VOLGOGRAD STATE TECHNICAL UNIVERSITY

Машинное обучение и нейросетевые модели

Лекция 6. Эстиматоры для вариационного инференса

Лектор: Кравченя Павел Дмитриевич

Волгоград 2025

План лекции

1. Проблема расчета градиента от эстиматора Монте-Карло.
2. Стохастический граф и его использование для оценки градиентов.
3. Репараметризация распределений.
4. Эстиматоры градиента ELBO и их свойства.
5. Способы уменьшения дисперсии эстиматора REINFORCE. Понятие baselines.
6. Способы определения baselines в Pyro.
7. Стохастический вариационный инференс для дискретных переменных.
8. Трюки репараметризации Gumbel-Max и Gumbel-Softmax.
9. Особенности Gumbel-Softmax эстиматора. Эстиматор Straight Through.
10. Сравнение разных подходов к выполнению байесовского вывода.

Вариационный инференс как задача оптимизации

- Решаем задачу **вариационного инференса** – определяем *вариационное* распределение, обладающее *нужными нам свойствами* (простота, легкость получения семплов), которым можно с достаточной точностью *заменить* сложное *апостериорное распределение*.
- Часто вариационное распределение (а иногда и правдоподобие модели) представляется в **параметрическом виде**: $q(\mathbf{z} \mid \boldsymbol{\varphi}) = q_{\boldsymbol{\varphi}}(\mathbf{z}), p(\mathbf{x} \mid \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$.
- Это позволяет свести *максимизацию* ELBO к *задаче оптимизации*, похожей на задачу *обучения модели* в *классических* ML и DL:

$$\boldsymbol{\theta}^{opt}, \boldsymbol{\varphi}^{opt} = \underset{(\boldsymbol{\theta},\boldsymbol{\varphi}) \in \Phi}{\arg\min} \left\{-\mathrm{ELBO}\left(p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x}), q_{\boldsymbol{\varphi}}(\mathbf{z})\right)\right\} = \underset{(\boldsymbol{\theta},\boldsymbol{\varphi}) \in \Phi}{\arg\min} \left\{-\mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right]\right\}.$$

- А задачу оптимизации можно решать *методами градиентного спуска*.

Проблема расчета градиента от эстиматора Монте-Карло

- Но есть нюанс: *непосредственное* вычисление градиента матожидания в *ELBO* затруднено, поскольку *матожидание* вычисляется с использованием *эстиматора Монте-Карло*, требующего расчета семплов.
- И в случае расчета *градиента* от ELBO по $\boldsymbol{\theta}$ *проблем нет*:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] = \nabla_{\boldsymbol{\theta}} \int \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \cdot q_{\boldsymbol{\varphi}}(\mathbf{z}) d\mathbf{z} = \int \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}) \cdot q_{\boldsymbol{\varphi}}(\mathbf{z}) d\mathbf{z} = \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})].$$

- Однако, расчет *градиента* от ELBO по $\boldsymbol{\varphi}$ требует расчета производной по $\boldsymbol{\varphi}$ от *матожидания* по распределению, которое *само зависит* от $\boldsymbol{\varphi}$:

$$\nabla_{\boldsymbol{\varphi}} \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] = \int \nabla_{\boldsymbol{\varphi}} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \cdot q_{\boldsymbol{\varphi}}(\mathbf{z})\right] d\mathbf{z} = \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\nabla_{\boldsymbol{\varphi}} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] + \int \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \cdot \nabla_{\boldsymbol{\varphi}} q_{\boldsymbol{\varphi}}(\mathbf{z}) d\mathbf{z}.$$

Преобразование градиента от матожидания

- Как посчитать *градиент от матожидания*? Попробуем *преобразовать* его:

$$\nabla_{\boldsymbol{\varphi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}\left[\frac{\log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] = \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\nabla_{\boldsymbol{\varphi}} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] + \int \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \cdot \nabla_{\boldsymbol{\varphi}} q_{\boldsymbol{\varphi}}(\mathbf{z}) d\mathbf{z}.$$

- Рассмотрим первое слагаемое. Распределение $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})$ не зависит от $\boldsymbol{\varphi}$:

$$\mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\nabla_{\boldsymbol{\varphi}} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] = -\mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}[\nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z})] = -\int \frac{\nabla_{\boldsymbol{\varphi}} q_{\boldsymbol{\varphi}}(\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \cdot q_{\boldsymbol{\varphi}}(\mathbf{z}) d\mathbf{z} = -\nabla_{\boldsymbol{\varphi}} \int q_{\boldsymbol{\varphi}}(\mathbf{z}) d\mathbf{z} = -\nabla_{\boldsymbol{\varphi}} 1 = 0.$$

