Machine Learning for Humans

@ GDG RAJKOT DevFest Season'18

GDG RAJKOT

This presentation is more fun with GIFs and animations, access it from here: https://goo.gl/Bu6qNr

GDG RAJKOT

Do you use machine learning?

GDG RAJKOT


MACHINE LEARNING
MACHINE LEARNING EVERYWHERE!
memegenerator.net

GDG RAJKOT

Hello!

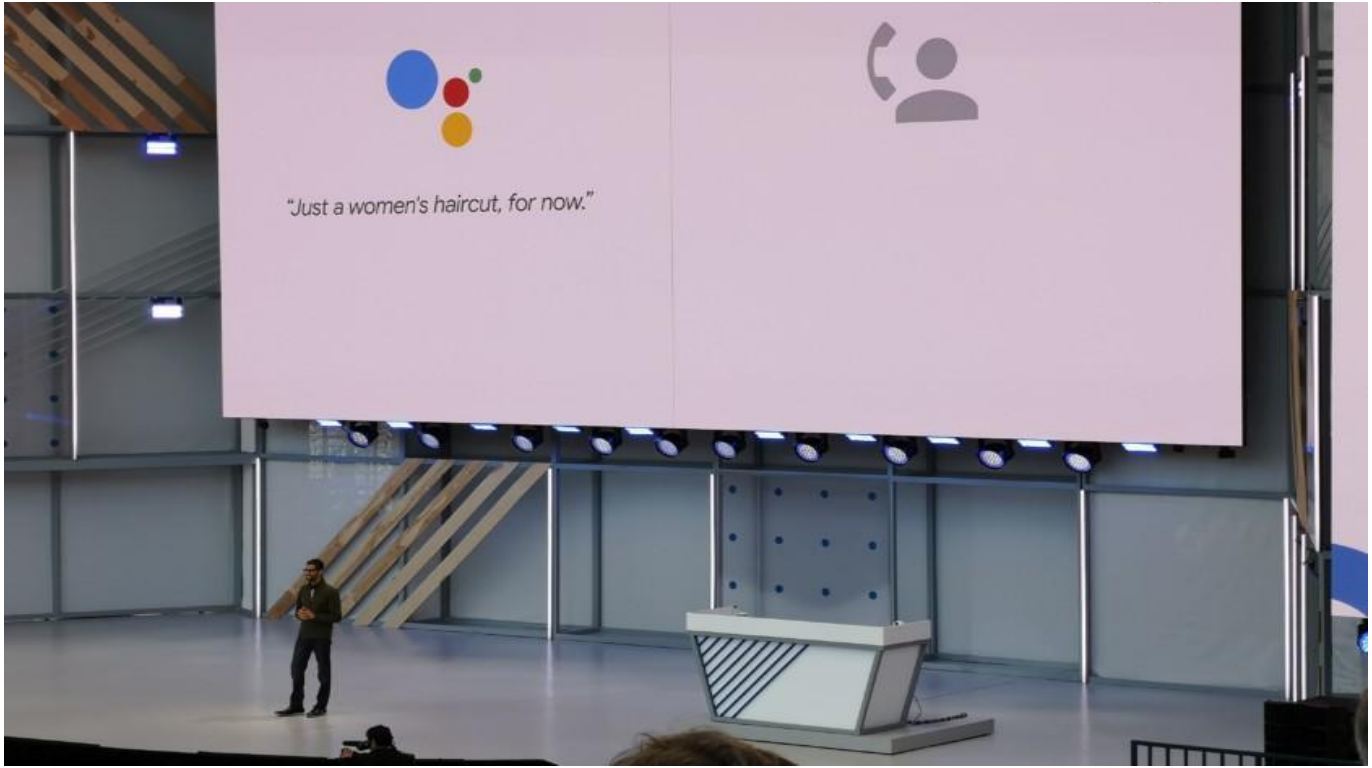
I AM Pratik Parmar

And I am here to bore you with Machine Learning.

GDG RAJKOT

- Shocking IT NEWS last year?

GDG RAJKOT

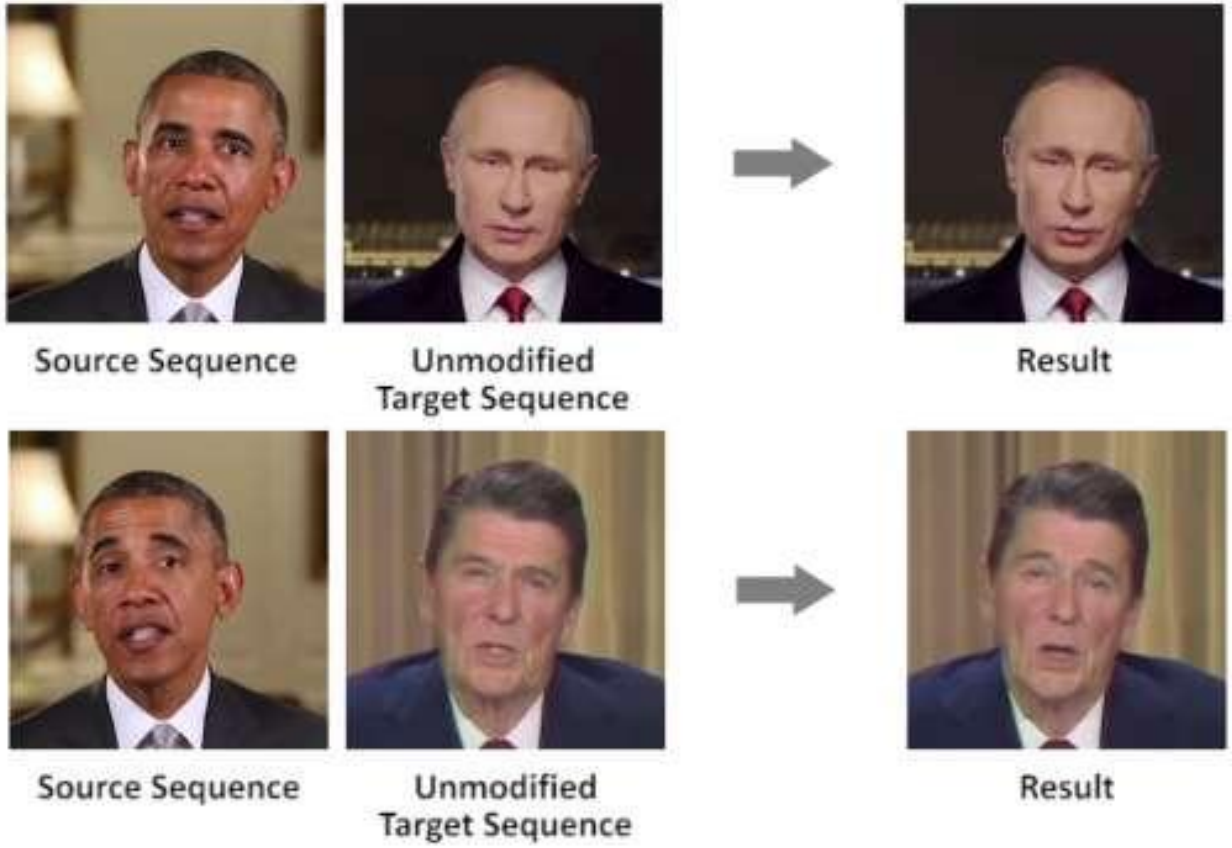


Google Duplex?

GDG RAJKOT

OpenAI's 'Dota 2' bots taking on pro teams ?

GDG RAJKOT



Deep Fake ?

GDG RAJKOT


PART 1
DEEP FAKES
Good or Bad??

What to expect?

- Why Machine Learning?
- Basic Machine Learning
- Neural Networks
- Image Classification
- Resources to learn ML

