

DIPLOMADO EN CIENCIA DE DATOS

MODELACIÓN SUPERVISADA SEMANA 2

Facultad de Estudios Superiores Acatlán

OUTLINE

Módulo 2 - Semana 2

- 1.- Continuación de Optimización de Funciones de Costo
- 2.- Regresión Lineal
- 3.- Sobreajuste y Desajuste de modelos supervisados

QUIZZES SEMANALES



Quizzes

- Cuatro quizzes, todos los Sábados de 8:15 a 8:30 am
- 10 preguntas (la mayoría de opción múltiple, una que otra pregunta abierta)
- Preguntas respecto a temas vistos una semana anterior a la fecha del quizz



QUIZ 1

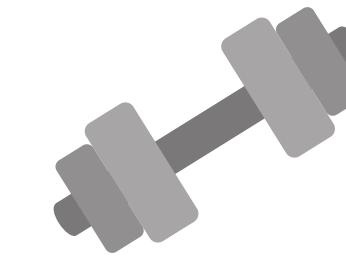


<https://b.socrative.com/login/student/>
Room name: IRENE2290

1.- Optimización de funciones de costo

OPTIMIZACIÓN DE FUNCIONES DE COSTO

ENTRENAMIENTO DE UN MODELO



¿Qué significa entrenar un modelo?

Entrenamiento de un modelo supervisado

- Dado un data set (\mathbf{x}_n, y_n) , deseamos encontrar los parámetros $\mathbf{W} = (w_0, w_1, \dots, w_D)$ óptimos. Tal que $y_n \approx f(\mathbf{x}_n)$
- Para determinar dichos parámetros, es necesario encontrar una función de costo $\mathcal{L}(\mathbf{W})$ y encontrar el punto óptimo $\mathbf{W} = (w_0, w_1, \dots, w_D)$, donde la función alcanza un mínimo global, i.e

distancia entre
predicción y valor real

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) \quad \text{tal que } \mathbf{W} \in R^D$$

- Existen varias maneras de dar solución al problema de optimización planteado anteriormente
- Deseable: encontrar el mínimo global para la función de costo

OPTIMIZACIÓN DE FUNCIONES DE COSTO

FUNCIONES DE COSTO Y PROPIEDADES



 Las funciones de costo juegan un papel importante en el entrenamiento de modelos supervisados

Funciones de Costo

- También llamada función de pérdida, es utilizada para entrenar parámetros que expliquen un set de datos
- Cuantifica que tan bueno es el desempeño del modelo. En otras palabras, cuantifica qué tan costosos son los errores cometidos por el modelo

Dos propiedades deseadas

- Cuando la variable target es numérica, se desea lo siguiente:
 - Función de costo sea simétrica alrededor del número cero
 - Función de costo penalice errores grandes de manera similar a los errores muy grandes

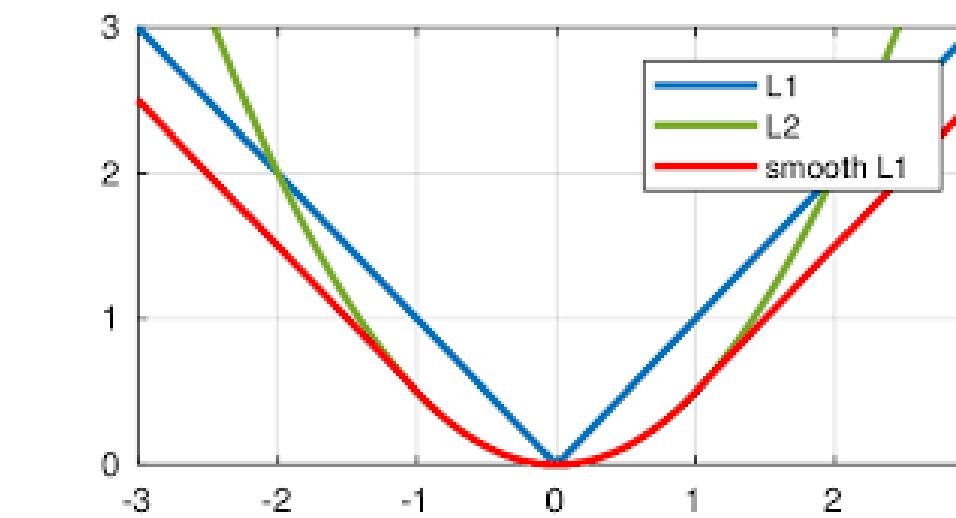
Funciones de costo populares

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

- ✓ Simétrica alrededor del cero
- ✗ No robusta cuando el data set contiene outliers

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

- ✓ Simétrica alrededor del cero
- ✓ Robusta cuando el data set contiene outliers



OPTIMIZACIÓN DE FUNCIONES DE COSTO

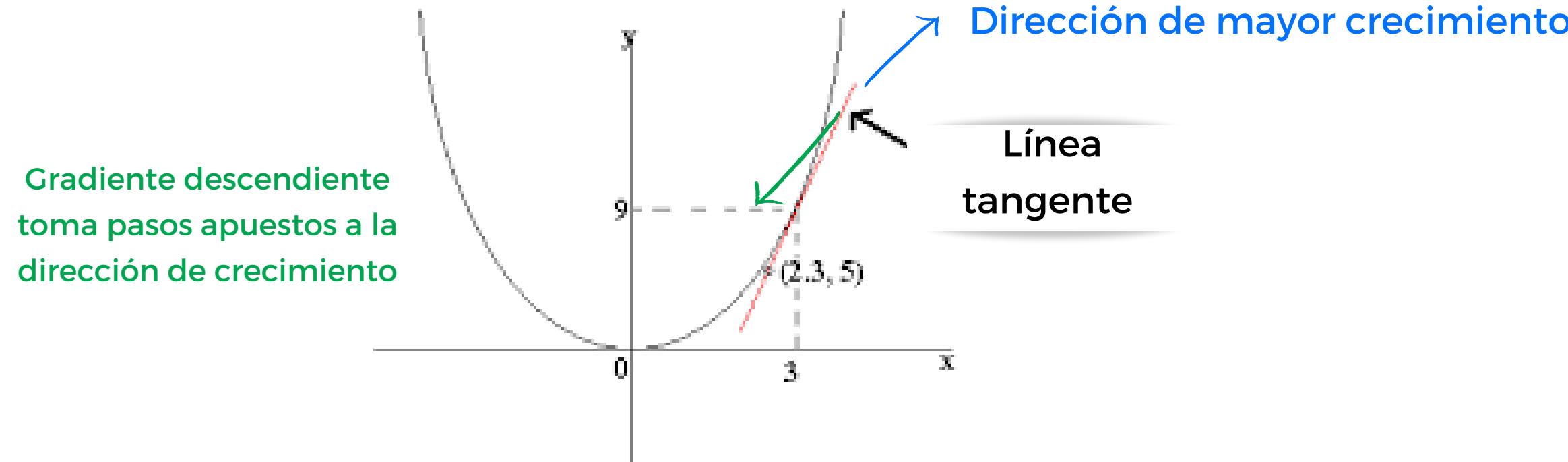
MÉTODOS DE OPTIMIZACIÓN - GRADIENTE DESCENDENTE



🎯 Algoritmo de optimización para encontrar parámetros óptimos tal que función de costo toque su punto mínimo al ser evaluada en esos parámetros

Optimización suave

- El gradiente (derivada) existe para una determinada función de costo
- Cuando el gradiente es evaluado en un punto, tenemos una pendiente tangente a la función en ese punto.
Dicha tangente apunta en la dirección del mayor incremento de la función
- Para minimizar una función de costo, damos de manera iterativa pasos en sentido contrario a la dirección del gradiente



OPTIMIZACIÓN DE FUNCIONES DE COSTO

MÉTODOS DE OPTIMIZACIÓN - GRADIENTE DESCENDENTE



🎯 La idea general de este algoritmo consiste en ir modificando parámetros de manera iterativa para minimizar el valor de la función de costo

Algoritmo

- El parámetro γ se tiene que optimizar.

