

Introduction to Deep Learning and IBM Model Asset Exchange

IBM Developer

Saishruthi Swaminathan | Developer Advocate | IBM San Francisco
Brendan Dwyer | IBM Developer | IBM San Francisco

Agenda

- 1 • What is Data Science? Why is it important?
- 2 • The Data Science Process | Pipeline
- 3 • Decoding Deep Learning
- 4 • Introducing IBM Model Asset Exchange
- 5 • Demo on IBM Model Asset Exchange
- 6 • Summary



What is Data Science?



Breaking down data science definition

Goal

- Find solution to the business problem

How?

- Transforming problems to well-posed questions

Using?

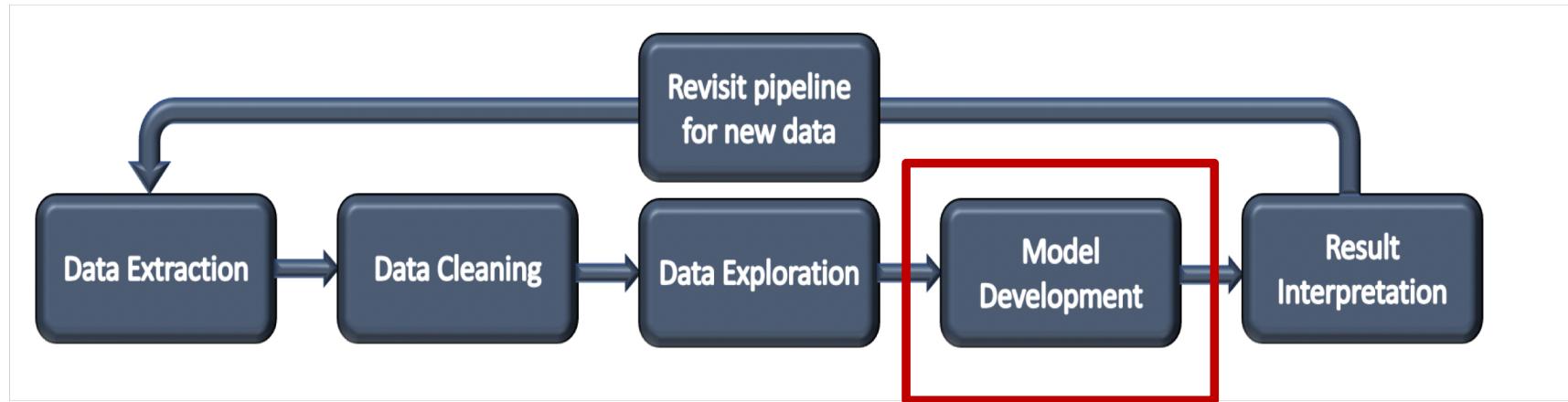
- Mathematics, programming and scientific method

Finally?

