Normalizing Flows: Planar Flow and Radial Flow

Pratikhya Ranjit, 190639 Rishabh Katiyar, 190702

January 29, 2022

1 Introduction

Normalizing Flows are generative models used to model probability distribution given the samples from the distribution. Normalizing flows transform easy-to-sample probability distributions into complex distributions using invertible transformation functions. This flow of probability mass function from one distribution to another justifies the name 'normalizing flows'. Two popular normalizing flows are planar flows and radial flows.

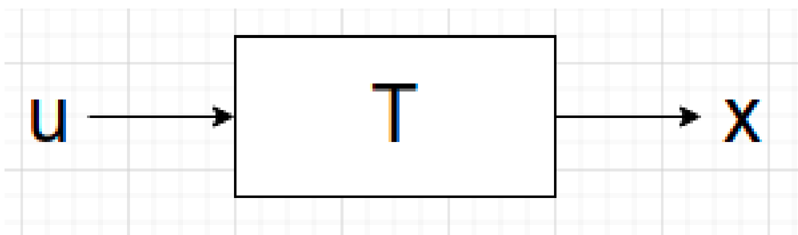

Figure 1: Represents $x = T(u)$. T represents the transformation function & can be linear or non-linear.

2 Training NF Model

The loss function for a normalising flow model is calculated using KL divergence between the estimated probability distribution $p_X(x;\theta)$ and the true distribution $p_X^*(x)$. Since it is difficult to calculate the integral of such functions, we use Monte-Carlo approximations to compute the loss. Thus, the loss function $L(\theta)$ is given by

$$L(\theta) = E_{x\sim p_X^*(x)}[log(p_X^*(x)) - log(p_X(x;\theta))]$$

$$= -E_{x\sim p_X^*(x)}[log(p_U(U;\psi)) + log|det(J_{T^{-1}}(x;\phi))|] \qquad where \qquad U = T^{-1}(X;\phi)$$

Here, $\theta = \{\phi, \psi\}$ are the set of learnable parameters where ϕ comes from transformation T and ψ is a set parameters present in U. $p_X^*(x)$ represents the true distribution and $p_X(x;\theta)$ represents the NF model.[PNR+21]

Using Monte-Carlo approximations, we compute the expectation value by sampling from the true distribution and hence compute the loss for parameters θ.

To construct $L(\theta)$ we need:

i Samples from true distribution $p_X^*(x)$

ii Inverse transformation T

iii Distribution $p_U(u)$

iv $log|det(J_{T^{-1}}(x;\phi))|$

We reconstruct the loss function by interchanging $p^*_X(x)$ and $p_X(x;\theta)$ using the definition of KL divergence. Therefore,

$$L(\theta) = D_{KL}[p_X(x;\theta)||p^*_X(x)]$$

$$= E_{x\sim p_X(x;\theta)}[log(p_X(x;\theta)) - log(p^*_X(x))]$$

Now, for the transformation $X = T(U)$, we have

$$(1) dX = |det(J_T(U))|dU$$

$$(2) p_X(x) * |det(J_T(U))| = p_U(u)$$

Using (1) and (2) we get

$$\int p_X(T(U)) * log[\frac{p(T(U))}{q(T(U))}]dX = \int p_X(T(U)) * log[\frac{p(T(U))}{q(T(U))}] * |det(J_T(U))|dU$$

$$= \int p_U(u) * log[\frac{p(T(U))}{q(T(U))}]dU$$

$$= E_{U\sim p_U(u)}[log[\frac{p(T(U))}{q(T(U))}]]$$

Therefore, $L(\theta)$ can be written as

$$L(\theta) = E_{U\sim p_U(u;\psi)}[log(p_X(x;\theta)) - log(p^*_X(x))]$$

We substitute (I) and (II) given below in the expression of $L(\theta)$.

I $log(p_X(x;\theta)) = log(p_U(u,\psi)) + log|det(J_T(u;\theta))|^{-1}$

II $log(p^*_X(x))$ can be computed by sampling u & the true distribution $p^*_X(x)$

To compute $L(\theta)$ we need :

i Distribution $p_U(u;\psi)$ to compute samples

ii $log|det(J_T(u;\theta))|$

iii The transformation $T(u)$

iv True distribution $p^*_X(x)$

Thus, by using KL divergence we compute $L(\theta)$ and then train our NF model.

