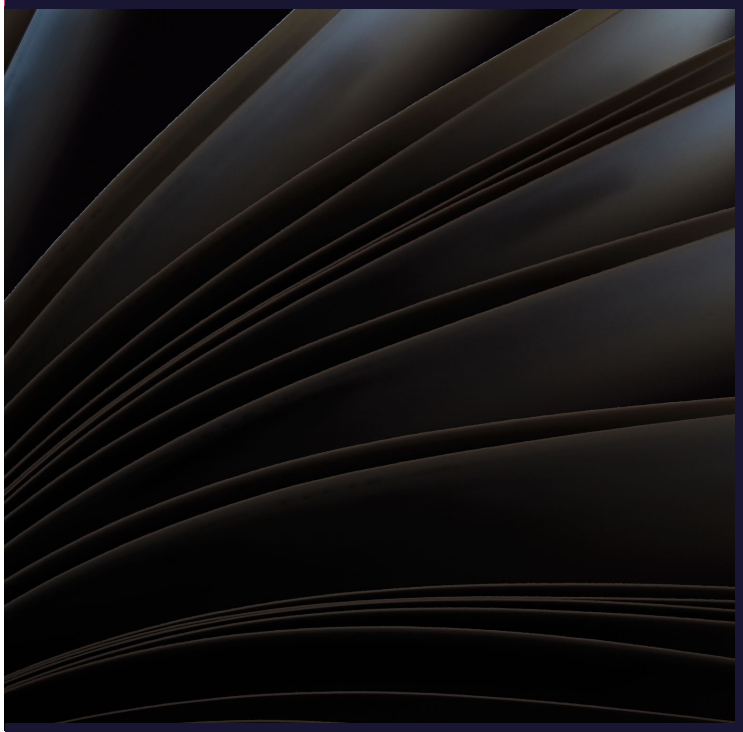


INTRODUCTION TO DEEP LEARNING

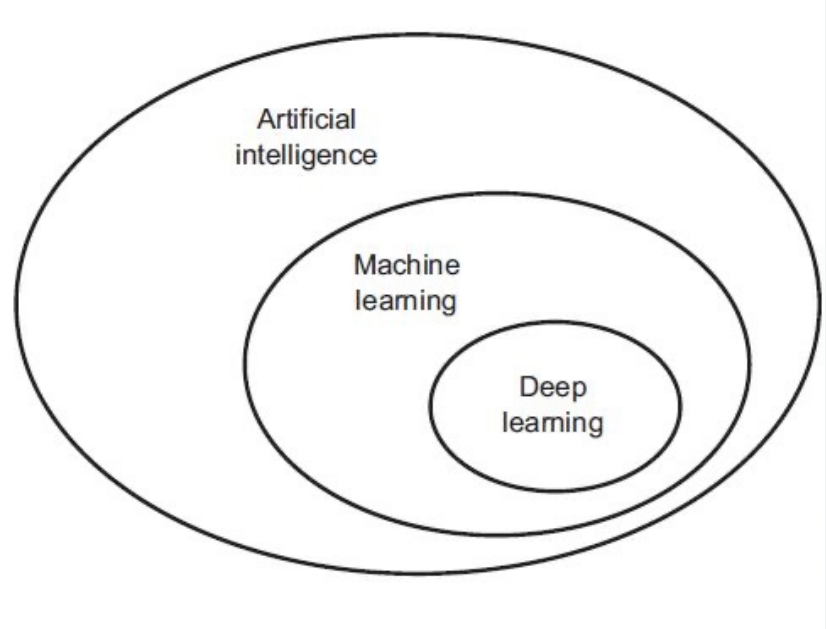
01211433 Vision and Control of Industrial Robots

Dr.Varodom Toochinda

Outline

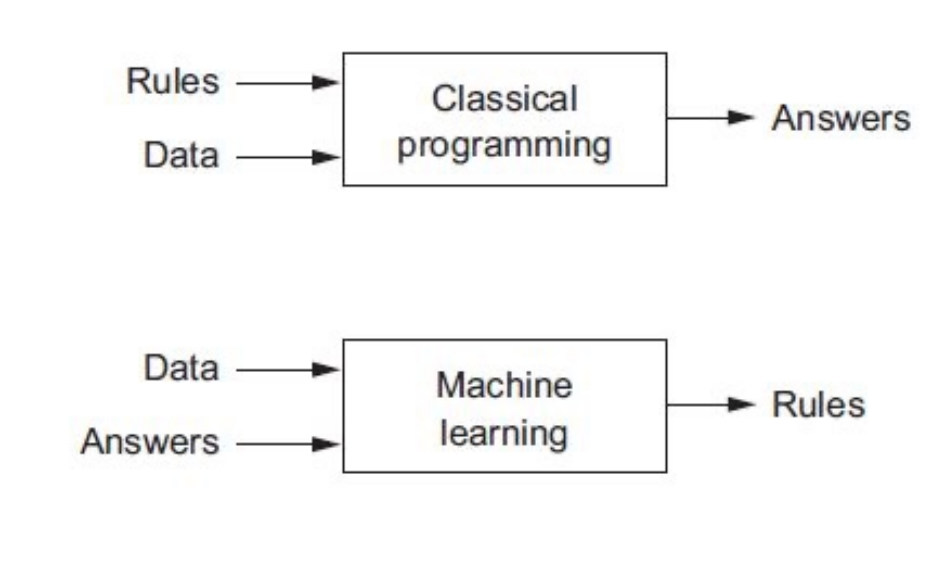
- Fundamentals of deep learning
- ML programming paradigm
- Deep learning structure
- Gradient descent algorithm
- Common NN architectures
- Deep learning development with tensorflow

What is Deep Learning?



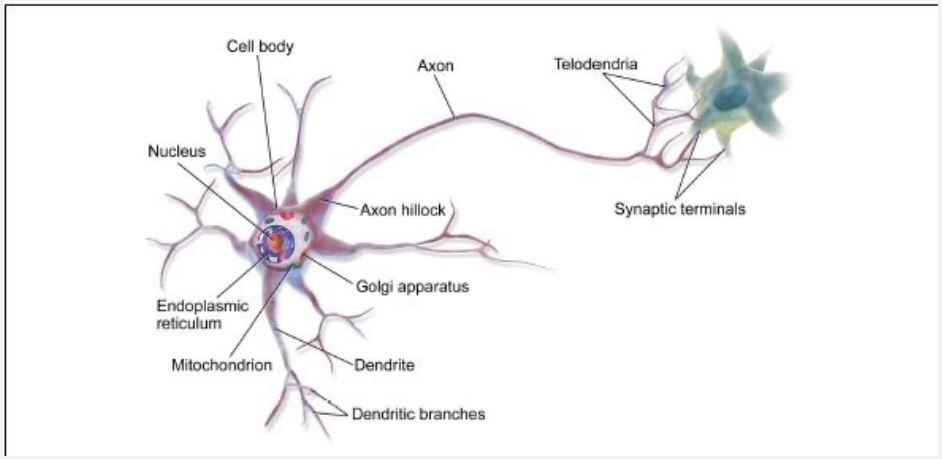
*From Deep Learning with Python, Francois Chollet, 2018

