

Introducción a las Redes Neuronales

Víctor Mijangos

Facultad de Ingeniería

(Basado en presentacione de Dra. Ximena Gutierrez-Vasques)



UNAM

Acerca del curso

Objetivo general: Presentar las nociones esenciales para comprender el diseño y la implementación de redes neuronales.

Las **redes neuronales** son una familia de algoritmos de aprendizaje estadístico que actualmente han tenido un auge importante en múltiples aplicaciones. Entre ellas el Procesamiento de Lenguaje Natural.

Github: https://github.com/ElotlMX/Curso_redes



Temario

1. Motivación del aprendizaje de máquina
2. Introducción al aprendizaje estadístico
3. Regresión Lineal y Logística
4. Perceptrón
5. Redes Neuronales FeedForward
6. Optimización de parámetros en redes neuronales
7. Radial Basis Network
8. AutoEncoders
9. Redes Neuronales Recurrentes
10. Transformers



Motivación para el Aprendizaje de Máquina



Deducción e inferencia

La **deducción** genera conocimiento partiendo de lo general y dirigiéndose a lo particular.

Generalmente, se parte de **premisas**, de los que se derivan en una **conclusión**.

Un ejemplo típico de deducción son los silogismos:

A1: Todos los hombres son mortales

A2: Sócrates es un hombre

C: Sócrates es mortal

La **inducción** parte de lo particular y se dirige a lo general.

La inducción recolecta **datos**, de los cuales extrae **patrones** que nos llevan a concluir algo sobre los datos en general.

Mientras que el conocimiento deductivo es verdadero (siempre que las premisas sean verdaderas) el conocimiento inductivo es **probabilístico**.



Ejemplo de método deductivo

Una **gramática** es un ejemplo de un método deductivo:

$$S \rightarrow X \cdot S \cdot Y | X \cdot Y$$

$$X \rightarrow a$$

$$Y \rightarrow b$$

Esta gramática determina todos los patrones de la forma:

$$L = \{a^n b^n : n > 0\}$$

Sin embargo, no todos los problemas pueden ser tratados deductivamente.



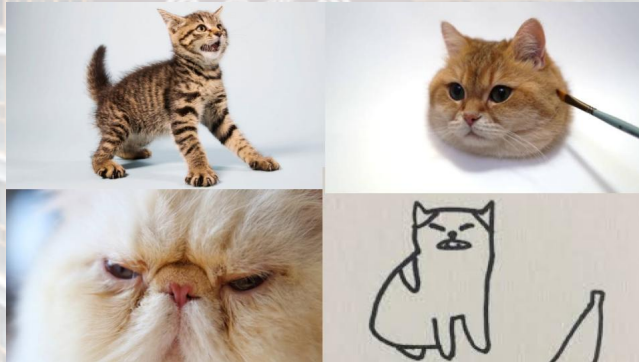
Necesidad de métodos inferenciales

- ▶ Existen tareas en las que determinar un algoritmo resulta una tarea sencilla.
- ▶ Sin embargo, existen tareas en que no es fácil determinar un algoritmo. Causas de esto son:
 - ▶ No existe *expertise* humano.
 - ▶ La solución puede cambiar con el tiempo o responder a casos particulares.
 - ▶ Hay tareas que, a pesar de realizarse de manera cotidiana, no somos capaces de describirlas.



Necesidad de métodos inferenciales

Imagínese una tarea en la que debe clasificarse un objeto como 'es gato' y 'no es gato'. ¿Qué tipo de algoritmo podría realizar esta tarea?



Necesidad de métodos inferenciales

Podemos preguntarnos ¿qué elementos caracterizan a un objeto? ¿Cómo podemos describirlo?

