Anomaly Detection with Spiking Neural Networks

- Eric Moreno (Caltech, MIT)

Mentors: Maurizio Pierini (CERN), Jean-Roch Vlimant (Caltech)

*October 27th, 2021*
Computers vs the Brain

- Neural networks trained with deep learning are state-of-the-art in AI/ML
- Training GPT-3: $12 million USD
- Training our brains:

  \[
  8.6 \times 10^{10} \text{ synapses} \times 7 \times 10^3 \text{ synapses} \times 10^3 \text{ precision} = 600 \text{ PetaFlops}
  \]

- World's most power supercomputer can run 442 PetaFlops at 30 MW!
- The brain only needs 12 W to run
- Best of both worlds: Biological neural nets + backprop

Tutorial on snnTorch: Jason Eshraghian ICONS 2021
Dense Neural Networks (DNNs)

- Most common type of artificial neural network
- Contains input layer, hidden layer(s), and a final output layer
- Able to learn complex non-linear relationships through Stochastic Gradient Descent (SGD)
- Loosely modeled after the neurons in biological brains
Dense Neural Networks (DNNs)

- Nodes receive input and combine with their internal state using an activation function
- Nodes receive outputs from predecessor nodes and perform a weighted sum
- Only compute as feedforward networks

Is this how biological neurons function?

Is this how biological neurons function?
Biological Neurons

- Connected by synapses (as in DNN)
- Single neurons can have variable electrochemical potential
- Fires all-or-nothing electrochemical (delta function) pulse after reaching a certain potential
Spiking Neural Networks (SNN)

- Utilize membrane potentials to mimic the behavior of organic information processing systems like the human brain.
- Very similar to DNN structure but replaces weighted sums with asynchronous spikes activated at a certain threshold.
- This architecture computes using asynchronous spikes (delta functions) that signal the occurrence of an event.

Membrane Potential

Membrane potential of a single spiking neuron. The neuron is firing when the action potential achieves a specific threshold. After activation the signal returns to the low, regular value.

Time-Series for SNNs

- **SNNs are perfect for time-series data** as they are inherently spatio-temporal.
- Spike timing and spike rates are able to encode and transmit information through the network.
- Spiking computation is event-driven:
  - When there is little to no information recorded, the SNN does not compute much.
  - When bursts of activity are recorded, thresholds are met and spikes are propagated.

Benefits of Spiking Networks

- Analog Computation
- Low power consumption
- Fast Inference
- Massive parallelism
- Excels with time-series data
Problems with Spiking Networks

- Experimental packages for SNNs
- Experimental Chipsets
- Non-differentiable
- Lack of training algorithms
- Currently don’t outperform DNNs
Project Goal

Bring SNN architectures to HEP applications
Our testing-grounds

- Detection of gravitational waves (GWs) at LIGO

Source: Elena Cuoco - Real Time Classifier for transient signals in Gravitational Waves, From raw data to classified triggers

Produces: 1-D time-series strain
Data is whitened and bandpassed

Source: https://www.gw-openscience.org/data/
1. Simulates typical detector noise conditions
2. Simulates GW waveforms for the following conditions:
   • Binary masses of 10-80 M☉
   • SNR of 5-20
   • Variable angles in the sky
3. Adds GW strain into noise for signal events
Currently used methods

**Matched Filtering**

- **Current method** used by LIGO
- Compares incoming GW data to bank of simulated waveforms
- Can only identify GWs that are available in GW banks (no exotic events)

**Deep Filtering**

- Convolutional Neural Networks (CNNs)
- Take time-series inputs, can determine detections and estimate parameters of events
- Still can miss events that aren’t included in training set

Unsupervised Learning: Autoencoder

• Encoders and decoders made of:
  • Dense Neural Networks
  • Recurrent Neural Networks (RNNs) such as LSTMs or GRUs which are good with dealing with time-dependent data
  • Convolutional layers
  • Spiking Neural Networks?!
Anomaly detection sequence:
1. Train autoencoder to encoder and decode data on data with no anomalies.
2. Compute the highest loss on the training dataset – set as threshold for anomalous detection
3. Run autoencoder for test data, identify events that fall above detection threshold
• SNUs enable modelling of SNNs in deep learning frameworks and training with backprop-through-time (BPTT)
• Resembles a recurrent architecture with a memory cell
SNN-AE Input

• Input must be spikes!

• Packages Include:
  • SNNtorch (recommended)
  • NeuroAIKit (recommended)
  • Nengo (harder)
  • SNNToolbox (useless)
ANNs: Unsupervised learning

- Unsupervised approaches have lower accuracy (no labels) but should have higher chance of detecting anomalous events
Future opportunities for SNNs

• SNNs live in the time domain
• Offer low-power consumption and fast inference on neuromorphic hardware
• Highly parallelizable
• Currently could serve as L1 triggers for LHC time-series data
• Future advances in the field of SNNs (backprop algorithms) could yield better performance than DNNs on specific time-series tasks
Thank you for your attention!

Questions?
Nengo Loihi – activation functions
Nengo Loihi – output

Potential of the output neurons:

Binary spike (classification):
SNNs supervised learning – SNN-TB

Accuracy results in simulation:
ANN: 94%
SNN: 91%

Executed on Loihi chip
Work under development: LMUs

- Type of RNN that provides a continuous time scale-invariant memory (SNN's LSTM analog)
- Works by orthogonalizing a time history by solving ODEs using the Legendre polynomials
- Allows the LMU to be resource efficient within a window of time
- Spiking neurons implement the hidden state of the LMU by nonlinearly encoding the memory vector

Work under development: LMUs

- Can maintain memory over much larger timesteps than an LSTM (around $10^8$ steps)
- Achieves SOTA performance on permuted sequential MNIST (better than LSTMs)
- Most difficult sequences to remember are noise signals which require $\mathcal{O}(d)$ dimensions to maintain window of $d$ timesteps