

Distributed TensorFlow & Stock Prediction

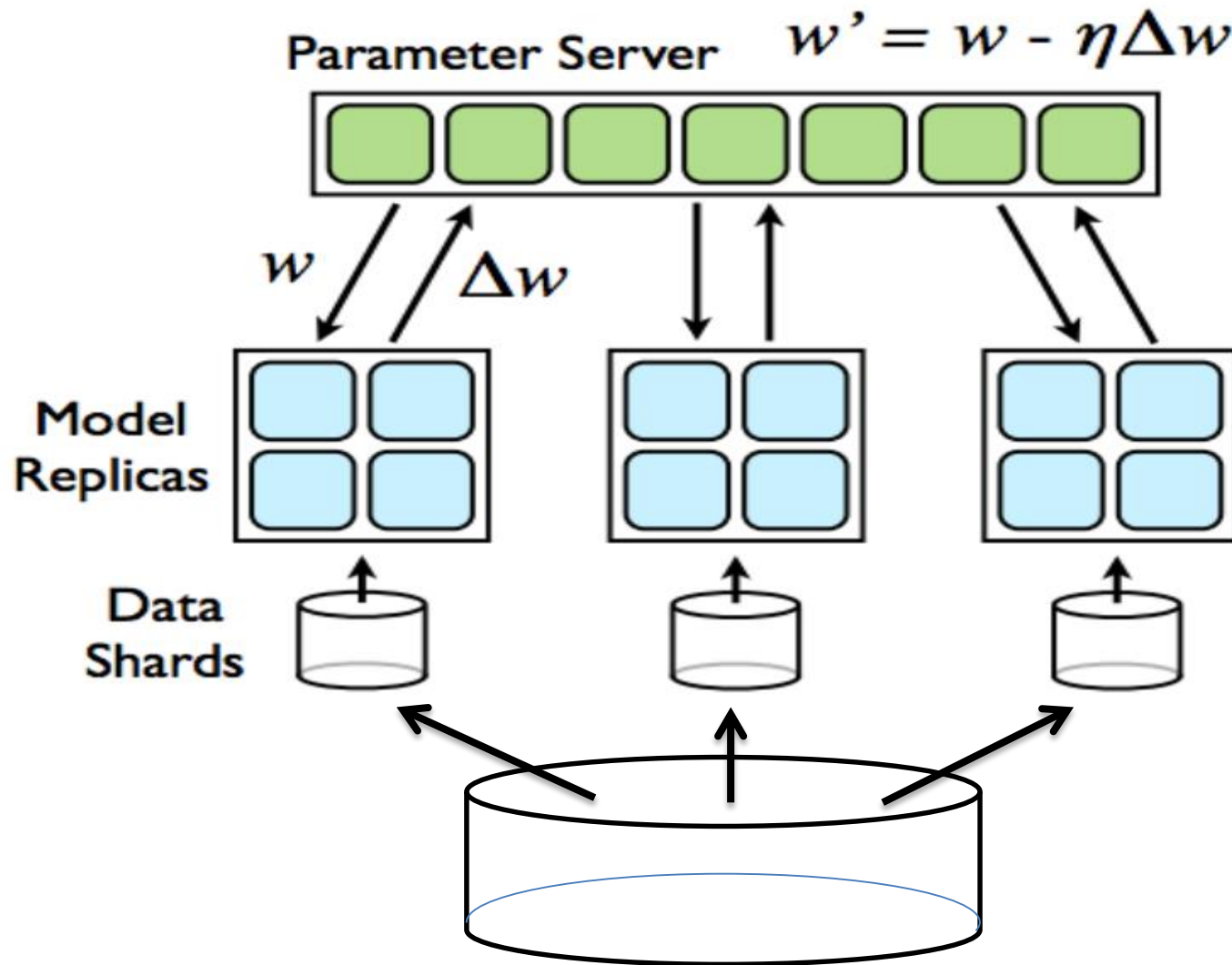
Yi Chin Chen

2016/09/21

Outline

- Distributed Training
- TensorFlow description
- Supervised Machine Learning
 - CNN model
- Reinforcement Learning
 - DQN model

