Fundamentos del Machine Learning

Que es la inteligencia artificial

- IA General: Trata de desarrollar un sistema que presenta la flexibilidad y versatilidad de la inteligencia humana para resolver un amplio rando de problemas cognitivos complejos.
- IA Especializada: Trata de desarrollar sistemas que pueden ser usados solo para las tareas para los que fueron diseñados.

Que es el machine learning

Es una rama de la IA que trata de desarrollar algoritmos que permitan a las máquinas aprender. Se busca desarrollar modelos computacionales que sean capaces de resolver problemas complejos usando como base ejemplos.

Cuando es apropiado usar machine learning

- Si no se tiene suficiente conocimiento explícito para obtener un algoritmo para resolver el problema, pero se tienen ejemplos de como se resuelve.
- Si el problema a resolver varía con el tiempo.
- Si los datos llegan continuamente y contienen nueva información que permite mejorar el sistema con el tiempo.

Generalización

Término usado para describir la capacidad de un modelo para clasificar o predecir nuevos datos correctamente. Hay 2 conceptos importantes relacionados con la generalización:

- Underfitting: El modelo no trabaja bien con los datos.
- Overfitting: El modelo trabaja demasiado bien con los datos, los memoriza, pero hace predicciones poco fiables con datos nuevos.

Hay que encontrar un balance entre estos 2 conceptos.

Praparación y limpieza de datos

Antes de usar datos para entrenar un modelo, suele ser necesario realizar ciertas preparaciones de los datos como:

- Normalizar los datos (Scaling)
- REcodificar las variables no-numéricas
- Eliminación de ruido y datos sin sentido

- Imputación de datos

Modelos lineales de aprendizaje supervisado

Primero debemos conocer las notaciones y definiciones:

- Vectores: Vienen como matrices en columna
  - Producto de vectores:

$$\mathbf{x}^\top \mathbf{y} = [x_1, x_2, \ldots, x_m]\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_m y_m$$

- Norma de un vector

$$||x|| = \sqrt{X^T X} = \sqrt{X_1^2 + x_2^2 + X^2_m}$$

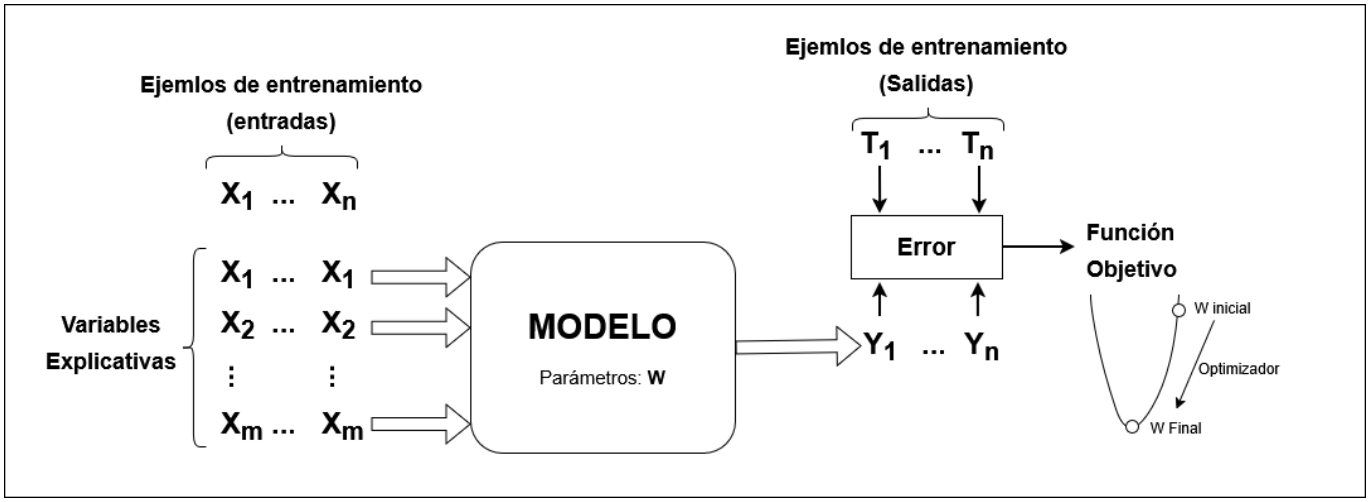
Métodos de regresión lineal

Tienen como objetivo predecir una o más variables continuas dado el valor de un set explicativo de variables repesentado por un vector X con dimensión m

$$ X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Para predecir los valores de las variables tenemos los siguientes elementos clave:

- **Variables explicativas**: variables de entrada del modelo
- **Ejemplos de entrenamiento**: Un grupo de n datos $x_1 ... x_n$ de las variables explicativas para el cual el valor de la variable a ser predecido es conocido $t_1,...,t_n$
- **Un Modelo**: Es una función parametrizable W que representa la relación entre x y t
- **Función objetivo** (o error o coste) que indica como de bien tiene que aproximar el modelo los datos de entrenamiento
- **Un método de optimización** para encontrar el modelo óptimo minimizando la función objetivo.



