A Machine Learning Investigation of One-Month Momentum

Ben Gum

Machine Learning Investigation of One-Month Momentum
Contents

- **Problem**

- Data

- Recent Literature

- Simple Improvements

- Neural Network Approach

- Conclusion

- Appendix : Some Background on Neural Networks

Machine Learning Investigation of One-Month Momentum
Problem

- In 1990, Jegadeesh (Evidence of Predictable Behavior in Security Returns) showed that there is a negative auto-correlation in one month active stock returns.

  - That is, a portfolio that is long last month's losers and short last month's winners has significant positive returns.

  - The intuition for this is that investors may over-value last month's winners, which can depress returns in the near future.

- Since 1990, this effect has decayed substantially, but others have published refinements of the one-month return reversal effect, by considering the daily returns within the trailing month (rather than the simple aggregate).

- Based on this, we investigate how we can predict month t+1 returns using the daily returns during month t. We use traditional techniques based on rather literature and then compare with a straight-forward neural network approach.

Problem : Metrics

- As with recent literature, we use the metric of the returns to an equal-weighted quintile difference portfolios.

  - That is, for a signal s, we sort the stocks by s then construct a long-short portfolio that goes long the highest quintile of s and is short the lowest quintile of s.

  - This is a rather standard metric that is independent of the overall market return and generally avoids issues with size biases (equal-weighted long-only portfolios often have a small cap tilt that can obscure the results of the underlying signal).

  - In addition, the quintile difference portfolio is generally investible and is not overly affected by outliers.

Machine Learning Investigation of One-Month Momentum
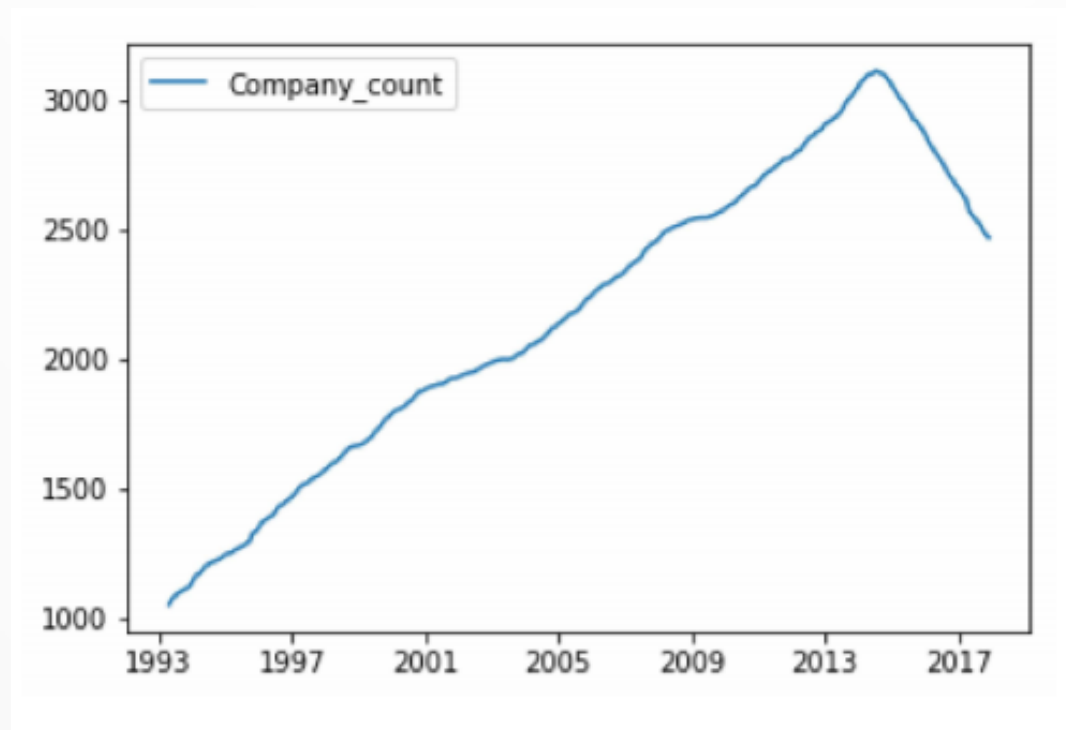## Contents

- ~~Problem~~

- **Data**

- Recent Literature

- Simple Improvements

- Neural Network Approach

- Conclusion

- Appendix : Some Background on Neural Networks

Data

- We get our daily stock returns from the Quandl Wiki database

  - Quandl has a number of handy financial databases, but for the exercise, we only need the daily returns (from daily adjusted close prices).

  - Quandl data is accessible via API or by mass download. For convenience we did a mass download of the entire Quandl Wiki database

  - Quandl Wiki daily price data begins in 1960 and continues to present .

- We also pull the daily S&P 500 (SPY) returns from the Yahoo Finance API.

  - This starts in 1993, so we have 25 years of total test data.

  - All of this is out of sample with respect to the Jegadeesh paper.

Data : Further Details

- Although the Quandl Wiki price data begins in 1960, the coverage is very thin initially.

  - During our test period of 1993-2017, the number of companies ranges from 1049 to 3113

  - In aggregate, we have 644,583 (company,month) data points for the study.

Machine Learning Investigation of One-Month Momentum
Contents

- ~~Problem~~

- ~~Data~~

- **Recent Literature**

- Simple Improvements

- Neural Network Approach

- Conclusion

- Appendix : Some Background on Neural Networks

Machine Learning Investigation of One-Month Momentum
Recent Literature

- We replicate and expand upon the findings in of Asness et al (Betting Against Correlations 2017)

  - By considering daily returns, they derive signals of market correlation and of positive skew of returns and show that both of these result in under-performance over the subsequent month

  - The intuition for this is similar to that of Jegadeesh. Investors over-pay for market correlations and positive skew and thus these metrics disappoint in the next month.

