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

На правах рукописи

Ашуха Арсений Павлович

АПРИОРНЫЕ ЗНАНИЯ ДЛЯ МОДЕЛЕЙ ГЛУБИННОГО ОБУЧЕНИЯ

РЕЗЮМЕ
диссертации на соискание
ученой степени кандидата
компьютерных наук

Москва — 2022

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., Национальный исследовательский университет «Высшая школа экономики».

1. Тема диссертации

Эта работа посвящена способам внедрения априорных знаний в модели глубокого обучения." Мы предлагаем *разреживающий вариационный дропаут (sparse variational dropout, Sparse VD)* — метод, который позволяет получить высокий уровень разреженности в глубоких нейронных сетях на основе разреживающего априорного распределения. Кроме того, было разработано т.н. *глубокое априорное распределение (deep weight prior, DWP)* — метод, который позволяет использовать априорные распределения над весами глубокой свёрточной нейронной сети, основанные на генеративных моделях. DWP может задавать более гибкие априорные распределения, которые могут учитывать зависимости между весами и охватывать несколько мод.

Далее мы рассматриваем ансамблирование с использованием *аугментации данных (data augmentation)*, которое также называют *тестовой аугментацией данных (test-time data augmentation, TTA)*. TTA — это способ исправить несовершенно усвоенные априорные знания модели для улучшения её прогностических характеристик. Мы предлагаем алгоритм для обучения политики аугментации данных для TTA, а также способ улучшения представлений нейросети с использованием TTA.

Актуальность работы

Машинное обучение — это область науки, которая изучает создание прогностических моделей на основе данных. Главной отличительной особенностью машинного обучения является высокое качество прогнозирования по сравнению с простыми подходами, основанными на правилах. Машинное обучение позволяет автоматически реконструировать правила и представления, необходимые для качественных предсказаний, даже если для этого требуется найти сложные зависимости в данных.

Наиболее успешным семейством моделей машинного обучения для сложно структурированных данных, таких как изображения, видео, естественные языки, аудио и молекулы, являются нейронные сети [22]. Глубокие нейронные сети (DNNs) можно определить как композицию параметризованных дифференцируемых модулей. Успех глубоких нейронных сетей в значительной степени основан на следующих факторах:

i) больших коллекции данных, которые включают от миллионов до сотен миллионов примеров;

ii) единообразных базовых блоках, которые позволяют создавать гибкие модели с большим количеством настраиваемых параметров;

ii) значительных объемах вычислительных мощностей, которые требуются для настройки больших моделей на больших наборах данных.

Сбор больших массивов данных, а также использование больших моделей не всегда возможны. Такие модели требуют много энергии даже на этапе предсказания и поэтому не под-

ходят для устройств с низким энергопотреблением, а сбор и разметка данных — крайне медленный и дорогостоящий процесс. Чтобы решить эту проблему, можно применять модели, которые используют априорные знания.

Понятие "априорные знания" происходит от латинской фразы *a priori* («из того, что было до») и обозначает информацию, независимую от рассматриваемого набора данных. Априорные знания могут быть использованы в алгоритме машинного обучения, чтобы улучшить его качество, уменьшить вычислительный бюджет, повысить скорость обучения и т.д. Априорная информация может быть интегрирована в модели глубокого обучения через:

i) **Проектирование моделей**. Проектирование моделей глубокого обучения часто подразумевает использование априорной информации. Наиболее заметным примером являются современные сверточные нейронные сети [18], которые используют информацию о пространственной структуре данных, а также эквивариантные сверточные сети [3], где ответ модели изменяется предсказуемо и плавно в соответствии с конкретными группами преобразований объекта.

ii) **Аугментацию данных**. Аугментация данных — это популярный инструмент, который искусственно расширяет количество объектов в данных, на которых обучается модель. При разработке стратегий аугментации данных используются предопределенные входные преобразования. Эти преобразования обычно сохраняют метку примера или изменяют ее предсказуемым образом. Разработка этих преобразований часто основывается на априорных знаниях о конкретной задаче.

iii) **Априорные распределения на параметры модели**. Байесовский подход является еще одним популярным способом интеграции априорных знаний в модели глубокого обучения. Предполагается, что априорные знания доступны в виде распределения над ненаблюдаемыми (скрытыми) переменными ν. Байесовский подход позволяет получить доступ к приближению апостериорного распределения (вариационному постериору) над скрытыми переменными $q_\phi(\nu)$ после наблюдения набора данных data, правдоподобия $p(\text{data} \mid \nu)$ и априорного распределения $p_{\text{prior}}(\nu)$.

$$\underbrace{p(\nu \mid \text{data})}_{\substack{\text{апостериорное распределение}}} = \frac{p(\text{data} \mid \nu) p_{\text{prior}}(\nu)}{\int_\nu p(\text{data} \mid \nu) p_{\text{prior}}(\nu)} \approx \underbrace{q_\phi(\nu)}_{\substack{\text{вариационный постериор} \\ \text{результат вариационного вывода}}} \tag{1}$$

В моделях глубокого обучения байесовский подход используется для нахождения аппроксимации апостериорного распределения над весами модели. Априорное распределение позволяет указать «предпочтения» в весах модели, которые будут учитываться при выводе приближения вариационного постериора.

Если набор данных достаточно велик, использование априорной информации не является строго обязательной. Например, архитектуры vision transformer [6] или MLP-mixer [31] позволя-

ют получить высокоточные модели для задач компьютерного зрения без явного использования сверток или других локальных преобразований, заданных «вручную» . Эти архитектуры обычно требуют большего количества данных и параметров по сравнению с моделями, которые используют априорные знания, но могут избежать ошибок, связанных с «ручной» интерграцией априорной информации при проектировании модели.

Люди, так же как и глубокие нейронные сети, активно используют априорную информацию. Строение мозга — это результат многолетней эволюции. Человеческий мозг спроектирован таким образом, чтобы люди быстро учились, избегали опасности и ежесекундно обрабатывали массу различных сигналов. Люди используют опыт, накопленный в течение жизни, для решения новых задач, но могут испытывать трудности, когда априорной информации недостаточно [7]. Поэтому мы предполагаем, что априорная информация необходима для искусственных нейронных сетей и будет важным компонентом для появления искусственного интеллекта.

Целью этой работы является разработка механизмов интеграции априорных знаний и использование этих механизмов для улучшения моделей глубокого обучения.