Inicializar vector $\mathbf{W}^{(0)}$ con un valor al azar o en ceros

Para $i = 1$ hasta `max_iteraciones`:

 Calcular el valor de la función de costo $\mathcal{L}(\mathbf{W}^{(t)})$

 Calcular el gradiente de la función de costo $\nabla \mathcal{L}(\mathbf{W}^{(t)})$

 Actualizar el valor de $\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \gamma \nabla \mathcal{L}(\mathbf{W}^{(t)})$

 Guardar valor de $\mathcal{L}(\mathbf{W}^{(t)})$ y $\mathbf{W}^{(t)}$

Regresar las listas de $\mathcal{L}(\mathbf{W}^{(t)})$ y $\mathbf{W}^{(t)}$

Tasa de aprendizaje γ

- Si la tasa de aprendizaje es muy pequeña, entonces el algoritmo tendrá que llevar a cabo muchas iteraciones para converger, lo tardará algo de tiempo
- Si la tasa de aprendizaje es muy alta, se corre con el riesgo de dar pasos muy grandes, inclusive saltar de un lado al otro. Existe la posibilidad de que el algoritmo diverja

OPTIMIZACIÓN DE FUNCIONES DE COSTO

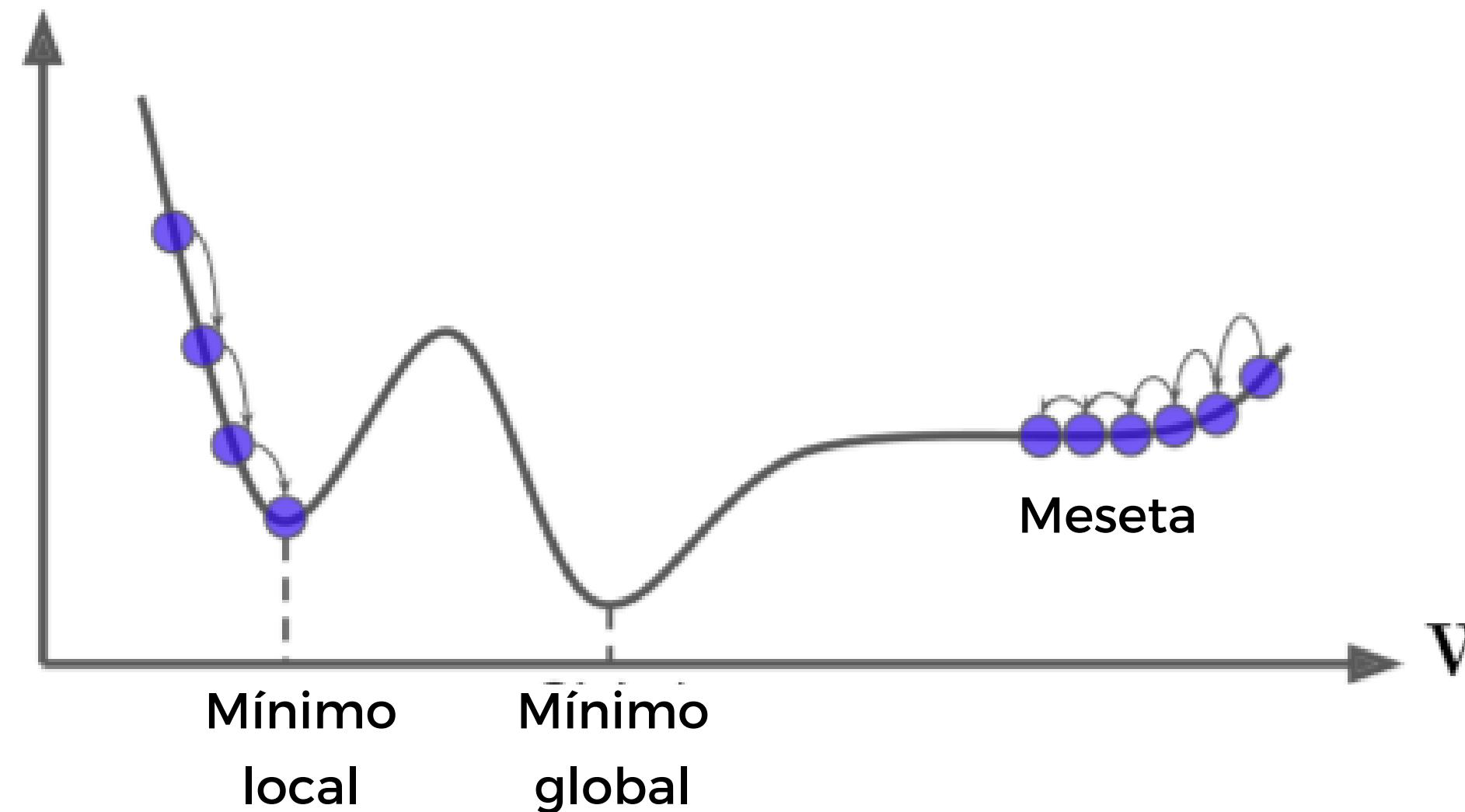
MÉTODOS DE OPTIMIZACIÓN - GRADIENTE DESCENDENTE



Un reto al que este algoritmo se enfrenta es cuando la función no es convexa

Escenario complicado para el algoritmo

Costo*



- Funciones de costo con formas irregulares pueden hacer la convergencia del algoritmo complicada
- Si el algoritmo inicia en el lado izquierdo de la imagen, convergería a un mínimo local
- Si el algoritmo inicia en el lado derecho de la imagen, tardaría mucho en cruzar toda la meseta, y si además el algoritmo termina pronto, jamás llegaría al mínimo global

OPTIMIZACIÓN DE FUNCIONES DE COSTO

FUNCIONES CONVEXAS



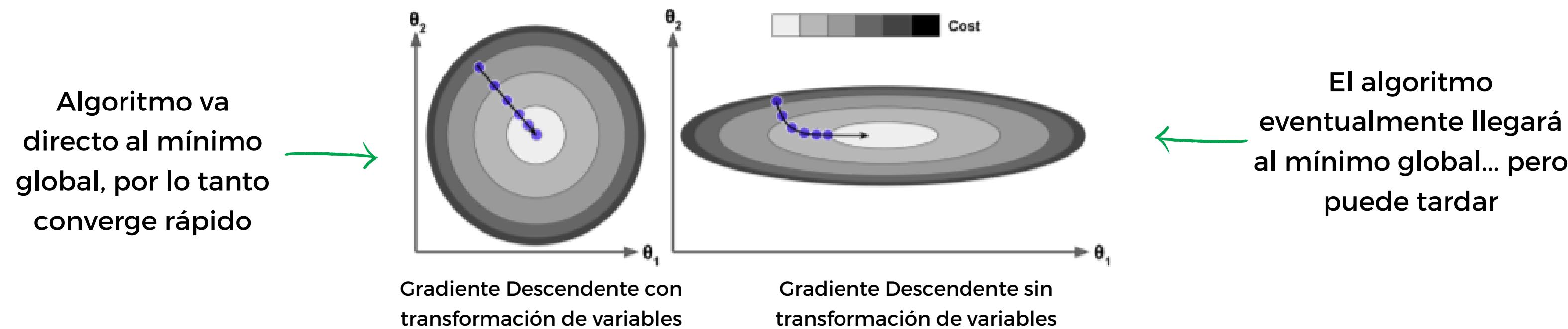
Para las funciones de costo convexas, los mínimos locales son mínimos globales

Funciones convexas

- Una función es convexa si una línea que une a dos puntos cualesquiera en la curva, nunca interseca con la misma en otras partes

Una función $f(\mathbf{X})$ es convexa, si para cualquier \mathbf{u}, \mathbf{v} y para $0 \leq \lambda \leq 1$ se tiene: $f(\lambda\mathbf{u} + (1 - \lambda)\mathbf{v}) \leq \lambda f(\mathbf{u}) + (1 - \lambda)f(\mathbf{v})$

- Lo anterior no hará milagros...