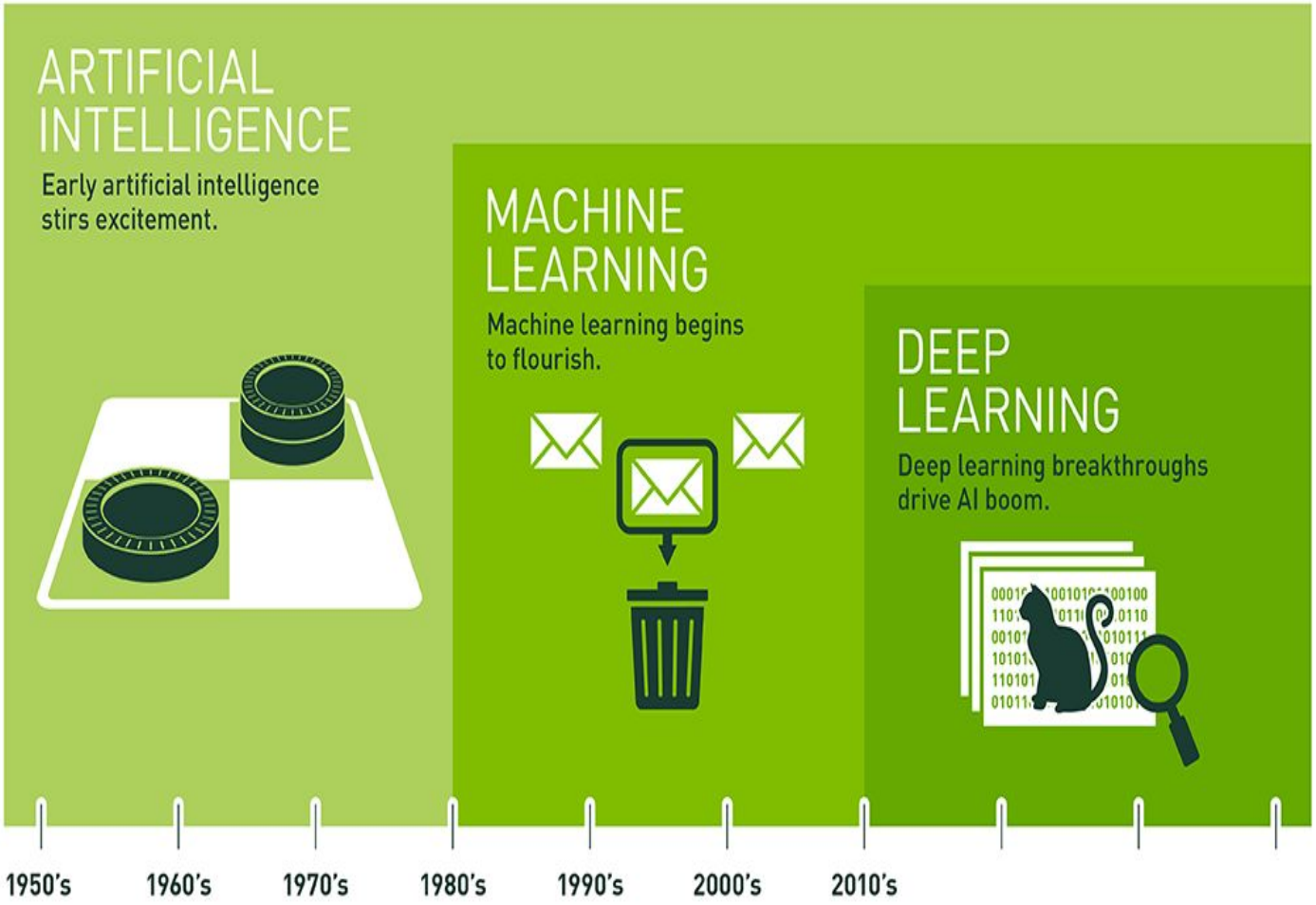
GDG RAJKOT

Intuition

+

Coding

GDG RAJKOT


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

GDG RAJKOT

Training	Prediction
many	*answer*
examples	*questions*

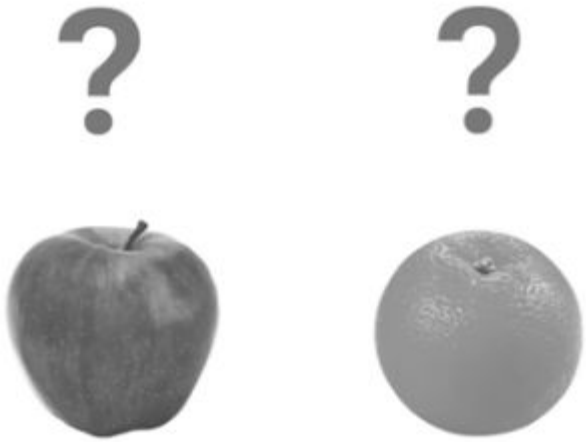
GDG RAJKOT

Basic Terminologies

- Features
- Labels
- Examples
  - Labeled examples
  - Unlabeled examples
- Models
  - Classification
  - Regression

GDG RAJKOT


“apple”

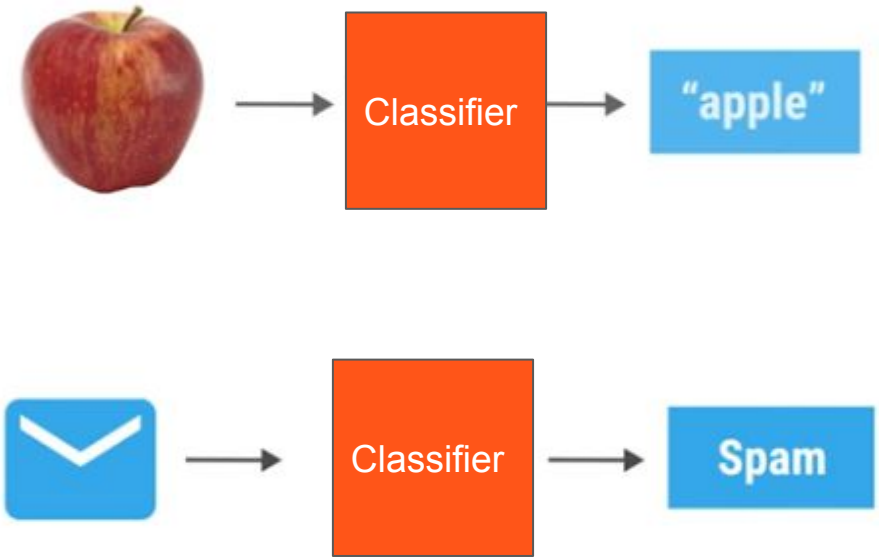


GDG
RAJKOT

```
def detect_colors(image):
    # lots of code

def detect_edges(image):
    # lots of code

def analyze_shapes(image):
    # lots of code

def guess_texture(image):
    # lots of code

def define_fruit():
    # lots of code

def handle_probability():
    # lots of code
```

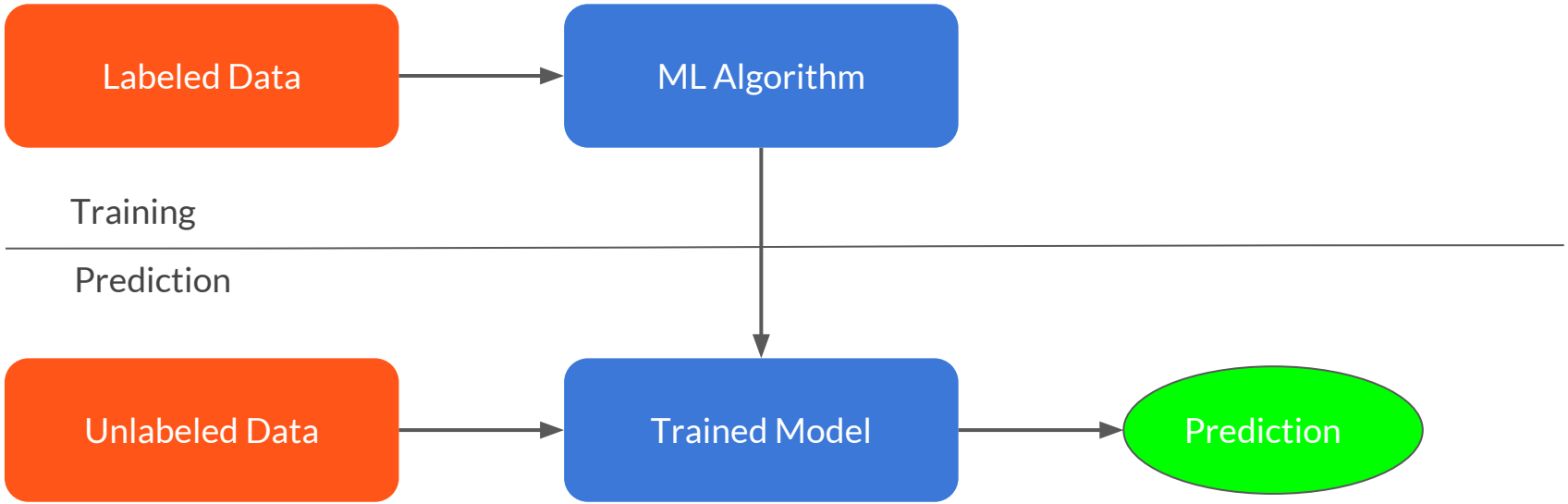
GDG RAJKOT

?



GDG RAJKOT

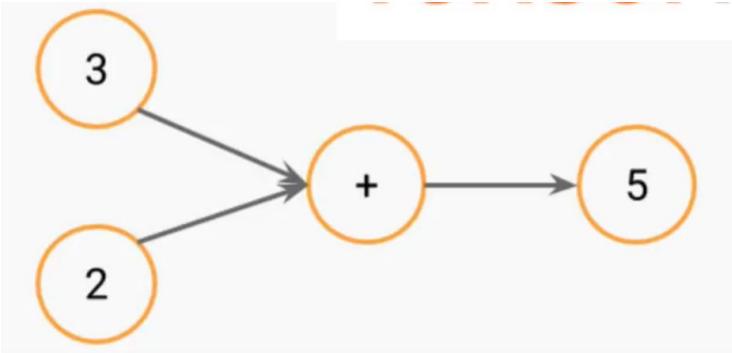
Workflow (supervised)



GDG RAJKOT

Scalar	Vector	Matrix	Tensor
1	$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 7 \end{bmatrix} & \begin{bmatrix} 3 & 2 \\ 5 & 4 \end{bmatrix} \end{bmatrix}$

