[Paper review 28]

Dennsity Estimation using Real NVP

(Laurent Dinh, et al., 2017)

[Contents]

1. Abstract

Advantages of "Flow based generative models "

- 1) tractability of the exact log-likelihood
- 2) tractability of exact latent-variable inference
- 3) parallelizability of both training and syntehsis

Glow

- simple type of generative flow, using "invertible 1 x 1 convolution"
- significant improvement in log-likelihood on standard benchmarks

2. Introduction

2 major problems in ML

- 1) data efficiency (ability to learn from few data points)
- 2) generalization (robustness to changes of the task)

Promise of generative models : overcome these 2 problems by

- learning realistic world models
- learning meaningful features of the input

Generative Modeling have advanced with likelihood-based methods

Likelihood-based methods : three categories

- 1) Autoregressive models
- 2) VAEs

- 3) Flow-based generative models (ex. NICE, RealNVP)

3) Flow-based generative model's merit

- exact latent-variable inference and log-likelihood evaluation
- efficient inference and efficient synthesis
- useful latent space for downstream tasks
- significant potential for memory savings

3. Background : Flow-based Generative Models

x : high-dimensional random vector

$x \sim p^*(x)$: unknown true distribution

log-likelihood objective : minimizing....

- (discrete x)

  $\mathcal{L}(\mathcal{D})=\frac{1}{N} \sum_{i=1}^N-\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right)$
- (continuous x)

  $\mathcal{L}(\mathcal{D}) \simeq \frac{1}{N} \sum_{i=1}^N-\log p_{\boldsymbol{\theta}}\left(\tilde{\mathbf{x}}^{(i)}\right)+c$
  - $\tilde{\mathrm{x}}^{(i)}=\mathrm{x}^{(i)}+u \text{ with } u \sim \mathcal{U}(0, a), \text{ and } c=-M \cdot \log a$
  - a : determined by the discretization level of the data
  - M : dimensionality of x

Generative process of most flow-based generative models

$$\begin{aligned}
&\mathrm{z} \sim p_{\boldsymbol{\theta}}(\mathrm{z}) \\
&\mathrm{x}=\mathrm{g}_{\theta}(\mathrm{z})
\end{aligned}$$

- z : latent variable
- $p_\theta(z)$: tractable density
- $g_\theta(\cdot)$: invertible (=biijective)

Change of variables + triangular matrix

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{x}) &=\log p_{\boldsymbol{\theta}}(\mathbf{z})+\log |\operatorname{det}(d \mathbf{z} / d \mathbf{x})| \\
&=\log p_{\boldsymbol{\theta}}(\mathbf{z})+\sum_{i=1}^K \log \left|\operatorname{det}\left(d \mathbf{h}_i / d \mathbf{h}_{i-1}\right)\right| \\
&=\log p_{\boldsymbol{\theta}}(\mathbf{z})+\sum_{i=1}^K \operatorname{sum}\left(\log \left|\operatorname{diag}\left(d \mathbf{h}_i / d \mathbf{h}_{i-1}\right)\right|\right)
\end{aligned}$$

4. Proposed Generative Flow

we propose a new flow

- built on NICE and RealNVP
- consists of a series of steps of flows
- combined with multi-scale architecture

4.1 Actnorm : scale & bias layer with data dependent initialization

actnorm layer :

- performs an affine transformation of the activations, using a "scale and bias" parameters per channel
- data dependent initialization

4.2 Invertible 1 x 1 convolution

permutation that reverses the ordering of the channels

(1x1 convolution with equal number of input & output channels = generalization of permutation operation)

log-determinant of an invertible 1x1 convolution of $h \times w \times c$ tensor $\mathbf{h}$ with $c \times c$ weight matrix $\mathbf{W}$

$\log\left|\det\left(\frac{d \operatorname{conv} 2\mathrm{D}(\mathbf{h};\mathbf{W})}{d\mathbf{h}}\right)\right| = h \cdot w \cdot \log|\det(\mathbf{W})|$

