First step Deep Learning with Tensorflow

Jiqiong QIU

About me

- Master of engineer degree at INSA de Lyon
- PhD in Color Formulation by Statistical Learning at UTC and BASF Coatings
  (EP 2887275A1: Method and system for determining a color formula)
- Data scientist at Sfeir

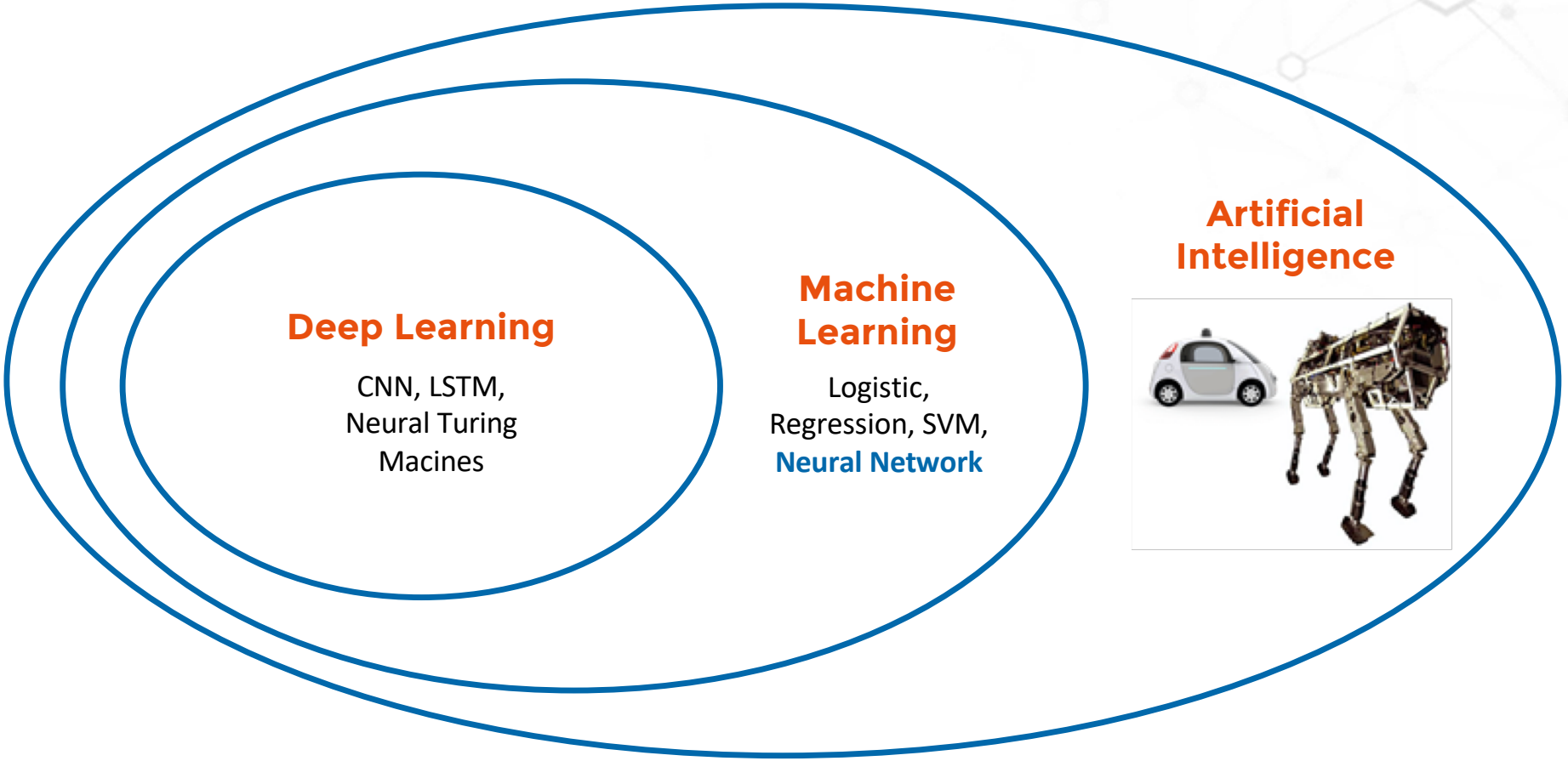
First step Deep Learning with Tensorflow

