PHY473R

Clément Lenoble & Gabriel Pereira de Carvalho

May 30, 2024

École polytechnique

Project overview

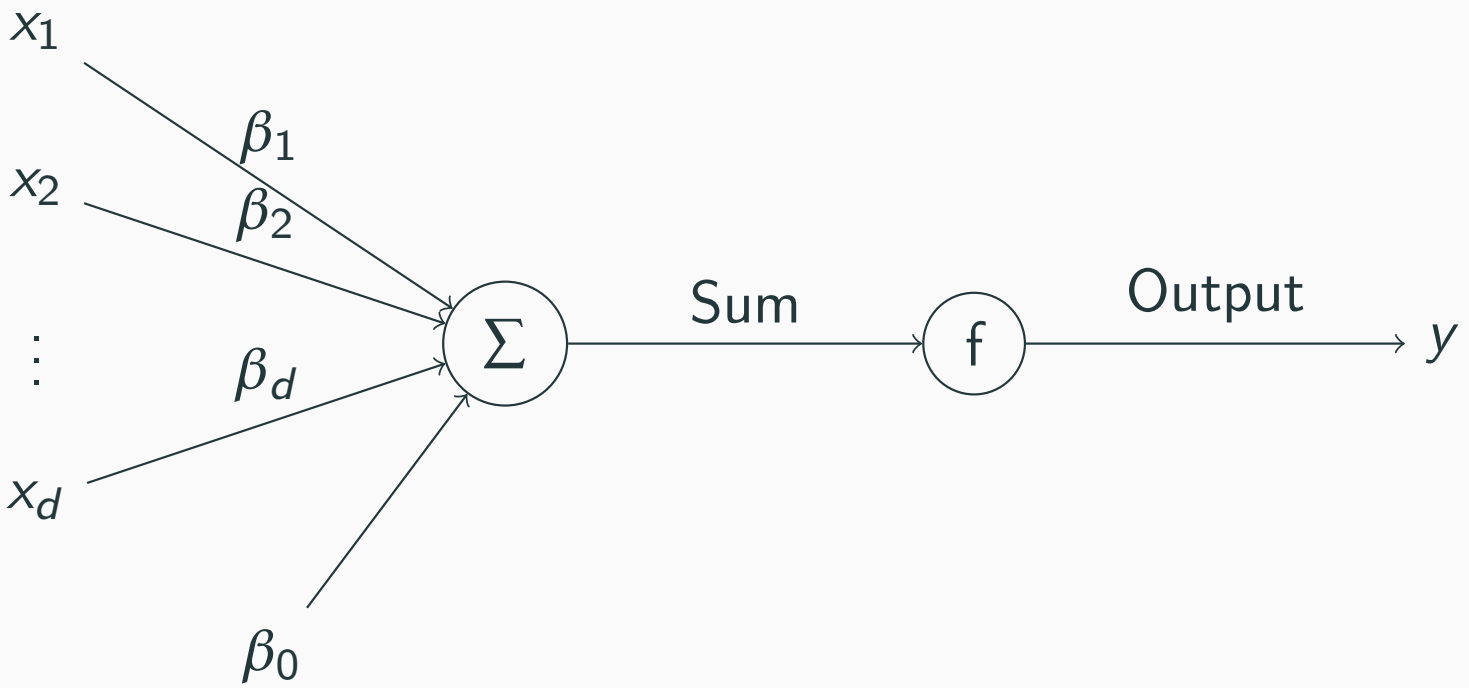
- Aim : character classification
- Hardware implementation (FPGA)
- Machine learning (perceptron)
- Results :
  - Efficient for binary classification
  - Multimodal classification
  - Working and independent solution with integrated camera, algorithm and results on FPGA

Table of contents

Theoretical question : Machine learning for characters recognition and FPGA implementation

Perceptron model

$x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, $\beta = (\beta_1, \ldots, \beta_d) \in \mathbb{R}^d$, $\beta_0 \in \mathbb{R}$