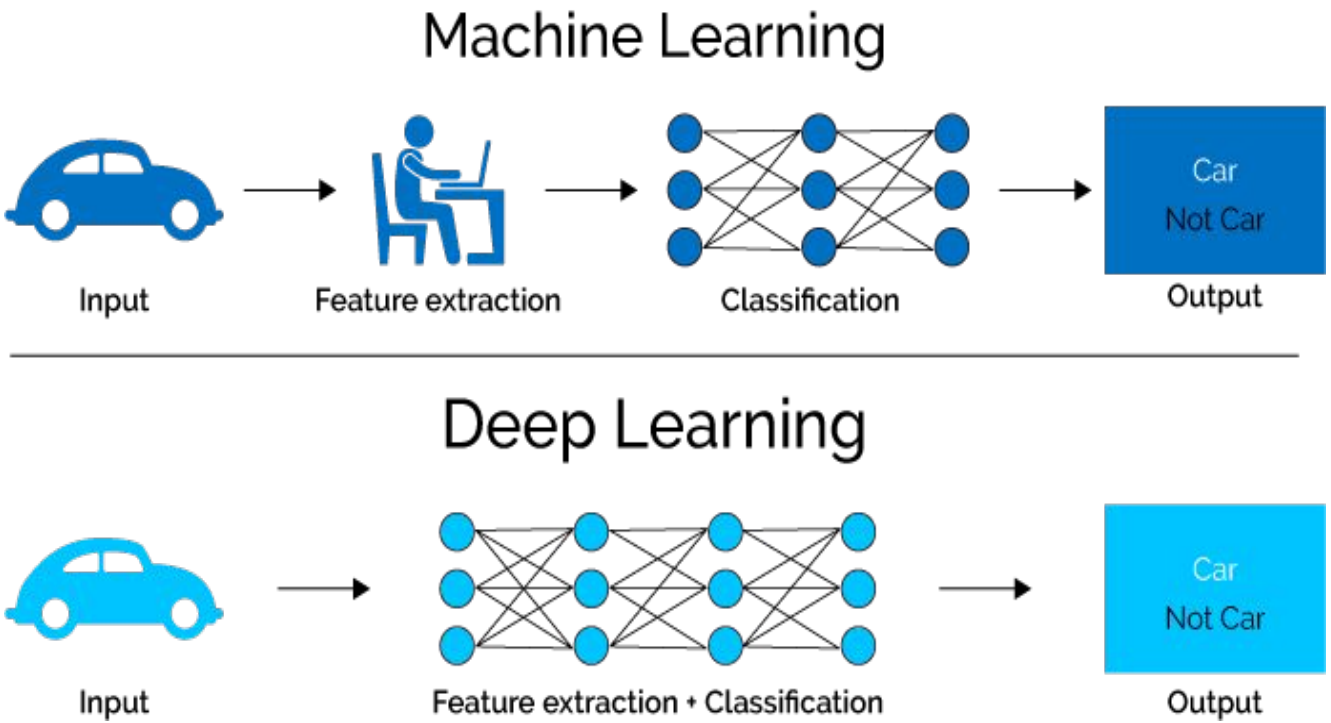
TensorFlow



Wanna be a cool guy like me?

GDG RAJKOT

Machine Learning vs Deep Learning



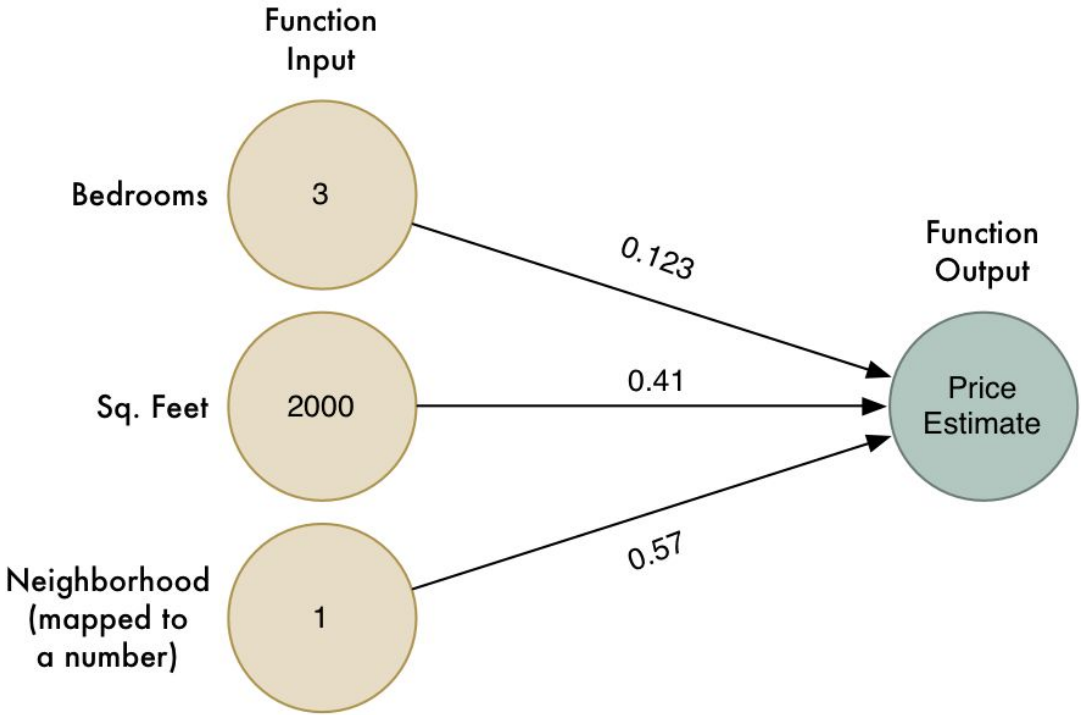
GDG RAJKOT

Neuron

A simple estimation function that takes in a set of inputs and multiplies them by weights to get an output.

