

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ  
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

На правах рукописи  
УДК 519.254

Бахтеев Олег Юрьевич

ПОСЛЕДОВАТЕЛЬНЫЙ ВЫБОР МОДЕЛЕЙ  
ГЛУБОКОГО ОБУЧЕНИЯ ОПТИМАЛЬНОЙ СЛОЖНОСТИ

05.13.17 — Теоретические основы информатики

Диссертация на соискание ученой степени  
кандидата физико-математических наук

Научный руководитель:  
д.ф.-м.н. В. В. Стрижов

Москва — 2018

# Оглавление

	Стр.
Введение . . . . .	3
Глава 1. Постановка задачи последовательного выбора моделей . . . . .	10
1.1. Оптимизация параметров в задаче выбора структуры модели . . . . .	16
1.2. Оптимизация гиперпараметров модели . . . . .	18
1.3. Порождение и выбор структуры модели глубокого обучения . . . . .	20
1.4. Метаоптимизация моделей глубокого обучения . . . . .	25
1.5. Выбор структур моделей специального вида . . . . .	26
Глава 2. Выбор модели с использованием вариационного вывода . . . . .	29
2.1. Постановка задачи оптимизации правдоподобия моделей . . . . .	32
2.2. Методы получения вариационной оценки правдоподобия . . . . .	36
2.3. Анализ методов выбора моделей . . . . .	44
Глава 3. Оптимизация гиперпараметров в задаче выбора модели . . . . .	49
3.1. Постановка задачи оптимизации гиперпараметров моделей . . . . .	52
3.2. Градиентные методы оптимизации гиперпараметров . . . . .	56
3.3. Анализ алгоритмов оптимизации гиперпараметров . . . . .	58
Глава 4. Выбор субоптимальной структуры модели . . . . .	65
4.1. Постановка задачи выбора структуры модели . . . . .	65
4.2. Обобщенная постановка задачи . . . . .	70
4.3. Анализ предложенного метода выбора структуры модели . . . . .	72
Глава 5. Анализ прикладных задач порождения и выбора моделей глубокого обучения . . . . .	76
5.1. Выбор модели автокодировщика (Попова) . . . . .	76
5.2. Модели парофраза (Смердов) . . . . .	78
5.3. Прореживание модели (Грабовой) . . . . .	79
Заключение . . . . .	84
Список основных обозначений . . . . .	86
Список иллюстраций . . . . .	87
Список таблиц . . . . .	89
Список литературы . . . . .	90
Список использованных источников . . . . .	90

## Введение

**Актуальность темы.** В работе рассматривается задача автоматического построения моделей глубокого обучения субоптимальной сложности.

Под сложностью модели понимается *минимальная длина описания* [1], т.е. минимальное количество информации, которое требуется для передачи информации о модели и о выборке. Вычисление минимальной длины описания модели является вычислительно сложной процедурой. В работе предлагается получение ее приближенной оценки, основанной на связи минимальной длины описания и *правдоподобия модели* [1]. Для получения оценки правдоподобия используются вариационные методы получения оценки правдоподобия [2], основанные на аппроксимации неизвестного другим заданным распределением. Под субоптимальной сложностью понимается вариационная оценка правдоподобия модели.

Одна из проблем построения моделей глубокого обучения — большое количество параметров моделей [3, 4]. Поэтому задача выбора моделей глубокого обучения включает в себя выбор стратегии построения модели, эффективной по вычислительным ресурсам. В работе [5] приводятся теоретические оценки построения нейросетей с использованием , при которых построение модели производится итеративно последовательным увеличением числа нейронов в сети. В работе [6] предлагается жадная стратегия выбора модели нейросети с использованием релевантных априорных распределений, т.е. параметрических распределений, оптимизация параметров которых позволяет удалить часть параметров из модели. Данный метод был к задаче построения модели метода релевантных векторов [7]. Альтернативой данным алгоритмам построения моделей являются методы, основанные на прореживании сетей глубокого обучения [8, 9, 10], т.е. последовательного удаления параметров, не дающих существенного прироста качества модели. В работах [11, 12] рассматривается послойное построение модели с отдельным критерием оптимизации для каждого слоя. В работах [13, 14, 15] предлагается декомпозиция модели на порождающую и разделяющую, оптимизируемых последовательно. В работе [16] предлагается метод автоматического построения сети, основанный на бустинге. В качестве оптимизируемого функционала предлагается линейная комбинация функции правдоподобия выборки и сложности модели по Радемахеру. В работах [17, 18, 19, 20] предлагается метод автоматического построения сверточной сети с использованием обучения с подкреплением. В [21] используется схожее представление сверточной сети, вместо обучения с подкреплением используется градиентная параметров, задающих структуру нейронной сети.

В качестве порождающих моделей в сетях глубокого обучения выступают ограниченные машины Больцмана [3] и автокодировщики [22]. В работе [23] рассматриваются некоторые типы регуляризации автокодировщиков, позволяющие формально рассматривать данные модели как порождающие модели с использованием байесового вывода. В работе [24] также рассматриваются ре-

гуляризованные автокодировщики и свойства оценок их правдоподобия. В работе [25] предлагается обобщение автокодировщика с использованием вариационного байесовского вывода [2]. В работе [26] рассматриваются модификации вариационного автокодировщика и ступенчатых сетей (англ. ladder network) [27] для случая построения многослойных порождающих моделей.

В качестве критерия выбора модели в ряде работ [28, 2, 29, 30, 31, 32] выступает правдоподобие модели. В работах [29, 30, 31, 32] рассматривается проблема выбора модели и оценки гиперпараметров в задачах регрессии. Альтернативным критерием выбора модели является минимальная длина описания [1], являющаяся показателем статистической сложности модели и заданной выборки. В работе [33] рассматривается перечень критериев сложности моделей глубокого обучения и их взаимосвязь. В работе [34] в качестве критерия сложности модели выступает показатель нелинейности, характеризуемый степенью полинома Чебышева, аппроксимирующего функцию. В работе [35] анализируется показатель избыточности параметров сети. Утверждается, что по небольшому набору параметров в глубокой сети с большим количеством избыточных параметров можно спрогнозировать значения остальных. В работе [36] рассматривается показатель робастности моделей, а также его взаимосвязь с топологией выборки и классами функций, в частности рассматривается влияние функции ошибки и ее липшицевой константы на робастность моделей. Схожие идеи были рассмотрены в работе [37], в которой исследуется устойчивость классификации модели под действием шума.

Одним из методов получения приближенного значения интеграла правдоподобия является вариационный метод получения нижней оценки интеграла [2]. В работе [38] рассматривается стохастическая версия вариационного метода. В работе [39] рассматривается алгоритм получения вариационной нижней оценки правдоподобия для оптимизации гиперпараметров моделей глубокого обучения. В работе [40] рассматривается получение вариационной нижней оценки интеграла с использованием модификации методов Монте-Карло. В работе [41] рассматривается стохастический градиентный спуск в качестве оператора, порождающего распределение, аппроксимирующее апостериорное распределение параметров модели. Схожий подход рассматривается в работе [42], где также рассматривается стохастический градиентный спуск в качестве оператора, порождающего апостериорное распределение параметров. В работе [43] предлагается модификация стохастического градиентного спуска, аппроксимирующая апостериорное распределение.

Альтернативным методом выбора модели является выбор модели на основе скользящего контроля [44, 29]. Проблемой такого подхода является возможная высокая вычислительная сложность [45, 46]. В работах [47, 48] рассматривается проблема смещения оценок качества модели и гиперпараметров, получаемых при использовании  $k$ -fold метода скользящего контроля, при котором выборка делится на  $k$  частей с обучением на  $k - 1$  части и валидацией результата на оставшейся части выборки.

Задачей, связанной с проблемой выбора модели, является задача оптимизации гиперпараметров [28, 2]. В работе [29] рассматривается оптимизация гиперпараметров с использованием метода скользящего контроля и методов оптимизации интеграла правдоподобия моделей, отмечается низкая скорость сходимости гиперпараметров при использовании метода скользящего контроля. В ряде работ [49, 50] рассматриваются градиентные методы оптимизации гиперпараметров, позволяющие оптимизировать большое количество гиперпараметров одновременно. В работе [49] предлагается метод оптимизации гиперпараметров с использованием градиентного спуска с моментом, в качестве оптимизируемого функционала рассматривается ошибка на валидационной части выборки. В работе [51] предлагается метод аппроксимации градиента функции потерь по гиперпараметрам, позволяющий использовать градиентные методы в задаче оптимизации гиперпараметров на больших выборках. В работе [52] предлагается упрощенный метод оптимизации гиперпараметров с градиентным спуском: вместо всей истории обновлений параметров для оптимизации используется только последнее обновление. В работе [42] рассматривается задача оптимизации параметров градиентного спуска с использованием нижней вариационной оценки интеграла правдоподобия.

### **Цели работы.**

1. Исследовать методы построения моделей глубокого обучения оптимальной сложности.
2. Предложить критерии оптимальной и субоптимальной сложности модели глубокого обучения.
3. Предложить метод выбора структуры модели глубокого обучения.
4. Предложить алгоритм построения модели субоптимальной сложности и оптимизации параметров.
5. Разработать алгоритм построения модели и проанализировать различные подходы к решению задачи автоматического построения моделей глубокого обучения и оптимизации параметров модели.

**Методы исследования.** Для достижения поставленных целей используются методы вариационного байесовского вывода [28, 2, 41]. Рассматриваются графовое представление нейронной сети [17, 21]. Для получения вариационных оценок правдоподобия модели используется метод, основанный на градиентном спуске [42, 41]. В качестве метода получения модели субоптимальной сложности используется метод автоматического определения релевантности параметров [28, 53] с использованием градиентных методов оптимизации гиперпараметров [49, 50, 52, 51].

### **Основные положения, выносимые на защиту.**

1. Предложен метод построения модели глубокого обучения субоптимальной сложности.
2. Предложен алгоритм оптимизации параметров, гиперпараметров и структурных параметров моделей глубокого обучения.
3. Проведено исследование свойства оптимизационных алгоритмов выбора модели.
4. Предложен метод выбора модели наиболее правдоподобной структуры, обобщающий различные алгоритмы оптимизации: оптимизация правдоподобия, последовательное увеличение сложности модели, последовательное снижение сложности модели, полный перебор вариантов структуры модели.
5. Предложены методы оптимизации параметров и гиперпараметров модели.
6. Предложен обобщенный метод выбора модели глубокого обучения.
7. Разработан программный комплекс для построения моделей глубокого обучения для задач классификации и регрессии.

**Научная новизна.** Разработан новый подход к построению моделей глубокого обучения. Предложены критерии субоптимальной и оптимальной сложности модели, а также исследована их связь. Предложен метод построения модели глубокого обучения субоптимальной сложности. Предложен метод оптимизации гиперпараметров модели, а также методов оптимизации модели. Предложен обобщенный метод выбора модели глубокого обучения.

**Теоретическая значимость.** В целом, данная диссертационная работа носит теоретический характер. В работе предлагаются критерии субоптимальной и оптимальной сложности, основанные на принципе минимальной длины описания. Исследуется взаимосвязь критериев оптимальной и субоптимальной сложности. Предлагаются градиентные методы для получения оценок сложности модели. Доказывается теорема об оценке энтропии эмпирического распределения параметров модели, полученных под действием оператора оптимизации. Доказывается теорема об обобщенном методе выбора модели глубокого обучения.

**Практическая значимость.** Предложенные в работе методы предназначены для построения моделей глубокого обучения в задачах регрессии и классификации; оптимизации гиперпараметров полученной модели; выборе модели из конечного множества заданных моделей; получения оценок переобучения модели.

**Степень достоверности и апробация работы.** Достоверность результатов подтверждена математическими доказательствами, экспериментальной проверкой полученных методов на реальных задачах выбора моделей глубокого обучения; публикациями результатов исследования в рецензируемых научных

изданиях, в том числе рекомендованных ВАК. Результаты работы докладывались и обсуждались на следующих научных конференциях.

1. “Восстановление панельной матрицы и ранжирующей модели в разнородных шкалах”, Всероссийская конференция «57-я научная конференция МФТИ», 2014.
2. “A monolingual approach to detection of text reuse in Russian-English collection”, Международная конференция «Artificial Intelligence and Natural Language Conference», 2015 [54].
3. “Выбор модели глубокого обучения субоптимальной сложности с использованием вариационной оценки правдоподобия”, Международная конференция «Интеллектуализация обработки информации», 2016 [55].
4. “Machine-Translated Text Detection in a Collection of Russian Scientific Papers”, Международная конференция по компьютерной лингвистике и интеллектуальным технологиям «Диалог-21», 2017 [56].
5. “Author Masking using Sequence-to-Sequence Models”, Международная конференция «Conference and Labs of the Evaluation Forum», 2017.
6. “Градиентные методы оптимизации гиперпараметров моделей глубокого обучения”, Всероссийская конференция «Математические методы распознавания образов ММРО», 2017 [57].
7. “Детектирование переводных заимствований в текстах научных статей из журналов, входящих в РИНЦ”, Всероссийская конференция «Математические методы распознавания образов ММРО», 2017 [58].
8. “ParaPlagDet: The system of paraphrased plagiarism detection”, Международная конференция «Big Scholar at conference on knowledge discovery and data mining», 2018.
9. “Байесовский выбор наиболее правдоподобной структуры модели глубокого обучения”, Международная конференция «Интеллектуализация обработки информации», 2018 [59].
10. “Variational learning across domains with triplet information”, TODO, «Conference on Neural Information Processing Systems», 2018.

Работа поддержана грантами Российского фонда фундаментальных исследований.

1. 19-07-00875, Развитие методов автоматического построения и выбора вероятностных моделей субоптимальной сложности в задачах глубокого обучения.
2. 16-37-00488, Разработка алгоритмов построения сетей глубокого обучения как суперпозиций универсальных моделей.
3. 16-07-01158, Развитие теории построения суперпозиций универсальных моделей классификации сигналов.
4. 14-07-3104, Построение и анализ моделей классификации для выборок малой мощности.

**Публикации по теме диссертации.** Основные результаты по теме диссертации изложены в ? печатных изданиях, ? из которых изданы в журналах, рекомендованных ВАК.

