**ARTIFICIAL NEURAL NETWORKS**

**Classification Using Artificial Neural Networks with Hyperparameter Tuning on Alphabets Data**

**1. Introduction**

This report presents the development and evaluation of an Artificial Neural Network (ANN) model for the classification of alphabet characters using the "Alphabets\_data.csv" dataset. The objective of this assignment is to apply data preprocessing techniques, construct a neural network, perform hyperparameter tuning, and evaluate the model using appropriate classification metrics. The emphasis is on understanding the structure and performance of ANNs and exploring the effect of hyperparameters on model performance.

**2. Data Exploration and Preprocessing**

The dataset "Alphabets\_data.csv" was loaded and examined to understand its structure. It contains 20,000 samples, each with 128 numerical features and a target label representing one of 26 English alphabet classes (A to Z).

**Key preprocessing steps performed:**

* **Handling Missing Values:** The dataset was checked for null values and cleaned if necessary.
* **Label Encoding:** The target labels (A-Z) were encoded using LabelEncoder followed by one-hot encoding to facilitate multi-class classification.
* **Normalization:** The feature values were scaled to a range of [0, 1] using MinMaxScaler to standardize the input for the ANN.

**3. Model Implementation**

An ANN model was constructed using TensorFlow and Keras. The model includes the following:

* Input layer matching the number of features (128).
* One or more hidden layers with ReLU activation.
* Output layer with 26 neurons and softmax activation for multi-class classification.

The dataset was split into training and testing sets (80/20 split). The model was compiled using categorical crossentropy loss and accuracy as the evaluation metric. Training was conducted over 20 epochs using an appropriate batch size.

**4. Hyperparameter Tuning**

Hyperparameter tuning was performed using RandomizedSearchCV in conjunction with KerasClassifier. The following hyperparameters were explored:

* Number of hidden layers (1–3)
* Number of neurons in each layer (64, 128, 256)
* Activation function (relu, tanh)
* Optimizers (adam, sgd)
* Learning rate (0.001, 0.01)
* Batch size (32, 64, 128)

The search identified a set of optimal hyperparameters which significantly improved model performance. The tuned model was retrained and evaluated using the test dataset.

**5. Evaluation**

The model was evaluated using accuracy, precision, recall, and F1-score. The final test accuracy achieved was approximately 93.9%. The classification report showed strong performance across all 26 classes, with precision, recall, and F1-scores above 0.90 for most classes.

**Summary of Results:**

* Default model accuracy (before tuning): Lower than final result.
* Tuned model accuracy: 93.9%
* Clear improvement observed after hyperparameter optimization.
* No signs of severe overfitting based on training/validation loss trends.

**6. Conclusion**

This assignment demonstrates the successful application of Artificial Neural Networks for a multi-class classification task. Through careful preprocessing and systematic hyperparameter tuning, model performance was significantly enhanced. The results confirm the importance of tuning network architecture and training parameters in neural network design. The project provides a comprehensive understanding of the ANN workflow from data preprocessing to final evaluation.

.