# Ciencia de Datos Aplicada a Ciencias de la Tierra

## Tema: Métodos de aprendizaje supervisado

Martín A. Díaz-Viera[1], Farhid M. Elisea Guerrero[2]

[1)] *mdiazv@imp.mx*, [2)] *felisea@imp.mx*

20 de marzo de 2024

# Contenido I

# Contenido II

# Introducción al aprendizaje supervisado

## Introducción al Aprendizaje Supervisado

# Aprendizaje Automatizado (ML)

- Algunos de los métodos más comunes implementados para "hacer que las máquinas aprendan"son:
  - Aprendizaje no supervisado
  - Aprendizaje supervisado
  - Aprendizaje reforzado

# Características del aprendizaje supervisado

- Es un método de aprendizaje automátizado para problemas en los que los datos están etiquetados.
- El objetivo es obtener una función que asigna vectores de entradas a etiquetas de salida.
- Produce una función inferida, para mapear nuevos ejemplos.
- La calidad estadística se mide mediante el error de generalización.

# Generalización del aprendizaje supervisado

Hay varias formas de generalizar el problema de aprendizaje supervisado:

- Aprendizaje semisupervisado: los valores de salida deseados se dan solo para un subconjunto de los datos
- Supervisión débil: se utilizan fuentes ruidosas, limitadas o imprecisas para los datos de entrenamiento.
- Aprendizaje activo: se recopilan nuevos ejemplos de forma interactiva, haciendo consultas a un humano.
- Predicción estructurada: cuando el valor de salida es un objeto complejo, como un árbol o un gráfico etiquetado.
- Aprendizaje para ordenar: cuando la entrada es un conjunto de objetos y la salida es un ordenamiento de esos objetos.

# Flujo del aprendizaje supervisado

Para un problema de aprendizaje supervisado, se deben realizar los siguientes pasos:

- Determinar el tipo de ejemplos de entrenamiento.
- Reunir un conjunto de entrenamiento.
- Determinar la representación de características de entrada de la función de aprendizaje.
- Determinar la estructura de la función de aprendizaje y el algoritmo de aprendizaje correspondiente.
- Completar el diseño.
- Evaluar la precisión de la función aprendida.

# Algoritmos de aprendizaje supervisado

Existe una amplia gama de algoritmos, con sus puntos fuertes y débiles. Pero hay cuatro cuestiones principales a considerar:

- Compensación de sesgo-varianza
- Complejidad de la función y cantidad de datos de entrenamiento
- Dimensionalidad del espacio de entrada
- Ruido en los valores de salida

# Clasificación de los métodos de aprendizaje supervisado

Los métodos de aprendizaje supervisado pueden ser del tipo:

- Regresión (Continuos)/Clasificación (Categóricos)
- Lineal/No lineal
- Simple/Ensamblado

# Métodos de aprendizaje supervisado

Según sea el valor de la predicción:

- Clasificación (Categóricos)/Regresión (Continuos)
  - Regresión logística
  - Bayes ingenuo
  - Vecino más cercano k-NN
  - Máquinas de soporte vectorial (SVM)
  - Árboles de decisión y bosque aleatorio
  - Redes neuronales (Perceptrón/Perceptrón multicapa)
- Regresión (Continuos)
  - Regresión lineal (simple o múltiple)
  - Regresión Bayesiana
  - Regresión cuantil
  - Regresión polinomial

# Métodos de aprendizaje supervisado

Según su forma funcional:

- Lineal
  - Regresión lineal (simple o múltiple)
  - Regresión logística
  - Regresión Bayesiana
  - Regresión cuantil
  - Regresión polinomial
  - Redes neuronales (Perceptrón)
- No lineal
  - Bayes ingenuo
  - Vecino más cercano k-NN
  - Máquinas de soporte vectorial (SVM)
  - Árboles de decisión y bosque aleatorio.
  - Redes neuronales (Perceptrón multicapa)

# Métodos de aprendizaje supervisado

Hay dos familias de **Métodos Ensamblados**:

- **Métodos de promediación**, se construyen varios estimadores de forma independiente y luego se promedia sus predicciones. En promedio, el estimador combinado suele ser mejor que cualquiera de los estimadores por separado porque se reduce su varianza. Ejemplos: métodos de embolsado, bosques de árboles aleatorios,etc.
- **Métodos boosting**, se construyen los estimadores secuencialmente y se intenta reducir el sesgo del estimador combinado. La motivación es combinar varios modelos débiles para producir uno fuerte. Ejemplos: AdaBoost, Gradient Tree Boost,etc.

