ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

На правах рукописи

Неклюдов Кирилл Олегович

БАЙЕСОВСКИЙ ПОДХОД В ГЛУБИННОМ ОБУЧЕНИИ: УЛУЧШЕНИЕ ДИСКРИМИНАТИВНЫХ И ГЕНЕРАТИВНЫХ МОДЕЛЕЙ

РЕЗЮМЕ
диссертации на соискание учёной степени
кандидата компьютерных наук

Москва — 2020

Диссертационная работа выполнена в федеральном государственном автономном образовательном учреждении высшего образования «Национальный исследовательский университет «Высшая школа экономики».

Научный руководитель: Ветров Дмитрий Петрович, к.ф.-м.н., профессор-исследователь, Национальный исследовательский университет «Высшая школа экономики».

Тема диссертации

В этой работе используется формализм Байесовской статистики для улучшения существующих моделей глубокого обучения различными способами. Основываясь на технике дважды-стохастического вариационного вывода [1], данная работа предлагает две вероятностные модели для глубоких дискриминативных сетей. Первая модель позволяет получить структурированную разреженность сверточных нейронных сетей (CNN) и, как следствие, их ускорение. Вторая модель улучшает оценку неопределенности в задаче классификации с помощью общепринятых архитектур CNN. Также в работе рассматриваются генеративные модели глубинного обучения. Рассматривая проблему генерации с точки зрения методов Монте Карло с Марковскими цепями (MCMC), в настоящей работе предлагается алгоритм, который улучшает производительность генеративных соревновательных сетей (GAN) [2]. А именно, в работе предлагается неявный алгоритм Метрополиса-Гастингса и проводится его асимптотический анализ. Этот алгоритм можно рассматривать как адаптацию обычного алгоритма Метрополиса-Гастингса [3] на случай когда предложное распределение задано неявно, а целевое распределение эмпирически.

Актуальность работы.

Машинное обучение — это научный подход, позволяющий строить модели (алгоритмы) на основе данных. Алгоритмы машинного обучения находят свое практическое применение в задачах, где решение не может быть строго формализовано или явно запрограммировано. Основными областями применения являются компьютерное зрение, моделирование естественных языков, распознавание речи. Здесь мы предоставляем неформальное описание концепции машинного обучения. Для точной формулировки, а также примеров алгоритмов и приложений мы отсылаем читателя к [4; 5].

Классическая дихотомия машинного обучения это разделение между обучением с учителем и обучением без учителя. Целью обучения с учителем является построение функции из объектов (обычно описываемых действительнозначными векторами, называемыми *признаки*), в *метки* (также действительнозначные вектора в наиболее общем виде). Далее мы будем называть эту функцию *модель*. Чтобы построить такую модель, обычно ее формулируют параметрическим образом, то есть определяют функцию (например, аналитическую) или алгоритм, который выдает

предсказание метки основываясь на входных значениях признаков и всех значениях параметров. Далее, среди всех возможных значений параметров мы выбираем конфигурацию, наиболее подходящую для нашей задачи. Процесс выбора называется *обучение*. Для обучения, обычно определяют *функцию потерь* в совместном пространстве предсказаний и меток. Функция потерь определяет меру «правильности» прогноза, например, она равна нулю, если прогноз верен и растет с величиной его ошибки. При заданной функции потерь и *обучающей* выборке (подмножество объектов с метками из генеральной совокупности данных), процесс обучения может быть сформулирован как задача оптимизации в пространстве параметров. То есть нам нужно найти такую конфигурацию параметров, которая дает минимум функции потерь на обучающей выборке. Обычно эта конфигурация находится методами градиентной оптимизации.

Основная цель обучения без учителя — обнаружить внутренюю структуру данных. Другими словами, для каждого объекта (его признакового описания) в имеющихся данных нам необходимы найти скрытую переменную, которая наилучшим образом описывает этот объект. Однако точное понятие о структуре (пространстве скрытых переменных) существенно зависит от природы данных и предполагаемого использования скрытых переменных. Чтобы снабдить читателя интуицией, мы кратко опишем несколько примеров задач обучения без учителя. Задача кластеризации является одной из наиболее распространенных задач в обучении без учителя и может быть неформально описана как «нахождение меток без меток в обучающих данных». То есть, каждому объекту обучающей выборки нам нужно присвоить метку, где метка является дискретной переменной, а объекты с одинаковой меткой образуют кластер. Иногда нам также необходимо определить структуру множества меток (его мощность и отношения порядка). Другой популярный подход в обучении без учителя — авто-кодировщики. Они кодируют объект в пространство скрытых представлений, а затем декодируют полученное скрытое представление обратно в исходное пространство, пытаясь восстановить исходный объект как можно точнее. Пространство скрытых представлений обычно выбирают исходя из желаемых свойств. Например, авто-кодировщик может “сжимать” объекты, отображая их в низкоразмерные скрытые представления. В контексте текущей работы, наиболее важными примерами обучения без учителя являются генеративные модели. В генеративных моделях мы предполагаем, что наблюдаемые эмпирические данные (обучающая выборка) представляют собой подмножество объектов из некоторо-

го неизвестного распределения. Тогда наша цель заключается в том чтобы восстановить это неизвестное распределение путем создания модели, которая может генерировать объекты из этого распределения. Наиболее популярными подходами к изучению таких моделей являются контрастная дивергенция [6], вариационные авто-кодировщики [7; 8], генеративные соревновательные сети [2], нормализующие потоки [9].

Основополагающим моментом в разработке методов машинного обучения и их приложений является выбор признакового описания объектов. Одни из самых сложных объектов для анализа — это изображения, тексты, звуки из-за их высокой размерности и сложной внутренней структуры. Для таких данных, до выхода прорывной работы [10], признаковые описания строились только на основе экспертных знаний в соответствующей области. Например, в компьютерном зрении, признаки SIFT [11] и HOG [12] были общепринятыми стандартами описаниями изображений. Глубинное обучение [13; 14] автоматизировало процесс построения признакового описания объектов. Основная идея этого подхода заключается в построении многоуровневой модели для извлечения представлений данных с несколькими уровнями абстракции. Модели глубокого обучения в значительной степени полагаются на искусственные нейронные сети, которые также известны как универсальные аппроксиматоры. Однако способность аппроксимировать любую функцию недостаточна для построения хорошей модели. Модель также должна учитывать природу данных, например, CNNs [15] являются инвариантными к сдвигам, что делает их хорошо подходящими для изображений, а LSTM [16] препятствует затуханию градиента на длинных последовательностей, что делает их подходящими для текстов.

