Maestría en Inteligencia Artificial Aplicada

Curvas de Aprendizaje

Sesgo – Varianza – Regularización

(*Bias – Variance – Regularization*)

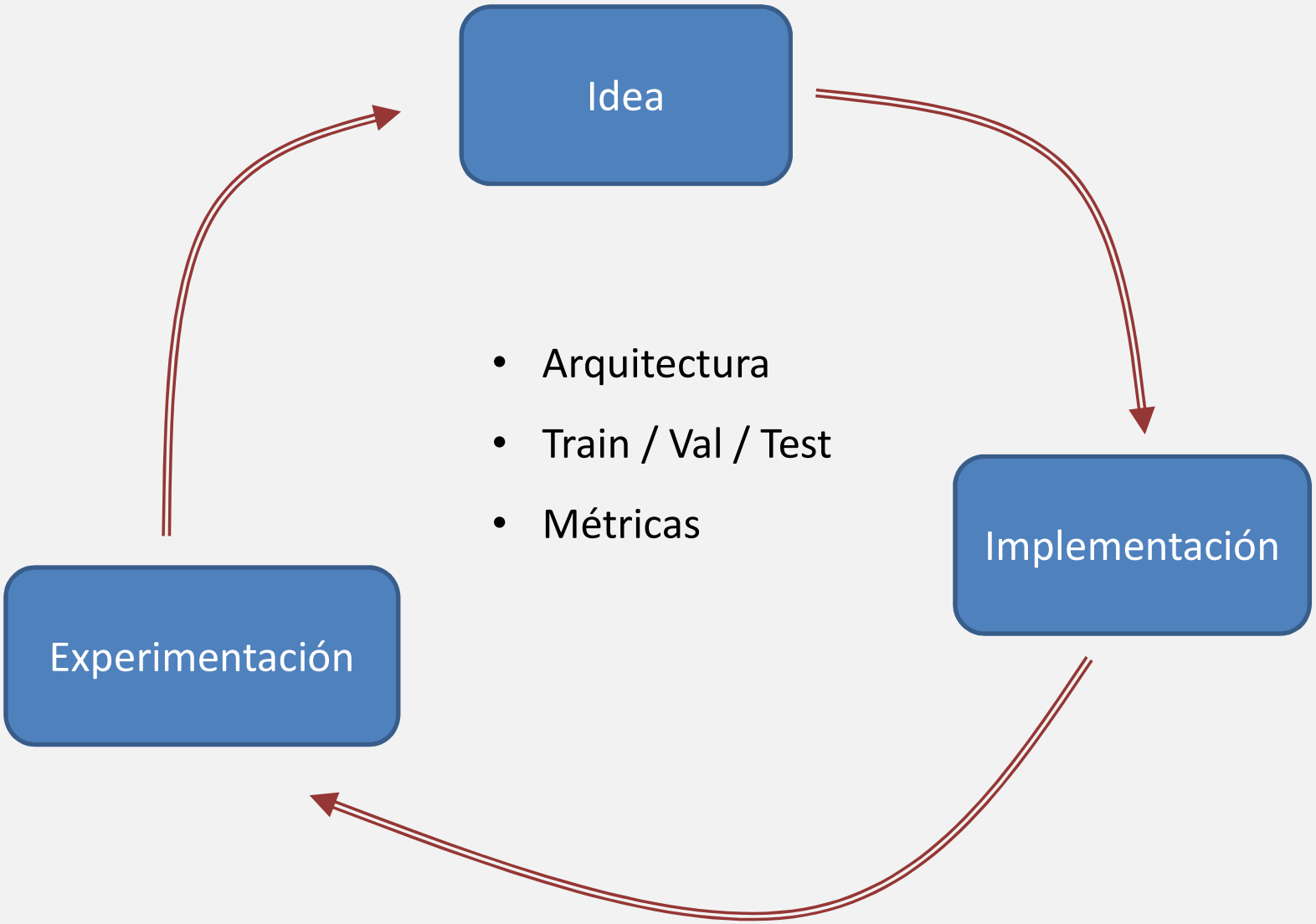
Dr. Luis Eduardo Falcón Morales
ITESM
Campus Guadalajara

Aprendizaje Supervisado:
Partición en datos de Entrenamiento, Validación y Prueba

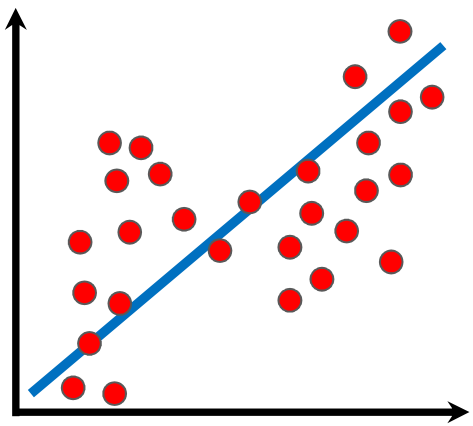
- **Datos de Entrenamiento:** para obtener los parámetros del modelo.
- **Datos de Validación:** para realizar el ajuste de los parámetros y seleccionar el mejor de los modelos.
- **Datos de Prueba:** para evaluar el desempeño del modelo.

En ocasiones la partición del conjunto original de datos se hace en los conjuntos de Entrenamiento y Prueba, porque en el conjunto de Entrenamiento se aplica Validación-Cruzada y ahí se divide a su vez el conjunto en entrenamiento y validación.

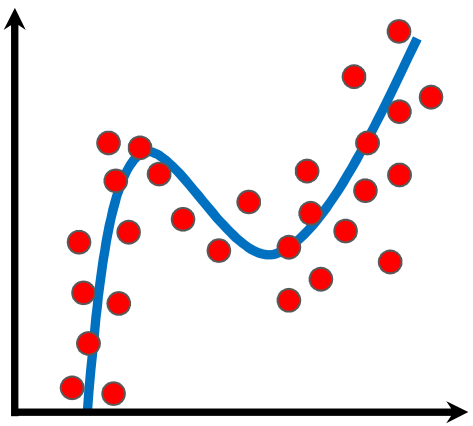
Proceso de construcción de un sistema de Aprendizaje Automático



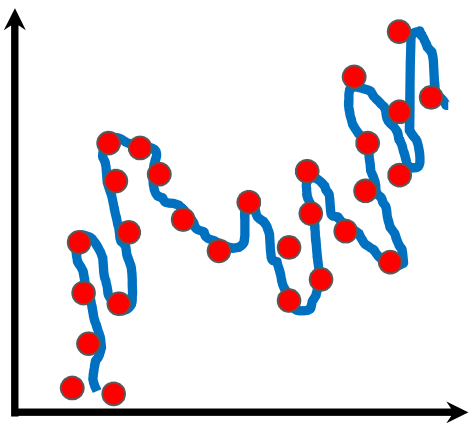
Segso vs Varianza / Subentrenado vs Sobreentrenado



