**CSE6243: Advanced Machine Learning** **Fall 2023**

# Lecture 17: Normalizing Flow Models

*Lecturer: Bo Dai* *Scribes: Ruomeng Ding*

**Note**: *LaTeX template courtesy of UC Berkeley EECS Department.*

## 17.1 Recap

**GANs**.

Sample generator:

$$\begin{aligned} \boldsymbol{\epsilon} &\sim \rho_\theta(\boldsymbol{\epsilon}) \\ \boldsymbol{x} &= G_\theta(\boldsymbol{\epsilon}) \end{aligned} \tag{17.1}$$

Loss function:

$$\min_G \max_D \mathbb{E}_{\boldsymbol{x}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{\epsilon}\sim\rho_\theta(\boldsymbol{\epsilon})}[\log(1 - D(G(\boldsymbol{\epsilon}))] \tag{17.2}$$

where $G(\cdot)$ denotes the generator and $D(\cdot)$ denotes the discriminator.

GANs do not require the computation of probability of $\boldsymbol{x}$. Today, we introduce Normalizing Flows, which enable the calculation of probability.

## 17.2 New Content

### 17.2.1 Basic: Change of Variable Theorem

Given $\boldsymbol{\epsilon} \in \mathbb{R}$, $\boldsymbol{x} = \mathbf{G}(\boldsymbol{\epsilon})$, $\boldsymbol{\epsilon} = \mathbf{G}^{-1}(\boldsymbol{x})$. The function $\mathbf{G}(\cdot)$ is invertible. We can obtain :

$$\begin{aligned} p(\boldsymbol{\epsilon})\mathrm{d}\boldsymbol{\epsilon} &= p(\mathbf{G}^{-1}(\boldsymbol{x}))\mathrm{d}(\mathbf{G}^{-1}(\boldsymbol{x})) \\ &= p(\mathbf{G}^{-1}(\boldsymbol{x}))(\mathbf{G}^{-1}(\boldsymbol{x}))'\mathrm{d}\boldsymbol{x} \end{aligned} \tag{17.3}$$

Now the question is how to infer the unknown probability density function of $\boldsymbol{x}$?

$$\begin{aligned} \int p(\boldsymbol{\epsilon})\mathrm{d}\boldsymbol{\epsilon} = \int p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1 \\ p(\boldsymbol{x}) = p(\mathbf{G}^{-1}(\boldsymbol{x}))\left|(\mathbf{G}^{-1}(\boldsymbol{x}))'\right| \end{aligned} \tag{17.4}$$

The multivariable version has a similar format:

$$p(\boldsymbol{x}) = p_\theta(\mathbf{G}^{-1}(\boldsymbol{x}))\left|\det \frac{\partial \mathbf{G}^{-1}}{\partial \boldsymbol{x}}\right| \tag{17.5}$$

where $\det \frac{\partial \mathbf{G}^{-1}}{\partial \boldsymbol{x}}$ is the Jacobian determinant of the function $\mathbf{G}^{-1}(\cdot)$.

### 17.2.2 Normalizing Flows

Given $\boldsymbol{\epsilon} \in \rho_\theta(\boldsymbol{\epsilon})$, $\boldsymbol{x} = \mathbf{G}_\theta(\boldsymbol{\epsilon})$,

$$p(\boldsymbol{x}) = p_\theta(\mathbf{G}_\theta^{-1}(\boldsymbol{x})) \left| \det \frac{\partial \mathbf{G}_\theta^{-1}}{\partial \boldsymbol{x}} \right| \tag{17.6}$$

To ensure that we can calculate $p(\boldsymbol{x})$, these four conditions should be satisfied:

[**Condition 1**] $\mathbf{G}^{-1}(\cdot)$ exists. The function $\mathbf{G}(\cdot)$ is invertible.

[**Condition 2**] $\left| \det \frac{\partial \mathbf{G}_\theta^{-1}}{\partial \boldsymbol{x}} \right| \neq 0$.

[**Condition 3**] The Jacobian determinant in [**Condition 2**] need to be easy to compute to decrease the computational cost. For example, the determinant of the diagonal matrix is equal to the product of its diagonal elements

[**Condition 4**]: $\boldsymbol{x} = \mathbf{G}_{\theta_1}\mathbf{G}_{\theta_2}...\mathbf{G}_{\theta_k}$. If any of $\mathbf{G}_{\theta_1}...\mathbf{G}_{\theta_k}$ satisfy [**Condition 1**] and [**Condition 2**], then the composite function $\mathbf{G}_{\theta_1}\mathbf{G}_{\theta_2}...\mathbf{G}_{\theta_k}$ will also fulfill [**Condition 1**] and [**Condition 2**].

