BIOENG 245

Project Part 2

Catalina Villouta

May 2022

In part 2 of the project I integrated the encoder -which was trained and analyzed in part 1- and coupled it with additional layers for performing a supervised classification task. Both the encoder and the additional layers were done using Tensorflow, allowing for a clean model and set up. The encoder parameters could have been refined for the classification task, but since the encoder had 100k parameters already and the sample had 700 data points, I decided to fix the encoder parameters and train only the additional layers. This is in fact an advantage of the autoencoder set up, which never *sees* the labels, hence the risk of overfitting is greatly reduced. Plus, the funnel-like architecture of the encoder-decoder with the added dropout in this project further reduces overfitting the train set. In fact, the encoder proved to be highly useful for the supervised training task. Without any meaningful dimensionality reduction, trying to forecast 10 possible labels using 700 samples and 765 features would have been an extremely difficult problem for classic ML and an impossible one for OLS. Regardless of the apparent mission impossible, I was able to build two models that leverage the encoder for predicting cell types. One I call the Small model, which adds only 1 output layer with previous dropout, and the other I call the Larger model, which has a fully connected hidden layer -also with previous dropout- in between the encoder and the final output layer. Both models perform exceptionally well, with similar performance using cross validation and the test set, as seen in Tables 1 and 2, with an accuracy of almost 80%. In this particular case I show here, the Small model has a slightly better performance, but changing the random state can flip the throne either way. So I tend to prefer the smaller, simpler model. In Figure 1 I show the confusion matrix of the Small model's prediction in the test set. The model does exceptionally well on the more numerous classes, and sees FN and FP for less numerous ones. Thus, a possible improvement here would be to take into account data imbalance in the modeling. Figure 2 shows the ROC curve and ROC AUC of the most numerous cell types. Again, results are excellent.

	accuracy	precision	recall	f1	roc_auc	categorical_crossentropy
small	0.798	0.643	0.659	0.645	0.952	0.641
larger	0.782	0.636	0.642	0.622	0.962	0.618

Table 1: Cross Validation Results

	accuracy	precision	recall	f1	roc_auc
small	0.829	0.703	0.684	0.684	0.971
larger	0.850	0.715	0.701	0.689	0.967

Table 2: Test Set Results

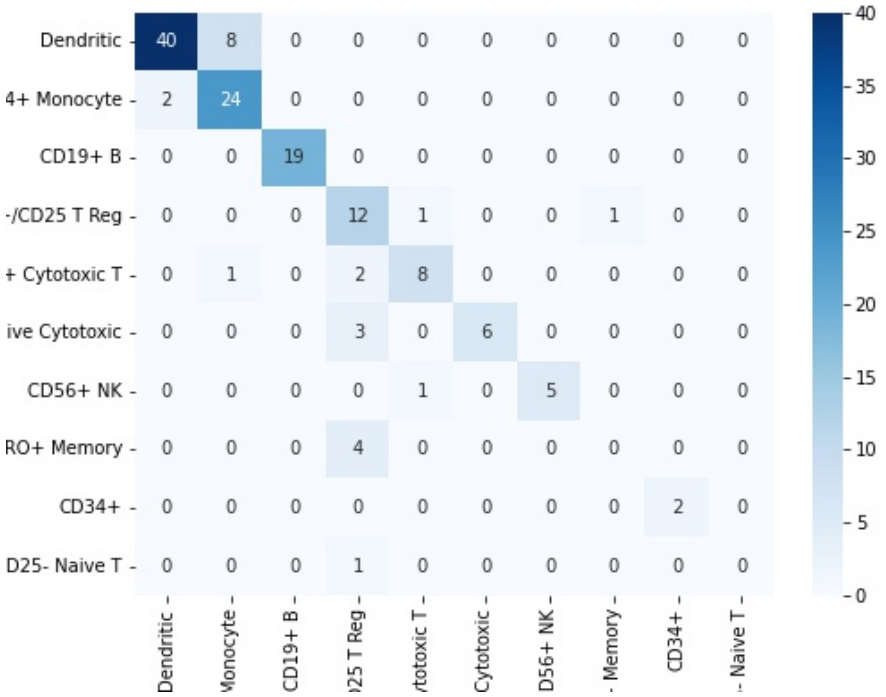

Figure 1: Confusion Matrix Small Model

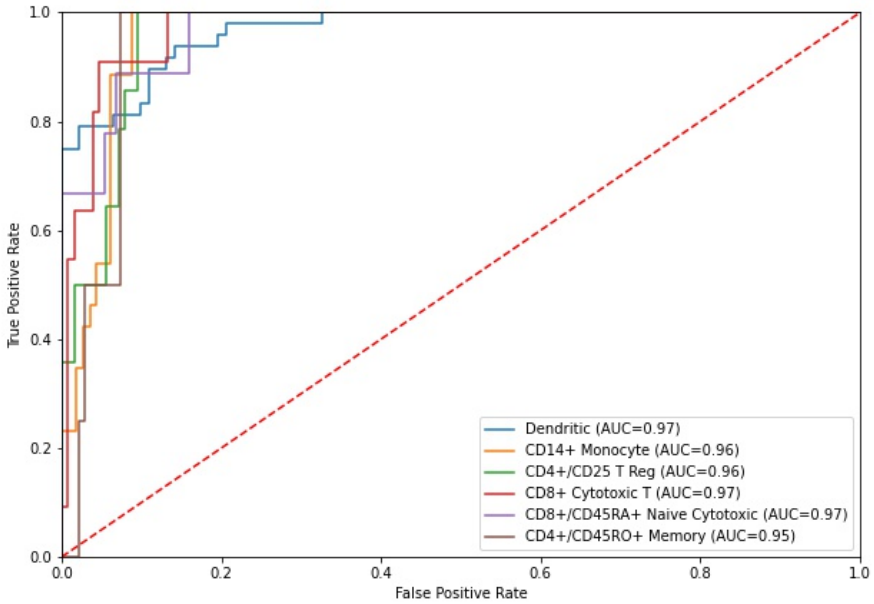

Figure 2: ROC Curve Small Model