Re-sampling on Long Short-term Memory Recurrent Neural Network to Predict Students' Final Grade

Xiaojun Liang

Research School of Computer Science, Australian National University
u6643605@anu.edu.au

Abstract. Normal neural network is known as the model with high capabilities of classifying unknown raw data after being well trained from a set of labelled data in a similar form and outlier detection and removal can help improve training process. In the previous experiment, Bimodal Distribution Removal (BDR) is applied as an outlier removal technique to improve the normal neural network classifier. However, neural network is memoryless and BDR is not a good choice for a small unbalanced dataset. Recurrent neural network is a class of artificial neural network which outperforms in learning sequence data, especially, Long Short-term Memory (LSTM) RNNs can process entire sequences of data. In this report, we focus on using LSTM and unbalanced training data re-sampling methods on a university students' final predicting project. It turns out LSTM outperforms normal neural network and BDR neural network and over-sampling can improve the training result.

Keywords: Deep learning, recurrent neural network, long short-term memory, unbalanced dataset, over-sampling, under-sampling, SMOTE, ADASYN

1 Introduction

In this experiment, the data I choose is a first-year grade sheet of course COMP1111 from The University of New South Wales students [1]. This data contains 10 partial assessment and a final mark for the 150 course participants. As a university student myself, preparing the final exam sometimes can be struggling, which stimulated my curiosity of the feasibility to predict the final marks from the partial marks of assessments. If the predicted marks are good enough, student may not have to participated in the final exam, or otherwise an unsatisfied predicted result may be an alarm to remind students to devote themselves to the final exam review. Take the Australian National University (ANU) marking system (HD for a mark of 80 or greater, D for a mark between 70 and 80, CR for a mark between 60 and 70, P for a mark between 50 and 60, F for a mark less than 50) as a reference, I devise this as a classification problem as to predicting the final grade classes.

Recurrent neural network is known as an advanced model for time series prediction. However, once the length of time series is out of RNN's capability, gradient vanishing in training can be a problem [2]. Compared with normal recurrent neural network (RNN), Long short-term memory (LSTM) replace nonlinear units in hidden layer to memory blocks, which makes the information storage facilitated [2]. Thus, LSTM outperforms RNN in long sequences problems. According to previous experiment, LSTM are stable in dealing with noise, distributed representations and continuous values [2]. Given that COMP1111 students' mark sheet dataset, since the assessment happens in a time series and the order of the assessment matters a lot, LSTM seems to be a good choice for this problem.

For machine learning algorithms, small datasets are always challenging with the potential risk of causing overfitting [3]. To make things worse, in most cases, small datasets are unbalanced. For a classification problem, class imbalance leads to a worse generalization on unknown data [4].

To handle class imbalance, re-sampling including under-sampling and over-sampling are commonly used techniques. Under-sampling removes data examples from majority classes [5], which is not solely suitable in small dataset since the smaller the dataset, the harder to get a high prediction accuracy through training. On the opposite, over-sampling is used to generate new instance from minority classes, which is proven to be more robust [6] and is more wildly used on small dataset.

Many over-sampling methods are designed for class imbalance problems. Random over-sampling randomly chooses samples from minority classes and duplicate them. Synthetic Minority Over-sampling Technique (SMOTE), is the method to randomly choose k nearest neighbors to even the minority class size to the majority class size [6]. Adaptive synthetic (ADASYN) sampling approach applied weight to minority class examples according to the level of difficulty and synthetic instances that are relatively harder to learn [7].

In this paper, given the dataset, which is small and unbalanced, my work is to build a LSTM classification model to predict the final grade base on the partial marks and adapt different over-sampling methods on the model to see if alleviating data imbalance can help improve training the model and which one works the best.

2 Method

2.1 Data pre-processing

The provided data has 16 features including individual course information (Regno, Crse/Prog, S, ES, Tutgroup) and 11 assessment marks (lab2, tutass, lab4, h1, h2, lab7, p1, f1, mid, lab10, final) crowed in one column and has mess information in first several rows. To transfer and map the data into an appropriate form, 4 steps of pre-processing needs to be done.

