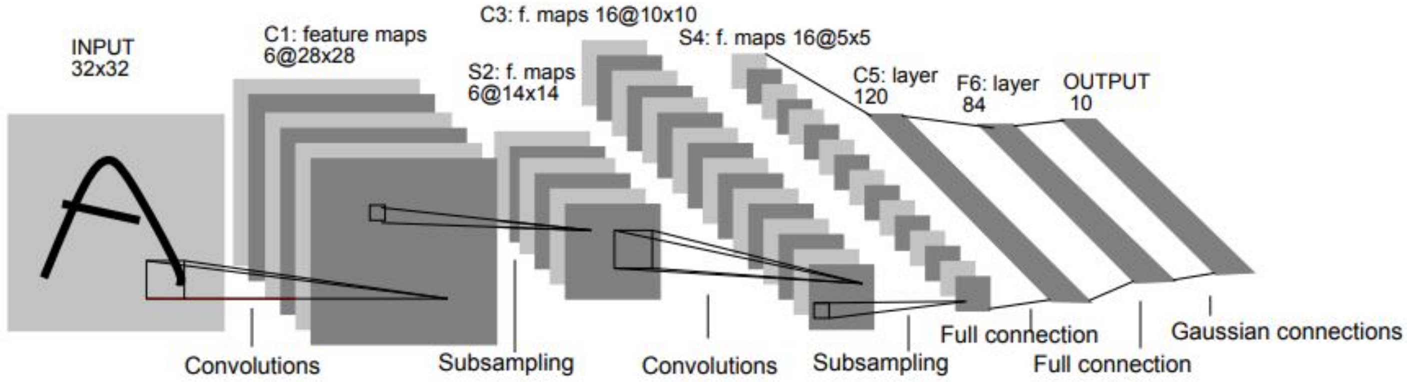
HW1: LeNet-5 with Post-training Quantization

LeNet is considered to be the first ConvNet.
Before we start, you may check this Tensorspace-LeNet to play with LeNet and get familiar with this neural architecture.



Ref.: LeCun et al., Gradient-Based Learning Applied to Document Recognition, 1998a

We are going to implement a neural architecture similar to LeNet-5 and train it with MNIST dataset.

Ref.: MNIST database from Wikipedia

After that, we will go through several steps to do Post-training Quantization, including

- Quantizing Weights
- Quantizing Activations

- Quantizing Biases

Action Items:

- □ Learn how to use Jupyter Notebook and write python code.
- □ Fill in all TODOs in `homework1.ipynb` and `nnutils/quantutils.py`.
- □ Answer all questions in `homework1.ipynb`

How to launch Jupyter Notebook?

You should choose either option 1 or option 2. If you know very well how to use Jupyter Notebook, you can just launch `homework1.ipynb` and start writing your homework. Make sure you have placed `homework1.ipynb` and `nnutils` folder in the same directory.

Option 1: with Google Colaboratory on the Cloud

1. Open your [Colab]
2. Unzip hw1.zip and upload homework1.ipynb to Colab.
3. Upload external `nnutils` folder.

   ```
   from google.colab import files

   uploaded = files.upload()

   for fn in uploaded.keys():
     print('User uploaded file "{name}" with length {length} bytes'.format(
         name=fn, length=len(uploaded[fn])))
   ```

   - Or you can mount `nnutils` via Google drive. Check this link for detailed information.
4. It may warn the missing package of `torchinfo`.
   - Run `!pip install torchinfo` in Colab before using it.
5. If you train the neural network from scratch, you should enable GPUs for the notebook:
   - Navigate to Edit→Notebook Settings
   - Select GPU from the Hardware Accelerator drop-down menu

- We don't want to install `torchinfo` again or upload any file when checking your homework. Comment out all comments you use in Step 3 and Step 4 before submitting your homework.

Option 2: with Conda on your computer

1. Install miniconda
2. Create a Conda virtual environment

   ```
   conda create --name vlsi
   conda activate vlsi
   ```

3. Install PyTorch
   - Check the official website and follow the procedures suitable for your computer.
4. Install the following packages for this homework

   ```
   conda install -c conda-forge matplotlib
   conda install -c anaconda jupyter
   conda install -c conda-forge torchinfo
   ```

5. Type `jupyter notebook` and launch Jupyter Notebook!

What do I need to submit?

1. Make sure you have done everything in `quantutils.py` under `nnutils` folder and `homework1.ipynb`.

   ```
   \nnutils
       \quantutils.py
       \...
   \homework1.ipynb
   ```

2. We don't want to install `torchinfo` again, upload/download any file to Colab, or retrain any models again when checking your homework. Comment out those lines of code for those processes!

   ```
   # from google.colab import files
   # uploaded = files.upload()

   # for fn in uploaded.keys():
   #     print('User uploaded file "{name}" with length {length} bytes'.format(
   #         name=fn, length=len(uploaded[fn])))
   ...

   # files.download(...)
   ...

   # train(net, trainloader, device)
   ...
   ```

```
# !pip install torchinfo
```

3. Uncomment those lines of code for loading the model in `homework1.ipynb`.

```
net.load_state_dict(torch.load('lenet.pt'))
...

net_with_bias.load_state_dict(torch.load('lenet_with_bias.pt'))
...
```

4. Click `Kernel` and then click `Restart Kernel & Run All` on the Jupyter Notebook of `homework1.ipynb`.
    - Make sure everything goes smoothly without any warining or error messages while running your `homework1.ipynb`!
5. Upload `quantutils.py`, `homework1.ipynb`, `lenet.pt`, and `lenet_with_bias.pt` to EECLASS. Do not zip these files or put them in a folder! Just upload these four separate files.

Troubleshooting

Reloading modules

You might need to run and modify `quantutils.py` and `homework1.ipynb` back and forth. If you have edited the module source file using an external editor and want to try out the new version without leaving the Python interpreter, you shoule reload these modules.

There are two alternatives:

- Autoreload
    - IPython extension to reload modules before executing user code. `autoreload` reloads modules automatically before entering the execution of code typed at the IPython prompt.

```
%load_ext autoreload
%autoreload 2
```

    - Check this link.
- Reload
    - reload() reloads a previously imported module.

```
import importlib
importlib.reload(module)
```

    - Check this link.

Sometimes there may be the following error message:

```
super(type, obj): obj must be an instance or subtype of type
```

The straightforward solution is to restart the kernel and run it all.

- Check this link.