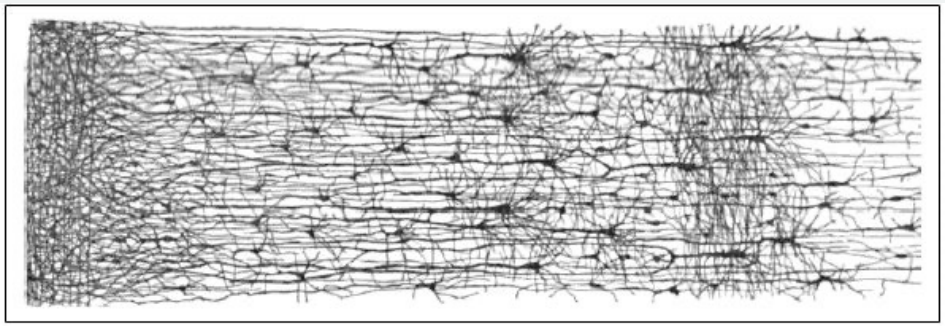
New programming paradigm in ML



*From Deep Learning with Python, Francois Chollet, 2018

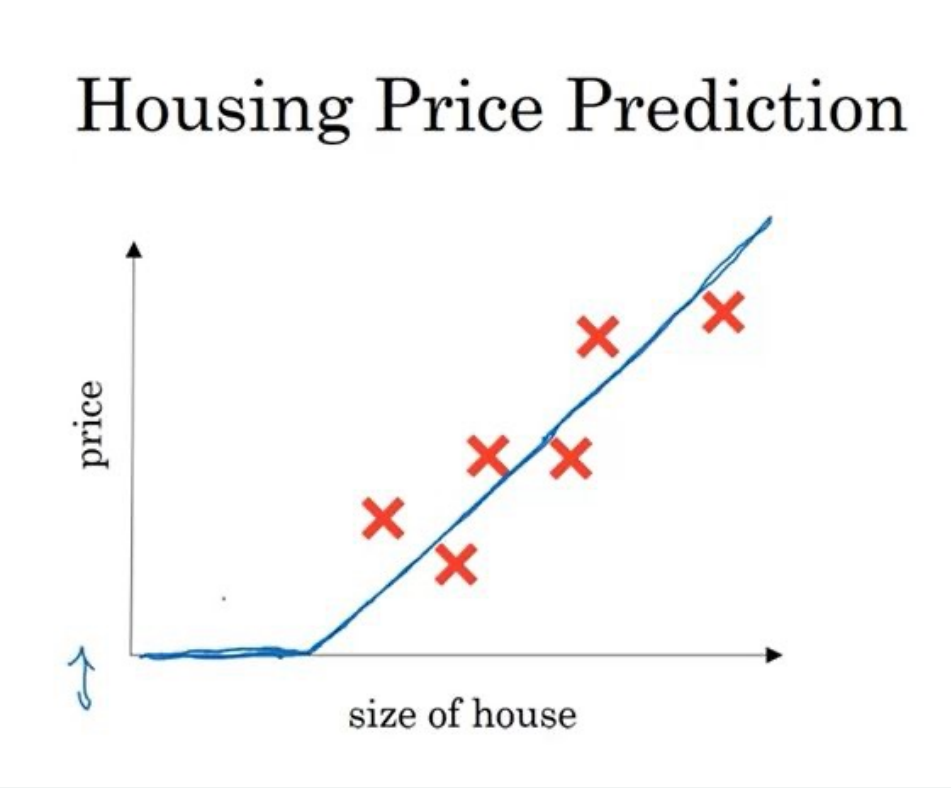
Motivation from biological neurons



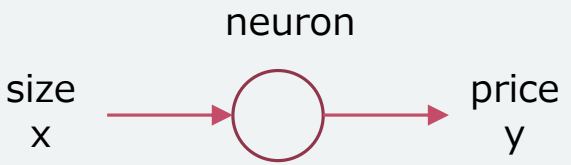


*From Hand on Machine Learning with Scikit Learn, Keras, and TensorFlow, Aurelien Geron. 2019

