Study to the use of Neural Networks using Scikit-Learn

Prediction of Abalone Age Groups

Rahul Kumar
Masters Student *of Data Science*
University of New South Wales
z5341134@ad.unsw.edu.au

***Abstract* — The report analyses the various effects of hyper parameters on Scikit-learns MLP Classifier to determine the best model to predict Abalone age groups. The final model was decided to have 1000 hidden neurons, with 2 hidden layers and solved with the Adam algorithm. This resulted in a preliminary best model accuracy of 59.1%**

I. Introduction

Neural Networks was developed as far back as 1943, in which Warren McCulloch and Walter Pitts developed a computational model for neural networks [1]. The development of the algorithm known as back propagation by Paul J. Werbos in 1980 was the next big break through for Neural Networks. However once there was an interest in deep learning, was when the popularity of Neural Networks really took off. Their use are more widespread than one would believe:

- Speech Recognition
- Image Recognition
- Autonomous Vehicles
- Medical Applications such as disease recognition

Neural Networks and deep learning still have some disadvantages and challenges. Neural networks requires an immense amount of data, where both the test data and training data needs to be similar, such that relationships can be built between old and new answers [2]. Deep learning in so far cannot decipher concepts that it produces in its results [2]. And the biggest factor is that Neural Networks and Deep learning is the lack of transparency and the black box nature of the models, which is a major problem when tasked with problems such as medical diagnosis and financial concerns [2]. Another factor that should be known are the numerous gradient descent optimisation algorithms available to the community to learn from, examples being Adam, which is based on adaptive estimates of lower order moments (Kingma, Lei Ba ICLR 2015) [3]. The more common SGD. These two optimiser methods are part of the testing done within the report.

The aim of this project to understand the use of neural networks, especially how various hyperparameters can be tuned, how different algorithms, hidden layers and neurons can affect prediction accuracy.

This report primarily analyses the data set of Abalone obtained from UCI's Machine Learning Repository and study "The Population Biology of Abalone in Tasmania. I. Blacklip Abalone from the North Coast and Islands of Bass Strait" [4]. The MLP Classifier from Scikit-Learn will be primarily used for analysis, in which the hidden layers, hidden neurons per layer and solver algorithm will be analysed. Finally based on these studies a best model will be picked and evaluated with the use of a confusion matrix.

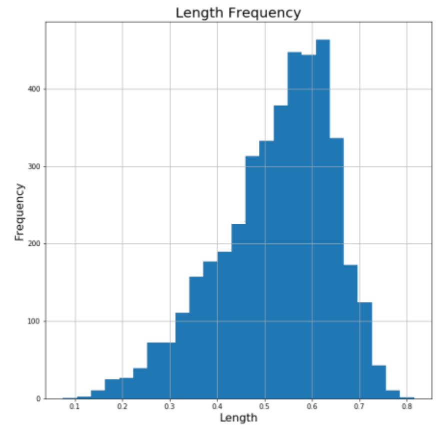
A. Overview

The paper is organised as follows, in Section II, the methodology and results of the study are given. Section III discusses the results and the validity of the methodology.

II. Results

A. Methodology

The database was discovered to be clean and structured, which is great for pre-processing. It resulted in less data clean up than anticipated. There were no missing values from the data source, and there were nine features that could