- Воспользуемся *равенством*: $\nabla_{\boldsymbol{\varphi}} q_{\boldsymbol{\varphi}}(\mathbf{z}) = q_{\boldsymbol{\varphi}}(\mathbf{z}) \cdot \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z})$. Это позволяет переписать соотношение следующим образом:

$$\nabla_{\boldsymbol{\varphi}} \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\frac{\log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] = \int \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \cdot q_{\boldsymbol{\varphi}}(\mathbf{z}) \cdot \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z})\, d\mathbf{z} = \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \cdot \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z})\right].$$

Score function estimator (REINFORCE)

- Данное выражение носит название **score function эстиматора** (или эстиматора **REINFORCE**):

$$\nabla_{\boldsymbol{\varphi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}\left[\frac{\log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] = \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \cdot \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z})\right].$$

- Градиент матожидания представлен как *матожидание*, а это означает, что его можно оценить с использованием *эстиматора Монте-Карло*:

$$\nabla_{\boldsymbol{\varphi}} ELBO \approx \frac{1}{S}\sum_{s=1}^{S}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}_s)}{q_{\boldsymbol{\varphi}}(\mathbf{z}_s)} \cdot \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z}_s)\right].$$

- Для работы эстиматора требуется, чтобы $\log q_{\boldsymbol{\varphi}}(z)$ был дифференцируем по параметру $\boldsymbol{\varphi}$.
- Данный эстиматор *несмещённый*, но имеет *очень большую дисперсию*.

- Вычисление *градиента от эстиматора* Монте-Карло может быть сильно упрощено, если возможно выполнить *репараметризацию* вариационного распределения:

$$\mathbf{z} \sim q_{\boldsymbol{\varphi}}(\mathbf{z}), \qquad \mathbf{z} = g(\boldsymbol{\varphi}, \boldsymbol{\varepsilon}), \qquad \boldsymbol{\varepsilon} \sim \xi(\boldsymbol{\varepsilon}),$$

где $g(\boldsymbol{\varphi}, \boldsymbol{\varepsilon})$ является детерминированной функцией относительно *параметра* $\boldsymbol{\varphi}$ и *случайной переменной* $\boldsymbol{\varepsilon}$, причем распределение $\xi(\boldsymbol{\varepsilon})$ *не зависит* от $\boldsymbol{\varphi}$, а из него *легко получать семплы*.

- В этом случае, градиент можно *оценить*, *перейдя к переменной* $\boldsymbol{\varepsilon}$:

$$\nabla_{\boldsymbol{\varphi}} \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\frac{\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] = \nabla_{\boldsymbol{\varphi}} \mathbb{E}_{\xi(\boldsymbol{\varepsilon})}\left[\frac{\log p_{\boldsymbol{\theta}}\big(\mathbf{x}, g(\boldsymbol{\varphi}, \boldsymbol{\varepsilon})\big)}{q_{\boldsymbol{\varphi}}(g(\boldsymbol{\varphi}, \boldsymbol{\varepsilon}))}\right] = \mathbb{E}_{\xi(\boldsymbol{\varepsilon})}\left[\nabla_{\boldsymbol{\varphi}} \frac{\log p_{\boldsymbol{\theta}}\big(\mathbf{x}, g(\boldsymbol{\varphi}, \boldsymbol{\varepsilon})\big)}{q_{\boldsymbol{\varphi}}(g(\boldsymbol{\varphi}, \boldsymbol{\varepsilon}))}\right].$$

- Для этого функция $g(\boldsymbol{\varphi}, \boldsymbol{\varepsilon})$ должна быть *дифференцируема по $\boldsymbol{\varphi}$*.

Пример трюка с репараметризацией

- В качестве примера выберем *семейство нормальных распределений* в качестве *вариационного распределения*:

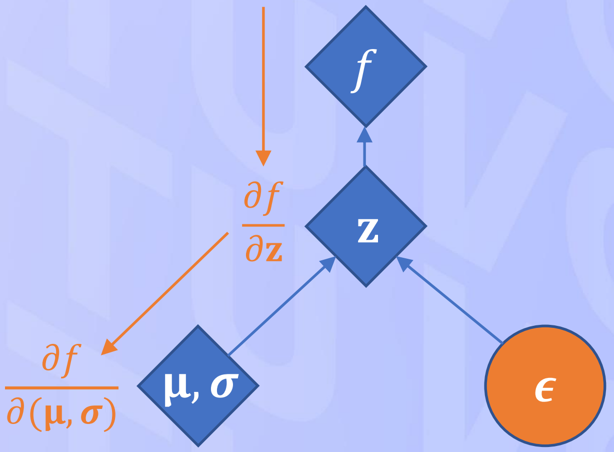
$$q_{\boldsymbol{\varphi}}(\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$$

