Zürich, July 2024

Intro to normalizing flows

(with Python examples)

Kamila Zdybał

kamilazdybal.github.io, kamila.zdybal@gmail.com

Please feel free to contact me with any suggestions, corrections or comments.

Preface

Keywords

normalizing flow | *probability density* | *Python*

1 Transformations between random variables

We begin with a preliminary example of transforming one probability density into another which will give the necessary intuition to move on towards understanding normalizing flows.

1.1 A simple example

Consider a random variable X sampled from a Gaussian normal distribution with zero mean and standard deviation equal to 1:

$$X \in \mathcal{N}(0, 1) \,. \tag{1}$$

We would like to transform samples of X using some nonlinear, invertible transformation, T. Hence, to go from $X \to Y$, we apply T:

$$Y = T(X) \tag{2}$$

Equivalently, since T is invertible, we can also go from $Y \to X$ in the following way:

$$X = T^{-1}(Y) \tag{3}$$

An example transformation that we will use here is the exponential function $T(\bullet) = \exp(\bullet)$, whose inverse is $T^{-1}(\bullet) = \ln(\bullet)$. Fig. 1 shows histograms of X and Y from this example, where we initially generated 10,000 samples of X.

Let's denote the distribution of X as p_X and the distribution of Y as p_Y. We now pose a following question: Given p_X and T, what is p_Y? It turns out that p_Y can be computed analytically using the change of variables formula:

$$p_Y(Y) = p_X(X) \frac{1}{|\text{det} J_T(X)|} \,, \tag{4}$$

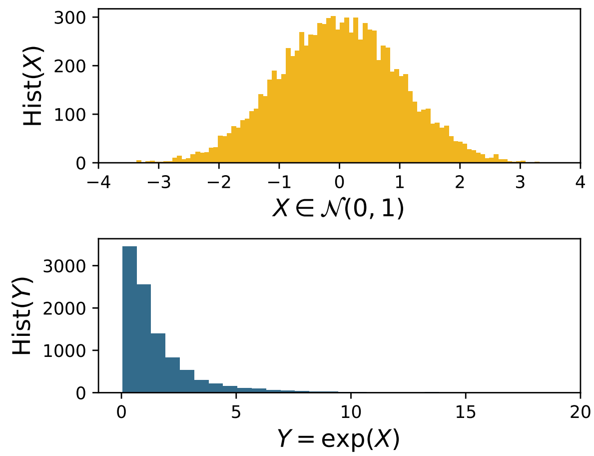

Figure 1: Histograms of two random variables, X and Y.

where $J_T(X)$ is the Jacobian of the transformation T. Eq. (4) is equivalent to

$$p_Y(Y) = p_X(T^{-1}(Y)) |\text{det} J_{T^{-1}}(Y)| \tag{5}$$

since $X = T^{-1}(Y)$ and, perhaps less trivially, $|\text{det} J_{T^{-1}}(Y)| = \frac{1}{|\text{det} J_T(X)|}$. In the case of an exponential function, the Jacobian of the transformation can be computed as

$$J_T(X) = \frac{d(\exp(X))}{dX} = \exp(X) \,. \tag{6}$$

Similarly, $J_{T^{-1}}(Y)$ is the Jacobian of the inverse transformation which in this case is

$$J_{T^{-1}}(Y) = \frac{d(\ln(Y))}{dY} = 1/Y \,. \tag{7}$$

Fig. 2 shows probability densities for X and Y, where p_X is computed as

$$p_X(X) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(X-\mu)^2}{2\sigma^2} \right) \tag{8}$$

with $\sigma = 1$ and $\mu = 0$, and p_Y is computed either from Eq. (4) or Eq. (5).

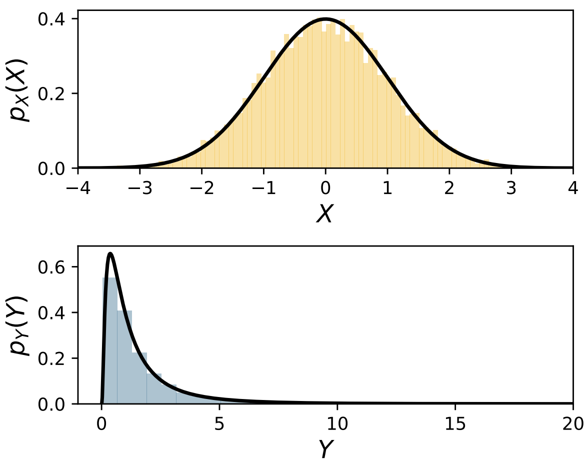

Figure 2: Probability density functions (PDFs) of two random variables, X and Y, where $X \in \mathcal{N}(0,1)$ and $Y = \exp(X)$.

1.2 Python computation

To compute p_Y using Eq. (4) in Python:

```
def p_transformed(X, p_X):
    p_Y = p_X(X,0,1)*np.abs(1/np.exp(X))
    return p_Y
```

and to compute p_Y using Eq. (5) in Python:

```
def p_transformed(Y, p_X):
    p_Y = p_X(T_inv(Y),0,1)*np.abs(1/Y)
    return p_Y
```

Both computations are equivalent, since $X = T^{-1}(Y)$ and $Y = \exp(X)$.