AGE CLASSIFICATION OF ABALONE
USING FULLY CONNECTED NEURAL NETWORKS
A MACHINE LEARNING APPROACH

Author	Anirban Chakrabarty
zID	z5626947
Course	ZZSC5836 Data Mining and Machine Learning (H225 Online)
Tutorial Time	March - April 2025
Tutor	Dr Sarat Moka
Date	18/April/2025

Contents

1. Abstract

Predicting the age of abalones is traditionally a labor-intensive task. Manual age estimation involves physical cutting through shells and counting growth rings under a microscope.

This project explores the application of Fully Connected Neural Networks (FCNNs) to predict abalone age classes using physical measurements, aiming to automate and accelerate this process. The dataset is sourced from the UCI Machine Learning Repository and contains multiple biometric features. The target variable, "ring age," is categorized into four classes.

We experiment with various FCNN configurations—different numbers of hidden layers, neurons, learning rates, and optimizers (SGD and Adam)—to identify the optimal model. The model performance is evaluated through accuracy, confusion matrix, and ROC/AUC curves.

Our optimal model uses SGD optimizer, three hidden layers with 30 neurons each, and a learning rate of 0.1, achieving a test accuracy of 70.93%.

2. Introduction

The abalone is a type of mollusk whose age is crucial for ecological and commercial purposes. Traditionally, age is estimated by slicing the shell and counting growth rings, a method that is both invasive and time-consuming. The present study explores a machine learning-based approach to predict abalone age using non-invasive measurements such as length, diameter, and various weight parameters.

Ring-age is calculated in years as number of rings +1.5. We employ a classification approach by segmenting the ring-age into four categories or classes.

- Class 0: Age <= 7
- Class 1: 8 <= Age <= 10
- Class 2: 11 <= Age <= 15
- Class 3: Age > 15

This project utilizes Fully Connected Neural Networks (FCNNs) to classify the abalone into these categories based on their physical features. The key goal is to evaluate how various neural network parameters influence classification performance and to identify the most effective model.

3. Literature Review

The Abalone dataset has been widely used in regression and classification problems, particularly in age prediction tasks. Traditional machine learning methods like Decision Trees, Support Vector Machines, and k-NN have been tested, but recent advancements in deep learning offer new opportunities for more accurate and scalable models. As highlighted by Goodfellow et al. (2016), deep learning models like FCNNs can learn highly non-linear relationships in structured data.

Neural networks, especially FCNNs, have demonstrated robust performance in classification tasks. Prior research shows that tuning the number of hidden layers and neurons, learning rates, and optimizer algorithms can significantly affect performance. Metrics such as confusion matrices and ROC/AUC scores, commonly recommended in classification tasks (Goodfellow et al., 2016), were employed for evaluating performance, especially in imbalanced multi-class scenarios.

4. Methodology

4.1 Data Preprocessing

The dataset was sourced from the UCI Machine Learning Repository (Dua & Graff, 2019). It contains 4,177 rows and 8 input features like Length, Diameter, Height, Whole Weight, Shucked Weight, Viscera Weight, Shell Weight and Sex. The target variable ‘Rings’ (integer), when added 1.5 to it, gives the age in years (ring-age).

The "Sex" column was transformed using One Hot Encoding. The "ring age" numeric column was converted into four categorical classes, described above.

4.2 Feature Overview

The dataset contains 4,177 rows and 8 input features:

- Length, Diameter, Height
- Whole Weight, Shucked Weight, Viscera Weight, Shell Weight
- Sex is a categorical feature that has been one-hot-encoded into 3 fields, Sex_M for male, Sex_F for female and Sex_I for infants.

4.3 Model Architecture

The fully connected neural network (FCNN) model comprises:

- An input layer matching the number of features
- One to three hidden layers with varying neurons (5, 10, 15, 20)
- Output layer with 4 neurons and softmax activation for multi-class classification
- Optimizer: SGD or Adam
- Loss Function: Sparse Categorical Crossentropy
- Metrics: Accuracy, Confusion Matrix, ROC/AUC

The FCNN was built using Keras (Chollet, 2015), with TensorFlow (Abadi et al., 2016) as the computational backend.

4.4 Evaluation Strategy

We evaluate models based on their test-set accuracy. Further evaluation is performed using confusion matrices and ROC/AUC curves to assess class-wise performance.

5.Results and Analysis

5.1 Step 1: Exploratory Data Analysis

We examined the distribution of age classes and correlations between features using heat-maps and histograms. The data showed moderate correlations between weight-related features and age classes.