- Известно, что *нормальное распределение* можно свести с *стандартному нормальному* с помощью *преобразования*:

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \mathcal{N}(\mathbf{0}, \mathbb{I}).$$

- Тогда $\mathbf{z} = g(\boldsymbol{\varphi}, \boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}$, $\boldsymbol{\varphi} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$.

$$\nabla_{\boldsymbol{\theta},\boldsymbol{\varphi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}\left[\frac{\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] = \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\mathbb{I})}\left[\nabla_{\boldsymbol{\theta},\boldsymbol{\mu},\boldsymbol{\sigma}} \frac{\log p_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon})}{q_{\boldsymbol{\phi}}(\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon})}\right].$$



- Однако, *не все распределения* имеют *представление* через $g(\boldsymbol{\varphi}, \boldsymbol{\varepsilon})$.

- Рассмотренные подходы к оценке градиента от ELBO обычно реализуют с помощью <u>эстиматоров</u>. Наиболее популярны <u>*два вида эстиматоров*</u>:

1. **Score function estimator** (likelihood ratio estimator, REINFORCE estimator) – применяется в общем случае, имеет <u>*большую дисперсию*</u>.

$$\nabla_{\boldsymbol{\varphi}} \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\frac{\log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})}\right] = \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z})}\left[\log\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\varphi}}(\mathbf{z})} \cdot \nabla_{\boldsymbol{\varphi}} \log q_{\boldsymbol{\varphi}}(\mathbf{z})\right].$$

2. **Pathwise estimator** – применяется в случае возможности применения <u>*трюка с репараметризацией*</u>. Имеет небольшую дисперсию, но требует <u>*представления распределения*</u> в виде:

$$\mathbf{z} \sim q_{\boldsymbol{\varphi}}(\mathbf{z}), \qquad \mathbf{z} = g(\boldsymbol{\varphi}, \boldsymbol{\epsilon}), \qquad \boldsymbol{\varepsilon} \sim \xi(\boldsymbol{\varepsilon}).$$

1. **Согласованность** (*Consistency*): при увеличении количества семплов из распределения результат работы эстиматора должен сходиться к истинному значению матожидания ELBO.
2. **Несмещённость** (*Unbiasedness*): результаты работы эстиматора должны являться центрированными относительно истинного значения матожидания ELBO.
3. **Минимальное значение дисперсии** (*Minimum variance*): из всех эстиматоров, обрабатывающих одинаковое количество семплов, предпочтителен эстиматор с наименьшим значением дисперсии.
4. **Эффективность** (Efficiency): предпочтителен эстиматор, требующий малого количества семплов и способный легко распараллеливаться.

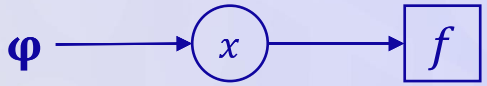
Оценка градиента с использованием стохастического графа

- Использование одного из видов эстиматоров определяется, как правило, возможностью *репараметризации распределений*.
- Поскольку оба эстиматора *несмещённые*, но score function эстиматор имеет *большую дисперсию*, чем pathwise, то стараются использовать последний. Score function эстиматор обычно применяется только если у распределения отсутствует эффективная *репараметризация*.
- Чаще всего при расчете градиентов от ELBO применяют оба эстиматора, каждый для своего множества параметров: $\mathbf{z}' \sim q_{\boldsymbol{\varphi}'}(\mathbf{z}')$ и $\mathbf{z}'' \sim q_{\boldsymbol{\varphi}''}(\mathbf{z}'')$.

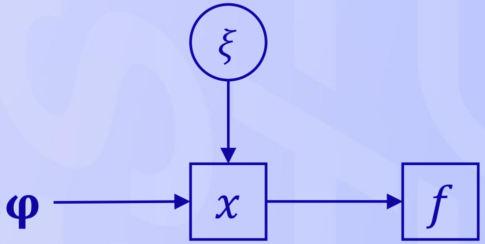
$$\nabla_{\boldsymbol{\varphi}} ELBO = \begin{bmatrix} \nabla_{\boldsymbol{\varphi}'} ELBO \\ \nabla_{\boldsymbol{\varphi}''} ELBO \end{bmatrix}.$$

Понятие стохастического графа

- Для удобства работы с градиентом от ELBO для произвольных моделей и распределений вводят понятие **стохастического вычислительного графа** – направленного ациклического графа со следующими типами вершин:
  - ✓ _Входные_ вершины Θ, включающие параметры, по которым рассчитывается градиент.
  - ✓ _Детерминированные_ вершины $\mathcal{D}$, определяемые как детерминированные функции.
  - ✓ _Стохастические_ вершины $\mathcal{S}$, осуществляющие реализацию семплов из распределения.
  - ✓ Вершины _функций ошибки_ $\mathcal{C}$, в которых заданы слагаемые функции ошибки (обычно *ELBO*).
- При визуализации _стохастические_ вершины обозначаются _кружком_, а _детерминированные_ – _прямоугольником_:



