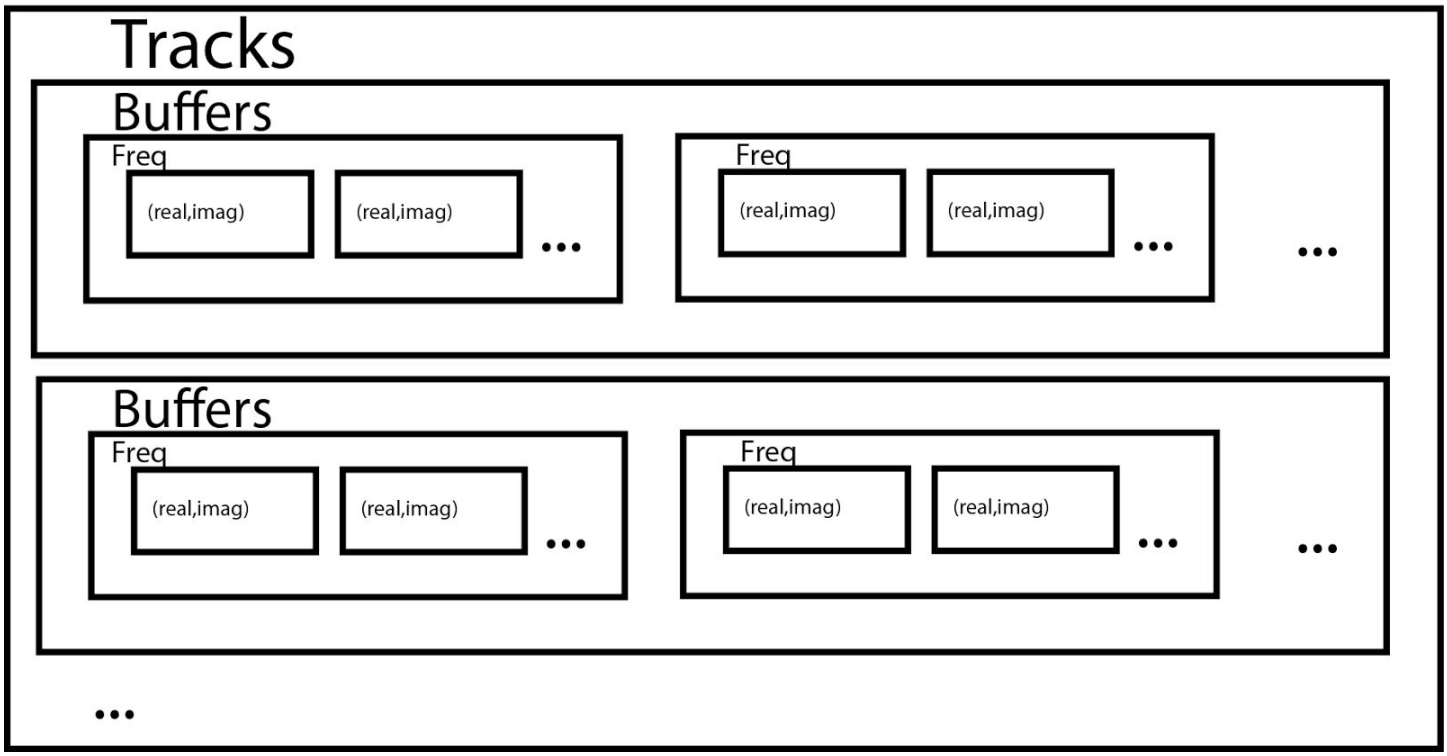
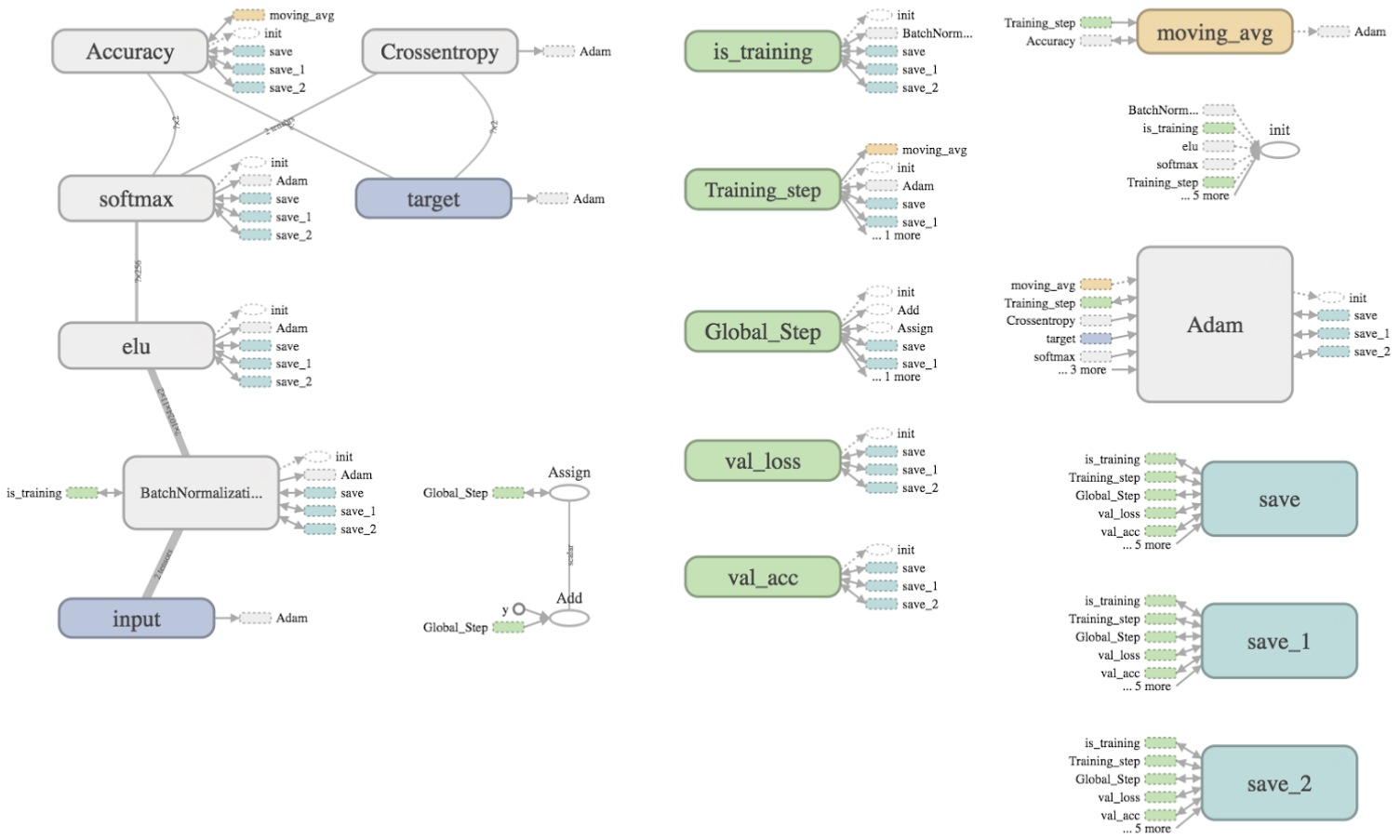
Machine Learning Drum Samples with Neural Networks