Distributed Training



Distributed Training

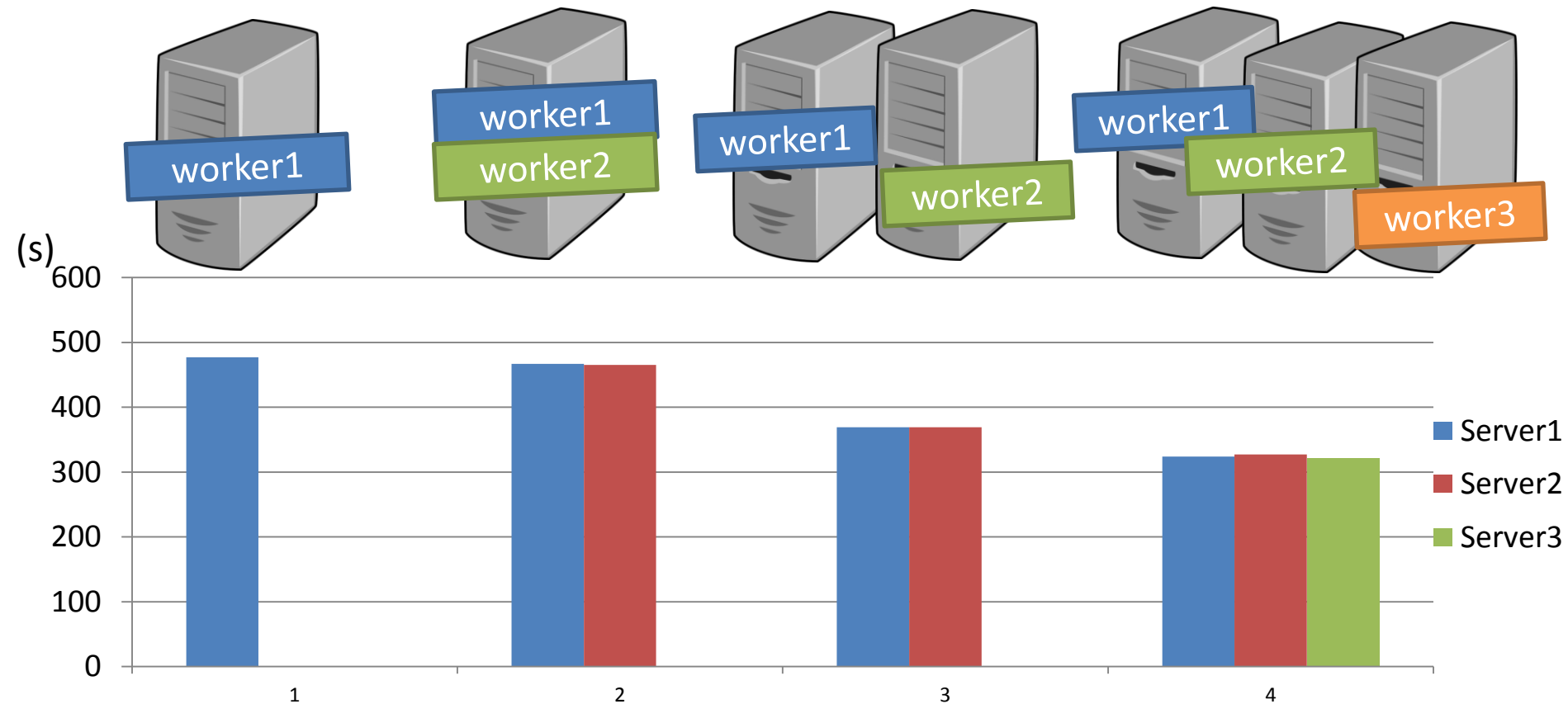
- Need to consider transmission time
- That is,

