MLP Coursework 1

s2045458

Abstract

In this report, the problem of overfitting and how dropout and weight penalty could mitigate the problem is investigated. The effect of hidden layer number and hidden layer units on overfitting is researched through a set of experiment. To evaluate the effect of dropout and weight penalty, two methods are implemented on the baseline network to perform experiment. The experiment results show dropout and weight penalty has significant contribution in mitigating overfitting. The combination of L1 penalty and dropout achieves best performance.

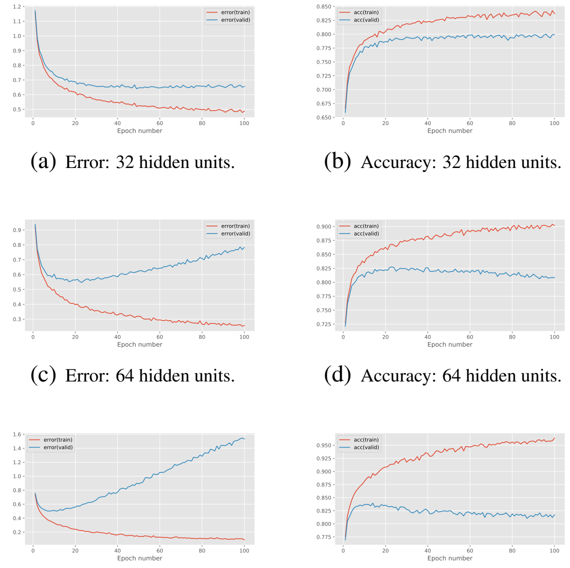
1. Introduction

The report of machine learning coursework is based on the classification of handwritten digits using neural network. There are two main tasks of this coursework. Identify the problem presented in "Figure 1" of coursework paper, which is overfitting, and investigate the width and depth of neural network. Another one is implementing dropout and weight penalty to solve the overfitting, improve validation accuracy and find the best configuration based on given baseline network. To accomplish the tasks of the coursework, they are broken down to several objectives:

- Prepare the neural network and training, validation, testing EMNIST dataset file.
- Vary the number of hidden units and layers to find whether they mitigates or worsens the overfitting problem.
- Implement dropout and weight penalty on given baseline network.
- Do experiments to find the effect of dropout and weight penalty on overfitting and select the best model configuration.

2. Problem identification

The problem addressed by the curves in the "Figure 1" of the coursework specification file is overfitting. As a modeling error, overfitting occurs when a network function is too closely or exactly fit to a limited set of training data points. To avoid overfitting, there are potential solution can be taken i) adjust the complexity of network corresponding to the training data set, ii) set prior constrain on network function based on advance information, iii) control the flexibility by implementing structural stabilization and regularization, common regularization methods include early stopping, weight penalty, dropout, etc.



(a) Error: 32 hidden units.

(b) Accuracy: 32 hidden units.

(c) Error: 64 hidden units.

(d) Accuracy: 64 hidden units.

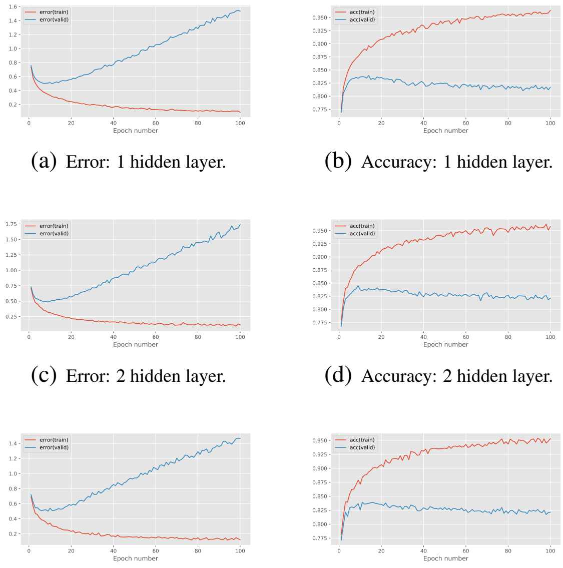
(e) Error: 128 hidden units.

(f) Accuracy: 128 hidden units.

