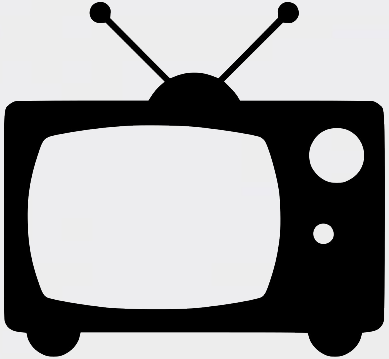
IoT Neural Networks:
Linear Integrate & Fire

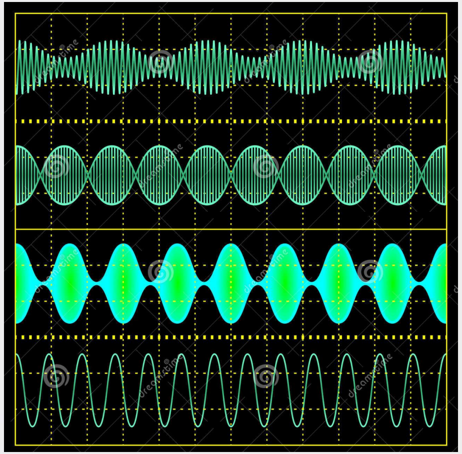
Simon Kaufmann, Owen Jow

Background

Radio Communication



Modulation

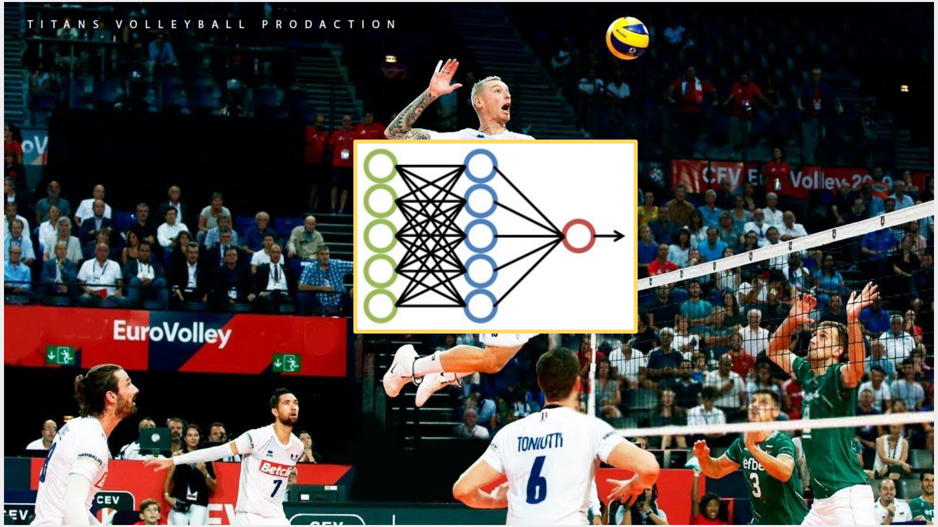


Goals

1. Train a Spiking Neural Network
2. Quantize the Spiking Neural Network

SNNs for Modulation Classification

- We want to use spiking neural networks (SNNs) to classify radio signals.
- We will train and validate the SNNs on the RadioML dataset.
    - RadioML dataset: a collection of signals and their associated classes.

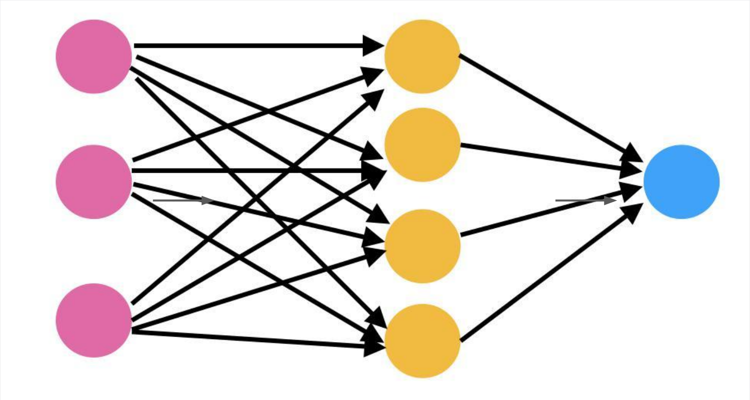


But what is a spiking neural network?
And how do you train one?
And how are the signals represented?

Stay tuned...