  - Asness at al add some additional reasoning regarding the contradiction with CAPM. CAPM says that investors who want to out-perform the market will simply buy a leveraged market portfolio. However when investors are unable or unwilling to lever, they may seek out positive skews and/or higher beta.

- Asness et al define SMAX (scaled max) as a metric of positive skew as a function of the daily total returns over the trailing month.

  - SMAX = (Average of highest 5 total daily returns)/(Stdev of the daily returns)

  - Note that max return and stdev of returns are highly correlated so the scaling removes this correlation, leaving a pure measure of positive skew.

Recent Literature : Signal Replication

- To replicate the results of Jegadeesh ad Asness et al, we calculate the following statistics for each company,month pair

  - **forward_return**, the return to the company in month t+1 (our dependent variable)

  - **spycorr**, the correlation of the daily returns in month t between the company and SPY

    - This is a variant of the market correlation metric from Asness et al

  - **ret**, the monthly return of the company in month t (Jegadeesh, 1990)

  - **max5**, the average of the largest five total daily returns of the company in month t

  - **min5**, the average of the smallest five total daily returns of the company in month t

  - **std**, the standard deviation of the total daily returns of the company in month t

  - **sret**, scaled return = ret/std

  - **smax**, scaled max = max5/std

  - **smin**, scaled min = min5/std

Recent Literature : Signal Replication, a Snapshot View

- The table below shows the signals for a handful of companies for January 2014

  - Note that we are not concerned with the relative scales of the independent variables (spycorr, …, smin) since we are using them to create quintile difference portfolios and are thus only concerned with rankings.

	forward_return	spycorr	ret	max5	min5	std	sret	smax	smin
A	0.02	0.56	0.07	0.02	-0.01	0.01	6.60	1.79	-0.70
AAL	0.33	-0.04	-0.60	0.03	-0.07	0.06	-10.25	0.45	-1.26
AAMC	0.08	-0.16	0.28	0.07	-0.04	0.05	6.21	1.60	-0.83
AAN	-0.09	0.15	0.03	0.01	-0.01	0.01	3.56	1.44	-1.16
AAOI	-0.09	0.28	0.14	0.05	-0.03	0.03	3.99	1.58	-1.00

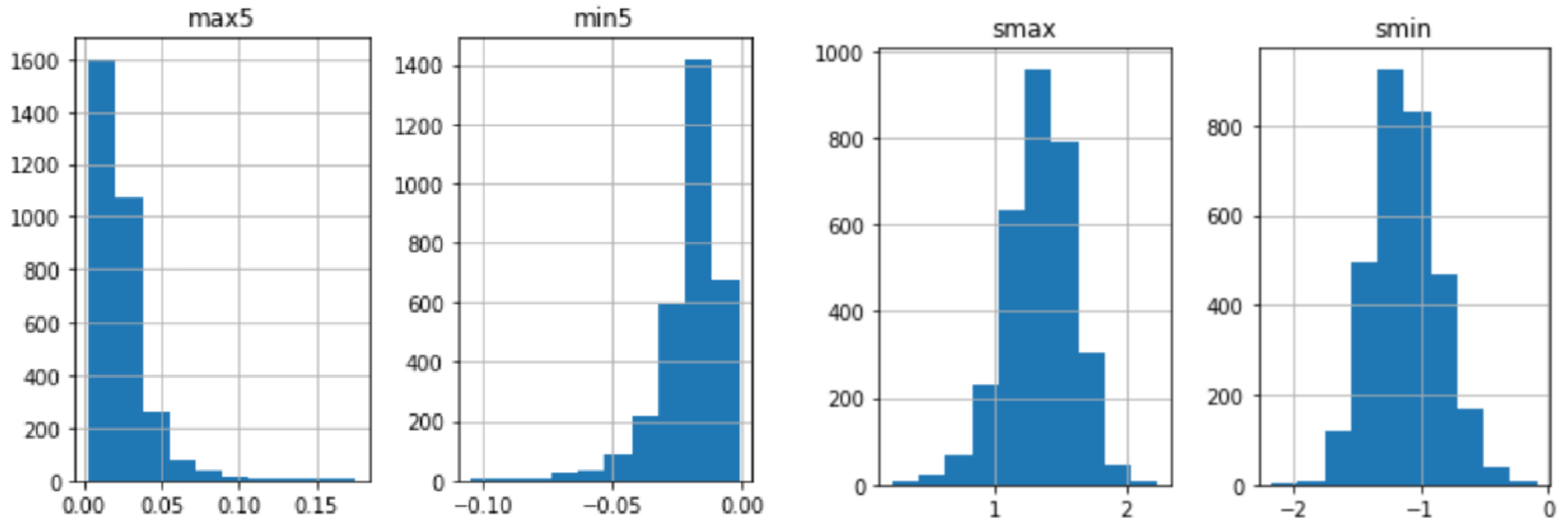# Recent Literature : Signal Replication, Snapshot Correlations

- The table below shows the correlations among our independent variables for January 2014

  - With the exception of spycorr, the correlations among the variables are fairly intuitive and representation of the time series as a whole

	spycorr	ret	max5	min5	std	sret	smax	smin
spycorr	1.00	-0.14	-0.38	0.37	-0.43	-0.02	0.07	-0.06
ret	-0.14	1.00	0.61	0.16	0.30	0.79	0.68	0.71
max5	-0.38	0.61	1.00	-0.61	0.91	0.29	0.29	0.40
min5	0.37	0.16	-0.61	1.00	-0.85	0.38	0.41	0.32
std	-0.43	0.30	0.91	-0.85	1.00	0.01	-0.05	0.15
sret	-0.02	0.79	0.29	0.38	0.01	1.00	0.87	0.85
smax	0.07	0.68	0.29	0.41	-0.05	0.87	1.00	0.83
smin	-0.06	0.71	0.40	0.32	0.15	0.85	0.83	1.00

## Recent Literature : Signal Replication, Snapshot Distributions

- The histograms below show the distributions of some independent variables for January 2014

  - Predictably, max5 and min5 are quite skewed. Dividing by std removes this skew.

