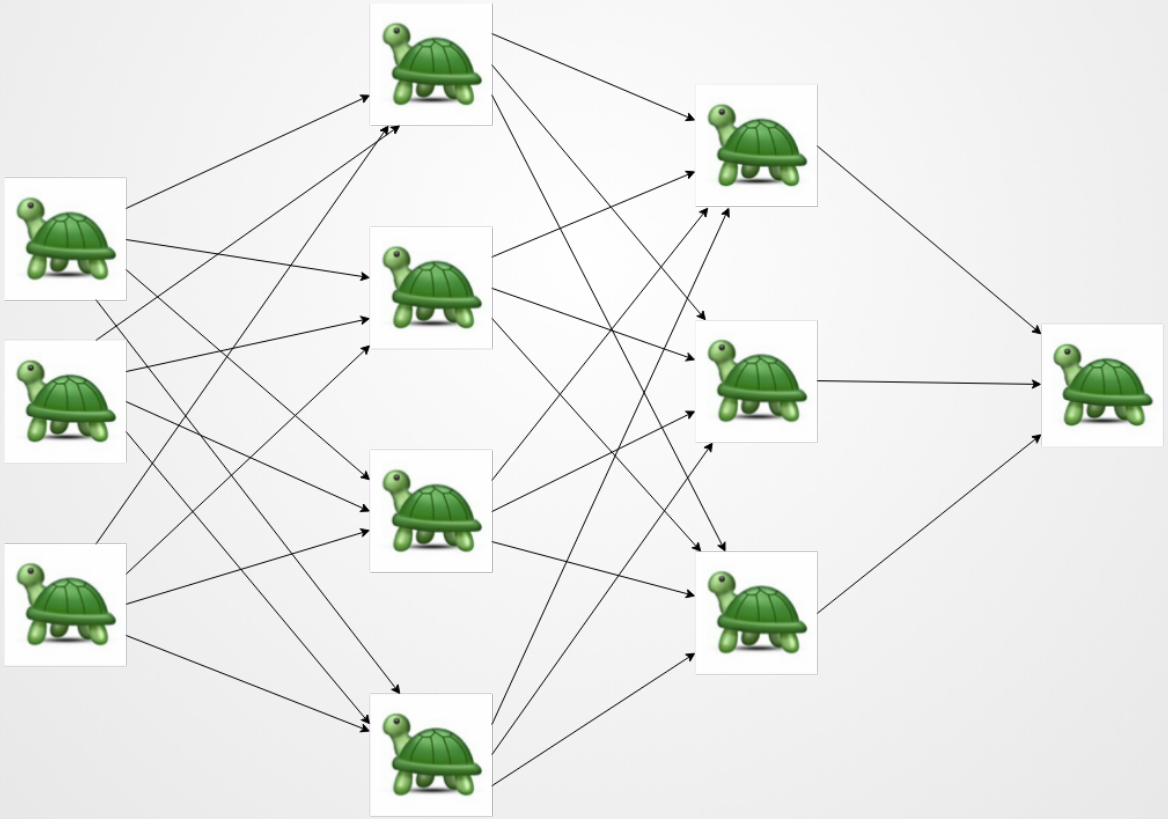
# Practical NLP Applications of Deep Learning

