Foundations of Neural Networks

Sheng-Chi, Huang
Master of Information technology
University of New South Wales
Changhua, Taiwan
z5471540@ad.unsw.edu.au

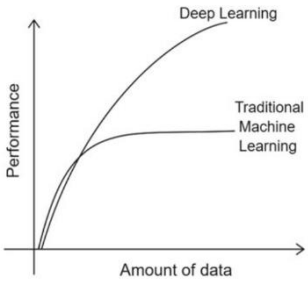
***Abstract*—Abalone is a rich nutrition food resource found in cold coastal regions. Its price is highly correlative to its age. However, it is difficult to determine the age of abalone as it needs to go through a detailed process. This project applies various deep learning methods and analyze different algorithms to propose an accessible way to determine the age of abalone.**

Keywords—Abalone datasets, Neural Networks, Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam), learning rate (lr), Root Mean Square Error (RMSE)

I. INTRODUCTION

As a type of consumable snail, abalone's price varies and has highly correlated relationship to its age. While it is an expensive procedure to determine the age of abalone which goes through the process of cutting the shell, staining, and counting the number of rings under a microscope. This increases the costs of both producers and consumers. This report adopts deep learning (supervised learning) technologies to reduce the cost of predicting the age of abalone.

The performance of traditional machine learning algorithms become stable when it reaches the threshold of training data, while deep learning can improve its performance with increasing amount of data [2].



[2]

In this project, we figure out the best hyperparameters and functions to fit into our model and choose the best result to build the models. We also apply various methodologies to lower the loss. Through the process, we achieve the loss (RMSE) to 2.01.

II. NEURAL NETWORK

A. *About the Data*

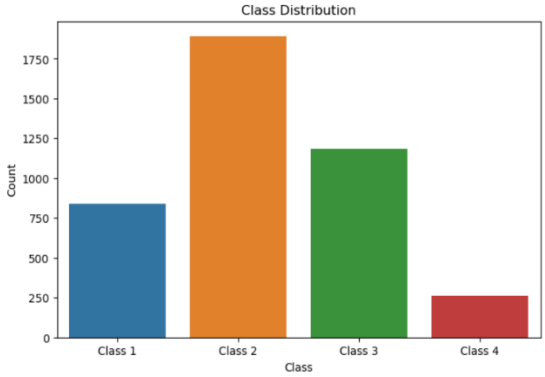
Firstly, we map the feature 'sex' into number ('M': 0, 'F': 1, 'I':2), and simply define the four classes.

Class 1: 0-7 years

Class 2: 8-10 years

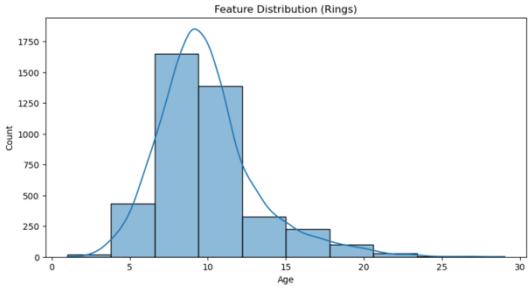
Class 3: 11-15 years

Class 4: greater than 15 years



Class	number
Class 1	839
Class 2	1891
Class 3	1186
Class 4	261
dtype: int64	

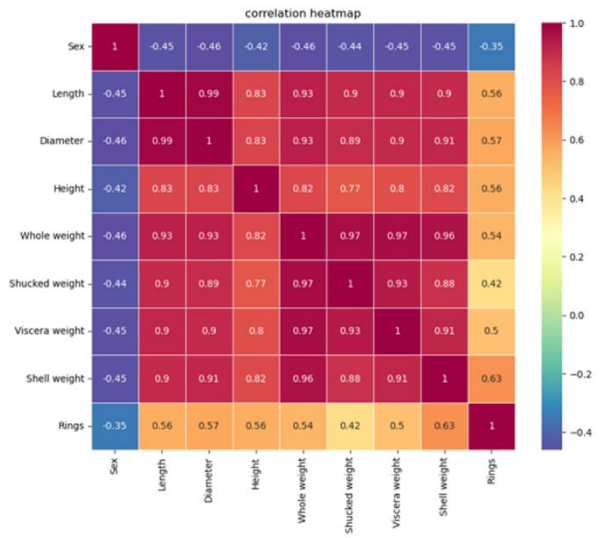
We can see that class 2 has the most count and follow by class 3, which is a normal distribution.