- **Proceso de entrenamiento**: El objetivo es construir el modelo para obtener los parámetros W óptimos para predecir el valor de t para un nuevo valor de x. Esti requiere un dataset de entrenamiento compuesto por n observaciones $x_1 ... x_n$ y un set de valores predecidos $t_1,....,t_n$ (Aprendizaje supervisado)

Métodos de clasificación lineal

Clasificación supervisada

Es similar a la regresión excepto en que el valor predecido toma valores dentro de un pequeño set discreto de datos. En el caso específico de clasificación binarias solo hay 2 posibles valores para cada item, por ejemplo:

- $t_i=0$ para datos de clase negativa
- $t_i=1$ para datos de clase positiva

Es una clasificación supervisada por que las etiquetas están disponibles para los datos entrenados.

Regiones de decisión

- Métodos de aprendizaje lineal: Las superficies de decisión generadas son funciones lineales de datos (hyperplanos)
- Métodos de aprendizaje no lineal: Las superficies de decisión que generan son funciones no lineales de los datos.

Conclusión sobre los clasificadores lineales

- Least Squares: No es una buena elección en general para clasificaciones ya que es muy sensible a los desbalances
- Regresión logística: El método más robusto ya que no es sensible a desbalances.

Métricas para la evaluación del error

Primero es necesario diferenciar entre error de función y métrica de evaluación:

- La función de pérdida es la función usada para optimizar el modelo. Minimizada durante el entrenamiento y, en general, es diferenciable en los parámetros del modelo
- Las Métricas de evaluación son utilizadas para juzgar el rendimiento del modelo. No interviene durante la optimización del proceso del modelo y no necesita ser diferenciable.

Métricas de error de evaluación para regresión

- **Mean Squared Error (MSE)**: Es la siguiente función donde y_i es la salida del modelo y t_i es la salida actual para el dato i. Pesa los errores grandes de forma más pesada debido al uso del error cuadrado.

$MSE = \frac{1}{n}∑^n_{i=1}(t_i-y_i)^2$

- **Root Mean Squared Errors (RMSE)**: Ventaja sobre MSE, facilita la interpretación ya que el error obtenido es relativo a las unidades de los datos.

$RMSE = \sqrt{\frac{1}{n}∑^n_{i=1}(t_i-y_i)^2}$

- **Mean Absolute Error (MAE)**: Trata todos los errores de la misma forma, mientras que MSE y RMSE le dan una mayor penalización a los errores altos. Poco adecuada si se quiere prestar atención a errores de predicción potencialmente grandes.

$MAE = \frac{1}{n}∑^n_{i=1}|t_i-y_i|$

- **Mean Absolute Percentage Error (MAPE)**: Tiene como ventajas que, al ser un procentaje, es independiente de la escala de los datos, es fácil de interprettar, aunque su valor puede ser mayor del 100% y que es resistente a los errores de outlier. También tiene varios puntos negativos, por ejemplo, hay problemas con la división si hay valores a ser predecidos equivalentes 0, si los valores son pequeños puede tener valores demasiado grandes y no es fiable frente a predicciones sistemáticamente mñas pequeñas que los valores actuales.

$MAPE = \frac{100}{n}∑^n_{i=1}|\frac{t_i-y_i}{t_i}|$

- **Symmetric Mean Absolute Percentaje Error (SMAPE)**:

- Ventajas:
  - Interpretación clara al ser procentajes
  - No afectado por outliers
  - Genera valores entre 0% y 100% solucionando el problema de MAPE
  - Es simétrico con respecto a predicciones bajas o altas.
- Desventajas:
  - Tiene problemas con la división si la predicción y el valor actual son equivalentes a 0
  - No es simétrico ya que las sobrepredicciones e infrapredicciones no se tratan de forma equivalente.