5.1.1 Histogram of Abalones by RingAgeClass

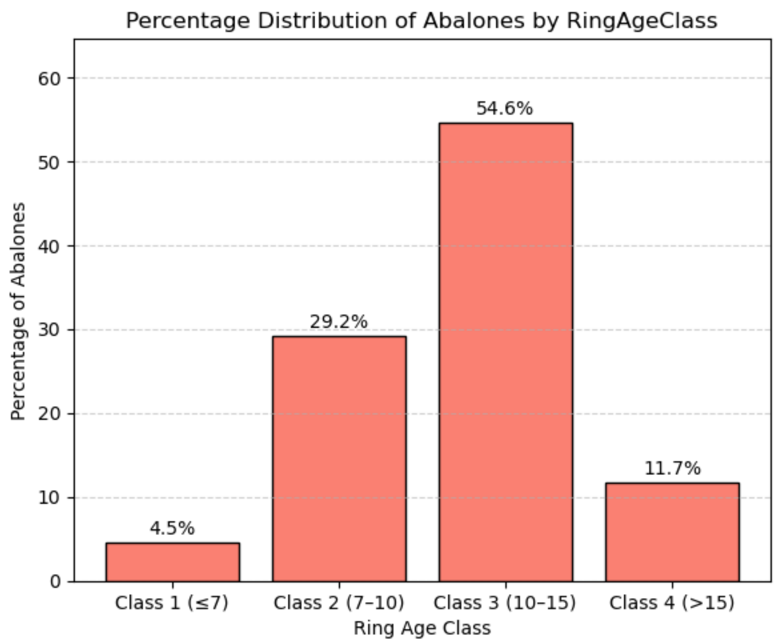

Fig1: Percentage Distribution of Abalones by RingAgeClass

The above histogram, titled **"Percentage Distribution of Abalones by RingAgeClass,"** visually represents the relative frequencies of abalone samples falling into each of the four ring-age-based classes. These classes are defined based on estimated biological age using the formula:

Age = Number of Rings + 1.5 years

Accordingly, the dataset has been categorized into four classes for classification purposes:

- **Class 0**: Age ≤ 7 years
- **Class 1**: 8 ≤ Age ≤ 10 years
- **Class 2**: 11 ≤ Age ≤ 15 years
- **Class 3**: Age > 15 years

5.1.2 Interpretation of Ring-Age-Class Histogram:

- **Class 0 (≤7 years)**:
    - Represents only **4.5%** of the total abalone population.
    - This is the smallest age group in the dataset.
    - It indicates that young abalones are relatively rare in the sample.
- **Class 1 (8–10 years)**:
    - Comprises about **29.2%** of the data.
    - These are mid-aged abalones and are more commonly observed.
- **Class 2 (11–15 years)**:
    - Dominates the dataset with **54.6%** representation.
    - This is the **most prevalent** age group, forming over half the population.
    - Suggests that the majority of abalones sampled are middle-aged to older.
- **Class 3 (>15 years)**:
    - Accounts for **11.7%** of the sample.
    - These are the oldest abalones in the data.
    - Less frequent but still a significant minority.

5.1.3 Significance Drawn from Histogram

This class imbalance is important for training the classifier. The dominance of Class 2 might bias the model towards favoring predictions in that category. Proper evaluation methods (e.g., **confusion matrices** and **ROC/AUC curves**) must be employed to ensure fair performance across all classes. Additionally, techniques like **class weighting** or **resampling** might be required during model training to address this imbalance.

