A Quant's Guide to TensorFlow for Prediction

Matthew Dixon[1]

Illinois Institute of Technology

June 7th 2017
@quiotaLLC

[1] Related work: M.F. Dixon, N. Polson V. Sokolov, Deep Learning for Spatio-Temporal Modeling: Dynamic Traffic Flows and High Frequency Trading, https://arxiv.org/abs/1705.09851

Deep Architectures in TensorFlow

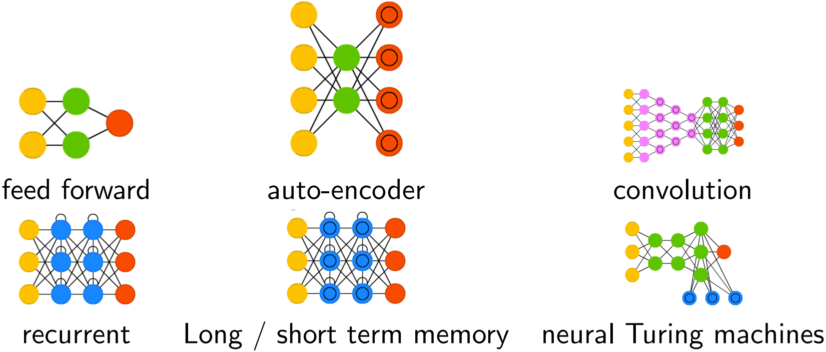
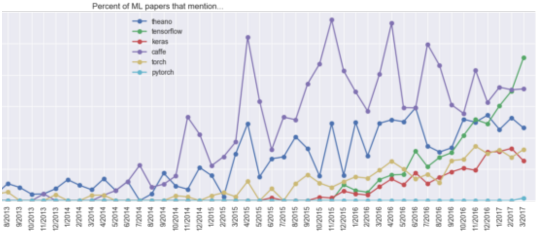


Figure: Most commonly used deep learning architectures for modeling. Source: `http://www.asimovinstitute.org/neural-network-zoo`

Growth of TensorFlow



% of papers	framework	has been around for (months)
9.1	tensorflow	16
7.1	caffe	37
4.6	theano	54
3.3	torch	37
2.5	keras	19
1.7	matconvnet	26
1.2	lasagne	23
0.5	chainer	16
0.3	mxnet	17
0.3	cntk	13
0.2	pytorch	1
0.1	deeplearning4j	14

Source : Andrej Karpathy‘s arXiv-sanity database

Tensorflow build for Intel Xeon Phi: https://github.com/tensorflow/tensorflow.git

Why Deep Learning in Finance?

- Capture complex, non-linear, relationships between variables to improve predictive power
- Regularization framework for automatic variable selection and prevention of over-fitting
- Not suitable for all problems in finance, but different architectures broaden the applicability

Example: Limit Order Book Updates

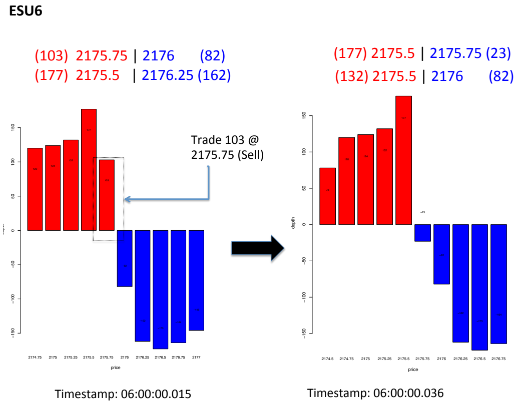
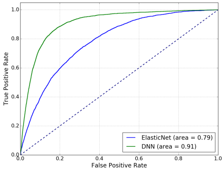
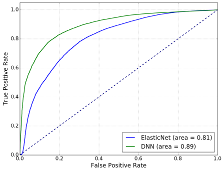


Figure: An exemplary sequence of limit order book updates before and after the arrival of a sell market order. The sell order is observed to match the supply of liquidity on offer at the best bid price and the entire book moves down by a tick. The sequence has been restricted to the top five levels of the order book.

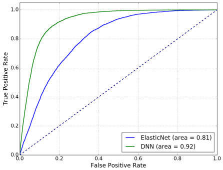
Receiver Operator Characteristics



(a) ROC curves of $\hat{Y} = 1$



(b) ROC curves of $\hat{Y} = 0$



(b) ROC curves of $\hat{Y} = -1$.

Table: The Receiver Operator Characteristic (ROC) curves of the deep learner and the elastic net method are shown for (left) downward, (middle) neutral, or (right) upward next price movement prediction.

Model Sensitivity

Hidden layers	DNN	EL-DNN
1	0.5057967179	0.5691572606
2	0.5340439642	0.5555855057
3	0.5724887077	0.578907192
4	0.5819864454	0.6474221372
5	0.5794411575	0.65784692

Approach 1: Statistical Inference based strategies

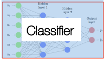
Strategy configuration c

Actions are independent of state

Minimize entropy

Features

X



Examples: RNNs, LSTMs

E.g.
-lagged prices,
depths

f(X)

Trading decision s(c)

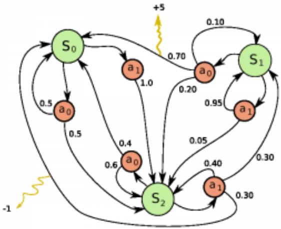
1 **buy**
0 hold
-1 sell

Use time series cross validation to assess performance

Pros: Simple and modular
Cons: Learning is separate from the reward

Approach 2: Markov Decision Process based strategies

Actions results in a change of state



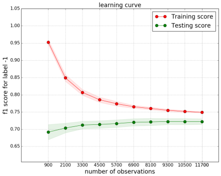
Define a set of states, actions and rewards

Goal: find an optimal set of decisions, Q-values, which maximize a utility function

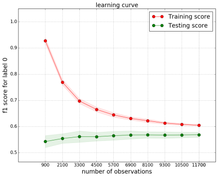
Examples: Q-Learning

Pros: Learn an optimal strategy based on optimal sequence of decisions
Cons: Complex and less programming support

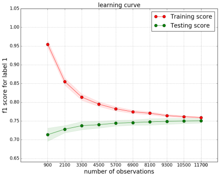
The Bias-Variance Tradeoff



(a) DNN F1-score of $\hat{Y} = 1$



(b) DNN F1-score of $\hat{Y} = 0$



(b) DNN F1-score of $\hat{Y} = -1$.

Table: The learning curves of the deep learner are used to assess the bias-variance tradeoff and are shown for (left) downward, (middle) neutral, or (right) upward price prediction. The variance is observed to reduce with an increased training set size and shows that the deep learning is not-overfitting. The bias on the test set is also observed to reduce with increased training set size.