2. Основные результаты и выводы

Новизну работы можно резюмировать следующим образом:

1. Мы предложили использовать *разреживающий вариационный дропаут* — метод, основанный на априорном распределении, поощряющем разреженность. Он позволяет разреживать глубокие нейронные сети. Таким образом, было впервые продемонстрировано, что вариационный дропаут [16] может применяться для прореживания моделей глубинного обучения. Чтобы сделать возможным обучение с использованием метода, мы предложили использовать параметризацию, которая обладает эффектом подавления шума в градиенте, и локальную репараметризацию для сверток.
2. Мы разработали *глубокое априорное распределение* — метод, позволяющий использовать выразительную генеративную модель, вариационный автоэнкодер (VAE) [17] в качестве априорного распределения над весами моделей глубокого обучения. Мы предложили вариационную нижнюю оценку, которая позволяет использовать априорные распределения, основанные на VAE, для вариационного вывода, а также для классического обучения.
3. Мы предлагаем *тестовую аугментацию данных (TTA) для ансамблей* — простой, но хорошо работающий метод, который позволяет интегрировать априорные знания в ансамбль моделей глубокого обучения. TTA улучшает прогностическую способность ансамблей [19; 21] с незначительными дополнительными вычислительными затратами.
4. Мы предлагаем *жадный поиск политики TTA* — метод, который может обучить политику тестовой аугментации данных. Другими словами, метод может автоматически выбирать

априорные знания, которые необходимо использовать во время TTA. Это помогает избежать слишком агрессивных аугментаций, которые вредны для качества прогнозирования.

5. Мы предлагаем метод *средних представлений*, который адаптирует TTA для улучшения представлений. Метод позволяет расширить область применение TTA до интеграции априорных знаний в представления.

Теоретическая и практическая значимость. Мы предлагаем метод разреживания, который может быть использован для сжатия и ускорения моделей глубокого обучения. Сжатие и ускорение важны для работы нейронных сетей на маломощных устройствах. Еще одним результатом этой работы является инструмент, который позволяет использовать гибкие (сложные) априорные распределения над весами модели, которые могут лучше отражать априорные знания о моделях глубокого обучения. Например, априорное распределение может представлять корреляции между весами и охватывать несколько мод. Мы также разработали методы ансамблирования с помощью тестовой аугментации данных (TTA). Использование этих методов повышает надежность модели, что крайне важно для потенциально опасных областей применения, таких как медицинская диагностика.

Методология и методы исследования. В этой работе мы применяем глубокое обучение, дважды стохастический вариационный вывод, вероятностное моделирование, генеративные модели, аугментацию данных, а также непрерывную и дискретную оптимизацию.

Воспроизводимость Мы приводим подробное описание предлагаемых методов и экспериментов. Код ко всем статьям находится в открытом доступе.

Результаты, выносимые на защиту.

1. Вывод о том, что *вариационный дропаут* способен выучить разреженные решения.
2. *Глубокое априорное распределение* — инструмент, который позволяет использовать неявные распределения для вариационного вывода.
3. *Аугментация данных во время предсказания (TTA)* для ансамблей.
4. *Жадный поиск политик для TTA* — алгоритм для выбора политики аугментации данных во время тестирования.
5. *MeTTA*, алгоритм использования TTA для улучшения представлений.

Личный вклад в результаты, выносимые на защиту. Автором диссертации были получены все заявленные результаты. Во всех упомянутых случаях как текст, так и экспериментальные результаты, представленные в статьях, являются результатом сотрудничества всех авторов. В первой статье «Вариационный Дропаут Разреживает Глубокие Нейронные Сети» автор предложил использовать локальную репараметризацию для свёрточных слоев, внес вклад в алгоритм обучения и обнаружил эффект разреженности в глубоких свёрточных сетях. В работе «Глубокое Априорное Распределение» автор внес основную идею и ввел функцию потерь. В

работе «Подводные Камни Оценки Неопределенности и Ансамблирования в Глубоком Обучении» автор предложил использовать аугментацию данных во время предсказания (TTA) для ансамблей. В статье «Средние Представления с Аугментацией Данных во Время Тестирования для Ансамблирования Представлений» автор предложил задействовать TTA для повышения качества представлений. В «Жадный Поиск Политик: Простой Безйлайн для Аугментации Данных во Время Тестирования» автор внес вклад в основную идею и провел эксперименты с ImageNet.

Публикации и апробация работы

* означает равный вклад соавторов

Публикации повышенного уровня.

1. *Дмитрий Молчанов*, Арсений Ашуха*, Дмитрий Ветров* Вариационный дропаут разреживает глубокие нейронные сети // Международная Конференции по Машинному Обучению, ICML, стр. 2498-2507. PMLR, 2017. Конференция ранга A* по версии CORE.

2. *Александр Лыжов*, Юлия Молчанова*, Арсений Ашуха*, Дматрий Молчанов*, Дмитрий Ветров* Жадный поиск политик: простой безйлайн для аугментации данных во время тестирования // Международная Конференция по Неопределенности в Искусственном Интеллекте, UAI 2020. Конференция ранга A* по версии CORE.

Публикации стандартного уровня.

1. *Андрей Атанов*, Арсений Ашуха*, Кирилл Струминский, Дмитрий Ветров, Макс Веллинг* Глубокое априорное распределение // Международная конференция по обучению представлений, ICLR 2019. Проиндексирована в SCOPUS. С 2021 ICLR является конференцией ранга A* по версии CORE.

Остальные публикации.

1. *Арсений Ашуха*, Александр Лыжов*, Дмитрий Молчанов*, Дмитрий Ветров* Подводные камни оценки неопределенности и ансамблирования в глубоком обучении // Международная конференция по обучению представлений, ICLR 2020. С 2021 ICLR является конференцией ранга A* по версии CORE.

2. *Арсений Ашуха, Андрей Атанов, Дмтрий Ветров* Средние представления с аугментацией данных во время тестирования для ансамблирования представлений // Неопределенность и надежность в глубоком обучении, ICML, 2021.