Simple Idea

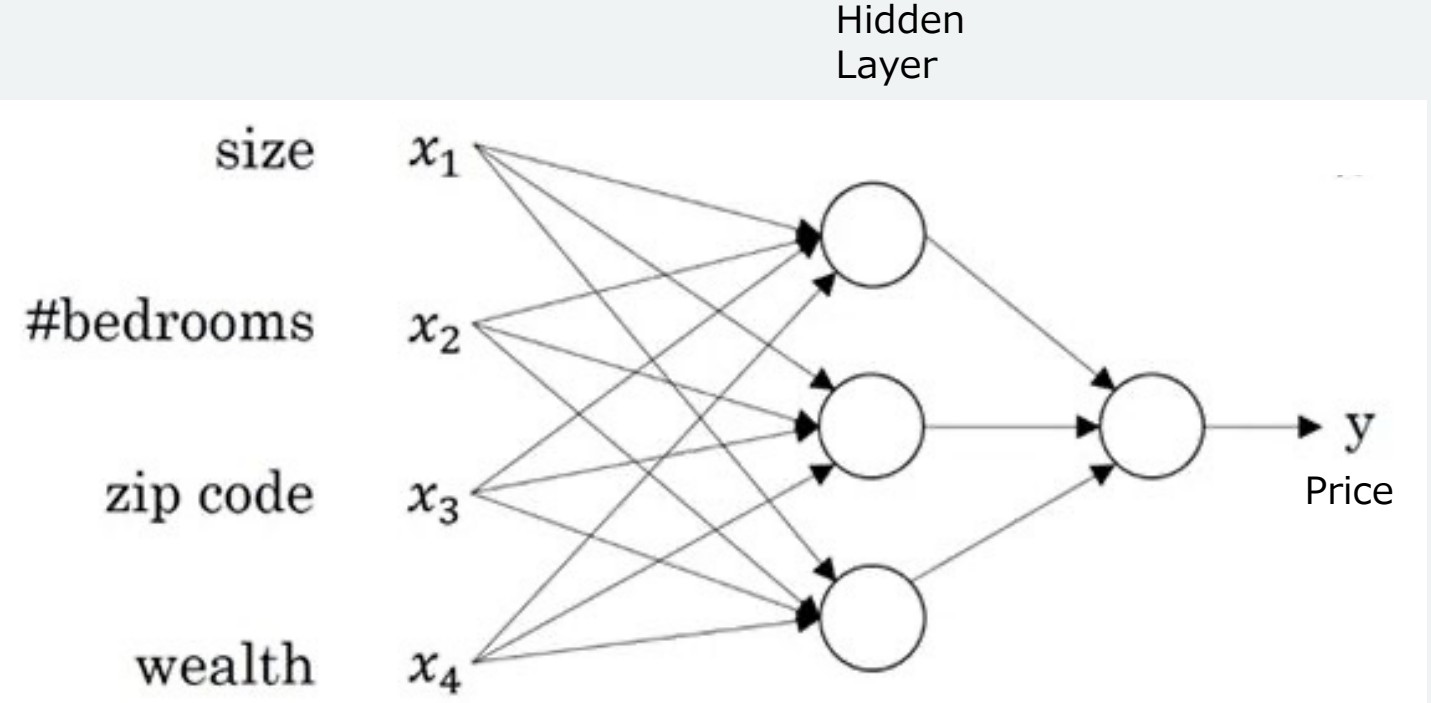


ReLu = Rectified Linear Unit

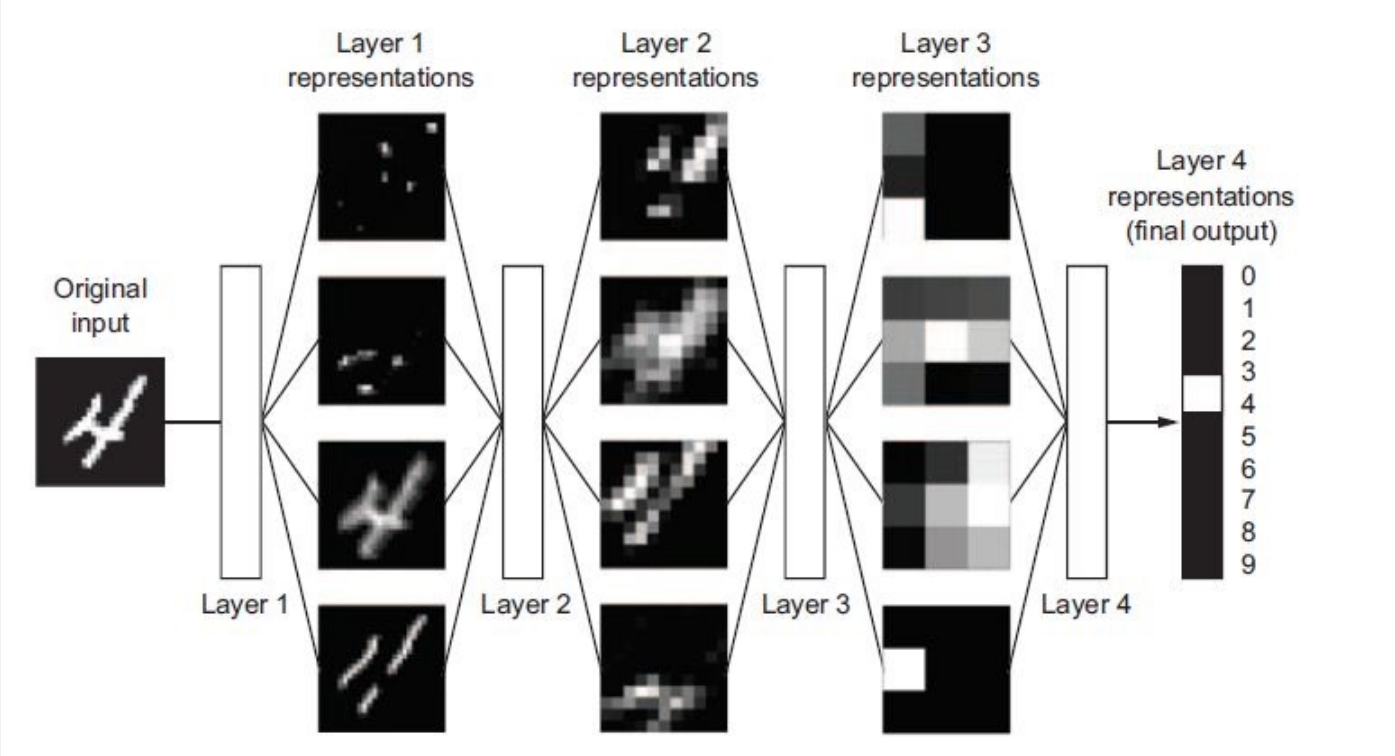


*Neural Network and Deeplearning course, deeplearning.ai, Coursera

Input Features

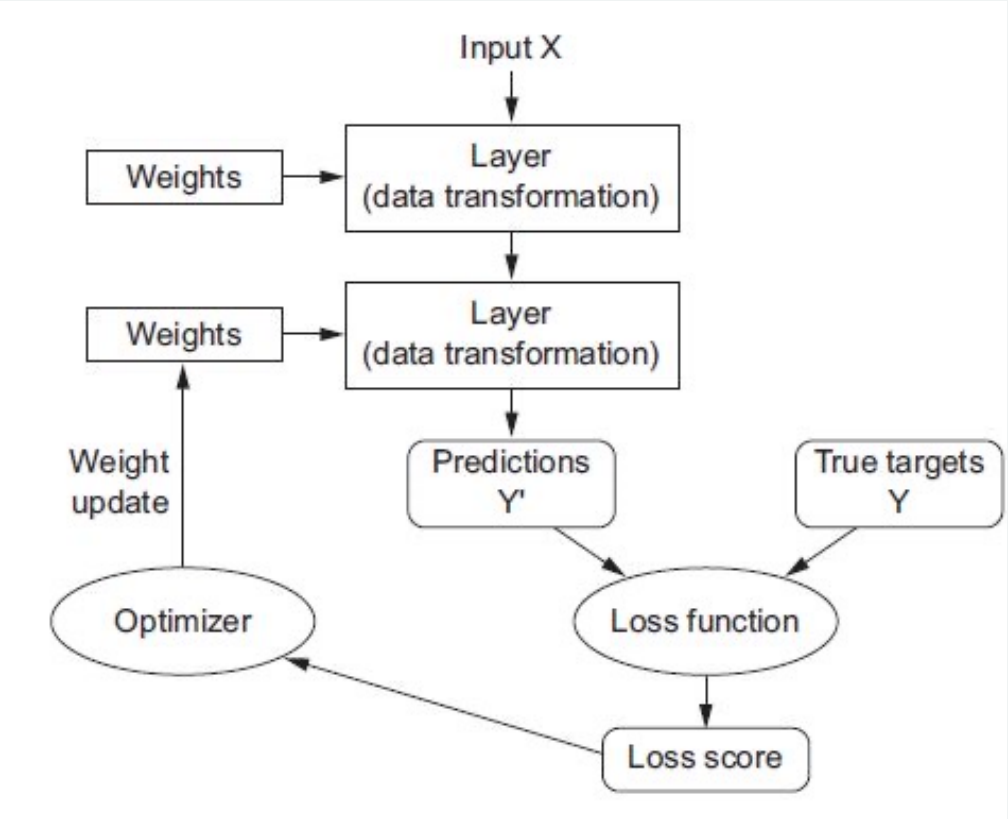


