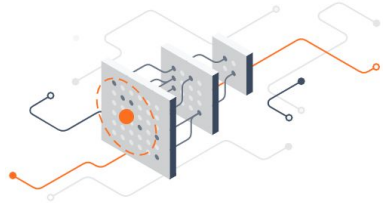
Machine Learning for Everyone

By Aaron Ma

I am Aaron Ma



- 11 years old
- Youngest TensorFlow contributor
- Robotics, machine learning, algorithms, self driving cars
- AI, ML, TensorFlow, Python, C++, HTML, CSS, TypeScript, JavaScript, Angular, Swift, ROS, AWS, Google Cloud, and more
- Tremendous passion in Computer Science
- Hard-core software and hardware
- Youngest graduated from Udacity Self Driving Car Engineer Nanodegree and more!
- Twitter **@aaronhma**

VERIFIED CERTIFICATE OF COMPLETION

June 18, 2019

UDACITY

Aaron Ma

Has successfully completed the

Self-Driving Car Engineer

NANODEGREE PROGRAM

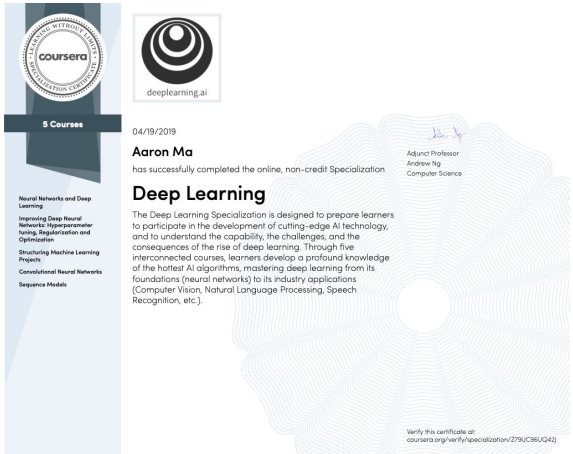
Sebastian Thrun
Founder, Udacity

Co-Created with

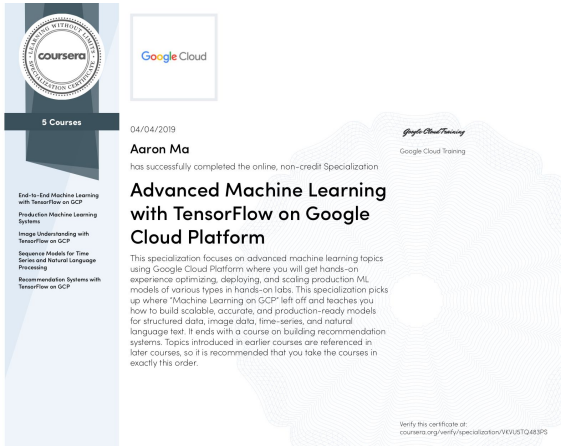
Mercedes Nvidia Uber ATG Didi BMW McLaren

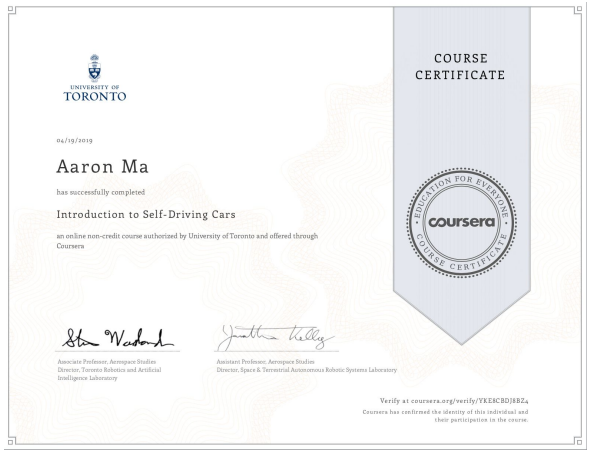
Udacity has confirmed the participation of this individual in this program.
Confirm program completion at confirm.udacity.com/C597KLNL

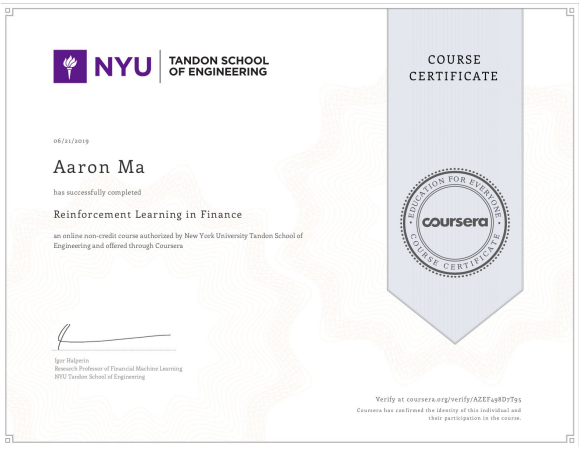

coursera
9 Courses
What is Data Science?
Open Source tools for Data Science
Data Science Methodology
Python for Data Science
Databases and SQL for Data Science
Data Analysis with Python
Data Visualization with Python
Machine Learning with Python
Applied Data Science Capstone
IBM
07/06/2019
Aaron Ma
has successfully completed the online, non-credit Specialization
IBM Data Science Professional Certificate
In this Professional Certificate learners developed and honed hands-on skills in Data Science and Machine Learning. Learners started with an orientation of Data Science and its Methodology, became familiar and used a variety of data science tools, learned Python and SQL, performed Data Visualization and Analysis, and created Machine Learning models. In the process they completed several labs and assignments on the cloud including a Capstone Project at the end to apply and demonstrate their knowledge and skills.
Joseph Santarcangelo
Senior Data Scientist
IBM
Polong Lin
Data Scientist, IBM
Alex Aklson, Ph.D.
Data Scientist
Rav Ahuja
AI & Data Science Program Director,
IBM Skills Network
Saeed Aghabozorgi
Sr. Data Scientist
IBM
Verify this certificate at:
coursera.org/verify/specialization/KQ2YJCKFBCD5

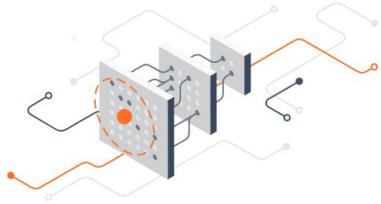

coursera
5 Courses
Neural Networks and Deep Learning
Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization
Structuring Machine Learning Projects
Convolutional Neural Networks
Sequence Models
deeplearning.ai
04/19/2019
Aaron Ma
has successfully completed the online, non-credit Specialization
Deep Learning
The Deep Learning Specialization is designed to prepare learners to participate in the development of cutting-edge AI technology, and to understand the capability, the challenges, and the consequences of the rise of deep learning. Through five interconnected courses, learners develop a profound knowledge of the hottest AI algorithms, mastering deep learning from its foundations (neural networks) to its industry applications (Computer Vision, Natural Language Processing, Speech Recognition, etc.).
Adjunct Professor
Andrew Ng
Computer Science
Verify this certificate at:
coursera.org/verify/specialization/Z79UC9EUQ42J


coursera
5 Courses
How Google does Machine Learning
Launching into Machine Learning
Intro to TensorFlow
Feature Engineering
Art and Science of Machine Learning
Google Cloud
07/07/2019
Aaron Ma
has successfully completed the online, non-credit Specialization
Machine Learning with TensorFlow on Google Cloud Platform
This five-course online specialization teaches course participants how to write distributed machine learning models that scale in Tensorflow, scale out the training of those models, and offer high-performance predictions. Also featured is the conversion of raw data to features in a way that allows ML to learn important characteristics from the data and bring human insight to bear on the problem. It also teaches how to incorporate the right mix of parameters that yields accurate, generalized models and knowledge of the theory to solve specific types of ML problems. Course participants experimented with end-to-end ML, starting from building an ML-focused strategy and progressing into model training, optimization, and productionalization with hands-on labs using Google Cloud Platform.
Verify this certificate at:
coursera.org/verify/specialization/PSEN6GC3CND3


coursera
5 Courses
End-to-End Machine Learning with TensorFlow on GCP
Production Machine Learning Systems
Image Understanding with TensorFlow on GCP
Sequence Models for Time Series and Natural Language Processing
Recommendation Systems with TensorFlow on GCP
Google Cloud
04/04/2019
Aaron Ma
has successfully completed the online, non-credit Specialization
Advanced Machine Learning with TensorFlow on Google Cloud Platform
This specialization focuses on advanced machine learning topics using Google Cloud Platform where you will get hands-on experience optimizing, deploying, and scaling production ML models of various types in hands-on labs. This specialization picks up where "Machine Learning on GCP" left off and teaches you how to build scalable, accurate, and production-ready models for structured data, image data, time-series, and natural language text. It ends with a course on building recommendation systems. Topics introduced in earlier courses are referenced in later courses, so it is recommended that you take the courses in exactly this order.
Google Cloud Training
Verify this certificate at:
coursera.org/verify/specialization/VKVUSTQ483PS


UNIVERSITY OF TORONTO
COURSE CERTIFICATE
04/19/2019
Aaron Ma
has successfully completed
Introduction to Self-Driving Cars
an online non-credit course authorized by University of Toronto and offered through Coursera
coursera
Associate Professor, Aerospace Studies
Director, Toronto Robotics and Artificial Intelligence Laboratory
Assistant Professor, Aerospace Studies
Director, Space & Terrestrial Autonomous Robotic Systems Laboratory
Verify at coursera.org/verify/YKE8CBDJ8BZ4
Coursera has confirmed the identity of this individual and their participation in the course.

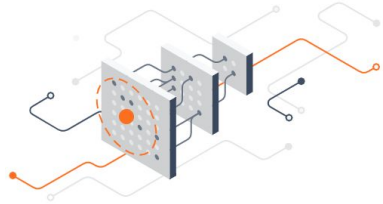

NYU TANDON SCHOOL OF ENGINEERING
COURSE CERTIFICATE
06/21/2019
Aaron Ma
has successfully completed
Reinforcement Learning in Finance
an online non-credit course authorized by New York University Tandon School of Engineering and offered through Coursera
coursera
Igor Halperin
Research Professor of Financial Machine Learning
NYU Tandon School of Engineering
Verify at coursera.org/verify/AZEFA98D7S5
Coursera has confirmed the identity of this individual and their participation in the course.



YOU


WELCOME BACK!


XXXMANTEDOYKISS.TUMBLR.



Journey to the Moon (maybe coming back!?!)

Blast off!

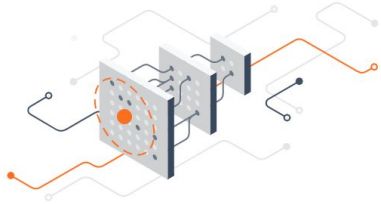
TensorFlow

ML Core Concepts


UNACCEPTABLE!
CN

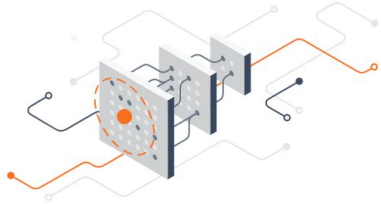

KOHL'S

What is ML?



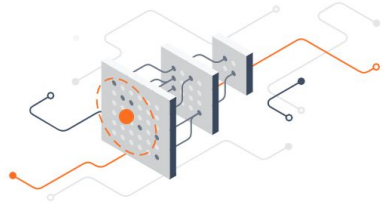
- Complex algorithm and statistical models

What is ML?



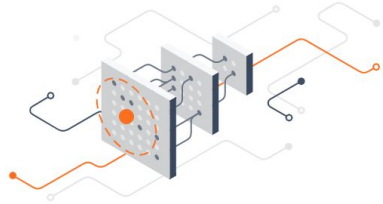
- Complex algorithm and statistical models
- Ability to learn and improve from experience without being explicitly programmed

What is ML?

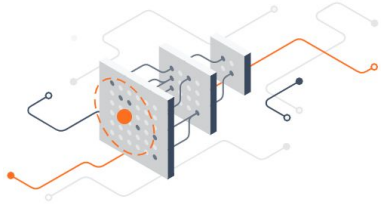


- Complex algorithm and statistical models
- Ability to learn and improve from experience without being explicitly programmed
- Allow computers to learn automatically without human intervention or assistance and adjust actions accordingly

What is ML?



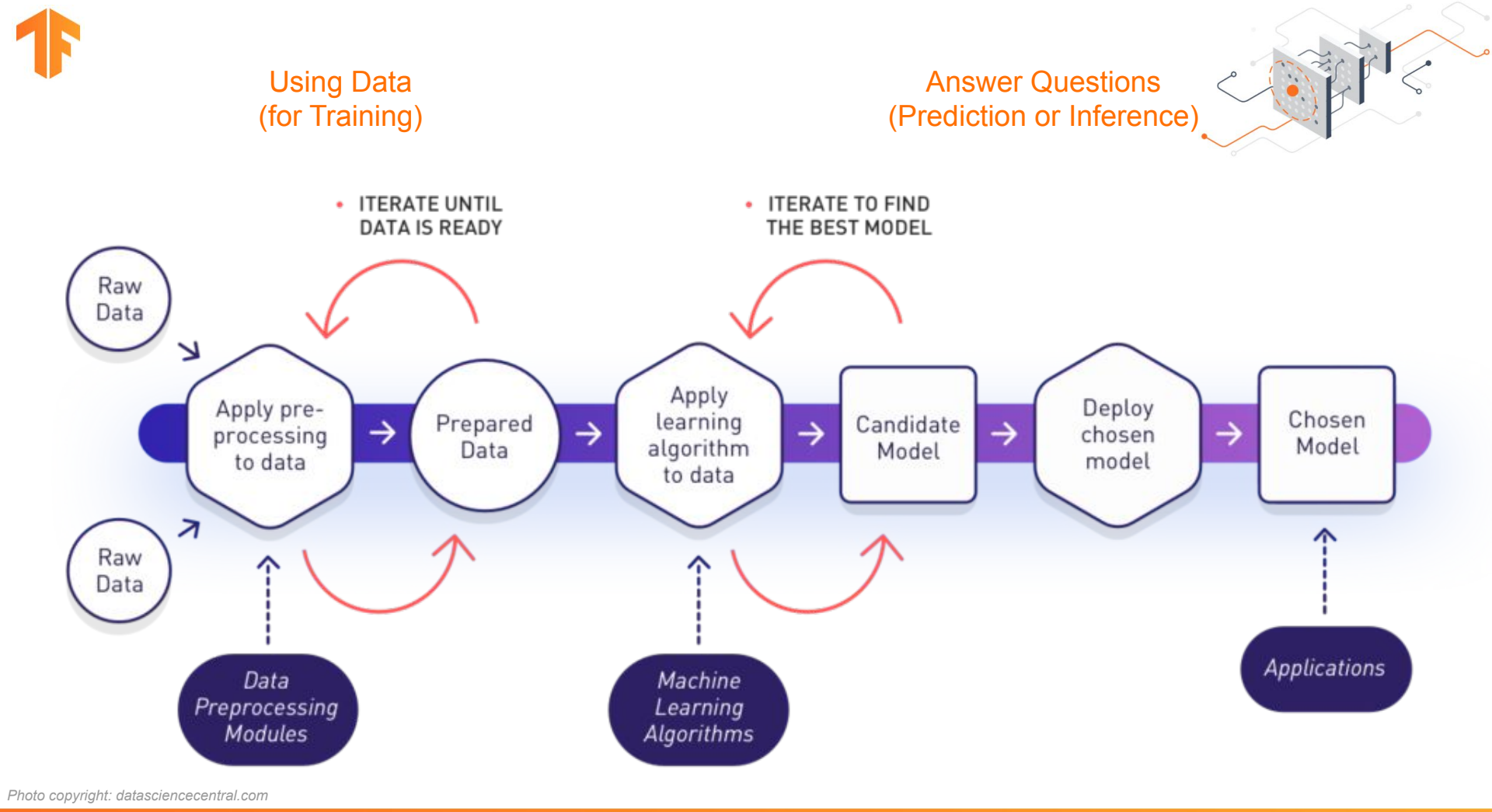
- Complex algorithm and statistical models
- Ability to learn and improve from experience without being explicitly programmed
- Allow computers to learn automatically without human intervention or assistance and adjust actions accordingly
- The goal is to find the features parameters (e.g. weights and biases) by tweaking input hyperparameters

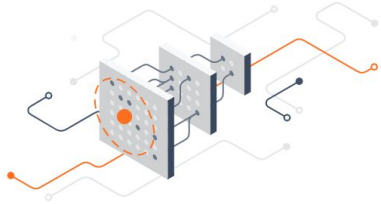


Machine Learning is

NOT Magic.

Rather tools and technology to **answer questions with data**!





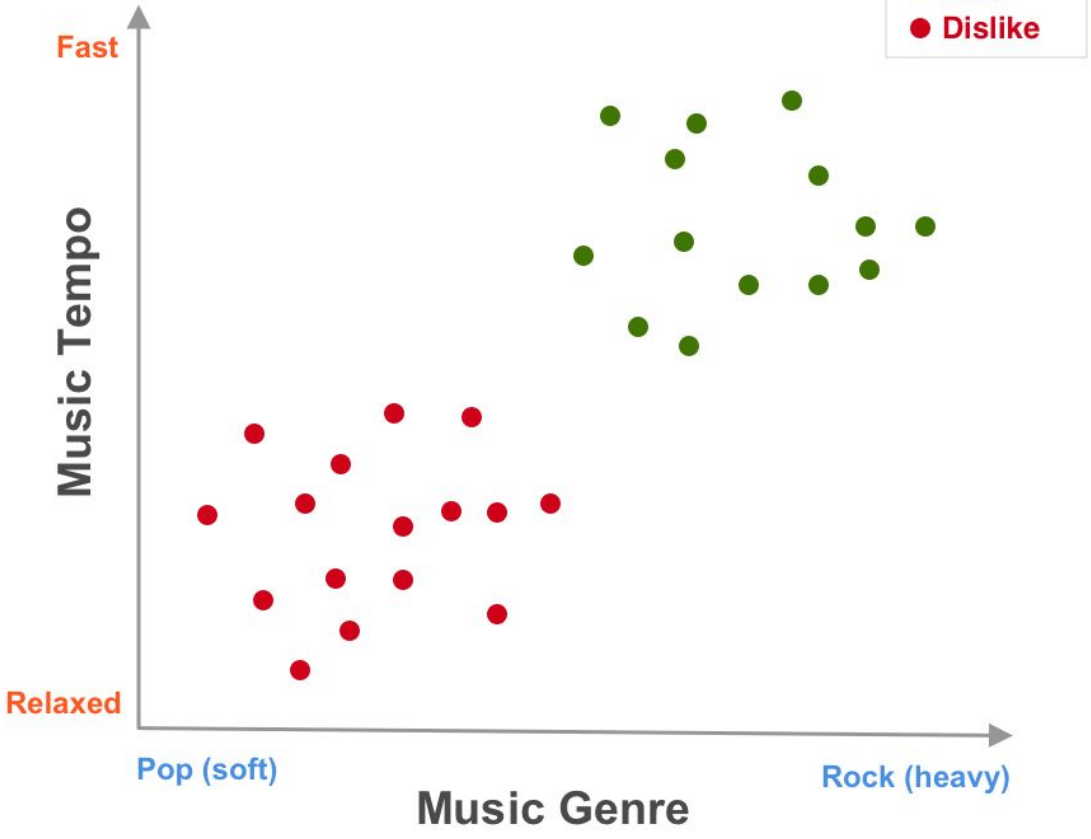
Human Friendly Way

Let's meet John

- Fast tempo
- Rock

- Slow tempo
- Pop

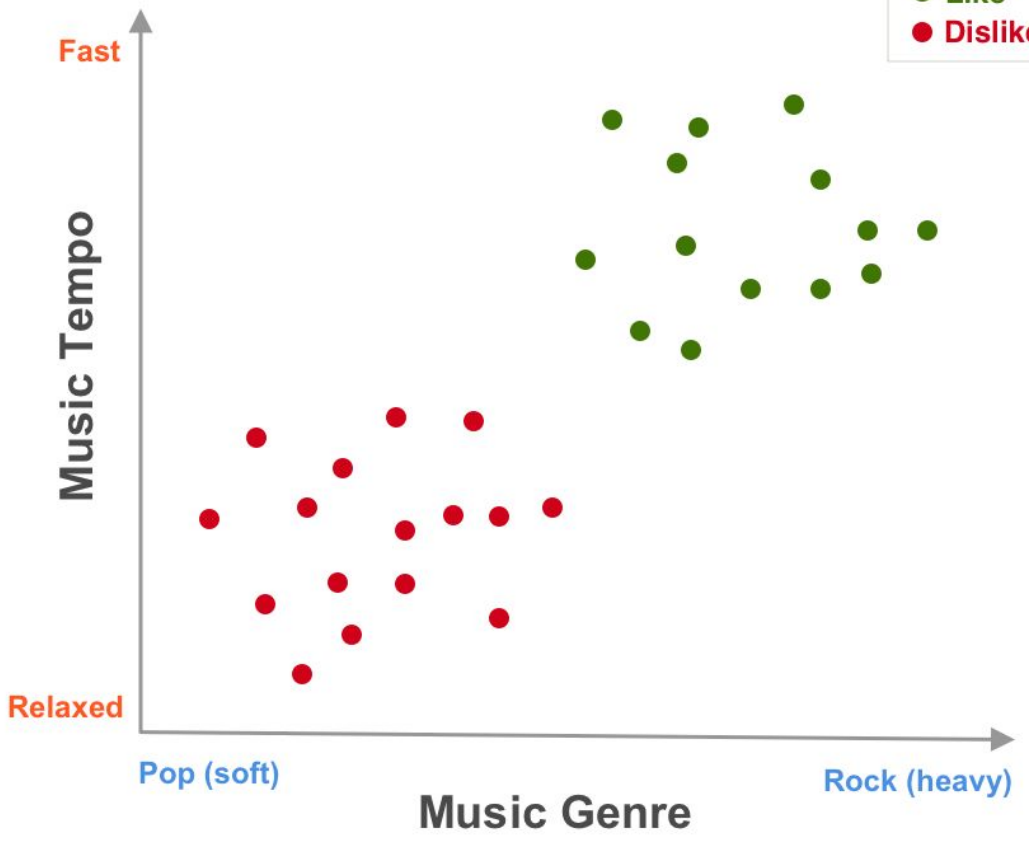
John


Legend
Like
Dislike
Fast
Music Tempo
Relaxed
Pop (soft)
Rock (heavy)
Music Genre

John

Listening a new song:

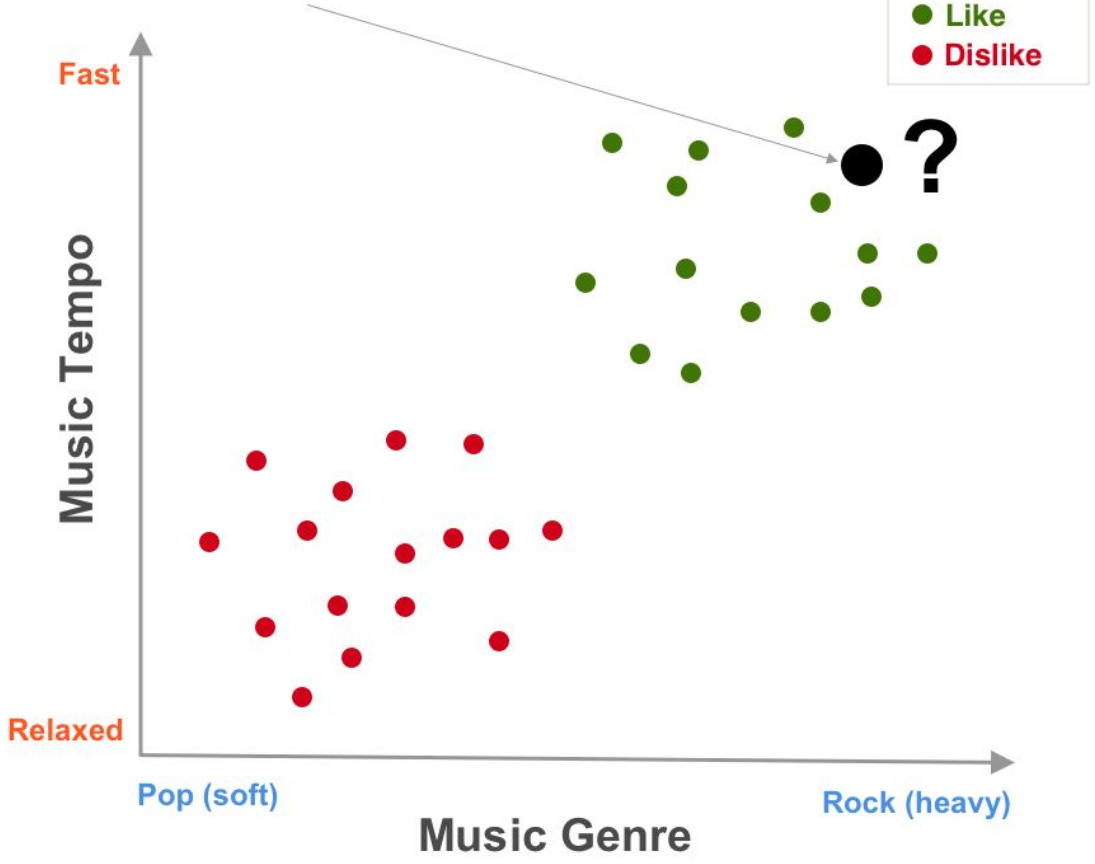
Metallica - Master of Puppets



John

Listening a new song:

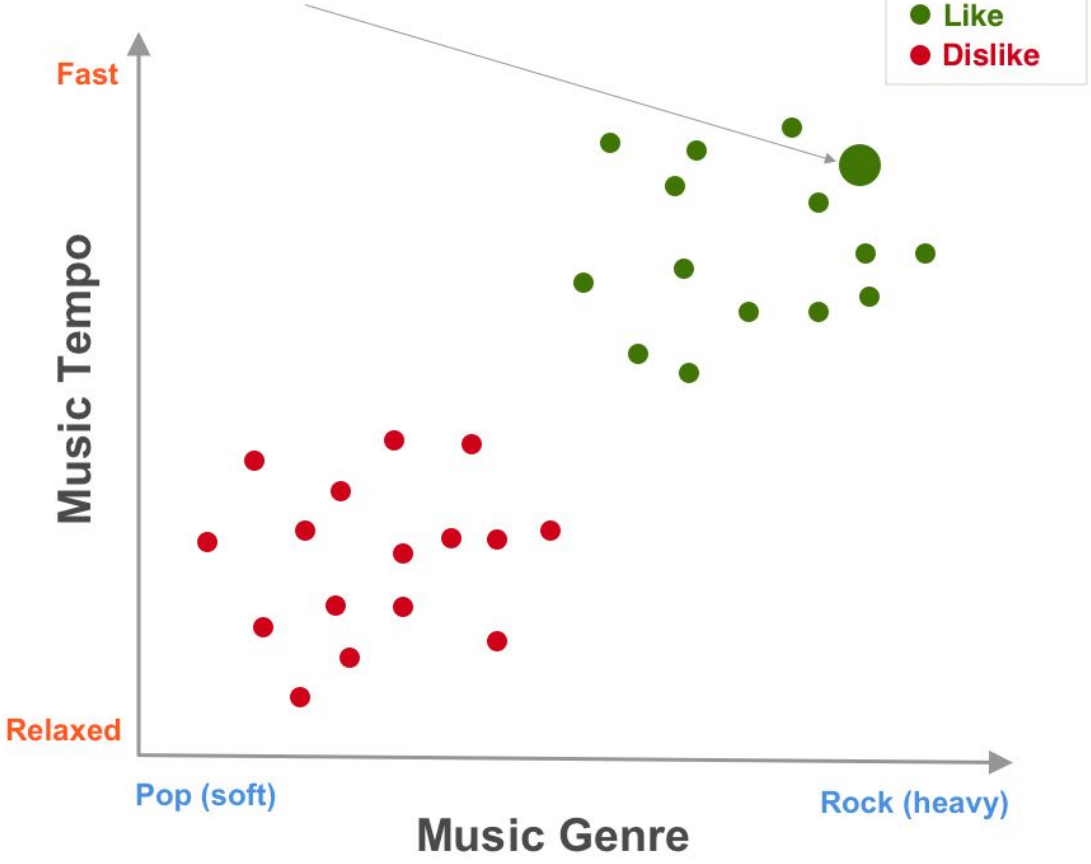
Metallica - Master of Puppets



John

Listening a new song:

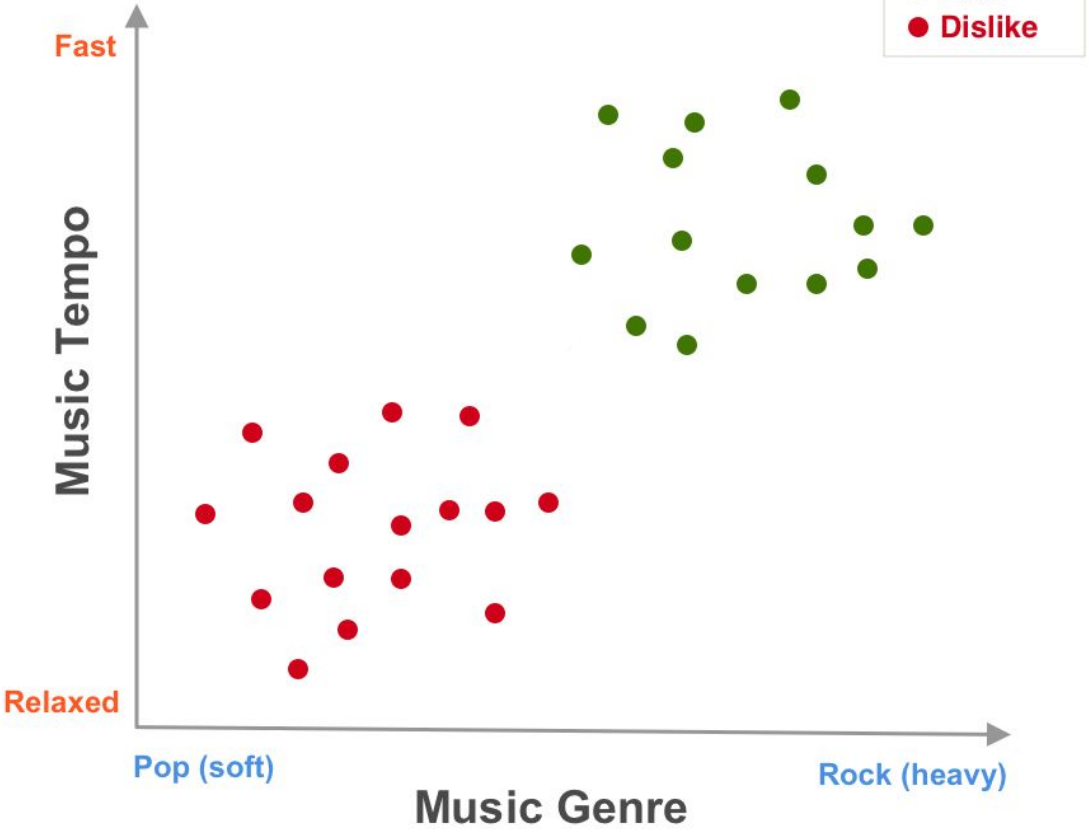
Metallica - Master of Puppets



Listening another new song:

Billie Eilish - Bad Guy

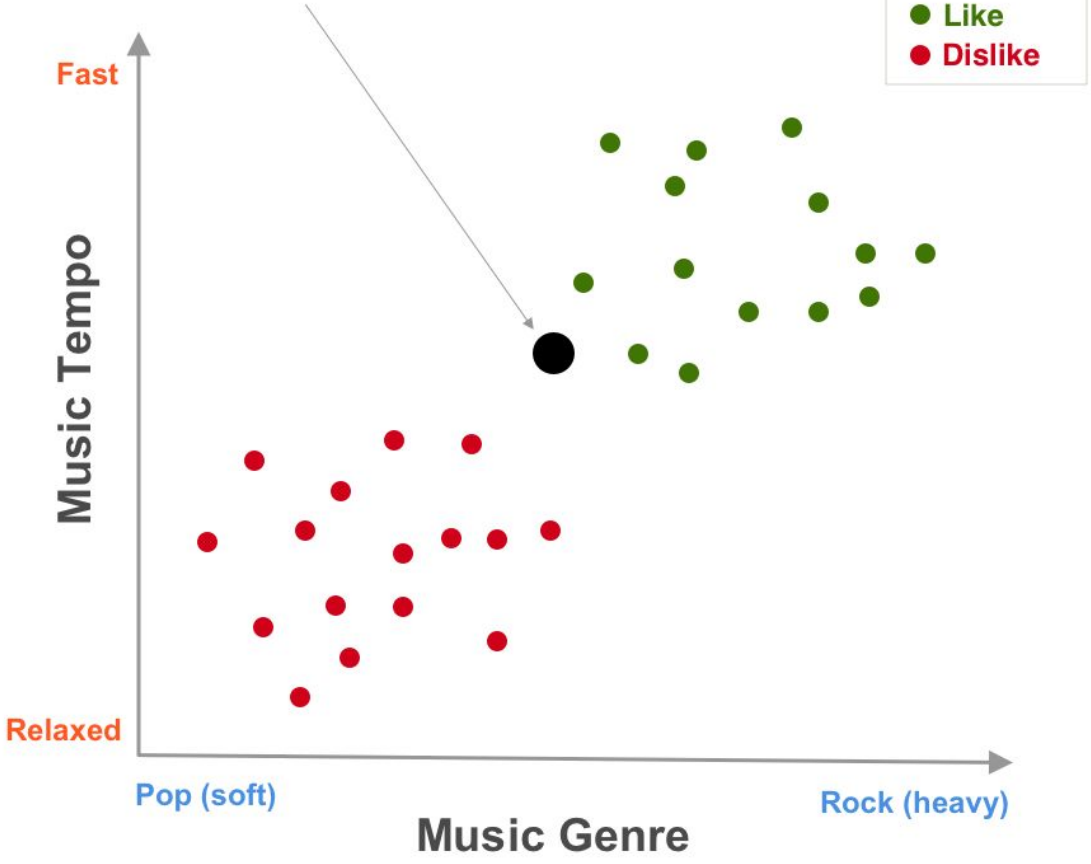
John



John

Listening another new song:

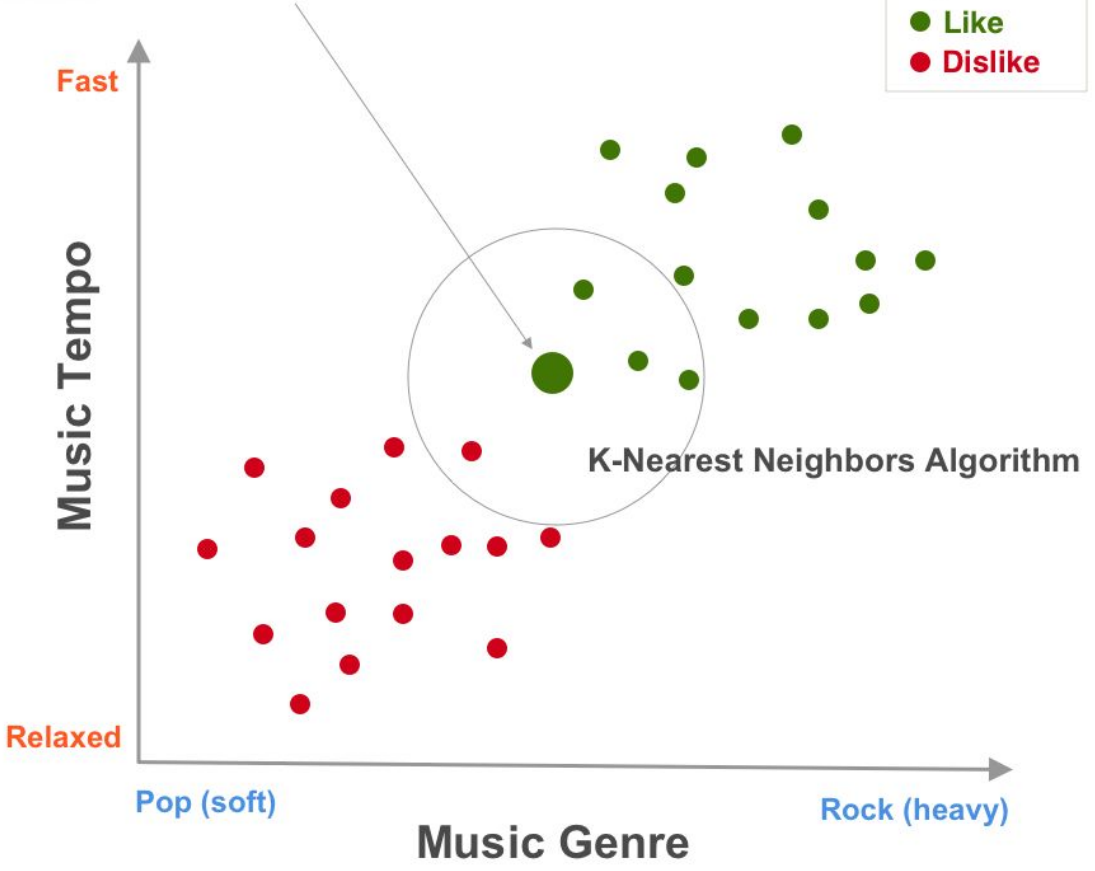
Billie Eilish - Bad Guy

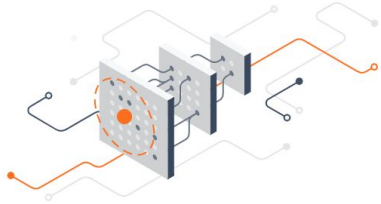


John

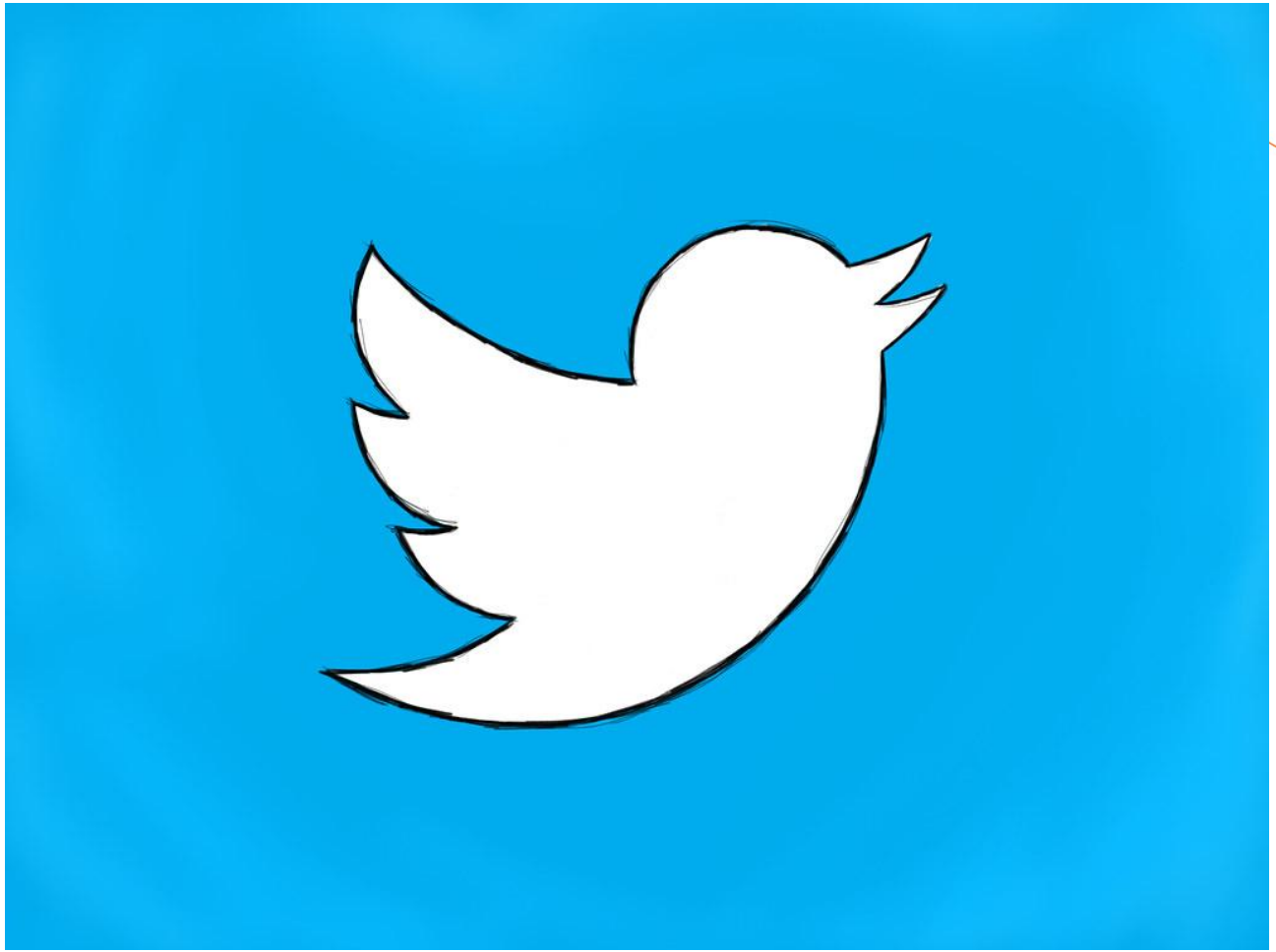
Listening another new song:

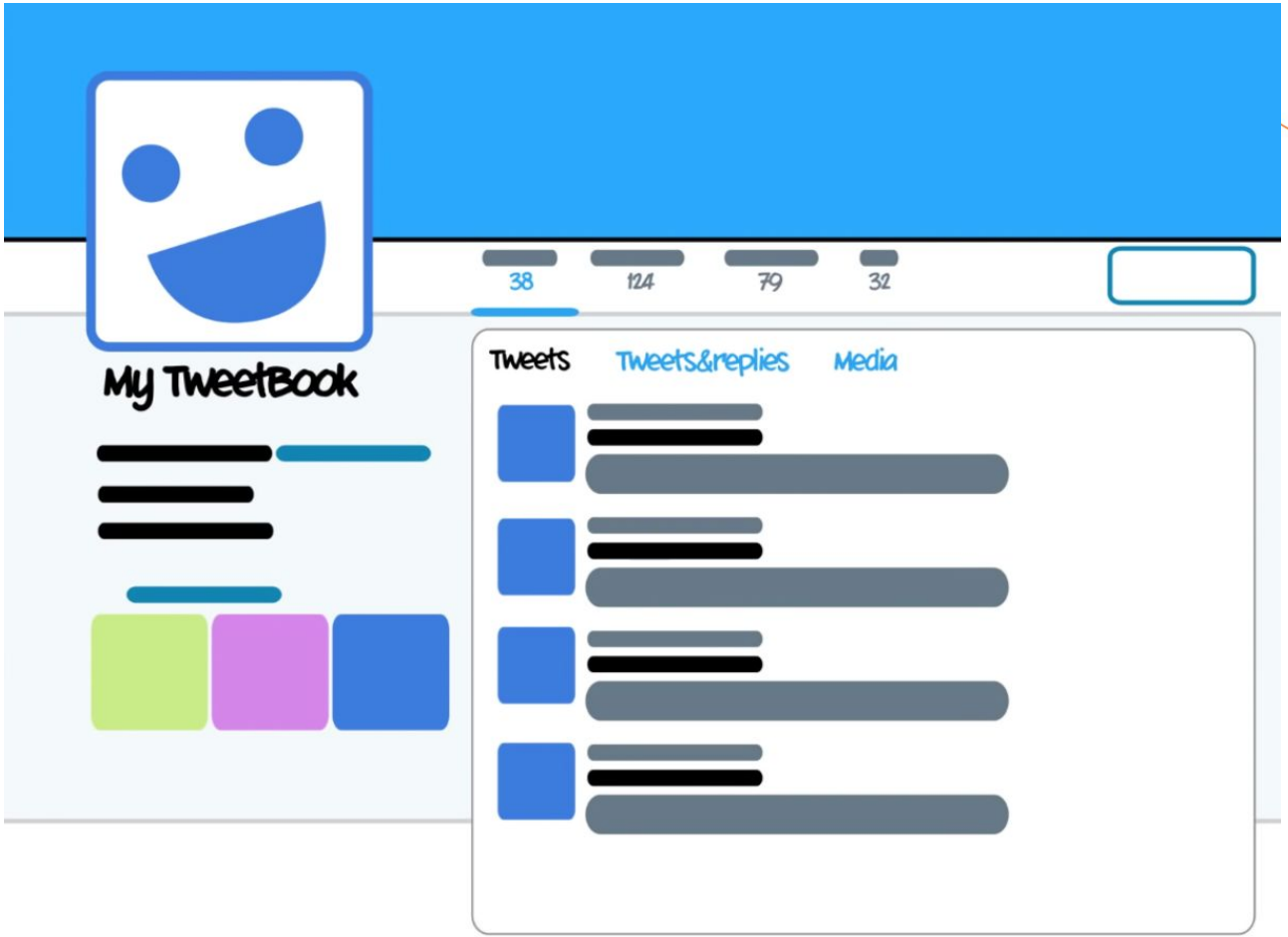
Billie Eilish - Bad Guy





Scenarios of ML

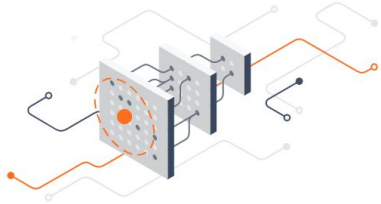



38
124
79
32
My TweetBook
Tweets
Tweets&replies
Media


My TweetBook
38
124
79
32
Tweets
Tweets&replies
Media

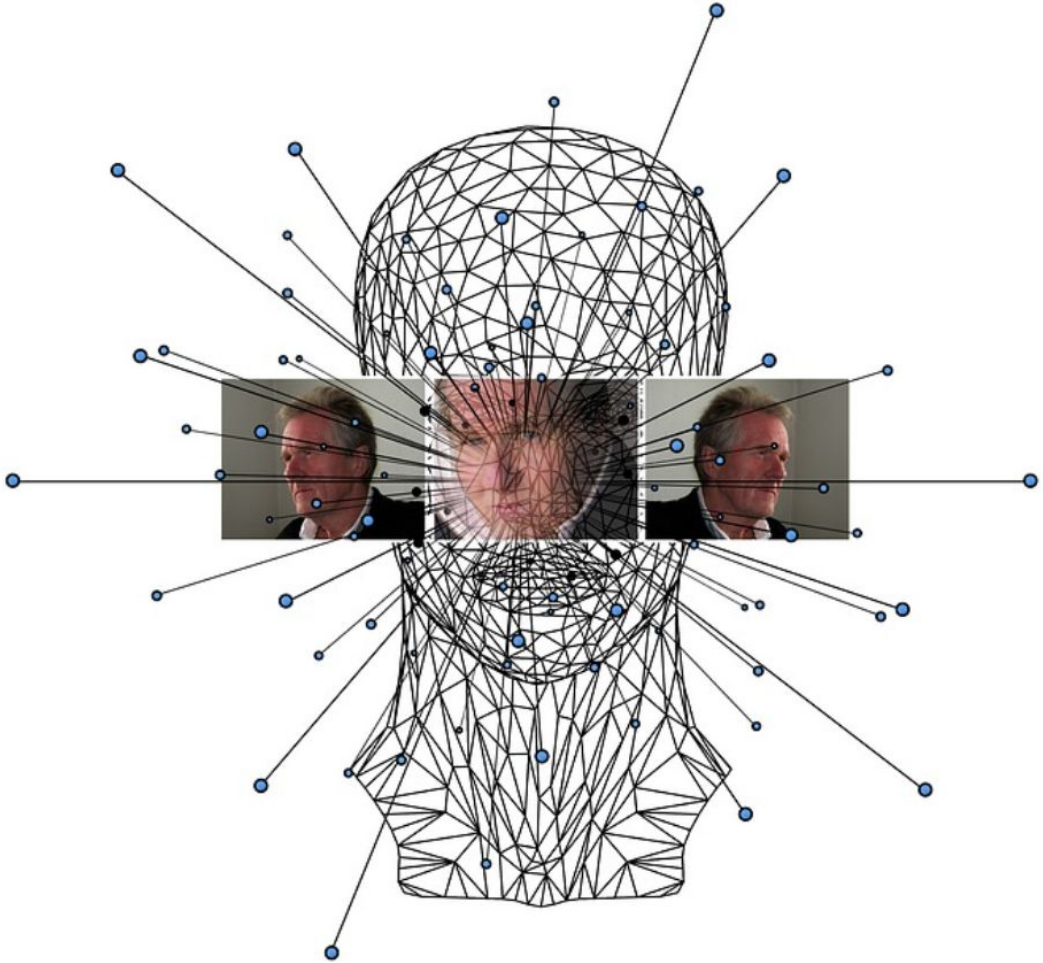

Content
Content is the
Content can be
quantify the
information


UBER

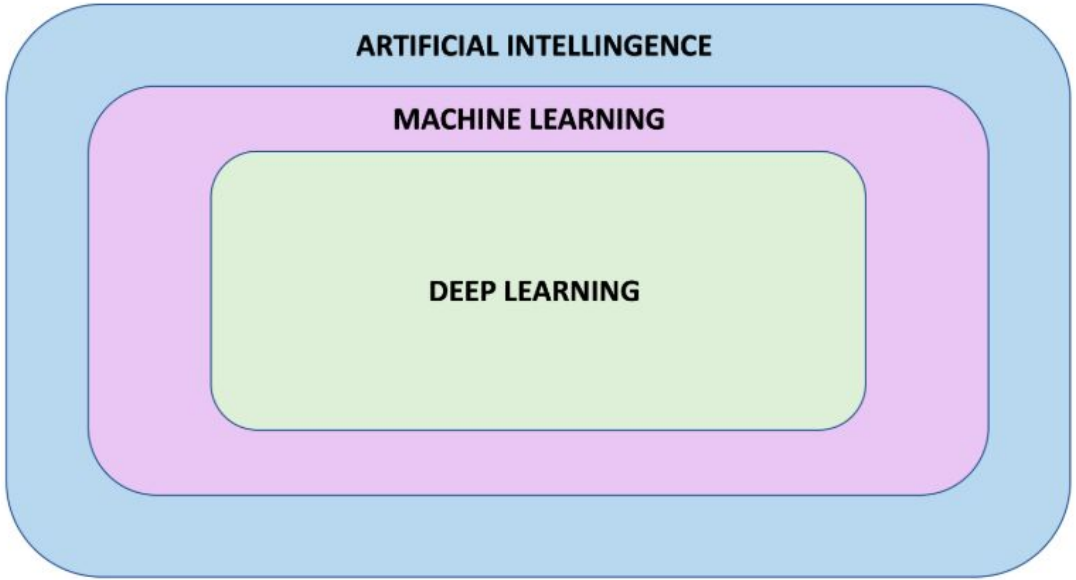


Differential pricing in real time based on:

- Demand (predictive model for area of high demand)
- Number of cars available
- Bad weather
- Rush hours / Real time traffics
- Travel distances
- Popular Destinations
- Special events (e.g Super Bowl, Concerts, etc)



Fields of Study in Machine Learning

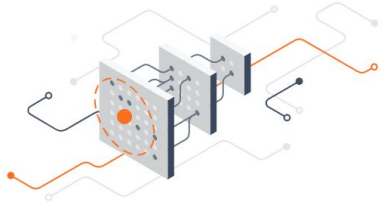


AI: The broad discipline of creating intelligent machines.

ML: A system that can learn from experience.

DL: A system that learn from experience on large data sets.

Types of Machine Learning

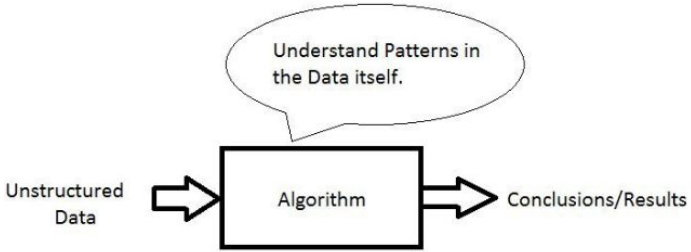


Supervised Learning

- **Task driven**
- Data -> input
- Label -> targets
- Iteration of training, algorithm finds desired output
- Regression and Classification problems

Unsupervised Learning

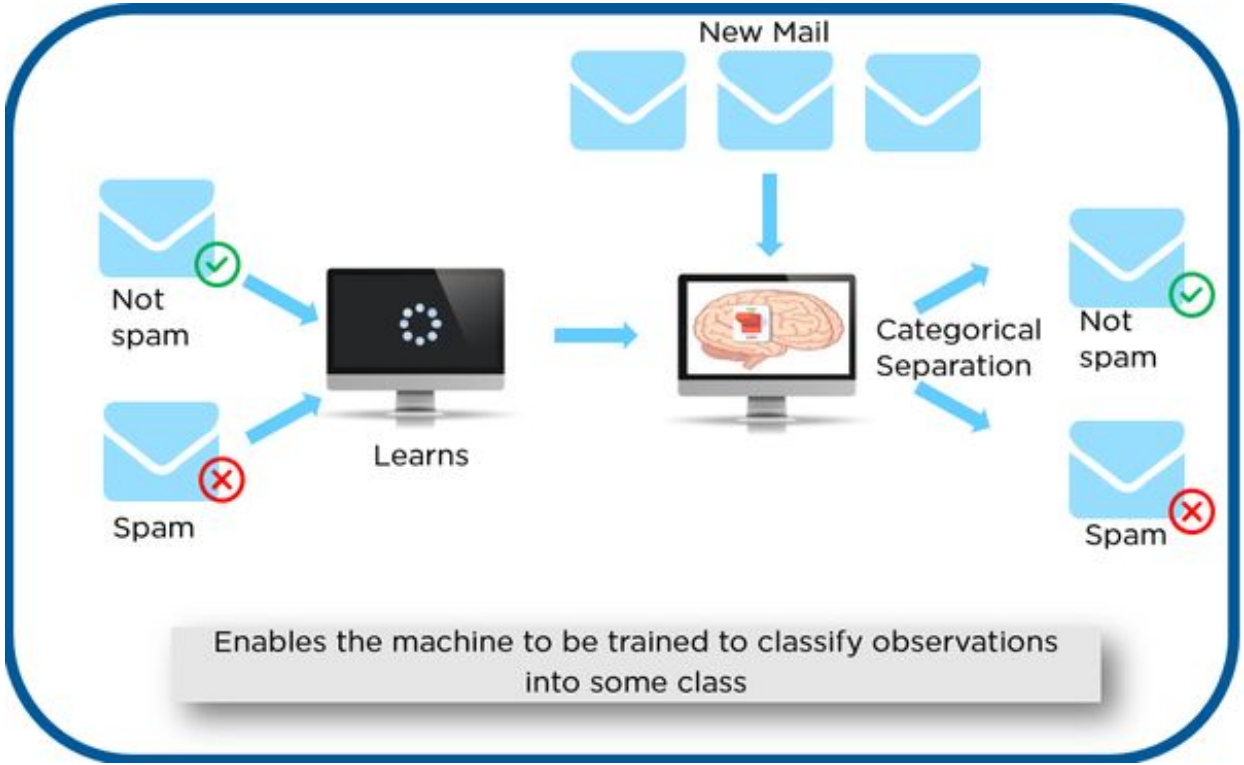
- **Data driven**
- No labels
- Not structured data
- Clustering problems (Group similar things)



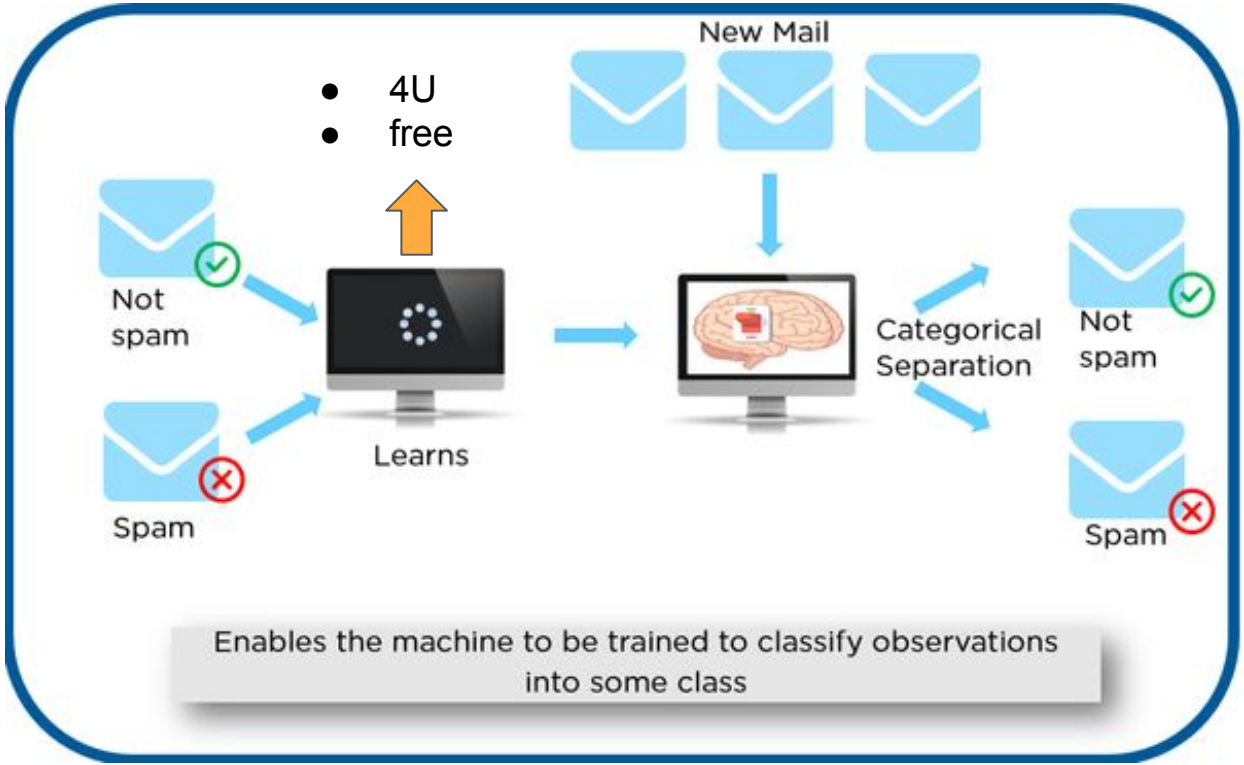
Reinforcement Learning

- **Algorithm driven**
- Reward-based learning by mapping state-action pairs to expected rewards
- learn from its experience
- Incentivize / Penalized