В этой работе мы в значительной степени полагаемся на формализм Байесовских рассуждений [17]. В качестве примера Байесовской модели, мы рассмотрим задачу обучения с учителем. Как и в общей постановке, у нас есть модель, которая определяет *правдоподобие*, например, вероятность получения правильной метки для заданного объекта и конкретных значений параметров. Помимо модели, мы также задаем *априорное* распределение параметров, которое содержит наши первоначальные знания о задаче. При заданных правдоподобии и априорном распределении, на этапе обучения мы теперь хотим найти не единственную конфигурацию параметров, а распределение в пространстве параметров (называемое *апостериорным*). Такой подход имеет ряд преимуществ, некоторые из которых мы перечисляем далее. Прежде всего, если точно определить

апостериорное распределение, то вместо одной модели можно использовать ансамбль моделей (веса пропорциональны плотности апостериорного распределения), который дает лучшее качество. Во-вторых, используя априорное распространение, можно инкорпорировать первоначальные знания о задаче в модель. В-третьих, можно выполнять инкрементальное обучение, включив информацию о ранее полученных данных в априорное распределение. Чтобы подчеркнуть основные принципы Байесовского вывода, мы приводим его связь с принципом максимальной энтропии [18], который является наиболее общим подходом к построению модели из данных. Эта связь точно описывается Джейнсом [17]:
“Байесовский метод и метод максимальной энтропии отличаются в одном аспекте. Обе процедуры предоставляют оптимальные способы построения модели исходя из информации, которой они обладают, но для Байесовского анализа мы можем выбрать модель; это означает выбор некоторого априорного распределения — или некоторой рабочей гипотезы — о наблюдаемом явлении. Обычно такие гипотезы выходят за рамки того, что непосредственно наблюдается в данных, и, в этом смысле, мы можем сказать, что Байесовские методы являются — или, по крайней мере, могут являться — спекулятивными. В то же время, если наши гипотезы верны, то мы ожидаем, что Байесовские методы дают лучшие результаты, чем метод максимальной энтропии; если же они ложны, результаты Байесовского вывода, вероятно, будут хуже.”

Таким образом, для разработки эффективного алгоритма (дискриминативного или генеративного) нам нужно выбрать модель, которая учитывает знание предметной области. В современном машинном обучении такие модели обычно используют подход глубинного обучения, например, CNN для изображений и LSTM для последовательностей. Использование моделей глубинного обучения в Байесовском выводе привело к возникновению отдельной области — Байесовское глубинное обучение. Центральным методом в Байесовском глубинном обучении является дважды стохастический вариационный вывод [1]. Основной работой в этой области является Вариационный Дропаут [19], который рассматривает CNN как модель правдоподобия в Байесовском выводе и интерпретирует слой дропаута как вариационное приближение апостериорного распределения. После этого, было продемонстрировано, что использование логравномерного распределения в качестве априорного, позволяет добиться разреженности в глубоких нейронных сетях [20]. Однако такую раз-

реженность нельзя использовать для ускорения глубоких нейронных сетях, поскольку она не обладает структурой. В данной работе предлагается модель, которая учитывает архитектуру свёрточных нейронных сетей (CNN) и пораждает структурированную разреженность, тем самым позволяя ускорить CNN. Помимо дропаут слоя, еще одной точкой соприкосновения традиционных алгоритмов глубинного обучения и Байесовским глубинным обучением является батч-нормализация. Рассматривая случайный выбор батча в качестве источника шума, мы предлагаем вероятностную модель батч-нормализации. Затем, используя предложенную модель, мы улучшаем производительность нескольких моделей с точки зрения оценки неопределенности.

Байесовское глубинное обучение в значительной степени опирается на тот факт, что современные модели глубинного обучения эффективно используют предметные знания об области задачи. Другим направлением использования моделей глубинного обучения в Байесовских методах является улучшение приближенного Байесовского вывода, когда точное определение апостериорного распределения невозможно. Одними из инструментов приближенного вывода являются методы Монте-Карло с Марковскими цепями (MCMC), которые можно использовать для описания апостериорного распределения с помощью сэмплирования. Выбор конкретного алгоритма MCMC является существенным для такой задачи. На практике обычно требуется получить алгоритм, который быстро сходится к областям высокой плотности вероятности и быстро перемещается между модами распределения. Чтобы построить такой алгоритм, возможно использовать семейство гибких моделей глубинного обучения для аппроксимации целевого распределения. Одними из первых примеров таких алгоритмов являются NICE-MC [21] и L2HMC [22]. В данной работе мы предлагаем альтернативный подход к обучению алгоритма сэмплирования, выводя функцию потерь для независимого предложного распределения в алгоритме Метрополиса-Гастингса. Помимо этого, мы выходим за рамки традиционной постановки задачи и рассматриваем адаптацию алгоритма Метрополиса-Гастингса на случай неявного предложного распределения и эмпирического целевого распределения. Мы называем эту адаптацию неявным алгоритмом Метрополиса-Гастингса и проводим ее эмпирический и теоретический анализ.

Целью работы является улучшение современных моделей глубинного обучения с использованием Байесовского подхода. Рассматривае-

мыми улучшениями являются улучшение качества и получение новых свойств моделей, таких как разреженность и оценка неопределенности.

Основные результаты и выводы

Новизна работы заключается в том, что впервые показаны следующий пункты.

1. Байесовские вероятностные модели позволяют получить структурированную разреженность в глубоких свёрточных нейронных сетях.
2. Батч-нормализация может быть сформулирована как вероятностная модель, с консистентными этапами обучения и применения.
3. Оптимизация симметричной КЛ-дивергенции позволяет получить эффективные предложные распределения в независимом алгоритме Метрополиса-Гастингса.
4. Используя приближение теста Метрополиса-Гастингса, возможно уменьшить ошибку приближения целевого распределения неявной вероятностной моделью.

Теоретическая и практическая значимость. Полученные результаты расширяют область применения свёрточных нейронных сетей путем сжатия и ускорения традиционных архитектур. Для аналитических целевых распределений (заданных как ненормированная плотность) в работе предлагается метод построения вычислительно эффективного сэмплера. Для неявных генеративных моделей, таких как GAN и VAE, в работе предлагается процедура фильтрации, которая демонстрирует консистентное улучшение на практике. Кроме того, мы выводим верхнюю оценку на расстояние между генерируемым распределением неявной модели и целевым эмпирическим распределением.

Методология и подход. Эта работа использует вероятностный подход в глубинном обучении. Для реализации алгоритмов были использованы следующие инструменты: Python; NumPy, PyTorch, Theano, Lasagne frameworks.

Достоверность полученных результатов достигается детальным описанием методов и алгоритмов, доказательством теорем, а также описанием экспериментов и предоставлением открытого кода, которые способствует воспроизводимости результатов.

Результаты выносимые на защиту:

1. Алгоритм структурированного разреживания свёрточных нейронных сетей.
2. Вероятностная формулировка батч-нормализации, и алгоритм для оценки неопределенности.
3. Адаптация общепринятого алгоритма Метрополиса-Гастингса на случай неявного предложного распределения и эмпирического целевого распределения.

Личный вклад в результаты выносимые на защиту. В двух основных статьях результаты получены лично автором, то есть автором предложены ключевые научные идеи, реализованы и проведены эксперименты, написаны статьи. Вклад других соавторов — обзор кода экспериментов, техническая помощь в дизайне экспериментов, обсуждение полученных результатов, редактирование текстов статей, обсуждение постановок задач и общее курирование исследований.

Публикации и апробация работы

Аспирант является основным автором в двух основных статьях по теме диссертации.

Публикации повышенного уровня.

1. *Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, Dmitry Vetrov* Structured Bayesian Pruning via Log-Normal Multiplicative Noise // Advances in Neural Information Processing Systems 30. 2017. P. 6775–6784. Rank A* conference, indexed by SCOPUS.
2. *Kirill Neklyudov, Evgenii Egorov, Dmitry Vetrov* The Implicit Metropolis-Hastings Algorithm // Advances in Neural Information Processing Systems 32. 2019. Rank A* conference, indexed by SCOPUS.