Score function estimator



Pathwise estimator

Оценка градиента с использованием стохастического графа

- Для *оценки градиента* от ELBO на *стохастическом графе* вводится <u>*суррогатная функция*</u>. Она является *дифференцируемой по параметрам* и вводится следующим образом:

$$\nabla_{\boldsymbol{\varphi}} ELBO = \nabla_{\boldsymbol{\varphi}} \mathbb{E}\left[\sum_{c \in \mathcal{C}} c\right] = \mathbb{E}\left[\nabla_{\boldsymbol{\varphi}} L(\Theta, \mathcal{S})\right],$$

$$L(\Theta, \mathcal{S}) = \sum_{\substack{\mathbf{w} \in \mathcal{S} \\ \boldsymbol{\varphi} \prec^D \mathbf{w}}} \log p(\mathbf{w} \mid \mathrm{DEPS}_{\mathbf{w}}) \cdot \hat{Q}_{\mathbf{w}} + \sum_{c \in \mathcal{C}} c(\mathrm{DEPS}_c),$$

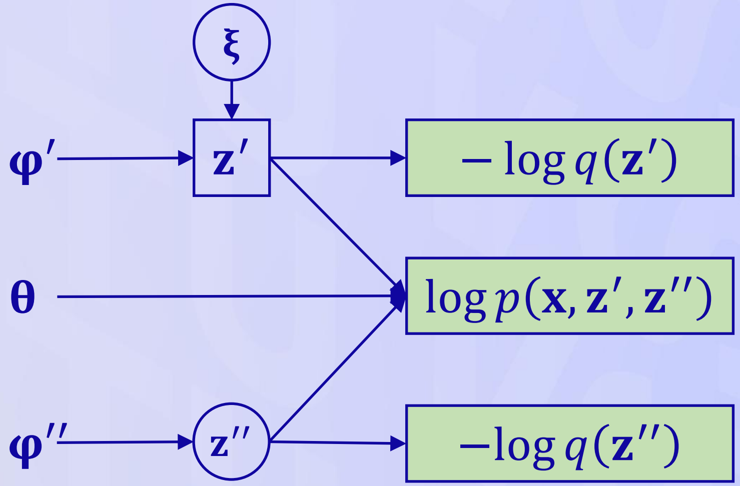
$v \prec^D w$ – детерминированное влияние v на w;
$v \prec w$ – влияние v на w;
$DEPS_c = \{w \in \Theta \cup \mathcal{S} : w \prec^D c\}$

- Если структура графа <u>*не учитывается*</u>, то $\hat{Q} = \sum_{c \in \mathcal{C}} c$,
- <u>Учет структуры графа</u> позволяет записать: $\hat{Q} = \hat{Q}_w = \sum_{\substack{c \in \mathcal{C} \\ w \prec c}} c$.

- Учет структуры графа при расчете *суррогатной функции* позволяет применить теорему Рао-Блеквелла (*Rao-Blackwell theorem*):
  - ✓ Пусть $\hat{\theta}_n$ – несмещенный эстиматор некоторого параметра $\boldsymbol{\theta} \in \Theta$, и пусть $T(\mathbf{X})$ – эффективная статистика для $\boldsymbol{\theta}$. Тогда:
    - ➢ $\tilde{\theta}_n = \mathbb{E}\left[\hat{\theta}_n \mid T(\mathbf{X})\right]$ – тоже несмещенный эстиматор $\boldsymbol{\theta}$,
    - ➢ $D_\theta\left[\tilde{\theta}_n\right] \leq D_\theta\left[\hat{\theta}_n\right] \quad \forall\boldsymbol{\theta}$.
- Под *эффективной статистикой* понимают функцию *выборки*, применение которой к оценке параметра будет *столь же эффективным*, как всей выборки
- Используя данную теорему, *можно показать*, что рассмотренный ранее эстиматор на основе стохастического графа имеет *меньшую дисперсию*.
- Такой *подход* к уменьшению дисперсии называется Rao-Blackwellization.