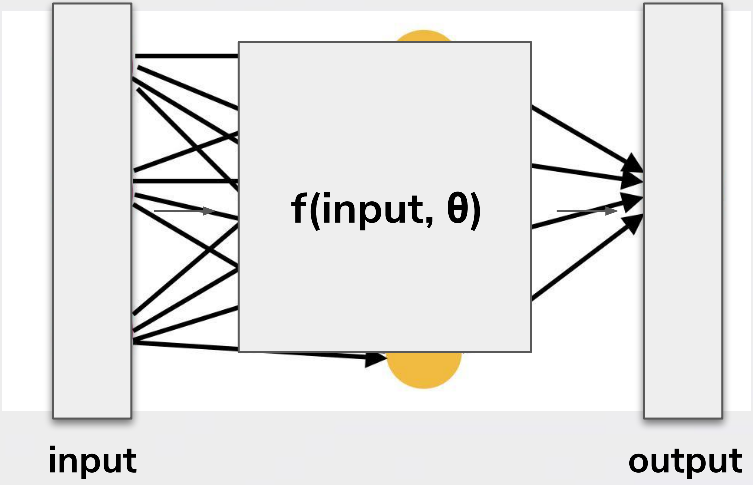
Neural Networks

A neural network is a parameterized function approximator.



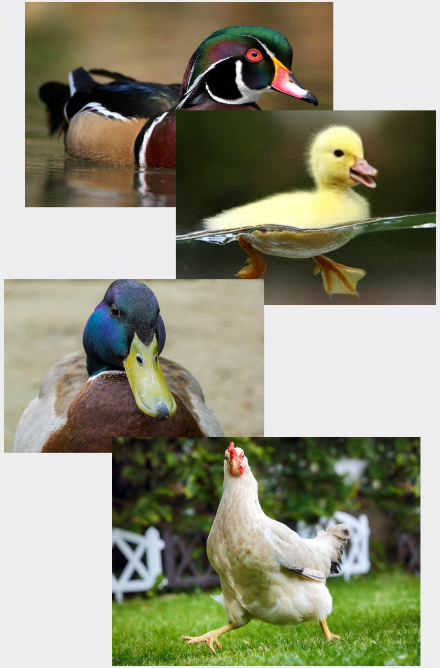
Neural Networks

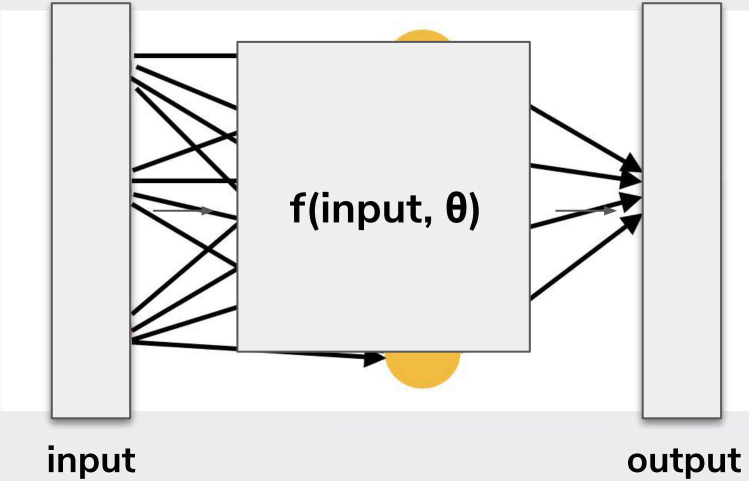
A neural network is a parameterized function approximator.



Neural Networks

In the simplest sense, a network learns by observing lots of input/output examples.