Figure 1. Error and accuracy curves of varying number of hidden units experiment on 1-hidden layer networks by using either 32, 64 and 128 Relu units.

Figure 1 visualizes the experiment of varying number of hidden units by using 32, 64, 128 hidden units on 1 hidden layer network. The error and accuracy curves illustrate the impact of the increasing number of hidden units on both training and validation performance. Both training and validation accuracy increases in line with the increasement of hidden unit numbers. However, obvious overfitting phenomenon occurs when the number of hidden units has increased to 64 and 128. The validation accuracy decreases after peak value in the process of 64 and 128 hidden units experiments. The situation in 128 hidden unit is worse. The experiment reveal the increasement of hidden units worsens the problem of overfitting.

Figure 2 visualizes the experiment of varying number of hidden layers by using 1, 2, 3 hidden layers with 128 Relu hidden units. The experiment results depicted shows the number of hidden layers do not have significant effect on validation performance. The results also shows the variation of hidden layer number does not worsen or mitigate overfitting. In this experiment, the result difference is not obvious enough to get a clear conclusion. Theoretically



(a) Error: 1 hidden layer. (b) Accuracy: 1 hidden layer.

(c) Error: 2 hidden layer. (d) Accuracy: 2 hidden layer.

(e) Error: 3 hidden layer. (f) Accuracy: 3 hidden layer.

Figure 2. Error and accuracy curves of varying number of hidden layers experiment with 128 Relu hidden units by using 1, 2 and 3 hidden layers.

speaking, the increasement of hidden layers do increase the complexity of the network and it should affect model performance to some extent. Considering the obtained experiment results and theoretical analysis, to get a clear and exact conclusion about the relationship between number of hidden layers, overfititng and validation accuracy, further experiments need to be performed.

3. Dropout and Weight Penalty

$$E_{Dropout} = \sum_i w_i p_i$$

Dropout is a regularization technique which ignores a certain set of units chosen at random during the training process. To be more specific, at each training stage, dropout layer randomly generates a dropout mask to handle the inputs based on predefined probability p. 1-p individual nodes are temporarily dropped out of the network and p individual nodes are left. In each mini-batch training, dropout randomly removes 1-p fraction of units. Individual hidden unit cannot provide features based on all other hidden units in each mini-batch. The randomly dropped hidden units will cause different missing features in different mini-batch training round. Dropout leads to an implicit model features combination, which alleviate overfitting. To implement dropout in coursework baseline network, four dropout layers are separately attached after Relu hidden units layers one by one. In dropout forward propagation function, a randomly generated dropout mask based on given dropout probability p is used to select valid inputs and temporarily clear the inputs of 1-p ignored hidden units. In dropout backward propagation, it returns gradients matrix handled by dropout mask.

