NBA Neural Net
2015-16 Update

Ben Taborsky

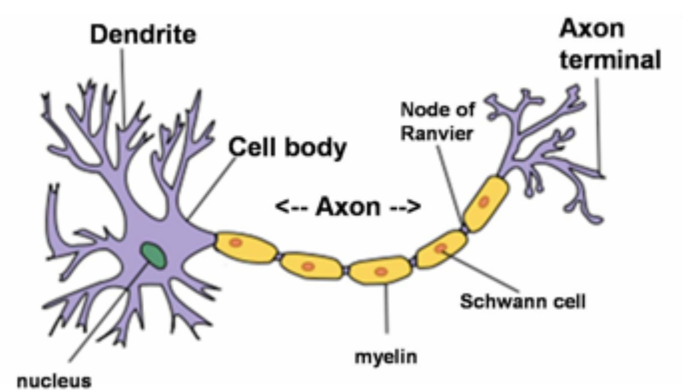
April 19, 2016

Contents

1. Review of Model

An NBA Neural Net

- About a year ago I trained a neural net to predict the outcomes of basketball games. This year, I tested it for real, betting on real basketball games with real money. These slides are an update on my experience, starting with a really quick review of the model.
- A neural net's structure is sort of based on brain biology. It is a series of connecting nodes that "fire" like neurons through their "dendrites," based on the values of inputs it receives on its "axon terminals."



Prediction	**Average CE Error**
Random Probability	1.000
50% Home Victory	0.693
60% Home Victory	0.671
Net Training Error	0.609
Net Validation Error	0.608
Westgate Money Line	0.581
Net Error on Backtest Years (adj)	0.595

- Neural networks are cool and useful because they can match any function with their nonlinearities, and they're a great way to learn about complicated patterns without imposing a lot of assumptions like you might have to with, for example, linear regression.
- It turned out to be possible to use just a relatively small amount of data to train a neural network that achieved a level of error close to, but not quite as low as, the Vegas money line.
- I thought it was close enough to wonder, if the predictions are very different, is this a glitch in the neural net or are the odds makers missing something?

