

Team-Improving-VAEs: Midterm Presentation

Callihan Bertley, Claire Gan, Rishi Leburu, Malia Walewski
June 25, 2024 — Emory University
Mentor: Dr. Deepanshu Verma



VAE Skeleton

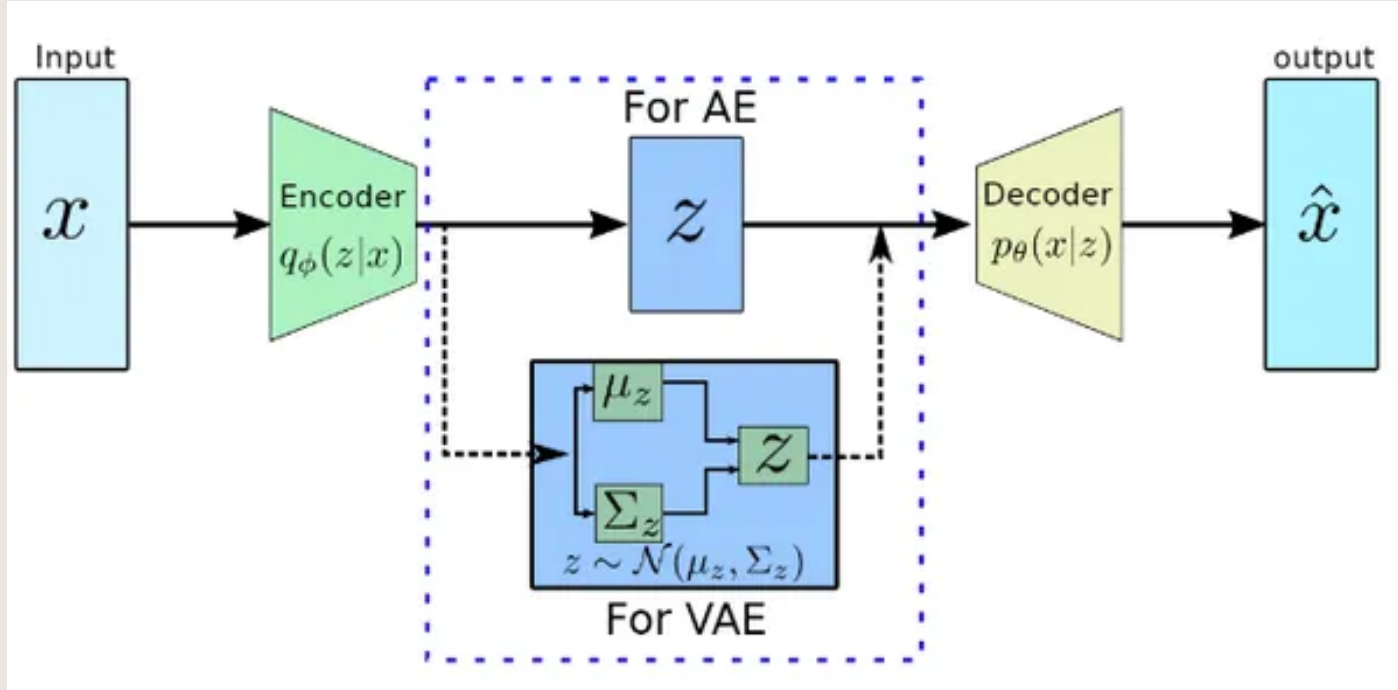


Image Source: "Generative AI with VAEs, GANs Transformers," Analytics Vidhya, 19 July 2023, <https://www.analyticsvidhya.com/blog/2023/07/generative-ai-with-vaes-gans-transformers/>. Accessed 24 June 2024.

[1]



Motivation

- VAEs default to a normal Gaussian to approximate the posterior
- The standard VAE makes weak assumptions about the latent space because it approximates it using a Gaussian. [2]



Objective

- Use Normalizing Flows to transform a Gaussian to better match the complex posterior distribution, $p_{\theta}(\mathbf{z}|\mathbf{x})$, of a VAE after training
- Experiment transforming the approximate posterior $q_{\psi}(\mathbf{z}|\mathbf{x})$ using different conditional NFs
 - To improve generalization error
 - Create better generative model



Introduction to NFs

- **Normalizing Flows** transform a simple probability distribution into a more complex distribution
- This transformation is done through a sequence of invertible and differentiable mappings.

