

## REDES NEURONALES ARTIFICIALES – DEEP LEARNING

SUPRESION DEL GRADIENTE – REGULARIZACIÓN – DROPOUT –  
INTRODUCCIÓN A DEEP LEARNING

LAURA DIAZ DÁVILA

# REDES NEURONALES

## APRENDIZAJE SUPERVISADO – UNIDIRECCIONALES – MULTICAPA

### EL PERCEPTRON GENERALIZADO

- 1. MLP BACK PROPAGATION**
- 2. DEEP LEARNING**



Pascalina (1642)

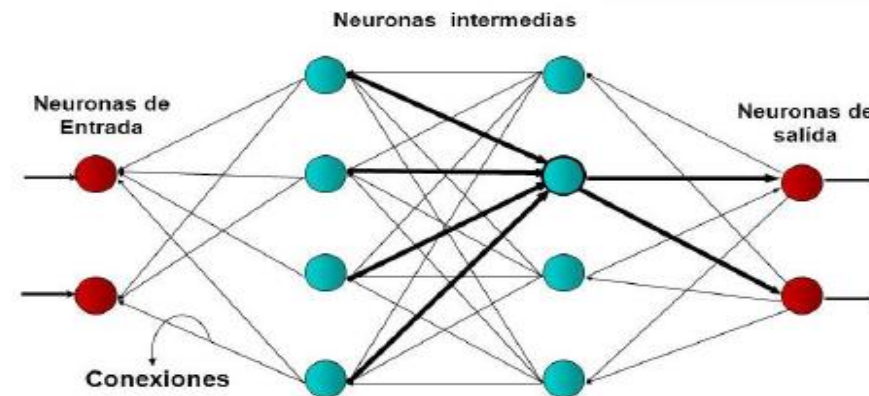