## Métodos de Clasificación

# Métodos de Clasificación

# Modelos no lineales de aprendizaje supervisado

Algunos de los métodos de clasificación son:

- Bayes ingenuo
- Vecino más cercano k-NN
- Máquinas de soporte vectorial (SVM)
- Árboles de decisión y bosque aleatorio.
- Redes neuronales (Perceptrón/Perceptrón multicapa)
- Métodos con reforzamiento (Boosting).

# Vecino más cercano k-NN

- Es un método estadístico de aprendizaje supervisado no paramétrico
- Se utiliza para clasificación y regresión.
- La entrada consta de los k ejemplos de entrenamiento más cercanos en un conjunto de datos.
- El resultado depende de si k-NN se usa para la clasificación o la regresión

# Vecino más cercano k-NN

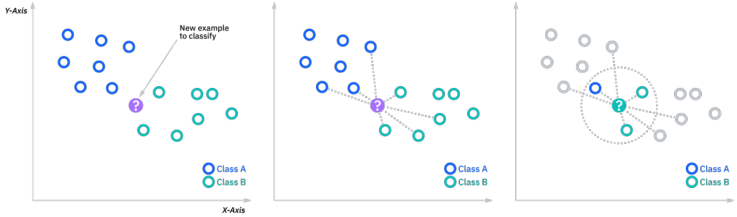



Figura 1: Vecino más cercano k-NN.

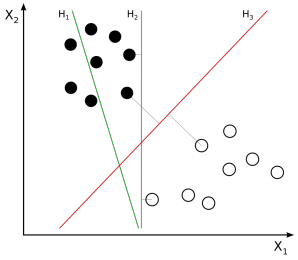
# Máquina de Soporte Vectorial (SVM)

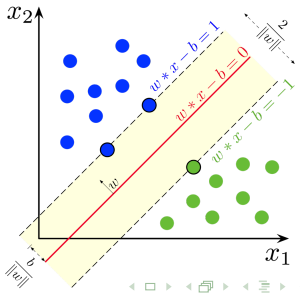
- Desarrollado en AT&T Bell Laboratories por Vladimir Vapnik (1995).
- Es un clasificador lineal binario no probabilístico
- Un concepto clave en este método es establecer separaciones lineales mediante hiperplanos.
- Maximiza el ancho de la brecha entre las dos categorías
- Los nuevos individuos se mapean en ese mismo espacio y se predice que pertenecen a una categoría según el lado de la brecha en el que se encuentran.

# Máquina de Soporte Vectorial (SVM)

- Los hiperplanos establecen límites entre datos multidimensionales.
- Maximiza el ancho de la brecha entre las dos categorías.

Figura 2: Máquina de Soporte Vectorial (SVM).



# Máquina de Soporte Vectorial (SVM)

- **Ventajas:**
  - Es idóneo para problemas de clasificación y de regresión
  - Tienen tolerancia al ruido y al sobreaprendizaje
  - Menor complejidad con respecto a las redes neuronales
- **Desventajas:**
  - Requieren de experimentación exhaustiva de los parámetros
  - Se dispara el tiempo de aprendizaje en conjuntos de datos masivos

# Máquina de Soporte Vectorial (SVM)

- **Software**
  Está disponible en plataformas populares como:
  - Statistics Toolbox, Mathworks Matlab R2014a.
  - Biblioteca e1071, lenguaje de programación R.
  - Biblioteca Accord.NET para Microsoft.NET.
  - Biblioteca Scikit-learn, lenguaje de programación Python.

