ACML Homework
Music by RNNs

Ibrahim Hadzic (i6200920), Alexander Reisach (i6197692)

github.com/Scriddie/midi-rnn

November 13, 2019

1 Getting Things Running

We adapted the code from *https://github.com/brannondorsey/midi-rnn* to tensorflow 2.1, and ran some experiments with the default settings. As can be seen in 1, although training loss is steadily decreasing, there is no clear trend visible for the validation performance.

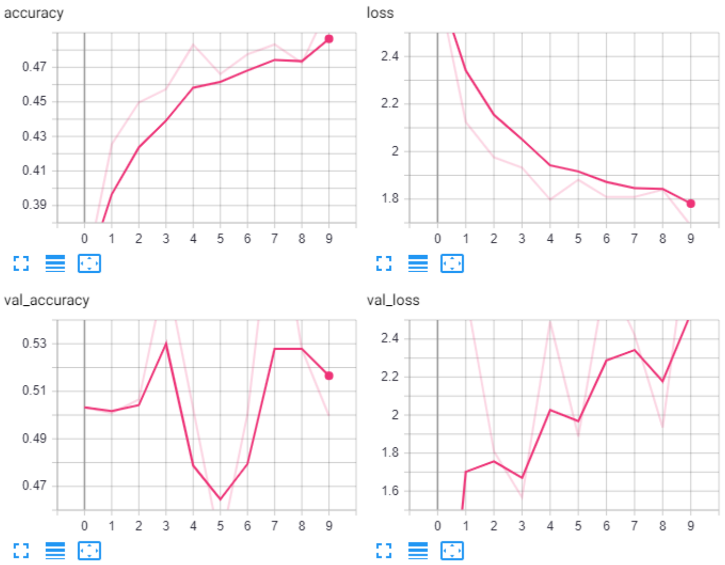


Figure 1: Tensorboard output for default settings

In order to test different and more computationally intensive settings, we use the Aachen cluster's GPU. We use keras' CuDNNLSTM layers (see file *train_gpu.py*) instead of regular LSTM layers to take advantage of the accelerated linear algebra (XLA) optimizations.

2 Experiments

Ever since the term ”random guessing” has come out of fashion, we have specialized in ”principled experimentation” instead, and this project is no exception. Following best practices from the internet (c.f. 2), we experimented in particular with the following parameters:

- number of (LSTM) layers and nodes per (LSTM) layer
- input window size
- numper of training epochs
- dropout ratio

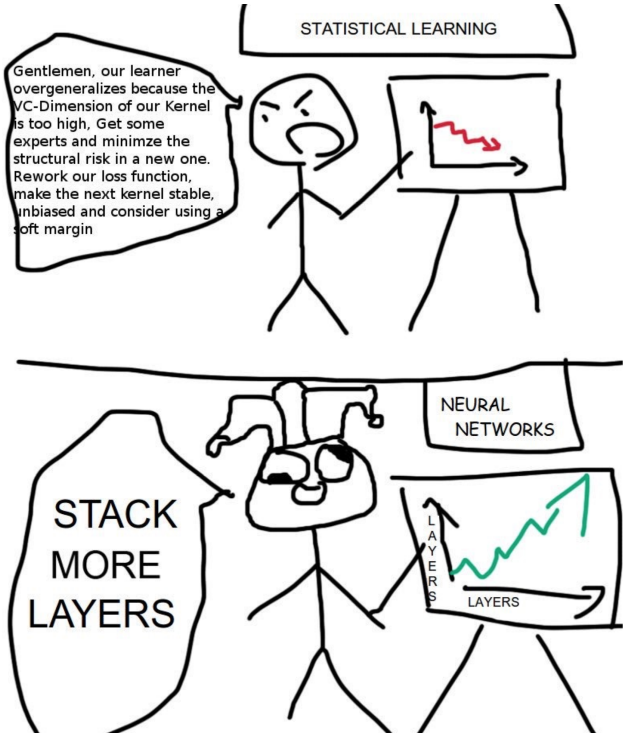
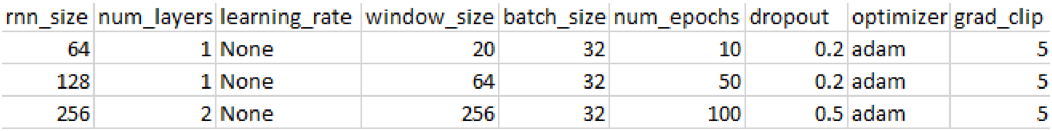

STATISTICAL LEARNING
Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high, Get some experts and minimze the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin
NEURAL NETWORKS
STACK MORE LAYERS
LAYERS
LAYERS

Figure 2: Good advice from the internet

In this spirit, we ran our model with the following settings:



rnn_size	num_layers	learning_rate	window_size	batch_size	num_epochs	dropout	optimizer	grad_clip
64	1	None	20	32	10	0.2	adam	5
128	1	None	64	32	50	0.2	adam	5
256	2	None	256	32	100	0.5	adam	5

Figure 3: Experiment Schedule

3 Results

All trained models can be found in the folder *cluster_experiments*. Each model contains a subfolder *generated* with *.mid* files containing model generated output. The first model, trained only for 10 epochs does produce sounds, but they can hardly be called music, and there seem to be frequent ”blank spots”, where there is no generation at all for a short period of time. The second model, trained for 50 epochs and with a larger hidden layer produces outcomes that sound more melodic and are largely free from ”blank spots”. The third model (two hidden layers with 256 nodes each for 100 epochs) produces the greatest variety of low-pitched and high-pitched notes. While there are some melodic elements here and there, there is still a huge gap to the original input files. To make sure the model does not only remember songs it has seen during training, we also generated music from a holdout set. We did not notice a difference in musical quality using this input as compared to before.