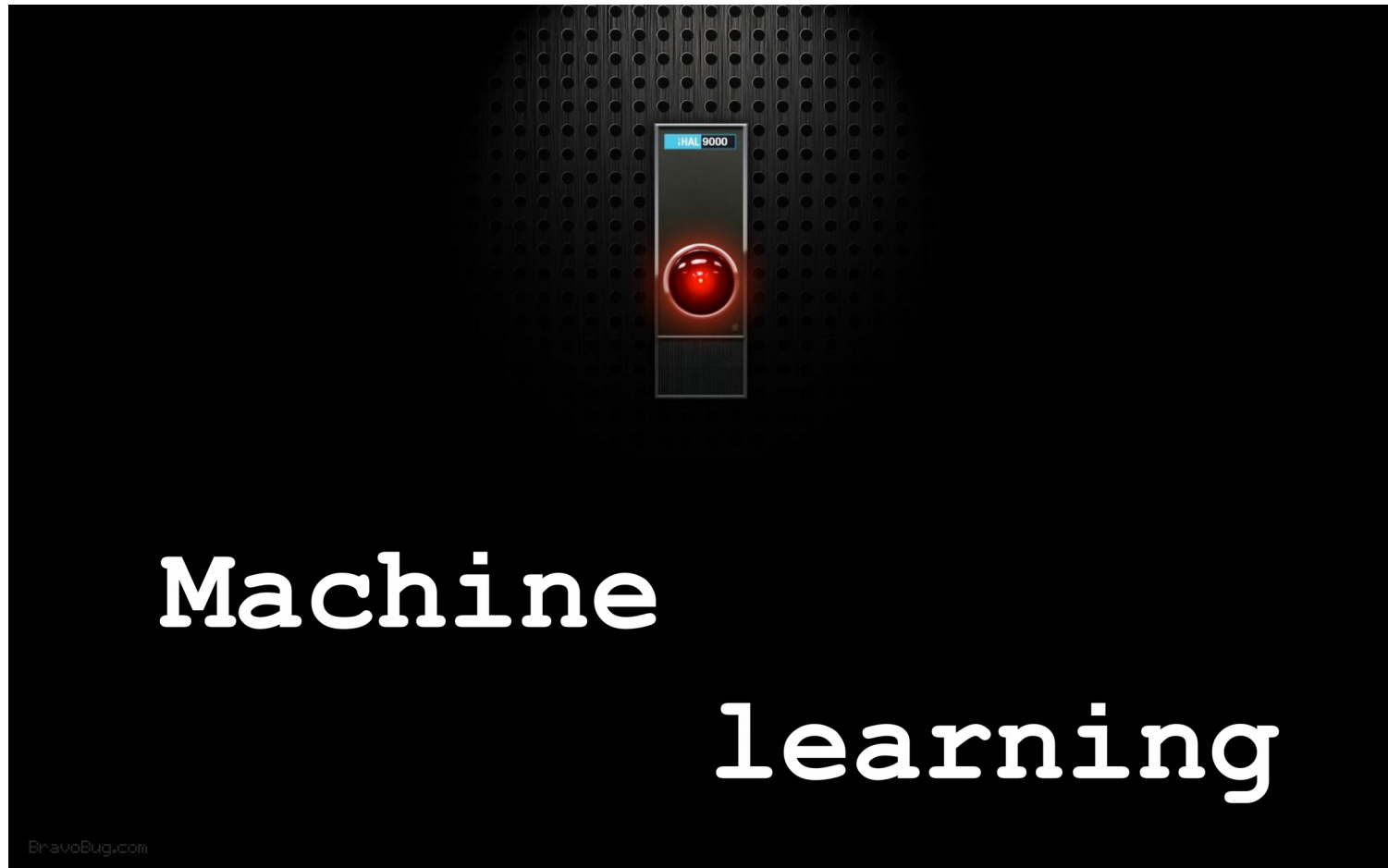


Introducción



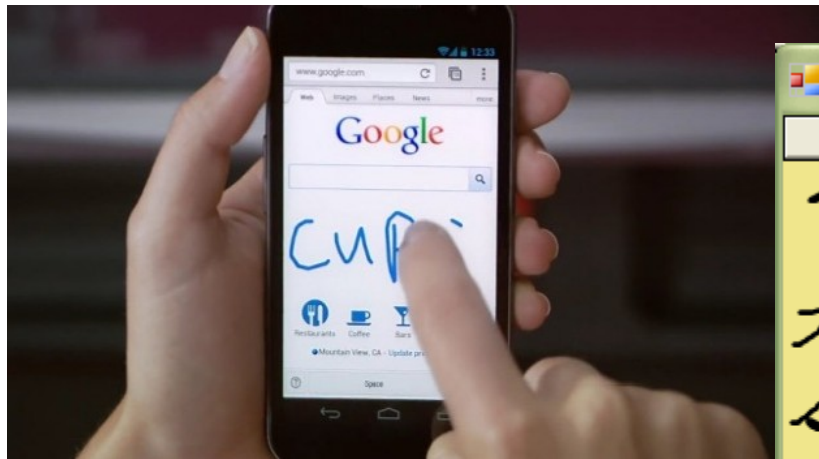
Qué es Machine Learning?

