Assignment: ANN Classification Tasks on the Iris Dataset

Your Name

October 24, 2024

1 Question 1

Task: Implement a simple artificial neural network (ANN) to classify the Iris dataset into its three species: Setosa, Versicolour, and Virginica. Evaluate the model's performance using accuracy, precision, recall, and F1-score.

Requirements:

- Use TensorFlow/Keras or PyTorch to build the ANN.
- Split the dataset into training and testing sets.
- Use at least one hidden layer in the neural network.
- Apply the softmax function in the output layer for multi-class classification.
- After training the model, compute accuracy, precision, recall, and F1-score.

Hints:

- Use `classification_report` from `sklearn.metrics` to compute precision, recall, and F1-score.
- Use a small number of neurons in the hidden layer since the dataset is small.
- The output layer should have 3 neurons for the 3 classes.

2 Question 2

Task: Train a neural network classifier on the Iris dataset to predict the species of a flower based on its features (sepal length, sepal width, petal length, petal width). Evaluate the model using a confusion matrix and calculate accuracy, precision, recall, and F1-score.

Requirements:

- Build a neural network using TensorFlow/Keras or PyTorch with at least one hidden layer.
- Use the ReLU activation function for hidden layers and softmax for the output layer.
- After training, create a confusion matrix to visualize the performance.
- Compute accuracy, precision, recall, and F1-score for each class (Setosa, Versicolour, Virginica).

Hints:

- Use `ConfusionMatrixDisplay` from `sklearn.metrics` for visualization.
- Use the Adam optimizer to train the model.
- Apply cross-entropy loss function for multi-class classification.

3 Question 3

Task: Perform multi-class classification on the Iris dataset using a neural network. Implement cross-validation to evaluate the model's performance and report accuracy, precision, recall, and F1-score.

Requirements:

- Use TensorFlow/Keras or PyTorch to build the ANN.
- Apply k-fold cross-validation (e.g., 5-fold) to evaluate the model's performance.
- Use at least one hidden layer with an appropriate activation function.
- After performing cross-validation, compute the average accuracy, precision, recall, and F1-score across all folds.

Hints:

- Use `KFold` or `StratifiedKFold` from `sklearn.model_selection` for cross-validation.
- Use `classification_report` from `sklearn.metrics` to compute metrics on each fold.
- Aggregate the results of each fold to calculate average performance.

4 Logistic Regression from Scratch

Consider the Iris dataset with sepal length and sepal width as the attributes, and Iris-Setosa as class c_1, and the Virginica as class c_2. There are $n_1 = 50$ points in c_1 and $n_2 = 100$ points in c_2.

Task: Train the logistic regression model and find the separating decision boundary and plot it. Do it from scratch without using a library.