$$E_{L1} = \sum_i \beta |w_i|$$

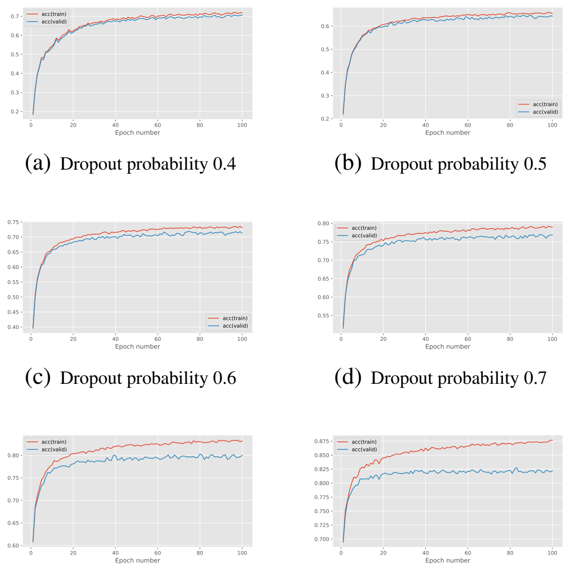
$$E_{L2} = \frac{1}{2} \sum_i w_i^2$$

Both L1 and L2 regularisation correspond to adding a term with effect of penalising large weights. L1 regularisation penalizes sum of absolute values of weights to the error. In L1 weights shrink to 0 at a constant rate. L2 generalisation penalizes sum of half square values of weights to the error. In L2 weights shrink to 0 at a proportional rate to the weight value. Therefore, when weight is small, L1 shrinks faster than L2. When weight is large, it reverses. Consequently, L1 tends to shrink weights to 0, leaving important features (feature selection). It encourages sparsity. L2 has none sparse solution, has no feature selections and gives better prediction while learning complex data patterns. Regularization is based on the assumption smaller weights generate simpler model. L1 L2 regularization penalise weights in different terms and reduce the model complexity, which mitigates overfitting. To implement L 1 and L2 penalty, "call" function is completed by adding a corresponding penalty term and multiplying it with a given coefficient. In "grad" function, it returns parameters multiplied by penalty coefficient.

4. Balanced EMNIST Experiments

In this section, the performance of different hyper-parameter settings and their combinations on multi-layer neural networks are investigated. Firstly, the functions related hyper-parameters (dropout probability and weight penalty coefficient) and other hyper-parameters (learning rate) are tested individually.

From given lecture material and empirical experience, the ideal dropout probability range lies between 0.5 to 0.8. To find the best dropout probability based on the baseline network with only dropout method, a set of tests is performed with dropout probabilities set from 0.4 to 0.9 with interval 0.1. Figure 3 visualises the experiment of finding best parameter setting of dropout method and the results prove the ability of dropout to mitigate overfitting. As depicted in Figure 3, the validation accuracy increases in line with dropout probabilities. When dropout probability reach 0.7 little overfitting occurs. When dropout probability reach 0.9, significant overfitting occurs. Based on the baseline network configuration, to avoid overfitting, dropout probability has to be less than 0.7. To pursue best validation performance, dropout has to be larger than 0.9. Considering both factors, the best dropout probability chosen for later experiment is 0.8. Additionally, compared to baseline network, all the dropout experiments with different dropout probabilities decrease the validation accuracy. The reason might be dropout ignore a fraction of hidden layer and the process weakens the network ability to learn features from training data. The results of experiments on individual



(a) Dropout probability 0.4

(b) Dropout probability 0.5

(c) Dropout probability 0.6

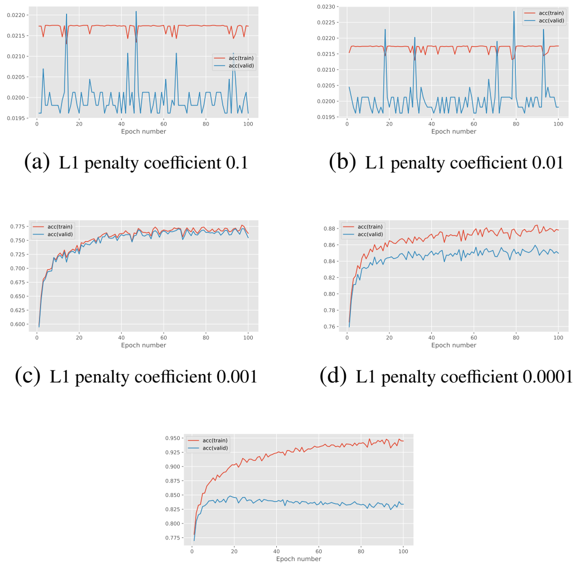
(d) Dropout probability 0.7

(e) Dropout probability 0.8

(f) Dropout probability 0.9

Figure 3. Accuracy curves of dropout probability experiment.

dropout method are likely to be underfitting rather than overfitting.