1. Бахтеев О.Ю., Попова М.С., Стрижов В.В. Системы и средства глубокого обучения в задачах классификации. // Системы и средства информатики. 2016. № 26.2. С. 4-22 [9].
2. Бахтеев О.Ю., Стрижов В.В. Выбор моделей глубокого обучения субоптимальной сложности. // Автоматика и телемеханика. 2018. №8. С. 129-147 [60].
3. Огальцов А.В., Бахтеев О.Ю. Автоматическое извлечение метаданных из научных PDF-документов. // Информатика и её применения. 2018.
4. Смердов А.Н., Бахтеев О.Ю., Стрижов В.В. Выбор оптимальной модели рекуррентной сети в задачах поиска парофраза. // Информатика и ее применения. 2019.
5. Грабовой А.В., Бахтеев О.Ю., Стрижов В.В. Определение релевантности параметров нейросети. // Информатика и её применения. 2019.
6. Бахтеев О.Ю. Восстановление панельной матрицы и ранжирующей модели по метризованной выборке в разнородных данных. // Машинное обучение и анализ данных. 2016. № 7. С. 72-77 [61].
7. Бахтеев О.Ю. Восстановление пропущенных значений в разнородных шкалах с большим числом пропусков. // Машинное обучение и анализ данных. 2015. № 11. С. 1-11 [62].

**Личный вклад.** Все приведенные результаты, кроме отдельно оговоренных случаев, получены диссидентом лично при научном руководстве д.ф.-м.н. В. В. Стрижова.

**Структура и объем работы.** Диссертация состоит из оглавления, введения, четырех разделов, заключения, списка иллюстраций, списка таблиц, перечня основных обозначений и списка литературы из 138 наименований. Основной текст занимает 100 страниц.

**Краткое содержание работы по главам.** В первой главе вводятся основные понятия и определения, формулируются задачи построения моделей глубокого обучения. Рассматриваются основные критерии выбора моделей. Рассматриваются существующие алгоритмы построения моделей глубокого обучения.

Во второй главе предлагается алгоритм построения субоптимальной модели глубокого обучения. Предлагаются методы оценки сложности модели.

В третьей главе рассматриваются методы оптимизации гиперпараметров модели.

В четвертой главе рассматривается обобщенный метод выбора модели глубокого обучения.

В пятой главе на базе предложенных методов описывается разработанный программный комплекс, позволяющий автоматически построить модель глубокого обучения субпотимальной сложности для заданной выборки для задачи классификации и регрессии. Работа данного комплекса анализируется на ряде выборок для задач классификации и регрессии. Результаты, полученные с помощью предложенных методов, сравниваются с результатами известных алгоритмов.

# Глава 1

## Постановка задачи последовательного выбора моделей

Проблема выбора структуры модели является фундаментальной в области машинного обучения интеллектуального анализа данных. Проблема выбора структуры модели глубокого обучения формулируется следующим образом: решается задача классификации или регрессии на заданной или пополняемой выборке  $\mathcal{D}$ . Требуется выбрать структуру нейронной сети, доставляющей минимум ошибки на этой функции и максимум качества на некотором внешнем критерии. Под моделью глубокого обучения понимается суперпозиция дифференцируемых по параметрам нелинейный функций. Под структурой модели понимается значения структурных параметров модели, т.е. величин, задающих вид итоговой суперпозиции.

Формализуем описанную выше задачу.

**Определение 1.** *Объектом* назовем пару  $(\mathbf{x}, y)$ ,  $\mathbf{x} \in \mathbb{X} = \mathbb{R}^n$ ,  $y \in \mathbb{Y}$ . В случае задачи классификации  $\mathbb{Y}$  является распределением вероятностей принадлежности объекта  $\mathbf{x} \in \mathbb{X}$  множеству классов  $\{1, \dots, Z\}$ :  $\mathbb{Y} \subset [0, 1]^Z$ , где  $Z$  – число классов. В случае задачи регрессии  $\mathbb{Y}$  является некоторым подмножеством вещественных чисел  $y \in \mathbb{Y} \subseteq \mathbb{R}$ . Объект состоит из двух частей:  $\mathbf{x}$  соответствует *признаковому описанию объекта*,  $y$  – *метке объекта*.

Задана простая выборка

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}, i = 1, \dots, m, \quad (1.1)$$

состоящая из множества объектов

$$\mathbf{x}_i \in \mathbf{X} \subset \mathbb{X}, \quad y_i \in \mathbf{y} \subset \mathbb{Y}.$$

**Определение 2.** *Моделью*  $\mathbf{f}(\mathbf{w}, \mathbf{x})$  назовем дифференцируемую по параметрам  $\mathbf{w}$  функцию из множества признаковых описаний объекта во множество меток:

$$\mathbf{f} : \mathbb{X} \times \mathbb{W} \rightarrow \mathbb{Y},$$

где  $\mathbb{W}$  – пространство параметров функции  $\mathbf{f}$ .

Специфика задачи выбора модели *глубокого обучения* заключается в том, что модели глубокого обучения могут иметь значительное число параметров, что приводит к неприменимости ряда методов оптимизации и выбора модели. Перейдем к формальному описанию семейства моделей глубокого обучения.

**Определение 3.** Пусть задан направленный граф  $(V, E)$ . Пусть для каждого ребра  $(j, k) \in E$  определен вектор базовых функций мощности  $K^{j,k}$ :  $\mathbf{g}^{j,k} = [\mathbf{g}_0^{j,k}, \dots, \mathbf{g}_{K^{j,k}}^{j,k}]$ . Пусть также для каждой вершины  $v \in V$  определена функция агрегации  $\mathbf{agg}_v$ . Граф  $(V, E)$  в совокупности со множеством векторов базовых функций  $\{\mathbf{g}^{j,k}, (j, k) \in E\}$  и множеством функций агрегаций

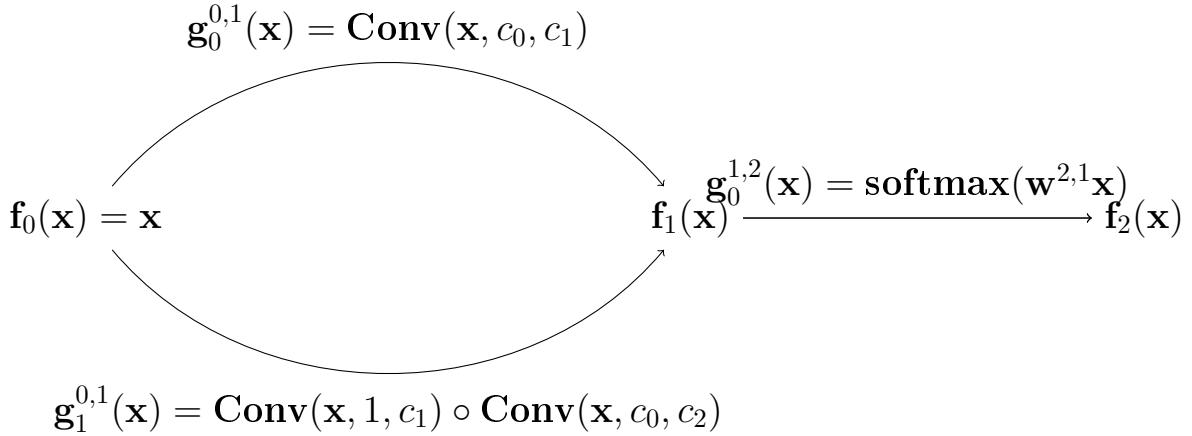


Рис. 1.1. Пример семейства моделей глубокого обучения: семейство описывает сверточную нейронную сеть.

$\{\mathbf{agg}_v, v \in V\}$  называется *семейством моделей*  $\mathfrak{F}$ , если функция, задаваемая как

$$\mathbf{f}_k(\mathbf{x}) = \mathbf{agg}_k (\{\langle \boldsymbol{\gamma}^{j,k}, \mathbf{g}^{j,k} \rangle (\mathbf{f}_j(\mathbf{x})) | j \in \text{Adj}(v_k)\}), \quad \mathbf{f}_0(\mathbf{x}) = \mathbf{x} \quad (1.2)$$

является моделью при любых значениях векторов,  $\boldsymbol{\gamma}^{j,k} \in [0, 1]^{K^{j,k}}$ .

Примером функций агрегации выступают функции суммы и конкатенации векторов.

**Определение 4.** Функции  $\mathbf{f}_1, \dots, \mathbf{f}_{|V|}$  из (1.2) назовем называть *слоями или подмоделями* модели  $\mathbf{f}$ .

Пример семейства моделей, которое описывает сверточную нейронную сеть, представлена на Рис. 1.1. Семейство задает множество моделей с двумя операциями свертки с одинаковым размером фильтра  $c_0$  и различным числом каналов  $c_1$  и  $c_2$ . Единичная свертка с  $c_1$  каналами  $\mathbf{Conv}(\mathbf{x}, c_1, 1)$  требуется для выравнивания размерностей скрытых слоев. Каждая модель семейства задается формулой:

$$\mathbf{f} = \mathbf{agg}_2 \left( \left\{ \boldsymbol{\gamma}_0^{1,2} \mathbf{g}_0^{1,2} \left( \mathbf{agg}_1 \left( \left\{ \boldsymbol{\gamma}_0^{0,1} \mathbf{g}_0^{0,1}(\mathbf{x}), \boldsymbol{\gamma}_1^{0,1} \mathbf{g}_1^{0,1}(\mathbf{x}) \right\} \right) \right) \right\} \right).$$

Положим, что функции агрегации  $\mathbf{agg}_1, \mathbf{agg}_2$  являются операциями суммы. Заметим, что к вершине “2” ведет только одно ребро, поэтому операцию суммы можно опустить. Итоговая формула модели задается следующим образом:

$$\mathbf{f} = \boldsymbol{\gamma}_0^{1,2} \mathbf{softmax} \left( \boldsymbol{\gamma}_0^{0,1} \mathbf{Conv}(\mathbf{x}, c_0, c_1)(\mathbf{x}) + \boldsymbol{\gamma}_1^{0,1} \mathbf{Conv}(\mathbf{x}, 1, c_1) \circ \mathbf{Conv}(\mathbf{x}, c_0, c_2)(\mathbf{x}) \right).$$

**Определение 5.** *Параметрами* модели  $\mathbf{f}$  из семейства моделей  $\mathfrak{F}$  назовем конкатенацию векторов параметров всех базовых функций  $\{\mathbf{g}^{j,k} | (j, k) \in E\}$ ,  $\mathbf{w} \in \mathbb{W}$ . Вектор параметров базовой функции  $\mathbf{g}_l^{j,k}$  будем обозначать как  $\mathbf{w}_l^{j,k}$ .

**Определение 6.** Структурой  $\Gamma$  модели  $\mathbf{f}$  из семейства моделей  $\mathfrak{F}$  назовем конкатенацию векторов  $\gamma^{j,k}$ . Множество всех возможных значений структуры  $\Gamma$  будем обозначать как  $\Gamma$ . Векторы  $\gamma^{j,k}, (j, k) \in E$  назовем *структурными параметрами модели*.

**Определение 7.** *Параметризацией* множества моделей  $M$  назовем семейство моделей  $\mathfrak{F}$ , такое что для каждой модели  $\mathbf{f} \in M$  существуют значение структуры модели  $\Gamma$  при котором функция  $\mathbf{f}$  совпадает с функцией (1.2).

**TODO** Можно доказать, что для любого множества хороших (дифференцируемых?) моделей существует параметризация.

Рассмотрим варианты ограничений, которые накладываются на структурные параметры  $\gamma_{j,k}$  семейства моделей. Цель данных ограничений — уточнение архитектуры модели глубокого обучения, которую требуется получить.

1. Структурные параметры лежат на вершинах булевого куба:  $\gamma_{j,k} \in \{0, 1\}^{K^{j,k}}$ . Структурные параметры  $\gamma_{j,k}$  интерпретируются как параметр включения или выключения компонент вектора базовых функций  $\mathbf{g}^{j,k}$  в итоговую модель.
2. Структурные параметры лежат внутри булевого куба:  $\gamma \in [0, 1]^{K^{j,k}}$ . Релаксированная версия предыдущих ограничений, позволяющая проводить градиентную оптимизацию для структурных параметров.
3. Структурные параметры лежат на вершинах симплекса:  $\gamma_{j,k} \in \bar{\Delta}^{K^{j,k}-1}$ . Каждый вектор структурных параметров  $\gamma_{j,k}$  имеет только одну ненулевую компоненту, определяющую какая из базовых функций  $\mathbf{g}^{j,k}$  войдет в итоговую модель. Примером семейства моделей, требующим такое ограничение является семейство полно связанных нейронных сетей с одним скрытым слоем и двумя значениями количества нейронов на скрытом слое. Схема семейства представлена на Рис. 1.5. Данное семейство можно представить как семейство с двумя базовыми функциями вида  $\mathbf{g} = \sigma(\mathbf{w}\mathbf{x})$ , где матрицы параметров каждой из функций  $\mathbf{g}^{1,1}, \mathbf{g}^{1,2}$  имеют фиксированное число нулевых столбцов. Количество этих столбцов определяет размерность итогового скрытого пространства (невырожденного?) или числа нейронов на скрытом слое.
4. Структурные параметры лежат внутри симплекса:  $\gamma_{j,k} \in \Delta^{K^{j,k}-1}$ . Релаксированная версия предыдущих ограничений, позволяющая проводить градиентную оптимизацию для структурных параметров. Значений структурных параметров  $\gamma_{j,k}$  интерпретируются как вклад каждой компоненты вектора базовых функций  $\mathbf{g}^{j,k}$  в итоговую модель.

Пример, иллюстрирующий представленные выше ограничения, изображен на Рис. 1.2. В данной работе рассматривается случай, когда на структурные параметры наложено ограничение 4. Данные ограничения позволяют решать задачу выбора модели как для семейства моделей типа многослойных полно связанных нейронных сетей, так и для более сложных семейств [21].