1. Divide the column into 15 features and a target column according to the metadata and drop the first five rows of useless information.
2. Investigate the values and fill missing value properly. When looking into the records, some students are high likely to drop the course in the beginning because they are not allocated to any tutorial groups and has no marks for all the assessment, thus, I choose to delete them. For the rest records, I filled the missing marks with 0 since students may not handle the assignment.
3. Drop the columns (Regno, Crse/Prog, S, ES, Tutgroup) since this information are not marks but some "outfield information" which I regard as unrelative to the final marks.
4. Map the final marks to ANU classes (HD, D, CR, P, F) and encode HD to 0, D to 1, CR to 2, P to 3 and F to 4 as class labels. The result of data pre-processing is a dataset with 146 records in the format of figure 3. Only 10 partial marks as input features and one target column with 5 classification labels as the neural network output. Before feed the data into the neural model, all the input needs to be normalized, for the neural network to fairly treat all the features.

As a small dataset with only 146 records left after cleaning, the class distribution is shown in Figure 1. The class 0 only have 10 examples and the class 1 have 26 examples while the last 3 classes have relatively even sizes. Obviously, this dataset is unbalanced, and training data re-sampling is needed in model training part.

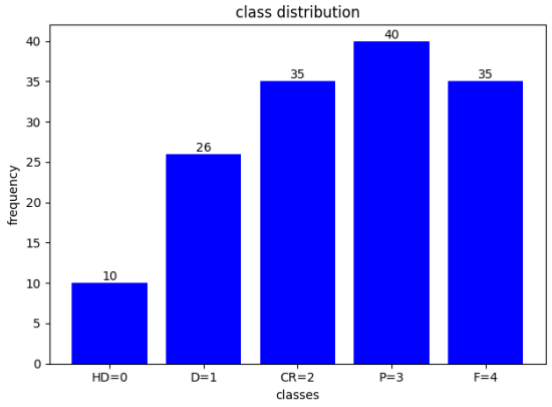

Fig. 1. Data class distribution diagram

2.2 Bimodal Distribution Removal (BDR)

In the technic paper [8], BDR is introduced to improve the training by remove the outliers based on the error distribution. During training, once the error variance is smaller than 0.1, BDR is triggered and those patterns with a bigger error than the mean error will be taken out as a subset. Then permanently remove the patterns whose error is not smaller than threshold 1, where $\overline{\delta_{ss}}$ and σ_{ss} are the mean and standard deviation of the subset. Repeat those steps until the error variance of training set is smaller than 0.01.

$$threshold = \overline{\delta_{ss}} + \alpha\sigma_{ss}(0 \leq \alpha \leq 1) \quad (1)$$

2.3 LSTM architecture

Based on the fact that was discussed in previous introduction, the COMP1111 dataset can be seen as a sequence and LSTM outperforms normal RNN in most cases. Thus, in this paper, LSTM is chosen for this problem. The simple LSTM neural network has one input layer with 10 input neurons for 10 input features, a LSTM layer with 50 neurons and an output layer with 5 neurons that each of them stands for one class label. Base on the fact that this is a multi-classification problem, Softmax function is generated for multi-classification [9] at the output layer. According to [10], cross entropy is most commonly used for classification models with softmax. At last, for optimizer, Adam and SGD are the most commonly used optimizer in neural network training. Adam is computationally efficient and can adjust learning rate in

its process [11], but sometimes it can be problematic because Adam has non-convergence problem [12]. Therefore, it will be justified later in the result comparison to see which one has better performance in addressing this problem.

2.4 Training data over-sampling

To improve the LSTM model training on the provided dataset, which is small and unbalanced, over-sampling the minority class is the top-priority strategy for re-sampling since we need to ensure the data is big enough. Three main over-sampling will be applied and compared in this paper: Random minority over-sampling with replacement, Synthetic Minority Over-sampling Technique (SMOTE) and Adaptive Synthetic (ADASYN) sampling approach.

Naive random minority over-sampling is a simple method to generates new samples by randomly copying the examples from the minority classes to even the class size. However, this may high likely to cause overfitting [6].