1. What is Deep Learning?
2. Why Tensorflow?
3. Digital Recognition using CNN
4. Resources

1 What is Deep Learning?

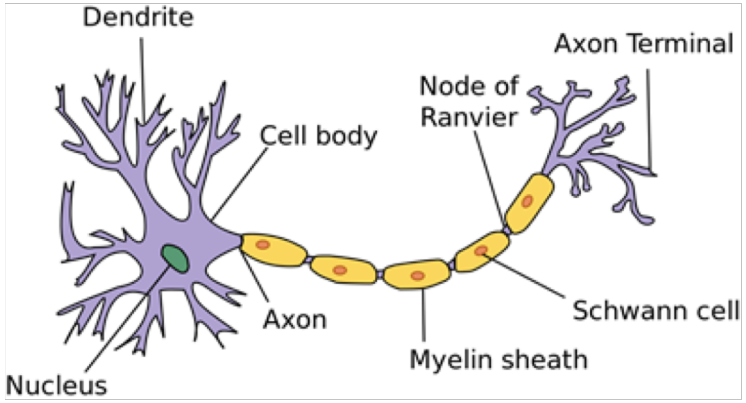
1 What is Deep Learning

1.1 From AI to Deep Learning

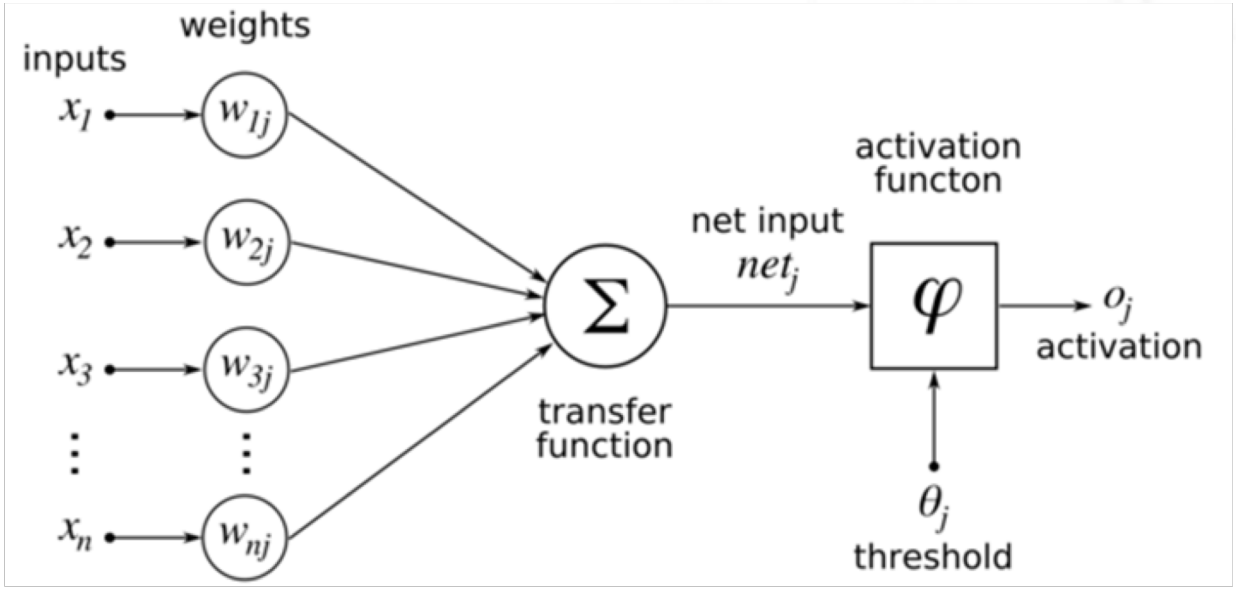


1.2 Neural Network

Neuron

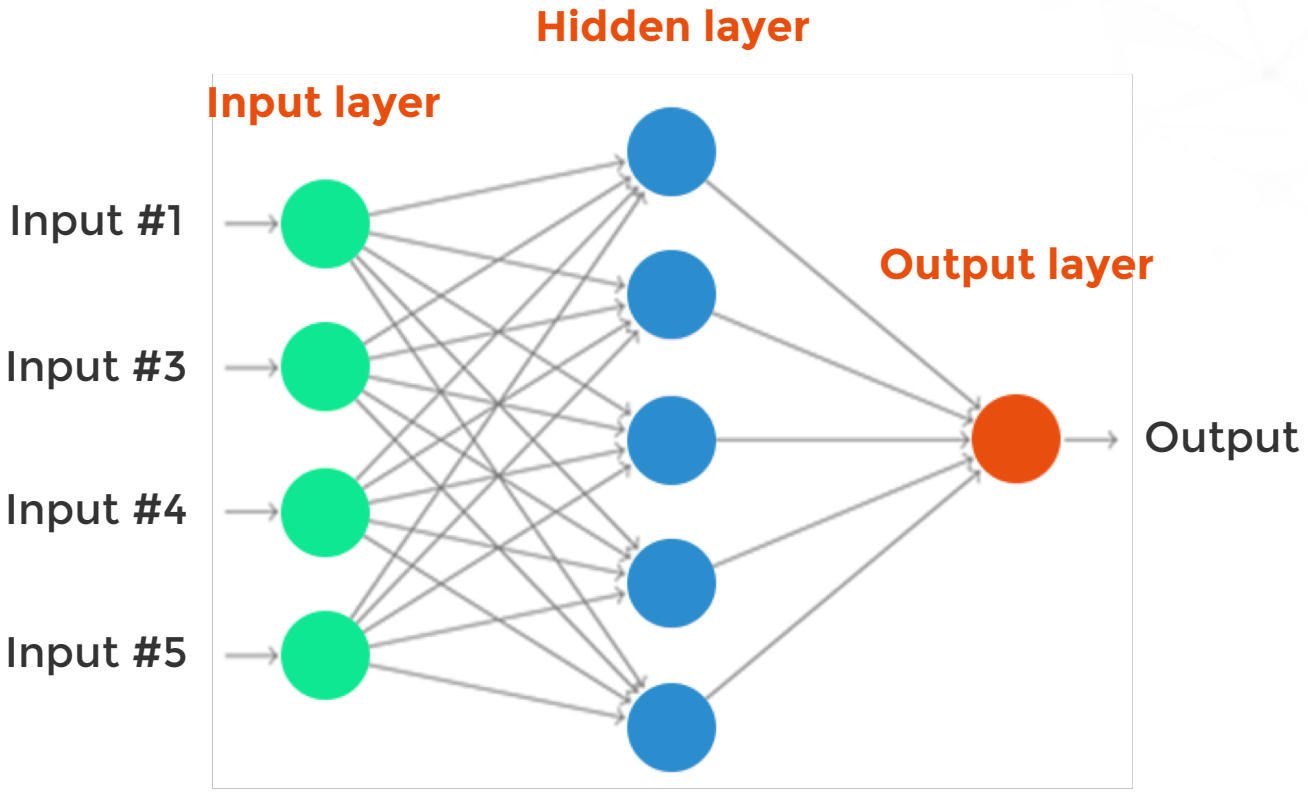


Artificial Neuron



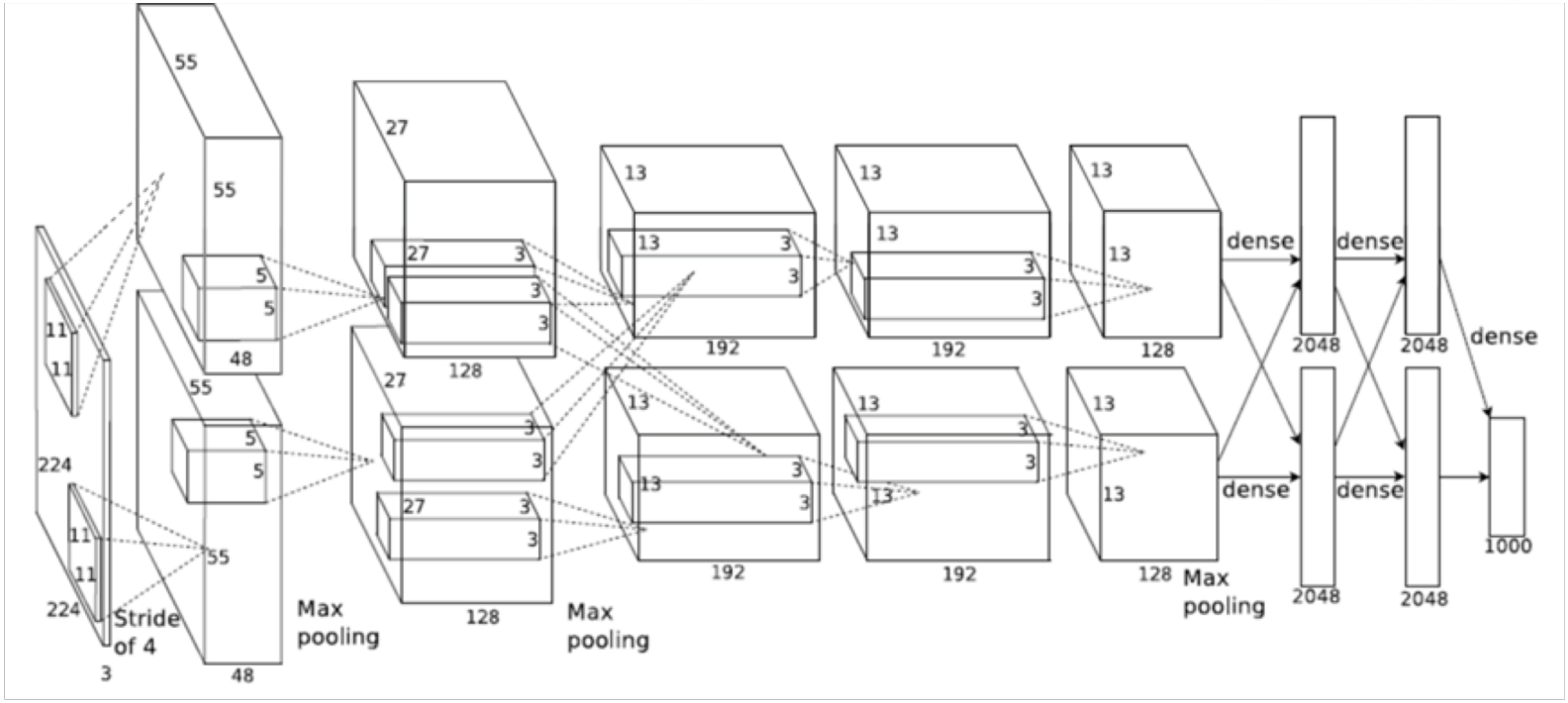
1.2 Neural Network

MLP: Multilayer Perceptron



1.3 Deep Learning

- More layer
- Different Neuron types

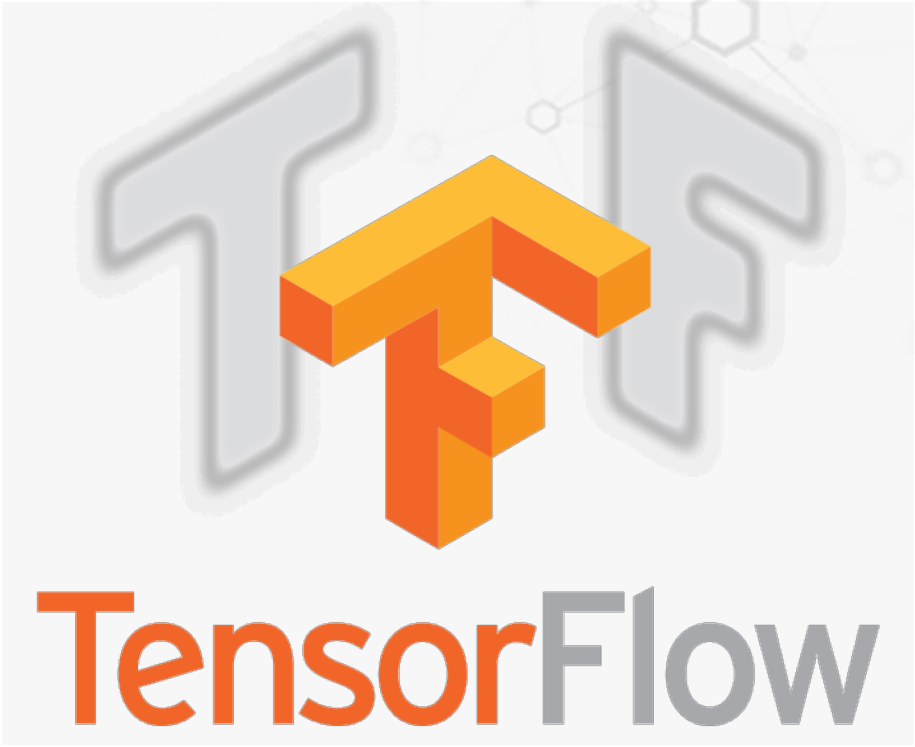


ALEXNET

2 Why Tensorflow?

2 Why Tensorflow?

- Open source by Google
- Python API
- Board
- Android (SDK) Mobile application



And more: http://deeplearning.net/software_links/

