Investigating Selective Learning Regularisation Techniques

Quinton Weenink
dept. Computer Science
University of Pretoria
Pretoria, South Africa
u13176545@tuks.co.za

***Abstract*—In this report it is found that active learning can improve the generalization performance of a Neural Network when applied to the Sloan Digital Sky Survey. Classification accuracy of the actively trained Neural Network is improved compared to other learning methods. Specifically the addiction of a selective learning to the training of the neural network allowed it to achieve both better classification accuracy as well as less generalization error. It was also found that for small networks regularisation schemes do not have a beneficial impact for active training.**

***Index Terms*—NN, Active learning, Selective learning**

I. Introduction

In this report we aim to investigate the effects of active learning and how it can improve the performance of a Neural Network when applied to the Sloan Digital Sky Survey. Generalization of the active trained Neural Network will be compared to other other learning techniques.

Regularisation techniques will be include for comparison as well as in conjunction with active learning to determine their influence on the results. Regularisation schemes such as weight decay, weight elimination and dropout aim to reduce generalization error in trained Neural Networks. By combining active learning and these regularisation schemes we might see even higher improvement in generalization.

For comparison passive learning techniques were used such as, batch and mini-batch, both widely used in research each with their own benefits.

II. Background

The following describes the concepts and structures used for the evaluation of the scenarios in this report.

A. Neural Network

Typically, feedforward Neural Networks are comprised of neurons connected in layers. Signals travel from the input layer through the hidden layer(s) to the output layer [1]. Each neuron is connected through a collection of weights to the neurons in the next layer as can be seen in Fig. 1. This study used a Neural Network with bias on every layer except the output layer.

Back propagation is an often used training method for Neural Networks and makes use of forward propagation to generate the Neural Network's output value(s). The weights are manipulated to train the network using the training error.

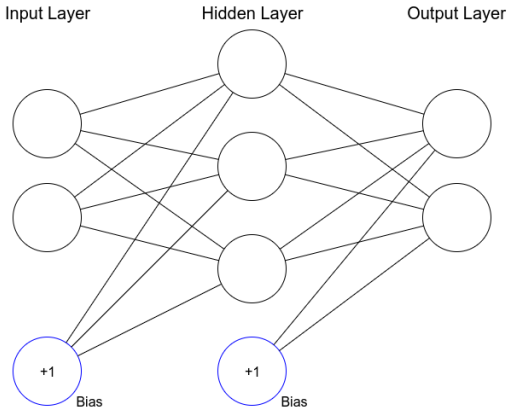

Fig. 1. Neural network with bias

$$g(net) = \frac{1}{1 - e^{-net}} \tag{1}$$

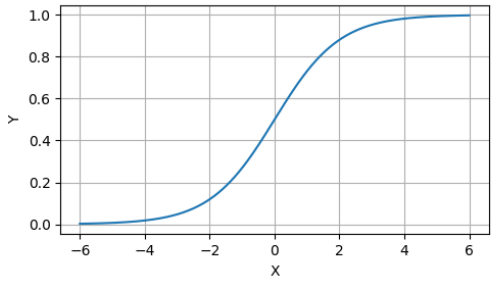

Fig. 2. Sigmoid activation function

Fig. (2) represents the output of a sigmoid function (1) which will be used as the activation function for Neural Networks in this this research.

B. Neural Network structure

11 input neurons, 11 hidden layer neurons and 3 output neurons where used all fully connected to each other. The

Softmax algorithm was run on the output layer as it clearly defines a winner for classifcation.

C. Regularisation

1) Dropout: A regularisation technique which implemented by randomly omitting neurons from the network with a probability of 0.5, so a hidden unit cannot rely on other hidden units being present.Dropout been imperially shown that it significantly reduces overfitting and gives major improvements over other regularization methods [2]

2) Weight decay: It has been observed in numerical simulations that a weight decay can improve generalization in a feed-forward neural network. [3]

D. Optimization techniques

The Adam optimization algorithm was selected for this research due to its performance and notably with data sets much like the *Sloan Digital Sky Survey* (SDSS) [4]. A learning rate of 0.01 was used throughout this research after tuning for optimum results.

E. Loss function

Cross-entropy was selected as the loss function due to issues with *mean squared error* (MSE) [5]

III. Learning

Neural network learning can be described in two ways, passive and active. Passive learning could be considered unsupervised while active learning could be considered as partially supervised as the samples that are presented for training are chosen. They are described in more detail below:

A. Passive Learning

1) Full batch: Batch training, opposite to stochastic training, feeds all available samples through the neural network and updates are calculated as one when back propagated through the neural network.