Deep neural network for digit classification



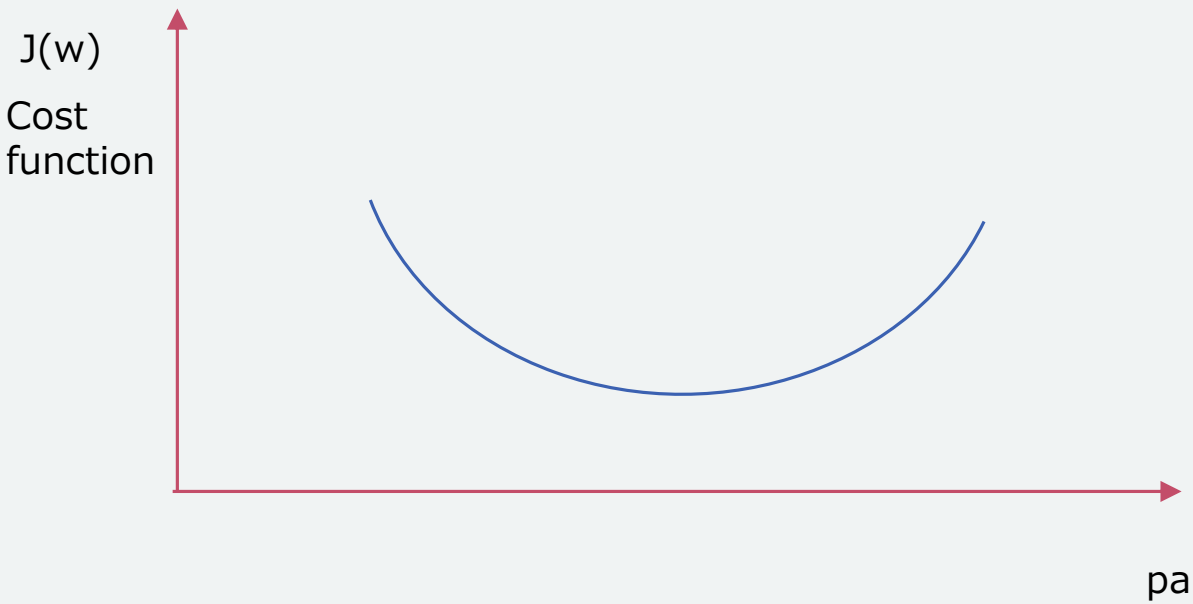
*From Deep Learning with Python, Francois Chollet, 2018

Deep learning structure



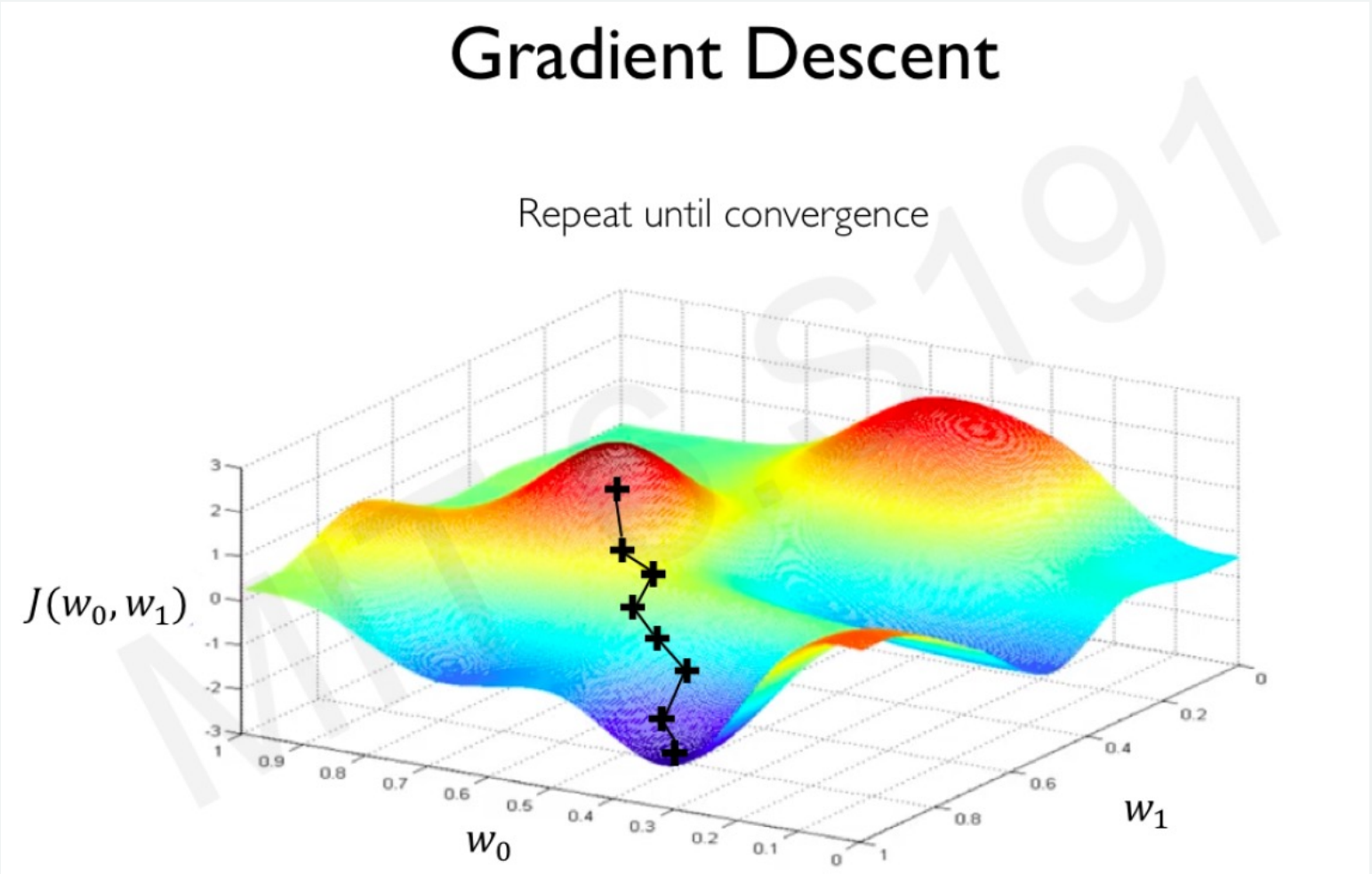
*From Deep Learning with Python, Francois Chollet, 2018