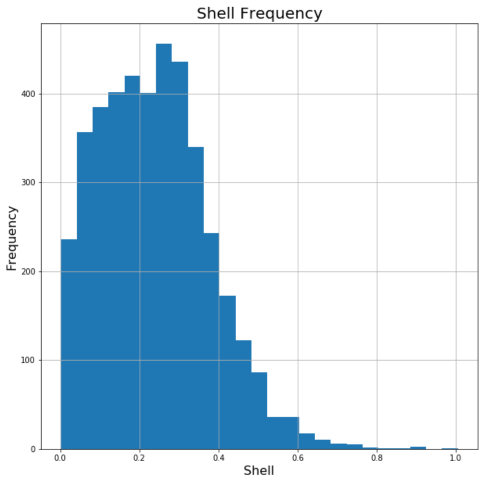


Figure 2. Shell Distribution

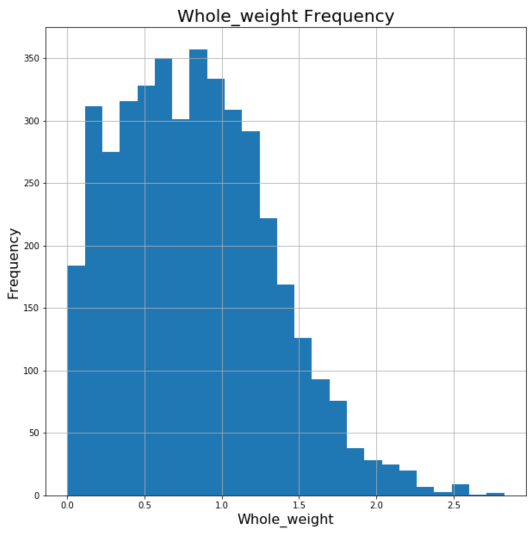


Figure 3. Weight Distribution

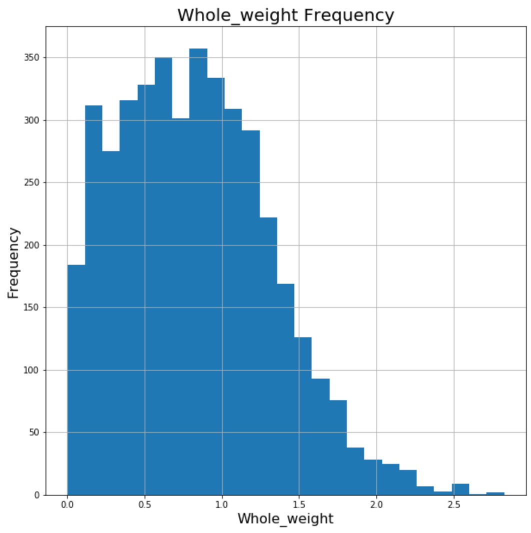


Figure 4. Diameter Distribution

be used to analyse, one which was determined to be the predicted value, which was Abalone age, or the Rings feature column. Looking at the raw data, the gender of abalone was well balanced however, majority of the remaining data was heavily skewed. This will introduce some bias in the training model. As it can be seen from Figures 1 through to 5, the diameters, weight shell and height distribution are heavily skewed to the left, while the length is skewed to the right. The data statistics can be seen on Table 6 in the Appendix

Table 7 located in the appendix displays the average mean across the genders, which helps in identifying specific relationships.

To simplify the data set into classification to satisfy our goal of understanding how to work with neural networks through

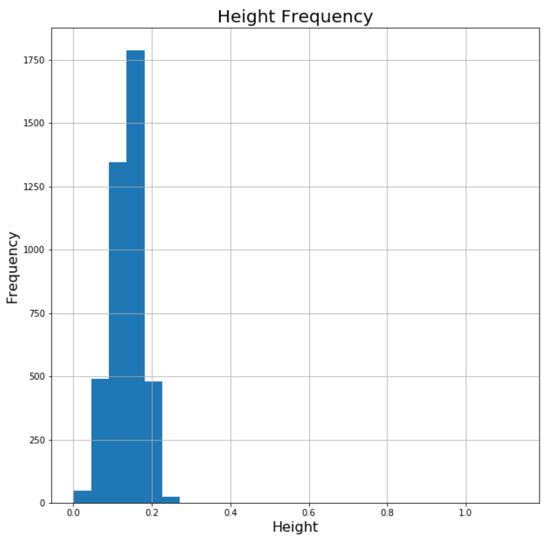


Figure 5. Height Distribution

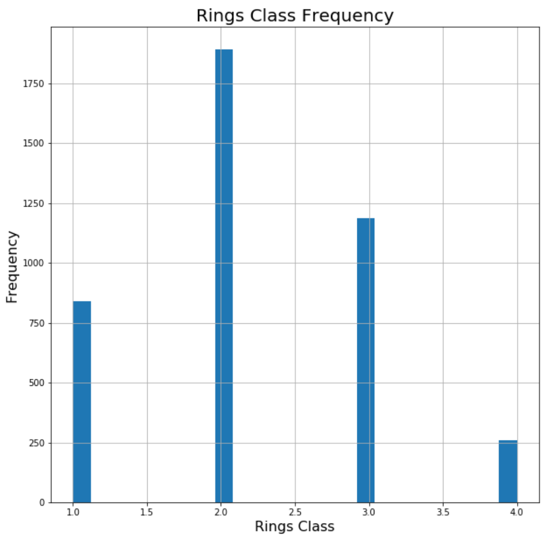


Figure 6. Ring Class Distribution

the use of MLP Classifier, the Ring Age was classified into 4 groups.

- Class 1: 0 - 7 years
- Class 2: 8- 10 years
- Class 3: 11 - 15 years
- Class 4: Greater than 15 years

After this was done, the distribution was plotted, which can be seen in Figure 6. Primarily prior the ring age were heavily skewed to the left, however the benefits of grouping the ages have normalised the distribution.

To discover some of the relationships between the dataset a seaborn pair plot was generated (Appendix, Figure 17). Looking at primarily the Rings, it can be seen that Height seems to have a high linear relationship, alongside length and diameter. Which leads to consideration that these features should be definitely considered as predictors. A correlation matrix was built to then confirm which would be the most ideal features to include for prediction.