From the distribution of the age count graph, we can see that the majority of the instances are around 10 rings.

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	class
1257	2	0.430	0.34	0.0	0.428	0.2065	0.0860	0.1150	8	Class 2
3996	2	0.315	0.23	0.0	0.134	0.0575	0.0285	0.3505	6	Class 1

In the dataset, two of the height value are 0, [3] I use the mean of height to replace the 0. After reviewing the dataset, we are ready to move the next phase.



B. *Tuning the Hyperparameters*

1) Number of neurons

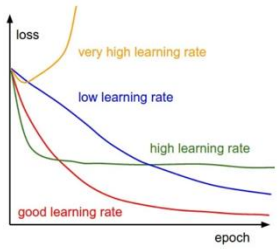
We first experiment on different numbers of the neurons to find an optimal number which maximize the performance. With all the other hyperparameters fixed, we adjust the number of the neurons to compare the result. With experimentations, we find that 20 hidden neurons (using ReLU, learning rate=0.01) has the best performance. However, there is an interesting obsersvatoin that when we change the activation funnction into tanh (lr = 0.01), the result would be slightly better.

```
best number of neuron:  20
rmse mean:  2.50813083052231
95% confindence interval: 2.489784267444818, 2.526477393599802
best number of neuron:  20
rmse mean:  2.551101329219123
95% confindence interval: 2.534345619780197, 2.5678570386580493
```

This can be related to several factors such as learning rate or the number of layer but normally, we use ReLU in hidden layer when dealing with classification task. So that we keep using ReLU as our activation function in later approaches.

2) Learning Rate

Learning rate is one of important hypermeters in neural network. As we can see in the figure [4], a good learning rate significantly reduce the loss. Therefore, we decide to set a range of number from 0.001 to 0.5 to find the best learning rate for our model.



[4]

```
best learning rate: 0.1, rmse: 2.4760975840948802
rmse mean:  2.607635006091377
95% confindence interval: 2.566590366555312, 2.6486796456274417
best learning rate: 0.1, rmse: 2.3802247327879993
rmse mean:  2.5999300695702297
95% confindence interval: 2.5576830126060353, 2.642177126534424
```

We iterate the learning rate by using the optimal number of hidden neurons (20) from the previous step. In this task, we use six numbers 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 to obtain the lowest RMSE value. And we find 0.1 is the optimal learning rate.

3) Number of Hidden Layers

A single hidden layer with a finite number of neurons can approximate continuous functions under reasonable assumptions about the activation function [5]. So we tend to keep the hidden layer as simple as possible.

```
{1: 2.401874392728322, 2: 2.344123125096761, 3: 2.237271998754138, 4: 2.3976349807461657}
best layer 3
rmse mean:  2.4202179906808716
95% confindence interval: 2.3881947273101636, 2.4522412540515797
```

During our process, we find that two hidden layers performs the best on RMSE.

Moreover, if we try the number of hidden layers from 1 to 4, we can see that the three-hidden-layers is the best among them. Based on this, we observe that the loss does not decrease when the number of layer increases.

Generally, a small dataset does not require a hidden layers more than five to achieve a high accuracy. Increasing the number of hidden layers may result in lowering the accuracy as the efficacy of the backpropagation will diminish. Adding more hidden layers to a neural network can indeed lead to overfitting, especially when the model becomes too complex [6].

```
{'L2 Strength': 0.0001, 'RMSE': 2.347312149406305}
best L2 0.0001
rmse mean:  2.4732499043482705
95% confindence interval: 2.4476794143759415, 2.4988203943205995
```