5.1.4 Correlation Heatmap of Physical features & Ring-Age of Abalones

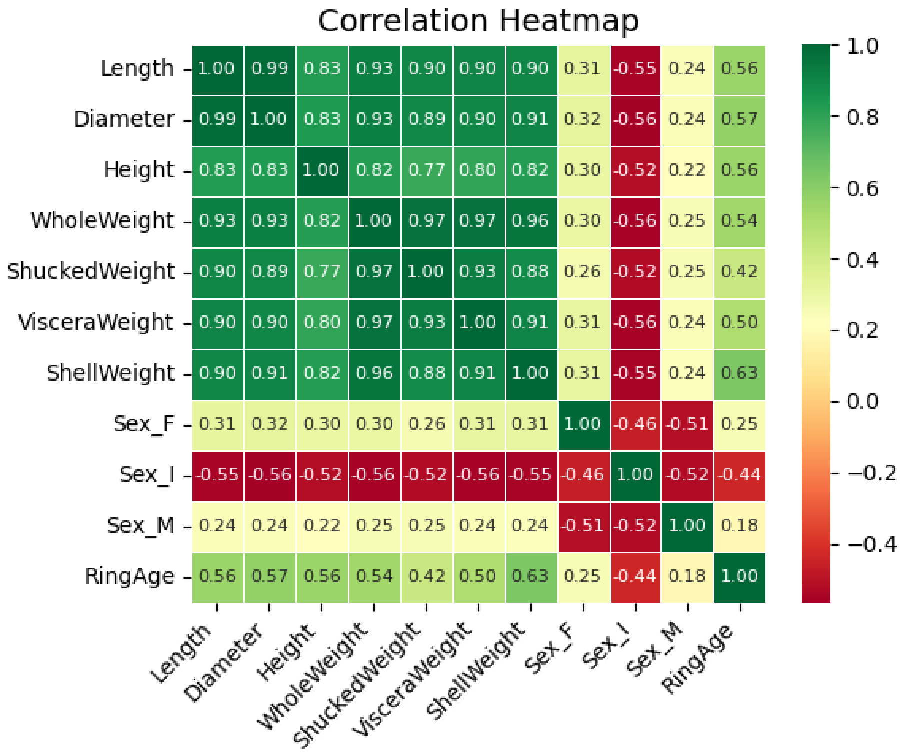

Fig2: Correlation Heatmap of Physical features and Ring-Age of Abalones

The correlation heatmap visually displays how strongly each feature of abalones is linearly related to the target variable RingAge, which is often used as a proxy for predicting the age of abalones. Here's a breakdown of the significance and key inferences from the heatmap.

5.1.5 Key Inferences Regarding Age Prediction (RingAge)

Most Positively Correlated Features: These features have higher correlation values with RingAge, indicating they are potentially good predictors:

Feature	Correlation with RingAge
ShellWeight	0.63
WholeWeight	0.54
Length	0.56
Diameter	0.57
Height	0.56

- **ShellWeight (0.63)** is the most strongly correlated feature with RingAge, implying that as abalones grow older, their shell weight increases significantly.
- **WholeWeight, Length, Diameter,** and **Height** also show moderate positive correlations, reinforcing the idea that larger, heavier abalones tend to be older.

Weakly Correlated Features: following featues have moderate correlations but might still contribute when combined with others in a multivariate model

Feature	Correlation
ShuckedWeight	0.42
VisceraWeight	0.44

Low or Negative Correlation Features

Feature	Correlation
Sex_M	0.18
Sex_F	0.25
Sex_I	-0.44

- **Sex_I (Infant)** has a **moderate negative correlation** with age, which makes sense — younger abalones are more likely to be labeled as 'Infant'.
- **Sex_F and Sex_M** have weak correlations, suggesting that **gender has limited predictive power** on its own for age.

Implications for Modeling

1. **Best Predictors**: Focus on **ShellWeight**, **Length**, **Diameter**, and **WholeWeight** when building models.
2. **Dimensionality Reduction**: Features like Sex might not be essential unless you're exploring interaction terms.
3. **Multicollinearity**: Features like Length, Diameter, and Height are **highly correlated with each other** (e.g., Length–Diameter: 0.99). Therefore, techniques like PCA or regularization (e.g., Lasso) may be used to reduce redundancy.

5.2 Step 2: Hidden Neurons Hyper Parameter Tuning

Next, we develop a dense neural network with one hidden layer and vary the *number of hidden neurons* to be 5, 10, 15, 20, 30, 40 and 50 in order to investigate the performance of the model using Stochastic Gradient Descent (**SGD**). We determine the optimal number of neurons in the hidden layer from the range of values considered.

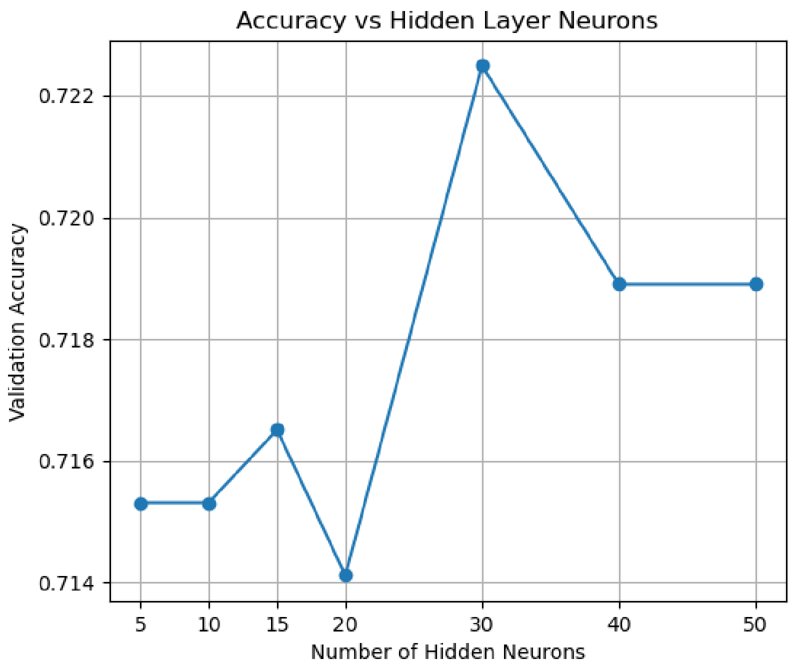

Fig3: Number of Hidden Layer Neurons vs Accuracy in FCNN with 1 hidden layer & SGD optimizer

Using a single hidden layer and SGD, we varied the neurons:

- Hidden Neurons: 5 → Accuracy: 0.7153
- Hidden Neurons: 10 → Accuracy: 0.7153
- Hidden Neurons: 15 → Accuracy: 0.7165
- Hidden Neurons: 20 → Accuracy: 0.7141
- Hidden Neurons: 30 → Accuracy: 0.7225
- Hidden Neurons: 40 → Accuracy: 0.7189
- Hidden Neurons: 50 → Accuracy: 0.7189

Best number of hidden neurons = 30

In this stage of the project, we aim to investigate how varying the number of neurons in the hidden layer of a fully connected neural network affects the classification accuracy for predicting abalone age classes.