Introducción

- Hay problemas en Informática que se pueden “definir” concretamente y son simples de convertir en un algoritmo
 - Ejemplo: Ordenar alfabéticamente una lista, calcular el balance de una cuenta.
- Hay otros que son simples de “entender” pero muy difíciles de “definir” y convertir en algoritmo
 - Ejemplo: Detectar una sonrisa en una cara, interpretar un gesto del lápiz como una letra dada

El Aprendizaje Automatizado introduce métodos que pueden resolver esas tareas “aprendiendo” la solución a partir de ejemplos de cómo se realiza la misma

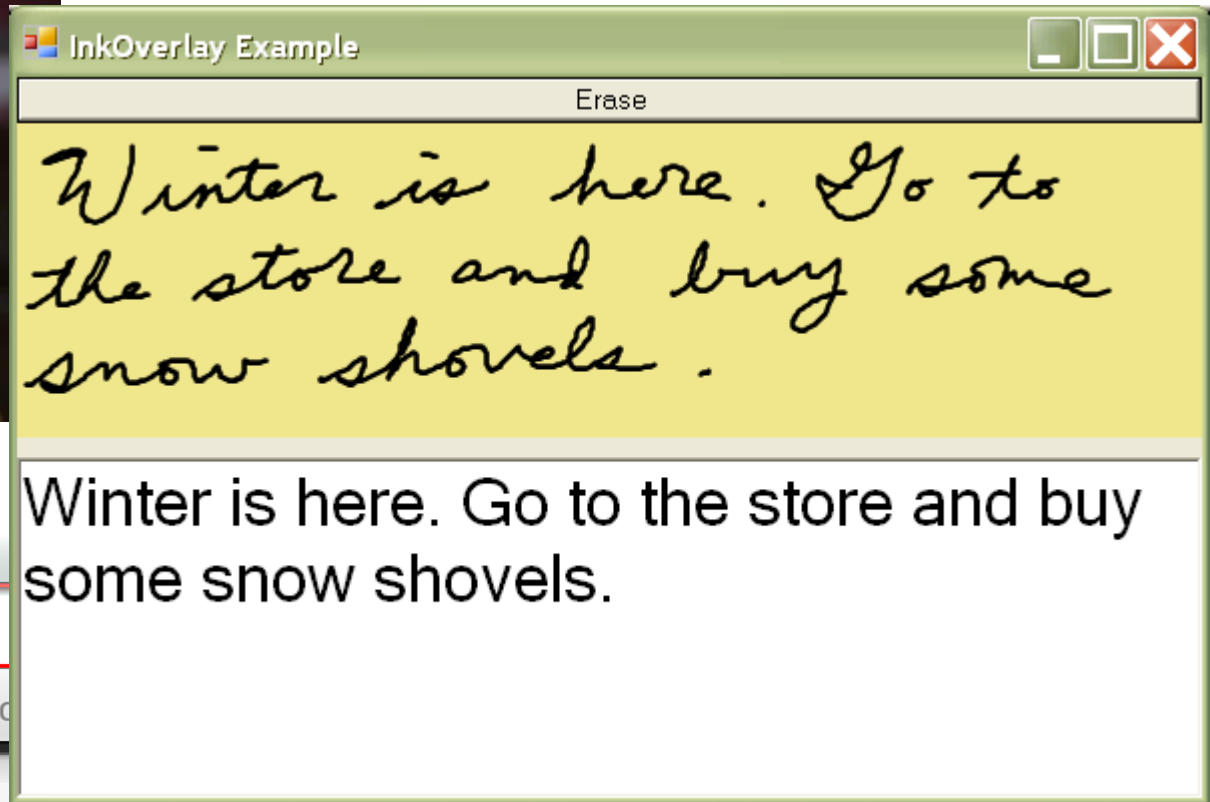
Introducción



Enter status

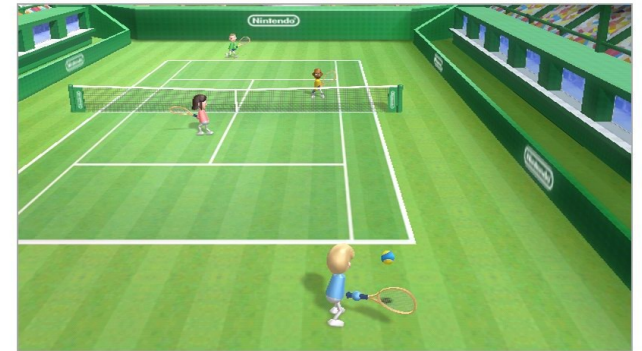
Android android Ardroid Andvoid Alo

WritePad for Android



WritePad for Android

Introducción



Introducción

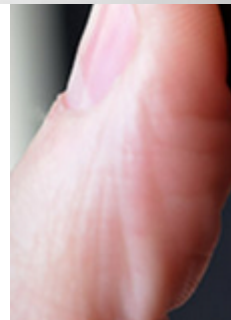


Siri

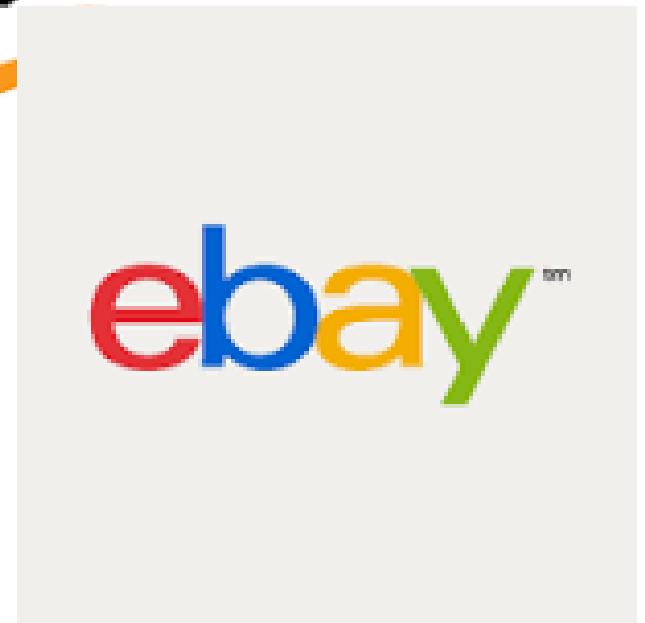
Use your voice to send messages, set reminders, search for information, and more.



Hi there. I'm Cortana.



Introducción



Problemas en ML

- Clasificación
- Regresión
- Ranking-Retrieval
- Detección de novedades
- Clustering
- Identificación de inputs relevantes
- Etc, etc.

Clasificación

Problema:

Dado un objeto (conjunto de características medidas de alguna forma) asignarle una (o varias) etiqueta de un conjunto finito.

Ejemplo:

asignar un símbolo alfanumérico a una secuencia de movimientos del lápiz en la pantalla táctil

Asignar automáticamente una noticia a diferentes grupos de interés (una o más clases)

Regresión

Problema:

Dado un objeto asignarle un número real.

Ejemplo:

Predecir la relación euro-dólar de mañana.

Predecir niveles de stock/ventas a futuro.

Búsqueda y Ranking

Problema:

Dado un objeto, asignarle y ordenar las respuestas más probables dentro de una base de datos.

Ejemplo:

Buscadores en Internet

Sistemas de recomendación

Detección de novedades

Problema:

Detectar "outliers", objetos que son diferentes a los demás.

Ejemplo:

Alarmas de comportamiento en compras con tarjeta.

Detección de fallas en equipos críticos.

Clustering

Problema:

Detectar grupos de objetos que tienen características similares.

Ejemplo:

Segmentación de consumidores/clientes a partir de sus patrones de compra/búsqueda. Marketing "dirigido".

Detección de inputs relevantes

Problema:

Dado uno de los problemas anteriores (u otro) y sus datos, averiguar cuales de las variables son responsables de la solución.

Ejemplo:

El "nuevo método científico": tomar muestras sanas y con alguna enfermedad. Analizar miles de variables con un método automático (MALDI-TOF, DNA-microchips) y buscar cuales de las variables monitoreadas son relevantes al problema.

Programas que aprenden?

“Se dice que un programa aprende si mejora su performance en una cierta tarea al incorporar experiencia”

Programas que aprenden?

Memorizar no es aprender

Generalizar es aprender

Como logramos generalizar?

Tengo estos datos:

8 – T

2 – T

5 – F

9 – F

4 – T

13 – F

Cual es la respuesta
para 12?

Y si agrego los datos:

14 – F

16 – T

Como logramos generalizar?

Para generalizar incorporamos “algo” a los datos: un bias.

En general usamos la “navaja de Occam”: La respuesta más simple que explica las observaciones es la válida

Distintos métodos de ML usan distintos bias

Métodos básicos

Arboles de decisión

- Probablemente el método más conocido para resolver problemas de clasificación
- Muy asociado a nuestra forma de proceder
- Uno de los primeros desarrollos en ML