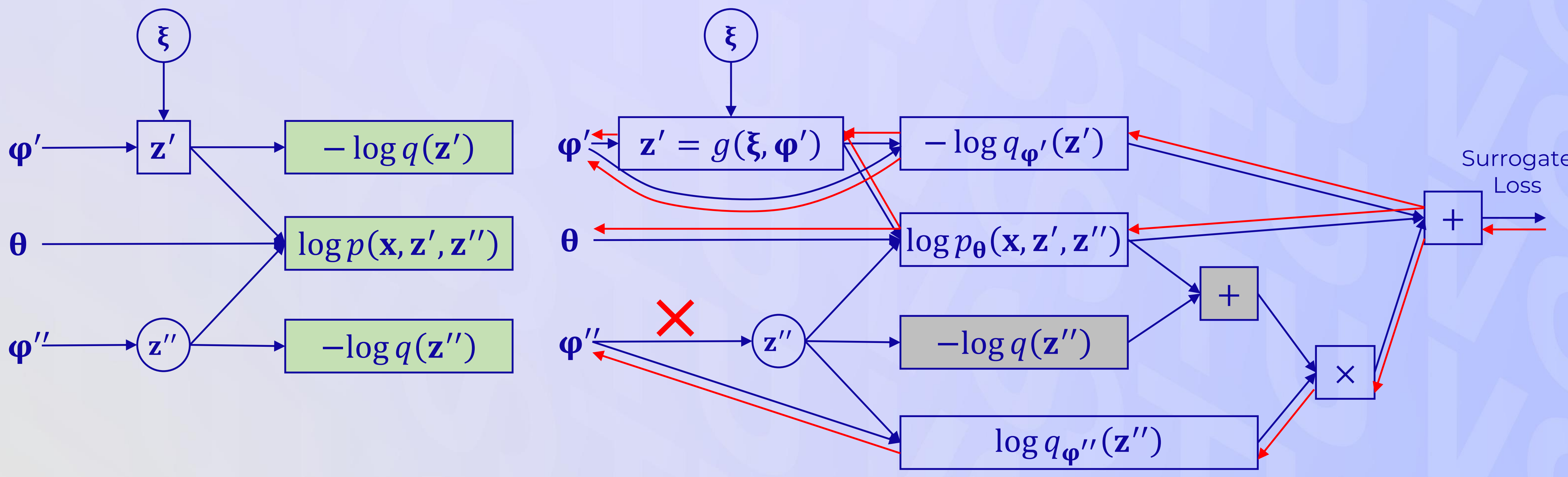
Оценка градиента с использованием стохастического графа

- Для расчета градиентов от ELBO *стохастический граф* преобразуется к *детерминированному вычислительному графу*, в который добавляются узлы, соответствующие *суррогатной функции ошибки*.
- Например, пусть $q_{\boldsymbol{\varphi}}(\mathbf{z})$ состоит из случайных величин, одно из которых репараметризуется, а другое – нет: $q_{\boldsymbol{\varphi}}(\mathbf{z}) = q_{\boldsymbol{\varphi}'}(\mathbf{z}') \cdot q_{\boldsymbol{\varphi}''}(\mathbf{z}'')$.



Оценка градиента с использованием стохастического графа

- Сведём представленный граф к *детерминированному вычислительному*.



Способ уменьшения дисперсии эстиматора REINFORCE

- Введем следующее <u>*расширение*</u> *эстиматора REINFORCE* $\hat{\mathcal{F}}$, добавив <u>константу</u> β (по $\boldsymbol{\phi}$):

$$\hat{\mathcal{F}} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}\left[\left(\log\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z})} - \beta\right)\cdot\nabla_{\boldsymbol{\phi}}\log q_{\boldsymbol{\phi}}(\mathbf{z})\right].$$

- Данный *эстиматор* можно представить в виде:

$$\hat{\mathcal{F}} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}\left[\log\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z})}\cdot\nabla_{\boldsymbol{\phi}}\log q_{\boldsymbol{\phi}}(\mathbf{z})\right] - \beta\cdot\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}[\nabla_{\boldsymbol{\phi}}\log q_{\boldsymbol{\phi}}(\mathbf{z})].$$

$$\hat{\mathcal{F}} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}\left[\log\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z})}\cdot\nabla_{\boldsymbol{\phi}}\log q_{\boldsymbol{\phi}}(\mathbf{z})\right].$$

- Таким образом, введённый эстиматор <u>*эквивалентен*</u> *REINFORCE* для $\forall\beta$.
- Переменные β называются **baselines**. <u>*Можно показать*</u>, что правильный подбор β <u>*снижает дисперсию*</u> эстиматора, оставляя его <u>*несмещённым*</u>.

