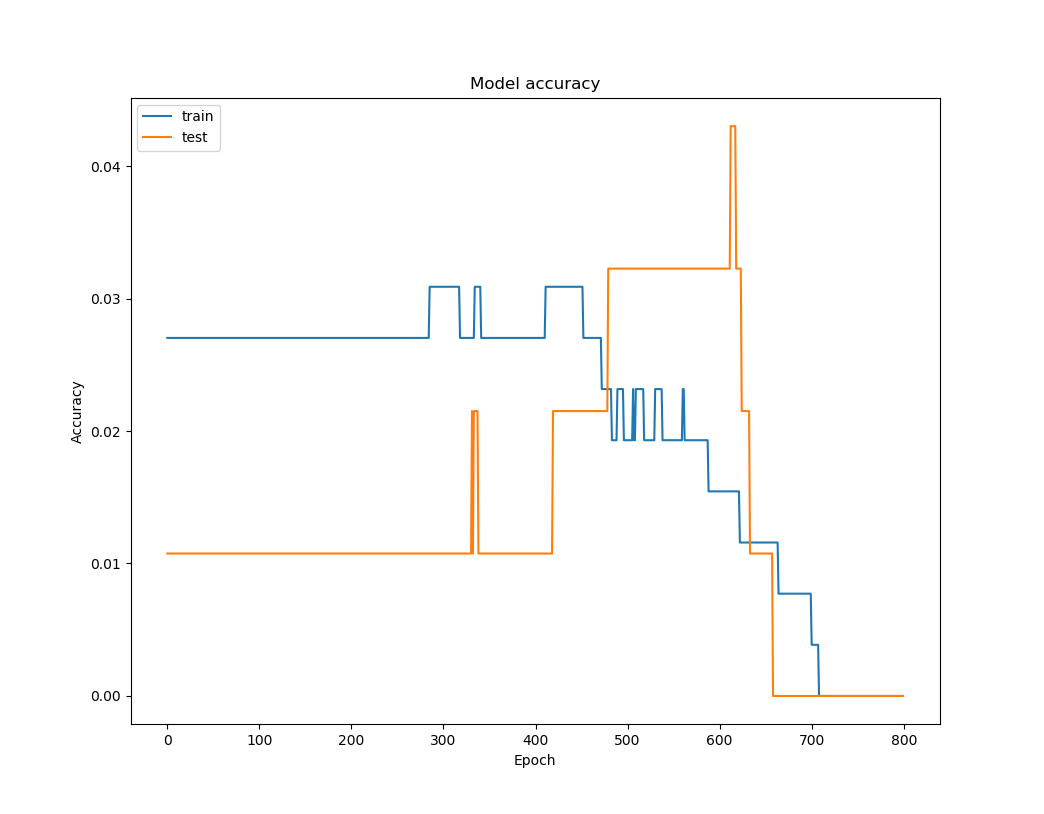
**RESULTS**

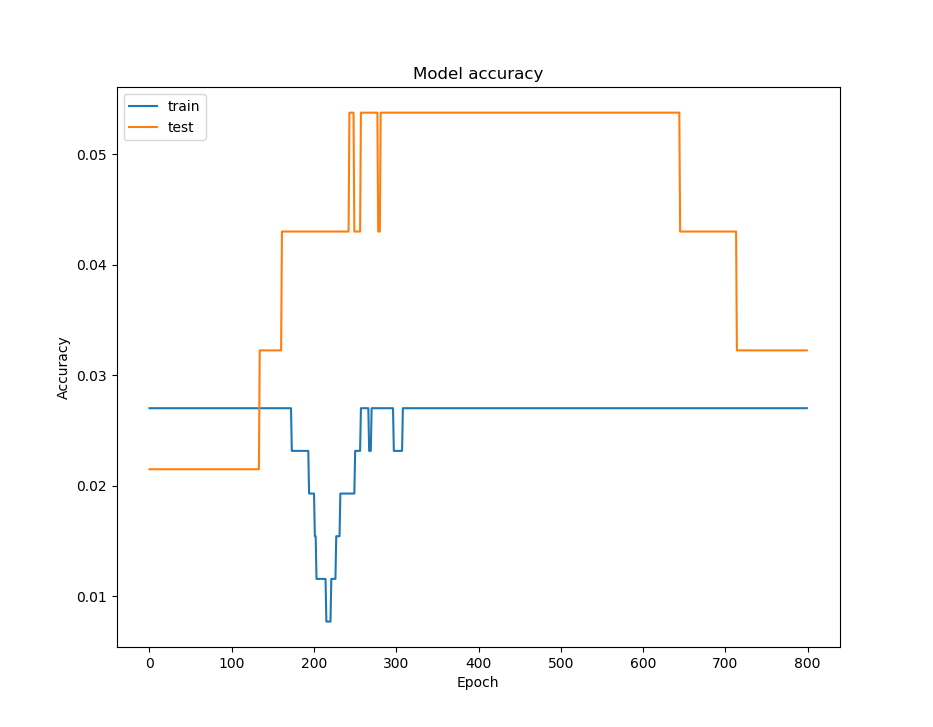
**Validation and training accuracies**

Every neural network is trained by traversing through a training dataset, and each pass through the training dataset is called an “epoch”. After every epoch, the accuracy of the network is tested separately using a different dataset called the validation dataset (Nielsen, 2015).

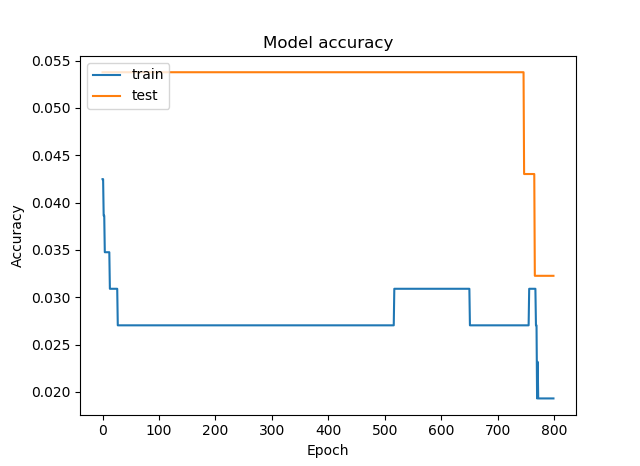
After three training sessions of 800 epochs each, the neural network showed a markedly low peak validation accuracy, or rate of correct predictions on the test dataset, of under 5.5%. Every session of 800 epochs showed stepwise and irregular increases and decreases of validation accuracy before consistently dipping to 0% towards the end of the training session. Below are graphs showing the validation (orange) and training (blue) accuracies of the neural network.



|  |
| --- |
| Figure 1. Training and validation accuracies of neural network, epochs 0-800 |



|  |
| --- |
| Figure 2. Training and validation accuracies of neural network, epochs 800-1600 |