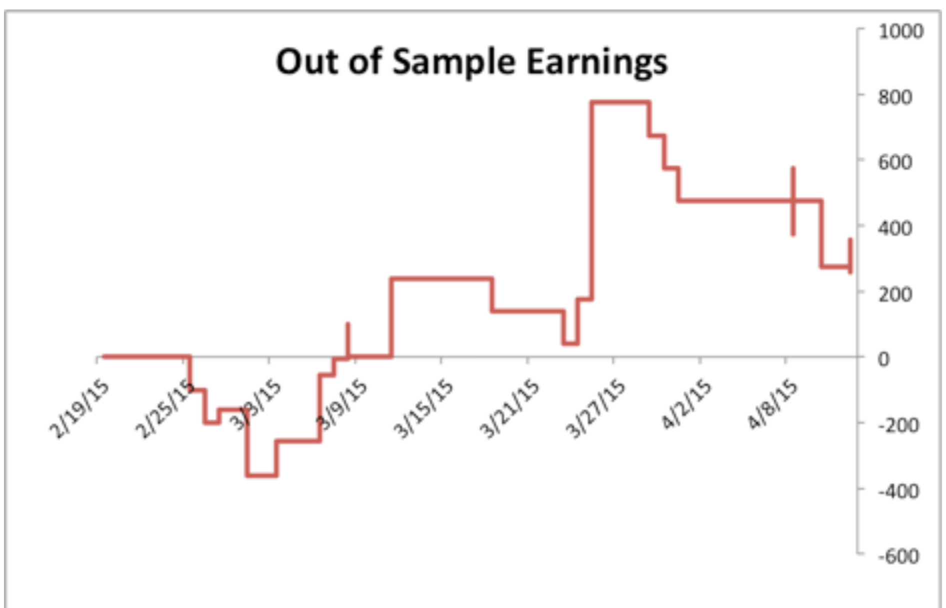
Betting Strategy Backtest

- To answer this question, I made up a strategy:
  - When the difference between the model's prediction and Vegas' prediction is greater than 15 percentage points, make a bet
  - The neural net has a hard time with very bad home teams, and never gives them less than a 27-28% victory probability. So I ignore the first rule if Vegas gives a home team less than a 40% chance.
    - Interesting note: these very improbable bets generally break even because you only have to win one or two to make up for losses, and sometimes Golden State loses at Milwaukee. But they add a lot of ugly variance.
  - The four seasons I tested *were not used* to train the neural net.
- According to these tests, the model seemed to be right often enough to be consistently profitable.

Annual Average Profit	1632
Annual Standard Deviation	962
Information Ratio	1.70
Max Drawdown	891

2014-15 Out of Sample Performance

- Of course, nobody should believe a backtest by itself, because there are all kinds of biases that could seep their way in. For example, I tried hard to come up with a good system for guessing what players I'd have known were playing before each game. But I also *really* wanted the model to work.
- So in mid-February 2015 I froze all the model's parameters and "paper-traded" using its predictions until the end of that season, trying to be as realistic as possible.
- Not an overwhelmingly positive result, but also not inconsistent with the backtest.
  - Most importantly, it didn't lose a ton of money.



2. 2015-16 Performance

Putting My Money Where My Mouth Is

- After the out of sample test was not a catastrophe, I decided to give the model a real test in the 2015-16 season.
- So, off I went to spend the winter in snowy Lake Tahoe, where I could snowboard straight from my little studio to the casino.
  - Until I broke my arm.

