Invertible Neural Networks and Inverse Problems

Generative Models and Likelihood

Sample from distribution learned from data

Want: density $p_\theta(x)$ s.t. training data has high likelihood

Let $\mathcal{D} = \{x_i\}_{i=1 \cdots N}$

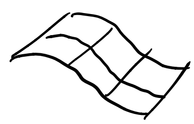
$$\mathcal{L}_\theta(\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} -\log p_\theta(x_i)$$

$$\min_\theta \mathcal{L}_\theta(\mathcal{D})$$

Variational Autoencoder (VAE)

Train $G_\theta : \mathbb{R}^k \to \mathbb{R}^n$

$z \mapsto x = G_\theta(z)$



As range(G_θ) is k-dim manifold, $p(x) = 0$ almost everywhere.

Define noisy observation model

$$p_\theta(x|z) = \mathcal{N}(x \mid G_\theta(z), \eta I)$$

Likelihood is defined everywhere

$$p_\theta(x) = \int p_\theta(x|z)\, p(z)\, dz$$

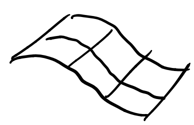
simple prior on z, eg $\mathcal{N}(0, I_k)$

Intractable, so optimize a lower bound.

Generative Adversarial Nets (GANs)

Train $G_\theta : \mathbb{R}^k \to \mathbb{R}^n$

$z \mapsto x = G_\theta(z)$



Range (G_θ) is k-dimensional

Train adversarially w/ a discriminator/critic

No direct modeling of likelihood

Invertible Neural Networks

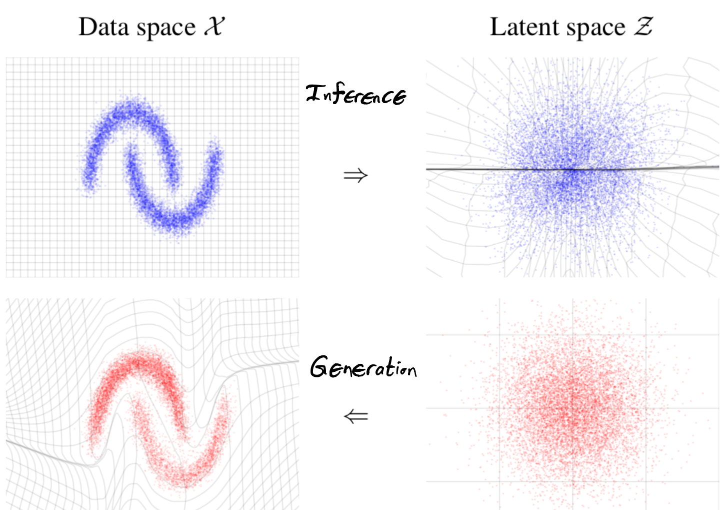
Train $G_\theta : \mathbb{R}^n \to \mathbb{R}^n$
$z \mapsto x = G_\theta(z)$

full dimensional
latent space

Allows exact calculation of likelihood of <u>any</u> image for <u>any</u> parameters θ.

<u>Can train by likelihood maximization</u>!

Impose prior on latent space: $z \sim N(0, I_n)$
Induces distribution in image space: $G_\theta(z)$



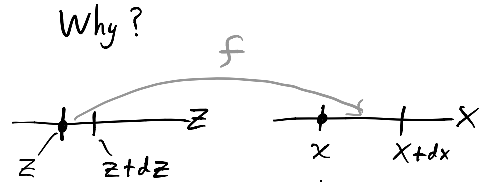
Probability Basics – Change of Variables

Let $z \in \mathbb{R} \sim p_z$, $f(z) = x \sim p_x$, f differentiable and strictly monotonic

What is relationship of p_z and p_x?

$$p_x(x) = \frac{p_z(z)}{\left|\frac{dx}{dz}\right|} = p_z(z(x)) \left|\frac{dz}{dx}(x)\right|$$

Why?



prob. $p_z(z)dz = p_x(x)dx$

$$p_x(x) = p_z(z)\, dz/dx$$

In multiple dimensions

Let $f: \mathbb{R}^n \to \mathbb{R}^n$ $\quad z = f(x)$

$$P_X(x) = P_Z(z) \left| \det \underbrace{\frac{\partial z(x)}{\partial x}}_{\text{Jacobian, } \mathbb{R}^{n \times n}} \right|$$