Доклады на конференциях

1. Стендовая презентация «Вариационный Дропаут Разреживает Глубокие Нейронные Сети» , Международная конференция по машинному обучению, Сидней, 2017.
2. Доклад о «Вариационный Дропаут Разреживает Глубокие Нейронные Сети» , Семинар исследовательской группы Байесовских методов, Москва, 2017.
3. Стендовая презентация «Глубокое Априорное Распределениеaw» , Международная конференция обучению представлений, Новый Орлеан, Луизиана, США, 2019 год.
4. Стендовая презентация «Жадный Поиск Политик: Простой Безйлайн для Аугментации Данных во Время Тестирования» , Неопределенность в искусственном интеллекте, удалённо, 2020.

Автор также внес свой вклад в следующие публикации

1. *Кирилл Неклюдов, Дмитрий Молчанов, Арсений Ашуха, Дмитрий Ветров* Структурированное Байесовское разреживание с помощью лог-нормального мультипликативного шума // Международная Конференция по Нейро-сетевым Системам Обработки Информации, NeurIPS 2017. Конференция ранга A* по версии CORE.

2. *Кирилл Неклюдов*, Дмитрий Молчанов*, Арсений Ашуха*, Дмитрий Ветров* Дисперсионные сети: когда ожидания не соответствуют вашим ожиданиям // Международная конференция по обучению представлений, ICLR 2019. Проиндексировано в SCOPUS. С 2021 ICLR является конференцией ранга A* по версии CORE.

3. *Андрей Атанов, Арсений Ашуха, Дмитрий Молчанов, Кирилл Неклюдов, Дмитрий Ветров* Оценка неопределенности с помощью стохастической батч нормализации // ICLR Симпозиум 2018 // Международный Симпозиум по Нейронным Сетям, стр. 261-269. Springer, Cham, 2019.

4. *Макс Кочуров, Тимур Гарипов, Дмитрий Подоприхин, Дмитрий Молчанов, Арсений Ашуха, Дмитрий Ветров* Байесовское инкрементное обучение для глубоких нейронных сетей // ICLR Симпозиум 2018.

5. *Евгений Никишин, Арсений Ашуха, Дмитрий Ветров* Доменная адаптация без учителя через общую скрытую динамику для обучения с подкреплением // Симпозиум по Байесовскому DL, NeurIPS 2019.

6. *Андрей Атанов, Александра Волохова, Арсений Ашуха, Иван Сосновик, Дмитрий Ветров* Полу-обусловленные нормализационные потоки для обучения на частично размеченных данных // Воркшоп по Обратимым Нейронным Сетям и Нормализационным Потокам, ICML, 2019.

Объем и структура работы. Диссертация содержит введение, содержание публикаций и заключение. Полный объем диссертации составляет 107 страниц.

3. Содержание работы

3.1. Вариационный дропаут разреживает глубокие нейронные сети

Глубокие нейронные сети *де-факто* стали инструментом для решения многих реальных задач машинного обучения — от детекции [34] и перевода [33] до сворачивания белков [15]. Однако отличная производительность всегда достигается за счет большого количества параметров, что приводит к высоким требованиям с точки зрения памяти и вычислений. Одним из способов решения этой проблемы является обучение разреженной модели, в которой большинство весов равно нулю. Таким образом, размер модели и вычислительные затраты будут уменьшены. В работе «Вариационный Дропаут Разреживает Глубокие Нейронные Сети» мы предлагаем модель под названием *разреживающий вариационный дропаут*, которая может обучать разреженные глубокие нейронные сети.

Описание модели

Мы рассматриваем проблему обучения с учителем с набором данных $D = (x_i, y_i)_{i=0}^N$, где x_i является объектом, а y_i — его меткой. Модель обучается с помощью вариационного вывода над весами глубокой нейронной сетей. Цель вариационного вывода состоит в том, чтобы настроить вариационное приближение $q_\phi(W) \approx p(W \mid D)$, где ϕ — параметры вариационного приближения, подлежащие обучению. Целевой функцией вариационного вывода является стохастическая оценка вариационной нижней оценки $\mathcal{L}(\phi)$ (ELBO):

$$\mathcal{L}(\phi) \simeq \mathcal{L}^{SGVB}(\phi) = L_{\mathcal{D}}^{SGVB}(\phi) - D_{KL}(q_\phi \| p) \to \max_{\phi} \tag{2}$$

$$L_{\mathcal{D}}(\phi) \simeq L_{\mathcal{D}}^{SGVB}(\phi) = \frac{N}{M} \sum_{m=1}^{M} \log p(y_m \mid x_m, \hat{W}), \quad \hat{W} \sim q(W \mid \phi), \tag{3}$$

где $L_{\mathcal{D}}^{SGVB}$ оценка логарифма правдоподобия, правдоподобие $p(y_m \mid x_m, \hat{W})$ задаётся с помощью глубокой нейронной сети, $p(W)$ априорное распределение на веса W, $D_{KL}(u, v) = \int dx\ u(x) \log \frac{u(x)}{v(x)}$ обозначает дивергенцию Кульбака — Лейблера, также называемую относительной энтропией, которая представляет собой асимметричное расстояние между двумя распределениями. $L_{\mathcal{D}}^{SGVB}$ отвечает за низкую ошибку предсказания и контролирует, насколько хорошо модель $p(y_m \mid x_m, \hat{W})$ работает в обучающем наборе данных, а $D_{KL}(q_\phi, p)$ контролирует, насколько хорошо модель удовлетворяет априорному распределению $p(W)$. Мы также предполагаем, что $q_\phi(W)$ поддерживает репараметризацию [17; 29].

В разреживающем вариационном дропауте мы используем полностью факторизованную Гауссовскую аппроксимацию в аддитивной параметризации:

$$q(w_{ij} \mid \theta_{ij}, \alpha\theta_{ij}) = \underbrace{\mathcal{N}(w_{ij} \mid \theta_{ij}, \alpha_{ij}\theta_{ij}^2)}_{\substack{\text{мультипликативная параметризация} \\ \textbf{(author?)}\ [16]}} = \underbrace{\mathcal{N}(w_{ij} \mid \theta_{ij}, \mu_{ij}^2)}_{\substack{\text{аддитивная параметризация} \\ \text{предлагается в этой работе}}}, \tag{4}$$

это обеспечивает снижение шума в градиенте по параметрам θ_{ij}. Модель использует логравномерное априорное распределение

$$p(\log|w_{ij}|) = \text{const} \quad \Leftrightarrow \quad p(|w_{ij}|) \propto \frac{1}{|w_{ij}|}. \tag{5}$$

Также алгоритм использует *локальную репараметризациию* (LRT) [16] как для полносвязанных, так и для свёрточных слоёв. LRT вычисляет распределение скрытых представлений аналитически на каждом слое, опираясь на предположение, что входные данные не являются случайной величиной. В результате LRT позволяет уменьшить дисперсию градиента. LRT может быть интерпретирован как различные реализации весов $\hat{W}$ из $q(W \mid \phi)$ для каждого объекта в минибатче.

