Улучшенный вариационный вывод с обратным авторегрессионным потоком

Плетнев Никита Вячеславович

Московский физико-технический институт
Факультет управления и прикладной математики
Кафедра интеллектуальных систем

15 декабря 2020 г.

Постановка задачи

Необходимо предложить новый вид Normalizing Flows,

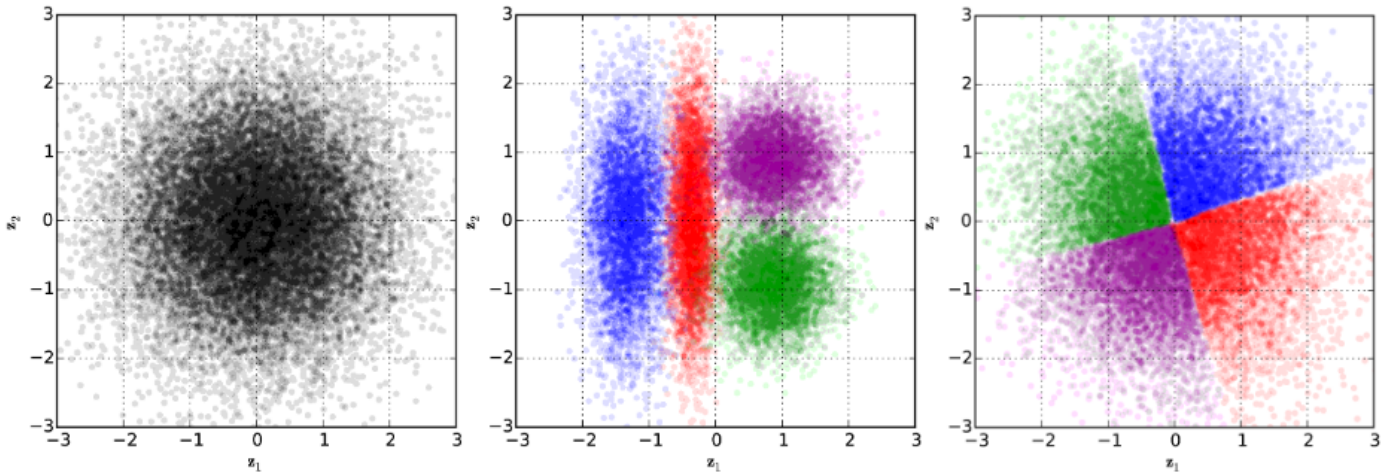
который будет хорошо масштабироваться на латентные пространства большой размерности.

Для чего это нужно

Normalizing Flows обеспечивает общую стратегию гибкого вариационного вывода на латентных переменных, что используется при построении вариационных автокодировщиков. Однако ранее опубликованные примеры потоков плохо масштабируются на латентные пространства большой размерности: требование параллелизации накладывает ограничения на апостериорное распределение. С другой стороны, оно должно быть достаточно гибким, чтобы приближать истинное распределение.

Пример применения построенного IAF

«Игрушечный» эксперимент на синтетических данных



(a) Prior distribution (b) Posteriors in standard VAE (c) Posteriors in VAE with IAF

Figure 1: Best viewed in color. We fitted a variational auto-encoder (VAE) with a spherical Gaussian prior, and with factorized Gaussian posteriors **(b)** or inverse autoregressive flow (IAF) posteriors **(c)** to a toy dataset with four datapoints. Each colored cluster corresponds to the posterior distribution of one datapoint. IAF greatly improves the flexibility of the posterior distributions, and allows for a much better fit between the posteriors and the prior.

Вариационный вывод и обучение

Обозначения

x — вектор наблюдаемых переменных,

z — вектор латентных переменных,

$p(x, z)$ — генеративная модель для совместного распределения,

$X = \{x_1, \ldots, x_N\}$ — датасет.

