Neural Networks

Anastasios Tefas
Assistant Professor
Department of Informatics
Aristotle University of Thessaloniki
Visiting Research Fellow in Tampere University of Technology

Presentation Outline

- Deep Learning in Big Financial Data Analysis
- What is Deep Learning
- Multilayer Perceptrons
- Deep Convolutional Neural Networks
- Recurrent Neural Networks
- Recent advances
- Conclusions

Recent Works on Deep Learning for Financial Data

- **Batres-Estrada, B. (2015). Deep learning for multivariate financial time series.** abstract
- **Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015, June). Deep learning for event-driven stock prediction. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (ICJAI) (pp. 2327-2333).** abstract
- **Dixon, M. F., Klabjan, D., & Bang, J. H. (2016). Classification-based Financial Markets Prediction using Deep Neural Networks. Available at SSRN 2756331.** abstract
- **Fehrer, R., & Feuerriegel, S. (2015). Improving Decision Analytics with Deep Learning: The Case of Financial Disclosures. arXiv preprint arXiv:1508.01993.** abstract
- **Heaton, J. B., Polson, N. G., & Witte, J. H. (2016). Deep Portfolio Theory. arXiv preprint arXiv:1605.07230.** abstract

Recent Works on Deep Learning for Financial Data

- **Rönnqvist, S., & Sarlin, P. (2016). Bank distress in the news: Describing events through deep learning. arXiv preprint arXiv:1603.05670.** abstract

- **Sharang, A., & Rao, C. (2015). Using machine learning for medium frequency derivative portfolio trading. arXiv preprint arXiv:1512.06228.** abstract

- **Sirignano, J. A. (2016). Deep Learning for Limit Order Books. arXiv preprint arXiv:1601.01987.** abstract

- **Takeuchi, L., Lee, Y. (2013). Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks.** abstract

- **Xiong, R., Nicholas, E. P., & Shen, Y. (2015). Deep Learning Stock Volatilities with Google Domestic Trends. arXiv preprint arXiv:1512.04916.** abstract

- **Zhu, C., Yin, J., & Li, Q. (2014). A stock decision support system based on DBNs. Journal of Computational Information Systems, 10(2), 883-893.** abstract

Case Studies

http://gregharris.info/a-survey-of-deep-learning-techniques-applied-to-trading/

Case Studies

http://gregharris.info/a-survey-of-deep-learning-techniques-applied-to-trading/

- **Limit Order Book Modeling**
  - **Sirignano (2016)** predicts changes in limit order books. He has developed a "spatial neural network" that can take advantage of local spatial structure, is more interpretable, and more computationally efficient than a standard neural network for this purpose. He models the joint distribution of the best bid and ask at the time of the next state change. Also, he models the joint distribution of the best bid and ask prices upon the change in either of them.

Case Studies

Price-based Classification Models

– **Dixon et al. (2016)** use a deep neural network to predict the sign of the price change over the next 5 minutes for 43 commodity and forex futures.
Architecture – Their input layer has 9,896 neurons for input features made up of lagged price differences and co-movements between contracts. There are 5 learned fully-connected layers. The first of the four hidden layers contains 1,000 neurons, and each subsequent layer tapers by 100 neurons. The output layer has 135 neurons (3 for each class {-1, 0, 1} times 43 contracts).

Case Study
Price-based Classification Models

- **Takeuchi and Lee (2013)** look to enhance the momentum effect by predicting which stocks will have higher or lower monthly returns than the median. **Architecture** – They use an auto-encoder composed of stacked RBMs to extract features from stock prices which they then pass to a feed-forward neural network classifier. Each RBM consists of one layer of visible units and one layer of hidden units connected by symmetric links. The first layer has 33 units for input features from one stock at a time. For every month t, the features include the 12 monthly returns for month t-2 through t-13 and the 20 daily returns approximately corresponding to month t. They normalize each of the return features by calculating the z-score relative to the cross-section of all stocks for each month or day. The number of hidden units in the final layer of the encoder is sharply reduced, forcing dimensionality reduction. The output layer has 2 units, corresponding to whether the stock ended up above or below the median return for the month. Final layer sizes are 33-40-4-50-2.