- Communicate results and its business impact

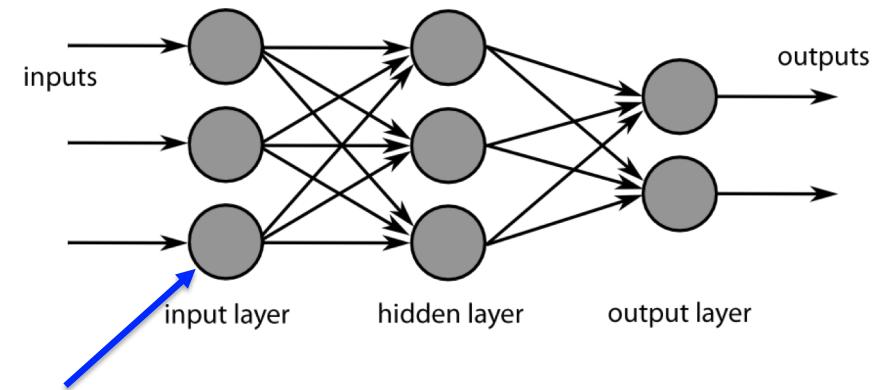


Data Science Pipeline



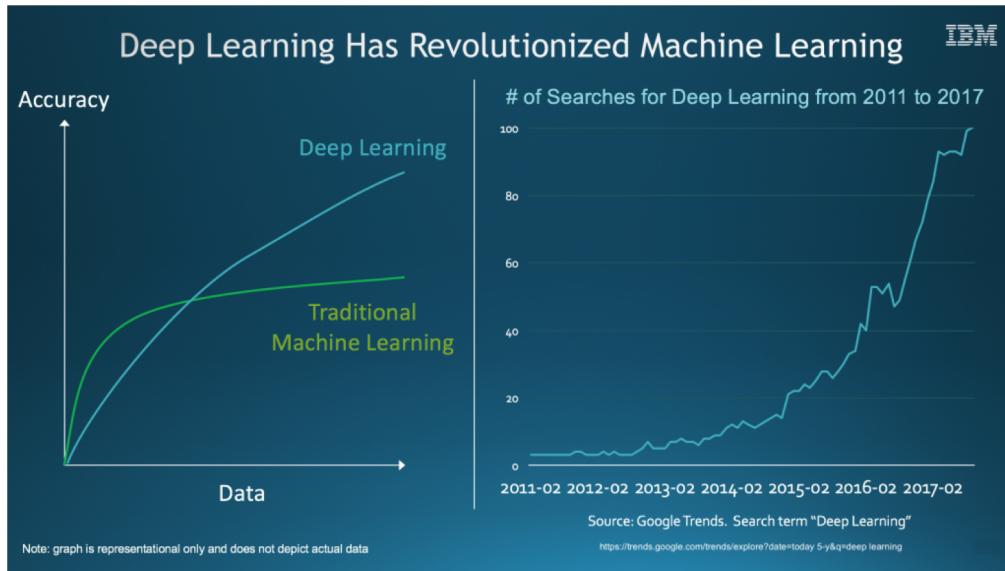
What is deep learning????

Machine learning with **large deep neural network**



Neuron

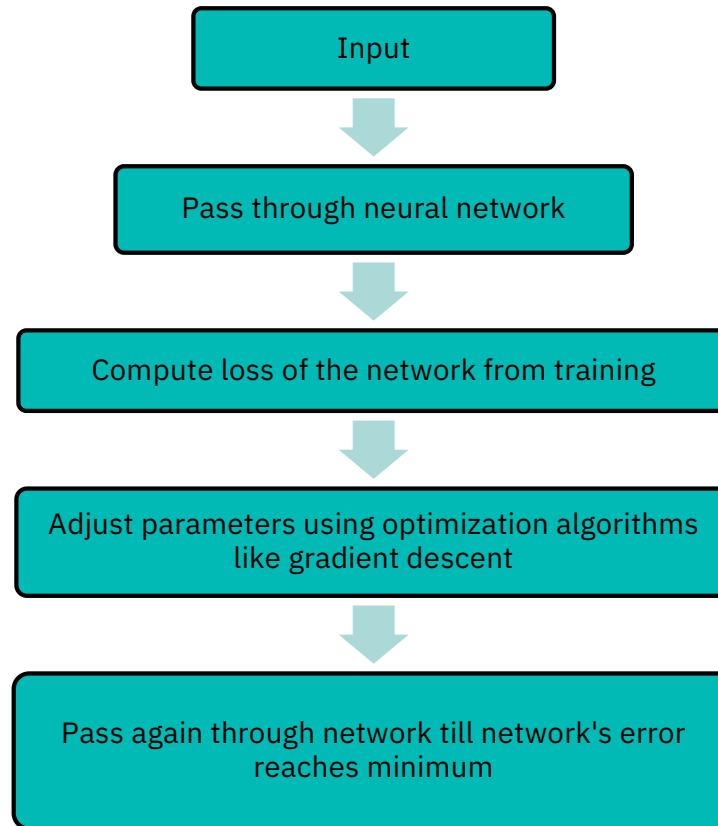
Why deep learning is becoming popular?



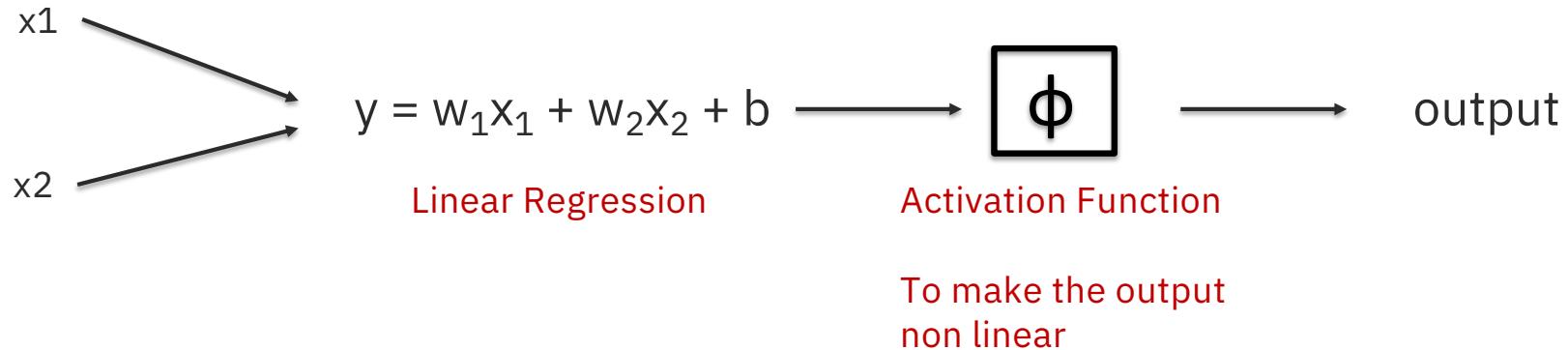
- Access to huge amount of data.
- Powerful computers increases computational efficiency.
- It is fantastic at supervised learning.
- Automatically learning important features.