$$T_{consumed} \approx T_{transmit} + T_{compute}$$

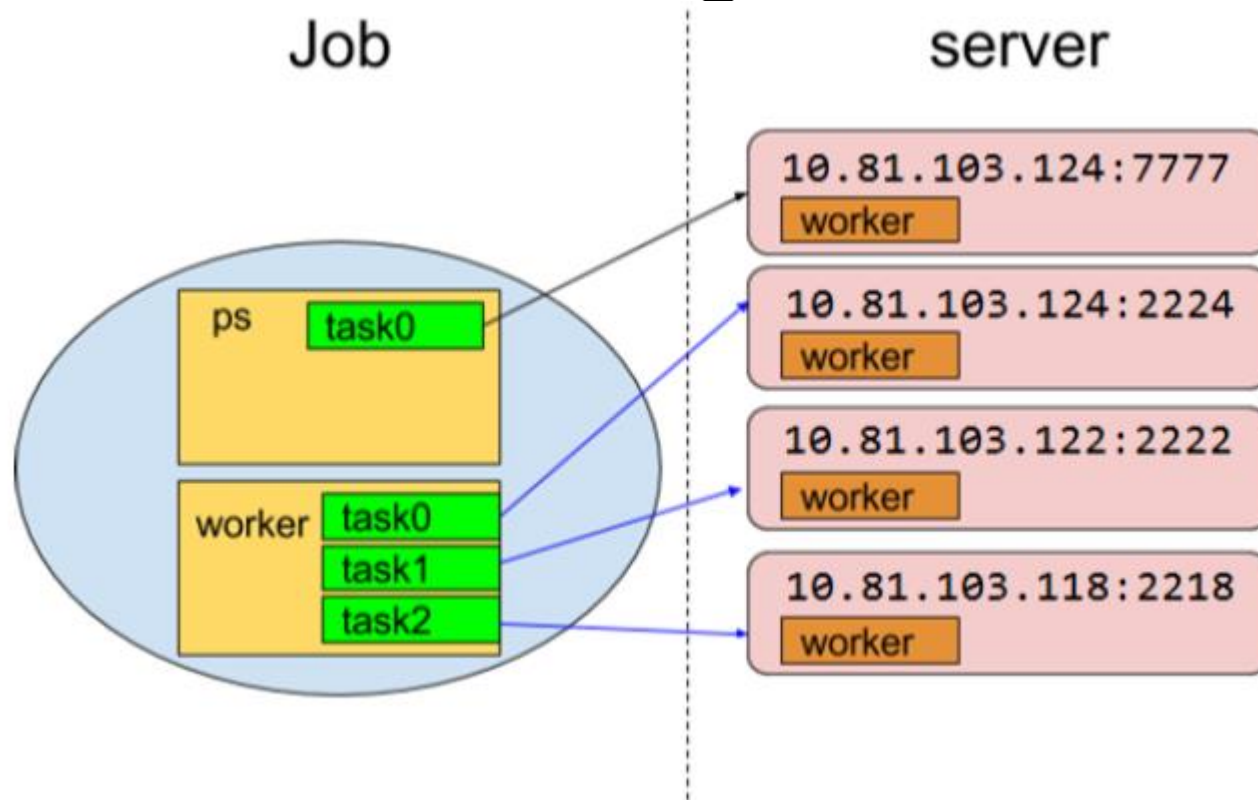
- If $T_{transmit} \gg T_{compute} \Rightarrow \text{low efficiency}$

Distributed Training

- Experiment on a 5-layer CNN model



Distributed Training on TensorFlow



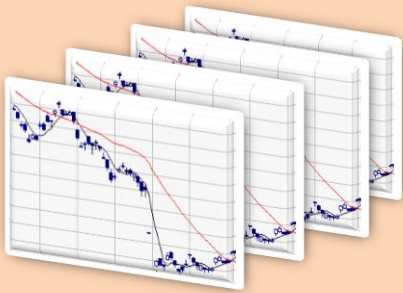
- Define a working cluster{"ps","worker"}
- Set task (here we do data parallelism, so each task is same)
- Assign task to server while connecting

Distributed Training

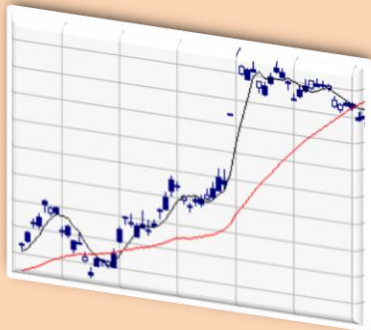
- For MNIST distributed training sample, please check the following work
- https://github.com/weitingforyou/tensorflow-stock-prediction/blob/master/distributed-tensorflow/mnist_replica_cnn.py

Stock movement prediction

Testing data



Historical data



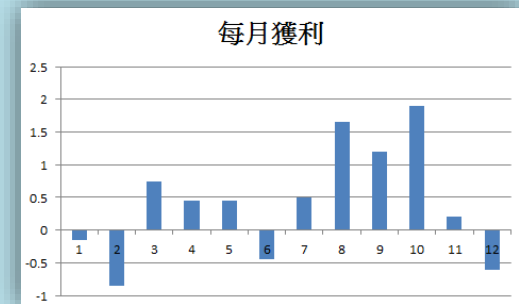
Today' price



Well-trained
model



Evaluation



Result

Buy?
Sell?



- A library for machine Intelligence
- Based on Python, C++
- Open Source
- Represent computations as graphs

TensorFlow

Machine Learning
Library
(ex, Weka)