Case Study
Price-based Classification Models

- **Batres-Estrada (2015)** predicts which S&P 500 stocks will have above-median returns for each given day, and his work appears to be heavily influenced by Takeuchi and Lee (2013).
  Architecture – He uses a 3-layer DBN coupled to an MLP. He uses 400 neurons in each hidden layer, and he uses a sigmoid activation function. The output layer is a softmax layer with two output neurons for binary classification (above median or below). The DBN is composed of stacked RBMs, each trained sequentially.

Case Study

Price-based Classification Models

- **Sharang and Rao (2015)** use a DBN trained on technical indicators to trade a portfolio of US Treasury note futures.
- **Zhu et al. (2016)** make trade decisions using oscillation box theory based on DBNs. Oscillation box theory says that a stock price will oscillate within a certain range in a period of time. If the price moves outside the range, then it enters into a new box. The authors try to predict the boundaries of the box. Their trading strategy is to buy the stock when it breaks through the top boundary or sell it when it breaks through the bottom boundary.
  Architecture – They use a DBN made up of stacked RBMs and a final back-propagation layer.

Case study
Text-based Classification Models

- **Rönnqvist and Sarlin (2016)** predict bank distress using news articles. Specifically, they create a classifier to judge whether a given sentence indicates distress or tranquility.
  Architecture – They use two neural networks in this paper. The first is for semantic pre-training to reduce dimensionality. For this, they run a sliding window over text, taking a sequence of 5 words and learning to predict the next word. They use a feed-forward topology where a projection layer in the middle provides the semantic vectors once the connection weights have been learned. The second neural network is for classification. Instead of a million inputs (one for each word), they use 600 inputs from the learned semantic model. The first layer has 600 nodes, the middle layer has 50 rectified linear hidden nodes, and the output layer has 2 nodes (distress/tranquil).

Case study
Text-based Classification Models

- **Fehrer and Feuerriegel (2015)** train a model to predict German stock returns based on headlines. **Architecture** – They use a recursive autoencoder with an additional softmax layer in each autoencoder for estimating probabilities. They perform three-class prediction {-1, 0, 1} for the following day's return of the stock associated with the headline.

Case study
Text-based Classification Models

- **Ding et al. (2015)** use structured information extracted from headlines to predict daily S&P 500 moves. Headlines are processed with Open IE to obtain structured event representations (actor, action, object, time). A neural tensor network learns the semantic compositionality over event arguments by combining them multiplicatively instead of only implicitly, as with standard neural networks.
  Architecture – They combine short-term and long-term effects of events, using a CNN to perform semantic composition over the input event sequence. They use a max pooling layer on top of the convolutional layer, which makes the network retain only the most useful features produced by the convolutional layer.

Case Study
Volatility Prediction

- **Xiong et al. (2015)** predict the daily volatility of the S&P 500, as estimated from open, high, low, close prices.
  Architecture – They use a single LSTM hidden layer consisting of one LSTM block. For inputs they use daily S&P 500 returns and volatilities. They also include 25 domestic Google trends, covering sectors and major areas of the economy.

Case Study

Portfolio Optimization

- **Heaton et al. (2016)** attempt to create a portfolio that outperforms the biotech index IBB. They have the goal of tracking the index with few stocks and low validation error. They also try to beat the index by being anti-correlated during periods of large drawdowns. They don't directly model the covariance matrix, rather it is trained in the deep architecture fitting procedure, which allows for nonlinearities. **Architecture** – They use auto-encoding with regularization and ReLUs. Their auto-encoder has one hidden layer with 5 neurons.

Where Can I use Deep Neural Networks?

- To extract feature vectors from time-series, text, multidimentional time-series, multi-modal data (news feeds + stock prices + …)
- To classify the extracted feature vectors (buy, sell, stay).
- To predict/forecast an event after we have extract appropriate feature vectors (e.g., stock price, maximum gain/loss, etc)
- To correlate sequences finding how a series of events (e.g., in politics) affects the events in finance (e.g., stock trade).
- To detect/localize important/interesting behaviors in the time-series
- To make focused decisions based on attention models (where to look in the data in order to decide).
- To extract sentiments/concepts/trends from text (news, tweets, blogs, etc) media data (radio, tv, youtube, etc) and social networks (facebook, tweeter, etc)

Why should I use Deep Neural Networks

- General learning machines that learn based on (big) data.
- There are models that are specialized to specific tasks (CNNs, LSTMs, etc)
- They have proven that they perform and generalize much better that other techniques for many difficult tasks (vision, speech, text, translation and general AI)
- They are not yet fully applied to financial data (opportunity for novel and efficient publishable solutions)

Breakthrough

Deep Learning: machine learning algorithms based on learning multiple levels of representation / abstraction.

