COMP 6321 Machine Learning Project

Hao Lei Yiming Ren Yishi Wang

Abstract

This report investigates two questions. First, for a given selection of data sets, can we say what is the 'best' classifier or the 'best' regressor in terms of good predictions? How much does the answer depend on the particular selection of data sets? How much does the answer depend on our computational constraints? We investigate these questions using data sets from the UCI repository. Second, we compare the interpretability of a decision tree classifier to that of a convolutional neural network. We compare the decision tree visualization to 'activation maximization', a technique to gain insight into the kinds of inputs that deep neural networks respond to. For high dimensional data set CIFAR10, we performed PCA and t-SNE embedding that project those features on a 2D space to visualize high-dimensional image data.

1. Introduction

For regression and classification tasks, different machine learning model performs differently in terms of training speed, accuracy, simplicity, interpretability and even the fairness of machine learning. In this project, we considered all first four metrics and trained 8 types of classification models and 7 types of regression models on various data sets from the UCI repository, the classifiers were evaluated by ROC, PR curve and regressors by MSE, R^2 score. By doing this pair-pair experiments, we found some models that perform well on all data sets and relatively easy to be train.

Furthermore, we attempted to interpret the decision tree and CNN models that were used for the CIFAR10[1] image labeling task. For decision tree, we plotted the tree structure with a max depth of 5 and for CNN we generated the activation maximization image using gradient ascent, which visualizes the expectation of CNN for a given class label.

In the end, for our novelty component, we tried to project the CIFAR10 images to 2 dimensions by doing PCA with 50 components and t-SNE embedding, provided an overall understanding of how these high dimensional features correlate with each other in a lower 2D space.

2. Methodology & Experimental Results

For data cleaning and pre-processing, we used Pandas data frame to load and analyze data. To deal with missing data, an `Imputer` that accepts different imputation strategies(i.e. `mean`, `most_frequent`) was built to handle different types of data set. The features and target then were encoded by `Encoder` and ready for training. Before the data was passed to train the models, we normalized it using `MinMaxScaler` that transforms the features by scaling them to a given ranges, in order to achieve a better prediction accuracy.

2.1. Classification Experiments

Ten different datasets were used for classification: 1. diabetic retinopathy; 2. default of credit card clients; 3. Breast Cancer Wisconsin; 4. Statlog (Australian credit approval); 5. Statlog (German credit data); 6. Steel plates faults; 7. adult; 8. yeast; 9. thoracic surgery data; 10. seismic-bumps.

These models were trained by using RandomizedSearchCV function with cross-validation fold size 3 to speed up the process, after 10-30 iterations the best estimator was returned and was used to predict the target. For classifier evaluation, we used AUC-ROC and AUC-PR curves. ROC is a probability curve and AUC represents the degree or measure of separability. AUC-ROC tells how much model is capable of distinguishing between classes[4]. But when the data set is imbalanced, AUC-PR is more suitable since it is more sensitive to imbalanced data thus performs better in the test set. The corresponding metric was selected according to the characteristics of the datasets.

As can be seen from Table 1, Random Forest and Logistic regression models performed the best with very high overall AUC-ROC, and also they have the advantage of the simplicity to understand and train.

For multiple class classification problems such as dataset 6 and 8, we used micro AUC-ROC metric to evaluate models as shown in Figure 1.

2.2. Regression Experiments

Classifications were carried out on another 10 datasets: 1. wine quality; 2. communities and crime; 3. QSAR aquatic toxicity; 4. Parkinson speech; 5. Facebook metrics; 6. bike sharing; 7. student performance; 8. concrete com-

Model \ Data	1	2	3	4	5	6	7	8	9	10	11	12	Average
KNN	0.70	0.72	0.99	0.99	0.55	0.50	0.69	0.91	0.87	0.92	0.54	0.62	0.75
SVM	0.79	0.71	1.00	1.00	0.59	0.55	0.75	0.97	0.90	0.94	0.50	0.51	0.77
Decision Tree	0.60	0.59	0.93	0.92	0.55	0.55	0.62	0.83	0.76	0.73	0.54	0.58	0.68
Random Forest	0.75	0.74	0.99	1.00	0.51	0.63	0.79	0.97	0.91	0.94	0.68	0.76	0.81
AdaBoost	0.72	0.75	0.99	1.00	0.64	0.57	0.74	0.75	0.91	0.84	0.61	0.79	0.78
Logistic Regression	0.83	0.71	1.00	0.98	0.74	0.56	0.76	0.95	0.85	0.93	0.58	0.75	0.80
Gaussian Naive Bayes	0.68	0.71	0.98	0.98	0.53	0.60	0.76	0.92	0.85	0.71	0.63	0.75	0.76
Neural Network	0.75	0.70	0.99	0.97	0.74	0.46	0.71	0.97	0.87	0.94	0.39	0.70	0.77

Table 1. Classification Result - ROC.

