QF 624: Machine Learning for Financial Applications

Automated Pattern Recognition, HMM, NLP

Master of Science in Quantitative Finance
Lee Kong Chian School of Business

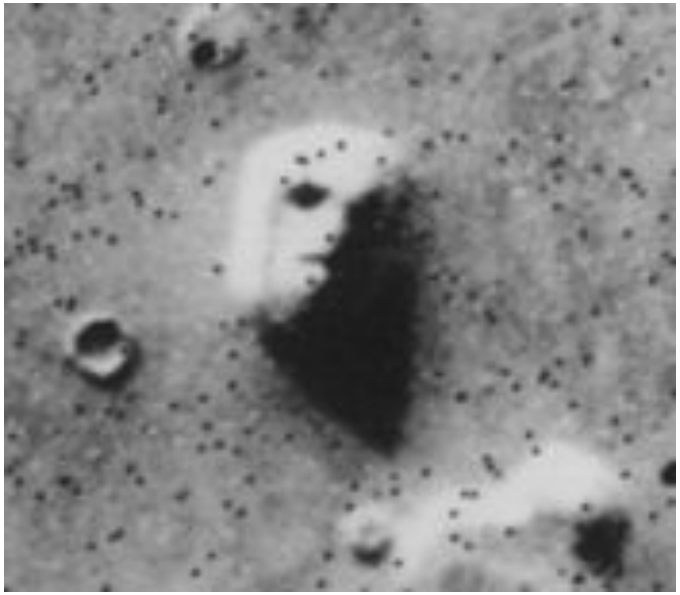
Saurabh Singal

July 2018

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Pareidolia

Pareidolia – seeing faces in a cloud (face on Mars)



Wikipedia defines **Pareidolia** *as "a psychological phenomenon in which the mind responds to a stimulus, usually an image or a sound, by perceiving a familiar pattern where none exists."*

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Example 13: Automated Pattern Recognition

Osler, Carol L. & P. H. Kevin Chang

Head and Shoulders: Not Just a Flaky Pattern

No 4, Staff Reports from Federal Reserve Bank of New York

Abstract:

"This paper evaluates rigorously the predictive power of the head-and-shoulders pattern as applied to daily exchange rates. Though such visual, nonlinear chart patterns are applied frequently by technical analysts, our paper is one of the first to evaluate the predictive power of such patterns. We apply a trading rule based on the head-and-shoulders pattern to daily exchange rates of major currencies versus the dollar during the floating rate period (from March 1973 to June 1994). "