OPTIMIZACIÓN DE FUNCIONES DE COSTO

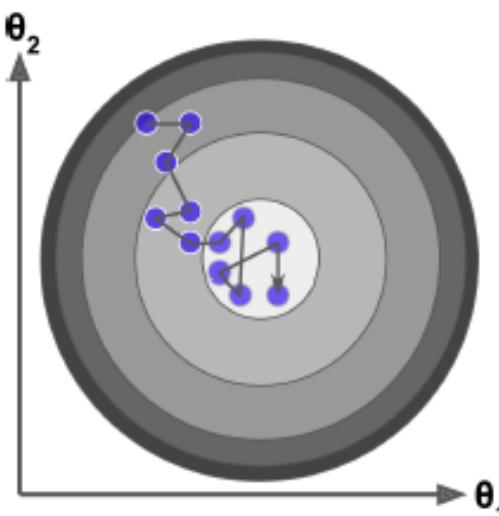
OTROS MÉTODOS DE OPTIMIZACIÓN



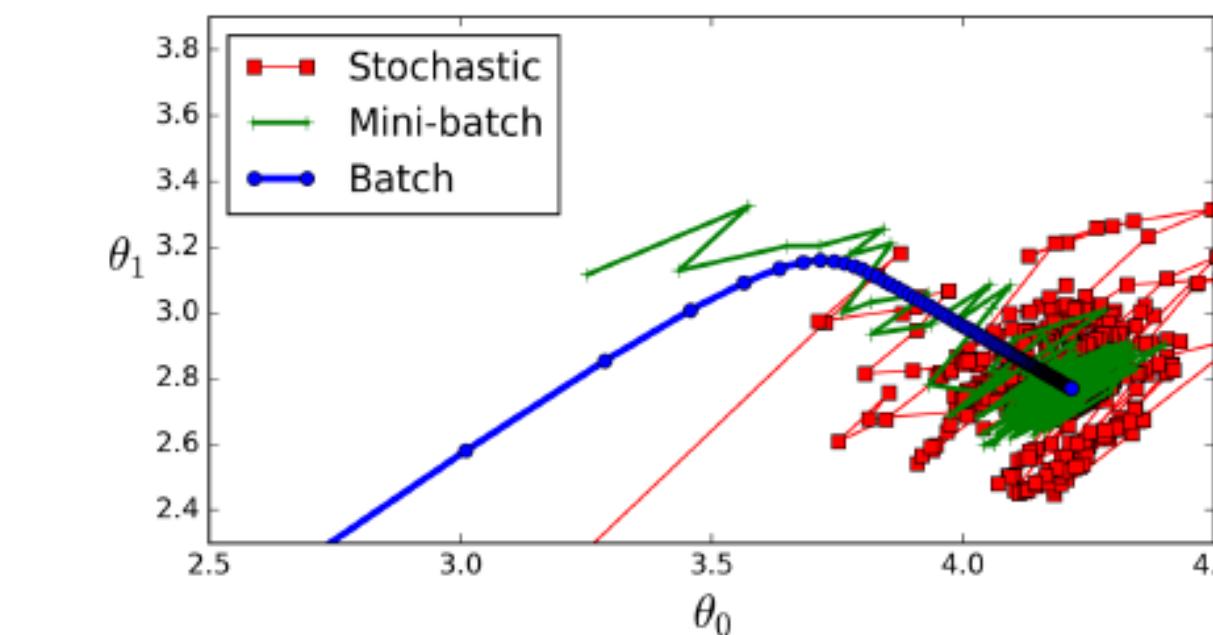
🎯 El algoritmo anterior toma el set de datos completo en cada iteración, tardando mucho en completar todo el ciclo iterativo cuando el set de datos es grande... otro problema se presenta cuando la función de costo no cuenta con gradiente en todos los puntos

Optimización suave

- Gradiente Descendente Estocástico
 - Opuesto a lo que hace el gradiente descendente, selecciona particiones del set de datos al azar en cada iteración. Lo anterior hace que el algoritmo termine de manera rápida, pero al ser un método estocástico, es inestable.
 - Puede ser que los valores finales para los parámetros sean buenos, pero no óptimos.



- Gradiente Descendente en Mini batch
 - En cada iteración del algoritmo se utilizan una serie de particiones del set de datos elegidas al azar (mini batches). Muy útiles para optimizar recursos tecnológicos, pues se puede parallelizar utilizando GPUs



2.- Regresión Lineal

REGRESIÓN LINEAL

CONCEPTOS BÁSICOS



Un modelo de regresión lineal asume una relación lineal entre la variable dependiente y las independientes



Definición

- Target numérica → Regresión
- Un set de datos está compuesto por el par (\mathbf{x}_n, y_n)
- Regresión lineal simple

$$y_n \approx f(\mathbf{x}_n) + \epsilon = w_0 + w_1 x_{n1} + \epsilon$$

- Regresión lineal múltiple

$$\begin{aligned} y_n &\approx f(\mathbf{x}_n) + \epsilon = w_0 + w_1 x_{n1} + \dots + w_D x_{nD} + \epsilon \\ &= \mathbf{x}_n^T \mathbf{W} \end{aligned}$$

- Función de costo

$$\mathcal{L}(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{x}_n^T \mathbf{W})^2$$

Problema: Deficiencia de rango

- Cuando el número de variables independientes es mayor que el número de observaciones que tenemos en un set de datos, el modelo de regresión lineal queda subdeterminado
- A este problema también se le conoce como problema $D > N$
- Si $D \leq N$, pero algunas variables independientes son (casi) colineales, entonces la matriz tiene problemas de mal condicionamiento (ill-conditioning)
- Solución: Utilizar programas que solucionan sistemas lineales (más de esto en breves)