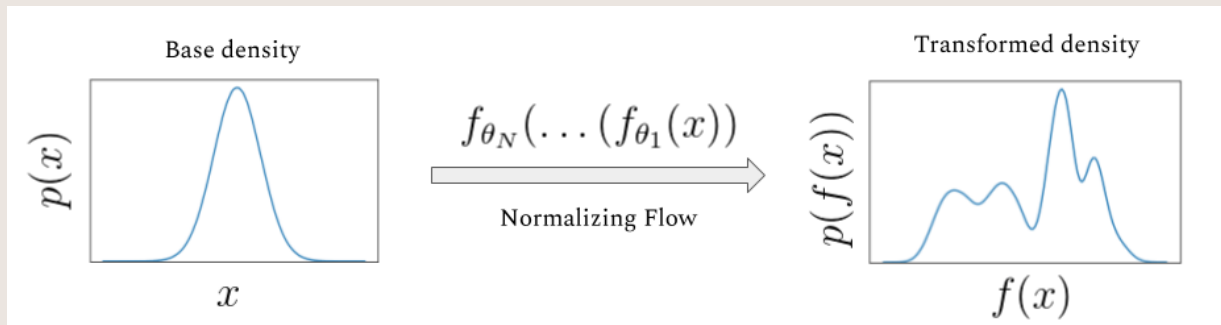
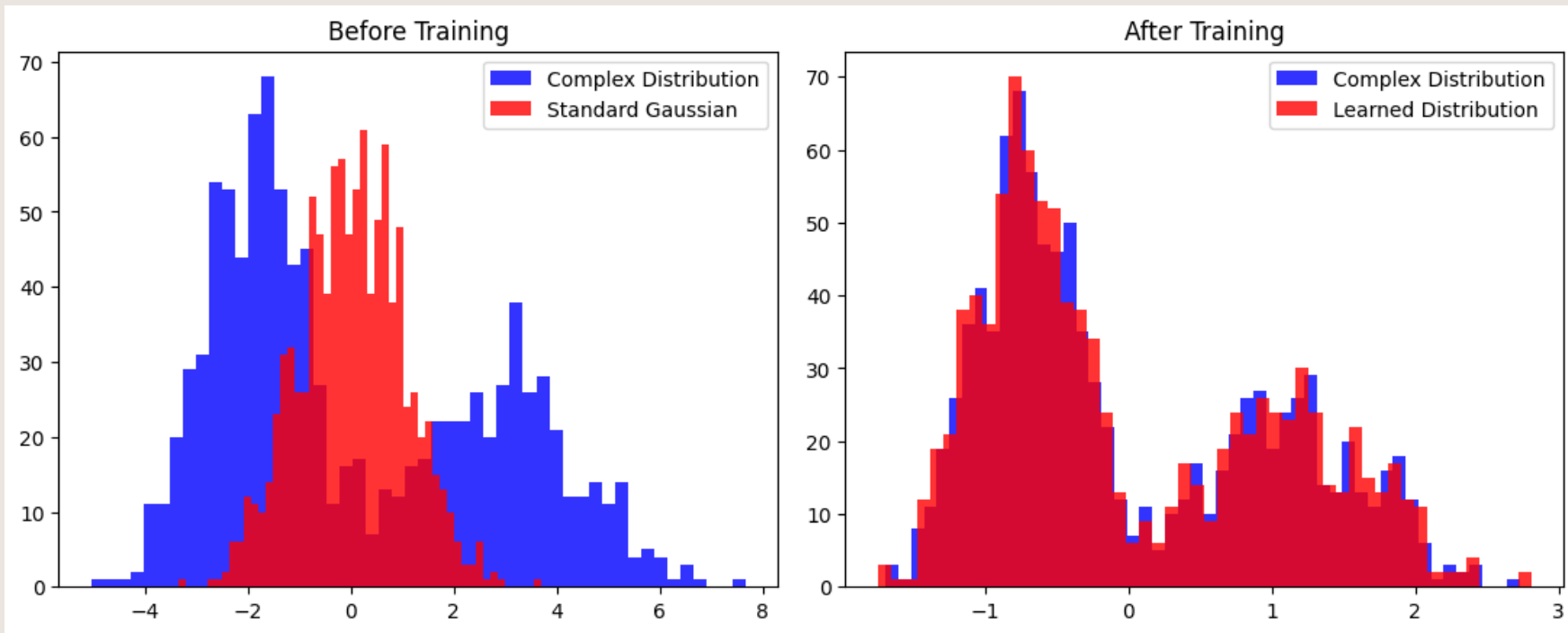