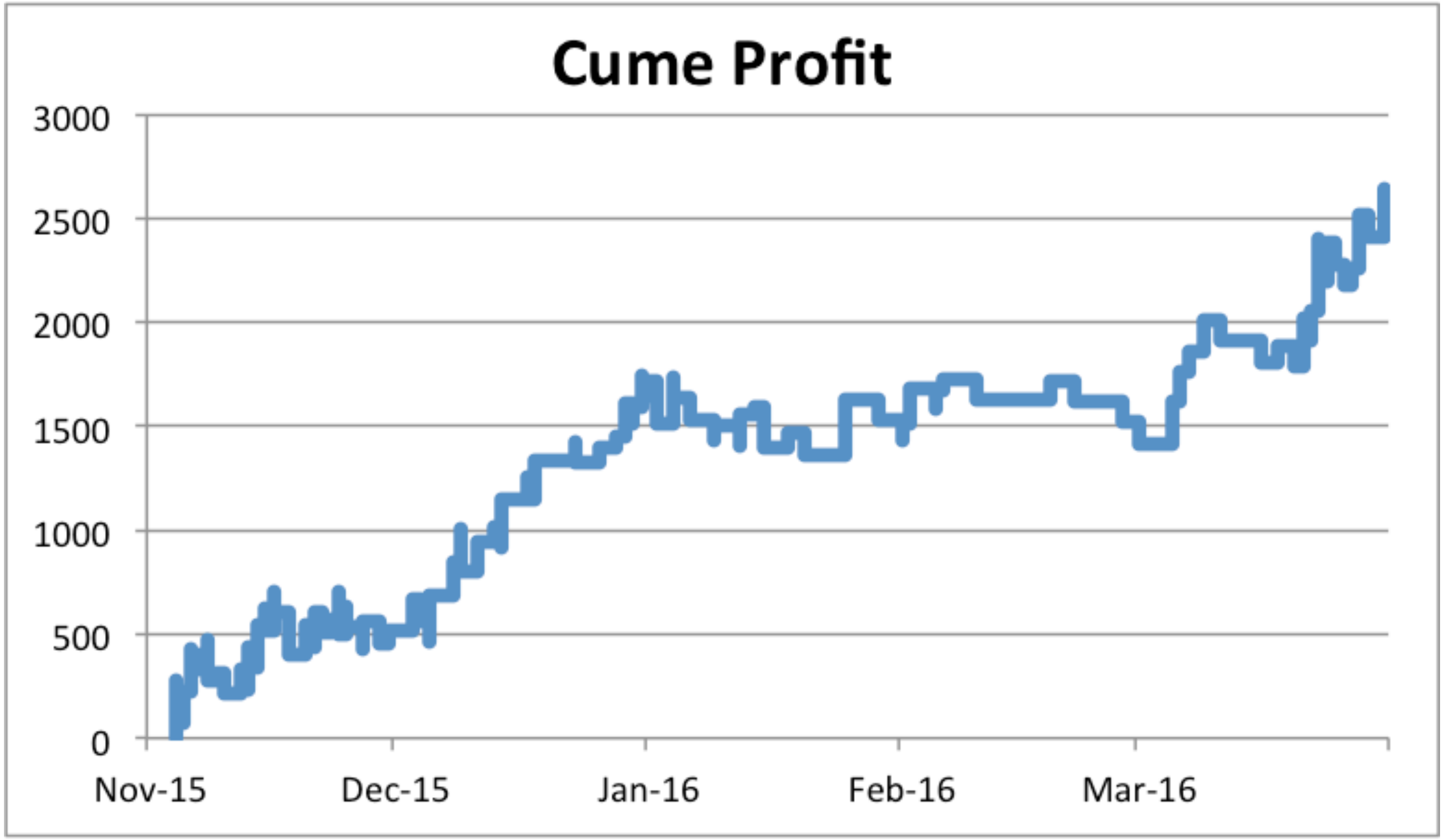
Lovely Lake Tahoe

Vicodin and sausage fingers

- I stuck with the strategy as described:
  - As long as the home team is given a 40%+ chance at victory, make a bet if the model disagrees with the line by more than 15 percentage points.
  - A few weeks into betting I realized that the model's predictions were doing very well in the 30-40% range for home victory so I lowered the floor to 30% on a provisional basis for the rest of the season (and made half-size bets, though that's not reflected in the results I show here).
  - Never, ever let personal feelings or opinions override the model. This can be hard.
    - Model, you think the Clippers have a 38% chance to beat the Warriors? Really? IN OAKLAND? Oh, dear.

Performance

With $100 bets on every model prediction, the cumulative profit for the year looks like this:



Profit	2644	Total Bets	118
Standard Dev	1326	Winning %	54.2%
IR	1.99	Max Drawdown	377

Breakdowns

- Bets are categorized according to Vegas' estimate of the home team's victory probability

Bets on Home Team

Min Prob	Max Prob	Count	Wins	Profit	St Dev	IR	Win Pct
30%	40%	22	12	1340.00	710.44	1.89	55%
40%	50%	29	13	80.00	627.67	0.13	45%
50%	60%	23	18	932.89	365.44	2.55	78%
60%	70%	17	9	-252.27	341.45	-0.74	53%
70%	80%	2	2	78.46	1.54	51.00	100%

Bets on Away Team

Min Prob	Max Prob	Count	Wins	Profit	St Dev	IR	Win Pct
50%	60%	1	0	-100.00	-	-	0%
60%	70%	11	6	430.00	443.66	0.97	55%
70%	80%	7	3	285.00	467.11	0.61	43%
80%	90%	6	1	-150.00	450.00	-0.33	17%