4) L2 Regularization

Now we try four different L2 regularization values: 0.0001, 0.001, 0.01, and 0.1. We can see that among these values, 0.0001 performs the best. L2 regularization helps to control the complexity of the neural network by adding a penalty to its weights. In this case, the best parameter 0.0001 shows that we apply a slight penalty to the weights without overly constraining their changes.

```
best learning rate: 0.1, rmse: 2.3802247327879993
rmse mean:  2.5999300695702297
95% confindence interval: 2.5576830126060353, 2.642177126534424
best learning rate: 0.1, rmse: 2.4760975840948802
rmse mean:  2.607635006091377
95% confindence interval: 2.566590366555312, 2.6486796456274417
```

These two models have the same hyperparameters without L2 regularization. Comparing these two models with the model in last section, we can see there is a significant improvement when using L2 regularization. By applying L2 regularization, we can prevent individual features from dominating the model and overfitting.

5) Adam And SGD

Before looking at the result, we introduce optimization algorithms: SGD and Adam. Stochastic Gradient Descent (SGD) uses mini-batch to calculate the gradient while Adam combines the ideas of momentum which helps in faster

convergence. So that we can assume Adam may has better performance later.

```
adam rmse mean:  2.4915058101376686
agd rmse mean:  2.502881662749396
adam 95% confindence interval: 2.4316222302123056, 2.5513893900630316
sgd 95% confindence interval: 2.4668097137700964, 2.538953611728696
```

After training, the RMSE loss from Adam and SGD are 2.491 and 2.502 respectively, which is same as our assumption.

```
adam rmse mean:  2.451550995451429
agd rmse mean:  2.4660532053654403
adam 95% confindence interval: 2.3889997523393776, 2.5141022385634804
sgd 95% confindence interval: 2.4147472897268516, 2.517359121004029
```

When the training iterations extended to 10,000 times, with the other hyperparameters fixed, both Adam and SGD have low RMSE losses. Surprisingly, SGD outperformed Adam with achieving a lower loss. This result highlights the effect of training time on the choice of optimizer.

This observation indicates that the training time is a crucial factor in determining the choice between Adam and SGD. However, it's important to recognize that the performance of an optimization algorithm can still be subject to the unique characteristics of the dataset.

Additionally, there is a new strategy 'SWATS' that combines Adam and SGD. It is a straightforward approach that shifts from Adam to SGD upon meeting a specified triggering condition. This condition is linked to the projection of Adam steps onto the gradient subspace [7].

III. Model

Based on the iterative processes so far, we find the optimal hyperparameters and functions as below:

Number of hidden neurons: 20
Learning rate: 0.1
Number of hidden layers: 2
L2 normalization: 0.0001,
Optimizer: Adam

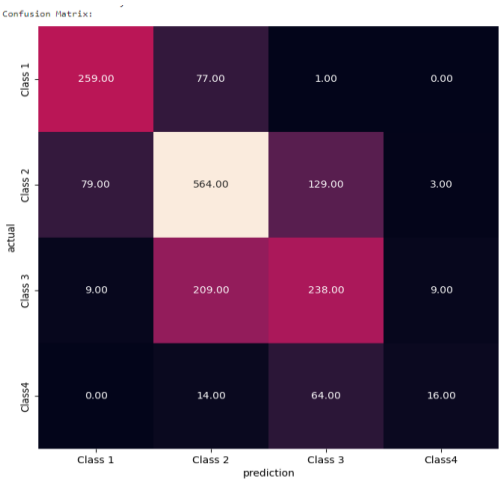
Now we try two different activations:

1) ReLU:

```
Classification Accuracy: 63.3154%
```

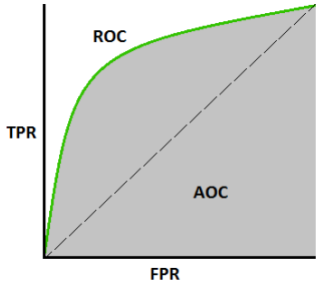
2) Tanh:

```
Classification Accuracy: 64.4524%
```