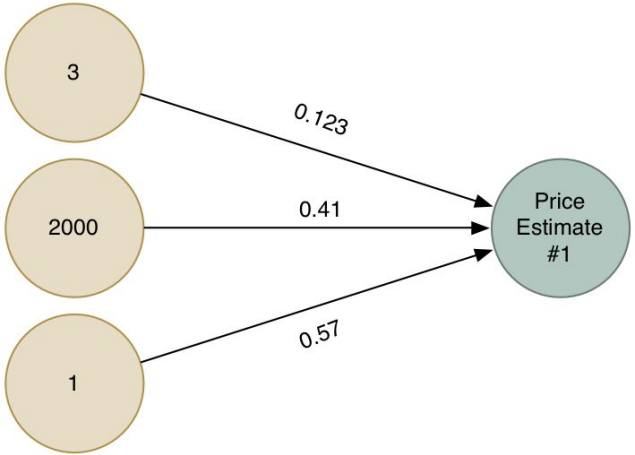
GDG RAJKOT

House price

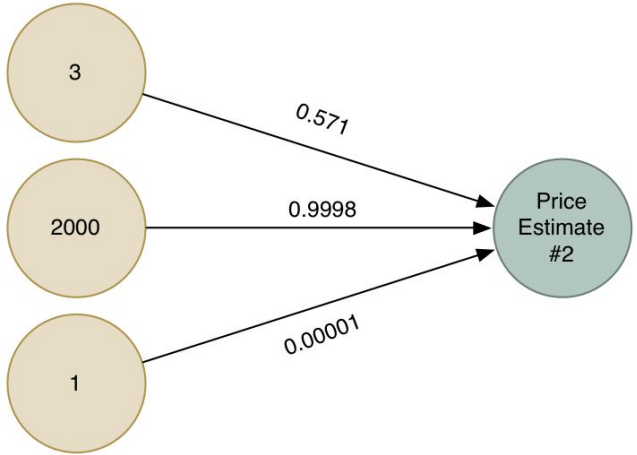


GDG RAJKOT

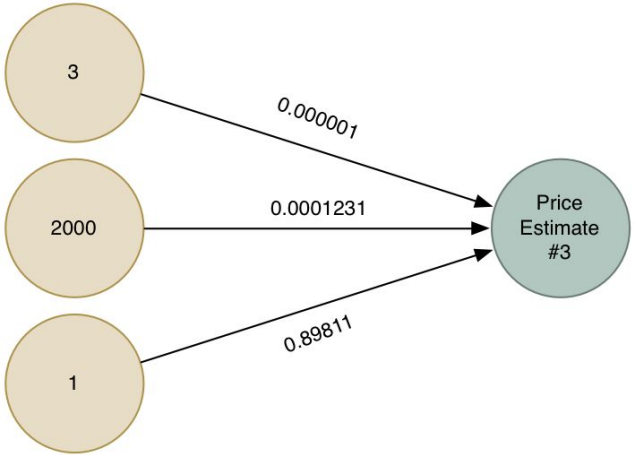
First set of weights



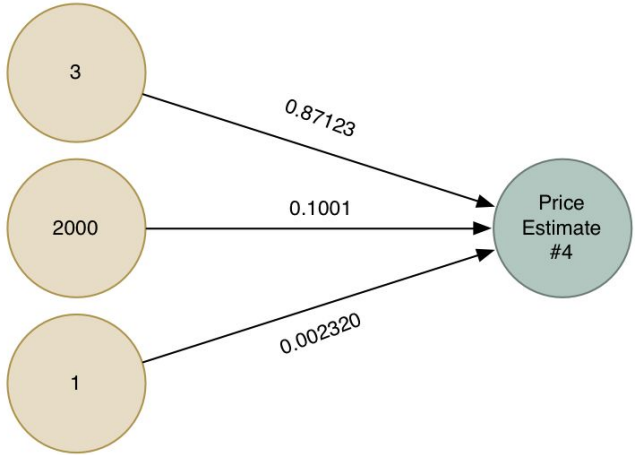
Second set of weights



Third set of weights

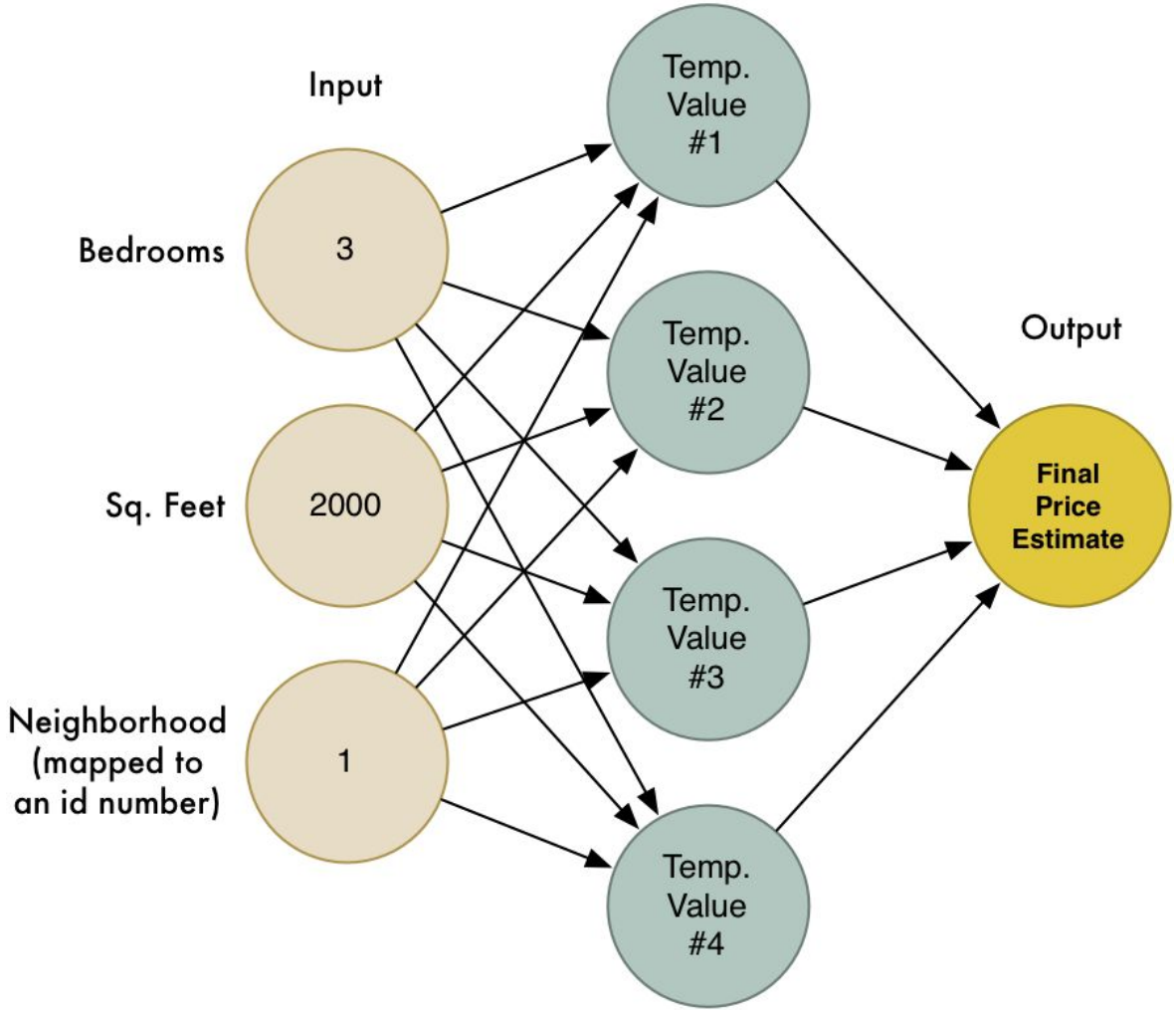


Fourth set of weights 4




Price Estimate #1
Price Estimate #2
Price Estimate #3
Price Estimate #4
0.20
0.15
0.05
0.60
Final Price Estimate

GDG RAJKOT


Input
Bedrooms
3
Sq. Feet
2000
Neighborhood
(mapped to
an id number)
1
Temp.
Value
#1
Temp.
Value
#2
Temp.
Value
#3
Temp.
Value
#4
Output
Final
Price
Estimate

GDG
RAJKOT

YESSS that is a neural network

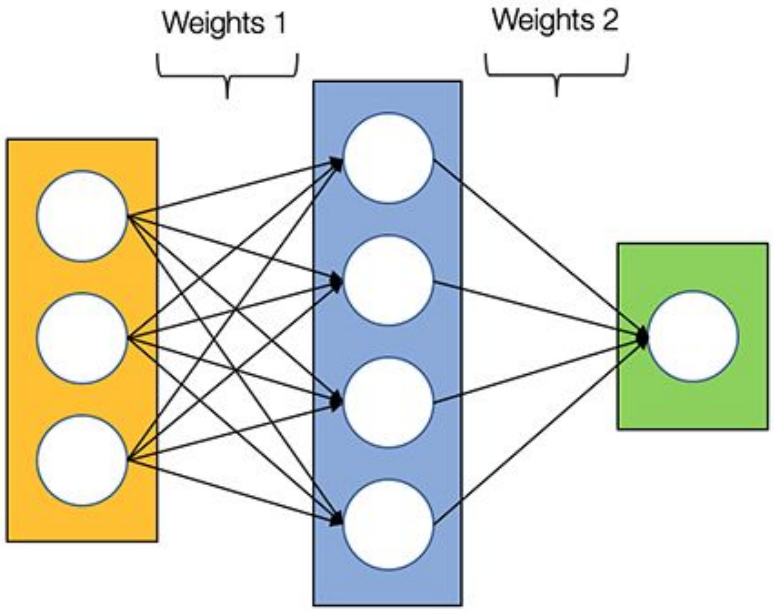
GDG RAJKOT

Neural Network

A mathematical function that maps a given input to a desired output.

GDG RAJKOT

Neural Network



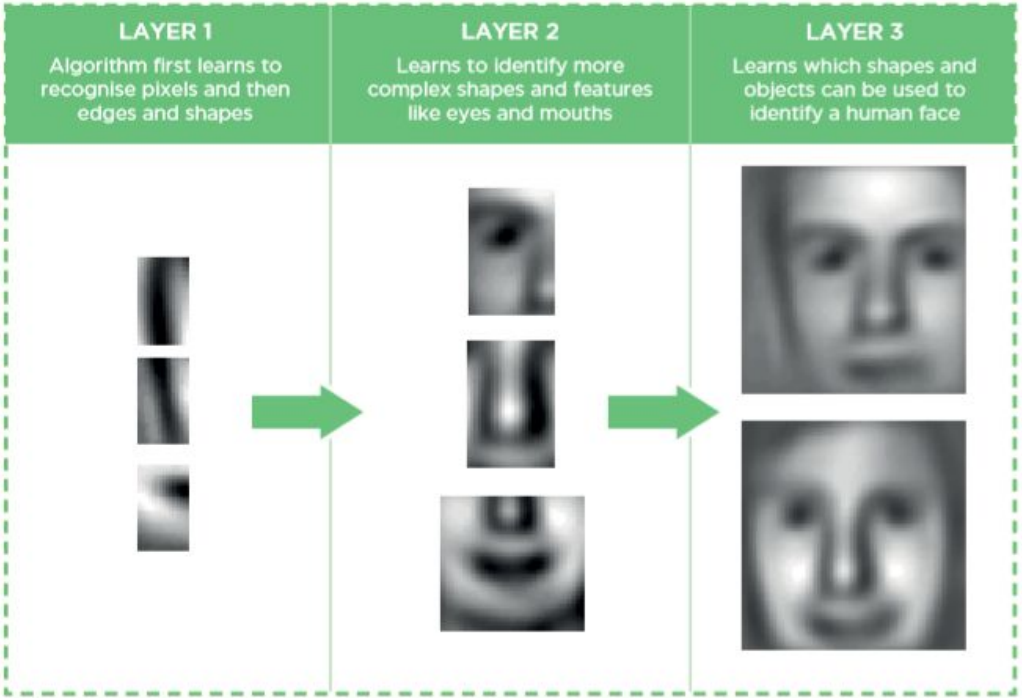
GDG RAJKOT

Components

- Input Layer - x
- Hidden layer
- Output layer - $\hat{y}$
- Weights and biases - W and b
- Activation function - σ

GDG RAJKOT

Working



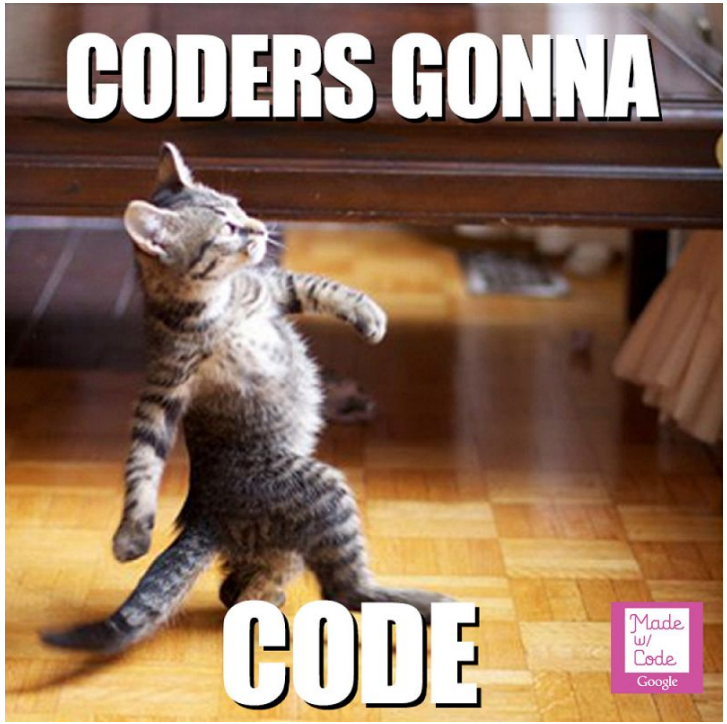
GDG RAJKOT

Tensorflow Playground

https://playground.tensorflow.org

GDG RAJKOT

Give some space because...



