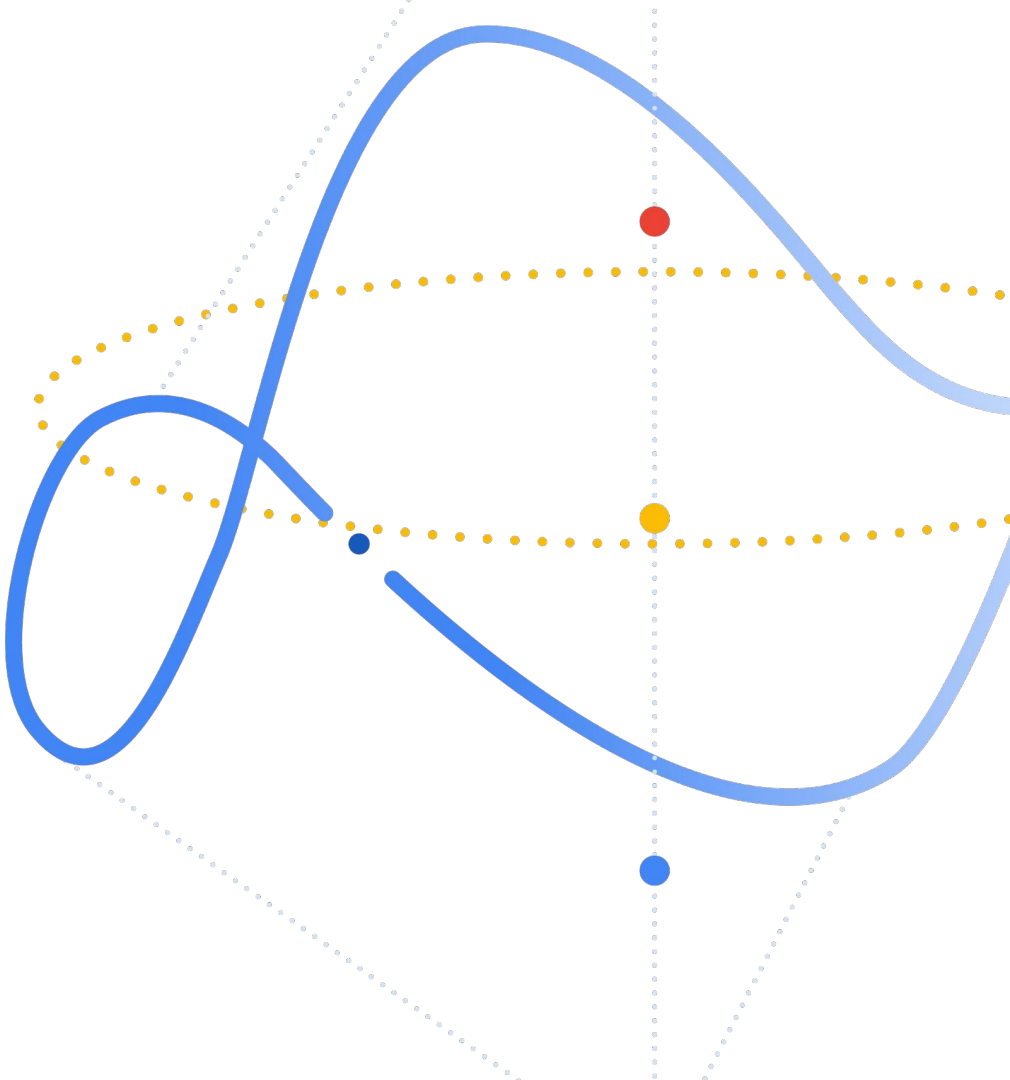
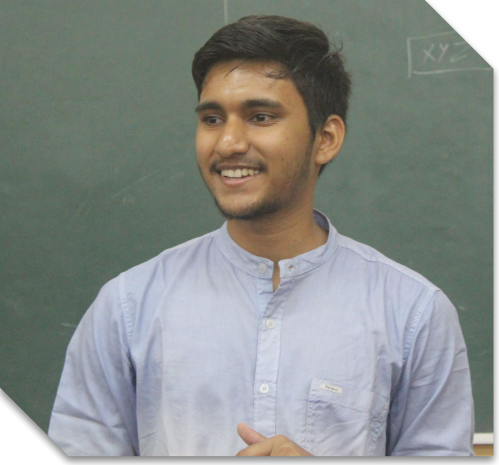
Introduction to ML

Explore ML



Google AI

Crowdsource
By Google



A Quick Introduction

Myself Srajan Chourasia

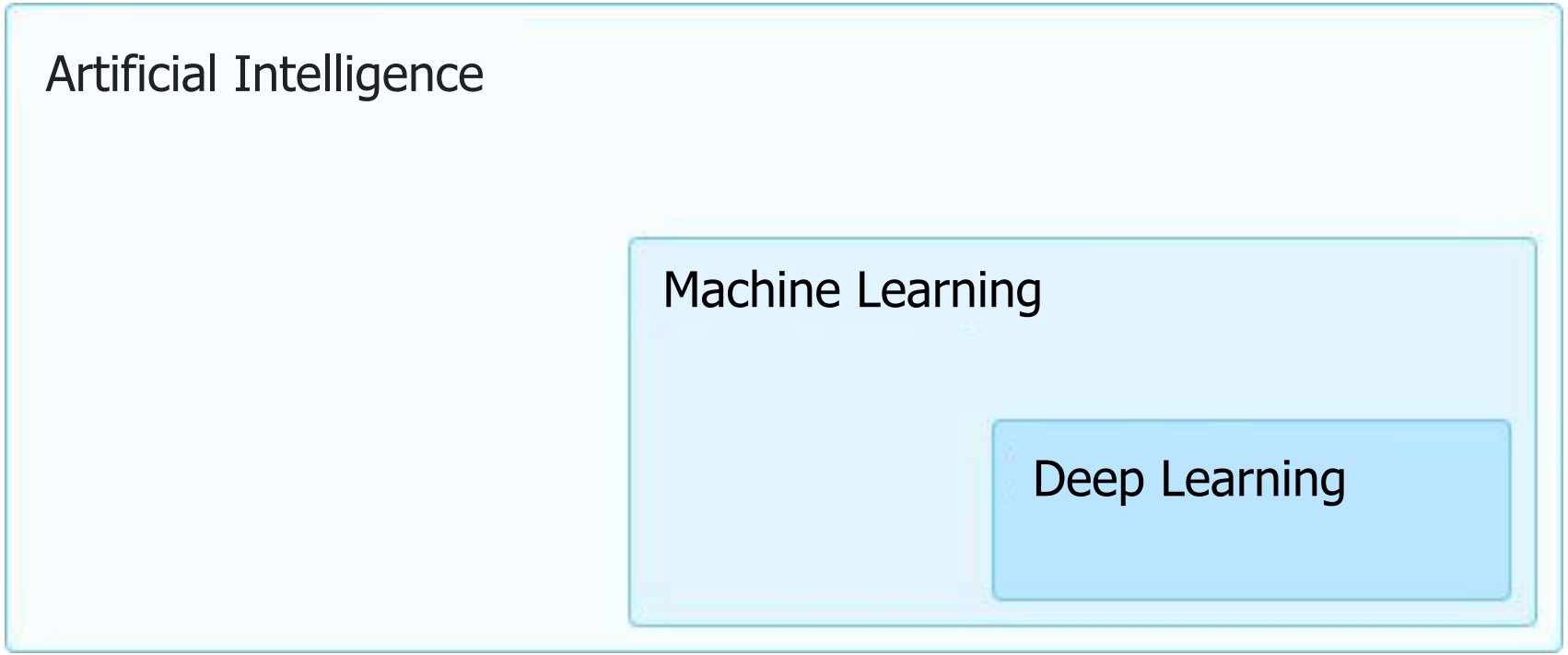
I mostly work in the domain of AI/ML especially in the field of Deep Learning and applying it in IOT/Robotics.