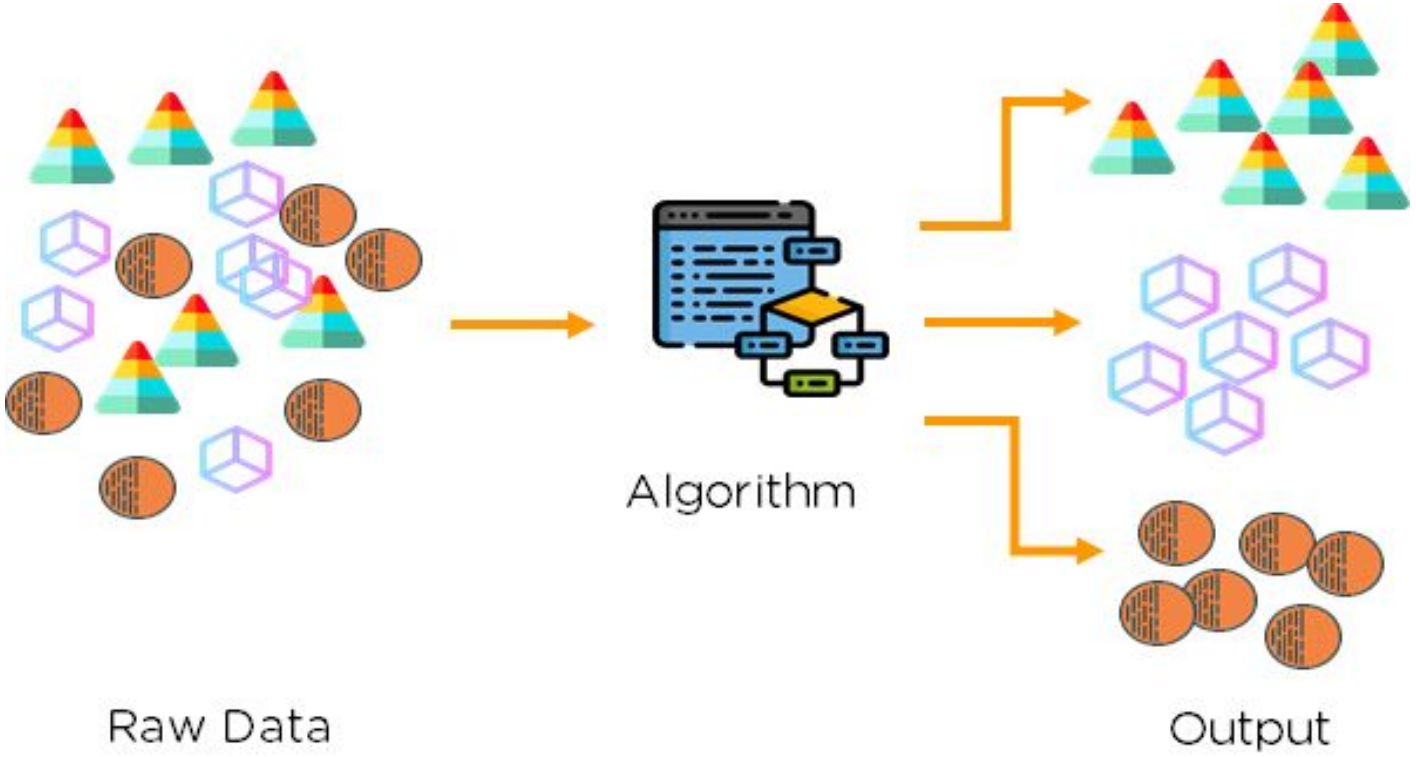
Supervised Learning: Classification



Supervised Learning: Classification

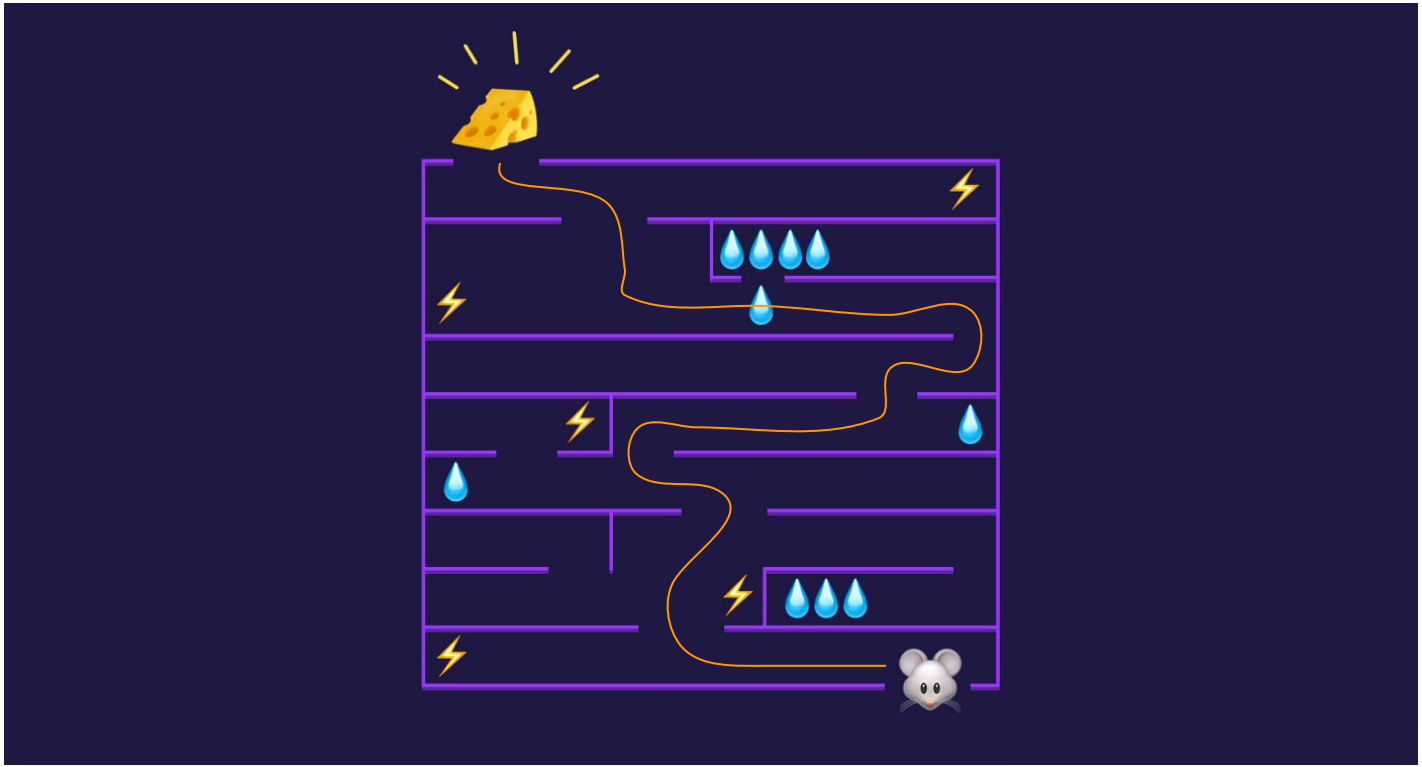


Unsupervised Learning: Clustering



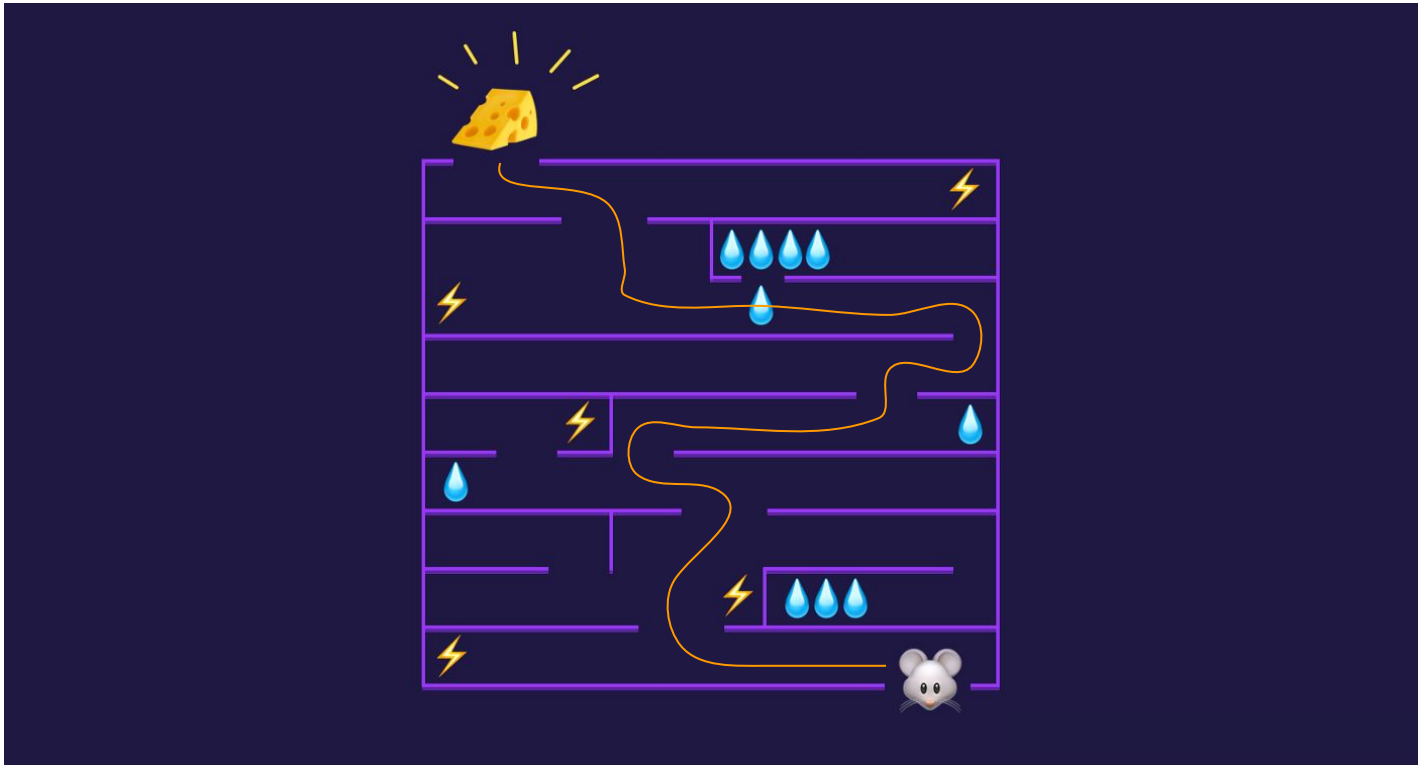
Reinforcement Learning

Reinforcement Learning



Reinforcement Learning

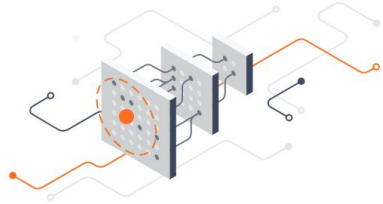
Reinforcement Learning



TensorFlow

Neural Network

What's a Neural Network?

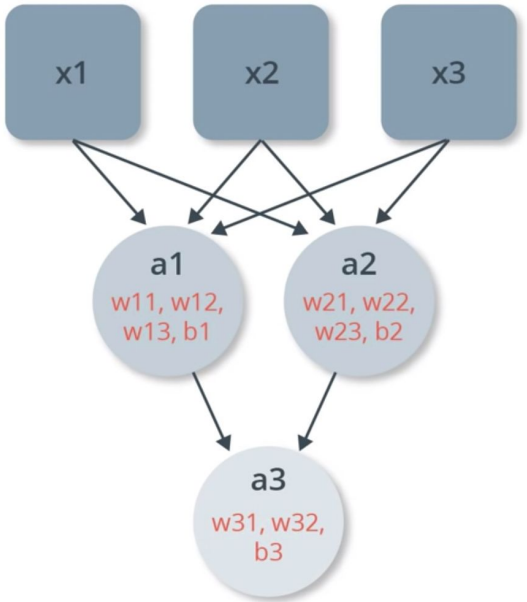


- Sort of computer software, inspired by the brain

What's a Neural Network?

- Sort of computer software, inspired by the brain
- Made up of units(aka neurons) to solve a problem together

A Neural Network



NN: a stack of layers.

Layer: is made up of units.

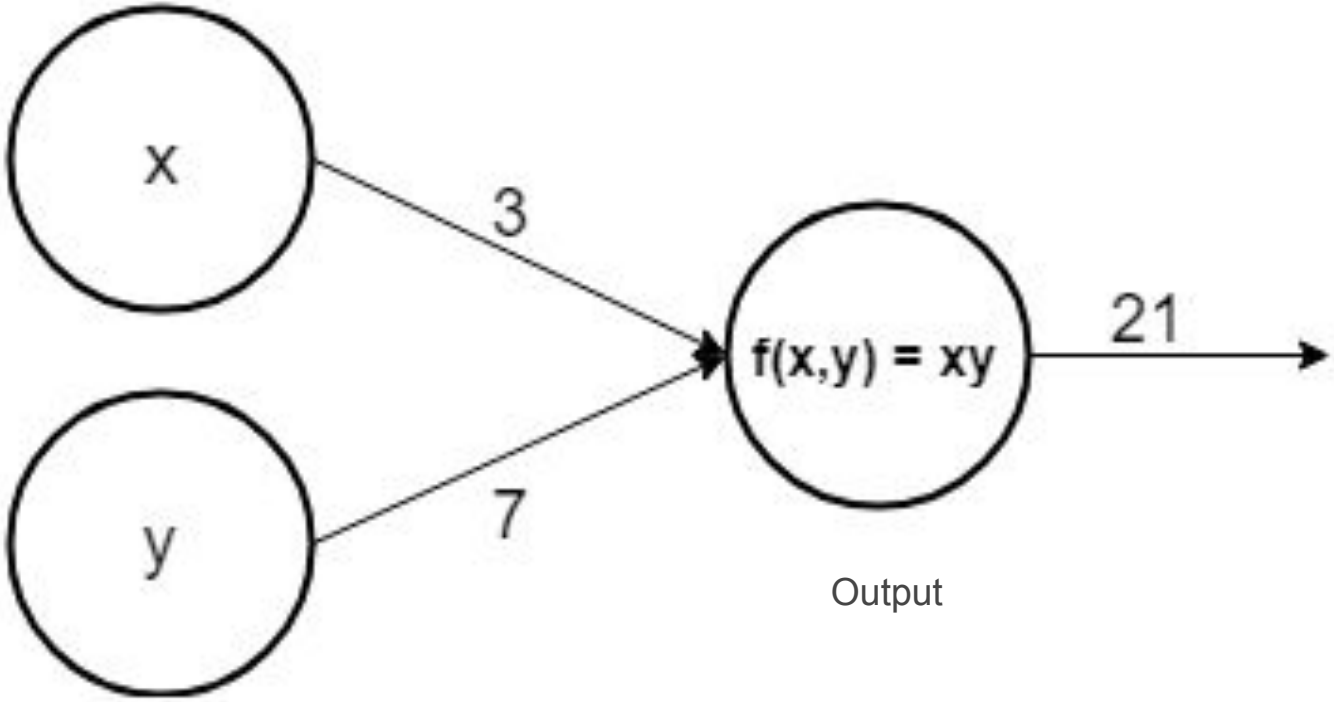
Unit: also called neurons, like the nerve cells in the biological brain.

Dense Layer: stack of layers are fully connected

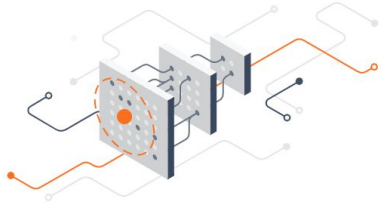
Weight & Biases: internal variables of model which are updated during training

Model: the representation of neural network

World's Simplest Neural Network

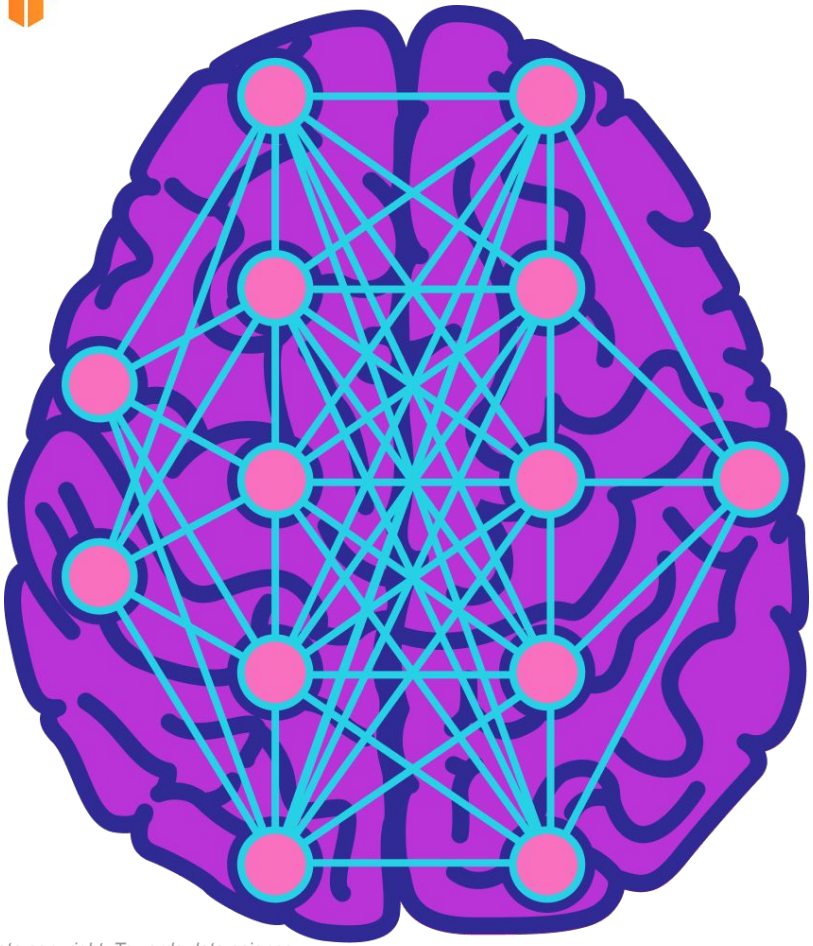


Weight is 1, Bias is 0, No hidden layer



World's Simplest Neural Network in Python

```
def neural_network (input, weight):
    return input * weight


inputs = [8.5, 9.5, 10, 9]
weight = 0.1
prediction = neural_network(inputs[0], weight)
print (prediction)     # 0.85
```



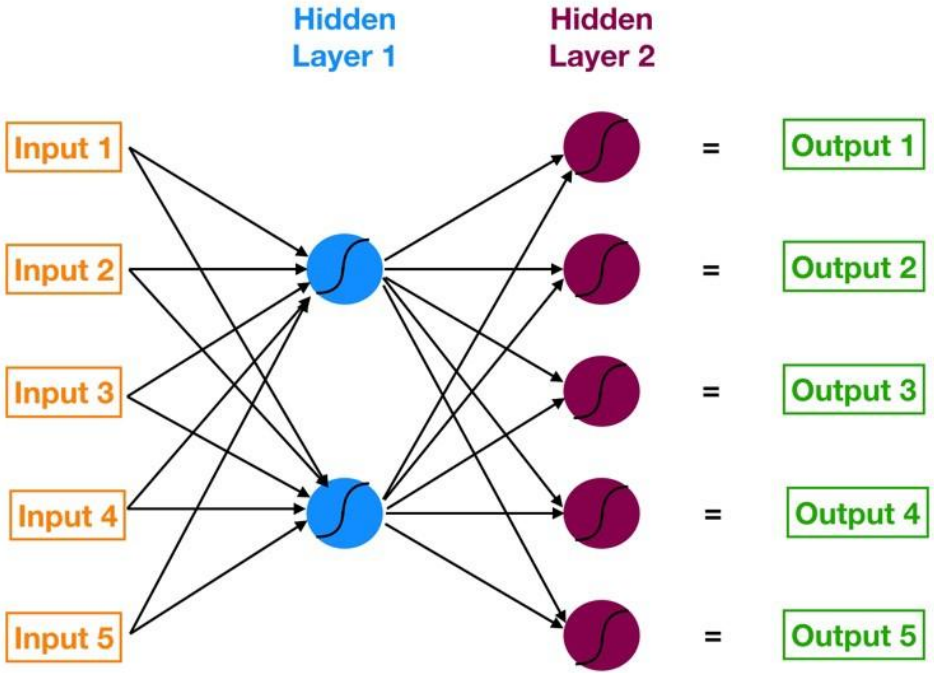
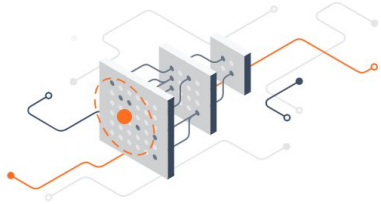
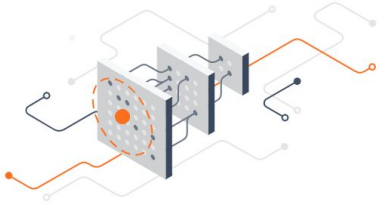


Photo copyright: Towards data science

1st Stage Separation!



