# Recognition of Sign Language Digits

João Nuno Valente

universidade de aveiro

# Table of contents

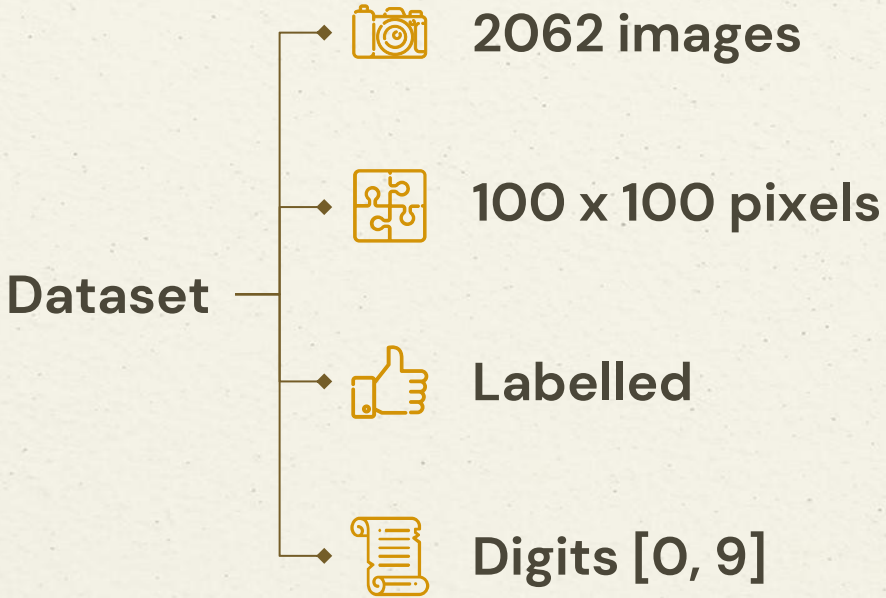
# 01
# Introduction

Contextualization of the problem

# Sign Language Digits

Sign languages are languages that use the visual-manual modality to convey meaning, instead of spoken words.

# Dataset




These were split into two parts, 80/20: 1.649 for the training set and 413 for the test set.

They were reshaped to 64×64 pixels and converted to grayscale.

For some reason, the labels were not properly assigned, but this was fixed.

The dataset consists of 10 classes.

# Neural Networks Approach

## Neural Networks

### FCNN

- Input layer: Takes in the raw data (e.g., pixel values of an image)
- Hidden layers: Each layer consists of neurons that apply an activation function

### CNN

- Primarily used for image classification, object detection
- Consists of convolutional layers
- Uses pooling layers to reduce spatial dimensions

### My approach

Involves testing both Neural Networks and comparing their results.