To maximize the log-likelihood, in this case we have:

$$\begin{aligned} \max_\theta \log p_\theta(\boldsymbol{x}) &= \max_\theta \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(\mathbf{x_i}) \\ &= \max_\theta \frac{1}{n} \sum_{i=1}^{n} (\log p_\theta(\mathbf{G}_\theta^{-1}(\boldsymbol{x}_i)) + \log \left| \det \frac{\partial \mathbf{G}_\theta^{-1}}{\partial \boldsymbol{x}_i} \right|) \end{aligned} \tag{17.7}$$

**Comparison with GAN**:

There is a delicate balance between the simplicity and the potency of the learning process.

1. When it comes to learning complex data distributions, GANs have demonstrated exceptional results, albeit with stability issues during the learning process.

2. On the other hand, Normalizing flows, due to their inherent constraints, offer more stability than GANs. However, these constraints could potentially result in less flexibility.

There are several concrete examples of Normalizing flows. For all the examples below, we have the assumption that $\boldsymbol{x} \in \mathbb{R}^d$, and $\boldsymbol{\epsilon} \in \mathbb{R}^d$.

### 17.2.3 Linear Flows

Linear mappings can express correlation between dimensions, $\mathbf{A} \in \mathbb{R}^{d \times d}$:

$$\boldsymbol{x} = \mathbf{A}\boldsymbol{\epsilon} + b \tag{17.8}$$

To satisfy the [**Condition 1**] and [**Condition 2**] mentioned in **section 17.3.1**:

$\mathbf{A} \neq 0$. $\mathbf{A}^{-1} \neq 0$. If $\mathbf{A}$ is an invertible matrix, the function is invertible.

If A is diagonal with nonzero diagonal entries, then its inverse can be computed in linear time and its determinant is the product of the diagonal entries.

$$|\det \mathbf{A}^{-1}| = |\det \mathbf{U}\boldsymbol{\lambda}^{-1}\mathbf{V}^{-1}| = (\prod_{i=1}^{n} |\boldsymbol{\lambda}_i|)^{-1} \tag{17.9}$$

where $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{d\times d}$. $\boldsymbol{\lambda}$ is the diagonal matrix. And $A = \mathbf{U}\boldsymbol{\lambda}\mathbf{V}^{-1}$. The result is an elementwise transformation and hence cannot express correlation between dimensions.

### 17.2.4 Planar Flows

Planar flows expand and contract the distribution along certain specific directions and take the form

$$\boldsymbol{x} = \boldsymbol{\epsilon} + \mathbf{u}h\left(\mathbf{w}^T\boldsymbol{x} + b\right) \tag{17.10}$$

where $\mathbf{G}_\theta(\boldsymbol{\epsilon}) = \mathbf{u}h\left(\mathbf{w}^T\boldsymbol{x} + b\right)$. $\mathbf{u}, \mathbf{w} \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}$ are parameters.$h : \mathbb{R} \to \mathbb{R}$ is a smooth non-linearity. The Jacobian determinant is

$$\begin{aligned} \det\left|\frac{\partial \mathbf{G}}{\partial \boldsymbol{\epsilon}}\right| &= \det\left|\mathbf{I} + \mathbf{u}h'\left(\mathbf{w}^T\boldsymbol{\epsilon} + b\right)\mathbf{w}^T\right| \\ &= |\det(\mathbf{I} + a\mathbf{u}\mathbf{w}^T)| \\ &= |1 + a\mathbf{w}^T\mathbf{u}| \end{aligned} \tag{17.11}$$

where $a = h'\left(\mathbf{w}^T\boldsymbol{\epsilon} + b\right) \in \mathbb{R}$. And $a\mathbf{w}^T\mathbf{u} \neq 1$. $\mathbf{u}h\left(\mathbf{w}^T\boldsymbol{x} + b\right)$ can be understood as a hidden layer that contains only one unit. A substantial number of planar flows can be stacked to achieve high expressivity.