3 Digital Recognition using CNN

3 Digital Recognition using CNN

3.1 Digital Recognition and Data Set

3.2 CNN

3.1 Digital Recognition and Data Set

MNIST database: Mixed National Institute of Standards and Technology database

- Images **28x28** pixels
- Training set: **60 000**
- Test set: **10 000**

3.2 CNN : Convolutionnal Neural Network

3.2.1 Convolution layer

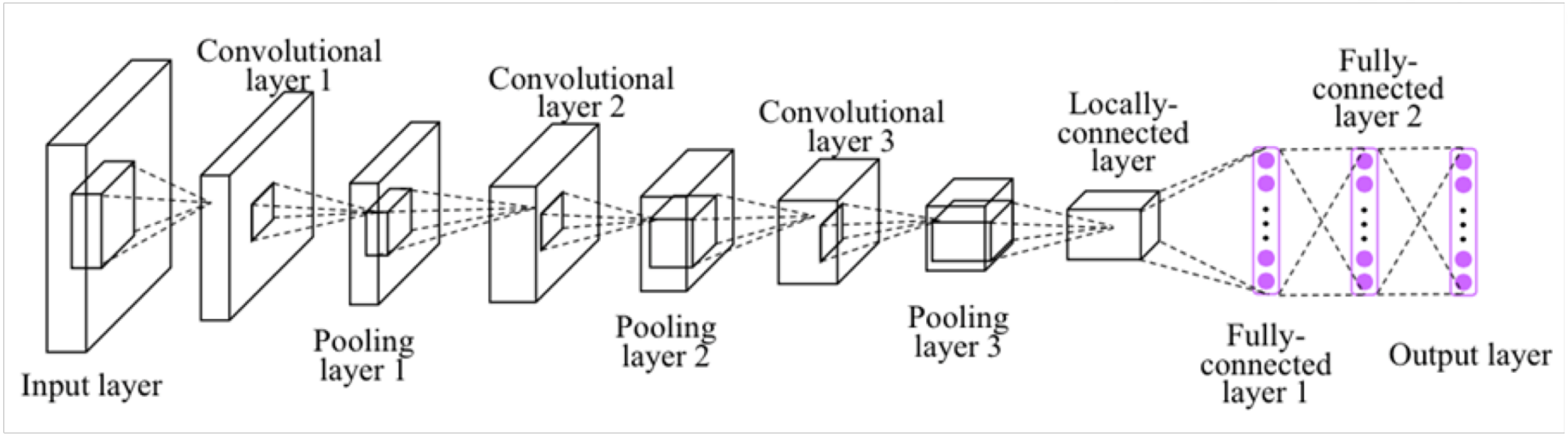
3.2.2 Pooling layer

3.2.3 MLP and more

3.2.4 Training and Results

3.2 Digital Recognition using CNN

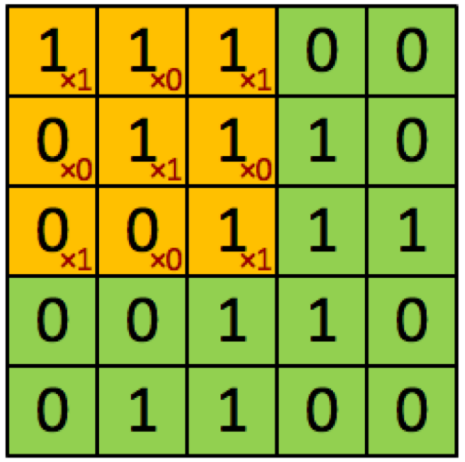
Convolutionnal Neural Network

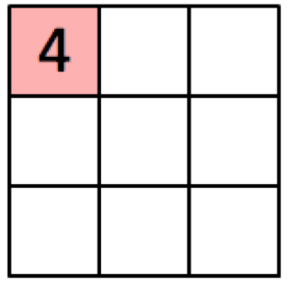


3.2.1 Convolution layer

Natural Images have the property of being “stationary”