Recent Literature : Signal Replication, Quintile Returns

- The chart and table below the returns to the quintile portfolios for each signal

  - The chart is rolling 10-year average to show the decay. These confirm that these signals work over the test period, but have decayed substantially.
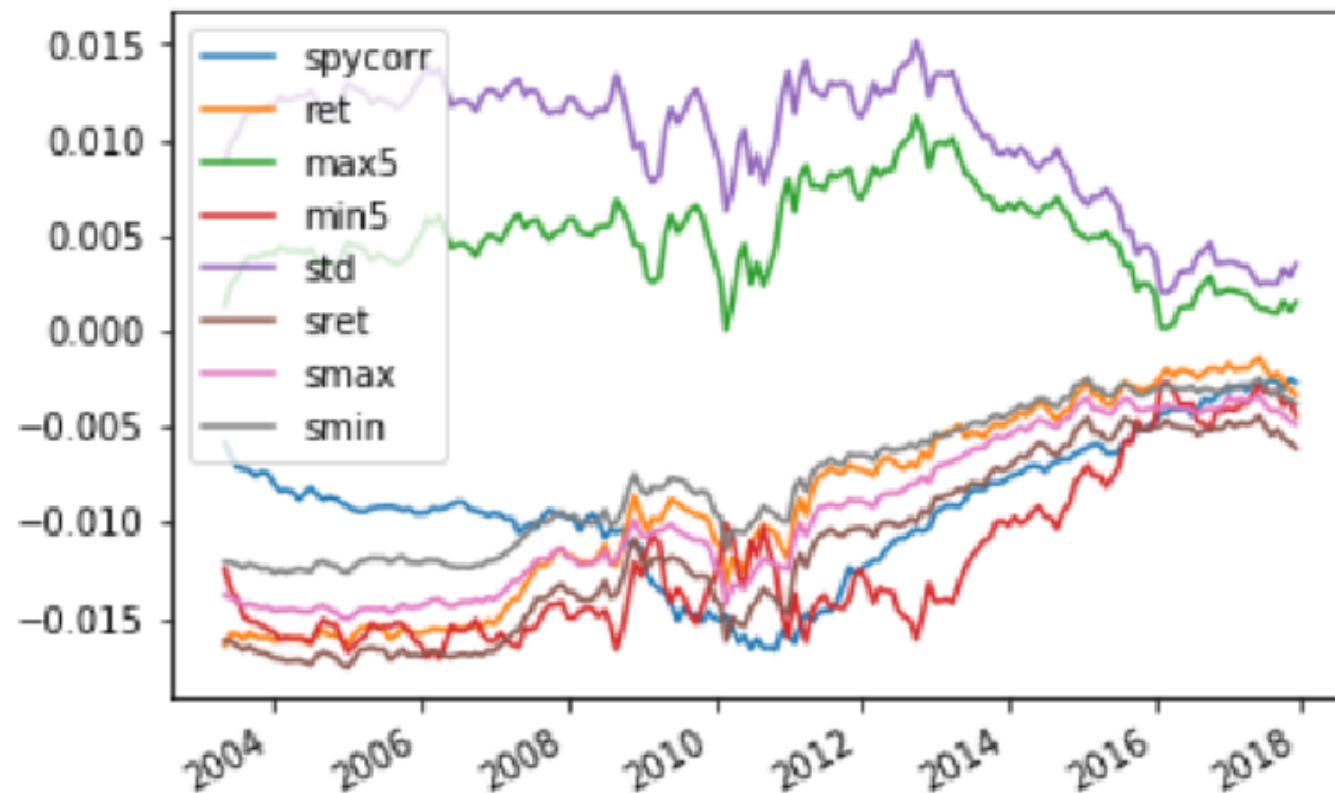
	Mean	Stdev	T-Stat
spycorr	-0.56%	3.26%	-2.97
ret	-0.86%	4.53%	-3.26
max5	0.31%	6.47%	0.83
min5	-0.93%	6.95%	-2.31
std	0.75%	6.94%	1.86
sret	-1.04%	3.96%	-4.52
smax	-0.88%	3.14%	-4.81
smin	-0.72%	3.12%	-3.98

Machine Learning Investigation of One-Month Momentum
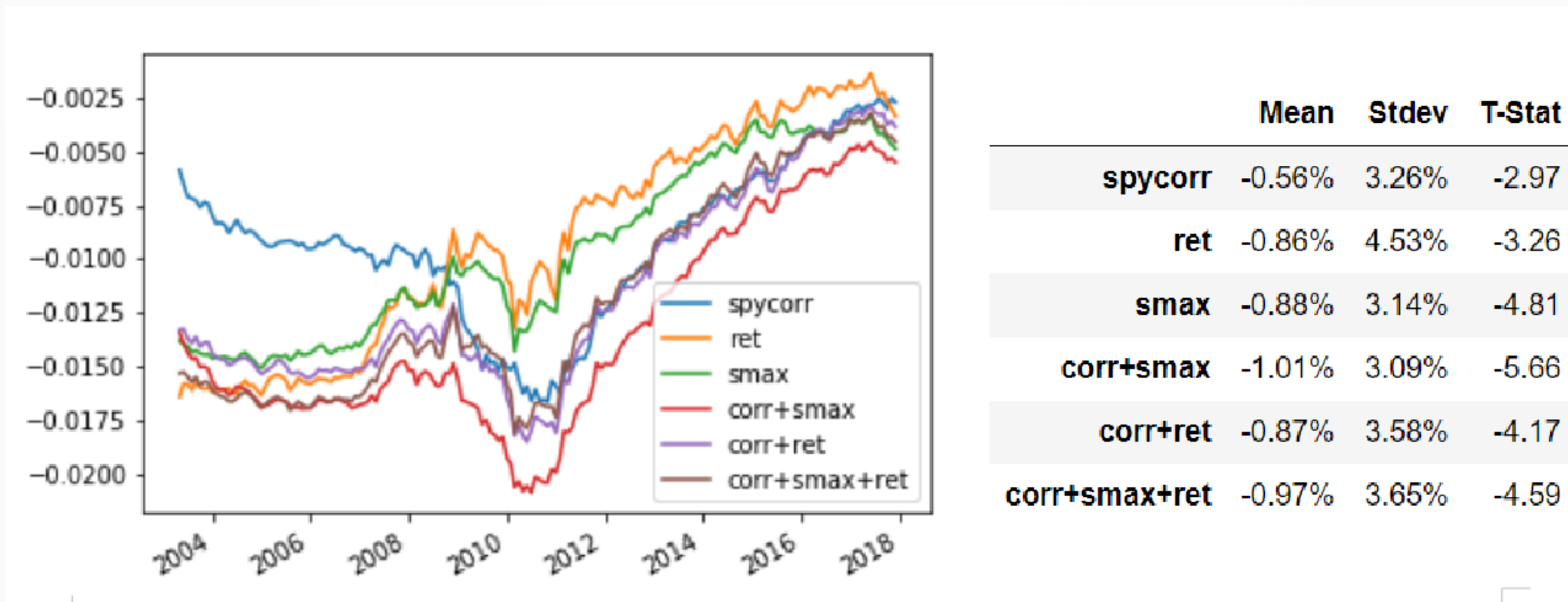## Contents