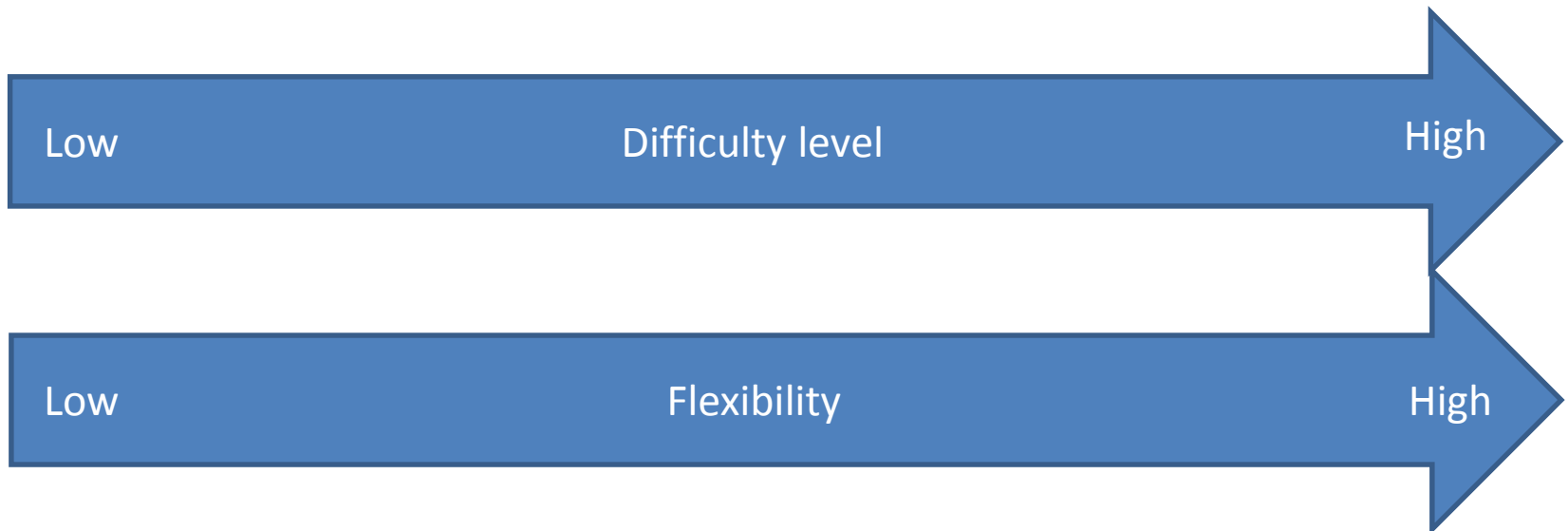
TensorFlow

Write by yourself

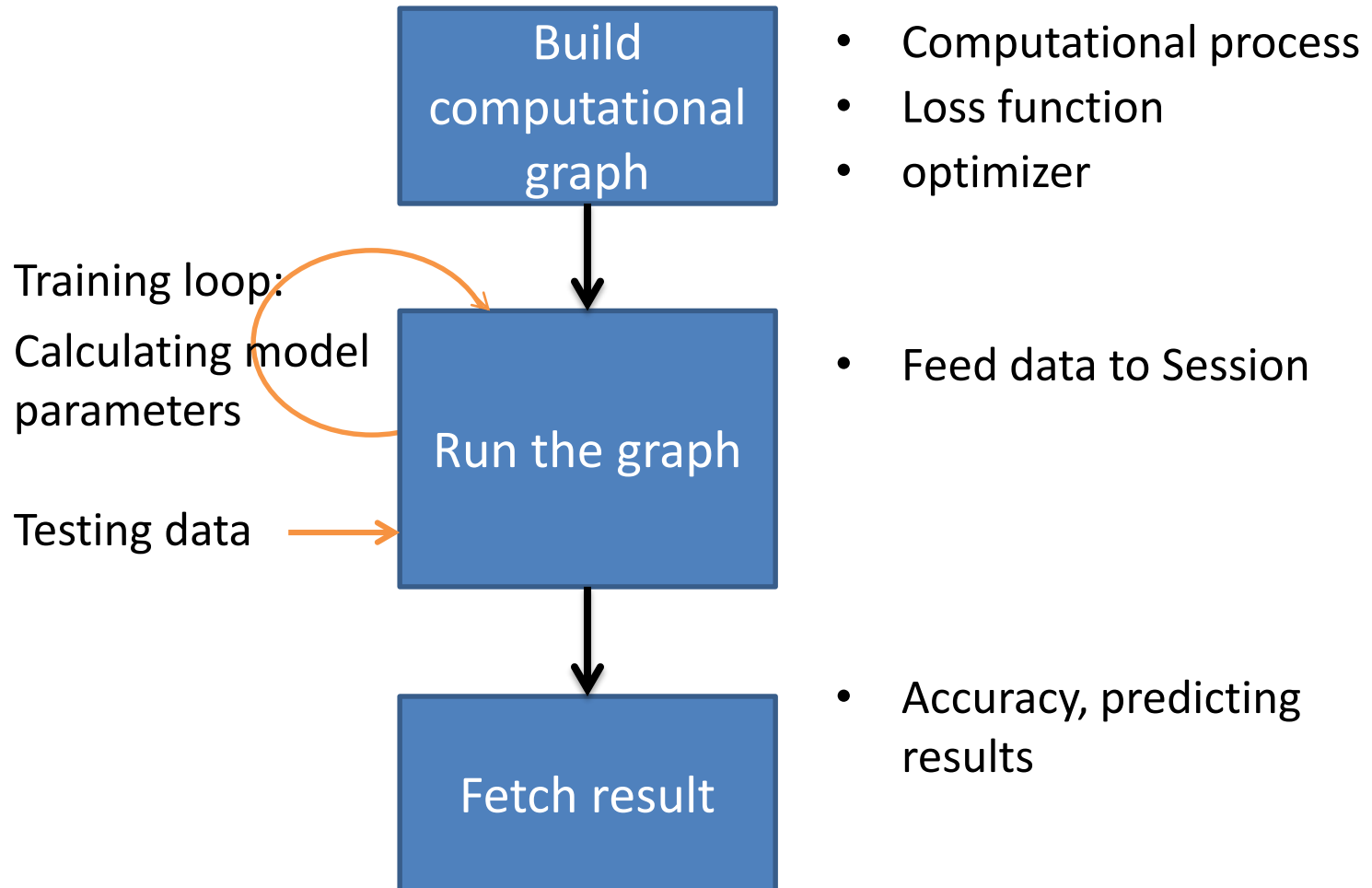
Load data and call
the library

Define your computational
graph, and run by Tensorflow

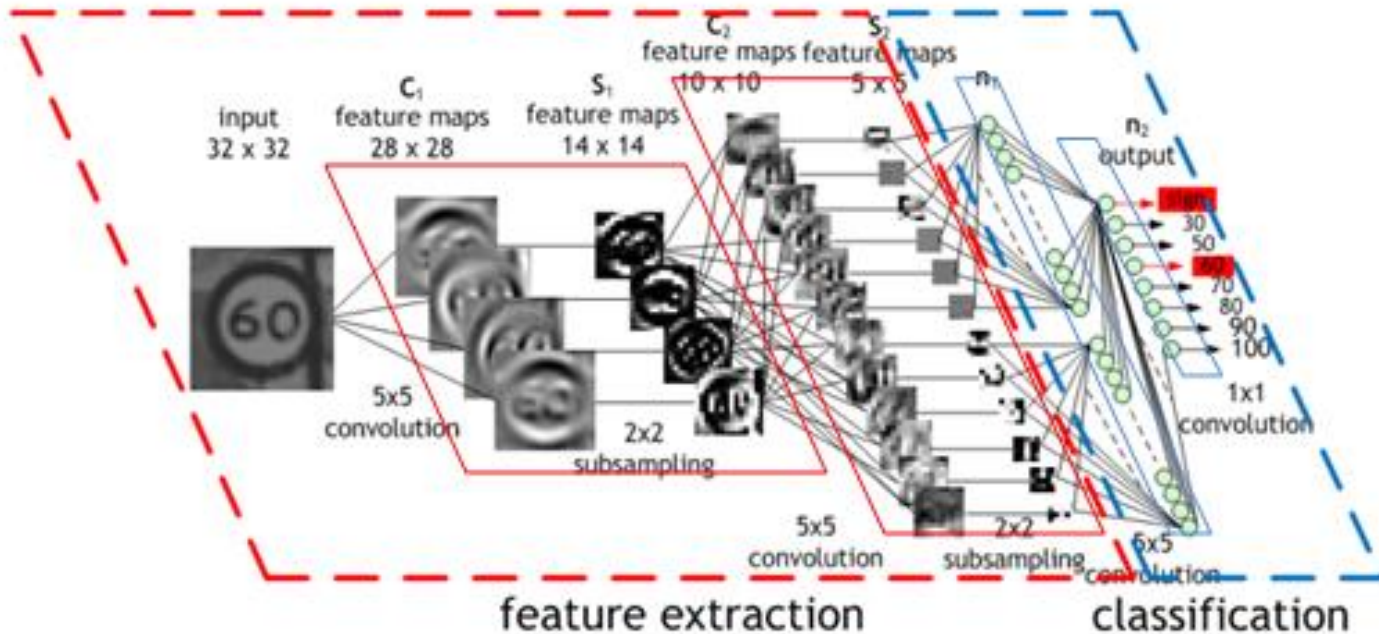
Write the process
about differentiation