### 17.2.5 NICE (Non-linear Independent Component Estimation)

The transformation in **NICE** is the affine coupling layer without the scale term, known as additive coupling layer. the input dimensions are split into two parts:

$$\left\{\begin{array}{ll} \boldsymbol{\epsilon}_1 & = \mathbf{x}_1 \\ \boldsymbol{\epsilon}_2 & = \mathbf{x}_2 - m_\theta(\boldsymbol{x}_1) \end{array}\right. \Leftrightarrow \left\{\begin{array}{ll} \boldsymbol{x}_1 & = \boldsymbol{\epsilon}_1 \\ \boldsymbol{x}_2 & = \boldsymbol{\epsilon}_2 + m_\theta(\boldsymbol{\epsilon}_1) \end{array}\right. \tag{17.12}$$

where $\boldsymbol{\epsilon}_1 \in \mathbb{R}^{d_1}$, $\boldsymbol{\epsilon}_2 \in \mathbb{R}^{d_2}$. And $\boldsymbol{x}_1 \in \mathbb{R}^{d_1}$, $\boldsymbol{x}_2 \in \mathbb{R}^{d_2}$. $d_1 + d_2 = d$. $d_1 + d_2 = d$. $\mathbf{m}(\cdot)$ is a scale and translation function. The Jacobian determinant is

$$\det\left|\frac{\partial \mathbf{G}^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right| = \begin{bmatrix} \frac{\partial \boldsymbol{\epsilon}_1}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{\epsilon}_1}{\partial \boldsymbol{x}_2} \\ \frac{\partial \boldsymbol{\epsilon}_2}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{\epsilon}_2}{\partial \boldsymbol{x}_2} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -m_\theta\prime(\mathbf{x_1}) & \mathbf{I} \end{bmatrix} = 1 \tag{17.13}$$

### 17.2.6 RealNVP (Real-valued Non-Volume Preserving)

**RealNVP** employs a normalizing flow by sequentially stacking a series of reversible bijective transformation functions. Similar to **NICE**, the input dimensions are divided into two sections.

$$\left\{\begin{array}{ll} \boldsymbol{\epsilon}_1 & = \mathbf{x}_1 \\ \boldsymbol{\epsilon}_2 & = \mathbf{x}_2 - m_\theta(\boldsymbol{x}_1) \odot \exp(-s_\phi(\boldsymbol{x}_1)) \end{array}\right. \Leftrightarrow \left\{\begin{array}{ll} \boldsymbol{x}_1 & = \boldsymbol{\epsilon}_1 \\ \boldsymbol{x}_2 & = \boldsymbol{\epsilon}_2 \odot \exp(s_\phi(\boldsymbol{x}_1)) + m_\theta(\boldsymbol{\epsilon}_1) \end{array}\right. \tag{17.14}$$

where $\boldsymbol{\epsilon}_1 \in \mathbb{R}^{d_1}$, $\boldsymbol{\epsilon}_2 \in \mathbb{R}^{d_2}$. And $\boldsymbol{x}_1 \in \mathbb{R}^{d_1}$, $\boldsymbol{x}_2 \in \mathbb{R}^{d_2}$. $d_1 + d_2 = d$. $\mathbf{s}(\cdot)$ and $\mathbf{m}(\cdot)$ are scale and translation functions. The $\odot$ operation is the element-wise product. The Jacobian determinant is

$$\det \left| \frac{\partial \mathbf{G}^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right| = \begin{bmatrix} \frac{\partial \boldsymbol{\epsilon}_1}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{\epsilon}_1}{\partial \boldsymbol{x}_2} \\ \frac{\partial \boldsymbol{\epsilon}_2}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{\epsilon}_2}{\partial \boldsymbol{x}_2} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -m_{\theta}\prime(\mathbf{x_1}) & \exp(s_{\phi}(\boldsymbol{\epsilon}_1)) \end{bmatrix} \tag{17.15}$$

### 17.2.7 Residual Flows

Residual Flows combines **NICE** and **RealNVP**. To be specific,

$$\boldsymbol{x} = \boldsymbol{\epsilon} + m_{\theta}(\boldsymbol{\epsilon}) \tag{17.16}$$

The Jacobian determinant is

$$\begin{aligned} \left| \det \frac{\partial \mathbf{G}}{\partial \boldsymbol{\epsilon}} \right| &= |\det(\mathbf{I} + m_{\theta}\prime(\boldsymbol{\epsilon}))| \\ \log \left| \det \frac{\partial \mathbf{G}}{\partial \boldsymbol{\epsilon}} \right| &= \log |\det(\mathbf{I} + m_{\theta}\prime(\boldsymbol{\epsilon}))| \\ &\approx \sum_{k=1}^{\infty} (-1)^{k+1} \frac{\mathrm{Tr}(m_{\theta}\prime(\boldsymbol{\epsilon}))^k}{k} \quad \textit{(Taylor Approx.)} \end{aligned} \tag{17.17}$$

where $k = 1$ or $2$ in practise.