- ~~Problem~~

- ~~Data~~

- ~~Recent Literature~~

- **Simple Improvements**

- Neural Network Approach

- Conclusion

- Appendix : Some Background on Neural Networks

Simple Improvements

- To improve upon the individual signals, we create simple combinations that are equal weighted averages of standardized signals (so that the relative weights are consistent through time).

  - We can see that corr+smax is now our best signal. Note that this could be further optimized (though at the risk of over-fitting in sample).

	Mean	Stdev	T-Stat
spycorr	-0.56%	3.26%	-2.97
ret	-0.86%	4.53%	-3.26
smax	-0.88%	3.14%	-4.81
corr+smax	-1.01%	3.09%	-5.66
corr+ret	-0.87%	3.58%	-4.17
corr+smax+ret	-0.97%	3.65%	-4.59

Machine Learning Investigation of One-Month Momentum
Contents

- ~~Problem~~

- ~~Data~~

- ~~Recent Literature~~

- ~~Simple Improvements~~

- **Neural Network Approach**

- Conclusion

- Appendix : Some Background on Neural Networks

Neural Network Approach : What, Why, and How

- What : I think of a neural network is a generalization of a regression

  - A regression transforms a few medium strength predictors into a strong predictor.

  - A Neural Network transforms many weak predictors into a few medium strength predictors and then into a strong predictor.

- Why : In this problem we have the ~20 daily returns from month t (ie weak predictors of return) and want a strong prediction of return in month t+1.

  - Instead of picking the medium strength from predictors literature (such as corr and smax in the previous section), we train the neural network to pick.

- How : In order to train on multiple months of data, we will need to modestly transform the inputs.

  - This will lose some information, but will prevent the network from trying to overfit based on aspects specific to particular months.

  - We limit ourselves to the last 20 business days of the month.

 - In addition, we transform all returns to active and do not use the SPY returns.

Neural Network Approach : Training/Testing split

- We will need to be careful about splitting our training and testing test sets so that our test results are realistic, since Neural Networks have to potential to substantially overfit.

- To do this, we use a technique called rolling lagged cross-validation.

  - For each year y in our testing window, we first train our network on the 60 months in years y-5 to y-1. We then test our network on the 12 months of year y.

  - For example, our first training set will be the 73,382 company,month pairs from 1993-1997.

  - Our training independent inputs are then a 73,382x20 matrix of daily returns and our dependent for training is a 73,382x1 matrix of the subsequent monthly returns

- Our initial model is consists of our 20 daily return inputs (weak predictors), 20 nodes in a hidden layer (medium strength predictors), and one output node (ideally a strong predictor of active returns).

- Although our model itself is a bit of a black box, the rolling lagged cross-validation assures us that all of our testing is out of sample.