Цель оптимизации

Как обычно, хотим максимизировать правдоподобие:

$$\log p(X) = \sum_{i=1}^{N} \log p(x_i).$$

Но в общем случае эта величина (или её производная) невычислима, поэтому применяется вариационная нижняя оценка:

$$\log p(x) \geq \mathcal{L}(x, \theta) = \mathbb{E}_{q(z|x)}[\log p(x, z) - \log q(z|x)] =$$
$$= \log p(x) - D_{KL}(q(z|x)||p(z|x)).$$

Требования

- Апостериорное распределение $q(z|x)$ должно быть эффективно вычислимо и дифференцируемо;
- Если размерность латентного пространства велика, то вычисление должно быть параллелизуемо.
- Это создаёт ограничения на семейство апостериорных распределений;
- С другой стороны, оно должно быть достаточно гибким, чтобы соответствовать истинному апостериорному распределению $p(z|x)$.

Подход к решению

Normalizing flows (Rezende and Mohamed, 2015)

Цепочка обратимых параметризованных преобразований:

$$z_0 \sim q(z_0|x), \quad z_t = f_t(z_{t-1}|x) \quad \forall t = 1..T$$

.

Если все якобианы вычислимы, то на последней итерации:

$$\log q(z_T|x) = \log q(z_0|x) - \sum_{t=1}^{T} \log \det \left| \frac{dz_t}{dz_{t-1}} \right|.$$

Сами авторы экспериментировали только с ограниченным семейством

$$f_t(z_{t-1}) = z_{t-1} + uh(w^T z_{t-1} + b).$$

(MLP с одним нейроном). Для обнаружения закономерностей высокой размерности нужна очень длинная цепочка.

Inverse Autoregressive Transformations

Основная идея

Для масштабируемости рассматриваем гауссовы версии VAE, такие как MADE, PixelCNN.
Переменная y моделируется: $y = \{y_i\}_{i=1}^{D}$.
$[\mu(y), \sigma(y)]$ — функция из вектора y в пару векторов (μ, σ).
Из-за авторегрессионной структуры матрица Якоби — нижнетреугольная с нулями на диагонали.

Сэмплирование

Последовательное преобразование «шума» $\epsilon \sim \mathcal{N}(0, I)$,
$y_0 = \mu_0 + \sigma_0 \cdot \epsilon_0$, $y_i = \mu_i(y_{1:i-1}) + \sigma_i(y_{1:i-1}) \cdot \epsilon_i$.
У нас есть параллелизуемое обратное преобразование:
$\epsilon = \frac{y - \mu(y)}{\sigma(y)}$.

И вычислимый якобиан: $\log\det\left|\frac{d\epsilon}{dy}\right| = -\sum_{i=1}^{D}\log\sigma_i(y)$