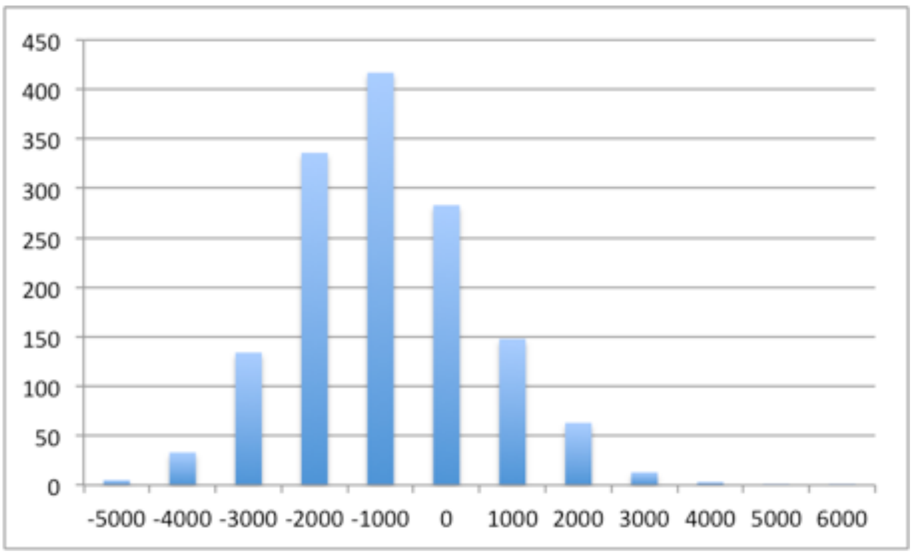
3. Analysis

Sometimes you just get lucky!

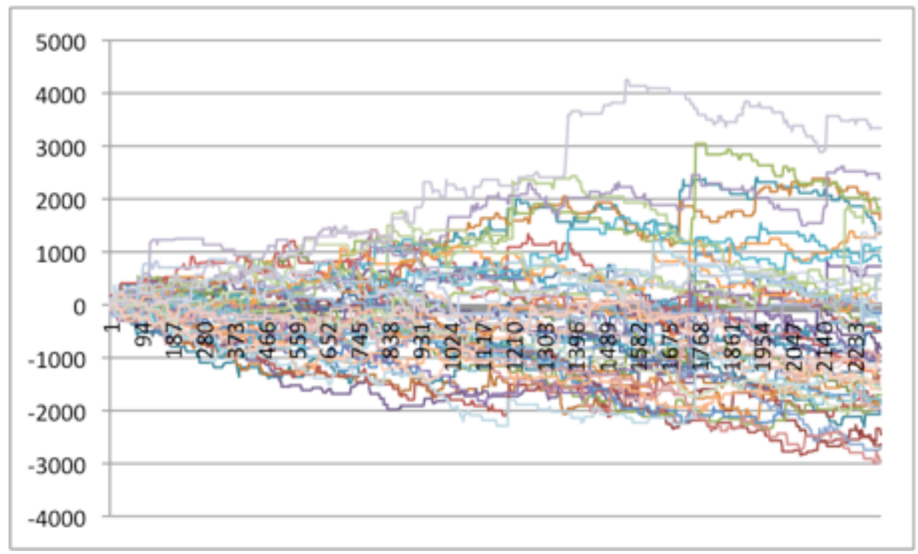
- The model's results, presented over the last few slides, look pretty good, but there is always a chance that a monkey throwing darts at an NBA schedule could do as well.

Is that what happened?

- To find out how well you could expect to do with approximately the same number of bets, but chosen randomly instead of by the model, I created 1000 sets of random bets.
  - On average they lost $500. This makes sense because Vegas takes transaction costs a little over 4%.
  - 4% transaction cost * $100 * 124 bets = $496
  - They did have huge variance, with a standard deviation of almost $1500
  - So, assuming that the model does NOT have explanatory power and the results were random, its profit of $2644 is 2.1 standard deviations above the mean, and turns out to fall in the 97th percentile
  - **In other words, it's possible but unlikely that the performance can be explained entirely by luck**



Histogram of 1000 random bet seasons
Frequency (y) vs Profit (x)
Profit shown is the bottom of the bucket