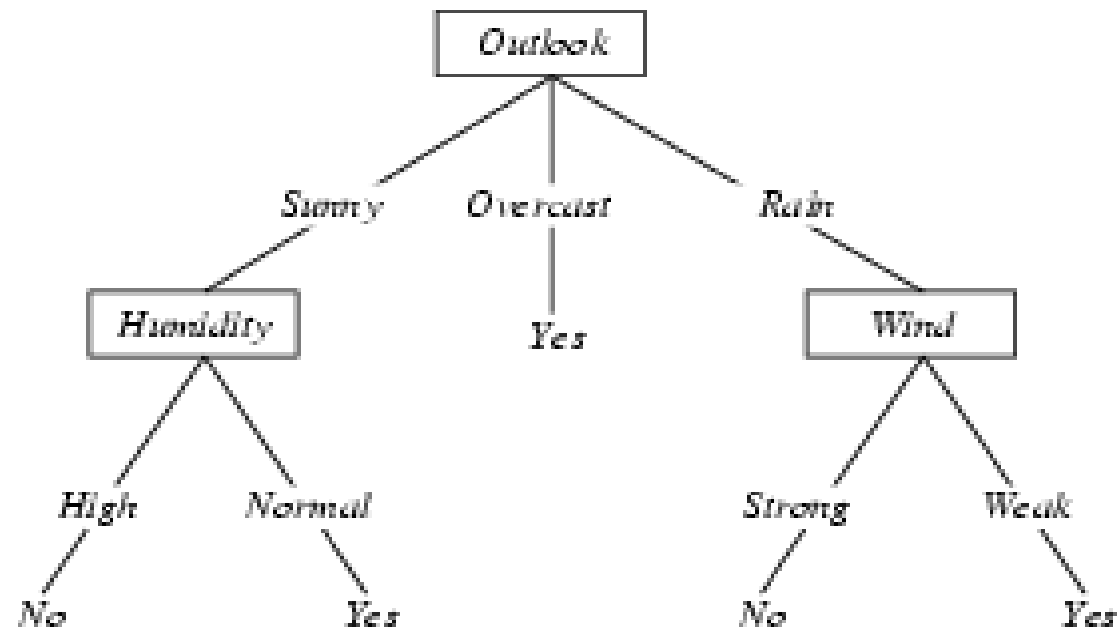
Arboles de decisión

Ejemplo: “Play Tennis”

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Arboles de decisión

Ejemplo: “Play Tennis”



Arboles de decisión

Cómo construimos el árbol?

- Hay variables más relevantes que otras.
- Si ponemos más alto las más relevantes, seguramente el árbol llegara antes a la solución, será mas simple.
- Un árbol más simple seguramente generalizará mejor (Occam).

Arboles de decisión

Cómo elegir que variable usar para dividir?

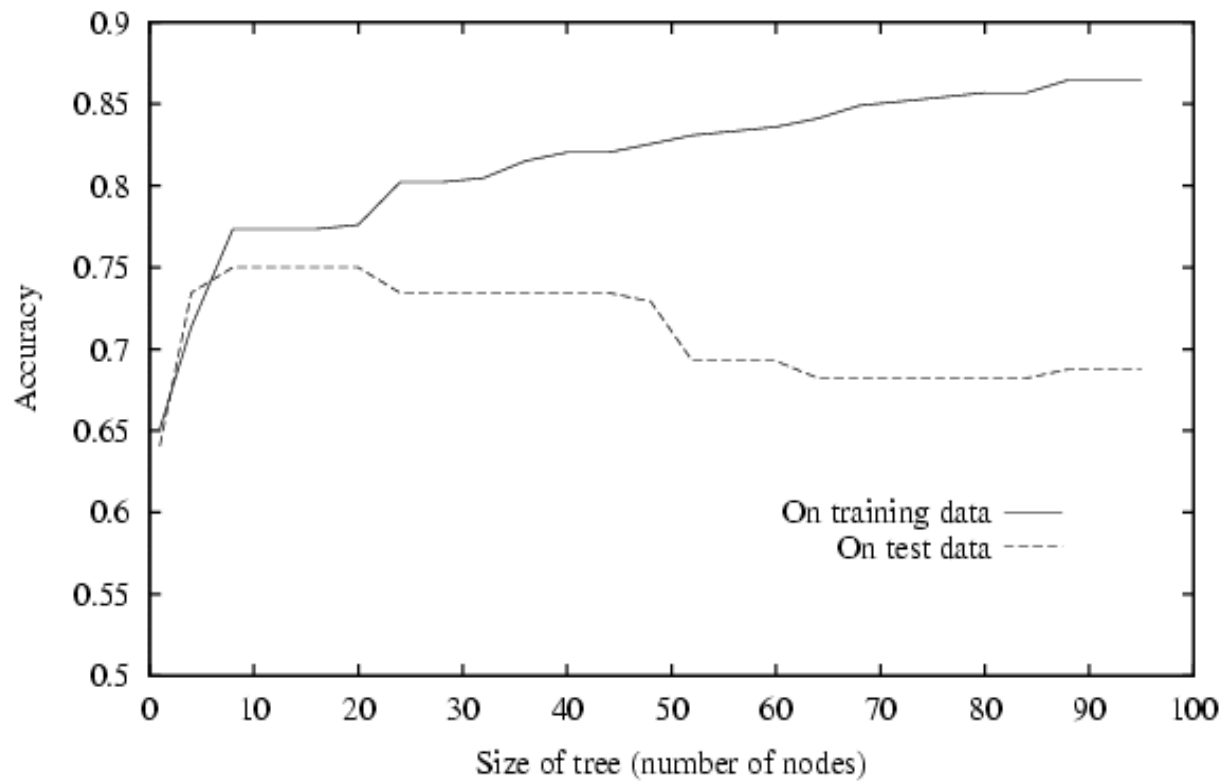
- Necesitamos la más “relevante”
- Busquemos la que da más información sobre la clase->Information Gain, correlación, etc.
- Dividimos e iteramos

Arboles de decisión

Hasta cuando dividimos?