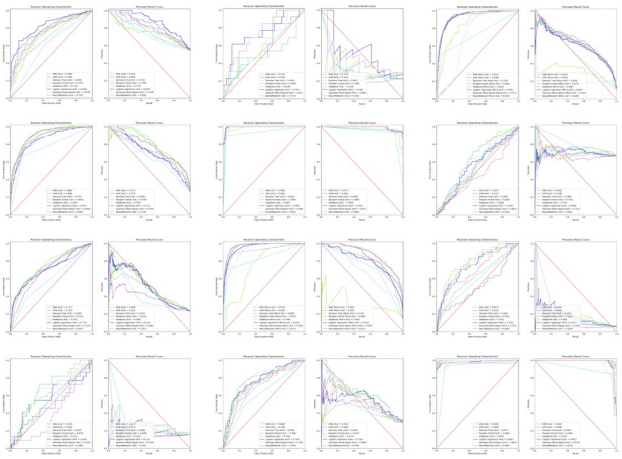

Figure 1. Classification result ROC & PR.

pressive strength; 9. SGEMM GPU kernel performance; 10. Merck Molecular activity challenge.

The process of training regression models is basically the same as training classification models. To evaluation the regressors, we used Mean-Squared-Error and R-Square score, R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression[3].

Among all these models, Random Forest and Neural Network are the best models with average R^2 score of 0.64 and 0.58. These two models also have the lowest average MSE with 20073.03 and 24494.43 respectively, the results are shown in Figure 2.

In the Merck Molecular Activity Challenge data set, there are 5878 total features but most of them are 0. For such a sparse matrix, we used PCA to reduce the dimension further to only 15, the result was amazingly good as the best model SVR has 0.622 R^2 and only 0.518 MSE as shown in Table 2.

Figure 2. Regression result MSE. & R^2.

2.3. Interpretability Experiments

This part we compared the interpretabilities of two types of models: Decision Tree and CNN. CIFAR10 vision dataset[1] was used to train and test those models.

2.3.1 Data Preprocessing

The CIFAR10 data was downloaded from the official website, loaded from disk and unpickled as numpy arrays. 5 batches of training data were combined into a 50000×3072 `np.uint8` feature array, and a 50000×1 `np.int32` target array. Likewise, one batch of testing data formed another 10000×3072 numpy array, and a 10000×1 target array. One 10×1 category names array was built from metadata.

For decision tree, in order to decrease the complexity of features, we took one step future and reduced the feature dimensions to 50 with PCA.

Each row of the feature arrays is one image. To rearrange them into *(N, C, H, W)* shape to feed into CNN, `reshape(-1,3,32,32)` was applied to both training and testing feature arrays. In addition, the RGB pixels value in the range [0, 255], we rescaled them into [-1, 1] by applying `Normalize()` of torchvision, the dtype now

Model \ Data	1	2	3	4	5	6	7	8	9	10	11	12	Average
SVR	0.35	0.46	0.63	0.51	0.36	0.14	0.46	0.84	0.79	0.21	0.07	0.44	0.44
Decision Tree	-0.06	0.03	0.22	-0.1	0.96	-1.02	0.73	0.68	0.78	0.99	0.34	0.08	0.30
Random Forest	0.41	0.47	0.64	0.4	0.93	0.32	0.85	0.81	0.9	0.99	0.47	0.53	0.64
AdaBoost	0.31	0.25	0.49	0.37	0.83	-0.18	0.54	0.75	0.76	0.68	0.51	0.37	0.47
Gaussian Process	0.32	0.45	0.44	0.47	0.39	0.17	0.7	0.82	0.21	0.78	0.08	0.37	0.43
Linear Regression	0.32	0.25	0.64	0.47	0.39	0.17	0.34	0.82	0.64	0.41	-0.03	0.37	0.40
Neural Network	0.32	0.31	0.63	0.47	0.78	0.16	0.85	0.82	0.89	0.99	0.42	0.37	0.58

Table 2. Regression result - R^2.

