Normalizing Flows/Autoregressive Models literature review

Mathematical background:

Given Data $X = \{x_1, \dots x_N\}$, maximize marginal log-likelihood $lnp(X) = \sum_{i=1}^{N} lnp(x_i)$

Apply variational inference and optimize the following lower bound:

$$lnp(X) \geq E_{q(Z|x)}[lnp(x|z)] - KL(q(z|x)||p(z))$$

Normalizing flow:

Start with an initial random variable with a simple distribution for generating $z^{(0)}$ and then apply a series of invertible transformation/mapping $f^{(t)}$ for $t = 1, \dots, T$. The last iteration givens a random variable $z^{(T)}$ that has a more flexible distribution. Once we choose transformation $f^{(t)}$ for which the Jacobian-determinant can be computed, the following lower bound turns into:

$$q(z') = q(z)\left|\det \frac{\partial f^{-1}}{\partial z'}\right| = q(z)\left|\det \frac{\partial f}{\partial z'}\right|^{-1}, z^{(T)} = f^{(T)} \cdot \dots \cdot f^{(1)} z^{(0)}$$

$$lnq\left(z^{(T)}|x\right) = lnq\left(z^{(0)}|x\right) - \sum_{t=1}^{T} \ln |det \frac{\partial f^{(t)}}{\partial z^{(t-1)}}|$$

$$lnp(X) \geq E_{q\left(z^{(0)}|x\right)}\left[lnp\left(x|z^{(T)}\right) + \sum_{t=1}^{T} \ln |det \frac{\partial f^{(t)}}{\partial z^{(t-1)}}|\right] - KL(q(z^{(0)}|x)||p(z^{(T)}))$$

Based on the above lower bound, normalizing flows can be divided into two kinds:

(1) General normalizing flows: formulate the flow for which the Jacobian-determinant can be relatively easy to compute

(2) Volume-preserving flows: formulate the flow such that the Jacobian-determinant equals 1 while allows flexible posterior distributions

[1] gives two simple flows form with invertible linear-time transformations (General normalizing flows)

- Planar flows:

  $f(z) = z + uh(w^T z + b), h(\cdot)\ has\ derivative\ h'(\cdot)\ and\ \psi(z) = h'(w^T z + b)w$

  $$\left|det \frac{\partial f}{\partial z}\right| = |1 + u^T \psi(z)|$$

- Radial flows:

  $f(z) = z + \beta h(a,r)(z - z_0), \left|det \frac{\partial f}{\partial z}\right| = [1 + \beta h(a,r)]^{d-1}[1 + \beta h(a,r) + \beta h'(a,r)r],$

  where $r = |z - z_0|, h(a,r) = 1/(a + r)$

Since not all functions of flows above are invertible, there are conditions for invertibility as examples:

For planar flows, if $h(x) = \tanh(x)$, then $f(z)$ should have $w^T u \geq -1$ to be invertible.

(check Appendix for planar flow invertibility normalization and Radial invertibility condition)

[2] Sylvester flows gives a generalization of planar flows to make a single transformation much more flexible.

Simple flows can be efficient for small problems but require a long chain of transformations for high-dimensional dependencies and may result in sub-optimal performance.

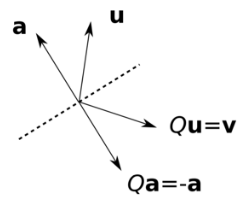
- Sylvester Normalizing flows: (a single layer with M hidden units)

  $$z' = z + Ah(Bz + b), A \in R^{D \times M}, B \in R^{M \times D}, b \in R^M\ and\ M \leq D$$

  Generally, the transformation above may not be invertible. Check special invertible case.

[3] and [4] give Householder transformations and corresponding flows (Volume-preserving flows)

[3] gives intuitive geometric explanation for reflection matrix:



Reflection matrix Q here sends a chosen axis vector a to its negative and reflects all over vectors through the hyperplane perpendicular to a. It has been proved that $Q = I - 2\frac{aa^t}{a^t a}$ and Q is also orthogonal.