Gradient descent



Learning rate

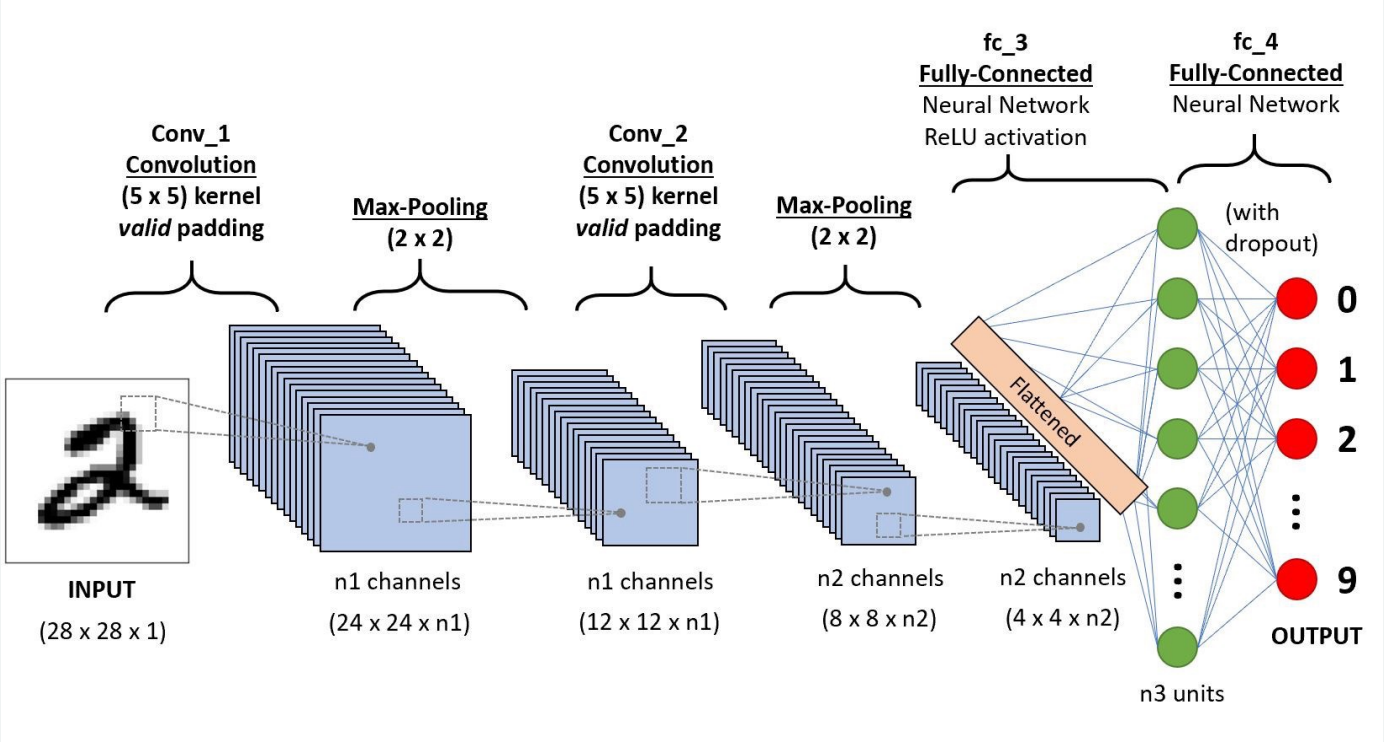
$$W := W - \alpha \frac{dJ(W)}{dw}$$



*From MIT 6S191 Deep Learning Lecture 1

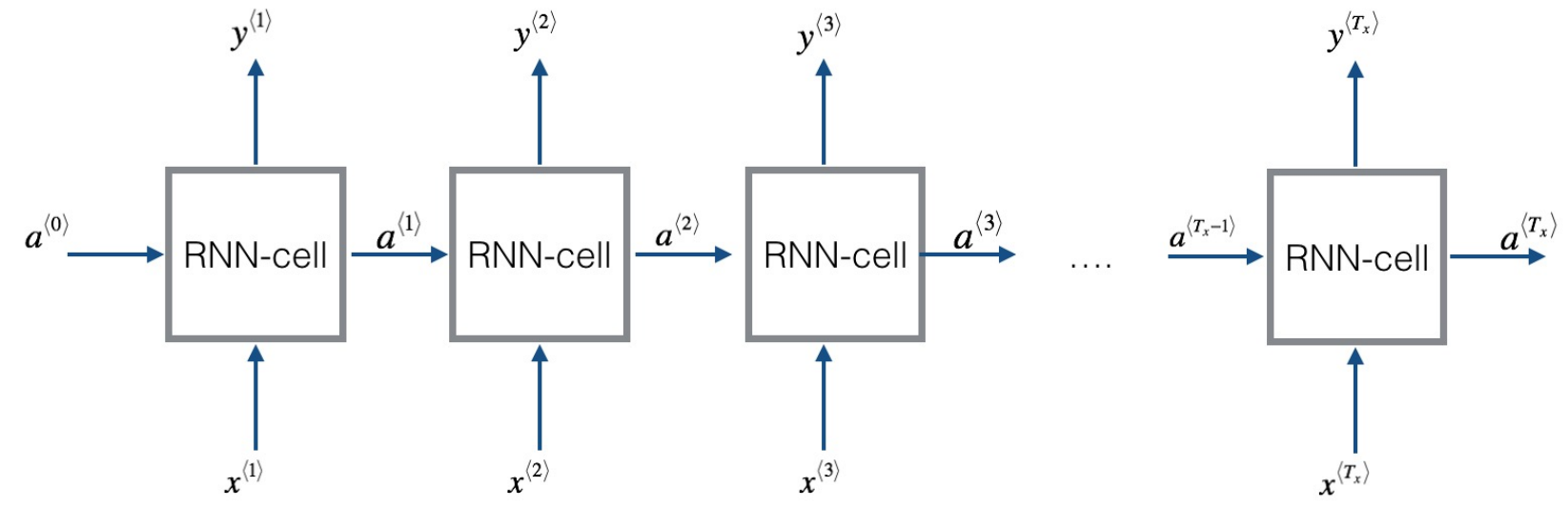
SOME COMMON NEURAL NETWORK ARCHITECTURES

