Student Names: Hamed Loghmani, Farinam Hemmatizadeh, Atefeh Gilvari
Student ID: 110107453, 110106214, 110082454

**Mushroom Classification using Neural Network**
Neural Network and Deep Learning: Project 1

University of Windsor

---

## Contents



## 1 Group Participation Details

To get a better learning experience on the technical and implementation aspects, all of the group members did their own implementation for the whole project. Then, we discussed and did code review sessions and combined the best features of each individual's implementation into one single repository on GitHub (`https://github.com/Hamedloghmani/DeepLearningWinter23`). Finally, we separated the report into three different equal parts and did a group review as the final stage.

## 2 Introduction

The objective of this study is to evaluate the accuracy and compare the performance of various classification algorithms in predicting the edibility of mushrooms. Mushrooms hold great importance for humans, and it is crucial to have reliable methods to determine their safety for consumption and detect any potential diseases that may affect them. In the field of agriculture, it is vital to identify applications and techniques for predicting the presence of diseases and classifying them as they can have a significant impact on crop yield [3]. Mushrooms can be either edible or poisonous, and distinguishing between the two is not always straightforward. The Audubon Society Field Guide of North American Mushrooms notes that there is no single feature that can be used to differentiate between the two [1, 2]. In Florida, which produces 15% of the berries grown in the United States, strawberries are a major crop and are grown during the winter season. To help farmers better predict disease outbreaks in their strawberry plants, a forecasting system has been developed called the Strawberry Advisory System (SAS). This system provides predictions of disease incidence and recommends fungicide applications to farmers, thereby reducing production costs by avoiding unnecessary fungicide use while maintaining crop yields [5]. Plant diseases have a significant impact on global food production, with almost 10% of food production being lost due to plant diseases. The development of accurate methods for predicting and classifying

plant diseases can help minimize these losses. The SAS is a promising example of how technology can be used to aid farmers in disease management and help reduce the negative impact of plant diseases on food production. With the increasing demand for food and the growing population, such technological solutions are becoming increasingly important to ensure food security for the future [4]. In mushroom classification, artificial neural networks (ANNs) can be used to predict the edibility of mushrooms based on their attributes. The ANNs function by simulating the human brain and consist of multiple weighted connections between input and output layers. During training, the weights of the connections between the layers are adjusted to ensure that the model can accurately predict the class labels of input data objects. This enables ANNs to learn complex patterns in the data and make accurate predictions on unseen data, which is useful in distinguishing between edible and poisonous mushrooms [6]. As a result, it is essential to establish guidelines to determine the edibility of mushrooms. In this study, we aim to train existing classification algorithms using a dataset of raw measurements to classify mushrooms as either edible or poisonous by using neural network layers. The outcomes of this study will contribute towards developing reliable methods to predict the safety of mushrooms for human consumption and aid in detecting diseases affecting them, ultimately leading to better crop yield.

### 2.1 Related Work

Until now, most research in this area has utilized collections of images of plants or fungi as datasets for testing classification algorithms. Prior investigations have discovered that decision trees are popular because of their simple interpretability, while support vector machines (SVM) and artificial neural networks (ANN) are often regarded as the most effective. However, k-nearest neighbour (KNN) and naïve Bayes have not been considered the top classification algorithms for agriculture, despite being easy to train and, therefore, widely employed in various plant and fungi disease classification studies [7]. In a study described in [2], the accuracy of three classification algorithms in predicting plant diseases was compared using a dataset of plant leaf images. The results of the study revealed that the decision tree algorithm outperformed the ANN and naïve Bayes algorithms in terms of accuracy. The study [8] aimed to assess the performance of various classifiers for predicting disease presence in a mushroom dataset and classifying diseases in a soybean dataset. The classifiers evaluated were ANN, naïve Bayes, KNN, SVM, decision tree, and random forest. The study found that all classifiers, except naïve Bayes, accurately predicted disease presence in the mushroom dataset. In the soybean dataset, ANN and KNN were the best classifiers, and ANN was more appropriate for plant datasets.