Способы определения baselines в Pyro

1. Baselines на основе <u>*скользящего среднего*</u> функции $\log\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z})}$ для *различных семплов*.

```
z = pyro.sample("z", dist.Bernoulli(...),
                infer=dict(baseline={'use_decaying_avg_baseline': True,
                                     'baseline_beta': 0.95}))
```

2. Baselines на основе <u>нейронной сети</u> (*neural baselines*).

```
z = pyro.sample("z", dist.Bernoulli(...),
                    infer=dict(baseline={'nn_baseline': baseline_module,
                                         'nn_baseline_input': x}))
```

где <u>baseline_module</u> – модуль <u>*нейросети*</u> Pyro с <u>*функционалом ошибки*</u>:

$$\mathcal{L} = \left(\log\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z})} - \beta\right)^2.$$

Стохастический вариационный инференс для дискретных переменных

- Рассмотренный ранее алгоритм *стохастического вариационного инференса* применим к *непрерывно изменяющимся переменным*.
- Как выполнить *вариационный инференс* для *дискретных переменных*?
- Применение алгоритмов *градиентного спуска* в случае дискретных переменных $\boldsymbol{\xi} = \boldsymbol{\xi}_{\boldsymbol{\phi}}$ *невозможно*, поскольку производная в этом случае не определена:

$$\frac{\partial \mathrm{ELBO}}{\partial \boldsymbol{\phi}} = \frac{\partial \mathrm{ELBO}}{\partial \boldsymbol{\xi}} \cdot \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\phi}}.$$

- Возможно применение *score function estimator.*
- Для *некоторых распределений* возможен *трюк с репараметризацией*, позволяющий свести данное *дискретное распределение* к непрерывному с близкими свойствами, по которому производная *легко вычисляется*.

Категориальное распределение и функция softmax

- Рассмотрим *репараметризацию*, позволяющую выполнить инференс для *дискретной переменной*, имеющей *категориальное распределение*.

$$\mathrm{P}(\xi_\varphi) = \mathcal{C}at(\xi; \varphi) = \begin{cases} \pi_1(\varphi), & \text{если } \xi = \xi_1, \\ \pi_2(\varphi), & \text{если } \xi = \xi_2, \\ & \dots \\ \pi_m(\varphi), & \text{если } \xi = \xi_m. \end{cases} \qquad \sum_i \pi_i = 1, \qquad \pi_i \geq 0 \ \forall i.$$

- Вспомним функцию, которая является *дифференцируемым аналогом* операции *argmax*. Она носит название *softmax*:

$$\mathbf{s} = \mathrm{softmax}(\mathbf{x}) = \frac{e^{\mathbf{x}}}{\sum_j e^{x_j}} = \frac{1}{\sum_j e^{x_j}} [e^{x_1}, e^{x_2}, \dots, e^{x_m}]^{\mathrm{T}}. \qquad \sum_i s_i = 1, \qquad s_i \geq 0 \ \forall i, \qquad x_i \in (-\infty; +\infty).$$

- Softmax может использоваться для *оценки* вектора вероятностей $\boldsymbol{\pi}(\phi)$.

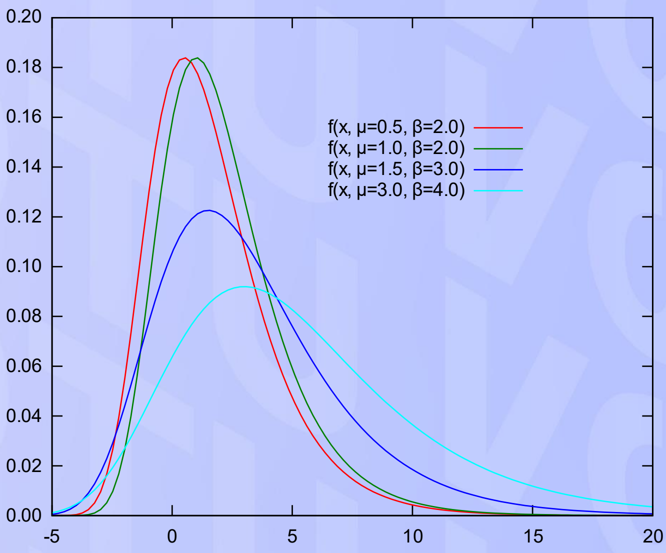
Трюк репараметризации Gumbel-Max

- Введем следующую *функцию*:

$$\xi = \text{one\_hot}\left(\arg\max_i [g_i + \log \pi_i]\right).$$

в которой π_i – рассмотренные ранее вероятности классов *категориального распределения*, а g_i – *семплы* из распределения Гумбеля (Gumbel):

$$g_i \sim \text{Gumbel}(0,1).$$



- *Семплирование* из категориального распределения можно заменить семплированием из функции ξ.
- Данный способ организации семплирования из *категориального распределения* называется *трюком репараметризации Gumbel-Max*.