GitHub

Google AI

Crowdsource
By Google

What is Machine Learning?


Artificial Intelligence
Machine Learning
Deep Learning

Machine Learning systems take inputs (data) to make useful predictions and decisions about previously unseen pieces of data.

Machine learning is a specific field of AI where a system learns to find patterns in examples in order to make predictions.

Computers learning how to do a task without being explicitly programmed to do so.

Machine Learning systems might:

- Label or classify data
- Predict numerical values
- Cluster similar pieces of data together
- Infer association patterns in data
- Create complex outputs

Machine Learning

Supervised

Model is trained on labeled data

stop_sign_1

stop_sign_2

stop_sign_3

stop_sign_4

Unsupervised

Model learns patterns from unlabelled data.

Google AI

Crowdsource
By Google

Features

Predicting the Price of a House

- Location
- Number of bedrooms
- Size of property

- Number of light switches?
- Color of house?

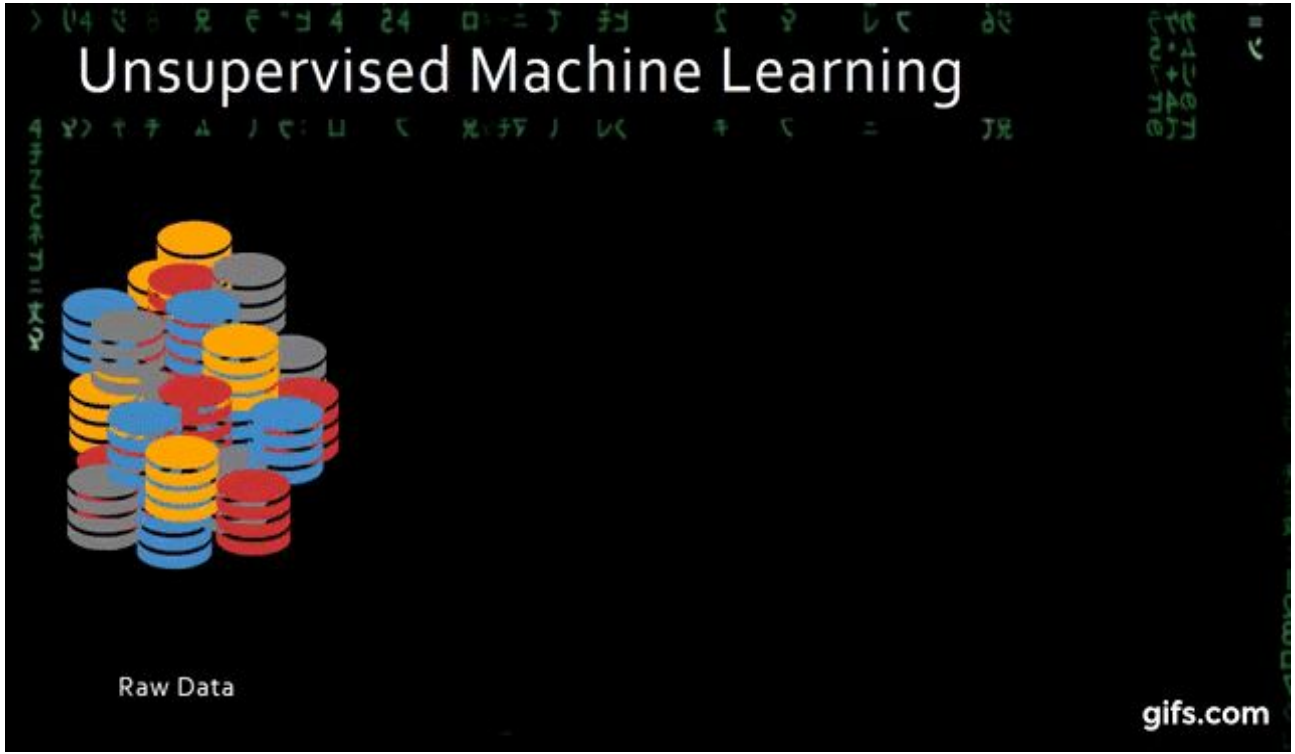
Features

Recommending which video a user should watch next

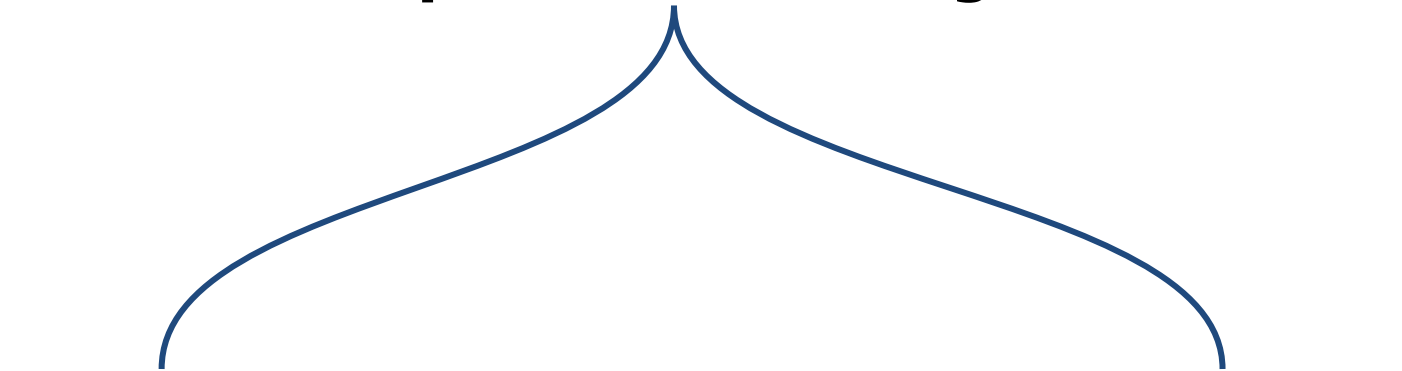
- Topic
- Popularity of a video/Number of views
- Creator of video

- Length of video?
- Age of video?