SMOTE generates new samples using the k-neighbors algorithm. A new synthetic sample X_{new} is generated from an existing sample X_i from its k nearest neighbors [13] (imbalanced learn doc). The imbalanced-learn API document also introduces an example of 3 nearest-neighbors over-sampling. For X_i, found the 3 nearest neighbors (in the Figure 2 blue circle). Then X_{new} is generated as formula 2:

$$X_{new} = X_i + \lambda\,(X_{zi} - X_i) \tag{2}$$

where λ is a random number in range [0, 1]. This method generates a sample between X_i and the chosen neighbor X_{zi} in Figure 2:

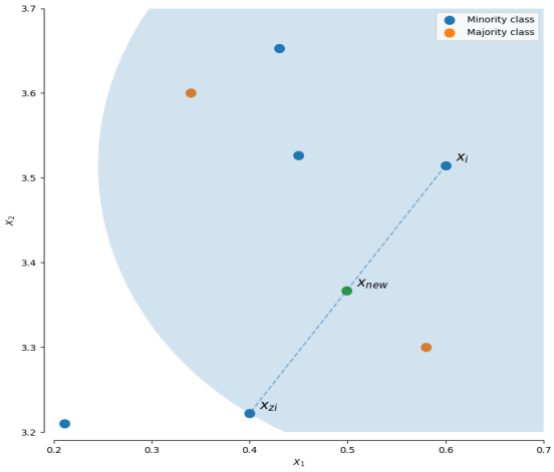

Fig. 2. 3 nearest-neighbors sample generation [13]

ADASYN introduces a new measurement of the difficulty to learn based on the similar algorithm with SMOTE. And based on that measurement, a weight distribution is applied to minority classes and automatically decide how many new samples need to be created [9]. The procedure of ADASYN can be summarized as following [9]:

1. Calculate the class imbalance degree d and compare d to the threshold d_{th}, where d_{th} is the maximum tolerated degree of class imbalance ratio.
2. If d < d_{th} , Calculate the number of samples that need to be created for the minority classes.
3. Within the chosen classes in step 2, calculate the number of examples that need to be generated for example X_i weighted on the learning difficulty level of X_i and then apply the k-neighbor algorithm introduced above.

Thus, ADASYN is not only capable of alleviating class imbalance but is also help the over-sampling focus more on those examples which are harder to learn.

2.5 Holdout validation and prediction accuracy

Holdout validation is a cross validation method that split the dataset into training set and testing set. A portion of the data is held out as testing data that is independent from the training samples and will be used to test the model. A repeated holdout is a commonly used method, where the average of repeatedly obtained estimates of error rate is called the repeated holdout estimate [14]. In this paper, instead of using the error rate, accuracy is more directly to show the prediction evaluation result. Therefore, to evaluate the performance of the LSTM model, 70% of the dataset is split to be training data and the remaining 30% is split to be testing data. Run the model 20 times and calculate the average testing accuracy.

3 Result and Discussion

3.1 LSTM optimizer comparison

To compare the performance and Adam and SGD, set the epochs to 2000 and run the model without any over-sampling for 20 times and calculate the average testing accuracy, the result is shown in Table 1.

Table 1. Average testing accuracy of Adam and SGD

	Adam	SGD
Without early stopping	53.70%	51.53%
With early stopping	55.85%	54%

As shown in Table 1, the testing accuracy of Adam is slightly higher than SGD. However, in these two situations, according to the testing loss curve, Adam are more likely to have the overfitting problem than SGD. Thus, early stopping is necessarily introduced to reduce the overfitting and the accuracy increased on both models. Still Adam is better than SGD and of much better efficiency. Therefore, Adam and early stopping is set up for further experiment.

3.2 Training data over-sampling distributions

As shown in Figure 3, in this training, the training set size is 105. The top left diagram shows the original training set class distributions, which is in the similar imbalance with the whole data distributions shown in Figure 1. Class 0 only has 8 samples which is significantly smaller than the other classes. The right top and the left bottom diagrams show the results of random over-sampling and SMOTE over-sampling, where both evens the class size to the majority class. The right bottom diagram shows the result of ADASYN, not evenly over-sampling the minority classes to the majority size but according to the difficulty level of learning.