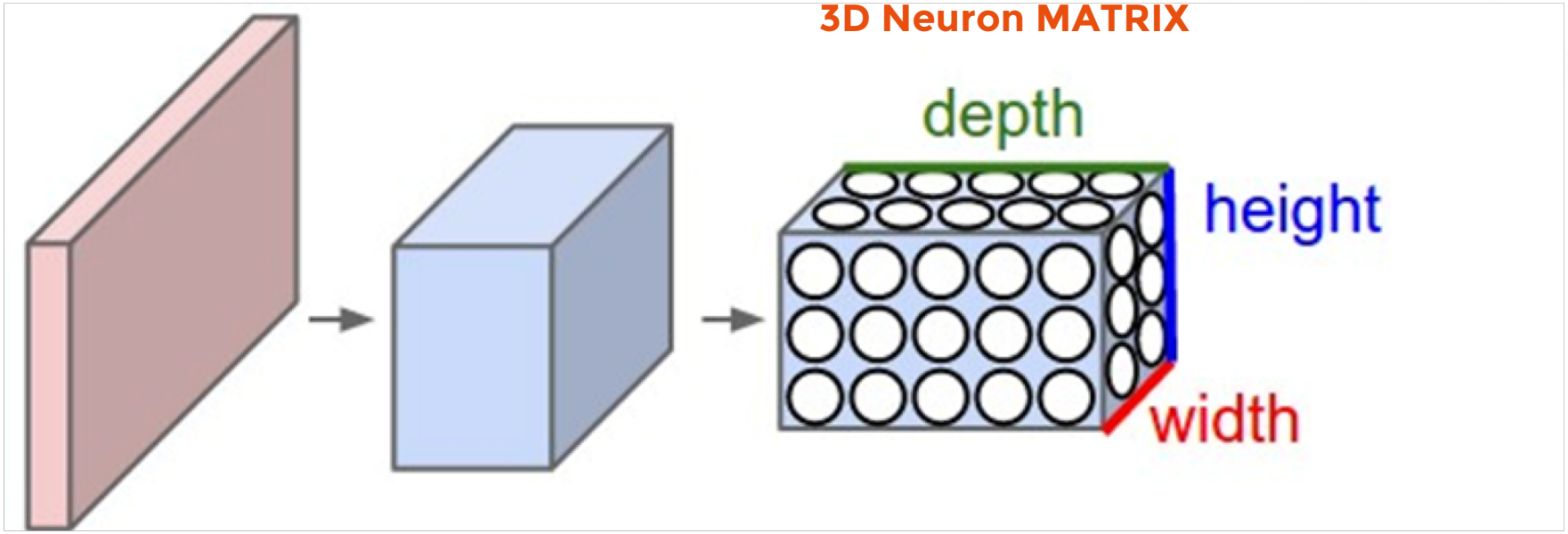
3.2.1 Convolution layer



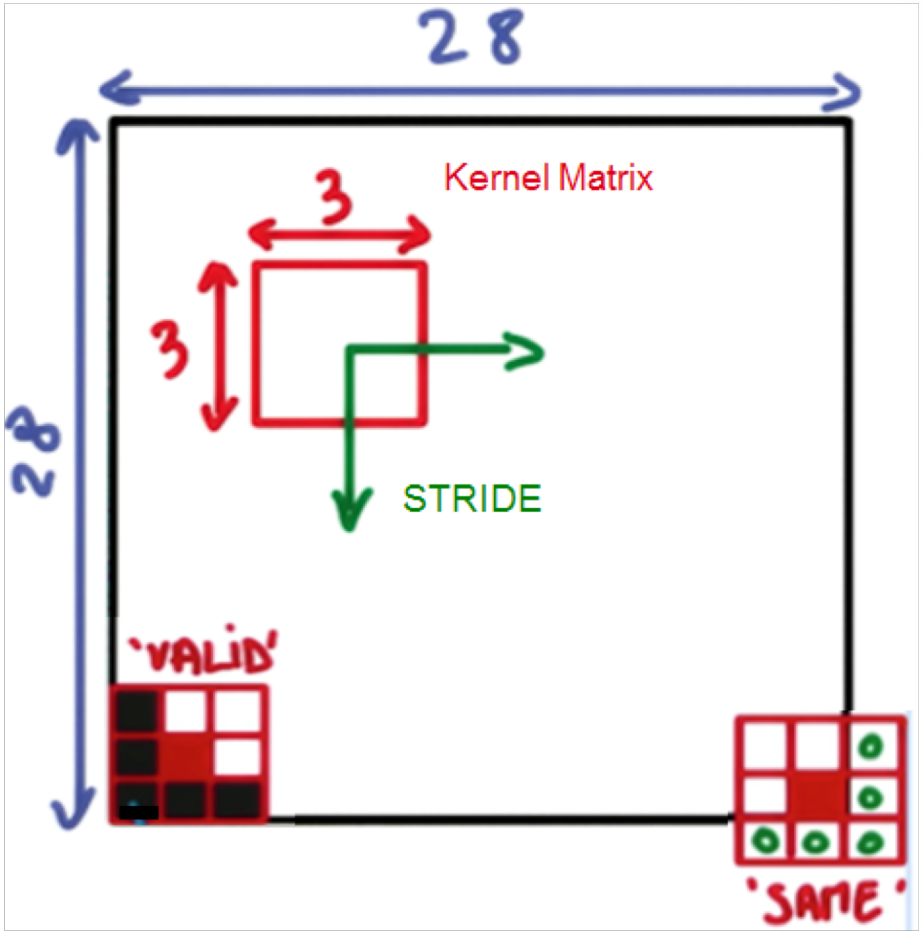


3.2.1 Convolution layer

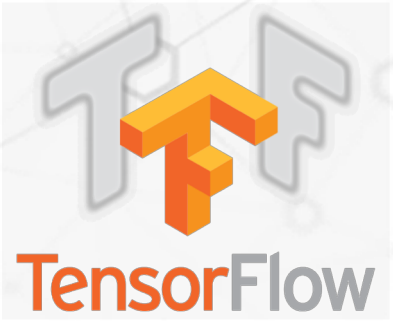
Input Depth = 3 (rgb)
Output Depth = 400



3.2.1 Convolution layer



3.2.1 Convolution layer: Tensorflow

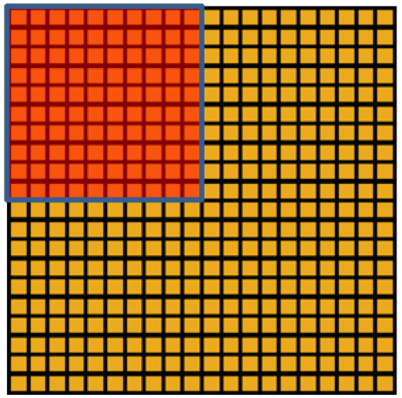



```
def conv2d(x, W):
    return tf.nn.conv2d(x, W, strides=[1, 1, 1, 1], padding='SAME')
```