- *Функция распределения* Гумбеля:

$$F = e^{-e^{-\frac{x-\mu}{\beta}}}.$$

- Отсюда можно получить *обратную функцию распределения*:

$$\ln F = -e^{-\frac{x-\mu}{\beta}} \quad \Longrightarrow \quad e^{-\frac{x-\mu}{\beta}} = -\ln F \quad \Longrightarrow \quad \frac{x-\mu}{\beta} = -\ln(-\ln F).$$

$$F^{-1} = x = -\beta \cdot \ln(-\ln F) + \mu.$$

- Таким образом, семплирование из *распределения Гумбеля* $\mathrm{Gumbel}(0,1)$, согласно *теореме об обратной функции*, выразится следующим образом:

$$g_i \sim -\ln(-\ln(\mathcal{U}[0,1])).$$

- И тогда:

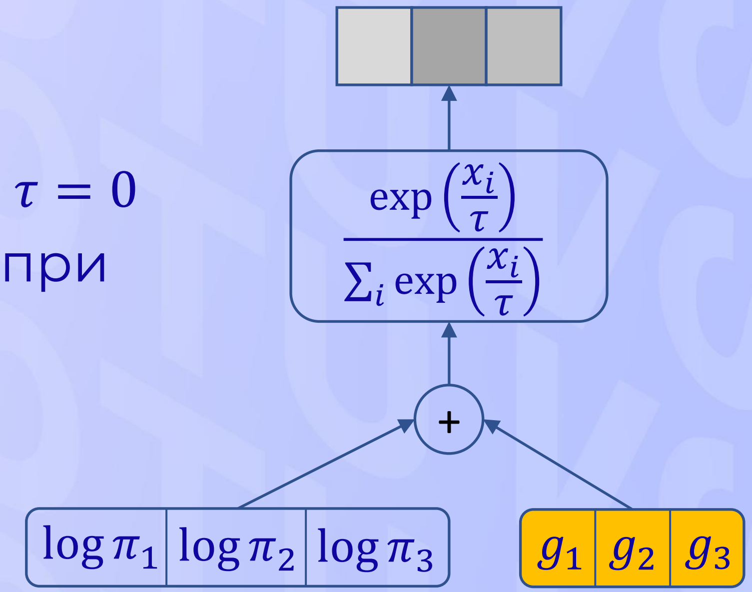
$$\xi = \mathrm{one\_hot}\left(\arg\max_i[\log\pi_i - \ln(-\ln(\mathcal{U}[0,1]))]\right).$$

Трюк репараметризации Gumbel-Softmax

- Однако, функция *argmax* тоже не является *дифференцируемой*.
- Идея: заменить данную функцию её *дифференцируемым аналогом*, и вместо *категориального* использовать для семплирования распределение:

$$y_i = \frac{e^{\frac{g_i + \log \pi_i}{\tau}}}{\sum_{j=1}^{m} e^{\frac{g_i + \log \pi_i}{\tau}}},$$

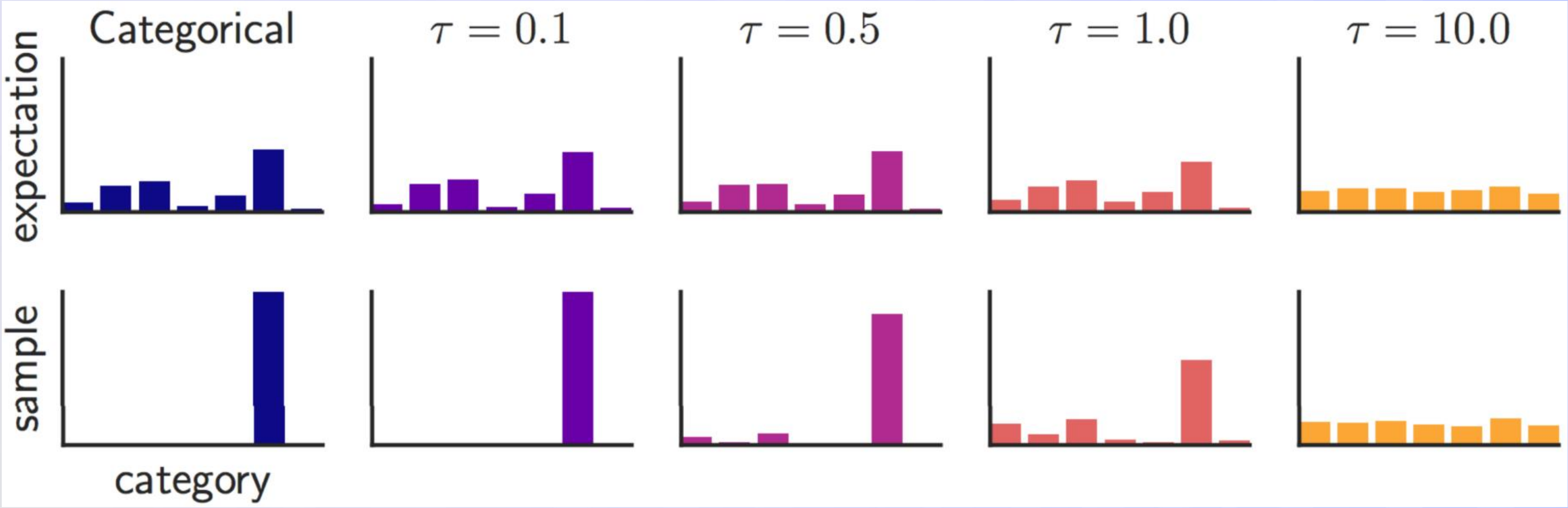
где τ называется *температурным* параметром. При $\tau = 0$ распределение тождественно *категориальному*, а при $\tau = \infty$ оно становится *равномерным*.