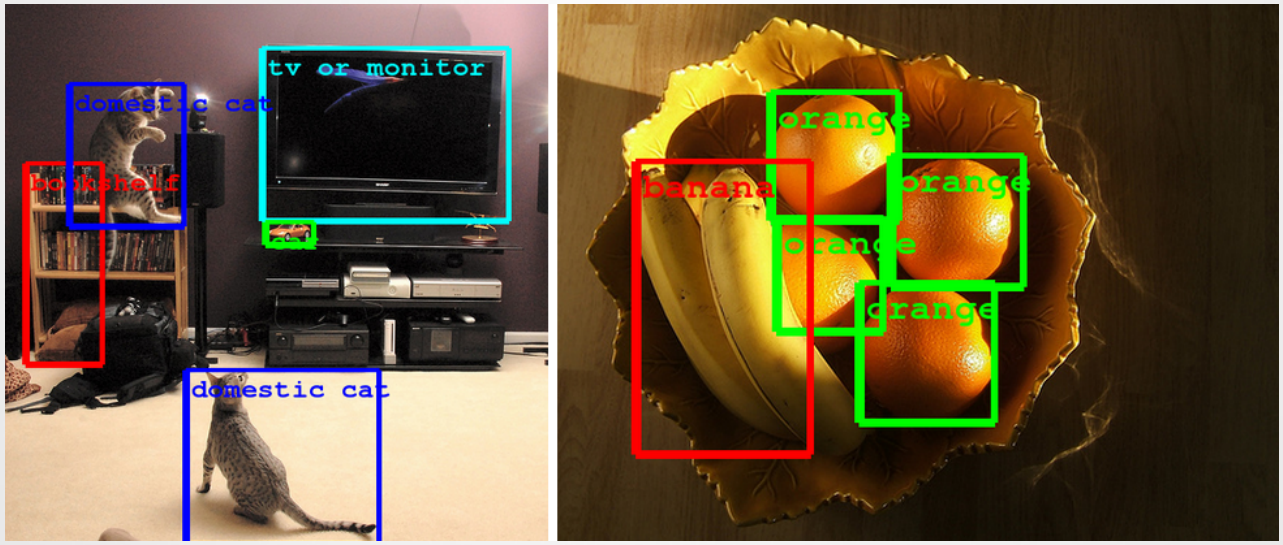

# What is Deep Learning?

# Turtles all the way down...



So what's the big deal?

# MASSIVE improvements in Computer Vision

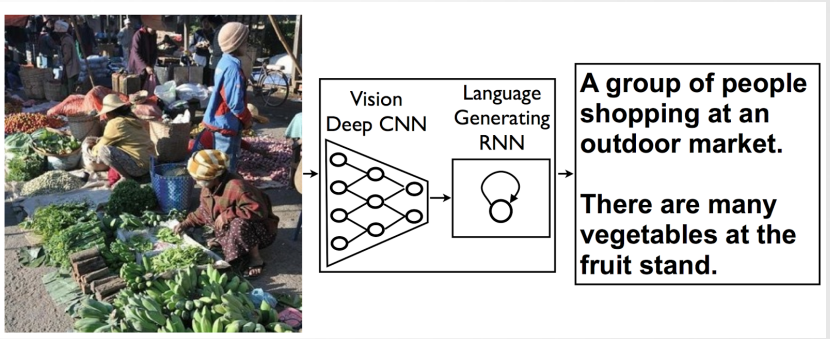


# Speech Recognition

- Baidu (with Andrew Ng as their chief) has built a state-of-the-art speech recognition system with Deep Learning
- Their dataset: 7000 hours of conversation couple with background noise synthesis for a total of 100,000 hours
- They processed this through a massive GPU cluster

# Cross Domain Representations

- What if you wanted to take an image and generate a description of it?
- The beauty of representation learning is it's ability to be distributed across tasks
- This is the real power of Neural Networks



But Samiur, what about NLP?

# Deep Learning NLP

- Distributed word representations
- Dependency Parsing
- Sentiment Analysis
- And many others ...

# Word Representations

## Standard

- Bag of Words
  - A one-hot encoding
  - 20k to 50k dimensions
  - Can be improved by factoring in document frequency

motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]

## Word embedding

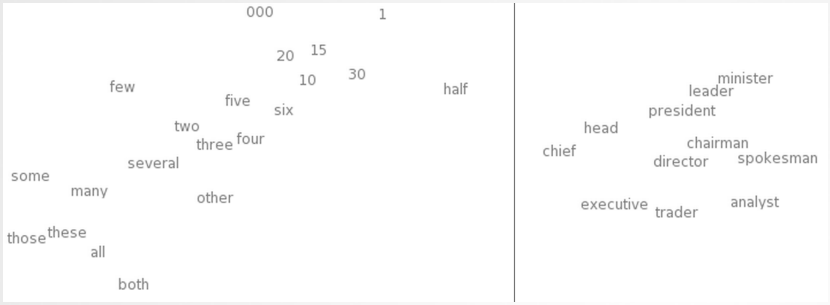
- Neural Word embeddings
  - Uses a vector space that attempts to predict a word given a context window
  - 200-400 dimensions