Modelo
Sub-entrenado

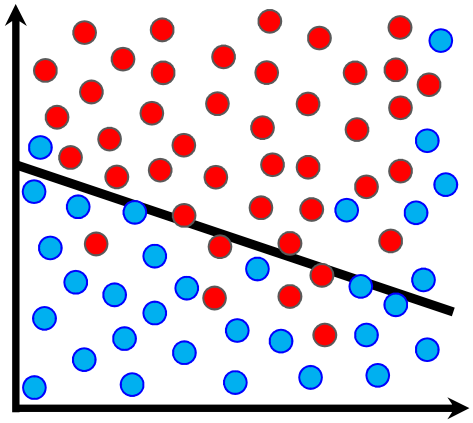


Modelo
Óptimo

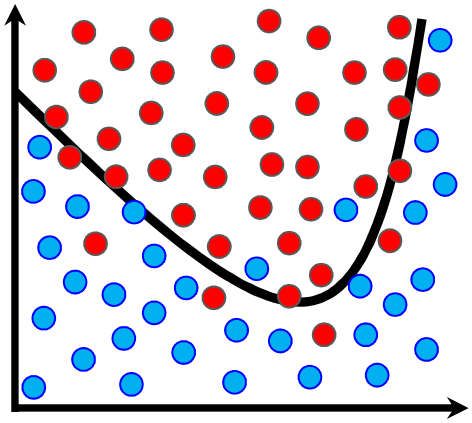


Modelo
Sobre-entrenado

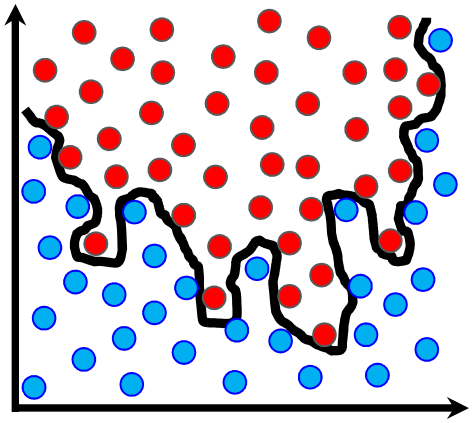
Problema de
Regresión



Error de sesgo alto



Sesgo y varianza
balanceados



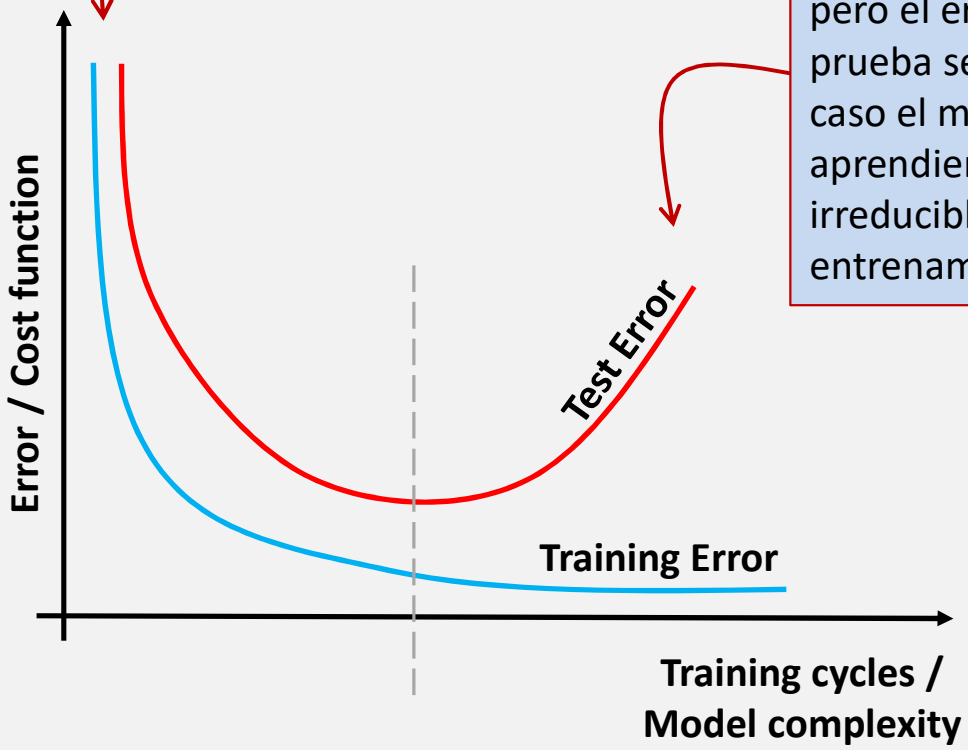
Error de varianza alta

Problema de
Clasificación

Sub-entrenamiento vs Sobre-entrenamiento

Sub-entrenamiento: tanto el error de los datos de entrenamiento como los de prueba son grandes.

Sobre-entrenamiento: el error de los datos de entrenamiento se mantiene disminuyendo, pero el error de los datos de prueba se incrementa. En este caso el modelo está aprendiendo inclusive el ruido irreducible de los datos de entrenamiento.



Underfitting vs Overfitting

Tipos de Errores

Los métodos de aprendizaje automático (machine learning) buscan obtener un modelo o función objetivo f que transforma datos de entrada X en datos de salida Y.

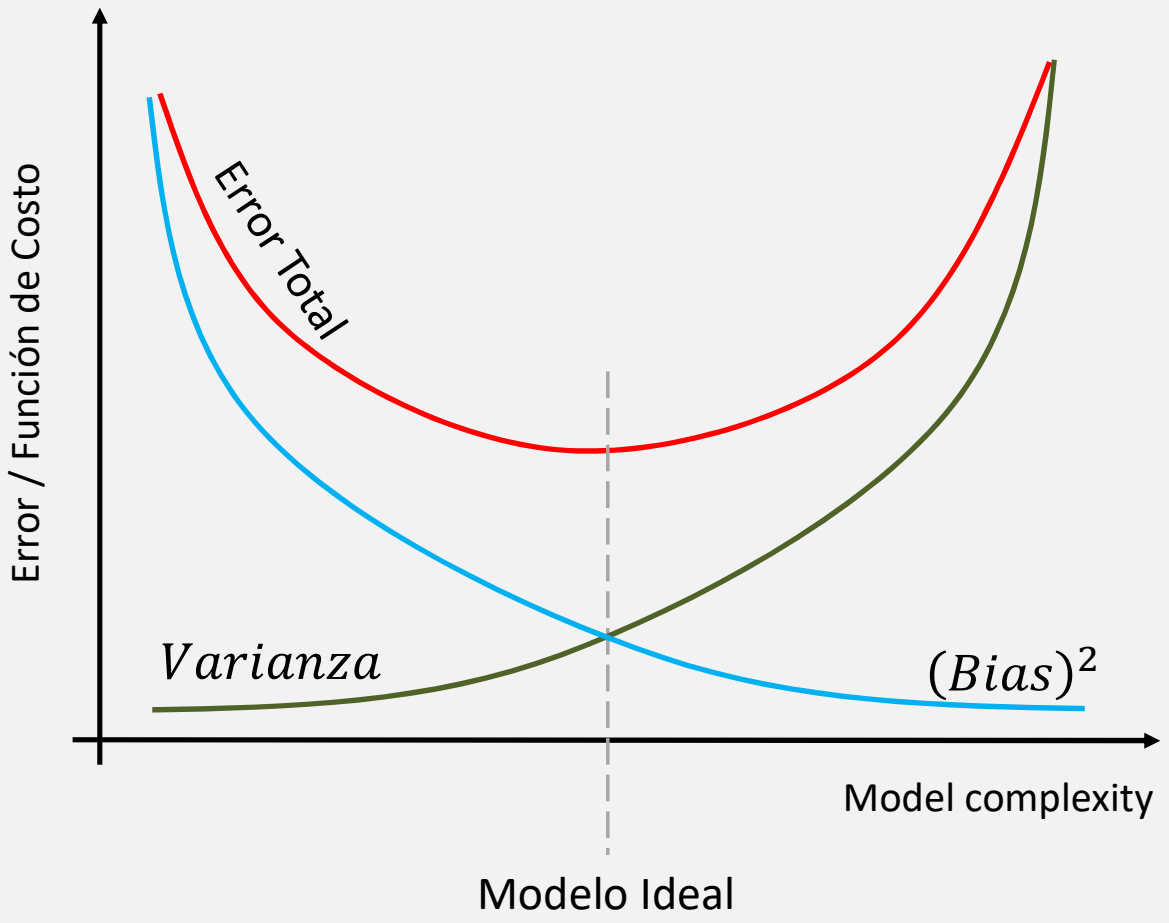
La aproximación del modelo se lleva a cabo minimizando el error total de aproximación:

$$Y = f(X) + error_{Total}(X)$$

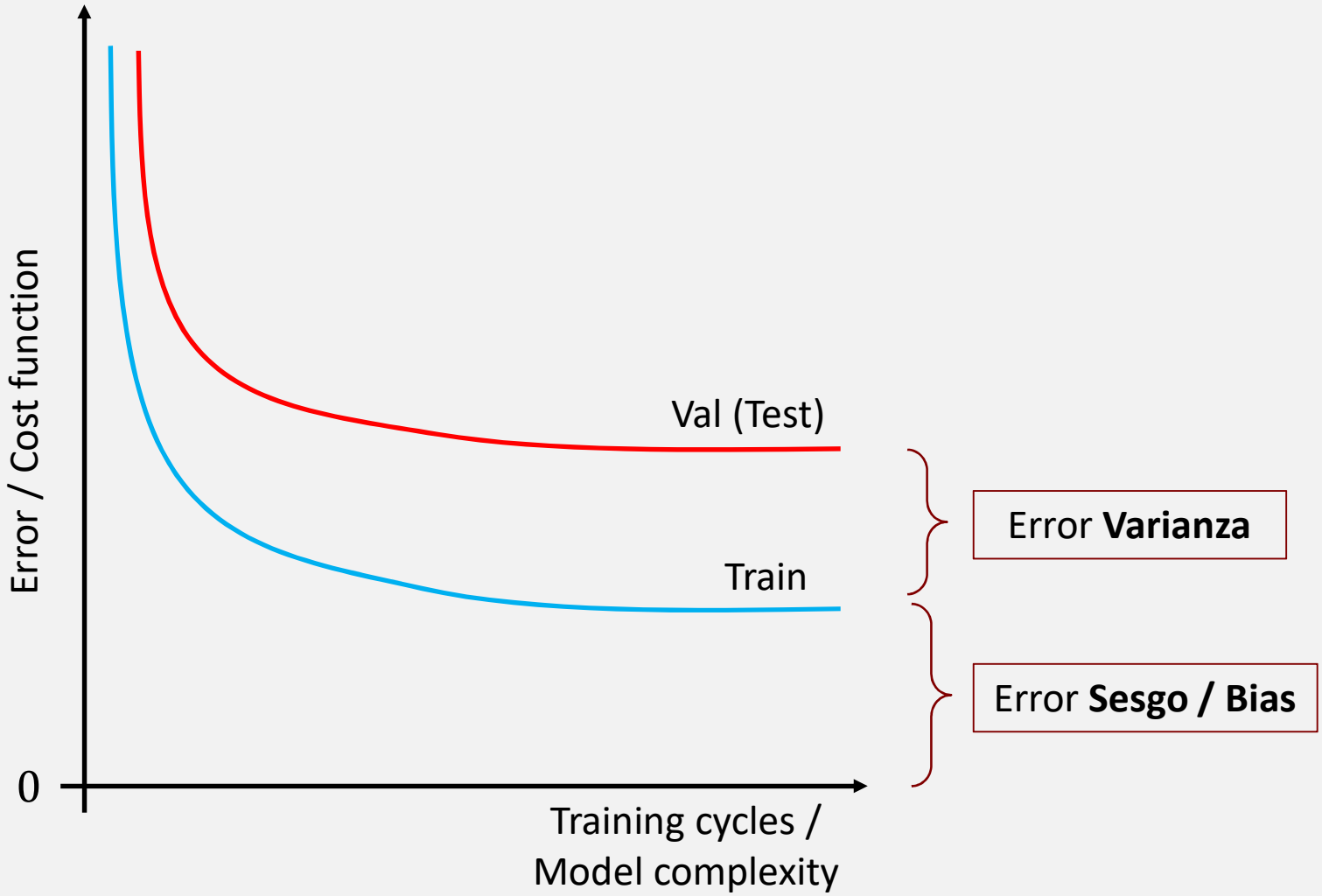
donde el error total del modelo de aprendizaje lo podemos descomponer como sigue:

$$error_{Total}(X) = \underbrace{\left(E\left[\hat{f}(X)\right] - f(X)\right)^2}_{\text{Error debido al } \textbf{Sesgo}} + \underbrace{E\left[\left(\hat{f}(X) - E\left[\hat{f}(X)\right]\right)^2\right]}_{\text{Error debido a la } \textbf{Varianza}} + \underbrace{error_{noise}}_{\text{Error } \textbf{irreducible} \text{ o ruido}}$$