- Hasta que tenga clases puras en todos los nodos
- Hasta que no tenga más variables disponibles
- O hay algo mejor?

Arboles de decisión



Sobreajuste!

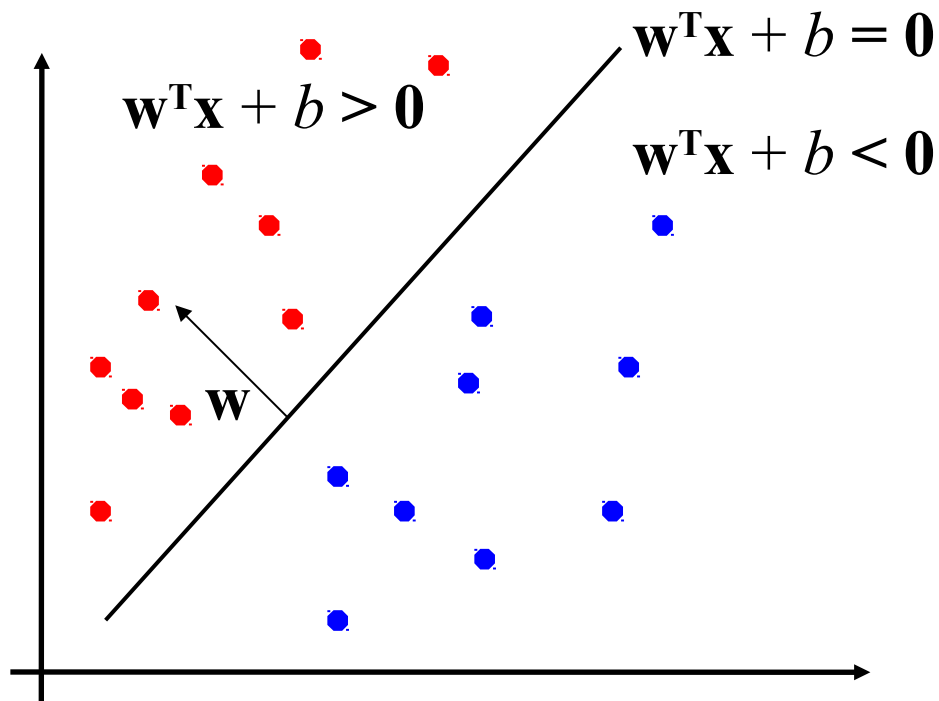
Árboles de decisión

Cómo controlar el sobreajuste?

- Necesitamos un conjunto independiente de datos, sobre el que podemos controlar la capacidad de generalización (“Validación”).
- Cuando deja de mejorar, detenemos el proceso.

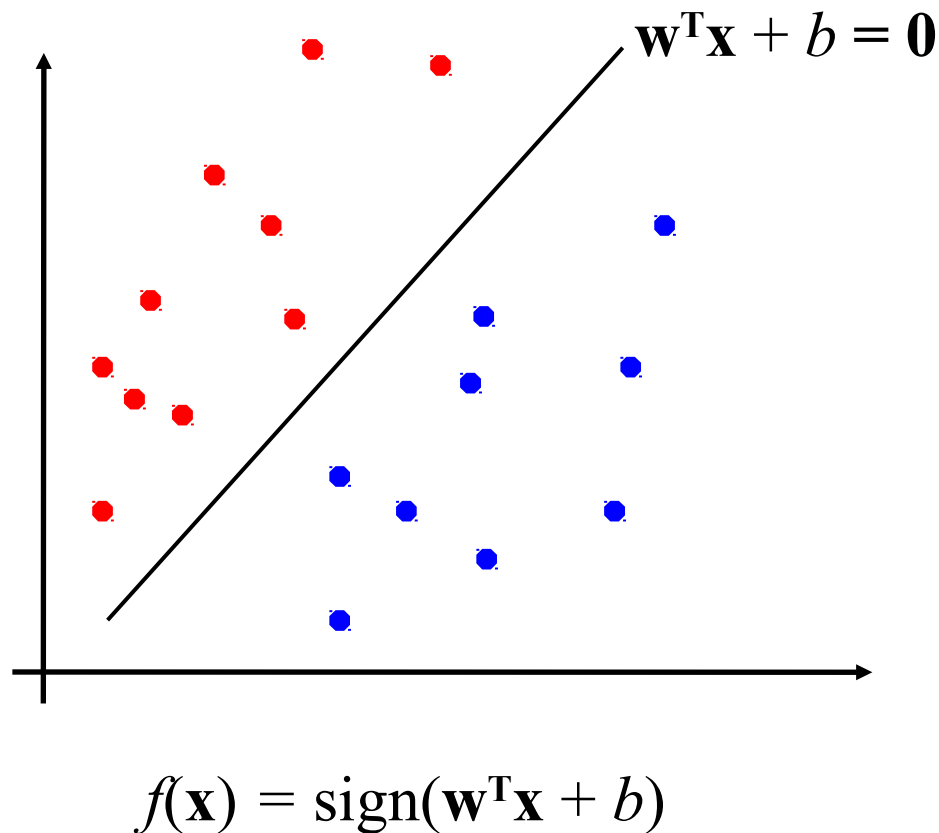
El perceptron

Si los datos están en un espacio vectorial...