Architectures of INNs

Normalizing flow – simple density in z mapped to complicated density in X by a sequence of invertible transformations

$$h_K = z \overset{f_K}{\longleftrightarrow} h_{K-1} \leftrightarrow \cdots \overset{f_2}{\longleftrightarrow} h_1 \overset{f_1}{\longleftrightarrow} X = h_0$$

$$\log P_X(x) = \log P_Z(z) + \sum_{i=1}^{K} \log \left| \det \left(\frac{dh_i}{dh_{i-1}} \right) \right|$$

Trick: Choose mappings f_i with triangular Jacobians

$$\log \left| \det \left(\frac{dh_i}{dh_{i-1}} \right) \right| = \mathrm{sum}\left(\log \left| \mathrm{diag}\ \frac{dh_i}{dh_{i-1}} \right| \right)$$

How can we use convolutions to build invertible transformations?

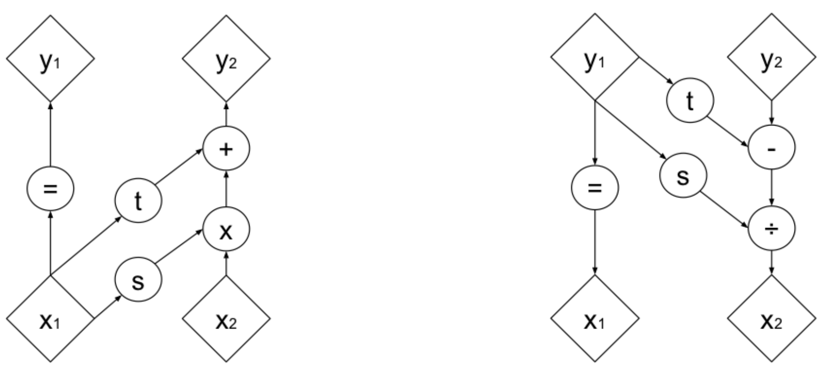
Affine coupling layer:

Given $X \in \mathbb{R}^D$, $d < D$

$$y_{1:d} = X_{1:d}$$

$$y_{d+1:D} = X_{d+1:D} \odot \exp(\underbrace{s}_{\text{scale}}(X_{1:d})) + \underbrace{t}_{\text{translation}}(X_{1:d})$$

CNN's – need not be invertible



(a) Forward propagation (b) Inverse propagation

Jacobian of affine coupling layer

$$\frac{\partial y}{\partial x} = \begin{pmatrix} I_{d \times d} & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}} & \operatorname{diag}(\exp(s(x_{1:d}))) \end{pmatrix}$$

lower triangular

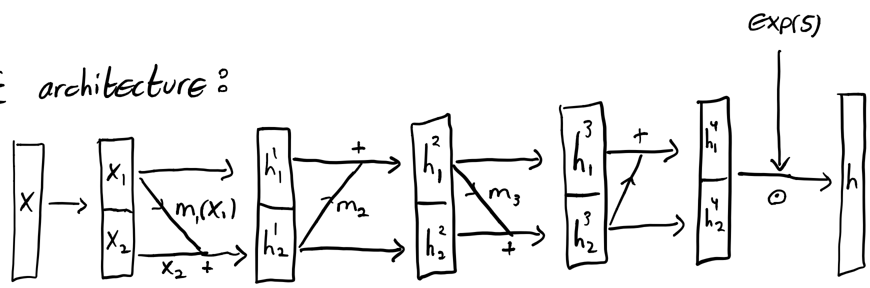
$$\det \frac{\partial y}{\partial x} = \exp\Big(\sum_j s(x_{1:d})_j\Big)$$