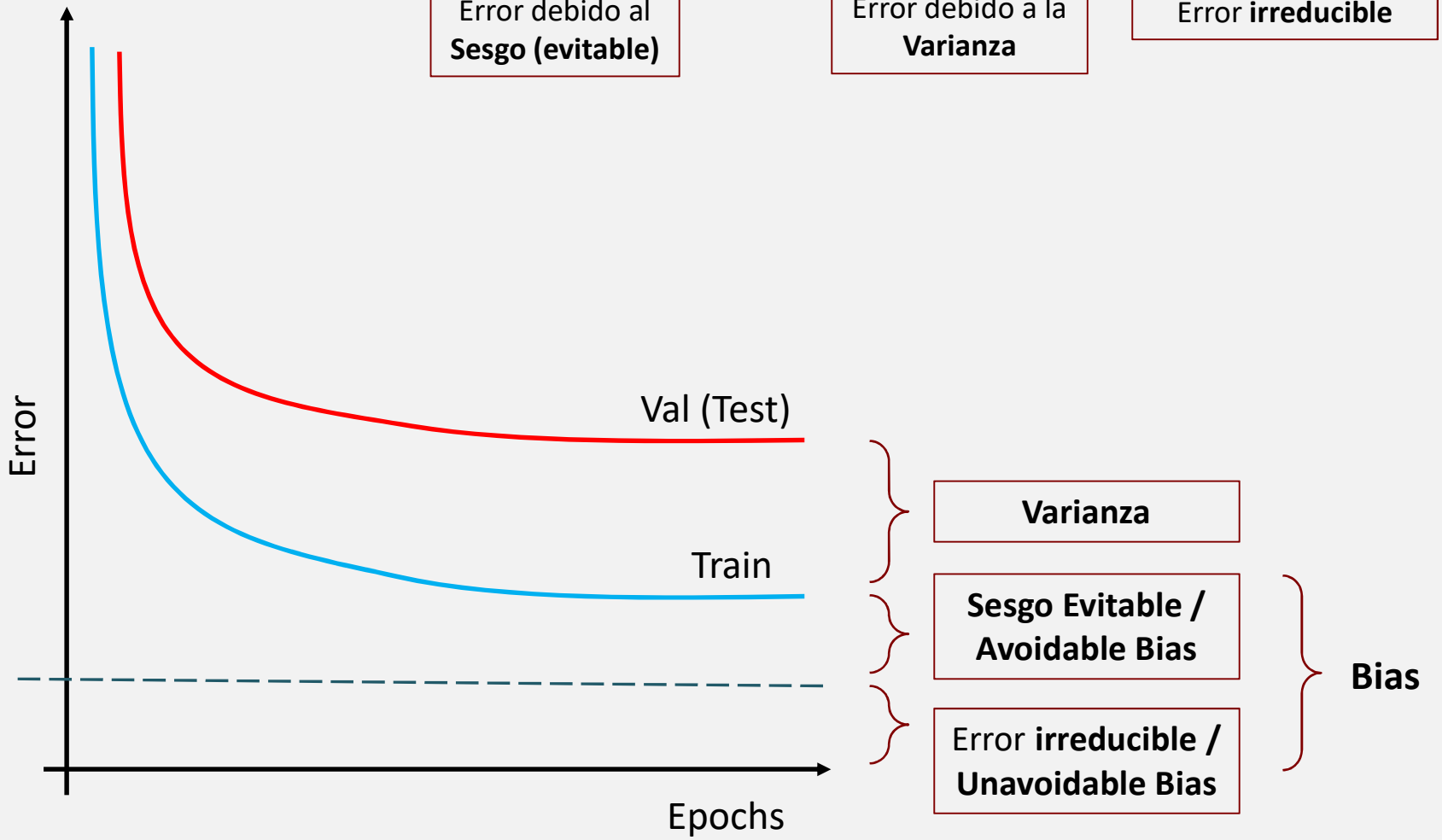
Sub-entrenamiento vs Sobre-entrenamiento



Análisis de los Errores: Sesgo y Varianza

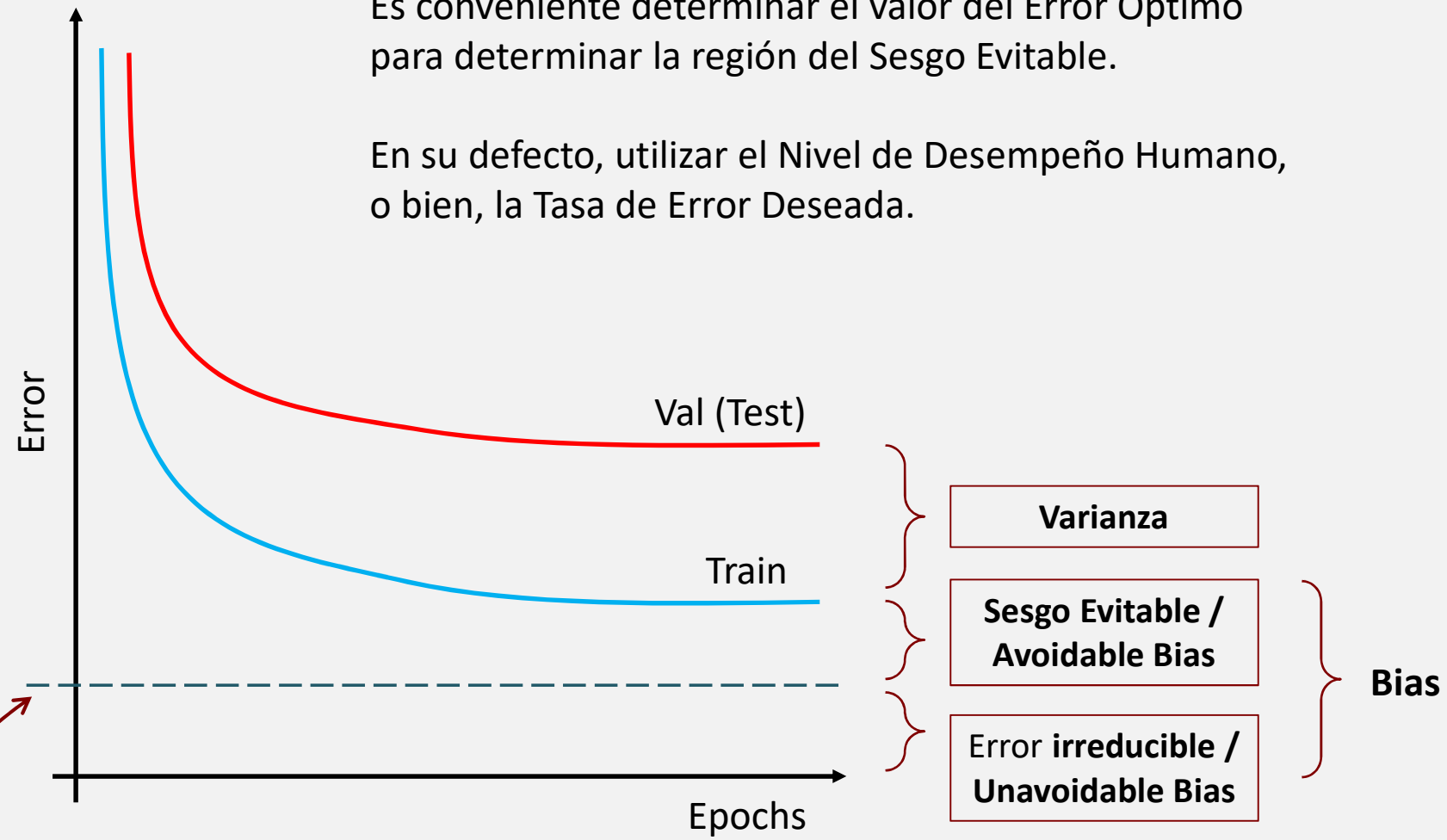


$$error_{Total}(X) = \underbrace{\left(E\left[\hat{f}(X)\right] - f(X)\right)^2}_{\text{Error debido al Sesgo (evitable)}} + \underbrace{E\left[\left(\hat{f}(X) - E\left[\hat{f}(X)\right]\right)^2\right]}_{\text{Error debido a la Varianza}} + \underbrace{error_{noise}}_{\text{Error irreducible}}$$



Es conveniente determinar el valor del Error Óptimo para determinar la región del Sesgo Evitable.