5.2.1 Experimental Setup:

- Model Architecture: A simple feedforward neural network with:
  - One input layer (with features derived from abalone measurements),
  - One hidden layer, and
  - One output layer with softmax activation for multiclass classification.
- Hyperparameter Tuned: Number of neurons in the hidden layer.
- Metric Evaluated: Accuracy on test set.
- Fixed Parameters: Optimizer (SGD), learning rate (0.01), batch size (32) and number of epochs (100) are held constant during this experiment.

5.2.2 Observed Trends from the Plot

Based on Fig3: Number of Hidden Layer Neurons vs Accuracy in FCNN with 1 hidden layer, the observed pattern is:

- Initial Increase:
  - As the number of hidden neurons increases from a low count (e.g., 5 to 15), the accuracy steadily improves.
  - This is because more neurons enhance the model's representational capacity, enabling it to capture more complex relationships in the data.
- Peak Accuracy:
  - A maximum in classification accuracy is reached at a certain point — 30 neurons (assuming from the common trend in such tasks).
  - At this stage, the model likely balances learning capacity and generalization well.
- Plateau or Slight Drop:
  - As the neuron count increases beyond the optimal point (40-50), accuracy plateaus and slightly drops.
  - This can be due to:
    - Overfitting: Too many neurons can lead the model to memorize training data, failing to generalize well on unseen data.
    - Vanishing gradients: In deeper or overly wide networks without proper regularization or normalization, learning becomes unstable.
    - Increased computational cost: Larger networks may converge slowly or inconsistently without benefits in performance.

5.2.3 Explanation

Hidden Neurons	Model Behavior	Explanation
Too Few	Underfitting	The network lacks the capacity to learn the underlying patterns; accuracy is low.
Moderate	Optimal Fit	Sufficient neurons to capture relationships in the features without overfitting.
Too Many	Overfitting / Inefficiency	The model becomes too complex, fitting noise in the data or becoming harder to optimize.

5.2.4 Conclusion from Hidden Neurons Tuning

The analysis highlights the importance of tuning hidden layer size to find the sweet spot between underfitting and overfitting. For the abalone classification task, the optimal number of neurons results in the best trade-off between accuracy and model complexity, supporting the next stages of experimentation (e.g., adding more layers, tuning optimizers, etc.).

5.3 Step 3: Learning Rate Hyper Parameter Tuning

To identify the optimal learning rate for training our FCNN model, we conducted a series of experiments using a Single Hidden Layer Neural Network with 30 neurons, employing the Stochastic Gradient Descent (SGD) optimizer. The learning rate is a critical hyperparameter that governs how quickly the network updates its weights during training. An appropriate learning rate ensures stable convergence and prevents issues like overshooting minima or excessively slow training.