- The chart and table below show the performance of five neural network model structures over our test window 1998-2017. The 'single' model with one hidden layer of 20 nodes does best.
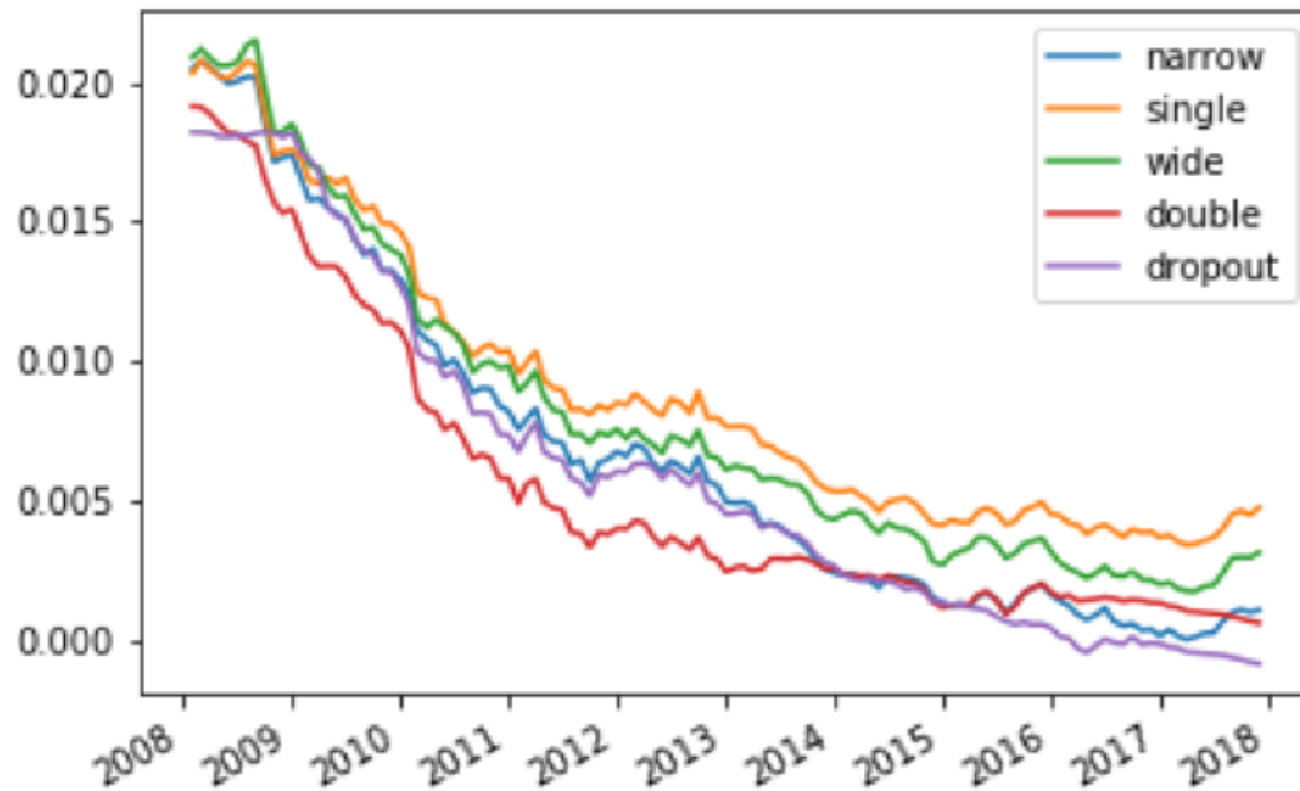
	Mean	Stdev	T-Stat
narrow	1.10%	3.34%	5.08
single	1.27%	3.47%	5.66
wide	1.23%	3.52%	5.38
double	1.01%	3.16%	4.94
dropout	0.87%	3.02%	4.46

Machine Learning Investigation of One-Month Momentum
Neural Network Approach : Comparison with Others

- Over the 1998-2017 test window, the neural network model out-performs the trailing return (Jegadeesh), SMAX (Asness et al), as well as our simple combination of idea, Smax + SpyCorr.

    - Though all models decay substantially in strength over time.

	Mean	Stdev	T-Stat
Neural Network	1.27%	3.47%	5.66
Trailing Return	0.76%	4.91%	2.40
Smax	0.83%	3.38%	3.81
Smax + SpyCorr	1.03%	3.32%	4.79

Neural Network Approach : Correlation with Others

- Over the 1998-2017 test window, the neural network model has only modest correlations with the trailing return (Jegadeesh), SMAX (Asness et al), and our simple combination of idea, Smax + SpyCorr.

  - This suggests that we could create an even stronger model by combining the neural network with our simple combination of fundamental signals.

	Neural Network	Trailing Return	Smax	Smax + SpyCorr
Neural Network	1.00	0.30	0.22	0.04
Trailing Return	0.30	1.00	0.90	0.52
Smax	0.22	0.90	1.00	0.67
Smax + SpyCorr	0.04	0.52	0.67	1.00

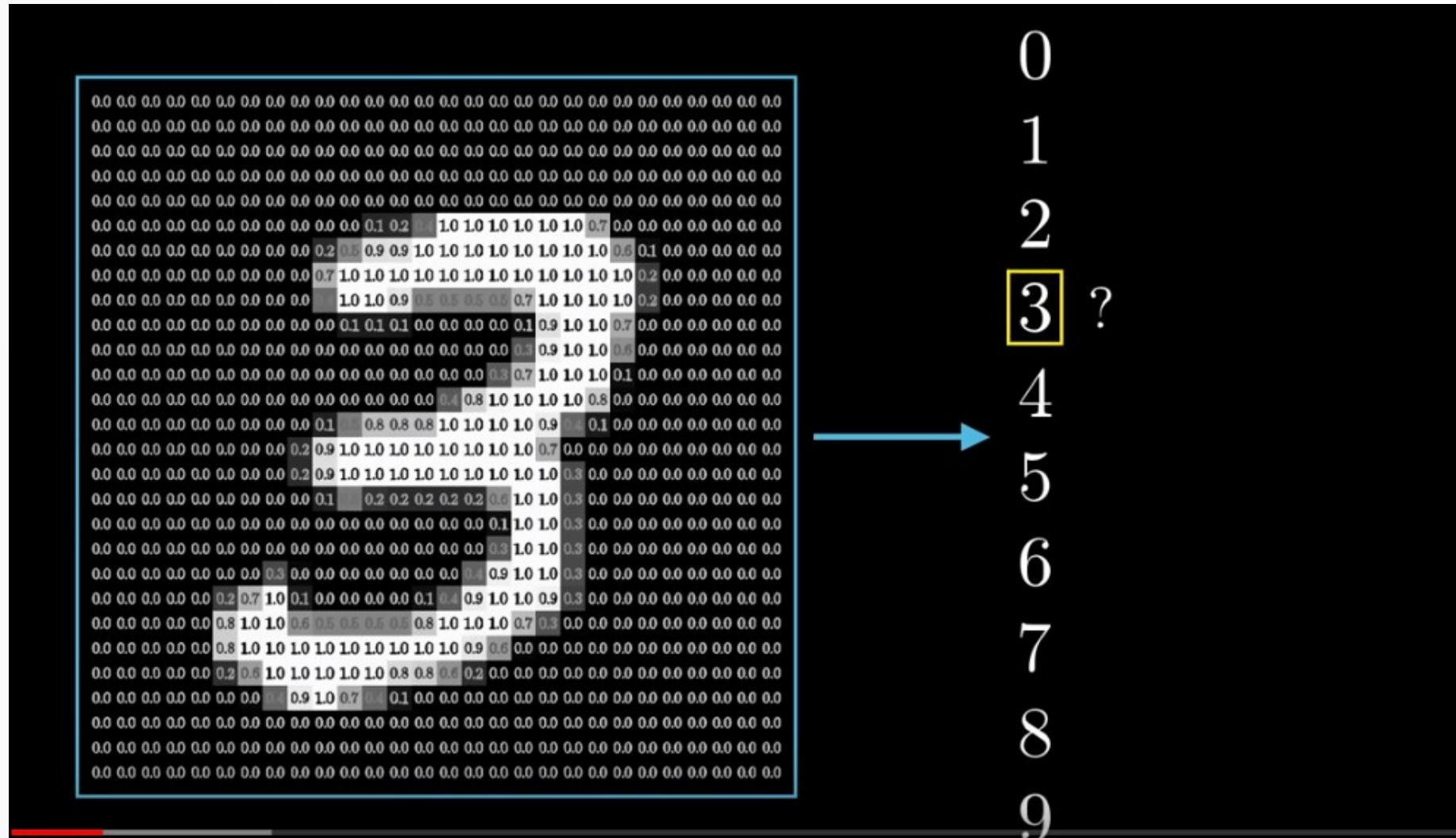# Machine Learning Investigation of One-Month Momentum
Contents