Neural Network Deep Dive

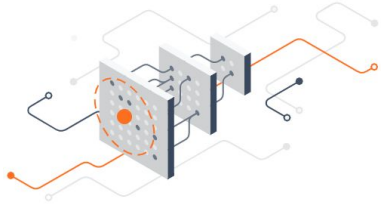


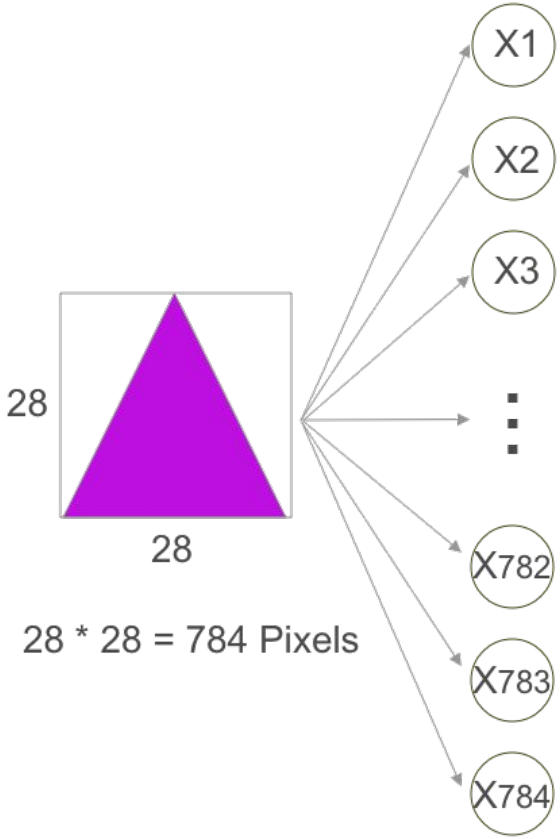

Input layer
Hidden layers
Output layer

Input layer

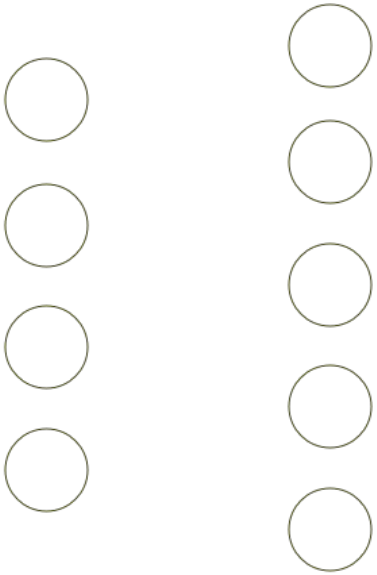
Hidden layers

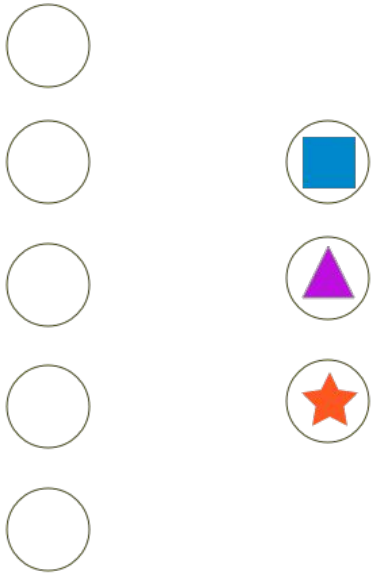
Output layer





28 * 28 = 784 Pixels

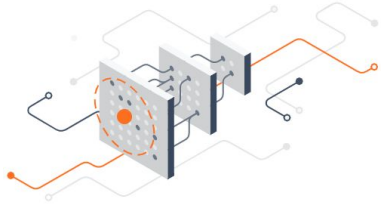


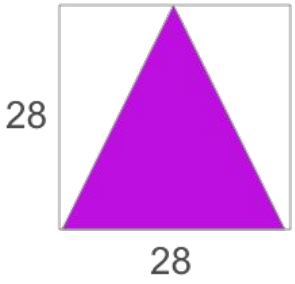


(X1 * 0.5 + X2 * 1.2 + X3 * -2.3 + … +
X782 * 3.8 + X783 * 0.5 + X784 * 4) + 9.8

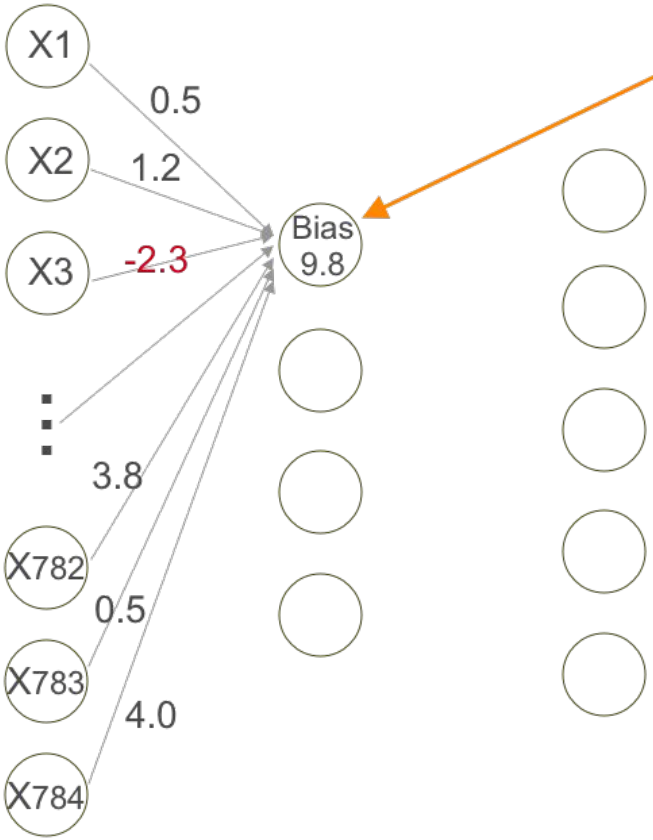
Activation Function
Sigmoid Fn [0, 1]

0.875



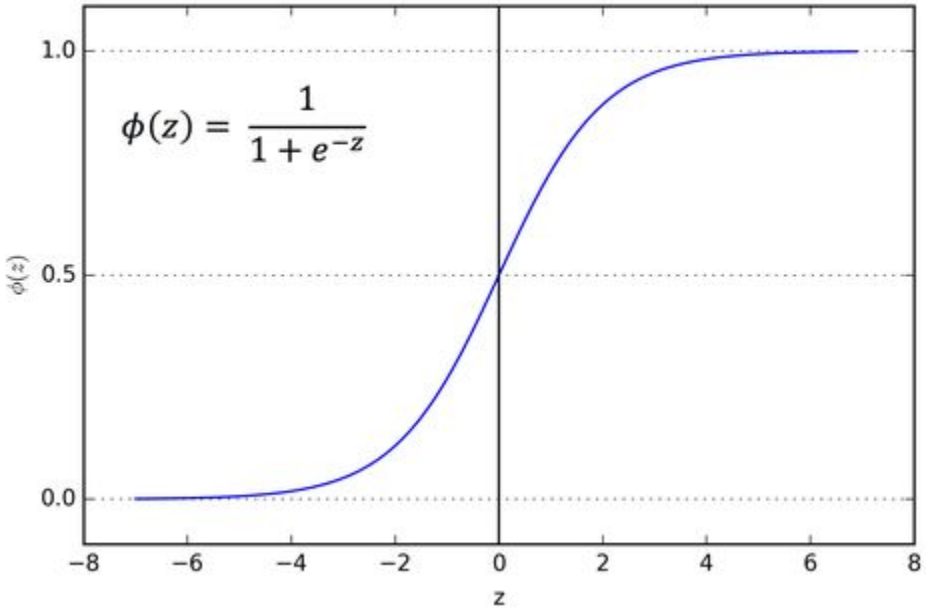


28 * 28 = 784 Pixels



$$y = wx + b$$

Sigmoid Function



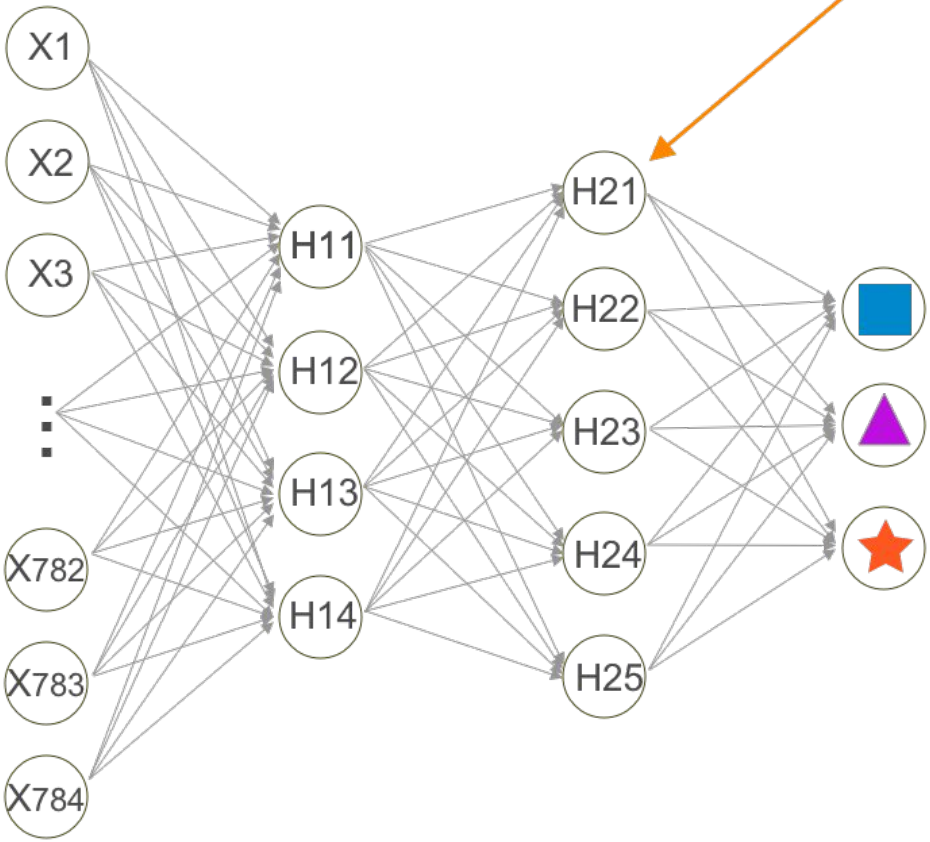
- Value between 0 - 1
- Useful for predicting probabilities as the output

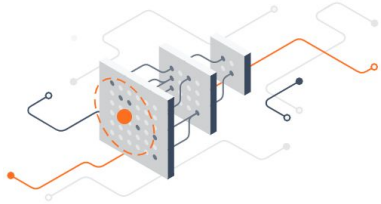
(H11 * 2.5 + H12 * 4.2 +
H13 * 1.3 + H14 * -9.2) +12.3

Activation Function
Sigmoid Fn [0, 1]