Предлагаемое решение: IAF — Inverse Autoregressive Flow

Схема

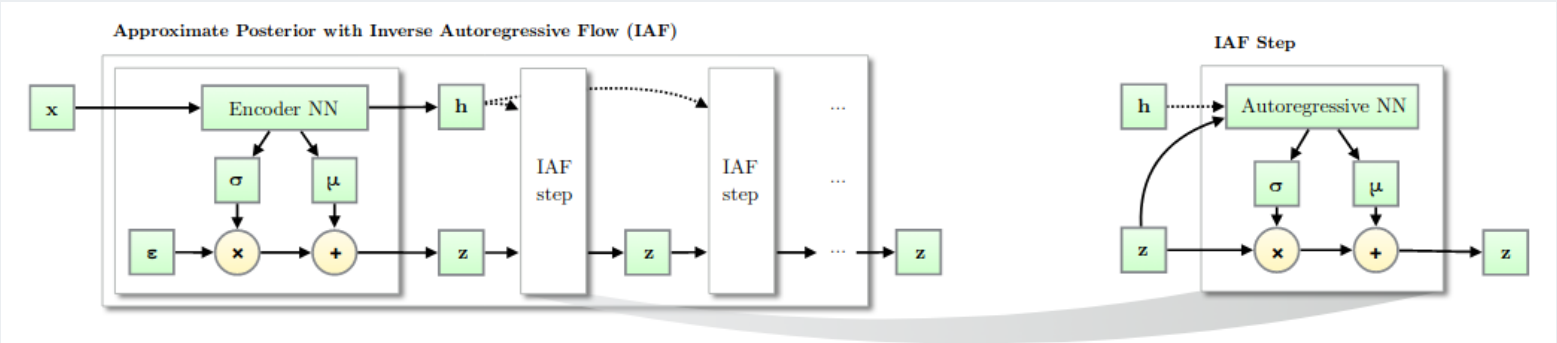


Figure 2: Like other normalizing flows, drawing samples from an approximate posterior with Inverse Autoregressive Flow (IAF) consists of an initial sample $\mathbf{z}$ drawn from a simple distribution, such as a Gaussian with diagonal covariance, followed by a chain of nonlinear invertible transformations of $\mathbf{z}$, each with a simple Jacobian determinants.

Формулы

$$z_0 = \mu_0 + \sigma_0 \odot \epsilon; \quad z_t = \mu_t + \sigma_t \odot z_{t-1}.$$

$$\log q(z_T|x) = -\sum_{i=1}^{D}\left(\frac{1}{2}\epsilon_i^2 + \frac{1}{2}\log(2\pi) + \sum_{t=0}^{T}\log\sigma_{t,i}\right)$$

Предлагаемое решение: IAF — Inverse Autoregressive Flow

Алгоритм

Algorithm 1: Pseudo-code of an approximate posterior with Inverse Autoregressive Flow (IAF)

Data:

$\mathbf{x}$: a datapoint, and optionally other conditioning information
$\boldsymbol{\theta}$: neural network parameters
$\texttt{EncoderNN}(\mathbf{x}; \boldsymbol{\theta})$: encoder neural network, with additional output $\mathbf{h}$
$\texttt{AutoregressiveNN}[*](\mathbf{z}, \mathbf{h}; \boldsymbol{\theta})$: autoregressive neural networks, with additional input $\mathbf{h}$
$\texttt{sum}(.)$: sum over vector elements
$\texttt{sigmoid}(.)$: element-wise sigmoid function

Result:

$\mathbf{z}$: a random sample from $q(\mathbf{z}|\mathbf{x})$, the approximate posterior distribution
l: the scalar value of $\log q(\mathbf{z}|\mathbf{x})$, evaluated at sample '$\mathbf{z}$'

$[\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{h}] \leftarrow \texttt{EncoderNN}(\mathbf{x}; \boldsymbol{\theta})$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$
$\mathbf{z} \leftarrow \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} + \boldsymbol{\mu}$
$l \leftarrow -\texttt{sum}(\log \boldsymbol{\sigma} + \frac{1}{2}\boldsymbol{\epsilon}^2 + \frac{1}{2}\log(2\pi))$
for $t \leftarrow 1$ **to** T **do**
  $[\mathbf{m}, \mathbf{s}] \leftarrow \texttt{AutoregressiveNN}[t](\mathbf{z}, \mathbf{h}; \boldsymbol{\theta})$
  $\boldsymbol{\sigma} \leftarrow \texttt{sigmoid}(\mathbf{s})$
  $\mathbf{z} \leftarrow \boldsymbol{\sigma} \odot \mathbf{z} + (1 - \boldsymbol{\sigma}) \odot \mathbf{m}$
  $l \leftarrow l - \texttt{sum}(\log \boldsymbol{\sigma})$
end

Преобразование как в LSTM для стабильности вычислений.

Эксперимент на MNIST

Model	VLB	$\log p(\mathbf{x}) \approx$
Convolutional VAE + HVI [1]	-83.49	-81.94
DLGM 2hl + IWAE [2]		-82.90
LVAE [3]		-81.74
DRAW + VGP [4]	-79.88	
Diagonal covariance	-84.08 ($\pm$ 0.10)	-81.08 ($\pm$ 0.08)
IAF (Depth = 2, Width = 320)	-82.02 ($\pm$ 0.08)	-79.77 ($\pm$ 0.06)
IAF (Depth = 2, Width = 1920)	-81.17 ($\pm$ 0.08)	-79.30 ($\pm$ 0.08)
IAF (Depth = 4, Width = 1920)	-80.93 ($\pm$ 0.09)	-79.17 ($\pm$ 0.08)
IAF (Depth = 8, Width = 1920)	-80.80 ($\pm$ 0.07)	**-79.10** ($\pm$ 0.07)