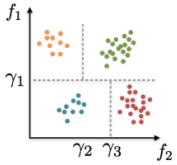
# Árboles de decisión y bosque aleatorio

## Árboles de decisión

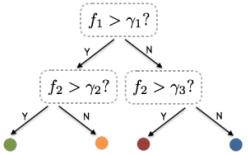
- Bosque aleatorio es un método basado en un conjunto de árboles de decisión
- Los árboles de decisión son modelos jerárquicos que dividen el dominio de datos en pequeñas partes
- Utilizan un pequeño número de cortes verticales y horizontales
- Toman en cuenta toda la información disponible al mismo tiempo
- Un árbol desarrolla un conjunto de reglas que bifurcan caracterizando los datos hacia una u otra rama de decisión
- El resultado final se obtiene por votación mayoritaria del conjunto de árboles.

# Árboles de decisión y bosque aleatorio

- Cada variable se divide por valores de corte aleatorios
- Las clases se dividen en pequeños grupos
- Estas se combinan para dar lugar a las clases finales
- El criterio de división permite crear árboles aleatorios,
- Al combinarlos se selecciona la mejor combinación de árboles
- Esto permite disminuir la varianza del estimador



(a) Geometrical interpretation



(b) Tree interpretation

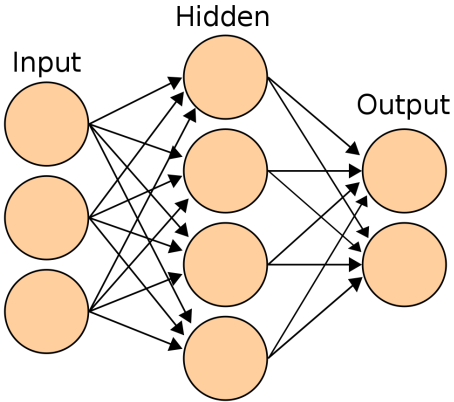
# Árboles de decisión y bosque aleatorio

- **Ventajas:**
  - Tolerancia al sobreaprendizaje
  - Bajo costo computacional.
  - No tiene un valor límite de variables de entrada.
  - Se puede asignar un valor de importancia a las variables.
  - Se utiliza para clasificación y regresión.
- **Desventajas:**
  - Pueden ser sensibles la presencia de ruido o instancias incorrectamente etiquetadas.
  - Se puede crear un bosque con relaciones muy complejas las cuales pueden dar resultados sobreestimados.
  - Pequeñas diferencias pueden crear un sinfín de árboles dentro del modelo
  - Se necesita de una base de datos de entrenamiento grande en comparación con los datos de prueba y validación para obtener resultados óptimos.

# Árboles de decisión y bosque aleatorio

- **Software**
  Está disponible en plataformas populares como:
  - Statistics Toolbox, Mathworks Matlab R2014a.
  - Biblioteca randomForest, lenguaje de programación R.
  - Biblioteca Accord.NET para Microsoft.NET.
  - Biblioteca Scikit-learn, lenguaje de programación Python.

# Redes neuronales




**Redes neuronales**

# Redes neuronales

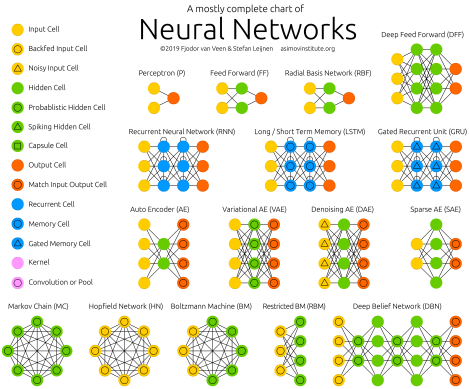



Figura 4: Tipos de redes neuronales (Leijnen and Veen, 2020 Leijnen and Veen (2020)).