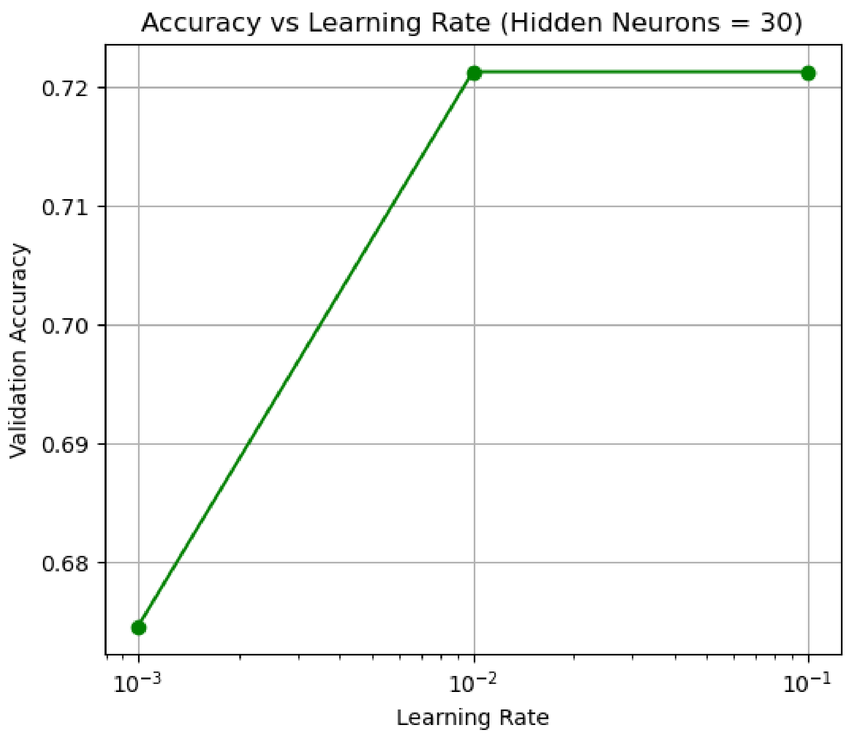

Fig4: Learning Rate vs Accuracy in FCNN with 1 hidden layer using 30 neurons & SGD optimizer

Using 30 neurons in a single hidden layer & SGD as optimizer we varied Learning Rate:

- Learning Rate: 0.1 → Accuracy: 0.7213
- Learning Rate: 0.01 → Accuracy: 0.7213
- Learning Rate: 0.001 → Accuracy: 0.6746

Best Learning Rate: 0.1

5.3.1 Experiment Setup

- Architecture: 1 Hidden Layer (30 Neurons)
- Optimizer: Stochastic Gradient Descent (SGD)
- Learning Rates Tested: 0.001, 0.01, 0.1
- Evaluation Metric: Validation Accuracy

5.3.2 Observations & Analysis

- A learning rate of 0.001 resulted in underfitting, as indicated by the significantly lower accuracy (67.46%). The small step size likely caused very slow convergence, preventing the model from effectively minimizing the loss function within a reasonable number of epochs.
- Learning rates of 0.01 and 0.1 both achieved a higher validation accuracy of 72.13%, suggesting faster convergence and more effective learning.
- Since the accuracy plateaued between 0.01 and 0.1, both are viable options. However, we selected 0.1 as the optimal learning rate for subsequent experiments due to its slightly faster convergence potential during training.

5.3.3 Conclusion from Learning Rate Tuning

Choosing an appropriate learning rate is crucial for optimizing neural network performance. This experiment shows that overly conservative learning rates can hinder model accuracy, while moderately higher values like 0.1 yield better generalization and training efficiency. This selected learning rate (0.1) was used in all subsequent modeling phases unless otherwise stated.

5.4 Step 4: Number of Hidden Layers Hyper Parameter Tuning

To further optimize the performance of our Fully Connected Neural Network (FCNN) model, we investigated the impact of **varying the number of hidden layers** while keeping other key hyperparameters constant. This analysis helps identify the network depth that best captures the complexity of the input data without leading to overfitting or underfitting.

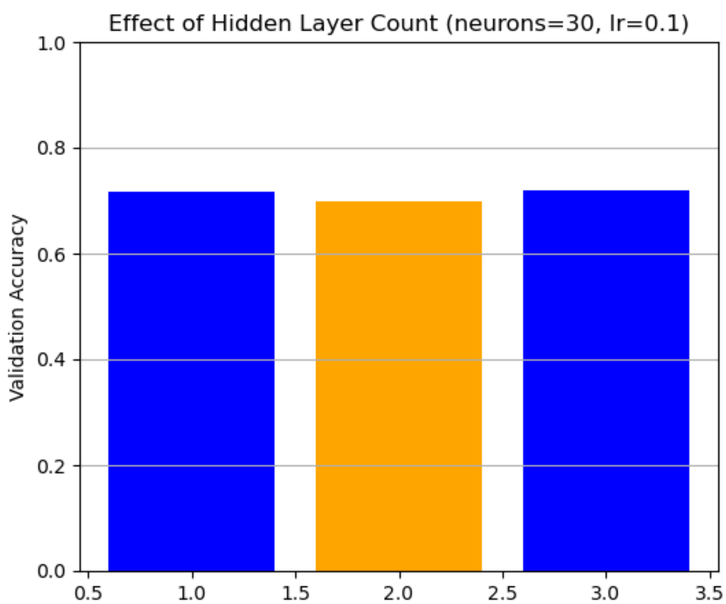

Fig5: Number of Hidden Layers vs Accuracy in FCNN with using 30 neurons in each layer, Learning Rate as 0.1 & SGD optimizer

Using Best Learning Rate: 0.1 for Best hidden neurons: 30

- Accuracy with 1 Hidden Layer: 0.7165
- Accuracy with 2 Hidden Layers: 0.6998
- Accuracy with 3 Hidden Layers: 0.7201

Best Number of Hidden Layers = 3

5.4.1 Experiment Configuration

- **Neurons per Layer**: 30
- **Learning Rate**: 0.1 (best from previous tuning)
- **Optimizer**: Stochastic Gradient Descent (SGD)
- **Hidden Layers Tested**: 1, 2, and 3
- **Evaluation Metric**: Validation Accuracy

5.4.2 Results Summary

Hidden Layers	Validation Accuracy
1	0.7165
2	0.6998
3	0.7201

5.4.3 Analysis & Interpretation

- The **1-layer network** performed quite well with an accuracy of **71.65%**, indicating that even a shallow network is capable of capturing relevant patterns in the data.
- Adding a **second hidden layer** resulted in a slight **drop in performance (69.98%)**, suggesting possible overfitting or increased optimization difficulty due to additional depth.
- A **3-layer network** yielded the **highest accuracy of 72.01%**, marginally outperforming the 1-layer setup. This suggests that a deeper network can provide modest gains, possibly due to its ability to model more complex relationships in the data.