Computation cost

- before : $\operatorname{conv} 2\mathrm{D}(\mathbf{h};\mathbf{W}) \rightarrow \mathcal{O}\left(h \cdot w \cdot c^2\right)$
- after : $\det(\mathbf{W}) \rightarrow \mathcal{O}\left(c^3\right)$,

  proof) LU Decomposition

  $\mathbf{W} = \mathbf{PL}(\mathbf{U} + \operatorname{diag}(\mathbf{s}))$

  $\log|\det(\mathbf{W})| = \operatorname{sum}(\log|\mathbf{s}|)$

  where P : permutation matrix & W random rotation matrix

4.3 Affine Coupling Layers

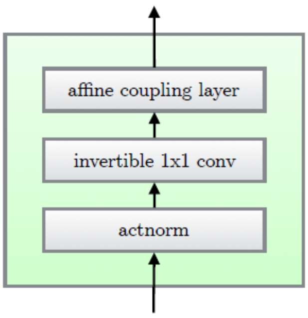
ex) $s = 1$: additive coupling layer

- Zero initialization

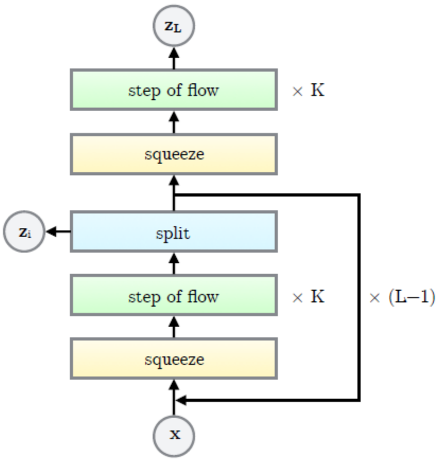
  (initialize the last convolution of each NN() with zeros)
- Split and Concatenation

  (splits $\mathbf{h}$ the input tensor into 2 halves)

Description	Function	Reverse Function	Log-determinant
Actnorm. See Section 3.1	$\forall i,j: \mathbf{y}_{i,j} = \mathbf{s} \odot \mathbf{x}_{i,j} + \mathbf{b}$	$\forall i,j: \mathbf{x}_{i,j} = (\mathbf{y}_{i,j} - \mathbf{b})/\mathbf{s}$	$h \cdot w \cdot \mathtt{sum}(\log\|\mathbf{s}\|)$
Invertible 1×1 convolution. $\mathbf{W}: [c \times c]$. See Section 3.2	$\forall i,j: \mathbf{y}_{i,j} = \mathbf{W}\mathbf{x}_{i,j}$	$\forall i,j: \mathbf{x}_{i,j} = \mathbf{W}^{-1}\mathbf{y}_{i,j}$	$h \cdot w \cdot \log\|\det(\mathbf{W})\|$ or $h \cdot w \cdot \mathtt{sum}(\log\|\mathbf{s}\|)$ (see eq. (10))
Affine coupling layer. See Section 3.3 and (Dinh et al., 2014)	$\mathbf{x}_a, \mathbf{x}_b = \mathtt{split}(\mathbf{x})$ $(\log \mathbf{s}, \mathbf{t}) = \mathtt{NN}(\mathbf{x}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{y}_a = \mathbf{s} \odot \mathbf{x}_a + \mathbf{t}$ $\mathbf{y}_b = \mathbf{x}_b$ $\mathbf{y} = \mathtt{concat}(\mathbf{y}_a, \mathbf{y}_b)$	$\mathbf{y}_a, \mathbf{y}_b = \mathtt{split}(\mathbf{y})$ $(\log \mathbf{s}, \mathbf{t}) = \mathtt{NN}(\mathbf{y}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{x}_a = (\mathbf{y}_a - \mathbf{t})/\mathbf{s}$ $\mathbf{x}_b = \mathbf{y}_b$ $\mathbf{x} = \mathtt{concat}(\mathbf{x}_a, \mathbf{x}_b)$	$\mathtt{sum}(\log(\|\mathbf{s}\|))$



(a) One step of our flow.



(b) Multi-scale architecture (Dinh et al., 2016).