$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$$

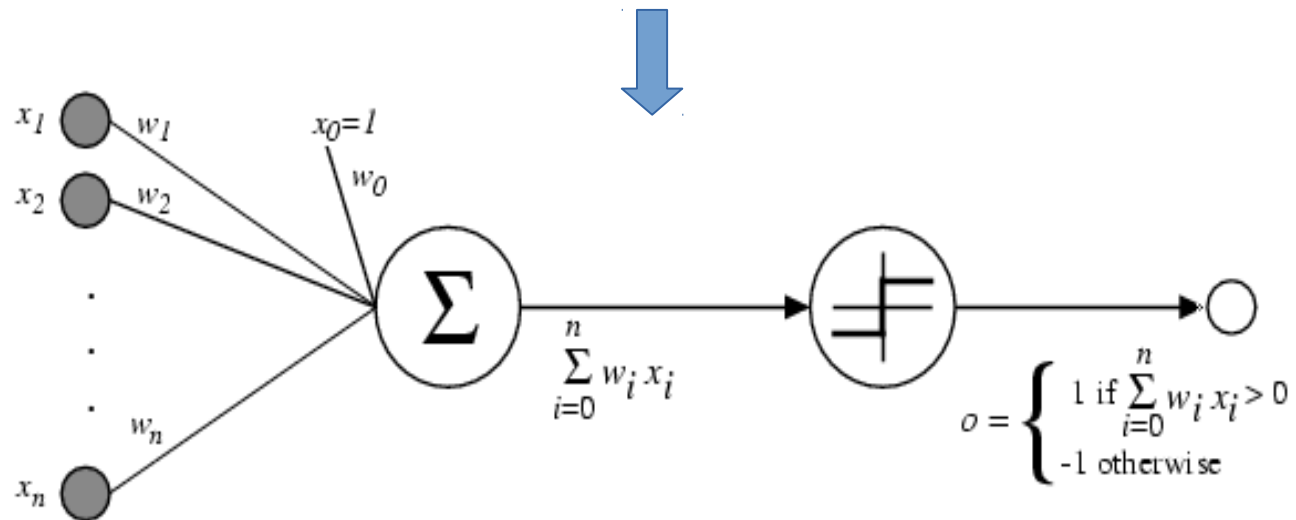
El perceptron



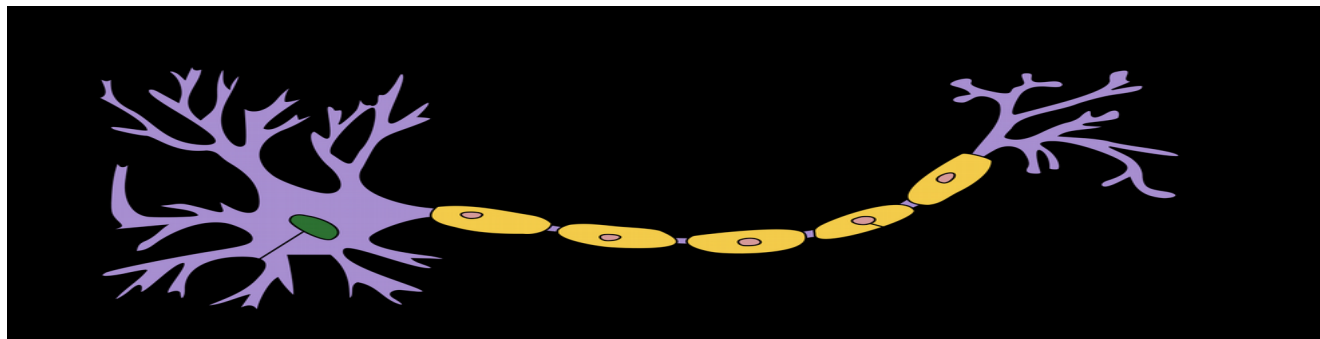
- Tenemos una regla de clasificación de forma fija
- Aprender: encontrar el mejor \mathbf{W} y b para el problema
- Regla de aprendizaje del perceptron: si es incorrecto nuevo \mathbf{W} hacia el ejemplo
- Converge a la solución