AUTOMATIZAMOS  
PROCESOS MENTALES

ESPIRITUAL

ETÉREO

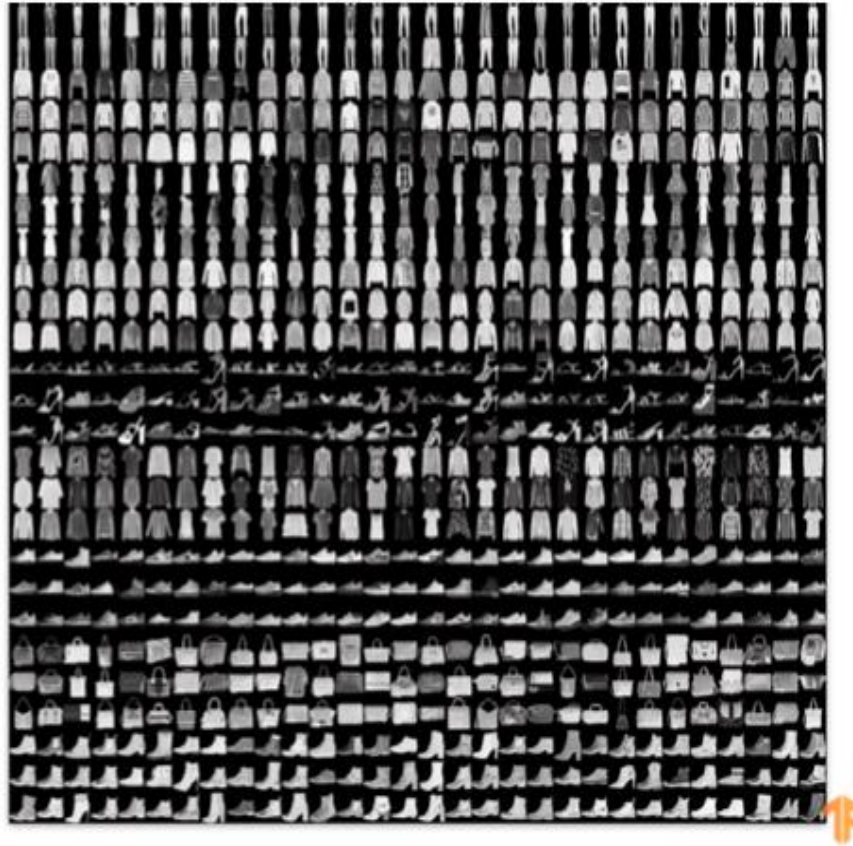
ANIMAMOS OBJETOS  
INHERTES, MÁQUINAS

CAJAS CADA VEZ MÁS  
NEGRAS

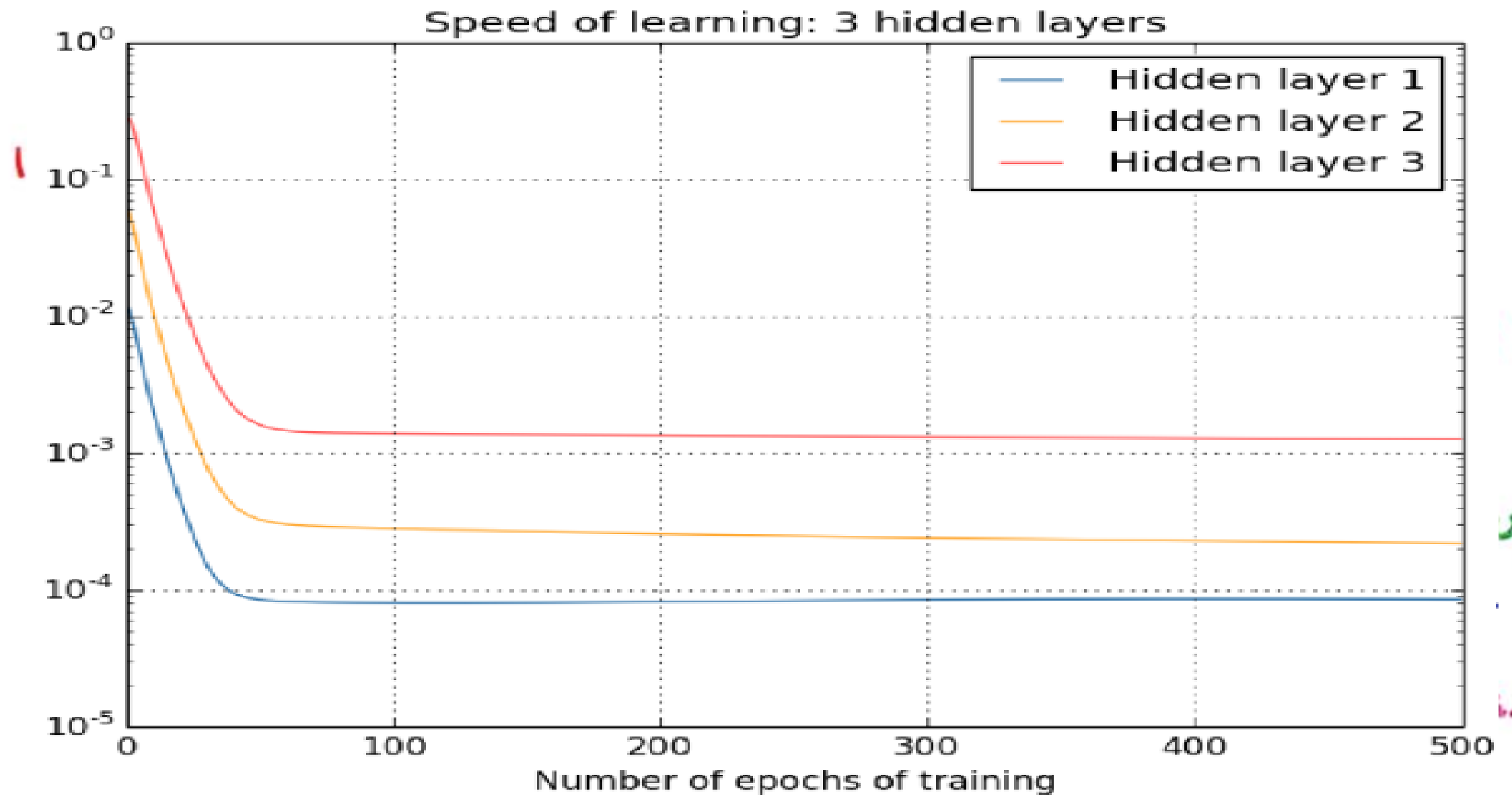


## Fashion MNIST

- 70k Images
- 10 Categories
- Images are 28x28
- Can train a neural net!



```
model = keras.Sequential([  