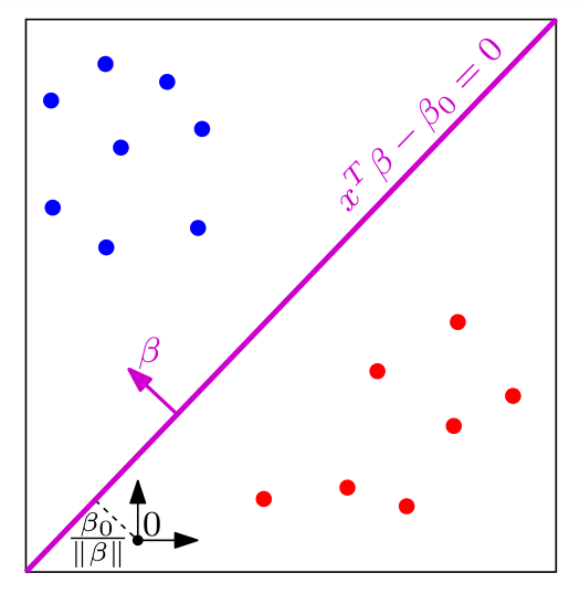
Output of the perceptron is given by:

$$y = f\left(\sum_{i=1}^{d} \beta_i x_i + \beta_0\right) \quad \text{in our case} \quad f(x) = \begin{cases} 0 & \text{si } x < 0, \\ 1 & \text{si } x \geq 0. \end{cases}$$

Perceptron model

Let us suppose the data is linearly separable in $\mathbb{R}^d$, which implies there exists a hyperplane $x^\top \beta + \beta_0 = 0$ that separates the two classes of data.

Therefore, $\forall x$, $y(x)$ will be either positive or negative, and the sign of $y(x)$ gives us the class.



$$y(x) = f\left(\sum_{i=1}^{d} \beta_i x_i + \beta_0\right)$$

$$f(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{if } x \geq 0. \end{cases}$$

Figure 1: Separating hyperplane

Perceptron model: Training

Let $X = ((x_1, y_1), ..., (x_n, y_n))$ be the training set,

$x_i = (x_{i,1}, ..., x_{i,d})$ is a training sample, and $y_i \in \mathbb{R}$ is the response associated to the input x_i.

$$y(x) = \beta_0 + \sum_{i=1}^{d} \beta_i x_i$$

The cost function J used is the Mean Squared Error (MSE) :

$$J(\beta) = \frac{1}{2n} \sum_{j=1}^{n} (y_j - y)^2$$

With replacing y by its value :

$$J(\beta) = \frac{1}{2n} \sum_{j=1}^{n} \left(y_j - \left(\beta_0 + \sum_{i=1}^{d} \beta_i x_{ji} \right) \right)^2$$

Perceptron model: Training

The aim of the training is to minimize $J(\beta)$.

The SGD method optimize, at each iterations, the value of the the weights and biases using gradient descent :

$$\begin{cases} \beta_0 & = \beta_0 - \eta \cdot \frac{\partial J}{\partial \beta_0} \\ \beta_i & = \beta_i - \eta \cdot \frac{\partial J}{\partial \beta_i} \end{cases} \tag{1}$$

Repeat until convergence, or stop criteria.

We have use python library for training and get the right biases

Data format

- Aim : keep all information while reducing the memory size
- Size : 256*256
- Black and white, with variable threshold

0	1	0	1
1	0	1	0
0	1	0	1
1	0	1	0

Flatten $\longrightarrow$

0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0

Tranining our model

Figure 2: MNIST dataset

Over **8000** to train our perceptron model (80% train/ 20% test).

Training the model

To fit the requirements of our FPGA project, we had to:

- Resize the images to 256x256.
- Transform them into black(0) and white(1) with pattern in white and background in black.
- Scale the weights, so we could approximate them with integers.
- Write the weights into a `.mif` file.