2) Mini batch: Mini batch training, much like full batch training, feeds multiple samples the neural network but rather than feeding all samples, a smaller batch is randomly selected and passed through. The batch size parameter determines the amount of samples that are passed through.

B. Active Learning

Active learning is any form of learning where the learning algorithm has control over what part of the input space it receives. [6]

1) Selective Learning: This report uses a type of Active Learning called Selective Learning. This learning approach only selects training samples which have the largest effect on the the Neural Network. The Selective learning algorithm used in this research selects samples based on the loss they would have if run through the Neural Network, specifically the batch includs the samples which had the largest loss when passed through the current Neural Network.

IV. Experimental Set-Up

A. Data sets selection and preparation

For this research the SDSS dataset was used and is comprised of a set of observations is described by 17 feature columns and 1 target column which identifies the observation to be a star, a galaxy or a quasar. Table I describes each of the features and weather they were included in the research or not.

TABLE I
NEUAL NETWORK FEATURES

Data Set	*Usage*	*Range*	*Mean*
objid	not included	–	–
ra	included	(8.2351, 260.884)	1.755300e+02
dec	included	(-5.38263, 68.5423)	1.483615e+01
u	included	(12.989, 19.5999)	1.861936e+01
g	included	(12.7995, 19.919)	1.737193e+01
r	included	(12.4316, 24.802)	1.684096e+01
i	included	(11.9472, 28.1796)	1.658358e+01
z	included	(11.6104, 22.8331)	1.642283e+01
run	not included	–	–
rerun	not included	–	–
camcol	not included	–	–
field	not included	–	–
specobjid	not included	–	–
class	target	(GALAXY, STAR)	–
redshift	included	(-0.00413, 5.3538)	1.437257e-01
plate	included	(266, 8410)	1.460986e+03
mjd	included	(51578, 57481)	5.294353e+04
fiberid	included	(1, 1000)	3.530694e+02

The features that were left out of were left out due them not having any relevance to the class they are classifying.

The dataset is comprised of 10000 items. The set was split into 3 sets. A training set D_T, for training the neural network. A testing set for evaluating the fitness of the neural network without contributing to the training process directly and the validation set D_G which is left only for validation when analysis is performed. The data was split into the training, testing and validation sets randomly with percentages 60%, 20%, 20% respectively.

All input to the Neural Network was scaled to the range $[-1, 1]$. While some research is found to do initial data manipulation to the SDSS input data, no data manipulation other than min max scaling was done.

One-hot encoding was used allowing the output to be as sparse as possible. One-hot encoding allows each of the neurons in the output layer to represent a specific class with all classification output was mapped to either 0 or 1.

Neural Network classification accuracy is used to determine how the neural network performed. The training error, E_T, was used in order to assess the resulting Neural Network accuracy.

$$E_T = 1 - \frac{\sum_{p=1}^{ND_T} correct(D_{Tp})}{ND_T} \tag{2}$$

Similarly to the training error the generalisation error, E_G, can be used to calculate the performance of the Neural Network on the set D_G. Each Neural Network was trained over 2000 iterations.

Additionally the generalisation factor $\rho_F = \frac{E_G}{E_T}$ was also used to measure the overfitting of the trained Neural Networks. A $\rho_F > 1$ would indicate a possibility of overfitting due to the generalisation error being greater than the training error. While a $\rho_f < 1$ or $E_G < E_T$ could indicate a lack of overfitting.

Batch sizes of 200 and 2000 were used as they had a notable impact on the performance of mini-batch training.

The dropout rate was set to 0.2 due to its poor performance with higher dropout rate. This poor performace could be as a result of the small size of the neural network. Weight decay is set to 0.01 after testing performance with other values for weight decay.

V. Results

The resulting E_T, E_G and ρ_F for all experiments conducted in this study are given in II and III.

Generally all methods performed well for the configurations tested in this research with all results being in the 98th percentile for E_G that is a result where almost all samples were classified as correct in the generalization set.

What first comes to mind when one sees the results in table II is the performance that can be achieved with active learning even with a batch size of 200. When compared with passive mini-batch the result are improved.

Fig. 6 does however indicate another problem that a small batch size could have when applying selective learning. One could presume that due to the small batch size the irregularity observed could be due to large weight updates taking place. This could result in step sizes being to large in some cases (other datasets) resulting in a neural network that never converges. The large irregularity can be observed more clearly when observing a single instance as in Fig. 3, this high irregularity is not observed in the other training methods.

Interestingly dropout seems to resolve these high fluctuations in the network as can also be seen in Fig. 4. Unfortunately the accuracy of the solution does not outperform the other configurations.