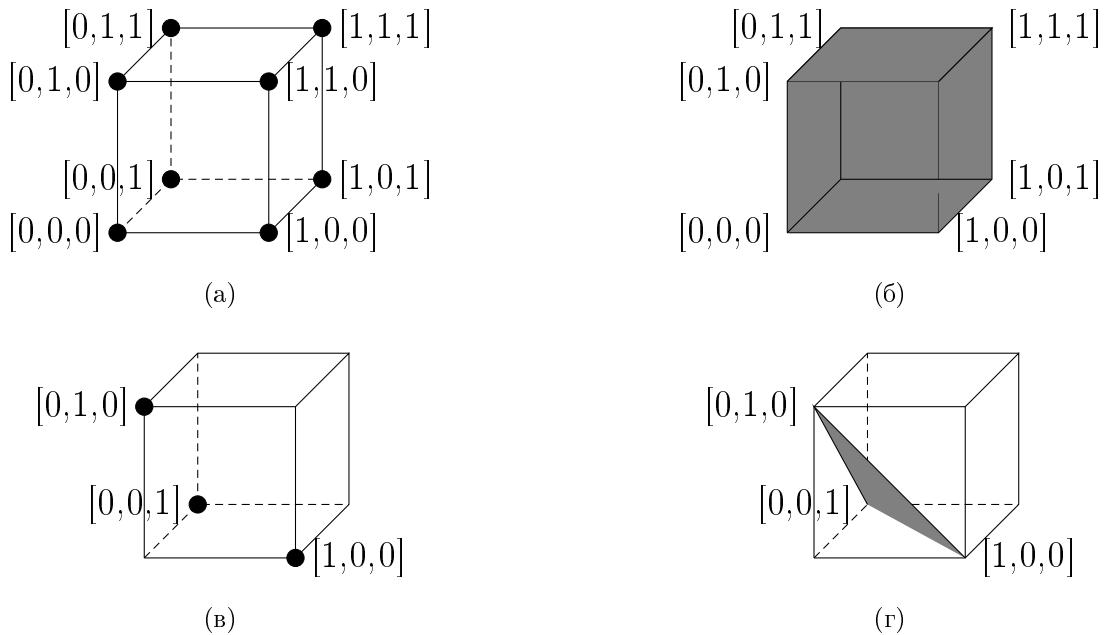


Рис. 1.2. Примеры ограничений для одного структурного параметра  $\gamma$ ,  $|\gamma| = 3$ .  
 а) структурный параметр лежит на вершинах куба, б) структурный параметр лежит внутри куба, в) структурный параметр лежит на вершинах симплекса, г) структурный параметр лежит внутри симплекса.

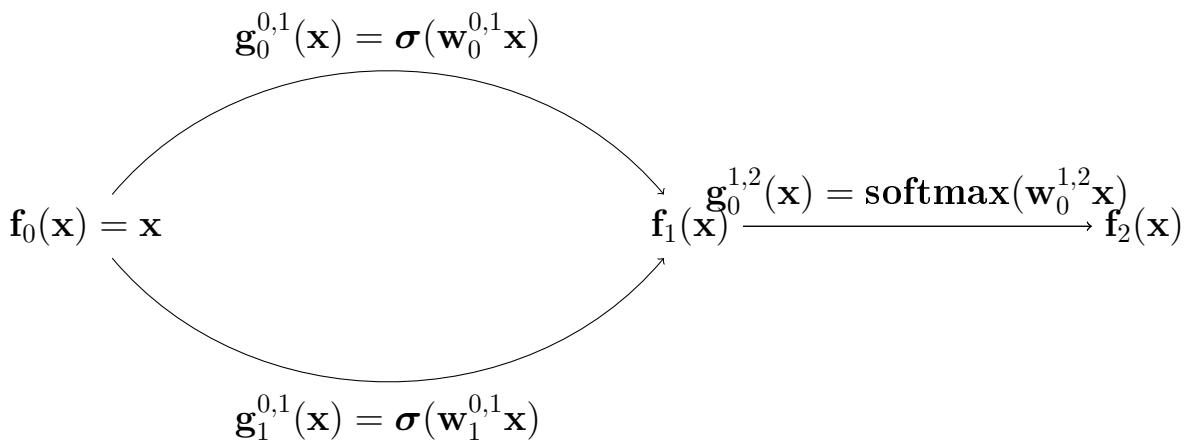


Рис. 1.3. Пример семейства моделей глубокого обучения: семейство описывает многослойную полно связную нейронную сеть с одним скрытым слоем и нелинейной функцией активации  $\sigma$ .

Для дальнейшей постановки задачи введем понятие вероятностной модели, и связанных с ним определений. Будем полагать, что для параметров модели  $\mathbf{w}$  и структуры  $\Gamma$  задано распределение  $p(\mathbf{w}, \Gamma | \mathbf{h})$ , соответствующее предположениям о распределении структуры и параметров.

**Определение 8.** Гиперпараметрами  $\mathbf{h} \in \mathbb{H}$  модели назовем параметры распределения  $p(\mathbf{w}, \Gamma | \mathbf{h})$ .

**Определение 9.** Априорным распределением параметров и структуры модели назовем вероятностное распределение, соответствующее предположениям о распределении параметров модели:

$$p(\mathbf{w}, \Gamma | \mathbf{h}) : \mathbb{W} \times \Delta\Gamma \times \mathbb{H} \rightarrow \mathbb{R}^+,$$

где  $\mathbb{W}$  — множество значений параметров модели.

Одной из возможных частных постановок задачи выбора структуры модели является *двуязычный байесовский вывод*. На первом уровне байесовского вывода находится апостериорное распределение параметров.

**Определение 10.** Апостериорным распределением назовем распределение вида

$$p(\mathbf{w}, \Gamma | \mathbf{y}, \mathbf{X}, \mathbf{h}) = \frac{p(\mathbf{y} | \mathbf{w}, \Gamma, \mathbf{X}, \mathbf{h}) p(\mathbf{w}, \Gamma | \mathbf{h})}{p(\mathbf{y} | \mathbf{X})} \propto p(\mathbf{y} | \mathbf{w}, \Gamma, \mathbf{X}, \mathbf{h}) p(\mathbf{w}, \Gamma | \mathbf{h}). \quad (1.3)$$

**Определение 11.** Вероятностной моделью глубокого обучения назовем совместное распределение вида

$$p(y, \mathbf{w}, \Gamma | \mathbf{x}, \mathbf{h}) = p(y | \mathbf{x}, \mathbf{w}, \Gamma) p(\mathbf{w}, \Gamma | \mathbf{h}) : \mathbb{Y} \times \mathbb{W} \times \Delta\Gamma \times \mathbb{R}^+.$$

**Определение 12.** Функцией правдоподобия выборки назовем величину

$$p(y | \mathbf{X}, \mathbf{w}, \Gamma) : \mathbb{Y} \times \mathbb{X} \times \mathbb{W} \times \Delta\Gamma \rightarrow \mathbb{R}^+.$$

Для каждой модели определена функция правдоподобия  $p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \Gamma)$ .

На втором уровне байесовского вывода осуществляется выбор модели на основе правдоподобия модели.

**Определение 13.** Правдоподобием модели назовем величину

$$p(y | \mathbf{X}, \mathbf{h}) = \int_{\mathbf{w}, \Gamma} p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \Gamma) p(\mathbf{w}, \Gamma | \mathbf{h}) d\mathbf{w} d\Gamma. \quad (1.4)$$

Получение значений апостериорного распределения и правдоподобия модели сетей глубокого обучения является вычислительно сложной процедурой. Для получения оценок на данные величины используют методы, такие как аппроксимация Лапласа [29] и вариационная нижняя оценка [39].

**Определение 14.** Апроксимирующим распределением назовем параметрическое распределение  $q(\mathbf{w}, \Gamma)$ , являющееся приближением апостериорного распределения параметров и структуры  $p(\mathbf{w}, \Gamma | \mathbf{X}, \mathbf{y}, \mathbf{h})$ .

**Определение 15.** *параметрами* модели  $\boldsymbol{\theta} \in \mathbb{R}^u$  назовем параметры аппроксимирующего распределения  $q$ .

**Определение 16.** Пусть задано аппроксимирующее распределения  $q$ . *Функцией потерь*  $L(\boldsymbol{\theta}, \mathbf{h}, \mathbf{X}, \mathbf{y})$  для модели  $\mathbf{f}$  назовем дифференцируемую функцию, принимаемую за качество модели на обучающей выборки при параметрах модели, получаемых из распределения  $q$ .

В качестве функции  $L$  может выступать логарифм правдоподобия выборки  $\log p(y|\mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma})$  и логарифм апостериорной вероятности  $\log p(\mathbf{w}, \boldsymbol{\Gamma}|\mathbf{y}, \mathbf{X}, \mathbf{h})$  параметров и структуры модели на обучающей выборке.

**Определение 17.** Пусть задано аппроксимирующее распределения  $q$  и функция потерь  $L$ . *Функцией валидации*  $Q(\mathbf{h}, \boldsymbol{\theta}, \mathbf{X}, \mathbf{y})$  для модели  $\mathbf{f}$  назовем дифференцируемую функцию, принимаемую за качество модели при векторе  $\boldsymbol{\theta}$ , заданном неявно.

В данной работе задача выбора структуры модели и параметров модели ставится как двухуровневая задача оптимизации:

$$\mathbf{h}^* = \arg \max_{\mathbf{h} \in \mathbb{H}} Q(\mathbf{h}, \boldsymbol{\theta}^*, \mathbf{X}, \mathbf{y}), \quad (1.5)$$

где  $\boldsymbol{\theta}^*$  — решение задачи оптимизации

$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta} \in \mathbb{R}^u} L(\boldsymbol{\theta}, \mathbf{h}, \mathbf{X}, \mathbf{y}). \quad (1.6)$$

**Определение 18.** *Выбором модели*  $\mathbf{f}$  назовем решение двухуровневой задачи оптимизации (1.5).

Рассмотрим для примера базовый вариант выбора модели с применением функций  $q, L, Q$ . Будем полагать, что задано разбиение выборки на обучающую  $\mathfrak{D}_{\text{train}}$  и валидационную  $\mathfrak{D}_{\text{valid}}$  части. Положим в качестве оптимизируемых параметров  $\boldsymbol{\theta}$  параметры  $\mathbf{w}$  и структуры  $\boldsymbol{\Gamma}$  модели:

$$\boldsymbol{\theta} = [\mathbf{w}, \boldsymbol{\Gamma}].$$

Пусть также задано априорное распределение  $p(\mathbf{w}, \boldsymbol{\Gamma}|\mathbf{h})$ . Положим в качестве функции  $L$  логарифм апостериорной вероятности модели:

$$L = \sum_{\mathbf{x}, y \in \mathfrak{D}_{\text{train}}} \log p(y, \mathbf{w}, \boldsymbol{\Gamma}|\mathbf{x}).$$

Положим в качестве функции  $Q$  правдоподобия выборки при условии параметров  $\mathbf{w}$  и структуры  $\boldsymbol{\Gamma}$ :

$$Q = \sum_{\mathbf{x}, y \in \mathfrak{D}_{\text{valid}}} \log p(y|\mathbf{x}, \mathbf{w}, \boldsymbol{\Gamma}).$$

Оптимизация параметров и структуры производится по обучающей выборке. Гиперпараметры  $\mathbf{h}$  выступают в качестве регуляризатора, чья оптимизация

производится по валидационной выборке. Подобная оптимизация позволяет предотвратить переобучение модели [49].

Частным случаем задачи выбора структуры глубокой сети является выбор обобщенно-линейных моделей. Отдельные слои полносвязанных нейросетей являются обобщено-линейными модели. Задачу выбора обобщено-линейной моделей сводится к задаче выбора признаков, методы решения которой делятся на три группы [63]:

1. Фильтрационные методы. Не используют какой-либо информации о модели, а отсекают признаки только на основе статистических показателей, учитывающих взаимосвязь признаков и меток объектов.
2. Оберточные методы анализируют подмножества признаков. Они выбирают не признаки, а подмножества признаков, что позволяет учесть корреляция признаков.
3. Методы погружения оптимизируют модели и проводят выбор признаков в единой процедуре, являясь комбинацией предыдущих типов отбора признаков.

### 1.1. Оптимизация параметров в задаче выбора структуры модели

Один из подходов к выбору оптимальной модели заключается в итеративном удалении наименее информативных параметров модели. В данном разделе собраны методы оптимизации структуры существующей модели.

**Алгоритмы прореживания параметров модели.** В [8] предлагается удалять неинформативные параметры модели. Для этого находится точка оптимума  $\boldsymbol{\theta}^*$  функции  $L$ , и производится разложение функции  $L$  в ряд Тейлора в окрестности  $\boldsymbol{\theta}^*$ :

$$L(\boldsymbol{\theta}^* + \Delta\boldsymbol{\theta}) - L(\boldsymbol{\theta}^*) = \frac{1}{2}\Delta\boldsymbol{\theta}^\top \mathbf{H}\Delta\boldsymbol{\theta} + o(\|\Delta\boldsymbol{\theta}\|^3), \quad (1.7)$$

где  $\mathbf{H}$  — гессиан функции  $L$ . Связь между параметрами не учитывается, поэтому гессиан матрицы  $L$  является диагональным. Положим в качестве операции удаления параметра замену его значения на ноль. Выбор наиболее неинформативного параметра сводится к задаче условной минимизации (1.7) при условиях вида

$$\theta_i + \Delta\theta_i = 0, \quad \theta_i \in \boldsymbol{\theta}.$$

В результате решения данной задачи минимизации каждому параметру определяется функция выпуклости

$$\text{saliency}(\theta_i) = \frac{\theta_i^2}{2H_{i,i}}.$$

Данная функция характеризует информативность параметра.

В [64] было предложено развитие данного метода. В отличие от [8] не вводится предположений о диагональности гессиана функции ошибок, поэтому удаление неинформативных параметров модели производится точнее. Для получения оценок гессиана и его обратной матрицы применяется итеративный алгоритм.

**Алгоритмы компрессии параметров модели.** В [65, 66, 10] предлагаются методы компрессии параметров сетей глубокого обучения. Основным отличием задачи прореживания от задачи компрессии выступает эксплуатационное требование: если прореживание используется для получения оптимальной и наиболее устойчивой модели, то компрессия производится для уменьшения потребляемых вычислительных ресурсов при сохранении основных эксплуатационных характеристик исходной модели [66]. В [10] предлагается итеративное использование регуляризации типа DropOut [67] для прореживания модели. В [65, 66] используются методы снижения вычислительной точности представления параметров модели на основе кластеризации параметров  $\mathbf{w}$  модели: вместо значений параметров предлагается хранить идентификатор кластера, соответствующего параметру, что существенно снижает количество требуемой памяти. В [66] предлагается метод компрессии, основанный на кластеризации значений параметров модели и представлении их в сжатом виде на основе кодов Хаффмана.

**Байесовские методы прореживания параметров модели.** Байесовский подход к порождению и выбору моделей заключается в использовании вероятностных предположений о распределении параметров и структуры в семействах моделей. Такой подход позволяет учитывать при выборе моделей не только эксплуатационные критерии качества модели, такие как точность итоговой модели и количество параметров в ней, но и некоторые статистические характеристики модели.