0.34

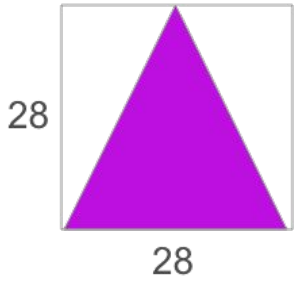
28 * 28 = 784 Pixels



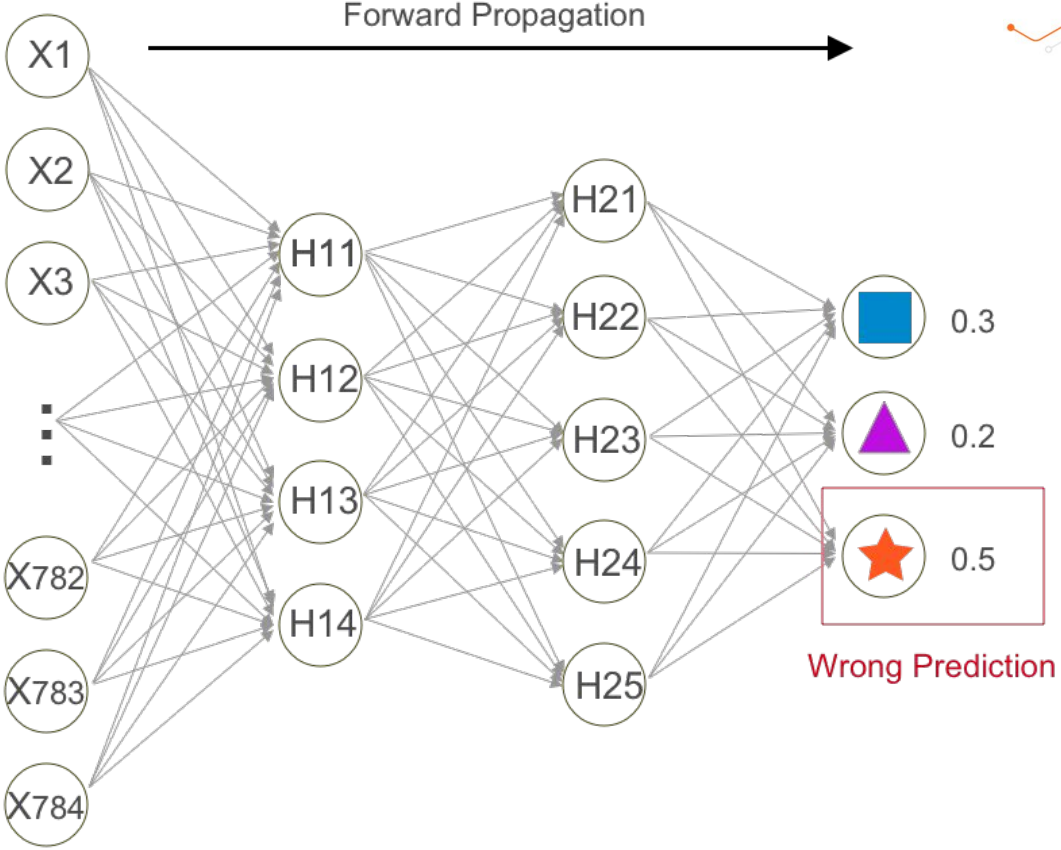


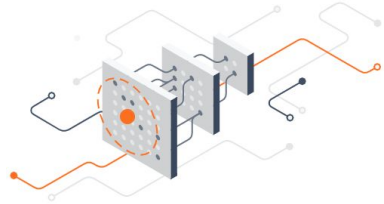
Input layer Hidden layers Output layer

Forward Propagation



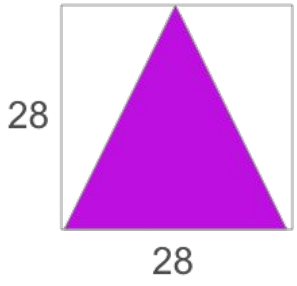
28 * 28 = 784 Pixels



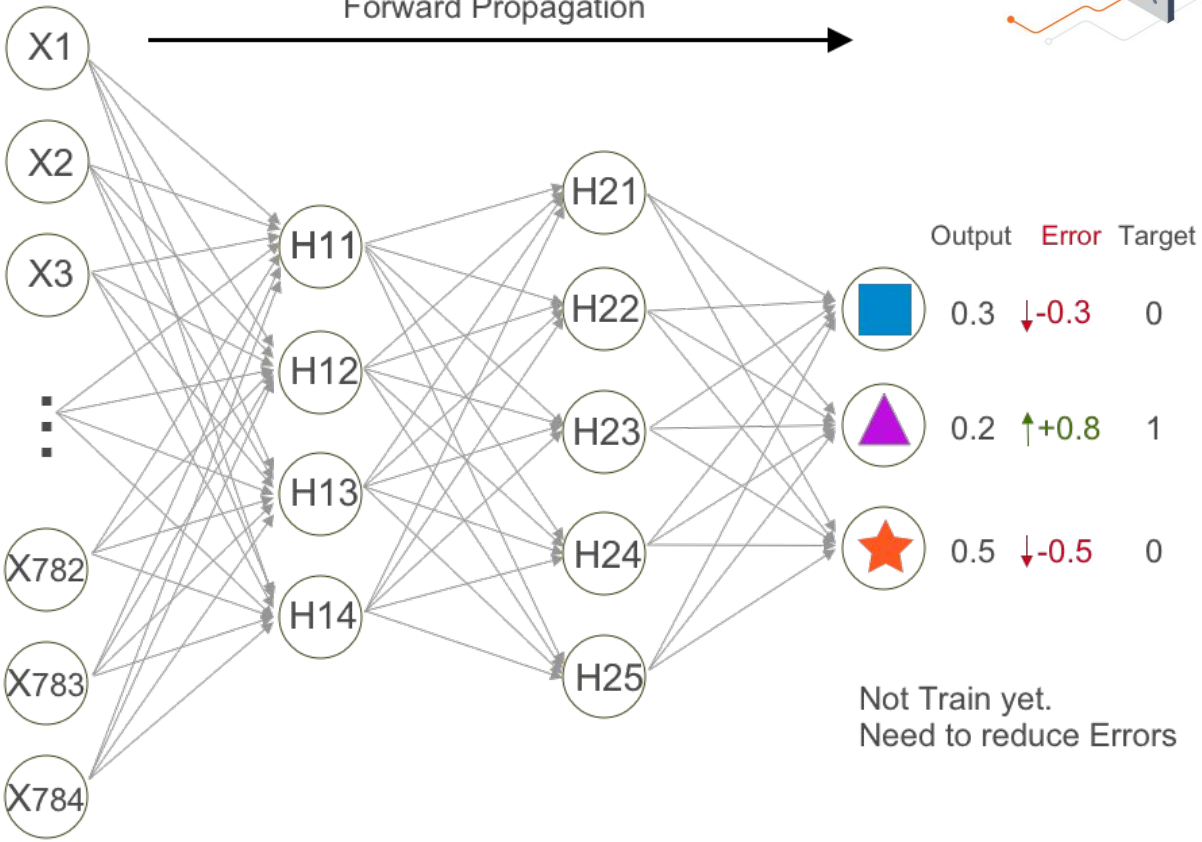


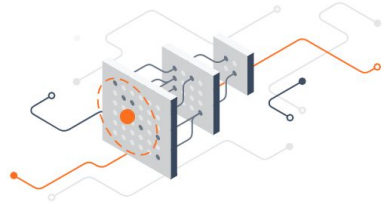
Input layer Hidden layers Output layer

Forward Propagation



28 * 28 = 784 Pixels





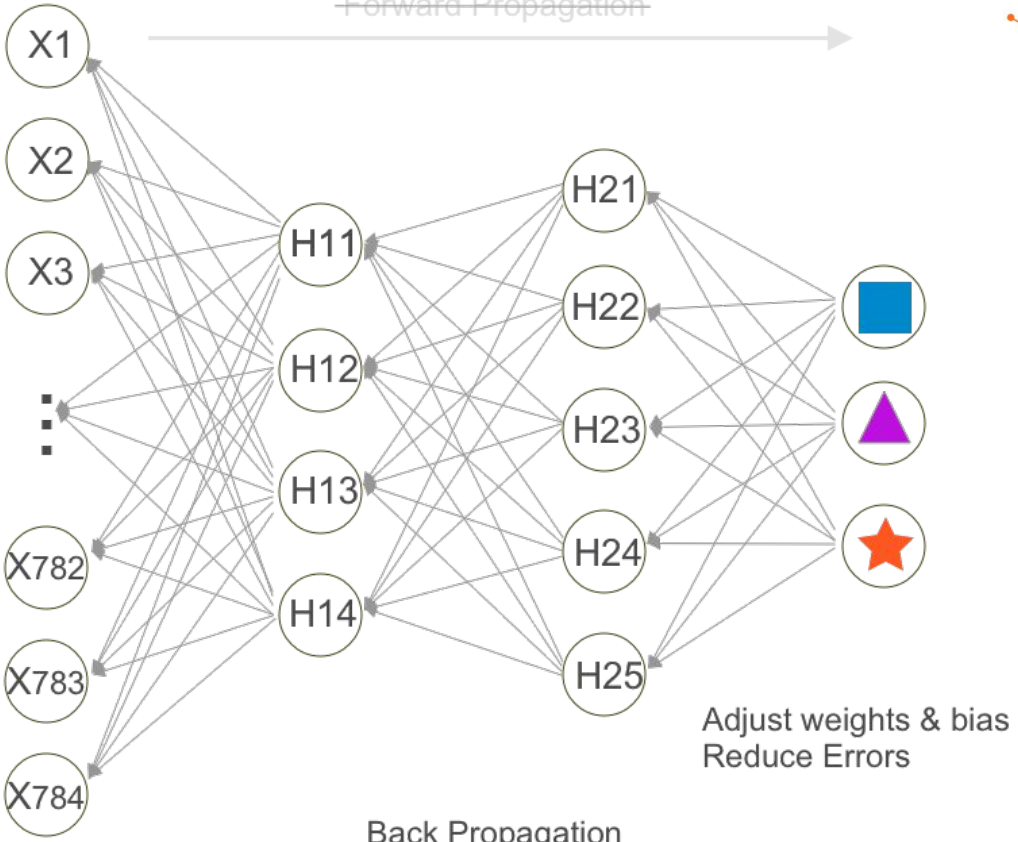
Input layer

Hidden layers

Output layer

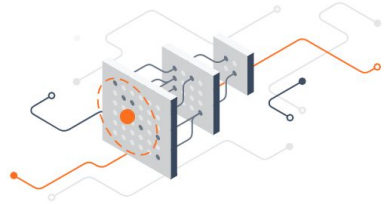
~~Forward Propagation~~

28 * 28 = 784 Pixels



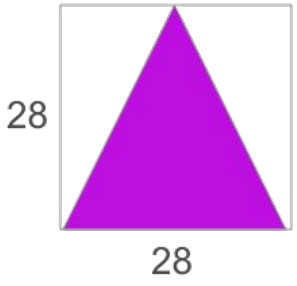
Adjust weights & bias
Reduce Errors

Back Propagation

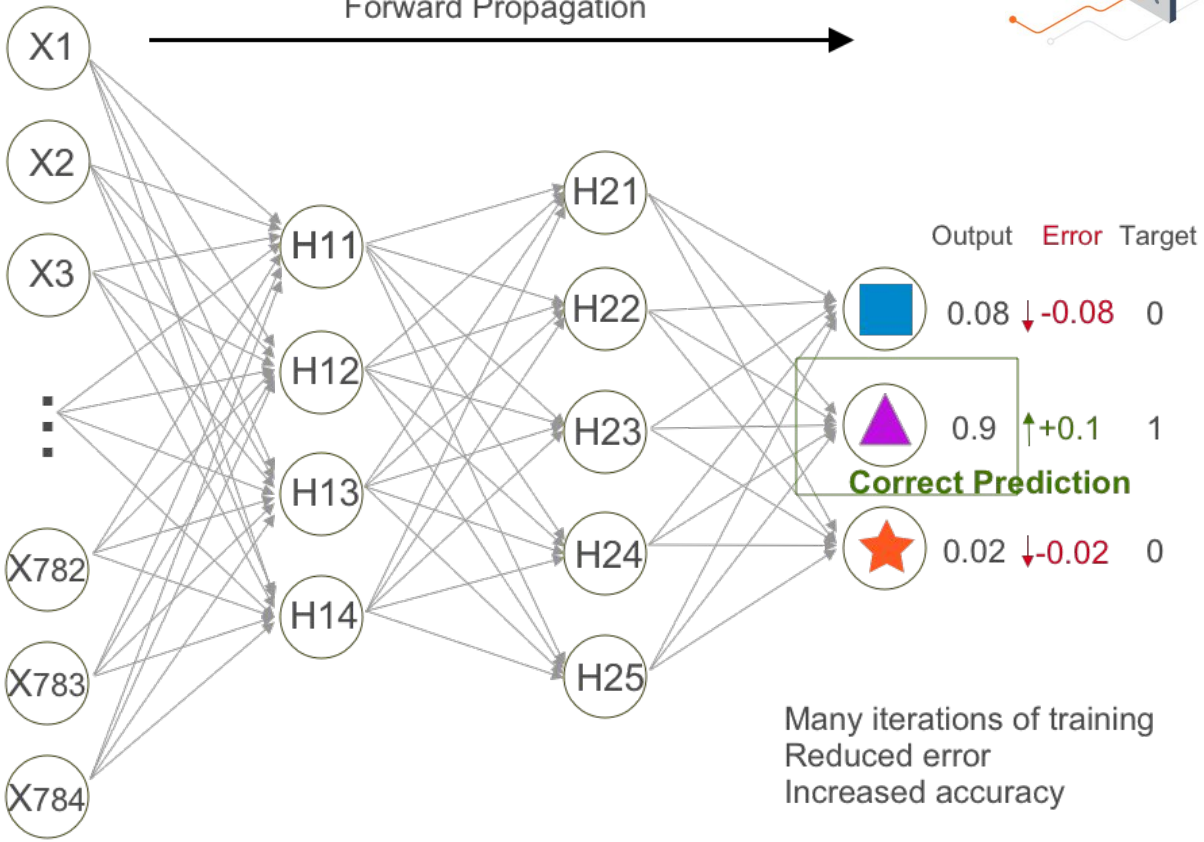


Input layer Hidden layers Output layer

Forward Propagation



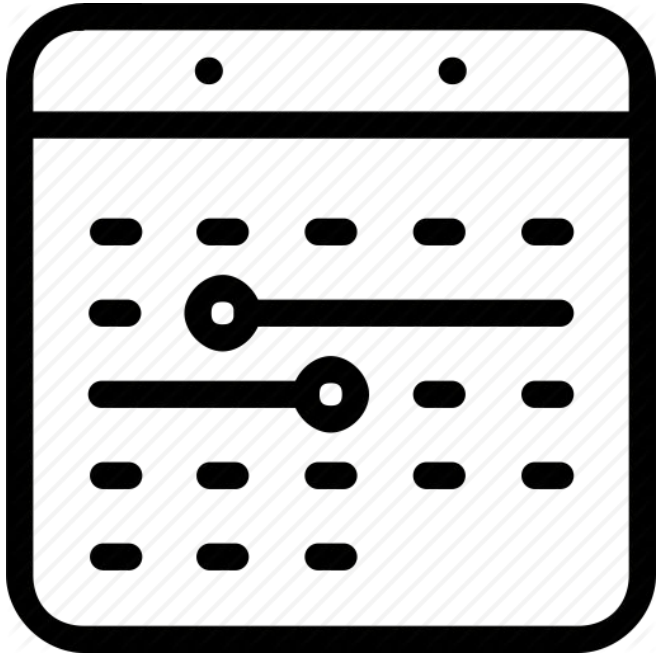
28 * 28 = 784 Pixels



Many iterations of training
Reduced error
Increased accuracy

How long does the training take?

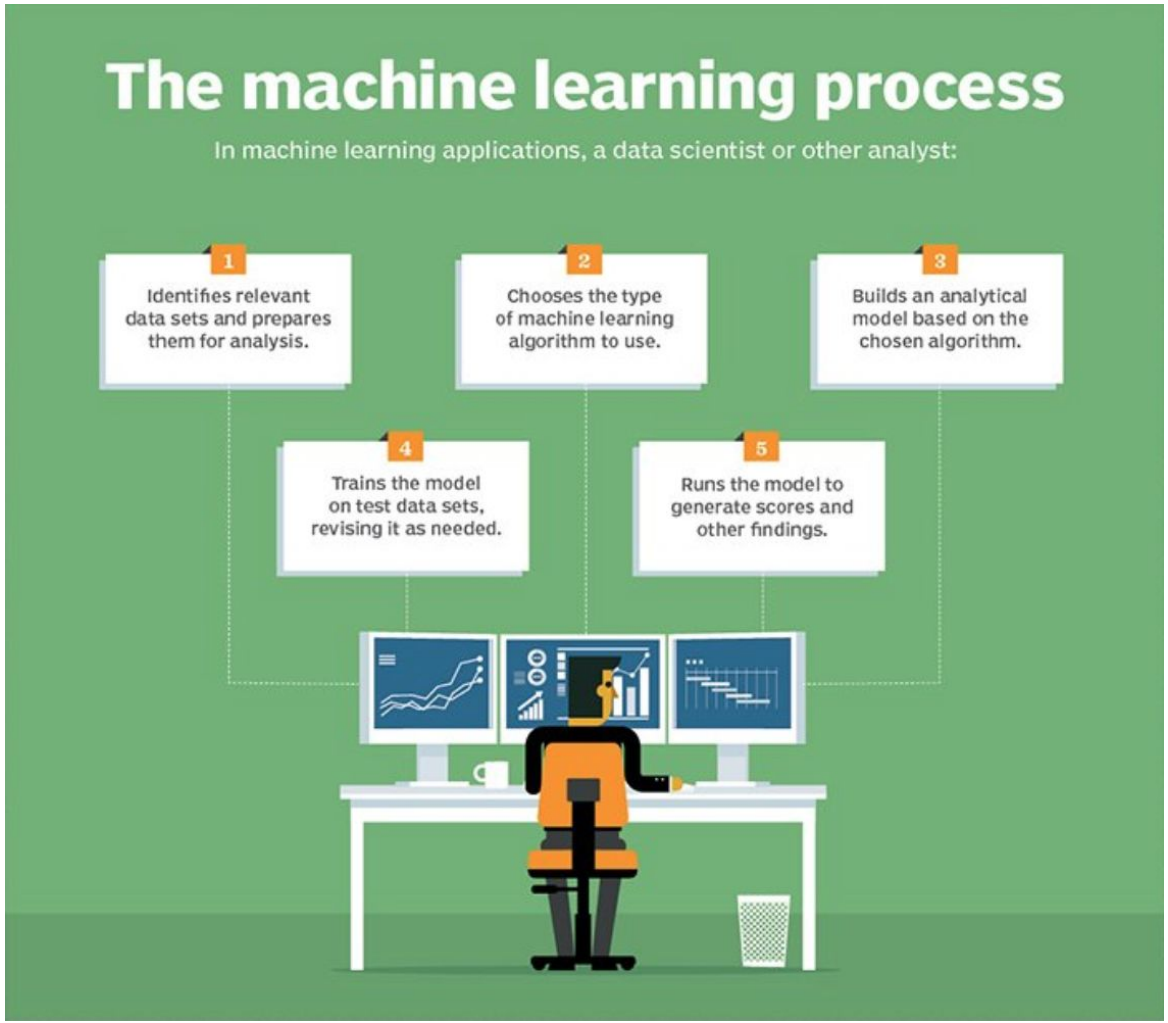
Hours

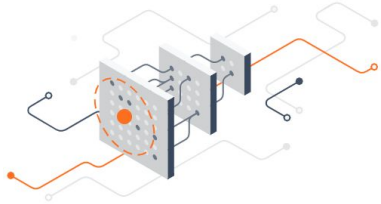


TensorFlow

ML Process

(5 simple steps)


The machine learning process
In machine learning applications, a data scientist or other analyst:
1
Identifies relevant data sets and prepares them for analysis.
2
Chooses the type of machine learning algorithm to use.
3
Builds an analytical model based on the chosen algorithm.
4
Trains the model on test data sets, revising it as needed.
5
Runs the model to generate scores and other findings.

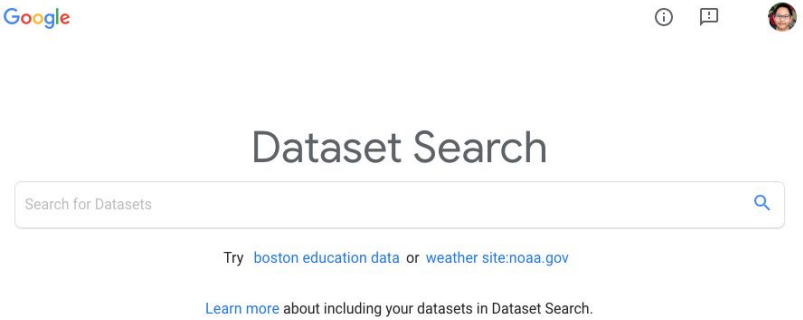


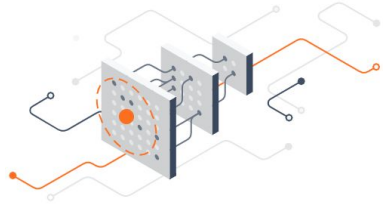
Step 1: Get Data

- Machine Learning depends largely on training data.
- Data preparation is a crucial step to make it suitable for ML.
- A large amount of data is generally required for the most common forms of ML.

Step 1: Data selection

- Google Dataset Search https://datasetsearch.research.google.com/
- Kaggle Datasets
- UCI ML Repository
- HackerEarth ML Datasets
- Amazon Datasets
- Microsoft Datasets
- Other sources (e.g. Government Datasets)



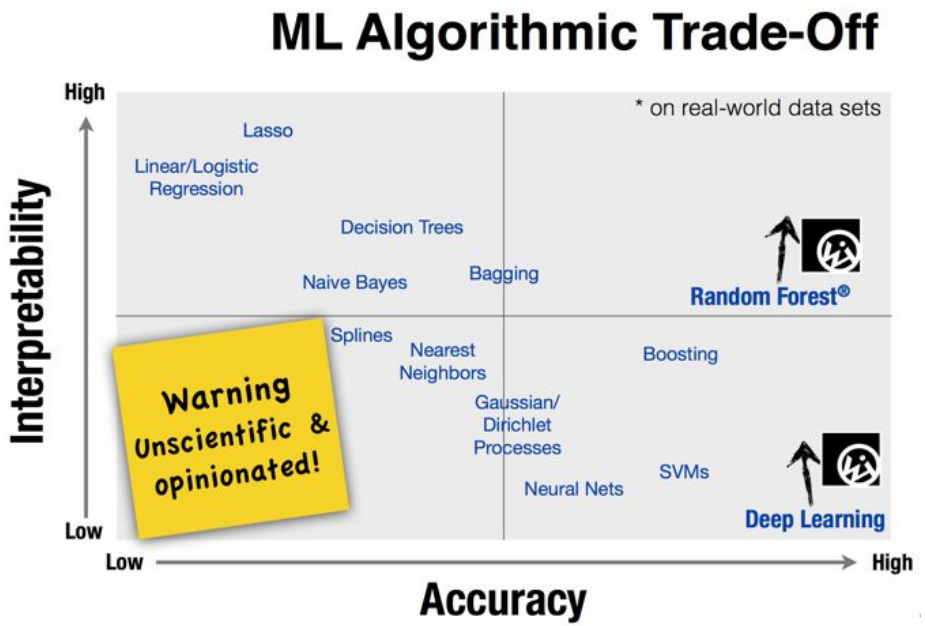


Step 2: Preprocess Data

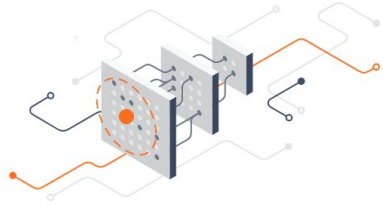
- Data preparation involves data selection, filtering, transformation, etc
- Formatting data to make it suitable for ML (structured format)
- Cleaning data to remove incomplete variables
- Sampling data further to reduce running times for algorithms and memory requirements.
- Separating data into training dataset, validation dataset and testing dataset
- Visualize Data using plots
- Data transformation methods: Scaling, Aggregation, etc

Step 3: Choose an Algorithm

- The algorithm used to train your model
- Choose **wisely**
- Popular libraries like Scikit-Learn or Keras
- Commonly used ML algorithms: Linear regression, Logistic regression, Decision trees, K-means, etc

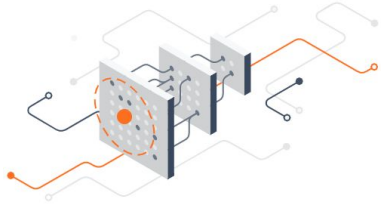


Step 4: Build & Compile Model



- Build a model layer by layer
- Each layer has weights that correspond to the layer the follows it
- Activation function for layer. It allows models to take into account nonlinear relationships
- Compiling the model takes parameters: optimizer, loss function, etc.

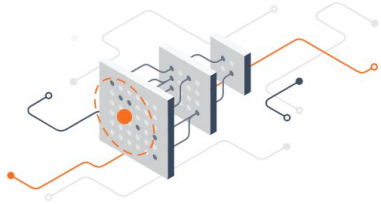
```
model.add (
  keras.layers.Dense(10, activation='relu')
  ...
)

model.compile (
  optimizer, loss, metrics, weights, ...
)
```



Step 5: Train the model

- Making a prediction based on the current state of the model
- Calculating how incorrect the prediction is
- Updating the weights or parameters of the network to minimize this error and make the model predict better
- Repeatedly iterating over the entire dataset for a given number of "epochs" until our model has converged(or define your own callback!)

```
model.fit (
  x, y, epochs=EPOCHS,verbose=VERBOSE,
  callback=CALLBACK
)
```

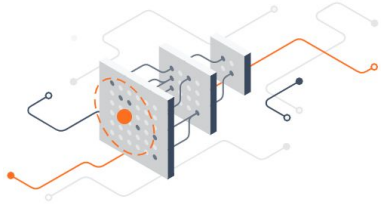


Step 6: Test the model

- See how well your model has learned the dataset during training
- Prediction result

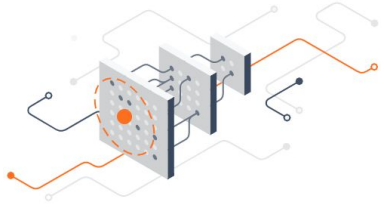
```
model.predict (
  predict values go here
)
```

REPEAT



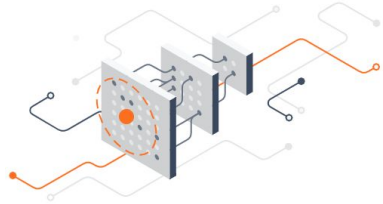
Human Friendly Way

Who Use ML?



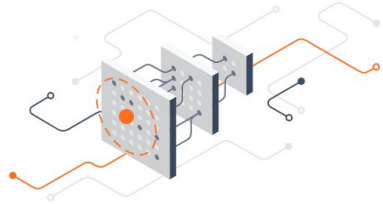
1. Researchers

Who Use ML?



1. Researchers
2. Data scientists

Who Use ML?

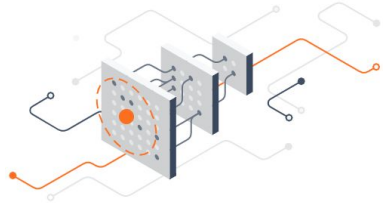


1. Researchers
2. Data scientists
3. ML engineers

Who Use ML?

1. Researchers
2. Data scientists
3. ML engineers
4. Developers

Who Use ML?



1. Researchers
2. Data scientists
3. ML engineers
4. Developers
5. Me

2nd Stage Separation!

TensorFlow

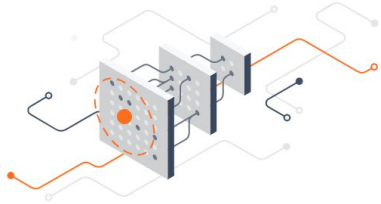
Traditional Learning

VS

Machine Learning


Fisher-Price
A
B
C
D
1
2
3
4


GOOD
JOB



Traditional Software Development VS ML

// Traditional Software Development

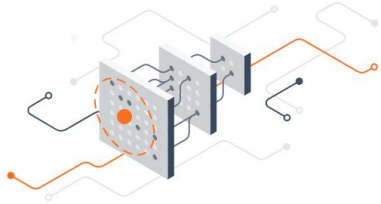
The input and the algorithm is known, and you write a function to produce an output

- Input data
- Apply logic to it
- Which produces a result
- Focused on code

// Machine Learning

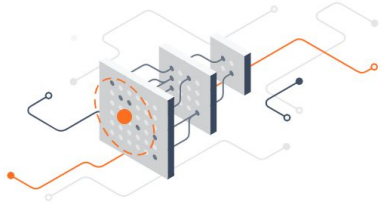
You know the input and the output, but you do not know the algorithm that creates the output given the input.

- Take pairs of input and output data
- Figure out the algorithm
- Focused on data



Traditional Software Development VS ML

```
// Traditional Software Development

def cToF(celcius):
  f = celsius * 1.8 + 32
  return f


cToF(0); // return fahrenheit value 32.
cToF(38);// return fahrenheit value 100.4.
```



Traditional Software Development VS ML

```
// Traditional Software Development

def cToF(celcius):
  f = celsius * 1.8 + 32
  return f
```

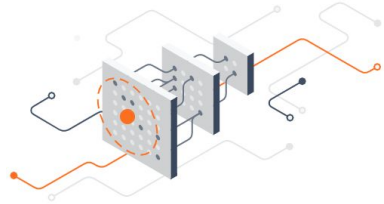
input

algorithm

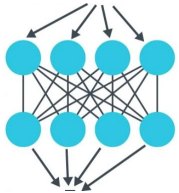
output

```
cToF(0); // return fahrenheit value 32.
cToF(38);// return fahrenheit value 100.4.
```

Traditional Software Development VS ML



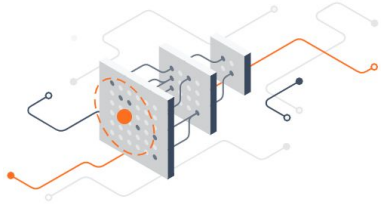
```
// Machine Learning

inputs = [0, 8, 15, 38]
outputs = [32, 46.4, 59, 100.4]

def cToF(inputs):
```



```
    return outputs
```

// A model. Algorithm: celsius * 1.8 + 32

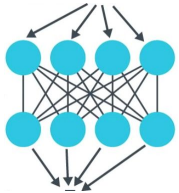
Traditional Software Development VS ML



```
// Machine Learning

inputs = [0, 8, 15, 38]
outputs = [32, 46.4, 59, 100.4]

def cToF(inputs):
```

input

output



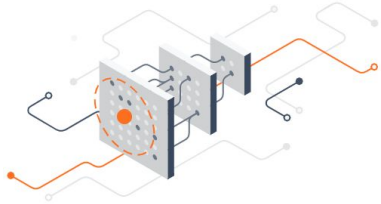
Iterations of training to find algorithm

```
    return outputs
```

// A model. Algorithm: celsius * 1.8 + 32

Simple Scenario

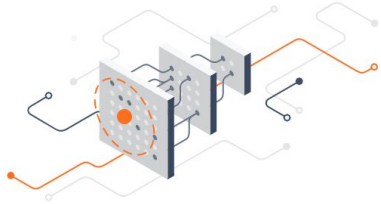
Scenario: Different language to say Hello?