Экспериментальные результаты

Мы тестируем разреживающий вариационный дропаут на задачах классификации MNIST [23], CIFAR-10 и CIFAR-100 [18]. При обучении на датасете MNIST метод позволяет сократить число используемых параметров до 280 раз по сравнению с исходной неразреженной сетью. Результаты представлены в таблице 1. При обучении на CIFAR уровень разреженности увеличивается с увеличением размера модели, в то время как ошибка остается на том же уровне, что и у неразреженных моделей.

Кроме того, мы показываем, что разреживающий вариационный дропаут удаляет 100% весов модели в отсутствии зависимости между объектами и метками, или, другими словами, когда разметка случайна [35].

Network	Method	Error %	Sparsity per Layer %	$\frac{\lvert W\rvert}{\lvert W_{\neq 0}\rvert}$
LeNet-300-100	Original	1.64		1
	Pruning	1.59	$92.0 - 91.0 - 74.0$	12
	DNS	1.99	$98.2 - 98.2 - 94.5$	56
	SWS	1.94		23
	(ours) Sparse VD	1.92	$98.9 - 97.2 - 62.0$	**68**
LeNet-5-Caffe	Original	0.80		1
	Pruning	0.77	$34 - 88 - 92.0 - 81$	12
	DNS	0.91	$86 - 97 - 99.3 - 96$	111
	SWS	0.97		200
	(ours) Sparse VD	0.75	$67 - 98 - 99.8 - 95$	**280**

Таблица 1: Сравнение различных методов разреживания (Pruning [13; 12], DNS [11], SWS [32]) на архитектурах LeNet-300-100 (3 слоя) и LeNet-5-Caffe (4 слоя). Sparse VD, предложенный в этой работе, обеспечивает высокий уровень разреженности при том же уровне точности.

Ретроспектива

Разреживающий вариационный дропаут был использован для разреживания нейронных сети в ведущих ИТ-компаниях. Однако дальнейшие исследования показали, что методы разрежи-

вания, основанные на концепции обрезки весов (pruning), могут показывать более хорошие результаты [10]. Как показывают дальнейшие исследования, разреженное решение, которое находит разреживающий вариационный дропаут, является только локальным оптимумом ELBO, а лучшие значения ELBO могут быть достигнуты с помощью менее гибкой вариационной аппроксимации $q(w_{ij}) = N(w_{ij} \,|\, 0, \sigma_{ij})$ [27].

Известно, что обучение глубоких нейронных сетей с шумом является сложным и нестабильным процессом. В меньшей степени это относится к методу, предложенному в данной главе. Все дисперсии весов инициализируются небольшими значениями и не претерпевают значительных изменений во время обучения. Поэтому использование разреживающего вариационного дропаута не ухудшает производительность, однако почти не вносит шума. Таким образом, реальное влияние стохастики в сети крайне мало, и SparseVD можно рассматривать как метод, основанный на разреживающем аддитивном регуляризаторе.

3.2. Глубокое априорное распределение для обучения нейронных сетей

Вариационный вывод — это инструмент, который после наблюдения данных позволяет преобразовать априорное распределение над параметрами моделей машинного обучения в приближение апостериорного распределения. Априорные распределения играют важную роль во многих современных методах квантования [32], разреживания [26; 28] и сжатия [25]. Однако используемые априорные распределения — недостаточно гибкие и не могут задавать корреляции между весами и охватывать несколько мод. В этой работе мы предлагаем *глубокое априорное распределение* — метод, который позволяет обучать и использовать сложные априорные распределения, определяемые с помощью неявной генеративной модели

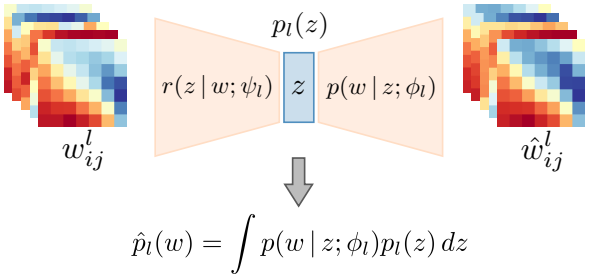
$$\hat{p}_l(w) = \int p(w \,|\, f_\phi(z)) p_l(z)\, dz, \tag{6}$$

где условное распределение $p(w \,|\, f_\phi(z))$ явное параметрическое распределение, f_ϕ нейронная сеть, а $p_l(z)$ априорное распределение в пространстве скрытых переменных, которое не зависит от обучаемых параметров. Распределение в уравнении (6) может быть интерпретировано как смесь бесконечного количества распределений (распределения «индексируются» вектором действительных чисел z). Неявные априорные распределения (6) могут задавать сложные распределения, которые, следовательно, могут лучше выражать априорные знания о параметрах модели.

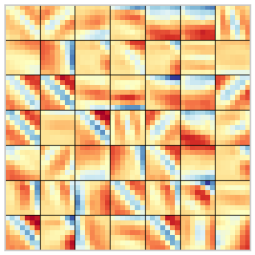
Обучение априорного распределения

Для того, чтобы обучить глубокое априорное распределение (DWP) (6), мы выполняем следующие шаги:

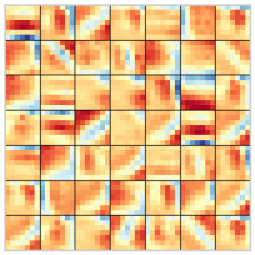
1. определяем архитектуру априорного распределения в виде вариационного автоэнкодера;
2. собираем набор нейросетей, обученных на доступном наборе данных;



(a) Обучение DWP



(b) Фильтры



(c) Семплы из DWP

Рис. 1: В подфигуре 1a мы показываем процесс обучения априорного распределения над свёрточными ядрами одного слоя. Сначала мы обучаем кодировщик $r(z \mid w; \phi_l)$ и декодер $p(w \mid z; \psi_l)$ с помощью VAE [17]. Затем мы используем декодер для построения априорного распределения $\hat{p}_l(w)$. В подфигуре 1b мы показываем несколько выученных ядер размера 7×7 из первого свёрточного слоя свёрточной нейронной сети, обученных на наборе данных NotMNIST. В подфигуре 1c мы показываем реализации из глубокого априорного распределения (DWP), которое было обучено на ядрах обученных нейронных сетей.