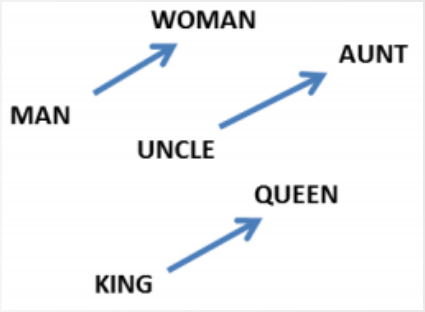
motel [0.06, -0.01, 0.13, 0.07, -0.06, -0.04, 0, -0.04]

hotel [0.07, -0.03, 0.07, 0.06, -0.06, -0.03, 0.01, -0.05]

Word embeddings make semantic similarity and synonyms possible

# Word embeddings have cool properties:



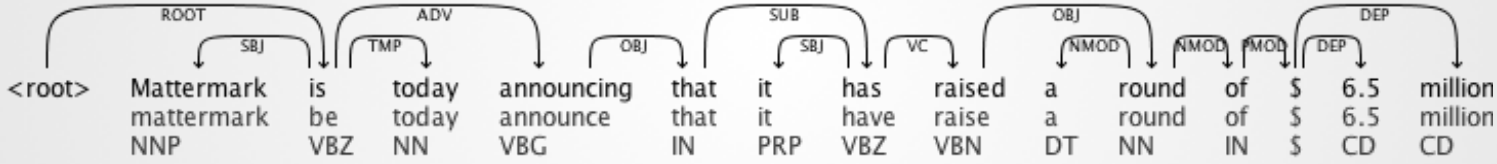




| Relationship | Example 1 | Example 2 | Example 3 |
|---|---|---|---|
| France - Paris | Italy: Rome | Japan: Tokyo | Florida: Tallahassee |
| big - bigger | small: larger | cold: colder | quick: quicker |
| Miami - Florida | Baltimore: Maryland | Dallas: Texas | Kona: Hawaii |
| Einstein - scientist | Messi: midfielder | Mozart: violinist | Picasso: painter |
| Sarkozy - France | Berlusconi: Italy | Merkel: Germany | Koizumi: Japan |
| copper - Cu | zinc: Zn | gold: Au | uranium: plutonium |
| Berlusconi - Silvio | Sarkozy: Nicolas | Putin: Medvedev | Obama: Barack |
| Microsoft - Windows | Google: Android | IBM: Linux | Apple: iPhone |
| Microsoft - Ballmer | Google: Yahoo | IBM: McNealy | Apple: Jobs |
| Japan - sushi | Germany: bratwurst | France: tapas | USA: pizza |

# Dependency Parsing

Converting sentences to a dependency based grammar

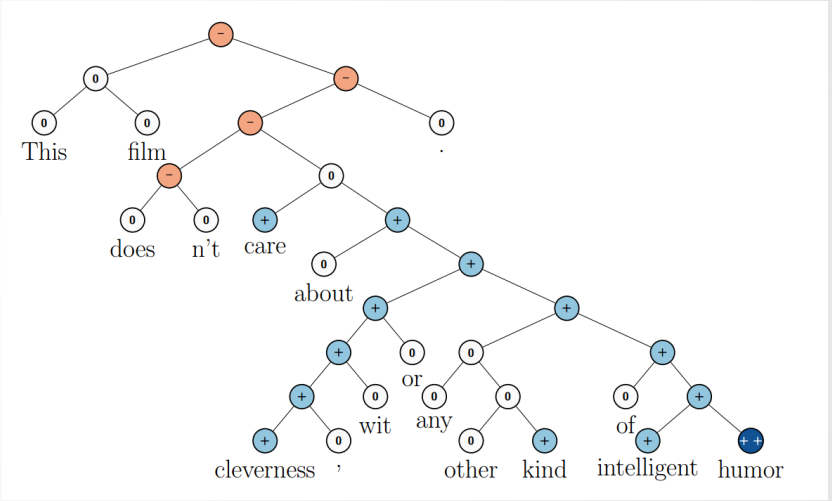
Simplifying this to the verbs and it's agents is called Semantic Role Labeling




| | Mattermark | is | today | announcing | that | it | has | raised | a | round | of | $ | 6.5 | million |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| announce.01 | A0 | | | | A1 | | | | | | | | | |
| raise.01 | | | | | | A0 | | | A1 | | | | | |
| round.01 | | | | | | | | | | | A1 | | | |

# Sentiment Analysis

- Recursive Neural Networks
  - Can model tree structures very well
  - This makes it great for other NLP tasks too (such as parsing)



Get to the applications part already!

