Efficient cancer classification from tissue slide images using CNNs

DS-GA 1006: 11-16 Project Update

Joyce Wu (jmw784)
Eduardo Fierro (eff254)
Raul Delgado Sanchez (rds491)

November 16, 2017

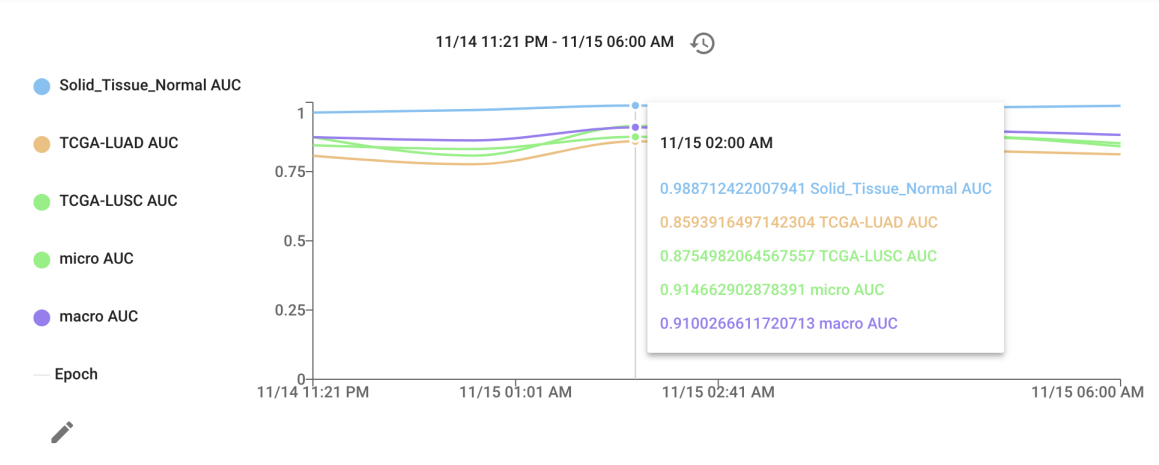
Update

Since the memo, we have now successfully implemented a function to evaluate per class AUC on a one-vs-all basis, a micro AUC and a macro AUC. This is important because our advisors used that same measure of success on their experiments and obtained a AUC of 0.97.

We have also successfully trained several models with different hyperparameter specifications. Our best result gives us a macro AUC of 0.91, which is a very good starting point.

		experiment	macro AUC ↑	File Name	Duration	Server Start T	Accuracy	Server End Ti
		Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
+	☐	joy4		train.py	04:52:39	11/8 08:30 PM		11/9 01:22 AM
+	☐	edu3_lr		train_decay_lr.py	39:57:02	11/12 04:45 PM		11/14 08:42 AM
+	☐	joy7	0.77254754483516	train.py	11:12:31	11/14 07:36 PM		11/15 06:49 AM
+	☐	joy6	0.86686363243863	train.py	12:49:05	11/14 12:12 PM		11/15 01:01 AM
+	☐	joy12	0.86842078585402	train.py	18:26:03	11/15 04:31 PM		11/16 10:57 AM
+	☐	edu4	0.87729027038281	train.py	24:20:37	11/15 11:43 AM		11/16 12:04 PM
+	☐	joy9	0.88252050837663	train.py	08:10:40	11/14 09:49 PM		11/15 06:00 AM
+	☐	edu5	0.90054836475341	train.py	23:45:59	11/15 12:22 PM		11/16 12:08 PM
+	☐	joy8	0.90840534929037	train.py	22:32:05	11/14 09:36 PM		11/15 08:09 PM

It appears that our models are achieving a very high AUC (~ 0.99) for classifying normal tissue vs cancerous tissue, but it is not doing as well telling apart TCGA-LUSC from TCGA-LUAD. This is consistent with our advisors' results as well.



We still have several experiments to run on this same model (a 4 layer CNN), mainly with different initializations, learning rate, optimizers, dropout rate, data augmentation, aggregation methods, etc. Most of these options are already implemented, so we just need to submit the jobs. We may also modify the architecture with more layers to see if it boosts performance.

In parallel, as our advisors requested us to replicate their previous work, we have a running version of Google's Inception V3 model implemented in PyTorch. We have the option to train it with pre-trained weights for all layers except the last layer (transfer learning, only updating the weights on the last layer), initialize it from the pre-trained weights but train all layers, or train the whole model from scratch. We have not yet implemented it into the main training code, which includes the functions to perform aggregate prediction and calculate AUC. We expect to have this implemented by the next meeting with our advisors, Tuesday of next week.

Finally, for the other types of cancers we wanted to try our model on (breast cancer and kidney cancer), we have run into problems tiling the images. In particular, the problems are related to memory and time limit issues with our jobs submitted in Prince. One of our advisors has offered to help us look into the problem. We expect to resolve these problems by Tuesday.