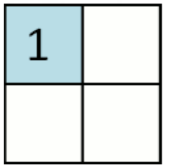
3.2.2 Pooling layer

Pooling (or Subsampling)

- **28x28** pixels
- 400 features (depth) over 3x3 inputs
- 2 strides
- 13x13x400 = 169x400= **67600**

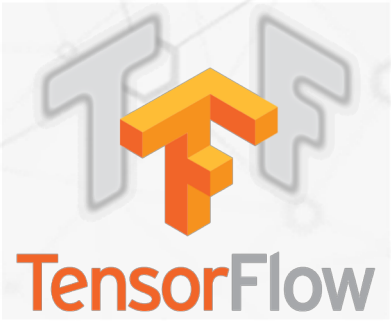


Convolved feature



Pooled feature

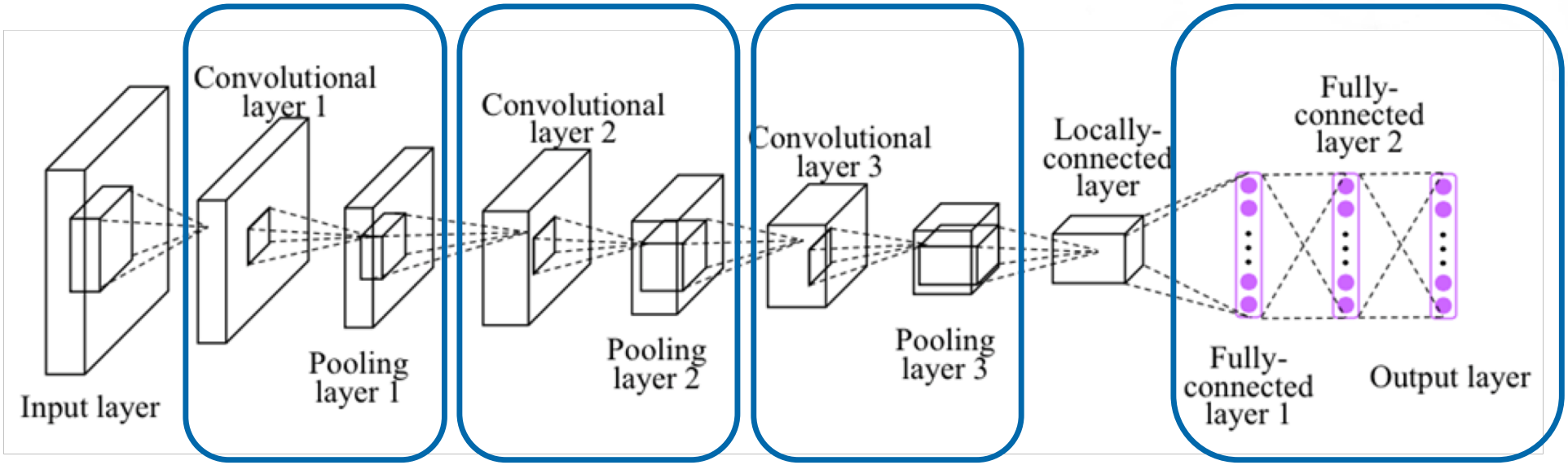
3.2.2 Pooling layer: Tensorflow




```
def max_pool_2x2(x):
    return tf.nn.max_pool(x, ksize=[1, 2, 2, 1],strides=[1, 2, 2, 1], padding='SAME')
```