Amazing improvements in error rate in object recognition, object detection, speech recognition, and more recently, in natural language processing / understanding

Machine Learning, AI & No Free Lunch

- Four key ingredients for ML towards AI

  1. Lots & lots of data
  2. Very flexible models
  3. Enough computing power
  4. Powerful priors that can defeat the curse of dimensionality

Deep Learning: Automating Feature Discovery

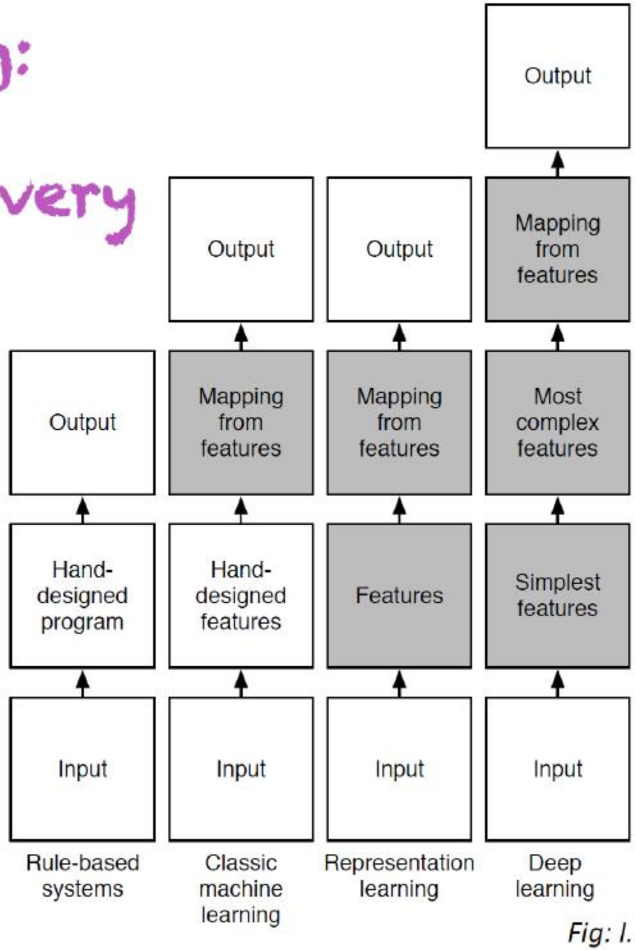


Fig: I. Goodfellow

Why does it work? No Free Lunch

- It only works because we are making some assumptions about the data generating distribution

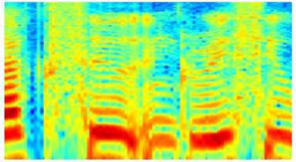
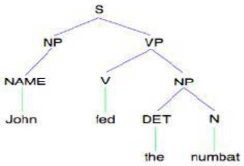
- Worse-case distributions still require exponential data

- But the world has structure and we can get an exponential gain by exploiting some of it


DEEP LEARNING EVERYWHERE
INTERNET & CLOUD
Image Classification
Speech Recognition
Language Translation
Language Processing
Sentiment Analysis
Recommendation
MEDICINE & BIOLOGY
Cancer Cell Detection
Diabetic Grading
Drug Discovery
MEDIA & ENTERTAINMENT
Video Captioning
Video Search
Real Time Translation
SECURITY & DEFENSE
Face Detection
Video Surveillance
Satellite Imagery
AUTONOMOUS MACHINES
Pedestrian Detection
Lane Tracking
Recognize Traffic Sign