## 3 Proposed Method

The present research implements a neural network architecture to classify data into one of two possible categories using a binary classification model. The model consists of three layers, including two hidden layers, each containing eight neurons and an output layer with a single neuron. This design allows the model to effectively learn complex patterns in the data and make accurate predictions based on the learned features. The hidden layers serve to extract relevant features from the input data, while the output layer provides the final classification decision. By utilizing this architecture, the model can achieve high accuracy and performance on the binary classification task at hand. Overall, the use of a three-layer neural network with specific neuron configurations represents a promising approach for binary classification tasks.

ReLU is chosen for the activation function in the hidden layers as it provides a simple and efficient way to introduce non-linearity into the model. It is computationally efficient and does not suffer from the vanishing gradient problem that can occur with other activation functions. By utilizing ReLU in the hidden layers, the model can effectively learn complex patterns in the data, leading to better accuracy and performance. Additionally, using ReLU can result in faster training times and better convergence properties compared to other activation functions, making it a popular choice for the classification task.

As we develop the model, we consider incorporating dropout layers at various points to prevent the model from becoming too tailored to the training data, which can lead to overfitting.

When dealing with a binary classification problem, the ideal activation function for the output layer is the sigmoid function. This function compresses the output values to a range between 0 and 1, which can be interpreted as the probability of belonging to one of two classes. By utilizing the sigmoid function in the final layer, we can effectively classify the data into these two distinct categories.

Binary cross-entropy is a commonly used loss function for binary classification problems, which is why it has been chosen in the context of classifying mushrooms in this work. This loss function is designed to measure the difference between the predicted probability distribution and the actual distribution of the target class. In the case of binary classification, this involves computing the difference between the predicted probability of belonging to one class and the actual binary label of the data (either 0 or 1). Binary cross-entropy is a popular choice because it is well-suited to problems where the classes are imbalanced, and it penalizes misclassification errors more strongly than correct predictions. By minimizing the binary cross-entropy loss during training, the model can learn to make better predictions and improve its accuracy on the classification task. Therefore, binary cross-entropy can be a suitable choice for classifying mushrooms since it involves predicting whether each sample belongs to one of two possible categories.

The present architecture incorporates various optimization algorithms to train the neural network model, including Adam, SGD, Adagrad, and RMSprop. These optimizers are commonly used in machine learning to update the weights of the network during training and improve its performance. By utilizing different optimizers, the model can explore different weight configurations and find the optimal solution for the given problem. In the evaluation section, the performance of each optimizer will be compared and analyzed to determine which one provides the best results for the given task.

# 4 Experiment

## 4.1 Dataset

The UCI machine learning repository hosts the mushroom dataset, which consists of 8,124 simulated examples of 23 species belonging to the Agaricus and Lepiota families [2]. The dataset includes 22 attributes that are categorical in nature, such as cap shape, odor, and habitat. Each species is labeled as either edible or poisonous, with any mushroom that cannot be categorized as edible automatically being classified as poisonous, irrespective of whether it is poisonous or not [8]. The mushroom dataset contains a set of categorical attributes that describe the characteristics of mushrooms. These attributes include features such as cap shape, cap color, bruises, odor, and population. Features of the dataset are summarized in table 1. We split the dataset into 70% for training and 30% for testing. This means that the majority of the data (70%) will be used to train the model, while the remaining 30% will be used to evaluate its performance. By doing this, we can assess how well the model will generalize to new data that it has not seen before.

## 4.2 Evaluation and Results

In the context of mushroom classification, different optimization algorithms have been utilized in the neural network architecture to improve the model's performance. The Adam optimizer stands out as a powerful option, utilizing adaptive learning rates and momentum to accelerate convergence and enhance accuracy. The SGD optimizer is a simple and efficient choice that updates the weights based on the gradient of the loss function with respect to the parameters. Adagrad, on the other hand, is an adaptive learning rate optimizer that can improve convergence on sparse datasets by adapting the learning rate for each parameter based on its historical gradient values. Finally, RMSprop is a stable and robust optimizer that combines the benefits of Adagrad and momentum, performing well on a wide range of problems.

