Normalizing flows in tf_utils

Sean Bittner

March 20, 2019

Planar flows [1]:

See section A.1 of [1] for reparameterization guaranteeing invertibility.

$$f(z) = z + h(w^\top z + b)$$

$$\psi(z) = h'(w^\top z + b)w$$

$$\left| \frac{df(z)}{dz} \right| = \left| 1 + u^\top \psi(z) \right|$$

We use $h = \tanh$. The elements of u and b are always initialized to 0, while the initial elements of w are drawn from a glorot uniform initializer.

Radial flows [1]:

See section A.2 of [1] for reparameterization guaranteeing invertibility.

$$f(z) = z + \beta h(\alpha, r)(z - z_0)$$

$$\left| \frac{df(z)}{dz} \right| = [1 + \beta h(\alpha, r)]^{d-1} \left[1 + \beta h(\alpha, r) + \beta h'(\alpha, r)r \right]$$

where $h(\alpha, r) = \frac{1}{\alpha + r}$, $r = |z - z_0|$, and $\alpha > 0$. A log-parameterization of α is used to enforce the constraint. $\log(\alpha)$, β, and z_0 all have the glorot uniform initializer.

Real NVP [2]:

In a real NVP (non-volume preserving) normalizing flow, there is a sequence of "masking" layers which parameterize bijective transformations until an output is produced. In each "masking" layer, a set of entries are chosen to be passed unchanged to the next masking layer. The remaining entries are scaled and shifted by arbitrary continuous differentiable functions of the unchanged elements.

For $b \in [0, 1]^D$, let x_{I_b} be the elements of arbitrary vector $x \in \mathcal{R}^D$, which have a 1 in the corresponding index of b. Similarly, let $x_{I_b^C}$ be the remaining elements of x (those with corresponding indices of b equal to 0). In a single masking layer of real NVP, we have the following transformation for mask b:

$$f(z)_{I_b} = z_{I_b}$$

$$f(z)_{I_b^C} = z_{I_b^C} \exp(s(z_{I_b})) + t(z_{I_b})$$

Ordering the elements of z such that the passed elements ($b_i = 1$) are at the early indices, and letting $d = \sum_i b_i$,

$$\frac{df(z)}{dz} = \begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{df(z)_{I_b^C}}{dz_{I_b}} & \text{diag}(\exp[s(z_{I_b})]) \end{bmatrix}$$

The log determinant jacobian of this masking layer is then

$$\frac{df(z)}{dz} = \sum_j s(z_{I_b})_j$$

The real NVP normalizing flow is constructed by a cascade of different masks. In common practice, a mask of a particular pattern is always followed by it's complement pattern. We adopt this pattern, and thus, our real NVP flows always have an even number of masks. We identify masks by their number of elements - D, the frequency of bit flip - f, and state of the first element. Below are some examples:

Sean Bittner
srb2201@columbia.edu

D	f	first elem	mask
8	1	1	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^\top$
8	1	0	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^\top$
8	4	1	$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^\top$

For a selection of $2k$ masks for a D dimensional distribution, we use the following sequence of masks: $f = 1, b_1 = 1$

$f = 1, b_1 = 0$

$f = \frac{D}{2}, b_1 = 1$

$f = \frac{D}{2}, b_1 = 0$

$f = 2, b_1 = 1$

$f = 2, b_1 = 0$

$f = \frac{D}{4}, b_1 = 1$

$f = \frac{D}{4}, b_1 = 0$

$f = 4, b_1 = 1$

$f = 4, b_1 = 0$

...

The parametization of functions s and t are L-layer fully-connected neural networks with U-units per layer.

Elementwise multiplication flows:

$$f(z) = a \circ z$$

$$\log\left(\left|\frac{df(z)}{dz}\right|\right) = \sum_{i=1}^{D} \log(|a_i|)$$

Shift flows:

$$f(z) = z + b$$

$$\log\left(\left|\frac{df(z)}{dz}\right|\right) = 0$$

Cholesky Product Flows:

We introduce a normalizing flow for constructing positive semi-definite matrices. This is done by parameterizing the entries of the Cholesky factor, with entries exponentiated along the diagonal.

$$f(z) = \text{vec}(L(z)L(z)^\top)$$

$$L(z) = \begin{bmatrix} e^{z_1} & 0 & \dots & 0 \\ z_2 & e^{z_3} & & \\ z_4 & z_5 & e^{z_6} & \\ \dots & & & \end{bmatrix}$$

$$\text{diag}(z) = \begin{bmatrix} z_1 \\ z_3 \\ z_6 \\ .. \end{bmatrix}, \text{diag}(L(z)) = \begin{bmatrix} e^{z_1} \\ e^{z_3} \\ e^{z_6} \\ .. \end{bmatrix}$$

Sean Bittner
srb2201@columbia.edu

$$vec(M) = \begin{bmatrix} M_{11} \\ M_{21} \\ M_{22} \\ M_{31} \\ ... \\ M_{DD} \end{bmatrix}$$

$$\left| \frac{dL(z)}{dz} \right| = \prod_{d=1}^{D} \text{diag}(L(z))_d = \prod_{d=1}^{D} e^{\text{diag}(z)_d}$$

$$\log(\left| \frac{dL(z)}{dz} \right|) = \log(\prod_{d=1}^{D} e^{\text{diag}(z)_d}) = \sum_{d=1}^{D} \text{diag}(z)_d$$

$$\Sigma(z) = L(z)L(z)^\top$$

$$\left| \frac{d\text{vec}(\Sigma(z))}{dL(z)} \right| = 2^D \prod_{d=1}^{D} L_{dd}^{D-d+1}$$

$$\log(\left| \frac{d\text{vec}(\Sigma(z))}{dL(z)} \right|) = D\log(2) \sum_{d=1}^{D} (D - d + 1)\log(\text{diag}(L(z))_d)$$

$$= D\log(2) \sum_{d=1}^{D} (D - d + 1)\text{diag}(z)_d$$

$$\log(\left| \frac{df(z)}{dz} \right|) = \log(\left| \frac{d\text{vec}(\Sigma(z))}{dL(z)} \frac{dL(z)}{dz} \right|) = \log(\left| \frac{d\text{vec}(\Sigma(z))}{dL(z)} \right| \left| \frac{dL(z)}{dz} \right|) = \log(\left| \frac{d\text{vec}(\Sigma(z))}{dL(z)} \right|) + \log(\left| \frac{dL(z)}{dz} \right|)$$

References

[1] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[2] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.