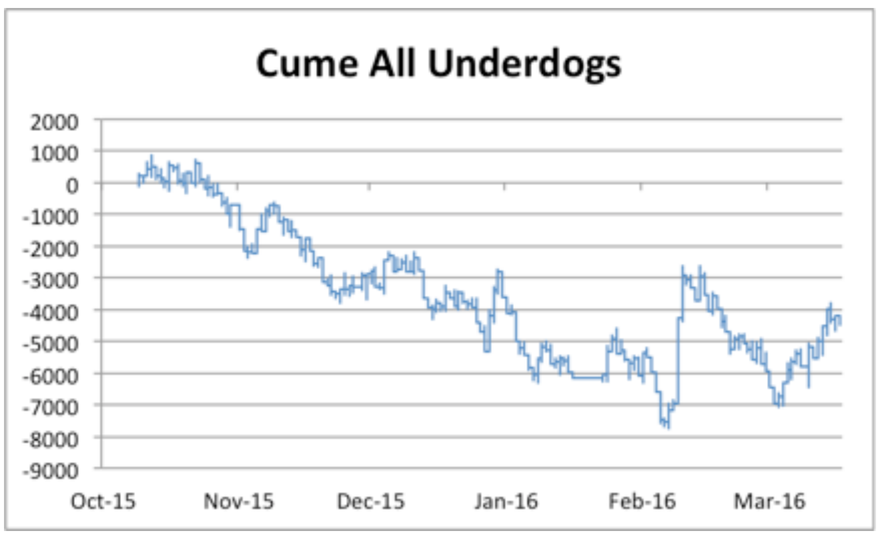
Time series of 50 random bet seasons

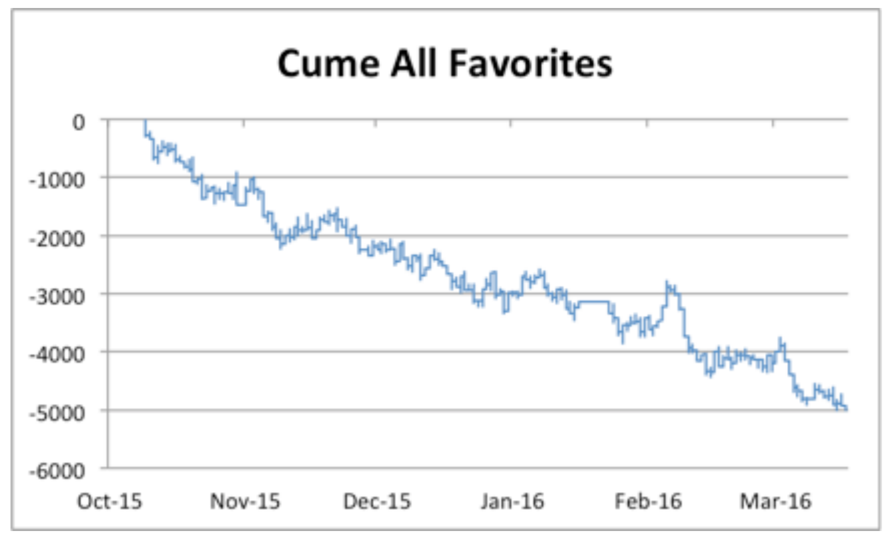
Other Possible Explanations

- The model was almost certainly better than random betting – but did it just luckily bet on the underdog or favorite a lot?
- What about certain bet-winning teams?

