

Durante la cursa vamos a utilizar las siguientes herramientas:

- Lenguaje de programación:
  - Python 3.8
  - Herramienta pip para instalar librerías de código y dependencias
- Librerías de código:
  - Numpy 1.18 y SciPy 1.5
  - PyTorch
- Consola interactiva de Python:
  - iPython y Google Colab
- Herramientas:
  - PyTest para tests, GitHub para repositorios y uWSGI para servidor web
- IDE recomendado:
  - VSCode: <https://code.visualstudio.com/>



La bibliografía es solo a modo de sugerencia y no será obligatorio el uso de dicho material. El curso está diseñado para ser completamente autocontenido.

- Deep Learning | Ian Goodfellow | <https://www.deeplearningbook.org/>



## Introduction to Deep Learning

- Loss function
  - Error cuadrático medio
  - Binary Cross Entropy
  - Cross Entropy
  - KL-Divergence
- Model
  - Non-linear neural network
    - Layers: Linear layer | Convolution layer | Recurrent layer
    - Activation Functions: Sigmoid, Softmax, ReLu, Tanh
- Optimization
  - Algorithms
    - Gradient Descent
  - Stochastic Gradient Descent
  - Mini-Batch Gradient Descent



## Qué estamos “Aprendiendo”?

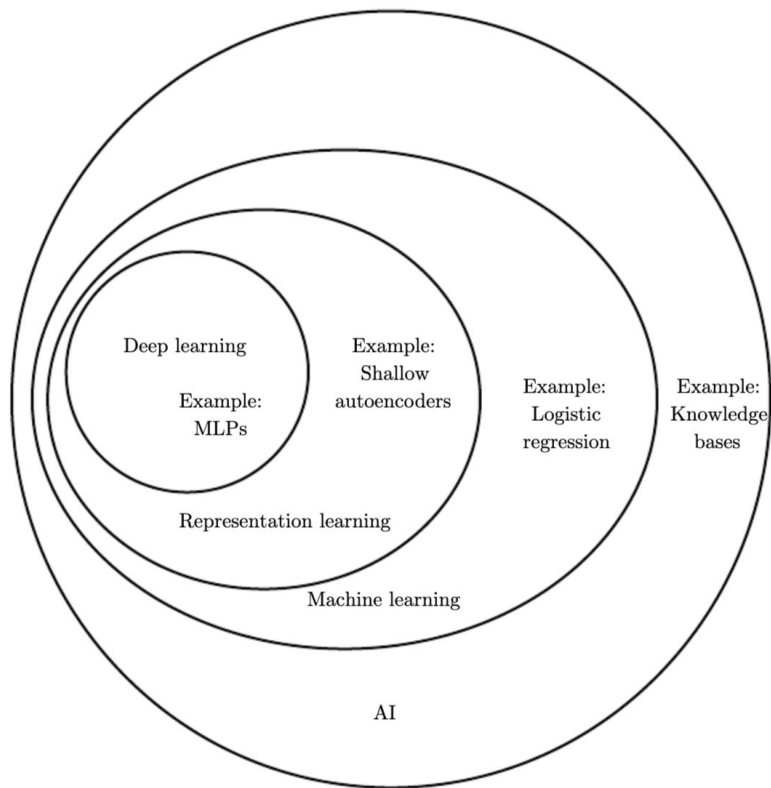
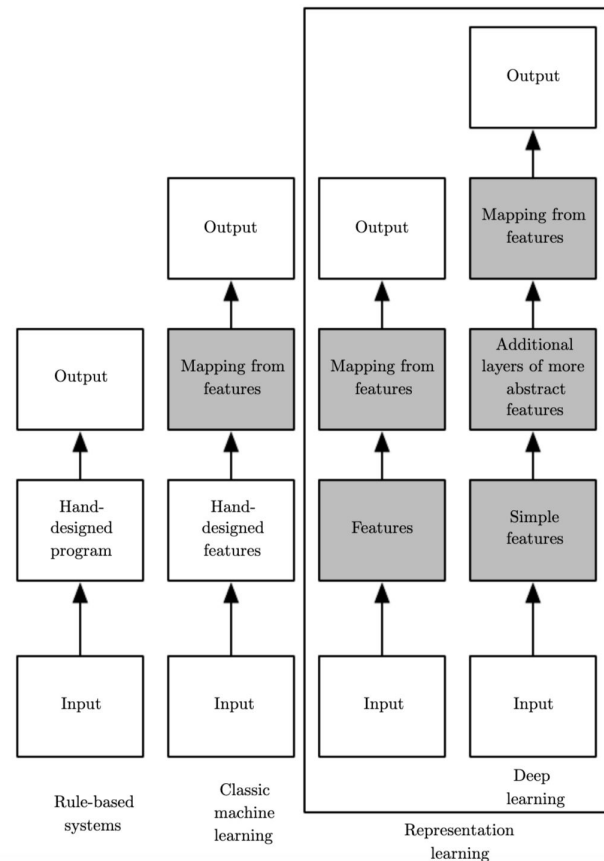
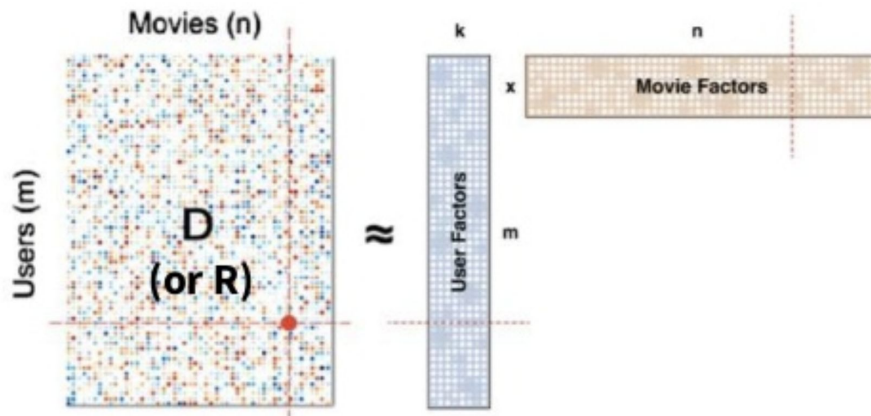