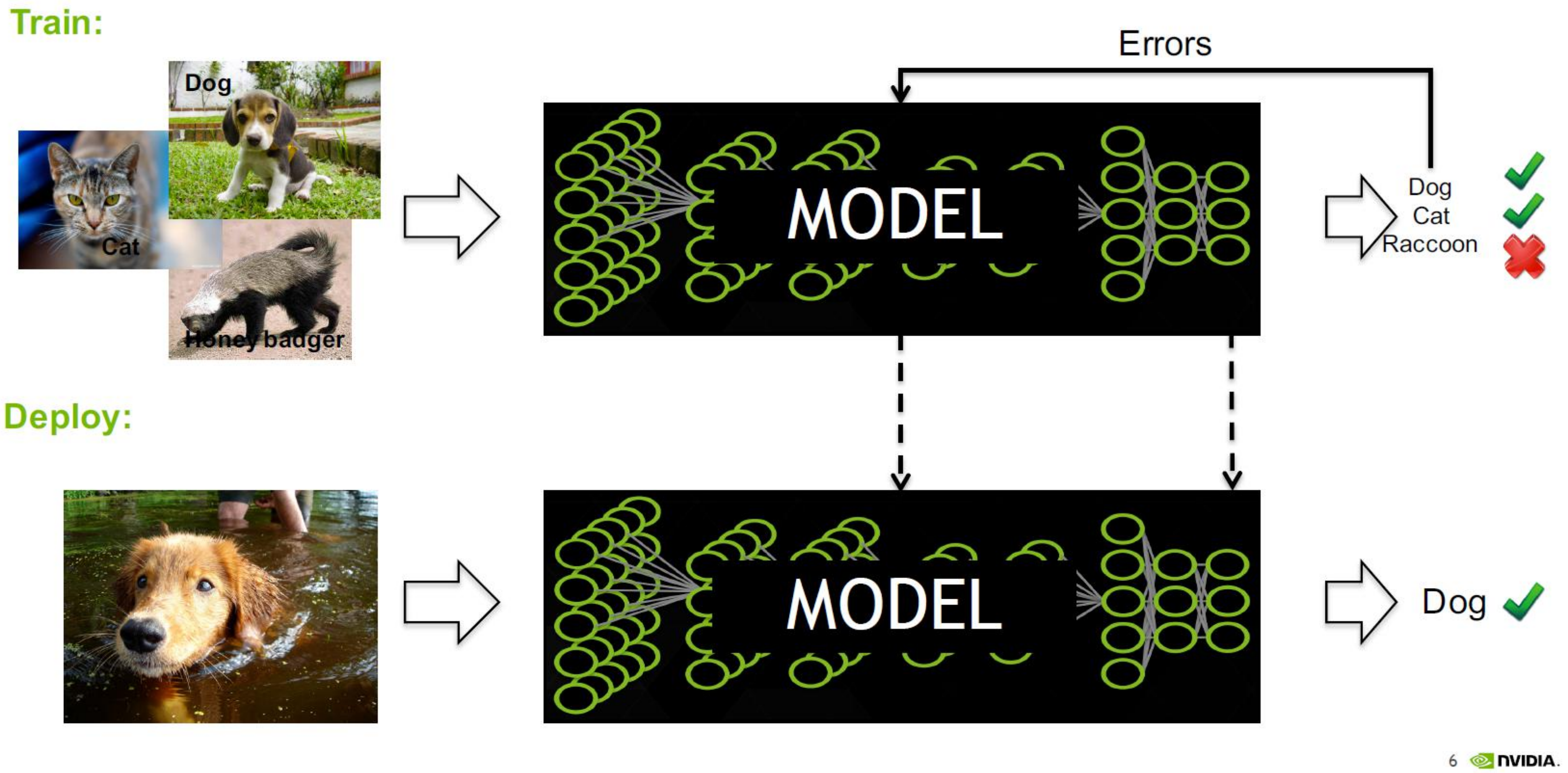
Traditional machine perception

Hand crafted feature extractors

Raw data	Feature extraction	Classifier/ detector	Result
		SVM, shallow neural net, …	
		HMM, shallow neural net, …	Speaker ID, speech transcription, …
		Clustering, HMM, LDA, LSA …	Topic classification, machine translation, sentiment analysis…