Hello in English

Hello

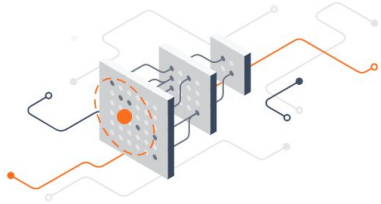
```
if (text == ‘hello’) {
  return language = ‘English’;
}
```



Hello in French

Bonjour

```
if (text == ‘hello’) {
  return language = ‘English’;
} else {
   return language = ‘French’;
}
```



Hello in Spanish

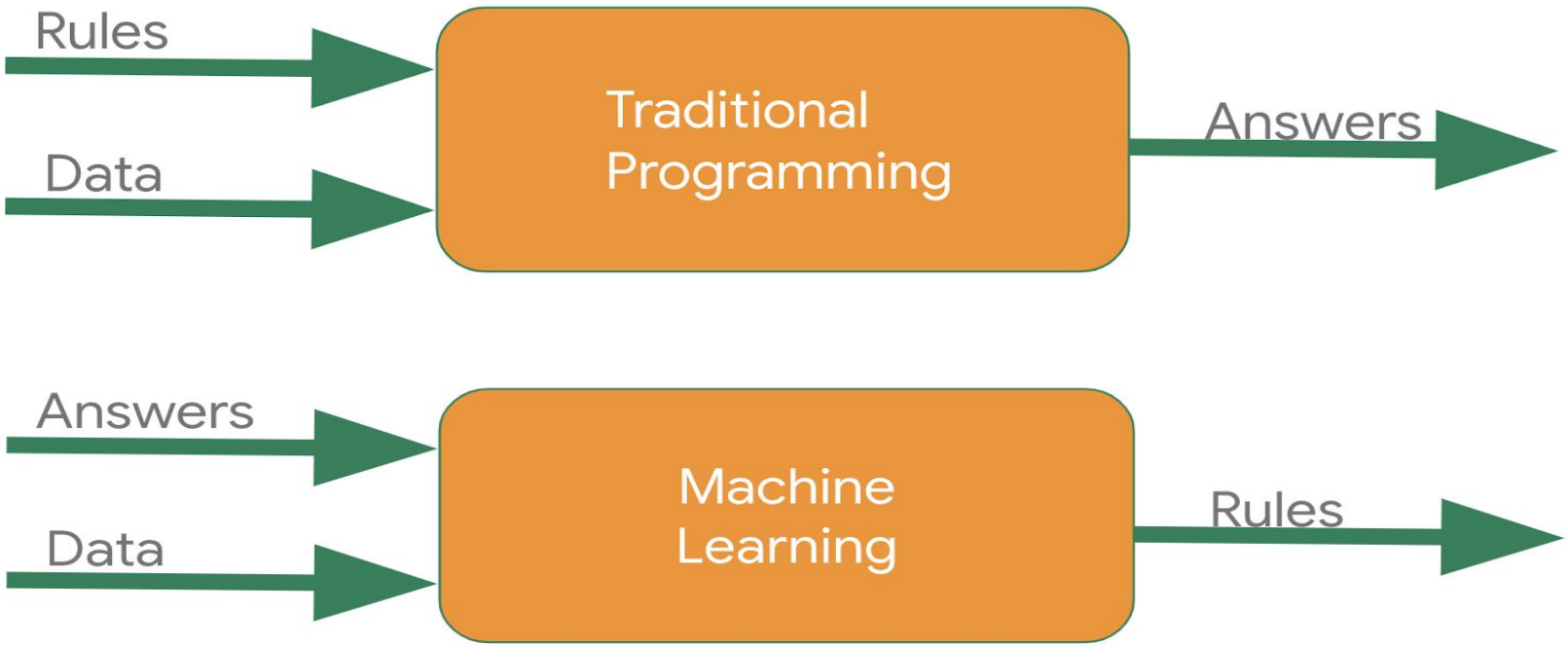
Hola

```
if (text == ‘hello’) {
  return language = ‘English’;
} else if (text == ‘bonjour’) {
  return language = ‘French’;
} else {
  return language = ‘Spanish’;
}
```

Which language is it? Hello in ?

?!!?

Traditional Programming VS ML



Now what?

Machine Learning

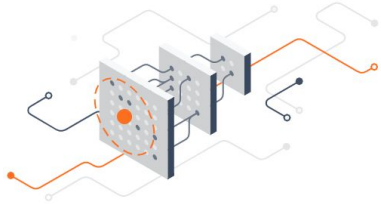

Data

```
counter = 1
start_time = time.time()

if self.load(self.checkpoint_dir):
    print(" [*] Load SUCCESS")
else:
    print(" [!] Load failed...")

for epoch in xrange(config.epoch):
    if config.dataset == 'mnist':
        batch_idxs = min(len(data_X), config.train_size) // config.batch_s
    else:
        data = glob(os.path.join("./data", config.dataset, "*.jpg"))
        batch_idxs = min(len(data), config.train_size) // config.batch_size

    for idx in xrange(0, batch_idxs):
        if config.dataset == 'mnist':
            batch_images = data_X[idx*config.batch_size:(idx+1)*config.batch_
            batch_labels = data_y[idx*config.batch_size:(idx+1)*config.batch_
        else:
            batch_files = data[idx*config.batch_size:(idx+1)*config.batch_
            batch = [get_image(batch_file, self.image_size, is_crop=self.is_
```

```
(training_data, training_label), (test_data, test_label) = source.load()

model = keras.models.Sequential([
  keras.layers.Dense(128, activation=nn.relu),
  ...
]);
model.compile(
  optimizer='adam',
  loss='sparse_categorical_crossentropy',
  metrics='accuracy'
)

model.fit( training_data, training_label, epochs = 30 )
model.evaluate( test_data, test_label )
```

Which language is it? Hello in ?

language = ‘Chinese’;



Much Better!

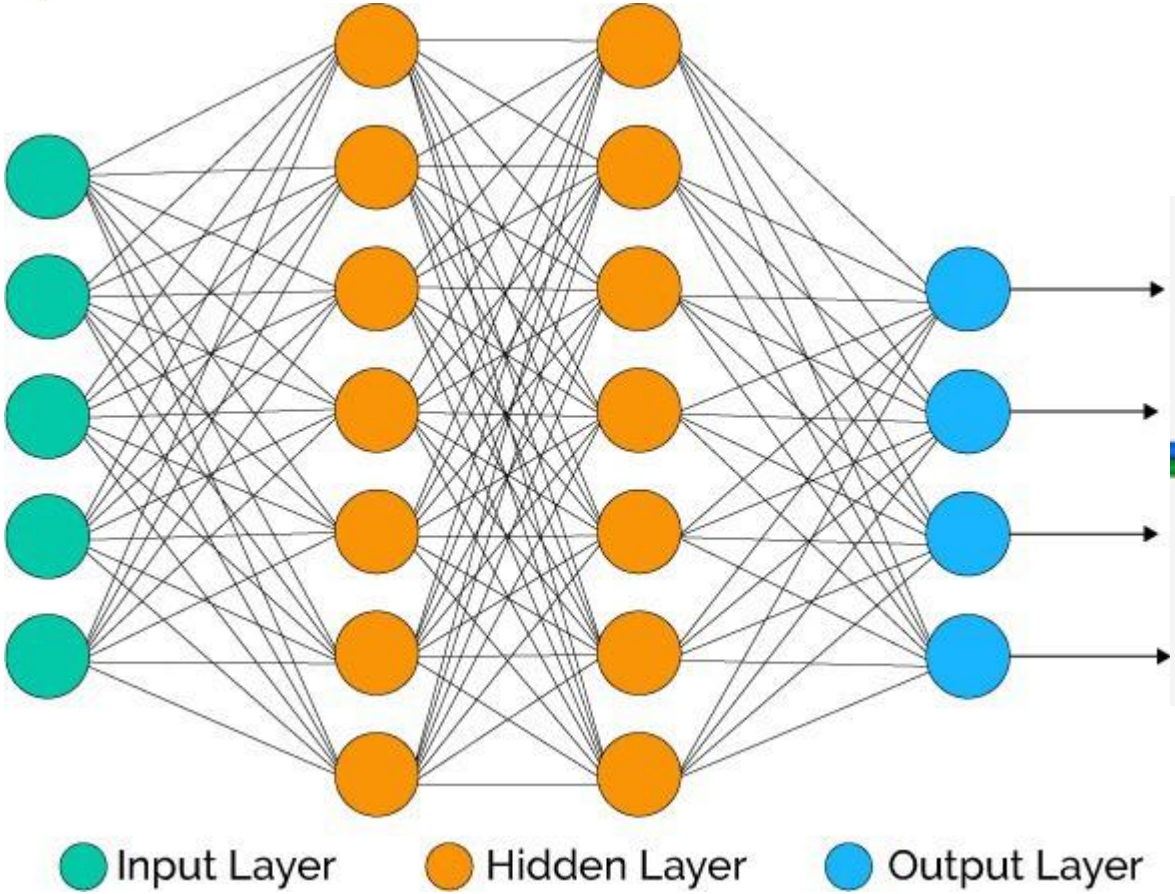
BERT (Bidirectional Encoder Representations from Transformers)

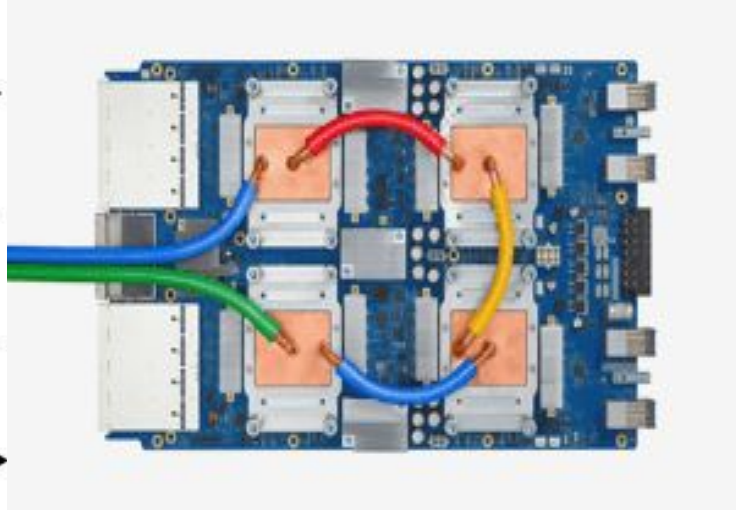
- State of the art language model for NLP
- Apply the bidirectional training of transformer, a popular attention model, to language modeling
- An attention mechanism that learns contextual relations between words/sub-words in a text
- A project by Google AI

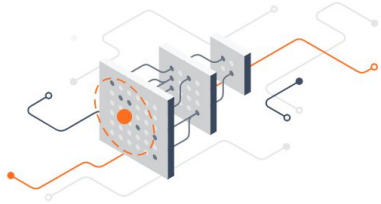
3rd stage ignition

TensorFlow

The History of ML


Input Layer
Hidden Layer
Output Layer





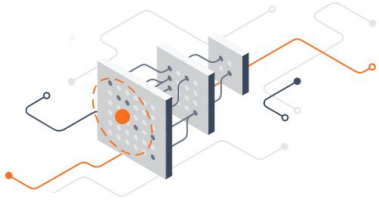
A Brief History of ML

- 1940 - The idea of ML was introduced
- 1951 - First neural network by Marvin Minsky from MIT
- 1952 - First RL Agent by Arthur Samuel
- 1974 - First ML ‘winter’
- 1996 - IBM’s Deep Blue defeats chess champion, Garry Kasparov
- 2000 - Backpropagation for NN is introduced
- 2014 - DeepMind is formed
- 2016 - DeepMind’s AlphaGo beats the best Go players
- 2030 - AI Robots taking over our jobs and daily tasks

Uh-oh.


JIM
JEFFERIES

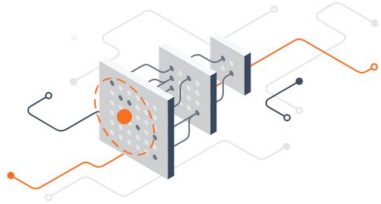

I can fix that.



TensorFlow

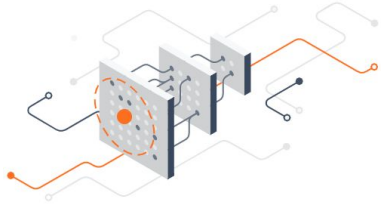
TensorFlow

- An open-source ML framework for everyone
- Built with C++
- 141K Github stars & >2400 contributors
- Powers nearly every single ML project @ Google
- No secret sauce
- Large framework
- Latest Release: 2.1 as of today



The History of TensorFlow

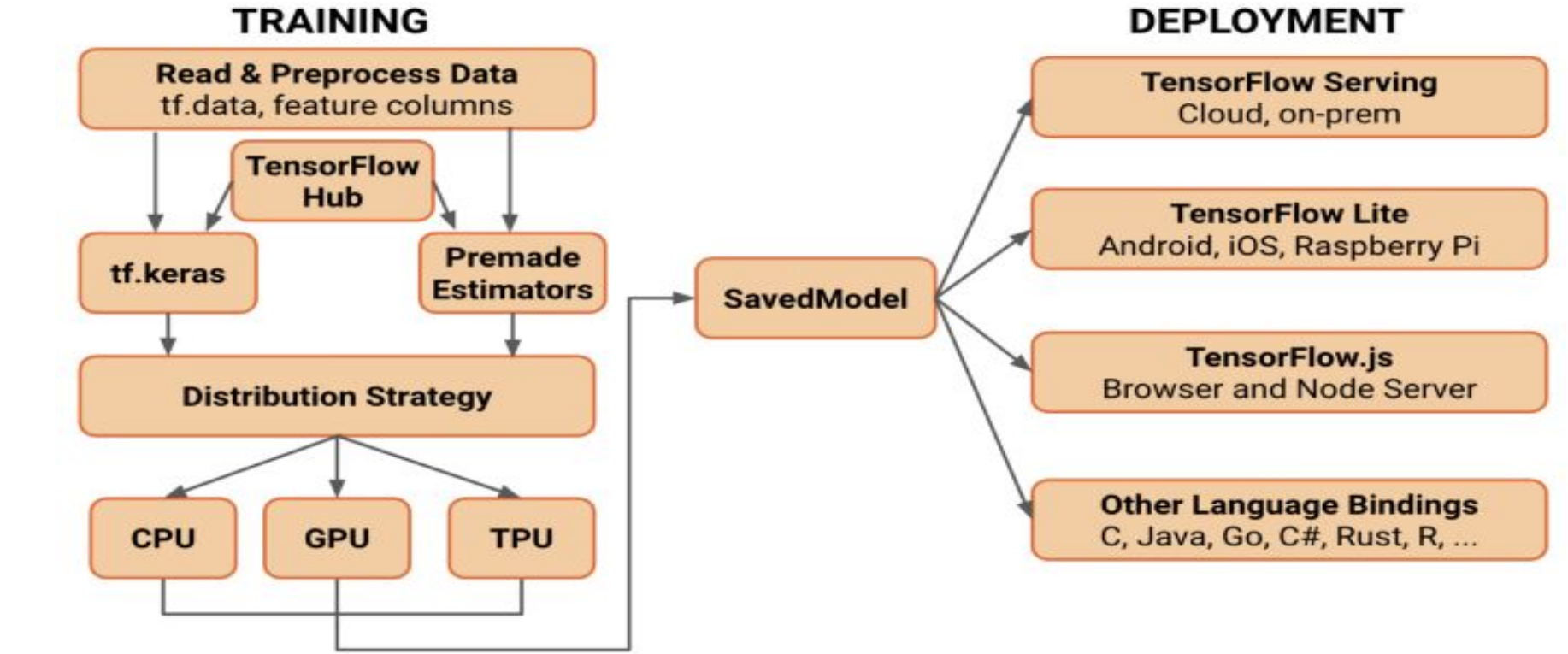
- Originally titled “DistBelief” and Google Brain team started developing it in 2011
- Open-sourced in 2015
- TensorFlow 1.0 announced in 2017
- The most hottest ML library in 2017
- TensorFlow 1.3 announced in 2017 which included Keras!
- TensorFlow 2.0 announced in September of 2019
- TensorFlow 2.1 announced in January of 2020



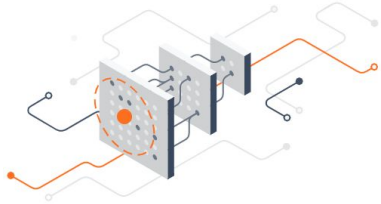
What is new in TensorFlow 2.x?

- Eager execution by default (Pythonic way)
- Keras as the high level API (tf.keras)
- tf.data simplified API to read training data using input pipelines
- tf.function decorator translates Python programs into TensorFlow graphs. (No more tf.Session())
- Still run TensorFlow 1.x code with 2.0 release (Pro tips: Do not do it!)

TensorFlow 2.x Architecture



```
[In]: import tensorflow as tf

[In]: tf.__version__

[Out]: 2.1.0
```

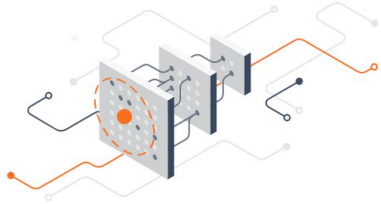


Everyone ❤ TensorFlow!

3rd stage Burnout

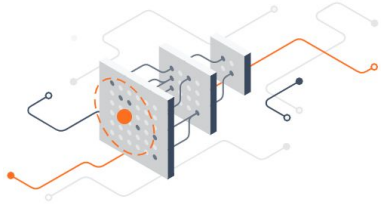
TensorFlow

Other tools for ML



Jupyter Notebook


Jupyter DataFrames
File Edit View Insert Cell Kernel Widgets Help
#DataFrames in Python
In [ ]: This is pandas DataFrame
In [6]: import pandas as pd
import numpy as np
In [7]: df = pd.DataFrame(data=np.array([[1,2,3],[4,5,6]], dtype=int), columns=['A','B','C'])
df
Jupyter
DataCamp



Google Colab

tutorial.ipynb

File Edit View Insert Runtime Tools Help

COMMENT SHARE

CODE TEXT CELL CELL

RAM Disk EDITING

Tutorial for Colaboratory

```
[1]  import matplotlib.pyplot as plt
     import pandas as pd
     pd.__version__
```

```
'0.22.0'
```

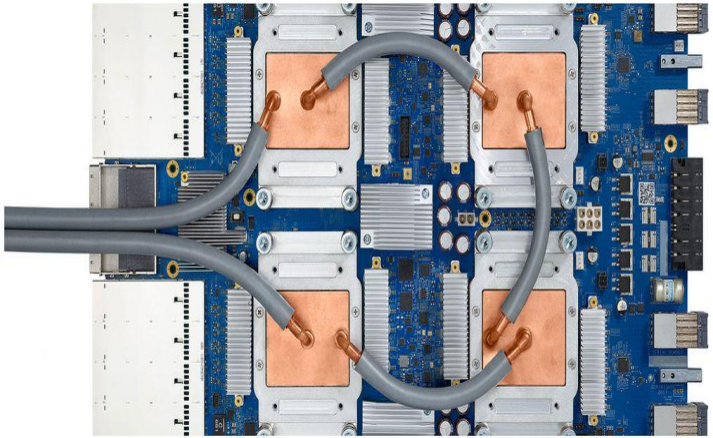
Ha, ha, ha! I've hidden the rest of the Colab!

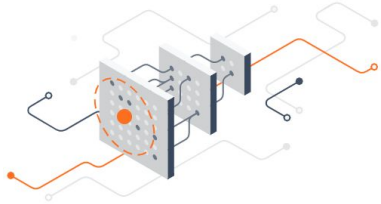

FREE
STUFF


Oh Yes!
IT'S
FREE

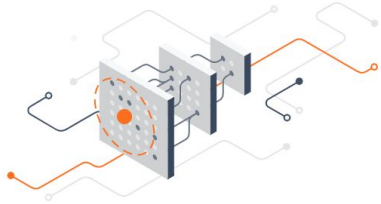

free
stuff


GTX980





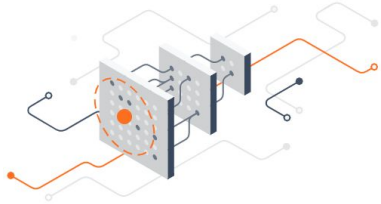
NumPy



What is Numpy?