Model results

Perceptron 0/1

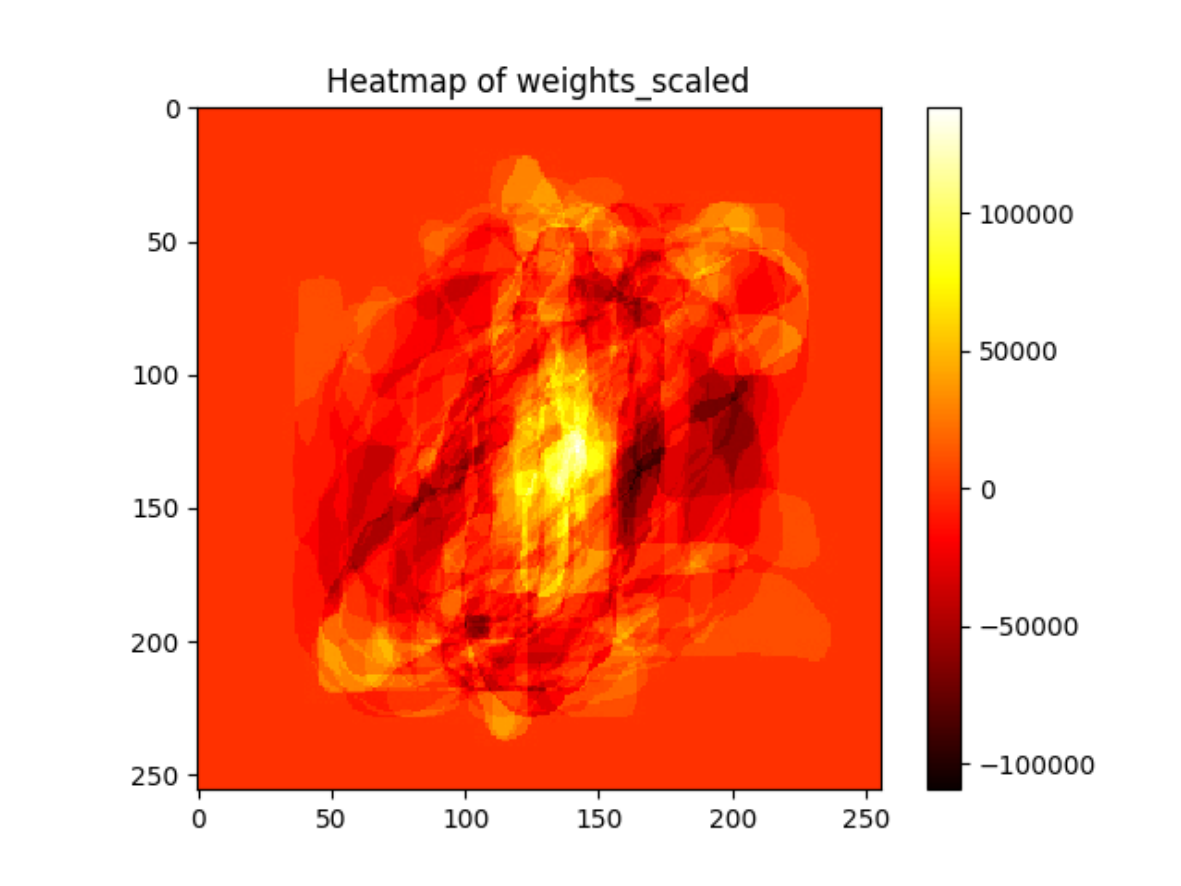


Figure 3: Heatmap for 0/1 perceptron

Accuracy on test data: 0,9989

Perceptrons 0/2 and 1/2

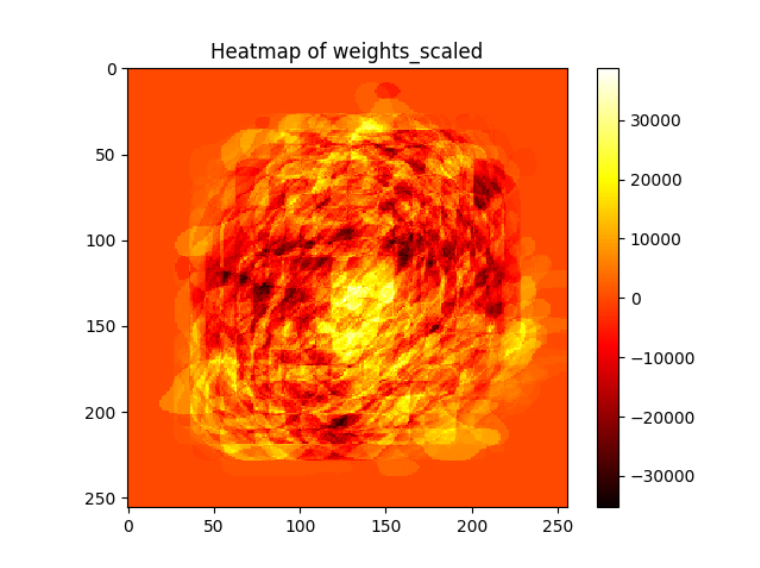


Figure 4: Heatmap for 0/2 perceptron

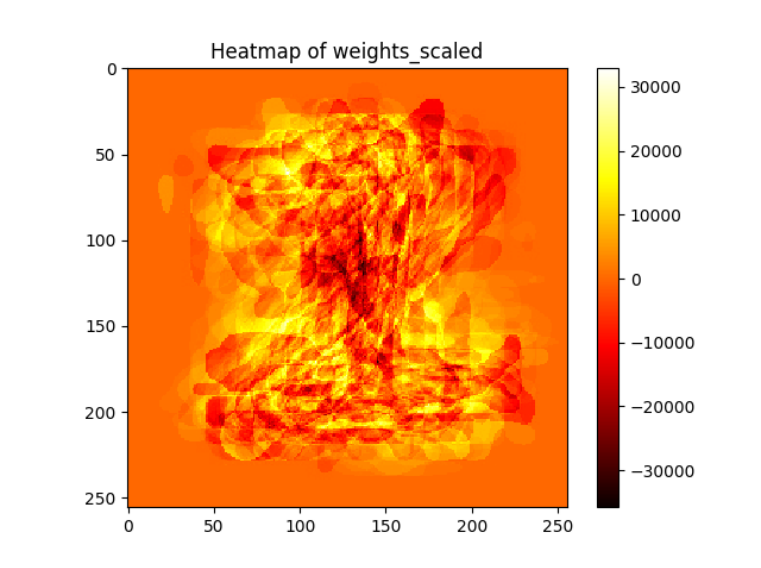


Figure 5: Heatmap for 1/2 perceptron

Circuit design

Control Unit (State Machine)

