

ETIQUETAS: MD2::(07)MachineLearning

Un programa aprende a partir de la experiencia  $E$  con respecto a una tarea  $T$  y una medida de rendimiento  $P$ , si  $\{\{c1::\text{su desempeño en la tarea } T, \text{ medido a través de } P, \text{ mejora con la experiencia } E\}\}$

## **¿Que cosas generaliza un programa que utiliza machine learning?**

Patrones en datos de ejemplos utilizados durante un entrenamiento

## **¿Para que se generalizan los patrones en datos de ejemplos utilizados durante un entrenamiento?**

Para encontrar esos patrones en ejemplos nunca antes vistos

## **¿Cuales son las "partes del problema"? (Machine Learning)**

- Tarea  $T$
- Experiencia  $E$
- Medida de rendimiento  $P$

## **¿Cómo se llama la columna que representa aquellos datos que se quieren predecir?**

Variable objetivo

## **¿Existe un enfoque único para la selección de características de experiencia?**

No

## **¿Cuales son los 3 objetivos de seleccionar que información se utiliza para entrenar un modelo?**

- Mejorar la métrica de rendimiento
- Evitar la maldición de la dimensionalidad
- Interpretabilidad y comprensión

## **¿Por qué al seleccionar solo las características más relevantes podemos mejorar la métrica de rendimiento?**

Porque reducimos la dimensionalidad/complejidad del problema

Extra: Solo nos quedamos con las que aportan información relevante

## **¿Que es la maldición de la dimensionalidad?**

A medida que aumenta el número de características, se vuelve más difícil obtener un buen rendimiento del modelo

## **¿Que 2 problemas nos da la maldición de la dimensionalidad?**

- Sobreajuste (overfitting)
- Mayor costo computacional

## **¿Por qué es importante elegir características relevantes para la interpretación y comprensión?**

Para comprender mejor los factores que influyen en nuestra tarea de aprendizaje

Extra: Desarrollamos criterio

## **¿Que variables no se pueden usar para predecir la variable objetivo? (2 cosas)**

- Variables que son consecuencia de la variable objetivo
- Variables que no se conocen al momento de predecir

## **¿Cómo necesitan los algoritmos de aprendizaje automatico que se representen los datos?**

Como valores numéricos

## **¿Que es el vector de características?**

Es la representación de un ejemplo en el espacio de características

## **¿Cómo se construye el vector de características?**

Se concatenan los valores de las variables de entrada

## **¿Que es una función de perdida?**

Una función que le asigna un número real a cada una de las predicciones del modelo

## ¿Que ventajas tienen los modelos lineales de machine learning? (3)

- Fáciles de interpretar
- Eficientes
- Pocos parámetros

$\theta_i$  es el "peso" que tiene el feature  $x_i$

## ¿Por qué se divide en conjunto de datos en 2 partes?

Para evitar el sobreajuste (overfitting)

## ¿Cuales son los 2 conjuntos en los que se divide el conjunto de datos?

- Conjunto de entrenamiento
- Conjunto de evaluación

## ¿Cuando hay underfitting?

Cuando el rendimiento en datos de entrenamiento es bajo

## ¿Cuando hay overfitting?

Cuando el rendimiento en datos de entrenamiento es alto y en datos de evaluación es bajo

## ¿Que es la regularización en machine learning?

Agregar una penalización adicional a la función de pérdida

## ¿Para que se utiliza la regularización en machine learning? (2 cosas)

- Para evitar el sobreajuste (overfitting)
- Para reducir la complejidad del modelo

## ¿Que se agrega en la función de perdida con la regularización L1?

$$\lambda \sum_{i=1}^n |\theta_i|$$

## ¿Que se agrega en la función de perdida con la regularización L2?

$$\lambda \sum_{i=1}^n \theta_i^2$$

## ¿Que cosas esta capturando el modelo cuando hay overfitting?

Ruido o detalles irrelevantes

## ¿Que es el optimizador?

Una funcion que encuentra el valor de los parametros que minimiza la funcion de perdida

