Normalizing Flows (NF)

goals	autoencoder	VAE	GAN	NF
small codes	hyperparameter		$\dim(z)$	$\dim(z)=\dim(x)$
good reconstruction $\hat{x}=D(E(x))\approx x$	✓	(✓)	✗	✓ (some: perfect)
accurate distribution $p(x)\approx p^*(x)$	✗	trade-off (✓)	✓	✓

idea of IVF: generalize inverse transformation method to arbitrary dimensions

- $x \in \mathbb{R}^D$, $p^*(x)$ high-dim density, $A \subseteq \mathbb{R}^D$ a region
- define $Pr(x \in A) = \int_A p^*(x)\, dx$
- let $z = f(x)$ an invertible encoding, $z_j = f_j(x_1, \dots, x_D)$, $x = f^{-1}(z) := g(z)$

$\tilde{A} = f(A)$ the image of A in z-space $\tilde{A} = \{z : z = f(x) \text{ for } x \in A\}$

- consistency: for all A: $Pr(z \in \tilde{A}) = \int_{\tilde{A}} q(z)\, dz \overset{!}{=} Pr(x \in A)$ (*)
- apply the multi-dimensional change-of-variables formula of Analysis II

$$\int_{\tilde{A}=f(A)} q(z)\, dz = \int_{g(\tilde{A})} q(z = f(x)) \left| \det J_f \right| dx$$

$\tilde{g}(\tilde{A}) = g(f(A)) = A$ Jacobian: $J_f = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \cdots & \frac{\partial f_1(x)}{\partial x_D} \\ \vdots & \ddots & & \\ \frac{\partial f_D(x)}{\partial x_1} & \cdots & & \frac{\partial f_D(x)}{\partial x_D} \end{pmatrix}$

- since consistency must hold for any A, the integrands must be equal
  $\Rightarrow$ multi-variate change-of-variables formula

$$\boxed{p(x) = q(z=f(x)) \left| \det J_f(x) \right|}$$

- goal: - pre-define $q(z)$ e.g. $q(z) = \mathcal{N}(0, \mathbb{I})$
  - learn $f(x)$, such that $p(x) \approx p^*(x)$ ($f(x) \hat{=}$ neural network encoder)
- recall 1-D: $q(z) = \text{uniform}(0,1) \Rightarrow f(x) = CDF_{p^*}(x)$
- for learning $q(z) = \mathcal{N}(0, \mathbb{I})$ is better: - non-zero gradients for gradient descent
  - supported on all of $\mathbb{R}^D$
- two difficult problems in practice.
  (1) how to ensure that $f(x)$ is invertible & efficiently compute $g(z) = f^{-1}(z)$
  (2) how to (efficiently) calculate $\det J_f$
- start with problem (2)

2-D determinant easy: $\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$, 3D ok: $\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \overset{a}{\underset{g}{}} \begin{matrix} a(ei - fh) \\ +b(f \cdot g - d \cdot i) \\ +c(d \cdot h - e \cdot g) \end{matrix}$

number of terms grows exponentially with D

- general solution with SVD

$$J = U \cdot \Lambda \cdot V^T \Rightarrow |\det J| = \det \Lambda = \prod_j \lambda_j$$

effort: $O(D^3)$

- exploit special case: if J is triangular

$$J = \begin{pmatrix} \neq 0 & & 0 \\ & \ddots & \\ \neq 0 & & \neq 0 \end{pmatrix}$$

$$\det J = \prod_j J_{jj}$$

we already talked about an instance: auto-regressive models

$$z_j = f_j(x_j; x_{<j}) \qquad J_f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & 0 & 0 & \cdots & 0 \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & 0 & \cdots & 0 \\ \vdots & & & & \\ \frac{\partial f_D}{\partial x_1} & \frac{\partial f_D}{\partial x_2} & \cdots & & \frac{\partial f_D}{\partial x_D} \end{pmatrix}$$