- ~~Problem~~

- ~~Data~~

- ~~Recent Literature~~

- ~~Simple Improvements~~

- ~~Neural Network Approach~~

- **Conclusion**

- Appendix : Some Background on Neural Networks

Conclusions: Findings and Questions

- Using daily and monthly returns from the US Stock Market from 1993-2017, we have:

  - Replicated signals from Jegadeesh and Asness et al and shown that while they continue to work, their effectiveness has decreased substantially over time.

  - Created a simple combination of signals that out-performs either of the components.

  - Trained and tested a neural network model that out-performs the all of the above.

- Questions:

  - Can we generate an improved signal by combining the models from literature with the neural networks?

  - Is there a way to make use of market/total returns to improve our neural network model?

  - Would a different structure of neural network give us an improved prediction?

Machine Learning Investigation of One-Month Momentum
Contents

- ~~Problem~~

- ~~Data~~

- ~~Recent Literature~~

- ~~Simple Improvements~~

- ~~Neural Network Approach~~

- ~~Conclusion~~
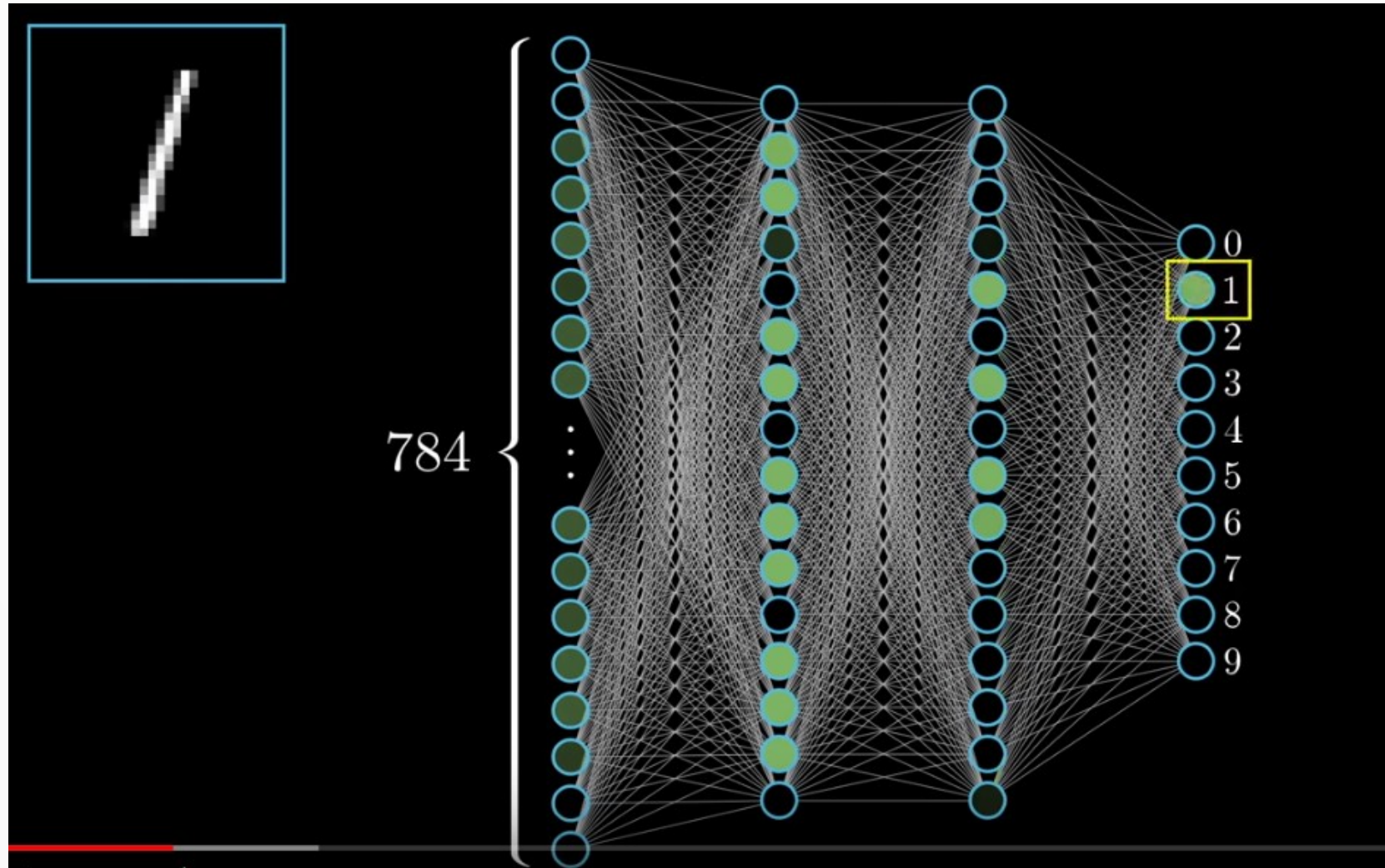
- **Appendix : Some Background on Neural Networks**

Some Background on Neural Networks

- There are many online resources on Neural Networks, but my favorite is the four video YouTube series by 3Blue1Brown. It gives a deep dive into using a neural network to recognize a hand-written digit.