Unsupervised Learning

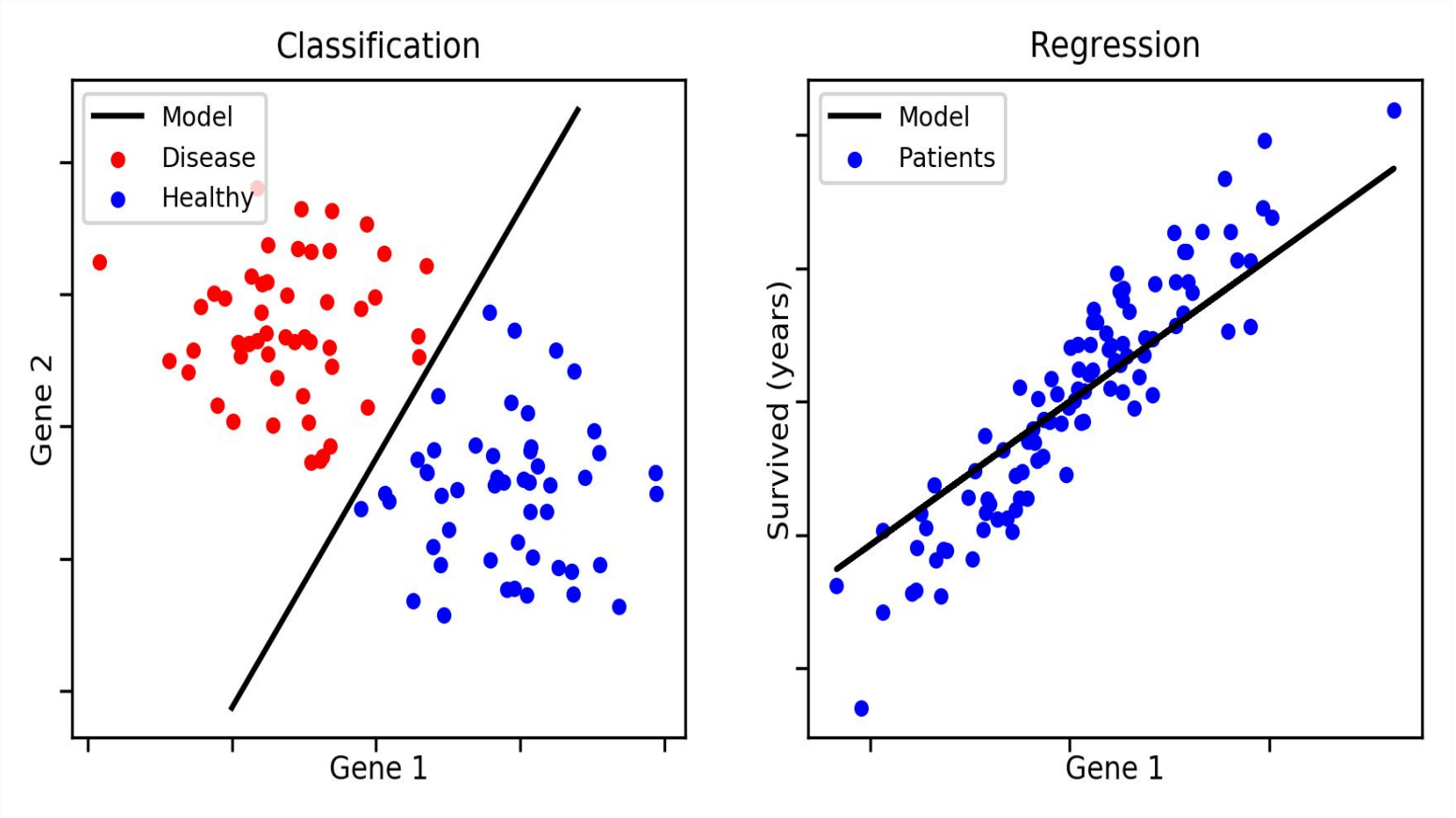


Supervised Learning



Regression

Classification


Classification
Model
Disease
Healthy
Gene 2
Gene 1
Regression
Model
Patients
Survived (years)
Gene 1

Type of Machine Learning?

Labeling email as spam or not-spam

- Classification
- Clustering
- Regression
- Sequence Prediction
- Style Transfer

Type of Machine Learning?

Labeling email as spam or not-spam

- **Classification**
- Clustering
- Regression
- Sequence Prediction
- Style Transfer

Type of Machine Learning?

Estimating arrival time based on time of day and traffic.

- Classification
- Clustering
- Regression
- Sequence Prediction
- Style Transfer

Type of Machine Learning?

Estimating arrival time based on time of day and traffic.

- Classification
- Clustering
- **Regression**
- Sequence Prediction
- Style Transfer

Type of Machine Learning?

Suggesting spelling corrections

- Classification
- Clustering
- Regression
- Sequence Prediction
- Style Transfer

Type of Machine Learning?