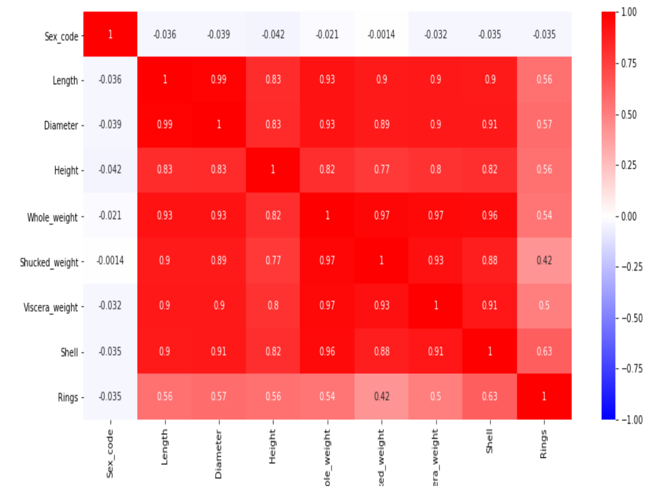


Figure 7. Data Correlation Heatmap

Some initial processing had to be done, in which the Sex (gender) was ordinally encoded, such that each category had a number, such that the correlation could be seen between rings and gender, this was saved as Sex_code. Expectedly, it can be seen that Abalone Sex has very little correlation to Abalone age. However, it is said that infancy can be correlated to various other factors such as abalone size and/or shell age [5]. As a result, it was still included in prediction, as a precautionary measure. To include the abalone sex, one-hot encoding was carried out, and new features were created, labelled as Sex_F, Sex_M, and Sex_I, where a value was 1 in the corresponding column and 0 in the others.

The final step of data preparation is normalising or scaling the data, to do this a simple linear scaling was applied with MinMaxScaler() from Scikit-Learn, as it is pre-processed scaling is always important [6]. Scaling in-fact ensures that your input values have the same weight as machine learning models are unable to differentiate between values [2].

The experiment was broken into the following tests cases. The first case was to tune the hyperparameter number of hidden neurons and analyse the effect of it. The number that showed the best performance, in which the metric used was end accuracy, was then used in the following test cases. The second test case was to analyse the effect of learning rate when using a SGD algorithm, after which the third test case was to analyse the effect of the number of hidden layers used in the model, again this was done with optimal number of neurons. Finally an analysis was done on the effect of different algorithms, primarily SGD and Adam.

The best performance tests across these would then be used on a final best model, in which the performance would be tested using a confusion matrix. To ensure the best solution, each case was run for 10 experimental runs.

B. *Results*

The results for each of the test case in tabular form is provided in the Appendix.

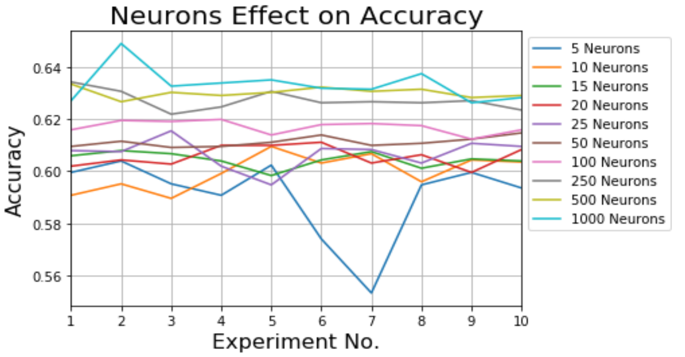


Figure 8. Effect of Hidden Neurons

Case 1: The Effect of Hidden Neurons

The accuracy variation of the number of hidden neurons can be noted in Figure 8 across each experiment. Figure 9 shows that as the number of neurons increase, the accuracy increases. The following table (Table 1.) shows the confidence interval for the last three test cases, for all the cases considered, Table 8 Can be found in the appendix.

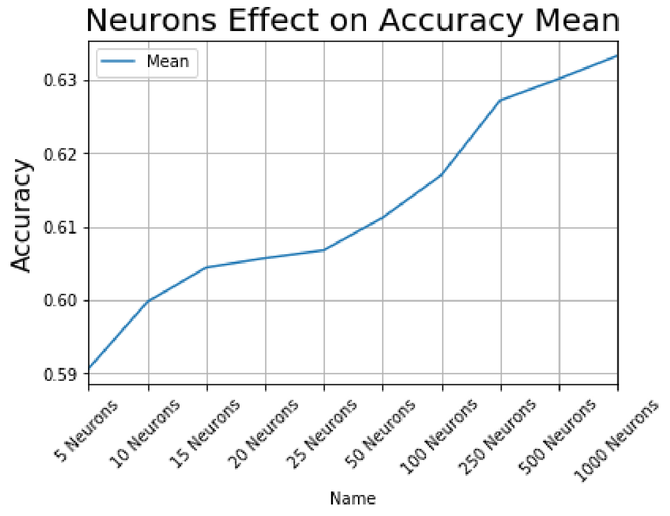


Figure 9. Effect of Hidden Neurons on Mean Accuracy

Name	250	500	1000
Mean	62.72%	63.01%	63.33%
Lower Bound	62.45%	62.87%	62.86%
Upper Bound	62.98%	63.15%	63.80%