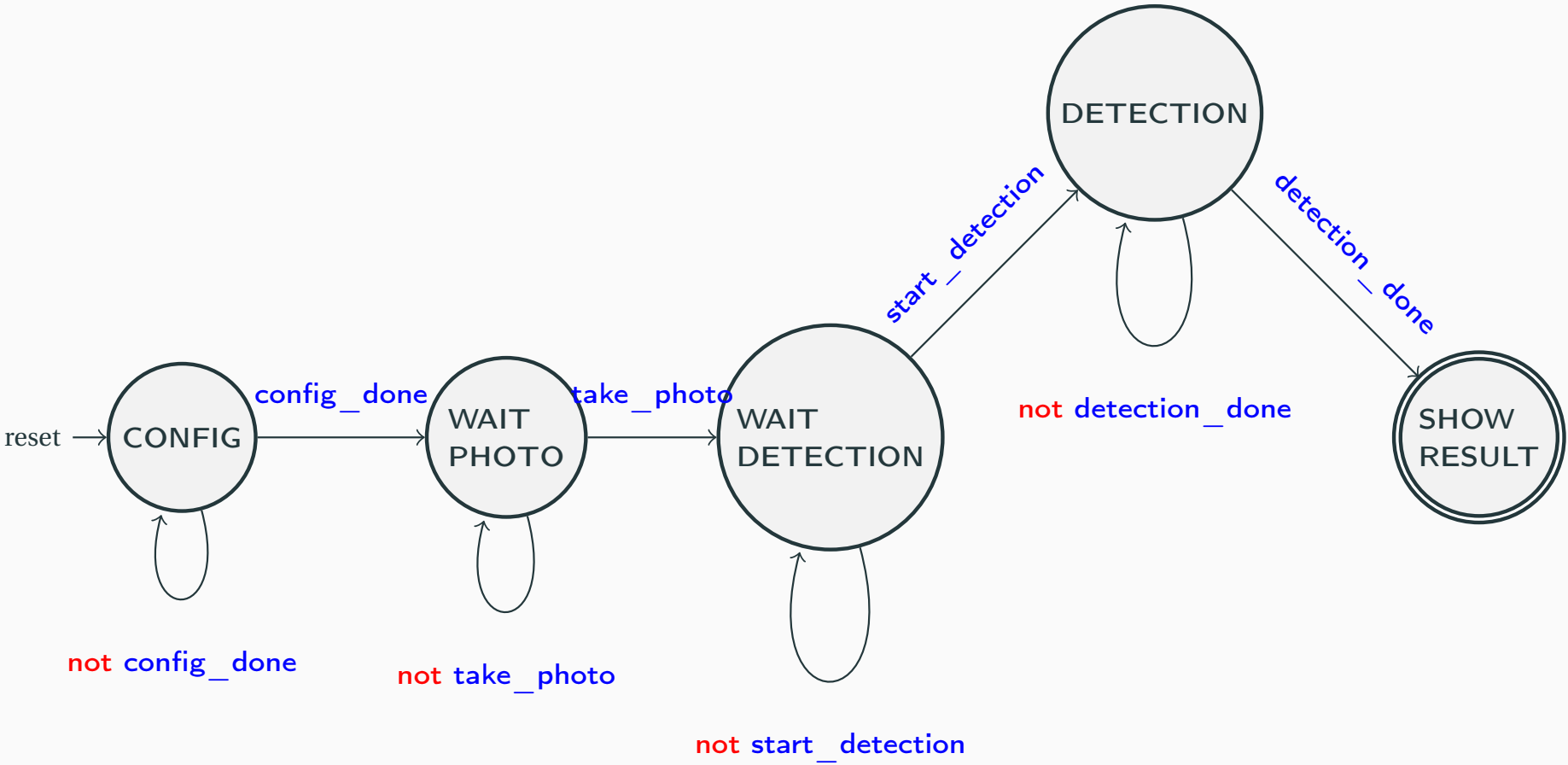
