapitulando...

- ✓ Entonces ya sabemos como calcular el coste del error

Función de Coste

- ✓ Y también como varía el coste ante un cambio en los parámetros

$$\begin{bmatrix} \frac{\partial error}{\partial \theta_1} \\ \\ \frac{\partial error}{\partial \theta_2} \end{bmatrix} = \nabla f$$

- ✓ Y como ir optimizando los parámetros

Descenso del Gradiente

Descenso del Gradiente

- ✓ La variación del coste, en función de la variación de los parámetros $\left[\frac{\partial Coste}{\partial w}\right]$ es muy difícil de calcular

- ✓ ¿Cómo variamos los valores de w para mejorar el coste?

El descenso de gradiente (Gradient Descent en inglés) es una ***técnica de optimización*** *ampliamente* ***utilizada en el campo del aprendizaje automático y la optimización numérica****.*

Se utiliza principalmente para resolver problemas de optimización.

Aunque no es un algoritmo de aprendizaje automático en sí mismo, es esencial en muchos algoritmos de entrenamiento de modelos de aprendizaje automático y se utiliza en diversas aplicaciones.

Principales áreas donde se usa el descenso del gradiente

- **Aprendizaje Automático Supervisado:**
  - Regresión Lineal: Se utiliza para ajustar los coeficientes de una regresión lineal para minimizar la función de costo (por ejemplo, el error cuadrático medio).
  - Regresión Logística: Se utiliza para encontrar los parámetros que maximizan la verosimilitud en la regresión logística, que se utiliza en clasificación binaria y multiclase.
- **Aprendizaje Automático No Supervisado:**
  - Agrupamiento (Clustering): Se utiliza en algoritmos de agrupamiento como el k-means para encontrar los centroides óptimos.
  - Reducción de Dimensionalidad: En técnicas como el Análisis de Componentes Principales (PCA), se utiliza para encontrar las proyecciones que retienen la mayor varianza.
- **Aprendizaje Profundo (Deep Learning):**
  - Redes Neuronales: El descenso de gradiente se aplica en la retropropagación (backpropagation) para ajustar los pesos de las capas ocultas de las redes neuronales profundas.
- **Aprendizaje por Reforzamiento:**
  - Aprendizaje Q: En el aprendizaje por refuerzo, se utiliza para aprender la función Q óptima en algoritmos como Q-Learning y Deep Q-Networks (DQN).
- **Optimización de Funciones en General:**
  - Optimización de Funciones No Convexas: El descenso de gradiente estocástico (SGD) se utiliza para optimizar funciones no convexas en diversos campos, como la economía y la ingeniería.

Thanks!

Any questions ?

You can find me at

- Twitter: @ruthy_root
- Email: ruth.chirinos@gmail.com