3.2.3 MLP and More

Convolution + Pooling **MLP**

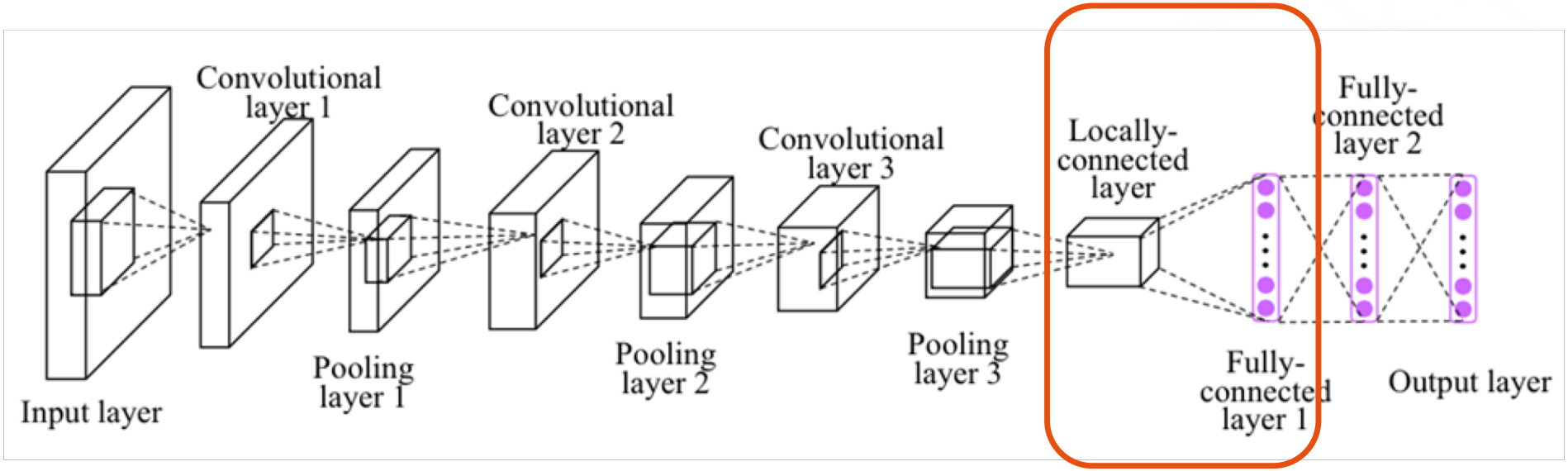


3.2.3 MLP and More

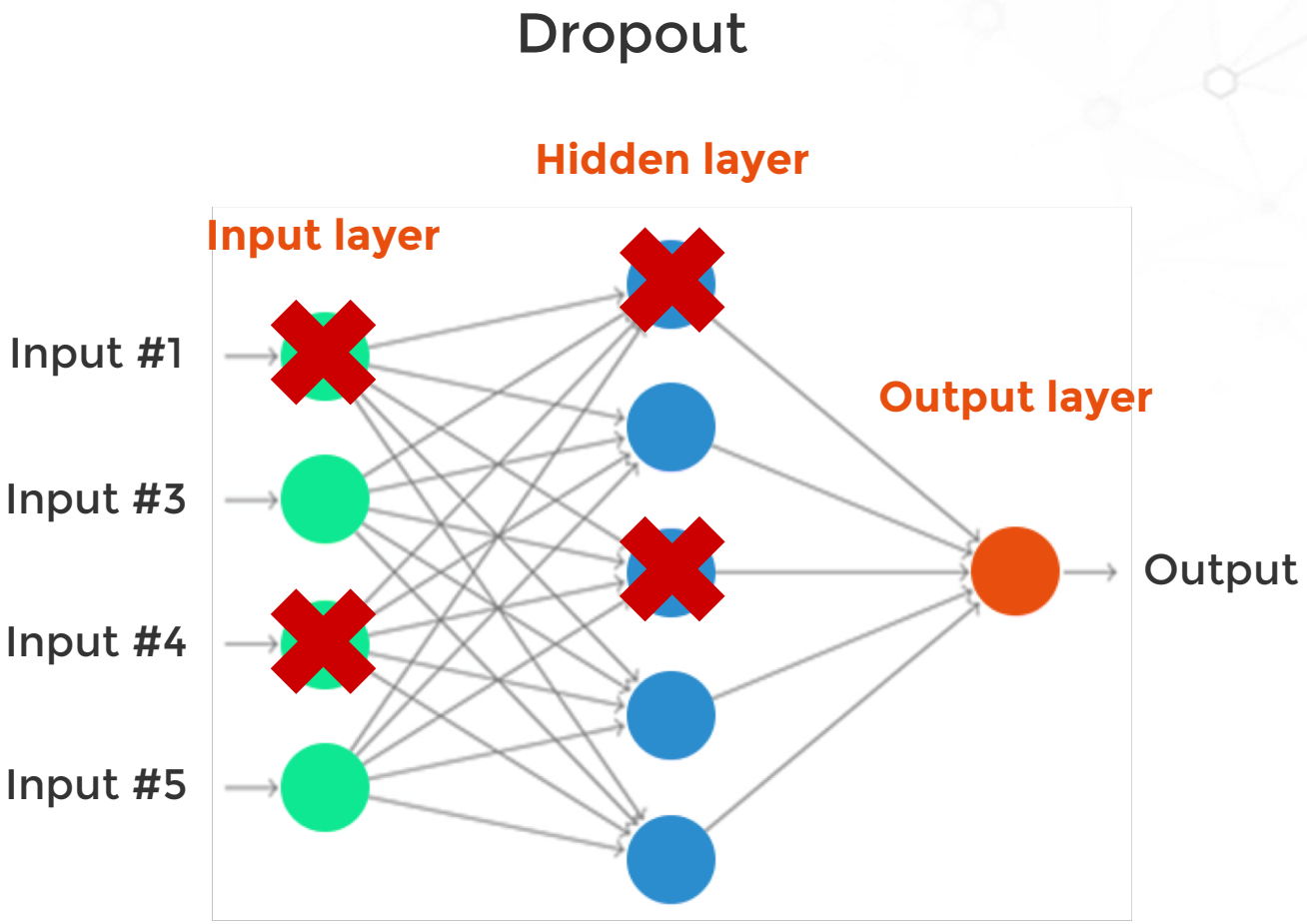
Convolution + Pooling

MLP

Reshape (3D to 1D)



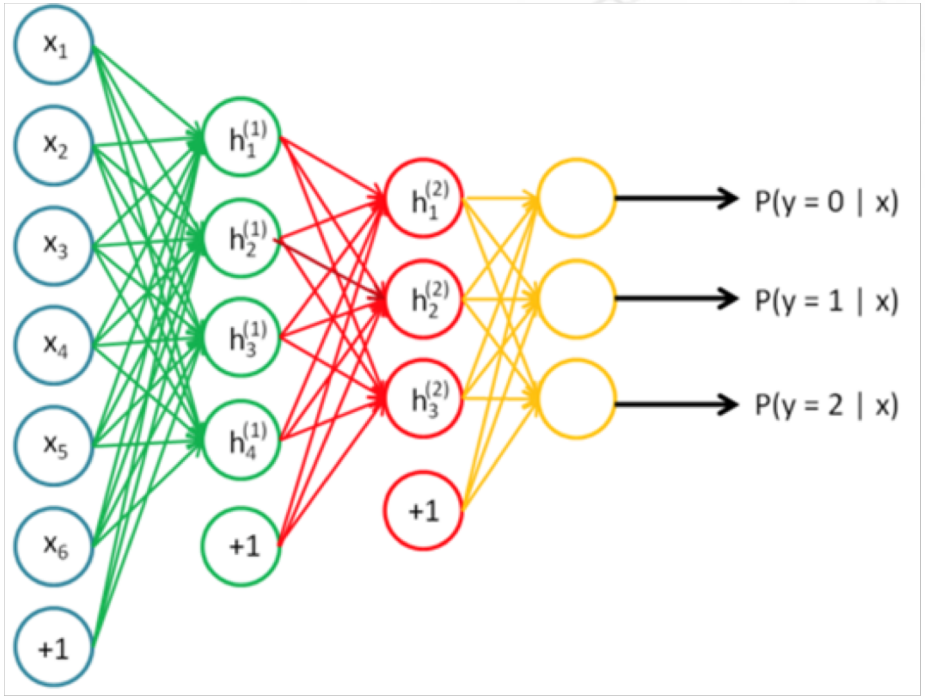
3.2.3 MLP and More



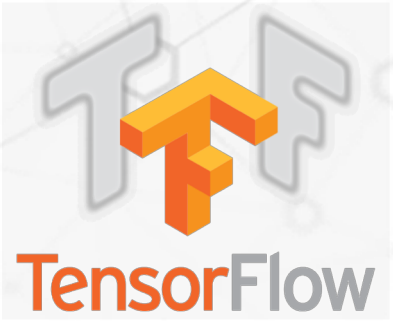
3.2.3 MLP and More

- Multiple Classification: [0,1,..,8,9]
- Softmax:

$$P(y = j|\mathbf{x}) = \frac{e^{\mathbf{x}^\top \mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\top \mathbf{w}_k}}$$



