LU-Net: Invertible Neural Networks Based on Matrix Factorization*

Robin Chan[†], Sarina Penquitt[†] and Hanno Gottschalk

March 30, 2023

LU-Net is a novel invertible neural network architecture motivated by the positive properties of normalizing flows. This network stands out with a simple and fast architecture and serves as a generative model. It is based on the factorization of quadratic weight matrices $\mathsf{A} = \mathsf{LU}$, where L is a lower triangular matrix with ones on the diagonal and U an upper triangular matrix. Instead of learning a fully occupied matrix A, we learn L and U separately. If combined with an invertible activation function Φ, such layers can easily be inverted whenever the diagonal entries of U are different from zero, see Fig. 1. Also, the computation of the determinant of the Jacobian matrix of such layers is cheap. Consequently, the LU architecture allows for cheap computation of the likelihood via the change of variables formula and can be trained according to the maximum likelihood principle.

(a) normalizing sequence:

$$\begin{aligned} \mathsf{x} &\mapsto \mathsf{Ux} \\ &\mapsto \mathsf{LUx} \\ &\mapsto \mathsf{LUx} + \mathsf{b} \\ &\mapsto \Phi(\mathsf{LUx} + \mathsf{b}) \\ &= \mathsf{f}(\mathsf{x}) =: \mathsf{z} \end{aligned}$$

(b) generating sequence:

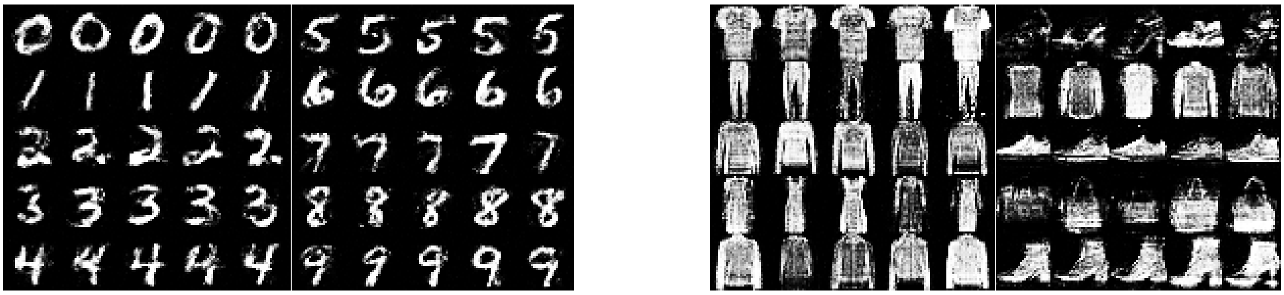
$$\begin{aligned} \mathsf{z} &\mapsto \Phi^{-1}(\mathsf{z}) \\ &\mapsto \Phi^{-1}(\mathsf{z}) - \mathsf{b} \\ &\mapsto \mathsf{L}^{-1}(\Phi^{-1}(\mathsf{z}) - \mathsf{b}) \\ &\mapsto \mathsf{U}^{-1}\mathsf{L}^{-1}(\Phi^{-1}(\mathsf{z}) - \mathsf{b}) \\ &= \mathsf{f}^{-1}(\mathsf{z}) = \mathsf{x} \end{aligned}$$

Figure 1: Illustration of the LU-Net design with one LU layer. Here, (a) represents the forward or "normalizing" direction and (b) the reversed or "generating" direction. Note that the weights are shared and that $(\mathsf{f}^{-1} \circ \mathsf{f})(\mathsf{x}) = \mathsf{x}$.

The numerical experiments are conducted in different settings. As a toy example, we apply LU-Net to learn a two-dimensional Gaussian Mixture. Then, we apply LU-Net to the image datasets MNIST as well as the more challenging Fashion-MNIST. With regard to LU-Net as generator, we obtain reasonable quality of the samples. The random samples can clearly be recognized as subset of MNIST (Fig. 2a) and Fashion-MNIST (Fig. 2b) and moreover, they can also be assigned easily to the corresponding classes.

*https://arxiv.org/abs/2302.10524

[†]equal contribution



(a) MNIST (b) Fashion MNIST

Figure 2: LU-Net: Randomly generated samples of (a) MNIST and (b) Fashion MNIST

In final experiments, we compare LU-Net with the popular normalizing flow architecture RealNVP. To make the models better comparable, we design them to be of similar size in terms of model parameters and conduct the same experiments for RealNVP as for the LU-Net. RealNVP is computationally more expensive than the LU-Net, which can be seen in Tab. 1 showing a comparison of the computional budget. RealNVP not only require more GPU memory than LU-Net but also considerably more time to train.

model	num weight parameters	GPU memory usage in MiB	num epochs training	test NLL in bits/pixel
LU-Net	4.92M	1,127	40	3.2424
RealNVP	5.39M	3,725	100	5.6819

model	train epoch in sec	optimization step in ms	density per image in ms	sampling per image in ms
LU-Net	7.32	1.2	37.10	45.15
RealNVP	99.88	56.0	259.15	1.03

Table 1: Model size and run time comparisons between LU-Net and RealNVP