DUCK

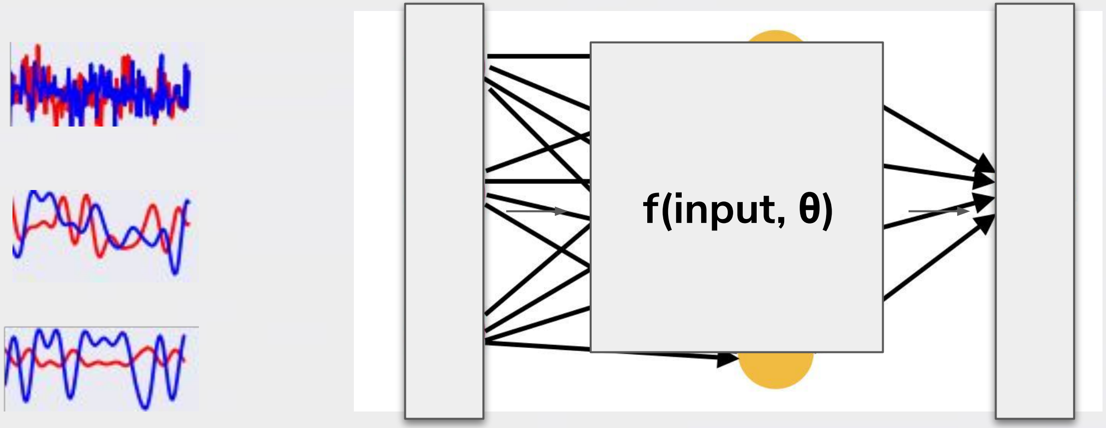
DUCK

DUCK

NOT A DUCK

Neural Networks

Previous work has achieved ~95% accuracy on the RadioML task using neural networks.



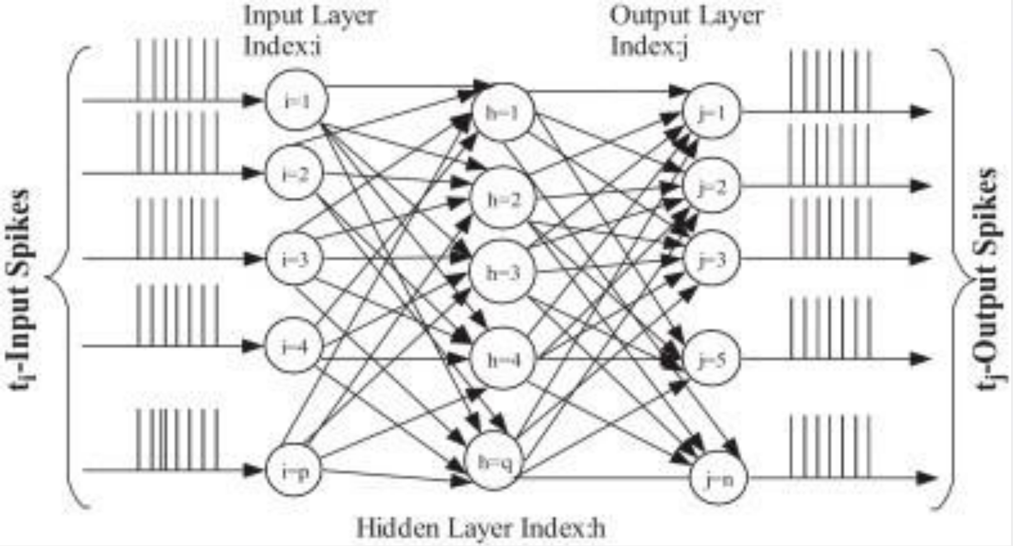
Neural Networks

Note: traditional neural networks need gradients for training!

$$\nabla f$$

Spiking Neural Networks

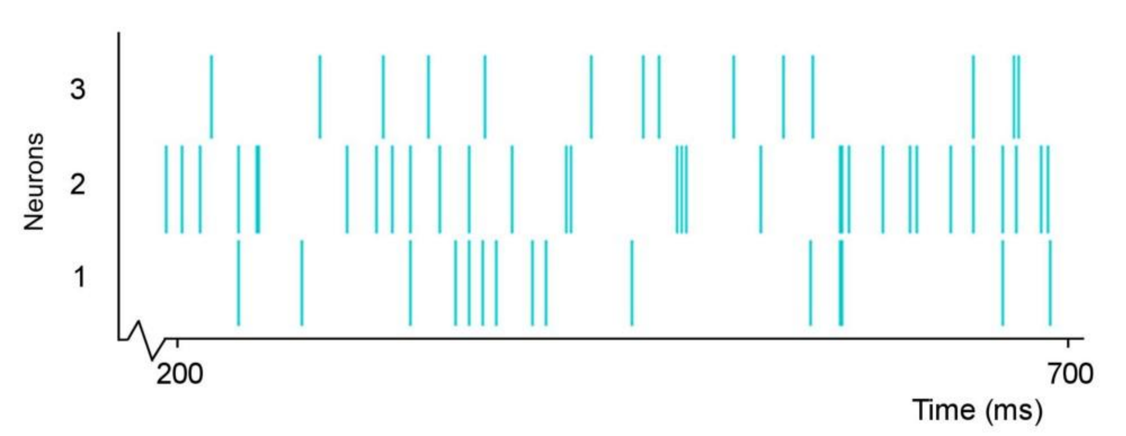
Spiking neural networks (SNNs) take and produce **spike trains** instead of continuous numbers. A spike train is a sequence of "all of nothing" (binary) events over time.