Favorites and Underdogs

- On the Breakdowns slide, it's pretty clear that a lot of the model's profits came from betting on underdogs. So, did it happen to do this in an amazing year for underdogs?
- The charts below show what the profit would have been like betting on the underdog or favorite in **every game**.
- Underdogs did mildly better. But saying this would not have been a good strategy is an understatement.
- ...wait a sec, why are they BOTH negative?
  - 1,157 games * 2 bets each * 4% transaction cost * $100 = $9256
  - All Underdogs Profit: -$4476
  - All Favorites Profit: -$4962
  - Total Profit: -$9438

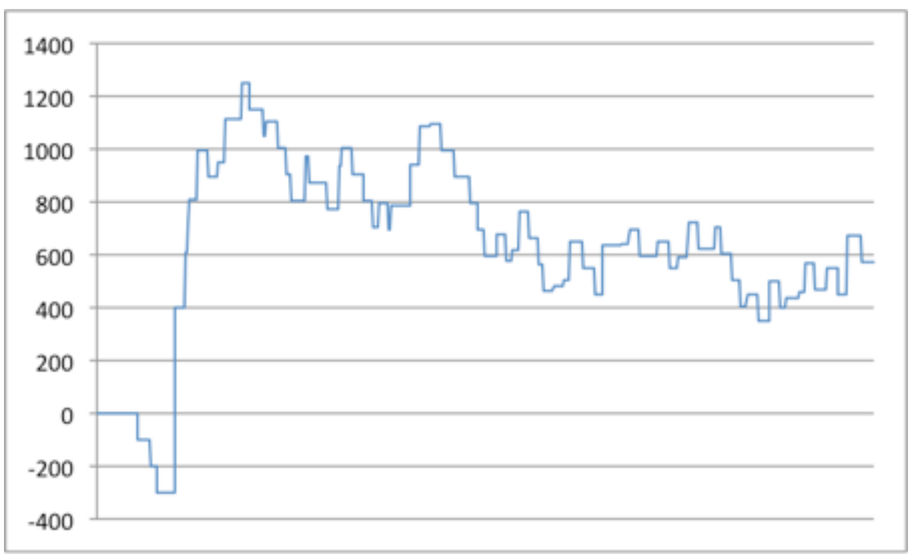




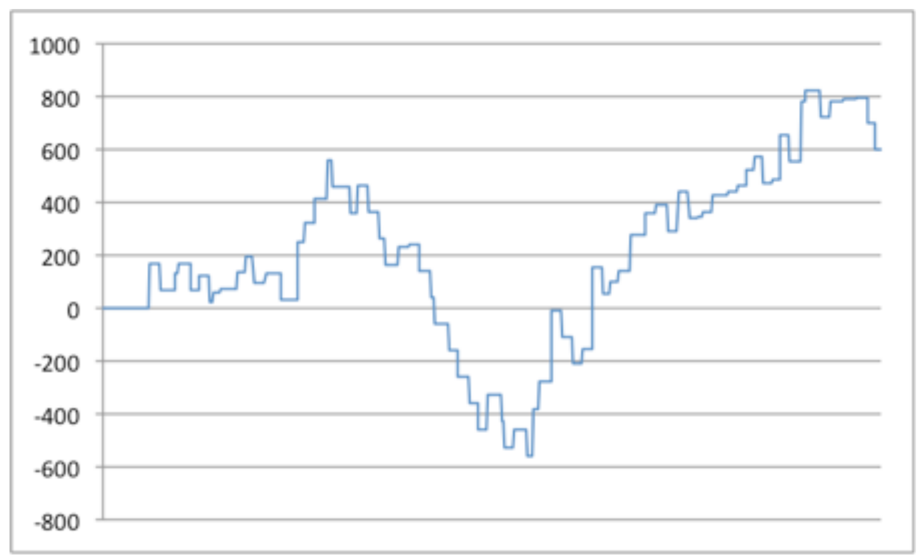
Prominent Teams

- The model made a little over half of its profits by betting against Houston 20 times and on Charlotte 15 times (actually 13 times for, twice against).
- Was this the source of the luck?
  - Both of these would have been decent bets repeated over and over all season, though with Houston the gains came all at once in the beginning.
  - With regard to Houston, the model captured three of the four big upsets that made up the early season gains. By the end It managed $816 in gains in 20 bets, beating the Always Bet Against strategy by about a third.
  - As for Charlotte, the model gained $653, out-earning the Always Bet On strategy while using many fewer bets.

