1. Introduction

**Student Number: 215920 & Kaggle Team Name: UUYW**

In recent year, there is a population addressing scenario memorability task by color, objects or pixel statistic [1] because of numerous applications like UI design, advertisement recommendation etc. The process of labeled data predicted by human is extremely strict, although the result of experiments is not exactly consistency [2]. In this work, we will use extracted feature by CNN and GIST to distinguish which image about city is memorable or not. Discuss how and why we selected this model and did pre-processing high dimensional, imbalanced and incomplete data.

## 2. Approach

Figure 3: the number of all samples in each class.

Multi-layer-perceptron (MLP) can handle with non-liner classification or regression through one or more non-linear layers called hidden layer. Backpropagation reclusively use chain rule in order to compute the gradient descent direction of every variable (features) in lost function. Add L2 regularization in lost function try to overcome over-fitting problem. Which proves MLP is a supervised learning algorithm. Input each hidden layer result to activation function which import non-linear factors improve the performance express any non-linear model. Otherwise, MLP is a combination of multiple liner function, an original perceptron.

In CNN model, we have a dropout process to do regularization in case of overfitting, randomly drop out proportional information to improve the robust of model.

Figure 1(left): x: dimension of CNN features, y: the number of lost values in x dimension.

Figure 2 (right): histogram of the number of lost value region (x) and the number of dimensions in each x region

The distribution (figure 2) of lost data tells us it is randomly picked fit with Gaussian distribution and the proportion (figure 1) of lost data is 20% which are similar with dropout process.

## 3. Methodology and results

Combine training data and additional data at first, otherwise, the number of samples does not support model to express well. Machine learning is typically proposed for domains that contain large dataset [3]. Then analyze class distribution (figure 3) by histogram, there are sets of imbalanced data, 1:0 approximately equals 7 : 1. The most algorithm are commonly focusing on classification of major samples and ignoring or misclassifying minority samples [4]. Thus, in order to balance data, the first thing to do is ignore class 1 with confidence 0.66 which may lose some unique information. It also minus the probability of misleading caused by unceMLrtain class. In addition, we regard all 0 class samples confidence as 1 because most confidence of them are 0.66, if separate them as two splits 38 and 288, severely disturb classifier learning each class features causes overfitting problem.

Consider there are a large quantity of 0 value existing in CNN data, attempt to fill all missing value with most common number to decrease loss of wrong imputed value. Weight time and accuracy, because of the high dimension it is not appropriate to adopt some ensemble method to complete value based similar feature. Like random forest imputer, it will spend a long time to math all data across each dimension. After that we can do feature selection through L1 norm with linear SVC (support vector classifier) to accelerate the convergency speed and relieve the curve of dimension.

At meantime, the training data is still imbalanced, try to cluster the major sample doing undersampling, combine the features in the same cluster which reserves as much information as possible while balancing data. As a result of that, clustering outperforms than the traditional under sampling, randomly discard extra data to keep a desired proportion [5]. We use the balanced data here because overfitting problem appears even adjust a little higher proportion, based on the test proportion easily to determine whether it looks like a reliable result.

Through a 5-fold cross-validation method to observe our model performance and adjust hyperparameters. However, the accuracy of cross-validation does not mean the true effect of testing set [6]. because of missing data imputed imprecisely and over-fitting problems. As for solve of MLP, we adopt lbfgs which can converge faster and perform better in small dataset (less than thousand). Another hyperparameter learning rate is responsible for the extent of gradient descent, too large may lead to walk around the optimal point, too small may lead to coverage slowly or even does not converge. Furthermore, we apply max-min normalization to speed up the convergency and improve our model expression power. In this problem, data range and dimension of features makes MLP converge quickly, so we won`t focus on tuning up learning rate besides L1 penalization parameterin in linear SVC, L2 penalization parameterin of MLP and hidden layer of MLP.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | 0.05 | 0.07 | 0.09 | 0.12 | 0.15 |
| Mean | 0.7746 | 0.7898 | 0.8035 | 0.7729 | 0.7515 |

Table 1. Mean accuracy of different C (L1 penalization parameter) when alpha = 0.0001, layer = 125

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedThe best hyperparameter acquired from validation is not the optimal hyperparameter in test, we only choose an approximate value around the local best value in case of overfitting. Additionally, the amount of folds and the actual folding strategy also influence the performance of CV [6].

Compared to the other activation functions sigmoid and tanh, the activation method we choose is Relu. Neural network will not easily happen gradient explosion and gradient disappearance. Relu is the most successful and widely-used activation thanks to its simplicity and effectiveness [7].

Figure 4: best predicting model based on average Spearman correlation across 5-fold validation [8].

4. Discussion

One way I tried to improve the model is build a classifier to distinguish the class 1 with different confidence called confidence classifier, Aggregate the results produced by first classifier based on test proportion. Denote class 1 confidence 1,0.66 as 1-1, 1-0.66 from the second classifier, denote 1,0 as 1,0 from results. Filter 1-1 matched with 1 from the first classifier, and 1-0.66 matched with 0 from first classifier. Recursively wash the results until no more match appears. At last, based on approximate test proportion determine 1-0.66 as 0, 1-1 as 1, if we have too many 1 or 0, make all residual as minor class. In fact, it is only more 11 0 class compared to expected. However, it does not work better than before, mainly caused by accuracy (around 75%) of confidence classifier. Work on a complete dataset may has some effect to process confidence problem.

Because of the high dimension and all non-zero data, paid more attention to CNN features. Probably we can build two multiple class classifiers to vote the results over confidence. Test Random forest imputer by GIST of complete training data, and the accuracy is more than 70%, not too bad. Two features may not be equally important but combine both are supposed to perform better than one of them.

What I learned most is analyze data to search suitable pre-processing method, wash out clean balanced data is much more vital than tuning classifier. Without the healthy data, the appropriate classifier or hyperparameter may work terribly in the testing data. I have tried extensive methods, e.g. CNN model without image caption(figure 4) [8], multiple weak classifier voting with random under sampling to offset imbalanced data while learning all samples information, or apply ensemble imputer for missing value [5], etc. Found key point should be the data processing instead of approach, inertial thinking blocks my eyes causes being stuck in circle.

References

medium. 2020. Memorability in Computer Vision. [ONLINE] Available at: https://medium.com/sciforce/memorability-in-computer-vision-763551e49d50. [Accessed 5 June 2020]

Isola, P. Xiao J., Torralba, A. and Oliva, A., 2011, June. What makes an image memorable?. In *CVPR 2011* (pp. 145-152). IEEE.

1. Menzies, T., 2001. Practical machine learning for software engineering and knowledge engineering. In *Handbook of Software Engineering and Knowledge Engineering: Volume I: Fundamentals* (pp. 837-862).
2. Williams, D.P., Myers, V. and Silvious, M.S., 2009. Mine classification with imbalanced data. *IEEE Geoscience and Remote Sensing Letters*, *6*(3), pp.528-532.
3. Longadge, R. and Dongre, S., 2013. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*.
4. Kärkkäinen, T., 2014, August. On cross-validation for MLP model evaluation. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)* (pp. 291-300). Springer, Berlin, Heidelberg.
5. Ramachandran, P., Zoph, B. and Le, Q.V., 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.
6. Squalli-Houssaini, H., Duong, N.Q., Gwenaëlle, M. and Demarty, C.H., 2018, April. Deep learning for predicting image memorability. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2371-2375). IEEE.