[4] gives Householder flows inspired by Reflection matrix:

(Transform an isotropic Gaussian (covariance matrix represented by the simplified matrix $\Sigma = \sigma^2 I$) into a non-isotropic Gaussian $N(u, \Sigma)$)

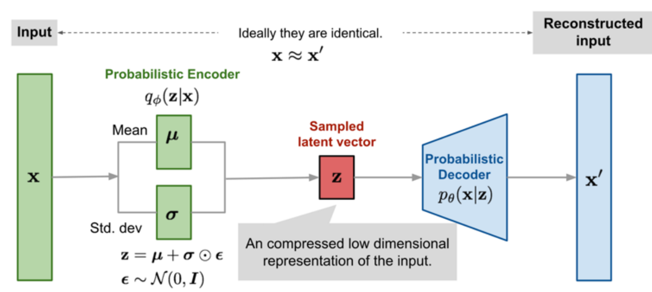
Given that any full covariance matrix Σ can be decomposed into $\Sigma = UDU^T$, where D is diagonal and U is orthogonal and $z^{(1)} = UZ^{(0)}, z^{(1)} \sim N(Uu, Udiag(\sigma^2)U^T)$, if we are able to model U and $diag(\sigma^2)$ coincides with true D, then we can resemble true covariance matrix Σ.

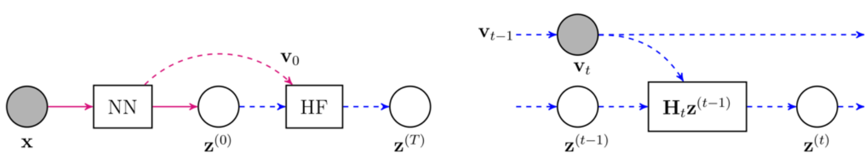
Model U through a sequence of Householder transformations: (Theorem 2)

$z^{(t)} = \left(I - 2\frac{v_t v_t^T}{||v_t||^2}\right) z^{(t-1)} = H_t z^{(t-1)}$, and orthogonal H_t and its absolute value of Jacobian-determinant is 1.

The encoder network NN would generate means and variances for the posterior and the first Householder vector v_0 to formulate Householder flow.

GitHub: https://github.com/jmtomczak/vae_householder_flow





(a) encoder network + Householder Flow (b) one step of the Householder Flow

Normalizing flows and autoregressive models can be combined together for density estimation:

[5] uses "masks" to make the output be autoregressive for a given ordering of the inputs.

Basic autoencoder: reconstruct x using hidden representation $h(x)$

$$h(x) = g(b + Wx), \hat{x} = sigm\big(c + Vh(x)\big), where\ sigm(a) = 1/(1 + \exp(-a))$$

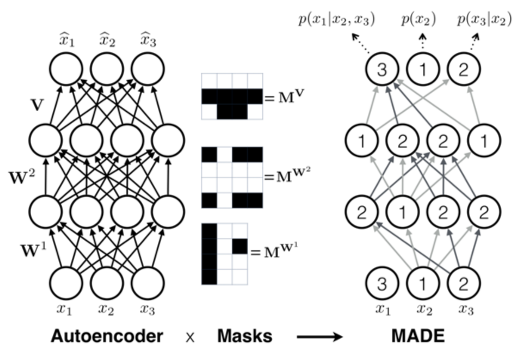
W and V are matrices that represent connections from the input to hidden layers and hidden to output layers.

Masked autoencoders: For a single hidden layer autoencoder under given ordering of inputs

$$h(x) = g(b + (W \odot M^W)x), \hat{x} = sigm\big(c + (V \odot M^V)h(x)\big)$$

here M^W and M^V are binary mask matrices to remove connections.

By permuting the ordering, we are able to generate L>1 hidden layers and formulate the network.



