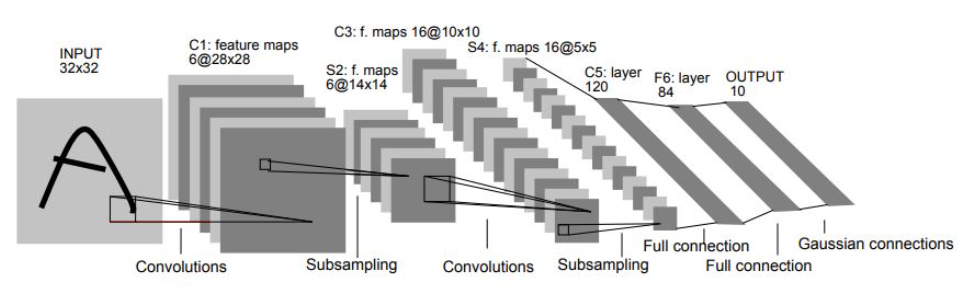
**Network On Chip bases designing and simulating neural network accelerator**

NoC architecture: 10x10 making the total count of PEs as 100.

The LeNet architecture under consideration:



With this architecture, the number of neurons in each layer is listed below:

(Note: The neurons in the input layer are not considered because as discussed the pixels will be placed in the memory)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Layer** | **Number of neurons** | **Kernels** | **Stride** | **Padding** | **Activation function** |
| Convolution | 4704 | 5x5x6 | 1 | 0 | sigmoid |
| Average pooling | 1176 | 2x2x6 | 2 | 0 | - |
| Convolution | 1600 | 5x5x16 | 1 | 0 | Sigmoid |
| Pooling | 400 | 2x2x16 | 2 | 0 | - |
| Fully connected | 120 | - | - | - | Sigmoid |
| Fully connected | 84 | - | - | - | Sigmoid |
| Fully connected | 10 | - | - | - | Sigmoid |

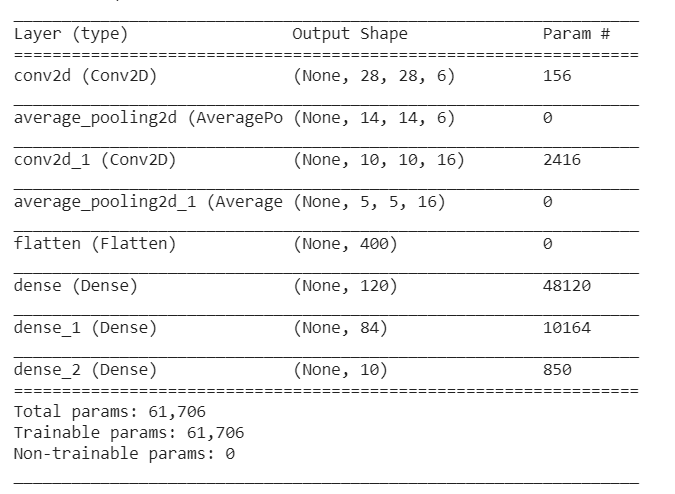
**Total number of neurons: 8094, so we can take the group size as 100.**

LeNet model in Keras is present here: <https://github.com/Bhagya-laxmi/Noc-based-CNN/tree/master/LeNet_arch>

The weights are present in the file Lenet\_weights.txt where the weights are organized as bias and kernel weights per kernel.

This github repo: <https://github.com/Bhagya-laxmi/Noc-based-CNN> has details about the ground work needed to start the implementation.

This github repo: <https://github.com/Bhagya-laxmi/nn_noxim> will contain the implementation from here on.

The Model summary: 

The dir\_x mapping will be used with XYX and YXY routing to balance the traffic across the interconnection.

The layer-wise grouping is shown here:

FC

FC

Average Pool

Convolution

Average Pool

Convolution

Gp 83

Gp 84

Gp 82

Gp 80

Gp 81

Gp 79

Gp 64

Gp 78

Gp 63

Gp 62

Gp 61

Gp 49

Gp 50

Gp 51

Gp 48

Gp 3

Gp 2

Gp 1

Gp 0

FC- Output

Gp 85