Spiking Neural Networks

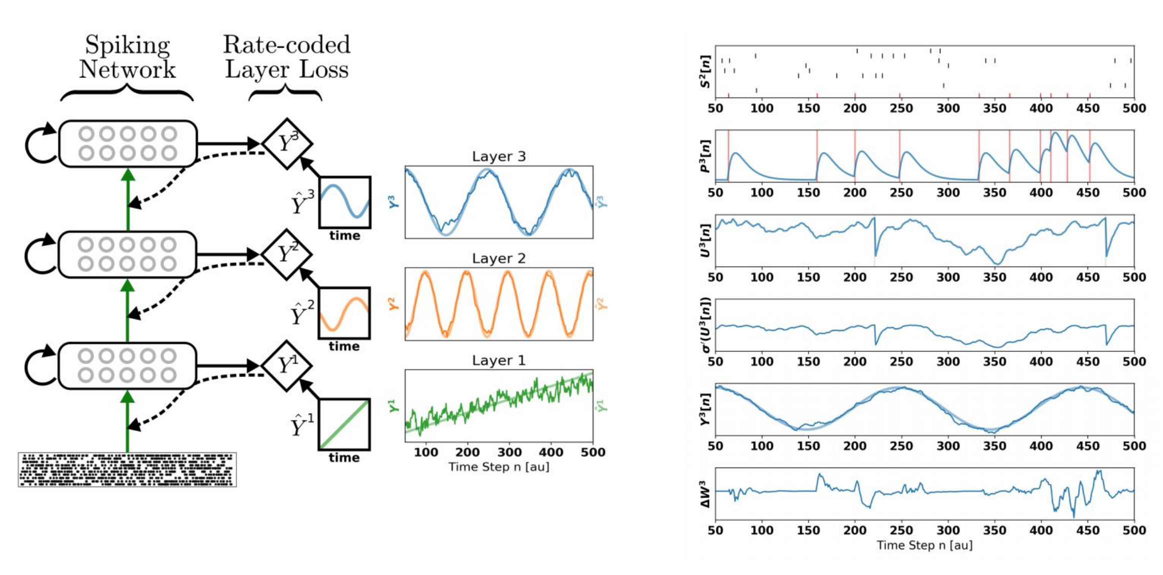
Advantages: less computation/power requirements on specialized hardware (only need to process spikes), models temporal data, theoretically more powerful neuron models…

Disadvantages: can't take gradients of spikes → can't train in the same way as before!



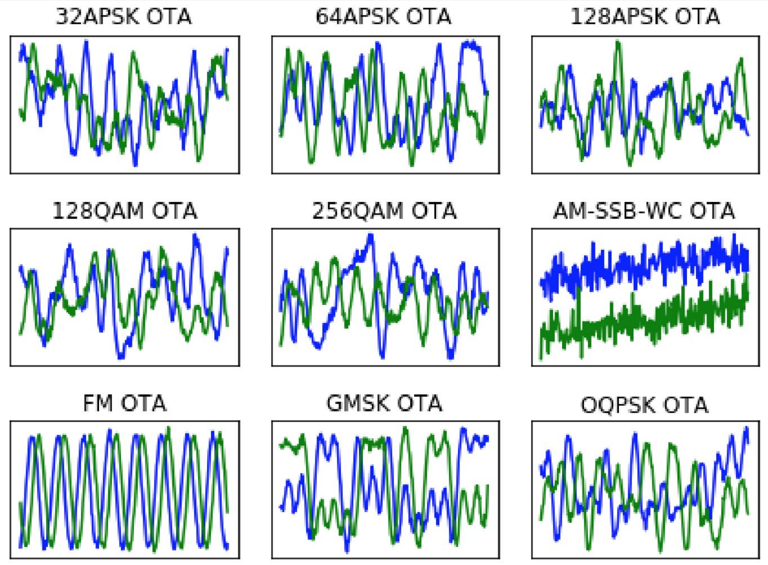
DCLL

We will follow the training structure outlined in the **Deep Continuous Local Learning (DCLL)** paper, where each layer is trained separately with local classification gradients.



