# Phase 1: establishing correlation between messages and actions

First of all, I have to make sure that there is significant correlation between messages and actions for the three agents. I accomplish this using normalized linear regression with tensorflow; setting messages as inputs and actions as outputs, I follow these steps:

1: get the average error over 100 random 20% samples of the dataset (x=message,y=action) before training

2: train

3: get the error after training

4: perform a significance test with a=0.05

5: repeat for each agent and for agent 3’s message to each agent

This allows us to prove that there is significant correlation between messages and actions, meaning that the messages have a real bearing on the actions of the agents. In other words, to some degree they have learned to communicate.

# Phase 2: establishing language commonality

Having established that the messages generated are significant, I want to ensure that the agents are “speaking the same language” to agent 3. To do this, I follow these steps:

1: use agent 3’s model from phase one to evaluate 100 random 20% samples of the datasets for agents 1 and 2; in other words, seeing how well agent 3 understands the messages agents 1 and 2 sends

2: perform a significance test comparing the error generated by agent 3’s model with that generated by the other agents’ models. The null hypothesis will be that agent 3’s model predicts less well than the other agents’ models, and the alternate hypothesis will be greater than or equal to.

This way, I will be able to prove that agent 3 does understand the messages sent in by the other agents.

# Phase 3: analyze agent 3’s behavior

Having established that agent 3 sends, receives, and understands meaningful messages, we can analyze why agent 3 behaves the way it does and assess whether it has learnt to lie. To do this, I follow these steps:

1: run through the whole of data sets for agent 3’s messages to agents 1 and 2 and record the z score of every data point

2: find the data points where the z values for agents 1 and 2 are very different from each other and record the information about these instances (top 5 instances)

3: explain the relation values leading up to the message and after the message for 10 time steps. Same goes for board states. Explain potential motivations for agent 3 to lie, and conclude that it learned to do so successfully.

# Potential areas for future study

The interior of the neural networks generated; why agents remember certain things but not others, what memories stay constant in the lstm, what part of the neural network lights up during a lie versus the truth (how it can detect when it needs to lie), what parts of agents 1 and 2’s networks light up during normal messages versus lies (what can they detect?)

# Weaknesses

The agents never actually learned to play checkers, so despite learning to communicate they did not actually complete the task. I could have run the network longer had I had more time, and that might have changed behavior, as would eliminating errors in the messaging with more training. In essence the model is not fully trained and a fully trained model would undoubtably show stronger results.

Stuff I changed so far:

1 sample t test > 2 sample t test

Many different dataset slices

Assessing over all the messages once rather than lots of little batches and getting the mean and stdev from that