SMU
SINGAPORE MANAGEMENT UNIVERSITY

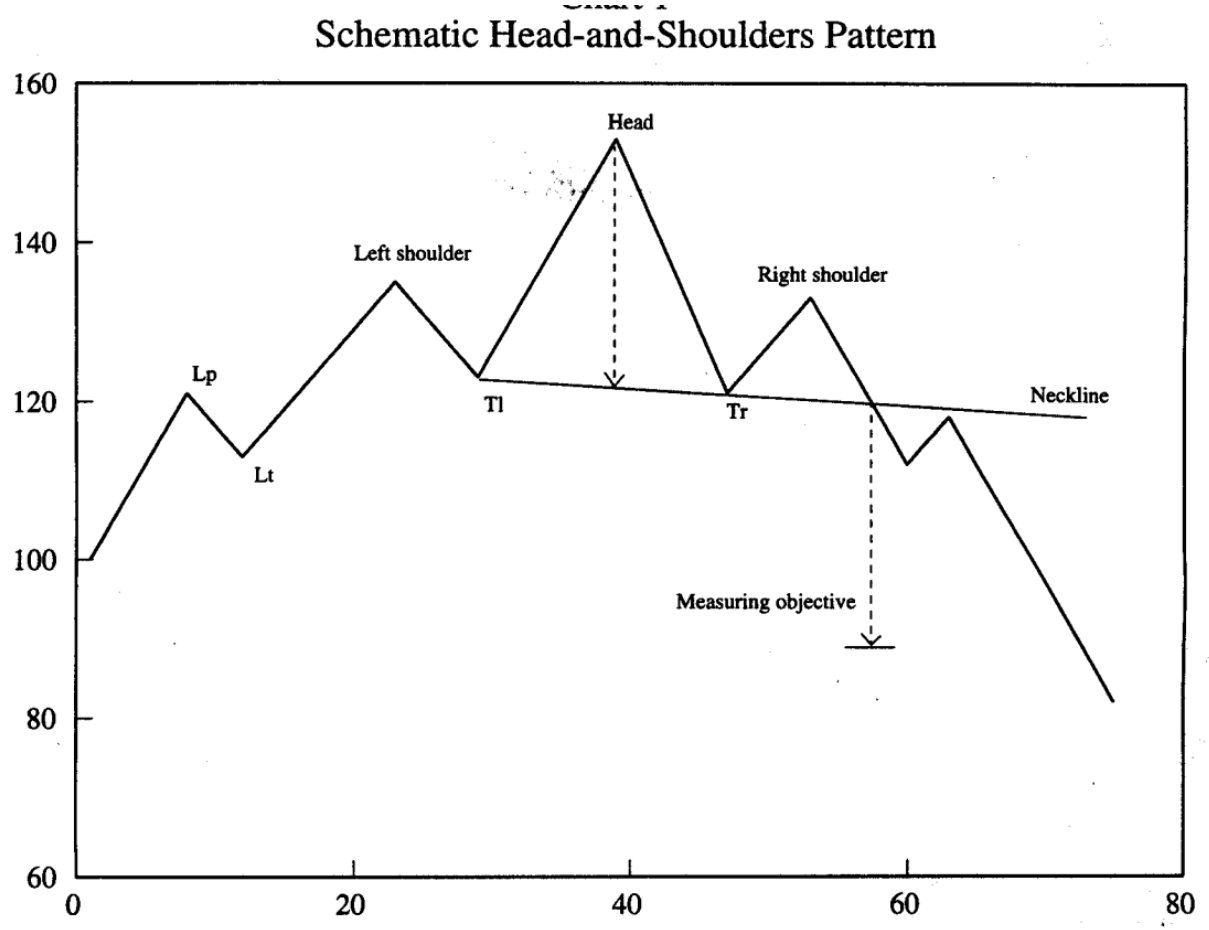
Head and Shoulders: Not Just a Flaky Pattern

- "We identify head-and-shoulders patterns using an objective, computer-implemented algorithm based on criteria in published technical analysis manuals. The resulting profits, replicable in real-time, are then compared with the distribution of profits for 10,000 simulated series generated with the bootstrap technique under the null hypothesis of a random walk."

- "Result: The trading rule has predictive power for 2 out of 6 FX crosses. If all 6 were to be traded, there would economically and statistically significant profits. The results are robust to changes in the parameters of the identification algorithm as well as sample period."

SMU
SINGAPORE MANAGEMENT UNIVERSITY

The Head Shoulders Pattern



Source: Head and shoulders: Not Just a Flaky Pattern, Carol Osler & Kevin Chang

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Andrew Lo: Technical Analysis

Lo, Andrew W., Harry Mamaysky and Jiang Wang

Foundations of Technical Analysis: Computational Algorithms, Statistical Inference and Empirical Implementation

Journal of Finance 55 (2000), 1705–1765.

Abstract:

Technical analysis, also known as ``charting'', has been a part of financial practice for many decades, but this discipline has not received the same level of academic scrutiny and acceptance as more traditional approaches such as fundamental analysis. One of the main obstacles is the highly subjective nature of technical analysis—the presence of geometric shapes in historical price charts is often in the eyes of the beholder.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Andrew Lo: Technical Analysis (2)

ABSTRACT (Contd.) In this paper, we propose a systematic and automatic approach to technical pattern recognition using non-parametric kernel regression, and apply this method to a large number of U.S. stocks from 1962 to 1996 to evaluate the effectiveness of technical analysis. By comparing the unconditional empirical distribution of daily stock returns to the conditional distribution—conditioned on specific technical indicators such as head-and-shoulders or double-bottoms—we find that over the 31-year sample period, several technical indicators do provide incremental information and may have some practical value.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Kernel Regression and Patterns

- The following patterns were analyzed on 3 different stock indices and 4 ETF's.