В работе [49] рассматривается задача оптимизации гиперпараметров. Авторы предлагают оптимизировать константы  $l_2$ -регуляризации отдельно для каждого параметра модели, проводится параллель с методами автоматического определения релевантности параметров (англ. automatic relevance determination, ARD) [28]. Идея автоматического определения релевантности заключается в выборе оптимальных начений гиперпараметров  $\mathbf{h}$  с дальнейшим удалением неинформативных параметров. Неинформативными параметрами являются те параметры, которые с высокой вероятностью равны нулю относительно априорного или апостериорного распределения.

В работе [39] был предложен метод, основанный на получении вариационной нижней оценки правдоподобия модели. В качестве критерия информативности параметра выступает отношение вероятности нахождения параметра в пределах апостериорного распределения к вероятности равенства параметра нулю:

$$\left| \frac{\mu_j}{\sigma_j} \right|,$$

где  $\mu_j, \sigma_j$  — среднее и дисперсия аппроксимирующего распределения  $q$  для па-

раметра  $w_j$ .

Идея данного метода была развита в [68], где также используются вариационные методы. В отличие от [39], в [68] рассматривается ряд априорных распределений параметров, позволяющих прореживать модели более эффективно:

1. Нормальное распределение с лог-равномерным распределением дисперсии. Для каждого параметра  $w \in \mathbf{w}$  задается группа параметров  $\omega \in \Omega$ , где  $\Omega$  — множество всех групп параметров:

$$p(\mathbf{w}, \mathbf{s}) \propto \prod_{\omega \in \Omega} \frac{1}{|\omega|} \prod_{w \in \omega} \mathcal{N}(w | \mathbf{0}, \boldsymbol{\omega}).$$

2. Априорное распределение задается произведением двух случайных величин  $s_{\text{general}}, s_{jk}$  с половинным распределением Коши  $\mathcal{C}^+$ : одно ответственно за отдельный параметр, другое — за общее распределение параметров:

$$s_{\text{general}} \sim \mathcal{C}^+(0, \lambda), \quad s_{jk} \sim \mathcal{C}^+(0, 1), \quad \hat{w}_{jk} \sim \mathcal{N}(0, 1), \quad w_{jk} \sim \hat{w}_{jk} s_{jk} s_{\text{general}},$$

где  $\lambda \in \mathbf{h}$  — параметр распределения.

## 1.2. Оптимизация гиперпараметров модели

В данном разделе рассматриваются работы, посвященные методам оптимизации гиперпараметров. Методы, используемые для оптимизации гиперпараметров моделей глубокого обучения должны быть эффективными по вычислительным затратам в силу высокой вычислительной сложности оптимизации параметров модели. В [133, 134] рассматривается задача оптимизации гиперпараметров стохастическими методами. В [133] проводится сравнение случного поиска значений гиперпараметров с переборным алгоритмом. В [134] производится сравнение случного поиска и алгоритмов, основанных на вероятностных моделях.

### Градиентные методы оптимизации гиперпараметров.

**Определение 19.** Назовем *оператором оптимизации* алгоритм  $T$  выбора вектора параметров  $\boldsymbol{\theta}'$  по параметрам предыдущего шага  $\boldsymbol{\theta}$ :

$$\boldsymbol{\theta}' = T(\boldsymbol{\theta} | L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\beta}), \quad (1.8)$$

где  $\boldsymbol{\beta}$  — параметры оператора оптимизации или *метапараметры*.

Пример схожего описания оптимизации модели с использованием оператора оптимизации можно найти в [41].

Частным случаем оператора оптимизации является оператор стохастического спуска:

$$T(\boldsymbol{\theta} | L, \mathbf{X}, \mathbf{y}, \mathbf{h}, \boldsymbol{\beta}) = \boldsymbol{\theta} - \beta_{\text{lr}} \nabla L(\boldsymbol{\theta}, \mathbf{h}, \hat{\mathbf{X}}, \hat{\mathbf{y}}),$$

где  $\beta_{\text{lr}}$  — шаг градиентного спуска,  $\hat{\mathbf{y}}, \hat{\mathbf{X}}$  — случайная подвыборка заданной мощности выборки  $\mathfrak{D}$ .

В случае оптимизации гиперпараметров оператор оптимизации применяется не к оптимизируемым параметрам  $\boldsymbol{\theta}$ , а к гиперпараметрам  $\mathbf{h}$ :

$$\mathbf{h} = T(\mathbf{h}|Q, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}^*, \boldsymbol{\beta}), \quad (1.9)$$

где  $\boldsymbol{\theta}^*$  — оптимизируемые параметры, полученные в ходе решения задачи оптимизации (1.6).

В случае, если для решения задачи (1.6) применяется несколько шагов оператора оптимизации (1.8),  $\boldsymbol{\theta}^*$  рассматривается как рекурсивная функция от начального приближения оптимизируемых параметров  $\boldsymbol{\theta}^0$  и вектора гиперпараметров  $\mathbf{h}$ :

$$\boldsymbol{\theta}^* = T \circ \cdots \circ T(\boldsymbol{\theta}^0|L, \mathbf{X}, \mathbf{y}, \mathbf{h}, \boldsymbol{\beta}) = \boldsymbol{\theta}^*(\boldsymbol{\theta}^0, \mathbf{h}). \quad (1.10)$$

Решение задачи оптимизации (1.9) при (1.10) является вычислительно сложным, поэтому применяются методы, аппроксимирующие применение градиентных методов при (1.10).

В [69] рассматривается оптимизация гиперпараметров градиентными методами для квадратичной функции потерь. В [49] в качестве оператора оптимизации гиперпараметров выступает метод градиентного спуска с моментом. Показано, что использование момента значительно снижает количество вычислительных ресурсов, требуемых для проведения оптимизации. В [70] предлагается аппроксимация градиентного метода, использующая предположение о линейности функции (1.10) от начального приближения  $\boldsymbol{\theta}^0$ . В [71] предлагается использовать численные методы для приближенного вычисления оператора оптимизации гиперпараметров. В [52] в качестве аппроксимации (1.10) предлагается рассматривать только последний шаг оптимизации:

$$\boldsymbol{\theta}^* \approx T(\boldsymbol{\theta}^{\eta-1}|L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\beta}),$$

где  $\eta$  — число шагов оптимизации.

**Суррогатный выбор моделей.** Идея суррогатных моделей заключается в аппроксимации модели или семейства моделей вычислительно менее сложной функцией.

В работе [72] предлагается моделировать качество модели  $Q$  (2.3) гауссовым процессом, параметрами которого выступают гиперпараметры исходной модели.

Одна из основных проблем использования гауссового процесса как суррогатной модели — кубическая сложность оптимизации. В работе [73] предлагается использовать случайные подпространства гиперпараметров для ускоренной оптимизации. В работе [74] предлагается комбинация из множества гауссовых моделей и линейной модели, позволяющая модели нелинейные зависимости гиперпараметров, а также существенно сократить сложность оптимизации.

В работе [75] предлагается рассматривать RBF-модель для аппроксимации качества  $Q$  исходной модели, что позволяет ускорить процесс оптимизации суррогатной модели. В [76] рассматривается глубокая нейронная сеть в качестве

суррогатной функции. Вместо интеграла правдоподобия (2.3), который оценивается в случае использования гауссового процесса в качестве суррогата, используется максимум апостериорной вероятности (2.4).

Одним из параметров гауссовых процессов является функция ядра гауссового процесса, полностью определяющая процесс в случае нулевого среднего. В работе [77] предлагается функция ядра, определенная на графах:

$$k(v_1, v_2) = r(d(v_1, v_2)),$$

где  $d$  — геодезическое расстояние между вершинами графа,  $r$  — некоторая вещественная функция,  $v_1, v_2 \in V$ .

В работе [78] рассматривается задача выбора структуры нейросети. Предлагается метод построения ковариационной функции для сравнения разнородных графов, соответствующих разным моделям нейронных сетей. Ковариационная функция основывается на метрике, заданной на некоторых характеристиках  $g(v)$  вершин, возможно не определенных для сравниваемых графов:

$$d_v((V_1, E_1), (V_2, E_2)) = \begin{cases} 0, & v \notin V_1, v \notin V_2, \\ \lambda_1 \sqrt{2} \sqrt{1 - \cos(\pi \lambda_2 \frac{g_1 - g_2}{\sup(g) - \inf(g)}),} & v \in V_1, v \in V_2, \\ \lambda_1 \text{ иначе}, & \end{cases}$$

где  $\lambda_1, \lambda_2$  — параметры функции  $d_v$ .

### 1.3. Порождение и выбор структуры модели глубокого обучения

В данном разделе рассматриваются работы, посвященные порождению и модификации структуры моделей. В отличие от работ, описанных в предыдущих разделах, в следующих работах рассматриваемым объектом является не отдельный параметр, а подмодель или группа параметров, входящая в эту подмодель.

**Графовое представление структуры модели.** Одним из возможных представлений структуры моделей глубокого обучения является графовое представление, в котором в качестве ребер графа выступают нелинейные функции, а в качестве вершин графа — представление выборки под действием соответствующих нелинейных функций. Данный подход к описанию модели является соответствует походам, описанным в [79], а также в библиотеках типа TensorFlow [80], Theano [81], Pytorch [82], в которых модель рассматривается как граф, ребрами которого выступают математические операции, а вершинами — результат их действия на выборку. В то же время, существуют и другие способы представления модели. В ряде работ, посвященных байесовской оптимизации [76, 75, 72], модель рассматривается как черный ящик, над которым производится ограниченный набор операций типа “произвести оптимизацию параметров” и “предсказать значение зависимой переменной по независимой пе-

ременной и параметрам модели". Подход, описанный в данных работах, также коррелирует с библиотеками машинного обучения, такими как Weka [83], RapidMiner [84] или sklearn [85], в которых модель машинного обучения рассматривается как черный ящик.

В [86] представлен обзор по графовому описанию моделей глубокого обучения, предлагается метод формального описания графовых сетей (англ. Graph Network), являющийся обобщением предложенных ранее графовых описаний моделей.

В работе [87] рассматриваются подходы к порождению моделей глубокого обучения. Предлагается формализация пространства поиска и формальное описание элементов пространства моделей. Приведем пример описания семейства моделей, соответствующего схеме из Рис. 1.1 при условии, что структурные параметры  $\gamma$  имеют только одну ненулевую компоненту:

(Concat

```
OR(
  (Conv2D [c0] [c1] [1] ,
  (Concat(
    (Conv2D [c0] [c2] [1] ,
    (Conv2D [1] [c1] [1])) ,
  (Affine [10]) ,
  (SoftMax)) .
```

TODO: в статье нет сотфмакса

**Прогнозирование графовых структур.** В работе [88] предлагается метод прогнозирования графовой структуры на основе линейного программирования. Предлагается свести проблему поиска графовой структуры к комбинаторной проблеме. В работе [89] предлагается метод прогнозирования структур деревьев, основанный на дважды-рекуррентных нейросетях (англ. doubly-reccurent), т.е. на сетях, отдельно прогнозирующих глубину и ширину уровней деревьев.

**Стochasticное порождение структур.** Одним из возможных направлений для порождения структур моделей глубокого обучения выступает стохастическое порождение структур. Данный тип порождения предполагает, что структуры порождаются случайно в соответствие оптимизируемым распределением, заданным на структурах  $q(\Gamma)$ . Затем выбирается одна, либо несколько наилучших структур с учетом валидационной функции  $Q$  или внешних, возможно недифференцируемых, критериев качества. Итоговая модель получается путем оптимизации параметров модели при выбранной структуре  $\Gamma$ . Заметим, что в ряде работ, одновременно порождается не только структура модели, но и итоговые параметры.

В работе [90] рассматривается порождение моделей, оптимизируемых без учителя. Модель представляется многослойным перцептроном вида:

$$\mathbf{f} = \mathbf{f}_{|V|} \circ \dots \circ \mathbf{f}_1(\mathbf{x}), \quad \mathbf{f}_i(\mathbf{x}) = \sigma(\mathbf{w}^i \odot \mathbf{H}^i \mathbf{x}),$$

где  $\mathbf{H}^i$  — бинарные матрицы, определяющие вклад каждого параметра из  $\mathbf{w}^i$  в

итоговую модель, знаком  $\odot$  обозначается покомпонентное перемножение.

Порождение моделей производится с использованием композиции процессов индийский буфетов. Процесс индийского буфет заключается в итеративном построении матрицы  $\mathbf{H}^i$  с ограниченным, но не заданным наперед количеством столбцов. Интерпретируя количество столбцов матрицы как размер  $i$ -го слоя предлагается метод, позволяющий выбирать стохастически порождать модели с различной размерностью скрытых слоев.

В работе [91] предлагается метод выбора модели сверточной нейронной сети. Используется функция потерь, основанная на аппроксимации априорного распределения процесса индийского буфета для каждой базовой функции  $\mathbf{g}_i$ , являющейся  $i$ -м отображением объектов:

$$L = \sum_{\mathbf{x}} \left\| \mathbf{x} - \sum_{j=1}^K \mathbf{x} - \sum_{j=1}^K \mathbf{w}^j * \mathbf{g}_i(\mathbf{x}) \right\|_2^2 + \lambda^2 K,$$

где  $K$  — параметр, отвечающий за количество сверток,  $\lambda$  — параметр алгоритма, знаком  $*$  обозначается операция свертки (TODO: проверить).

В работе [92] предлагается ввести априорное распределение Бернулли на структурные параметры  $\boldsymbol{\gamma}^i$ .

В [93] рассматривается задача выбора архитектуры с помощью большого количества параллельных запусков обучения моделей. Предлагаются критерии ранней остановки процедуры оптимизации обучения моделей.

**Последовательный выбор структуры модели.** В работах [94, 16] предлагается наращивание моделей, основанное на бустинге. Рассматривается задача построения нейросетевых моделей специального типа:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}_{|V|} \circ \mathbf{f}_{|V|-1} \circ \dots \mathbf{f}_0(\mathbf{x}), \quad \mathbf{f}_{i+1}(\mathbf{x}) = \boldsymbol{\sigma}(\mathbf{f}_i(\mathbf{x})) + \mathbf{f}_i(\mathbf{x}),$$

приводится параметризация модели, позволяющая рассматривать декомпозицию модели на слабые классификаторы. В [16] рассматривается задача выбора полносвязной нейронной сети для задачи бинарной классификации,  $Z = 2$ . На каждом шаге построения выбирается одно из двух расширений модели, каждое из которых рассматривается как слабый классификатор: сделать модель шире или сделать модель глубже. Пример работы AdaNet представлен на Рис. 1.4. Построение модели заканчивается при условии снижении радемахеровской сложности:

$$\mathfrak{R} = \frac{1}{m} \mathsf{E}_{b_1, \dots, b_{|V|}} \sup_{\mathbf{w}} \sum_{i=1}^m b_i \arg \max_{c=\{0,1\}} f^c(\mathbf{x}_i, \mathbf{w}),$$

где  $b_i$  — реализация случайной дискретной величины, равновероятно принимающей значений  $-1$  и  $1$ ,  $f^c$  —  $c$ -я компонента модели  $\mathbf{f}$ .

