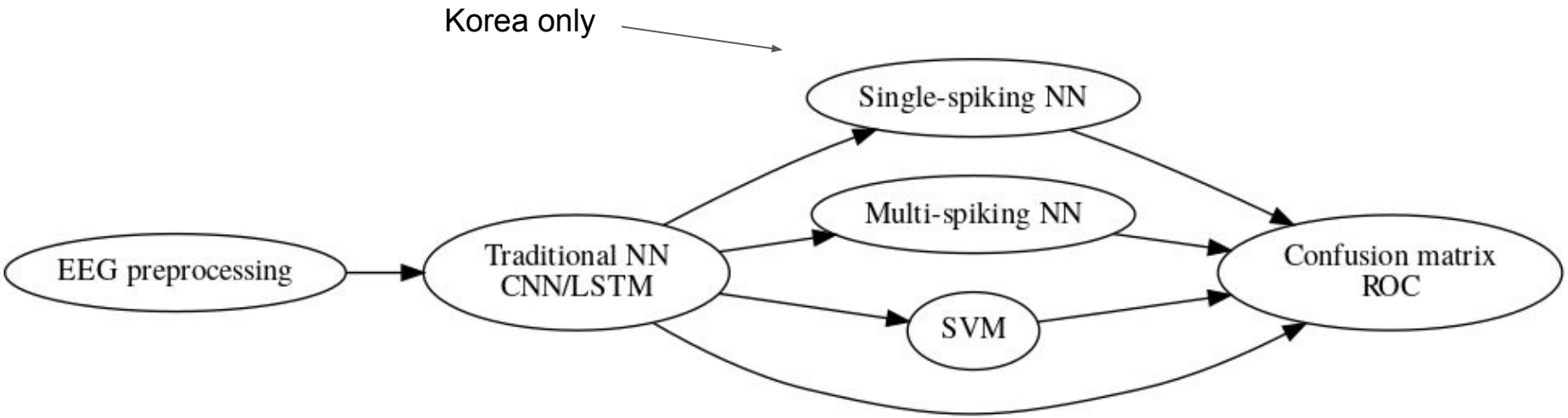
Predicting seizures with artificial spiking neural networks

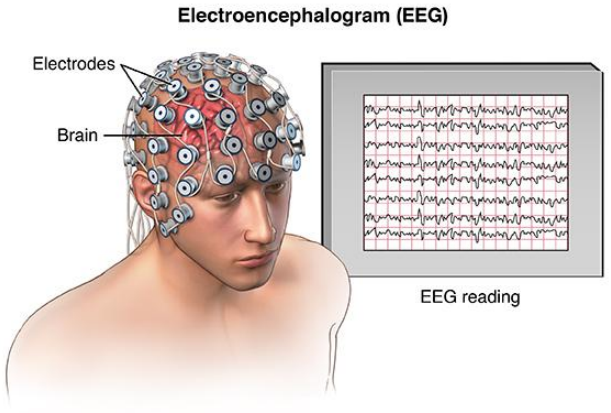
Jeremy Angel, Matthew Wootten, Junwon Kim, and Jimin Chae



Our data processing pipeline

EEG Preprocessing

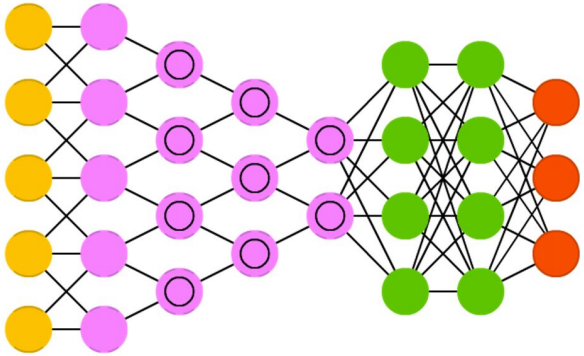
- Reading the raw EEGs
- Picking positive and negative samples
- Picking out features we know will be important



```
def usecsToBytes(usecs):
    return 4 * int(round(usecs / eegLengthUsecs * channelFileLengthDwords))

usecs = 1
seconds = 1000000 * usecs
minutes = 60 * seconds
# Time before the seizure's start when the sample starts
sampleStartOffset = 10 * minutes
# Duration of the sample
sampleDuration = 10 * seconds
sampleDurationBytes = usecsToBytes(sampleDuration)

f = open(channelFile, 'rb')

for (num, seizureStartUsecs) in enumerate(seizureStartsUsecs):
    sampleStartUsecs = seizureStartUsecs - sampleStartOffset
    sampleStartBytes = usecsToBytes(sampleStartUsecs)

    f.seek(sampleStartBytes)
    sample = f.read(sampleDurationBytes)
    outputName = channelFile.split('.')[0] + "-positive-" + str(num) + ".raw32"
    g = open(outputName, 'wb')
    g.write(sample)
    g.close()

f.close()
```