Table 1. Mean and 95% Confidence Interval for Case 1

It can be seen that as the number of Neurons increase, the higher the accuracy, as a result, the optimal number of Neurons selected was 1000.

Case 2: The Effect Learning Rate

Learning Rate effect was an interesting analysis, as the expected effect of as Learning Rate was decreased in value, i.e. made close to 0, the better the accuracy, however this was not the case in our observations, seen in both Figures

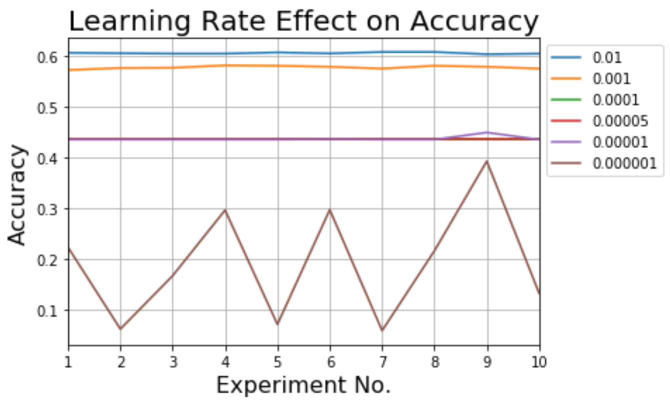


Figure 10. Effect of Learning Rate

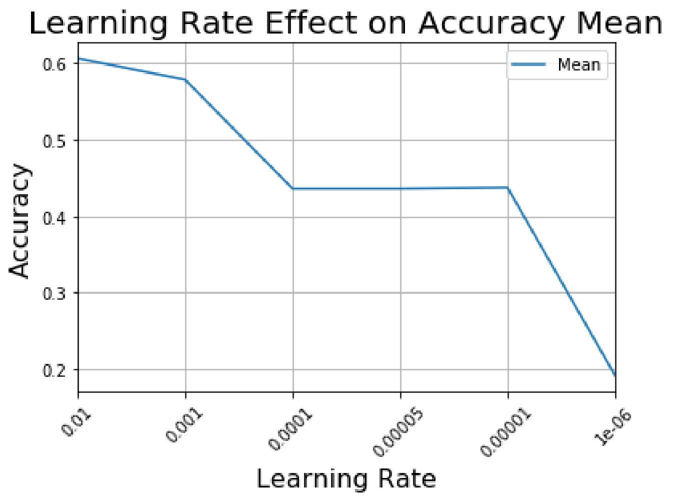


Figure 11. Effect of Learning Rate on Mean Accuracy

10 and 11. This was a surprise, as learning rate made smaller was expected to make the accuracy better as learning rate affects whether our model converges correctly.

Table 3. shows the average accuracy and Confidence Interval for the most interesting cases, which were a learning rate of 0.01, 0.001, 0.00005 and 0.000001.

Name	0.01	0.001	0.00005	0.000001
Mean	60.65%	57.85%	43.60%	19.21%
Lower	60.55%	57.63%	43.60%	11.04%
Upper	60.76%	58.06%	43.60%	27.38%

Table 2 Mean and 95% Confidence Interval for Case 2

Case 3: The Effect of Hidden Layers

The effect of Hidden Layers and variation of accuracy can be seen in Figure 12, which is more chaotic. However when looking at the Figure 13, it can be seen that at 1000 Neurons, having two hidden layers gives the best accuracy for the test runs made. With a sharp drop in accuracy when running with 3 layers. It should be noted that with an increase in layers computational time drastically increased.

Table 3 shows the mean and 95% confidence interval for all the 4 tests, and it can be seen that having 2 layers is indeed the most ideal case for this data set.