В работе [95] рассматривается задача порождения сверточных нейронных сетей. Предлагается проводить последовательный выбор структуры модели по

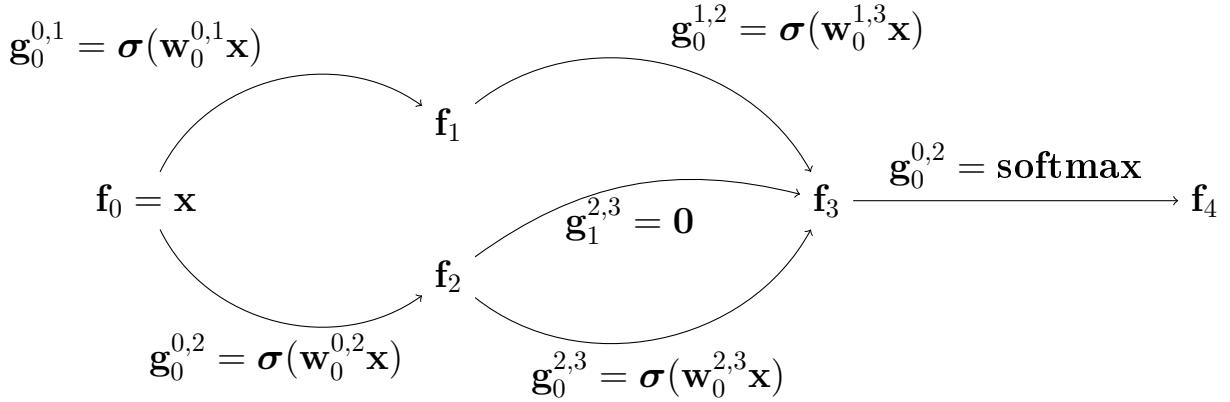


Рис. 1.4. Пример итерации алгоритма AdaNet [16]. Рассматриваются две альтернативные модели: модель с углублением сети (соответствует занулению функции  $\mathbf{f}_2$  с использованием базовой функции  $\mathbf{g}_1^{2,3}$ ) и модель с расширением сети (соответствует базовой функции  $\mathbf{g}_0^{2,3}$ ).

В качестве функции агрегации для подмодели  $\mathbf{f}_3$  выступает конкатенация:  $\text{agg}_3 = \text{concat}$ .

восходящему числу параметров: начиная от сетей с одной подмоделью и итеративно увеличивая количество подмоделей. В силу высокой вычислительной сложности данного подхода, вместо последовательного порождения моделей, предлагается провести оптимизацию рекуррентной нейронной сети, которая предсказывает качество модели по заданным подмоделям, и на основе данного предсказания выбрать наилучшую модель.

В работе [96] предлагается метод анализа структуры сети на основе линейных классификаторов, построенных на промежуточных слоях нейросети. Схожий метод был предложен в [97], где классификаторы на промежуточных уровнях используются для уменьшения вычислений при выполнении вывода и предсказаний. Промежуточные классификаторы работают как решающий список.

В работе [98] предлагается инкрементальный метод оптимизации нейросети. На первом этапе модель декомпозируется на несколько подмоделей, при которой модель последовательностью слоев  $\mathbf{f}_1, \dots, \mathbf{f}_{|V|}$ . Проводится последовательная оптимизация моделей вида:

- 1)  $\mathbf{f} = \mathbf{f}_{|V|}(\mathbf{x})$ ;
- 2)  $\mathbf{f} = \mathbf{f}_{|V|-1} \circ \mathbf{f}_{|V|}(\mathbf{x})$ ;
- 3) ...
- 4)  $\mathbf{f} = \mathbf{f}_1 \circ \dots \circ \mathbf{f}_{|V|}(\mathbf{x})$ .

**Оптимизация структуры модели на основе обучения с подкреплением.** В [17] предлагается итеративная схема выбора архитектуры сверточной нейросети с использованием обучения с подкреплением. Распределение структур и параметров  $q(\mathbf{w}, \Gamma)$  задается рекуррентной нейронной сетью, которая определяет значение параметров модели и наличие ребер с ненулевыми

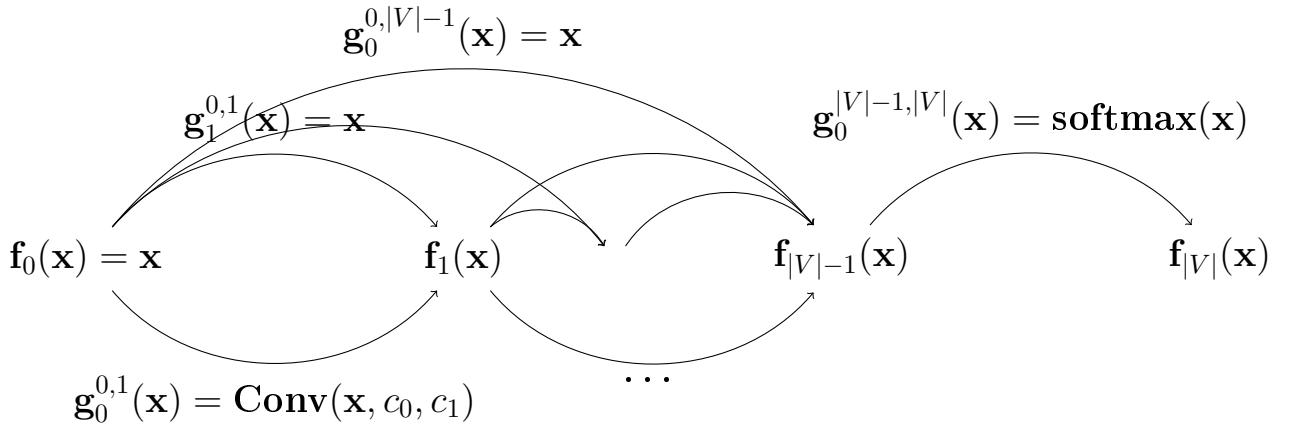


Рис. 1.5. Пример семейства моделей глубокого обучения, описываемый в [17]. Каждая подмодель  $\mathbf{f}_j$  является линейной комбинацией базовых функций: свертки и результата работы предыдущих подмоделей (англ. skip-connection).

операциями между вершинами графов модели. Параметры рекуррентной нейронной сети оптимизируются на основе значения функции  $Q$ , получаемого на каждой итерации алгоритма.

В работе [18] предлагается алгоритм построения регрессионной модели для оценки финального качества модели и ранней остановки оптимизации моделей. Он позволяет существенно ускорить поиск моделей, представленный в [17]. В [20] рассматривается задача переноса архитектуры нейросети, чья структуры была выбрана по выборке, меньшей мощности. Как и в [17] предлагается метод параметризации сверточной нейронной сети в виде графа. Предложенная параметризация позволяет задать более мощное семейство моделей, чем в [17]. Модель представляется в виде последовательности суперпозиций подмоделей, называемых клетками (англ. normal cell и reduction cell). Каждая из этих клеток содержит следующее множество нелинейных операций  $\mathbf{g}$ , состоящее из тождественной операции  $\mathbf{g}(\mathbf{x}) = \mathbf{x}$ , а также множество сверток с фиксированным количеством каналов и размером фильтров и функций субдискретизации или пулинга. Алгоритм выбора структуры модели рекуррентной сетью выглядит следующим образом на шаге  $j$ :

- 1) выбрать вершину  $v'$  из вершин  $v_{j-1}, v_{j-2}$  из данной клетки или вершину из предыдущих клеток;
- 2) выбрать вершину  $v''$  из вершин  $v_{j-1}, v_{j-2}$  из данной клетки или вершину из предыдущих клеток;
- 3) выбрать базовую функцию  $\mathbf{g}'$  для применения к вершине  $v'$ ;
- 4) выбрать базовую функцию  $\mathbf{g}''$  для применения к вершине  $v''$ ;
- 5) выбрать функцию агрегации результатов применения операций  $\mathbf{g}', \mathbf{g}''$ : сумму или конкатенацию.

В отличие от предыдущих работ, в работе [19] предлагается подход к инкрементальному обучению нейросети, основанном на модификации модели, полу-

ченной на предыдущем шаге. Рассматриваются две операции над нейросетью: расширение и углубление сети.

В работах [99, 100, 101] рассматриваются методы деформации нейросетей. В работе [101] предлагается метод оптимального разделения нейросети на несколько независимых сетей для уменьшения количества связей и, как следствие, уменьшения сложности оптимизации модели. В работе [99] предлагается метод сохранения результатов оптимизации нейросети при построении новой более глубокой или широкой нейросети. В работе [100] рассматривается задача расширения сверточной нейросети, нейросеть рассматривается как граф.

В работе [21] используется представление модели из [20]. Вместо обучения с подкреплением используются градиентная оптимизация структуры и параметров, выполненная в единой процедуре.

#### 1.4. Метаоптимизация моделей глубокого обучения

Задача выбора структуры модели тесно связана с раздел машинного обучения под названием *метаобучение* или *метаоптимизация*. Под метаобучением понимаются алгоритмы машинного обучения [102], которые:

- 1) оценивают и сравнивают методы оптимизации моделей;
- 2) оценивают возможные декомпозиции процесса оптимизации моделей;
- 3) на основе полученных оценок предлагают оптимальные стратегии оптимизации моделей и отвергают неоптимальные.

В работе [103] предлагается подход к адаптивному изменению параметров сети. В качестве оператора оптимизации параметров рассматривается величина:

$$T(\boldsymbol{\theta}|L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\beta}) = \boldsymbol{\theta} + \mathbf{f}_{\text{optim}}(\mathbf{f}_{\text{mod}}(\boldsymbol{\theta})),$$

где  $\mathbf{f}_{\text{mod}}$  — функция, определяющая номер параметра из  $\boldsymbol{\theta}$ , подлежащего оптимизации, а  $\mathbf{f}_{\text{optim}}$  — величина изменения параметра. В [103] также предлагается подмодель  $\mathbf{f}_{\text{ana}}$ , определяющая номер параметра, подлежащего дальнейшему анализу. Подход, описанный в данной работе, предполагает оптимизацию и анализ не только самой модели  $\mathbf{f}$ , но и дополнительных моделей  $\mathbf{f}_{\text{mod}}, \mathbf{f}_{\text{ana}}, \mathbf{f}_{\text{optim}}$ .

В работе [104] рассматривается оптимизация метапараметров (шага градиентного спуска  $\beta_{\text{lr}}$  и начального распределения параметров  $\boldsymbol{\theta}^0$ ). Рассматривается задача оптимизации параметров модели в случае, когда количество примеров невелико. Для этого проводится оптимизация оператора оптимизации, который выглядит следующим образом:

$$T(\boldsymbol{\theta}|L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\beta}) = \boldsymbol{\theta}^0 - \boldsymbol{\beta} \nabla L(\boldsymbol{\theta}^0, \mathbf{h}, \mathbf{X}, \mathbf{y}),$$

где векторы  $\boldsymbol{\theta}^0$  и  $\boldsymbol{\beta}$  являются параметрами оператора  $T$ . Задача оптимизации параметров оператора  $T$  рассматривается как задача многозадачного обучения (англ. multitask learning), когда оператор оптимизируется с учетом нескольких

различных выборок и различных функций  $L$ , определенных отдельно для каждой выборки.

В работе [105] рассматривается задача восстановления параметров модели по параметрам другой модели, чьи параметры были получены оптимизацией функции потерь на выборке меньшей мощности. Задачу можно рассматривать как задачу нахождения параметров некоторого оператора оптимизации  $T : \boldsymbol{\theta}^0 \rightarrow \boldsymbol{\theta}$ , где  $\boldsymbol{\theta}^0$  — параметры модели, оптимизированной на небольшой выборке. Предлагается функция оптимизации:

$$T = \arg \min ||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0||_2^2 + \beta_\lambda L(\boldsymbol{\theta}, \mathbf{h}, \hat{\mathbf{X}}, \hat{\mathbf{y}}),$$

где  $\boldsymbol{\theta}$  — параметры модели, обученной по полной выборке  $\mathfrak{D}$ ,  $\hat{\mathfrak{D}}$  — выборка меньшей мощности,  $\beta_\lambda$  — настраиваемый метапараметр.

В работе [106] рассматривается оптимизация метапараметров оператора оптимизации с помощью модели долгой краткосрочной памяти LSTM, которая выступает альтернативе аналитических алгоритмов, таких как Adam [107] или AdaGrad [108]. LSTM имеет небольшое число параметров, т.к. для каждого метапараметра используется свой экземпляр модели LSTM с одинаковыми параметрами для каждого экземпляра. Оптимизируемый функционал является суммой значений функции потерь  $L$  на нескольких шагах оптимизации:

$$Q = \sum_{t=1}^{\eta} L(\boldsymbol{\theta}^t),$$

где  $\eta$  — число шагов оптимизации,  $\boldsymbol{\theta}^t$  — оптимизируемые параметры модели на шаге оптимизации  $t$ .

## 1.5. Выбор структур моделей специального вида

В данном разделе представлены работы по поиску оптимальных моделей со структурами специального вида.

В работе [109] рассматривается оптимизация моделей нейросетей с бинарной функцией активации. Задача оптимизации сводится к задаче mixed integer программирования, которая решается методами выпуклого анализа. В работе [110] предлагается метод построения сети глубокого обучения, структура которой выбирается с использованием обучения без учителя. Критерий оптимальности модели использует оценки энергетических функций и ограниченной машины Больцмана.