REGRESIÓN LINEAL

CORRELACIÓN \neq CAUSALIDAD

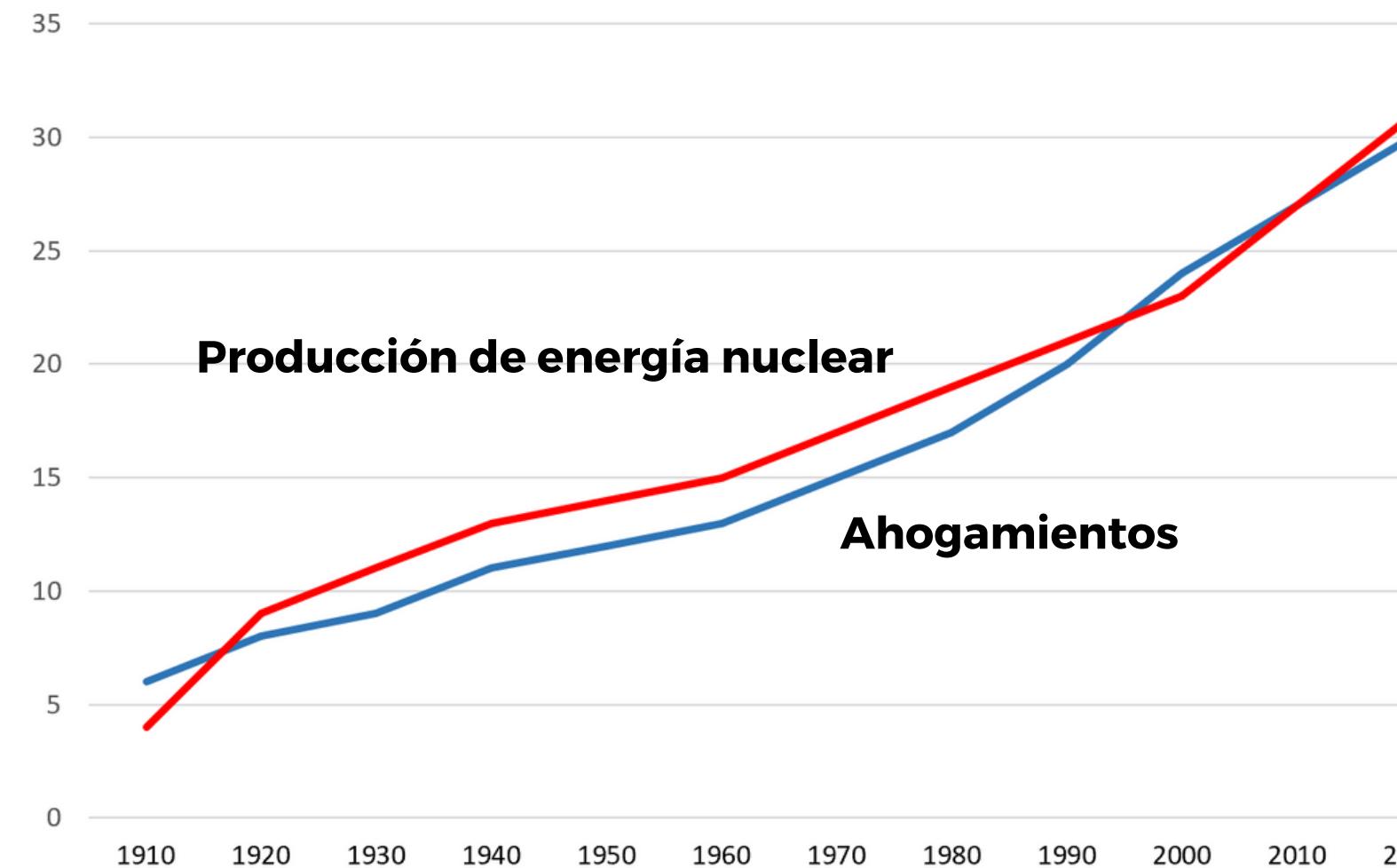


🎯 La regresión lineal encuentra la correlación lineal entre inputs y output, más no la relación causal del fenómeno.

Correlación Espuria

¿Qué ven de raro en la siguiente imagen?

Número de ahogamientos vs. producción de energía nuclear



Correlación no es igual a causalidad, interpreta tus resultados con precaución

REGRESIÓN LINEAL

FORMAS DE RESOLVER UN PROBLEMA DE REGRESIÓN LINEAL



🎯 Pocas funciones de costo permiten tener una fórmula para determinar parámetros óptimos. Dicha fórmula engloba la inversa de una matriz. ¿Qué podría afectar la determinación de parámetros óptimos usando la solución de mínimos cuadrados?

Mínimos cuadrados

- Función de costo MSE permite tener una fórmula para determinar los parámetros óptimos para una regresión lineal

$$\mathbf{W} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Gradiente descendente

- Algoritmo de optimización para encontrar parámetros tal que minimicen una función de costo

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \gamma \nabla \mathcal{L}(\mathbf{W}^{(t)})$$

Librería Numpy

- Módulo de álgebra lineal de Numpy
- Muy útil cuando la inversa de un sistema lineal no existe, pues la aproxima utilizando la inversa de Moore-Penrose a través del método de descomposición de valores singulares

```
np.linalg.pinv(X).dot(y)
```

Librería Sklearn

- Método más utilizado (el que usaremos para este módulo)

```
from sklearn.linear_model import LinearRegression  
LinReg = LinearRegression()  
LinReg.fit(X_train, y_train)  
LinReg.predict(X_test)
```

REGRESIÓN LINEAL

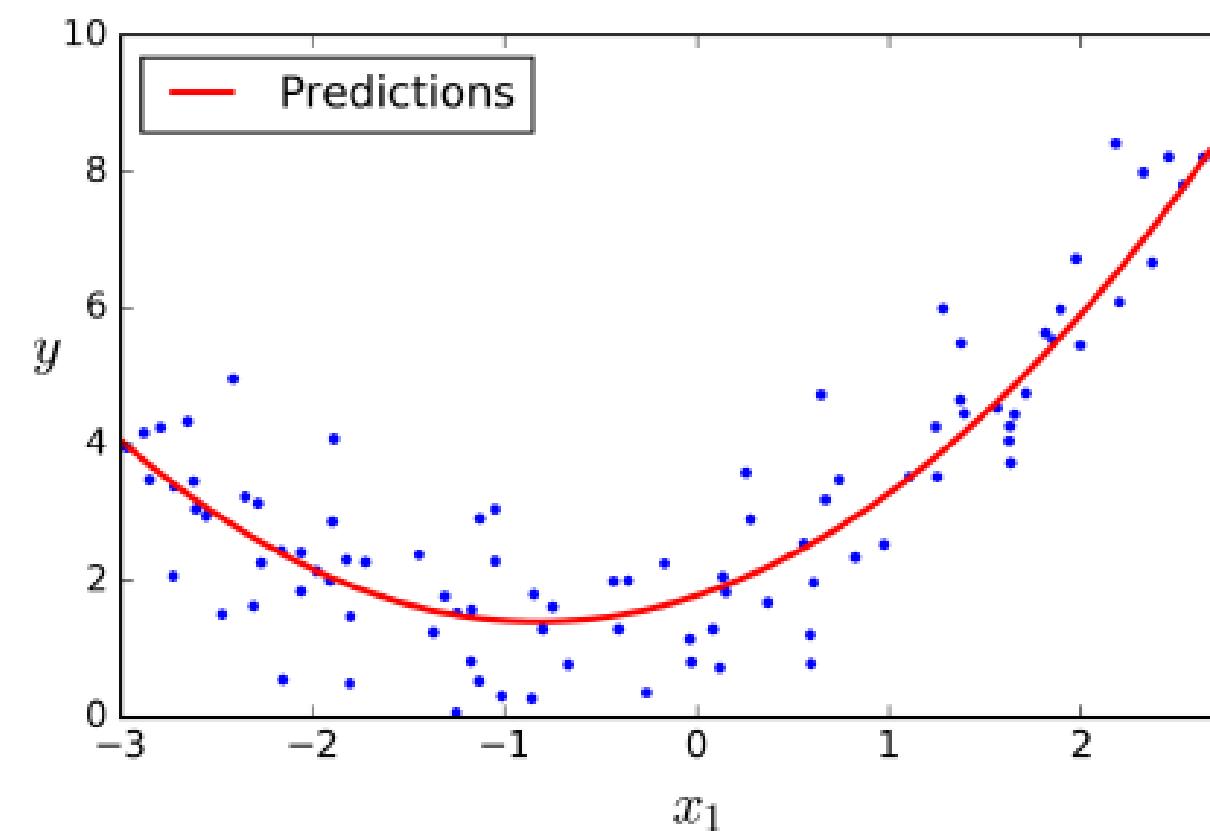
UN PROBLEMA CON LOS MODELOS LINEALES



🎯 Los modelos lineales son muy sencillos para poder modelar distribuciones complejas. Una forma de incrementar la potencia de éstos modelos es al aumentar o extender el set de datos con una base polinomial de grado M

Aumentar o extender el set de datos para poder ajustar de mejor manera un modelo lineal

¿Cómo podríamos aumentar el input de datos tal que podamos tener la predicción en color rojo?



