УДК 004.89 (519.24)

НЕЙРОБАЙЕСОВСКИЕ МЕТОДЫ

Г. С. Афанасенко

Белорусский государственный университет, Беларусь, Минск, redstray12@yandex.ru

Данная статья основана на результатах курсовой работы. Здесь рассматриваются основные концепции байесовских нейронных сетей, их математические основы и реализация в виде программного фреймворка. Приводится теоретическое обоснование байесовского подхода, рассматриваются его отличия от частотного подхода в статистике. Описывается процесс построения байесовских нейронных сетей, а также реализованный фреймворк для их обучения, выполненный на C++. Представлены результаты тестирования фреймворка на популярных наборах данных, таких как FashionMNIST и CIFAR10.

Ключевые слова: байесовская статистика; байесовские нейронные сети; вариационный вывод; вариационная нижняя оценка; C++ биндинги; нейронные сети; байесовский подход.

NEUROBAYESIAN METHODS

G. S. Afanasenko

Belarussian state university, Belarus, Minsk, redstray12@yandex.ru

This article discusses the basic concepts of Bayesian neural networks, their mathematical foundations and implementation in the form of a software framework. The theoretical justification of the Bayesian approach is given, and its differences from the frequency approach in statistics are considered. The process of building Bayesian neural networks is described, as well as an implemented framework for their training in C++. The results of testing the framework on popular datasets such as FashionMNIST and CIFAR10 are presented.

Keywords*:* Bayesian statistics; Bayesian neural networks; variational inference; variational lower bound; C++ bindings; neural networks; Bayesian approach.

Введение

В современном мире развитие нейронных сетей и глубинного обучения происходит очень быстро. Регулярно появляются новые методы обучения нейронный сетей, новые архитектуры, новые слои или приёмы, каждые из которых имеют свои особенные полезные эффекты и поведение. К сожалению, из-за многомерности данных, их сложной структуры и общей

сложности задач обучения нейронных сетей, люди всё меньше и меньше могут объяснить, почему тот или иной трюк, слой работают. Получение истинного понимания принципа работы нейронных сетей позволяют усовершенствовать уже существующие методы и изобрести новые. По этой теме сделано большое количество исследований и работ на английском языке.

Целью курсовой работы будет разобрать самые базовые концепции и понятия нейробайесовских методов, собрать их в одной работе и предоставить весь материал, чтобы другие заинтересованные люди могли ознакомиться с ними на русском языке. Более подробно, в этой работе будет проведён разбор со всеми математическими выкладками ряда самых известных зарубежных статей по этой теме с дальнейшим объяснением их на более понятном языке. Как результат, курсовая работа будет собой представлять небольшой учебник по нейробайесовским методам.

Теоретические сведения

Байесовская статистика отличается от частотной трактовки вероятностей тем, что вероятность в байесовском подходе интерпретируется как степень уверенности в истинности гипотезы. Основной инструмент — теорема Байеса:

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{P(D)} \tag{1}$$

$$p(\theta, D_x, D_y) = p(\theta \mid D_x, D_y)p(D_x, D_y) \tag{2}$$

Рассмотрим вероятностную модель (2), где $D_x = (x_1, x_2, ..., x_n), D_y = (y_1, y_2, ..., y_n), n \gg 1, \theta \in \mathbb{R}^d$

Использование байесовского подхода вместо частотного позволяет перейти от точечной оценки параметров вероятностных моделей к оценке их распределения в условиях неопределённости.

Для получение апостериорного распределения будем пробовать приблизить апостериорное распределение $p(\theta \mid D_x, D_y)$ параметризованным распределением $q(\theta \mid \varphi)$ путём минимизации дивергенции Кульбака-Лейблера $KL(q(\theta \mid \varphi) \| p(\theta \mid D_x, D_y))$. То есть

$$\hat{\varphi} = \arg\max_{\varphi} KL(q(\theta \mid \varphi) \| p(\theta \mid D_x, D_y)) \tag{3}$$

Для того, чтобы считать значение самой функции и её градиента, нам бы потребовалось уметь считать значение $p(\theta \mid D_x, D_y)$, которое мы и пытаемся найти. Поэтому требуется найти альтернативный оптимизируемый функционал. Если мы распишем KL-дивергенцию, то получим (4).

$$KL(q(\theta \mid \varphi) \| p(\theta \mid D_x, D_y)) = \\ = \underbrace{\mathbb{E}_{\theta \sim q(\theta|\varphi)}[\log q(\theta \mid \varphi)] - \mathbb{E}_{\theta \sim q(\theta|\varphi)}[\log p(\theta, D_x, D_y)]}_{-ELBO(\varphi, D)} + \log p(D_x, D_y) \tag{4}$$

Получившееся значение $ELBO(\varphi, D)$ называется вариационной нижней оценкой(Evidence Lower Bound). Оно также записывается, как $L(\varphi, D)$. С учётом свойств вариационной нижней оценки задача о минимизации дивергенции Кульбака-Лейблера по параметрам φ эквивалентна задаче о максимизации вариационной нижней оценки по тем же параметрам φ.