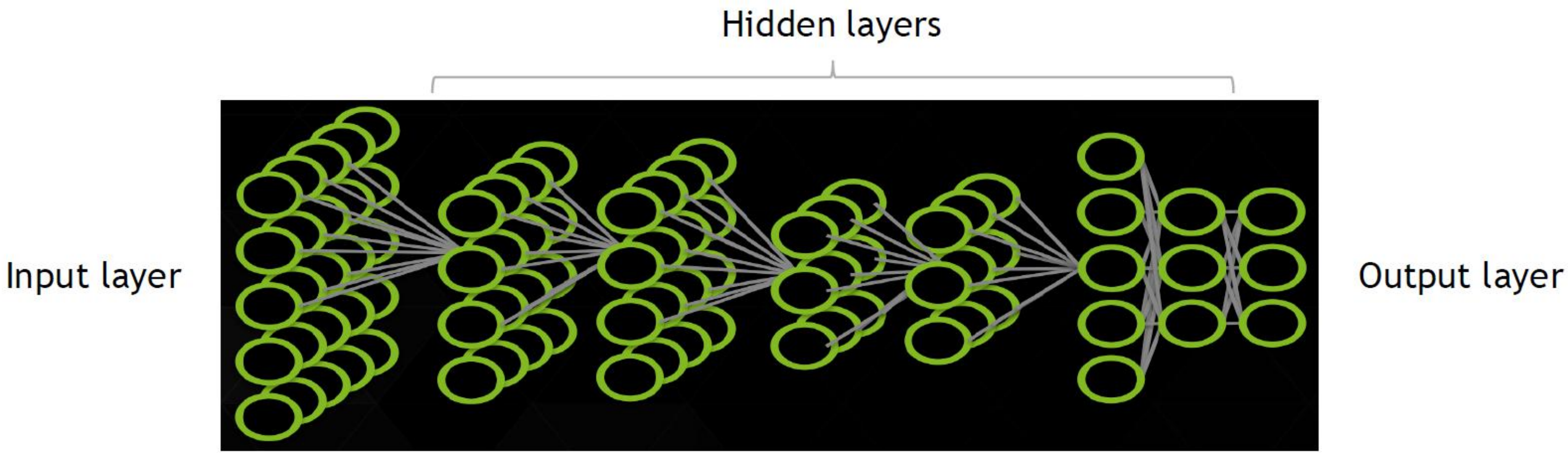
5 NVIDIA

Deep learning approach



Artificial neural network

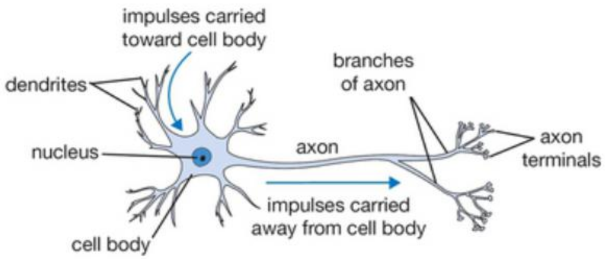
A collection of simple, trainable mathematical units that collectively learn complex functions



Given sufficient training data an artificial neural network can approximate very complex functions mapping raw data to output decisions