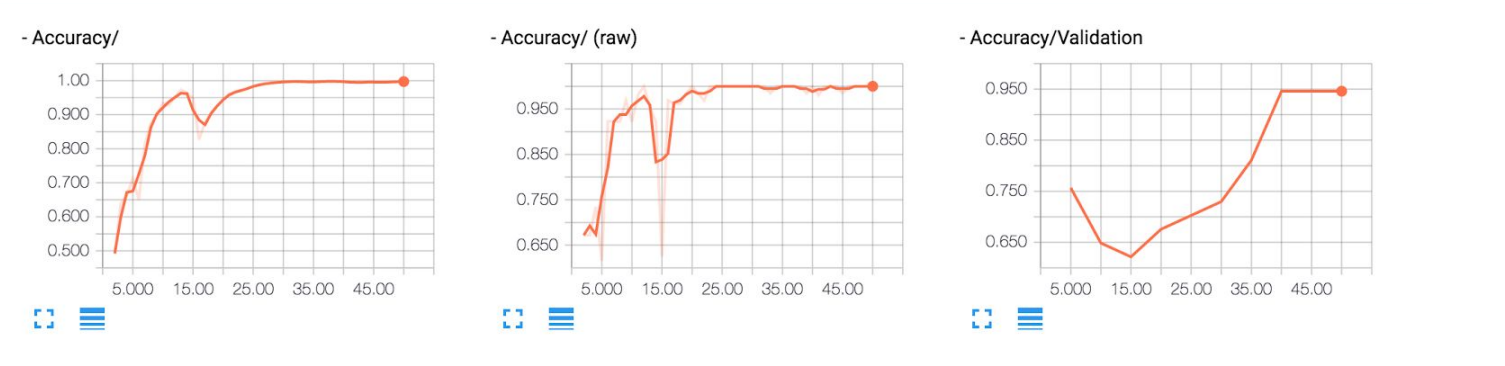
Corey Sery and John Ottenlips

Tensor - typed multiple dimension array



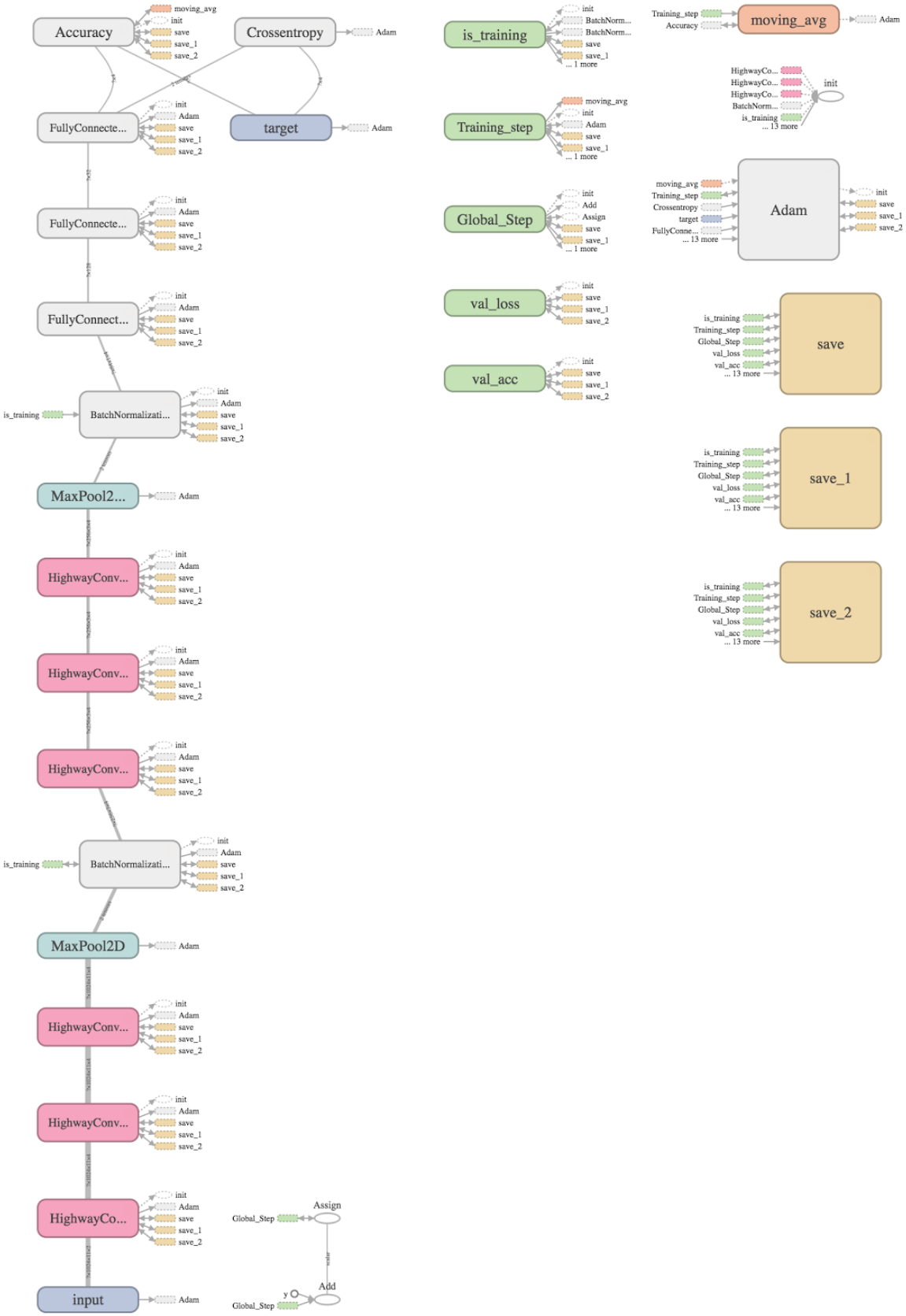
Multilayer Perceptron Network

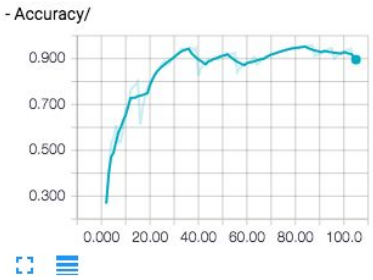


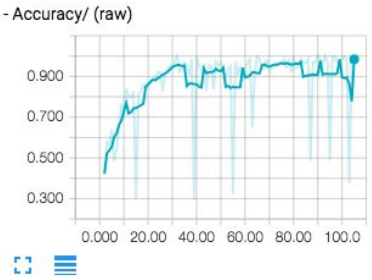


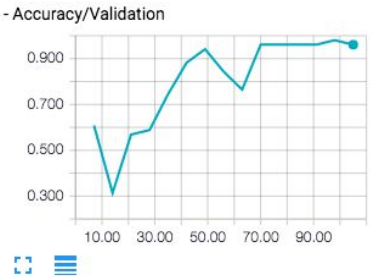

```
Training Step: 50  | total loss: 0.04457
| Adam | epoch: 010 | loss: 0.04457 - acc: 0.9972 | val_loss: 0.07630 - val_acc: 0.9459 -- iter: 308/308
--
Predictions for each class:  [ 13.  26.]
Validation accuracy :  1.0

Process finished with exit code 0
```

