**Artificial Neural Networks**

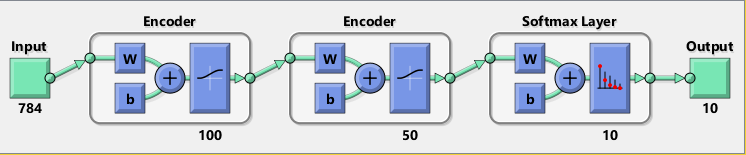
*Exercise Session 4 – Deep learning -Stacked Autoencoders and Convolutional Neural Networks,*

*report by,*

*Marimuthu Ananthavelu (r0652832), KU Leuven.*

**Stacked Autoencoders:**

Autoencoders are trained in an unsupervised, greedy, layer-wise fashion. It means No labels, beginning training with just the first layer of the network and then adding new layers on the move. The weights can be learned using a variety of techniques ranging from "batch" gradient descent to mini-batch stochastic gradient descent (SGD), to quasi-Newton methods like L-BFGS.In this work, the digits classification has been performed with the help of artificial dataset in which each digit is of 28 x 28 pixels images. The network architecture is as below;



The idea is that the weights learned in an unsupervised manner to minimize reconstruction error for the representation learning task offer a good starting point to initialize a network for a supervised discriminative task such as classification or similarity. I.e., the network learns something about the underlying distribution by looking at the unlabeled data, allowing it to discriminate between labeled data. However, the weights still need to be "fine-tuned" for this new task. So ideally it is 2 step training process with Pre-training and Post-training.

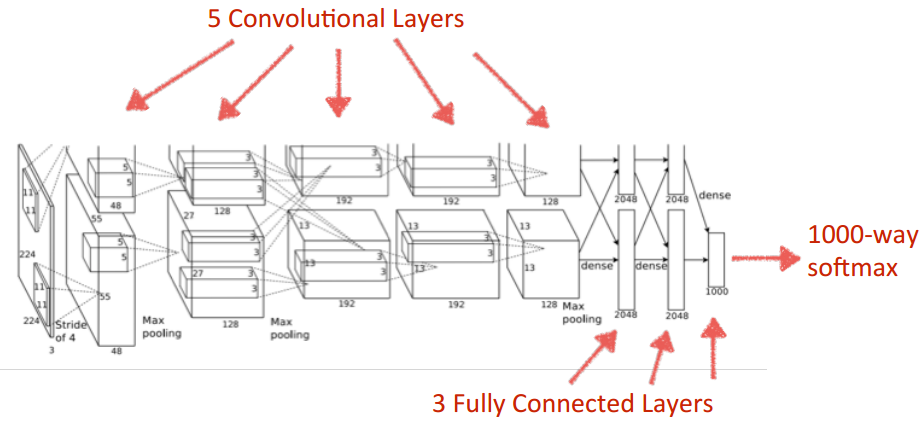
***Convolutional Neural Networks:***

**First Convolutional Layer(layer-2)**

The weights obtained from these layers captures the basic image features, such as edges, dots, bright spots, dark spots and blobs. In our dataset, they represent edges of the airplane wings, ship floor windows, laptop mousepads etc. These "primitive" features are then processed by deeper network layers, which then combine the early features to form higher level image features better suited for classification. The filters on the first layer convolve around the input image and “activate” (or compute high values) when the specific feature it is looking for is in the input volume.

**Convolutional Layers -1 to 5 (layers 2 to 6)**

The dimension of the input at the start of layer 6 is a result of convolution at the earlier layers. Convolution and Pooling reduces the input sizes and not because of the ReLU or the Cross-Channel Normalization layer. Thus from figure 4.3, the dimension of input at layer 6 will be a vector of 4096 inputs features.



**4.2.3 Final dimension of the problem:**

For the Caltech101 dataset[[1]](#footnote-1), the images of objects belong to 101 categories. The number of neurons (or features) at the final fully connected layer before the final softmax layer is 4096. The convolution net in this exercise process input images of size 227x227x3 = 154587 features. Thus the final dimension used for classification is reduced by a factor of 15457/4096 ~ 38.

1. http://www.vision.caltech.edu/Image\_Datasets/Caltech101 [↑](#footnote-ref-1)