Другие публикации.

1. *Andrei Atanov, Arsenii Ashukha, Dmitry Molchanov, Kirill Neklyudov, Dmitry Vetrov* Uncertainty Estimation via Stochastic Batch Normalization // In International Symposium on Neural Networks, pp. 261-269. Springer, Cham, 2019.

Доклады на конференциях и семинарах.

1. Seminar of Bayesian methods research group, Москва, 20 мая 2017. Доклад: “Group sparsity in convolutional neural networks”.

2. “Conference on Neural Information Processing Systems 2017”, Лос Анджелес, США, 9 декабря 2016. Доклад: “Structured Bayesian Pruning via Log-Normal Multiplicative Noise”.
3. Seminar of Bayesian methods research group, Москва, 05 октября 2018. Доклад: “Metropolis-Hastings View on Variational Inference and Adversarial Training”.
4. Machine Learning Seminar at Lebedev Physical Institute, 19 февраля 2019. Доклад: “How Neural Networks Help MCMC and How MCMC Helps Neural Networks”.
5. “International Symposium on Neural Networks”, Москва, 10 июля 2019. Доклад: “Uncertainty Estimation via Stochastic Batch Normalization”.
6. “Conference on Neural Information Processing Systems 2019”, Ванкувер, Канада, 11 декабря 2019. Доклад: “The Implicit Metropolis-Hastings Algorithm”.

Объем и содержание работы. Диссертация состоит из введения, содержания публикаций и заключения. Полный объем диссертации 64 страницы.

1 Содержание работы

1.1. Структурированное Байесовское Разреживание

В первой главе описывается модель структурированного Байесовского разреживания, которая способна индуцировать произвольные паттерны структурированной разреженности параметров нейронной сети или её промежуточных представлений. Модель использует стохастический вариационный вывод для настройки своих параметров. Помимо этого, в этой главе описан корректный аналог лог-равномерного априорного распределения, вызывающего разреженность, [20; 23], который позволяет нам сформулировать правильную вероятностную модель и избежать проблем, возникающих при использовании вырожденного априорного распределения. Таким образом, мы можем получить новый Байесовский метод регуляризации нейронных сетей, который приводит к структурированной разреженности. Дополнительно, предложенная модель может быть представлена в виде отдельного слоя, что существенно упрощает ее имплементацию практически без дополнительных вычислительных затрат и позволяет инкорпорировать модель в существующие архитектуры нейронных сетей.

Для заданной вероятностной модели $p(y \mid x,\theta)$, наша цель найти оптимальные значения параметров θ модели используя обучающую выборку $\mathcal{D} = \{(x_i,y_i)\}_{i=1}^N$. Априорные знания о параметрах θ задаются с помощью априорного распределения $p(\theta)$. Используя теорему Байеса мы получаем апостериорное распределение $p(\theta \mid \mathcal{D}) = p(\mathcal{D} \mid \theta)p(\theta)/p(\mathcal{D})$.

Однако вычисление апостериорного распределения с помощью теоремы Байеса обычно требует вычисление интегралов, которые не выражается аналитически. Для преодоление трудностей интегрирования, на практике прибегают к приближенным техникам вывода. Один из самых распространённых приближенных методов это вариационный вывод. В этом подходе неизвестное распределение $p(\theta \mid \mathcal{D})$ приближается другим параметрическим распределением $q_\phi(\theta)$ с помощью минимизации дивергенции Кульбака-Лейблера $\mathrm{KL}(q_\phi(\theta) \,\|\, p(\theta \mid \mathcal{D}))$. Минимизация КЛ-дивергенции эквивалентна максимизации вариационной нижней оценки $\mathcal{L}(\phi)$.

$$\mathcal{L}(\phi) = L_D(\phi) - \mathrm{KL}(q_\phi(\theta) \,\|\, p(\theta)), \tag{1}$$

$$\text{где } L_D(\phi) = \sum_{i=1}^{N} \mathbb{E}_{q_\phi(\theta)} \log p(y_i \mid x_i,\theta) \tag{2}$$

Предложенная модель может быть сформулирована как отдельный дропаут слой с входным вектором $x \in \mathbb{R}^I$, который является I-мерным признаковым описанием одного объекта, и выходным вектором $y \in \mathbb{R}^I$ того же размера. Под входным вектором x обычно подразумеваются активации предыдущего слоя. В таком случае выходной вектор y будет использоваться в качестве входного вектора следующего слоя. Мы следуем общему способу построения дропаут слоёв (3). Каждый входной признак x_i умножается на случайную величину θ_i имеющую плотность $p_{noise}(\theta)$. Например, для бинарного дропаута $p_{noise}(\theta)$ является полностью факторизованным распределением Бернулли $p_{noise}(\theta_i) = \mathrm{Bernoulli(p)}$, а для Гауссовского дропаута распределение является полностью факторизованным Гауссовским распределением $p_{noise}(\theta_i) = \mathcal{N}(1,\alpha)$.

$$y_i = x_i \cdot \theta_i \qquad\qquad \theta \sim p_{noise}(\theta) \tag{3}$$

Далее мы следуем Байесовскому подходу по отношению к переменной θ. Чтобы получить разреженное решение, в качестве априорного распределения $p(\theta)$ можно выбрать полностью факторизованное транкированное лог-равномерное распределение. Для аппроксимационного семей-

Таблица 1: Сравнение разных техник разреживания (SparseVD [20]) для VGG-подобных архитектур на CIFAR-10. StructuredBP соответствует оригинальной модели структурного Байесовского разреживания, а StructuredBPa соответствует той же модели со шкалирование КЛ-дивергенции. k — мультипликатор ширины слоя задающий число фильтров и нейронов в каждом слое сети (width$(k) = k \times$ original width)

k	Метод	Ошибка %	Элементов в слое	CPU	GPU	FLOPs
1.0	Original	7.2	64 – 64 – 128 – 128 – 256 – 256 – 256 – 512 – 512 – 512 – 512 – 512 – 512 – 512	1.00×	1.00×	1.00×
	SparseVD	7.2	64 – 62 – 128 – 126 – 234 – 155 – 31 – 81 – 76 – 9 – 138 – 101 – 413 – 373	2.50×	1.69×	2.27×
(ours)	StructuredBP	7.5	64 – 62 – 128 – 126 – 234 – 155 – 31 – 79 – 73 – 9 – 59 – 73 – 56 – 27	2.71×	1.74×	2.30×
(ours)	StructuredBPa	9.0	44 – 54 – 92 – 115 – 234 – 155 – 31 – 76 – 55 – 9 – 34 – 35 – 21 – 280	3.68×	2.06×	3.16×
1.5	Original	6.8	96 – 96 – 192 – 192 – 384 – 384 – 384 – 768 – 768 – 768 – 768 – 768 – 768 – 768	1.00×	1.00×	1.00×
	SparseVD	7.0	96 – 78 – 191 – 146 – 254 – 126 – 27 – 79 – 74 – 9 – 137 – 100 – 416 – 479	3.35×	2.16×	3.27×
(ours)	StructuredBP	7.2	96 – 77 – 190 – 146 – 254 – 126 – 26 – 79 – 70 – 9 – 71 – 82 – 79 – 49	3.63×	2.17×	3.32×
(ours)	StructuredBPa	7.8	77 – 74 – 161 – 146 – 254 – 125 – 26 – 78 – 66 – 9 – 47 – 55 – 54 – 237	4.47×	2.47×	3.93×