- Vemos que la distribución de puntos azules tiene una forma cuadrática
- Podríamos agregar el cuadrado de cada variable independiente al set de entrenamiento original
- Lo anterior nos dará un set de datos aumentado para poder ajustar un modelo de regresión lineal y determinar los parámetros óptimos

3.- Sobreajuste y Desajuste de modelos supervisados



SOBREAJUSTE Y DESAJUSTE

EVITAR FENÓMENOS QUE AFECTEN LA GENERALIZACIÓN DE UN MODELO



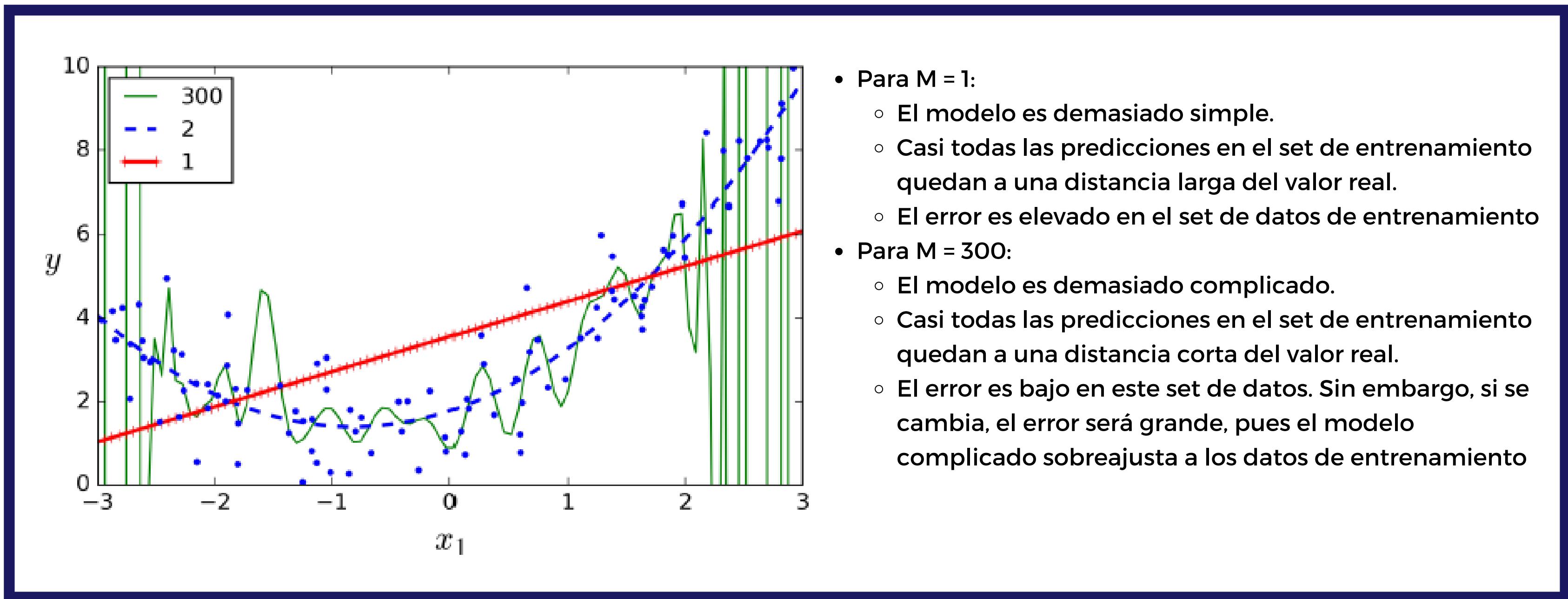
Nuestro objetivo al entrenar un modelo es encontrar la distribución subyacente que nos ayude a generalizar predicciones en otros sets de datos diferentes al set de datos de entrenamiento, más no tomar una foto de éste último.

SOBREAJUSTE Y DESAJUSTE

MODELOS SIMPLES Y COMPLICADOS



🎯 Los modelos lineales por si solos son muy simples. Si extendemos el set de datos utilizando polinomios de grado M podemos aumentar la complejidad del mismo. Sin embargo, podríamos incurrir en el sobreajuste de datos de entrenamiento



SOBREAJUSTE Y DESAJUSTE

RAZONES Y SOLUCIONES



¿Qué es mejor, desajustar o sobreajustar a los datos de entrenamiento?

Desajuste a los datos de entrenamiento

- ¿Por qué?
 - El modelo es demasiado simple como para aprender la estructura de datos subyacente
 - Variables independientes tienen mucho ruido
 - El modelo comete muchos errores
- ¿Cómo solucionarlo?
 - Seleccionar un modelo más robusto, con más parámetros a optimizar
 - Incluir mejores variables (ingeniería de datos)
 - Reducir restricciones en el modelo (reducir el parámetro del término regularizador)

Sobreajuste a los datos de entrenamiento

- ¿Por qué?
 - El modelo es demasiado complejo, tal que ajusta la estructura de datos y el error subyacente
 - Existe mucha varianza entre las predicciones en el set de entrenamiento y el set de validación
- ¿Cómo solucionarlo?
 - Simplificar el modelo, al imponer restricciones al número de parámetros a optimizar
 - Incrementar el número de observaciones en el set de datos de entrenamiento
 - Reducir el ruido en el set de datos de entrenamiento (arreglar errores en los datos, tratamiento de outliers)
 - Agregar un término de regularización

SOBREAJUSTE Y DESAJUSTE

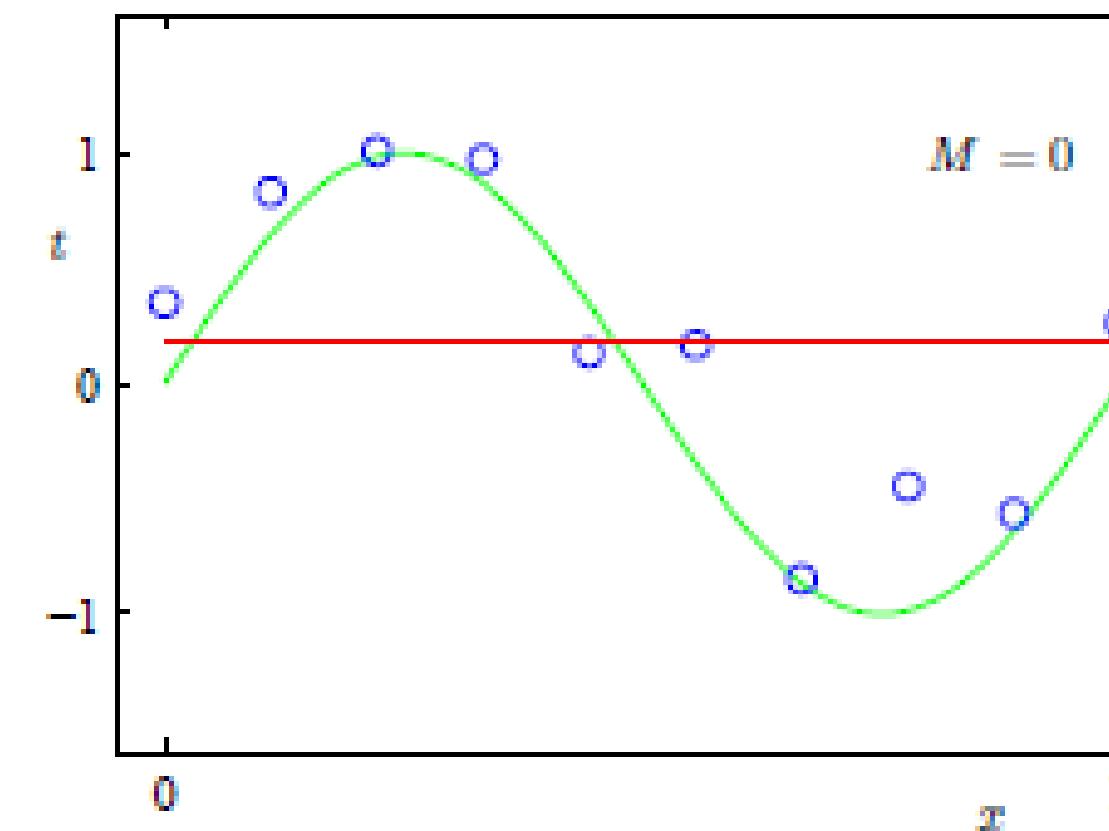
EJEMPLOS



¿Qué pasará si utilizamos un modelo constante o uno lineal para predecir la variable target en un nuevo set de datos?