(a) L1 penalty coefficient 0.1

(b) L1 penalty coefficient 0.01

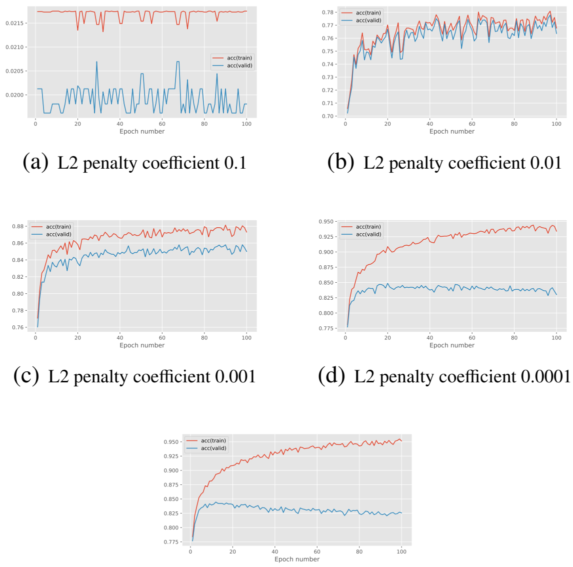
(c) L1 penalty coefficient 0.001

(d) L1 penalty coefficient 0.0001

(e) L1 penalty coefficient 0.00001

Figure 4. Accuracy curves of L1 penalty coefficient experiment.

As indicated in coursework material, the penalty coefficient usually lies in the range of 0.1 to 0.00001. To find the best parameter setting of L1 coefficient, a set of L1 coefficient penalty test from 0.1 to 0.00001 is performed. The result is illustrated in Figure 4. While L1 penalty coefficient decreasing from 0.1 to 0.00001, Figure 4(a) and Figure 4(b) indicate when L1 coefficient equals to 0.1 and 0.01, the training and validation accuracy do not increase at all. That is, the L1 coefficient is too large that large penalties suppress weights to be large in magnitude. Consequently, the network is not capable to learn the importance of features and model has not been trained at all. Figure 4(c) illustrates a well-performed training and no overfitting occurs when L1 penalty equals to 0.001. Figure 4(d) shows validation accuracy is keep increasing and little overfitting occurs when L1 penalty equals to 0.0001. Figure 4(e) shows a strong overfitting phenomenon when L1 coefficient equals to 0.00001. That is, the L1 coefficient is too small to prevent overfitting at this stage. Considering both validation accuracy and overfitting situation, 0.0001 is the best value for L1 coefficient.



(a) L2 penalty coefficient 0.1

(b) L2 penalty coefficient 0.01

(c) L2 penalty coefficient 0.001

(d) L2 penalty coefficient 0.0001

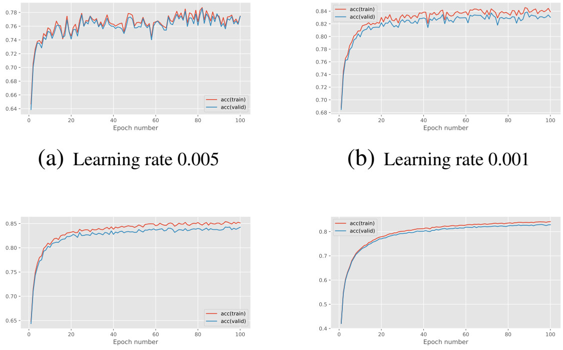
(e) L2 penalty coefficient 0.00001

Figure 5. Accuracy curves of L2 penalty coefficient experiment.

Same as L1 penalty coefficient, L2 penalty coefficient value usually lies in range 0.1 to 0.00001. Figure 5 visualises the L2 coefficient experiment. Figure 5(a) shows a too large penalty that the network fail to learn features, when L2 coefficient equals to 0.1. In Figure 5(b), no overfitting occurs but the accuracy curve cannot stabilize. That is, the feature scaling is bad under this configuration. Figure 5(c) shows a high validation accuracy training and little overfitting occurs when L2 coefficient equals to 0.001. While L2 coefficient keep decreasing, Figure 5(d) and Figure 5(e) show strong overfitting phenomenons occur when :2 coefficient equals to 0.0001 and 0.00001. In this experiment, the best validation accuracy obtained when L2 penalty coefficient equals to 0.001. Additionally, only little overfitting occurs at this stage. 0.001 is the best L2 penalty coefficient parameter setting of the network which only applies L2 regularization method.

The experiments for L1 penalty and L2 penalty prove weight penalty do mitigate overfitting because these two methods penalize the loss function to help simplifying the model to avoid overfitting. From the experiments, when L1 coefficient equals 0.0001 and L2 coefficient equals to 0.001 and 0.0001, validation accuracy is increased. The other weight penalty coefficient decrease validation accuracy. That is, weight penalty only positively contribute to validation accuracy when implements appropriate parameter settings.