Only involves forward eval of s,
so s & t can be arbitrarily complicated

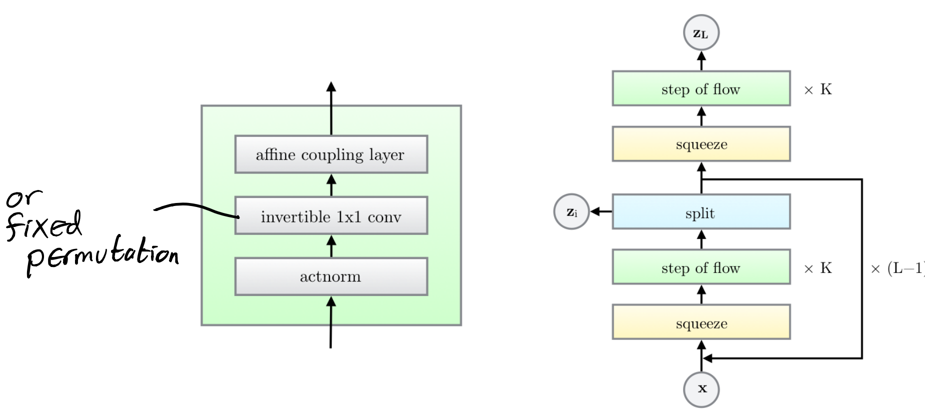
Architectures using coupling layers

Roughly: coupling layer → shuffle components → coupling layer → shuffle components

NICE architecture:



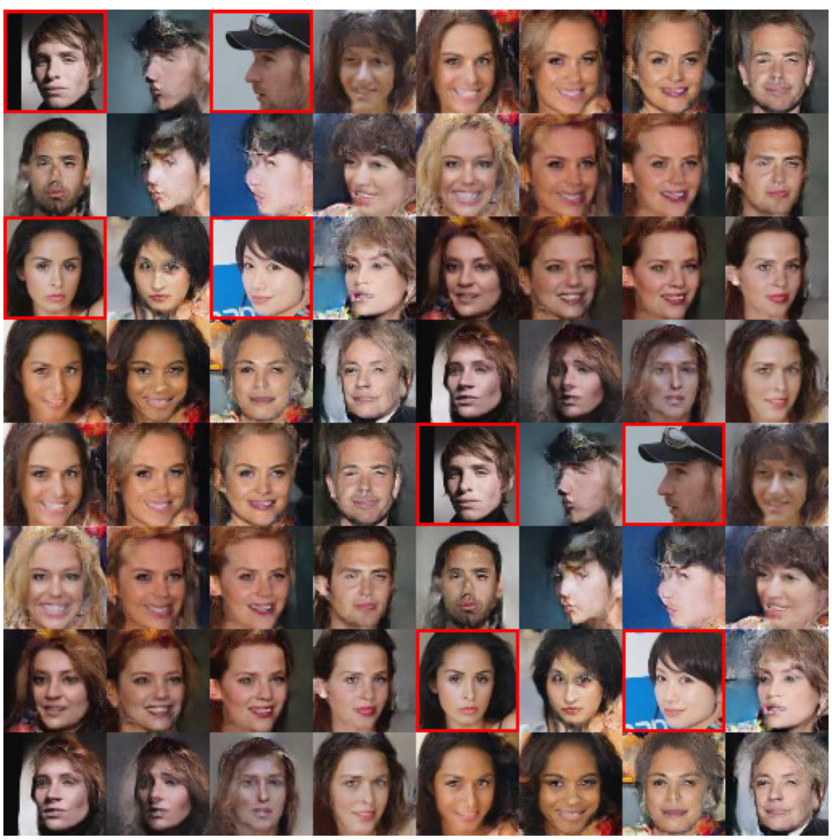
Glow / Real-NVP Architecture



(a) One step of our flow. (b) Multi-scale architecture (Dinh et al., 2016).

Samples from INNs

Source: Glow paper



Source: Real-NVP

Using INNs as priors for inverse problems

A trained INN provides a density $p(x)$ induced by $x = G_\theta(z)$ w/ $z \sim \mathcal{N}(0, I_n)$, $G_\theta : \mathbb{R}^n \to \mathbb{R}^n$

Consider image $x_* \in \mathbb{R}^n$
measurements $y = A x_* + \eta$ for $A \in \mathbb{R}^{m \times n}$, $\eta \in \mathbb{R}^n$