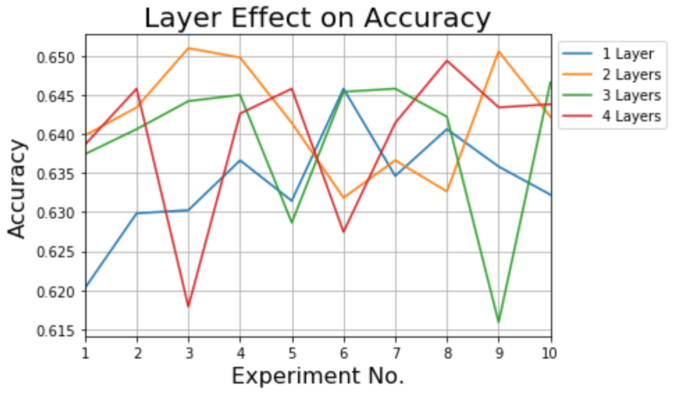


Figure 12. Effect of Hidden Layers

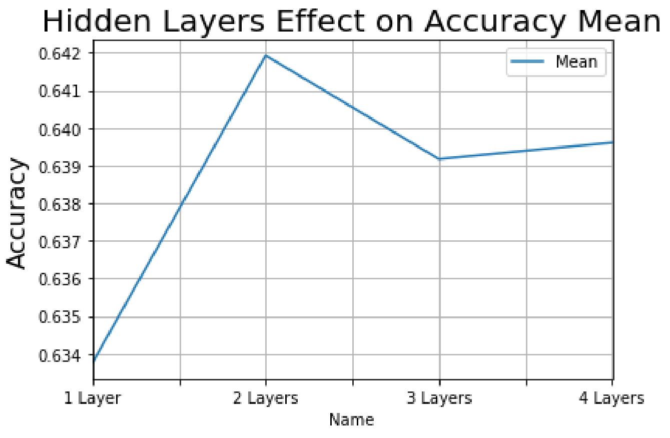


Figure 13. Effect of Hidden Layers on Mean Accuracy

Name	1 Layer	2 Layers	3 Layers	4 Layers
Mean	63.37%	64.19%	63.92%	63.96%
Lower	62.88%	63.69%	63.22%	63.27%
Upper	63.86%	64.69%	64.62%	64.65%

Table 3. Mean and 95% Confidence Interval for Case 3

Case 4: The Effect of Algorithms

Two algorithms were analyzed in this case, Adam and SGD.

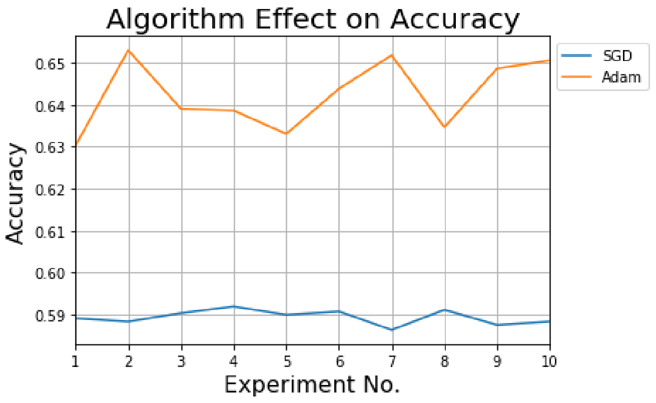


Figure 14. Effect of Algorithm

Name	SGD	Adam
Mean	58.94%	64.23%
Lower	58.81%	63.63%
Upper	59.06%	64.83%

Table 4. Mean and 95% Confidence Interval forCase 4

It can be seen that between the two, the Adam algorithm has a large variance between experiments, while SGD has a lower variance. However, the accuracy is completely different for the two, in Table 4 it can be seen that Adam has the higher mean accuracy.

The Loss vs Epoch can be shown between the two solvers to examine the difference between learning with each model algorithm and is shown in Figure 15.

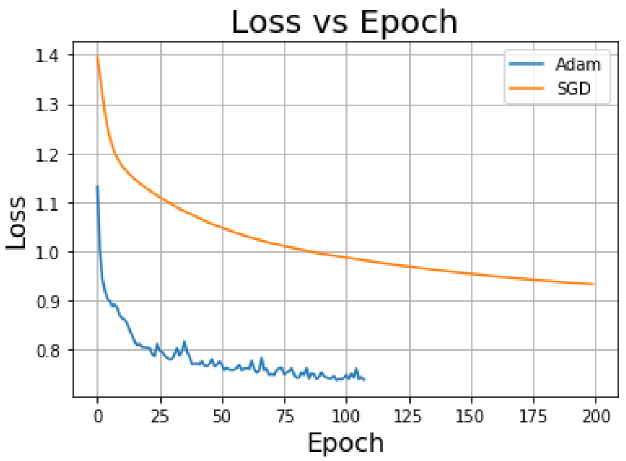


Figure 15. Effect of Learning between Algorithm

Based on each case study the best hyper parameters were chosen to discover the best model for predicting values. 1000 Neurons were chosen as the number of Hidden Neurons, there were two hidden layers and the Adam algorithm was used.