3. обучаем априорное распределение на весах обученных сетей.

Обученное априорное распределение может быть использовано для вариационного вывода с новым набором данных. Рисунки 1a показывают, как априорное распределение может быть построено на основе обученной вариационной модели автоэнкодера. Рисунки 1b, 1c демонстрируют реализации из обученной генеративной модели в сравнении с обученными ядрами.

Вариационный вывод с неявными априорными распределениями

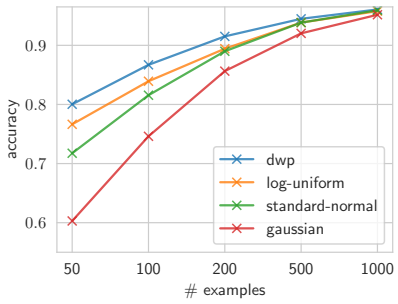
Мы рассматриваем проблему обучения с учителем на наборе данных $D = (x_i, y_i)_{i=0}^N$, где x_i объект, а y_i его метка. Мы обучаем модель $p(y_i \mid x_i, W)$ с вариационным выводом, оптимизируя следующую функцию выгоды:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \mathbb{E}_{q_\theta(W)} \log p(y_i \mid x_i, W) - D_{\mathrm{KL}}(q_\theta(W) \| p(W)) \to \max_\theta, \tag{7}$$

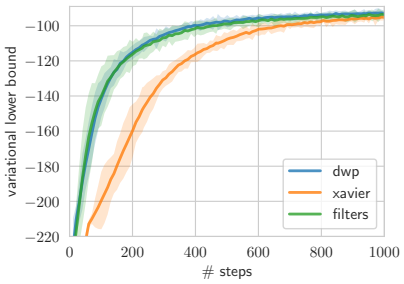
где W обозначает веса нейронной сети, $q_\theta(W)$ вариационное распределение, которое может быть репараметризовано [17; 9], а $p(W)$ априорное распределение.

Мы рассматриваем нейронную сеть с L свёрточными слоями и обозначаем параметры l-го свёрточного слоя как $w^l \in \mathrm{R}^{I_L \times O_l \times H_l \times W_l}$, где I_L количество входных каналов, O_L количество выходных каналов, H_L и W_l пространственные размеры ядер. Параметры нейронной сети обозначаются как $W = (w^1, \ldots, w^L)$. Вариационное приближение $q_\theta(W)$ и априорное распределение $p(W)$ имеют следующую факторизацию по слоям, фильтрам и каналам:

$$q_\theta(W) = \prod_{l=1}^{L} \prod_{i=1}^{I_l} \prod_{j=1}^{O_l} q(w_{ij}^l \mid \theta_{ij}^l) \qquad p(W) = \prod_{l=1}^{L} \prod_{i=1}^{I_l} \prod_{j=1}^{O_l} p_l(w_{ij}^l), \tag{8}$$



(a) ConvNet on MNIST



(b) VAE on MNIST

Рис. 2: (2a) Для разного размера обучающих выборок мы демонстрируем точность модели, обученной вариационным выводом с полностью факторизованными вариационными аппроксимациями с различными априорными распределениями: *глубокое априорное распределение* (dwp), лог-равномерное (log-uniform), стандартное нормальное (standard-normal), обученное нормальное с полной матрицей ковариации (gaussian). Мы обнаружили, что вариационный вывод с глубоким априорным распределением на веса обеспечивает лучшую среднюю точность, чем обучение с другими априорными распределениями. (2b) Инициализация весов моделей с глубокими приоритетами веса или изученными фильтрами значительно увеличивает скорость обучения по сравнению с инициализацией Ксавье. Это дает некоторое доказательство того, что *глубокое априорное распределение* выучивает репрезентативную аппроксимацию истинного распределения ядер.

где $w_{ij}^l \in \mathrm{R}^{H_L \times W_l}$ — это ядро j-го канала в i-м фильтре l-го свёрточного слоя.

KL-дивергенция с неявным априорным распределением (формула 6) не может быть вычислена в явной форме или оценена без смещения. Чтобы сделать оценку вариационной нижней оценки возможным, мы вводим вспомогательную нижнюю оценку KL-дивергенции:

$$D_{\mathrm{KL}}(q(W)\|\hat{p}(W)) = \sum_{l,i,j} D_{\mathrm{KL}}(q(w_{ij}^l|\theta_{ij}^l)\|\hat{p}_l(w_{ij}^l)) \leq \sum_{l,i,j} \big(- H(q(w_{ij}^l \,|\, \theta_{ij}^l)) + \\ + \mathbb{E}_{q(w_{ij}^l \,|\, \theta_{ij}^l)} \big[D_{\mathrm{KL}}(r(z \,|\, w_{ij}^l; \psi_l)\|p_l(z)) - \mathbb{E}_{r(z \,|\, w_{ij}^l; \psi_l)} \log p(w_{ij}^l \,|\, z; \phi_l) \big] \big) = D_{\mathrm{KL}}^{bound}, \quad (9)$$

где $r(z \,|\, w; \psi_l)$ является вспомогательной моделью вывода для априорного распределения для l-го слоя $\hat{p}_l(w)$. Итоговая вспомогательная вариационная нижняя оценка $\mathcal{L}^{aux}(\theta, \psi)$ имеет следующий вид

$$\mathcal{L}^{aux}(\theta, \psi) = L_D - D_{\mathrm{KL}}^{bound} \leq \mathcal{L}(\theta) = L_D - D_{\mathrm{KL}}(q_\theta(W)\|\hat{p}(W)). \quad (10)$$

Экспериментальные результаты

Мы продемонстрировали, что использование *глубокого априорного распределения* может быть полезным при обучении с ограниченными размеченными данными. Инициализация весов с помощью *глубокого априорного распределения* позволяет процессу обучения быстрее сходиться. Результаты показаны на рис. 2.