Image Source: "Normalizing Flows for Variational Inference," by B. M. Geboe, <https://gebob19.github.io/normalizing-flows/>. Accessed June 24, 2024



Practice NF Model





Inverse Autoregressive Transformations (IAT)

- When a random sample \mathbf{z}_0 is passed through a function the pdf of the transformed sample is as below:

$$q_\psi(\mathbf{z}_1 | \mathbf{x}) = q_\psi(\mathbf{z}_0 | \mathbf{x}) \left| \det \left(\frac{d\mathbf{z}_0}{d\mathbf{z}_1} \right) \right| \quad (1)$$

$$\log(q_\psi(\mathbf{z}_1 | \mathbf{x})) = \log(q_\psi(\mathbf{z}_0 | \mathbf{x})) - \log \left(\left| \det \left(\frac{d\mathbf{z}_1}{d\mathbf{z}_0} \right) \right| \right) \quad (2)$$

- Applying T transformations sequentially:

$$\log(q_\psi(\mathbf{z}_T | \mathbf{x})) = \log(q_\psi(\mathbf{z}_0 | \mathbf{x})) - \sum_{t=1}^T \log \left(\left| \det \left(\frac{d\mathbf{z}_t}{d\mathbf{z}_{t-1}} \right) \right| \right) \quad (3)$$



Inverse Autoregressive Transformations (IAT)

- Computational difficulties can be solved using Inverse Auto Regressive Transformations
- Multiply a sample by σ and add μ (outputs of an autoregressive neural network)
- Networks are built such that the parameters σ and μ for the j^{th} dimension of the transformed variable only depend on the $[0, j - 1]$ inputs and the context vector, h . [2]

$$\implies \mathbf{z}_1^j = \mu^j(\mathbf{z}_0^{j-1}, \dots, \mathbf{z}_0^1, h) + \sigma^j(\mathbf{z}_0^{j-1}, \dots, \mathbf{z}_0^1, h)\mathbf{z}_0^j \quad (4)$$



Inverse Autoregressive Transformations (IAT)

$$\frac{dz_1}{dz_0} = \begin{bmatrix} \sigma^1 & 0 & 0 & 0 \\ \dots & \sigma^2 & 0 & 0 \\ \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \sigma^K \end{bmatrix} \quad (5)$$

- If T transformations are applied sequentially:

$$\log(q_\psi(\mathbf{z}_T | \mathbf{x})) = \log(q_\psi(\mathbf{z}_0 | \mathbf{x})) - \sum_{t=1}^T \sum_{k=1}^K \log(\sigma_t^k) \quad (6)$$



Inverse Autoregressive Flows (IAF)

- Inverse Autoregressive Flows (IAFs) apply a series of invertible, autoregressive transformations to a base distribution

