Handwritten Letter Recognition using EMNIST Dataset

Introduction

Handwritten letter recognition is a fundamental challenge in machine learning, with applications in digitizing documents, postal services, and assistive technologies. The EMNIST dataset extends the classic MNIST dataset to include both uppercase and lowercase letters, posing additional complexity due to similarities between characters (e.g., "i" vs. "j").

We will develop a deep learning model to classify handwritten letters from the EMNIST dataset, Experiment with hyperparameters and architectures to optimize performance and Analyze model behavior, including misclassification patterns and the impact of lettercase.

Methodology

Data Exploration

- **Dataset Overview**:
    - EMNIST Letters (26 classes for uppercase and lowercase letters).
    - 124,800 training and 20,800 testing grayscale images (28x28 pixels).
    - Balanced class distribution across letters, but uppercase and lowercase versions of the same letter (e.g., "C" vs. "c") were treated as distinct classes.

- **Visual Inspection**:
    - visualize 9 training samples.
    - Observed variations in stroke thickness, rotation, and writing styles.
    - Ambiguities in lowercase letters (e.g., "i" vs. "l").
    - Uppercase letters exhibited more consistent shapes, while lowercase letters had overlapping features (e.g., "q" vs. "g").

- **Class Distribution Analysis**:
    - Uppercase classes showed slightly higher accuracy in preliminary tests, likely due to clearer structural distinctions.
    - No significant class imbalance, but case-based differences were noted (e.g., fewer instances of rarely used letters like "Z").

Data Preprocessing

- **Flattening**: Images (28x28 pixels) were converted into 784-dimensional vectors.

- **Normalization**: Pixel values scaled to [0, 1].
- **One-Hot Encoding**: Labels transformed into categorical vectors for 26 classes.
- **Train/Test Split**: 124,800 training and 20,800 testing samples.

Model Architecture

- **Baseline**: A fully connected neural network with:
    - Input layer (784 units).
    - Hidden layers (varied across experiments, e.g., 128, 64, or 512 neurons).
    - Output layer (26 units with Softmax activation).
- **Activations**: ReLU, GELU, Sigmoid, and Tanh tested.
- **Optimizer**: Adam with learning rate scheduling (exponential decay).

Training Process

- **Experiments**: 8 trials varying layers, neurons, and activations.
- **Regularization**: Early stopping (patience=5) to prevent overfitting.
- **Validation**: 5-fold cross-validation for robust evaluation.

Results

Performance Metrics

- **Best Model**: Experiment 7 (1 hidden layer, 512 neurons, Tanh) achieved:
- **Accuracy**: ~91% (test).
- **Loss**: ~0.29 (test).
- **Precision** ~92% (test).
- **Recall**: and ~90% (test).

Key Observations

- **Uppercase vs. Lowercase**: Uppercase letters were easier to classify due to distinct shapes (e.g., "I" vs. "L"), while lowercase letters (e.g., "i" vs. "l") caused confusion.
- **Misclassifications**: Common errors included "i" vs. "j" and "q" vs. "g".
- **Batch Size**: Smaller batches (8) led to faster convergence but higher noise; larger batches stabilized training.

Visualizations

- **Training Curves**: Validation accuracy plateaued after ~10 epochs.
- **Confusion Matrix**: Highlighted class-specific errors.
- **Misclassified Samples**: Visual inspection revealed ambiguous handwriting.

Discussion

Key Findings

- **Model Simplicity**: A single hidden layer with 512 neurons outperformed deeper architectures, suggesting EMNIST's simplicity.
- **Data Challenges**: Lowercase letters and stylistic variations increased complexity.
- **Hyperparameter Sensitivity**: Learning rate scheduling and early stopping were critical for stability.

Challenges

- Letter similarity and case sensitivity.
- There is a slight overfitting.

Lessons Learned

- **Data Preprocessing**: Normalization and one-hot encoding were vital for convergence.
- **Interpretability**: Visualizations (confusion matrices) provided actionable insights.

Conclusion

The best model achieved **91% test accuracy** using a single hidden layer, demonstrating that simpler architectures can effectively handle EMNIST. Key challenges included letter similarity and case sensitivity. Future work could explore CNNs or address class imbalance for further improvements.