**Machine Learning for Robotics: Fashion MNIST Project Report**

**Project Overview**

This project, conducted as part of the *Machine Learning for Robotics* course (Section CS-Z), focuses on preprocessing and dimensionality reduction of the Fashion MNIST dataset using Python. The dataset, comprising 70,000 grayscale images of 10 clothing categories, is processed to prepare it for machine learning tasks, with a specific emphasis on applying Principal Component Analysis (PCA) for dimensionality reduction. The project demonstrates data loading, normalization, splitting, flattening, and PCA transformation, ensuring the data is optimized for subsequent modeling.

Group Members:

* Ali Haider (21I-1522)
* Arsal Mairaj (21I-1520)
* Shayan (22I-2392)

Submission Date: April 28, 2025

**Dataset Description**

The Fashion MNIST dataset is a collection of 70,000 grayscale images, each of size 28x28 pixels, representing 10 clothing categories (e.g., t-shirt, trouser, dress). It is split into:

* **Training Set**: 60,000 images
* **Test Set**: 10,000 images

Each image is associated with a label (0–9) corresponding to a specific clothing category. The dataset is widely used as a benchmark for image classification tasks, offering a more complex alternative to the classic MNIST digit dataset.

**Objectives:**

1. Load and verify the Fashion MNIST dataset.
2. Normalize pixel values to the range [0, 1].
3. Split the dataset into training (70%), validation (15%), and test (15%) sets.
4. Flatten images into 1D vectors for non-neural network models.
5. Apply PCA to reduce dimensionality while retaining 95% and 98% of the explained variance.
6. Visualize the cumulative explained variance using an elbow plot.
7. Transform validation and test sets using the fitted PCA model and save the results.

**Methodology**

**1. Environment Setup**

The project uses Python with the following libraries:

* **Tensor Flow**: For loading the Fashion MNIST dataset.
* **NumPy**: For numerical operations and array manipulation.
* **Scikit-learn**: For data splitting (train\_test\_split) and PCA.
* **Matplotlib**: For plotting the PCA elbow plot.
* **Seaborn**, **UMAP**, **TSNE**: Imported but not used in the provided code (potentially for future visualizations).
* **Jupyter Notebook**: As the development environment.

**2. Data Loading**

The Fashion MNIST dataset is loaded using tf.keras.datasets.fashion\_mnist.load\_data(). The shapes are verified:

* Training data: (60,000, 28, 28), Labels: (60,000,)
* Test data: (10,000, 28, 28), Labels: (10,000,)

**3. Data Normalization**

Pixel values, originally in the range [0, 255], are normalized to [0, 1] by:

* Converting the data type to float32.
* Dividing by 255.0.
* Verification confirms min=0.0 and max=1.0 for both training and test sets.

**4. Data Splitting**

The dataset is split into:

* **Training Set (70%)**: 42,000 images
* **Validation Set (15%)**: 9,000 images
* **Test Set (15%)**: 9,000 imagesThe process involves:
* Splitting the full training set (60,000 images) into 70% training and 30% temporary using train\_test\_split with test\_size=0.3 and random\_state=42.
* Splitting the temporary set (18,000 images) equally into validation and test sets (test\_size=0.5).
* The original test set is replaced with the new test set (9,000 images).
* Shapes are verified to ensure correct splitting.

**5. Data Flattening**

To prepare for non-neural network models (e.g., SVM, logistic regression), the 28x28 images are flattened into 1D vectors of 784 features (28 \* 28):

* Training: (42,000, 784)
* Validation: (9,000, 784)
* Test: (9,000, 784)
* The first sample’s shape is checked before (28, 28) and after (784,) flattening to confirm correctness.

**6. Dimensionality Reduction with PCA**

PCA is applied to the flattened training data (42,000 samples, 784 features) to reduce dimensionality while retaining significant variance.

**Steps:**

1. **Fit PCA**:
   * A PCA model is fitted on the flattened training data to compute principal components.
2. **Cumulative Explained Variance**:
   * The cumulative sum of explained variance ratios is calculated.
   * Components retaining ≥95% and ≥98% variance are identified:
     + 95% variance: 187 components
     + 98% variance: 348 components
3. **Elbow Plot**:
   * A plot visualizes the cumulative explained variance versus the number of components.
   * Horizontal lines mark 95% and 98% variance thresholds.
   * Vertical lines indicate the corresponding number of components.
4. **Transform Data**:
   * Two PCA models are created:
     + pca\_95: Retains 187 components (95% variance).
     + pca\_98: Retains 348 components (98% variance).
   * Training, validation, and test sets are transformed:
     + PCA 95% shapes: Train (42,000, 187), Val (9,000, 187), Test (9,000, 187)
     + PCA 98% shapes: Train (42,000, 348), Val (9,000, 348), Test (9,000, 348)
5. **Save Results**:
   * PCA-transformed datasets are saved as .npy files for future use.

**Results**

* **Data Preparation:**
  + The dataset was successfully loaded, normalized, split, and flattened.
  + Normalization ensured pixel values are in [0, 1], suitable for machine learning.
  + Splitting produced balanced training, validation, and test sets.
  + Flattening converted images into 784-dimensional vectors, enabling compatibility with non-neural models.
* **PCA:**
  + PCA reduced the dimensionality from 784 features to:
    - 187 features (95% variance)
    - 348 features (98% variance)
  + The elbow plot effectively illustrated the trade-off between the number of components and explained variance.
  + Transformed datasets were correctly shaped and saved, ensuring reproducibility.

**Discussion**

* **Normalization**: Essential for stabilizing model training and improving convergence.
* **Splitting**: The 70-15-15 split ensures sufficient data for training while reserving validation and test sets for evaluation.
* **Flattening**: Necessary for traditional machine learning models but not for convolutional neural networks, which leverage spatial structure.
* **PCA**:
  + Reducing to 187 components (95% variance) significantly lowers computational cost while retaining most information.
  + Using 348 components (98% variance) preserves more details but increases complexity.
  + The elbow plot aids in visualizing the point of diminishing returns for additional components.
* **Limitations**:
  + The code includes unused libraries (TSNE, UMAP, Seaborn), suggesting potential plans for further visualization not implemented here.
  + No classification models were trained, limiting insights into PCA’s impact on performance.
* **Future Work**:
  + Apply classification models (e.g., SVM, logistic regression) on PCA-transformed data to evaluate accuracy.
  + Explore TSNE or UMAP for visualizing high-dimensional data in 2D/3D.
  + Compare PCA with other dimensionality reduction techniques (e.g., auto encoders).

**Conclusion:**

The project successfully preprocesses the Fashion MNIST dataset and applies PCA for dimensionality reduction. The dataset is normalized, split, and flattened, with PCA reducing features to 187 (95% variance) and 348 (98% variance). The elbow plot provides a clear visualization of variance retention. The transformed data is saved for future use, laying a strong foundation for subsequent machine learning tasks. This work demonstrates proficiency in data preprocessing and dimensionality reduction, critical for efficient model training in robotics and computer vision applications.