ства мы выбираем транкированное лог-нормальное распределение. Тогда модель слоя может быть сформулирована следующим образом.

$$y_i = x_i \cdot \theta_i \quad p(\theta_i) = \text{LogU}_{[a,b]}(\theta_i) \quad q(\theta_i \,|\, \mu_i, \sigma_i) = \text{LogN}_{[a,b]}(\theta_i \,|\, \mu_i, \sigma_i^2) \tag{4}$$

Эксперименты показывают, что предложенная модель приводит к высокому уровню групповой разреженности и значительному ускорению сверточных нейронных сетей с незначительным падением точности. Мы демонстрируем производительность нашего метода на архитектурах LeNet и VGG, используя наборы данных MNIST и CIFAR-10. Результаты для VGG-подобных архитектур на наборе данных CIFAR-10 представлены в Таблице 1. Для каждой архитектуры мы приводим количество сохраненных нейронов и фильтров, и полученное ускорение. Мы также демонстрируем, что оптимизация в пространстве вариационных параметров (μ, σ) приводит к улучшению качества модели и позволяет более эффективно выполнять разреживание по сравнению с оптимизацией только дисперсии шума. В качестве дополнительного свойства, мы показываем, что сеть структурированное Байесовское разреживание не переобучается на случайно размеченных данных, что является недостатком при обучении общепринятых архитектур.

1.2. Стохастическая Батч-нормализация

Во второй главе мы исследуем технику батч-нормализации, и предлагаем ее вероятностную интерпретацию. Слой батч-нормализации является неотъемлемой частью любой глубокой сверточной архитектуры. В нашей работе мы рассматриваем батч-нормализацию как стохастический слой и предлагаем способ ансамблирования сетей использующих её. Однако, наивный способ требует больших затрат на вычисление и память.

Поэтому мы предлегаем аналог — стохастическую батч-нормализацию (SBN) — эффективную и масштабируемую приближенную технику.

Мы рассматриваем задачу обучения с учителем на обучающей выборке $D = \{(x_i, y_i)\}_{i=1}^N$. Цель обучения заключается в нахождении параметров θ предсказательного распределения $p_\theta(y \mid x)$ моделируемого нейронной сетью. Для решения этой задачи обычно используется стохастический градиентный спуск, например, [24].

Батч-нормализация пытается сохранить нулевое мат. ожидание и единичную дисперсию активаций всех слоев нейронной сети. Чтобы добиться этого слой батч-нормализации вычисляет выборочное среднее $\mu(\mathcal{B})$ и дисперсию $\sigma^2(\mathcal{B})$ по одному батчу $\mathcal{B}$ во время обучения и использует аккумулированные статистики во время предсказания:

$$\mathrm{BN}^{\text{train}}_{\gamma,\beta}(x_i) = \frac{x_i - \mu(\mathcal{B})}{\sqrt{\sigma^2(\mathcal{B}) + \epsilon}} \cdot \gamma + \beta \qquad \mathrm{BN}^{\text{test}}_{\gamma,\beta}(x_i) = \frac{x_i - \hat{\mu}}{\sqrt{\hat{\sigma}^2 + \epsilon}} \cdot \gamma + \beta \quad (5)$$

где γ, β — обучаемые параметры батч-нормализации (масштаб и сдвиг) и ϵ константа использующаяся для численной стабильности. Во время обучения среднее и дисперсия вычисляются по случайно выбранному батчу ($\mu(\mathcal{B})$, $\sigma(\mathcal{B})$), в то время как во время применения используется экспоненциальное сглаживание статистик ($\hat{\mu}$, $\hat{\sigma}^2$). Далее мы избавляемся от этой неконсистентности в предложенной вероятностной модели.

Из уравнения (5) видно что при вычислении батч-нормализации для x_i результат зависит от всего батча $\mathcal{B}$. Эта зависимость может быть переформулирована в терминах статистик батча $\mu(\mathcal{B}), \sigma(\mathcal{B})$:

$$p_\theta(y_i \mid x_i, \mathcal{B}_{\setminus i}) = p_\theta(y_i | x_i, \mu(\mathcal{B}), \sigma(\mathcal{B})), \quad (6)$$

где $\mathcal{B}_{\setminus i}$ — батч без x_i. Из-за случайности выбора батчей во время обучения, для конкретного x_i, $\mathcal{B}_{\setminus i}$ является случайной величиной, тогда статистики батча могут быть интерпретированы как случайные величины. Условное распределение $p_\theta(\mu, \sigma \mid x_i, \mathcal{B}_{\setminus i})$ является произведением двух дельта-функций Дирака в точках $\mu(\mathcal{B})$ и $\sigma(\mathcal{B})$, т.к. статистики являются детерминированными функциями батча, и распределение среднего и дисперсии при заданном x_i это мат. ожидание по распределению батчей. Во время предсказания мы усредняем распределение $p_\theta(y|x, \mu, \sigma^2)$ по статистикам:

$$p_\theta(\mu, \sigma | x_i) = \mathop{\mathbb{E}}_{\mathcal{B}_{\setminus i}} \delta_{\mu(\mathcal{B})}(\mu)\delta_{\sigma(\mathcal{B})}(\sigma) \qquad p_\theta(y|x) = \mathop{\mathbb{E}}_{p_\theta(\mu,\sigma|x)} p(y|x, \mu, \sigma) \quad (7)$$

В работе, мы демонстрируем что во время обучения батч-нормализация (5) выполняет несмещенную одноточечную Монте-Карло оценку градиента на нижнюю оценку маргинального правдоподобия (7). Таким образом, такая вероятностная модель соответствует батч-нормализации во время обучеия. Однако, на этапе предсказания батч-нормализация обычно использует экспоненциальное сглаживание статистик $\mathbb{E}\,\mu \approx \hat{\mu}, \mathbb{E}\,\sigma \approx \hat{\sigma}$, которое может рассматриваться как смещенная оценка (7):

$$\mathop{\mathbb{E}}_{p_\theta(\mu,\sigma|x_i)} p(y_i|x_i,\mu,\sigma) \approx p_\theta(y|x,\mathbb{E}\,\mu,\mathbb{E}\,\sigma)$$

Прямая Монте-Карло оценка может быть использована для несмещенной оценки (7), однако, она имеет ограничения на практике. Действительно, для получения одного сэмпла распределения статистик (7) нам необходимо сделать прямой проход для всего батча. Таким образом, для получения Монте-Карло оценки для одного тестового объекта, нам необходимо сделать несколько прямых проходов с разными батчами из обучающей выборки. Для устранения этого недостатка мы предлагаем стохастическую батч-нормализацию.