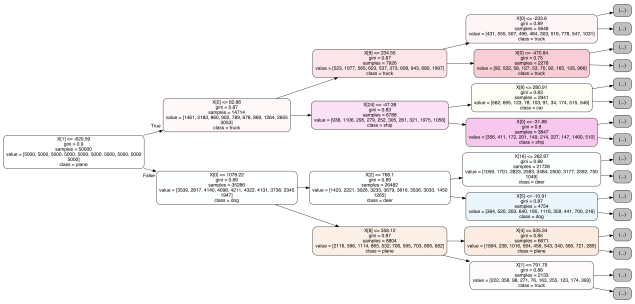

Figure 3. Decision tree.

is `np.float32`.

2.3.2 Decision Tree

The decision tree was trained using PCA output data with 50 features, the prediction score was increased from 0.243 to 0.277 due to dimension reduction. We plot 3 depth of the tree for interpretation shows as Figure 3.

2.3.3 Convolutional Neural Network

At first, a CNN with two convolutional layers of both 64 filters were constructed, followed by 2 fully-connected linear layers. After running for a few epochs, the accuracy rate still could only reach approximately 60%.

Then attempts were made to adjust the filter numbers and kernel sizes. However, it showed that adjusting the CNN parameters only without changing the network architecture could not boost the accuracy to the next level.

In the end, a CNN with 3 convolutional layers followed by 4 fully-connected linear layers was selected, with incremental filter numbers 32-64-128. Parameters search was also made to search for the optimized filter size, padding and stride values. The final CNN architecture is shown as Figure 4. As for hyper-parameters, the training rate was 0.001, momentum was 0.9, and weight decay was 1×10^{-3}. The model was trained with 30 epochs in one hour.

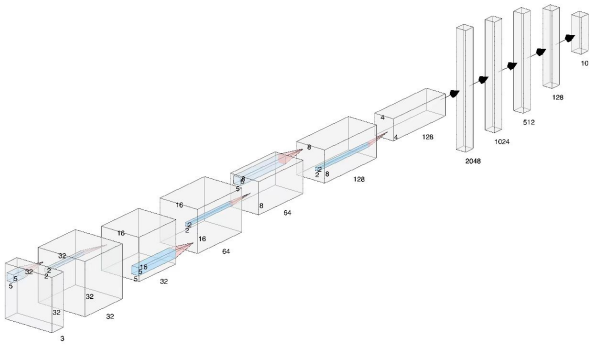

Figure 4. CNN architecture.

Category	Accuracy	Category	Accuracy
Airplane	85 %	Dog	64 %
Automobile	89%	Frog	84 %
Bird	68 %	Horse	79 %
Cat	67 %	Ship	88 %
Deer	74 %	Truck	82 %

Table 3. CNN accuracies by categories.

This model reached an overall accuracy of 78.68% after 30 epochs, 1 hour trained. The accuracies by categories are listed in the Table 3.

2.3.4 Interpretability

Interpretability can be defined as the degree to which a human can understand the cause of a decision[2]. Decision tree is usually a very intuitive type of model for human to navigate through and understand the decision paths. However, in the case of image inputs, the input features are very long array of pixel values, no longer text or numbers that a human can easily interpret. On the other hand, neural network seems more like a black box and very difficult for us to understand how the decisions are made. However, by using activation maximization technique, we could visualize