Ретроспектива

Глубокое априорное распределение достаточно сложно обучить для больших моделей и на больших наборах данных. Однако, у нас нет знаний о том, в на каких задачах DWP работает лучше всего. Например, [20] успешно применили *глубокое априорное распределение* для трехмерной магнитно-резонансной томографии (МРТ).

Генеративные модели, которые генерируют веса нейронных сетей (гиперсети), в последнее время стали популярными [30; 8; 2]. *Глубокое априорное распределение* можно рассматривать как небольшой шаг к генеративным моделям обученных нейронных сетей, что может стать важной темой для будущих исследований генеративных моделей.

3.3. Аугментация данных во время тестирования улучшает ансамбли без дополнительных затрат ресурсов

Аугментация данных во время тестирования (TTA) [18] — это простой метод, который усредняет прогнозы модели по различным аугментациям (искажениям) входных данных. TTA делает прогнозы инвариантными к аугментациям и позволяет улучшить прогностические характеристики модели глубокого обучения.

Мы предлагаем использовать TTA для ансамблей. В этом случае каждый член ансамбля применяется к случайному искажению объекта (формулы 12, 13).

$$p(y \mid x) = \frac{1}{S}\sum_{i=1}^{S} p(y|x, \theta_i) \tag{11}$$

(a) Классический ансамбль

$$p(y \mid x) = \frac{1}{S}\sum_{i=1}^{S} p(y \mid \hat{x}_i, \theta_i), \tag{12}$$

$$\text{where } \hat{x}_i \sim p_{\text{aug}}(\cdot \mid x) \tag{13}$$

(b) Ансамбль с TTA

TTA улучшает ансамбли с незначительными дополнительными вычислительными затратами во время вывода. Эмпирическая эффективность метода продемонстрирована с помощью различных ансамблей ResNet-50 [14] в ImageNet (рис. 4). Интересно, что одна модель (single model) с TTA работает на том же уровне ошибки, что и методы, которые требуют значительно большего количества параметров и вычислительного бюджета. TTA — это простой, но мощный подход, который не получил достаточно внимания в недавних работах.

3.4. Жадный поиск политик для аугментации данных во время тестирования

Аугментация данных — это одна из популярных техник, применяемых для обучения глубоких нейронных сетей. Она позволяет искусственно увеличить размер датасета и интегрировать априорные знания об инвариантности предметной области в глубокую нейронную сеть. Однако в большинстве случаев обученные глубокие нейронные сети оказываются не полностью инвариантными к искажениям, которые были внесены аугментациями. Другими слова-

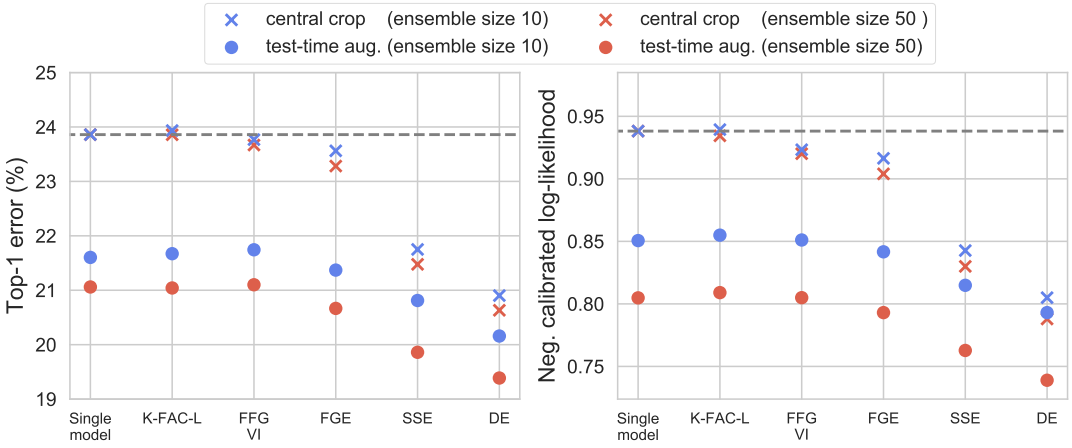


Рис. 4: Как читать результаты: $\times \xrightarrow[10\text{ samples}]{\text{test-time aug}} \bullet$, $\times \xrightarrow[50\text{ samples}]{\text{test-time aug}} \bullet$. Ошибка предсказаний и отрицательное калиброванное лог-правдоподобие (чем меньше, тем лучше) для различных методов ансамблирования на датасете ImageNet. Мы исследуем качество двух режимов. *Central-crop evaluation* ($\times\times$) означает, что каждый член ансамбля применяется к центральной части изображения, а *test-time data augmentation* ($\bullet\bullet$) означает, что каждый член ансамбля применяется к отдельному случайному искажению изображения. **Аугментация данных во время тестирования значительно улучшает ансамбли без дополнительных вычислительных затрат.**

ми, глубокие нейронные сети не полностью усваивают априорные знания вносимые аугментацией. Чтобы устранить эту проблему, используется аугментация данных во время тестирования. Она учитывает несколько разных искажений для каждого объекта на этапе предсказания. Однако современные методы аугментации данных спроектированы специально для аугментации во время обучения и обычно не работают наилучшим образом во время предсказания. Одна из причин заключается в том, что аугментация данных с высоким уровнем шума, например, randaugment [4], может действовать как регуляризация во время обучения. Но высокий уровень шума может ухудшить производительность во модели во время предсказания.

В этой работе мы используем априорные знания о наборе преобразований, сохраняющих метки, чтобы выучить оптимальную политику для аугментации данных во время тестирования. Основная цель состоит в том, чтобы продемонстрировать, что обучение политики аугментации данных во время тестирования возможно.

Поиск жадной политики (GPS) итеративно создаёт политику аугментации данных во время тестирования. Мы рассматриваем задачу обучения политики TTA для одной предварительно обученной классификационной модели. Прогнозы модели будут усреднены по семплам искажений из обученной политики, для получения финального предсказания. Иллюстрация примеров, искаженных обученной GPS-политикой, доступна на рис. 5.