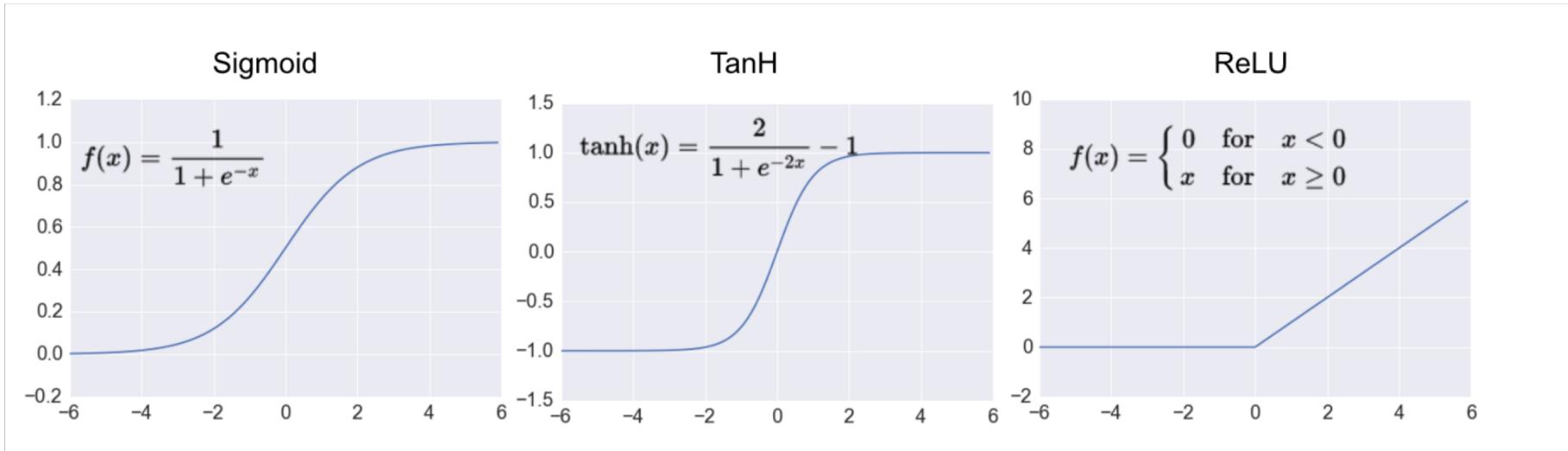
Flow



What happens inside a single neuron?



Activation Function



- Preferred over Sigmoid
- Output values have mean approximately zero.
- Slow optimization

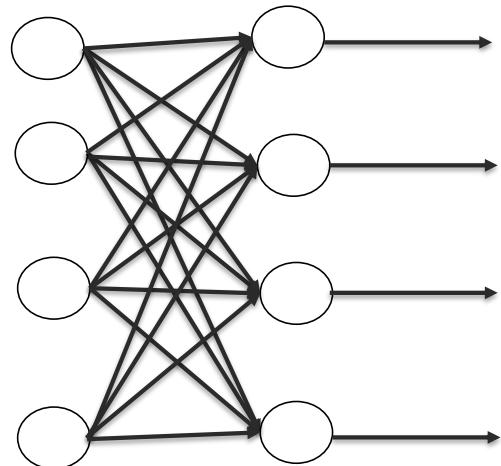
- Preferred for hidden layers
- Fast optimization

