Machine Learning: A Brief Update

A. What is training and testing accuracy?

Training accuracy is how well the model makes predictions based on data it has already seen. Testing accuracy is how well the model makes predictions based on new data (never seen before).

B. The big questions -- why was the accuracy so high before? Why isn't it as high now? Why was the accuracy so low during the demo?

I am currently using a much larger training and testing dataset as mentioned before, but I realize there is a better reason why the accuracy is lower now…

I realize the 90% I reported earlier was the *training* accuracy of the model. Ideally, the training accuracy = testing accuracy, but this isn't always true. I suspect the model from the last demo had a high training accuracy (90% I saw locally) and a low testing accuracy (~0% during demo) (explained further in Section D). I should have recorded the training and testing accuracies of the model from the start, but I didn't know there would be such a large discrepancy between the two accuracies.

C. What's up with the model now?

As you may recall, we are running a multiclass classification neural network model to predict the angle of the door in 15 degree increments. I am recording the training and testing accuracy.

There are 2 separate predictions being produced, one based on magnitude and one based on phase data. I don't think we reached a conclusion on the best way to deal with phase and magnitude data. I had predicted that the magnitude and phase model predictions would be similar. But I have seen a difference up to 10% between the magnitude and phase accuracies, so this is not the case. Everything in this document applies to both the magnitude and phase models.

Currently, the *testing* accuracy of both models is plateauing around 40-60%, regardless of the training accuracy. Meaning, in the last few days, I have seen the training accuracy fluctuate between 40-80%, but the testing accuracy stays close to 50% regardless.

D. Why is this happening?

Again, ideally training accuracy = testing accuracy. When training accuracy >> testing accuracy, this is called “overfitting”. The model gets very used to the data it has seen in the past, and it is not as good at predicting new data. This is what is happening with the model.

E. What is the solution?

I’ve done my research on how to fix overfitting, and here’s what I tried based on the Internet’s advice: modifying the # of input layers, # of nodes, different activation functions, # epochs, # batches, graphing the accuracy and loss, etc. I have seen the training accuracy increase. But like I mentioned, the testing accuracy is still stuck around 40-60%.

It would be amazing if we had more resources other than the Internet. ML is different from regular programming because there is not one “blanket” solution that can fix a problem. Someone else’s solution on the Internet might work for their model, but won’t necessarily work for our model. So it would have been great to have an experienced person (like a professor!) to ask questions on different approaches we could take, specific to our model and context.

F. What do we have to show from the ML side?

- The multiclass neural network code (15, 30, etc degrees).
- The binary-class neural network code (open vs close).
- Spreadsheet recording all the parameters I changed and the resulting training and testing accuracy.
- Can generate graphs showing accuracy and loss metrics.
- Rationale for why we made certain decisions throughout this process (why we started with a linear regression model → why we chose a neural network → specifics about the model → etc).

G. What could have helped?

- Data preprocessing. I think preprocessing the data more before feeding it into the model could have helped increase the accuracy, not sure.
- More expertise in machine learning. I started Senior Design as a junior with basic experience in machine learning, nothing to this scale. I learned a lot in this process of choosing, constructing, and training the model. Again, it would have been great if we had some outlet for questions. In the real world/industry, there would have been SMEs we could talk to for advice. Senior Design felt very isolating in that we can’t turn to our advisor or professors for help. (I reached out to about 4 professors in total. The one professor that replied to us eventually ghosted us. lol)

H. What can we do from here?

We can go back to the binary neural network model (open vs close) so we have something to show for our project. Any other ideas on what to do?

The end. Thanks for reading.