How to code in TensorFlow



Convolution Neural Network



1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

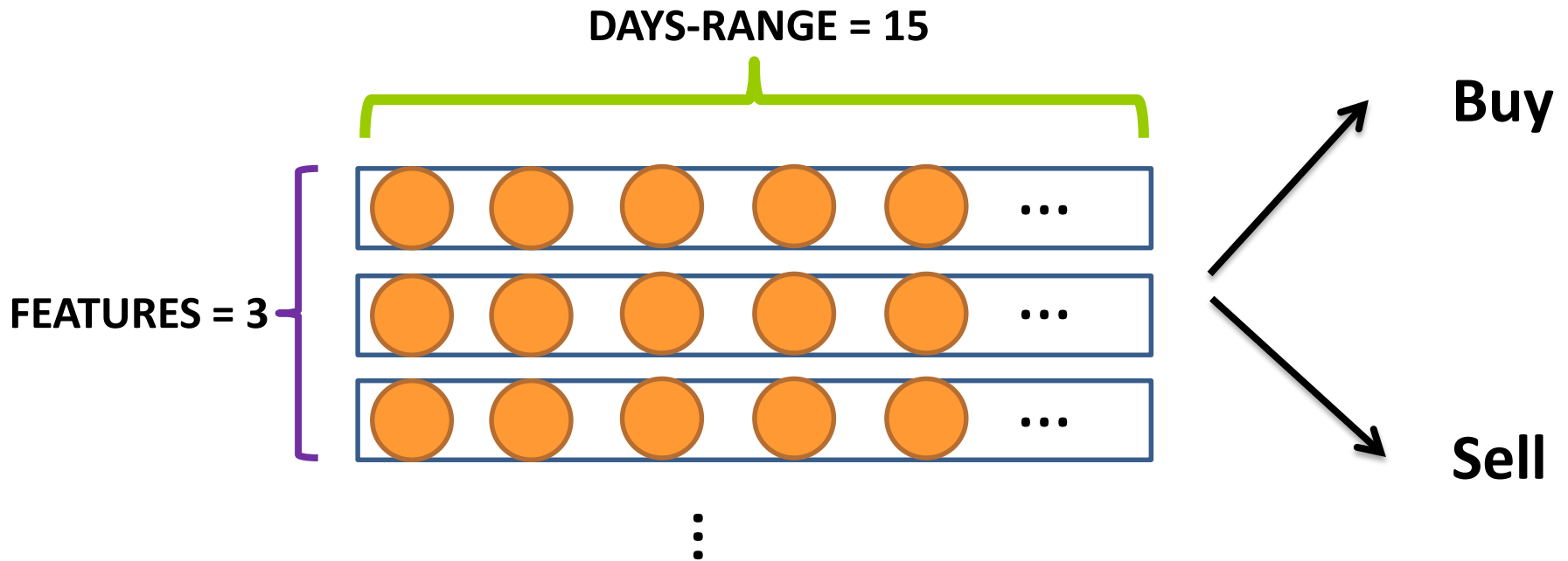
4		

Convolved
Feature

An image of a traffic sign is filtered by 4 5x5 convolutional kernels which create 4 feature maps, these feature maps are subsampled by max pooling. The next layer applies 10 5x5 convolutional kernels to these subsampled images and again we pool the feature maps. The final layer is a fully connected layer where all generated features are combined and used in the classifier (essentially logistic regression)

