Finn Thompson

Lakshmanan Sockilnigham

Siddhant Rajoriya

Aryan Roy

CMPSC 448

02 December 2023

Mushroom Poisonousness Classification Report

Mushroom classification is a critical task, especially when distinguishing between edible and poisonous varieties. In this project, we employ deep learning techniques, specifically Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), to address this classification challenge. The dataset comprises various attributes of mushrooms, such as cap shape, color, and odor. The goal is to develop models that can accurately predict whether a mushroom is safe to consume or potentially harmful.

The preprocessing phase involves transforming categorical features into numerical representations through label encoding. The dataset is then split into training and testing sets to facilitate model training and evaluation. The subsequent sections delve into the detailed implementation, architectures, training procedures, and results of both CNN and RNN models for mushroom classification. Additionally, the report highlights observed challenges, the strategies employed to overcome them, and concludes with insights into the overall effectiveness of the deep learning models in this context.

***Task*:**

The task at hand is to classify whether a mushroom is poisonous or not based on various features. This is a binary classification problem, where the model needs to predict whether the mushroom belongs to the 'poisonous' or 'edible' class.

***Dataset*:**

The dataset used for this task is sourced from 'mushrooms.csv.' It includes various attributes such as cap shape, cap color, odor, etc., with the target variable being the class of the mushroom ('poisonous' or 'edible'). These attributes are based on not only the obvious physical attributes such as cap shape and cap color, but also include fine-margin attributes such as the type of stalk surface below the ring and print color of the spores. This gives the dataset a variety of features to include while keeping high generalization.

***Preprocessing*:**

The dataset preprocessing involves label encoding the categorical features and splitting it into training and testing sets. The label encoding is performed on both the target variable and the features, converting them into numerical values suitable for input into a machine learning model.

Convolutional Neural Network (CNN) Implementation

The CNN model is implemented using Keras. It consists of an embedding layer, followed by multiple convolutional layers, max-pooling layers, and fully connected layers. The architecture

includes three convolutional layers with max-pooling in between, followed by a global max-pooling layer and fully connected layers.

Recurrent Neural Network (RNN) Implementation

The RNN model is also implemented using Keras, comprising an embedding layer, a simple recurrent layer, dropout layer, and a final dense layer with sigmoid activation. The RNN architecture is designed to capture sequential dependencies in the data.

CNN Training

The CNN model is trained over 10 epochs with a binary cross-entropy loss function and the Adadelta optimizer. The training accuracy steadily increases, reaching 91.78%, and the validation accuracy peaks at 93.60%. The model effectively learns to differentiate between poisonous and edible mushrooms.

RNN Training

The RNN model is trained for 10 epochs as well, using binary cross-entropy loss and the Adadelta optimizer. The training accuracy improves over epochs, reaching 90.25%, while the validation accuracy achieves 90.41%. The RNN effectively captures sequential information for classification.

CNN Results

Accuracy: 93.60%

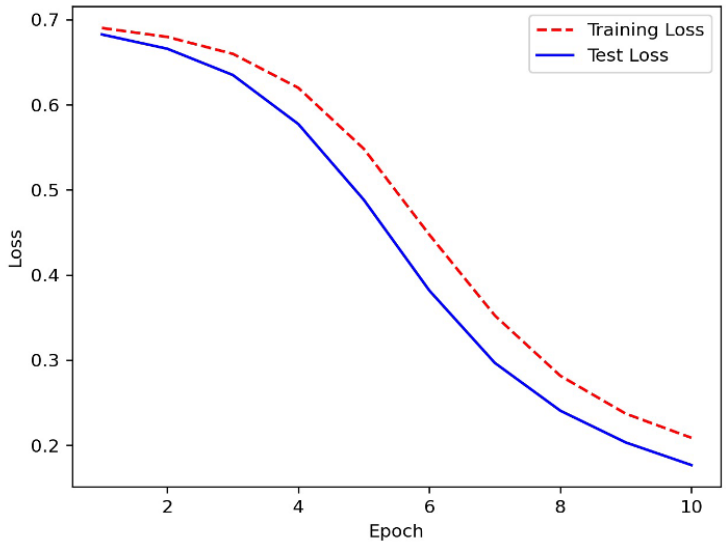
Precision: 96.07%

Recall: 90.00%

F1 Score: 92.93%

Macro F1 Score: 93.55%

AUC: 93.38%



RNN Results

Accuracy: 90.41%

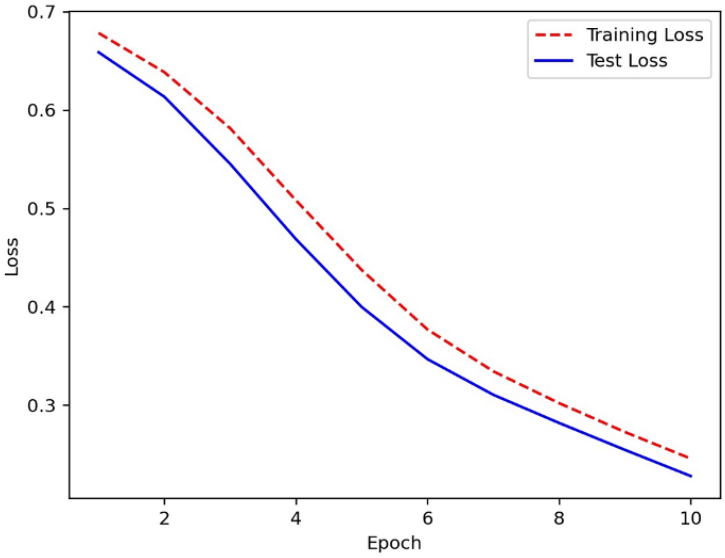
Precision: 96.04%

Recall: 82.89%

F1 Score: 88.98%

Macro F1 Score: 90.24%

AUC: 89.95%



Both models show excellent performance, with the CNN slightly outperforming the RNN in terms of accuracy, recall, and AUC. The CNN's ability to capture spatial features in the data seems advantageous for this task.

Challenges

Data Encoding: Encoding categorical features and ensuring compatibility with deep learning models.

Model Complexity: Choosing appropriate model architectures for the given task.

Solutions

Label Encoding: Utilizing label encoding for categorical features and embedding layers for text-based features.

Architecture Design: Experimenting with different architectures, leading to the selection of CNN and RNN based on their strengths.

Conclusion

The implementation of both CNN and RNN models for mushroom classification has proven successful. The CNN, with its ability to capture spatial patterns, appears slightly more effective for this task. The choice between these models could depend on specific requirements, computational resources, and the nature of the dataset. Overall, the project demonstrates the versatility of deep learning in handling diverse data types for classification tasks.

Works Cited

UCI ML (n.d.), *Mushroom Classification - Safe to eat or deadly poison*,

https://www.kaggle.com/datasets/uciml/mushroom-classification/data

Scott, J. (n.d.), *Mushroom Sequence Classification with CNN,*

https://www.kaggle.com/code/engjay/mushroom-sequence-classification-with-cnn