

Report

Marcin Mazur

July 23, 2022

1 Architecture

For the *Coupling Layers* I used simple MLPs with *LeakyReLU* and *Batch-Norm1d*. I tried many activation functions: *ReLU*, *Sigmoid*, *Tanh*, but *LeakyReLU* turned out to give the best results. When it comes to batch norm I tried to minimize momentum as much as possible (as far as training did not become unstable). In *RealNVP* I decided to go for 10 Coupling Layers of 5 flavours (different sizes of MLPs). Each flavour first was applied to first half of input vector and then second half (simply use the same configuration twice, one after another).

2 Optimization

For the optimization I used *Adam* with warm-up and reducing on plateau. Model tended to learn best till approximately 60 epoch, after that progress was very slow.

3 Results

I reached minimal -1600 loss, but could not get better scores. I think that was caused by the fact that model was not complex enough (more complex models gave me 'CUDA out of memory' errors, so I stuck to the final model presented in notebook).

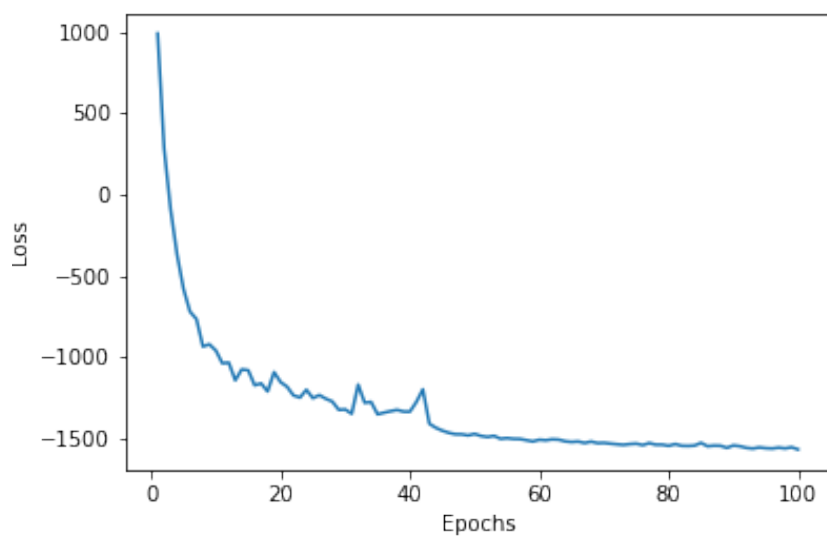


Figure 1: Loss

As you can see fives produced by sampling from normal distribution and passed to model inverse flow are pretty good. The samples produced by model definitely look very alike and tend to be better (when it comes to symmetry, precision, etc.) than handwritten digits.

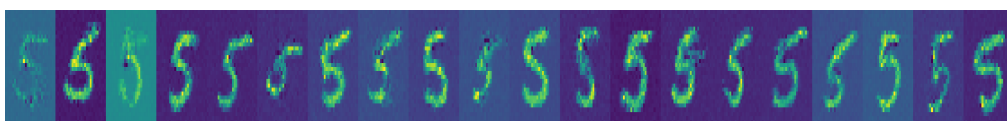


Figure 2: Samples