HS	Head and Shoulders
IHS	Inverted Head and Shoulders
BT	Broadening Tops
BB	Broadening Bottoms
TT	Triangle Tops
TB	Triangle Bottoms
RT	Rectangle Tops
RB	Rectangle Bottoms
PBULL	Proprietary Bull
PBEAR	Proprietary Bear

SMU
SINGAPORE MANAGEMENT UNIVERSITY

The origins of Markov Chain theory.

- Suppose you are given a body of text and asked to guess whether the letter at a randomly selected position is a vowel or a consonant. Since consonants occur more frequently than vowels, your best bet is to always guess consonant.

- Suppose we decide to be a little more helpful and tell you whether the letter preceding the one you chose is a vowel or consonant. Is there now a better strategy you can follow?

- In 1913, A.A. Markov was trying to answer the above problem analysed twenty thousand letters from Pushkin's poem Eugene Origin. He found that 43% letters were vowels and 57%, consonants. So in the first problem, one should always guess "consonant" and can hope to be correct 57% of the time.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Pushkin's Poetry and Markov Chains

- However, a vowel was followed by consonant 87% of the time. A consonant was followed by a vowel 66% of the time. Hence, guessing the opposite of the preceding letter would be a better strategy in the second case. Clearly, knowledge of the preceding letter is helpful.
- The real insight came when Markov took the analysis a step further. Markov investigated whether knowledge about the preceding two letters confers any additional advantage. He found that there was no significant advantage to knowing the additional preceding letter. This leads to the central idea of a Markov chain - while the successive outcomes are not independent, only the most recent outcome is of use in making a prediction about the next outcome.

Hidden Markov Models and Regime Specific Strategies

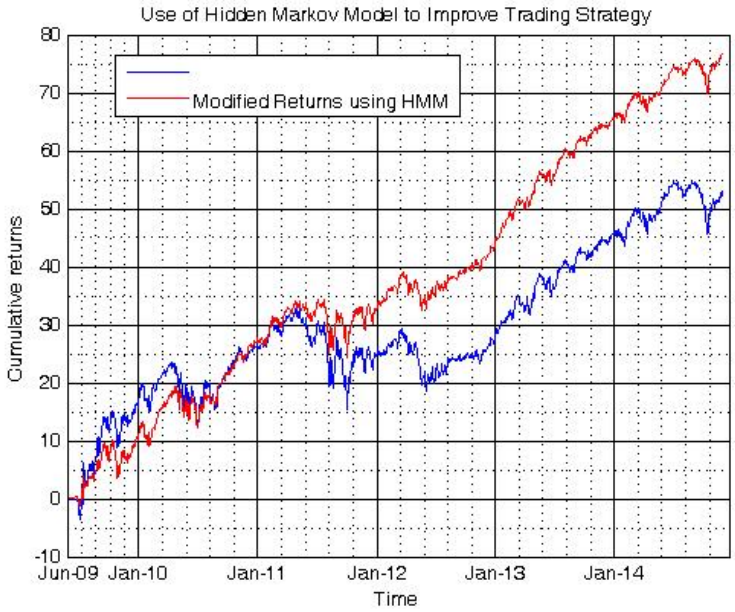
- We know that the stock market not only changes direction (bull versus bear markets) but also changes volatility.
- Calm, peaceful periods of low volatility are punctuated by turbulent periods; occasionally there are episodes of panic.
- A 4 –state HMM can be fit (peaceful, stable, turbulent, panic-riven). Aggressive/Conservative strategies and portfolio mix can be appropriate in different states.