GPS организован следующим образом:

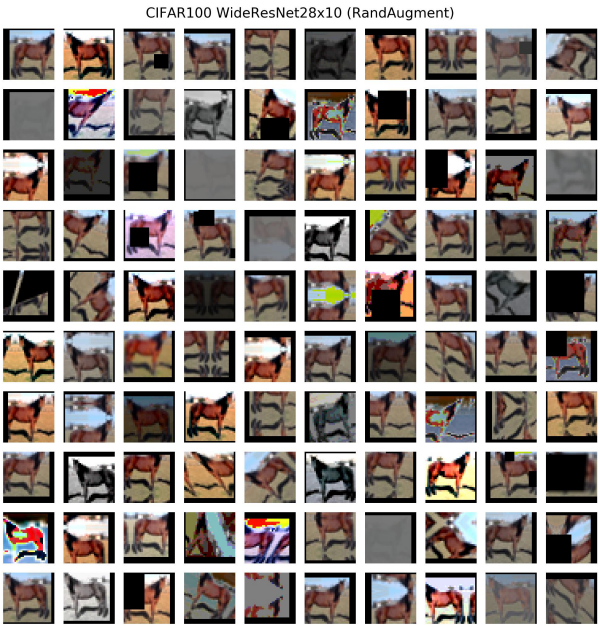


Рис. 5: Иллюстрация политики аугментации данных обученной жадным поиском политики.

i) Первым делом GPS семплирует большой набор под-политик. Каждая под-политика представляет собой комбинацию двух-трех случайно выбранных преобразований со случайно выбранными магнитудами, которые контролируют максимальную величину искажения.

ii) GPS выбирает подполитику, которая максимально улучшает текущую политику, добавляет ее в текущую политику.

iii) Повторять шаг, *ii* пока количество под-политик меньше, чем требуется.

Мы обнаружили, что обычная точность в качестве критерия поиска работает значительно хуже, чем откалиброванный отрицательный логарифм правдоподобия. Точность, вероятно, является слишком «шумной» , а предсказания требуют калибровки, чтобы уменьшить влияние возможную неправильную температуру. Несмотря на простоту, *жадный поиск политики* работает лучше, чем более сложная оптимизация на основе RL [24].

Экспериментальные результаты

Эмпирически мы демонстрируем, что *жадный поиск политики* улучшает результаты классической TTA для данных внутри и вне домена без каких-либо дополнительных вычислительных затрат во время вывода. GPS также может быть применен к ансамблям. Пример результатов показан на рисунке 6.

3.5. Улучшение представлений с помощью ансамблирования

Ансамблирование, это популярный инструмент, который улучшает оценки неопределенности и качество моделей глубокого обучения. Однако ансамбли обычно применяются для улучше-

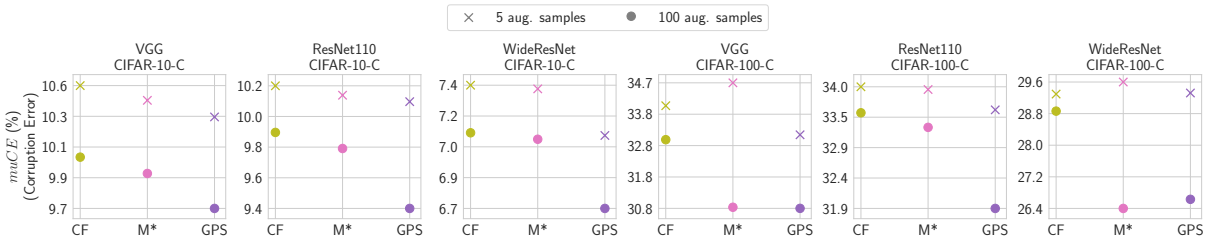


Рис. 6: Средняя ненормализованная ошибка *(muCE)* на поврежденных (out-of-domain) данных из CIFAR-10/100 для различных стратегий TTA: случайная вырезанная часть изображения и горизонтальные перевороты (CF), модифицированное RandAugment с M, найденное с помощью поиска по сетке (M^*) и GPS политика (GPS). Обучаемые методы TTA были обучены на чистых, неповрежденных данных. В большинстве случаев политики GPS более устойчивы к сдвигу домена по сравнению с альтернативными стратегиями.

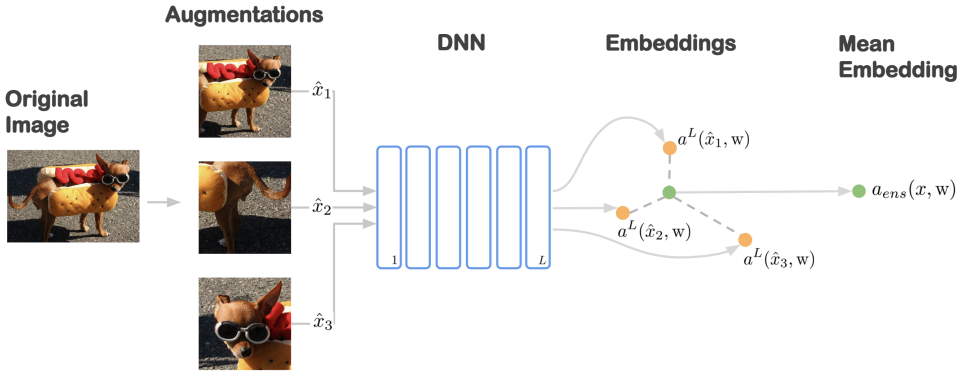


Рис. 7: Чтобы получить *среднее представления*, MeTTA усредняет активации одной сети для разных искажений объекта. MeTTA не влияет на этап обучения и может быть применена к предобученной сети.

ния прогнозов модели и не могут быть применены для улучшения качества представлений, потому что независимые сети имеют несогласованные представления. Однако повышение качества представлений важно для многих проблем, например, для поиска изображений, сопоставления, верификации и систем рекомендаций.