# Redes neuronales

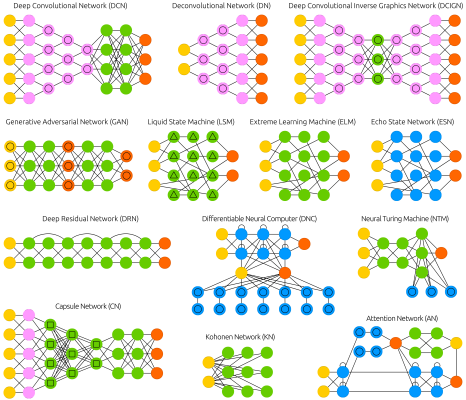
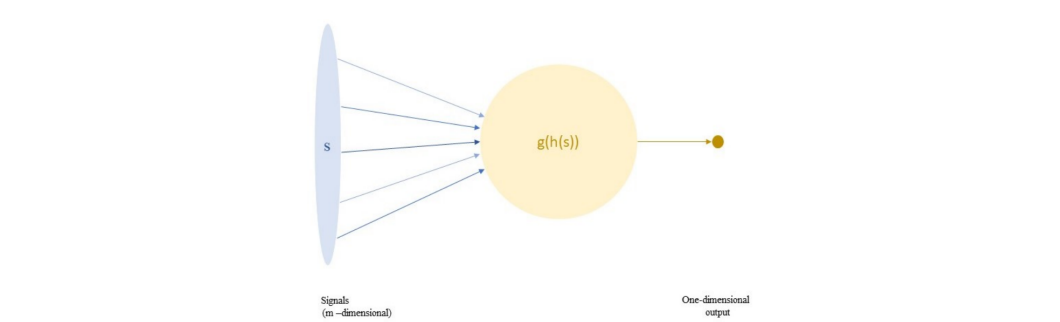



Figura 5: Tipos de redes neuronales (Leijnen and Veen, 2020 Leijnen and Veen (2020)).

# Redes neuronales

- Sistemas informáticos inspirados en las neuronas biológicas
- Colección de nodos conectados llamados neuronas artificiales
- Cada conexión puede transmitir una señal a otras neuronas.
- Una neurona artificial recibe una señal, luego la procesa y puede enviar señales a las neuronas conectadas a ella
- Las neuronas se agregan en capas
- Diferentes capas pueden realizar diferentes transformaciones en sus entradas.
- Las señales viajan desde la capa de entrada hasta la capa de salida, posiblemente después de atravesar las capas varias veces.

# Actividad de una neurona (Lichtner-Bajjaoui, 2020)




- Cada neurona recibe una señal $s = (s_1, s_2, ..., s_m)$ con $m$ entradas y un vector de pesos $w$
- Se aplica una función $h(s, w)$, que puede ser un producto escalar o una distancia
- Finalmente, pasa a una función de activación $g$ que genera una salida unidimensional $g(h)$.

# Redes neuronales

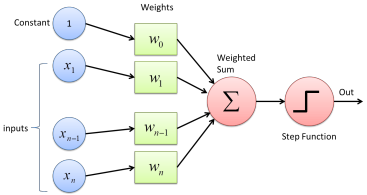



Figura 6: Perceptrón.

- La señal en una conexión es un número real
- La salida es alguna función de la suma de sus entradas.
- Las conexiones tienen un peso que se ajusta con el aprendizaje.
- La señal se envía cuando cruza un umbral.

# Redes neuronales (Perceptrón)

Figura 7: Perceptrón unicapa (derecha) y multicapa (izquierda).

- La unidad básica de una red neuronal es el **Perceptrón**.
- Detecta características o tendencias en los datos de entrada.
- Es un algoritmo para el aprendizaje supervisado de clasificadores binarios.
- Ese algoritmo permite aprender a las neuronas artificiales.

# Redes neuronales

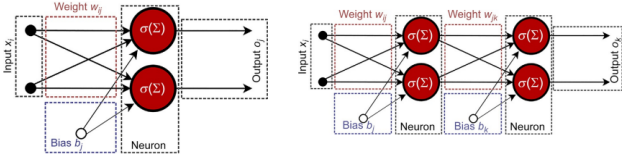



Figura 8: Perceptrón unicapa (derecha) y multicapa (izquierda).