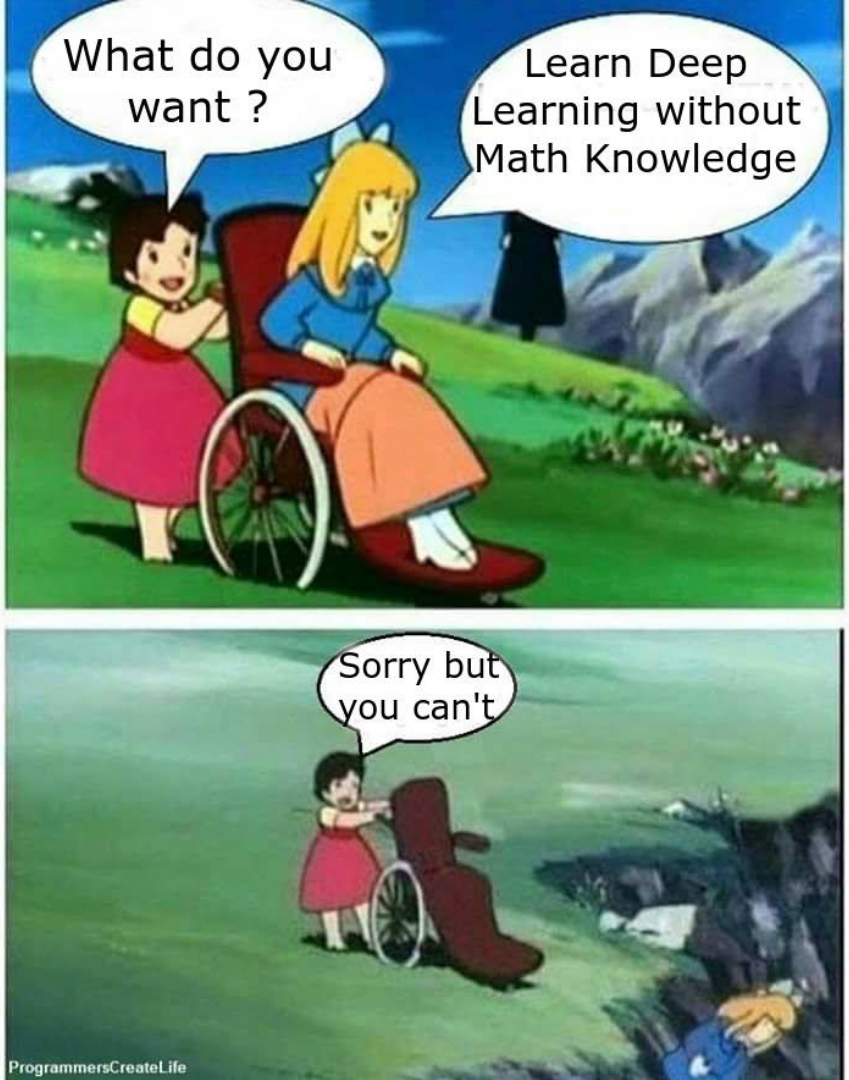
Google Colab

http://colab.research.google.com/

GDG RAJKOT

Free things

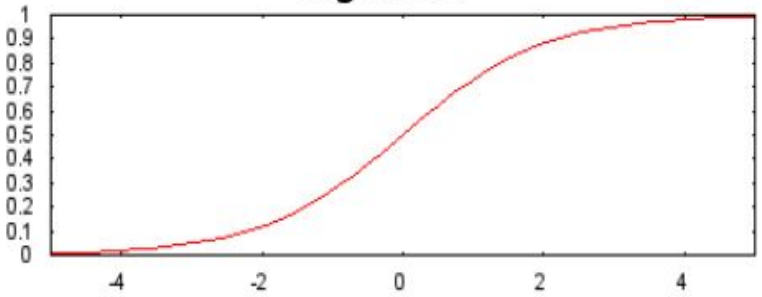
While learning Deep Learning



Activation Function

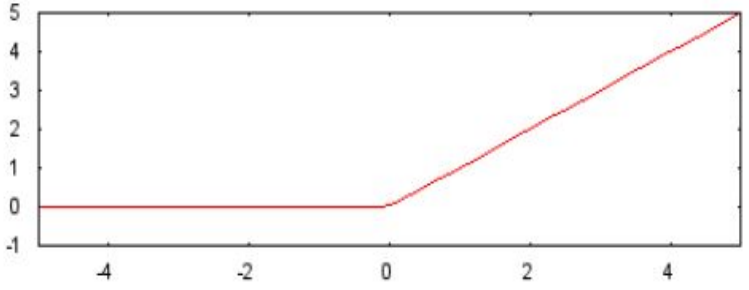
It's just a thing (**node**) that you add to the output end of any neural network. It is also known as **Transfer Function**.

Sigmoid



$$f(t) = \frac{1}{1+e^{-t}}$$

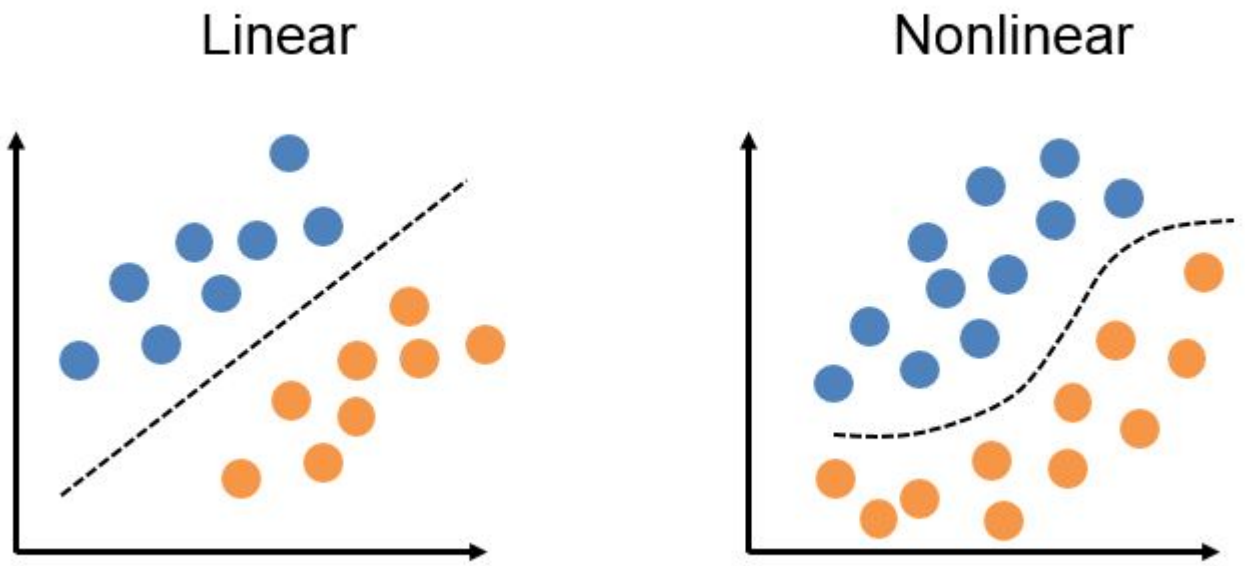
ReLU



$$f(t) = max(0, t)$$

GDG RAJKOT

Why activation function?



GDG RAJKOT

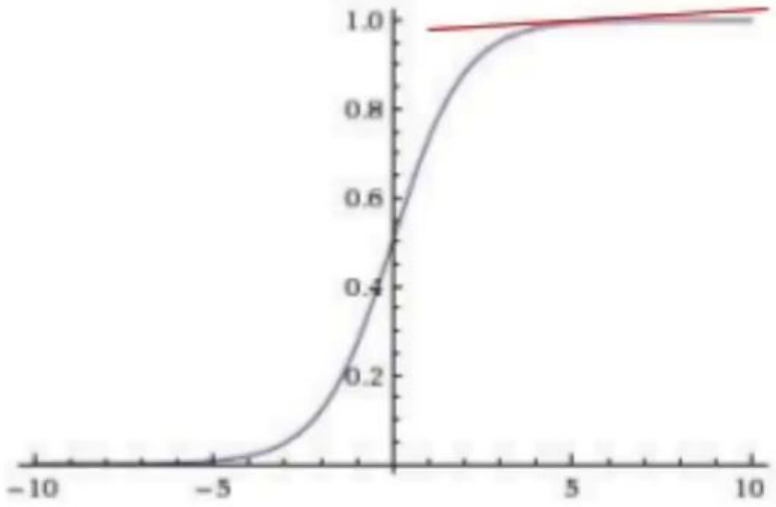
Problems with sigmoids

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

0 < output < 1

Avoid in practice

- Sigmoids saturate and kill gradients
- Sigmoids slow convergence
- Sigmoids are not zero-centered
- OK to use on last layer