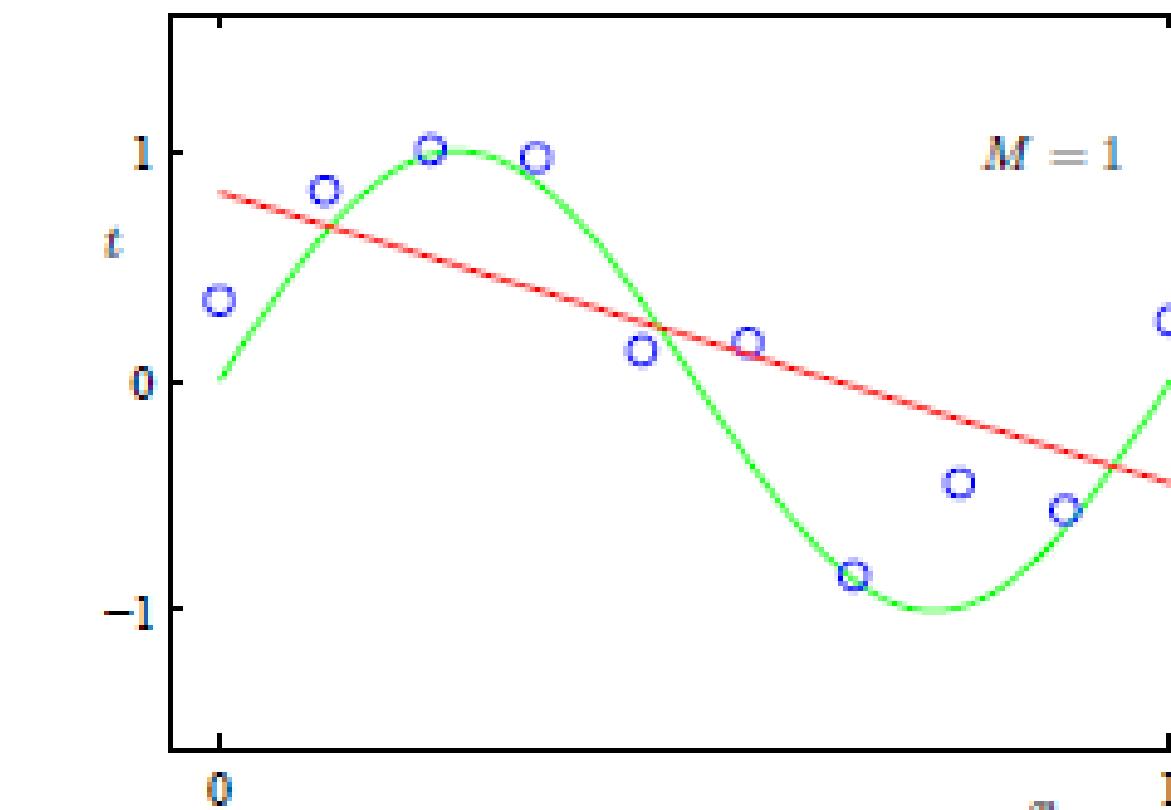
Desajuste a los datos de entrenamiento

- Modelo es una constante, presenta un gran error



Desajuste a los datos de entrenamiento

- Modelo es un poco más complejo que una constante. Sin embargo sigue presentando un gran error



SOBREAJUSTE Y DESAJUSTE

EJEMPLOS

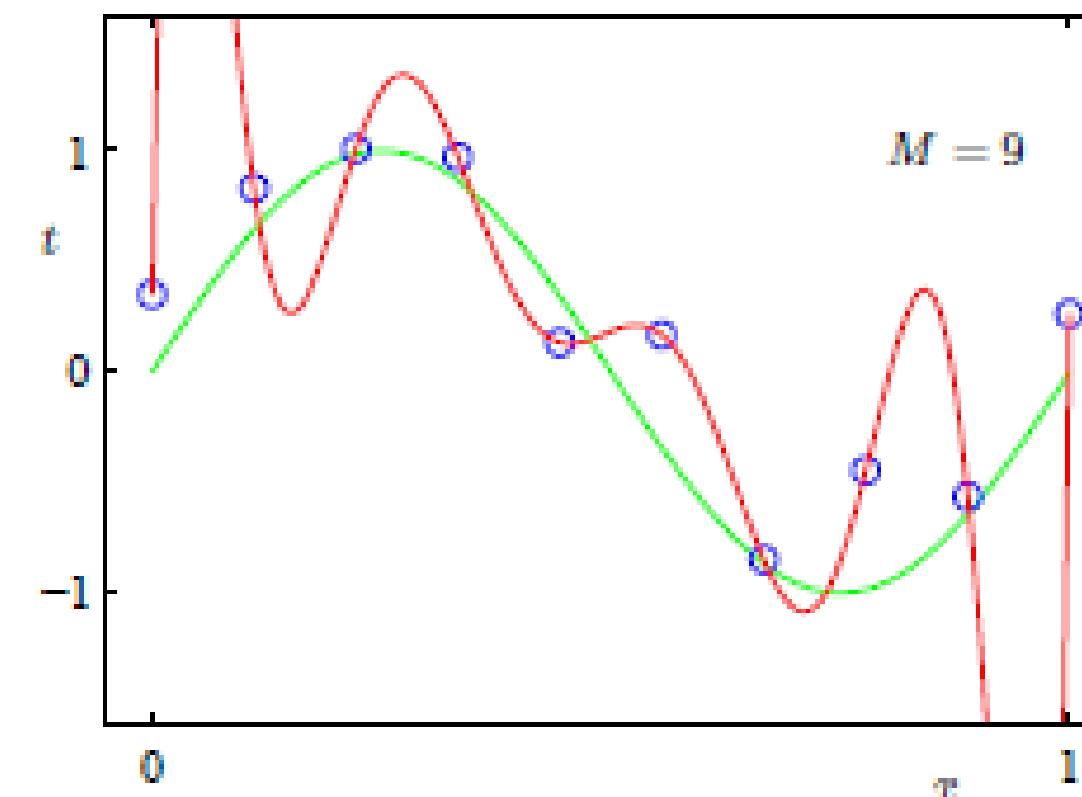


¿Qué pasará si utilizamos un modelo con un set de datos extendido (polinomio grado 9) para predecir la variable target en un nuevo set de datos?



Sobreajuste a los datos de entrenamiento

- Modelo es muy complejo, ajusta a cada uno de los datos de entrenamiento



SOBREAJUSTE Y DESAJUSTE

PREGUNTAS



¿Qué es mejor, desajustar o sobreajustar a los datos de entrenamiento?

→ Ninguno! ambos fenómenos son malos para los modelos. Recordemos que nuestro objetivo es minimizar la función de costo (error), por lo tanto un punto intermedio será lo ideal

¿Qué pasará si utilizamos un modelo constante o uno lineal para predecir la variable target en un nuevo set de datos?

→ Tendremos un error promedio elevado en el set de entrenamiento, debido a que el modelo no logró aprender la distribución de datos subyacente

¿Qué pasará si utilizamos un modelo con un set de datos extendido (polinomio grado 9) para predecir la variable target en un nuevo set de datos?