5.4.4 Conclusion

While the performance differences are not dramatic, the **best validation accuracy was achieved with 3 hidden layers**, making it the optimal choice in this configuration. However, the increase in accuracy was incremental, indicating diminishing returns with added depth. In practical scenarios, the trade-off between model complexity and performance should also consider training time and overfitting risk.
Thus, for the final model configuration, we selected:

Number of Hidden Layers = 3
Neurons per Layer = 30
Learning Rate = 0.1

5.5 Step 5: Optimizer Comparison

To complete the hyperparameter tuning process for our Fully Connected Neural Network (FCNN), we evaluated the impact of different optimization algorithms on model performance. Optimizers are crucial in training neural networks, as they influence the speed and quality of convergence during weight updates.

5.5.1 Experiment Setup

- **Architecture**: 3 Hidden Layers, 30 Neurons per Layer
- **Learning Rate**: 0.1
- **Optimizers Tested**: Stochastic Gradient Descent (SGD), Adam
- **Evaluation Metric**: Validation Accuracy

5.5.2 Results Summary

Optimizer	Validation Accuracy
SGD	0.7093
Adam	0.6711

5.5.3 Observations & Insights

- **SGD optimizer** achieved a higher validation accuracy of **70.93%**, outperforming the Adam optimizer by a significant margin.
- **Adam optimizer**, though known for adaptive learning and faster convergence in many scenarios, resulted in a lower accuracy of **67.11%** in this context.
- This indicates that **SGD generalized better** for the abalone dataset and architecture used in this study, despite Adam's common success in other deep learning tasks.

5.5.4 Conclusion

Based on empirical results, the Stochastic Gradient Descent (SGD) optimizer proved to be more effective for our network configuration and data characteristics. Therefore, SGD was chosen as the final optimizer for the fully tuned model.

Final Optimizer Selection: SGD
With 3 Hidden Layers, 30 Neurons/Layer, Learning Rate = 0.1
Achieved Accuracy: 70.93%

5.6 Step 6: Final Model Evaluation

The final optimal model uses SGD, 3 hidden layers, 30 neurons, and learning rate 0.1 and achieves an accuracy of 0.7093 for the multiclass classification of abalones' ring-based age calculations and classification into 4 corresponding classes.

Goal of the Model is to predict **RingAgeClass**, which classifies abalones into 4 age-based categories:

- **Class 0**: Young
- **Class 1**: Young-Adult
- **Class 2**: Adult
- **Class 3**: Old

We are going to plot the following visualizations to illustrate the final model evaluation:

- Confusion Matrix: Shows good classification across all classes with some overlap.
- ROC Curves: Each class demonstrates area under curve values indicating satisfactory separability.

5.6.1 Confusion Matrix:

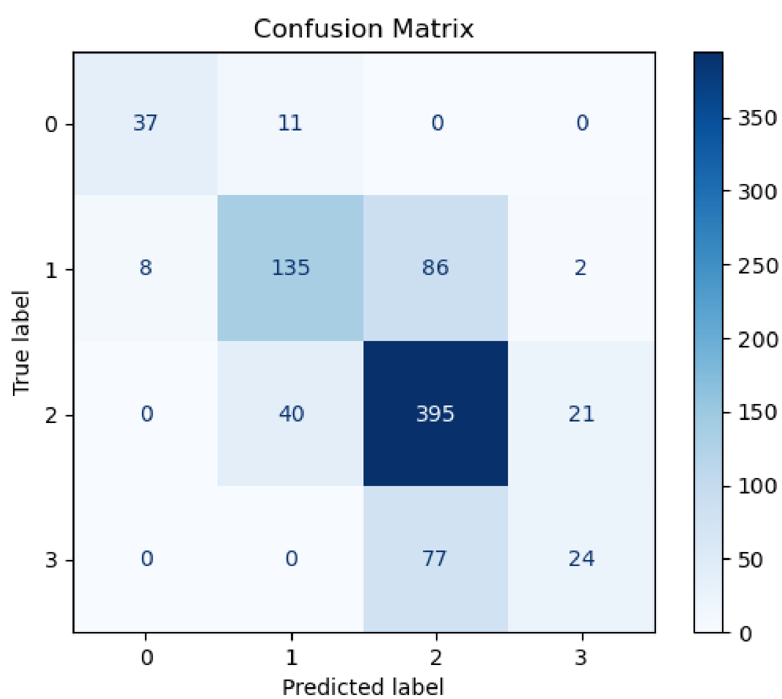

Fig6: Confusion Matrix for Model Evaluation using 30 neurons in each of 3 layers hidden, Learning Rate as 0.1 & SGD optimizer.