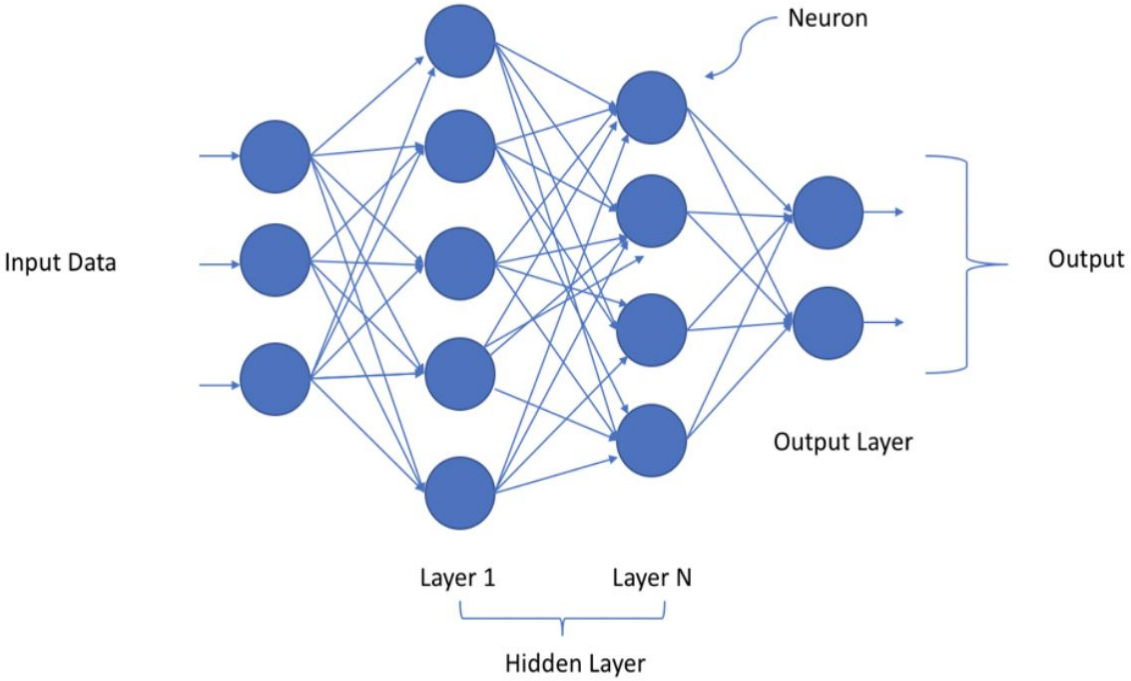
Suggesting spelling corrections

- Classification
- **Clustering**
- Regression
- Sequence Prediction
- Style Transfer

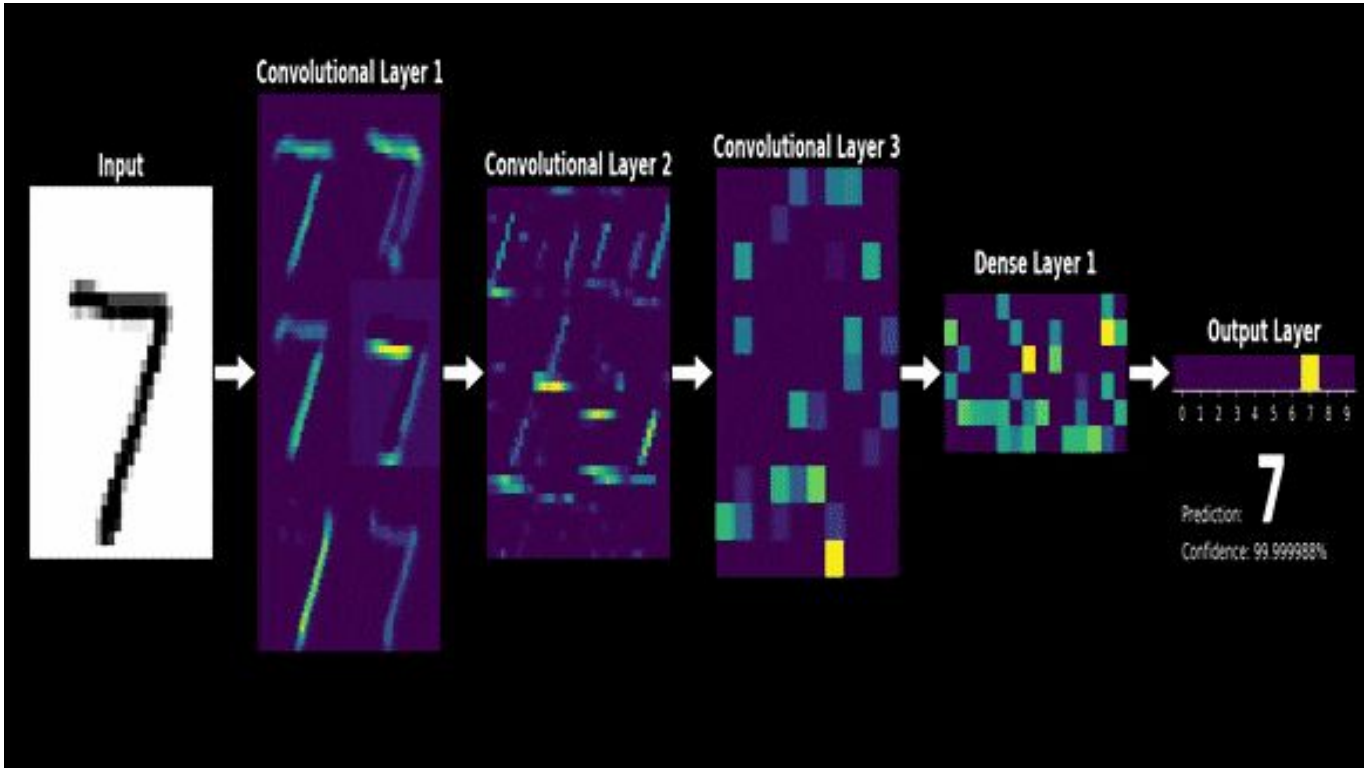
Deep Learning

- ❑ Simple Neural Networks
- ❑ Deep Neural Networks
- ❑ Convolutional neural networks (CNNs)
- ❑ Recurrent neural networks (RNNs)

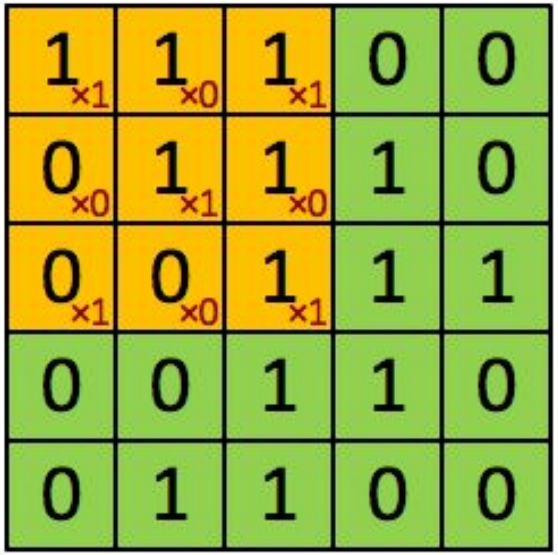
Deep Neural Networks



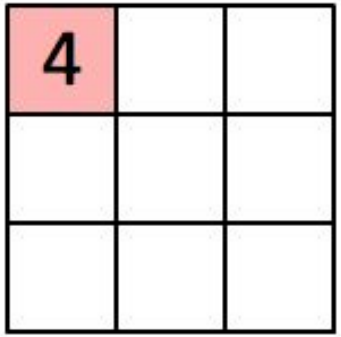
Convolutional neural networks (CNNs)