В работах [111, 112] рассматривается выбор архитектуры сети с использованием *суперсетей*: связанных между собой подмоделей, образующих граф, каждый путь из нулевой вершины в последнюю которого определяет модель глубокого обучения. Пример графа, описывающего суперсеть представлен на Рис. 1.6. В работе [112] рассматриваются стохастические суперсети, позволяющие выбрать структуру нейросети за ограниченное время оптимизации. Схожий

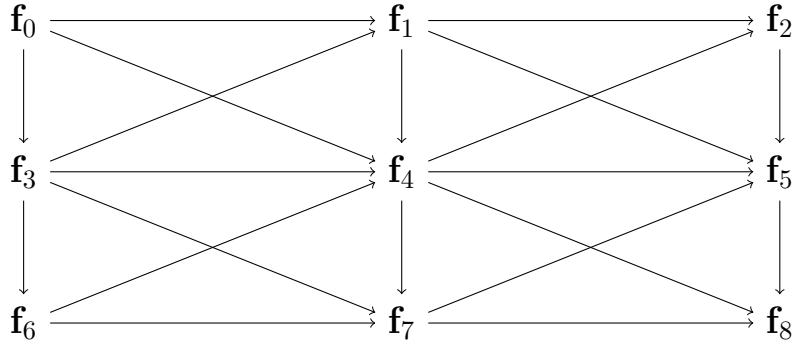


Рис. 1.6. Пример суперсети. Каждый путь из подмодели  $f_0$  в конечную модель  $f_8$  задает модель глубокого обучения.

подход был предложен в работе [111], где предлагается использовать эволюционные алгоритмы для запоминания оптимальных подмоделей и переноса этих моделей в другие задачи.

**Порождающие модели.** Порождающими моделями называются модели, приближающие совместное распределение объектов и соответствующих им меток  $p(\mathbf{X}, \mathbf{y})$ . Частным случаем порождающих моделей являются модели, приближающие только распределение векторов объектов  $\mathbf{X}$ . Подобный случай будем считать частным случаем классификации при пустом множестве меток классов ( $Z = 0$ ).

В ряде работ [113, 114, 115, 116, 117] рассматривается подход к построению порождающих моделей глубокого обучения, при котором каждая подмодель  $f_i$  приближает распределение некоторой случайной величины  $\mathbf{z}_i$ , которая влияет на итоговое распределение  $p(\mathbf{X}, \mathbf{y}) = \int_{\mathbf{z}_1, \dots, \mathbf{z}_{|V|}} p(\mathbf{X}, \mathbf{y} | \mathbf{z}_1, \dots, \mathbf{z}_{|V|}) p(\mathbf{z}_1, \dots, \mathbf{z}_{|V|}) d\mathbf{z}_1 \dots d\mathbf{z}_{|V|}$ . Подобный подход позволяет использовать вероятностную интерпретацию для каждой отдельной подмодели.

В работе [113] рассматривается обобщение вариационного автокодировщика на случай более общих графических моделей. Предлагается проводить оптимизацию сложных графических моделей в единой процедуре. Для вывода предлагается использовать нейронные сети. Другая модификация вариационного автокодировщика представлена в работе [114], авторы рассматривают использование процесса сломанной трости в вариационном автокодировщике, тем самым получая модель со стохастической размерностью скрытой переменной. В [115] рассматривается смесь автокодировщиков, где смесь моделируется процессом Дирихле.

В работе [116] предлагается подход к оптимизации неизвестного распределения с помощью вариационного вывода. Предлагается решать задачу оптимизации итеративно, добавляя в модель новые компоненты вариационного распределения, проводится аналогия с бустингом.

В работе [117] рассматривается задача построения порождающих моделей с

дискретными значениями скрытых переменных  $\mathbf{z}_i$ , предлагается критерий для послойного обучения порождающих моделей:

$$Q = \sum_{\mathbf{x}} \log \sum_i p(\mathbf{x}|\mathbf{z}_i) q(\mathbf{z}) \rightarrow \max,$$

где  $q$  — аппроксимирующее распределение для случайной величины  $\mathbf{z}$ .

В работе [53] рассматривается метод ARD для снижения размерности скрытого пространства вариационных порождающих моделей. Скрытая переменная параметризуется как произведение некоторой случайной величины  $\mathbf{z}$  на вектор, отвечающий за релевантность каждой компоненты скрытой переменной. Схема порождения выборки  $\mathbf{X}$  представлена на Рис. 1.7.

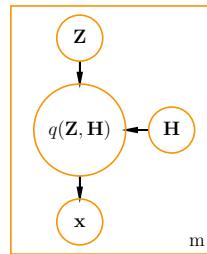


Рис. 1.7. Схема порождения вектора объектов  $\mathbf{X}$ , представленная в [53].

В данной работе предлагается метод последовательного порождения моделей глубокого обучения, основывающийся на применении вариационного вывода. Вариационный вывод позволяет получить оценки правдоподобия модели с небольшими вычислительными затратами, а также проследить потенциальное начало переобучения модели без использования контрольной выборки. Для регуляризации структуры модели предлагается ввести априорное распределение на структуре, позволяющее проводить оптимизацию модели и ее структуры в различных режимах. В качестве метода оптимизации гиперпараметров выступают градиентные методы, что позволяет эффективно производить оптимизацию большого числа гиперпараметров, сопоставимого с числом параметров модели.

## Глава 2

### Выбор модели с использованием вариационного вывода

Рассматривается задача выбора моделей глубокого обучения субоптимальной сложности. Под сложностью модели понимается минимальная длина описания совокупности выборки и модели классификации или регрессии. Под субоптимальной сложностью понимается приближенная оценка минимальной длины описания, полученная с использованием байесовского вывода и вариационных методов. Вводятся вероятностные предположения о распределении параметров. На основе байесовского вывода предлагается функция правдоподобия модели. Для получения оценки правдоподобия применяются вариационные методы с использованием градиентных алгоритмов оптимизации. Проводится вычислительный эксперимент на нескольких выборках.

Проблема выбора модели является одной из ключевых задач машинного обучения. Под моделью понимается суперпозиция функций, решающая задачу классификации или регрессии. В данной работе в качестве критерия выбора модели предлагается субоптимальная сложность модели. Под сложностью модели понимается *минимальная длина описания* [1], т.е. минимальное количество информации, которое требуется передать о модели и о выборке. Вычисление минимальной длины описания модели является вычислительно сложной процедурой. В работе предлагается получение ее приближенной оценки, основанной на связи минимальной длины описания и *правдоподобия модели* [1]. В общем случае данная величина является трудновычислимой. Для получения оценки правдоподобия используются вариационные методы получения оценки правдоподобия [2], основанные на аппроксимации неизвестного другим заданным распределением. Под субоптимальной сложностью понимается вариационная оценка правдоподобия модели.

В работе [33] рассматривается ряд критериев сложности моделей глубокого обучения и их взаимосвязь. В работе [34] в качестве критерия сложности модели выступает показатель нелинейности, характеризуемый степенью полинома Чебышева, аппроксимирующего функцию. В работе [35] анализируется показатель избыточности параметров сети. Утверждается, что по небольшому набору параметров в глубокой сети с большим количеством избыточных параметров возможно спрогнозировать значения остальных. В работе [36] рассматривается показатель робастности моделей, а также его взаимосвязь с топологией выборки и классами функций, в частности рассматривается влияние функции ошибки и ее липшицевой константы на робастность моделей. Схожие идеи были рассмотрены в работе [37], в которой исследуется устойчивость классификации модели под действием шума. В ряде работ [28, 2, 29, 30, 31, 32] в качестве критерия выбора модели выступает правдоподобие модели. В работах [29, 30, 31, 32] рассматривается проблема выбора модели и оценки гиперпараметров в задачах регрессии. Альтернативным критерием выбора модели является минимальная длина описания [1], являющаяся показателем статистической сложности моде-

ли и заданной выборки. В работе [1] рассматриваются различные модификации и интерпретации минимальной длины описания, в том числе связь с правдоподобием модели.

Одним из методов получения приближенного значения правдоподобия модели является вариационный метод получения нижней оценки правдоподобия [2]. В работе [38] рассматривается стохастическая версия вариационного метода. В [39] рассматривается алгоритм получения вариационной нижней оценки правдоподобия для оптимизации гиперпараметров моделей глубокого обучения. В работе [40] рассматривается взаимосвязь градиентных методов получения вариационной нижней оценки интеграла с методом Монте-Карло. В [41] рассматривается стохастический градиентный спуск в качестве оператора, порождающего распределение, аппроксимирующее апостериорное распределение параметров модели. В работе отмечается, что стохастический градиентный спуск не оптимизирует вариационную оценку правдоподобия, а приближает ее только до некоторого числа итераций оптимизации. Схожий подход рассматривается в работе [42], где также рассматривается стохастический градиентный спуск в качестве оператора, порождающего апостериорное распределение параметров. В работе [43] предлагается модификация стохастического градиентного спуска, аппроксимирующая апостериорное распределение.

Альтернативным методом выбора модели является выбор модели на основе скользящего контроля [44, 29]. Проблемой такого подхода является возможная высокая вычислительная сложность [45, 46]. В работах [47, 48] рассматривается проблема смещения оценок качества модели и гиперпараметров, получаемых при использовании  $k$ -fold метода скользящего контроля, при котором выборка делится на  $k$ -частей с обучением на  $k - 1$  части и валидацией результата на оставшейся части выборки.

Задачей, связанной с проблемой выбора модели, является задача оптимизации гиперпараметров [28, 2]. В работе [118] рассматривается оптимизация гиперпараметров с использованием метода скользящего контроля и методов оптимизации интеграла правдоподобия моделей, отмечается низкая скорость сходимости гиперпараметров при использовании метода скользящего контроля. В работах [49, 50] рассматриваются градиентные методы оптимизации гиперпараметров, позволяющие оптимизировать большое количество гиперпараметров одновременно. В работе [49] предлагается метод оптимизации гиперпараметров с использованием градиентного спуска с моментом. В качестве оптимизируемого функционала рассматривается ошибка на валидационной части выборки. В работе [41] отмечается возможность использовать градиентный метод для оптимизации гиперпараметров с использованием вариационной нижней оценки интеграла правдоподобия в качестве оптимизируемого функционала.

Одна из проблем построения моделей глубокого обучения — большое число параметров модели [3], которое достигает нескольких миллионов, а оптимизация модели достигает десятков дней [119]. Задача выбора модели глубокого обучения включает в себя выбор стратегии построения модели, эффективной

по вычислительным ресурсам. В работе [42] рассматривается задача оптимизации параметров градиентного спуска с использованием нижней вариационной оценки интеграла правдоподобия.

В работе [5] приводятся теоретические оценки построения нейросетей с использованием жадных стратегий, при которых построение модели производится итеративно последовательным увеличением числа нейронов в сети. В работе [6] предлагается жадная стратегия выбора модели нейросети с использованием релевантных априорных распределений, т.е. параметрических распределений, оптимизация параметров которых позволяет удалить часть параметров из модели. Данный метод был к задаче построения модели метода релевантных векторов [7]. Альтернативой данным алгоритмам построения моделей являются методы, основанные на прореживании сетей глубокого обучения [8, 9], т.е. последовательного удаления параметров, не дающих существенного прироста качества модели. В работах [11, 12] рассматривается послойное построение модели с отдельным критерием оптимизации для каждого слоя. В работах [13, 14, 15] предлагается декомпозиция модели на порождающую и разделяющую, оптимизируемых последовательно.

В качестве порождающих моделей в сетях глубокого обучения выступают ограниченные машины Больцмана [3] и автокодировщики [22]. В работе [23] рассматриваются алгоритмы регуляризации автокодировщиков, позволяющих формально рассматривать данные модели как порождающие модели с использованием байесового вывода. В работе [24] рассматриваются регуляризованные автокодировщики и свойства оценок их правдоподобия. В работе [25] предлагается обобщение автокодировщика с использованием вариационного байесовского вывода [2]. В работе [26] рассматриваются модификации вариационного автокодировщика и ступенчатых сетей [27] для случая построения многослойных порождающих моделей.

В данной работе предлагается метод получения вариационной нижней оценки правдоподобия модели с использованием модифицированного алгоритма стохастического градиентного спуска. Модификация заключается в добавлении шумовой компоненты. Эта компонента позволяет получить более точные оценки правдоподобия модели для сравнения моделей и выбора наиболее адекватной из них. Рассматривается ряд модификаций базового алгоритма. В качестве базового алгоритма выступает алгоритм оптимизации параметров модели с использованием стохастического градиентного спуска без контроля переобучения. Он заключается в итеративном вычислении градиента по параметрам от функции правдоподобия обучающей выборки и изменении значений параметров с его учетом. Приводится сравнение с алгоритмом получения вариационной нижней оценки, представленном в [39]. Рассматриваются следующие модификации базового алгоритма: оптимизация с кросс-валидацией с использованием и без использования регуляризации модели, алгоритм получения вариационной оценки правдоподобия модели с применением нормального распределения, алгоритм получения вариационной оценки правдоподобия с использованием сто-

хастического градиентного спуска, алгоритм получения вариационной оценки правдоподобия с использованием стохастической динамики Ланжевена. Данные алгоритмы решают следующие проблемы оптимизации моделей градиентным спуском: оптимизация модели с меньшими затратами вычислительных ресурсов, быстрая сходимость оптимизации, контроль переобучения и выбор наиболее адекватной модели. Под переобучением понимается потеря обобщающей способности модели с увеличением правдоподобия обучающей выборки [28]. Переобучение характерно для моделей с большим количеством параметров, сопоставимым с мощностью обучающей выборки, что встречается в случае выбора моделей глубокого обучения [3, 119]. Также алгоритмы имеют дальнейшую возможность применения к градиентным алгоритмам оптимизации гиперпараметров, описанным в [49].

Свойства представленных в данной работе алгоритмов исследуются на выборках, на которых проверялась работа алгоритма вероятностного обратного распространения ошибок [120], где авторы акцентируют внимание на оптимизации параметров модели.

## 2.1. Постановка задачи оптимизации правдоподобия моделей

Задана выборка

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}, i = 1, \dots, m, \quad (2.1)$$

состоящая из множества пар «объект-метка»

$$\mathbf{x}_i \in \mathbf{X} \subset \mathbb{R}^n, \quad y_i \in \mathbf{Y} \subset \mathbb{Y}.$$

Метка  $y$  объекта  $\mathbf{x}$  принадлежит либо множеству:  $y \in \mathbb{Y} = \{1, \dots, Z\}$  в случае задачи классификации, где  $Z$  — число классов, либо некоторому подмножеству вещественных чисел  $y \in \mathbb{Y} \subseteq \mathbb{R}$  в случае задачи регрессии.

Моделью глубокого обучения  $\mathbf{f}$  назовем суперпозицию функций

$$\mathbf{f}(\mathbf{w}, \mathbf{X}) = \mathbf{f}_1(\mathbf{f}_2(\dots \mathbf{f}_K(\mathbf{w}, \mathbf{X}))) : \mathbb{R}^{m \times n} \rightarrow \mathbb{Y}^m, \quad (2.2)$$

где  $\mathbf{f}_k$  — подмодели, параметрическое семейство дважды дифференцируемых по параметрам вектор-функций,  $k \in \{1, \dots, K\}$ ;  $\mathbf{w} \in \mathbb{R}^u$  — вектор параметров моделей.