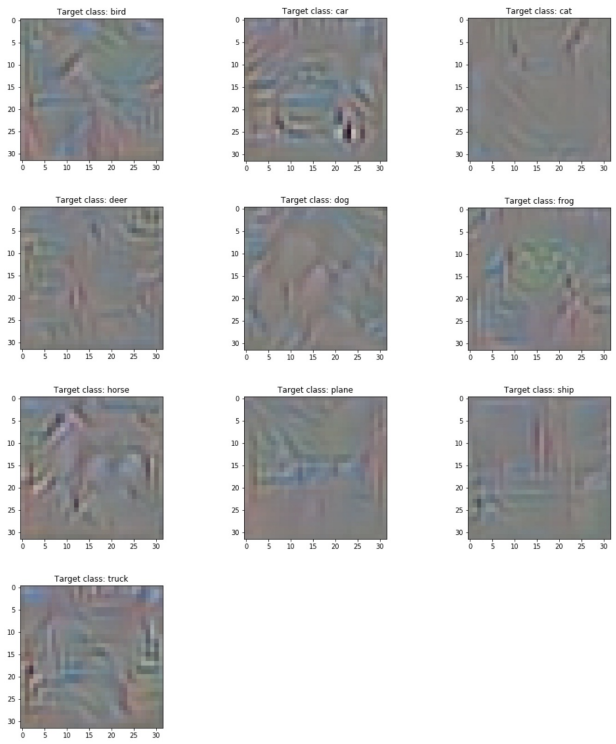

Figure 5. Activation result.

the input that maximizes the activation of a specific class categorization.

For example, the Figure 5 shows the maximized activation of every class. The images were visualized by performing "gradient ascent" with the trained CNN model.

Whereas for decision trees, it is really difficult for someone to visualize the decision made at X[2349]$\leq$144.5 for example. Especially after our PCA dimensionality reduction step, to interpret the decision paths of a decision tree becomes nearly impossible.

In summary, the interpretability of the decision tree is usually very good if the input features are human-understandable and intuitive features such as disease symptoms or a small number of meaningful values. However, when the input features of a decision tree are computer encoded values such as a long array of RGB pixels, it becomes impossible for humans to interpret. CNN wins over the decision tree on this specific case, we can clearly identify the activation image features for classification.

3. Novelty component

After generated the activation images, we also applied PCA and t-SNE on this data set to visualize the high dimension feature in a 2D space. First, we reduced the dimension of images from 32×32 to 50 components, from the explained_variance_ratio_ attribute we noticed that the first

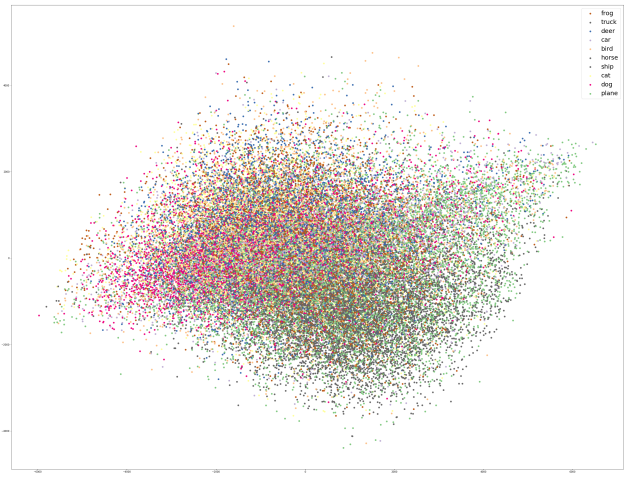

Figure 6. PCA.

Figure 7. t-SEN.

and second principal components explain almost 40% of the variance in the reduced data set. The visualization of these two components was shown as Figure 6. From the image, we can see that some certain classes are clustered together but the boundaries are not very clear.

After done PCA the data was then embedded into a 2D space by applying t-SNE. Compared with PCA, the t-SNE separated the classes further with more obvious gaps between them as shown Figure 7.

4. Conclusions

Overall, the random forest model performs the best in both classification and regression tasks. It can handle binary features, categorical features, and numerical features. There is very little pre-processing needs to be done, the data does not need to be rescaled or transformed[5]. The train-

ing speed the random forest is acceptable since it is parallelizable, meaning that we can split the process to multiple machines to run.

On the other hand, the worst model among all these models should be the decision tree model. It is prone to be overfitting and only works well in very limited situations.

For image categorization problem, CNN wins over decision tree in terms of accuracy (27.7% v.s. 78.7%) and easier interpretability with the assistance of the visualization of activation maximization, although CNN takes much longer time to train.

References

[1] Alex Krizhevsky. *The CIFAR-10 dataset*. URL: https://www.cs.toronto.edu/~kriz/cifar.html. (accessed: 11.2019).

[2] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial Intelligence* 267 (2019), pp. 1–38. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2018.07.007. URL: http://www.sciencedirect.com/science/article/pii/S0004370218305988.

[3] *Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?* URL: https://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit.

[4] *Understanding AUC - ROC Curve*. URL: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

[5] *Why Random Forest is My Favorite Machine Learning Model*. URL: https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706.