NN Codelab

http://tiny.cc/ml_nn

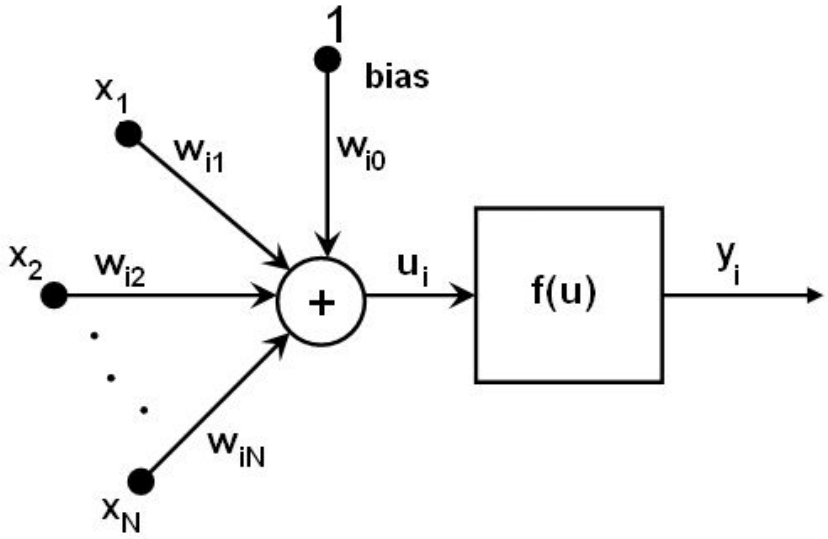
GDG RAJKOT

Sigmoid and Numpy

```
import numpy as np


def sigmoid(x):
    return 1.0/(1 + np.exp(-x))
def sigmoid_derivative(x):
    return x * (1.0 - x)
```

GDG RAJKOT

Weight and biases



Neural Network class

```
class NeuralNetwork:
  def __init__(self, x, y):
     self.input     = x
     self.weights1   = np.random.rand(self.input.shape[1],4)
     self.weights2   = np.random.rand(4,1)
     self.y              = y
     self.output      = np.zeros(self.y.shape)
```

GDG RAJKOT

Training

Output of simple 2 layer neural network is:

$$Y = \sum (weight * input) + bias$$

GDG RAJKOT

Feedforward

Calculating the predicted output, $\hat{y}$

$$\hat{y} = \sigma(W_2\sigma(W_1x + b_1) + b_2)$$

GDG RAJKOT

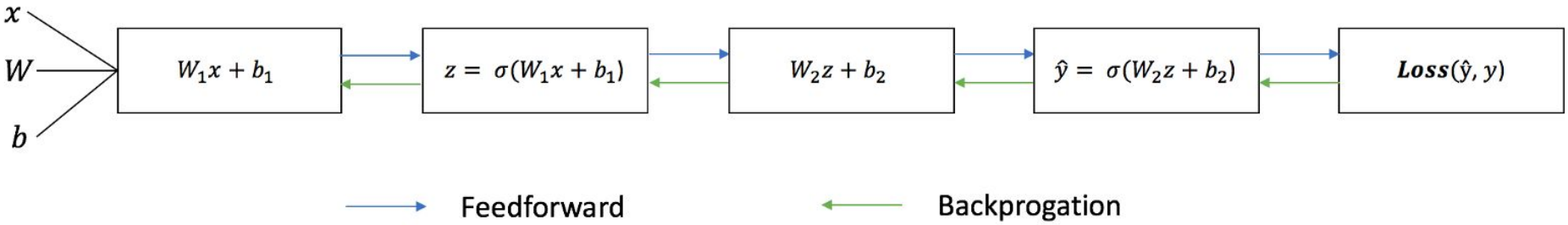
Feedforward

```
def feedforward(self):
    self.layer1 = sigmoid(np.dot(self.input, self.weights1))
    self.layer2 = sigmoid(np.dot(self.input, self.weights2))
```

GDG RAJKOT

Backpropogation

Updating the weights and biases



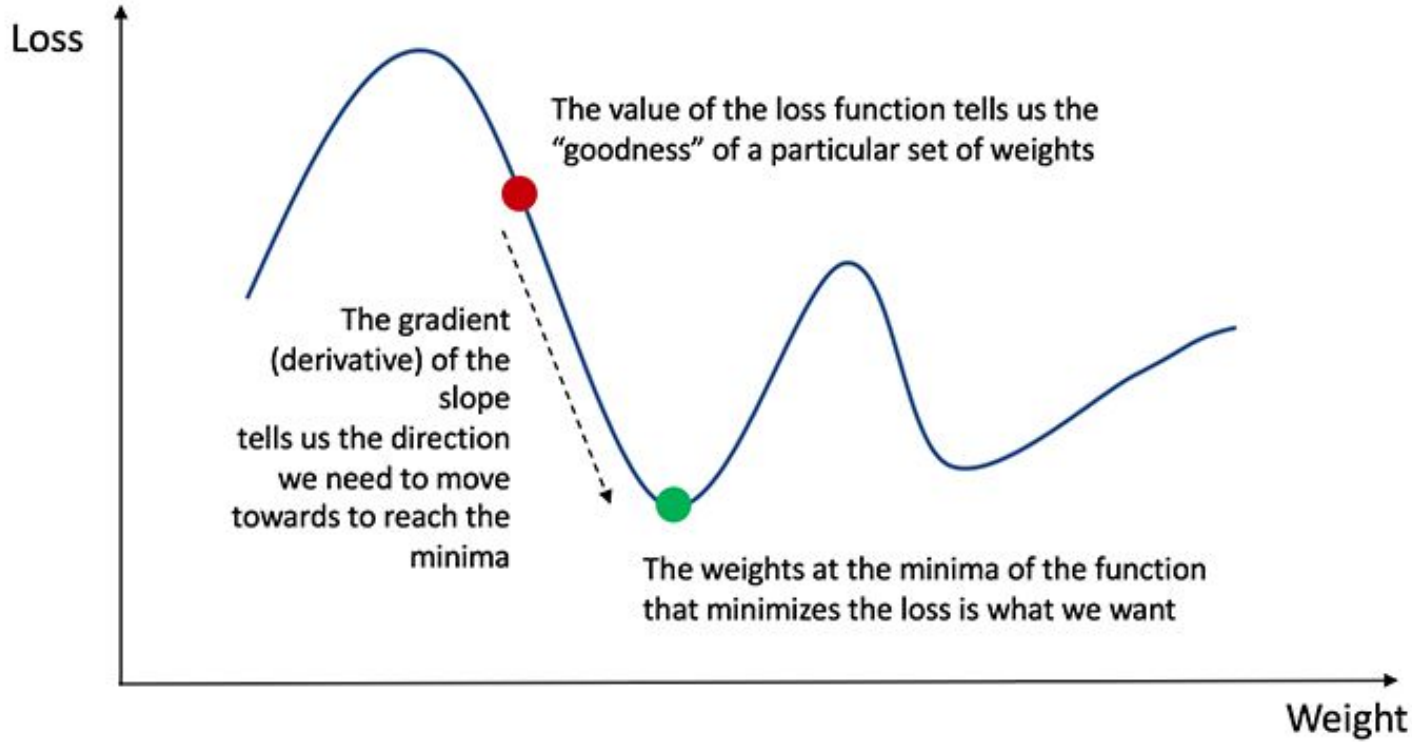
GDG RAJKOT

Loss Function

$$Sum - of - Squares\ Error = \sum_{i=1}^{n} (y - \hat{y})^2$$

GDG RAJKOT

Gradient Descent Algorithm



GDG RAJKOT

Derivative of Loss

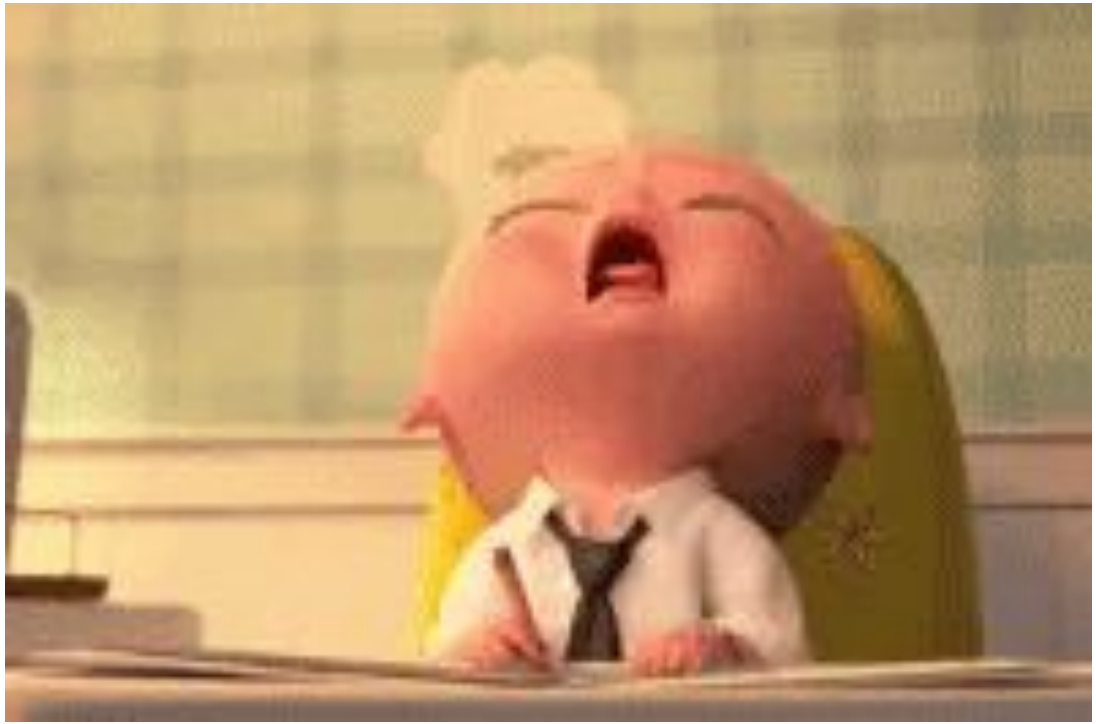
$$Loss(y, \hat{y}) = \sum_{i=1}^{n} (y - \hat{y})^2$$

