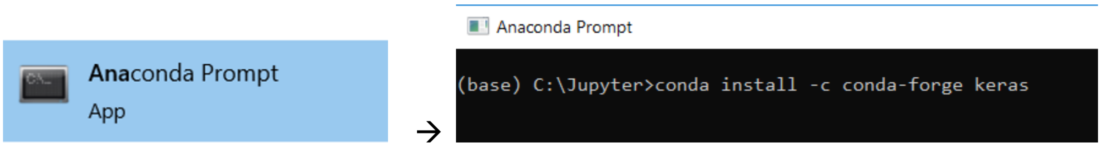
Python based GUI for hardware code generation

Software requirements and setting up the environment:

1. Install Anaconda Python: Anaconda can be downloaded from https://www.anaconda.com/distribution/#download-section

2. Install Keras: Once Anaconda is installed, Keras framework can be installed through Anaconda. Open Anaconda prompt → enter the command “conda install –c conda-forge keras”



```
(base) C:\Jupyter>conda install -c conda-forge keras
```

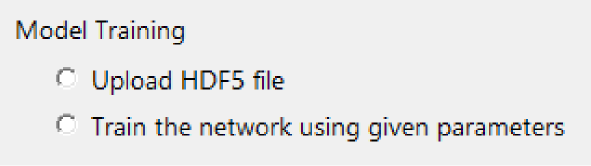
3. Install Xilinx Vivado: Xilinix Vivado will be required for the simulation and synthesis of verilog code with tcl script (This functionality is not included in the GUI yet).

The filename of the main script to be executed is ‘gui.py’.

Note: All the file templates and python scripts need to be kept in the same folder where the main script named ‘gui.py’ is kept.

Training a model and generate trained parameters:

There are two options to obtain trained parameters for the model. First option is to directly upload the HDFS file which contains the model parameters and the other option is to generate the HDFS file by training the model.

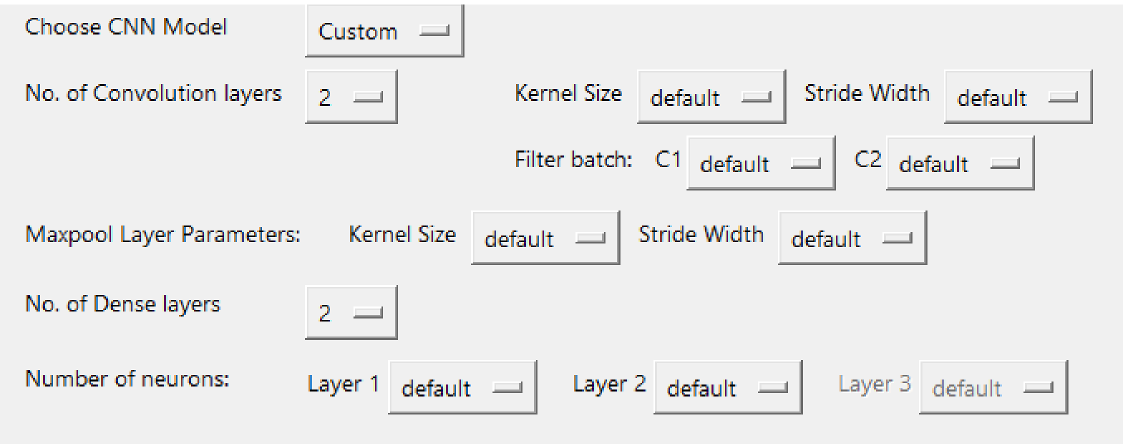


Model Training

○ Upload HDF5 file

○ Train the network using given parameters

In the second option, user will have options to choose a pre-designed model or a custom model. If the default model is chosen, it will select the optimized LeNet-1 model with kernel size 3x3. The options to customize the design will be enabled if the user chooses custom model. The parameters that the user can choose includes the number of convolution layers, convolution kernel size, stride width, filter batch size for each convolution layer, maxpool kernel size, stride width, number of dense layers and number of neurons in each dense layer.



Choose CNN Model: Custom

No. of Convolution layers: 2

Kernel Size: default

Stride Width: default

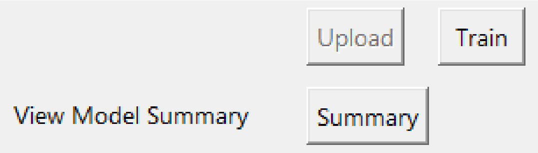
Filter batch: C1 default, C2 default

Maxpool Layer Parameters: Kernel Size default, Stride Width default

No. of Dense layers: 2

Number of neurons: Layer 1 default, Layer 2 default, Layer 3 default

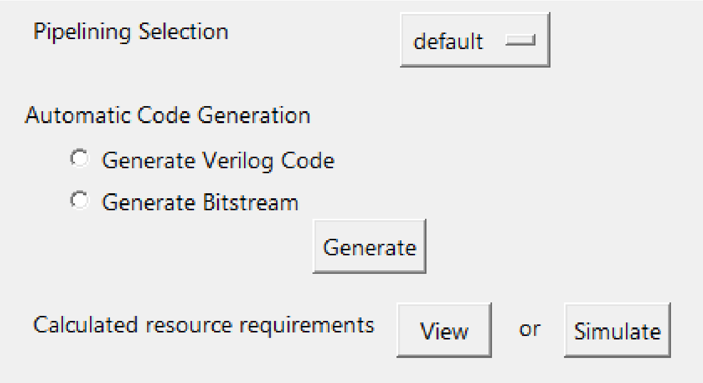
Model will be trained and will generate the HDFS file after clicking the 'Train' button. Once the model is trained, the model summary can be viewed by clicking the model summary button.



Hardware Code generation:

For hardware code generation, the pipelining option can be chosen. The default option will be the fully pipelined version.

For code generation, either the Verilog code or the bit-stream can be generated by choosing the respective options. The generate button will start the code generation. The resource requirements of the chosen model can be viewed by clicking the view button if user had chosen a predefined model or if the resource requirements are already available for the user defined custom model. If user opts for a custom model of which the resource requirements are not available, the simulate button will simulate the Verilog code generated with a tcl file and will show the resource requirements (This feature is not available at the moment).



Pipelining Selection: default

Automatic Code Generation

○ Generate Verilog Code

○ Generate Bitstream