Table 1: Dataset features

| Features | Value |
|---|---|
| **Dataset Characteristics** | Multivariate |
| **Attribute Characteristics** | Categorical |
| **Number of Instances** | 8,124 |
| **Number of Attributes** | 22 |
| **Missing Value** | Yes |

The performance of the binary classification model was evaluated using different values of the learning rate, specifically 0.01, 0.03, and 0.1. The learning rate is a hyperparameter that determines how much the model's parameters are updated in each iteration during training. The results show that a learning rate of 0.01 leads to the highest accuracy and lowest loss, indicating that the model can learn more effectively and make more accurate predictions with a lower learning rate. On the other hand, if the learning rate is too low, the model can take too long to converge or get stuck in a local minimum, or lead to overfitting, and if we choose higher learning rates, the model can overshoot the optimal parameters and fail to converge, therefore, becomes underfitted.

It is essential to assess the accuracy and loss of the model over the epochs to ensure effective training and optimal performance. To achieve this, two plots are created to visualize the accuracy and loss of the model on the training and validation sets. Figure 1 illustrates the loss of the neural network configuration as the number of epochs varies. Loss is a metric that quantifies the error in the model's predictions. It measures how well the model performs in predicting the expected outcome. A low-value loss means that the model's prediction is perfect. The plot shows that both the training and testing samples decrease concurrently as the number of epochs increases, and the loss function values decrease with each epoch, indicating that the model is improving as the training progresses and the prediction is perfect.

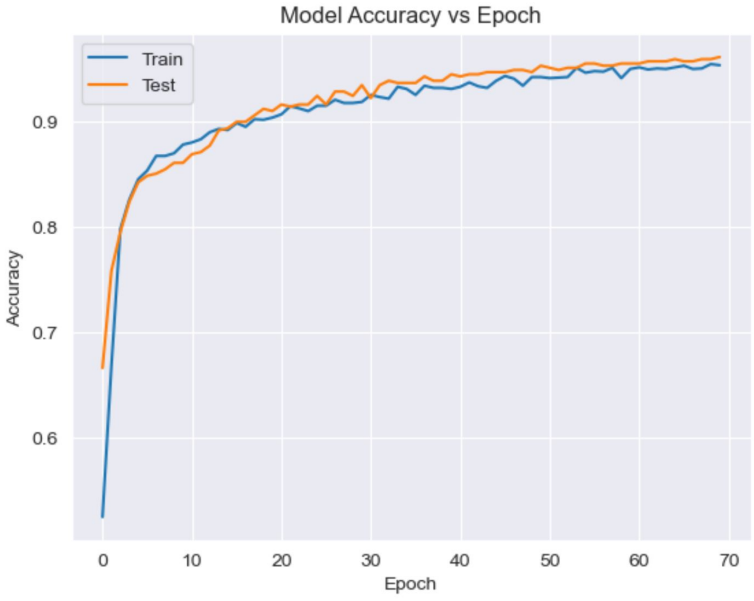


Figure 1: Model Accuracy Improvement Over Epochs

Figure 2 exhibits the accuracy of the neural network setup as the number of epochs varies. Accuracy

Table 2: Evaluation summary for different optimizers and best parameters

| optimizer | best lr | dropout | training epochs | mean train accuracy | mean test accuracy |
|---|---|---|---|---|---|
| Adam | 0.01 | 0.2 | 100 | 98.82% | 99.9% |
| SGD | 0.01 | 0.2 | 100 | 91.41% | 95.2% |
| RMSprop | 0.01 | 0.2 | 100 | 98.55% | 99.8% |
| Adagrad | 0.01 | 0.2 | 100 | 94.35% | 95.9% |

refers to the number of true positives and true negatives divided by the total number of predictions made by the model. It indicates how frequently the model makes accurate predictions. The plot shows that both the training and testing samples increase together with the number of epochs, indicating that the model's mean accuracy is 99.7% with Adam optimizer and 0.1 learning rate.