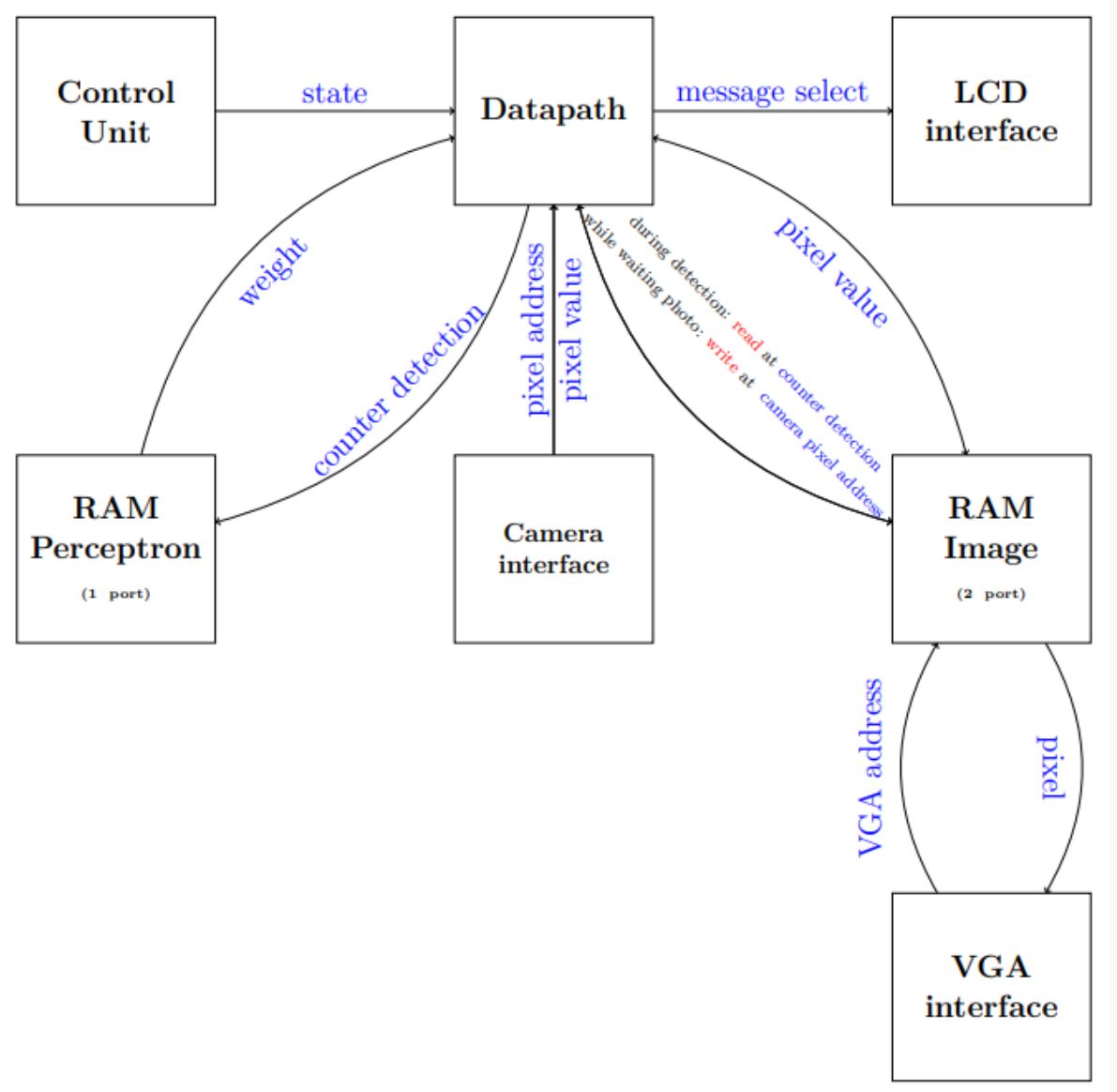