Для каждой модели определена функция правдоподобия  $p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f})$ , где  $\mathbf{x}$  — строка матрицы  $\mathbf{X}$ ,  $\mathbf{y}$  — вектор меток зависимой переменной  $y$ . Множество всех рассматриваемых моделей обозначим  $\mathfrak{F}$ . Для каждой модели  $\mathbf{f}$  из конечного множества моделей  $\mathfrak{F}$  задано априорное распределение параметров  $p(\mathbf{w}|\mathbf{f})$ .

**Определение 20.** Сложностью модели  $\mathbf{f}$  назовем правдоподобие модели:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{f}) = \int_{\mathbf{w} \in \mathbb{R}^u} p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f})p(\mathbf{w}|\mathbf{f})d\mathbf{w}. \quad (2.3)$$

Модели  $\mathbf{f} \in \mathfrak{F}$  имеют различные размерности  $u$  соответствующих векторов параметров. Также заданы различные априорные распределения их параметров  $p(\mathbf{w}|\mathbf{f})$ .

**Определение 21.** Модель классификации  $\mathbf{f}$  назовем оптимальной среди моделей  $\mathfrak{F}$ , если достигается максимум интеграла (2.3).

Требуется найти оптимальную модель  $\mathbf{f}$  среди заданного множества моделей  $\mathfrak{F}$ , а также значения ее параметров  $\mathbf{w}$ , доставляющие максимум апостериорной вероятности

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f})p(\mathbf{w}|\mathbf{f})}{p(\mathbf{y}|\mathbf{X}, \mathbf{f})}. \quad (2.4)$$

*Пример 1.* Рассмотрим задачу линейной регрессии:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{A}^{-1}),$$

где  $\mathbf{A}$  — диагональная матрица. Правдоподобие зависимой переменной имеет вид

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}) = (2\pi)^{-\frac{m}{2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})\right), \quad (2.5)$$

априорное распределение параметров модели имеет вид

$$p(\mathbf{w}|\mathbf{f}) = (2\pi)^{-\frac{n}{2}} |\mathbf{A}|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{w}^\top \mathbf{A} \mathbf{w}\right). \quad (2.6)$$

Правдоподобие модели (2.3) в этом примере вычисляется аналитически [118]:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{f}) = (2\pi)^{-\frac{m}{2}} |\mathbf{A}|^{\frac{1}{2}} |\mathbf{H}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})^\top(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})\right) \exp\left(-\frac{1}{2}\hat{\mathbf{w}}^\top \mathbf{A} \hat{\mathbf{w}}\right), \quad (2.7)$$

где  $\hat{\mathbf{w}}$  — значение наиболее вероятных (2.4) параметров модели:

$$\hat{\mathbf{w}} = \arg \max p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f}) = (\mathbf{A} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y},$$

$\mathbf{H}$  — гессиан функции потерь  $L$  модели:

$$\mathbf{H} = \nabla \nabla_{\mathbf{w}} \left( \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{1}{2}\mathbf{w}^\top \mathbf{A} \mathbf{w} \right) = \mathbf{A} + \mathbf{X}^\top \mathbf{X},$$

$$L = -\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}).$$

**Пример 2.** Рассмотрим задачу классификации, в которой модель — нейросеть с softmax-слоем на выходе:

$$\mathbf{f} = \mathbf{f}_{\text{SM}}(\mathbf{f}_2(\dots \mathbf{f}_K(\mathbf{x}))),$$

$\mathbf{f}_2, \dots, \mathbf{f}_K$  — дифференцируемые функции,  $\mathbf{f}_{\text{SM}}$  — многомерная логистическая функция:

$$\mathbf{f}_{\text{SM}} = \frac{\mathbf{f}_2(\dots \mathbf{f}_K(\mathbf{x}))}{\sum_{r=1}^Z \exp(f_{r,2}(\dots \mathbf{f}_K(\mathbf{x})))},$$

где  $f_{r,2}$  —  $r$ -я компонента функции  $\mathbf{f}_2$ . Компонента  $r$  вектора  $\mathbf{f}_{\text{SM}}$  определяет вероятность принадлежности объекта  $\mathbf{x}$  к классу  $r$ . Логарифм правдоподобия зависимой переменной аналогично (2.5) имеет вид

$$\log p(y|\mathbf{x}, \mathbf{w}, \mathbf{f}) = \log \hat{f}_{\hat{r}, \text{SM}}(\mathbf{f}_2(\dots \mathbf{f}_K(\mathbf{x}))),$$

где  $\hat{f}_{\hat{r}, \text{SM}}$  соответствует ненулевой компоненте вектора  $y$ :

$$\hat{r} \in \{1, \dots, Z\} : y_r > 0,$$

$y_r$  — компонента вектора  $y$ .

Интеграл правдоподобия (2.3) модели является трудновычислимым для данного семейства моделей. Одним из методов вычисления приближенного значения правдоподобия является получение вариационной оценки правдоподобия.

В качестве функции, приближающей логарифм интеграла (2.3), будем рассматривать его нижнюю оценку, полученную при помощи неравенства Йенсена [2]. Получим нижнюю оценку логарифма правдоподобия модели, используя неравенство

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X}, \mathbf{f}) &= \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w} + D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})) \geq \quad (2.8) \\ &\geq \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w} = \\ &= -D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f})) + \int_{\mathbf{w}} q(\mathbf{w}) \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}) d\mathbf{w}, \end{aligned}$$

где  $D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f}))$  — расстояние Кульбака–Лейблера между двумя распределениями:

$$\begin{aligned} D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f})) &= - \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{w}|\mathbf{f})}{q(\mathbf{w})} d\mathbf{w}, \\ p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f}) &= p(\mathbf{y}|\mathbf{X}, \mathbf{f})p(\mathbf{w}|\mathbf{f}). \end{aligned}$$

**Определение 22.** Вариационной оценкой логарифма правдоподобия модели (2.3)  $\log p(\mathbf{y}|\mathbf{X}, \mathbf{f})$  называется оценка  $\log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{f})$ , полученная аппроксимацией неизвестного апостериорного распределения  $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})$  заданным распределением  $q(\mathbf{w})$ .

Будем рассматривать задачу нахождения вариационной оценки как задачу оптимизации. Пусть задано множество распределений  $Q = \{q(\mathbf{w})\}$ . Сведем задачу нахождения наиболее близкой вариационной нижней оценки интеграла (2.3) к оптимизации вида

$$\hat{q}(\mathbf{w}) = \arg \max_{q \in Q} \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w} | \mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w}.$$

В данной работе в качестве множества  $Q$  рассматривается нормальное распределение и распределение параметров, неявно получаемое оптимизацией градиентными методами.

Оценка (3.11) является нижней, поэтому может давать некорректные оценки для правдоподобия (2.3). Для того, чтобы оценить величину этой ошибки, докажем следующее утверждение.

**Утверждение 1.** Пусть задано множество  $Q = \{q(\mathbf{w})\}$  непрерывных распределений. Максимизация вариационной нижней оценки

$$\int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w} | \mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w}$$

логарифма интеграла (2.3) эквивалентна минимизации расстояния Кульбака–Лейблера между распределением  $q(\mathbf{w}) \in Q$  и апостериорным распределением параметров  $p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \mathbf{f})$ :

$$\hat{q} = \arg \max_{q \in Q} \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w} | \mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w} \Leftrightarrow \hat{q} = \arg \min_{q \in Q} D_{KL}(q(\mathbf{w}) || p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \mathbf{f})), \quad (2.9)$$

$$D_{KL}(q(\mathbf{w}) || p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \mathbf{f})) = \int_{\mathbf{w}} q(\mathbf{w}) \frac{q(\mathbf{w})}{p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \mathbf{f})} d\mathbf{w}.$$

*Доказательство.* Доказательство непосредственно следует из (3.11). Вычитая из обеих частей равенства  $D_{KL}(q(\mathbf{w}) || p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \mathbf{f}))$ , получим

$$\log p(\mathbf{y} | \mathbf{X}, \mathbf{f}) - D_{KL}(q(\mathbf{w}) || p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \mathbf{f})) = \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w} | \mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w},$$

где  $\log p(\mathbf{y} | \mathbf{X}, \mathbf{f})$  — выражение, не зависящее от  $q(\mathbf{w})$ .  $\square$

Таким образом, задача нахождения вариационной оценки, близкой к значению интеграла (2.3) сводится к поиску распределения  $\hat{q}$ , аппроксимирующего распределение  $p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \mathbf{f})$  наилучшим образом. Доказательство утверждения 1 см. в Приложении.

**Определение 23.** Модель  $\mathbf{f}$  назовем субоптимальной на множестве моделей  $\mathfrak{F}$  по множеству распределений  $Q$ , если модель доставляет максимум нижней вариационной оценке интеграла (4.5)

$$\max_{q \in Q} \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w} | \mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w}. \quad (2.10)$$

Субоптимальность модели может быть также названа вариационной оптимальностью модели или LB-оптимальностью (*Lower Bound — нижняя граница*) модели.

Вариационная оценка (3.11) интерпретируется как оценка сложности модели по принципу минимальной длины описания [1], где первое слагаемое определяет количество информации для описания выборки, а второе слагаемое — длину описания самой модели [39].

$$\text{MDL}(\mathbf{y}, \mathbf{f}) = \text{Len}(\mathbf{y}|\hat{\mathbf{w}}, \mathbf{f}) + \text{COMP}(\mathbf{f}),$$

где  $\text{Len}(\mathbf{y}|\hat{\mathbf{w}}, \mathbf{f})$  — длина описания матрицы  $\mathbf{y}$  с использованием модели  $\mathbf{f}$  и оценки вектора параметров  $\hat{\mathbf{w}}$ , полученных методом наибольшего правдоподобия, а  $\text{COMP}(\mathbf{f})$  — величина, характеризующая *параметрическую сложность* модели, т.е. способность модели описать произвольную выборку из  $\mathbb{R}^n$  [1].

В данной работе решается задача выбора субоптимальной модели при различных заданных множествах  $Q$ .

## 2.2. Методы получения вариационной оценки правдоподобия

Ниже представлены методы получения вариационных нижних оценок (2.10) правдоподобия (2.3). В первом подразделе рассматривается метод, основанный на аппроксимации апостериорного распределения  $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})$  (2.4) многомерным гауссовым распределением с диагональной матрицей ковариаций. В последующих разделах рассматриваются методы, основанные на различных модификациях стохастического градиентного спуска.

### Аппроксимация нормальным распределением

В качестве множества  $Q = \{q(\mathbf{w})\}$  задано параметрическое семейство нормальных распределений с диагональными матрицами ковариаций:

$$q \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}), \quad (2.11)$$

где  $\mathbf{A}_q$  — диагональная матрица ковариаций,  $\boldsymbol{\mu}_q$  — вектор средних компонент.

Пусть априорное распределение  $p(\mathbf{w}|\mathbf{f})$  (3.4) параметров модели задано как нормальное:

$$p(\mathbf{w}|\mathbf{f}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{A}^{-1}).$$

Тогда оптимизация (4.5) имеет вид

$$\int_{\mathbf{w}} q(\mathbf{w}) \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}) d\mathbf{w} - D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f})) \rightarrow \max_{\mathbf{A}_q, \boldsymbol{\mu}_q}, \quad (2.12)$$

где расстояние  $D_{\text{KL}}$  между двумя гауссовыми величинами рассчитывается как

$$D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f})) = \frac{1}{2} (\text{Tr}[\mathbf{A}\mathbf{A}_q^{-1}] + (\boldsymbol{\mu} - \boldsymbol{\mu}_q)^T \mathbf{A} (\boldsymbol{\mu} - \boldsymbol{\mu}_q) - u + \ln |\mathbf{A}^{-1}| - \ln |\mathbf{A}_q^{-1}|).$$

В качестве приближенного значения интеграла

$$\int_{\mathbf{w}} q(\mathbf{w}) \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}) d\mathbf{w}$$

предлагается использовать формулу

$$\int_{\mathbf{w}} q(\mathbf{w}) \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}) d\mathbf{w} \approx \sum_{i=1}^m \log p(y_i|\mathbf{x}_i, \mathbf{w}_i),$$

где  $\mathbf{w}_i$  — реализация случайной величины из распределения  $q(\mathbf{w})$ .

Итоговая функция оптимизации (2.12) имеет вид

$$\mathbf{f} = \arg \max_{\mathbf{A}_q, \boldsymbol{\mu}_q} \sum_{i=1}^m \log p(y_i|\mathbf{x}_i, \mathbf{w}_i) - D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f})). \quad (2.13)$$

*Пример 2.* Пусть задана выборка  $\mathfrak{D}$ , в которой переменная  $y$  не зависит от  $\mathbf{x}$ :

$$y \sim \mathcal{N}(\mathbf{w}, \mathbf{B}^{-1}), \quad (2.14)$$

$$\begin{aligned} \mathbf{B}^{-1} &= \begin{pmatrix} 2 & 1,8 \\ 1,8 & 2 \end{pmatrix}, \\ p(\mathbf{w}|\mathbf{f}) &= \mathcal{N}(\mathbf{0}, \mathbf{I}). \end{aligned}$$

График аппроксимации распределения параметров представлен на рис. 2.1, а. Как видно из графика, с использованием метода (2.13) получено грубое приближение апостериорного распределения  $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})$ , что может существенно занизить оценку правдоподобия модели.

Данный пример показывает, что качество итоговой аппроксимации распределения  $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})$  значительно зависит от схожести распределений  $\hat{q}$  и  $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})$ . В силу диагональности матрицы  $\mathbf{A}_q$  и полного ранга матрицы  $\mathbf{B}$  итоговое распределение  $\hat{q}$  не может адекватно приблизить данное распределение  $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})$ .

### Аппроксимация с использованием градиентного метода

В качестве множества распределений  $Q = \{q(\mathbf{w})\}$ , аппроксимирующих неизвестное распределение  $\log p(\mathbf{y}|\mathbf{X}, \mathbf{f})$ , используются распределения параметров, полученные в ходе их оптимизации.