$$SMAPE = \frac{100}{n}∑^n_{i=1}\frac{|t_i-y_i|}{|t_i|+|y_i|}$$

Métricas de evaluación del error para la clasificación supervisada

Tenemos una matriz de confusión para dos clases de problemas:

- **Error de tipo I**: Falsos positivos (FP)
- **Error de tipo II**: Falsos Negativos (FN)

Predicción de clasificador/Clase Real	**Positivo**	**Negativo**
Positivo	Verdaderos Positivos (TP)	Falsos positivos (FP)
Negativo	Falsos negativos(FN)	Verdaderos negativos (TN)

- **Precisión**: Tasa global de aciertos del sistema. Es problemática cuando se trabaja con datasets desbalanceados.

$$Exactitud= \frac{Aciertos Globales}{Casos Totales} = \frac{TP+TN}{TP+TN+FP+FN} $$

- **Sensibilidad**: de todos los ejemplos positivos cuantos fueron predecidos como positivos.

$$Sensibilidad = \frac{Aciertos de casos positivos}{Casos positivos} = \frac{TP}{TP+FN}$$

- **Especificdad**: De todos los casos negativos, cuantos fueron predecidos como negativos.

$$Especifidad = \frac{Aciertos de casos negativos}{Casos negativos} = \frac{TN}{TN+FP}$$

- **Precisión**: De todos los ejemplos clasificados como positivos, cuantos eran de verdad positivos.

$Precisión = \frac{Aciertos para casos clasificados como positivos}{Casos clasificados como positivos} = \frac{TP}{TP+FP}$

- **Puntuación F-1**: Media harmónica entra la Exactitud y la sensibildiad. Ventaja: mejor mñetrica que exactitud si hay desbalance. Desventaja: no tiene en cuenta los verdaderos negativos.

$Puntuación F-1 = 2 + \frac{Exactitud*Sensibilidad}{Precisión+Sensibilidad}$

- **Curva ROC (Receiver Operating Characteristics)**: Muestra gráficamente la sensibilidad en comparación con los falsos positivos como función del clasificador de discriminación.
- **Aárea bajo la curva ROC (AUC)**: Calcula el área bajo la ROC, su valor se encuentra entre 0 (peor caso) y 1 (Mejor caso)
- **Macro Media**: Calcula la media de las métricas de todas las clases

$Macro-Media = \frac{1}{c}∑^c_{i=1}Métrica de la clase i$

- **Media pesada**: Calcula la media de la metrcica pero pesada por el número de casos por clase.

$media pesada = \frac{Σ^c_{i=1} num. datos clase i * métrica de la clase i}{num. datos locales}$

- **Micromedia**: Para cada métrica de verdaderos positivos individuales, pfalsos positivos y falsos negativos, se suman juntos para cada clase:

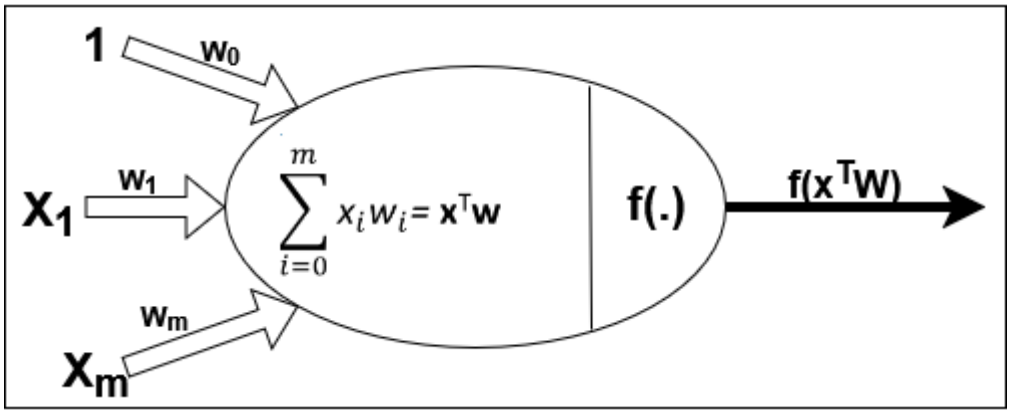
$MicroMedia = \frac{TP1 + TP2 + TP3}{TP1+TP2+TP3+FP1+FP2+FP3}$

Modelos no lineales de aprendizaje supervisado

Los métodos de aprendizaje no lineales permten la creación de regiones de decisión complejas para separar datos de diferentes clases. En general suelen rendir mejor que los modelos lineales.