Maximum Likelihood formulation

$$\max_x \ p(x) \quad \text{st} \quad Ax = y \qquad (\text{noiseless case})$$

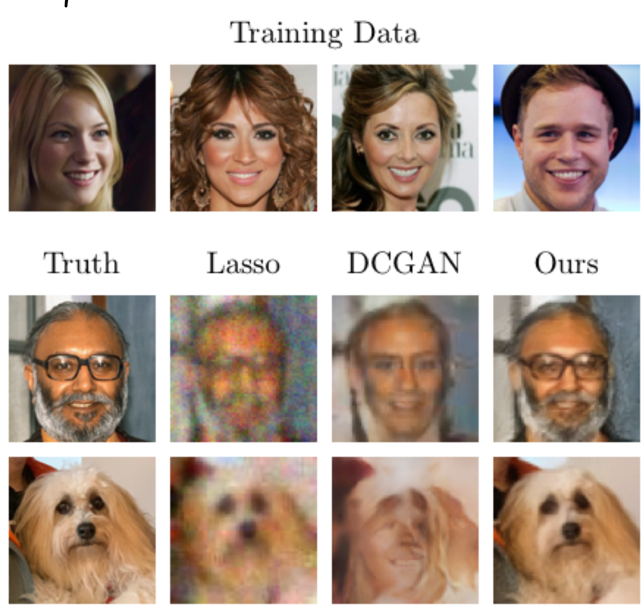
$$\min_x \ -\log p(x) + \gamma \, \|Ax - y\|^2 \qquad (\text{noisy case})$$

$$\min_z \ \|A G(z) - y\|^2 \quad \text{w/ } z_0 = 0 \qquad (\text{optimization in latent space})$$

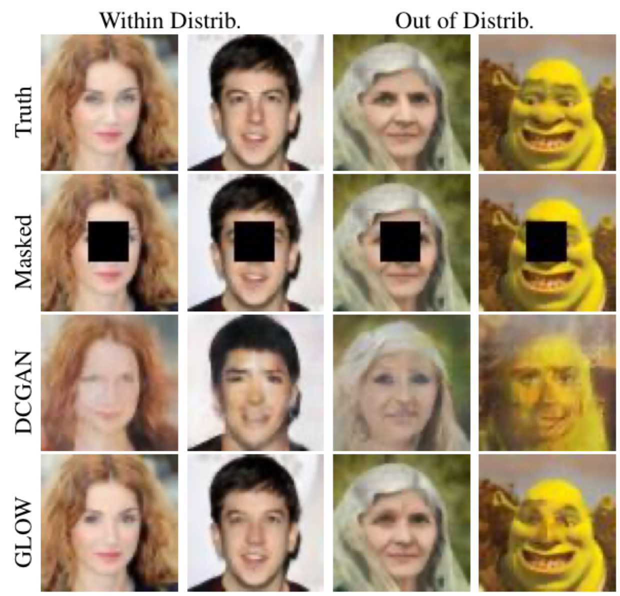
See: Asim et al. 2019
Whang et al. 2020

Out of distribution performance of INNs for Compressed Sensing

Compressed Sensing
w/ $m = 0.2\,n$
Gaussian measurements



Inpainting w/ Glow prior



Theory for INNs in linear case

Suppose $G \in \mathbb{R}^{n \times n}$ has sigular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$

Let $P_G(x)$ be given by $x = Gz$ for $z \sim \mathcal{N}(0, I_n)$

Let $A \in \mathbb{R}^{m \times n}$ have iid $\mathcal{N}(0,1)$ entries

Theorem (Asim, Daniels, Leong, Hand, Ahmed)

Let $x_0 \sim P_G$ for fixed G.

If $4 \leq m < n$, then the MLE estimate

$$\hat{x} = \arg\max_{x} P_G(x) \quad \text{st} \quad Ax = Ax_0$$

obeys

$$\sum_{i > m} \sigma_i^2 \leq \mathbb{E}_A \mathbb{E}_{x_0 \sim P_G} \|\hat{x} - x_0\|^2 \lesssim m \sum_{i > m-2} \sigma_i^2 .$$

Note: As $m \to n$, error $\to 0$, unlike w/ GAN theory

Also see Gaussian width based theory in Whang et al. (nonlinear case)

Other approaches for inverse problems w/ INNs

Approximate forward operator by INN, get inverse for free (supervised learning)

Ardizzone et al. 2018

Limitations of INNs

- Computationally expensive
- Do not have explicit low-dimensional signal representations

Strengths of INNs

- Exact likelihood calculation
- Exact latent-variable inference
- Memory savings in back-prop
- All signals are in their range
  ⇒ Strong out-of-distribution performance