Convolution

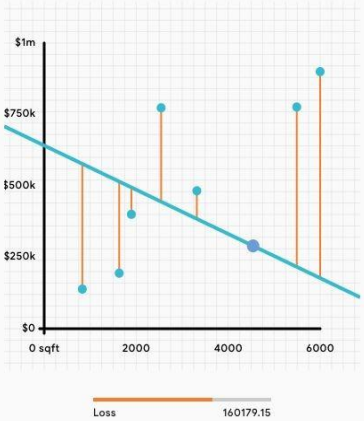


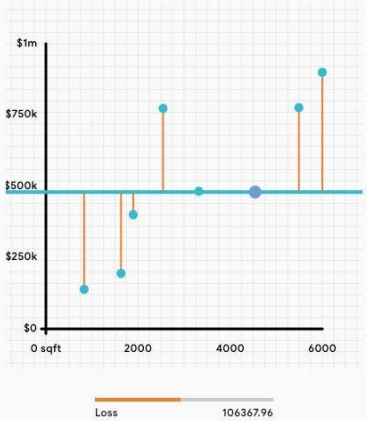
Image

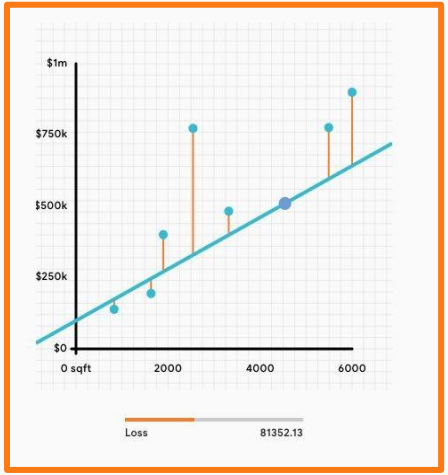


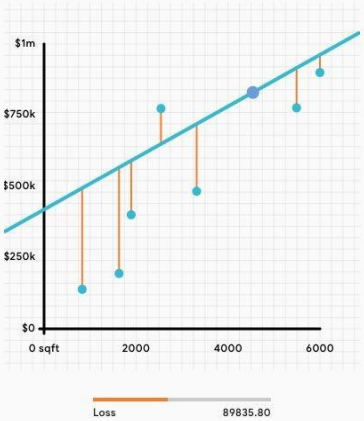
Convolved Feature

Loss

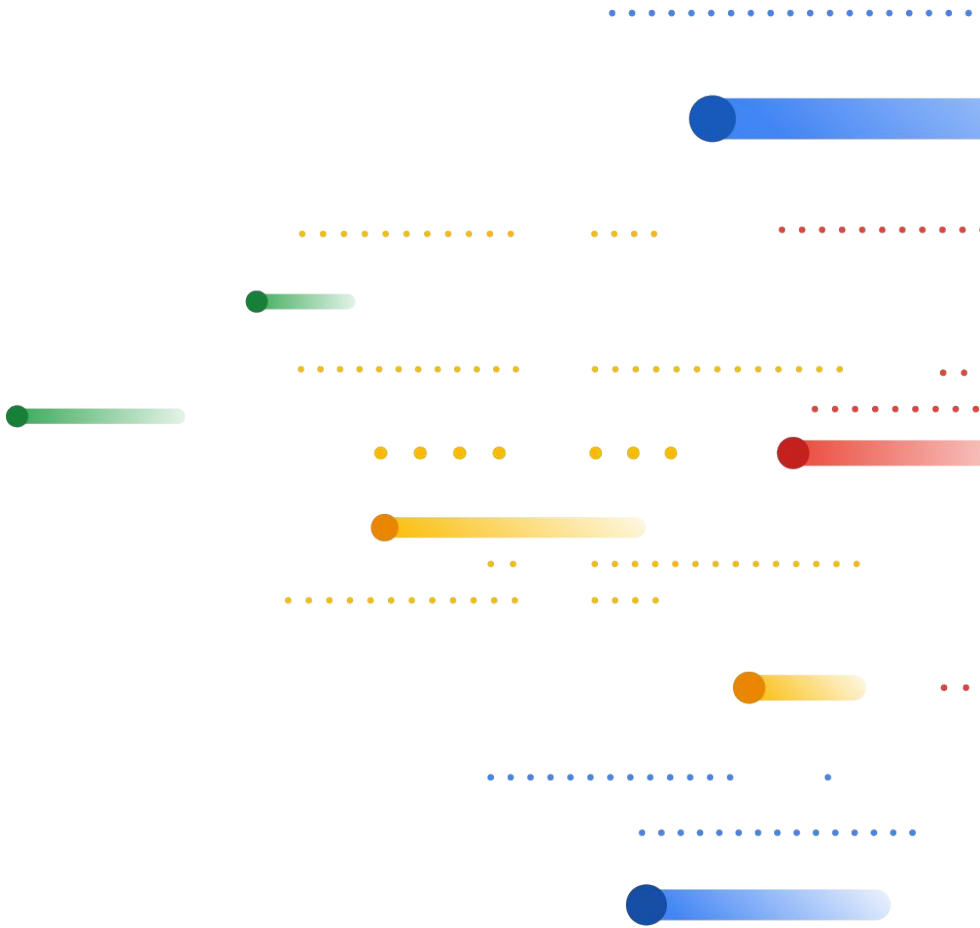

$1m
$750k
$500k
$250k
$0
0 sqft
2000
4000
6000
Loss
160179.15


$1m
$750k
$500k
$250k
$0
0 sqft
2000
4000
6000
Loss
106367.96


$1m
$750k
$500k
$250k
$0
0 sqft
2000
4000
6000
Loss
81352.13


$1m
$750k
$500k
$250k
$0
0 sqft
2000
4000
6000
Loss
89835.80

Let’s Build a CNN Model to Predict Handwritten Digits!



Google AI

Crowdsource
By Google

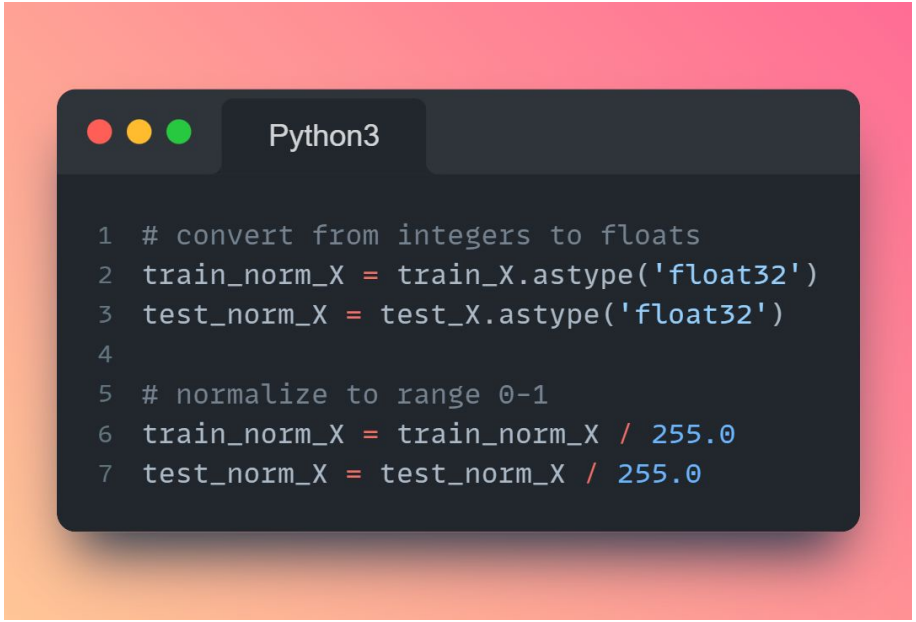
We will Use the MNIST Dataset!