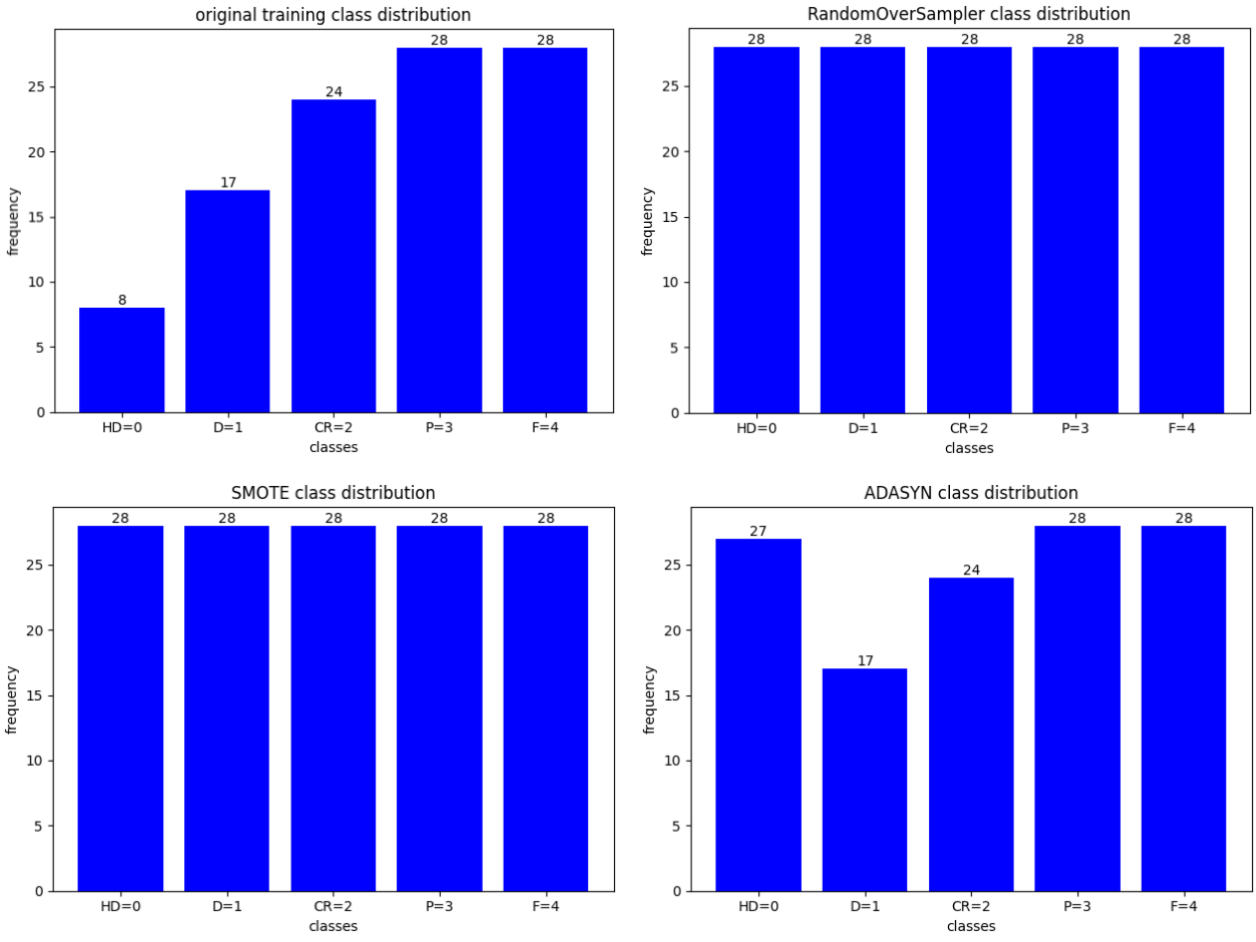

Fig. 3. Class distribution on different over-sampling method

3.3 Over-sampling methods comparison

Given the LSTM neural network, Adam and early stopping is proven to be the ideal settings with a better performance. In this session, 3 over-sampling methods is applied to the LSTM structure and the result of repeated holdout testing accuracy is shown in Table 2.

