Deep Learning Image Classification

Tony Liang

2024-04-06

Metrics Evaluation

> What metrics did you use for evaluating your networks, and why? How would you expect your chosen metrics to behave in a dataset that has mostly negative labels (i.e. very few tumor cell annotations)?

Given the classification problem is binary (two classes), hence I used various classic binary classification metrics like accuracy, precision, recall, and f1-score. These metrics all measures different aspect of correctly predicted tumor images, or classified tumors/non-tumors (Lever, 2016). Moreover, they're easily accessible through python library scikit-learn (Pedregosa *et al.*, 2011), hence selected to evaluate the networks.

Out of these metrics, accuracy might be the metric that is most affected when labels are mostly negative. This is because accuracy represent the following:

$$\text{accuracy} = \frac{TP + TN}{P + N} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

Then, if one dataset has mostly negative labels, the accuracy could still be very high, given it could classify these negative labels almost as perfect, while not considering the imbalanced distribution of labels, hence misleading metric. That is also why we need to introduce other metrics like precision and recall to measure correct positive identifications or actual positives that were identified correctly.

Classification results

> How did your results differ when testing your initial trained network on the training dataset and the two test datasets? Why do you think this happened?

ResNet18 Baseline

We first train the network with the slide images with resnet (He *et al.*, 2016) as its baseline under the pytorch framework (Paszke *et al.*, 2019), then tested this trained network against each of our training dataset, test dataset 1, and test dataset 2. We would expect that the evaluation on the training dataset has the highest performance, given we are evaluating on the same data that a network or a model was trained on. I think this step is more like a sanity check, hence we should not only rely on the performance on the training set.

Then, the performance on those test datasets varied. Test data 1 had higher metrics given the images collected are more balanced (i.e. having almost equivalent numbers of images with and without tumour cells annotation). But, test data 2 dropped drastically compared to test data 1, because of its uneven distribution of labels (very few tumor cell annotations). Lastly, we expect these test datasets should have lower performance than on the train dataset, as these serve the purpose to "generalize" or estimate how well our model would work on future unseen data.

Table 1 summarizes the classification metrics on each of the datasets evaluation on our trained network with baseline as resnet18 and default hyperparameters of learning rate $1e^{-4}$, batch size of 32 and number of epochs 5.

Table 1: Baseline Results of ResNet18 classification performance on tumor datasets

Dataset	Accuracy	Precision	Recall	F1 Score
Train data	0.920	0.948	0.889	0.917
Test data 1	0.914	0.938	0.909	0.923
Test data 2	0.750	0.676	1.000	0.806

Table 2: Baseline Results of ResNet50 classification performance on tumor datasets

Dataset	Accuracy	Precision	Recall	F1 Score
Train data	0.989	0.987	0.991	0.989
Test data 1	1.000	1.000	1.000	1.000
Test data 2	0.521	0.521	1.000	0.685

ResNet50

Other than the initial baseline, I also tried to train network using another backbone, resnet50 (He *et al.*, 2016). Its corresponding results are shown at table 2.

Hyperparameter Results

> How did the performance on the two test sets change when you modified hyperparameters and used a cross-validation strategy to choose the best model?

The previous results were sort of baseline using default and suboptimal hyperparameters, hence we could access model performance on the two sets using optimal hyperparameter sets. These could be estimated by implementing a cross-validation (cv) strategy (Berrar *et al.*, 2019) into our train and evaluation of network. That is, for a grid of hyperparameter a, $a_1, ..., a_n$, hyperparameter b, $b_1, ..., b_m$ or more, run cross validation on each combination of these hyperparameters and get their mean score of folds (using accuracy to rank these), and find such combination that achieves highest mean accuracy score. Then, these set of parameters could be used to train a "best" model and evaluate on all datasets again.

Table 3 is summary of best model performance on each dataset with optimal hyperparameters

Table 4 that illustrate the mean cv score of each hyperparameter combination. I tuned epochs and learning rate.

Differences in performance

> Hypothesize why there is a difference in performance between the performance of the model on test set 1 and test set 2 in your 'Baseline results' table. Hint: do the differences go away with your attempts in #2 and #3?

The different performance across two test dataset could be due to suboptimal hyperapameters, like too

Table 3: Hyperameter Optimal Results of ResNet18 classification performance on tumor datasets

Dataset	Accuracy	Precision	Recall	F1 Score
Train data	0.998	0.996	1	0.998
Test data 1	1.000	1.000	1	1.000
Test data 2	0.521	0.521	1	0.685

Table 4: Mean cross-validation accuracy for hyperparameter combination pair of ResNet18

lr	epochs	mean_cv_score
0.001	9	1.000
0.010	5	1.000
0.010	7	1.000
0.010	3	0.998
0.100	3	0.998
0.001	3	0.996
0.001	7	0.996
0.010	9	0.996
0.100	5	0.996
0.001	5	0.993
0.100	9	0.976
0.100	7	0.971
0.000	7	0.962
0.000	9	0.960
0.000	5	0.876
0.000	3	0.798

few rounds of training so model could not learn all patterns yet, or learning rate being too small/big that optimizes slowly or skips minimas. Moreover, data quality also matter, it could be due to poor quality or similar images in both normal and tumor images. Specifically, the images of test data 2 had a higher hue of coloring, such that its background color (pink red? fuchsia?) is very similar to the tumor annotation (red), hence make this hard to classify and reducing performance upon evaluation.

With optimal hyperparameters, the performance on train and test 1 have all bumped up compared to using those default ones (or inital ones) for our trained model with backbone resnet18. But, test 2 had a worse performance now.

Future dataset attempt

> What would you expect to happen when you try this on a dataset from another research center? What types of challenges do you expect to encounter?

It is hard to tell if our trained network can perform equally well with future datasets from another research center, most likely it would perform poorly, given the differences in histology and annotations from centres. And, other factors like batch normalization and technology differences could also affect the data quality, thus affecting on the classification performance as well. In general, we would need to train on more datasets, so it could generalize.

References

Berrar D. & others (2019). Cross-validation.

He K., Zhang X., Ren S. & Sun J. (2016). Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p. 770–778.

Lever J. (2016). Classification evaluation: It is important to understand both what a classification metric expresses and what it hides. Nature methods 13 (8): 603–605.

Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L. & others (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems 32.

Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V. & others (2011). Scikit-learn: Machine learning in python. the Journal of machine Learning research 12: 2825–2830.