Background:
Generalization: The concept of handling different objects by a common property. The set of objects of the common property are called a class, the elements are called instances.
Neuron: all instances of input signals below an energy threshold are ignored.

Generalization in Machine Learning: All inputs close enough to each other result in the same output. This is for an assumed or trained definition of close enough (generalization distance).
Memory: Only identical being close enough implies no generalization

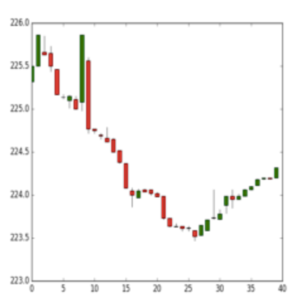
Capacity Requirement of a Dataset("estimated memory equivalent capacity"): Build a static-parameter machine learner to memorize the training data.(Essentially finding how many simple biased layers are needed to perfectly classify). Assume exponential improvement through training. This is, the memory-equivalent capacity can be minimally logarithmic of the size of the static-parameter machine learner. This estimate is very significantly lower than the amount of bits needed to make a table of the data for cases where useful information is present. Only when your data is truly random(e.g. gaussian noise) will this estimate be the same as the size of a memory lookup table.
Memory-equivalent Capacity for Neural Networks :
1. The output of a single neuron is maximally one bit.
2. The memory capacity of a single neuron is the number of parameters in bits.
3. The memory capacity of neurons in a single layer is additive.
4. The memory capacity of neurons in a subsequent layer is limited by the output of the layer it depends on.

Generalization Progression: Estimate the capacity needed to memorize 10%, 20%, . . . , 100% of training table. If the capacity does not stabilize at the higher percentages there is not enough training data to generalize or the representation function(s).

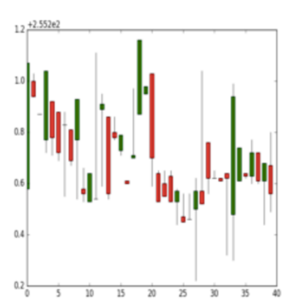
Question: Can I predict the rise and fall of bitcoin using financial charts of its prices.

Hypothesis: I can determine 5-10 features of these charts that very confidently(85%+ accuracy) predict the direction of price change.

Data: CSV file containing full price history of bitcoin converted into matplotlib-finance visual charts with 40 price points in each chart labeled up or down based on the future price change. Training set is 550000 of these charts. Created validation and test sets with 30% and 10% of the size. Classes are balanced with equal amount of up and down.
Example data:



UP



DOWN

Feature extraction: Trained Res Net-18 binary classifier(state of the art neural network for computer vision tasks) to 99.99% training classification accuracy which achieved 87.6% validation classification accuracy. (Prior work using the same price history and data generator with AlexNet achieved 70% validation accuracy with a much smaller dataset)

Analysis thus far:
Embedded the data into 512 dimensions(features in the last layer of ResNet) feature space to see if this network is extracting generalizable information for bitcoin prediction. Generalization Progression results show that the information in this feature space is potent enough that you only need about 70% of the data for the memory equivalent capacity to level off.

Current Analysis:
1) Training a logistic regressor on the features with L1 norm penalty to enforce sparsity in the weights. I then get the indices of the top weights based on the L1 norm contributions of the weights based off of https://rakeshchada.github.io/Sentiment-Neuron.html interpretation of https://arxiv.org/pdf/1704.01444.pdf. This will allow me to choose 5-10 out of the 512 features that can model most of the accuracy and will likely generalize better to unseen data. Once completed will find the memory equivalent capacity of these features and check the generalization progression (compare which levels off sooner)

2)Testing several low dimensional embeddings of the feature space such as T-SNE and PCA to compress into 2 or 3 dimensions to see if the two classes separate and cluster in an aesthetically pleasing way and I will evaluate based on subjective visual appeal.

Testing:
1) Train an ensemble of estimated memory equivalent capacity sized neural networks on the 512 features and test accuracy on held out set.
2) Compare results to ensemble of estimated memory equivalent capacity sized neural networks on the 5-10 dimension features.

Future directions: Train sentiment analysis network on bitcoin and crypto related hashtagged posts from twitter. Embed data in the sentiment networks feature space and go through a similar pipeline of testing generalization on the data, using L1 logistic regression to find the 5-10 most important features. I will then test generalization on these features. Finally, I will test accuracy and generalization progression improvements of combining the high dimensional sentiment features with high dim resNet features and combining the low dim sentiment features with the low dim resNet features