Sign Language Recognition Using Neural Networks

A Two-Layer Implementation for Static Gesture Recognition

Oleksandr Solovei

Universidade de Aveiro

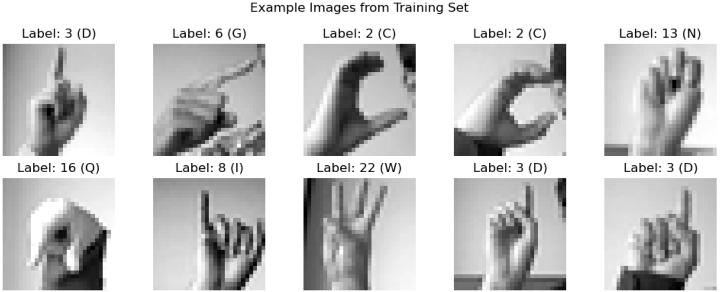
November 28, 2024

Overview

Problem Statement & Motivation

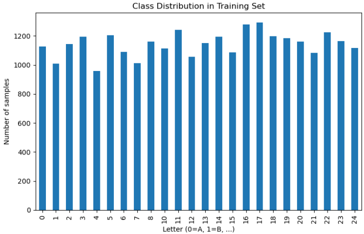
- **Goal:** Develop an accessible sign language recognition system
- **Approach:** Two-layer neural network for static gesture classification
- **Dataset:** Sign Language MNIST
    - 27,455 training images
    - 7,172 test images
    - 24 ASL letters (excluding J, Z which require motion)
- **Impact:** Enhanced communication tools for deaf community

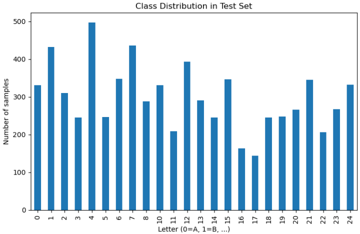
Dataset Characteristics



- 28×28 grayscale images (784 features)
- Centered hand gestures
- Varying lighting conditions

Dataset Distribution





- Imbalanced classes (144-498 samples per class)
  - Most frequent: Letter E (498 samples)
  - Least frequent: Letter Q (144 samples)

Dataset Examples

Preprocessing Pipeline

- **Data Normalization**
  - Pixel scaling: $X_{normalized} = \frac{X}{255}$
  - Range: [0, 1]
- **Label Processing**
  - One-hot encoding (24 classes)
  - Label adjustment for J, Z gaps
- **Data Splitting**
  - Training Set: 27,455 samples
  - Testing Set: 7,172 samples

Model Architecture

- **Network Structure**
  - Input Layer: 784 neurons (28×28 pixels)
  - Hidden Layer: 256 neurons
  - Output Layer: 24 neurons (one per letter)
- **Activation Functions**
  - Hidden Layer: Sigmoid
  - Output Layer: Sigmoid
- **Parameters**
  - Total trainable parameters: 207128
  - Xavier initialization

Xavier Initialization

- **Layer-specific scaling factors:**

$$\epsilon_1 = \sqrt{\frac{6}{784+256}} \approx 0.084 \qquad \epsilon_2 = \sqrt{\frac{6}{256+24}} \approx 0.149$$

- **Weight matrices** initialized uniformly:

$$W_1 \sim U(-\epsilon_1, \epsilon_1) \qquad W_2 \sim U(-\epsilon_2, \epsilon_2)$$

- **Bias Vectors:**

$$b_1 = 0 \in \mathbb{R}^{256\times 1} \qquad b_2 = 0 \in \mathbb{R}^{24\times 1}$$

Benefits

- Prevents vanishing/exploding gradients
- Maintains activation variance across layers
- Enables faster convergence

Momentum in Training

- **Standard Gradient Descent:**

$$W = W - \alpha \nabla L$$

Only uses current gradient

- **Momentum Update (= 0.9):**

$$v = \beta v - \alpha \nabla L$$

$$W = W + v$$

Accumulates previous updates

- **Benefits:**
  - Accelerates training in consistent directions
  - Helps escape local minima
  - Reduces oscillations in gradient updates

Training Strategy

- **Optimization Parameters**
  - Initial learning rate: 0.1
  - Momentum (): 0.9
  - Batch size: 64
  - Total iterations: 80
- **Regularization Techniques**
  - L2 penalty (): 0.01
  - Decay: 0.95 / 50 steps
- **Loss Function Components**
  - Binary cross-entropy
  - L2 regularization term

Mathematical Framework

Forward Propagation:

$Z_1 = W_1X + b_1$
$A_1 = \sigma(Z_1)$

$Z_2 = W_2A_1 + b_2$
$A_2 = \sigma(Z_2)$

Loss Function:

$$L_{BCE} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{24}[y_k^{(i)}\log(a_k^{(i)}) + (1 - y_k^{(i)})\log(1 - a_k^{(i)})] \quad (1)$$

$$L_{total} = L_{BCE} + \frac{\lambda}{2m}\sum_{i,j}(W_{1_{ij}}^2 + W_{2_{ij}}^2) \quad (2)$$

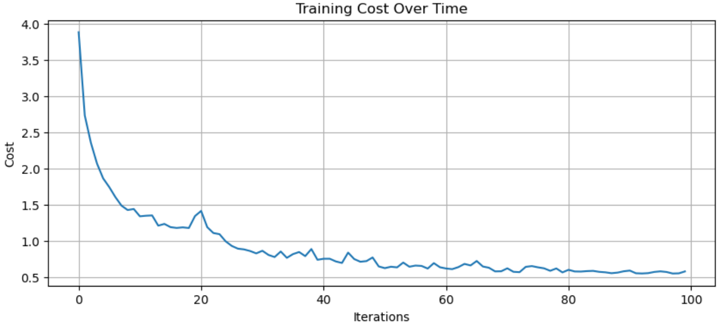
Weight Updates:

$v_W = \beta v_W - \alpha\frac{\partial L}{\partial W}$
$W = W + v_W$

Learning Rate Decay:

$\alpha_t = \alpha_0 \cdot 0.95^{\lfloor t/50 \rfloor}$

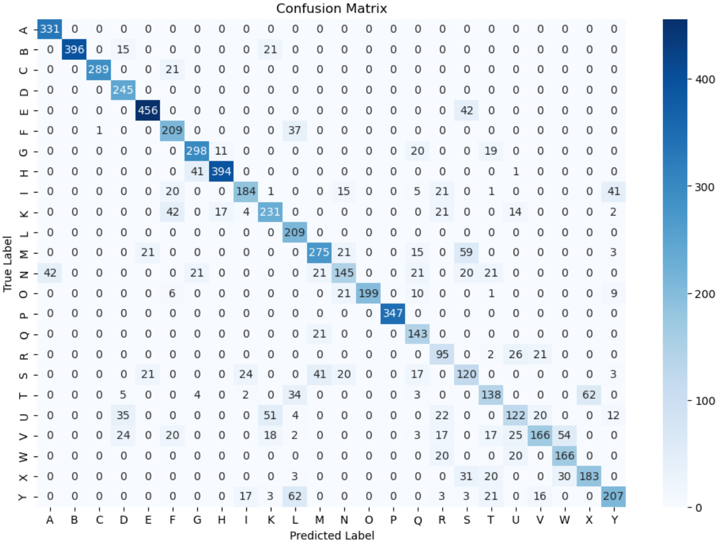
Cost Over Time Analysis



Performance Analysis

- **Overall Metrics**
  - Test Accuracy: 77.36%
  - Weighted F1-score: 0.77
  - Macro F1-score: 0.75
- **Best Performing Letters (F1-score)**
  - O: 1.00
  - B: 0.96
  - C: 0.96
  - A: 0.94
- **Challenging Letters (F1-score)**
  - R: 0.46
  - Q: 0.55
  - M: 0.57
  - S: 0.57

Confusion Matrix



Interactive Web Implementation

- **User Interface:**
  - Live webcam feed
  - Image capture functionality
  - Preview of original capture
  - Processed 28×28 preview
  - Top-3 predictions display
  - Confidence percentages
- **Processing Pipeline:**
  - Real-time grayscale conversion
  - Size normalization to 28×28
  - Pixel value normalization (0-1 range)
- **Network Visualization:**
  - Layer-by-layer activation monitoring
  - Neuron activity visualization
  - Confidence distribution display

Improvements & Extensions

- **Model Enhancements**
  - Data augmentation for underrepresented classes
  - Feature engineering for hand shape detection
  - Deeper architecture exploration
- **Practical Applications**
  - Real-time recognition system
  - Mobile application development
  - Educational tools
- **Research Extensions**
  - Dynamic gesture recognition
  - Multi-modal approaches
  - Transfer learning exploration

Conclusion

- **Key Achievements**
  - Successful static gesture recognition
  - Balanced performance-complexity trade-off
  - Identified clear paths for improvement
- **Impact**
  - Foundation for accessible communication tools
  - Benchmark for future implementations
  - Insights for sign language recognition systems