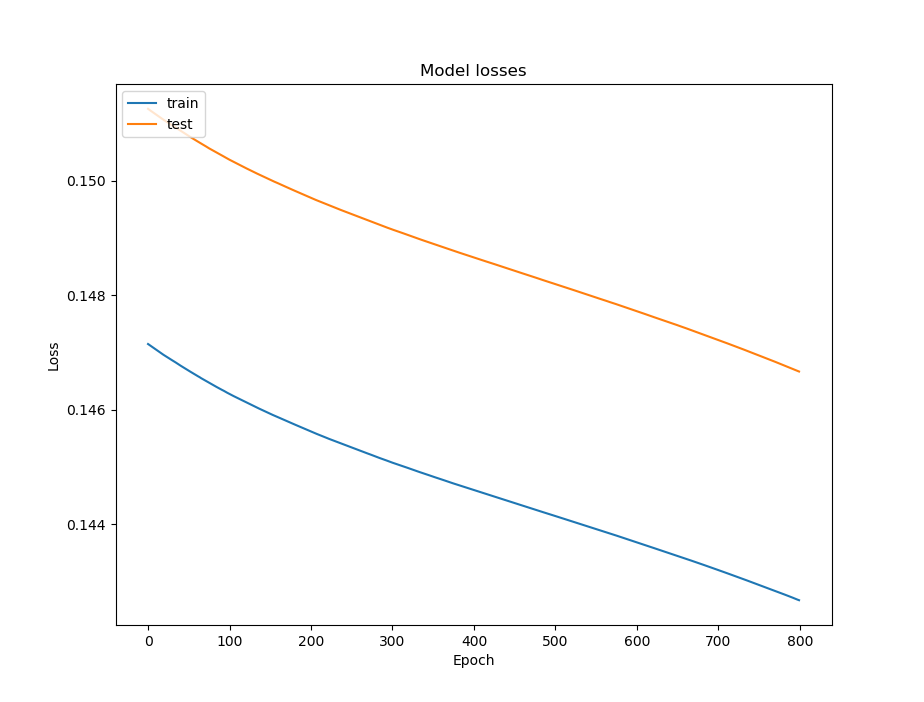


|  |
| --- |
| Figure 3. Training and validation accuracies of neural network, epochs 1600-2400 |

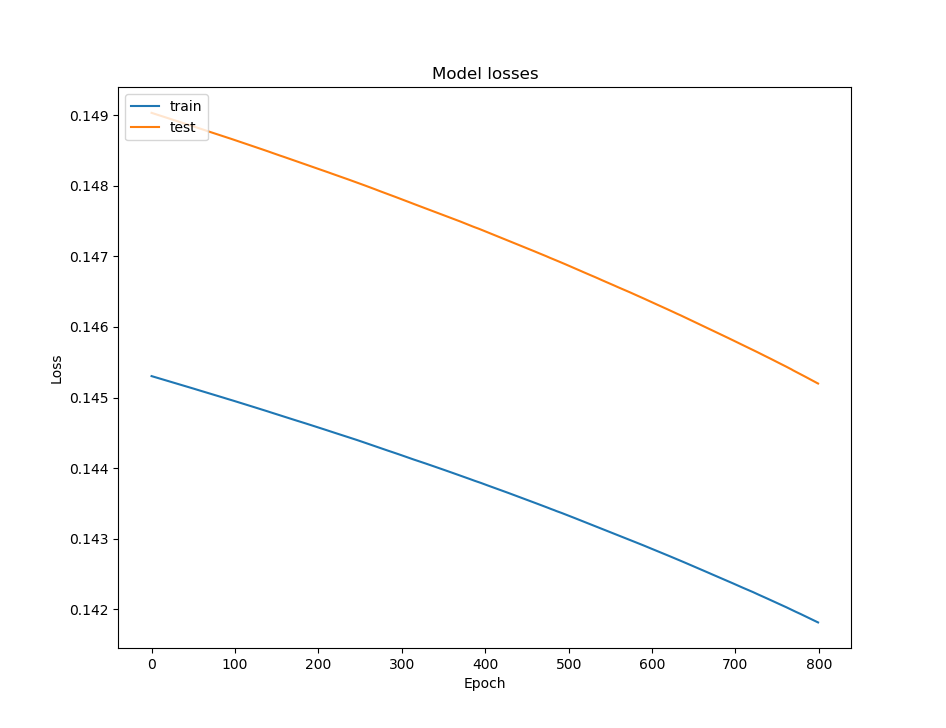
**Validation and training losses**

The loss function is a special series of mathematical operators that shows how far the neural network is from the correct prediction. As such, lower values are considered favorable (Nielsen, 2015).