# Tools

- Python
  - Theano/PyLearn2
  - Pybrain
  - Gensim (for word2vec)
  - nolearn (uses scikit-learn)
- Java/Clojure/Scala
  - DeepLearning4j
  - neuralnetworks
- APIs
  - Meta Mind

# Problem: Funding Sentence Classifier

Build a binary classifier that is able to take any sentence from a news article and tell if it's about funding or not.

eg. "Mattermark is today announcing that it has raised a round of $6.5 million"

# Word Vectors

- Used Gensim's Word2Vec implementation to train unsupervised word vectors on the UMBC Webbase Corpus (~100M documents, ~48GB of text)
- Then, iterated 20 times on text in news articles in the tech news domain (~3M documents, ~900MB of text)

# Sentence Vectors

- How can you compose word vectors to make sentence vectors?
  - Use paragraph vector model proposed by Quoc Le
  - Feed into an RNN constructed by a dependency tree of the sentence proposed by Richard Socher
  - Convolution Neural Networks proposed by Yoon Kim
  - Use heuristic function to combine the string of word vectors

# What did we try?

- TF-IDF + Naive Bayes
- Word2Vec + Composition Methods
- Word2Vec + TF-IDF + Composition Methods
- Word2Vec + TF-IDF + Semantic Role Labeling (SRL) + Composition Methods

# Composition Methods

| | Mattermark | is | today | announcing | that | it | has | raised | a | round | of | $ | 6.5 | million |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| announce.01 | A0 | | | | A1 | | | | | | | | | |
| raise.01 | | | | | | A0 | | | A1 | | | | | |
| round.01 | | | | | | | | | | | A1 | | | |

Where $w_i$ represents the i'th word vector,

$w_v$ the word vector for the verb, and a0 and a1 are agents

1. Additive

$$\sum_{i=1}^{N} \vec{w}_{ij}$$

2. Multiplicative

$$\prod_{i=1}^{N} \vec{w}_{ij}$$

3. Circular Convolution

$$\sum_{i=1}^{N} \vec{w}_{i-1} \circledast \vec{w}_i$$

4. Circular Convolution /Additive

$$\vec{w}_v \circledast \sum_{i=1}^{N_{a0}} \vec{w}_{ij}^{a0} + \sum_{i=1}^{N_{A1}} \vec{w}_{ij}^{a1} \circledast \vec{w}_v + \sum_{i=1}^{N_{rem}} \vec{w}_{IJ}$$

5. Tensor Product/Additive

$$\sum_{i=1}^{N} \left[ \vec{w}_v \odot \sum_{i=1}^{N_{a0}} \vec{w}_{ij}^{a0} + \sum_{i=1}^{N_{a1}} \vec{w}_{ij}^{a1} \odot \vec{w}_v \right]_{ij} + \sum_{i=1}^{N_{rem}} \vec{w}_{ij}$$

6. Circular Convolution/Dot Product

$$\left[ \vec{w}_v \circledast \sum_{i=1}^{N_{a0}} \vec{w}_{ij}^{a0} \right] \cdot \left[ \sum_{i=1}^{N_{a1}} \vec{w}_{ij}^{a1} \circledast \vec{w}_v \right] + \sum_{i=1}^{N_{rem}} \vec{w}_{ij}$$

# What worked?

- Word2Vec + TFIDF + SRL + Circular Convolution
  - The first method with simple TFIDF/Naive Bayes performed extremely poorly because of it's large dimensionality
  - Combining TFIDF with Word2Vec provided a small, but noticeable improvement
  - Adding SRL and a more sophisticated composition method increased performance by almost 5%

# What else is possible?

# Document Vectors

- Can we apply this method to generate general purpose document vectors?
  - We are currently using LDA (a topic analysis method) or simple TFIDF  to create document vectors
  - How will this method compare to the already proposed paragraph vector method by Quoc Le?

# Document Search

- Can we associate these document vectors with much smaller query strings?
  - eg. Search for artificial intelligence against our companies and get better results than keyword search

# Mattermark is Hiring!

Contact me at:

- @samiur1204
- samiur@mattermark.com