Для оптимизации вариационной нижней оценки воспользуемся градиентным спуском. Наложив определённые условия на вероятностную модель, мы получим значение градиента (5).

$$\frac{\partial(-L(\varphi))}{\partial \varphi} = \mathbb{E}_{q(\theta|\varphi)}[\frac{\partial \log q(\theta \mid \varphi)}{\partial \varphi}(\log q(\theta \mid \varphi) - \log p(D_y \mid \theta, D_x) - \log p(\theta))] \tag{5}$$

Для подсчёта интеграла приблизим его с помощью метода Монте-Карло [1]. Описанный метод оптимизации (6) называется стохастический вариационный вывод и является самым базовым подходом. Однако, такой метод имеет большой недостаток — большую дисперсию значений. Позднее было предпринято ряд попыток улучшения метода [2], [3].

$$\theta_k \sim q(\theta \mid \varphi), k = \overline{1 \ldots K} \\ -\text{grad}L(\varphi) \approx \frac{1}{K} \sum_{k=1}^{K} \frac{\partial \log q(\theta \mid \varphi)}{\partial \varphi}(\log q(\theta_k \mid \varphi) - \log p(D_y \mid \theta_k, D_x) - \log p(\theta_k)) \tag{6}$$

Рассмотрим метод [2], который называется «трюк с репараметризацией». Основная идея метода заключается в том, чтобы репараметризировать оцениваемые параметры (7). После чего мы можем переписать градиент с учётом этой репараметризации (8)

$$\theta = g(\varepsilon, \varphi), \varepsilon \sim r(\varepsilon) \tag{7}$$

$$\varepsilon_k \sim r(\varepsilon), k = \overline{1...K}$$
$$\theta_k = g(\varepsilon_k, \varphi)$$
$$\frac{\partial(-L(\varphi))}{\partial\varphi} \approx \frac{1}{K}\sum_{k=1}^{K}\frac{\partial}{\partial\varphi}(\log q(\theta_k \mid \varphi) - \log p(D_y \mid \theta_k, D_x) - \log p(\theta_k)) \tag{8}$$

Такой метод сильно упрощает процесс обучения с помощью градиентного спуска, и позволяет использование механизма автодифференцирования из любого популярного фреймворка. Также такой метод имеет на порядки меньшую дисперсию. Существует ряд других подходов, позволяющих уменьшить дисперсию стохастического градиента, один из которых мы также рассмотрим ниже.

Предыдущие теоретические выкладки применимы во многих областях глубинного обучения и обучения с подкреплением. В частности, обыкновенный стохастический вариационный вывод лёг в основу алгоритма REINFORCE, который является разновидностью policy-gradient алгоритма в обучении с подкреплением. Кроме того, возможно использование полученных формул для объяснения устройства работы Dropout слоя в нейронных сетях.

Байесовские нейронные сети

Использование нейробайесовских методов не ограничивается теорией. Их использование возможно на практике при разработке новых архитектур нейронных сетей. К примеру, возможно применение байесовского подхода для линейного слоя нейронной сети. Как известно, обыкновенный линейный слой можно записать в виде (9).

$$y = Wx + b, x \in \mathbb{R}^n, y \in \mathbb{R}^m, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \tag{9}$$

$$\begin{aligned} &W \sim p(W \mid x) \\ &b \sim p(b \mid x) \\ &y = Wx + b \end{aligned} \tag{10}$$

Если применить байесовский подход к линейному слою, то получим (10). Как и раньше мы хотим приблизить истинные апостериорные распределения параметров с помощью некоторых параметрических $q(W \mid \varphi_W), q(b \mid \varphi_b)$ и затем использовать их в результате функционирования нейронной сети. Обучаем такой слой по тем же формулам градиентов, которые указаны

выше. Однако, в [3] предложили другой способ, который позволяет ещё сильнее уменьшить дисперсию стохастического градиента за счёт решения проблем с появляющимися ковариациями между разными примерами одного батча при обучении. Этот метода называется локальная репараметризация. Основная идея в том, чтобы применять метод Монте-Карло не к параметрам, а сразу к результату работы слоя (11)

$$y \sim p(y \mid x) \tag{11}$$

Это возможно не во всех случаях, но наиболее часто используемый случай — это нормальные распределения, которые позволяют сделать локальную репараметризацию.

Фреймворк на C++

Реализация фреймворка происходила на C++20 с дальнейшим биндингом этого фреймворка в Python 3.12, как полноценного пакета-расширения. В качестве системы сборки плюсовой части использовался CMake, а для биндинга внутрь Python — PyBind11. Чтобы создать пакет-расширение с возможностью дальнейшего распространения в интернете требуется создать wheel-пакет с расширением .whl, который будет возможно установить обычной командой pip install *.whl. Для сборки такого wheel-пакета будет использован scikit-build 0.18, который автоматически компилирует C++ часть проекта и собирает её в файл общей библиотеки .so, а также собирает итоговой .whl-пакет, в который подсоединяет файлы общей библиотеки.

С точки зрения функционала фреймворка, он покрывает базовые потребности в обучении нейронных сетей подобно уже известным PyTorch, TensorFlow. При проектировании архитектуры было уделено внимание тому, чтобы интерфейс функций, классов и методов был знаком всем пользователям уже известных библиотек.

Заключение

В ходе работы были разобраны основные научные статьи по нейробайесовским методам, выполнены математические выкладки и объяснения на русском языке.

Дано объяснение о том, что из себя представляют байесовские сети и байесовские нейронные сети, в том числе сравнение с классическими нейронными сетями. Также было дано несколько обоснований полезности

нейробайесовских методов, которые были подкреплены практическими экспериментами на наборах данных CIFAR10, FashionMNIST.

Также был реализован основной функционал фреймворка для обучения нейронных сетей, который, однако, требует будущей доработки.

Библиографические ссылки

1. *Ranganath R.* Black box variational inference. 17th International Conference on Artificial Intelligence and Statistic. 2014.

2. *Diederik P. Kingma, Max Welling.* Auto-encoding variational bayes. 2nd International Conference on Learning Representations. 2014.

3. *Diederik P. Kingma, Tim Salimans, and Max Welling.* Variational dropout and the local reparameterization trick. In Proceedings of the 28th International Conference on Neural Information Processing Systems 2015. № 28.