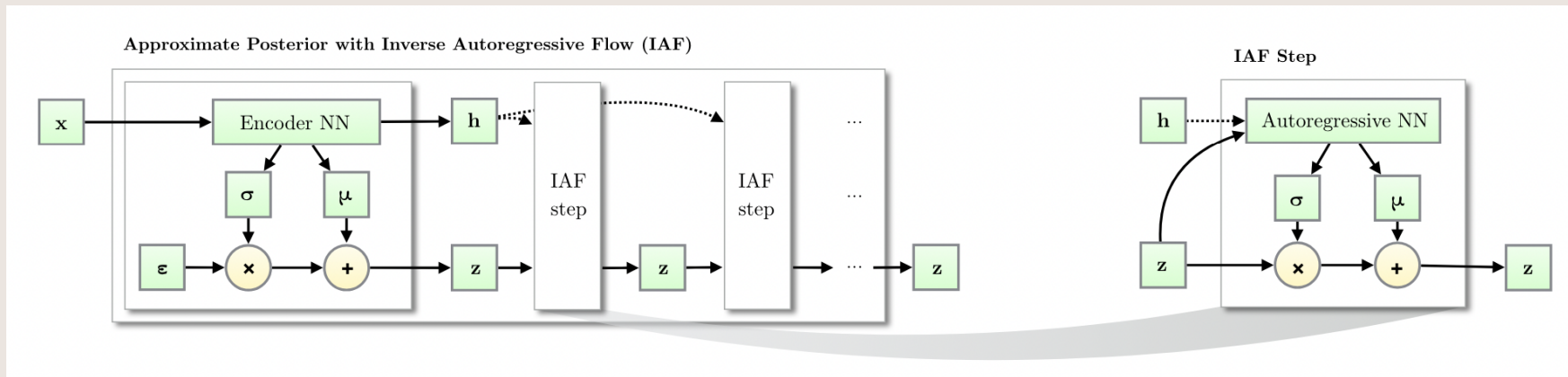


Image Source: L. Midgley, "Improving variational inference with inverse autoregressive flow," University of Cambridge, Tech. Rep., 2021. [Online]. Available: <https://github.com/lollcat/Autoencoders-deep-dive/blob/Pytorch/Report.pdf>.



Inverse Autoregressive Flows (IAF)

- The main advantage of IAFs/IATs are that they retain most of the properties of a standard Gaussian
- They are easy to compute (so easy to sample from as well) and parallelizable for high dimensional z



The KL Divergence & The ELBO

$$D_{kl} \left(q_{\psi}(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z}|\mathbf{x}) \right) = -(\text{ELBO}) + \log p_{\theta}(\mathbf{x}) \quad (7)$$

$$\text{ELBO} = \underbrace{\mathbb{E}_{\mathbf{z} \sim q_{\psi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})]}_{\text{reconstruction term}} - \underbrace{D_{kl} \left(q_{\psi}(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}) \right)}_{\text{regularization term}} \quad (8)$$

VAEs allow for efficient training of generative models by minimizing the negative evidence lower bound (ELBO). [3]



Coding is hard

```
30 # Backward pass and optimization step
31 loss.backward(retain_graph=True)
--> 32 optimiser.step()
33
34
----- 2 frames -----
/usr/local/lib/python3.10/dist-packages/torch/autograd/graph.py in _engine_run_backward(t_outputs,
*args, **kwargs)
742     unregister_hooks = _register_logging_hooks_on_whole_graph(t_outputs)
743     try:
--> 744         return Variable._execution_engine.run_backward( # Calls into the C++ engine to run
the backward pass
745             t_outputs, *args, **kwargs
746         ) # Calls into the C++ engine to run the backward pass

RuntimeError: one of the variables needed for gradient computation has been modified by an inplace
operation: [torch.FloatTensor [10000, 400]], which is output 0 of AnStridedBackward0, is at version 2;
expected version 1 instead. Hint: the backtrace further above shows the operation that failed to
compute its gradient. The variable in question was changed in there or anywhere later. Good luck!
```