Figure 6: Control unit's state machine

Datapath


Control Unit
state
Datapath
message select
LCD interface
weight
counter detection
pixel address
pixel value
during detection: read at counter detection
while waiting photo: write at camera pixel address
pixel value
RAM Perceptron
(1 port)
Camera interface
RAM Image
(2 port)
VGA address
pixel
VGA interface

Demo!!

Next steps

- One vs One or One vs All
  - 2 architecture of multimodal classification using perceptrons
  - $O(\frac{n(n-1)}{2})$ vs $O(n)$ classifiers (and weights in memory)
  - No theoretical argument $\Longrightarrow$ test both
- Multi layer Perceptron
- Convolutional Neural Networks

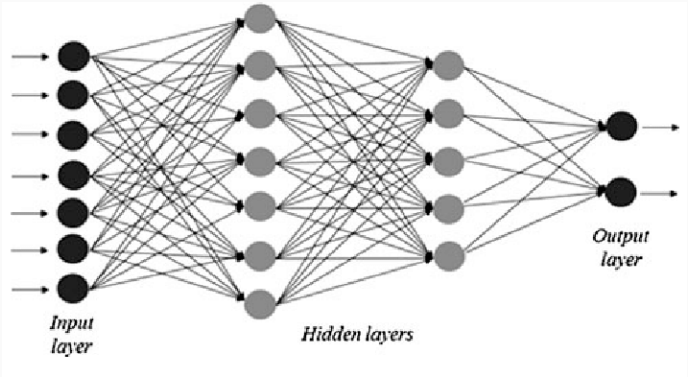


Figure 7: MLP

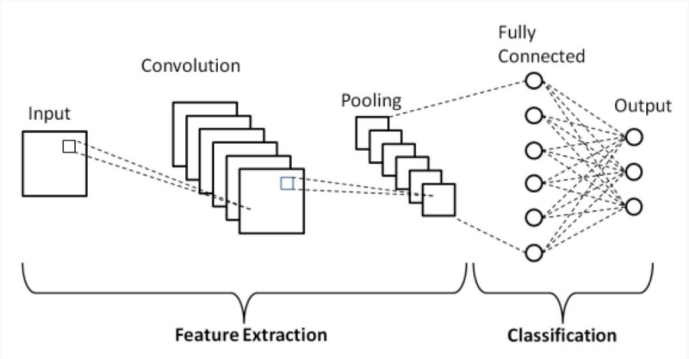


Figure 8: CNN

Conclusion

- This project lays the foundation of AI implementation using FPGA
- Very interesting work on the technical field, but also project management
- We are grateful to the teaching team for their hints and support