The model is evaluated using a confusion matrix and a ROC Curve is displayed for each Classification shown in Table 5 and Figure 16.

	Precision	Recall	f1-score	Support
Class 1	0.821	0.706	0.760	521
Class 2	0.538	0.836	0.655	1093
Class 3	0.553	0.267	0.361	744
Class 4	0.000	0.000	0.000	149
Accuracy	-	-	0.591	2507
Macro Avg	0.478	0.453	0.444	2507
Weighted Avg	0.569	0.591	0.550	2507

Table 5. Confusion Matrix Classification Report For Best Model

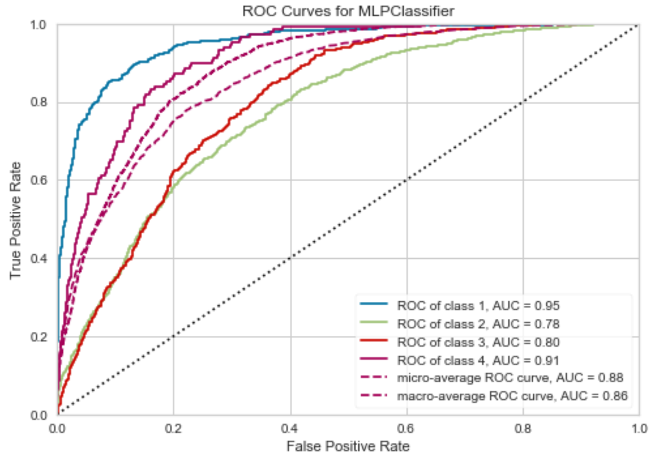


Figure 16. ROC AUC Graph of Best Model

It can be noted that the model at this stage can be improved further as the overall accuracy and precision and recall for specific classes can be improved further.

III. Discussion

The experiment and analysis shows a variety of results which were expected and some which were not entirely expected. Looking at Case 1, it was expected that the more neurons a model has, the better the accuracy. The downside to having 1000 neurons were the computational time that the model ended up using to learn from the test data. So, the tradeoff between accuracy is processing time which can be improved with access to better computational power. The variations between experiments were high for a very small amount of neurons, and to ensure the best reproducibility, would be to improve the number of neurons. To improve analysis, the random state of weight and bias initialization should be set, as such the true effects of neurons can be identified, which can build some consistency between experiments.

With analysis between various learning rates, the experimental results were a genuine surprise. It was expected that the with decreasing of learning rate significantly improves the generalization of accuracy [7]. The larger the learning rate the higher the chance of over correcting [7]. To improve on this case, it would be much better to analyse the effect of learning rate on Loss and Epoch. Using learning rates much greater than 0.01 as done by Wilson and Martinez [7] to values of 100, 50 and 10 would give a much better indication of effects. It would be prudent to analyse the learning rate based on the Sum Square Error. It would allow a decision to be made on that the learning rate of 0.01 being the ideal rate for a data set of this size.

The analysis of Hidden Layer was as expected, there would be a peak of accuracy for an ideal number of hidden layers. There is belief that implementing less than 3 layers in a neural network will result in a loss of accuracy, however with an increase in more than 3 will result in large computational times [8]. This was noted in the experimental run over here, the analysis for 2 layers took 1 minute with 1 Layer and 1000 neurons, however with 4 layers the computational time took well over 10 minutes. The methodology for testing could be improved by analyzing the effects on a variety of metric scores than accuracy, and if time permitted, on a larger number of layers.

The final effect of neural networks analyzed was the effect of algorithm chosen. It can be seen that there is a large difference in accuracy between the SGD method and Adam, this is understandable as SGD updates its model parameters according to the gradient loss, based of a set learning rate of one sample, while Adam adapts the learning rate for each parameter [9]. However this is different compared to the results provided by Wang and Wiens, where SGD has fractionally better accuracy than Adam. The difference could be attributed to the data set in itself. To improve the analysis done in this report, it would be better off to analyze a variety of metrics as done by Wang and Wiens, alongside more algorithms such as AdaSGD, AdaBound.

This final section of the discussion will go report about the best model. It can be seen that the final accuracy is very low, sitting at 59.1%, with Class 1, having the highest Precision, recall and f1score. Class 2, has passable metrics, with Class 3 and 4 having terrible Precision, recall and f1. This would primarily be due to the heavy skewed data set. This can be vastly improved by balancing the data set through various means, and doing hyper parameter tunings with GridSearchCV for the most ideal combination compared to just choosing the best values for each hyper parameters. Furthermore, more analysis on the best hyperparameters can be done, with a variety of metrics such as precision and recall.

