DT8122 Project Assignment: Diffusion and BFN

Ali Eftekhari Milani

September 2024

1 Diffusion

This section deals with the implementation of a diffusion model for the MNIST dataset.

1.1 Network

The network used is a simple U-Net architecture without attention mechanism. An Exponential Moving Average (EMA) is used to smooth model parameters and stabilise training. Furthermore, the Channel Shuffle operation proposed in the ShuffleNet paper[1] is used for increased model efficiency. The time step information is incorporated through an MLP-based embedding with a SiLU activation function.

1.2 Challenges

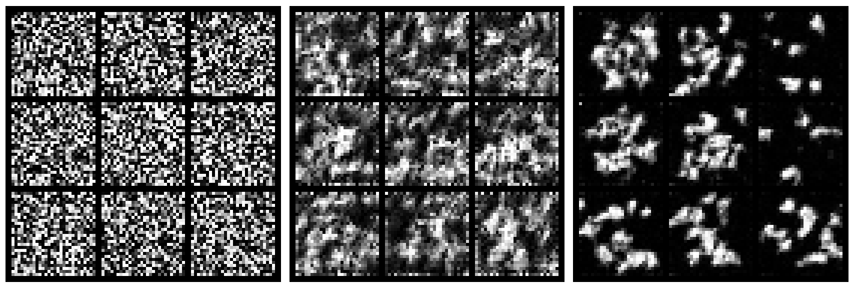
One of the main challenges was finding a trade-off between model complexity and the quality of results. Existing implementations were too large to be trained on a laptop without a GPU, and simple CNN architectures did not learn adequately. What ended up working was sticking to the U-Net architecture without attention mechanism and with reduced size, coupled with model parameter smoothing using EMA and some operations suggested in the ShuffleNet paper.

1.3 Plots

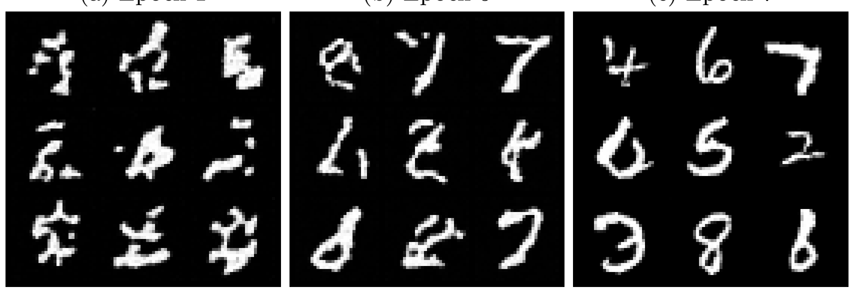
During training, 9 digits are sampled and saved at each epoch. In Figure 1, they are shown at six different epochs during training, depicting the training progress.

Once the training is complete, starting with an initial sample from the prior, the trained model is used to implement the reverse diffusion and the outputs at ten different diffusion steps selected uniformly among the 1000 steps are shown in Figure 2.

[1] arxiv:1707.01083v2



(a) Epoch 1 (b) Epoch 5 (c) Epoch 7



(d) Epoch 14 (e) Epoch 18 (f) Epoch 70

Figure 1: Diffusion - Nine digits sampled at different stages of the training.

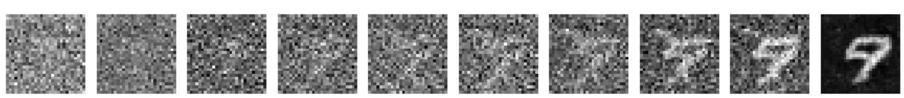


Figure 2: Diffusion - The reverse diffusion process of a given sample from the prior at ten diffusion steps.

2 Bayesian Flow Network

This section deals with the implementation of the Bayesian Flow Network (BFN) for the MNIST dataset.

2.1 Comparison of Diffusion and Bayesian Flow Networks

2.1.1 Similarities

Both models are probabilistic generative models that rely on stochastic processes (diffusion models on a Markov process, BFNs on Bayesian inference) to learn the underlying data distribution. Both models operate based on a sequential transformation process.