3.2.3 MLP and More: Tensorflow



- **Reshape**

```
tf.reshape
```

- **Dropout**

```
tf.nn.dropout(h_fc1, keep_prob)
```

- **Output of CNN**

```
tf.nn.softmax
```

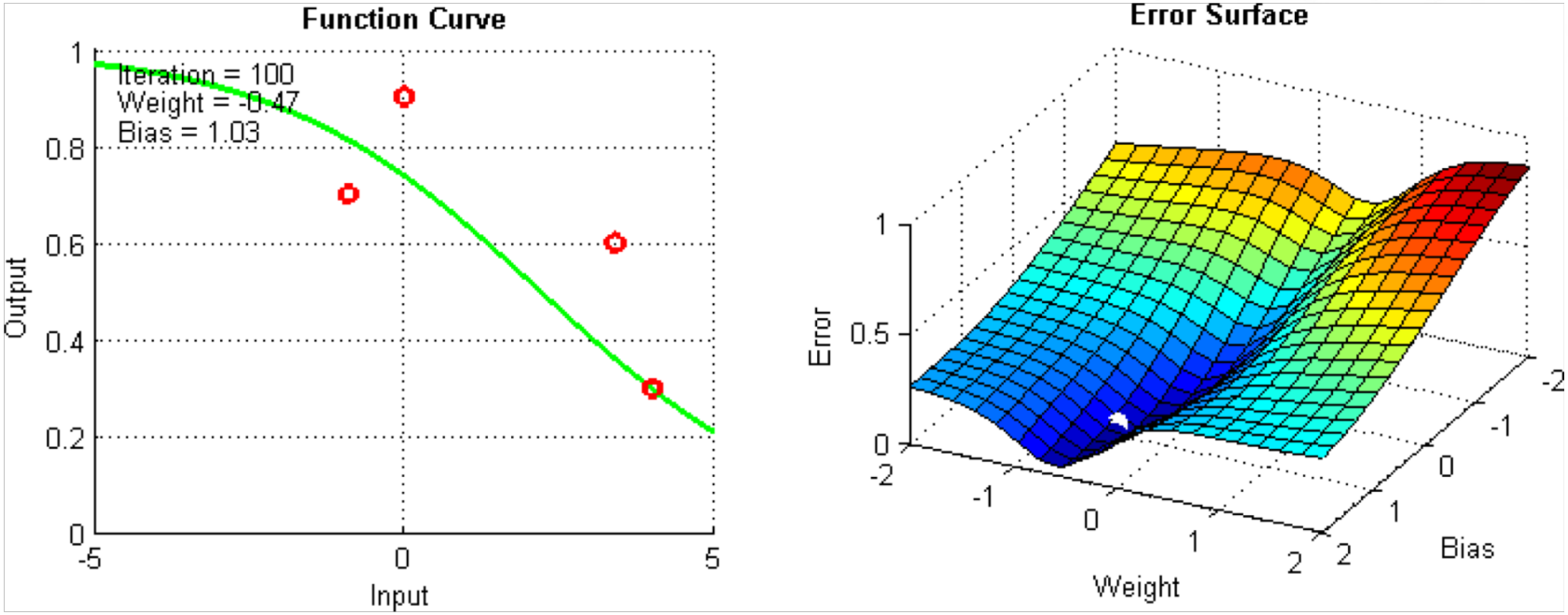
3.2.4 Training and Results

- Cost Function: $$\mathrm{H}(p,q) = -\sum_{x} p(x) \log q(x).$$

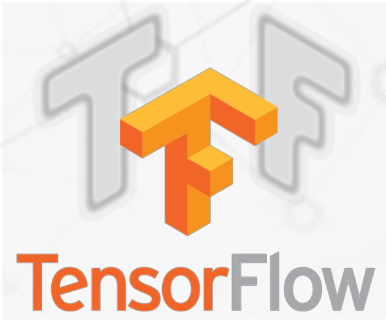
- Cost Function is defined by:
  - Input Values (images, noted as x)
  - Weight matrix (w)

3.2.4 Training and Results

Training process



3.2.4 Training and Results: Tensorflow



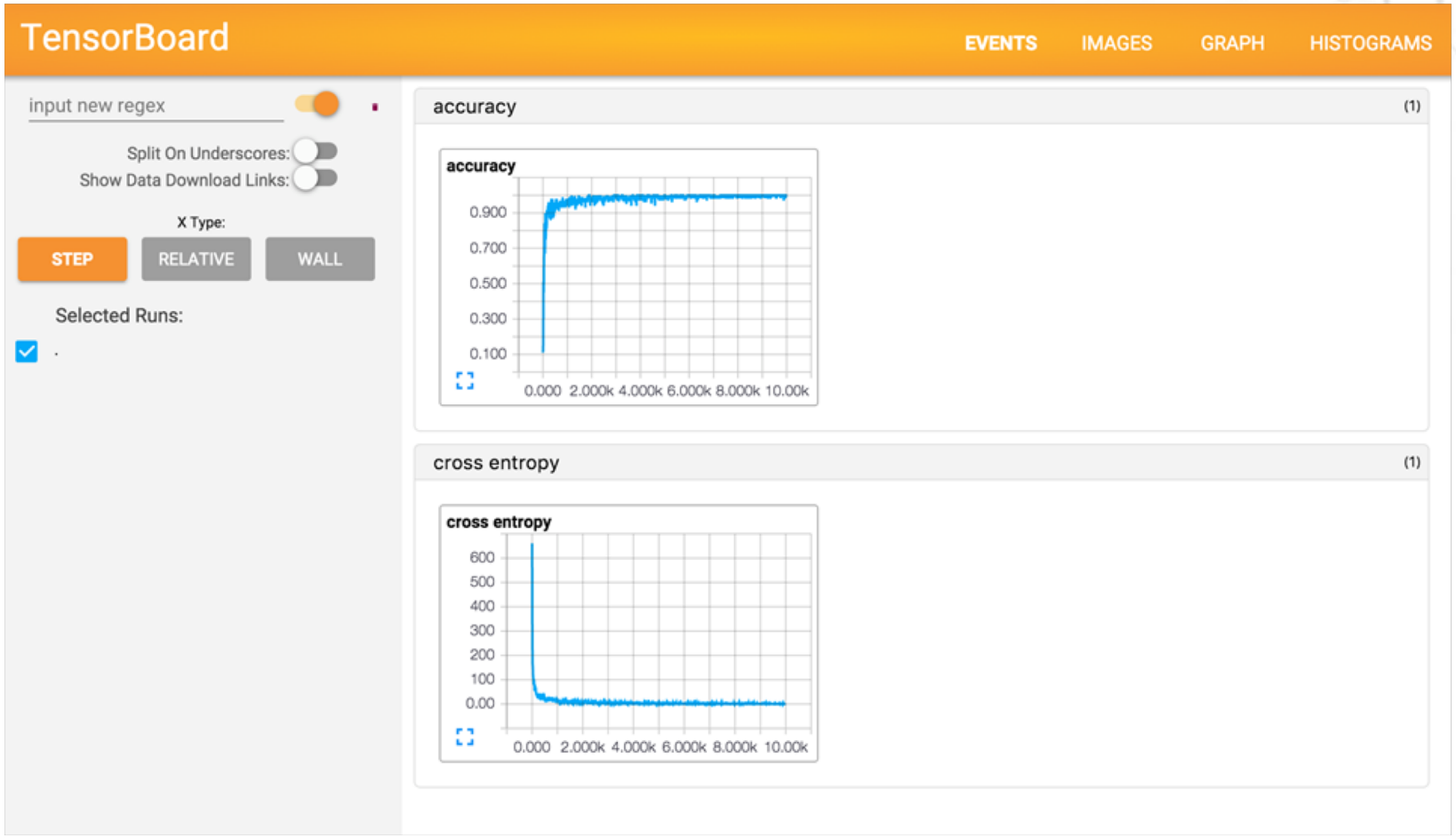
- **Cost Function**

```
cross_entropy = -tf.reduce_sum(y_*tf.log(y_conv))
```