Profits from 73 bets against Houston

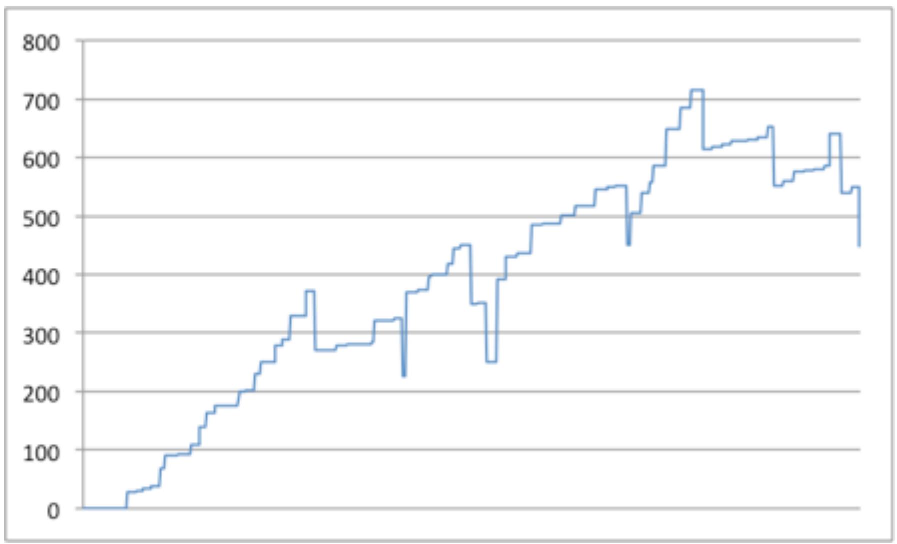


Profits from 73 bets on Charlotte

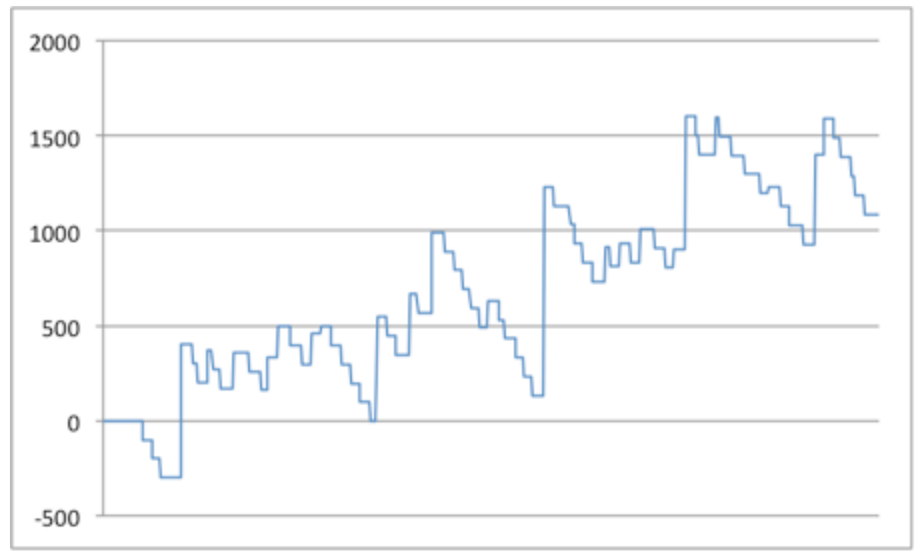


Other good bets?

74 bets on the Warriors. Not bad!



But not nearly as good as betting on the Nets...



Conclusion

- Testing the NBA neural net in a real-life situation resulted in a successful betting season!
- While the performance may or may not be replicable, it is unlikely that it was due to luck.
- Betting is very stressful.
- Approximately 60% range of motion recovered in left wrist.