Weekly Overview

- Goals: Running Spiking YOLO v3 and deepening my understanding of the ANN to SNN conversion code in SNN Tool Box

Difficulties

- Ran into some numpy errors on Spiking YOLO and would only see if my solutions worked after the code ran for an hour
- SNN Tool Box appears to have some code removed from the GitHub
Spiking YOLO v3 Progress

- Training on the COCO Dataset
  - 82,783 Labelled Training Images
  - 40,504 Labelled Validation Images
- Training at roughly 30 epochs per day, 300 epochs will complete around 6/27
  - 10 GB VRAM
Pipeline

1. Initialize an empty SNN
2. Load the testing data used for the ANN
   a. No modifications are made to data yet
3. Load the ANN
4. Parse the ANN
   a. Converts the ANN from supported frameworks (Keras, Pytorch, Lasagne, Caffe) to Keras ANN
5. EvaluatesParsed ANN
6. Converts Parsed ANN to SNN
   a. Build to Empty SNN using Parsed ANN as input
Build

Setup Layers

1. Defines Input Shape
2. Iterates through each layer and adds converted equivalent:

**Build Dense / Conv**

- **Input:** Dense / Conv2D Layer
- **Operation:**
  - Carries over all weights
  - Creates new biases

\[
NewBias = \frac{OldBias \times cm \times (thresh - rest)}{duration}
\]

Statistically equivalent to ReLU when averaged over all timesteps

- **cm:** Membrane Capacitance
  - Default: 1
- **thresh**
  - Default: 1
- **rest**
  - Default: 0
- **duration:** Length of spike train
  - Default: 200
Build cont’d

**Build Average Pool**

- Input: Average Pooling Layer
- Output:
  - Connections: A list where each entry is a tuple containing the source neuron index, the target neuron index, and weights which are equal to each neuron divided by the W x H of the pool

**Build Max Pool**

- Hidden, but likely similar logic to Avg Pool

**Build Flatten**

- Hidden, but likely uses the same connections list since a flattening doesn’t do any numerical operations
Pipeline

1. Initialize an empty SNN
2. Load the testing data used for the ANN
   a. No modifications are made to data yet
3. Load the ANN
4. Parse the ANN
   a. Converts the ANN from supported frameworks (Keras, Pytorch, Lasagne, Caffe) to Keras ANN
5. Evaluates Parsed ANN
6. Converts Parsed ANN to SNN
   a. Build to Empty SNN using Parsed ANN as input
   b. Compile SNN
Compile

1. Removes Counter from layers that Tensorflow added
2. Multiplies all biases by timestep value
3. Compiles:
   a. SGD
   b. Categorical Cross Entropy
1. Initialize an empty SNN
2. Load the testing data used for the ANN
   a. No modifications are made to data yet
3. Load the ANN
4. Parse the ANN
   a. Converts the ANN from supported frameworks (Keras, Pytorch, Lasagne, Caffe) to Keras ANN
5. Evaluates Parsed ANN
6. Converts Parsed ANN to SNN
   a. Build to Empty SNN using Parsed ANN as input
   b. Compile SNN
7. Run Simulation
   a. Takes Sample from testing data
   b. Sorts into batches
   c. Collects performance metrics
   d. ***Unable to locate where / when testing data is converted into spike trains
What’s Next For Me?

● Completing the Spiking YOLO v3 Exercise
  ○ If possible find a pre-trained model since one week is too long
  ○ Find code for how an image is encoded into spike trains

● Examining Attention Models for potential conversion