See Appendix 4: Hidden Markov Models

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Example 14: HMM Improves Strategy Returns

Trading Signals enhanced by Proprietary HMM



SMU
SINGAPORE MANAGEMENT UNIVERSITY

Andrew Viterbi

Andrew Viterbi is an American electrical engineer

- Invented the Viterbi algorithm which is a dynamic programming based algorithm for finding the most likely sequence of hidden states that could have resulted in the sequence of observed events.
- Helped develop CDMA standard for cell phones
- Co-founded Qualcomm Inc.
- University of Southern California's Viterbi School of Engineering, is named in his honor in 2004 in recognition of his $52 million gift.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Natural Language Processing

- Linguistics is the scientific study of language, including its grammar, semantics, and phonetics. Classical linguistics involved devising and evaluating rules of language.
- Computational linguistics is the modern study of linguistics using the tools of computer science.
- The examples for sentiment analysis using bag of words/word embeddings are from the book *Deep Learning for NLP* by Jason Brownlee (and his blogs at machinelearningmastery.com)

See Appendix 5: Neural Networks;
Appendix 6: Backpropagation explained in depth

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Stop Words

- A stop word is a commonly used word (such as *the, a, is, are, just*) that provide little context
- They are removed during a text clean-up or pre-processing stage …
- ….because they provide more noise than signal for a computational linguistic analysis project.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Why is Vocabulary defined?

- Defining a vocabulary of known or preferred words is important in any natural language processing task.
- The larger the vocabulary, the larger the dimension of the vector space, and the larger is the representation of documents.
- It is more efficient to select only those words that are believed to have predictive power or informational content

Bag-of-Words Model

- Bag-of-Words (BoW) is a representation of text data used for document classification and feature extraction
- The task of text modelling is more complicated because Machine Learning cannot work on raw text directly; we need to represent text as numeric vectors.
- We make a vocabulary of known words and then compute the score for each word, that is, each word count is a feature.

Bag of Words-2

- One simple intuition is that two documents are similar if they have similar words.
- We can learn *something* about the of a document by examining these word-scores.
- An n-gram is an n-token sequence of words: a 2-gram or bigram is a two-word sequence of words like “how are” and a 3-gram or a trigram is a three-word sequence like “how are you”.
- It is called a bag-of-words , because any information about the order or structure of words in the document is discarded.
- One limitation of Bag of Words is that word order is not important; thus we cannot infer meaning from context
- For example
  - This is important vs Is this important
  - Good vs “not Good”

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Word Scoring Schemes used for Text Encoding by Tokenizer

- Binary - Where words are marked as present (1) or absent (0).
- Count - Where the occurrence count for each word is marked as an integer.
- Frequency - Where words are scored based on their frequency of occurrence within the document.
- TF-IDF - Where each word is scored based on its frequency, and words that are common across all documents are penalized.

Term Frequency-Inverse Document Frequency: TF-IDF

- A problem with scoring word frequency is that highly frequent words start to dominate in the document, but they may not contain as much informational content
- One approach is to rescale the frequency of words by how often they appear in all documents, so that the scores for frequent words like *the* or *that* are also frequent across all documents are penalized. This approach to scoring is called Term Frequency - Inverse Document Frequency, or TF-IDF for short, where:
  - *Term Frequency*: is a scoring of the frequency of the word in the current document.
  - *Inverse Document Frequency*: is a scoring of how rare the word is across documents.

Introducing the Movie Review Data

- The Movie Review Data is a collection of movie reviews retrieved from the **imdb.com** website.
- The reviews were collected and made available by Bo Pang and Lillian Lee for research work on NLP.
- The dataset comprises 1,000 positive and 1,000 negative movie reviews and is called the polarity dataset.
- It has become a standard dataset for natural language processing and sentiment analysis research, similar to Iris dataset for tutorials on clustering or the MNIST dataset for introductory computer vision.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Bag of Words & Movie Review Sentiment

- The essential steps are
  - Clean the data (remove punctuation, stop words, convert to lower case, etc.)
  - Create a vocabulary
  - Create tokens and encode strings to numeric output
  - The score each document by frequency of each word in the vocabulary. If there are N words in the vocabulary, each document is represented by a vector of length N, and the *N*-th entry is the count (or frequency) of the N-th word in the vocabulary
  - Fit a neural network to training data

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Example 15: Sentiment Analysis using Bag of Words