Чтобы снизить затраты на вычисления и память Монте-Карло оценки, мы предлагаем приблизить распределение статистик батч-нормализации $p_\theta(\mu, \sigma \,|\, x_i)$ с помощью полностью факторизованного параметрического семейства $p_\theta(\mu, \sigma \,|\, x_i) \approx r(\mu)r(\sigma)$. Мы параметризуем $r(\mu)$ и $r(\sigma)$ следующим образом:

$$r(\mu) = \mathcal{N}(\mu|\mathrm{m}_\mu, \mathrm{s}^2_\mu) \qquad\qquad r(\sigma) = \mathrm{Log}\mathcal{N}(\sigma|\mathrm{m}_\sigma, \mathrm{s}^2_\sigma) \tag{8}$$

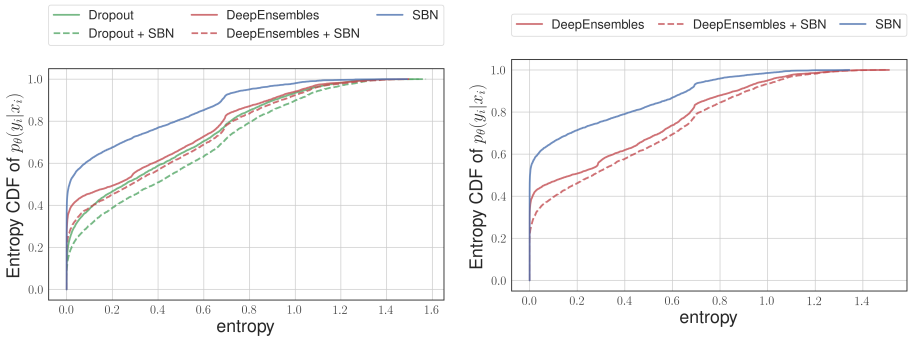
Данная аппроксимация хорошо работает на практике, в частности мы демонстрируем, что она позволяет точно приблизить маргинальные распределения. Поскольку аппроксимация больше не зависит от обучающих данных, сэмплы для каждого слоя могут быть получены без выполнения прямого прохода для всего батча, что позволяет эффективно вычислять предсказания. На практике, чтобы сделать прогноз для одного тестового объекта, можно продублировать объект K раз и пропустить весь батч через сеть только один раз, независимо сэмплируя статистики для каждой копии и усредняя результаты для такого “виртуального батча”. Эта процедура в K раз быстрее по сравнению с прямой оценкой Монте-Карло.

Чтобы найти параметры $\{m_\mu, s_\mu, m_\sigma, s_\sigma\}$ мы минимизируем КЛ-дивергенцию между распределением порождаемым батч-нормализацией

(7) и нашим приближением $r(\mu)r(\sigma)$ для каждого объекта:

$$D_{\mathrm{KL}}\left({}^{1}\!/_{N}\sum_{i=1}^{N} p_\theta(\mu,\sigma \mid x_i) \,\middle|\middle|\, r(\mu)r(\sigma)\right) \longrightarrow \min_{m_\mu, s_\mu, m_\sigma, s_\sigma}$$

Поскольку r принадлежит экспоненциальному семейству, эта задача минимизации эквивалетна приравниванию моментов распределений и не требует вычисления градиентов. В нашей реализации мы используем экспоненциальное сглаживание для аппроксимации достаточных статистик среднего и дисперсии распределения. Это можно сделать для любой предварительно обученной сети с батч-нормализацией.



(а) Результаты для VGG-11

(б) Результаты для ResNet-18

Рис. 1: Эмпирическая функция распределения энтропии на внедоменных данных. VGG-11 и ResNet-18 на пяти классах из CIFAR-10, не использовавшихся во время обучения. SBN соответствует модели со стохастической батч-нормализацией. Чем правее и ниже идут графики, тем лучше оценка неопределенности.

Чтобы показать что наш метод масштабируется на глубокие сверточные архитектуры, мы приводим результаты экспериментом на VGG-подобных и ResNet архитектурах. Мы разбиваем CIFAR-10 на два набора данных (CIFAR-5), и изображаем график эмпирической функции распределения энтропии на Рис. 1. Мы обучали модели на случайно выбранных 5 классах и оценивали уровень неопределенности на оставшихся.

1.3. Выбор Предложного Распределения для Метрополиса-Гастингса

Значительную часть методов MCMC можно рассматривать как алгоритм Метрополиса-Гастингса (MH) с различными предложными распределениями. С этой точки зрения, проблема построения сэмплера мо-

жет быть сведена к вопросу — как выбрать предложное распределение MH алгоритма? Пытаясь ответить на этот вопрос, мы предлагаем обучать независимое предложное распределение, которое максимизирует вероятность принятия в MH алгоритме. Как мы демонстрируем, данный подход тесно связан с вариационным выводом. Для Байесовского вывода предложенный метод выгодно отличается от альтернатив на задаче сэмплирования из апостериорного распределения. В рамках того же подхода мы выходим за рамки классических методов MCMC и выводим с нуля подход генеративных соревновательных моделей (GAN), рассматривая генератор как предложное распределение и дискриминатор как тест Метрополиса-Гастингса.

Вероятность принятия в MH алгоритме тесно связана с детальным балансом. В крайнем случае, когда вероятность принятия достигает максимального значения, распределения $p(x')q(x \mid x')$ и $p(x)q(x' \mid x)$ должны совпадать (с точностью до множества меры нуль) в совместном пространстве предыдущей точки x и предложенной точки x'. В таком случае, мы можем сказать что условие детального баланса выполнено:

$$p(x')q(x \mid x') = p(x)q(x' \mid x) \quad \forall x,x'. \tag{9}$$

Оказывается, что вероятность принятия определяет расстояние между распределениями $p(x')q(x \mid x')$ и $p(x)q(x' \mid x)$, или насколько хорошо выполнено условие детального баланса для предложного распределения $q(\cdot \mid \cdot)$. Мы формализуем эту связь в следующей теореме.

***Теорема* 1.** *Для случайной величины* $\xi = \frac{p(x')q(x \mid x')}{p(x)q(x' \mid x)}, x \sim p(x), x' \sim q(x' \mid x)$

$$\begin{aligned} \text{AR} &= \mathbb{E}_\xi \min\{1, \xi\} = 1 - \frac{1}{2}\mathbb{E}_\xi |\xi - 1| = \\ &= 1 - \text{TV}\Big(p(x')q(x \mid x') \Big\| p(x)q(x' \mid x)\Big), \end{aligned} \tag{10}$$

где TV — *расстояние полной вариации.*

Переформулировка в терминах полной вариации позволяет нам получить нижнюю оценку на вероятность принятия с помощью неравенства Пинскера.

$$\text{AR} \geq 1 - \sqrt{\frac{1}{2} \cdot \text{KL}\Big(p(x')q(x \mid x') \Big\| p(x)q(x' \mid x)\Big)}. \tag{11}$$

Мы предлагаем использовать вероятность принятия или его нижнюю границу в качестве целевой функции для обучения предложного распределения. Однако, использование этого для Марковского предложного распределения может привести к вырожденному решению решению $q(x' \mid x) = \delta(x' - x)$, которое дает максимальную вероятность принятия. Это происходит из-за того, что условие детального баланса и вероятность принятия не учитывают автокорреляцию цепочки. В этой работе мы добиваемся нулевой автокорреляции предложенных сэмплов и исключаем вырожденное решение, рассматривая независимые предложные распределения.