- Las neuronas se organizan en múltiples capas (aprendizaje profundo).
- Las neuronas de una capa se conectan solo con las neuronas de las capas anterior y posterior.
- La capa que recibe datos externos es la capa de entrada.
- La capa que produce el resultado final es la capa de salida.
- Entre ellas hay cero o más capas ocultas, redes de una sola capa y sin capas.

# Redes neuronales

- **Ventajas:**
  - Aprendizaje.
  - Auto organización.
  - Tolerancia a fallos
  - Flexibilidad
  - Tiempo real
- **Desventajas:**
  - Complejidad de aprendizaje para grandes tareas
  - Tiempo de aprendizaje elevado.
  - No permite interpretar lo que se ha aprendido
  - Elevada cantidad de datos para el entrenamiento

# Redes neuronales

- **Software**

  Está disponible en plataformas populares como:

  - `Statistics and Machine Learning Toolbox`, Matlab.
  - Biblioteca `neuralnet`, lenguaje de programación R.
  - Bibliotecas `Scikit-learn`, `TensorFlow`, `PyTorch`, `NeuroLab`, lenguaje de programación Python.

# Métricas de desempeño

Existen varios índices cuantitativos para evaluar el rendimiento de clasificación de diferentes algoritmos

- La exactitud (**accuracy**) es la proporción de muestras correctamente clasificadas.
- La **precisión** es la proporción de verdaderos positivos entre las instancias clasificadas como positivas.
- La sensibilidad (**recall**) es la proporción de verdaderos positivos entre todas las instancias positivas en los datos.
- **F1** es una media armónica ponderada de precisión y sensibilidad.
- **ROC AUC** (área bajo la curva (AUC) de la característica operativa del receptor (ROC)) representa el grado o la medida de "separabilidad".

# Métricas de desempeño

Tabla de contingencia o matriz de confusión de un experimento de **P** casos positivos y **N** casos negativos para alguna condición (https://en.wikipedia.org/wiki/Precision_and_recall).

| | | **Predicted condition** | |
|---|---|---|---|
| | Total population = P + N | **Predicted Positive (PP)** | **Predicted Negative (PN)** |
| **Actual condition** | **Positive (P)** [a] | **True positive (TP),** hit[b] | **False negative (FN),** type II error, miss, underestimation[c] |
| | **Negative (N)**[d] | **False positive (FP),** type I error, false alarm, overestimation[e] | **True negative (TN),** correct rejection[f] |

# Métricas de desempeño

- La **Precisión** es la proporción de verdaderos positivos entre las instancias clasificadas como positivas.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

- La sensibilidad (**Recall**) es la proporción de verdaderos positivos entre todas las instancias positivas en los datos.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

- La **F1** es una media armónica ponderada de precisión y sensibilidad.

$$F1 = \frac{2Precision \times Recall}{Precision + Recall} \tag{3}$$

# Métricas de desempeño

- La exactitud (**Accuracy**) es la proporción de muestras correctamente clasificadas con respecto al total.

$$Accuracy = \frac{TP + TN}{Total(P + N)} \tag{4}$$

- La **Prevalencia** es la proporción de positivos con respecto al total.

$$Prevalencia = \frac{P}{Total(P + N)} \tag{5}$$

## Métodos de Regresión

# Métodos de Regresión

# Métodos de Regresión

Algunos de los métodos de regresión son:

- Regresión lineal (simple o múltiple)
- Regresión Bayesiana
- Regresión cuantil
- Regresión polinomial
- Redes neuronales (Perceptrón/Perceptrón multicapa)

# Regresión

- El **análisis de regresión** es un conjunto de procedimientos estadísticos para estimar las relaciones entre una variable **dependiente** y una o más variables **independientes**.

- El análisis de regresión se utiliza principalmente para dos propósitos conceptualmente distintos:

  1. Predicción, tiene una superposición con el aprendizaje automátizado.
  2. Inferir relaciones causales entre las variables independientes y dependientes.

# Regresión lineal simple

- El análisis de regresión es un conjunto de procedimientos estadísticos para estimar las relaciones entre una variable dependiente y una o más variables independientes.