Highway Convolutional Network

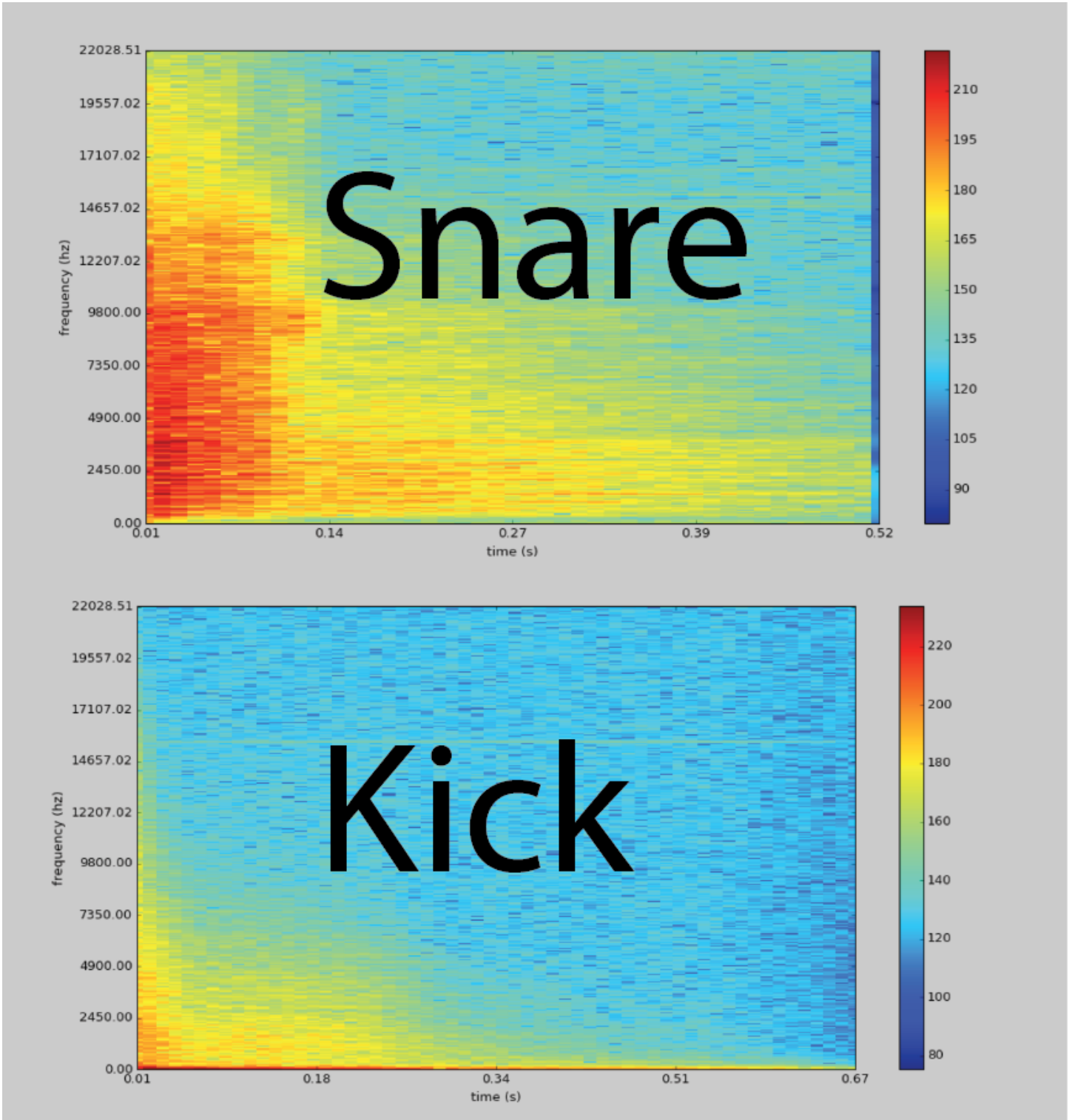









```
Training Step: 105  | total loss: 0.50834
| Adam | epoch: 015 | loss: 0.50834 - acc: 0.8957 | val_loss: 0.18966 - val_acc: 0.9608 -- iter: 421/421
--
Predictions for each class:  [ 12.   5.  14.  21.]
Actual for each class:  [ 12.   5.  14.  22.]
Validation accuracy :  0.9811320754716981
```


Snare
22028.51
19557.02
17107.02
14657.02
12207.02
9800.00
7350.00
4900.00
2450.00
0.00
frequency (hz)
0.01
0.14
0.27
0.39
0.52
time (s)
210
195
180
165
150
135
120
105
90
Kick
22028.51
19557.02
17107.02
14657.02
12207.02
9800.00
7350.00
4900.00
2450.00
0.00
frequency (hz)
0.01
0.18
0.34
0.51
0.67
time (s)
220
200
180
160
140
120
100
80

Goal: Our goal was to make a neural network that classified the difference between snare and kick drum samples. To do this, we decided to construct neural nets from the tensorflow library and use a data set of .wav files.

Process: For the data set, we collected kick and snare tracks from multiple libraries. We down sampled the tracks to 11.025k. This allowed us to run the process faster, the cost was loss of higher audio frequencies. We created a 4d tensor, or multidimensional array, of our audio. The dimensions were of tracks, batches, frequencies, and complex amplitude. Then we passed this tensor to two different neural net architectures. The first one was a multilayer perceptron network that used the Adam method for stochastic optimization. This network also had a batch normalization, softmax and elu hidden layer. The second network we used was a convolutional highway network which also used the Adam method. The difference with the convolutional network is that it has layers of depth as well as width to it. For a convolutional highway network all of the hidden layers are fully connected across unimpeded "information highways" to make it more efficient when processing deep layers. To prevent over fitting our data, we divided our data into three randomized sets: testing, training and validation. This way we know the model has generalized and will work with unseen data.

Results: We were able to classify our tracks with up to 98% accuracy with both a multilayer perceptron network and a convolutional highway network.

Future: We would like to use the model to generate audio as well as classify it, like what the Google Deep Dream project does with images. We would also like to work with bigger data sets, and longer audio samples, and unsupervised learning.

Citations

Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Highway networks. *arXiv preprint arXiv:1505.00387*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

https://www.tensorflow.org/versions/r0.12/api_docs/python/index.html (2016)

http://tflearn.org/ (2016)