**InputLayer Class:** This class is the first layer of a neural network (the input layer), it is followed by one or more hidden layers. It has an Attribut shape wich will be used for the first layer.

**Abstract ConvolutionalLayer Class:** In this class convolutional layer the input keeps its original shape in order to exploit this correlation between the pixels.

**CPUConvolutionalLayer Class:** this class inherits from the Abstract ConvolutionalLayer Class using the CPU’s implementation by using the CpuWorker Class

**FPGAConvolutionalLayer Class:** this class inherits from the Abstract ConvolutionalLayer Class using the FPGA’s implementation by using the FpgaWorker class

**MaxPoolLayer Class:** These layer take the output feature maps from the convolutional layers, and its Max function is to reduce the size of the input feature maps to reduce the amount of parameters in the network.

The pooling layer works in a similar way to the convolutional layer since it operates on a subregion on the input feature maps and performs a filter operation.

**ResponseNormalizationLayer Class:** The Class Response Normalization is kept separate from the others since it may require multiple memory access patterns.

**DenseLayer Class:** this layer can use convolutional blocks for calculations, both of them have access to these blocks.

**TrainableLayer Class:** this class to train the numeral network using a weight matrix

**ActivationFunction Class:** Activation function is introduced, its purpose is to allow small changes in the weights or bias to only cause a small change on the output, this property is helpful when training a network.

**Sigmoid Class:** sigmoid is an activation function. Same as the linear classifier this network also has ten outputs since it is the same classification problem, however the hidden layer has 25 neurons.

**Tanh Class:** is an activation function

**Sofmax Class:** is an activation function

**ReLU Class:** is an activation function