CSE 5194 Class Project

Subash Chebolu

Chebolu.1@osu.edu

1. Introduction

The purpose of this project is to provide technical insights on how certain machine learning models perform against a particular dataset introduced later. This project should enhance the understanding of certain performance aspects that are tied to the combination of model and data and will provide a certain hypothesis on the results. Although the scope of this project is quite small, due to it pertaining to a single dataset and specially handpicked models, it should provide some knowledge on how to approach larger questions within this field.

2. Data

The data used in this project is a diagnostic breast cancer dataset providing certain variables of several cases of breast cancer tumors. The dataset has 569 samples of breast cancer data with 32 variables associated with each sample. The id variable which is a unique identifier for each tumor is an integer and the diagnosis variables which indicates whether a tumor is malignant or benign is a character either ‘M’ or ‘B’ respectively and the remaining 30 variables are all float values that are

numerical measures of certain aspects of the tumor. The variables in the dataset by name are: id, diagnosis, radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_mean, compactness_mean, concavity_mean, concave points_mean, symmetry_mean, fractal_dimension_mean, radius_se, texture_se, perimeter_se, area_se, smoothness_se, compactness_se, concavity_se, concave points_se, symmetry_se, fractal_dimension_se, radius_worst, texture_worst, perimeter_worst, area_worst, smoothness_worst, compactness_worst, concavity_worst, concave points_worst, symmetry_worst, and fractal_dimension_worst. The variables are best explained by the data dictionary provided by the Kaggle website [1]. The variables are meant to explain statistical distributions of certain aspects of a tumor since a mean value, standard deviation and worst value are given for the different aspects measured from the tumor. The only preprocessing required to get this data into the right form for the models was the random sampling of 80-20 where 80% of the data would be training data and 20% would be testing data and the encoding removal of the id column in the data so that the network doesn't just remember the value based on each samples unique identifier and lastly the encoding of the 'B' and 'M' characters in the diagnosis column for benign and malignant tumors to 0 and 1 respectively for classification.

3. Models

The models that were used for experimentation in this project are certain handpicked models that seemed to align well with the data type. All the models for this project were built in python using the PyTorch, Numpy and pandas libraries primarily [2].

The first model is a dense, fully connected model where the input vector is fed into a dense layer of variable size to which a sigmoid function is applied after which the output is sent to another dense layer of size 2 which flows into another sigmoid and lastly a log softmax function is applied for classification. The second model is a 1-dimensional cnn model where the initial dense layer of the dense model is switched out with a 1-dimensional convolutional layer with kernel size 5 and variable filters. The filters are then flattened out and the remaining functions and layer are the exact same as the dense model. The reason these two models were picked in particular is due to these models working well with the 1-dimensional data vector given for each sample. The lack of a temporal dimension and nonsequential structure of the data showed that any type of recurrent network would be unfit for this data. The models' accuracy was measured as a percent of samples in the testing data that were predicted correctly. Lastly, the optimizer used for both models was the adadelta optimizer with a learning rate of 0.1 and the loss function was the negative log-likelihood loss.

4. Results

The following chart indicates the progression of accuracy as the training epochs increased for the dense model with a hidden layer size of 100.

Epoch # - dense model	Accuracy - percent predicted correctly in test data
50	0.675438596491

100	0.675438596491
150	0.859649122807
200	0.877192982456
250	0.868421052632
300	0.877192982456
350	0.894736842105
400	0.894736842105
450	0.90350877193
500	0.912280701754

The below chart indicates the progression of accuracy as the training epochs increase for the cnn model with an output feature size of 100.

Epoch # - cnn model	Accuracy - percent predicted correctly in test data
50	0.640350877193
100	0.763157894737

150	0.789473684211
200	0.824561403509
250	0.842105263158
300	0.842105263158
350	0.842105263158
400	0.842105263158
450	0.842105263158
500	0.859649122807

5. Analysis

The chart above show that the accuracy did indeed go up as the number of training epochs went up showing that the model was learning and that it was doing a better and better job of generalizing to the dataset since the testing data is different from the training data. The most interesting detail here is that even though the initial accuracy values at 50 epochs are comparable at about 66%, the final accuracies are significantly different from the dense model predicting nearly 5% more sample points correctly than the cnn model. This shows that the cnn model either requires more training or is not a good model for the data being delivered to it. This is further accentuated by the fact that

the accuracy seems to stagnate for the cnn model between 250 to 450 epochs where it is possible it was struggling to gather more detail and generalize better whereas the dense model seemed to be learning much quicker and didn't saturate it ability to learn more and generalize better about the data through the entire run. Overall, the dense model was a better fit and worked better for this data than the cnn model. A reason for this could be that the localized nature of a convolutional layer where it only computes based on the surrounding neighbors of data is detrimental when working with this dataset since that doesn't explain as much about the data as a fully view on the data like a fully connected, dense layer.

6. Future Works

The future direction of this work could involve larger models instead of a single key layer and a broader set of data. Being able to apply the knowledge discovered here to many datasets and see if the hypothesis created here and the results found here are similar in different contexts and not only for this data would be very beneficial in discovering techniques on how to create the best model for certain learning problems.

7. Conclusion

The experiments here determine the effectiveness of a dense model and a cnn model on a breast cancer tumor dataset. The results show that the dense model is better at generalizing and learning about this data than the cnn and some hypotheses

are stated as to why this is the case. Future work should focus on bringing light to these types in a general context such that better understanding of machine learning models and their interactions with data could be better explained.

8. References

[1] “Breast Cancer Wisconsin (Diagnostic) Data Set.” *Kaggle*, 25 Sept. 2016, www.kaggle.com/uciml/breast-cancer-wisconsin-data.

[2] Chebolu, Subash. “subashch6/519401_ClassProject.” *GitHub*, github.com/subashch6/519401_ClassProject.