Redes Neuronales Artificiales

Muchos modelos de Machine Leargin están inspirados por la biología. Las redes neuronales artificiales definen funciones de las entradas que son calculadas por las neuronas. El modelo matemático de una neurona artificial sería el siguiente:



Las neurona están organizadas en una serie de capas que definen la arquitectura de la red:

- Feed Forward Networks: Unidireccional, solo se puede avanzar a la siguiente capa, no se puede ir para atrás
- Recurrent Networks: Bidireccionales, pueden avanzar en ambas direcciones.

El entrenamiento de las redes trata de encontrar el peso óptimo W^*

$$W^* = argmin_w \sum^n_{i=1}E(y_i, t_i)$$

t_i es la salida deseada e y_i es la salida para los datos i. Las funciones de error más utilizadas son:

$$MSE = E(y_i,t_i) = \frac{1}{2}(y_i-t_i)^2$$

$$Entropía Cruzada = E(y_i,t_i) = -t_i log(y_i)$$

- **Error Back Propagation**: Método efinciente para calcular el gradiente necesitado para realizar la optimización de los pesos en una red multicapa.
    1. Con un bloque de datos de entrenamiento, propagasop la salida hacia adelante para calcular el error en la salida
    2. Propagamos hacia atrás el error para obtener el gradiente del error en cada peso
    3. Usamos el gradiente para actualizar los pesos.

Región de decisión

- Si la red no tiene capas ocultas, entonces la región de decisión es un hiperplano (Clasificador

lineal)
- Si la red tiene 1 capa oculta, la región de decisión es convexa (Abierta o cerrada)
- Si la red tiene 2 capas ocultas, la región de decisión es una combinación de redes convexas.

Redes Neuronales Convolucionales (CNN)

Modelo de red profunda que es capaz de capturar dependencias espaciales y temporales en una imagen aplicando filtros relevantes. La arquitectura es mucho mejor para datos de imágenes debido a la reducción en el número de parámetros involucrados yy el reuso de pesos.

Capa convolucional

Consiste en un grupo de filtros (Kernels) que sin aplicados a la imagen para que pueda ser aprendida. Los filtros se activan cuando cierto tipo de característica visual es detectada. El objetivo de la operación convolucional es extraer características relevantes de la imagen. La operación convolucional realiza productos de puntos entre los filtros y las regiones locales de la imagen de entrada.

La arquitectura de una CNN suele tener más de una capa convolucional. La primera capa convolucional es responsable de capturar las características a bajo nivel como bordes, colores, orientación de gradientes… Las consiguientes capas covolucionales en la arquitecutra capturan características de alto nivel.

Capa ReLU

Utilizadas normalmente después de las capas convolucionales para transformas las operaciones lineales retiradas por esta. Es una funciónde activación defenida por $max(0,x)$. No afecta al tamaño del volúmen de salida de la capa convolucional.

Capa Pooling

Se encarga de reducir el tamaño de la salida de la capa convolucional. Este tipo de capas se usan para reducir el número de parámetros en la red y así controlar el overfiting. Los pixeles vecionos en las imagenes tienden a tener valores similares, por lo que las capas convolucionales van a producir valores similares para los pixeles vecinos. Realiza transformacionaes que son similares a rotaciones y translaciones. Es común insertar períodicamente una capa poling entre sucesivas capas covolucionales en las CNN. Existen varios tipos de Pooling:

- Max Pooling: Devuelve el valor máximo de la parte de la imagen cubierta por el filtro (kernel)
- Pooling medio: Devuelve la media de todos los valores de la parte de la imagen cubierta por el filtro (kernel)

Capa completamente conectada

Permite aprender combinaciones no lineales de las características de alto nivel dadas por la capa

convolucional. La entrada de esta capa es una transformación aplanada en un vector columna. A través del proceso de entrenamiento, el modelo es capáz de distinguir entre características dominantes y clasificarlas usando una función de salida softmax.

Recomendaciones para un buen entrenamiento de modelos

- Normalizar los datos: Extrae la media y divide por la desviación típica de cada pixel
- Incrementar los datos disponibles: Añadir imagenes rotadas...
- Usar datos balanceados entre las clases
- Randomizar el orden de los datos antes de entrenar un grupo
- Inicializar aleatoriamente los pesos con variaza apropiada
- Usar set de validación apra comprobar el rendimiento obtenido y evitar el overfitting
- Si el dataset disponible es pequeño, entrena primero el modelo con un dataset grande y luego modifica le modelo con el dataset objetivo.