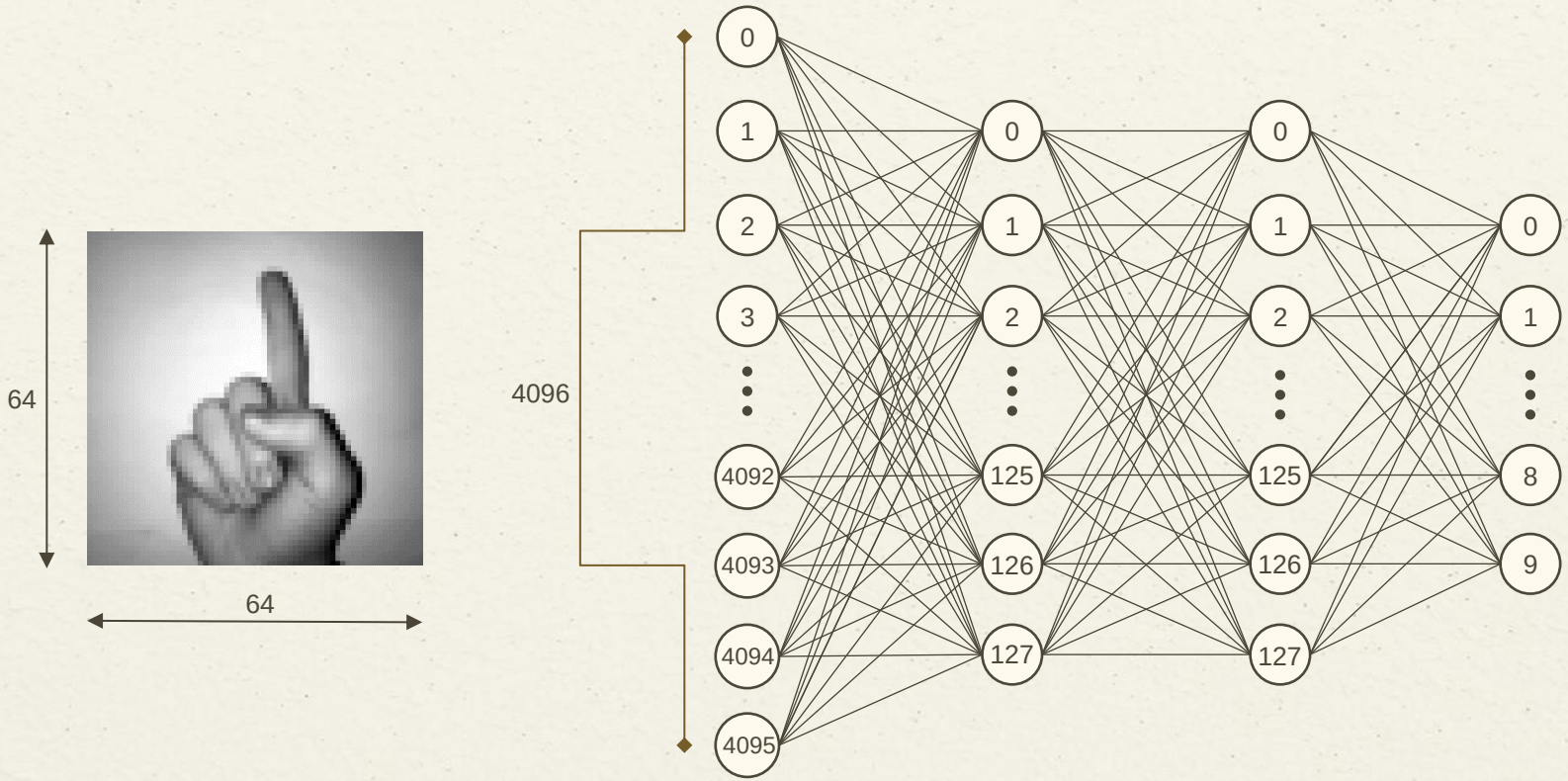
# 02
# FCNN

Fully Connected Neural Network architecture

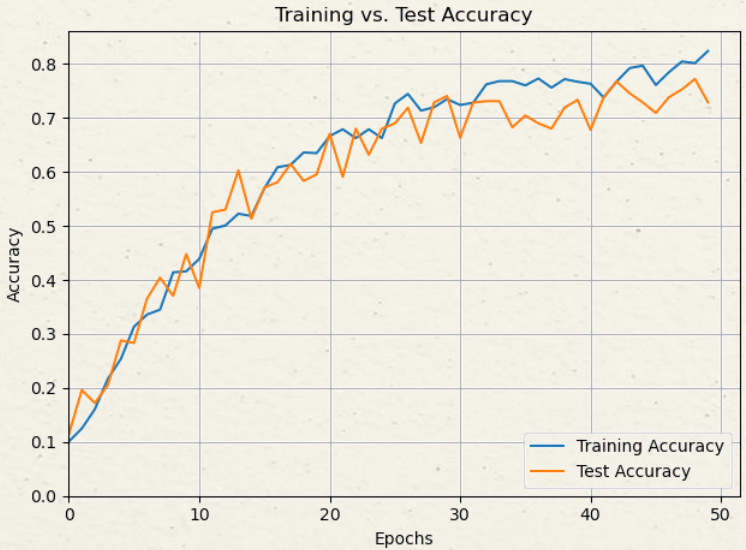
# FCNN Architecture I

- Input Layer:
  - It flattens the 64x64 image into a 1D vector of 4096 values
- First Hidden Layer:
  - 128 neurons with **ReLU** activation.
- Second Hidden Layer:
  - Another 128 neurons with **ReLU** activation.
- Output Layer:
  - 10 neurons, used for the 10-class classification using **Softmax.**

# FCNN Architecture II



# Analysis of the results




## Parameters

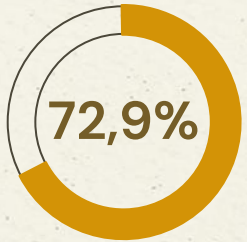
The total number of parameters is 542,218.

## Epochs

50 iterations were performed.

## Accuracy

The accuracy using the test set was 72,9%.