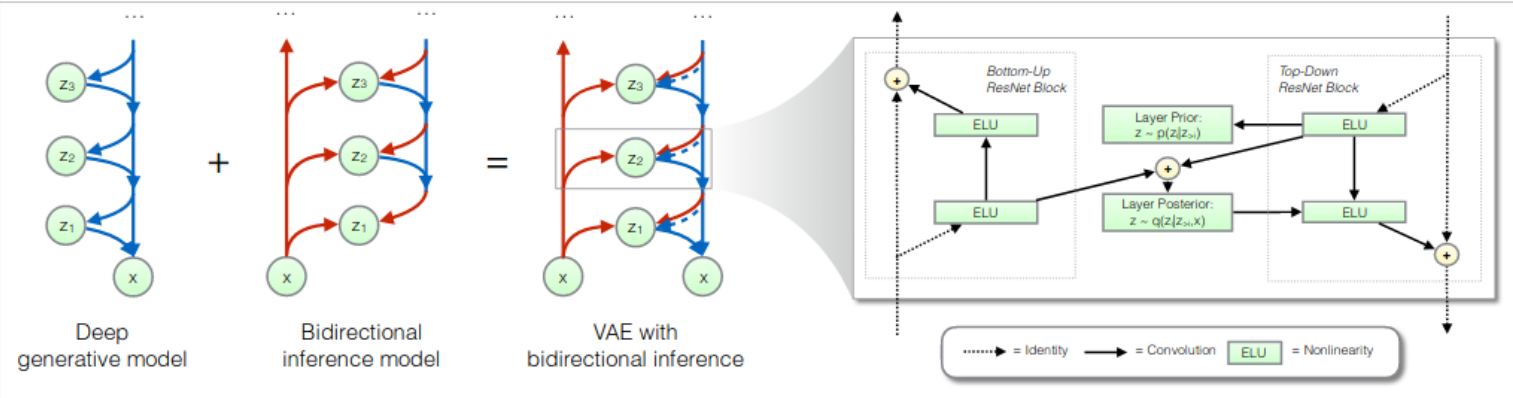


Figure 3: Overview of our ResNet VAE with bidirectional inference. The posterior of each layer is parameterized by its own IAF.

Эксперимент на CIFAR-10: латентные вектора в памяти

Table 2: Our results with ResNet VAEs on CIFAR-10 images, compared to earlier results, in *average number of bits per data dimension* on the test set. The number for convolutional DRAW is an upper bound, while the ResNet VAE log-likelihood was estimated using importance sampling.

Method	bits/dim $\leq$
Results with tractable likelihood models:	
Uniform distribution (van den Oord et al., 2016b)	8.00
Multivariate Gaussian (van den Oord et al., 2016b)	4.70
NICE (Dinh et al., 2014)	4.48
Deep GMMs (van den Oord and Schrauwen, 2014)	4.00
Real NVP (Dinh et al., 2016)	3.49
PixelRNN (van den Oord et al., 2016b)	**3.00**
Gated PixelCNN (van den Oord et al., 2016c)	**3.03**
Results with variationally trained latent-variable models:	
Deep Diffusion (Sohl-Dickstein et al., 2015)	5.40
Convolutional DRAW (Gregor et al., 2016)	3.58
ResNet VAE with IAF (Ours)	**3.11**

При этом наш метод сэмплирует изображение за 0.05 секунд, а PixelNet — за 52 секунды.

Значимость результатов

Предложены новый тип потоков — Inverse Autoregressive Flow, который хорошо масштабируется в латентные пространства большой размерности.

Проведены эксперименты, показывающие эффективность подхода: получены близкие к наилучшим результаты при намного более быстром сэмплировании.

Использованные источники

[1] Diederik P. Kingma, Tim Salimans: Improved Variational Inference with Inverse Autoregressive Flow, https://arxiv.org/pdf/1606.04934.pdf