Words for **Cats**

Fluffy lap-monster or lean, mean hunter? Find the right words to describe your four-legged friend below.

| Adjectives | Nouns | Verbs | Negatives | Breeds | Phrases | Quotes |
|------------------|-------|---------------------------|-----------|----------------|---------|-------------------------|
| action | | companion | | leash | | running water |
| affection | | confirmation | | life companion | | scratcher |
| agility | | couch potato | | listener | | senses |
| alpha cat | | curiosity | | litter box | | show off |
| animal | | disposition | | lovebug | | sleep |
| animal shelter | | ears | | lover | | smell |
| antics | | fast learner | | markings | | snuggle bug |
| attention-seeker | | feline | | master | | snuggler |
| beast | | fluff ball | | mate | | source of entertainment |
| behavior | | four-legged family member | | meow | | stranger |

Necesitamos encontrar las características que mejor describan a un objeto.



Necesidad de métodos inferenciales

Una opción alternativa: Podemos dejar que la máquina observe una serie de ejemplos y extraiga las categorías relevantes.

Justificación: Los datos son abundantes y baratos, el *conocimiento* es caro y escaso.

Objetivo: Construir modelos generales a partir de ejemplos particulares (**inducción**).

Diremos, entonces, que buscamos que la máquina **aprenda** a partir de estos ejemplos.



¿A qué nos referimos con aprender?

Se busca construir sistemas computacionales que sean capaces de *mejorar en una tarea basados en la experiencia* [1].

Surge la pregunta:

¿Cómo podemos construir sistemas computacionales que mejoren en una tarea a partir de la experiencia? ¿Cuáles son las leyes fundamentales que gobiernan el “aprendizaje”?



Una aproximación al problema del aprendizaje

Se definen los siguientes elementos en el problema del aprendizaje:

1. Una tarea T (el problema al que nos enfrentamos).
2. Una medida de desempeño P (cómo evaluar qué también lo hace la máquina).
3. Experiencia E (datos, ejemplos).

Entonces, el problema de aprendizaje se puede definir como [1]:

Un programa de computadora se dice que aprende de la experiencia E con respecto a una tarea T y una medida de desempeño P , si su desempeño con respecto a T , medido con P , mejora con la experiencia E .



Máquinas que aprenden



Aprendizaje de máquina - Ejemplos

• Filtrar Spam

T: predecir qué correos son spam para un usuario determinado.

P: % de correos correctamente clasificados

E: Un conjunto de correos previamente etiquetados como spam/ no spam



• Robot driving

T: Manejar en la ciudad usando visión y sensores

P: distancia recorrida sin cometer un error

E: Una secuencia de imágenes y las acciones de manejo relacionadas (grabadas de un humanos manejando)



Aprendizaje de máquina - Ejemplos

- **Diagnóstico médico**

T: Sugerir un tratamiento dado los síntomas

P: % de predicciones de tratamiento que fueron correctas

E: Una base de datos, registros médicos que contengan los síntomas del paciente y la enfermedad diagnosticada.



- **Reconocimiento de escritura**

T: Reconocer y clasificar palabras dentro de una imagen

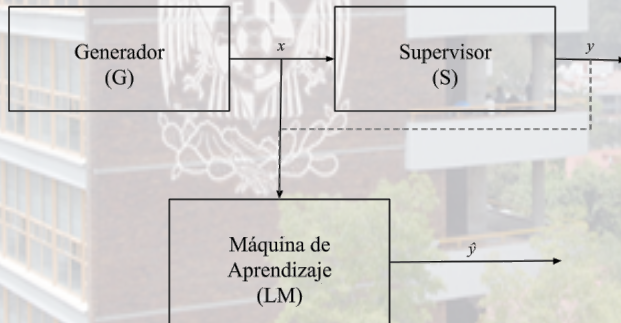
P: % de palabras correctamente clasificadas

E: Una base de datos con imágenes de palabras escritas a mano, cada una asociada a una etiqueta indicando qué palabra es.



Modelo general de aprendizaje a partir de ejemplos

A partir de los elementos citados, se puede proponer un modelo general de aprendizaje a partir de ejemplos [2]:



Modelo general de aprendizaje a partir de ejemplos

Este modelo cuenta con los siguientes elementos:

1. **Generador:** Se encarga de generar la experiencia (ejemplos) a partir de una distribución $p(x)$.
2. **Supervisor:** Determina la clase a la que pertenece un ejemplo. Se trata de conocimiento experto.
3. **Máquina de Aprendizaje:** Es una función que toma un ejemplo de entrada y le asigna una clase con base en una distribución $q(x)$.