# 03
# CNN

Convolutional Neural Network architecture

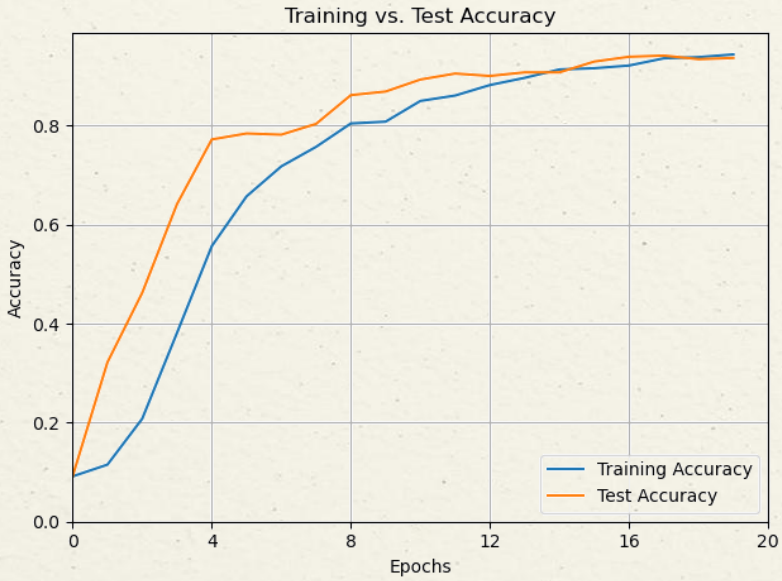
# CNN Architecture I

- Input Layer:
  - L0: Accepts grayscale images of 64×64 pixels.
- Convolutional & Pooling Layers:
  - L1: Conv2D (32 filters, 3×3 kernel, ReLU activation) → Extracts features from the image.
  - L2: MaxPooling2D (2×2) → Reduces spatial dimensions.
  - L3: Conv2D (64 filters, 3×3 kernel, ReLU activation) → Deeper feature extraction.
  - L4: MaxPooling2D (2×2) → Reduces spatial dimensions.
  - L5: Conv2D (128 filters, 3×3 kernel, ReLU activation) → Captures more complex patterns.
  - L6: MaxPooling2D (2×2) → Reduces spatial dimensions.

# CNN Architecture II

- Fully Connected Layers:
  - L7: Flatten Layer → Converts the feature maps into a 1D vector for the Dense layers.
  - L8: Dense (128 neurons, ReLU activation) → Fully connected layer for learning high-level features.
  - L9: Dropout (0.5) → Reduces overfitting by randomly disabling 50% of neurons during training.
- Output Layer:
  - L10: Dense (10 neurons, Softmax activation) → Produces probabilities for 10 classes.

# Analysis of the results




## Parameters

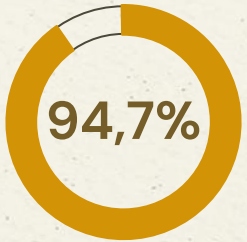
The total number of parameters is 683,914.

## Epochs

20 iterations were performed.

## Accuracy

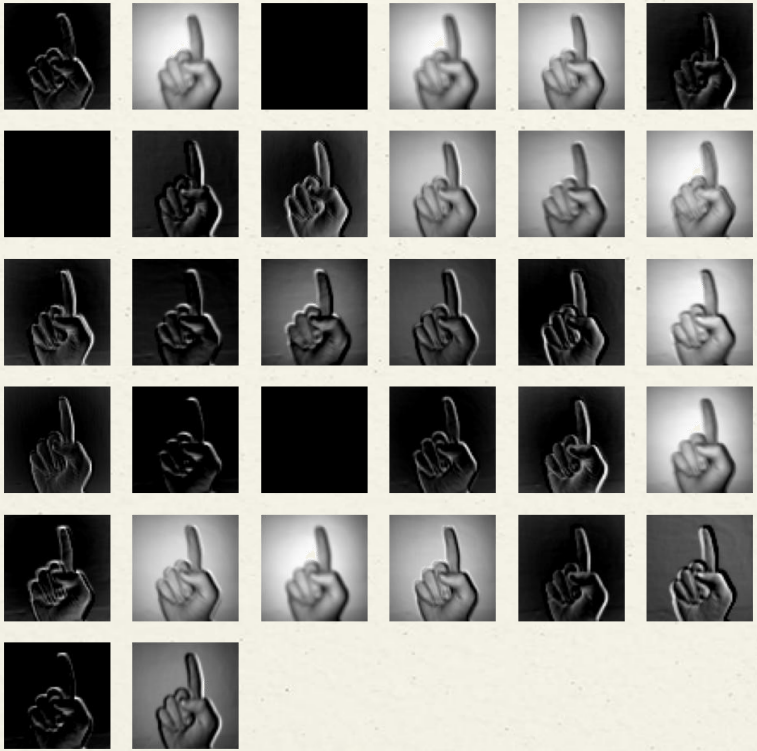
The accuracy using the test set was 94,7%.



