Deep Learning Final Project Papers

Riley Fiske – CSC 380 – 1 May 2024

Housekeeping

The files to support this project can be found on GitHub, where I have a branch that is forked from the gmtanner-cord/CSC380 Repository. The discussed models (minus my Large Transfer Model h5 file that is 450mb) are in the models folder, and adapted scripts are in the scripts folder. The h5 file could not be pushed to GitHub since it was over 100mb. These models will be brought to the final exam period on Wednesday, both on USB and synced via OneDrive.

Model Performance

For the final project for this class, I tweaked different parts of the Convolutional Neural Network Model from-scratch and the Transfer Learning Model to try and optimize accuracy, true positive rates, and true negative rates for image classifications of Niblet. Different adjustment techniques were utilized for both the CNN models from-scratch and for the Transfer Learning Models, which proved to be less fruitful.

Building Convolutional Neural Network Models from Scratch

I created three different from-scratch models of this type. My first model (cnn_model_rdf_2024_04_29_model2_fixedmetrics.keras) was adjusted by changing the MaxPool2D pooling layers to AveragePooling2D pooling layers. My second model (cnn_model_rdf_2024_04_29_model3.keras) went through and added a normalization layer after every round of convolution before pooling, as well as GlobalAveragePooling2D right before flattening the layers. My third model (cnn_model_rdf_2024_04_29_model3_newaugmentation.keras) took my second model as is, and manipulated the image augmentation at the beginning further, by adding random brightness, random contrast, and gaussian noise. We will analyze the performance of these three models later on.

Transfer Learning Models and Evaluating their Performance

I noticed right away that the Transfer Learning Models' accuracy rates were much lower than the accuracy rates from the from-scratch models. I decided to beef up the Transfer Learning Model from Small to Large and observe the change in these rates using the same training and test sets. Unfortunately, the accuracy did not change much at all between the small and large models, staying consistently low on overall accuracy and an increasingly high false positive rate.

Transfer Learning (Large) CF			
		Expected	
		Niblet	Not Niblet
Predicted	Niblet	90	84
	Not Niblet	0	18

Transfer Learning (Small) CF			
		Expected	
		Niblet	Not Niblet
Predicted	Niblet	90	87
	Not Niblet	0	15

As can clearly be seen, both models have hardly over 50% accuracy, with heavy favor towards selecting false positives. Therefore, these models are not going to be considered in the final consideration for best model, and the best performing model should hopefully lie among the from-scratch models.

Evaluating Performance of the From-Scratch CNNs

The from-scratch CNNs performed much better than the Transfer Learning Models, although they also seemed to exhibit the same tendency of higher false positive rates than false negative rates:

CNN (Model 1) CF			
Accuracy: 0.7448			
		Expected	
		Niblet	Not Niblet
Predicted	Niblet	86	45
	Not Niblet	4	57

CNN (Model 2) CF			
Accuracy: 0.8854			
		Expected	
		Niblet	Not Niblet
Predicted	Niblet	81	13
	Not Niblet	9	89

CNN (Model 3) CF			
Accuracy: 0.5833			
		Expected	
		Niblet	Not Niblet
Predicted	Niblet	81	71
	Not Niblet	9	31

Surprisingly enough, Model 3, with the enhanced image augmentation features, performed much poorer than Model 2 that had less image augmentation. The same number of true positive and false negatives were observed, but Model 2 had a higher true negative rate than Model 3 (0.696 vs. 0.127). This is most likely due to the random brightness and contrast introduced into Model 3 that may be distorting the main identifiable characteristic of Niblet (his yellow, maroon, and green colors). Model 1 seems to be a nice middle ground between Models 2 and 3, and interestingly enough had the least number of false negatives and the highest number of true positives.

Final Thoughts

From the five models I created, there was a clear winner with Model 2 from the CNNs from-scratch. The normalization used in Model 2 seems to have pushed it over the edge in comparison to Model 1, its closest competitor. Further experimentation could show if the normalization caused this increase in accuracy or if other factors contributed more (GlobalAveragePooling, MaxPooling vs. AveragePooling, etc.). Regardless, it seems that the from-scratch models performed significantly better than the transfer learning models given the same training and test set.

Course Reflection

While it was rollercoaster of a semester in-terms of this course and its direction, I am overall satisfied with how it turned out in the end and feel that I was able to take away some things from the class, despite having had completed several intensive Data Science courses prior to taking this course. One of the incentives I had for taking this course was to get my hands dirty with Python in the Data Science world, and I feel that my proficiency in Python programming has increased greatly over the course of the semester, both through this class as well as through purposefully choosing Python for coding challenges in other classes. With the final project, I feel that I was able to go out and explore these higher-class methods within Python and write the code to adapt them and optimize them pretty confidently on my own, which further proved that proficiency as well as furthered my understanding of the functionality and architecture of these neural network systems, that I had only encountered before briefly in Data Mining in Spring 2022.

I see myself using the skills picked up in this class in the future frequently as I plan to pursue Data Science further after graduation (either in industry, graduate school, or both), and both Python and Neural Networks are very prevalent in the world of Data Science. I feel confident in R and those skills, so I feel good knowing now that I also have a basic understanding of how to adapt those skills into Python to accomplish the same tasks. I plan to continue building these skills this summer in my (hopeful) Data Science Internship (location pending, still waiting on two replies from places).

As for Deep Learning Models' place in society, we see their mark being made every day, especially with the popularization of large language models since the public release of Chat-GPT. I am a believer in the benefits they bring to our society, but I also recognize the dangers and risks they bring. To name a few, I believe deep learning models can enable humans to reach data/statistical insights not before reachable using big data, and it simplifies the mechanical process so that humans can spend their time doing what they're good, asking questions, reasoning, and drawing conclusions. Dangers can arise in the ability of users to oversimplify tasks that don't need simplifying which can result in cutting necessary corners in education or development due to the convenience of an all-knowing "answer key". There are many more benefits and risks to using these models, but as is the case with any new innovation, the risk/reward ratio will balance itself out over time, and we are just on the cusp of these models' full potential to change humanity forever, for better or worse.