- Let us run some python code: **smu_moview_senti_BoW.ipynb**

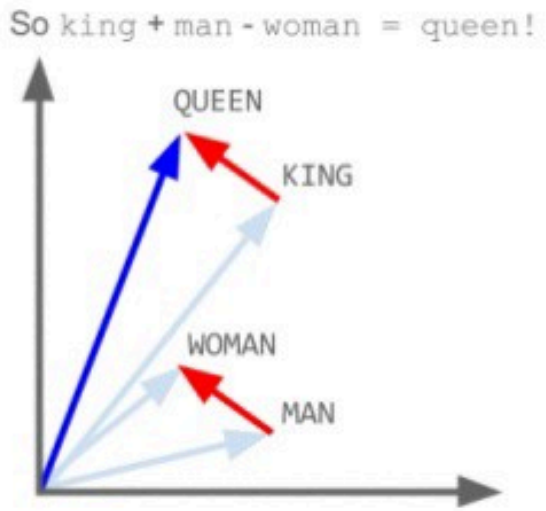
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 50)                740250
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 51
=================================================================
Total params: 740,301
Trainable params: 740,301
Non-trainable params: 0
_________________________________________________________________
Epoch 1/5
 - 2s - loss: 0.4617 - acc: 0.7910
Epoch 2/5
 - 1s - loss: 0.0746 - acc: 0.9895
Epoch 3/5
 - 1s - loss: 0.0191 - acc: 1.0000
Epoch 4/5
 - 1s - loss: 0.0088 - acc: 1.0000
Epoch 5/5
 - 1s - loss: 0.0052 - acc: 1.0000
Review: [Best movie ever! It was great, I recommend it.]
Sentiment: POSITIVE (59.050%)
Review: [This is a bad movie.]
Sentiment: NEGATIVE (61.744%)
```

Word Embedding Models: Google's Word2Vec, Stanford's GloVe

- A Word Embedding model provides dense vector representation of words that succeeds in capturing the semantic regularity in the language.
- The vector space representation of the words provides a projection where words with similar meanings are locally clustered within the space) words with similar meaning are represented by vectors which are close to each other.
- This is more more powerful than a Bag of Words approach which is simple (sparse vector representation of word counts or frequency) and can describe documents but not the meaning of the words
- Word2vec is a word embedding model that was developed by Tomas Mikolov at Google in 2013. Stanford University researchers (Pennington, Socher and Manning) came up with GloVe, Global Vectors for Word Representation which we will illustrate.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Linear Algebra on words:
queen = (king - man) + woman



This comes from the deck of slides: http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec-57135994

SMU
SINGAPORE MANAGEMENT UNIVERSITY

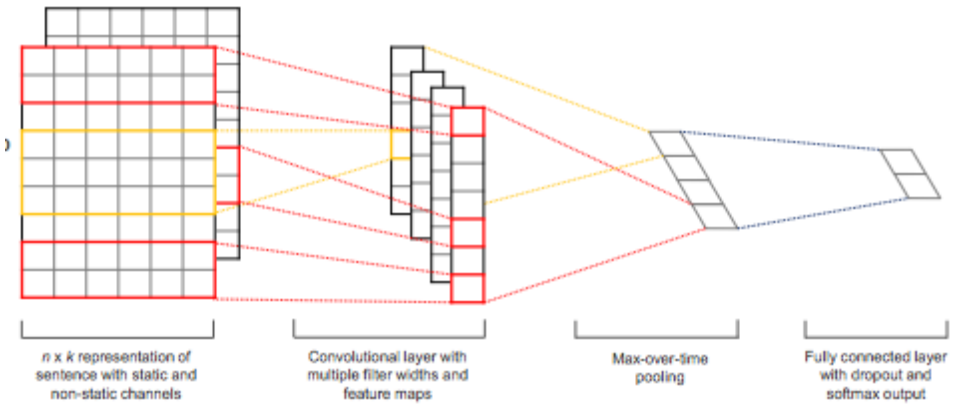
CNN for Sentence Classification

- "Convolutional Neural Networks for Sentence Classification" by Yoon KIM, New York University describes how to use CNN's in NLP tasks.

- Yoon Kim used an architecture which is a slight variant of the one in "Natural Language Processing (Almost) from Scratch" by R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuglu, P. Kuksa. *Journal of Machine Learning Research 2011*

SMU
SINGAPORE MANAGEMENT UNIVERSITY

CNN for Sentence Classification-2



- All the sentences are mapped to embedding vectors and then used as inputs.
- Convolution operations on the inputs using kernels of different sizes are used to extract feature maps.
- The feature maps are used as inputs to a max-pooling layer.
- Finally, a fully-connected layer with dropout & softmax out put is used for making the final prediction.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Example 16: Sentiment Analysis using Word Embedding + CNN Model

- We will use the Movie Review Polarity Dataset
- As before, we will clean the dataset by stripping it of punctuation, unnecessary white space, convert upper to lower case, and discard stop-words and any words with non-alphabetic characters.
- We will also define a vocabulary (of preferred words)

Example 16: Sentiment Analysis using Word Embedding + CNN Model-2

We will use a three component architecture:

- Word Embedding: is a vector space representation of each word; words which have similar meaning in language are also “close” in this representation
- Convolution Model: A CNN which will be used for feature extraction, effectively extracting meaningful sub-structures that are useful in overall prediction task
- Fully Connected Model: This interprets the features extracted by the CNN and makes predictions (note that CNN and fully connected layer to interpret the features and make predictions can be inside one neural network).

Example 16: Sentiment Analysis using Word Embedding + CNN Model-3

We can estimate a word embeddings model and then use a CNN :

Run the notebook **`smu_movie_senti_embeddings.ipynb`**