En su defecto, utilizar el Nivel de Desempeño Humano, o bien, la Tasa de Error Deseada.



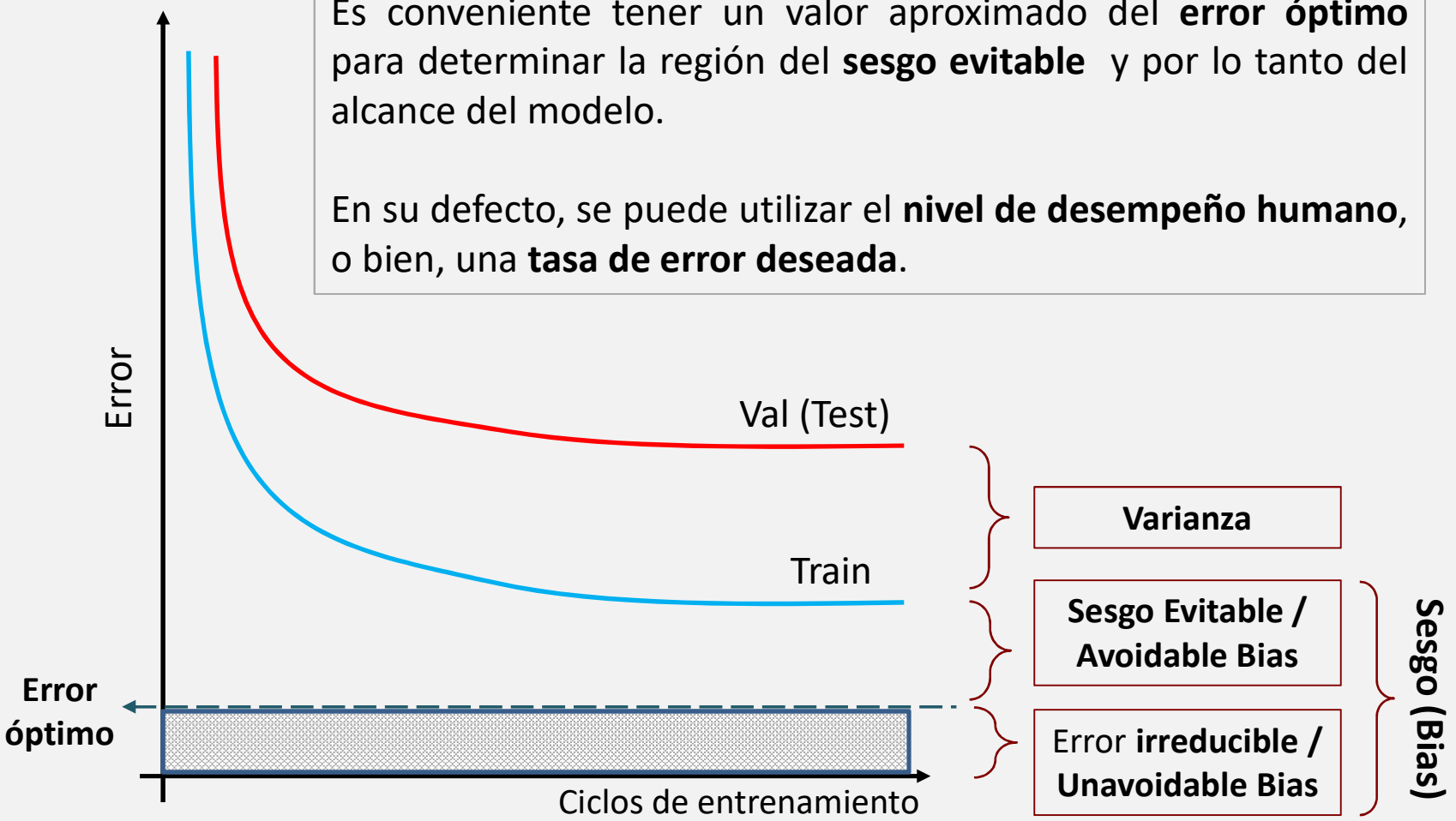
Optimal Error / Bayes Error rate

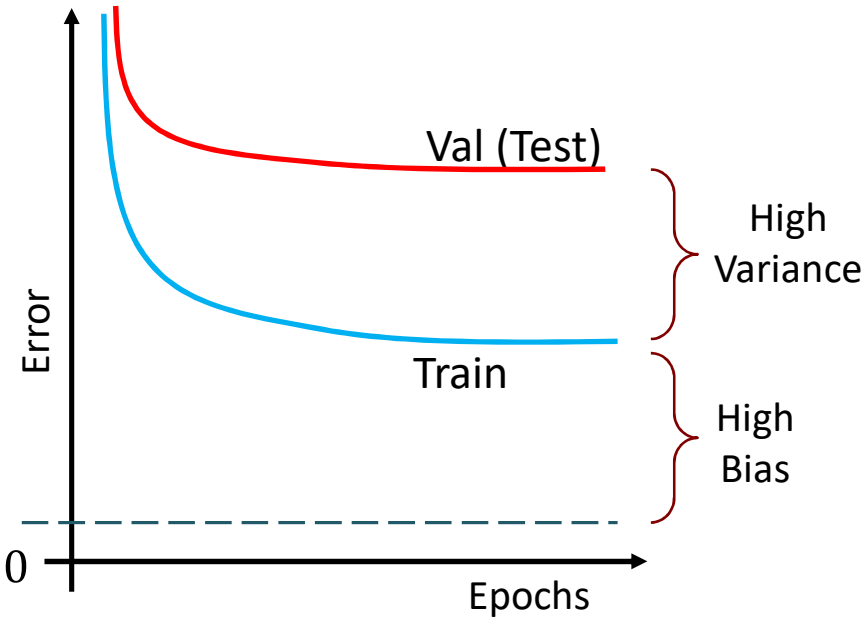
O bien, sustituirlo por:

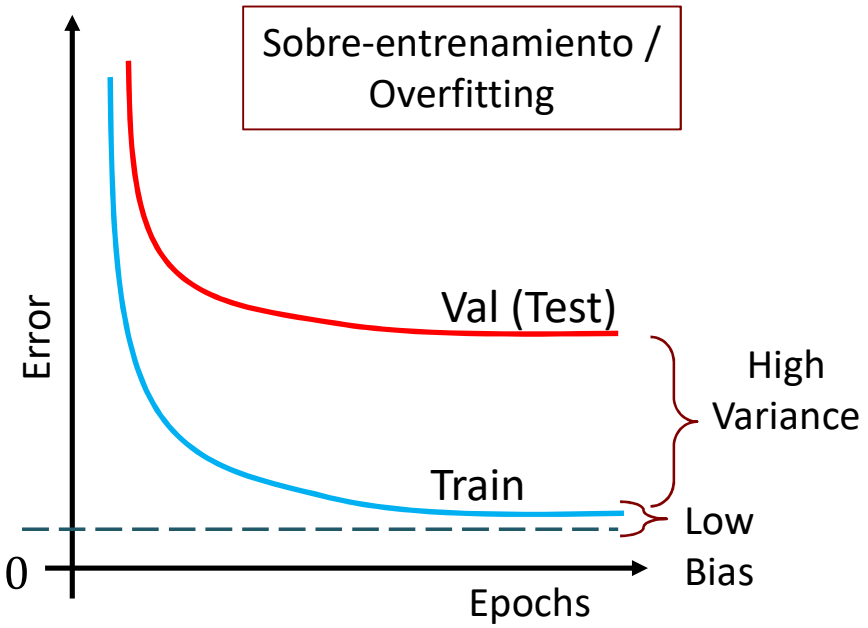
Human Level Performance / Desired Error rate

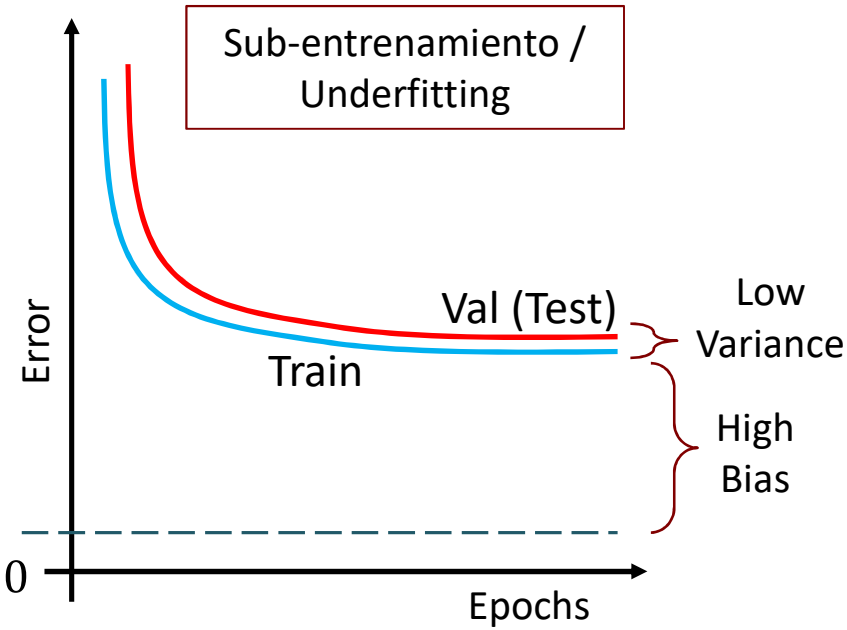
Es conveniente tener un valor aproximado del **error óptimo** para determinar la región del **sesgo evitable** y por lo tanto del alcance del modelo.

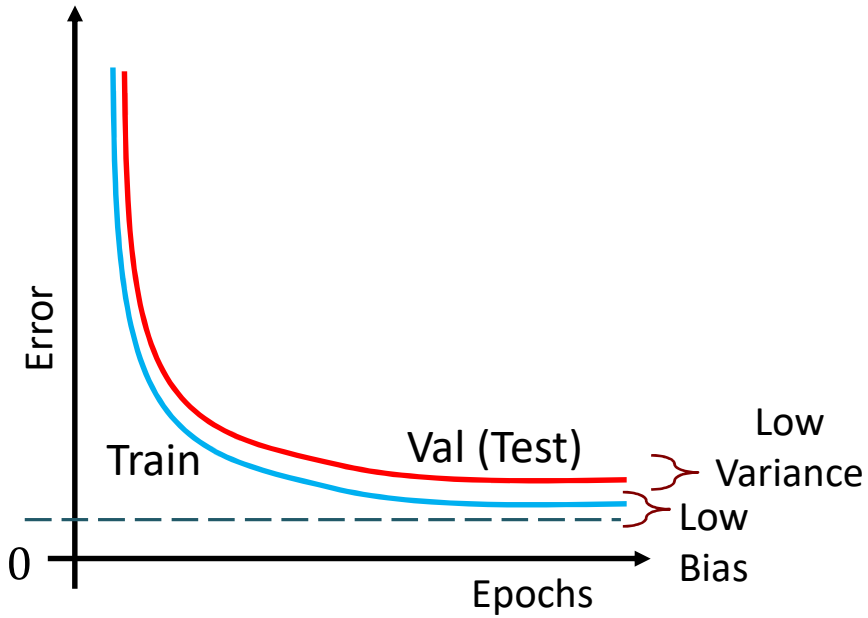
En su defecto, se puede utilizar el **nivel de desempeño humano**, o bien, una **tasa de error deseada**.