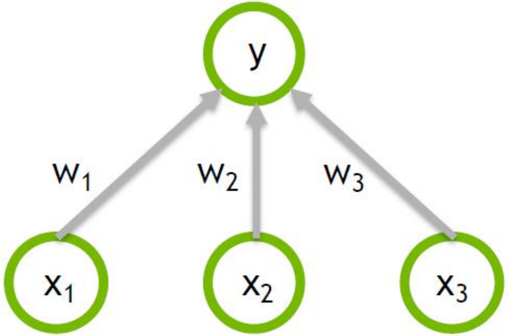
Artificial neurons

Biological neuron



From Stanford cs231n lecture notes

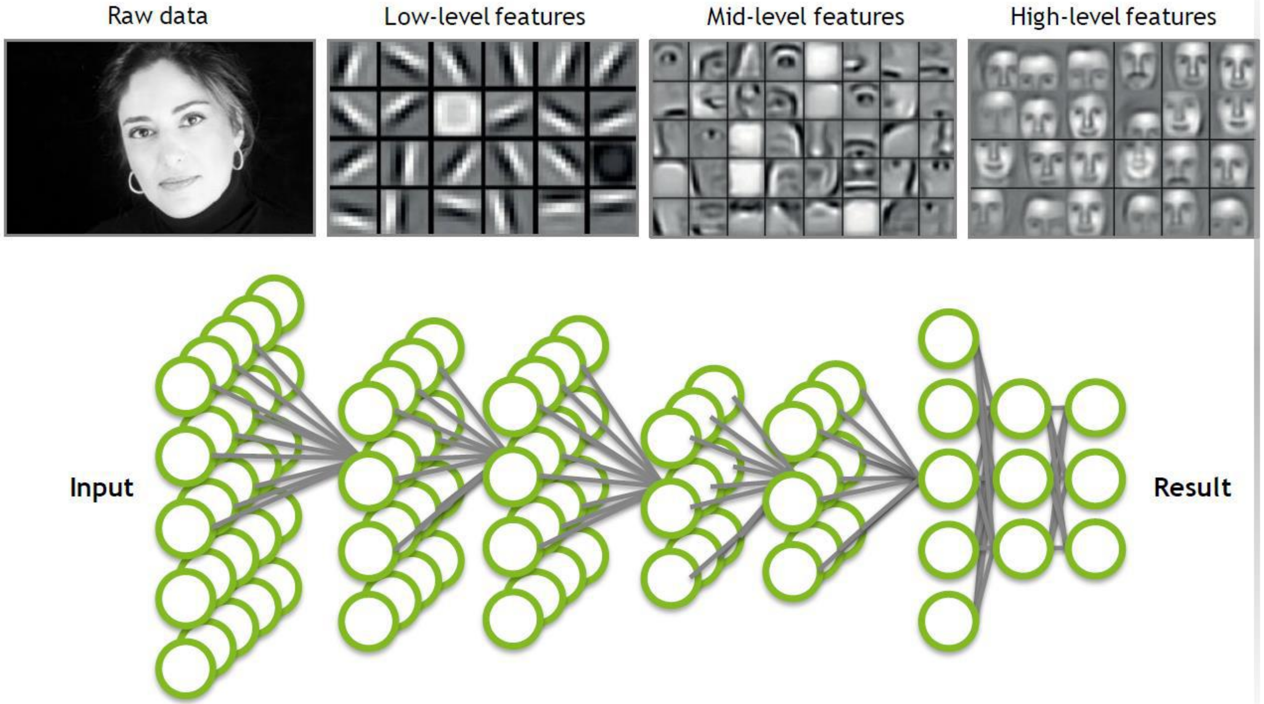
Artificial neuron



$$y = F(w_1 x_1 + w_2 x_2 + w_3 x_3)$$

$$F(x) = \max(0, x)$$

Deep neural network (dnn)



Application components:

Task objective
e.g. Identify face
Training data
10-100M images
Network architecture
~10 layers
1B parameters
Learning algorithm
~30 Exaflops
~30 GPU days

9 NVIDIA.

Deep learning benefits

- Robust
  - No need to design the features ahead of time – features are automatically learned to be optimal for the task at hand
  - Robustness to natural variations in the data is automatically learned
- Generalizable
  - The same neural net approach can be used for many different applications and data types
- Scalable
  - Performance improves with more data, method is massively parallelizable

Baidu Deep Speech 2

End-to-end Deep Learning for English and Mandarin Speech Recognition

English and Mandarin speech recognition

Transition from English to Mandarin made simpler by end-to-end DL

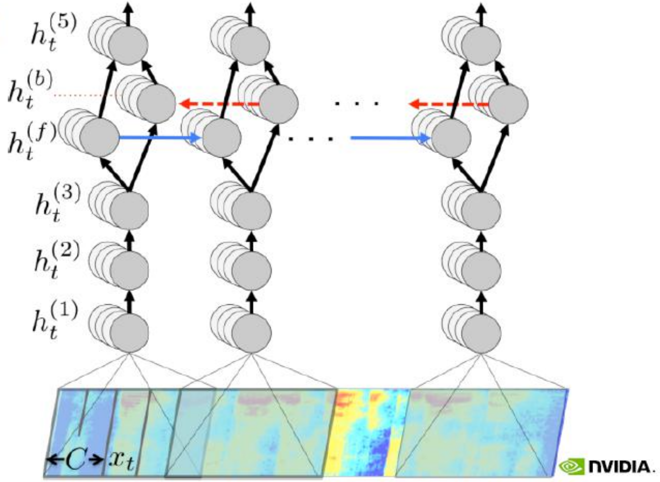
- No feature engineering or Mandarin-specifics required

More accurate than humans

- Error rate 3.7% vs. 4% for human tests

http://svail.github.io/mandarin/

http://arxiv.org/abs/1512.02595



AlphaGo

First Computer Program to Beat a Human Go Professional

Training DNNs: 3 weeks, 340 million training steps on 50 GPUs

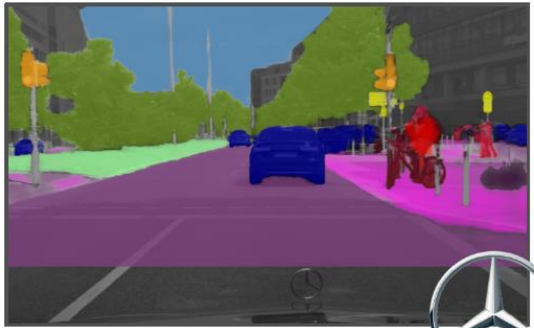
Play: Asynchronous multi-threaded search