# 04
# Visualization

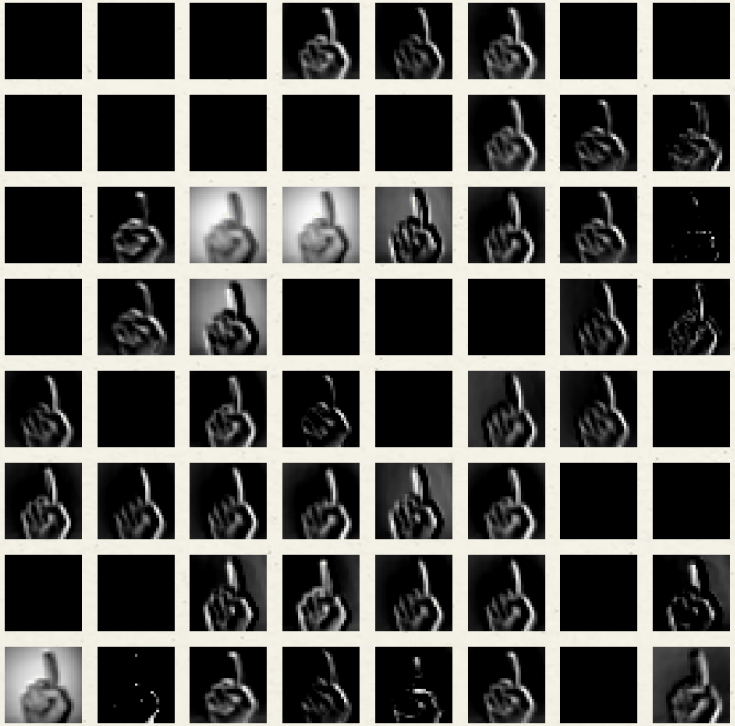
Image example through a convolution transformation