Мы предлагаем ***средние представления с TTA (MeTTA)*** — простой метод ансамблирования представлений. Метод усредняет представления с L-го слоя модели $a^L(\,\cdot\,; \mathrm{w})$ одной модели по различным искажениям объекта x

$$a_{ens}(x; \mathrm{w}) = \mathbb{E}_{\hat{x} \sim p_{aug}(\cdot \,|\, x)} a^L(\hat{x}; \mathrm{w}) \cong \frac{1}{S} \sum_{s=1}^{S} a^L(\hat{x}_s; \mathrm{w}), \text{ where } \hat{x}_s \sim p_{aug}(\cdot \,|\, x), \tag{14}$$

где S количество аугментаций (искажений) для одного изображения, w веса сети, а $a_{ens}(x; \mathrm{w})$ - среднее представления. Эмпирически метод работает как с моделями обученными с учителем, так и с моделями обученными самообучением (таблица 2). Иллюстрация MeTTA доступна на рисунке 7. Интерполяция между представлениями полученными из центрально части изображения и представлениями полученными с помощью MeTTA показана на рисунке 8. MeTTA

Problem	Model	Width	SK	# Params (M)	Central crop Embeddings	Mean embeddings with TTA $N = 10$	$N = 32$
Self-supervised features (SimCLRv2)	ResNet50	1×	False	24	71.7	73.3 (+1.6%)	73.8 (+2.1%)
		1×	True	35	74.6	75.8 (+1.2%)	76.2 (+1.6%)
	ResNet101	2×	False	170	77.0	78.1 (+1.1%)	78.5 (+1.5%)
		2×	True	257	79.0	79.8 (+0.8%)	79.9 (+0.9%)
	ResNet152	3×	True	795	79.8	80.3 (+0.4%)	80.7 (+0.9%)
Supervised features	ResNet50	1×	False	24	76.6	78.0 (+1.4%)	78.5 (+1.9%)
		1×	True	35	78.5	79.7 (+1.2%)	80.2 (+1.7%)
	ResNet101	2×	False	170	78.9	80.2 (+1.3%)	80.6 (+1.7%)
		2×	True	257	80.1	81.0 (+0.9%)	81.3 (+1.3%)
	ResNet152	3×	True	795	80.5	81.4 (+0.9%)	81.9 (+1.4%)

Таблица 2: Сравнение различных методов вывода эмбедингов. В таблице представлена точность на наборе данных ImageNet [5] для линейной оценки представлений со 100% метками. Для экспериментов с самообучением (self-supervised), мы использовали модели, которые были предобучены с помощью SimCLRv2 [1] `https://github.com/google-research/simclr`. SK расшифровывается как селективные ядра.

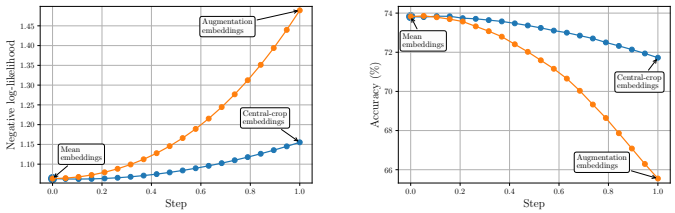


Рис. 8: Отрицательный логарифм правдоподобия (слева) и точность (справа) для линейно интерполированных представлений вида $(1-\alpha)\cdot x+\alpha\cdot y$, где x среднее представление, а y представление центральной области картинки. Показатели для каждого шага α усредняются по валидационным изображениям а так-же по различным аугментациям для оранжевой кривой.

показывает, как ансамбли могут быть применены для повышения качества представлений. Это может открыть множество новых приложений.

4. Заключение

В заключительном разделе мы резюмируем основной научный вклад этой работы.

1. Мы предложили *разреживающий вариационный дропаут* — метод разрежения глубоких нейронных сетей. Метод использует вариационный вывод с лог-равномерным априорным распределением. Чтобы сделать возможным обучение *разреживающего вариационного дропаута*, было предложено использовать аддитивную параметризацию и локальную репараметризацию, которые уменьшают дисперсию градиентов. Метод позволяет обучать модели с высоким уровнем разреженности (до 270 раз в наших экспериментах).

2. Мы предложили *глубокое априорное распределение* — метод, который позволяет обучать генеративную модель на ядрах свёрточной нейронной сети и использовать обученную генеративную модель в качестве априорного распределения во время обучения свёрточных нейронных сетей. Чтобы обучаться с *глубоким априорным распределением*, мы разработали специальную форму вариационного вывода, которая может работать с неявным априорными распределениями. Обучение с *глубоким априорным распределением* улучшает качество в условиях ограниченных данных и позволяет быстрее сходиться.

3. Мы предложили *аугментацию данных во время вывода* для ансамблей, что позволяет разнообразить прогнозы и повысить качество и способность ансамблей оценивать неопределенность.

4. Мы предложили *жадный поиск политики* — метод, который позволяет обучить политику аугментации данных для применения во время тестирования. Усреднение прогнозов по искажениям из обученной политики повышает производительность нейросети в внутри домена и за его приделами.

5. Мы предложили MeTTA — метод, который использует ансамблирование для повышения качества представлений.

Список литературы

[1] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[2] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628--8638, 2021.

[3] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990--2999. PMLR, 2016.

[4] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Rachit Dubey, Pulkit Agrawal, Deepak Pathak, Thomas L Griffiths, and Alexei A Efros. Investigating human priors for playing video games. *arXiv preprint arXiv:1802.10217*, 2018.

[8] Emilien Dupont, Yee Whye Teh, and Arnaud Doucet. Generative models as distributions of functions. *arXiv preprint arXiv:2102.04776*, 2021.

[9] Michael Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. *arXiv preprint arXiv:1805.08498*, 2018.

[10] Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

[11] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. In *Advances In Neural Information Processing Systems*, pages 1379--1387, 2016.

[12] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[13] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pages 1135--1143, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[15] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Kathryn Tunyasuvunakool, Olaf Ronneberger, Russ Bates, Augustin Žídek, Alex Bridgland, et al. High accuracy protein structure prediction using deep learning. *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book)*, 22:24, 2020.

[16] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575--2583, 2015.

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097--1105, 2012.

[20] Anna Kuzina, Evgenii Egorov, and Evgeny Burnaev. Bayesian generative models for knowledge transfer in mri semantic segmentation problems. *Frontiers in neuroscience*, 13:844, 2019.

[21] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

[22] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541--551, 1989.

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278--2324, 1998.

[24] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *arXiv preprint arXiv:1905.00397*, 2019.

[25] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. *arXiv preprint arXiv:1705.08665*, 2017.

[26] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. *arXiv preprint arXiv:1701.05369*, 2017.

[27] Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variance networks: When expectation does not meet your expectations. In *International Conference on Learning Representations*, 2019.

[28] Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, and Dmitry P Vetrov. Structured bayesian pruning via log-normal multiplicative noise. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[29] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, 2014.

[30] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

[31] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[32] Karen Ullrich, Edward Meeds, and Max Welling. Soft weight-sharing for neural network compression. *arXiv preprint arXiv:1702.04008*, 2017.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998--6008, 2017.

[34] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[35] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.