2.1.2 Differences

The diffusion model operates based on a forward process where noise is added sequentially and a backward process where the network learns to sequentially denoise the image. In contrast, the BFN operates with a single process where it iteratively gains information about the data.

2.1.3 Learning discrete distributions

When learning discrete distributions, diffusion models need special techniques like quantization or discrete state space modelling to handle the fact that they traditionally operate in continuous spaces. Several adaptations exist (like categorical diffusion models), but the process of mapping discrete data to a continuous space and back can introduce complexity and inefficiencies. In contrast, BFNs, due to their flexible nature with priors and posteriors, are inherently more capable of handling discrete distributions. The Bayesian framework allows for directly modelling discrete priors and updating them through the posterior, making it a more natural fit for discrete data. Therefore, BFNs do not require a modification to handle discrete variables, as the flow-based transformations can be applied to both continuous and discrete distributions, especially when combined with appropriate priors (e.g., categorical distributions).

2.2 Plots

An available implementation of the BFN[2] has been used in this section. The model is trained for 20 epochs. Since the U-Net architecture in this implementation is larger than the one used in the diffusion model, the training time was considerably longer, around 9 hours for 20 epochs of training.

After the completion of the training, nine digits are sampled from the output distribution and are shown in Figure 3. Furthermore, a generated digit across ten steps is plotted in Figure 4.

[2] https://github.com/Algomancer/Bayesian-Flow-Networks

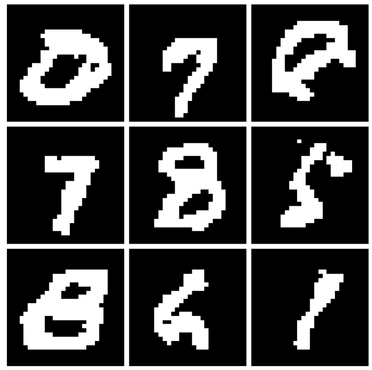
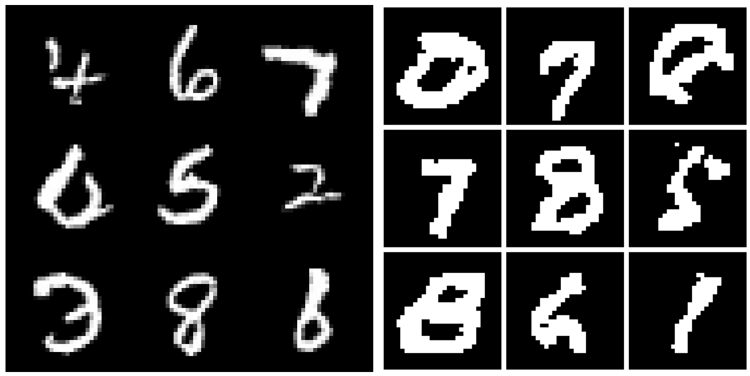


Figure 3: BFN - Nine digits sampled from the output distribution.

Figure 4: BFN - A digit plotted across ten steps.

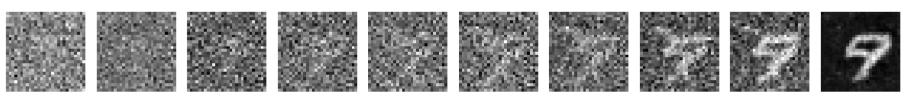
2.3 Comparison of the results of Diffusion and BFN

Nine digits sampled by the trained Diffusion model and BFN are plotted together in Figure 5. In addition, the generation process across 10 steps is compared in Figure 6. Both models appear to generate acceptable results, with slightly better-looking results for the Diffusion model. However, a fair and rigorous comparison is not possible since the size of the models used in the two methods are different, and the training is done for different numbers of epochs.



(a) Diffusion (b) BFN

Figure 5: Nine digits sampled at different stages of the training: A comparison of Diffusion and BFN results.



(a) Diffusion

(b) BFN

Figure 6: A digit plotted across ten steps, starting with a sample from the prior and ending with the final generated image: A comparison of Diffusion and BFN results.