# L1: Convolution



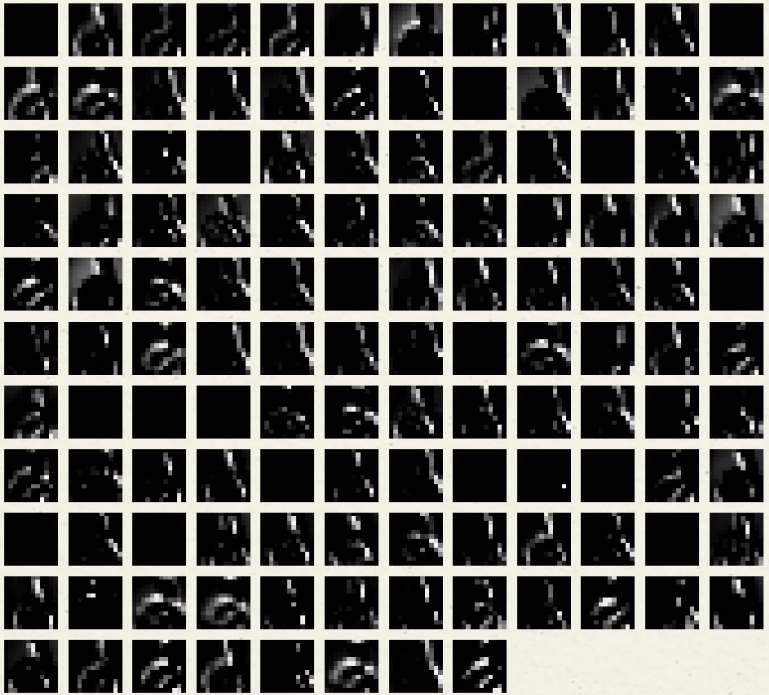
# L2: MaxPooling

# L3: Convolution

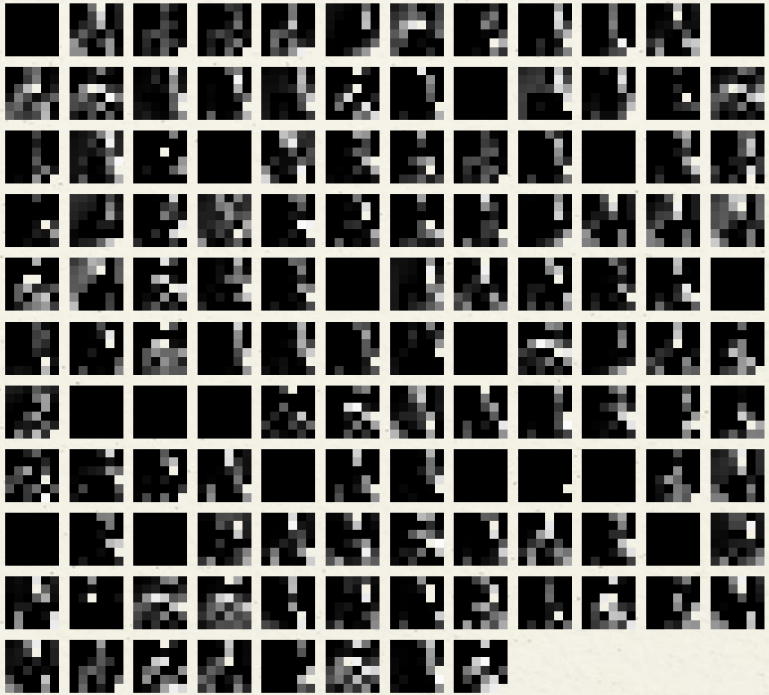


# L4: MaxPooling

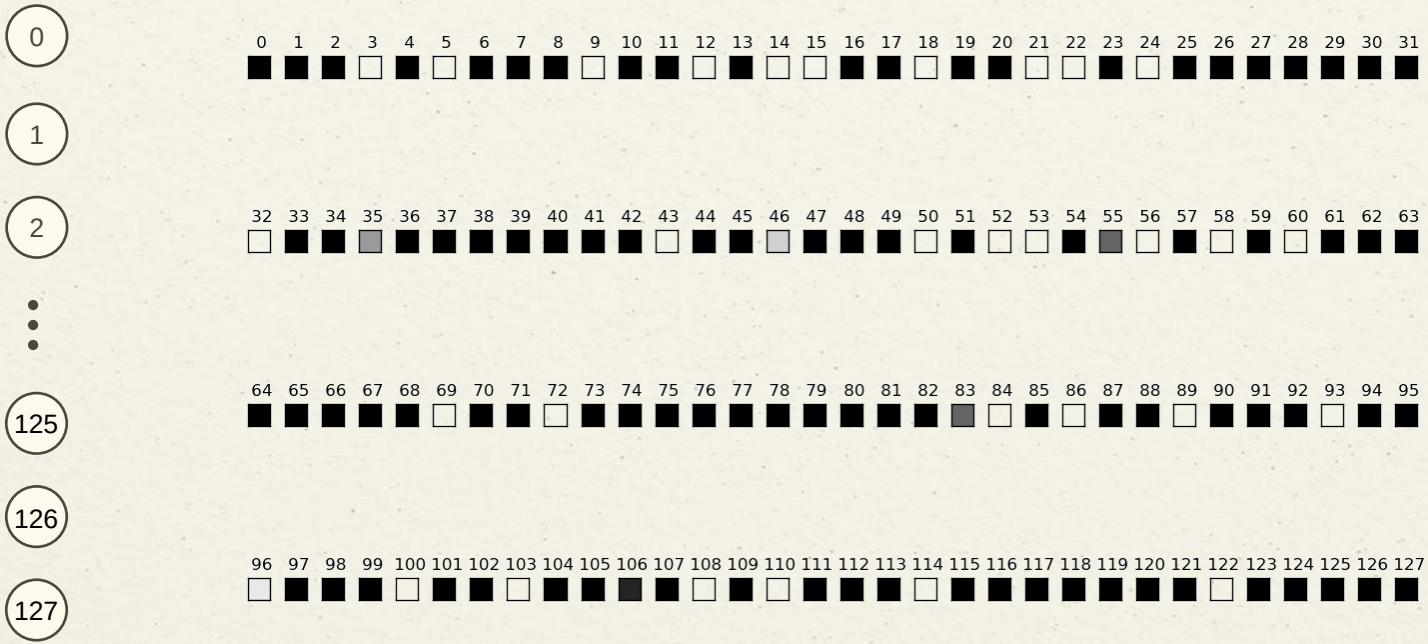
# L5: Convolution



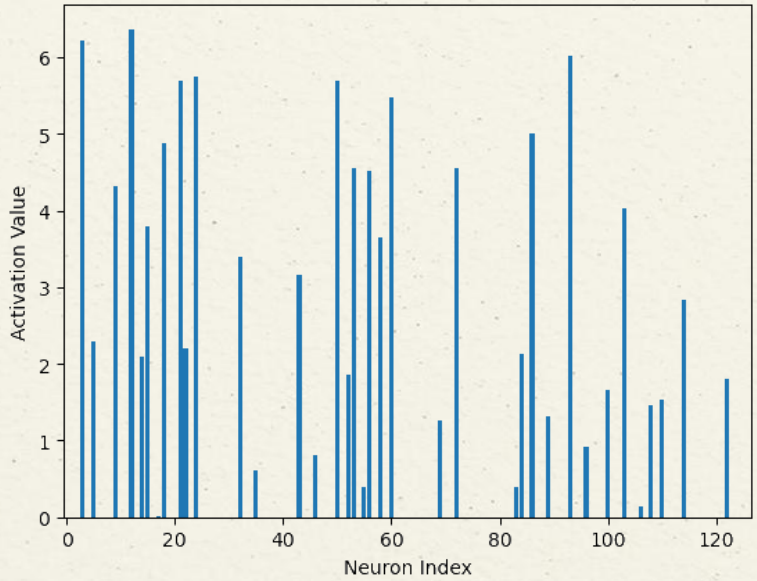
# L6: MaxPooling



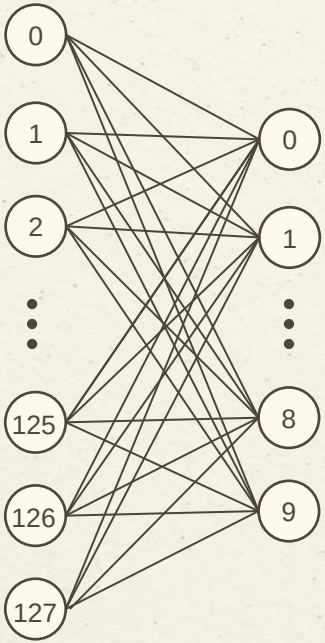
# L8 | Fully connected

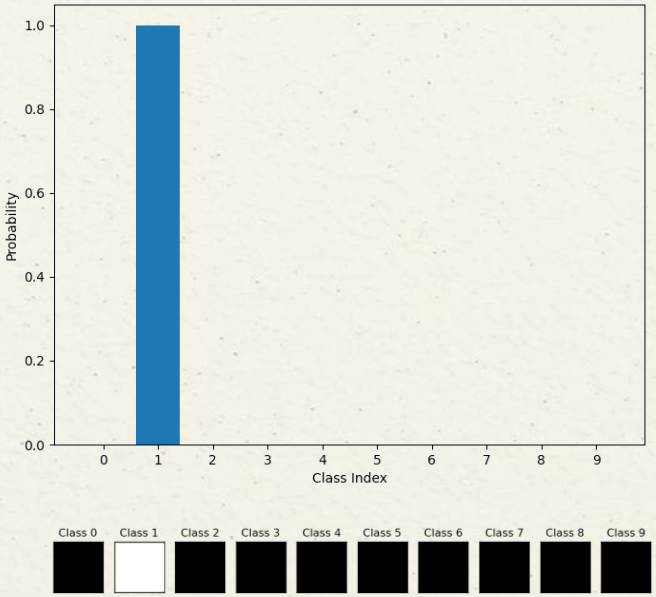


# L8 | Fully connected – Bar plot

# L10 | Final result



# 05
# Final remaks

Challenges and useful resources

# Challenges

## What is the best architecture configuration?

After some testing, I found these configurations interesting. However, I have no idea why they work, while others do not.

## What is the impact of the batch size?

I just kept changing the batch size to see which value would perform better.

## How many epochs are too many epochs?

At some point, I had 200 epochs, at which point a friend of mine politely asked, 'What exactly are you training that needs 200 epochs?'.

# Useful resources

## 3Blue1Brown channel

This YouTube channel has a fascinating playlist that visually explains how Neural Networks work.

## Michael Nielsen's Book

*Neural Networks and Deep Learning* is a fantastic free online book that covers the mathematics of Neural Networks.

# Thanks!

## Do you have any questions?

- jnvalente@ua.pt
- JoaoNunoValente.github.io
- @JoaoNunoValente