X-Ray CNN Update **Ryan Taylor**
November 8, 2019

Python Adaptation, pt. III

The breakthrough has been achieved, through modification of the categorical Y sets and batch parameters. The output of the code is below, with added timestamps to indicate run times.

```
runfile('/Users/ryan/MURA-Network.py', wdir='/Users/ryan')
Data initialized in 27.22952699661255 seconds.
Using TensorFlow backend.
(51, 2) , (16, 2)
2019-11-17 11:14:11.697249: I
tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports
instructions that this TensorFlow binary was not compiled to use: AVX2
FMA
2019-11-17 11:14:11.748477: I
tensorflow/compiler/xla/service/service.cc:168] XLA service
0x7fab70231ba0 executing computations on platform Host. Devices:
2019-11-17 11:14:11.748540: I
tensorflow/compiler/xla/service/service.cc:175]   StreamExecutor device
(0): Host, Default Version
Epoch 1/3
```

```
10/10 [==============================] - 180s 18s/step - loss: 39.8984 -
accuracy: 0.5000 - val_loss: 23.6576 - val_accuracy: 0.5000
Epoch 2/3
10/10 [==============================] - 177s 18s/step - loss: 94.6570 -
accuracy: 0.5000 - val_loss: 117.9384 - val_accuracy: 0.3125
Epoch 3/3
10/10 [==============================] - 134s 13s/step - loss: 73.7994 -
accuracy: 0.5217 - val_loss: 307.5919 - val_accuracy: 0.5000
Network completed in 512.7943470478058 seconds.
```

The program correctly identified 52.17% of the 15 test samples over 3 complete iterations of the 50 training samples, thus achieving about 5.22% efficiency per input sample. This efficiency calculation (# correct identified / # training samples) is indicative of positive change per each iteration, which is confirmed by the relative improvements over each epoch (on average about 0.7% improvement per epoch).

The results of the network are further indicative of correct network structure due to two accurate results:

1. The network, over a short period (about 6 batches, 30 samples), is able to achieve 100% accuracy on training and validation sets with uniform results (all samples in both sets, in this case, were positive)
2. Once the training and validation sets were randomized to include both positive and negative results, the network achieved about 50% accuracy as indicated above, which should be around the baseline for a network with two possible predictions (given that it has learned very little)

In order to achieve close to 80% accuracy, thus, the network must be run for about 34.28 minutes. However, this is only in an ideal situation; the reality is that it will decline asymptotically given the small and non diverse set of data. Here is a comparison of the current state versus the ideal state:

State	Current	Ideal
Training Samples	50	9751
Test Samples	15	658
Epochs	3	100+
Batch Size	5	50+

At the current speed, the network would take 3,514,720 seconds to complete in the ideal state, or 40.68 days. As such, improving computing power is paramount to achieving a desirable outcome.

Current State

As of the project's current state, it works. There are significant obstacles to its efficient functioning, however. It is not currently obvious what the best performance is that the network can achieve in its current state, which will be reassessed ideally after it is put to the test on the full dataset. Doing so will likely require reserving computing power on the cloud, which might be done through AWS. But overall, however slowly, the network is learning.