→ Tendremos un error promedio elevado en el set de validación, debido a que el modelo memorizó los datos de entrenamiento, al cambiar dichos datos, el modelo va a cometer muchos errores

SOBREAJUSTE Y DESAJUSTE

ZOOM EN LAS CAUSAS DEL ERROR



Para entender cómo mitigar el desajuste y sobreajuste de datos, es necesario entender las diferentes fuentes de error que un modelo puede tener

Descomposición del error

$$MSE = \mathbb{E}[y - \hat{f}_{S_{Train}}(\mathbf{X}_n)^2]$$

$$= \mathbb{E}[f(\mathbf{X}_n) + \epsilon - \hat{f}_{S_{Train}}(\mathbf{X}_n)^2]$$

Después de varios pasos....

$$= Var[f(\mathbf{X}_n) - \hat{f}_{S_{Train}}(\mathbf{X}_n)] + Var(\epsilon) + [\mathbb{E}[f(\mathbf{X}_n)] - \mathbb{E}[\hat{f}_{S_{Train}}(\mathbf{X}_n)]]^2$$



Varianza

Dispersión de las predicciones

Ruido

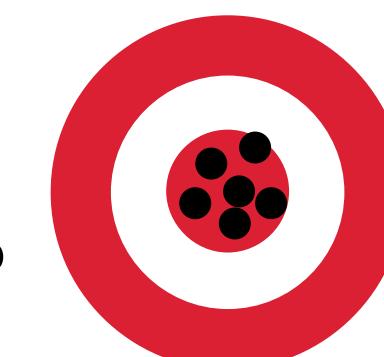
No está en nuestro control

Sesgo

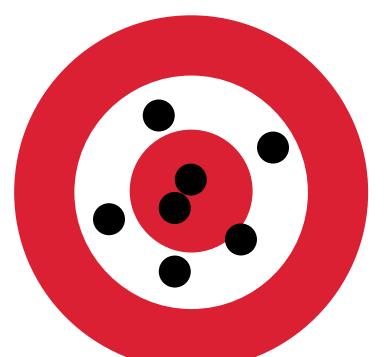
Error entre la predicción promedio y la distribución real

¿Cuál crees que sea el escenario ideal?

Baja varianza



Bajo sesgo



**Alto
sesgo**

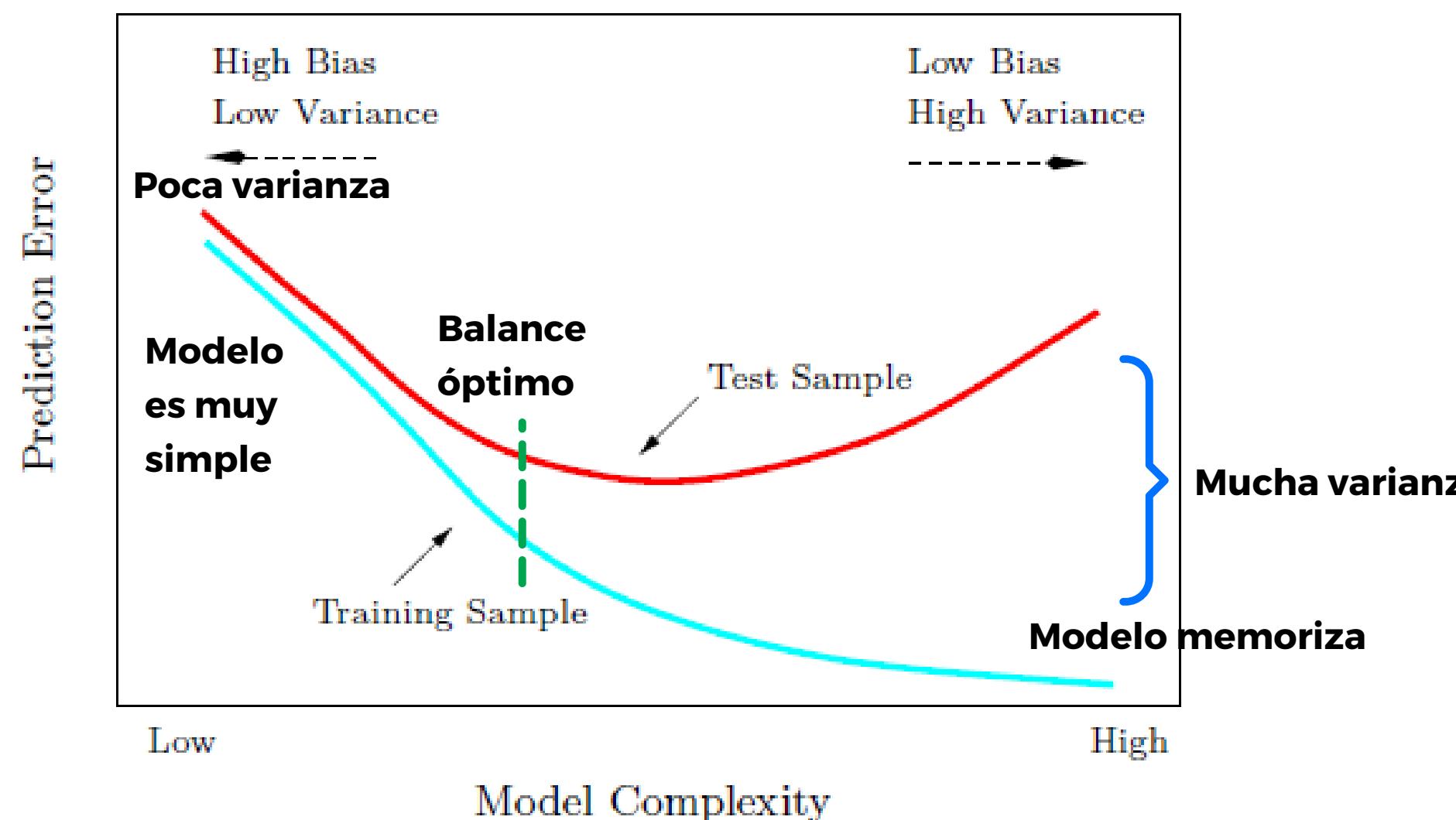
SOBREAJUSTE Y DESAJUSTE

BIAS-VARIANCE TRADE-OFF



Queremos minimizar el error, por lo tanto, tenemos que encontrar un balance entre sesgo y varianza. Lo anterior sucede cuando por cada incremento en el sesgo, se tiene un reducción en la varianza, y viceversa

Optimización del trade-off entre el sesgo y la varianza



- Si logramos llegar al balance óptimo entre las dos principales fuentes de error, lograremos encontrar el modelo que ayude a generalizar a futuros sets de datos.
Aka: Nos ayude a predecir de manera más eficiente y correcta

SOBREAJUSTE Y DESAJUSTE

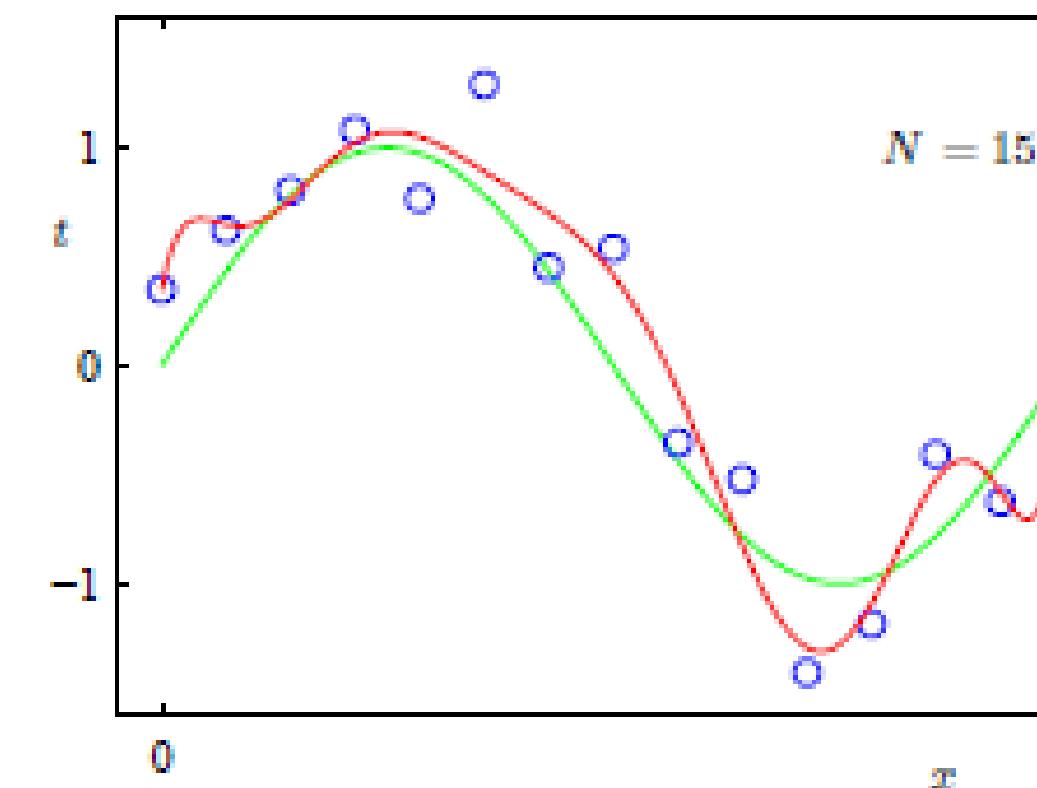
UNA FORMA DE MITIGAR EL SOBREAJUSTE DE DATOS