The ROC for Class 1 is exceptional, however it can be seen for 2 and 3 it is much closer to the 45 degree diagonal. The AUC gives a great indication that Class 2 and Class 3 can be focused and improved upon, again using a better combination of hyper parameters. This is based on carrying out numerous more neural network modelling, for a variety of more hyperparameters than what was tested in this report. There are numerous methodologies to improve this process such as using initialization strategies such as the normalized variance preserving initialization for the activation functions.[10]

The biggest issue to improve prediction and improve training capabilities would be to use, a more balanced data set, a larger number of experimentations and spending time optimizing the best model.

IV. Conclusion

In conclusion, having a large number of neurons and good number of hidden layers can really improve the accuracy of a neural network. However the decision of algorithm is very important. Neural networks are vastly different from other Machine Learning models, in that majority of developing the ideal model is trial and error, based on the data set you are using. All this relies on the delicate balance between computational time and accuracy. The final model in this case only provided an accuracy of 59.1%, however this can still be further improved with enough time.

References

[1] S. K. Sarvepalli, “Deep learning in neural networks: The science behind an artificial brain.” Unpublished, 2015. Retrieved from https://www.researchgate.net/publication/331400258_Deep_Learning_in_Neural_Networks_The_science_behind_an_Artificial_Brain

[2] G. Marcus, “Deep learning: A critical appraisal,” *arXiv [cs.AI]*, 2018. Retrieved from https://arxiv.org/ftp/arxiv/papers/1801/1801.00631.pdf

[3] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv [cs.LG], 2014. Retrieved from https://arxiv.org/pdf/1412.6980.pdf

[4] Warwick J Nash, Tracy L Sellers, Simon R Talbot, Andrew J Cawthorn and Wes B Ford (1994) "The Population Biology of Abalone (_Haliotis_ species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait", Sea Fisheries Division, Technical Report No. 48 (ISSN 1034-3288)

[5] D. Tarbath, “Estimates of Growth and Natural Mortality of the Blacklip Abalone in Tasmania,” Technical Report Series, no. 3, Dec. 1999. Retrieved from https://www.imas.utas.edu.au/__data/assets/pdf_file/0007/1172914/Tech_Report_3_AbaloneGrowthMortality.pdf

[6] C. M., Neural Networks for Pattern Recognition. Oxford, England: Clarendon Press, 1995. Pg 296.

[7] D. R. Wilson and T. R. Martinez, “The need for small learning rates on large problems,” in IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222), 2002. Retireved from https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=2092&context=facpub

[8] M. Uzair and N. Jamil, “Effects of Hidden Layers on the Efficiency of Neural networks,” in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 2020. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9318195

[9] J. Wang and J. Wiens, “AdaSGD: Bridging the gap between SGD and Adam,” *arXiv [cs.LG]*, 2020. Retireved from https://arxiv.org/pdf/2006.16541.pdf

[10] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Training very deep networks,” *arXiv [cs.LG]*, 2015. Retrieved from https://proceedings.neurips.cc/paper/2015/file/215a71a12769b056c3c32e7299f1c5ed-Paper.pdf

Appendix

	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell	Rings
count	4177	4177	4177	4177	4177	4177	4177	4177
mean	0.5240	0.4079	0.1395	0.8287	0.3594	0.1806	0.2388	9.9337
std	0.1201	0.0992	0.0418	0.4904	0.2220	0.1096	0.1392	3.2242
min	0.0750	0.0550	0.0000	0.0020	0.0010	0.0005	0.0015	1.0000
25%	0.4500	0.3500	0.1150	0.4415	0.1860	0.0935	0.1300	8.0000
50%	0.5450	0.4250	0.1400	0.7995	0.3360	0.1710	0.2340	9.0000
75%	0.6150	0.4800	0.1650	1.1530	0.5020	0.2530	0.3290	11.0000
max	0.8150	0.6500	1.1300	2.8255	1.4880	0.7600	1.0050	29.0000

Table 6. Abalone Data Description

Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell	Rings
F	0.58	0.45	0.16	1.05	0.45	0.23	0.30	11.13
I	0.43	0.33	0.11	0.43	0.19	0.09	0.13	7.89
M	0.56	0.44	0.15	0.99	0.43	0.22	0.28	10.71