$$\frac{\partial\, Loss(y,\hat{y})}{\partial W} = \frac{\partial Loss(y,\hat{y})}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial z} * \frac{\partial z}{\partial W} \qquad \text{where } z = Wx + b$$

$$= 2(y - \hat{y}) * \text{derivative of sigmoid function} * x$$

$$= 2(y - \hat{y}) * z(1\text{-}z) * x$$

I know, math sucks



GDG RAJKOT

Backpropogation

```
def backprop(self):
  d_weights2 = np.dot(self.layer1.T,
                      (2*(self.y - self.output) * sigmoid_derivative(self.output)))
  d_weights1 = np.dot(self.input.T,
                      (np.dot(2*(self.y - self.output) * sigmoid_derivative(self.output),
                      self.weights2.T) * sigmoid_derivative(self.layer1)))

  self.weights1 += d_weights1
  self.weights2 += d_weights2
```

Don’t worry, I’ve already wrote this code for you guys in notebook ☻

GDG RAJKOT

ML in real life?

It’s showtime

GDG RAJKOT

Search your photos

Videos

Movies

Animations

Collages

SHOW MORE

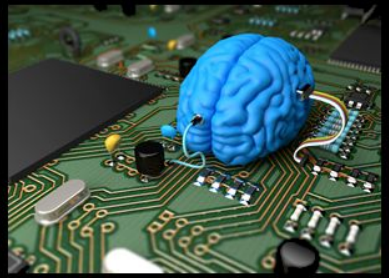
GDG RAJKOT

Once upon a time...

GDG RAJKOT

Deep Learning

What society thinks I do



What my friends think I do

What other computer scientists think I do

What mathematicians think I do

What I think I do

```
In [1]:
import keras
Using TensorFlow backend.
```

What I actually do

Keras Basic

- High level neural network API
- Capable of running on top of -
  - Tensorflow
  - CNTK
  - Theano

GDG RAJKOT

Sequential Model

```
from keras.models import Sequential

model = Sequential()
```

GDG RAJKOT

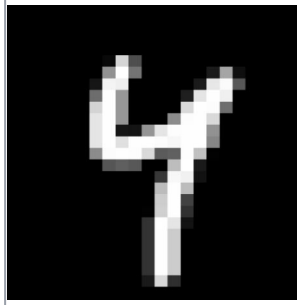
Add more layers?

```
from keras.layers import Dense

model.add(Dense(units=64, activation='relu', input_dim=100))
model.add(Dense(units=10, activation='softmax'))
```

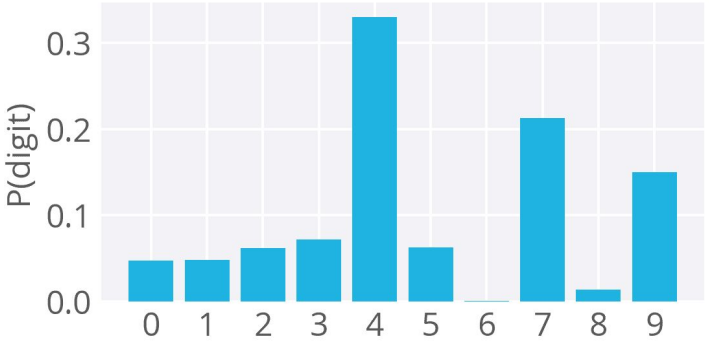
GDG RAJKOT

WTH is Softmax?

- Sigmoid is kind of Multi Class Sigmoid



$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$



GDG RAJKOT

Compile

```
model.compile(loss='categorical_crossentropy',
        optimizer='sgd',
        metrics=['accuracy'])
```

GDG RAJKOT

Train and Test

```
model.fit(x_train, y_train, epochs=5, batch_size=32)
loss_and_metrics = model.evaluate(x_test, y_test, batch_size=128)
```

GDG RAJKOT

Epoch & Batch size

- Epoch:
  - One forward and one backward pass of all the training samples.

- Batch size:
  - Number of training samples in one epoch.

GDG RAJKOT

Handwritten Digit Classification

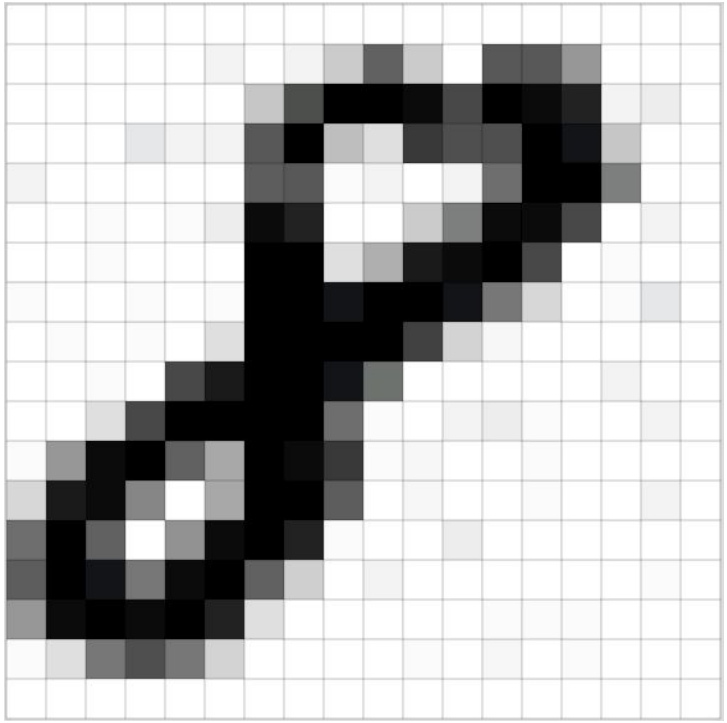
with CNN

GDG RAJKOT

http://tiny.cc/ml_digit

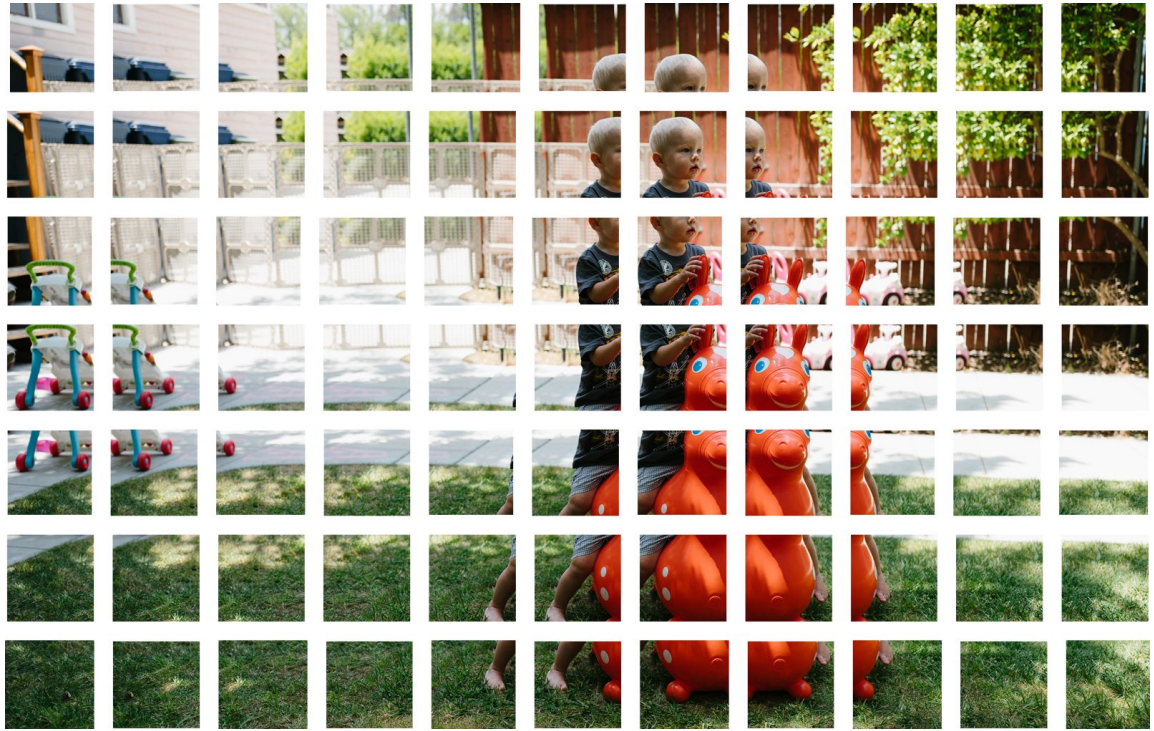
GDG RAJKOT

MNIST



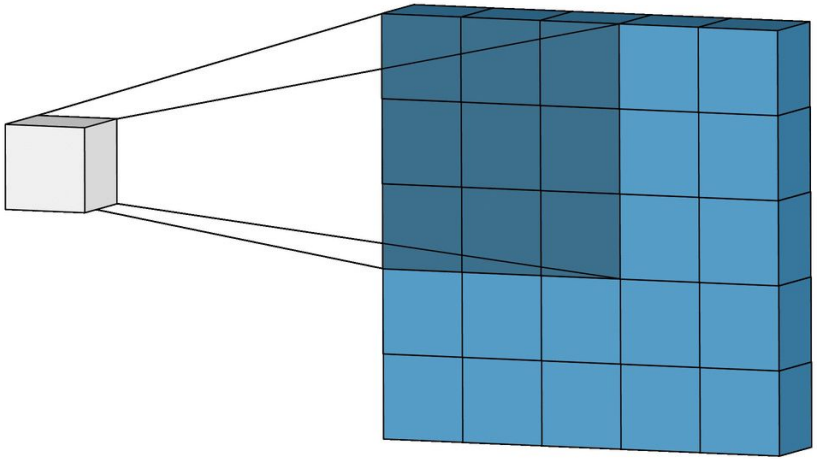
Convolution

Convolution



GDG RAJKOT

Convolution



$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$	0	0
$0_{\times 0}$	$1_{\times 1}$	$1_{\times 0}$	1	0
$0_{\times 1}$	$0_{\times 0}$	$1_{\times 1}$	1	1
0	0	1	1	0
0	1	1	0	0