"triangular map"

"Knothe-Rosenblatt re-arrangement"

- if $f(x)$ is a multi-layer network, $f(x)$ is a composition of functions $f^{(\ell)}$

$$f = f^{(L)} \circ f^{(L-1)} \circ \dots \circ f^{(1)} \qquad z^{(\ell)} = f^{(\ell)}(z^{(\ell-1)}), \quad z^{(0)} := x \quad z := z^{(L)}$$

Jacobians of each layer $\quad J^{(\ell)} = \frac{\partial f^{(\ell)}(z^{(\ell-1)})}{\partial z^{(\ell-1)}}$

Jacobian of composition $\quad J = \frac{\partial f(x)}{\partial x} = J^{(L)} \cdot J^{(L-1)} \cdot \dots \cdot J^{(1)}$

$$\det J = \det J^{(L)} \cdot \dots \cdot \det J^{(1)}$$

$\Rightarrow$ determinant of multi-layer network is easy when layer determinants are easy

$\Rightarrow$ popular architecture: define all $f^{(\ell)}(z^{(\ell-1)})$ as auto-regressive functions

- return to problem (1): choose $f^{(\ell)}$ auto-regressive and easy to invert

major architecture: coupling layer $\Rightarrow$ network "Real NVP" [Dinh et al 2017]

- idea: in each layer, change only half of the dimensions of $z^{(\ell-1)}$
  pass the remaining dimensions on unchanged ("skip connection")

$z_j^{(\ell)} = z_j^{(\ell-1)}$ for $j = 1, \dots, \tilde{D}$ $\qquad \tilde{D} = \lfloor D/2 \rfloor$

$z_j^{(\ell)} = f_j^{(\ell)}\left(z_j^{(\ell-1)}, \underbrace{z_{1:\tilde{D}}^{(\ell-1)}}_{\subseteq z_{<j} \text{ since } \tilde{D} < j}\right)$ for $j = \tilde{D}+1, \dots, D$

$$J^{(\ell)} = \begin{pmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_j^{(\ell)}}{\partial z_{j'}^{(\ell-1)}} \text{ for } j' \le \tilde{D} & & & \frac{\partial f_{\tilde{D}+1}^{(\ell)}}{\partial z_{\tilde{D}+1}^{(\ell-1)}} \; 0 \\ & & & 0 \quad \ddots \\ & & & \frac{\partial f_D^{(\ell)}}{\partial z_D^{(\ell-1)}} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_{\tilde{D}} & \vdots & 0_D \\ \cdots & \vdots & \cdots \\ * 0 & \vdots & \operatorname{diag}\left(\frac{\partial f_j^{(\ell)}}{\partial z_j^{(\ell-1)}}\right) \\ & \vdots & \end{pmatrix}$$