Для независимых предложных распределений, нижняя оценка из (11) может быть переписана как симметричная КЛ-дивергенция между $p(\cdot)$ и $q(\cdot)$

$$\text{AR} \geq 1 - \sqrt{\frac{1}{2}\Big(\text{KL}(q(x)\|p(x)) + \text{KL}(p(x)\|q(x))\Big)}, \tag{12}$$

глобальный максимум которой достигается при $q(x) = p(x)$, и, следовательно, при максимальной вероятности принятия $\text{AR} = 1$. В работе мы демонстрируем что полученная нижняя оценка проводит связь предложенного подхода с вариационным выводом и генеративными соревновательными моделями. В контексте Байесовского вывода, полученная нижняя оценка может быть предпочтительной оптимизации вероятность принятия, т.к. может быть оценена только по батчу данных.

В этой главе, мы предлагаем алгоритм для обучения параметров ϕ независимого предложного распределения $q_\phi(x)$. В качестве целевых функций для оптимизации мы используем вероятность принятия MH алгоритма и её нижнюю оценку. Для удобства, мы переписываем эти функции как функции потерь следующим образом. Максимизация вероятности принятия эквивалентна минимизации потерь:

$$\mathcal{L}_{\text{AR}}(\phi) = -\text{AR} = -\mathbb{E}_{\substack{x \sim p(x) \\ x' \sim q_\phi(x')}} \min\left\{1, \frac{p(x')q_\phi(x)}{q_\phi(x')p(x)}\right\}. \tag{13}$$

Для максимизации нижней оценки, функция потерь может быть записана как

$$\mathcal{L}_{\text{KL}}(\phi) = \text{KL}(q_\phi(x)\|p(x)) + \text{KL}(p(x)\|q_\phi(x)) = \tag{14}$$

$$= -\mathbb{E}_{\substack{x \sim p(x) \\ x' \sim q_\phi(x')}} \log\left(\frac{p(x')q_\phi(x)}{q_\phi(x')p(x)}\right). \tag{15}$$

Таблица 2: Краткое описание двух постановок для целевого $p(x)$ и предложного $q(x)$ распределений.

Постановка	**Плотность** $p(x)$	**Сэмплы из** $p(x)$	**Плотность** $q(x)$	**Отношение плотностей**
Постановка с плотностью	дана	независимый МН	дана	дано
Эмпирическая постановка	недоступна	даны	недоступна	соревновательное обучение

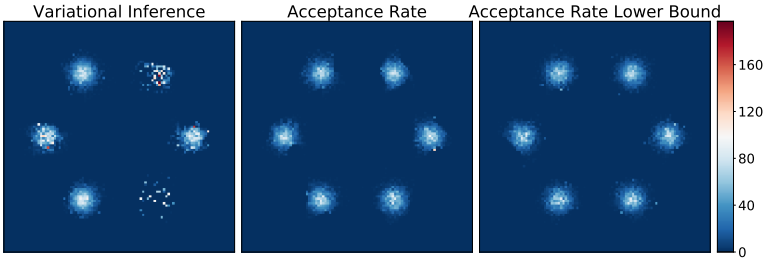


Рис. 2: Двумерные гистограммы $25 \cdot 10^3$ сэмплов из МН алгоритма с разными предложными распределениями. Слева направо предложные распределения обучены с помощью вариационного вывода, максимизации вероятности принятия, максимизации нижней оценки на вероятность принятия.

Для оценки $\mathcal{L}(\phi)$, необходимо вычислить отношение плотностей на сэмплах из целевого распределения $x \sim p(x)$ и предложного $x' \sim q_\phi(x')$. В зависимости от формы в которой задано целевое распределение, у нас возникают разные проблемы при оценке функции потерь.

Если целевое распределение задано как ненормированная плотность (мы называем это *постановка с плотностью*), мы предлагаем использовать явные вероятностные модели, в качестве предложного распределения, чтобы вычислять отношение плотностей точно. Для получения сэмплов из целевого распределения, в этой постановке мы предлагаем использовать независимый МН алгоритм с текущим предложным распределением.

Если целевое распределение задано эмпирически (мы называем это *эмпирическая постановка*), сэмплы из целевого и предложного распределений доступны, однако мы не можем вычислить отношение плотностей, для этого мы предлагаем приблизить это отношение с помощью соревновательного обучения.

Краткого описание обоих постановок приведено в Таблице 2.

Мы демонстрируем эмпирические результаты для обеих постановок. В постановке с плотностью, предложенный алгоритм оказывается предпочтительнее в задаче сэмплирования из апостериорного распреде-

ления Байесовской логистической регрессии. В эмпирической постановке, мы демонстрируем улучшения различных GAN моделей с помощью применения MH алгоритма на этапе выполнения.

В качестве иллюстрации, мы демонстрируем двумерные гистограммы сэмплов из смеси шести Гауссиан для разных методов (Рис. 2).

1.4. Неявный Алгоритм Метрополиса-Гастингса

В предыдущей главе мы предлагаем использовать дискриминатор GAN для фильтрации нереалистичных сэмплов генератора. Здесь мы обобщаем и обосновываем эту идею, вводя неявный алгоритм Метрополиса-Гастингса. Для любой неявной вероятностной модели и целевого распределения, представленного эмпирически, неявный алгоритм Метрополиса-Гастингса обучает дискриминатор для оценки отношения плотности, а затем генерирует цепочку сэмплов. Поскольку аппроксимация отношения плотностей вносит ошибку на каждом шаге цепочки, крайне важно проанализировать стационарное распределение такой цепочки. С этой целью мы приводим теоретический результат, утверждающий, что функция потерь дискриминатора является верхней оценкой на расстояние между целевым распределением и стационарным распределением цепочки.

Algorithm 1 Неявный алгоритм Метрополиса-Гастингса

input целевой набор данных $\mathcal{D}$
input неявная модель $q(x \mid y)$
input обученный дискриминатор $d(\cdot,\cdot)$
  $y \sim \mathcal{D}$ инициализация из данных
  for $i = 0 \dots n$ **do**
    сгенерировать предложную точку $x \sim q(x \mid y)$
    $P = \min\{1, \frac{d(x,y)}{d(y,x)}\}$
    $x_i \begin{cases} x, \text{ с вероятностью } P \\ y, \text{ с вероятностью } (1-P) \end{cases}$
    $y \leftarrow x_i$
  end for
output $\{x_0, \dots, x_n\}$

Целью неявного алгоритма Метрополиса-Гастингса является сэмплирование из эмпирического целевого распределения $p(x)$, $x \in \mathbb{R}^D$, в то время как доступны сэмплы только из неявного предложного распределе-

ния $q(x \mid y)$. При заданном дискриминаторе $d(x,y)$, алгоритм генерирует цепочку сэмплов как описано в Алгоритме 1.

Мы строим наши рассуждения, сначала предполагая, что цепь генерируется с использованием некоторого дискриминатора, а затем последовательно вводим условия на дискриминатор и ограничиваем сверху расстояние между цепочкой и целевым распределением. Наконец, мы выводим верхнюю границу, которую можно минимизировать относительно параметров дискриминатора. Мы рассматриваем случай неявного Марковского предложного распределения, но все выводы справедливы и для независимых предложных распределений.