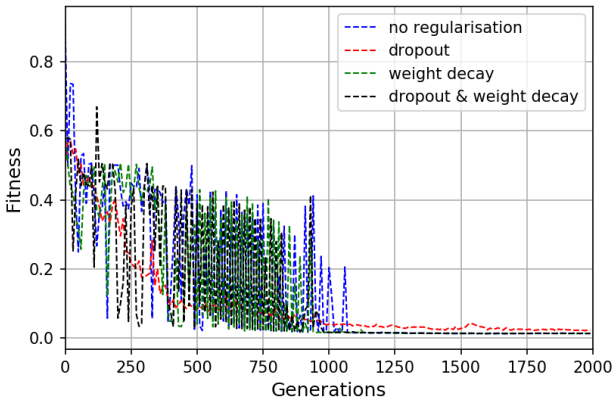

Fig. 3. E_C during active training over 2000 iterations. This figure was for one sample, and was chosen due to the irregular training pattern active learning seems to induce

TABLE II
Results after 2000 iterations with a batch size of 200. Means are reported over 30 samples with standard deviations in parenthesis

Full Batch			
regularisation	E_T	E_G	ρ_F
none	0.053333 (0.027516)	0.055117 (0.028879)	0.998072 (0.002288)
dropout	0.067789 (0.026438)	0.070733 (0.028854)	0.996767 (0.004074)
weight decay	0.041633 (0.016134)	0.043300 (0.017917)	0.998228 (0.002208)
dropout & weight decay	0.050217 (0.026461)	0.052517 (0.028698)	0.997508 (0.002836)
Random Mini-batch			
regularisation	E_T	E_G	ρ_F
none	0.128311 (0.001318)	0.135017 (0.001691)	0.992308 (0.001410)
dropout	0.127950 (0.008376)	0.134000 (0.010510)	0.993041 (0.002990)
weight decay	0.128950 (0.002140)	0.135350 (0.001867)	0.992654 (0.001372)
dropout & weight decay	0.130239 (0.003812)	0.136650 (0.003760)	0.992631 (0.001802)
Selective Learning			
regularisation	E_T	E_G	ρ_F
none	0.011633 (0.000591)	0.016550 (0.000472)	0.995026 (0.000522)
dropout	0.025289 (0.004769)	0.029283 (0.004783)	0.995906 (0.002999)
weight decay	0.011772 (0.000867)	0.016367 (0.000645)	0.995351 (0.000964)
dropout & weight decay	0.011422 (0.000544)	0.016483 (0.000652)	0.994880 (0.000459)

Figures 3, 4, 5, 6 were chosen based on weather a distinguishable observation could be made.

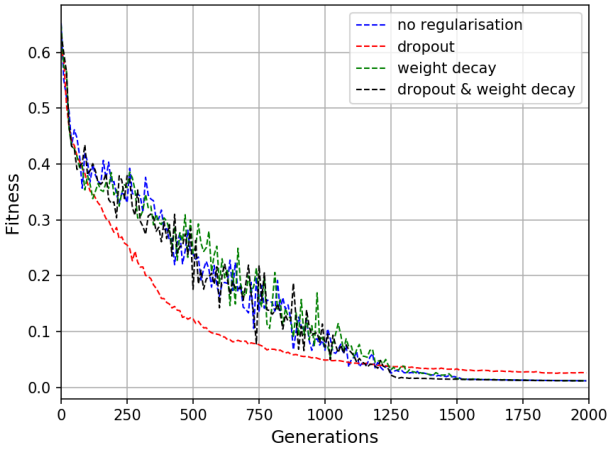

Fig. 4. E_C during active training over 2000 iterations.

TABLE III
RESULTS AFTER 2000 ITERATIONS WITH A BATCH SIZE OF 2000. MEANS ARE REPORTED OVER 30 SAMPLES WITH STANDARD DEVIATIONS IN PARENTHESIS

Full Batch			
regularisation	E_T	E_G	ρ_F
none	0.106456 (0.026626)	0.100683 (0.024244)	1.006541 (0.002907)
dropout	0.153944 (0.002047)	0.150083 (0.003472)	1.004571 (0.005143)
weight decay	0.090628 (0.036755)	0.086050 (0.033668)	1.005176 (0.003712)
dropout & weight decay	0.107572 (0.025046)	0.101683 (0.022831)	1.006669 (0.002731)
Random Mini-batch			
regularisation	E_T	E_G	ρ_F
none	0.113950 (0.030494)	0.108233 (0.027633)	1.006567 (0.003755)
dropout	0.160228 (0.010006)	0.157850 (0.009527)	1.002848 (0.004516)
weight decay	0.113356 (0.031720)	0.107650 (0.029037)	1.006547 (0.003594)
dropout & weight decay	0.098517 (0.036816)	0.094267 (0.033412)	1.004876 (0.004278)
Selective Learning			
regularisation	E_T	E_G	ρ_F
none	0.029272 (0.001034)	0.025483 (0.001172)	1.003904 (0.001097)
dropout	0.086189 (0.003251)	0.083367 (0.004651)	1.003093 (0.004743)
weight decay	0.029272 (0.001343)	0.025717 (0.001476)	1.003663 (0.001081)
dropout & weight decay	0.029200 (0.001327)	0.025367 (0.001056)	1.003950 (0.001076)