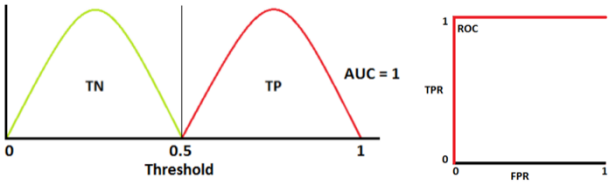


The confusion matrix shows the model does not perform very well on class 4. In contrast, the recall rate reaches 76.85% in class 1, 72.77% in class 2 and 51.18% in class 3. The reason is that the numbers of the instances in each class are imbalance. We only have 261 samples in class 4 (minority classes) in the training set., and the training data would just have more less. It is challenging to train a model with class imbalance. Also, due to the limited samples in class 4, the model barely classified the instances into class 4. Many studies have explored whether class imbalances hinder classifier induction or if other factors might explain these deficiencies. Some strategies such as clustering can be more effective, especially in the small dataset and when the problem is complicated [8].

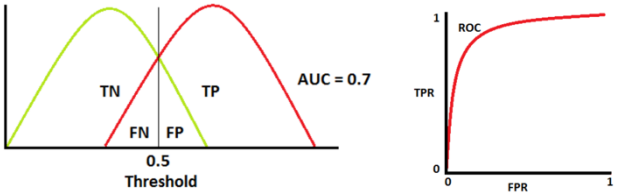
1) Area Under the ROC Curve and Receiver Operating Characteristic



The Area Under the ROC Curve (AUC) is a metric used to evaluate the performance of classification models. ROC, Receiver Operating Characteristic, is a graphical tool for visualizing the performance at different classification thresholds, telling how much the model is capable of distinguishing between the classes.

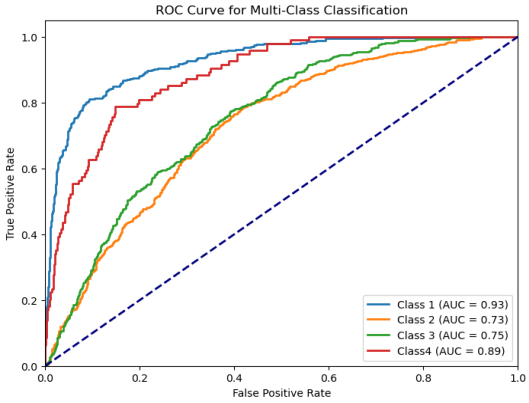


[10]



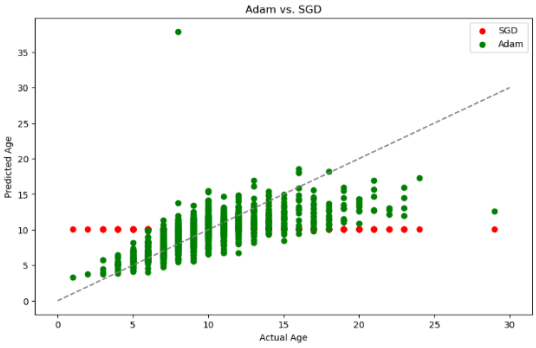
[10]

The figures show that when AUC is 1.0, there is 100% chance to distinguish the class correctly. If there is an overlap, for instance, AUC is 0.7, which means the model have 70% chance to correctly distinguish the class [9]. However, AUC equally treats different types of errors and does not reflect the asymmetry in error costs. Also, it only provides ranking information without considering specific predicted probabilities [10].



In this graph, we can see that class 4 still has a high AUC value. The ROC curve is constructed by plotting the true positive rate (TPR) and false positive rate (FPR). If the ROC curve for Class 4 rises at high FPR values, the AUC can still be high even if the misclassification rate is high.

IV. Regression Model



1000 iteration:

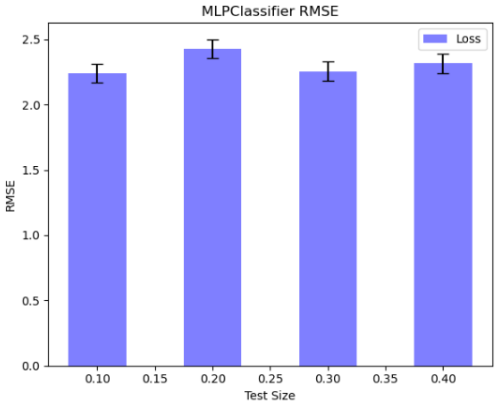
```
Adam RMSE mean: 2.2465604122887144
Adam 95% confidence interval: 2.189062935451165, 2.304057889126264
SGD RMSE mean: 3.2560022919065963
SGD 95% confidence interval: 3.2213740911522146, 3.290630492660978
```