Application on Stock Market

- Task:
 - Making correct decisions based on historical data



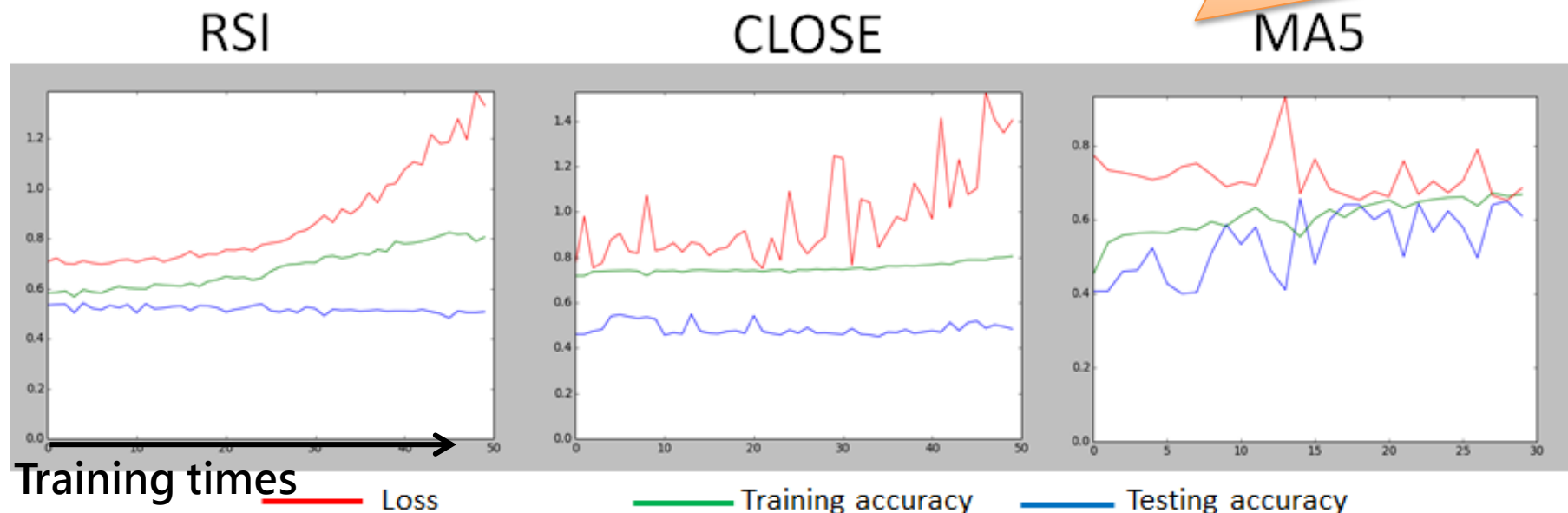
CNN code

https://github.com/kimber-chen/Tensorflow-for-stock-prediction/blob/master/2.CNN_tsc/CNN_Classifier.ipynb

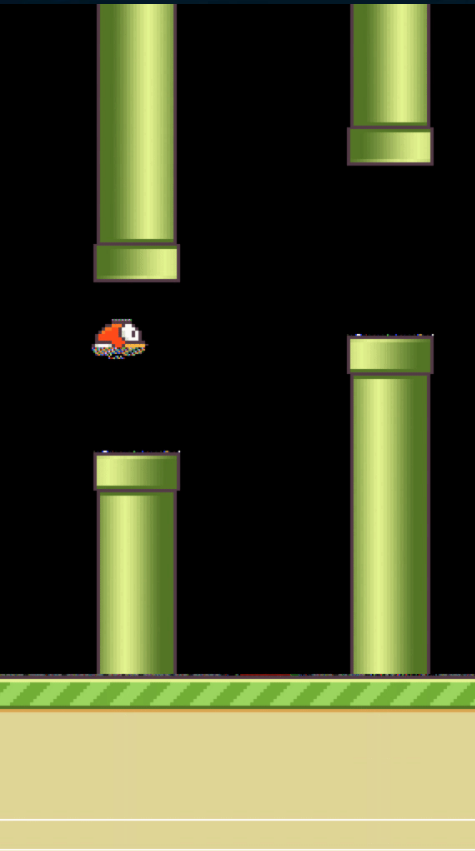
- Weight, bias and conv function
- How to build graph
- Training and Evaluating_accuracy
- sess.run()

Which Indicators

Predict 2330.tw(TSMC) close price movement of 5 days later. Accuracy between 58%~60%



- `Loss=tf.nn.softmax_cross_entropy_with_logits(h_fc2,y_)`
- The difference between prediction and real label