Ядро перехода в неявном алгоритме Метрополиса-Гастингса это

$$t(x \mid y) = q(x \mid y) \min\left\{1, \frac{d(x,y)}{d(y,x)}\right\} + \tag{16}$$

$$+ \delta(x-y) \int dx' q(x' \mid y) \left(1 - \min\left\{1, \frac{d(x',y)}{d(y,x')}\right\}\right). \tag{17}$$

Далее мы хотим чтобы стационарное распределение t_∞ нашей Марковской цепочки было как можно ближе к целевому распределению p. В качестве меры близости распределений мы рассматриваем стандартную метрику для анализа в МСМС — *расстояние полной вариации*

$$\|t_\infty - p\|_{TV} = \frac{1}{2} \int |t_\infty(x) - p(x)| dx. \tag{18}$$

Мы предполагаем, что предложное распределение $q(x \mid y)$ задано, но различные $d(x,y)$ приводят к разным t_∞. Поэтому мы хотим вывести верхнюю оценку на расстояние $\|t_\infty - p\|_{TV}$ и минимизировать её относительно параметров дискриминатора $d(x,y)$. Мы выводим эту верхнюю оценку в три этапа.

Когда переходное ядро удовлетворяет условию миноризации, Марковская цепочка сходится "быстро" к стационарному распределению. Мы формализуем это утверждение в следующем Предложении.

***Предложение* 1.** *Рассмотрим переходное ядро $t(x \mid y)$ которое удовлетворяет условию миноризации $t(x \mid y) > \varepsilon \nu(x)$ для некоторого $\varepsilon > 0$, и распределения ν. Тогда расстояние между двумя последовательными шагами уменьшается как:*

$$\|t_{n+2} - t_{n+1}\|_{TV} \leq (1-\varepsilon) \|t_{n+1} - t_n\|_{TV}, \tag{19}$$

где распределение $t_{k+1}(x) = \int t(x \mid y) t_k(y) dy$.

Чтобы гарантировать условие миноризации для нашего переходного ядра $t(x \mid y)$, мы требуем чтобы предложное распределение $q(x \mid y)$ удовлетворяло условию миноризации с некоторой константой ε и распределением ν (заметьте, что для независимых распределений условие миноризации выполняется автоматически с $\varepsilon = 1$). Также мы ограничиваем область определения дискриминатора как $d(x,y) \in [b,1] \ \forall x,y$, где b это некоторая положительная константа, которая является гиперпараметром алгоритма. Из этого требования следует

$$t(x \mid y) \geq bq(x \mid y) > b\varepsilon\nu(x). \tag{20}$$

Выбирая целевое распределение $p(x)$ в качестве начального распределения $t_0(x)$ нашей цепочки $t(x \mid y)$, мы сводим задачу оценки расстояния $\|t_\infty - p\|_{TV}$ к задаче оценке расстояния $\|t_1 - p\|_{TV}$:

$$\|t_\infty - p\|_{TV} \leq \frac{1}{b\varepsilon} \|t_1 - p\|_{TV}. \tag{21}$$

Однако, наивная оценка $t_1(x)$ приводит к смещенной оценки $\|t_1 - p\|_{TV}$, так как мат. ожидание находится внутри нелинейной функции. Для преодоления этой проблемы мы оцениваем сверху это расстояние в следующем предложении.

***Предложение* 2.** *Для ядра $t(x \mid y)$ неявного алгоритма Метрополиса-Гастингса, расстояние между начальным распределением $p(x)$ и распределением $t_1(x)$ имеет следующую верхнюю оценку*

$$\|t_1 - p\|_{TV} \leq 2 \left\| q(y \mid x)p(x) - q(x \mid y)p(y)\frac{d(x,y)}{d(y,x)} \right\|_{TV}, \tag{22}$$

где TV-расстояние справа вычисляется в совместном пространстве $(x,y) \in \mathbb{R}^D \times \mathbb{R}^D$.

Чтобы сверху ограничить TV-расстояние $\|\alpha - \beta\|_{TV}$ с помощью КЛ-дивергенции $\mathrm{KL}(\alpha\|\beta)$ можно использовать неравенство Пинскера:

$$2\|\alpha - \beta\|_{TV}^2 \leq \mathrm{KL}(\alpha\|\beta). \tag{23}$$

Однако, в неравенстве Пинскера подразумевается что α и β являются распределениями, в то время как это не всегда верно для функций $q(x \mid y)p(y)\frac{d(x,y)}{d(y,x)}$ в (22). В следующем предложении мы расширяем неравенство Пинскера на случай когда одна из функций не нормированна.

Предложение 3. *Для распределения $\alpha(x)$ и некоторой положительной функции $f(x) > 0\ \forall x$ следующее неравенство имеет место:*

$$\|\alpha - f\|_{TV}^2 \leq \left(\frac{2C_f + 1}{6}\right)(\widehat{\mathrm{KL}}(\alpha\|f) + C_f - 1), \tag{24}$$

где C_f — нормировочная константа функции f: $C_f = \int f(x)dx$, и $\widehat{\mathrm{KL}}(\alpha\|f)$ формальное вычисление КЛ-дивергенции

$$\widehat{\mathrm{KL}}(\alpha\|f) = \int \alpha(x) \log \frac{\alpha(x)}{f(x)} dx. \tag{25}$$

Объединяя все результаты, мы получаем следующую верхнюю оценку.

$$\|t_\infty - p\|_{TV}^2 \leq \frac{1}{b^2\varepsilon^2} \|t_1 - p\|_{TV}^2 \leq \tag{26}$$

$$\leq \frac{4}{b^2\varepsilon^2} \left\| q(y \mid x)p(x) - q(x \mid y)p(y)\frac{d(x,y)}{d(y,x)} \right\|_{TV}^2 \leq \tag{27}$$

$$\leq \left(\frac{4 + 2b}{3\varepsilon^2 b^3}\right) \Bigg(\underbrace{\mathbb{E}_{\substack{x \sim p(x) \\ y \sim q(y \mid x)}} \left[\log \frac{d(y,x)}{d(x,y)} + \frac{d(y,x)}{d(x,y)} \right]}_{\text{потери дискриминатора}} - \tag{28}$$

$$-1 + \mathrm{KL}\left(q(y \mid x)p(x) \middle\| q(x \mid y)p(y) \right) \Bigg)$$

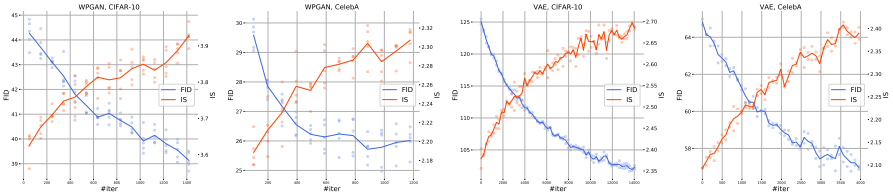


Рис. 3: Монотонные улучшение в терминах FID и IS для обучения дискриминатора с помощью кросс-энтропии. Во время итераций, мы вычисляем метрики 5 раз (точки), а затем усредняем их (сплошные линии). Для одного вычисления метрики мы используем 10^4 сэмплов. Высокие значения IS и низкие значения FID свидетельствуют о более высоком качестве. Качество оригинального генератора соответствует нулевой итерации на каждом графике.