Traditional NN

- Does more preprocessing
- Trains more quickly, so we can iterate faster
- Provides input for the subsequent steps



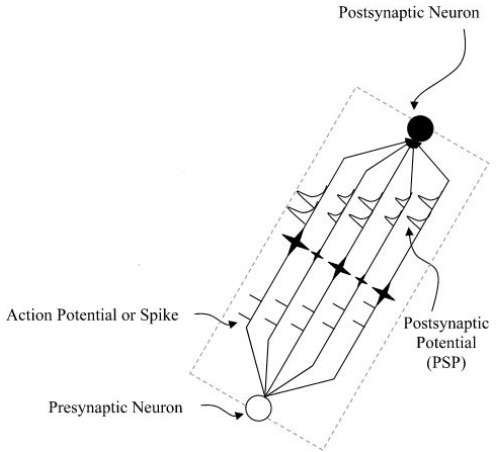
Right now, we have a basic convolutional network, using PyTorch (not pictured because it’s pretty long).

This use of traditional networks was recommended by Dr. Turaga at HHMI

Multi-spiking NN

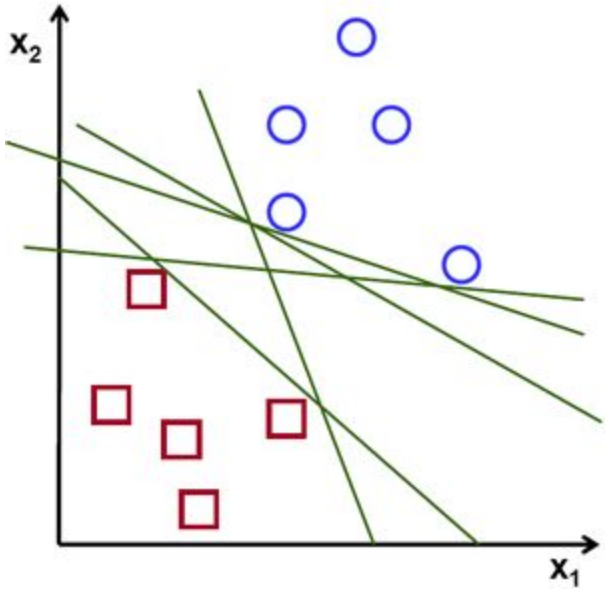
- What we have now: a replication
- What we are working on: a different form of the replication that batches together much of the arithmetic
- What we hope to end up with: a network that trains much more quickly (a couple orders of magnitude more quickly)

Dr. Turaga at HHMI gave us some ideas for the more efficient form of the replication



SVMs

- Classify something by just drawing a plane between the examples and non-examples
- The inputs will be the outputs of the traditional neural net
- Currently, we haven't made progress on this control.



This control was recommended by Dr. Scheffer at HHMI.

Post-processing

Collect the data into something that looks like this:

Confusion Matrix		Actual	
		Seizure	Not seizure
Predicted	Seizure		
	Not seizure		

Final metrics

AUROC: area under the receiver operating characteristic curve

This is a measure of how good predictions are over a variety of thresholds

- ROC curve: a plot of sensitivity vs. specificity
  - Sensitivity: correctly predicted positives as a proportion of all real positives
  - Specificity: correctly predicted negatives as a proportion of all real negatives

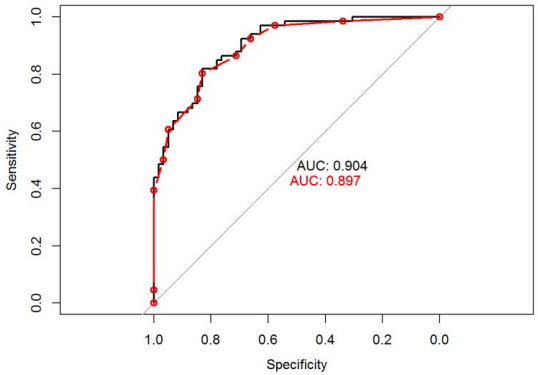


Image by Bob Horton

These statistical metrics were recommended by Dr. Oguz at NIH

Progress up to Progress Report 1

- We have a single multi-spiking neuron that can take inputs and receive outputs in isolation.
- We have an initial more advanced preprocessing step based on EEG signal steepness instead of signal height
- Discussion with Dr. Scheffer

Progress After Progress Report 1

- Multi-spiking neural network in python without the use of neural networking libraries
- Preliminary work on a slightly different mathematical formulation that *may* run faster (if it works)
- Convolutional neural network in python using PyTorch
- Some lightly preprocessed data for the convolutional network
- Discussions with Dr. Turuga and Dr. Oguz

Immediate Future

- Optimizing Multi-Spiking Neural Network using neural networking libraries
- Preparing data for running through the CNN
  - This will be (one of our) controls