Diagrama de Venn de algoritmos



Bloques que se aprenden en Deep Learning

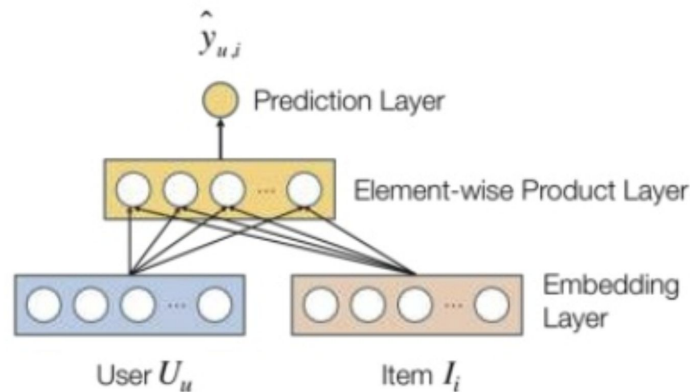
## Qué estamos “Aprendiendo”? Ejemplo en Sistemas de Recomendación



$$\nabla \left( \sum_{u,i: R_{u,i} \neq 0} (R_{u,i} - U_u \cdot I_i)^2 \right) = \vec{0}$$

- El objetivo es aprender todos los  $U_u$  y  $I_i$  que optimizan el error cuadrático medio.
- $U_u$  e  $I_i$  son representaciones densas de usuarios e ítems.
- $U_u$  e  $I_i$  son conocidos con el nombre de embeddings, representaciones densas.
- Los embeddings también los podemos aprender con una red neuronal.

## Qué estamos “Aprendiendo”? Ejemplo en Sistemas de Recomendación

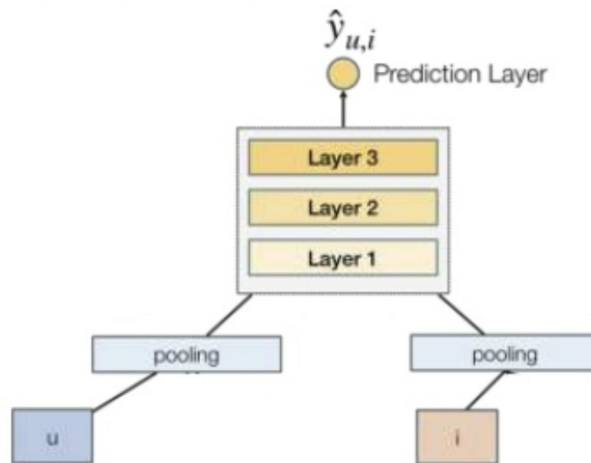


$$\nabla \left( \sum_{u,i: R_{u,i} \neq 0} (R_{u,i} - U_u \cdot I_i)^2 \right) = \vec{0}$$

- Una red neuronal muy simple para resolver el problema es la que vemos en la imagen.
- Lo que aprendemos son los embeddings de los usuarios y los embeddings de los ítems.
- La operación es el producto interno entre  $U_u$  e  $I_i$ .
- El output es simplemente una neurona mas su funcion de activacion.
- Podríamos usar Binary Cross Entropy como función de costo.