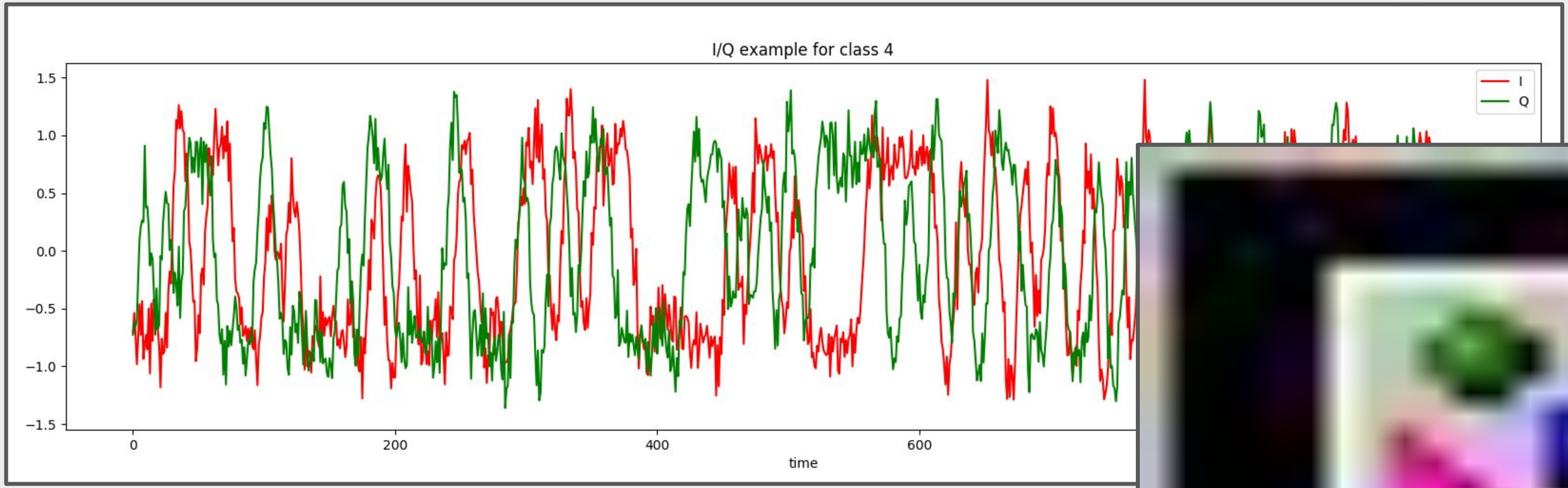
SNNs for Modulation Classification

- We want to identify the modulation classes for different signals.
- RadioML represents each signal as I/Q data (real/imaginary signal components).



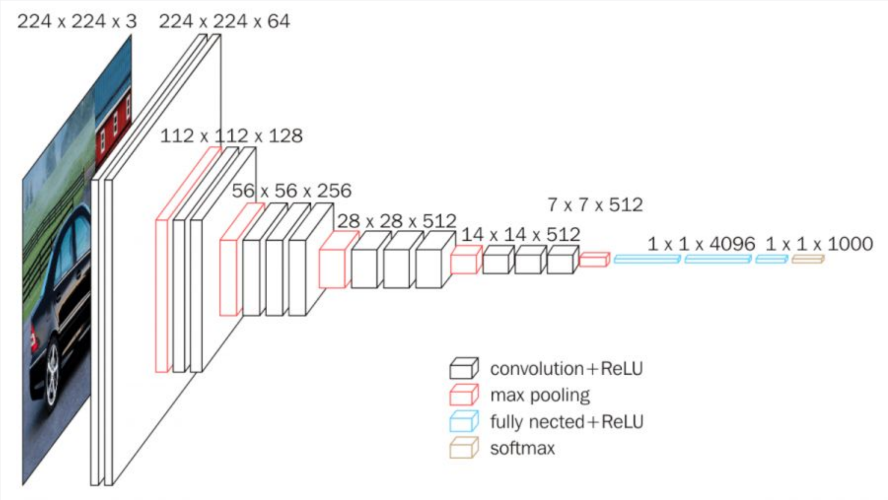
SNNs for Modulation Classification

- For use with an SNN, we convert each signal to a series of events in the complex plane (i.e. discretize each into a 2D image, with I and Q axes).



SNNs for Modulation Classification

- Currently, we're using a convolutional network architecture based on VGG-10.
  - Except now each layer takes and produces spikes,
    and maintains the requisite state to do so.