El perceptron

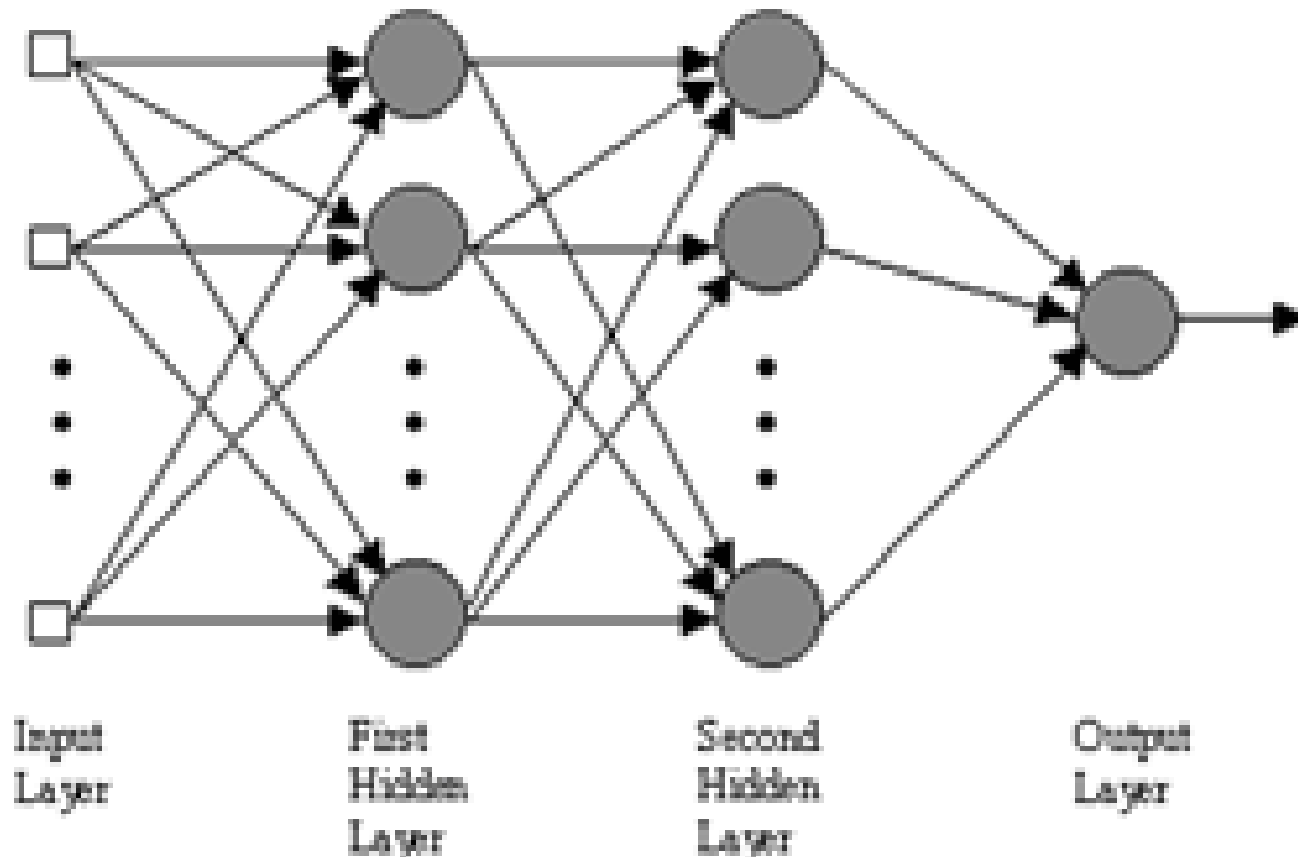
$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b) \quad \text{si } b = w_0$$



$$o(x_1, \dots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + \dots + w_n x_n > 0 \\ -1 & \text{otherwise.} \end{cases}$$



El perceptron multicapa o red neuronal



El perceptron multicapa

Como elegir los parámetros para algo tan complejo como la función implementada por un MLP?

Hay que buscar por algún método valores que hagan mínimo el error sobre los datos, que tengan el valor correcto en la salida

Descenso por el gradiente!

El perceptron multicapa

Descenso por el gradiente:

Definir la función error a minimizar

- Calcular el gradiente
- Mover los parámetros en la dirección que minimiza el error
- Iterar hasta converger

Heurísticas para mejorar la convergencia

Aprendizaje por instancias

Aproximar la función objetivo sólo localmente

Modelar sólo al momento de hacer predicciones

Ejemplo: k-vecinos

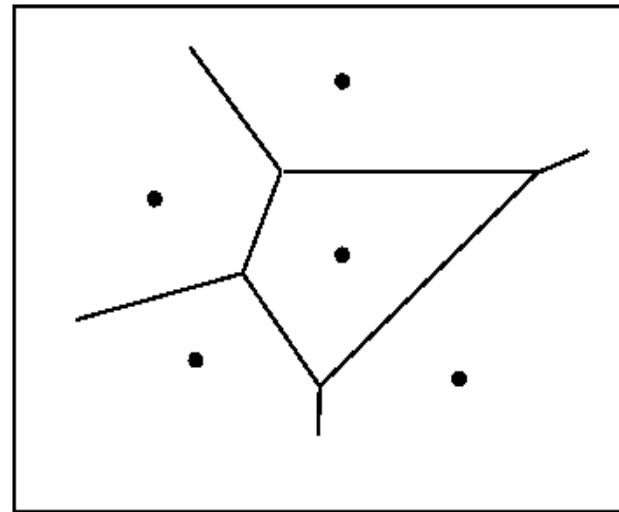
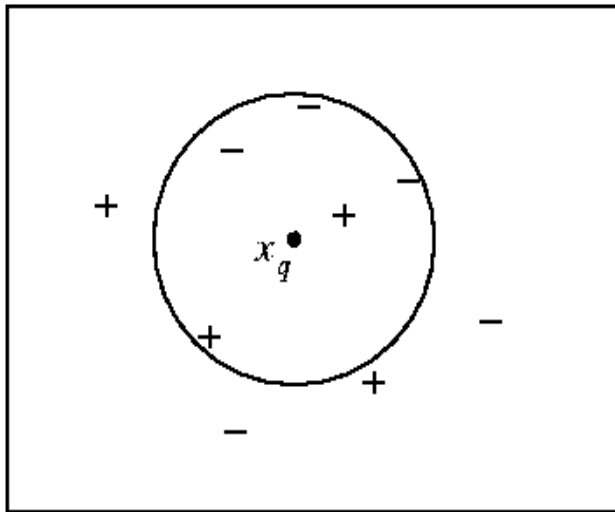
Aprendizaje por instancias