Convolutional Neural Networks (CNN)

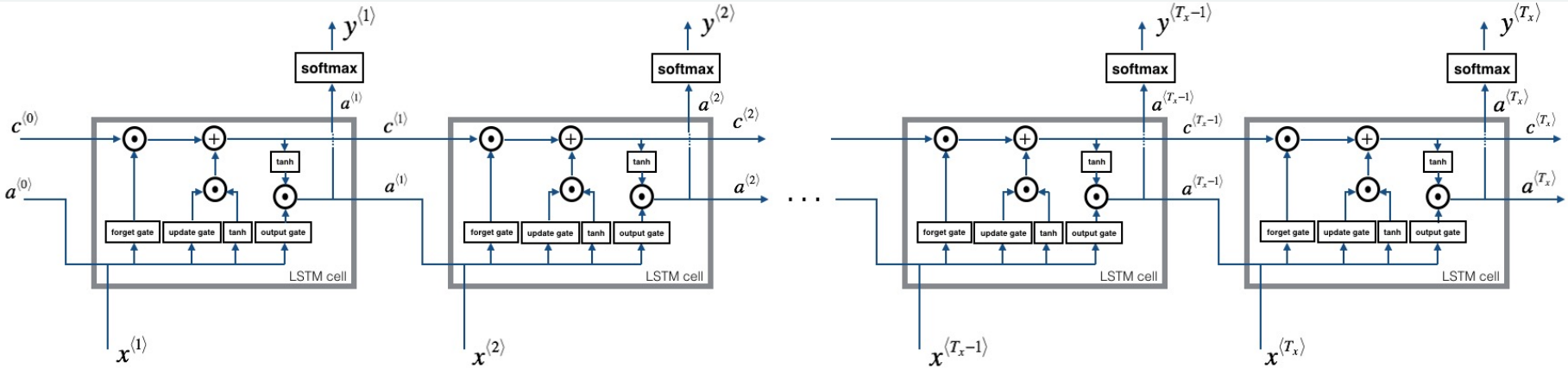


https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

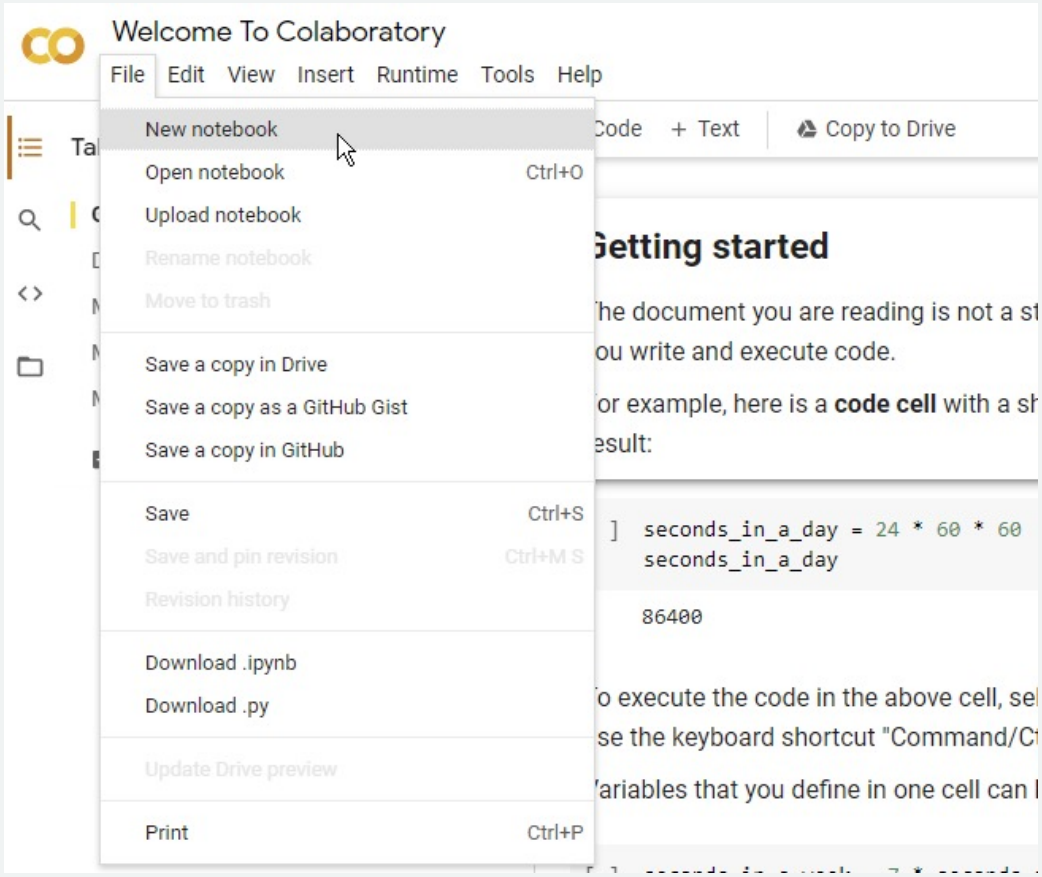
Recurrent Neural Networks (RNN)



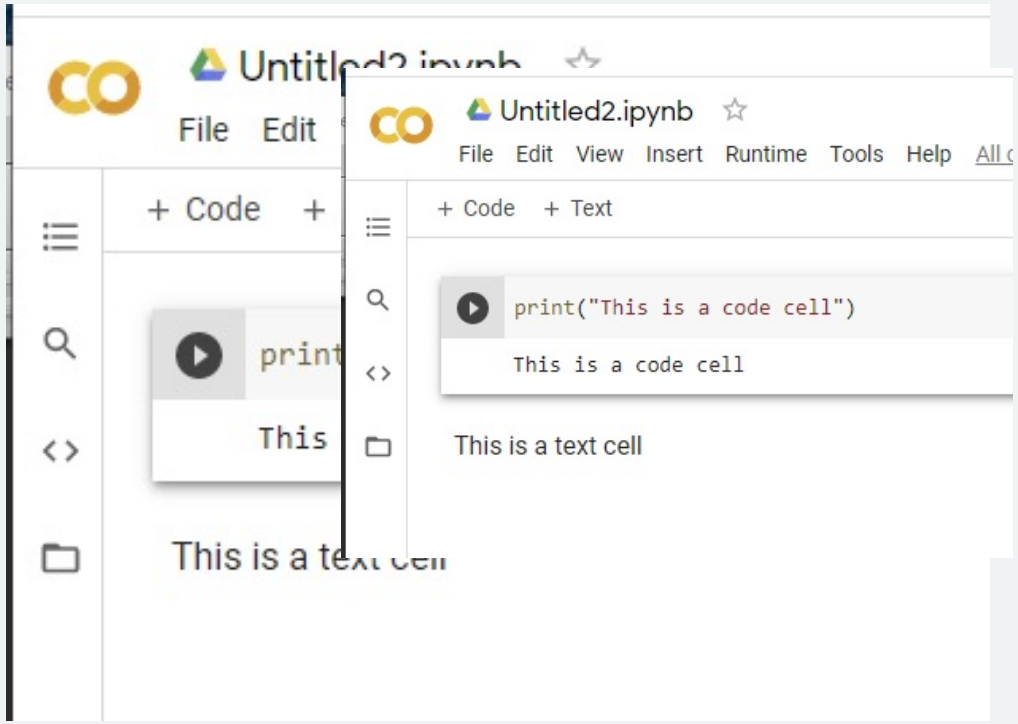
Long Short Term Memory (LSTM)