First Import the MNIST DataSet of HandWritten Digits.

Python3

```python
# loading the mnist dataset
from tensorflow.keras.datasets import mnist
from matplotlib import pyplot as plt

# load dataset
(train_X, train_y), (test_X, test_y) = mnist.load_data()

# reshape dataset to have a single channel
train_X = train_X.reshape((train_X.shape[0], 28, 28, 1))
test_X = test_X.reshape((test_X.shape[0], 28, 28, 1))

# summarize loaded dataset
print(f'> Train Dataset: X (Image Shape) = {train_X.shape} and Y (Labels Shape) = {train_y.shape}')
print(f'> Test Dataset: X (Image Shape) = {test_X.shape} and Y (Labels Shape) = {test_y.shape}')
```

Let's Normalising the Images



One-Hot Encoding

id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

One-Hot Encoding

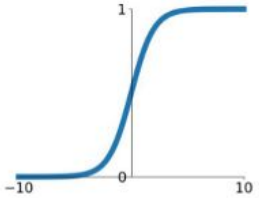
Python3

```python
# import the function to do one-hot encoding from keras
from keras.utils import to_categorical

# one-hot encoding of train and test labels
train_labels = to_categorical(train_y)
test_labels = to_categorical(test_y)

# shape of one-hot encoded labels
print(f'> Train Labels: Y (Labels Shape) = {train_labels.shape}')
print(f'> Test Labels: Y (Labels Shape) = {test_labels.shape}')
```

Activation Functions!

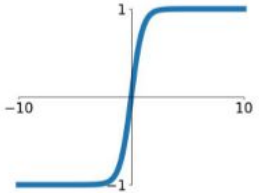
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



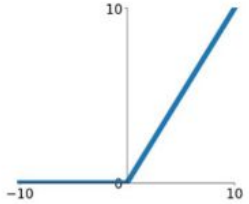
tanh

$$\tanh(x)$$



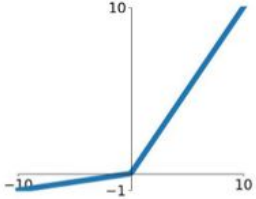
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

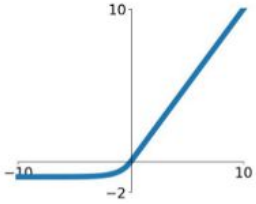


Maxout

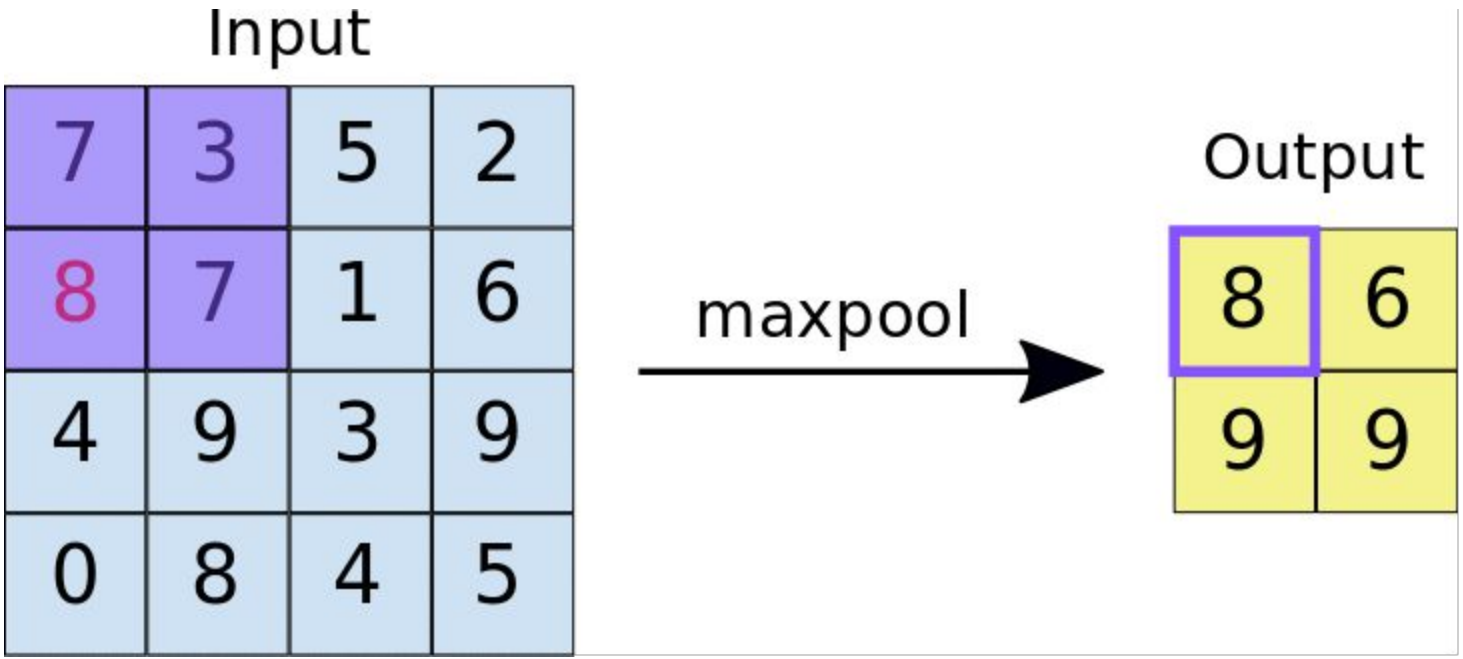
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