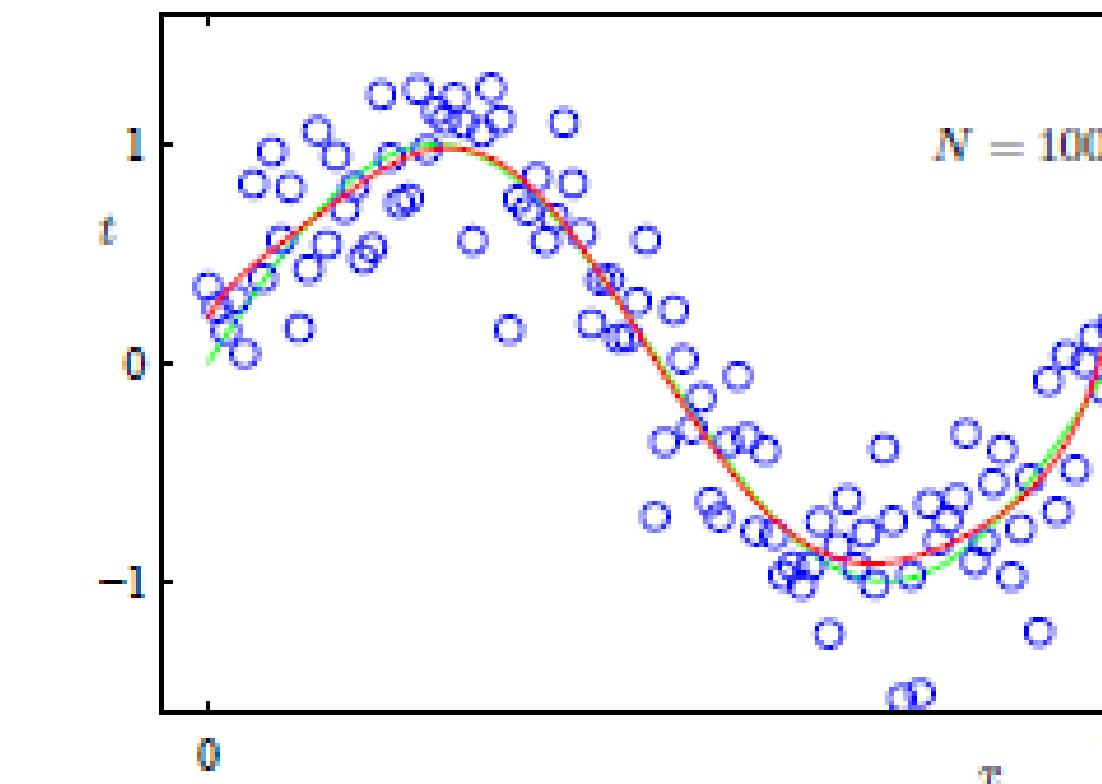
Entre más datos para entrenar tengamos, el modelo podrá generalizar de mejor manera

Aumentar el número de datos de entrenamiento para reducir el sobreajuste de datos

- El modelo intenta ajustar cada uno de los datos de entrenamiento



- Entre más datos de entrenamiento tengamos, el modelo tendrá más dificultad para sobreajustar a los datos de entrenamiento



SOBREAJUSTE Y DESAJUSTE

OTRA FORMA DE MITIGAR EL SOBREAJUSTE DE DATOS



Agregar un término de regularización a la función de costo



Término de regularización para mitigar el sobreajuste de datos

- El término Omega nos ayuda a regularizar la función de costo, penalizando a los modelos complejos y favoreciendo a los simples

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) + \underbrace{\Omega(\mathbf{W})}_{\text{Término regularizador}}$$

Término de regularización comunes

- Norma L2
 - Parámetros grandes para \mathbf{W} serán penalizados, mientras que los pequeños estarán ok
- Norma L1
 - La solución óptima para \mathbf{W} será poco densa (sparse)

$$\Omega(\mathbf{W}) = \lambda \|\mathbf{W}\|_2^2$$

$$\Omega(\mathbf{W}) = \lambda \|\mathbf{W}\|_1$$

SOBREAJUSTE Y DESAJUSTE

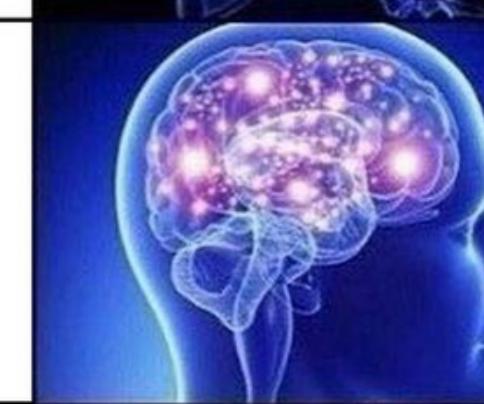
FORMAS DE EVALUAR SI NUESTRO MODELO SUFRE DE SOBREAJUSTE



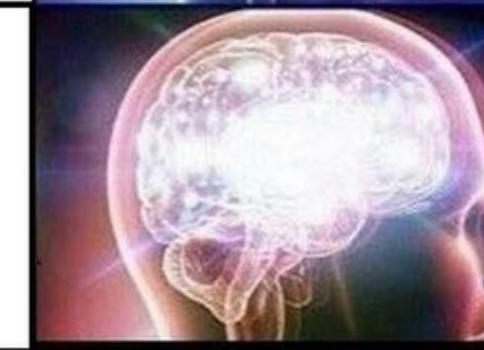
Entrenar y evaluar el modelo en un mismo set de datos de entrenamiento



Partir el set de datos original en train y test. Entrenar en el set de entrenamiento y evaluar en el de validación



Partir el set de datos original en train y test. Entrenar en train y evaluar en test. Obtener un set de datos independiente (holdout) y hacer una evaluación extra



Utilizar K-Fold Cross-Validation para determinar los parámetros óptimos para el término de regularización que nos ayude a mitigar el sobreajuste



K-Fold Cross-Validation nos ayuda a obtener un estimación insesgada de la generalización del error y de su varianza. Por lo tanto, nos ayuda a ver qué tan bien o qué tan mal nuestro modelo generalizará a nuevos sets de datos

K-FOLD CROSS-VALIDATION

FORMAS DE EVALUAR SI NUESTRO MODELO SUFRE DE SOBREAJUSTE



Divide el set de datos K veces, tal que todos los datos sirvan para entrenar y evaluar un modelo de manera distinta cada vez. Al final del ciclo iterativo, el promedio de las métricas obtenidas en cada iteración será la generalización del performance del modelo (error, otras métricas). También puedes calcular la desviación estándar de las métricas, para determinar que tan estable es el modelo

El mismo set de datos será utilizado 5 veces por separado