Progress - SNN

Set up a first RadioML network:

- Cleaned up DCLL library
- Prepared RadioML data for SNN
- Modified network architecture to align with VGG model
- Trained and evaluated the network on RadioML data

Result: 40% accuracy (requires further optimization)

Next: Parameter tuning, experimentation with network architectures

Quantization

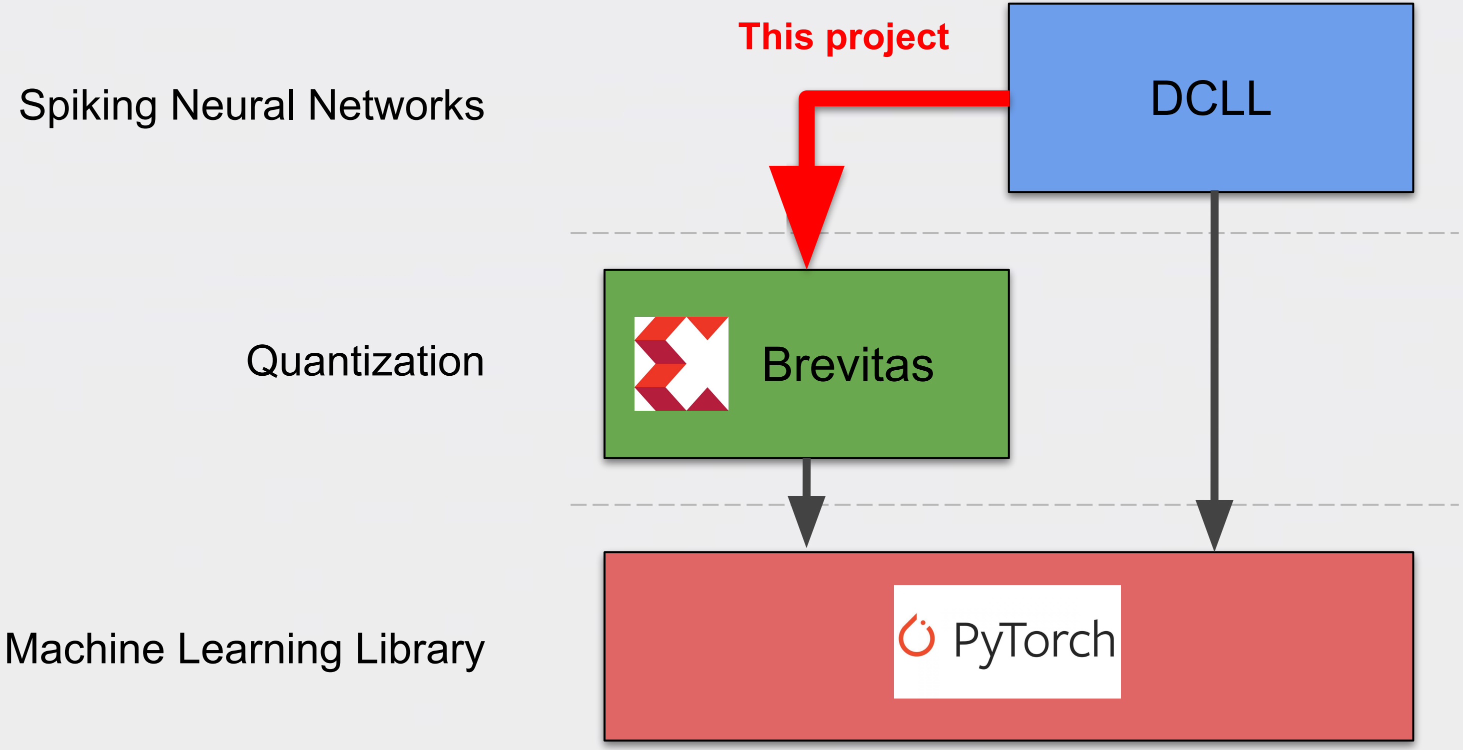
What? Reduce bits for number representation

Why? Save memory

Faster computation

How? Brevitas - Library for quantization aware training

Quantization



Progress - Quantization

Progress Quantization:

- [x] Weights
- [ ] Neuron State
- [ ] Other parameters

Results: MNIST Network

Regular:	98.8%
8-bit:	97.6%
4-bit:	96.5%

Next: Experiment with further quantization; adapt for RadioML network

Conclusion

So far:

- Trained an SNN to classify modulations of signals.
- Quantized weights of simple SNN.

Next steps:

- Tune hyperparameters to achieve better performance.
- Quantize remaining parts of the SNN and apply this to RadioML as well.