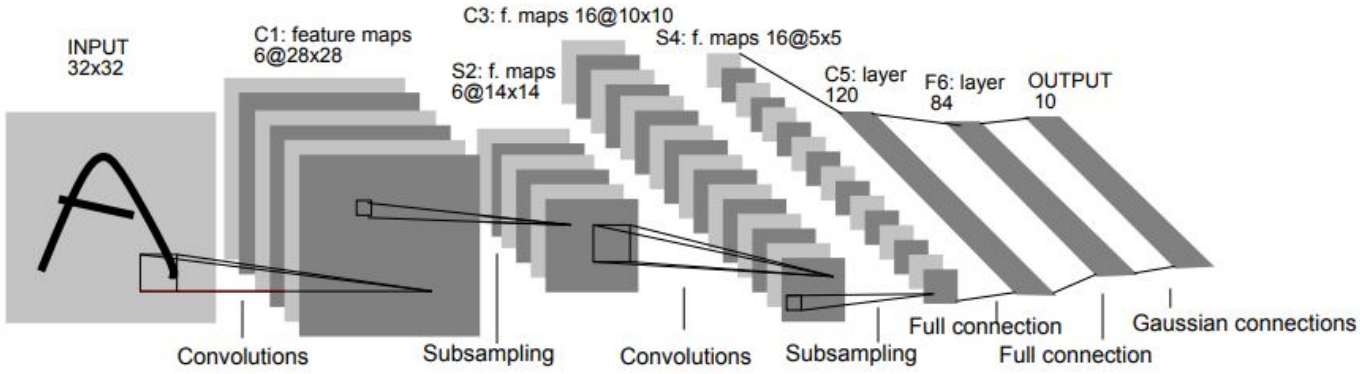
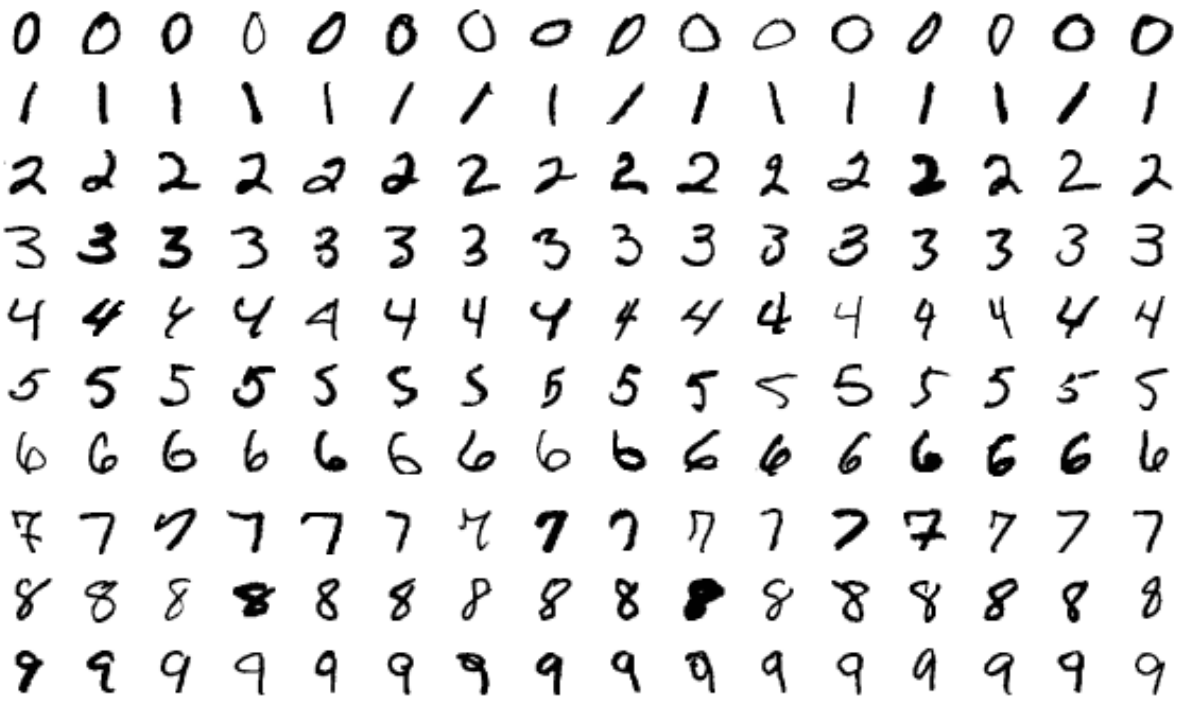
HW1: LeNet-5 Quantization and Inference