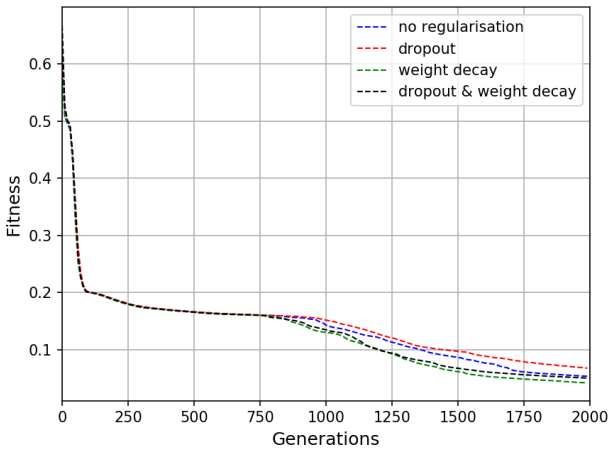

Fig. 5. E_C during full batch training over 2000 iterations.

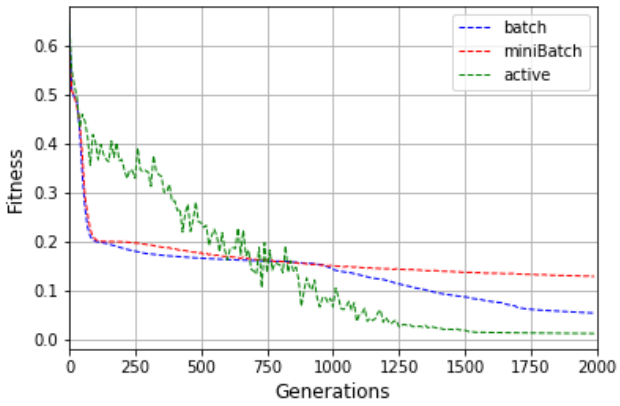

Fig. 6. E_C during training over 2000 iterations. Figure represents training with no regularisation applied

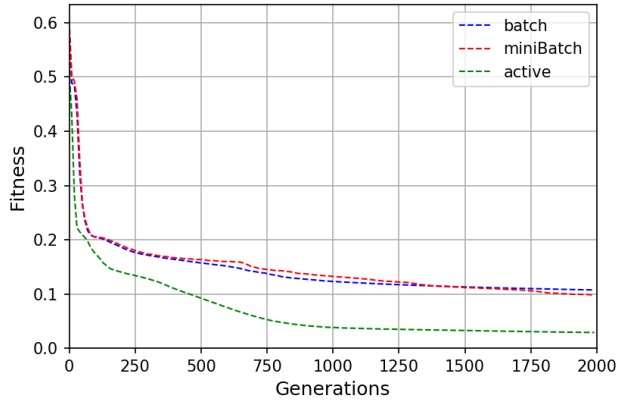

Fig. 7. E_C during training over 2000 iterations with a batch size of 2000. Figure represents training with no regularisation applied

VI. CONCLUSION

In conclusion active learning performs adequately on the SDSS dataset outperform the other training methods it was compared against. Regularisation schemes do not appear to have any substantial improvement on the generalization of Neural Network trained with active learning. Perhaps due to the small size of the Neural Network defined in this research the benefits of regularisation could not be observed.

Further research could expand the datasets trained using active learning as well as verify the effect dropout has on active learning with small batch sizes.

REFERENCES

[1] A. Rakitianskaia and A. Engelbrecht, "Measuring saturation in neural networks," in *Symposium Series on Computational Intelligence*, IEEE. IEEE, Dec. 2015. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7376778/

[2] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[3] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in neural information processing systems*, 1992, pp. 950–957.
[4] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[5] J. Shore and R. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Transactions on information theory*, vol. 26, no. 1, pp. 26–37, 1980.
[6] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, no. 2, pp. 201–221, May 1994. [Online]. Available: https://doi.org/10.1007/BF00993277