  - The follow slides are a brief sampling of that deep dive.

- Digit/image recognition is a very powerful application of neural networks, but differs from the stock return prediction in several fundamental ways.

  - Digit/image recognition is a classification problem, in that the dependent variable is one of a finite number of classes, such as the digits 0-9.

    - Stock return prediction has a continuous dependent variable. While the general methodology is similar, the continuous neural network will have a different functions at each node.

  - Digit/image recognition has the advantages that:

    - More images/digits can always be generated for training.

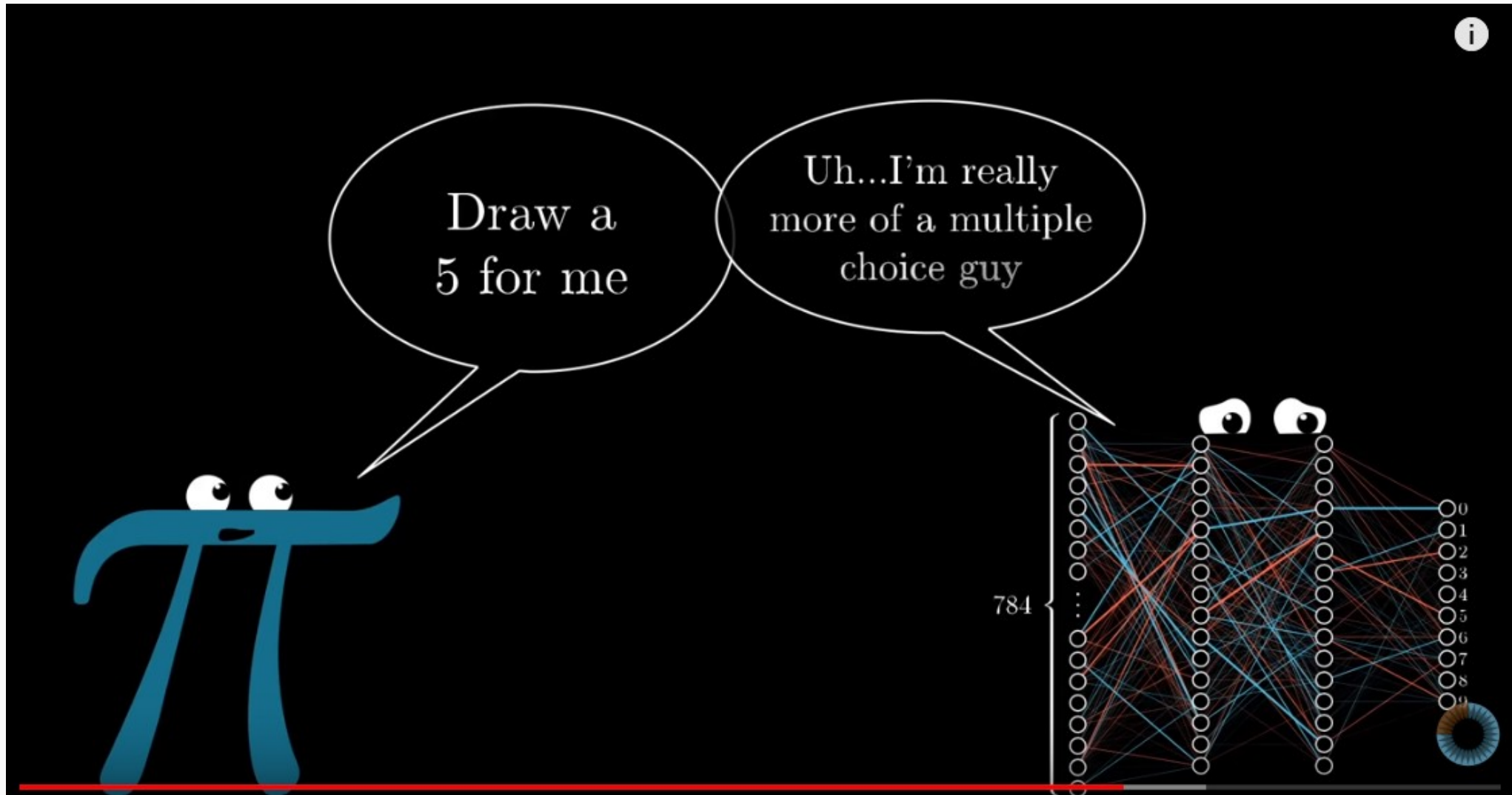   - **Yesterday's digits/images are very similar to tomorrow's. Returns might not be.**

BLUE1BROWN SERIES S3 • E1 But what *is* a Neural Network? | Deep learning, chapter 1

BLUE1BROWN SERIES S3 • E1 But what *is* a Neural Network? | Deep learning, chapter 1

Neural Networks by 3Blue1Brown : Classifier Limitations

BLUE1BROWN SERIES S3 • E2 Gradient descent, how neural networks learn | Deep learning, chapter 2