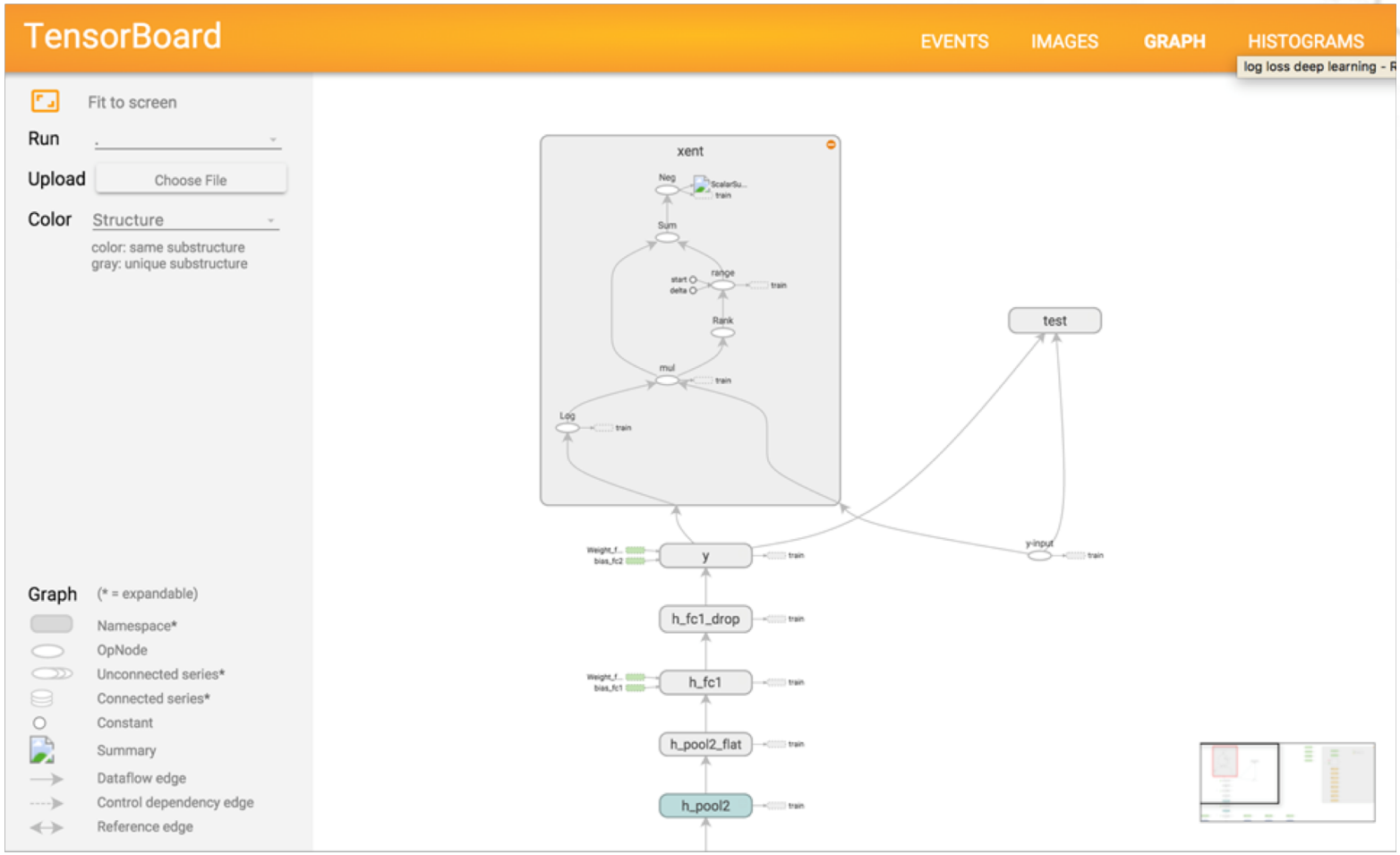
- **Training process**

```
train_step = tf.train.AdamOptimizer(1e-4).minimize(cross_entropy)
```

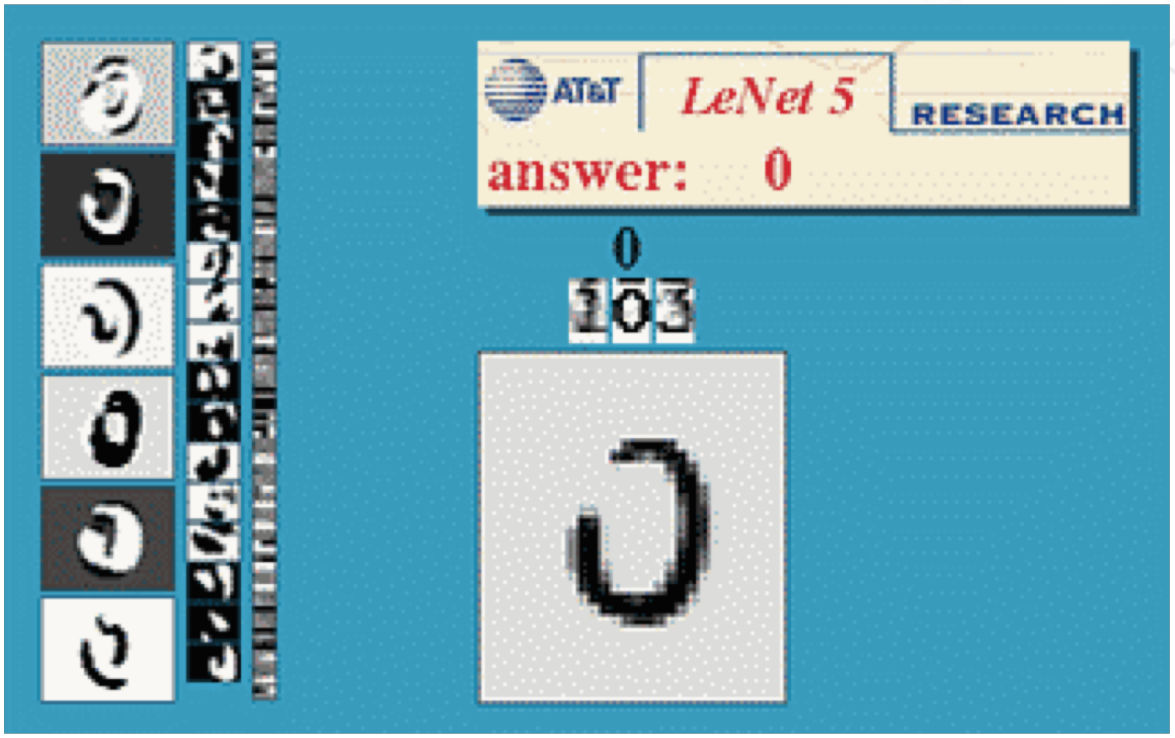
3.2.4 Training and Results



3.2.4 Training and Results



3.2.4 Training and Results



4 Resources

4 Resources

- For this tutorial:

  https://github.com/Sfeir/demo-tenserflow

  https://docs.google.com/presentation/d/1ch4YiKD83wERmmEFRvFIQ98Mtz65aGsLP3uWiTElY2I/edit?usp=sharing

- MOOC:

  CS229: Machine Learning

  UD730: Deep Learning - Taking machine learning to the next level

  CS231n: Convolutional Neural Networks for Visual Recognition

  CS224d:Deep Learning for Natural Language Processing

- Summer school:

  IPAM 2012

  Montreal 2015

- Website

  http://deeplearning.net/ https://www.topcoder.com/ https://www.kaggle.com/

Thank you.

by Jiqiong QIU