Source: <http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/>



Scale March 7, 2019 / © 2019 IBM Corporation

Softmax activation



$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

$$\begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix} \Rightarrow \begin{bmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{bmatrix} = \begin{bmatrix} 148.4 \\ 7.4 \\ 0.4 \\ 20.1 \end{bmatrix}$$

$$\sum_j e^{y_j} = 148.4 + 7.4 + 0.4 + 20.1 \\ = 176.3$$

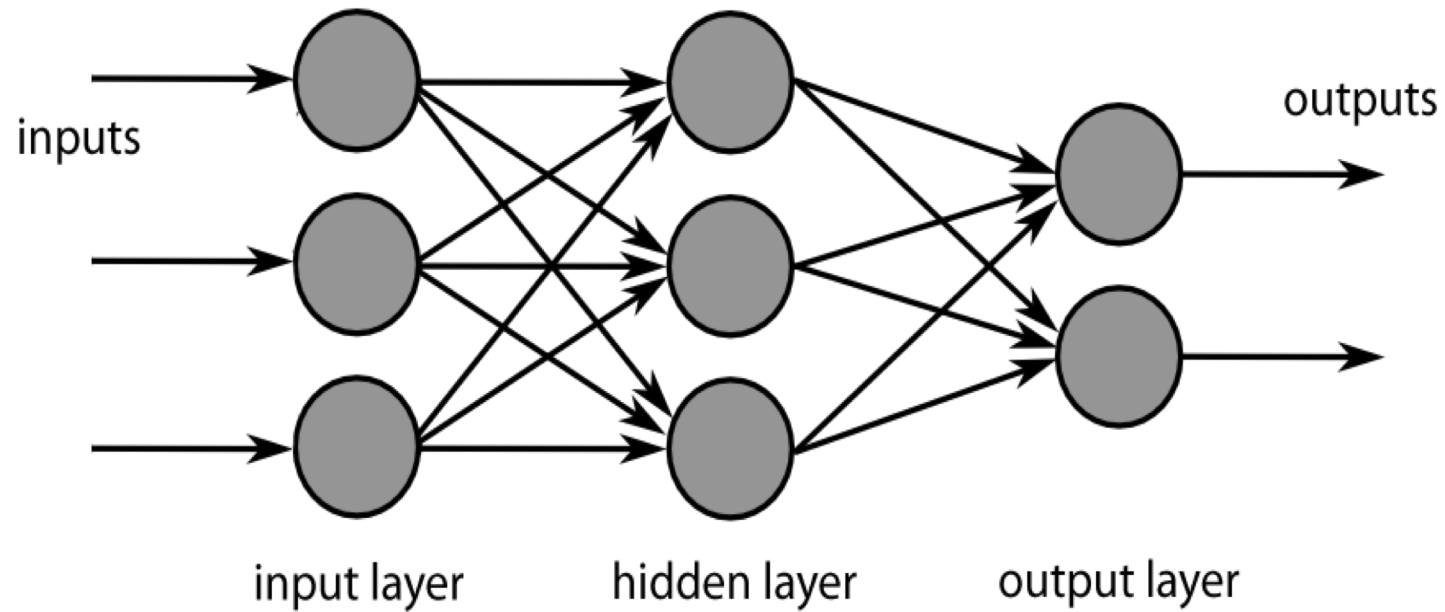
$$\text{output} = \begin{bmatrix} 148.4 / 176.3 \\ 7.4 / 176.3 \\ 0.4 / 176.3 \\ 20.1 / 176.3 \end{bmatrix} = \begin{bmatrix} 0.842 \\ 0.042 \\ 0.002 \\ 0.114 \end{bmatrix}$$



<https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d>

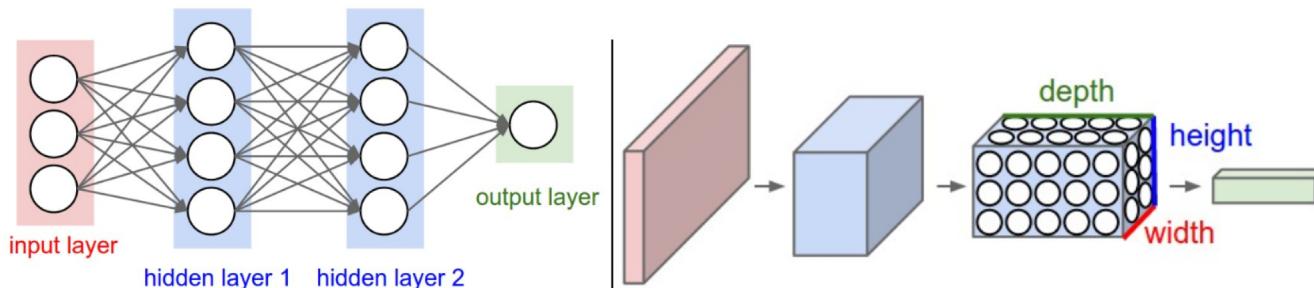
Scale March 7, 2019 / © 2019 IBM Corporation

Multilayer Perceptron



Convolutional Neural Network

- Layers are organized in 3-dimensions: width, height and depth.
- Not all neurons in one layer is connected to the other.
- Network performs series of convolution and pooling to extract features.



Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

<http://cs231n.github.io/convolutional-networks/>



Scale March 7, 2019 / © 2019 IBM Corporation

Convolutional Neural Network

Convolution

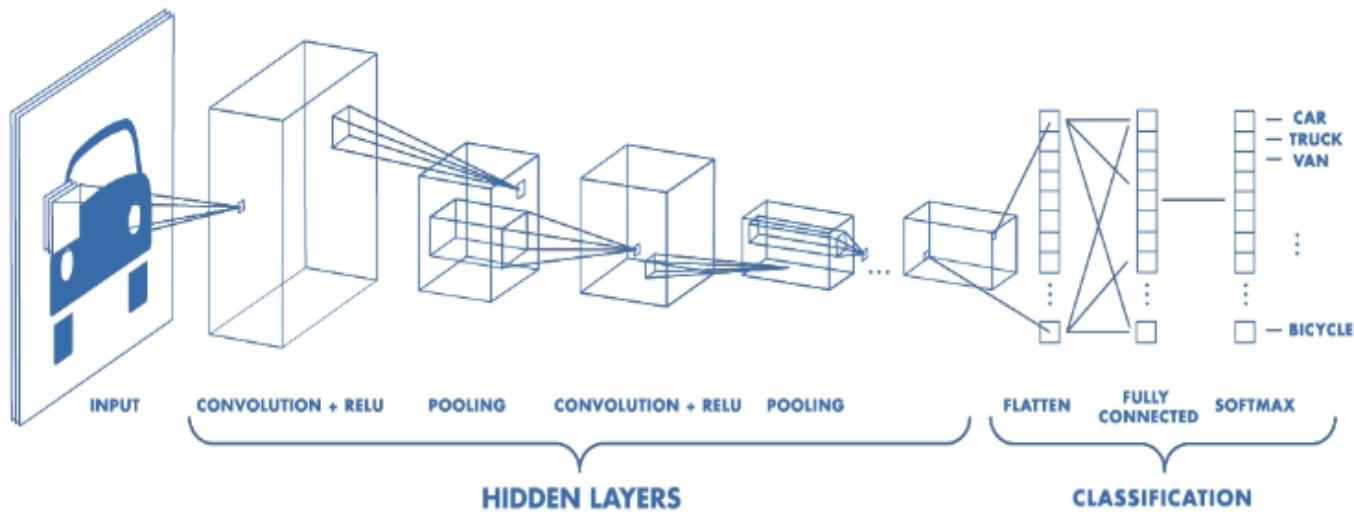
- Combination of two functions to produce a third function.
- Image + Kernel = Feature map

Pooling

- Reduce the dimensionality to reduce the number of parameters and computation in the network.
- This shortens the training time and controls overfitting.



Building CNN



<https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html>



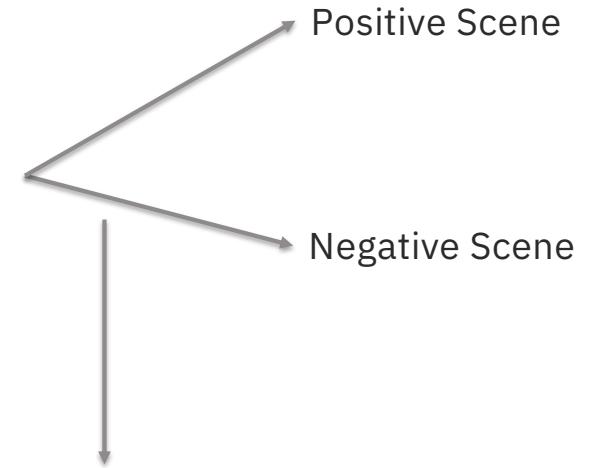
Silicon Valley Code Camp | October 13-14, 2018 / © 2018 IBM Corporation

The real game!



Image Caption Generator

**A group of young
men playing a game
of soccer .**



Sentiment Analysis



Scale March 7, 2019 / © 2019 IBM Corporation

Challenges in data model development:

- Know what are the available models for the problem.
- Pick out the best working code
- Solve installation and set-up issues
- Get the code up and running
(not all the models run in one go)



IBM Model Asset Exchange

Model Asset Exchange:

- One place for all **state-of-art** open source deep learning models.
- Wide variety of domains

Provides:

- Tested **code** and **IP**
- Both **trainable** and **trained** version

Started with the goal of:

- Making **deep learning models** easy to consume.