Neural Networks by 3Blue1Brown : A Single Node

BLUE1BROWN SERIES S3 • E1 But what *is* a Neural Network? | Deep learning, chapter 1
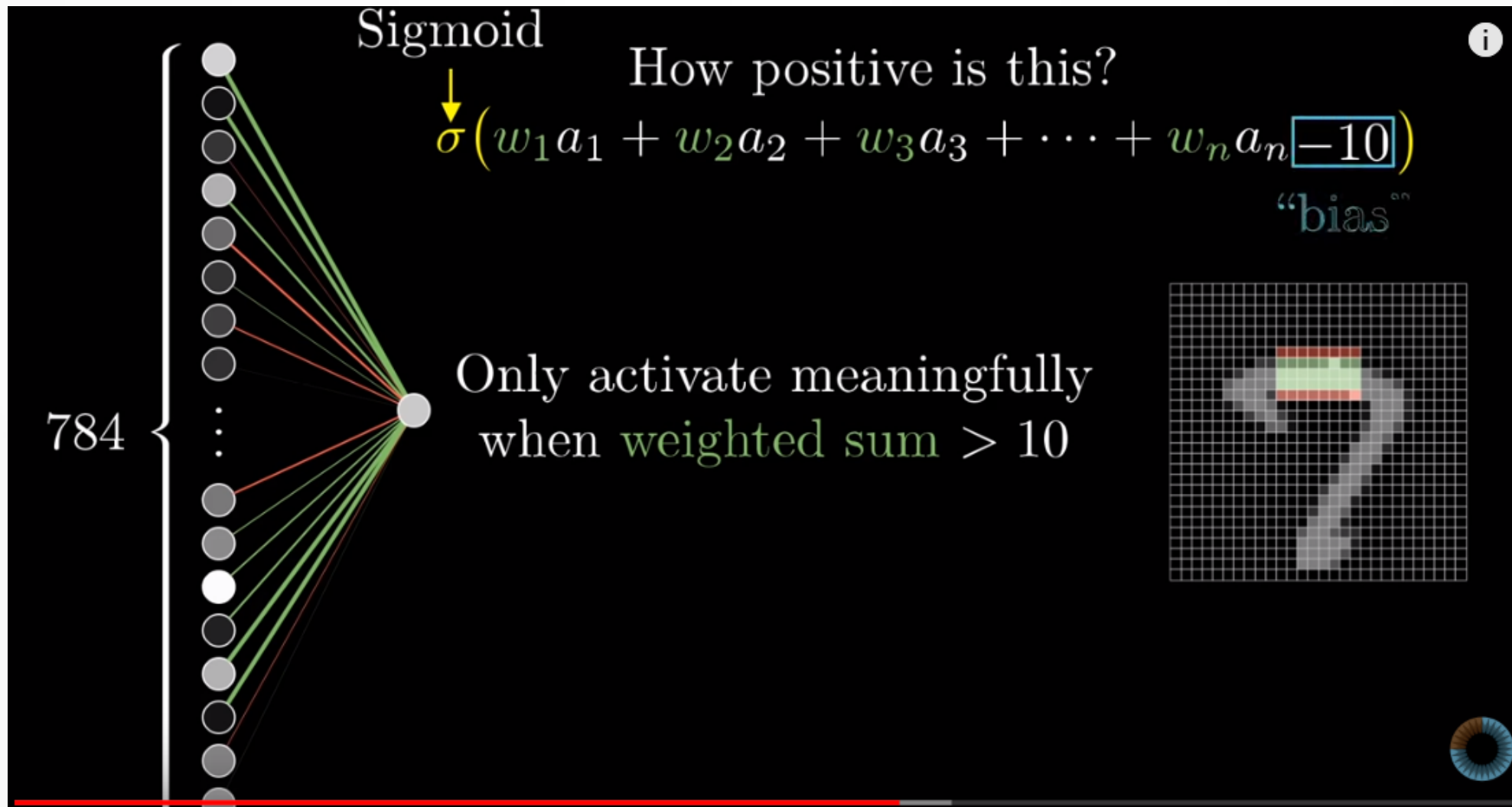
Machine Learning Investigation of One-Month Momentum
Neural Networks by 3Blue1Brown : 13,002 Parameters!

$$784 \times 16 + 16 \times 16 + 16 \times 10$$
weights

$$16 + 16 + 10$$
biases

$$13,002$$

784

BLUE1BROWN SERIES S3 • E1 But what *is* a Neural Network? | Deep learning, chapter 1

BLUE1BROWN SERIES S3 • E2 Gradient descent, how neural networks learn | Deep learning, chapter 2

BLUE1BROWN SERIES S3 • E2 Gradient descent, how neural networks learn | Deep learning, chapter 2

	2	5	0	4	1	9		Average over all training data
w_0	−0.08	+0.02	−0.02	+0.11	−0.05	−0.14	⋯	→ −0.08
w_1	−0.11	+0.11	+0.07	+0.02	+0.09	+0.05	⋯	→ +0.12
w_2	−0.07	−0.04	−0.01	+0.02	+0.13	−0.15	⋯	→ −0.06
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
$w_{13,001}$	+0.13	+0.08	−0.06	−0.09	−0.02	+0.04	⋯	→ +0.04

BLUE1BROWN SERIES S3 • E3 What is backpropagation really doing? | Deep learning, chapter 3

$$\boxed{\frac{\partial C_0}{\partial a_k^{(L-1)}}} = \underbrace{\sum_{j=0}^{n_L-1} \frac{\partial z_j^{(L)}}{\partial a_k^{(L-1)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial C_0}{\partial a_j^{(L)}}}_{\text{Sum over layer } L}$$

$$z_j^{(L)} = \cdots + w_{jk}^{(L)} a_k^{(L-1)} + \cdots$$

$$a_j^{(L)} = \sigma(z_j^{(L)})$$

$$C_0 = \sum_{j=0}^{n_L-1} (a_j^{(L)} - y_j)^2$$

$a_k^{(L-1)} \rightarrow$

0.11

0.83 $\leftarrow a_0^{(L)}$

0.40

0.63 $\leftarrow a_1^{(L)}$

0.81

BLUE1BROWN SERIES S3 • E4 Backpropagation calculus | Deep learning, chapter 4