After the experiments performed on several networks which only implements individual regularization method, the best parameter setting for each method is obtained. That is, L1 penalty coefficient 0.0001, L2 penalty coefficient 0.001, Dropout probability 0.8. To investigate the performance of the combination of these regularization methods, the parameter settings obtained in individual method experiments will be used as a start point. To be specific, the combination of dropout and L1 penalty network will use parameter setting dropout probability = 0.8, L1 penalty coefficient = 0.0001. The combination of dropout and L2 penalty network will use the parameter setting dropout probability = 0.8, L2 penalty equals to 0.001. Based on the regularization settings, the combination experiment will investigate the best learning rate for two combination separately.



(a) Learning rate 0.005

(b) Learning rate 0.001

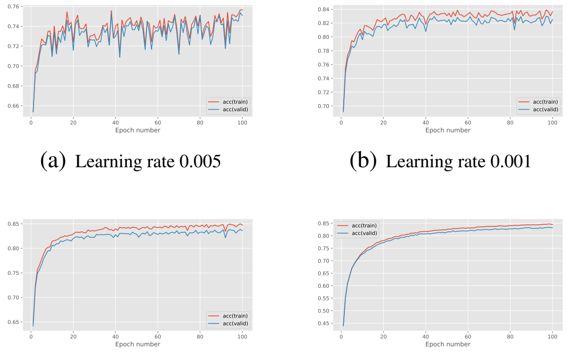
(c) Learning rate 0.0005

(d) Learning rate 0.0001

Figure 6. Accuracy curves of learning rate experiment with dropout probability 0.8 and L1 penalty coefficient equals to 0.0001.

In the dropout and L1 penalty combination experiment, the default learning rate 0.001 is chosen as a start point. As showed in the result figures of previous experiments, there exists obvious curves fluctuations. So the learning rate tends to be set smaller values. 0.005, 0.001(default learning rate), 0.0005, 0.0001 are selected to perform combination experiment. The result is showed in Figure 6. Compared to results obtained in dropout and L1 penalty individual experiments, the combination of L1 penalty and dropout shows less overfitting. These are obvious fluctuation in Figure 6(c) and 6(d), which means learning rate value is too large. In Figure 6(c), the learning rate 0.0005 experiment obtains a good result with high validation accuracy and little overfitting. Figure 6(d) illustrates little underfitting when learning rate equals to 0.0001. Considering both overfitting issue and validation accuracy, learning rate 0.0005 is the best parameter setting for L1 penalty and dropout experiment.

Similar to combination experiment of dropout and L1 penalty, the combination experiment of dropout and L2 uses same learning rates. The result is illustrated in Figure 7. Figure 7(a) and Figure 7(b) indicate learning rate is too larger when it equals to 0.005 and 0.001. Compare Figure 7(c) and 7(d), the validation accuracy is similar, but learning rate equals to 0.0001 has better performance in preventing overfitting. So learning rate 0.0001 is the best parameter setting for L2 penalty and dropout combination.



(a) Learning rate 0.005

(b) Learning rate 0.001

(c) Learning rate 0.0005

(d) Learning rate 0.0001

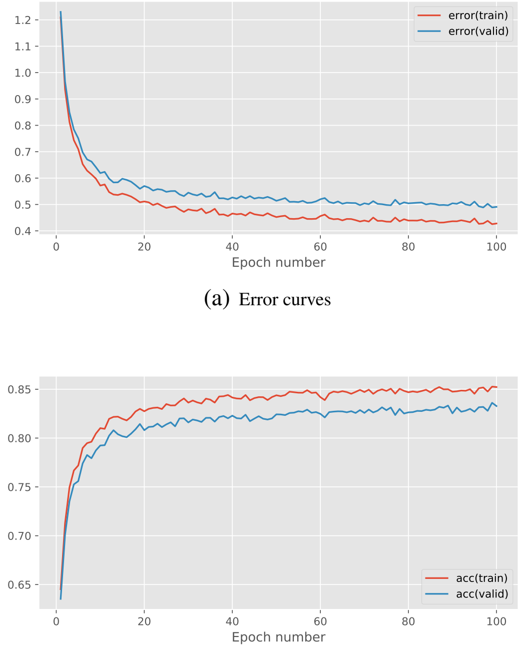
Figure 7. Accuracy curves of learning rate experiment with dropout probability 0.8 and L2 penalty coefficient equals to 0.001.