Model Asset Exchange Work Flow

Find a state-of-art open source deep learning model specific to domain

Validate license terms

Perform model health check & code clean up

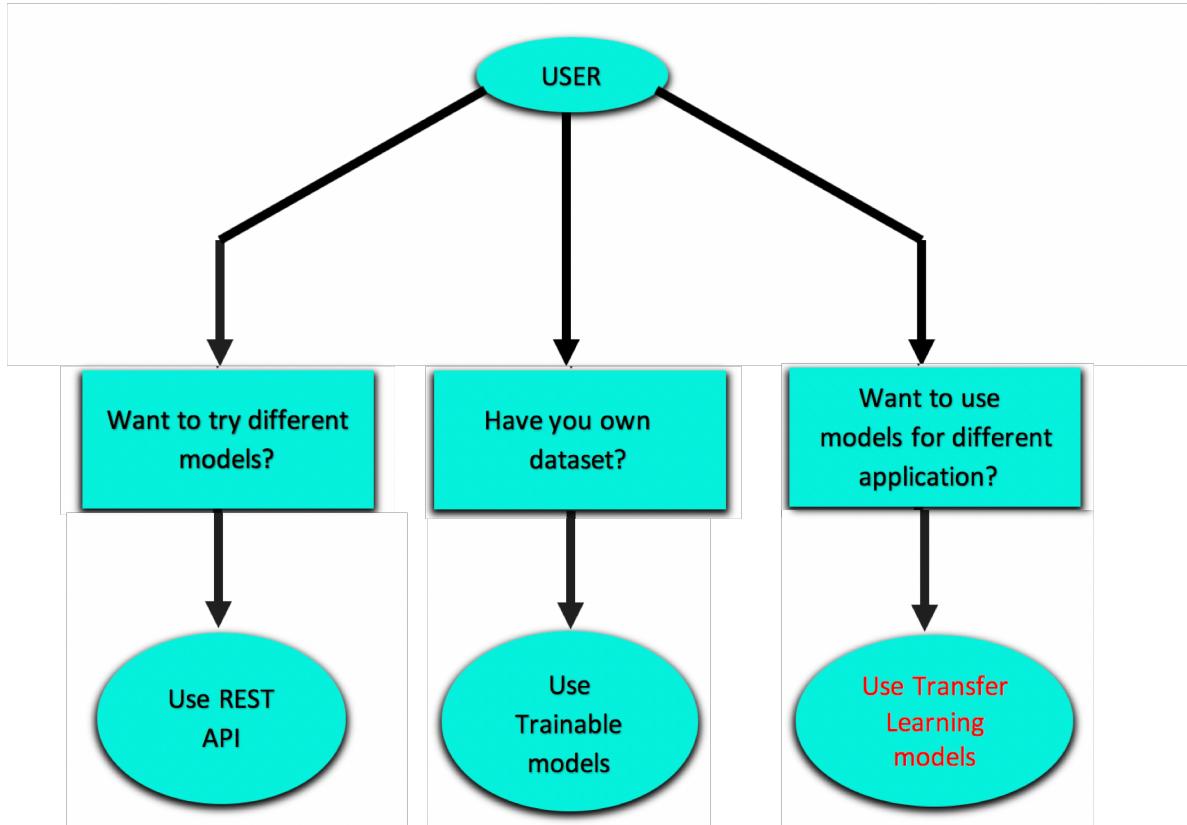
Wrap models in docker container with MAX framework to provide REST API

Test model and continuous integration

Regular model Update (version control)



The Next Steps...

