## Project Documentation: Dimensionality Reduction Techniques on CIFAR-10

### Project Title: Optimizing Model Performance with Dimensionality Reduction

### 1. Business Objective

\*\*Enhance Model Efficiency and Performance through Dimensionality Reduction\*\*

The objective of this project is to optimize classification model performance by applying dimensionality reduction techniques, specifically Principal Component Analysis (PCA), to the CIFAR-10 dataset. This aims to improve computational efficiency and model generalization by reducing data dimensions while retaining essential characteristics necessary for accurate classification.

#### Business Constraints

1. \*\*Data Integrity\*\*: Ensure that the dimensionality reduction approach does not significantly distort or lose vital data characteristics necessary for accurate model classification.

2. \*\*Computational Resources\*\*: Account for computational resource limitations, especially when training models on both high-dimensional and reduced datasets.

3. \*\*Model Complexity\*\*: Find a balance between reducing data dimensions and retaining enough data complexity to capture critical patterns for accurate classification.

#### Success Criteria

1. \*\*Model Performance\*\*: Achieve comparable or improved classification accuracy on the reduced dataset relative to the original dataset, targeting at least 85% accuracy.

2. \*\*Computational Efficiency\*\*: Demonstrate at least a 30% reduction in training time with the reduced dataset, reflecting efficiency gains.

3. \*\*Trade-off Analysis\*\*: Provide a detailed report discussing trade-offs between dimensionality reduction and model performance, covering accuracy, training time, and complexity impacts.

### 2. Data Collection

The CIFAR-10 dataset, consisting of 60,000 32x32 color images across 10 classes, is used. This dataset serves as a benchmark for image classification tasks, with images split into 50,000 for training and 10,000 for testing. The original high-dimensional dataset (32x32x3 for each image) will be compared to a PCA-reduced version to analyze trade-offs in model performance and efficiency.

### 3. Project Workflow

#### Step 1: Import Required Libraries

The project begins by importing essential libraries, including:

- \*\*TensorFlow\*\*: For constructing and training the deep learning model.

- \*\*PCA\*\* and \*\*StandardScaler\*\* from scikit-learn: For dimensionality reduction and data standardization.

#### Step 2: Load the CIFAR-10 Dataset

The CIFAR-10 dataset is loaded using TensorFlow’s dataset API. Each image is 32x32 pixels with three color channels, giving a high-dimensional input representation.

#### Step 3: Data Preprocessing - Flatten and Standardize

1. \*\*Flattening\*\*: Each image (32x32x3) is reshaped into a 1D vector for PCA application.

2. \*\*Standardization\*\*: The flattened data is standardized to zero mean and unit variance to prepare it for PCA, ensuring that all features contribute equally to the variance.

#### Step 4: Apply PCA for Dimensionality Reduction

PCA is applied to reduce the dataset’s dimensionality:

- \*\*Number of Components\*\*: PCA is set to retain 150 components, capturing a significant portion of variance while reducing dimensionality.

- \*\*Variance Check\*\*: The cumulative variance explained by the components is examined to ensure that essential information is retained.

#### Step 5: Build and Train Model on Original Dataset

A Sequential neural network is initialized to classify images in the original high-dimensional dataset:

1. \*\*Model Architecture\*\*:

- A fully connected \*\*Dense layer\*\* with ReLU activation is used for hidden layers.

- \*\*Softmax activation\*\* is used in the output layer for multi-class classification.

2. \*\*Model Compilation\*\*:

- \*\*Optimizer\*\*: Adam optimizer is chosen for efficient gradient updates.

- \*\*Loss Function\*\*: Sparse categorical cross-entropy is used for multi-class classification.

- \*\*Metrics\*\*: Accuracy is tracked to monitor performance.

3. \*\*Training\*\*:

- The model is trained on the original dataset with 10 epochs, a batch size of 64, and 20% of data used for validation.

- Model performance is evaluated on the test dataset, with accuracy and loss recorded.

#### Step 6: Build and Train Model on PCA-Reduced Dataset

The same Sequential model is then trained on the PCA-reduced dataset:

1. The model architecture, compilation, and training parameters mirror those used for the original data.

2. Model training and evaluation are repeated on the reduced dataset, with test accuracy and loss recorded.

#### Step 7: Trade-Off Analysis

To assess the impact of dimensionality reduction, a trade-off analysis is conducted:

1. \*\*Accuracy Difference\*\*: Calculate the difference in accuracy between models trained on original and PCA-reduced data.

Formula:

\[

\text{accuracy\\_difference} = (\text{test\\_accuracy\\_orig} - \text{test\\_accuracy\\_pca}) \times 100

\]

2. \*\*Training Time Reduction\*\*: Measure and compare the training time for both datasets to evaluate computational efficiency improvements.

3. \*\*Reporting\*\*: A comprehensive written report discusses:

- Accuracy and computational efficiency comparisons.

- Observed trade-offs between dimensionality reduction and model accuracy.

- Practical considerations for using dimensionality reduction in classification tasks.

### 4. Summary and Insights

This project demonstrated the application of PCA on CIFAR-10 to reduce dimensionality and improve model efficiency. By reducing training time and resource usage by at least 30% while maintaining target accuracy, the project successfully showcased the value of dimensionality reduction for deep learning applications. The final report offers insights into balancing model complexity and performance, assisting stakeholders in making data-driven decisions for efficient and scalable AI solutions.