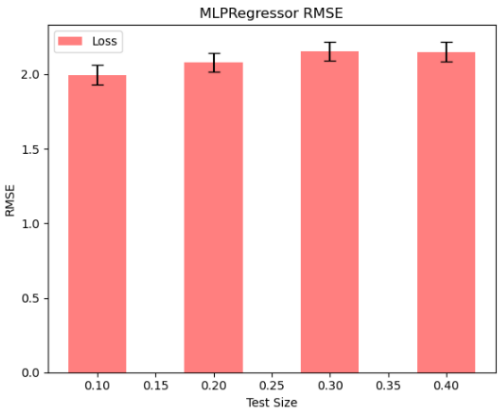
10000 iteration:

```
Adam RMSE mean: 2.2186211188405274
Adam 95% confidence interval: 2.159920307726752, 2.2773219299543026
SGD RMSE mean: 3.254061902741207
SGD 95% confidence interval: 3.221995191963703, 3.2861286135187115
```

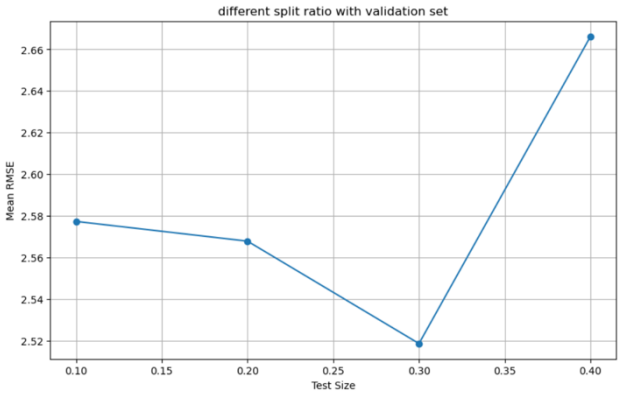
Apparently, the loss become more when we use regression in this case. MLPRegressor is trained to predict continuous numerical output. Both of MLPClassifier and MLPRegressor are supervised learning model which employ multi-layer perception.

V. Feature Additional Visualization and Hybrid Dropout and Weight Decay





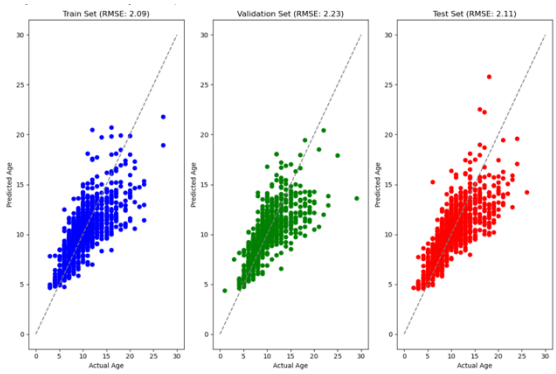
We try to know the different ratio to split test and train set in regression and classify model.



Here we plot the differences between different train/test split set. with validation set (0.2). The points are (0.1, 2.5773), (0.2, 2.5678), (0.3, 5186), and (0.4, 2.6662).

```
32/32 [==============================] - 0s 1ms/step
32/32 [==============================] - 0s 1ms/step
32/32 [==============================] - 0s 1ms/step
32/32 [==============================] - 0s 1ms/step
32/32 [==============================] - 0s 1ms/step
32/32 [==============================] - 0s 1ms/step
32/32 [==============================] - 0s 1ms/step
32/32 [==============================] - 0s 1ms/step
32/32 [==============================] - 0s 1ms/step
Best dropout rate: 0.1
Best weight decay (λ): 0.001
```

We use TensorFlow to adopt dropout in our model and create two list:
dropout_rates_list = [0.1, 0.2, 0.3],
weight_decays_list = [0.001, 0.01, 0.1]
to find out the best value for the model.