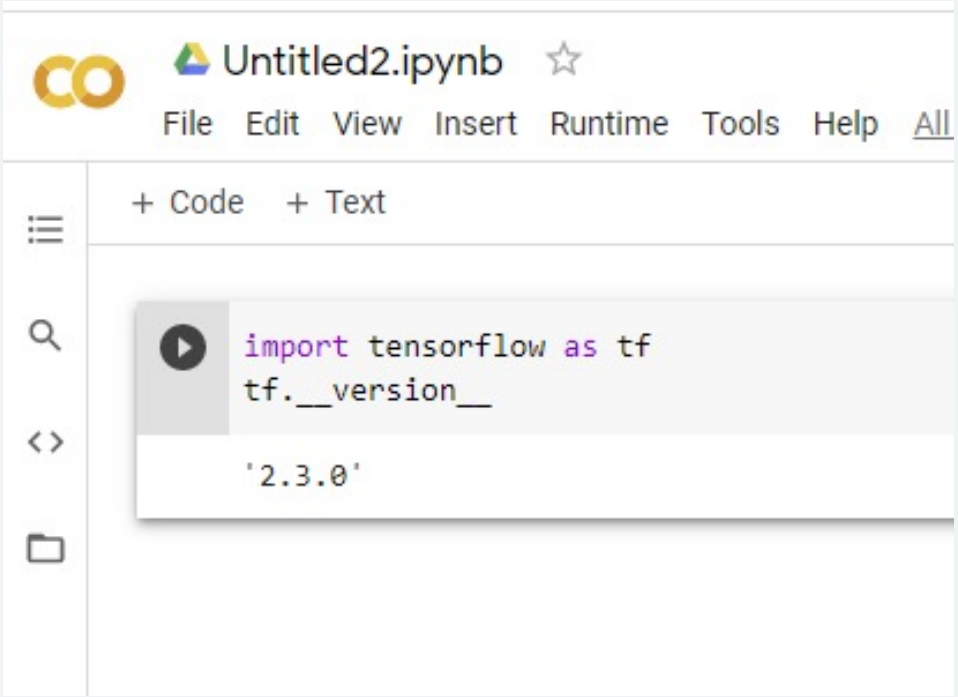
Begin Deep Learning Development in Colab



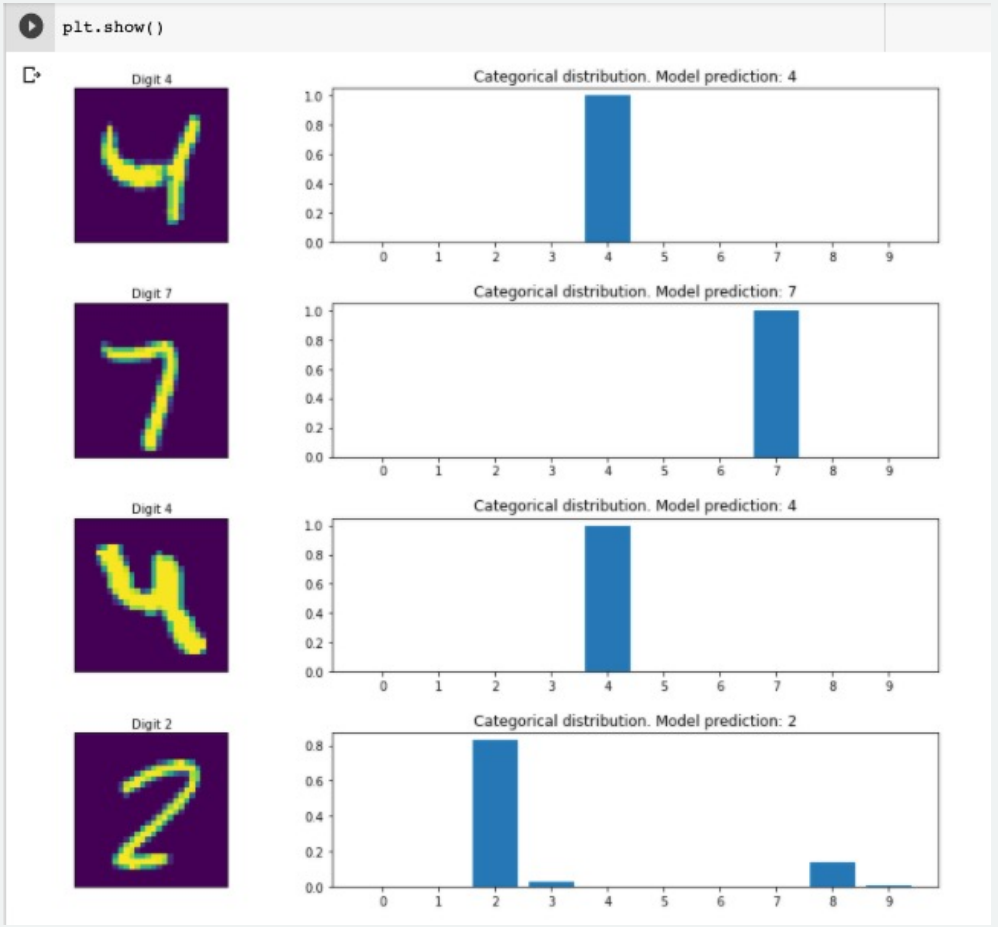
2 types of cell in notebook



TensorFlow is already installed in Colab



CNN_example1.ipynb

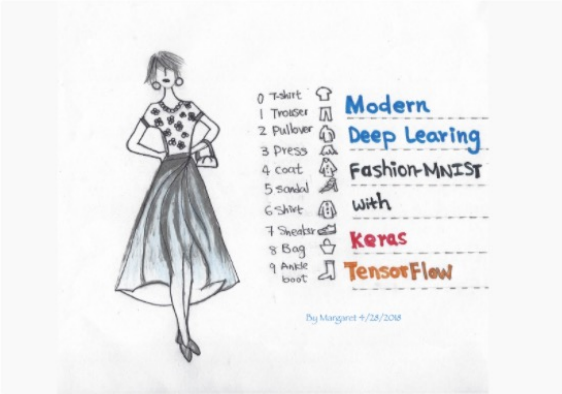


HW 6 : use the provided CNN_example1.ipynb notebook
Train your model to recognize fashion_mnist dataset.
The model must have the following layers