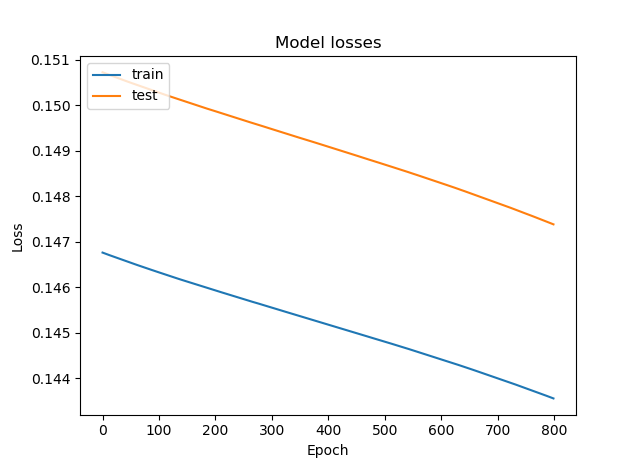
The same training sessions returned decreasing values of the loss function as the training session proceeded. The losses decreased in a smooth fashion, but the graphs formed were not of the same kind. Below are graphs showing the validation (orange) and training (blue) losses of the neural network.



|  |
| --- |
| Figure 4. Training and validation losses of neural network, epochs 0-800 |



|  |
| --- |
| Figure 5. Training and validation losses of neural network, epochs 800-1600 |



|  |
| --- |
| Figure 6. Training and validation losses of neural network, epochs 1600-2400 |

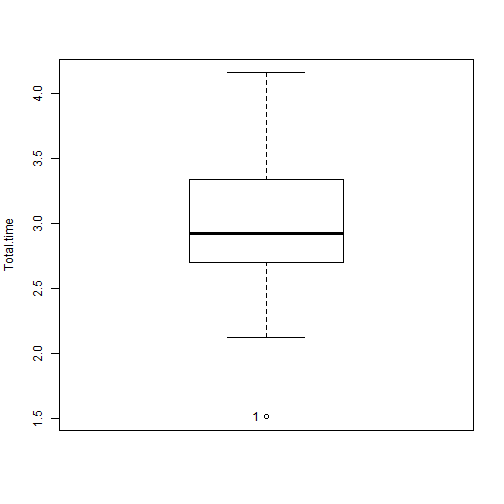
**Response time**

The study also aims to provide a neural network that responds within a commonly-accepted standard of 10 milliseconds (ms). The response time of the neural network is defined here as the time elapsed, as recorded by Python’s built-in timeit class, between a user input on the MIDI keyboard and a predicted chord output on the screen. 30 random chord samples were picked from the full dataset of 444 chords, and response times were taken using an algorithm.

Using a T-test for one mean set at 95% confidence, it was found that the neural network performed favorably against this standard, posting a mean response time of about 2.95 ms and a t-statistic adequately significant to reject the null hypothesis that the response time of the neural network is greater than or equal to 10 ms.

|  |  |
| --- | --- |
| **Mean (ms)** | 2.95466667 |
| **t-statistic** | -17.19 |
| **t-critical** | -1.699 |
| **null hypothesis outcome** | reject, since -17.19 < -1.699 |

|  |
| --- |
| Table 7. Results of t-test for one mean on neural network response time. |



|  |
| --- |
| Figure 8. Boxplot of total response times of neural network to MIDI inputs in milliseconds. Note: Null hypothesis is that response time is less than or equal to 10 ms. The program used to make this boxplot does not output units on the x-axis. |

**DISCUSSION**

The neural network performs atypically from the perspective of validation and training accuracies. It is expected that the training and validation accuracies would increase as the training proceeds. However, it is behaving as expected in terms of validation and training losses, as they decrease during training (Nielsen, 2015).

Bodik (2018) suggests that this problem is caused by the input and output formats of the neural network being too complex for the neural network to properly classify input data. This can cause the neural network to underfit, i.e. not be able to adapt to the characteristics of the training data. This is manifested by consistent but unfavorable values of validation accuracy. (Bodik, 2018; Nielsen, 2015). It is therefore recommended that steps to reduce this underfitting be taken in the future.

**References**

Bodik, S. (2018, Feb 15). *Answer to “Low loss and low accuracy. What is the reason?”* [Online forum comment]. Retrieved March 5, 2018, from https://stats.stackexchange.com/posts/328889/revisions.

Nielsen, M. A. (2015). *Neural networks and deep learning*. Determination Press.