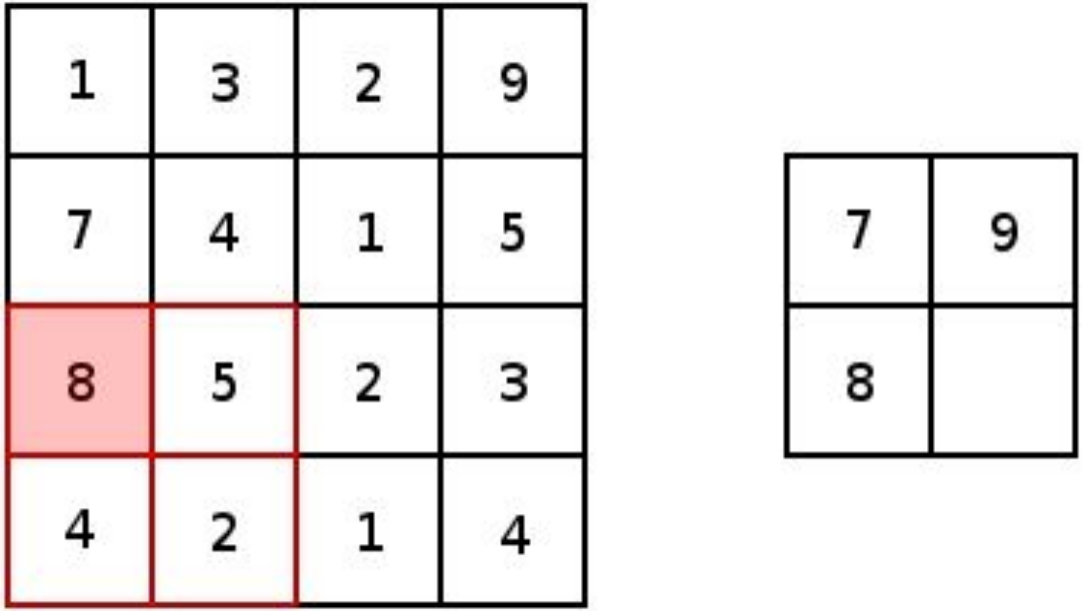
Image

4		

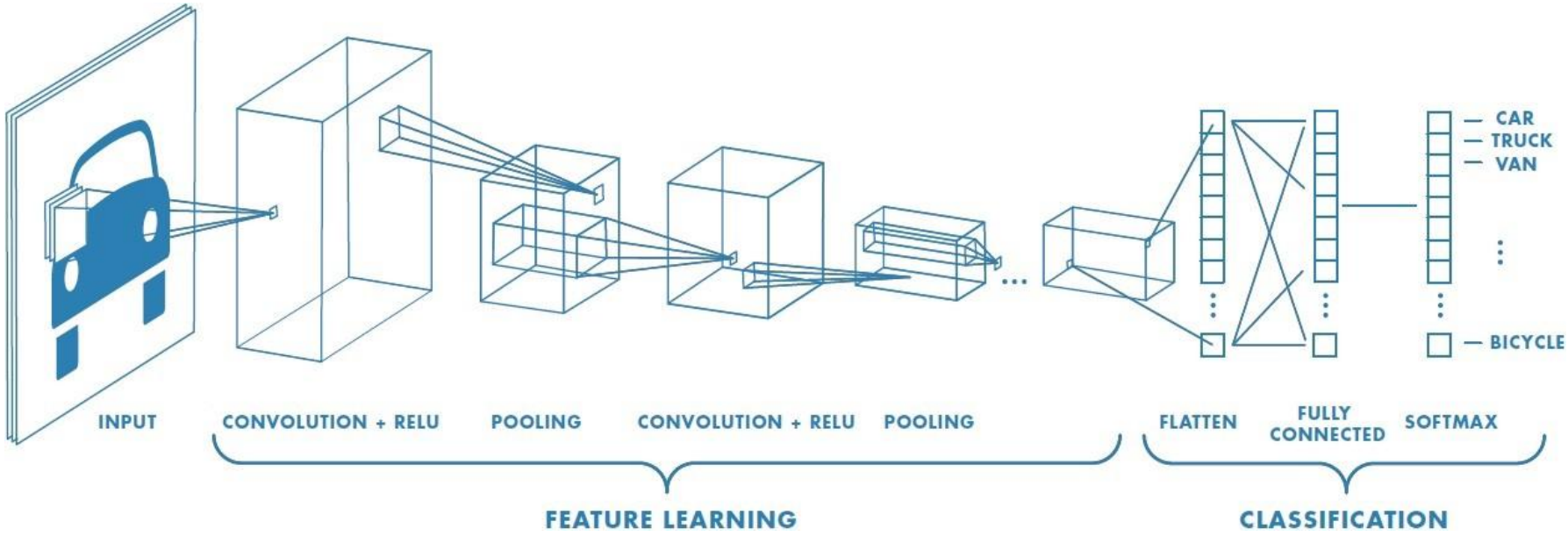
Convolved Feature

GDG RAJKOT

MaxPooling



Convolutional Neural Network (CNN)



GDG RAJKOT

Load Data

```
img_rows, img_cols = 28, 28
batch_size = 128
num_classes = 10
epochs = 12

(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

GDG RAJKOT

Plot Images and Labels

```
plt.subplot(221)
plt.imshow(x_train[0], cmap=plt.get_cmap('gray'))
plt.title(y_train[0])
```

GDG RAJKOT

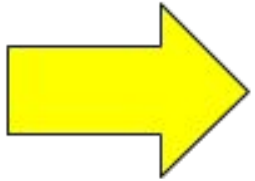
Preprocess Input Data

- Reshape
- Convert data type to float32
- Normalize to the range [0,1]

GDG RAJKOT

One Hot Encoder

Color
Red
Red
Yellow
Green
Yellow



Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

GDG RAJKOT

Preprocess Class Labels

y_train = keras.utils.to_categorical(y_train, num_classes)

y_test = keras.utils.to_categorical(y_test, num_classes)

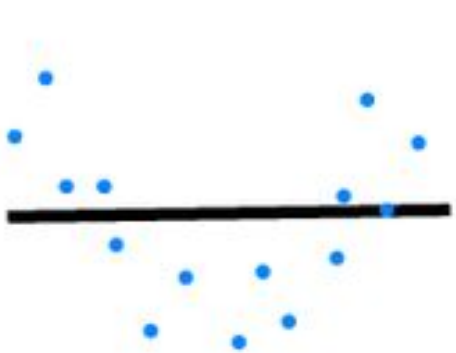
GDG RAJKOT

Model Architecture

```
model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=input_shape))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

GDG RAJKOT

Overfitting



Underfitting

Desired

Overfitting

Model Architecture

```
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

print(model.summary())
```

GDG RAJKOT

Deploy

- REST API
- Tensorflow Serving
- Tensorflow Extended (TFX)

GDG RAJKOT

Recap

- Basic Machine Learning
- Applications of Machine Learning
- Neural Network
- Image Classification
- Deployment

GDG RAJKOT

Too much ML?

Now what?

http://tiny.cc/ml_learn

- Machine Learning Crash Course, a course from Google that introduces machine learning concepts.
- CS 20: Tensorflow for Deep Learning Research, notes from an intro course from Stanford.
- CS231n: Convolutional Neural Networks for Visual Recognition, a course that teaches how convolutional networks work.
- Machine Learning Recipes, a video series that introduces basic machine learning concepts with few prerequisites.
- Deep Learning with Python, a book by Francois Chollet about the Keras API, as well as an excellent hands on intro to Deep Learning.
- Hands-on Machine Learning with Scikit-Learn and TensorFlow, a book by Aurélien Geron's that is a clear getting-started guide to data science and deep learning.
- Deep Learning, a book by Ian Goodfellow et al. that provides a technical dive into learning machine learning.

GDG RAJKOT

Thanks to:

- Google Developers Group, Rajkot
- Women Techmakers, Rajkot
- Charmi Chokshi for content
- GDG India Community members
- Tensorflow for being awesome !
- Keras for being even more awesome !!!

NO QUESTIONS PLEASE 👽