LeNet is considered to be the first Convolutional Neural Network (CNN).
Before we start, you can check Tensorspace-LeNet to play with LeNet and get familiar with this neural network architecture.



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

We will implement a neural network architecture similar to LeNet-5 and train it with the MNIST dataset.



Ref.: MNIST database from Wikipedia

Part 1: Post-training Quantization and Quantization Aware Training

After that, we will go through several steps for Post-training Quantization (PTQ), including

- Quantizing Weights
- Quantizing Activations
- Quantizing Biases

For Quantization-aware Training (QAT), you need to trace the code and answer the questions. The code for QAT is already implemented in `homework1-1.ipynb`.

Action Items:

- [ ] Learn how to use Jupyter Notebook and write Python code.
- [ ] Fill in all TODOs in `homework1-1.ipynb` and `quantutils.py`.
- [ ] Answer all questions in `homework1-1.ipynb` and write down your answers in the `studentID_hw1_report`.

Part 2: Model Inference of the QAT LeNet

We have trained a CNN model similar to LeNet with QAT.

In this part, we will implement the functional inference model in Python, a high-level programming language, and adjust the bit width of the partial sums in each layer.

Action Items:

- [ ] Implement a high-level functional model for each layer of the CNN, including convolution, pooling, and fully-connected layers with 8-bit quantization of the input activations, output activations, and weights accordingly.
- [ ] Pass all unit tests of `OpTestCase`.
- [ ] Fill in all TODOs in `homework1-2.ipynb` and `functional.py`.
- [ ] Answer all questions in `homework1-2.ipynb` and write the answers in the `studentID_hw1_report.pdf`.

How to launch Jupyter Notebook?

You should choose either Option 1 or Option 2. If you are familiar with Jupyter Notebook, you can just launch `homework1.ipynb` and start writing your homework.

Option 1: Using Google Colaboratory on the Cloud

1. Open your Colab
2. Upload `homework1-1.ipynb` and `homework1-2.ipynb` into Colab.

```
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])))
```

3. There may be warnings for missing packages of `torchinfo` in Part1 or `numba` in Part2.
    - Run `!pip install torchinfo` in Colab before using it.
    - Run `!pip install numba==0.55.1` in Colab before using it.
4. 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

- When checking your homework, we will not install `torchinfo` and `numba` again or upload any additional files. Before submitting your homework, comment out all the comments you used in Step 3.

Option 2: Using 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

Part1 (`homework1-1.ipynb`)

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

Part2 (`homework1-2.ipynb`)

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

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

What do I need to submit?

Please follow the rule to upload the following files to EECLASS, otherwise you will get a 20-point penalty.

1. Make sure you have done everything in `homework1-1.ipynb`, `quantutils.py`, `homework1-2.ipynb`, and `functional.py`.
2. We don't want to install `Numba` again, upload/download any files 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(...)
...

# !pip install numba
```

3. Click `Kernel` and then click `Restart Kernel & Run All` on the Jupyter Notebook of `homework1-2.ipynb`.
    - Make sure everything goes smoothly without any warning or error messages while running your `homework1-2.ipynb`!
4. Upload `quantutils.py`, `parameter.zip`, `homework1-2.ipynb`, `functional.py`, and `studentID_hw1_report` to EECLASS. Do not zip these files or put them in a folder! Just upload these five separate files.

Troubleshooting

Reloading modules

You might need to run and modify `homework1-1.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 should 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
```

  - Please refer to the link.
- Reload
  - reload() reloads a previously imported module.

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

  - Please refer to the link.

You may occasionally encounter 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.

- Please refer to the link.