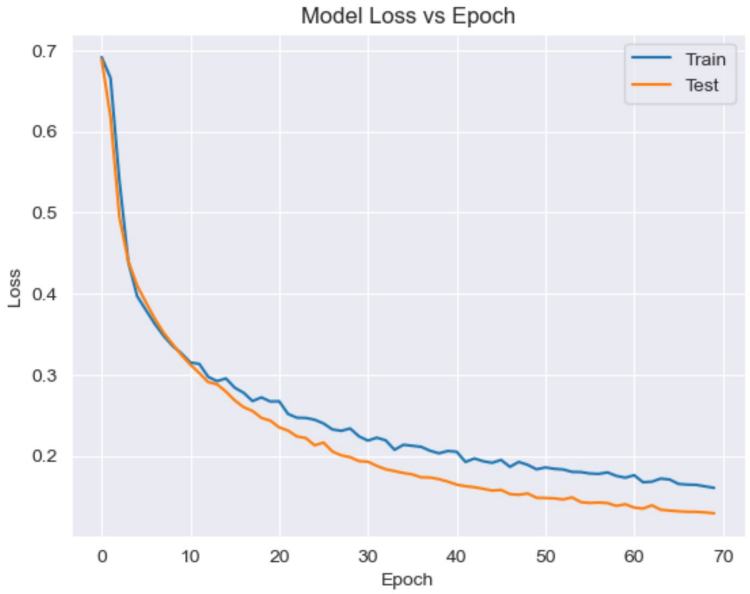


Figure 2: Model Loss Reduction Over Epochs

However, incorporating dropout layers can aid in enhancing the model's ability to generalize to novel and unseen data. As a result, by adding a dropout layer to the model, its accuracy decreased to 98.8%. This decrease in accuracy indicates that the model has not been overfitted and can generalize well to new and unseen data.

The model's weight update process is influenced by the batch size, whereas the number of epochs determines how many times the model is trained in total. A batch size of 100 is set, indicating that the model is trained on 100 samples during each iteration. Finally, although we have tested our neural models with different number of epochs, with the learning rate of 0.01 all of them converged with almost 100 epochs (mostly less). Hence, in order to avoid computational overhead we kept 100 epochs as the best epoch number. The summary of our evaluation results is available in Table 2.

## References


[1] A Knopf. "The Audubon society field guide to North American mushrooms". In: *GH Lincoff (Pres.), New York* (1981).

[2] JC Schlimmer. *UCI repository of machine learning databases*. 1981.

[3] LV Madden, G Hughes, and ME Irwin. "Coupling disease-progress-curve and time-of-infection functions for predicting yield loss of crops". In: *Phytopathology* 90.8 (2000), pp. 788–800.

[4] Richard N Strange and Peter R Scott. "Plant disease: a threat to global food security". In: *Annu. Rev. Phytopathol.* 43 (2005), pp. 83–116.

[5] W Pavan, CW Fraisse, and NA Peres. "Development of a web-based disease forecasting system for strawberries". In: *Computers and electronics in agriculture* 75.1 (2011), pp. 169–175.

[6] Shomona Gracia Jacob and R Geetha Ramani. "Data mining in clinical data sets: a review". In: *training* 4.6 (2012).

[7] David Camilo Corrales, Juan Carlos Corrales, and Apolinar Figueroa-Casas. "Towards detecting crop diseases and pest by supervised learning". In: *Ingeniería y Universidad* 19.1 (2015), pp. 207–228.

[8] Maria Morgan, Carla Blank, and Raed Seetan. "Plant disease prediction using classification algorithms". In: *IAES International Journal of Artificial Intelligence* 10.1 (2021), p. 257.