Modelo general de aprendizaje a partir de ejemplos

El objetivo de la Máquina de Aprendizaje es, entonces, encontrar una distribución (estimada a partir de los ejemplos) que determine una clase \hat{y} a un ejemplo x , que sea lo más cercana a la clase y asignada por el Supervisor.

Para hacer esto, la máquina de aprendizaje requiere definir una **función de desempeño**.

Llamaremos a esta función, “función de Riesgo”, y, en forma general, la entenderemos como:

$$R(f) = \mathbb{E}[L(y, \hat{y})]$$

En este caso, f es la LM tal que $f(x) = \hat{y}$.



Algunos tipos de aprendizaje

Según ciertas características, podemos clasificar los modelos de aprendizaje:

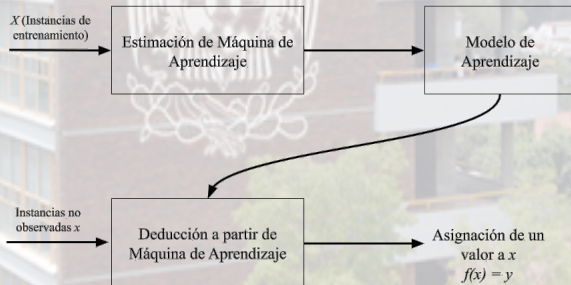
1. **Supervisado:** Cuando información de S entra en la LM para aprender f .
 - 1.1 **Regresión:** Cuando se aprende una función $f : X \rightarrow \mathbb{R}$.
 - 1.2 **Clasificación:** Cuando se aprende una función $f : X \rightarrow \{c_1, c_2, \dots, c_n\}$.
2. **No-supervisado:** Cuando la LM no tiene información de S para aprender f .
 - 2.1 **Clustering:** Cuando se aprende una función $f : X \rightarrow \{c_1, c_2, \dots, c_n\}$.
3. **Semi-supervisado:** Se trata de modelos híbridos (supervisados con no-supervisado).



Elementos de un problema de aprendizaje

Durante una tarea de aprendizaje, se tienen dos etapas:

1. **Entrenamiento:** Se estima la función f a partir de una muestra de ejemplos.
2. **Evaluación:** Se determina la capacidad de generalizar del modelo de aprendizaje.



Elementos de un problema de aprendizaje

Para llevar a cabo el entrenamiento y la evaluación, generalmente, se dividen los datos en 3 subconjuntos:

1. **Entrenamiento:** El subconjunto de datos que servirá para estimar el modelo de aprendizaje (70 %).
2. **Validación:** Es un subconjunto de datos que se prueba constantemente para ajustar ciertos requerimiento del modelo de aprendizaje (10-15 %).
3. **Evaluación:** Es el subconjunto a partir del cuál se determina la capacidad de generalizar del modelo estimado (15-20 %).



Algunos ejemplos

Algunos ejemplos de **algoritmos de aprendizaje supervisado** son:

1. Árboles de decisión y bosques aleatorios.
2. k Nearest Neighborhoods (KNN).
3. Bayes Naïve.
4. Support Vector Machine (SVM).

Algunos ejemplos de **algoritmos de aprendizaje no-supervisado** son:

1. k -means.
2. Clustering jerárquico.



Lecturas y herramientas

Libros:

- ▶ Tom Mitchel (1997). *Machine Learning*. McGraw Hill.
- ▶ Ethem Alpaydin (2010). *Introduction to Machine Learning*, 2d edition. The MIT Press.
- ▶ Stephen Marsland (2014). *Machine Learning: An Algorithmic Perspective*. CRC Press.
- ▶ Shai Shalev-Schwartz y Shai Ben-David (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Paqueretría:

- ▶ Sklearn (Python): Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011



References



Tom M Mitchell.

Machine learning and data mining.

Communications of the ACM, 42(11), 1999.



Vladimir Vapnik.

Statistical learning theory. 1998, volume 3.

Wiley, New York, 1998.



The End