- El análisis de regresión se utiliza principalmente para dos propósitos conceptualmente distintos:
  1. Predicción, tiene una superposición con el aprendizaje automátizado.
  2. Inferir relaciones causales entre las variables independientes y dependientes.

# Regresión Lineal Bayesiana

- Es un modelado condicional en el que la media de una variable se describe mediante una combinación lineal de otras variables,
- El objetivo es obtener la probabilidad posterior de los coeficientes de regresión
- Predicción $y$ fuera de la muestra condicionada a los valores observados $X$
- La versión más usada es el modelo lineal normal, en el que $y$ dado $X$ tiene una distribución Gaussiana

# Regresión cuantil

- Mientras que el método de mínimos cuadrados estima la media condicional de la variable de respuesta a través de los valores de las variables predictoras
- La regresión por cuantiles estima la mediana condicional (u otros cuantiles: .25, .75) de la variable de respuesta
- La regresión cuantil es una extensión de la regresión lineal que se utiliza cuando no se cumplen las condiciones de la regresión lineal.

# Regresión polinomial

- La relación entre la variable independiente $x$ y la variable dependiente $y$ se modela como un polinomio de grado $n$ en $x$.
- Ajusta una relación no lineal entre el valor de $x$ y la media condicional correspondiente de $y$, denotada como $E(y|x)$.
- Como problema de estimación estadística es lineal,
- La función de regresión $E(y|x)$ es lineal en los parámetros desconocidos que se estiman a partir de los datos.

# Métricas de desempeño

Existen varios índices cuantitativos para evaluar el rendimiento de los diferentes algoritmos

- Train. score (%) - Puntaje de entrenamiento en porciento
- Train. Time (s) - Tiempo de entrenamiento en segundos
- $R^2$ - Coeficiente de determinación
- MaxRE - Error residual máximo
- MAE - Error absoluto medio
- MSE - Error cuadrático medio
- MDAE - Error absoluto mediano
- MAPE - Porcentaje del error absoluto medio

# Flujo de trabajo del Aprendizaje Supervisado

## Flujo de trabajo del Aprendizaje Supervisado

# Flujo de Trabajo General

1. Generación de la base de datos
2. Análisis exploratorio de los datos
3. Métodos de aprendizaje no supervisados y supervisados

# Generación de la base de datos

1. **Uniformizar**: Diferentes formatos, fuentes, frecuencias, unidades, escalas, nombres de variables, etc.
2. **Revisar**: Calidad, cantidad, estructura de los datos.
3. **Depurar**: Detectar datos repetidos, fuera de rango, erróneos, etc.
4. **Consolidar**: Bases de datos estructurada para el majejo flexible de la información por las diferentes variables, categorías, en espacio y/o tiempo.

# Análisis exploratorio de los datos

1. **Univariado**: Análisis estadístico global, por unidades, por secciones, etc.
2. **Bivariado**: Análisis de dependencia global, por unidades, por secciones, etc.
3. **Multivariado**: Análisis de regresión, de agrupamiento, de componentes principales, etc

# Etapas del aprendizaje supervisado

- Datos de entrenamiento y validación
- Análisis exploratorio de los datos (univariado, bivariado, multivariado)
- Selección de variables
- Entrenamiento
- Validación
- Predicción

# Datos de entrenamiento y validación

- Usualmente se divide la base de datos que se dispone en dos subconjuntos:
  - Datos de **entrenamiento**
  - Datos de **validación**
- Típicamente en la proporción de 70/30 %, aunque se pudiera usar otra proporción.
- La selección se realiza de manera aleatoria.
- Mediante una semilla se puede prefijar la selección.
- Otras opción es el método **K-fold**: se dividen los datos en $k$-grupos de muestras de igual tamaño (equivalente a la estrategia *Leave One Out*).
- Sistemáticamente, se entrena usando $(k-1)$ grupos y se valida con el grupo que ha quedado omitido.