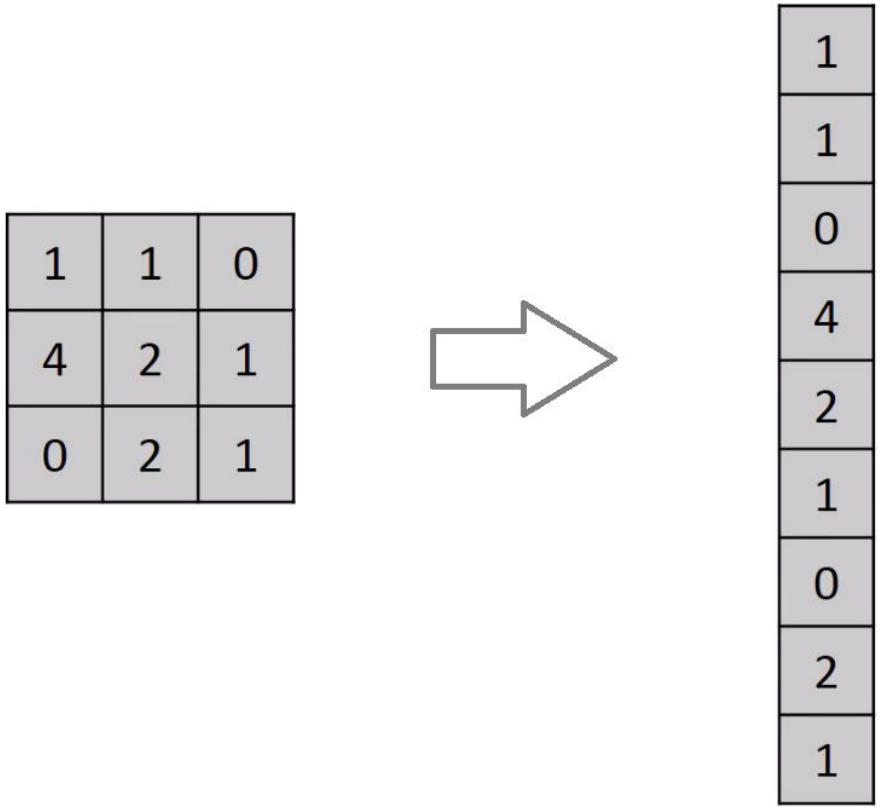
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Max Pooling



Flattening



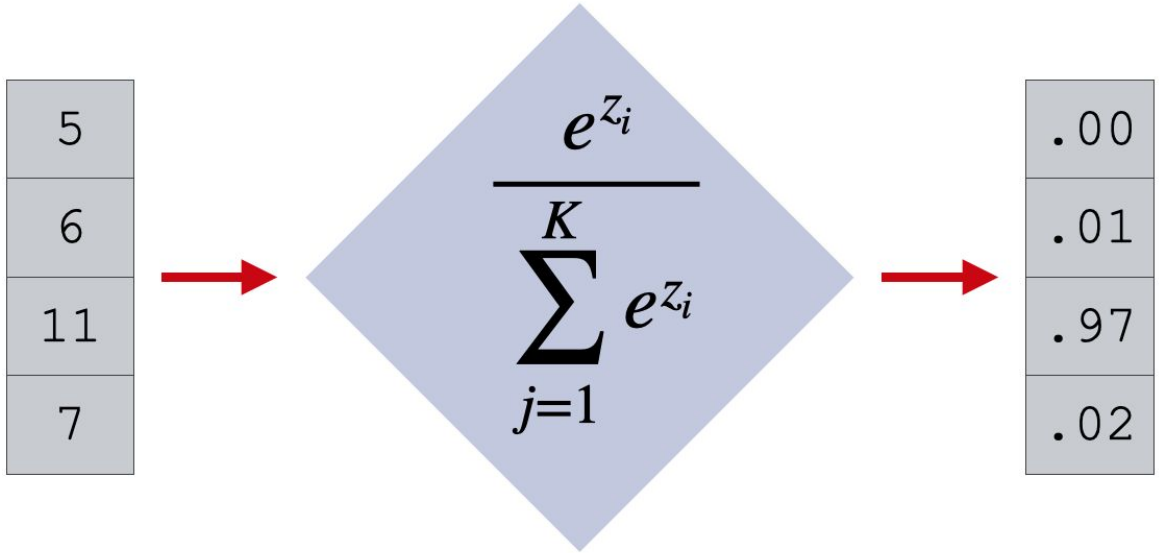
Let's build the model architecture

Python3

```python
# import all the necessary libraries
import keras
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from keras import models
from keras.datasets import mnist

# Define the CNN model
model = models.Sequential()

# Input Layer
# First Convolution Layer
model.add(Conv2D(filters = 64, kernel_size = (3,3), activation='relu', input_shape=(28, 28,1)))

# Max Pool Layer
model.add(MaxPooling2D((2, 2)))

# Second Convolution Layer
model.add(Conv2D(filters = 32, kernel_size = (3, 3), activation='relu'))

# Max Pool Layer
model.add(MaxPooling2D((2, 2)))

# Flatten Layer
model.add(Flatten())

# Hidden Layers (Dense Layer)
model.add(Dense(500, activation='relu'))
model.add(Dense(200, activation='relu'))

# Final Layer
# Output Layer
model.add(Dense(10, activation='softmax'))

# print the model summary
model.summary()
```

SoftMax

SOFTMAX TRANSFORMS A VECTOR OF NUMBERS INTO A VECTOR OF RELATIVE "PROBABILITIES"



SoftMax

Input pixels, $\mathbf{x}$

Forward propagation

Feedforward output, $\mathbf{y}_i$

cat	dog	horse
5	4	2
4	2	8
4	4	1

Softmax function

Softmax output, $\mathbf{S}(\mathbf{y}_i)$

cat	dog	horse
0.71	0.26	0.04
0.02	0.00	0.98
0.49	0.49	0.02

Shape: (3, 32, 32)

Shape: (3,)

Shape: (3,)

Let's Compile and Train the model now!

Python3

```python
# Set the Optimizer
from tensorflow.keras.optimizers import Adam

# set the learning rate
adam = Adam(learning_rate = 1e-3)

# compile the model to use the training data to train the model
model.compile(loss = 'categorical_crossentropy',
              optimizer = adam,
              metrics = ['accuracy'])

# Train the model on the training dataset
model.fit(train_norm_X, train_labels,
          batch_size = 100,
          epochs = 10,
          verbose = 1)
```