Мы приводим эмпирические результаты предложенного алгоритма и проверку теории как для независимых, так и для Марковских предложных распределений. В обоих случаях сэмплирование с помощью неявного алгоритма МН лучше, чем сэмплирование напрямую из генератора.

Для независимых предложных распределений, мы проверяем наш теоретический результат, демонстрируя монотонные улучшения качества на протяжении всего обучения дискриминатора (Рис. 3). Более того, использование Марковского предложного распределения в неявном алгоритме Метрополиса-Гастингса выгодно отличается от использования независимых предложных распределений.

Заключение

Основные результаты работы могут быть резюмированы следующим образом

1. Предложена модель Структурированного Байесовского Разреживания (SBP) для получения структурированной разреженности свёрточных нейронных сетей. Модель может быть сформулирована как дропаут слой индуцирующий мультипликативный шум на выходы предыдущего слоя. Основой модели является априорное распределение индуцирующее разреженность и настройка параметров распределения с помощью стохастического вариационного вывода. SBP слой может использовать произвольные паттерны для разреживания его входов и адаптивной регуляризации. Мы применяем SBP для того чтобы снизить число используемых нейронов и фильтров в свёрточной нейронной сети и демонстрируем значительное практическое ускорение без внесения модификаций в имплементацию нейросетей.
2. Предложена вероятностная формулировка батч-нормализации и алгоритм оценки неопределенности. Мы рассматриваем вероятностный подход и разрабатываем новый алгоритм, который ведет себя консистентно на этапах обучения и тестирования. Мы сравниваем эффективность предложенного алгоритма с аналогичными методами для задач классификации изображений и оценки неопределенности.
3. Подходя к проблеме сэмплирования с точки зрения алгоритма Метрополиса-Гастингса мы предлагаем эффективные решения как для эмпирических, так и для аналитических распределений. В этом подходе мы демонстрируем, что естественной функцией потерь для оптимизации независимого предложного распределения является симметричная КЛ-дивергенция. По сравнению с вариационным выводом эта процедура учитывает прямую КЛ-дивергенцию, тем самым способствуя покрытию различных

мод. Для эмпирических распределений, например, набора изображений, оптимизация симметричной КЛ эквивалентна обычному обучению генеративных соревновательных сетей (GAN). Тем не менее, наш подход позволяет нам аппроксимировать алгоритм Метрополиса-Гастингса и получить стабильные улучшения по сравнению с GAN.

4. Мы предлагаем неявный алгоритм Метрополиса-Гастингса для сэмплирования из эмпирического целевого распределения, используя неявную вероятностную модель в качестве предложного распределения. В теоретической части статьи мы выводим верхнюю оценку расстояния между целевым распределением и стационарным распределением Марковской цепочки. Тем самым, мы обосновываем предложенную ранее эвристическую процедуру и выводим функцию потерь для случая Марковского предложного распределения. Более того, постобработка в неявном алгоритме Метрополиса-Гастингса может рассматриваться как усовершенствование любой неявной модели. В экспериментальной части статьи мы эмпирически проверяем предложенный алгоритм на реальных наборах данных (CIFAR-10 и CelebA). Для обеих задач, фильтрация с помощью предложенного алгоритма уменьшает ошибку приближения целевого распределения неявной моделью.

Список литературы

1. *Titsias M.*, *Lázaro-Gredilla M.* Doubly stochastic variational Bayes for non-conjugate inference // International Conference on Machine Learning. — 2014. — С. 1971—1979.
2. *Goodfellow I.*, *Pouget-Abadie J.*, *Mirza M.*, *Xu B.*, *Warde-Farley D.*, *Ozair S.*, *Courville A.*, *Bengio Y.* Generative adversarial nets // Advances in neural information processing systems. — 2014. — С. 2672—2680.
3. *Hastings W. K.* Monte Carlo sampling methods using Markov chains and their applications. — 1970.
4. *Bishop C. M.* Pattern recognition and machine learning. — springer, 2006.
5. *MacKay D. J.* Information theory, inference and learning algorithms. — Cambridge university press, 2003.
6. *Hinton G. E.* Training products of experts by minimizing contrastive divergence // Neural computation. — 2002. — Т. 14, № 8. — С. 1771—1800.
7. *Hinton G. E.*, *Dayan P.*, *Frey B. J.*, *Neal R. M.* The”wake-sleep”algorithm for unsupervised neural networks // Science. — 1995. — Т. 268, № 5214. — С. 1158—1161.
8. *Kingma D. P.*, *Welling M.* Auto-encoding variational bayes // ICLR. — 2014.
9. *Rezende D. J.*, *Mohamed S.* Variational inference with normalizing flows // arXiv preprint arXiv:1505.05770. — 2015.
10. *Krizhevsky A.*, *Sutskever I.*, *Hinton G. E.* Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. — 2012. — С. 1097—1105.
11. *Lowe D. G.* [и др.]. Object recognition from local scale-invariant features. // iccv. Т. 99. — 1999. — С. 1150—1157.
12. *Dalal N.*, *Triggs B.* Histograms of oriented gradients for human detection //. — 2005.
13. *LeCun Y.*, *Bengio Y.*, *Hinton G.* Deep learning // nature. — 2015. — Т. 521, № 7553. — С. 436.
14. *Goodfellow I.*, *Bengio Y.*, *Courville A.*, *Bengio Y.* Deep learning. Т. 1. — MIT Press, 2016.
15. *LeCun Y.*, *Boser B.*, *Denker J. S.*, *Henderson D.*, *Howard R. E.*, *Hubbard W.*, *Jackel L. D.* Backpropagation applied to handwritten zip code recognition // Neural computation. — 1989. — Т. 1, № 4. — С. 541—551.
16. *Hochreiter S.*, *Schmidhuber J.* Long short-term memory // Neural computation. — 1997. — Т. 9, № 8. — С. 1735—1780.

17. *Jaynes E. T.* Probability theory: The logic of science. — Cambridge university press, 2003.

18. *Jaynes E. T.* On the rationale of maximum-entropy methods // Proceedings of the IEEE. — 1982. — Т. 70, № 9. — С. 939—952.

19. *Kingma D. P.*, *Salimans T.*, *Welling M.* Variational dropout and the local reparameterization trick // Advances in Neural Information Processing Systems. — 2015. — С. 2575—2583.

20. *Molchanov D.*, *Ashukha A.*, *Vetrov D.* Variational dropout sparsifies deep neural networks // arXiv preprint arXiv:1701.05369. — 2017.

21. *Song J.*, *Zhao S.*, *Ermon S.* A-nice-mc: Adversarial training for mcmc // Advances in Neural Information Processing Systems. — 2017. — С. 5140—5150.

22. *Levy D.*, *Hoffman M. D.*, *Sohl-Dickstein J.* Generalizing Hamiltonian Monte Carlo with Neural Networks // arXiv preprint arXiv:1711.09268. — 2017.

23. *Kingma D. P.*, *Salimans T.*, *Welling M.* Variational Dropout and the Local Reparameterization Trick // Advances in Neural Information Processing Systems 28 / под ред. C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett. — Curran Associates, Inc., 2015. — С. 2575—2583.

24. *Kingma D. P.*, *Ba J.* Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980. — 2014.