- Данное преобразование называется *трюком репараметризации* Gumbel-Softmax.
- Говорят, что $\mathbf{y}$ – это *continuous relaxation* of $\boldsymbol{\xi}$.

Особенности Gumbel-Softmax эстиматора

- При $\tau > 0$ рассматриваемое распределение <u>*не является тождественным категориальному*</u>. Поэтому, *эстиматор градиентов* является <u>*смещённым*</u>.
- Для <u>малых</u> τ градиенты имеют <u>*высокую дисперсию*</u>. Подбор τ позволяет *регулировать* требуемые *смещение* и *разброс* (*variance-bias trade-off*).



- Применение репараметризации Gumbel-Softmax создает распределение, *не тождественное категориальному*. Однако, данная репараметризация вводится *только* с целью *вычислить производные* в процессе оптимизации.
- Идея: *семплирование* следует выполнять из *категориального* распределения, а *расчет градиентов* проводить с использованием *дифференцируемой функции*, введённой посредством *репараметризации*.
- Данный способ расчета реализует *Straight-Through эстиматор*.
- Этот *смещённый эстиматор* использует *argmax*-версию распределения при семлировании (*forward pass*) и *softmax*-версию при расчете градиентов (*backward pass*).
- Т.о., в эстиматоре реализуется *аппроксимация*: $\nabla_{\boldsymbol{\phi}}\boldsymbol{\xi} \approx \nabla_{\boldsymbol{\phi}}\mathbf{y}$.

- *Straight-Through Gumbel-Softmax эстиматор* во фреймворке Pyro реализуется посредством *распределения*:

```
class RelaxedOneHotCategoricalStraightThrough(temperature, probs=None, logits=None, validate_args=None)
```

- Данное распределение имеет следующие *особенности*:
  - Метод *rsample*() возвращает семпл из *дискретного* распределения.
  - Метод *log_prob*() возвращает логарифм вероятности для семпла из *непрерывного* распределения с использованием *Gumbel-Softmax*.
  - *Градиент по параметрам* рассчитывается с использованием семплов из *непрерывного* распределения.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Сравнение эстиматоров

Эстиматор	Непрерывность z	Поддерживает backprop	Точность аппроксимации $q(z)$	Требуется существование $\frac{\partial f}{\partial z}$
Score function	Непрерывный или дискретный	**Да**	Точный	**Нет**
Pathwise	Непрерывный	**Да**	Точный	Да
Gumbel-max	Дискретный	Нет	**Точный**	
Gumbel-softmax	Дискретный	**Да**	Смещённый	Да
Straight-through Gumbel	Дискретный	**Да**	Точный при forward, смещенный при backward	Да

- Существует *два основных подхода* к выполнению *байесовского вывода*:
  - **Sample-based методы** (на основе *марковских цепей*). Данные методы обеспечивают *точное решение* задачи байесовского вывода, но требуют *большого времени моделирования* и *серьёзных вычислительных ресурсов*. Это существенно *ограничивает* применение этих методов для сложных моделей с большим количеством латентных переменных (в частности, для *нейросетей*).
  - **Вариационный инференс**. Является менее требовательным к вычислительным ресурсам, но обеспечивает лишь *приближенное решение* задачи вывода. *Точность* решения определяется выбранным *вариационным распределением*.

VOLGOGRAD
STATE
TECHNICAL
UNIVERSITY

Демонстрация практических примеров

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Заключение

1. Поговорили про различные способы оценки градиента от ELBO.
2. Ввели понятие репараметризации случайных величин.
3. Рассмотрели эстиматоры градиента ELBO и поговорили об их свойствах.
4. Поговорили о способах уменьшения дисперсии эстиматоров, ввели baselines для снижения дисперсии эстиматора REINFORCE.
5. Определили способ выполнения вариационного инференса с дискретными переменными, разобрались с трюками Gumbel-Max и Gumbel-Softmax.
6. Рассмотрели особенности эстиматоров Gumbel-Softmax и Straight-Through.
7. Поговорили об организации эстиматоров в Pyro и на практических примерах разобрали их применение.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Спасибо за внимание!

Волгоград 2025