Cross Entropy Loss

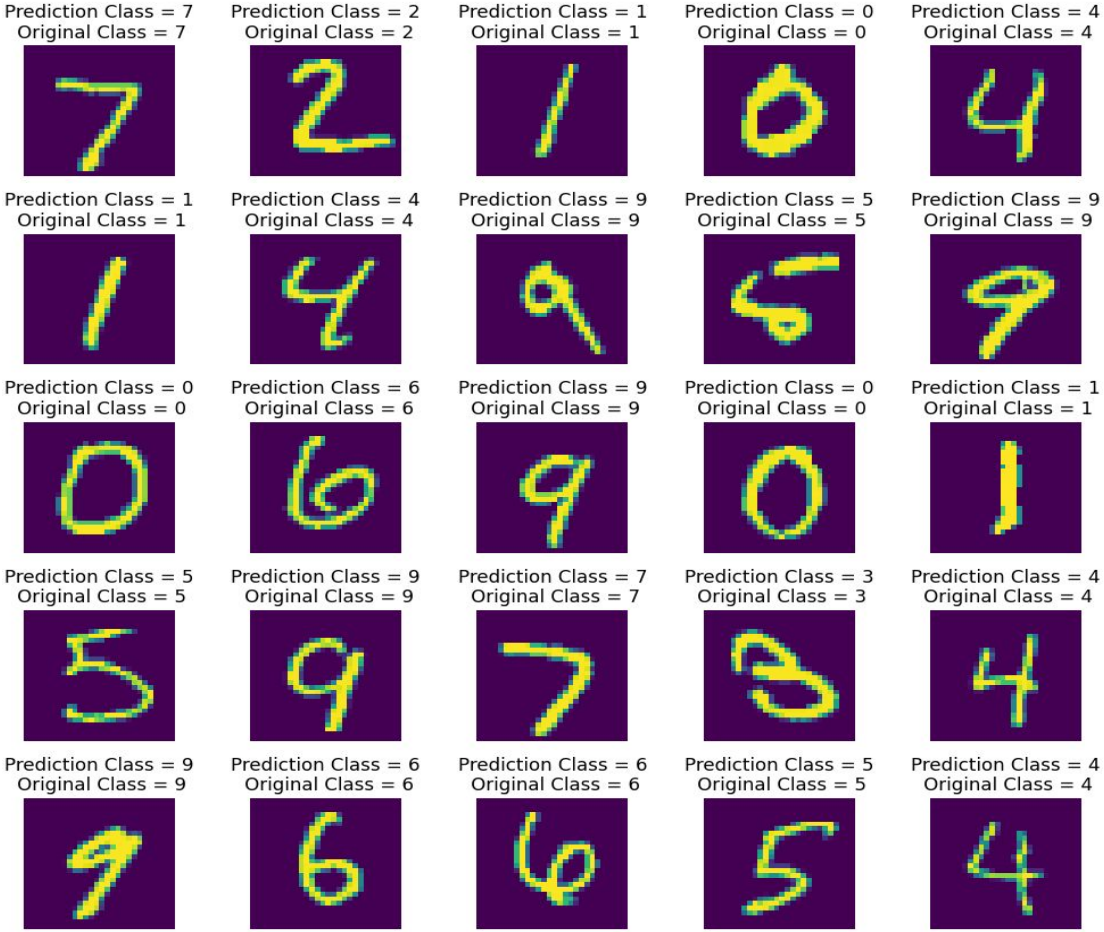
$$CE = -\sum_{neuron=1}^{classes} y_{true_{neuron}} * \ln\left(y_{pred_{neuron}}\right)$$

Let's Test Our model!

Python3

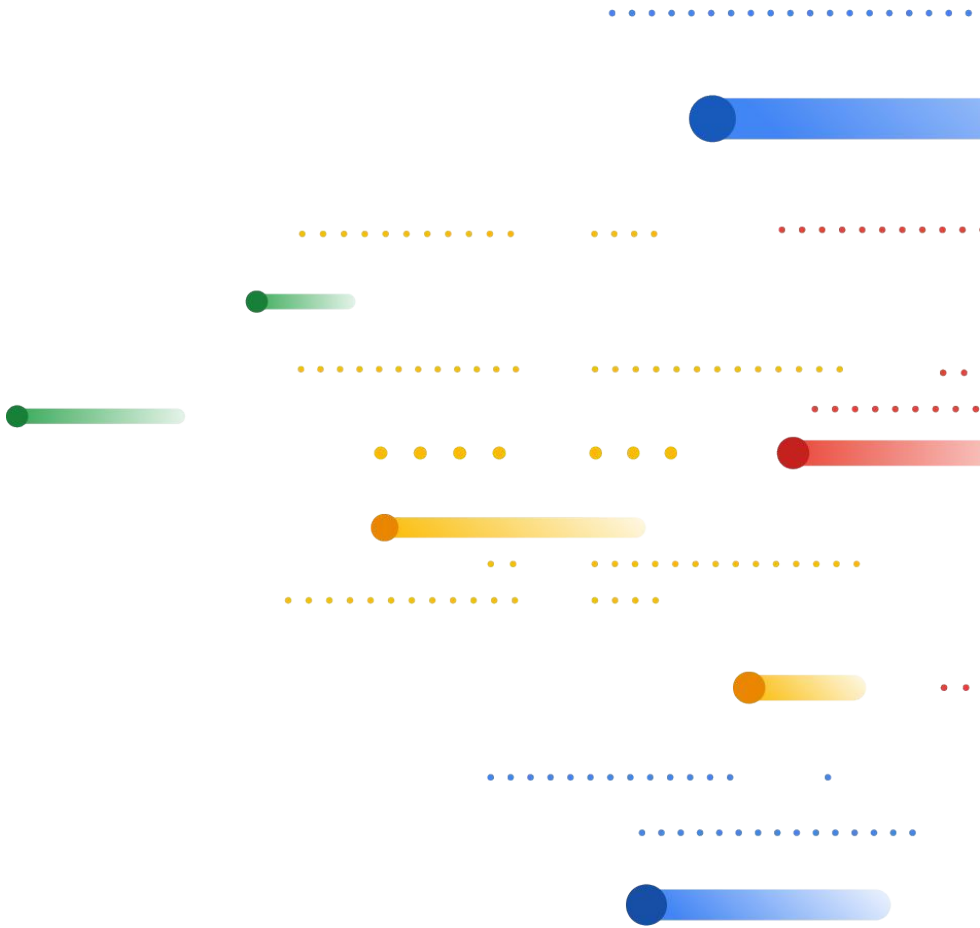
```python
import numpy as np

# Test the model's accuracy with the test data
predictions = model.predict(test_norm_X)
predictions = np.argmax(predictions, axis = 1)

test_loss, test_acc = model.evaluate(test_norm_X, test_labels)

# print the test accuracy
print('> Test accuracy:', test_acc)
```

Final Output



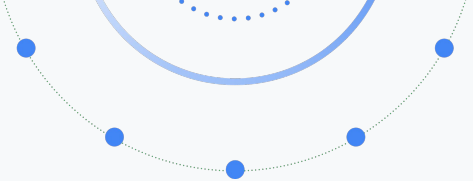
Task

Build a CNN Model Using Fashion MNIST Dataset



Q & A

GitHub Repository: Explore ML



Thank You!

Connect with me:

GitHub

LinkedIn