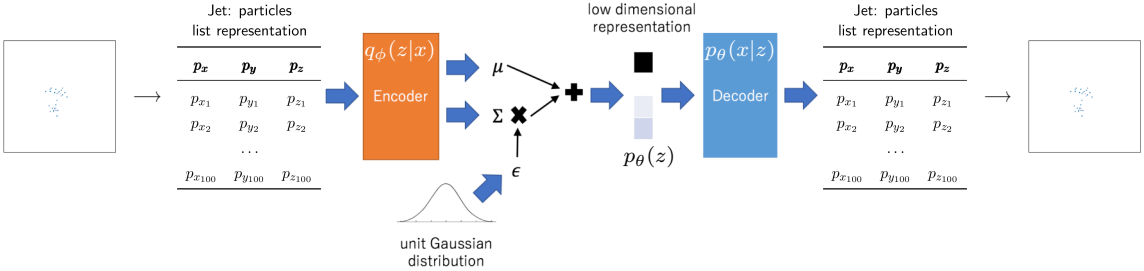
Normalizing Flows

Breno Orzari

Sprace

Recap on VAE



- Its main feature is the capability of generating new outputs through the encoding of the training inputs into a low dimensional representation
- Variational inference is used to impose the form of the encoded distribution of input data

VAE loss function

- $L(\theta, \phi) = -E_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log \ p_\theta(\mathbf{x}|\mathbf{z})] + D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$
  - First term is the reconstruction loss
    - MSE, cross entropy, etc.
  - Second term is KL divergence
    - $q_\phi(\mathbf{z}|\mathbf{x})$ is taken to be multivariate gaussians
    - $p_\theta(\mathbf{z})$ is taken to be a set of standard gaussian
    - Act as a regularizer
- There is no guarantee that assuming gaussians is the best approach for every type of data
  - A way to "transform" this assumption is necessary, that is also not computationally expensive
    - Normalizing flows

Normalizing flows for variational inference (arXiv:1505.05770)

- Application of a invertible transformation (flow) K times in a row over the latent vector
  - Transformation have learnable parameters that will be optimized to improve generation of data
    - Flexibilizes the prior to improve generation capability of the network
- For just one transformation $f : \mathbb{R}^D \rightarrow \mathbb{R}^D$ with inverse transformation $f^{-1} = g$ such that $g \circ f(\mathbf{z}) = \mathbf{z}$
  - The resulting random variable $\mathbf{z}' = f(\mathbf{z})$ has a distribution
    - $q(\mathbf{z}') = q(\mathbf{z}) \left| \det \frac{\partial f}{\partial \mathbf{z}} \right|^{-1}$
- Considering the f transformation composed K times
  - $\mathbf{z}_K = f_K \circ \cdots \circ f_1(\mathbf{z}_0)$
    - $q_K(\mathbf{z}_K) = q_0(\mathbf{z}_0) \prod_{k=1}^{K} \left| \det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \right|^{-1}$

Normalizing flows for variational inference

- $L(\theta, \phi) = E_{q_0(\mathbf{z}_0)}\left[\log\ q_0(\mathbf{z}_0)\right] - E_{q_0(\mathbf{z}_0)}[\log p(\mathbf{x}, \mathbf{z}_K)] - E_{q_0(\mathbf{z}_0)}\left[\sum_{k=1}^{K} \log \left|\det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}}\right|\right]$
  - Using that $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$
    - $L(\theta, \phi) = -E_{q_0(\mathbf{z}_0)}\left[\log\ p(\mathbf{x}|\mathbf{z}_K)\right] + D_{KL}(q_0(\mathbf{z}_0)||p(\mathbf{z}_K)) - E_{q_0(\mathbf{z}_0)}\left[\sum_{k=1}^{K} \log \left|\det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}}\right|\right]$
    - First term is the reconstruction loss
    - Second term is the KL divergence*
    - Third term is the "correction" from the flows
- One of the most common types of flows is the linear flow
  - $f(\mathbf{z}) = \mathbf{z} + \mathbf{u}h(\mathbf{w}^T\mathbf{z} + b)$
    - $\mathbf{u}, \mathbf{w}$ and b are free parameters

Tests using flows

- Types of flows
  - Planar ($\mathbf{u}$ and $\mathbf{w}$ are vectors; b is scalar)
  - Orthogonal Sylvester ($\mathbf{u}$ and $\mathbf{w}$ are matrices parametrized by $\mathbf{R}$, $\widetilde{\mathbf{R}} \rightarrow$ triangular and $\mathbf{Q} \rightarrow$ orthogonal; $\mathbf{b}$ is vector)
  - Triangular Sylvester
  - Householder Sylvester
  - IAF ($\mathbf{z}' = \mu' + \sigma' \odot \mathbf{z}$, where μ' and σ' are the outputs of an autorregressive NN)
- Tests using a gaussian distribution to generate 3×100 matrices
  - Tested each flow using the same common hyperparameters
  - Value of the features is very important
    - Might change the way we are using particles features
  - Smaller learning rate (need to be sure of this)
  - Comparing D_{KL} of gen and input distributions
    - $D_{KL} = \sum_i D^i_{KL}(p_{gen}||p_{input})$, where i is each of the 3 "features"

Tests with gaussian with $\mu = 1.0$, $\sigma = 0.5$

n_{flows}	Planar	IAF	Householder	Triangular	Orthogonal
0	0.2431	0.2431	0.2431	0.2431	0.2431
1	0.1355	0.2206	0.1610	0.2343	0.2185
2	0.1190	0.2970	0.1970	0.1676	0.1830
5	0.2572	0.3071	0.1517	0.1324	0.1606
10	0.1185	0.3446	0.1171	0.1817	0.1086
15	0.1448	0.3247	0.1716	0.1591	0.2025
20	0.1180	0.2566	0.1323	0.1426	0.1692
30	0.1130	0.2156	0.1149	0.1639	0.1749
40	0.1085	0.4205	0.1356	0.1387	0.1369
60	0.1131	0.2532	0.1714	0.1180	0.1795
80	0.1487	0.1437	0.1442	0.1257	0.1389

Tests with gaussian with $\mu = 1.0$, $\sigma = 0.5$: qualitative performance