Aprendizaje con k-vecinos

- Dado un ejemplo X , buscar sus k puntos más cercanos con alguna distancia.
- Predecir usando sólo los k -vecinos:
 - Clasificación: Voto sobre la clase
 - Regresión: Promedio de los valores

Aprendizaje por instancias

Voronoi Diagram



Aprendizaje por instancias

k es un parámetro a elegir, no cualquier valor sirve, el resultado depende de k .

Cómo elegir k (o cualquier parámetro general)?

- Separar un conjunto de validación dentro de los datos que se usan para aprender.
- Evaluar los posibles valores sobre ese conjunto, elegir el que da menor error.

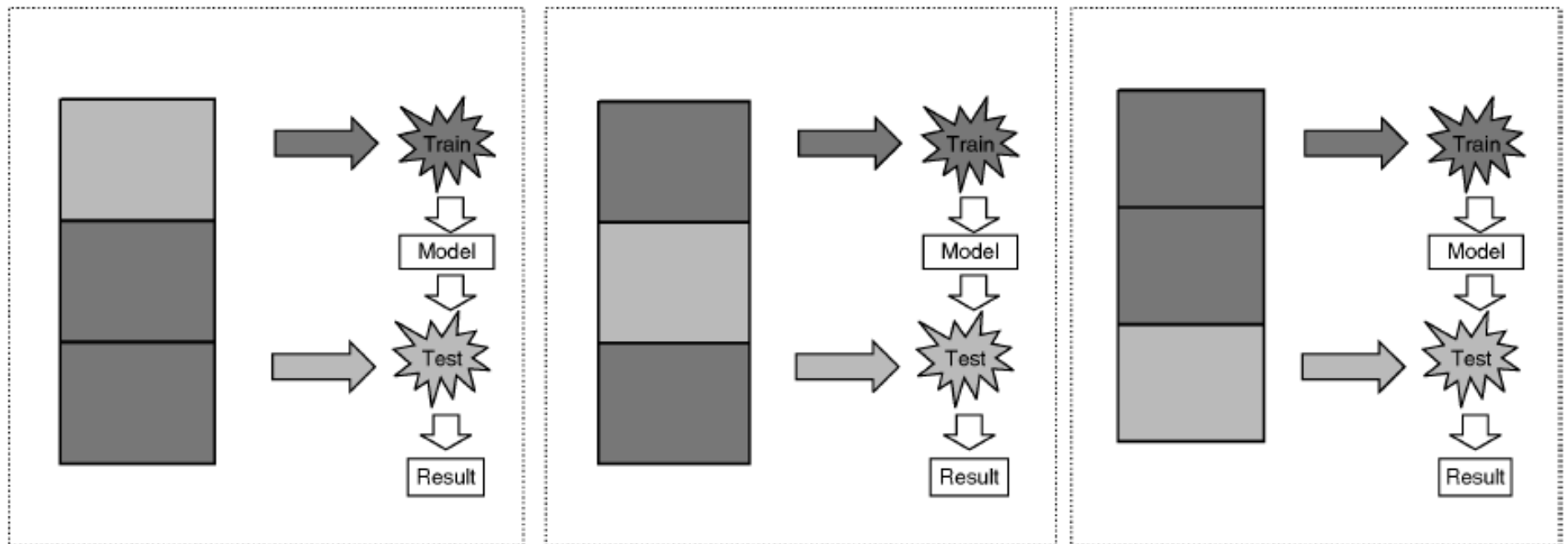
Estimación del error

Como estimar qué tan bueno es lo que aprendí?

La solución más simple es medirlo en una muestra de datos similar a la que voy a usar a futuro: Conjunto de test

Estimación del error

Si no tengo un conjunto de test, estimo con Cross-Validation



Cross-Validation. Figure 1. Procedure of three-fold cross-validation.

Límite: Leave-one-out

Como resolver un problema en ML

- Identificar el problema y conseguir conocimiento experto
- Conseguir datos, muchos datos!
- Elegir un método adecuado (o varios)
- Entrenar varios modelos con el conjunto de train, evaluarlos con el conjunto de validación, elegir el mejor
- Estimar el error con el conjunto de test