## Qué estamos “Aprendiendo”? Ejemplo en Sistemas de Recomendación

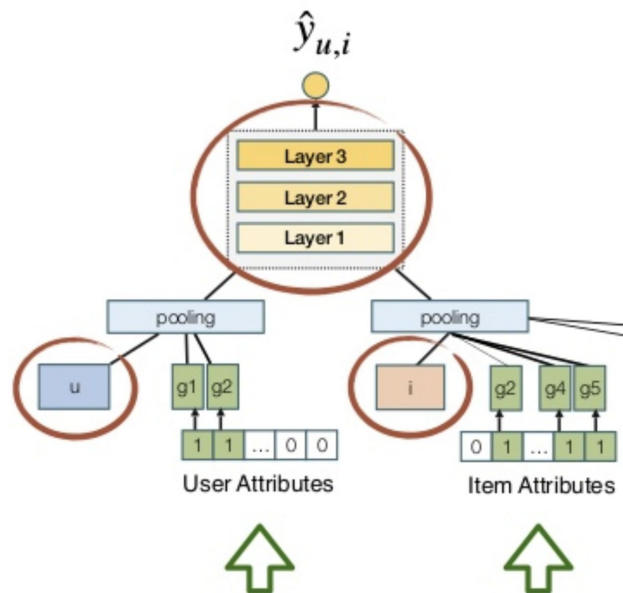
$$\nabla \left( \sum_{u,i: R_{u,i} \neq 0} (R_{u,i} - U_u \cdot I_i)^2 \right) = \vec{0}$$



- Una red neuronal mas compleja es la que vemos en la imagen
- Lo que aprendemos son los embeddings de los usuarios y los embeddings de los ítems.
- La operación es el producto interno entre  $U_u$  e  $I_i$ .
- El output es simplemente una neurona mas su funcion de activacion.
- Podríamos usar Binary Cross Entropy como función de costo.

## Qué estamos “Aprendiendo”? Ejemplo en Sistemas de Recomendación

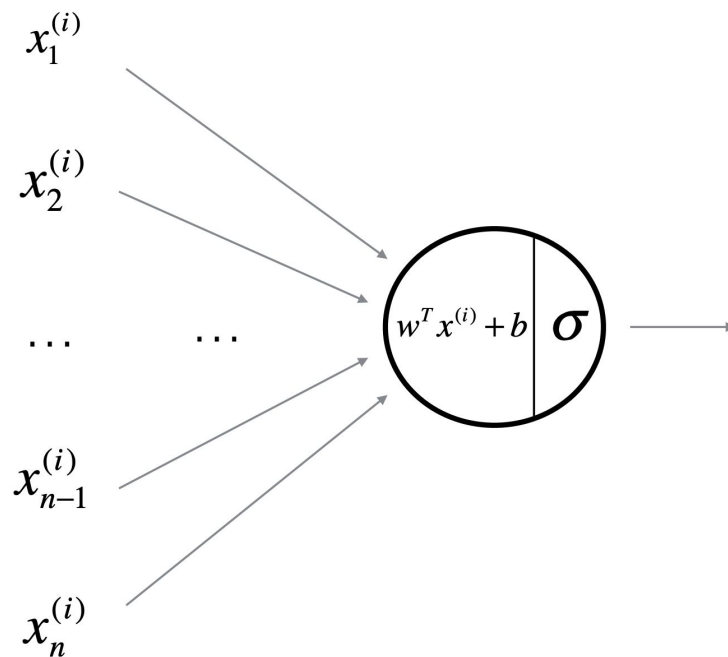
$$\nabla \left( \sum_{u,i: R_{u,i} \neq 0} (R_{u,i} - U_u \cdot I_i)^2 \right) = \vec{0}$$



- Círculos rojos -> lo que aprendemos con Gradient Descent
- Flechas verdes -> dataset de entrada (input)



## Unidad básica - Neurona



## Unidad básica - Layer