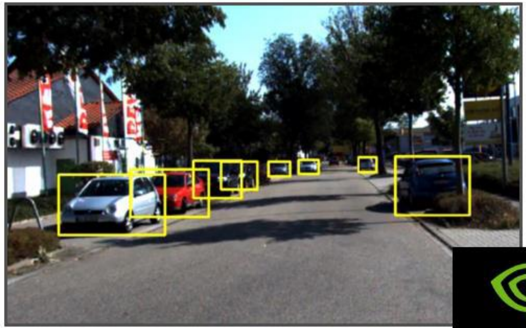
- Simulations on CPUs, policy and value DNNs in parallel on GPUs
- Single machine: 40 search threads, 48 CPUs, and 8 GPUs
- Distributed version: 40 search threads, 1202 CPUs and 176 GPUs

Outcome: Beat both European and World Go champions in best of 5 matches

http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html
http://deepmind.com/alpha-go.html

Deep Learning for Autonomous vehicles

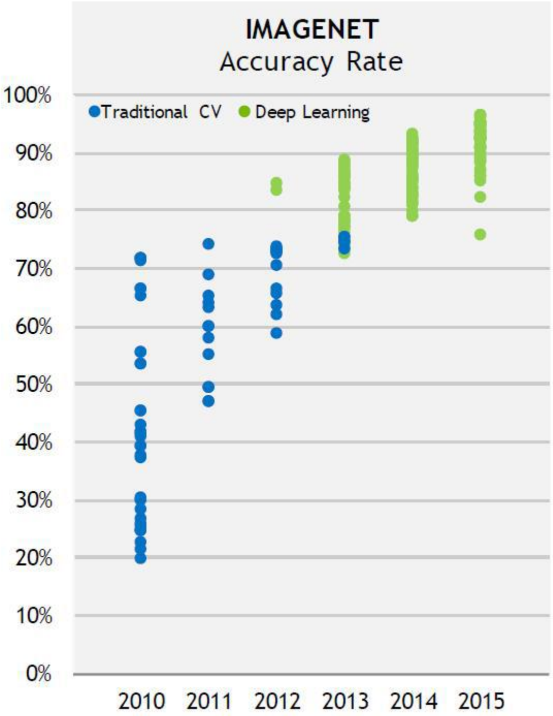


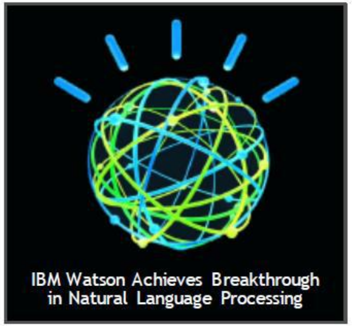


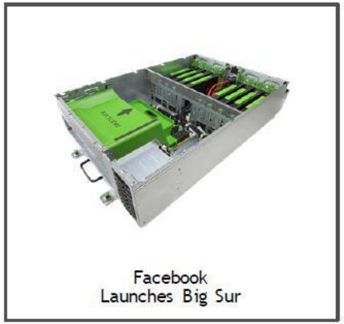
Deep Learning Synthesis

Texture synthesis and transfer using CNNs. Timo Aila et al., NVIDIA Research

THE AI RACE IS ON




IBM Watson Achieves Breakthrough in Natural Language Processing


Facebook Launches Big Sur

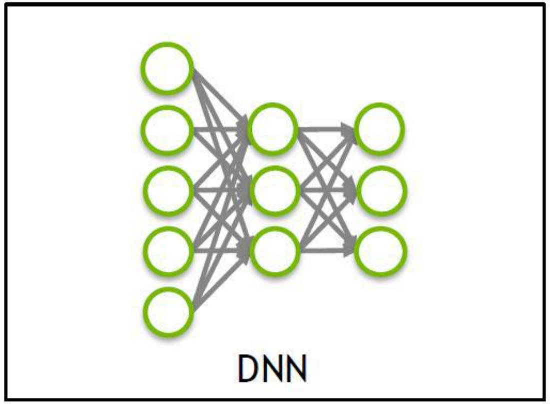
Baidu Deep Speech 2 Beats Humans

Google Launches TensorFlow


Toyota Invests $1B in AI Labs

Microsoft & U. Science & Tech, China Beat Humans on IQ

The Big Bang in Machine Learning



"Google's AI engine also reflects how the world of computer hardware is changing. (It) depends on machines equipped with GPUs... And it depends on these chips more than the larger tech universe realizes."