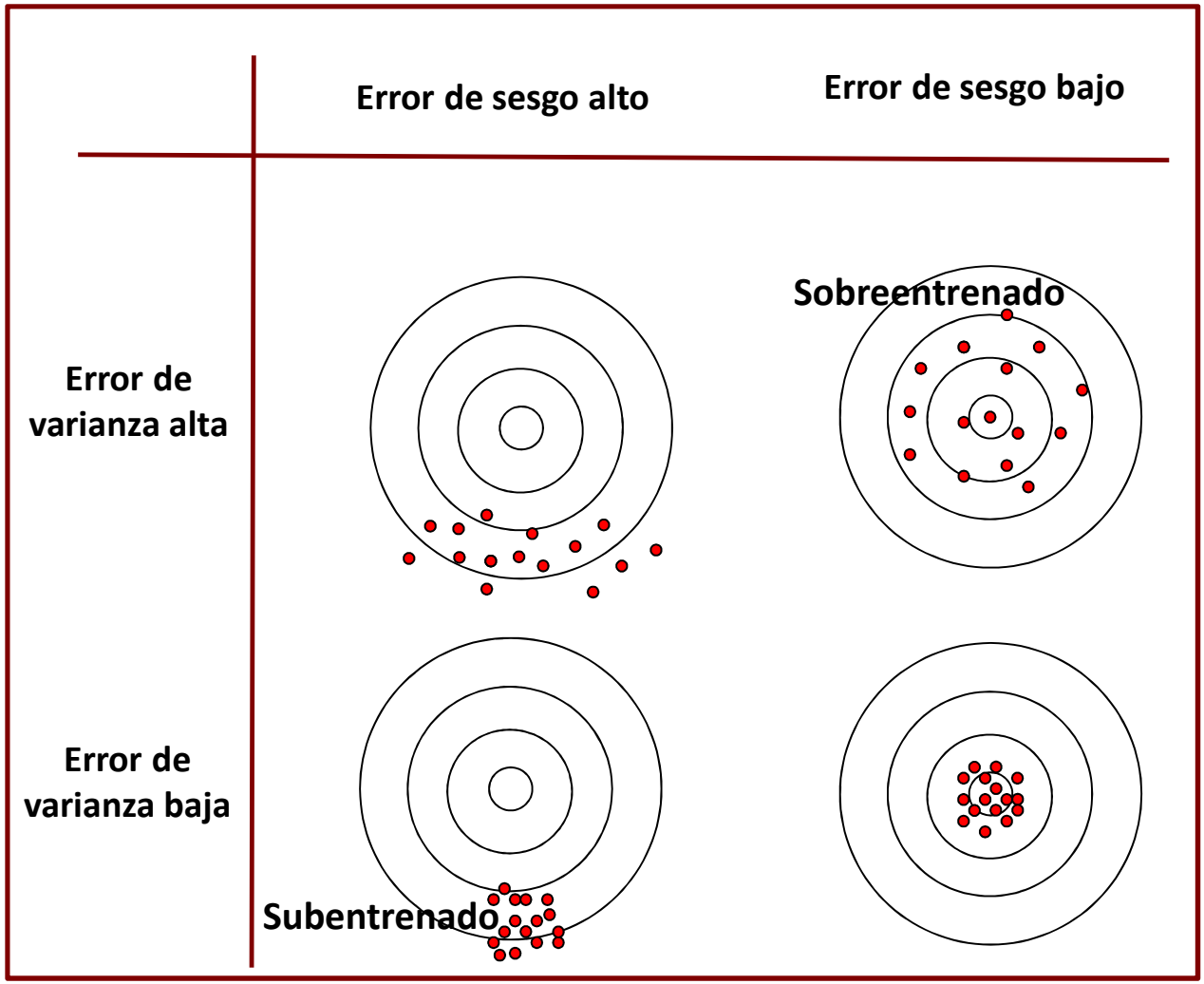



Val (Test)
Train
High
Variance
High
Bias
Error
0
Epochs

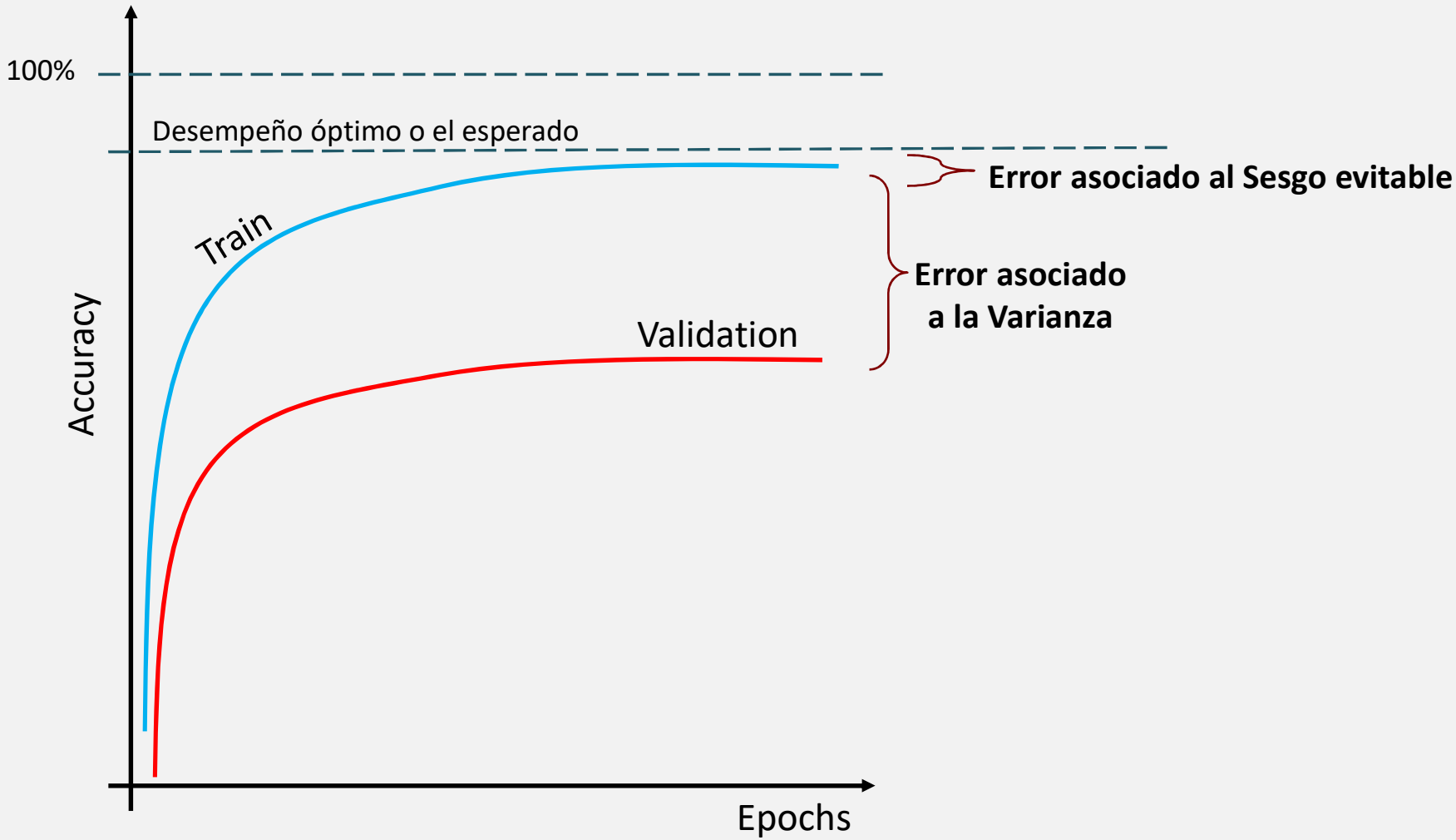

Sobre-entrenamiento /
Overfitting
Val (Test)
Train
High
Variance
Low
Bias
Error
0
Epochs


Sub-entrenamiento /
Underfitting
Val (Test)
Train
Low
Variance
High
Bias
Error
0
Epochs

