Classification des Caractères Tifinagh (niveaux de gris) avec un Réseau de Neurones Multiclasses

hanane IQLI

June 2025

1 Abstract

In this project, we developed a multiclass neural network based on a multilayer perceptron (MLP) architecture, featuring two hidden layers with ReLU activation functions and a softmax output layer for classifying 33 Tifinagh characters. The model was trained on grayscale images resized to 32×32 pixels and achieved an overall test accuracy of 89%. Performance was particularly strong for certain characters, such as yat and yagg, with F1-scores exceeding 0.95. However, other classes exhibited significant precision-recall imbalances. The results demonstrate good generalization capabilities, while also highlighting areas for improvement, especially in recognizing visually similar or structurally complex characters. Potential enhancements include adding more hidden layers and applying targeted data augmentation strategies.

2 Introduction

Handwritten character recognition is a major challenge in computer vision, especially for under-explored scripts such as Tifinagh, used by Amazigh communities. This project aims to develop a multiclass neural network to classify these characters, relying on a multilayer perceptron (MLP) with two hidden layers containing 64 and 32 neurons, respectively. The model is trained on images resized to 32×32 pixels and flattened into 1024-dimensional feature vectors.

3 Methodology

3.1 Forward Operation

The forward propagation process in a multilayer neural network proceeds layer by layer through the following steps.

Propagation in Layer l

For a layer l, the linear output $Z^{[l]}$ and the activation $A^{[l]}$ are given by:

$$Z^{[l]} = A^{[l-1]}W^{[l]} + b^{[l]} \tag{1}$$

$$A^{[l]} = g^{[l]}(Z^{[l]}) \tag{2}$$

- $A^{[l-1]}$: output from the previous layer (or input to the network if $l = 1$),
- $W^{[l]}$: weight matrix of layer l,
- $b^{[l]}$: bias vector of layer l,
- $g^{[l]}$: activation function used in layer l.

Input Layer

$$A^{[0]} = X, \quad \text{where } X \in \mathbb{R}^{m\times 1024}$$

Each input is a 32×32 image flattened into a 1024-dimensional vector. The input batch contains m examples.

Hidden Layers ($l = 1, 2$)

Both hidden layers use the ReLU activation function:

$$g^{[l]}(z) = \text{ReLU}(z) = \max(0, z) \tag{3}$$

This function introduces non-linearity while remaining easy to differentiate.

Output Layer ($l = 3$)

The output layer applies the *softmax* function, enabling the outputs to be interpreted as class probabilities:

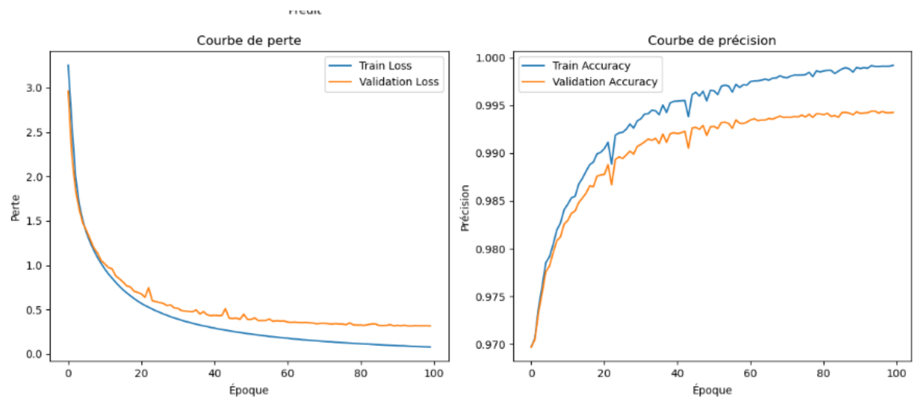
$$A^{[3]} = \text{softmax}(Z^{[3]}), \quad \hat{y}_{i,c} = \frac{e^{z_{i,c}}}{\sum_{j=1}^{33} e^{z_{i,j}}} \tag{4}$$

- $\hat{y}_{i,c}$ is the predicted probability that example i belongs to class c,
- The sum of probabilities across the 33 classes is equal to 1 for each example.

Overall Model Architecture

- Input: vectors of size 1024 (from 32×32 images),
- Layer 1: $1024 \to 64$ neurons, ReLU activation,
- Layer 2: $64 \to 32$ neurons, ReLU activation,
- Output layer: $32 \to 33$ neurons, softmax activation.

4 Resultants



The figure above displays training and validation metrics for the model, particularly loss and accuracy curves. The Train Loss and Validation Loss show how the model's error decreases over epochs, while the Train Accuracy and Validation Accuracy demonstrate improved performance, reaching high precision levels between 0.970 and 0.995. The x-axis spans from 0 to 100 epochs. The high accuracy values, especially nearing 0.995, indicate strong performance on both training and validation datasets with minimal overfitting. The smooth curves suggest stable training. Overall, the results reflect a highly effective model with strong generalization capabilities.

Evaluation of Tifinagh Alphabet Recognition

Global Performance

- **Overall accuracy**: 89% (5637 test samples)
- **Number of classes**: 33 Tifinagh characters
- **Distribution**: Approximately 171 samples per class

Key Observations

- **Best-performing characters** (F1-score ¿ 0.93):
  - yad, yagg, yagh, yat, yax, yan
- **Notable issues**:
  - Class **ya**: Very low precision (0.47) but high recall (0.96)
  - Classes with low recall (¡0.80): yatt, yss, yu

5 Discussion

The proposed architecture, based on a two-layer neural network using ReLU in the hidden layers and softmax at the output, demonstrates promising results for Tifinagh character classification. With an overall accuracy of 89%, the model shows strong ability to distinguish most of the 33 characters, especially for classes like yat and yagg, which achieve F1-scores above 0.95. However, performance varies significantly across classes. For example, class ya shows a striking imbalance between precision (0.47) and recall (0.96), suggesting overgeneralization, possibly due to visual similarities or training data imbalance. Other classes such as yatt and yss suffer from low recall (¡0.80), indicating that the model struggles to identify them reliably.

To improve these outcomes, several enhancements could be explored: adding a third hidden layer to capture more complex features, applying targeted data augmentation for problematic classes, or incorporating attention mechanisms to better identify distinctive character features. While the current model provides a strong foundation, such improvements could enhance both consistency and accuracy, reducing confusion between visually similar characters. Continued experimentation with these approaches may help bridge performance gaps and adapt the model for real-world applications like Tifinagh OCR systems.

6 Conclusion

The current two-layer architecture (ReLU hidden layers and softmax output) achieves a commendable 89% accuracy in Tifinagh character recognition, revealing both strengths and areas for improvement. The model performs well with distinctive characters like yat (F1=0.96) and yagg (F1=0.95), showcasing effective feature extraction. However, three key limitations remain: (1) significant precision-recall imbalance in confusing classes (e.g., ya: precision=0.47, recall=0.96), (2) low recall for structurally complex characters (e.g., yatt, yss), and (3) potential underfitting suggested by consistent underperformance across similar character groups.

7 Reference

https://github.com/iqlihanane/Classification-des-Caract-res-Tifinagh-niveaux-de-gris-avec-un-R-seau-de-Neurones-Multiclasses.git