3 Planar Flow

The transformation function for planar flow is given by

$$X = T(U) = U + C * h(W^T U + b) \tag{1}$$

Here X & U are vectors. Let's say U is a D-dimensional vector; C & W are also D-dimensional vectors and b is a scalar. C,W & b are learnable parameters and h is a non linear function which could be a tanh or a sigmoid function.

The total number of learnable parameters in this transformation are 2D + 1.

This is the forward map T(U). The inverse map $T^{-1}(X)$ is non - trivial to find here and can be approximated.
Computation of the Jacobian which is needed in computing the loss function is as follows

$$x_i = u_i + c_i * h(\sum_{k=1}^{D} w_k u_k + b) \tag{2}$$

$$J_{ji} = \frac{\partial x_i}{\partial u_j} = \delta_{ij} + c_i * h'(W^T U + b) * w_j \tag{3}$$

$$J = I + h'(W^T U + b) W C^T \tag{4}$$

Computing the determinant of the Jacobian will require Sylvester's identity for computing determinants which states that

$$det(I_n + AB) = det(I_m + BA) \tag{5}$$

In the above equation I_n is n x n Identity vector, A is n x m dimensional vector, B is m x n dimensional vector and I_m is m x m Identity vector.

$$det(J) = |I + h'(W^T U + b) W C^T| \tag{6}$$

On applying the Sylvester's Identity the determinant becomes

$$det(J) = |1 + h'(W^T U + b) C^T W| \tag{7}$$

C^T is 1xD dimensional vector and W is Dx1 dimensional, hence $C^T W$ becomes a scalar and the overall determinant becomes really easy to calculate.

4 Radial Flow

The transformation function for radial flow is given by

$$X = T(U) = U + \frac{\beta * (U - U_0)}{\alpha + r} \tag{8}$$

U & U_0 are D-dimensional vectors. α & β are scalars. r is the *L2* norm between U & U_0. α, β & U_0 are the learnable parameters and the total number is D + 2.

Calculating the inverse of the transformation is difficult and we use approximation to compute the same.

Computation of the Jacobian which is needed in computing the loss function is as follows

$$x_i = u_i + \frac{\beta * (u_i - u_{0i})}{\alpha + \sqrt{\sum_{k=1}^{D} (u_k - u_{0k})^2}} \tag{9}$$

$$J_{ji} = \frac{\partial x_i}{\partial u_j} = \delta_{ij} + \frac{\beta * \delta_{ij}}{\alpha + r} - \frac{\beta * (u_i - u_{0i})}{(\alpha + r)^2} * \frac{2(u_j - u_{0j})}{2r} \tag{10}$$

$$J = I(1 + \frac{\beta}{\alpha + r}) - \frac{\beta (U - U_0)(U - U_0)^T}{(\alpha + r)^2 r} \tag{11}$$

Using another property of determinant in *Eq.(12)* along with Sylvester's Identity

$$det(\alpha A_{nxn}) = \alpha^n det(A_{nxn}) \tag{12}$$

$$J = (1 + \frac{\beta}{\alpha + r}) * (I - \frac{\beta * (U - U_0)(U - U_0)^T}{(\alpha + r)^2 * r * (1 + \frac{\beta}{\alpha + r})}) \tag{13}$$

$$det(J) = (1 + \frac{\beta}{\alpha + r})^D * (1 - \frac{\beta * (U - U_0)^T (U - U_0)}{(\alpha + r)^2 r (1 + \frac{\beta}{\alpha + r})}) \tag{14}$$

$$det(J) = (1 + \frac{\beta}{\alpha + r})^D * (1 - \frac{\beta * r^2}{(\alpha + r)^2 r (1 + \frac{\beta}{\alpha + r})}) \tag{15}$$

$$det(J) = (1 + \frac{\beta}{\alpha + r})^{D-1} * (1 + \frac{\alpha * \beta}{(\alpha + r)^2}) \tag{16}$$

Thus the complexity of calculating det(J) is O(D).

All the equations derived so far can be substituted suitably to calculate the loss function and train the NF model.

References

[PNR+21] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference, 2021.