Table 2. Over-sampling LSTM average testing accuracy

Over-sampling methods	No over-sampling	Random	SMOTE	ADASYN
Average testing accuracy	55.85%	56.8%	59.41%	52%

As shown in Table 2, without training data over-sampling, the LSTM average testing accuracy is 55.85%. Random over-sampling and SMOTE over-sampling have a higher testing accuracy than the original training dataset and especially the SMOTE over-sampling method outperforms than other method with the 59.41% testing accuracy. However, ADASYN performs worst in the three over-sampling methods and even worse than the non-resampling LSTM model. Since the methodology of ADASYN do not given the equal chance to each minority instance to be selected, one drawback of ADASYN is that outliers cannot be identified and therefore decrease the prediction accuracy [15].

However, compared with the 47.64% average testing accuracy of normal neural network and 51.03% average testing accuracy of BDR neural network in previous experiment, the over-sampling-based LSTM shows great improvement.

4 Conclusion and Future work

In this paper, provided with the COMP1111 grade sheet of University of New South Wales students, which can be regard as a sequential data, I devised a classification problem and built a LSTM recurrent neural network to predict the students final grade over the partial assessment marks. It turns out LSTM works better than a normal neural network and a BDR neural network. And Adam fits this model better than SGD.

The size and distribution of the training data has tremendous impact on the training result. This paper investigates some re-sampling methods to deal with class imbalance in a small data. To avoid making the dataset shrinking, over-sampling methods including random over-sampling, SMOTE and ADASYN are implemented and compared on the repeated holdout accuracy. It turns out in this case, SMOTE works the best.

However, over-sampling brings noises when generating new samples, in the future, hybrid re-sampling methods that combines over-sampling and under-sampling is worth of research. Moreover, neural network fine tuning is always a hot topic to spend effort on.

5 Reference

1. Che, E., Choi, Y., Gedeon, T. D.: Comparison of extracted rules from multiple networks. In: International Conference on Neural Networks, vol. 4, pp. 1812-1815. IEEE (1995)
2. Li, Y., Zhu, Z., K, D., Han, H., Zhao, Y.: EA-LSTM: Evolutionary attention-based LSTM for time series prediction. In: Knowledge-Based Systems, vol. 181, pp. 104785. (2019)
3. Perez, L., Wang, J.: The Effectiveness of Data Augmentation in Image Classification using Deep Learning. arXiv preprint arXiv: 1712.04621 (2017).
4. Buda, M., Maki, A., Mazurowski, MA.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks 106, 249-259 (2018).
5. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. In: Intelligent data analysis vol. 6(5), pp. 429-449 (2002).
6. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer P.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16, 321-357 (2002).
7. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the 5th IEEE International Joint Conference on Neural Networks, pp. 1322-1328 (2008)
8. Slade, P., Gedeon, T.D.: Bimodal distribution removal. In: International Workshop on Artificial Neural Networks. pp. 249-254. Springer (1993)
9. Jiang, M., Liang, Y., Feng, X., Fan, X., Pei, Z., Xue, Y., Guan, R.: Text classification based on deep belief network and softmax regression. In: Neural Computing and Applications, vol. 29(1), pp. 61-70 (2018)
10. Zhang, X., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In: Advances in neural information processing systems, pp. 8778-8788 (2018)
11. Kingma, D., Ba, J.: ADAM: A Method for Stochastic Optimization. In: 3rd International Conference for Learning Representations, San Diego (2015)

12. Reddi, S. J., Kale, S., Kumar, S.: On the Convergence of Adam and Beyond. arXiv preprint arXiv:1904.09237 (2019).
13. Imbalaned-learn User Guide.: Over-sampling, https://imbalanced-learn.readthedocs.io/en/stable/over_sampling.html
14. Kim, J.H.: Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. In: Computational statistics & data analysis, vol. 53(11), pp. 3735-3745 (2009)
15. Dattagupta, S.J.: A performance comparison of oversampling methods for data generation in imbalanced learning tasks. Diss. (2018)