Optimising Latent Space Representations with Variational Autoencoders in Pyro

Deniz Alkan (da626)

03 December 2025

Dataflow programming

- major focus of the course: we cannot escape DAGs
- time travel back to Session 2: CIEL
- Eiko: can it run Monte Carlo simulations?
- Deniz: hmm[1]..., yes, no, no, yes, maybe, I don't know, yes
- yes... if we provide the randomness via a seed given as input
- but what if we did randomized computations a completely different way?
- inference *built in* to our program?

[1] Not Hidden Markov Model

Probabilistic Programming

- Deep Learning: complicated neural networks for highly non-linear relationships between data.
- learning? **Optimization** with backpropagation (*automatic differentiation*).
- Probabilistic Machine Learning: probabilistic models relationships between variables.
- learning? **Inference** & optimization with *probabilistic programming*.

Pyro: a probabilistic programming language (PPL)

- NumPy alone isn't enough for running complicated inference tasks
- We would need to calculate the KL divergence, the ELBO, run optimization algorithms all by hand-coding the math.
- Pyro [2], built on PyTorch, provides a framework to automatically track relationships between observed and hidden variables, run inference and optimization with various architectures.
- Why Pyro over NumPyro, TensorFlow Probability, PyMC, etc.? Most flexibility for deep generative models — what I will be using Pyro for!

VAEs

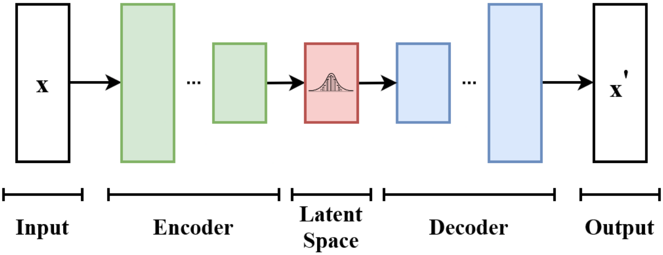

Figure: VAE Architecture[2]

We can also impose conditions such as group-equivariance on the input, output, and latent spaces for structured representation learning. [1]

[2]By EugenioTL - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=107231101

Aim & Plan

1. Implement simple VAE architecture on Pyro (prototype of existing work)
2. Impose group-equivariant and invariant structure over the input, latent, and output spaces in Pyro
3. Benchmark using the MNIST archetype, and extend to large datasets: OpenF1 (goal: strategy generation in Formula 1)

- Aim: Use probabilistic programming to conduct representation learning

Relation to my research

My research: Using *Quantum Entanglement Measures* for Disentangled **Representation Learning** with **Generative Models**

- Interested in how group-equivariant structures between input, output, latent spaces give rise to (dis)entangled group representations in the latent space.
- Example (group-equivariance/invariance): is a cat still a cat if its upside down?
- Example (disentanglement): does changing the color of a car change its shape?
- How can we apply quantum entanglement measures to our latent space representations and encourage learning disentangled representations?

Bibliography I

[1] Riccardo Ali, Pietro Liò, and Jamie Vicary. *Parameter-free approximate equivariance for tasks with finite group symmetry*. arXiv:2506.08244 [cs]. June 2025. DOI: 10.48550/arXiv.2506.08244. URL: http://arxiv.org/abs/2506.08244 (visited on Nov. 12, 2025).

[2] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. "Pyro: Deep Universal Probabilistic Programming". In: *Journal of Machine Learning Research* (2018).