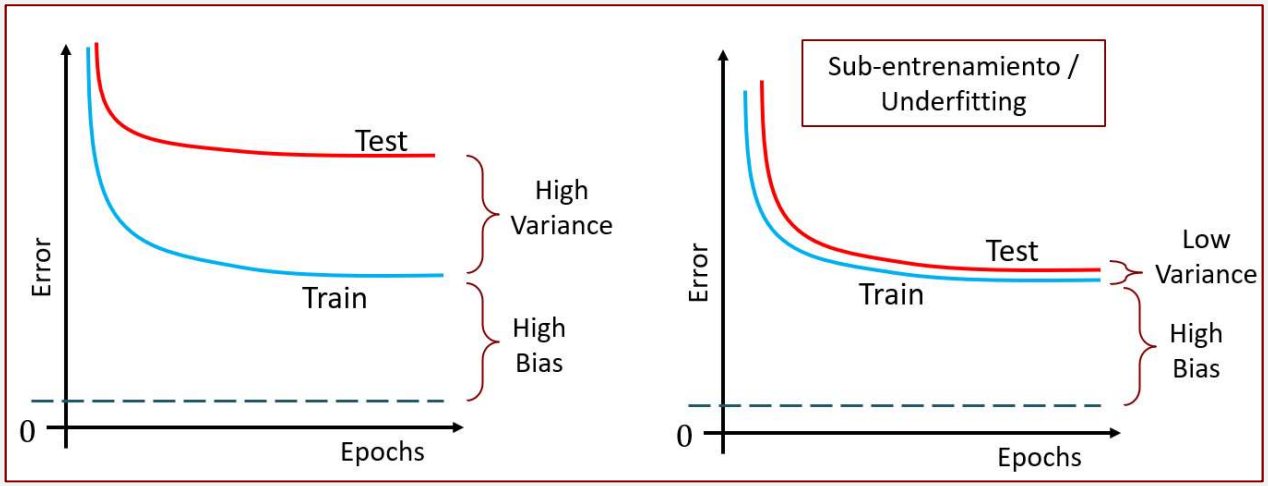

Train
Val (Test)
Low
Variance
Low
Bias
Error
0
Epochs


Error de sesgo alto
Error de sesgo bajo
Error de varianza alta
Error de varianza baja
Sobreentrenado
Subentrenado

Conclusiones análogas se obtienen analizando las gráficas de desempeño (accuracy) de los conjuntos de entrenamiento y validación. Por ejemplo, el caso de *overfitting* se podría ver como sigue:



Cómo enfrentar el problema del **Sesgo Alto** (*High Bias*)

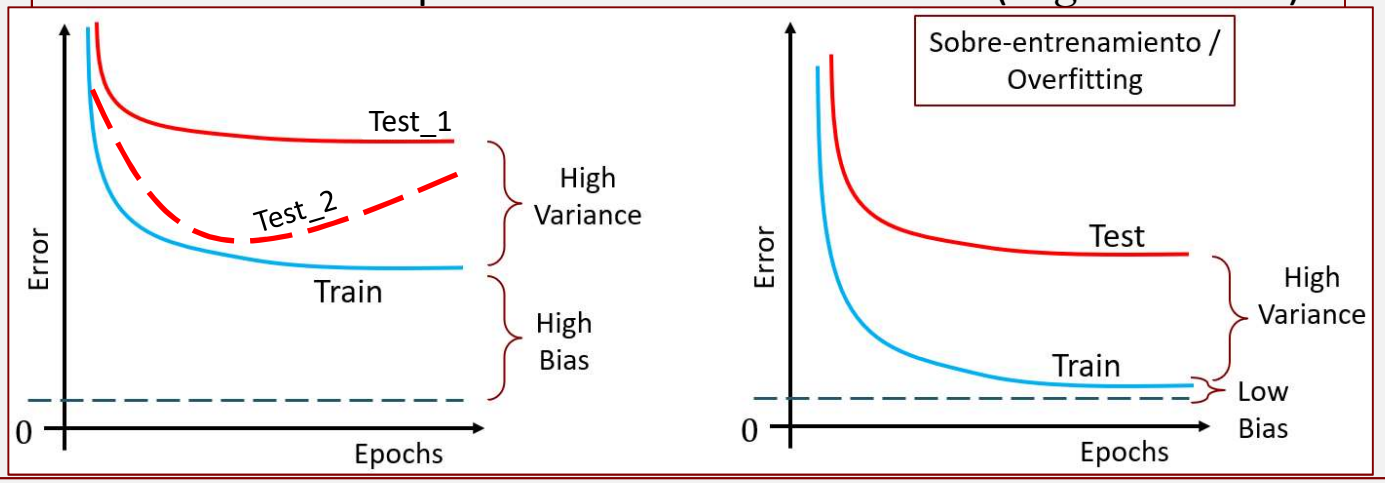


- Incrementar el tamaño del modelo: más capas, más neuronas, mayor arquitectura, otro modelo más complejo.
- Agregar más variables (*features*) apoyándose en un análisis de errores del conjunto de Validación.
  - Incluir además regularización (ℓ_2, ℓ_1, Transformación de datos) si al incrementar la complejidad del modelo aumenta la varianza.
- Aplicar un *learning rate* dinámico que vaya disminuyendo con el número de épocas.

Toma en cuenta que varios de estos casos nos lleva al dilema del Sesgo y la Varianza (*Bias-Variance tradeoff*) y habrá que estarse moviendo entre los casos del sesgo alto y varianza alta.

Estos ajustes requerirán en general una mayor capacidad de cómputo.

Cómo enfrentar el problema de la **Varianza Alta** (*High Variance*)



Bias-Variance tradeoff

- Incluir regularización (ℓ_2, ℓ_1, *Transformación de los datos*).
- Realizar un análisis de errores en el conjunto de validación para detectar y atacar los errores más comunes:
  - Simplificar el modelo quitando variables: en la perspectiva actual de DL se dejan todas y que el modelo decida su importancia (no en ML que no se tienen muchos datos).
  - O bien, agregar nuevas variables si se detecta que ayudarán a disminuir los tipos de errores detectados (esto también puede disminuir el Bias).
- Detener el entrenamiento antes de que la varianza se empiece a incrementar (*early stopping*).
- Simplificar el modelo disminuyendo las capas y el número de neuronas. No recomendable en DL (Deep Learning).
- Disminuir el tamaño de paso (learning rate) si la gráfica tiene muchas oscilaciones muy bruscas, e incrementar el número de épocas (epochs).
- Ajustar los conjuntos de entrenamiento y prueba para que sean más parecidos.
- Agregar más datos al conjunto de entrenamiento (siempre recomendable).

Regularización L1 / L2

Recordemos que se desea minimizar una función de costo J, dado un conjunto de N datos muestrales, para obtener los pesos β_j de un modelo $\hat{y}(\beta_0, \beta_1, \dots, \beta_m)$, a partir de un conjunto de datos de entrenamiento X, Y.

En el caso de que un modelo use la suma de los cuadrados de los errores como función de costo, tendríamos:

Penalización L_1 (lasso):

$$J_{costo\_L1} = \sum_{k=1}^{N} (y_k - \hat{y}_k)^2 + \lambda \sum_{j=1}^{m} |\beta_j|$$

Penalización L_2 (ridge):

$$J_{costo\_L2} = \sum_{k=1}^{N} (y_k - \hat{y}_k)^2 + \lambda \sum_{j=1}^{m} \beta_j^2$$

Observa que no se incluye el término constante β_0 en las penalizaciones.

Penalización de red elástica (L_1 y L_2):

$$J_{costo\_elastic} = \sum_{k=1}^{N} (y_k - \hat{y}_k)^2 + \lambda \sum_{j=1}^{m} |\beta_j| + \eta \sum_{j=1}^{m} \beta_j^2$$

Constantes de penalización: $\lambda > 0, \eta > 0$