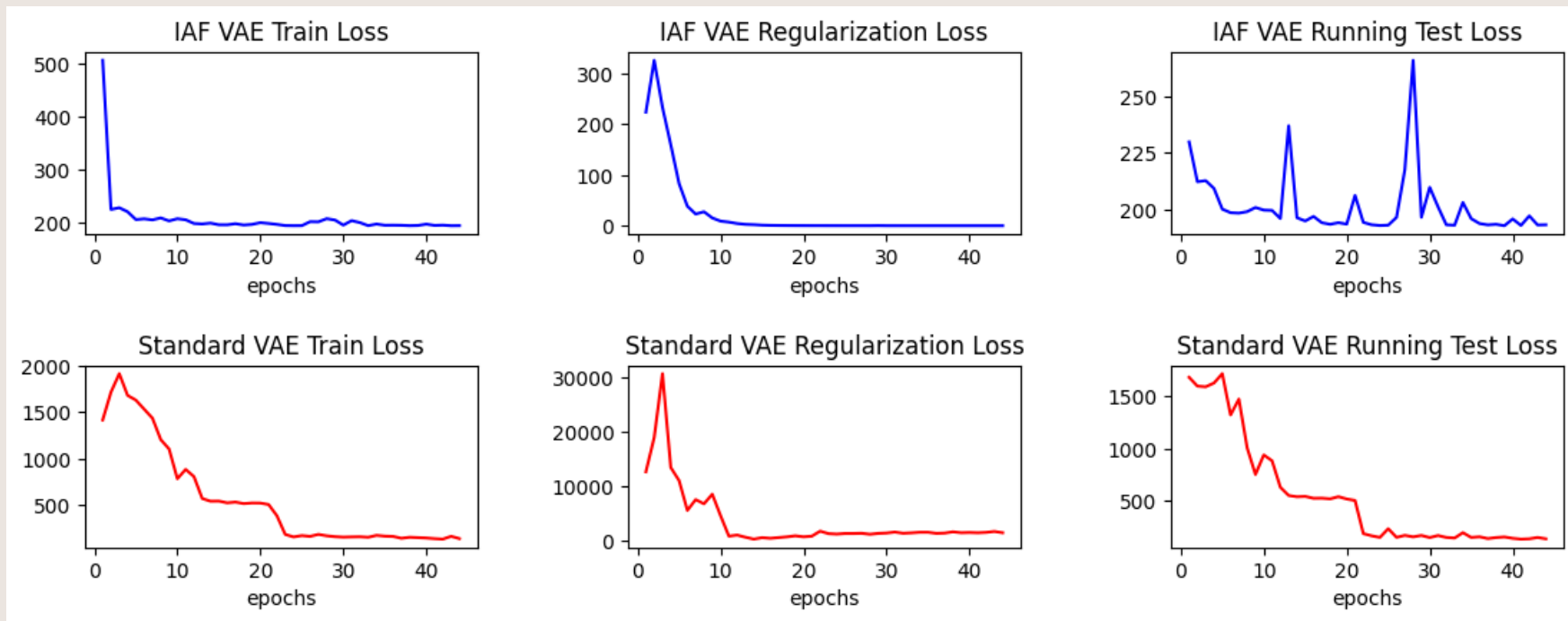
Train Model

12s completed at 3:33 PM

fied by an inplace
ard0, is at version 2;
n that failed to
later. Good luck!



IAF implementation into VAE





Next Steps

Implement Partially Convex Potential Maps (PCP-Map) [4]

- another type of architecture to transform simple distributions to complex ones
- experiment and compare results with IAF-VAE and standard VAE

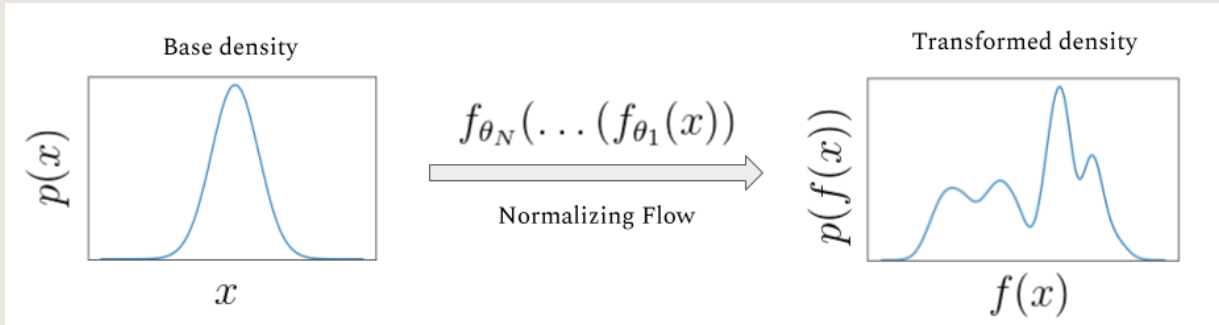


Image Source: "Normalizing Flows for Variational Inference," by B. M. Geboe, <https://gebob19.github.io/normalizing-flows/>. Accessed June 24, 2024



References

- [1] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," 2014.
 - [2] L. Midgley, "Improving variational inference with inverse autoregressive flow," University of Cambridge, Tech. Rep., 2021. [Online]. Available: <https://github.com/lollcat/Autoencoders-deep-dive/blob/Pytorch/Report.pdf>
 - [3] L. Ruthotto and E. Haber, "An introduction to deep generative modeling," *GAMM-Mitt.*, vol. 44, no. 2, pp. Paper No. e202 100 008, 24, 2021. [Online]. Available: <https://doi.org/10.1002/gamm.202100008>
 - [4] Z. O. Wang, R. Baptista, Y. Marzouk, L. Ruthotto, and D. Verma, "Efficient neural network approaches for conditional optimal transport with applications in bayesian inference," 2023. [Online]. Available: <https://arxiv.org/abs/2310.16975>
- Special thanks to NSF funding awards DMS-2051019 and DMS-1751636

EMORY