## ¿Cómo es la función que se usa para el descenso del gradiente?

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \alpha \frac{\partial}{\partial \theta_i} \text{loss}(\theta_i^{(t)})$$

El proceso del descenso por el gradiente implica ajustar los parámetros del modelo  $\{\theta_i\}$  en la dirección opuesta al gradiente de la función de costo

## ¿Que indica el gradiente de la función de costo?

La dirección de máximo crecimiento de la función de costo

## ¿Cuales son los hiperparámetros de un modelo lineal de machine learning?

- Cantidad de iteraciones
- Learning rate
- Regularización
- Estandarización

## ¿Por qué realizar estandarización de los datos?

Porque algunos algoritmos (como el descenso por gradiente) pueden tener dificultades para converger si los features tienen escalas muy diferentes

## ¿Que es la estandarización de los datos?

Transformar los datos para que tengan media 0 y desviación estándar 1

## ¿Que problemas tiene la estandarización de los datos?

Como muevo la media a 0 se acerca el gradiente a 0 y tenemos el problema de vanishing gradient

## **¿Cómo evitar el problema de vanishing gradient en la estandarización de los datos?**

Mover la media a 1 o -1

## **¿Que es el batching para calcular el gradiente de la función de costo?**

Calcular el gradiente de la función de loss sobre un subconjunto de los datos

## **¿Cuales son las 3 ventajas de usar batching para calcular el gradiente de la función de costo?**

- Mayor eficiencia computacional
- Mayor estabilidad del gradiente
- Regularización efectiva

## **¿Cual es la desventaja de usar batching para calcular el gradiente de la función de costo?**

El batch elegido puede no ser representativo

## **¿Porque tenemos mayor eficiencia computacional al usar batching para calcular el gradiente de la función de costo?**

Porque los calculos entran en la memoria de la GPU

## **¿Porque tenemos mayor estabilidad del gradiente al usar batching para calcular el gradiente de la función de costo?**

Porque se reducen las fluctuaciones en la estimación del gradiente

## **¿Cómo elegir un batching correcto?**

- Recorrer todo el dataset
- Elegir un tamaño de batch que sea representativo
- Tomar batches aleatorios y varias iteraciones

## **¿Que sucede si el tamaño del batch es pequeño?**

Me aseguro que entra en memoria

## **¿Que sucede si el tamaño del batch es grande?**

Se minimiza el impacto de los outliers

## **¿Cómo se soluciona el problema de exploding gradient?**

Recortando el gradiente si es mayor a un valor umbral

## **¿Que cosa no pueden hacer los modelos lineales?**

XOR

## **¿Cuales son los hiperparámetros que se agregan en un modelo de machine learning no lineal?**

- Arquitectura de la red
- Función de activación

## **¿Cómo se compone la arquitectura de la red en un modelo de machine learning no lineal?**

- Cantidad de capas
- Cantidad de neuronas por capa

## **¿que ventaja tiene aumentar la cantidad de neuronas de una red?**

- Tiene la capacidad de representar problemas más complejos

## **¿Que 2 desventajas tiene aumentar la cantidad de neuronas de una red?**

- Mayor cantidad de datos necesarios para el entrenamiento
- Más complicado de entrenar

## **¿Que cosa introducen las funciones de activación en un modelo de machine learning?**

No linealidad

## ¿En que consiste el dropout?

En desactivar aleatoriamente un porcentaje de neuronas en cada iteración

El dropout es una técnica de `{{c1::regularización}}`

## ¿Que problema trae el vanishing gradient?

Que el modelo se entrena muy lento

## ¿Que problema trae el exploding gradient?

las actualizaciones son muy grandes y podemos “pasarnos de largo” los minimos locales o diverger

## ¿Cómo solucionar el vanish gradient? (3)

- Mejorar la inicialización de los pesos
- Usar funciones de activación que no lleven a valores muy pequeños
- Usar optimizadores adaptativos