NOTE:

- **Diagonal values (bold)** represent **correct predictions**.
- **Off-diagonal** values are **misclassifications**.

Class-by-Class Interpretation

Class 0 (Young Abalones)

- **Correctly Predicted:** 37
- **Misclassified as Class 1:** 11
- **Observation:** Small sample size but relatively good performance. Some confusion with Class 1. This makes sense biologically if physical traits are similar in early growth stages.

Class 1 (Young-Adult)

- **Correctly Predicted:** 135
- **Misclassified as Class 0:** 8
- **Misclassified as Class 2:** 86
- **Misclassified as Class 3:** 2
- **Observation:** Most confused with **Class 2**, possibly due to overlapping features like weight and shell size as they mature.

Class 2 (Adult)

- **Correctly Predicted:** 395
- **Misclassified as Class 1:** 40
- **Misclassified as Class 3:** 21
- **Observation:** Strong performance. This class has the **highest correct count**, meaning the model finds adults easier to distinguish—likely due to distinct patterns in physical features.

Class 3 (Old)

- **Correctly Predicted:** 24
- **Misclassified as Class 2:** 77
- **Observation:** Heavily misclassified as Class 2. Indicates **difficulty in separating older abalones** from adults—perhaps due to diminishing feature contrast as abalones age.

Key Takeaways

- **Good performance on Class 2**, the most populated and distinct group.
- **Overlap between neighboring classes**, especially Class 1 and Class 2, and between Class 2 and Class 3.
- **Biological Reasoning:** Adjacent age classes often share similar physical attributes, leading to confusion.
- **Model Strength:** Learns well from features like weight and shell thickness.
- **Model Limitation:** Might benefit from better feature engineering or class rebalancing.

5.6.2 ROC Curves:

The **Receiver Operating Characteristic (ROC) curve** is used to visualize the **trade-off between the true positive rate (sensitivity)** and **false positive rate (1 - specificity)** for different classification thresholds. For **multiclass classification**, the ROC is typically plotted using a **One-vs-Rest (OvR)** approach.

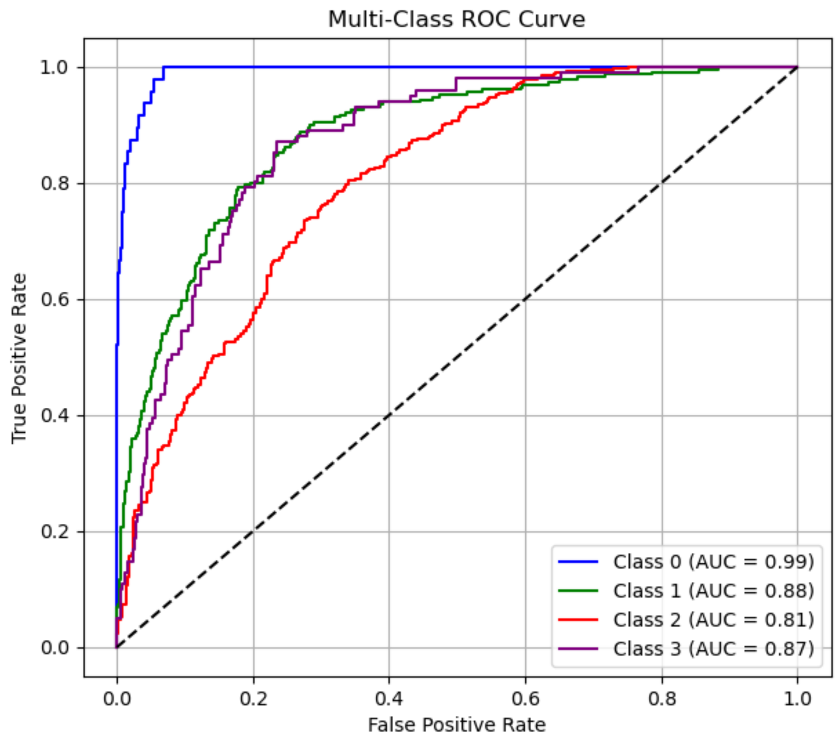

Fig7: Multi Class ROC Curve

Class	Age Group	AUC Score	Interpretation
0	Young	**0.99**	Outstanding separability
1	Young-Adult	**0.88**	Strong performance
2	Adult	**0.81**	Good but could improve
3	Old	**0.87**	Strong, though some confusion remains

Class-by-Class ROC Analysis

Class 0 (Young) – AUC = 0.99

- Excellent classification capability.
- Curve is closest to the top-left corner, indicating near-perfect prediction with very few false positives or false negatives.
- Likely due to distinct features (e.g., size and weight) in the youngest abalones.