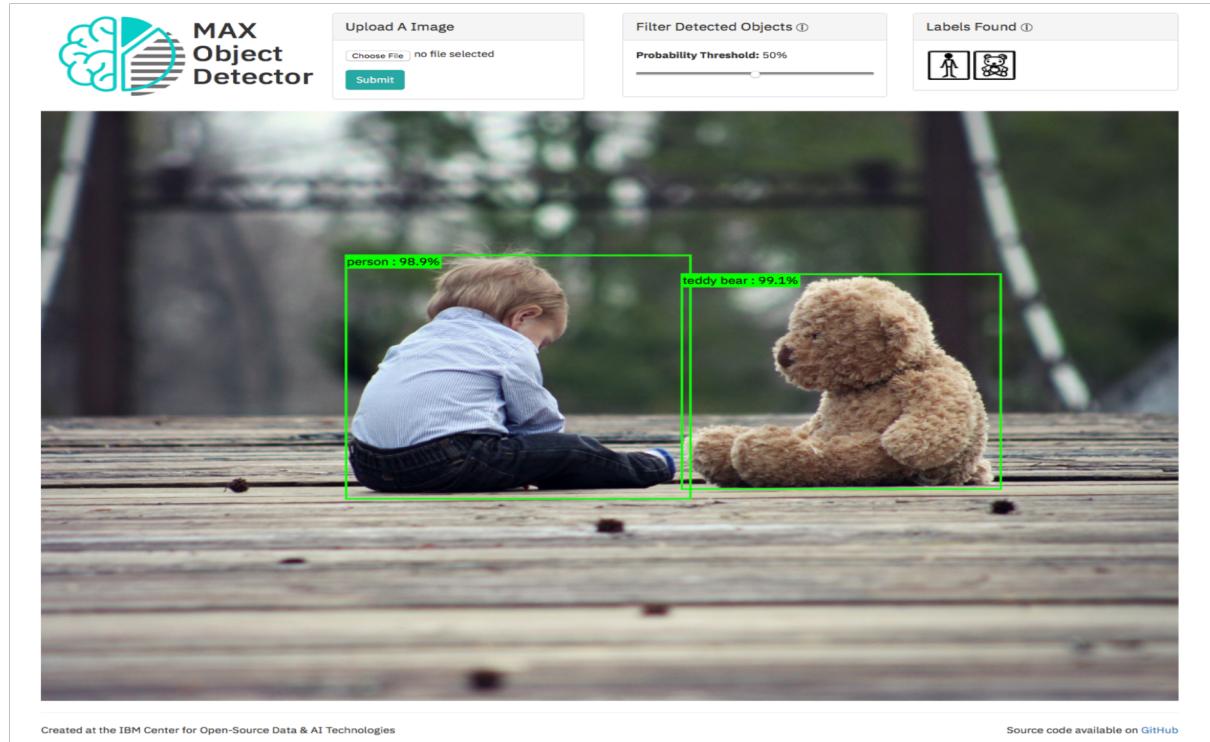


Demo of IBM Model Asset Exchange

Model : Object Detector

Model Asset Exchange link : [MAX](#)

Link to the model : [MAX-Object-detector](#)



Scale March 7, 2019 / © 2019 IBM Corporation

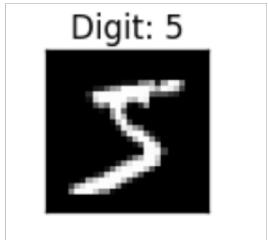
Summary

-
- 1 • Data science helps in resolving the business problem
 - 2 • To get answers for data science question, use data pipeline
 - 3 • Neural networks – MLP and CNN
 - 4 • MAX - Free, open-source models categorized into variety of domains
 - 5 • Tested code and IP
 - 6 • Build and deploy web service in less than 30 seconds
 - 7 • Start training on Watson Studio in minutes



Convolutional Neural Network - Workshop

Goal : MNIST Digits Classification



- A large database of handwritten digits (0 to 9).
- The database contains 60,000 training images & 10,000 testing images.
- Each image is of size 28x28.

<https://bit.ly/2SLa6Ep>



Scale March 7, 2019 / © 2019 IBM Corporation

To-do: Use Keras functions to create a simple CNN

[https://bit.ly/2SLa6Ep.](https://bit.ly/2SLa6Ep)

Use either `Jupyter notebook` or `Google colab`

1. Use [`CNN_MNIST_Workshop.ipynb`](#) notebook
2. Complete Code under `Build CNN model` using [`CNN_MNIST_1_Conv_layer.ipynb`](#) as reference. Reference program has 1 layer Convolution. For workshop, create two layer convolution network.