is triangular

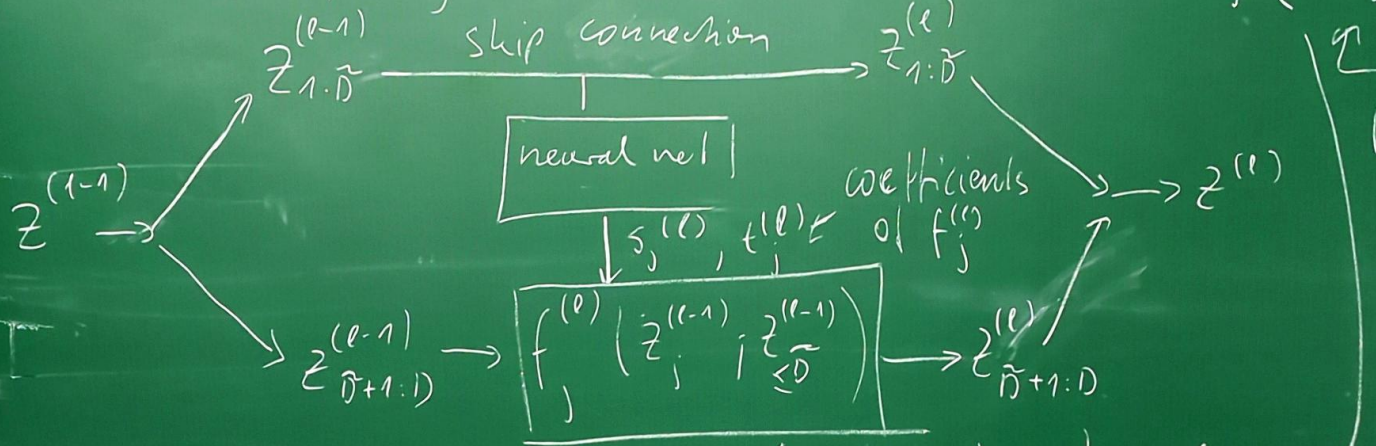
$$\det J^{(\ell)} = \prod_{j=\tilde{D}+1}^{D} \frac{\partial f_j^{(\ell)}\left(z_j^{(\ell-1)}, z_{\le \tilde{D}}^{(\ell-1)}\right)}{\partial z_j^{(\ell-1)}}$$

- simplest invertible function: affine $z_j^{(\ell)} = s \cdot z_j^{(\ell-1)} + t$

$s_j^{(\ell)}\left(z_{\leq D}^{(\ell-1)}\right)$ $\quad$ $t_j^{(\ell)}\left(z_{\leq D}^{(\ell-1)}\right)$

↳ two neural nets $\mapsto$ $(D \cdot L$ in total$)$

"affine coupling layer"



- for $f_j^{(\ell)}$ to be invertible, we need $s_j^{(\ell)} \neq 0$

$\Rightarrow$ we actually learn $\tilde{s}_j^{(\ell)}\left(z_{\leq D}^{(\ell-1)}\right)$ and set $s_j^{(\ell)} = \exp\left(\tilde{s}_j^{(\ell)}\right) > 0$

(in practice it is numerically more stable to set $s_j^{(\ell)} = \exp\left(\tanh\left(\tilde{s}_j^{(\ell)}\right)\right)$

- forward: $z_j^{(\ell)} = \exp\left(s_j^{(\ell)}\left(z_{\leq \tilde{D}}^{(\ell-1)}\right)\right) \cdot z_j^{(\ell-1)} + t_j^{(\ell)}\left(z_{\leq \tilde{D}}^{(\ell-1)}\right) \quad j > \tilde{D}$

$z_j^{(\ell)} = z_j^{(\ell-1)} \quad j \leq \tilde{D}$

- inverse: $z_j^{(\ell-1)} = z_j^{(\ell)}$ for $j \leq \tilde{D}$ thanks to skip connection

$$z_j^{(\ell-1)} = \left(z_j^{(\ell)} - t_j^{(\ell)}\left(z_{\leq \tilde{D}}^{(\ell-1)}\right)\right) \cdot \exp\left(-s_j^{(\ell)}\left(z_{\leq \tilde{D}}^{(\ell-1)}\right)\right) \quad \text{for } j > \tilde{D}$$

$\Rightarrow$ invertible layer constructed from non-invertible s and t (always executed in same direction)

$\Rightarrow$ choose s and t according to application (fully connected, convolutional, ...)

determinant of a affine coupling layer:

$$\det J^{(\ell)} = \prod_{j=\tilde{D}+1}^{D} \frac{\partial f_j^{(\ell)}\left(z_j^{(\ell-1)}; z_{\leq \tilde{D}}^{(\ell-1)}\right)}{\partial z_j^{(\ell-1)}} = \prod_{j=\tilde{D}+1}^{D} \exp\left(s_j^{(\ell)}\left(z_{\leq \tilde{D}}^{(\ell-1)}\right)\right)$$