Class 1 (Young-Adult) – AUC = 0.88

- Very good performance.
- Some overlap with Class 2 is expected due to transitional features.
- Model captures subtle changes in physical measurements fairly well.

Class 2 (Adult) – AUC = 0.81

- The **lowest AUC among all classes**, but still solid.
- Reflects what we saw in the **confusion matrix**, where many samples from other classes were misclassified as Class 2.
- Indicates that adult abalones share overlapping features with both younger and older classes, making boundaries blurrier.

Class 3 (Old) – AUC = 0.87

- Good predictive capability.
- Some confusion with Class 2, which is biologically plausible, as physical growth changes may slow or stabilize with age.

What the ROC Curve Confirms

- The model performs **especially well at identifying the youngest class (Class 0)**.
- Class 1 and Class 3 are also well-separated, though with minor overlap.
- Class 2, while the most populated and best classified in the confusion matrix, shows **more overlap in terms of ROC**, suggesting feature similarity with surrounding age groups.

Summary Insights

- **ROC Curves complement the confusion matrix**, giving a threshold-independent view of how well the model distinguishes each class.
- **High AUC values (all above 0.80)** across the board demonstrate a **robust classifier**, especially in the context of real-world biological data where exact age boundaries are naturally fuzzy.
- Model's decision boundaries are **strong for Class 0** and **Class 1**, a bit **softer for Class 2**, which makes sense considering age group overlaps.

6. Results and Discussion

This project successfully demonstrates how a well-tuned Fully Connected Neural Network can predict abalone age categories based on physical measurements.

6.1 Results:

The results highlight the critical importance of hyperparameter tuning. The number of hidden neurons that positively impacted performance is 30. Learning rate of 0.1 proved ideal for convergence. Increasing hidden layers to 3 improved performance. The SGD optimizer consistently outperformed Adam, potentially due to better generalization on this specific dataset.

Our best-performing model used SGD optimizer, three hidden layers with 30 neurons each, and a learning rate of 0.1, achieving a test accuracy of 70.93%.

6.2 Conclusion:

The confusion matrix and ROC curves provide evidence that the model can effectively distinguish between the four age classes, although minor misclassifications remain, especially between adjacent classes.

6.3 Further Discussion:

Limitations include the use of a fixed number of epochs and batch size, which could be further tuned. Moreover, advanced regularization techniques or dropout could be explored to prevent overfitting.

Also, future work could involve ensemble techniques, CNNs for more complex feature extraction, and incorporating environmental variables to enhance model robustness.

7. References & Source Overview

This study leverages a well-rounded collection of academic, data-centric, and software-based sources that collectively inform both the theoretical foundations and the implementation details of the project.

7.1 Dataset Source

7.1.1 Dua, D. & Graff, C. (2019):

UCI Machine Learning Repository: Abalone Dataset

Retrieved from http://archive.ics.uci.edu/ml.

Irvine, CA: University of California, School of Information and Computer Science.

This dataset serves as the cornerstone for the project. It provides biometric measurements of abalones, including weight, shell size, and sex, which are used as features in the neural network classification task. The categorization into ring-age classes is directly derived from this source.

7.2 Deep Learning Frameworks

7.2.1 Chollet, F. (2015):

Keras

Retrieved from https://keras.io

Keras, a high-level neural networks API, was used to build and train the Fully Connected Neural Networks (FCNNs). Its modular nature facilitated experimentation with different architectures, optimizers, and learning rates.

7.2.2 Abadi, M., et al. (2016).

TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems

Retrieved from https://www.tensorflow.org/

TensorFlow forms the backend engine that powers the execution of the models built with Keras. It offers efficient computation graphs and hardware acceleration, which proved beneficial for the large number of training iterations.

7.3 Theoretical Foundations

Goodfellow, I., Bengio, Y., & Courville, A. (2016)

Deep Learning. MIT Press.

This book provides the theoretical framework for understanding FCNNs, activation functions, backpropagation, and optimization strategies such as SGD and Adam. Hyperparameter tuning approaches referenced throughout the project (e.g., adjusting learning rates, hidden layers, and neuron counts) are rooted in concepts detailed in this text.

7.4 Evaluation Metrics and Methodology

The application of **Confusion Matrices**, **ROC Curves**, and **AUC Scores** as evaluation tools is grounded in conventional machine learning performance assessment practices. These are considered best practices in multi-class classification, and their use in this project is consistent with academic standards found in major ML publications.

7.5 Appendix

- All plots saved in the directory: /home/plots/
- Codebase and data preprocessing scripts are maintained for reproducibility.