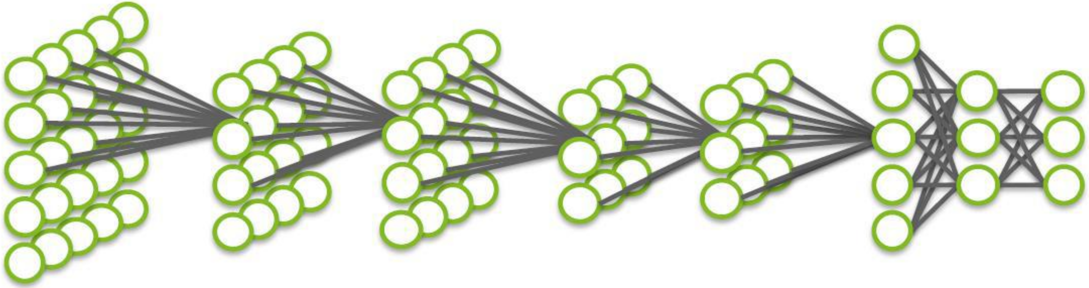
WIRED

Three Kinds of Networks

DNN - all fully connected layers

CNN - some convolutional layers

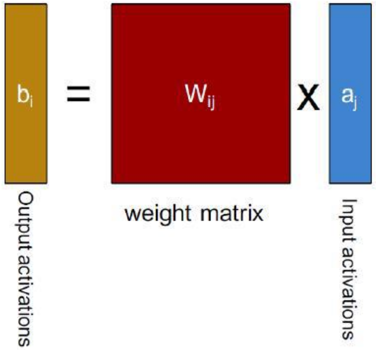
RNN - recurrent neural network, LSTM

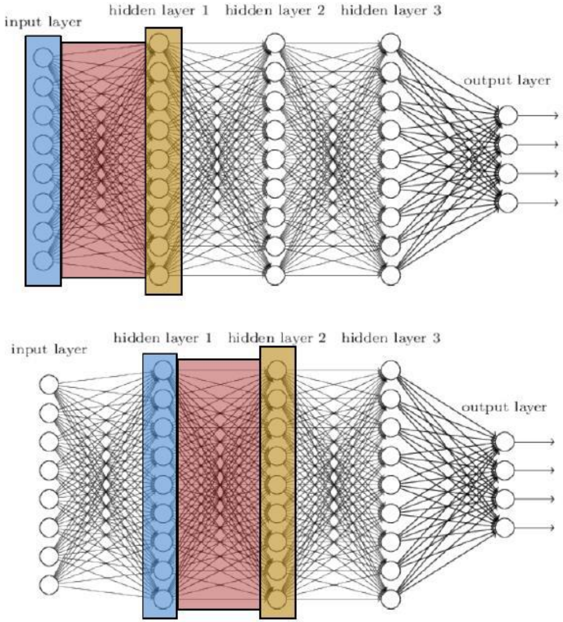


19 NVIDIA.

DNN

Key operation is dense M x V





Repeat for each layer

Backpropagation uses dense matrix-matrix multiply starting from softmax scores

20 NVIDIA.

Used Material and Useful resources

- http://gregharris.info/a-survey-of-deep-learning-techniques-applied-to-trading
- http://vision.stanford.edu/teaching/cs231n/syllabus.html
- http://www.deeplearningbook.org/
- http://www.iro.umontreal.ca/~bengioy/yoshua_en/talks.html
- https://developer.nvidia.com/deep-learning-courses
- http://arxiv.org/abs/1602.06561
- http://www.slideshare.net/SebastienJehan/deeplearning-in-finance
- http://deeplearning.net/tutorial/

Summary and Conclusions

- Deep learning is one of the most powerful machine learning tool you have available nowadays.
- You can find several deep learning methods that are well suited for your problem.
- Not much work in financial data analysis and thus good opportunity for novel work and publications

Thank You

Questions