Solution:

1.

```
model = Sequential()  
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(28,28,1)))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
model.add(Conv2D(64, (3, 3), activation='relu'))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
model.add(Flatten())  
model.add(Dense(50, activation='relu'))  
model.add(Dense(num_classes, activation='softmax'))
```

2. pred = model.predict(x_test)



Web App Creation Workshop Link

<https://bit.ly/2Hkc7oP>

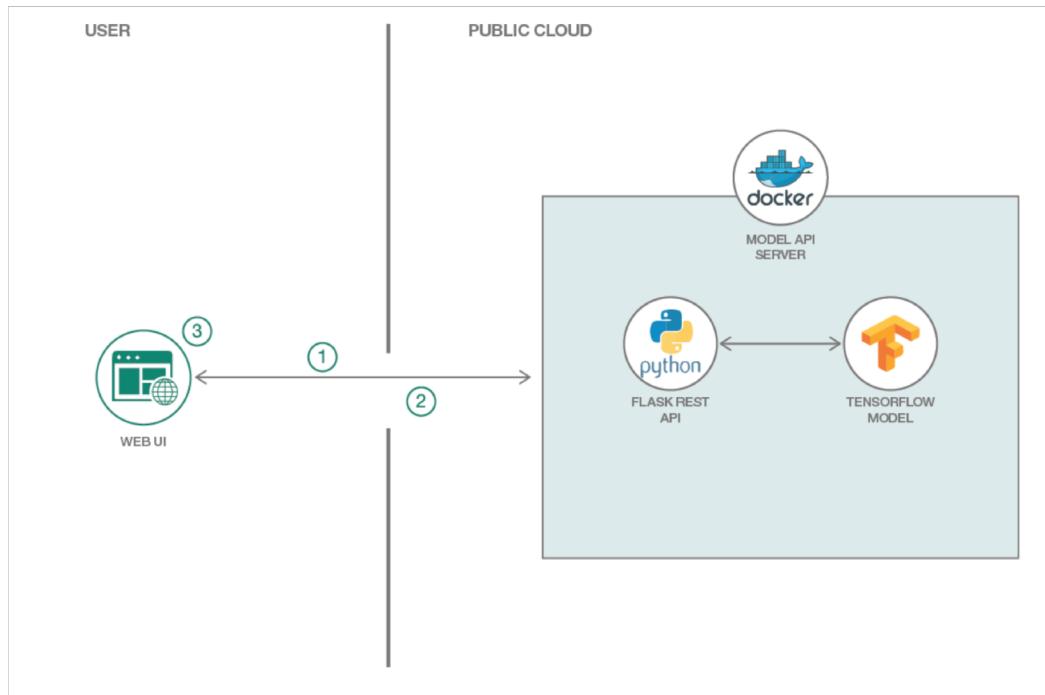
In app.py

Fill **TODO T1** with model URL which is '**/model/predict**'

Model API returns predictions in JSON format. To retrieve result, replace **TODO T2** With JSON key '**predictions**'.



Workshop – Create a web app using MAX Object Detector



- User uses Web UI to send an image to Model API.
- Model API returns object data and Web UI displays detected objects.



To-do 1: Prepare input data for prediction

```
# TODO R1: review inference request payload
# Required inference request parameter: image (JPG/PNG encoded)
files = {
    'image': image_encoded.tostring(),
    'Content-Type': 'multipart/form-data',
}

# Optional inference parameter: threshold (default: 0.7, range [0,1])
data = {'threshold': '0.5'}
```



To-do 2: Send input image to Model API

```
# TODO T1: replace model URL placeholder  
model_url = args.ml_endpoint.rstrip('/') + '**TODO**'  
  
# Send image file form to model endpoint for prediction
```

Fill **TODO** with model URL which is **'/model/predict'**



To-do 3: Retrieve results from Model API

```
# TODO T2: uncomment next line and replace placeholder
# result = output_data['**TODO**']

if len(result) == 0:
    msg = 'No objects detected, try uploading a new image'
    return render_template('index.html', error_msg=msg)
```

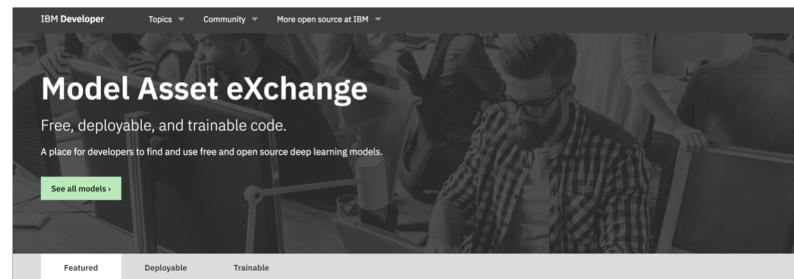
Model API returns predictions in JSON format. To retrieve result, replace ****TODO**** With JSON key **'Predictions'**.



Resources



<https://developer.ibm.com/>



<https://developer.ibm.com/exchanges/models/>



Scale March 7, 2019 / © 2019 IBM Corporation



saishruthis



SSaishruthi



saishruthi.tn@ibm.com



Scale March 7, 2019 / © 2019 IBM Corporation