```
Vocabulary size: 14781
Maximum length: 1244

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 1244, 100)         1478100
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 1237, 32)          25632
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 618, 32)           0
_________________________________________________________________
flatten_1 (Flatten)          (None, 19776)             0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                197770
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 11
=================================================================
Total params: 1,701,513
Trainable params: 1,701,513
Non-trainable params: 0
_________________________________________________________________
Epoch 1/5
 - 17s - loss: 0.6902 - acc: 0.5233
Epoch 2/5
 - 16s - loss: 0.4947 - acc: 0.7822
Epoch 3/5
 - 17s - loss: 0.0995 - acc: 0.9744
Epoch 4/5
 - 17s - loss: 0.0097 - acc: 1.0000
Epoch 5/5
 - 16s - loss: 0.0029 - acc: 1.0000
Train Accuracy: 100.000000
Test Accuracy: 86.500000
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Example 17: Sentiment Analysis using GloVe

We can use Stanford's GloVe which is already estimated

Run the notebook **smu_movie_senti_glove.ipynb**

```
# load embedding from fileraw_embedding =
load_embedding(os.path.join(project_folder,'data/glove.6B/glove.6B.100d.txt'))
```

```
Train Accuracy: 99.78 percent
Test Accuracy: 71.00 percent
```

```
Review: [Everyone will enjoy this film. I love it, recommended!]
Sentiment: POSITIVE (51.382%)
Review: [This is a bad movie. Do not watch it. It sucks.]
Sentiment: POSITIVE (50.233%)
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Recurrent Neural Networks (RNN)

- Recurrent Neural Networks are powerful when it comes to modeling sequences. The RNN introduces recurrence by the use of loops, which allows us to model time dependence by passing information from one step to the next.
- The RNN is therefore a good framework for connecting the past information to the present, because there is time dependence.
- A Recurrent Neural Network (RNN) is able to do whatever an HMM can do.
- To summarize, RNN is powerful in two situations
  - Whenever temporal dependence in important
  - Wherever contextual information is important

Vanishing and Exploding Gradients

- During the process of training a neural network, gradient descent is utilized to update the network weights in the right direction and by the right amount. In each iteration of training, network weights are updated proportional to the partial derivative of the error function with respect to the current weight.
- There can be problems if back-propagated error either
  - blows up (exploding gradient) or
  - decays exponentially (vanishing gradient).
- Exploding gradient results in unstable model, (e.g., model weights become very large) and training loss changes very rapidly at each update. Clipping the gradients at a pre-defined threshold can control exploding gradients.