By using default learning rate, the validation accuracy of the combination of dropout and weight penalty is worse than individual weight penalty method but better than dropout method. The phenomenon might result from that dropout weaken the network ability to learn data features. The best validation accuracy found using L1 penalty is slightly better than the best validation accuracy using L2 penalty. The reason might be L2 penalty is good at learning complex model but L1 penalty is better at feature selection. The data and the model used in this coursework is not complex enough to exhibit the power of L2 penalty. For the problem of overfitting, both weight penalty and dropout show their ability to mitigate overfitting and the combination of them exhibit stronger ability by comparing to baseline network performance.

Comparing all the best configuration obtained in these experiment and considering both overfitting issue and validation accuracy, the combination of dropout and L1 penalty with dropout probability equals to 0.8, L1 penalty coefficient equals to 0.0001, learning rate equals to 0.0005 is the best model configuration. The Figure 8 shows error and accuracy curves of the best model configuration evaluated on training set and testing set. The test performance result are 0.49(error) and 0.83(accuracy).

5. Literature Review: Paper Title

The paper will be reviewed in this section is Maxout Network (I. Goodfellow, 2013). The paper designed and introduced a new model called maxout to utilize a model averaging method called dropout. Firstly, authors argued the opinion that dropout can only be used as model performance enhancement. They tried to design a model which could directly improve the performance of dropout in perspective of model averaging technique. Then, the details



(a) Error curves

(b) Accuracy curves

Figure 8. The error and accuracy curves of best model configuration obtained in experiments with dropout probability equals to 0.8, L1 penalty coefficient equals to 0.0001, learning rate equals to 0.0005. The figure is obtained by performing test on training and testing set. Both "error(valid)" and "acc(valid)" here indicates the result on test set

of dropout as a contributive technology which is usually implemented in deterministic feedforward architectures are reviewed. After that, the concept of maxout model, a simple feedforward architecture, is introduced and the compatibility between dropout and maxout is analysed theoretically. In addition, the strength of maxout as a universal approximator is proved through deduction based on two reliable propositions, which generates a conclusion that "A two hidden layer unit maxout network can approximate any continuous function f(v) arbitratily well on the compact domain C". Subsequently, the designed maxout model is evaluated on four benchmark datasets. With different maxout layers layouts implemented, the combination of dropout and maxout method obtained extreme low test error rate in the groups of best methods corresponding to four benchmark datasets. The result turns out to be remarkable. After practically demonstrated the effectiveness of maxout networks, the authors identified the reasons why maxout network is remarkably compatible with dropout. In training datasets, dropout technique provides larger input linear domains for maxout hidden units. Another reason is dropout bagging style training phrase is improved by maxout network.

In this coursework, the default activation function is Relu. The performance of dropout method on Relu layers is investigated when perform individual dropout experiment on the baseline network. The paper provided another way to improve the network performance of dropout. It selected dropout as the start point and specially designed a new activation function for it, which was regarded as an indiscriminately applicable tool. The paper comprehensively evaluated and proved the strength of dropout and maxout combination method theoretically by deduction and practically by experiments on four benchmark datasets. Based on the information provided in paper, the optimization experiment of coursework baseline network can be further extended or improved by replacing Relu activation function with maxout.

6. Conclusions

The report presents results of exploring the effect of hidden units and layers on overfitting and the investigation of weight penalty and dropout. Both dropout and weight penalty could mitigate the problem of overfitting and their combination has better performance. Dropout always decreases the validation accuracy. weight penalty could improve validation accuracy while using approximate parameter setting. The experiment can be further extended by implementing maxout which is mentioned in (I. Goodfellow, 2013) to cooprate with dropout to pursue better validation performance.

References

I. Goodfellow, D. Warde-Farley, M. Mirza A. Courville Y. Bengio. Maxout network. 2013. URL https://arxiv.org/abs/1302.4389.