- Numerical Python
- Open source library that can be used for scientific and numerical computing
- Combination of C and Python
- Efficient computation on arrays
- Can process a homogeneous multi-dimensional array

Pandas



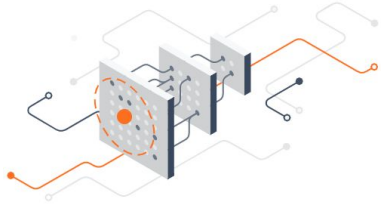
What is Pandas?

- High-performance data manipulation and analysis tool
- Load, prepare, manipulate, model, analyze, clean, transform (on Data)

In [20]:
```
url = 'http://46.101.230.157/dilan/pandas_tutorial_read.csv'
column_names = ['my_datetime', 'event', 'country', 'user_id', 'source', 'topic']
pd.read_csv(url, delimiter=';', names = column_names)
```

Out[20]:

	my_datetime	event	country	user_id	source	topic
0	2018-01-01 00:01:01	read	country_7	2458151261	SEO	North America
1	2018-01-01 00:03:20	read	country_7	2458151262	SEO	South America
2	2018-01-01 00:04:01	read	country_7	2458151263	AdWords	Africa
3	2018-01-01 00:04:02	read	country_7	2458151264	AdWords	Europe
4	2018-01-01 00:05:03	read	country_8	2458151265	Reddit	North America
5	2018-01-01 00:05:42	read	country_6	2458151266	Reddit	North America
6	2018-01-01 00:06:06	read	country_2	2458151267	Reddit	Europe
7	2018-01-01 00:06:15	read	country_6	2458151268	AdWords	Europe
8	2018-01-01 00:07:21	read	country_7	2458151269	AdWords	North America
9	2018-01-01 00:07:29	read	country_5	2458151270	Reddit	North America
10	2018-01-01 00:07:57	read	country_5	2458151271	AdWords	Asia



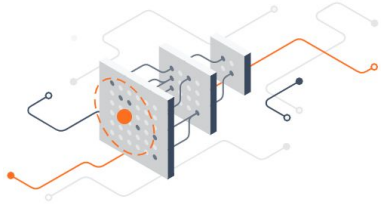
Human Friendly Way

```
[In]: import pandas as pd
```

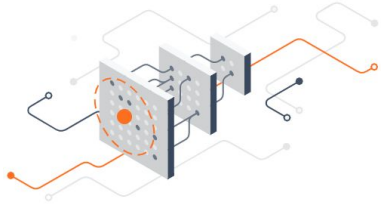
```
[In]: df = pd.read_csv('pandas.csv')

[In]: print(df)
```

```
[In]: df.drop(to_drop,inplace=True,axis=1)
```


Great meal!

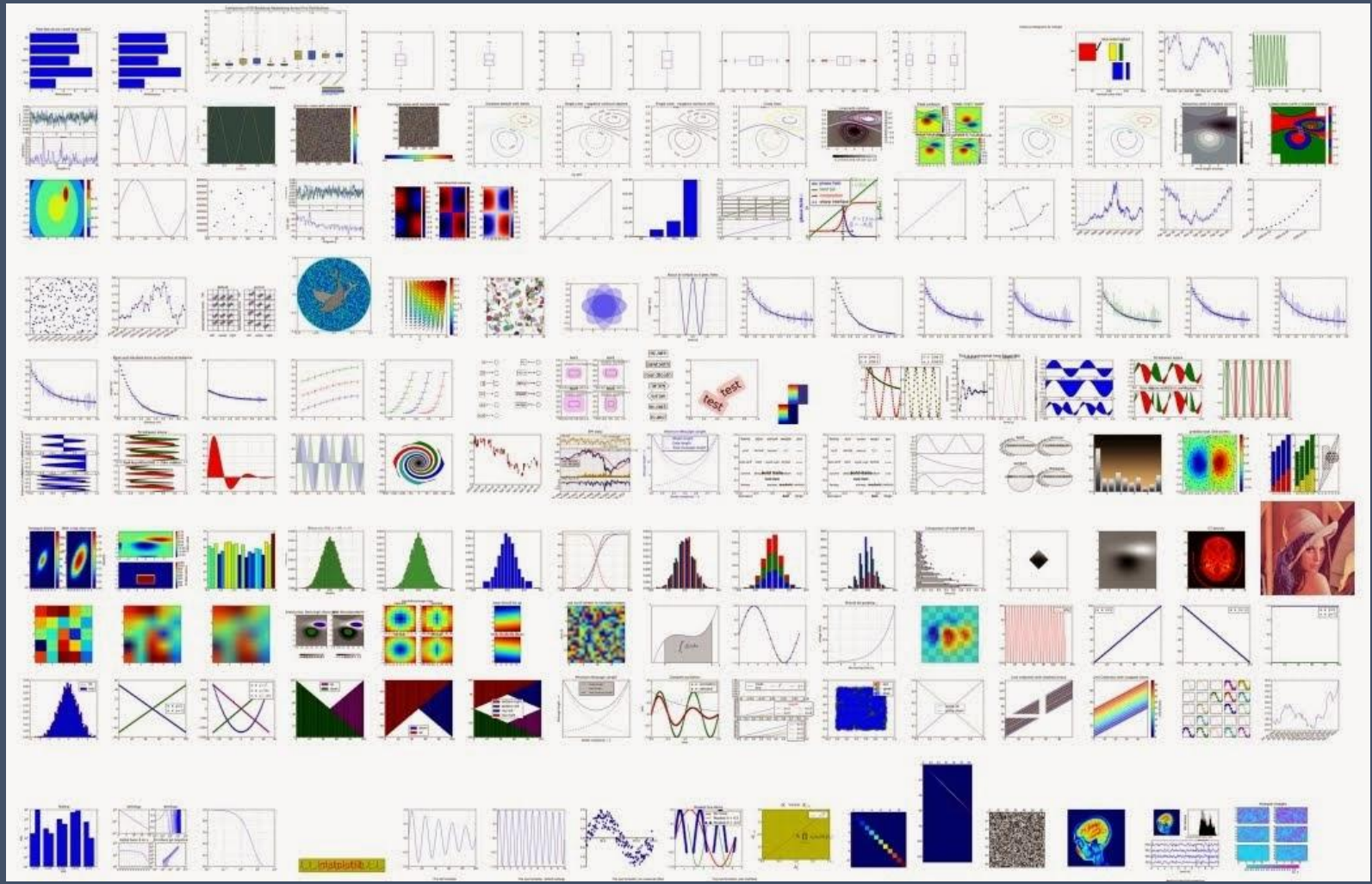


Matplotlib



What is Matplotlib?

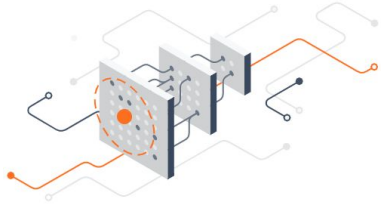
- Multi-platform data visualization library for 2D plots of arrays
- Create high-quality graphs, charts, and figures


test
test
matplotlib

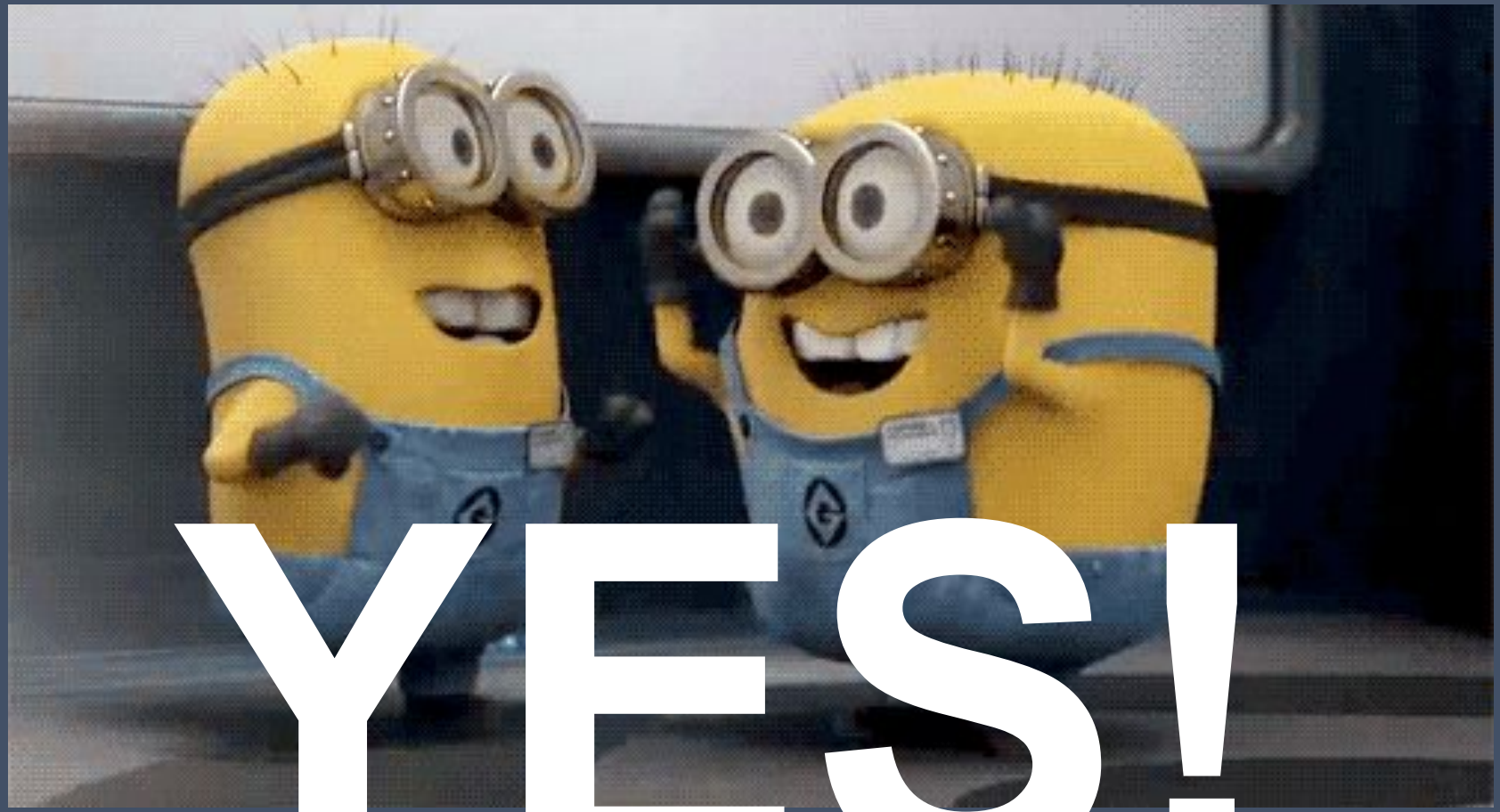
TensorFlow

2nd Edition
Updated for
TensorFlow 2

Project 1: Fashion MNIST



Coding?!?


YES!

Demo

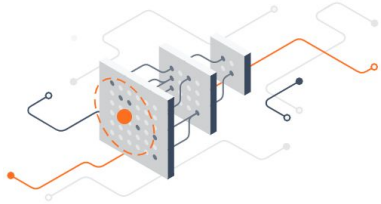
https://bit.ly/pycon20-fm

MOON LANDING!

TensorFlow

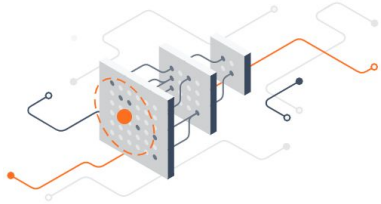
Grand Finale

What If



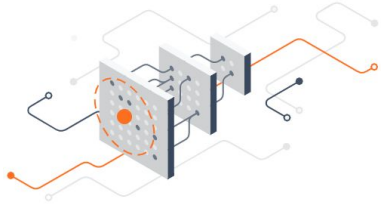
What If

WHAT????!!!!



What If

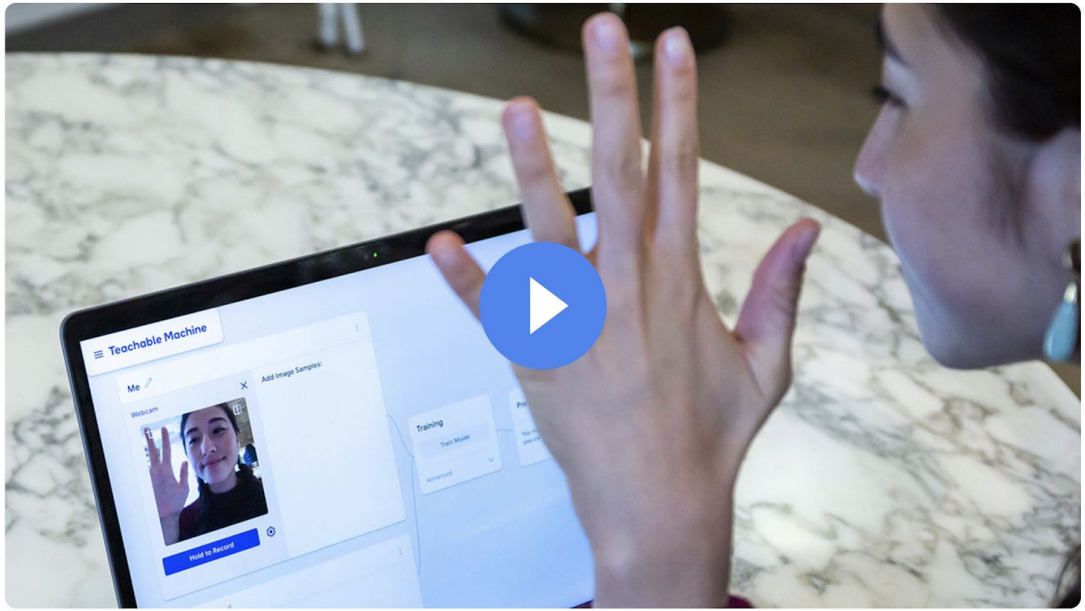
No Code required to create ML Model?



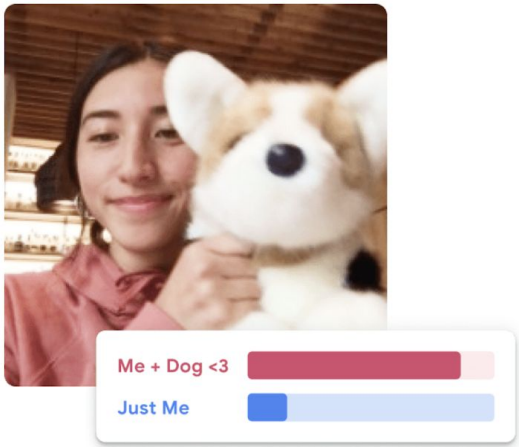
What If

No Code required to create ML Model?
No ML experience required?

Teachable Machine

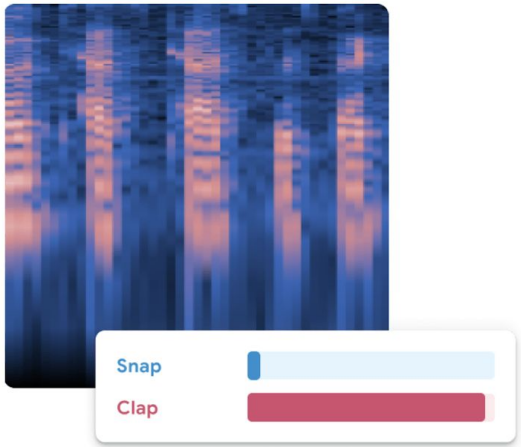


Teachable Machine is a web-based tool that makes creating machine learning models fast, easy, and accessible to everyone.



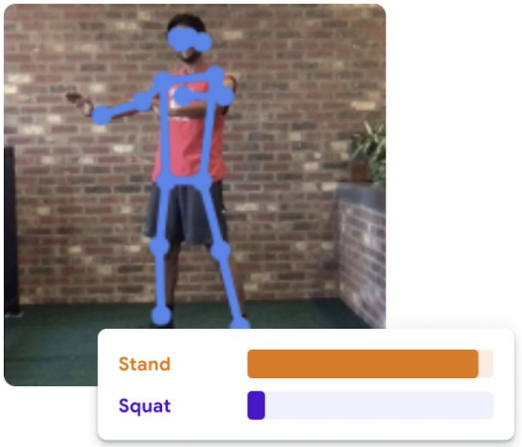
Images

Teach a model to classify images using files or your webcam.



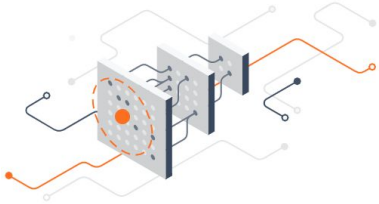
Sounds

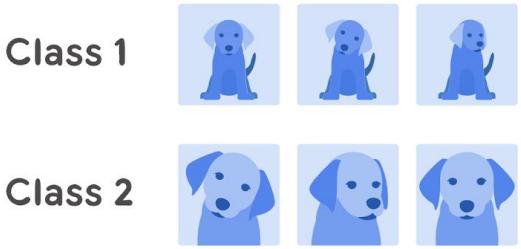
Teach a model to classify audio by recording short sound samples. (WAV/MP3/etc file support coming soon.)



Poses

Teach a model to classify body positions using files or striking poses in your webcam.





1 Gather

Gather and group your examples into classes, or categories, that you want the computer to learn.

2 Train

Train your model, then instantly test it out to see whether it can correctly classify new examples.

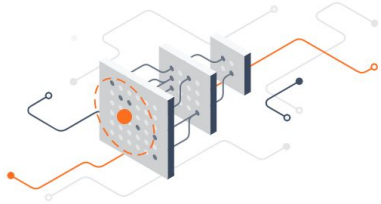
3 Export

Export your model for your projects: sites, apps, and more. You can download your model or host it online for free.

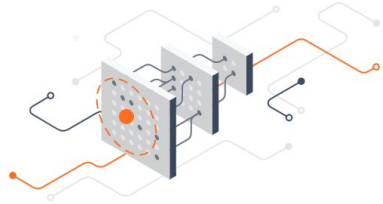
Demo

https://bit.ly/pycon20-3


TAKEAWAYS
TAKEAWAYS EVERYWHERE
quickmeme.com

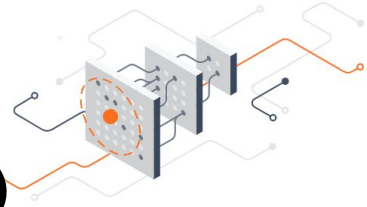


Machine Learning is **Everywhere**



Machine Learning is for **Everyone**


WE WANT YOU



With a little help from:

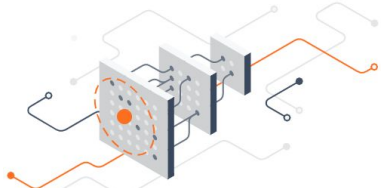
Google

With a little help from:

TensorFlow

We're Back On Earth! You survived!

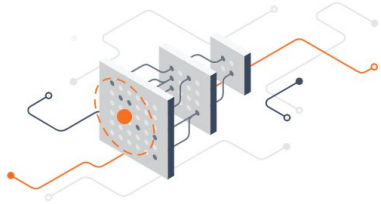
#PartyTime



AI is akin to building a rocket ship. You need a huge engine and a lot of fuel. The rocket engine is the learning algorithms but the fuel is the huge amounts of data we can feed to these algorithms.

Andrew Ng

The Inevitable: Understanding the 12 Technological Forces That Will Shape Our Future



Next steps

- Official Tensorflow site: https://www.tensorflow.org
- Machine Learning Crash Course: https://developers.google.com/machine-learning/crash-course/
- Awesome Tensorflow: https://github.com/jtoy/awesome-tensorflow
- TensorFlow Playground: https://playground.tensorflow.org
- Teachable Machine: https://teachablemachine.withgoogle.com/
- ML Zero to Hero (YouTube): https://www.youtube.com/playlist?list=PLQY2H8rRoyvwLbzbnKJ59NkZvQAW9wLbx
- Towards Data Science (blog, commonly referred to as TDS): https://towardsdatascience.com/machine-learning/home

NOW! Give Yourself a round of applause!

Slide: https://bit.ly/pycon20

Aaron Ma

aaronhma.com

hi@aaronhma.com

Available for 14/7

@aaronhma