- 2 Conv2D layers,
- 2 MaxPooling2D layers
- 2 BatchNormalization() layers
- 1 Dropout() layer

Train your model to achieve test accuracy > 0.9

This is a tutorial of how to classify the Fashion-MNIST dataset with tf.keras, using a Convolutional Neural Network (CNN) architecture. In just a few lines of code, you can define and train a model that is able to classify the images with over 90% accuracy, even without much optimization.




```
# Run your function to evaluate the model

test_loss, test_accuracy = evaluate_model(model, scaled_test_images, test_labels)
print(f"Test loss: {test_loss}")
print(f"Test accuracy: {test_accuracy}")
```

```
313/313 [==============================] - 3s 9ms/step - loss: 0.2742 - accuracy: 0.9002
Test loss: 0.2741547226905823
Test accuracy: 0.9002000093460083
```

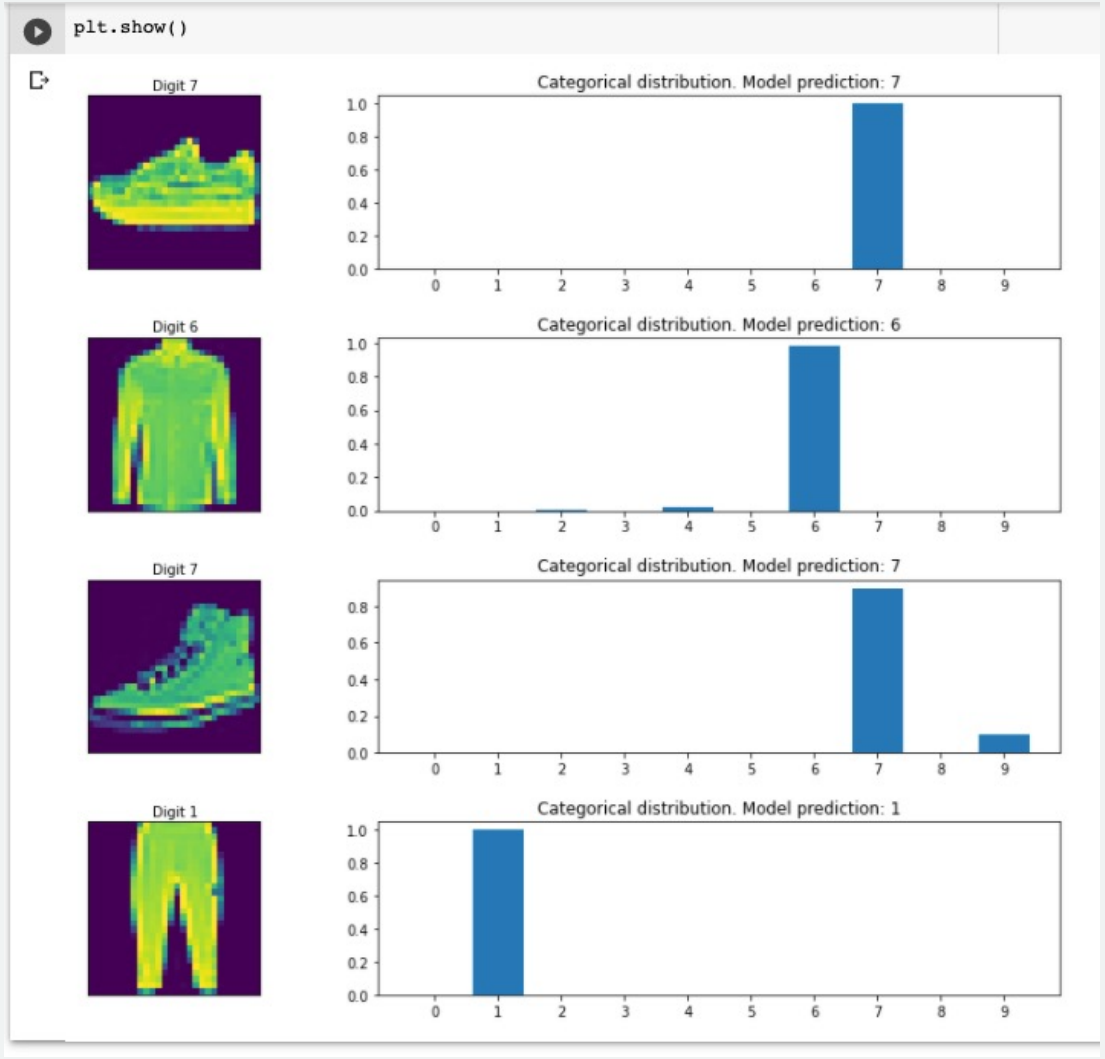

0 T-shirt
1 Trouser
2 Pullover
3 Press
4 coat
5 sandal
6 Shirt
7 Sneaker
8 Bag
9 Ankle boot

Categories

```
[50] model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_5 (Conv2D)            (None, 28, 28, 16)        160
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 14, 14, 16)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 14, 14, 32)        4640
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
batch_normalization_1 (Batch (None, 7, 7, 32)          128
_________________________________________________________________
dropout_1 (Dropout)          (None, 7, 7, 32)          0
_________________________________________________________________
batch_normalization_2 (Batch (None, 7, 7, 32)          128
_________________________________________________________________
flatten_3 (Flatten)          (None, 1568)              0
_________________________________________________________________
dense_9 (Dense)              (None, 64)                100416
_________________________________________________________________
dense_10 (Dense)             (None, 64)                4160
_________________________________________________________________
dense_11 (Dense)             (None, 10)                650
=================================================================
Total params: 110,282
Trainable params: 110,154
Non-trainable params: 128
_________________________________________________________________
```

Fashion MNIST prediction



References

- F. Chollet. Deep Learning with Python. Manning Publications Co. 2018.
- A. Geron. Machine Learning with Scikit-Learn, Keras & TensorFlow. O'Reilly Media, Inc. 2019.
- I. GoodFellow, Y. Bengio and A. Courville. Deep Learning. www.deeplearningbook.org.
- Coursera
  - Deep Learning Specialization. Deeplearning.ai.
  - DeepLearning.AI TensorFlow Developer
- MIT 6.S191 Introduction to Deep Learning http://introtodeeplearning.com/