[6] put up Masked Autoregressive flow: implementation of normalizing flow uses MADE as a building block. (improve MADE by modelling the density of its internal random numbers)

$$p(x_i|X_{1:i-1}) = N(x_i|u_i, (expa_i)^2), where\ u_i = f_{u_i}(X_{1:i-1})\ and\ a_i = f_{a_i}(X_{1:i-1})$$

Recursion and reparameterization:

$$x_i = u_i expa_i + \mu_i, where\ \mu_i = f_{\mu_i}(X_{1:i-1})\ and\ a_i = f_{a_i}(X_{1:i-1})\ and\ u_i \sim N(0,1)$$

Then implement f_{μ_i} and f_{a_i} with masking (MADE) and enable transforming from data X to μ_i and a_i

Comparison with Inverse Autoregressive Flows in [7]: (check equivalence section in [6])

$$x_i = u_i expa_i + \mu_i, where\ \mu_i = f_{\mu_i}(\mu_{1:i-1})\ and\ a_i = f_{a_i}(u_{1:i-1})\ and\ u_i \sim N(0,1)$$

[7] implements f_{μ_i} and f_{a_i} with masking (MADE) and enable transforming from previous $u_{1:i-1}$

[8] gives a volume-preserving flow: (enrich linear Inverse Autoregressive Flow)

GitHub: https://github.com/jmtomczak/vae_vpflows

idea: $z^{(1)} = Lz^{(0)}$ and the Jacobian-determinant of L must be 1

possible options for choosing L:

(1) Orthogonal matrix as Householder flow (HF)

(2) Lower-triangular matrix with 1 on the diagonal as Linear inverse autoregressive flow

Further represent variation in data, introduce K such matrices $\{L_1(x), \dots, L_K(x)\}$ and operate linear transformation through convex combination: (property: $|\det(\sum_{k=1}^{K} y_k(x) L_k(x))| = 1$)

$$z^{(1)} = \left(\sum_{k=1}^{K} y_k(x) L_k(x)\right) z^{(0)}, y(x) = Softmax\big(NN(x)\big) = [y_1(x), \dots y_K(x)]^T$$

Further flows with complicated neural structure or higher-level dynamics: [9] and [10]

Reference:

[1] Rezende, Danilo Jimenez, and Shakir Mohamed. "Variational inference with normalizing flows." *arXiv preprint arXiv:1505.05770* (2015).
[2] Berg, Rianne van den, et al. "Sylvester normalizing flows for variational inference." *arXiv preprint arXiv:1803.05649* (2018).
[3] Kerl, John. "The Householder transformation in numerical linear algebra." *Rapport technique, University of Arizona* (2008): 18.
[4] Tomczak, Jakub M., and Max Welling. "Improving variational auto-encoders using householder flow." *arXiv preprint arXiv:1611.09630* (2016).
[5] Germain, Mathieu, et al. "Made: Masked autoencoder for distribution estimation." *International Conference on Machine Learning*. 2015.
[6] Papamakarios, George, Theo Pavlakou, and Iain Murray. "Masked autoregressive flow for density estimation." *Advances in Neural Information Processing Systems*. 2017.
[7] Kingma, Durk P., et al. "Improved variational inference with inverse autoregressive flow." *Advances in neural information processing systems*. 2016.
[8] Tomczak, Jakub M., and Max Welling. "Improving variational auto-encoders using convex combination linear inverse autoregressive flow." *arXiv preprint arXiv:1706.02326* (2017).
[9] Huang, Chin-Wei, et al. "Neural autoregressive flows." *arXiv preprint arXiv:1804.00779* (2018).
[10] Marino, Joseph, et al. "Improving Sequential Latent Variable Models with Autoregressive Flows." *Symposium on Advances in Approximate Bayesian Inference*. 2020.

Websites for reference:

http://akosiorek.github.io/ml/2018/04/03/norm_flows.html
https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html (VAE figure)