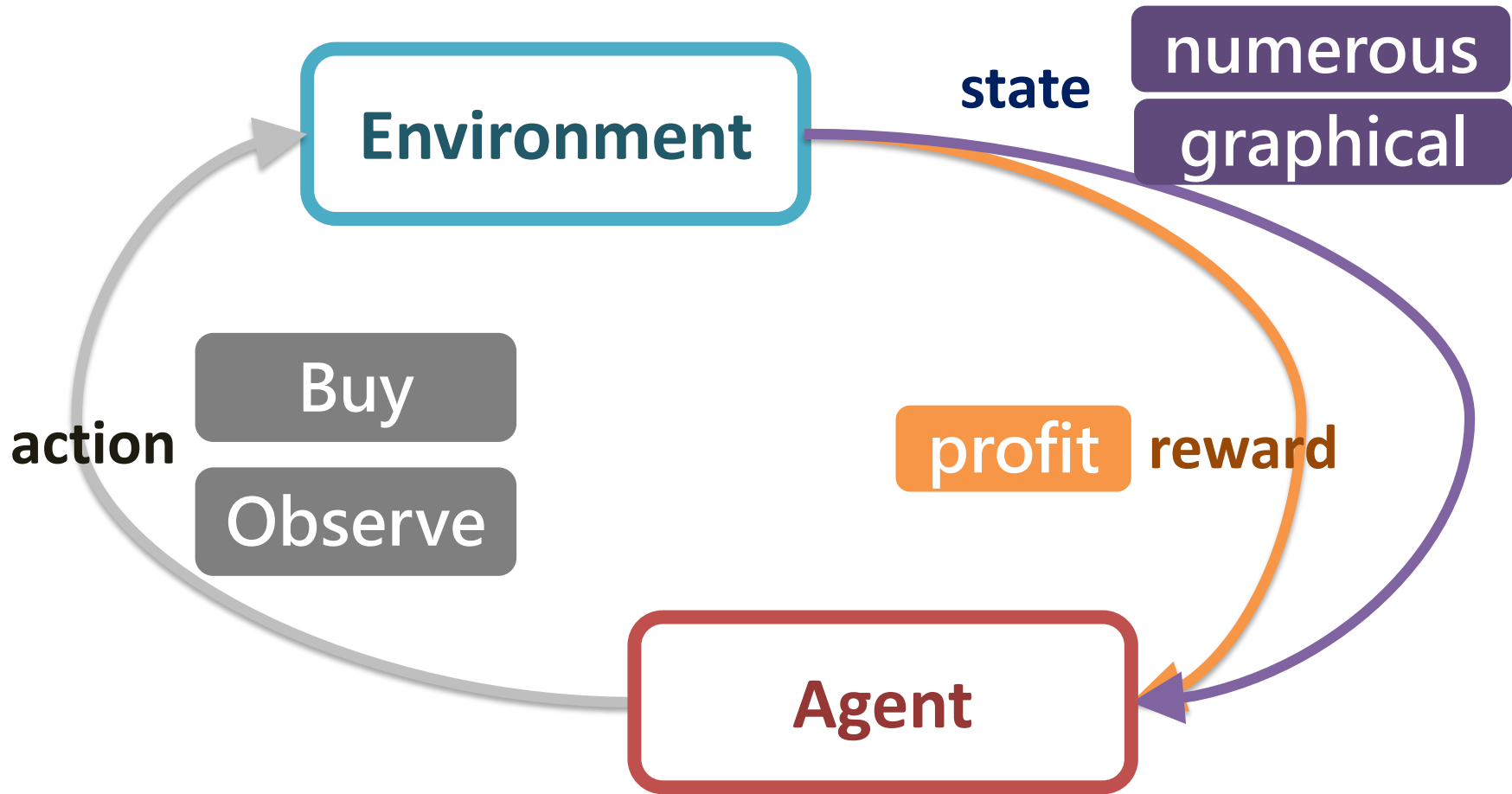


DQN

HUMAN-LEVEL CONTROL THROUGH **DEEP REINFORCEMENT LEARNING**

- Published in [Nature](#) on 26th February 2015
- Combines Deep Neural Networks with Reinforcement Learning
- Master a diverse range of Atari 2600 games with only the raw pixels and score as inputs.

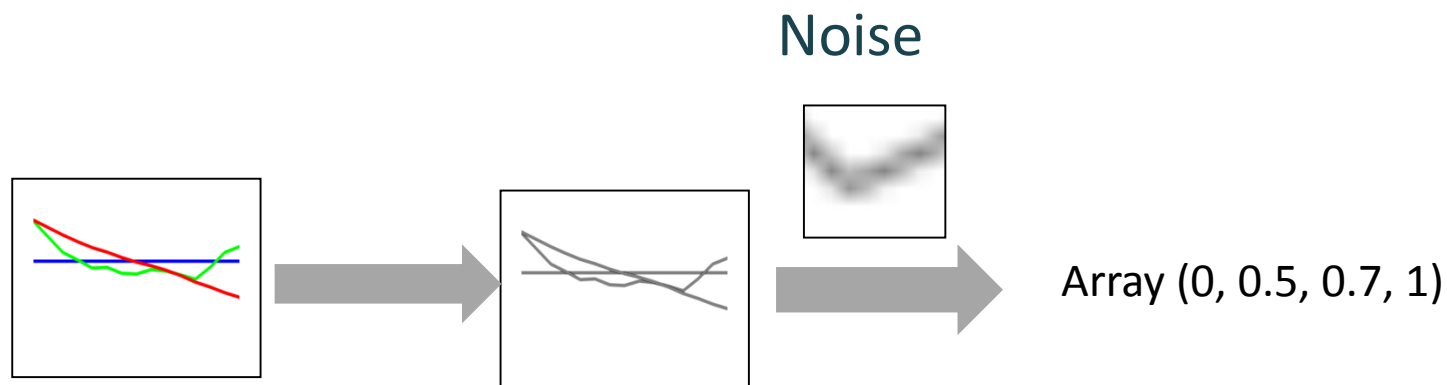
Deep Q Network



DQN code

https://github.com/kimber-chen/Tensorflow-for-stock-prediction/blob/master/1.DQN_CNN_image/Test%20model%20by%20yearline.ipynb

- Convert image to array :



- Trading strategy : sell while meet +10% profit or -5% loss.

Thank you