We use the best values found from the last step and build the model. The figure shows the prediction of training, validating and testing set.

```
RMSE with drop out and weight decay: 2.106494024131662
```

The RMSE is 2.10649 after the validation set, dropout and weight decay are adopted with comparison to the model without these processes, the RMSE was 2.4.

VI. Discussion

In this report, we have applied various functions to construct our model and fine-tune the best hyperparameters. The process of setting the hyperparameters and designing the network structure often requires a considerable amount of trials. So that we try to keep the structure simple to control these hyperparameters [11].

Moreover, after doing this research, we recognize that tuning hypermeters is not only about adjusting one hyperparameter, but interatively experiment with all various configurations. In other word, the optimal value in one specific condition does not fit all.

By completing this project, we have gained a deeper understanding of machine learning. Machine learning is finding a balance between selecting appropriate functions and trying the hyperparameters to get the lowest loss. In this case, we utilized the MLPClassifier and find the most suitable hyperparameters for it. In week 9, the professor introduced the concept of clustering, triggering our interest in a potentially valuable approach for our dataset. Nevertheless, the initial experimentation with K-means clustering yields a higher loss compared to my previous model. This can be attributed to solving this classification task. While we only have a limited dataset with 4177 instances, it is insufficient to adopt K-means here. As a result, this project continues with the MLPClassifier and conducts a comparative analysis with neural networks

VII. Conclusion

In this multi-classification task, we conclude that classifier is better than regressor in determining the age of abalone. We use a large learning rate and two hidden layers with 20 neurons. The result is acceptable and we have summarized some key points: 1) As the dataset is not very large, using a shallow neural network to prevent distortion of the original data and to avoid selecting too many neurons to maintain data integrity is ideally. 2) In terms of the number of epochs, we can set a large number at first and use an early stop to halt the training when the result is not going to improve. 3) Adequate dataset size and balanced class distribution are important to develop an accurate model. Overall, the classifier's performance work fair. The number of instances is limited to train the models and the class imbalance also affects the training process. It's a good idea to implement dynamic construction of neural networks in the future to solve similar problems.

References

[1] R. Tibshirani, "[neural networks: A review from Statistical Perspective]: Comment," *Statistical Science*, vol. 9, no. 1, 1994. doi:10.1214/ss/1177010645

[2] A. E. Hassanien, A.-B. M. Salem, R. Ramadan, and T. Kim, Advanced Machine Learning Technologies and Applications First International Conference ; Proceedings. Berlin: Springer, 2012.

[3] D. Buntarto, "Classification method for estimating the numbers of rings of abalone," Medium, https://medium.com/analytics-vidhya/classification-method-for-estimating-the-numbers-of-rings-of-abalone-bb13264dd186.

[4] Lavanya Shukla, "Designing Your Neural Networks," Medium, Sep. 23, 2019. https://towardsdatascience.com/designing-your-neural-networks-a5e4617027ed

[5] P. Raut and A. Dani, "Correlation between number of hidden layers and accuracy of artificial neural network," *Algorithms for Intelligent Systems*, pp. 513–521, 2020. doi:10.1007/978-981-15-3242-9_49

[6] J. Heaton, "The Number of Hidden Layers," Heaton Research, Dec. 28, 2018. https://www.heatonresearch.com/2017/06/01/hidden-layers.html

[7] Nitish Shirish Keskar, Richard Socher: Improving Generalization Performance by Switching from Adam to SGD. 2017(https://arxiv.org/abs/1712.07628v1)

[8] Kanellopoulos, D. (2006). Handling imbalanced datasets: A review. p.12.

[9] S. Narkhede, "Understanding AUC - ROC Curve," Medium, Jun. 15, 2021. https://medium.com/towards-data-science/understanding-auc-roc-curve-68b2303cc9c5

[10] Madong, "Auc disadvantage Zhihu, https://zhuanlan.zhihu.com/p/92792702(https://48hours.ai/files/AUC.pdf)

[11] L. N. Smith, "A DISCIPLINED APPROACH TO NEURAL NETWORK HYPER-PARAMETERS: PART 1 – LEARNING RATE, BATCH SIZE, MOMENTUM, AND WEIGHT DECAY," Apr. 2018.