    keras.layers.Flatten(input_shape=(28, 28)),  
    keras.layers.Dense(128, activation=tf.nn.relu),  
    keras.layers.Dense(10, activation=tf.nn.softmax)  
])
```

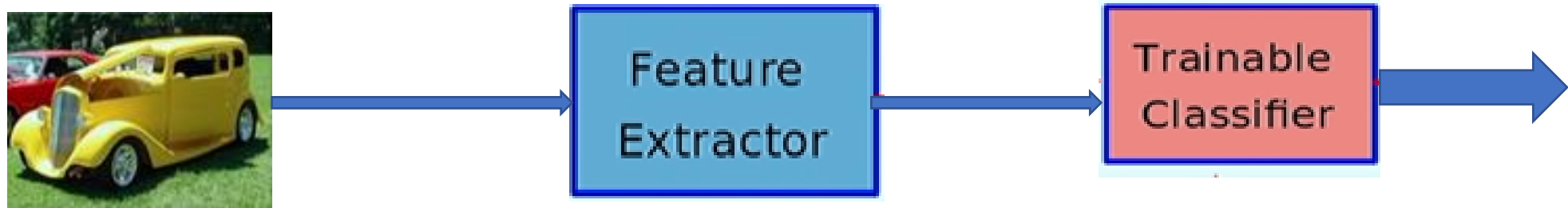


rij

late

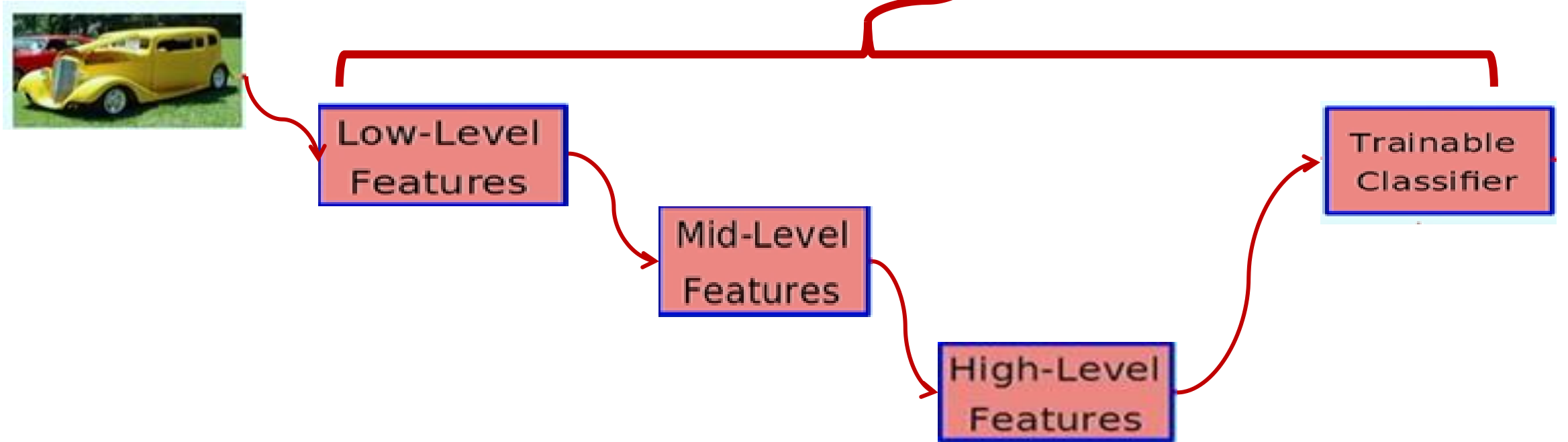
# Redes Neuronales Artificiales

EL PARADIGMA ESTÁNDAR DEL APRENDIZAJE DE  
PATRONES: MACHINE LEARNING “TRADICIONAL”