- Vanishing gradient prevents the weight from changing its value between updates, effectively stopping the neural network from further training. Vanishing gradient are more difficult to detect and control. (Utilizing ReLU instead of sigmoid activation, using regularization will help)
- Error flow analysis in existing RNNs found that long time lags were not effective to existing architectures, and Long Short-Term Memory (LSTM) is an RNN architecture specially designed to address the vanishing gradient problem.

LSTM

- A special type of RNN called LSTM (Long Short Term Memory) is used to model temporal dependence (time series prediction) as well as handle contextual information (what is the current state of the market).

- This allows the network to learn when to forget previous hidden states and when to update hidden states given new information
- LSTM can be used for sequence predictions.
- LSTM is powerful in speech recognition, machine translation and (when combined with CNN's) for image captioning.

RNN: "The Unreasonable Effectiveness of Recurrent Neural Networks" by Andrej Karpathy

Let us read from this excellent article:

What makes Recurrent Networks so special?

A glaring limitation of Vanilla Neural Networks (and also Convolutional Networks) is that their API is too constrained:

- They accept a fixed-sized vector as input (e.g. an image) and produce a fixed-sized vector as output (e.g. probabilities of different classes).
- These models perform this mapping using a fixed amount of computational steps (e.g. the number of layers in the model).
- The core reason that recurrent nets are more exciting is that they allow us to operate over *sequences* of vectors: Sequences in the input, the output, or in the most general case both.

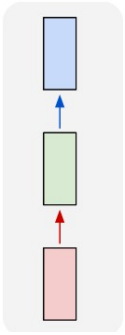
Lets look at examples may make this more concrete

SMU
SINGAPORE MANAGEMENT UNIVERSITY

RNN: “The Unreasonable Effectiveness of Recurrent Neural Networks” by Andrej Karpathy

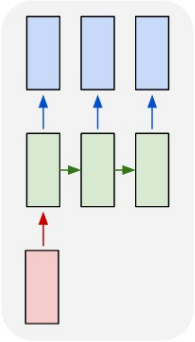
Each rectangle is a vector and arrows represent functions (e.g. matrix multiply). Input vectors are in red, output vectors are in blue and green vectors hold the RNN's state

one to one



(1) Vanilla mode of processing without RNN, from fixed-sized input to fixed-sized output (e.g. image classification).

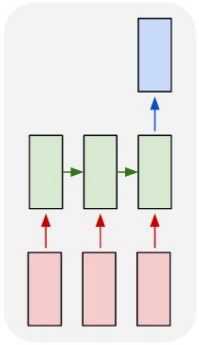
one to many



(2) Sequence output (e.g. image captioning takes an image and outputs a sentence of words).

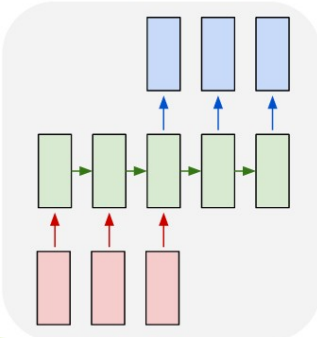
SMU
SINGAPORE MANAGEMENT UNIVERSITY

many to one



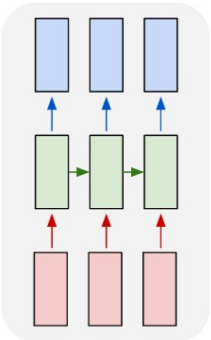
(3) Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment).

many to many



(4) Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French).

many to many



(5) Synced sequence input and output (e.g. video classification where we wish to label each frame of the video). Notice that in every case are no pre-specified constraints on the lengths sequences because the recurrent transformation (green) is fixed and can be applied as many times as we like.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

J.M. Keynes

- When the facts change, I change my mind. What do you do, sir? (attributed to Keynes)

SMU
SINGAPORE MANAGEMENT UNIVERSITY