Table 7.. Gender Average Statistics

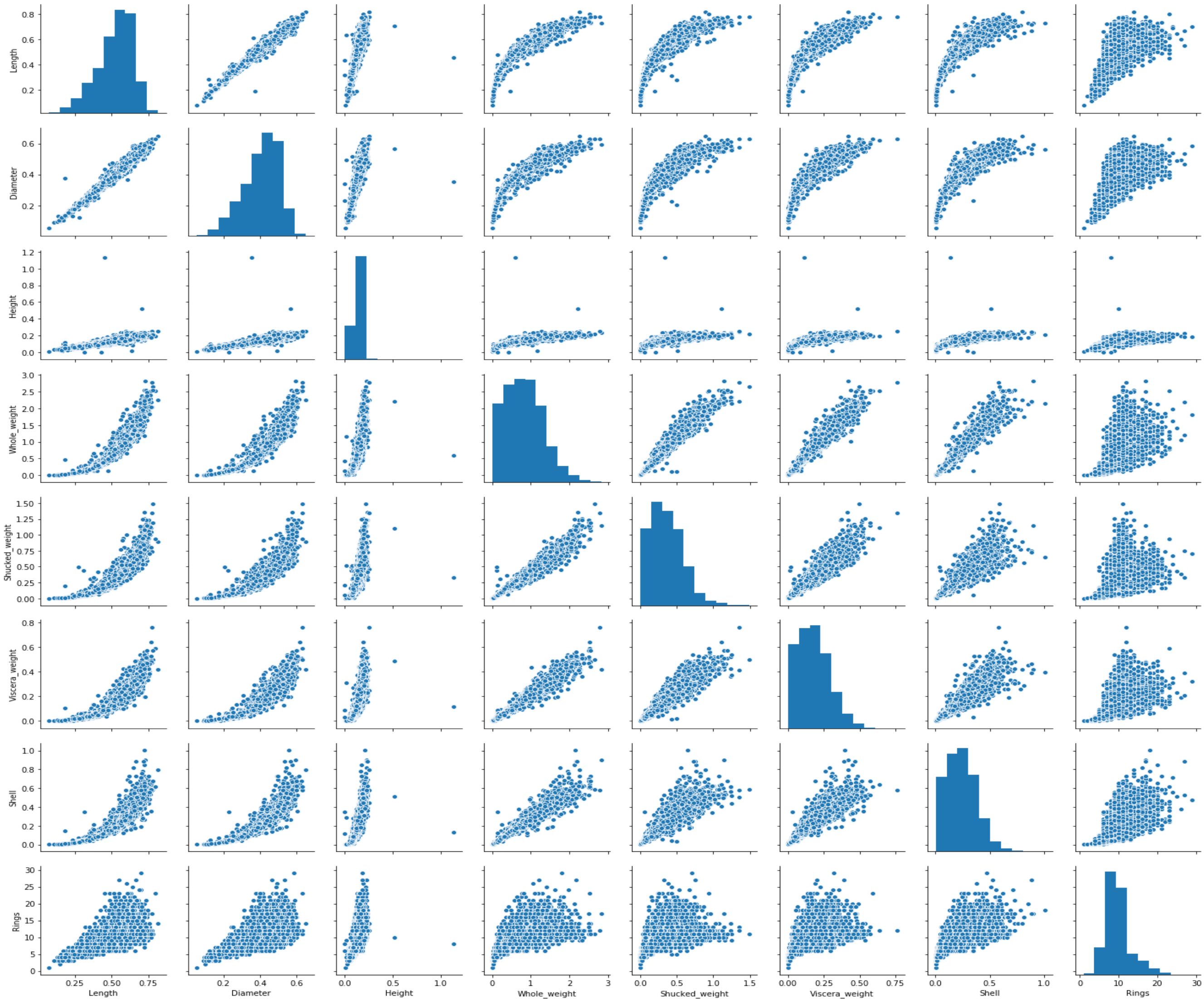
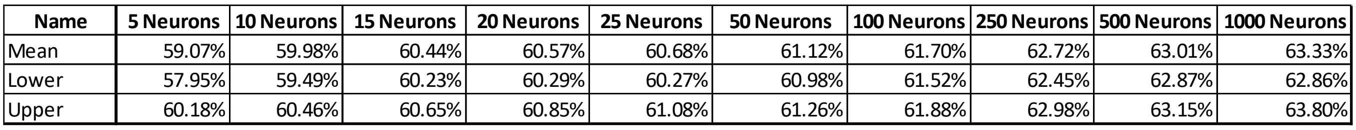


Figure 17. Data Set Pairplot



Name	5 Neurons	10 Neurons	15 Neurons	20 Neurons	25 Neurons	50 Neurons	100 Neurons	250 Neurons	500 Neurons	1000 Neurons
Mean	59.07%	59.98%	60.44%	60.57%	60.68%	61.12%	61.70%	62.72%	63.01%	63.33%
Lower	57.95%	59.49%	60.23%	60.29%	60.27%	60.98%	61.52%	62.45%	62.87%	62.86%
Upper	60.18%	60.46%	60.65%	60.85%	61.08%	61.26%	61.88%	62.98%	63.15%	63.80%

Table 8. Complete Case 1 Data