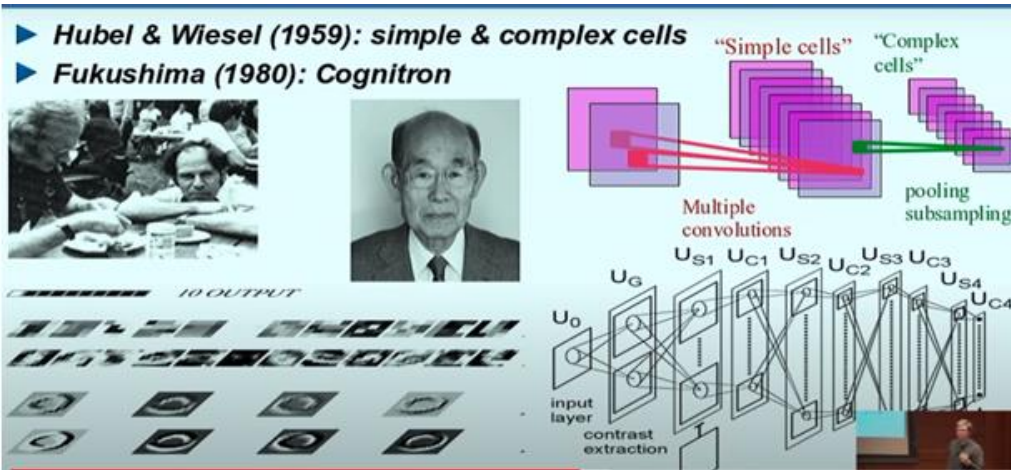


# ASÍ TRABAJA DEEP LEARNING

*ASÍ ENTRENAN, ASÍ APRENDEN, AUTOMATIZANDO MÁS PROCESOS MENTALES*



# CONVOLUTIONAL NEURAL NETWORKS



ARQUITECTURA FEED- FORWARD

APRENDIZAJE SUPERVISADO

IMITAN EL SISTEMA VISUAL DE LOS MAMÍFEROS

FUNCIONAN MUY BIEN PARA IMÁGENES

YANN LE CUN (1998) BACKPROPAGATION

APRENDEN EN BASE A EJEMPLOS

FILTRAN LAS IMÁGENES DE MODO QUE SUS CARACTERÍSTICA EMERJAN.

Fuente: Conferencia de Epistemología de Deep Learning – Yann Le Cun -2019

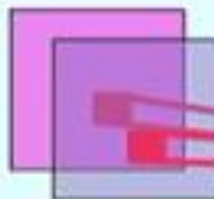
SU “MAGIA” CONSISTE EN QUE APRENDEN, A PARTIR DE MUCHOS EJEMPLOS, “EL FILTRO ADECUADO”. A ESTO SE LE DENOMINA “EXTRACCIÓN DE CARACTERÍSTICAS”



# CONVOLUTIONAL NEURAL NETWORKS

► Hubel & Wiesel (1959): simple & complex cells

► Fukushima (1980): Cognitron



10 OUTPUT



```
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

## Ecosistema de TensorFlow

TensorFlow ofrece una colección de flujos de trabajo para desarrollar y entrenar modelos mediante Python o JavaScript, y poder implementarlos con facilidad en la nube, de forma local, en el navegador o en el dispositivo, más allá del lenguaje que se use.

<https://www.tensorflow.org/learn>

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(64, (3,3), activation='relu',
                           input_shape=(28, 28, 1)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(10, activation='softmax')
])
```

Fuente: [https://www.youtube.com/watch?v=x\\_VrgWTKkiM](https://www.youtube.com/watch?v=x_VrgWTKkiM)

¡GRACIAS!