# Selección de variables

Índices de importancia relativa de las variables para una clasificación dada (Russell and Norvig, 2016):

- **Info. gain** nos dice qué tan importante es una variable dada (Shannon, 1948).
- **Gain ratio** es una relación entre la ganancia de información y la información intrínseca del atributo (Raschka and Mirjalili, 2017).
- **Gini** se puede considerar como un criterio para minimizar la probabilidad de clasificación errónea (Zaffar et al., 2018; Zani, 1994).
- **ANOVA** es la diferencia entre los valores promedio de la variable en diferentes clases.
- **Chi-Square** Dado un conjunto de datos sobre dos "eventos", podemos comparar el conteo observado O y el esperado E.

# Entrenamiento

Durante la fase de entrenamiento,

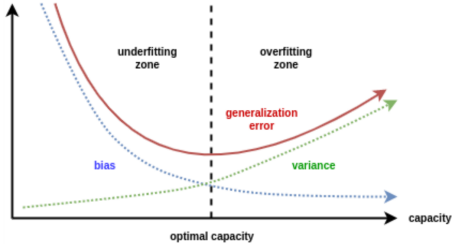
- El modelo se alimenta con el subconjunto de **entrenamiento** que contiene datos etiquetados.
- Indican la relación de las variables de entrada con las variables de salida.
- Se realiza el ajuste de los **hiperparámetros** del modelo
- Esto se pudiera realizar aplicando un método de optimización para encontrar la configuración de parámetros óptima
- Se obtienen métricas de desempeño

# Validación

Durante la fase de validación,

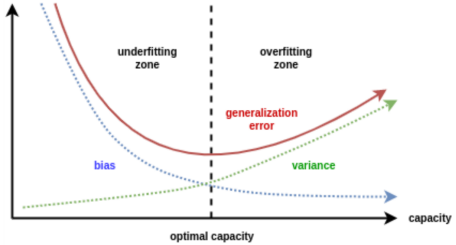
- El modelo previamente entrenado se aplica al subconjunto de **validación**.
- Los resultados se comparan con los valores de salida que se tienen en el subconjunto de validación.
- Se evalúa la calidad del modelo entrenado mediante métricas de desempeño.

# Underfitting vs. Overfitting (Lichtner-Bajjaoui, 2020)




- Por un lado, el modelo puede aproximarse muy bien para una muestra de entrenamiento dada,
- Por otro lado, aplicando la aproximación a otras muestras el resultado podría no ser bueno.
- La red no ha aprendido lo suficiente (underfitting). Aproximación sesgada con baja varianza.
- La red ha sobre aprendido (overfitting). Aproximación insesgada con alta varianza.

# Sesgo vs. Varianza (Lichtner-Bajjaoui, 2020)




- Al aumentar la dimensión de las muestras de entrenamiento se puede reducir el sesgo.
- Tiene el efecto de darle a las muestras de entrenamiento más información,
- Pero al proporcionar más información, la varianza en general aumentará.
- Esto crea una compensación entre varianza y sesgo.

# Predicción

Durante la fase de predicción,

- El modelo previamente entrenado y validado se aplica a un nuevo conjunto de datos con las mismas variables de entrada.
- Se obtienen los valores de las variables de salida de acuerdo al modelo

# Interpretación de los resultados

Finalmente,

- Se debe realizar la revisión de los resultados obtenidos de manera que sean consistentes según la experiencia del área de aplicación.
- Se recomienda realizar análisis estadísticos y gráficos.

# Referencias

Leijnen, S. and Veen, F. v. (2020). The neural network zoo. *Proceedings*, 47(1).

Lichtner-Bajjaoui, A. (2020). *A Mathematical Introduction to Neural Networks*. Phd thesis, Universitat de Barcelona, Barcelona, Spain.

Raschka, S. and Mirjalili, V. (2017). *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*. Packt Publishing. 2nd Edition.

Russell, S. and Norvig, P. (2016). *Artificial Intelligence: A Modern approach*. Pearson Education, Inc., publishing as Prentice Hall.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

## Siguiente tema:

**Ejemplos**

**de**

**Aplicaciones**