Представим неравенство (3.11)

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{f}) \geq \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w} = E_{q(\mathbf{w})}(\log p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})) - S(q(\mathbf{w})), \quad (2.15)$$

где  $S$  — энтропия распределения:

$$S(q(\mathbf{w})) = - \int_{\mathbf{w}} q(\mathbf{w}) \log q(\mathbf{w}) d\mathbf{w},$$

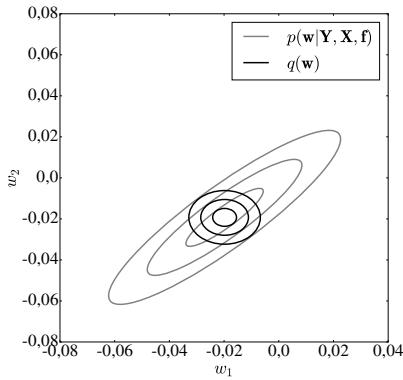
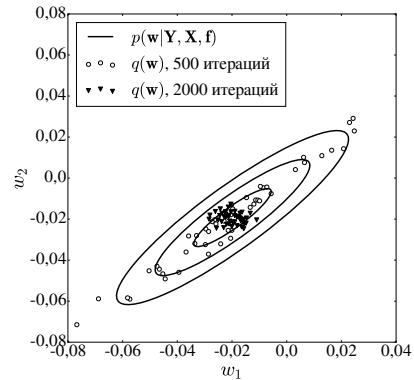
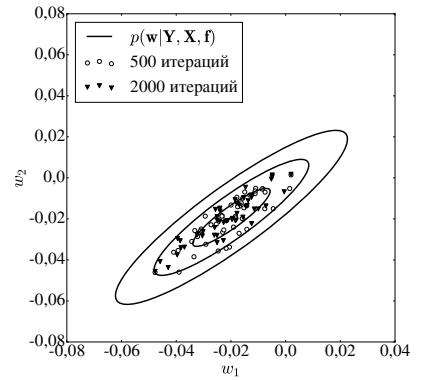
Рис. 2.0. *a*Рис. 2.0. *б*Рис. 2.0. *в*

Рис. 2.1. Аппроксимация распределения *a*) нормальным распределением, *б*) распределением, полученным с помощью градиентного спуска, *в*) с использованием стохастической динамики Ланжевена.

$$p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f}) = p(\mathbf{w}|\mathbf{f})p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}),$$

$\mathbb{E}_{q(\mathbf{w})}(\log p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f}))$  — матожидание логарифма вероятности  $\log p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})$ :

$$\mathbb{E}_{q(\mathbf{w})}(\log p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})) = \int_{\mathbf{w}} \log p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})q(\mathbf{w})d\mathbf{w}.$$

Оценка распределений производится при оптимизации параметров. Оптимизация выполняется в режиме мультистарта [121], т.е. при запуске оптимизации параметров модели из нескольких разных начальных приближений. Основная проблема такого подхода — вычисление энтропии  $S$  распределений  $q(\mathbf{w}) \in Q$ . Ниже представлен метод получения оценок энтропии (2.19)  $S$  и оценок правдоподобия (2.15).

Запустим  $r$  процедур оптимизаций модели  $\mathbf{f}$  из разных начальных приближений:

$$L(\mathbf{w}^1, \mathbf{y}, \mathbf{X}), \dots, L(\mathbf{w}^r, \mathbf{y}, \mathbf{X}) \rightarrow \min,$$

где  $r$  — число оптимизаций,  $L$  — оптимизируемая функция потерь

$$L = - \sum_{i=1}^m \log p(y_i, \mathbf{w}|\mathbf{x}_i, \mathbf{f}) = -\log p(\mathbf{w}|\mathbf{f}) - \sum_{i=1}^m \log p(y_i|\mathbf{x}_i, \mathbf{w}, \mathbf{f}). \quad (2.16)$$

Пусть начальные приближения параметров  $\mathbf{w}^1, \dots, \mathbf{w}^r$  порождены из некоторого начального распределения  $q^0(\mathbf{w})$ :

$$\mathbf{w}^1, \dots, \mathbf{w}^r \sim q^0(\mathbf{w}).$$

Для описания произвольного градиентного метода оптимизации параметров модели введем понятие оператора оптимизации. Оно используется для вычисления оценки энтропии распределения, полученного под действием этой оптимизации.

**Определение 24.** Назовем оператором оптимизации алгоритм  $T$  выбора вектора параметров  $\mathbf{w}'$  по параметрам предыдущего шага  $\mathbf{w}$ :

$$\mathbf{w}' = T(\mathbf{w}).$$

Рассмотрим оператор градиентного спуска:

$$T(\mathbf{w}) = \mathbf{w} - \gamma \nabla L(\mathbf{w}, \mathbf{y}, \mathbf{X}), \quad (2.17)$$

где  $\gamma$  — длина шага градиентного спуска.

Пусть значения  $\mathbf{w}^1, \dots, \mathbf{w}^r$  — реализации случайной величины из некоторого распределения  $q(\mathbf{w})$ . Начальная энтропия распределения  $q(\mathbf{w})$  соответствует энтропии распределения  $q^0(\mathbf{w})$ , из которого были порождены начальные приближения оптимизации параметров  $\mathbf{w}^1, \dots, \mathbf{w}^r$ . Под действием оператора  $T$  распределение параметров  $\mathbf{w}_1, \dots, \mathbf{w}_r$  изменяется. Для учета энтропии распределений, полученных в ходе оптимизации, формализуем метод, представленный в [41].

**Теорема 1.** Пусть  $T$  — оператор градиентного спуска,  $L$  — функция потерь, градиент  $\nabla L$  которой имеет константу Липшица  $C_L$ . Пусть  $\mathbf{w}^1, \dots, \mathbf{w}^r$  — начальные приближения оптимизации модели, где  $r$  — число начальных приближений. Пусть  $\gamma$  — длина шага градиентного спуска, такая что

$$\gamma < \frac{1}{C_L}, \quad \gamma < \left( \max_{g \in \{1, \dots, r\}} \lambda_{\max}(\mathbf{H}(\mathbf{w}^g)) \right)^{-1}, \quad (2.18)$$

где  $\lambda_{\max}$  — наибольшее по модулю собственное значение гессиана  $\mathbf{H}$  функции потерь  $L$ .

При выполнении неравенств (2.18) разность энтропий распределений  $q'(\mathbf{w}), q(\mathbf{w})$  на смежных шагах почти наверное сходится к следующему выражению:

$$S(q'(\mathbf{w})) - S(q(\mathbf{w})) \approx \frac{1}{r} \sum_{g=1}^r (-\gamma \text{Tr}[\mathbf{H}(\mathbf{w}'^g)] - \gamma \text{Tr}[\mathbf{H}(\mathbf{w}'^g)\mathbf{H}(\mathbf{w}'^g)]) + o_{\gamma^2 \rightarrow 0}(1), \quad (2.19)$$

где  $\mathbf{H}$  — гессиан функции потерь  $L$ .

*Доказательство.* Предварительно приведем две леммы [122, 123], требуемые для доказательства теоремы.

**Лемма 1.** Пусть  $T$  — оператор градиентного спуска,  $L$  — дважды дифференцируемая функция потерь, градиент  $\nabla L$  которой имеет константу Липшица  $C_L$ . Пусть для длины шага  $\gamma$  выполнено неравенство  $\gamma < \frac{1}{C_L}$ . Тогда  $T$  является диффеоморфизмом.

**Лемма 2.** Пусть  $\mathbf{w}$  — случайный вектор с непрерывным распределением  $q(\mathbf{w})$ . Пусть  $T$  — биективное отображение вектора  $\mathbf{w}$  в пространство той же размерности. Пусть  $q'(\mathbf{w})$  — распределение вектора  $T(\mathbf{w})$ . Тогда справедливо утверждение

$$S(q'(\mathbf{w})) - S(q(\mathbf{w})) = \int_{\mathbf{w}} q'(\mathbf{w}) \log \left| \frac{\partial T(\mathbf{w})}{\partial \mathbf{w}} \right| d\mathbf{w}. \quad (\text{П.1})$$

Рассмотрим очередной шаг оптимизации. При  $\gamma < \frac{1}{C}$  оператор градиентного спуска  $T$  является диффеоморфизмом, а значит, и биекцией, справедлива формула (П.1). По усиленному закону больших чисел

$$S(q'(\mathbf{w})) - S(q(\mathbf{w})) \approx \frac{1}{r} \sum_{g=1}^r \log \left| \frac{\partial T(\mathbf{w}'^g)}{\partial \mathbf{w}} \right|.$$

Логарифм якобиана  $\log \left| \frac{\partial T(\mathbf{w}'^g)}{\partial \mathbf{w}} \right|$  оператора  $T$  запишем как

$$\log \left| \frac{\partial T(\mathbf{w}'^g)}{\partial \mathbf{w}} \right| = \log |\mathbf{I} - \gamma \mathbf{H}| = \sum_{i=1}^u \log (1 - \gamma \lambda_i), \quad (\text{П.2})$$

где  $\lambda_i$  —  $i$ -е собственное значение гессиана  $\mathbf{H}$ .

При  $(\gamma \lambda_i)^2 \leq (\gamma \lambda_{\max})^2 < 1$  выражение (П.2) раскладывается в ряд Тейлора:

$$\sum_{t=1}^u \log (1 - \gamma \lambda_i) = -\gamma \text{Tr}[\mathbf{H}(\mathbf{w}'^g)] - \gamma^2 \text{Tr}[\mathbf{H}(\mathbf{w}'^g) \mathbf{H}(\mathbf{w}'^g)] + o_{\gamma^2 \rightarrow 0}(1).$$

Просуммировав полученные выражения для каждой точки мультистарта и вынеся  $o_{\gamma^2 \rightarrow 0}(1)$  за скобки, получим выражение (2.19), что и требовалось доказать.  $\square$

Получим итоговую формулу для оценки правдоподобия модели.

**Утверждение 2.** Оценка (2.15) на шаге оптимизации  $\tau$  представима в виде

$$\log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{f}) \approx \frac{1}{r} \sum_{g=1}^r L(\mathbf{w}_\tau^g, \mathbf{X}, \mathbf{y}) + S(q^0(\mathbf{w})) + \frac{1}{r} \sum_{b=1}^\tau \sum_{g=1}^r (-\gamma \text{Tr}[\mathbf{H}(\mathbf{w}_b^g)] - \gamma^2 \text{Tr}[\mathbf{H}(\mathbf{w}_b^g) \mathbf{H}(\mathbf{w}_b^g)]) \quad (2.20)$$

с точностью до слагаемых вида  $o_{\gamma^2 \rightarrow 0}(1)$ , где  $\mathbf{w}_b^g$  —  $g$ -я реализация параметров модели на шаге оптимизации  $b$ ,  $q^0(\mathbf{w})$  — начальное распределение.

*Доказательство.* Представим энтропию распределения  $q^\tau(\mathbf{w})$  следующим образом:

$$S(q^\tau(\mathbf{w})) = S(q^0(\mathbf{w})) - S(q^0(\mathbf{w})) + S(q^1(\mathbf{w})) - S(q^1(\mathbf{w})) + \dots - S(q^{\tau-1}(\mathbf{w})) + S(q^\tau(\mathbf{w})).$$

Каждая разность энтропий вида  $S(q^b(\mathbf{w})) - S(q^{b-1}(\mathbf{w}))$  по теореме с точностью до  $o_{\gamma^2 \rightarrow 0}(1)$  представима в виде

$$S(q^b(\mathbf{w})) - S(q^{b-1}(\mathbf{w})) \approx \frac{1}{r} \sum_{g=1}^r (-\gamma \text{Tr}[\mathbf{H}(\mathbf{w}_b^g)] - \gamma^2 \text{Tr}[\mathbf{H}(\mathbf{w}_b^g) \mathbf{H}(\mathbf{w}_b^g)]). \quad (\text{П.3})$$

Формула (2.20) получается подстановкой в выражение (2.15) суммы выражений вида (П.3), а также начальной энтропии  $S(q^0(\mathbf{w}))$ .  $\square$

В [41] предлагается алгоритм приближенного вычисления для выражения, находящегося под знаком суммы в (2.20):

$$-\gamma \text{Tr}[\mathbf{H}(\mathbf{w}^g)] - \gamma^2 \text{Tr}[\mathbf{H}(\mathbf{w}^g) \mathbf{H}(\mathbf{w}^g)] \approx \mathbf{r}_0^\top (-2\mathbf{r}_0 + 3\mathbf{r}_1 - \mathbf{r}_2),$$

где вектор  $\mathbf{r}_0$  порождается из нормального распределения:

$$\mathbf{r}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \mathbf{r}_1 = \mathbf{r}_0 - \gamma \mathbf{r}_0^\top \nabla \nabla L, \quad \mathbf{r}_2 = \mathbf{r}_1 - \gamma \mathbf{r}_1^\top \nabla \nabla L.$$

Заметим, что при приближении параметров модели к точке экстремума оценка правдоподобия устремляется в минус бесконечность в силу постоянно убывающей энтропии. Таким образом, чем ближе градиентный метод приближает параметры модели к точке экстремума, тем менее точной становится оценка правдоподобия модели. Один из методов борьбы с данной проблемой будет представлен в разделе 3.3. Доказательство теоремы и утверждения 2 см. в Приложении.

### Модификация алгоритма оптимизации модели.

В качестве оператора  $T$  предлагается использовать псевдослучайный стохастический градиентный спуск, т.е. градиентный спуск, оптимизирующий параметры  $\mathbf{w}^1, \dots, \mathbf{w}^r$  по некоторой случайной подвыборке  $\hat{\mathbf{X}}, \hat{\mathbf{y}}$ , одинаковой для каждой точки старта  $\mathbf{w}^1, \dots, \mathbf{w}^r$ :

$$T(\mathbf{w}) = \mathbf{w} - \frac{m}{\hat{m}} \gamma \nabla L(\mathbf{w}, \hat{\mathbf{y}}, \hat{\mathbf{X}}), \quad (2.21)$$

где  $\hat{\mathbf{X}}$  — случайная подвыборка выборки  $\mathbf{X}$ , одинаковая для всех точек мультистарта,  $\hat{\mathbf{y}}$  — соответствующие метки классов,

$$|\hat{\mathbf{X}}| = \hat{m}.$$

Как и версия алгоритма с использованием градиентного спуска (2.21), основной проблемой модифицированного алгоритма оценки интеграла (2.10) является грубость аппроксимации исходного распределения  $p(\mathbf{w}|\mathbf{f}, \mathfrak{D})$ .

Рассмотрим пример 2 (2.14). График аппроксимации распределения  $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})$  представлен на рис. 2.1,б. Как видно из графика, градиентный спуск сходится к mode распределения. При небольшом количестве итераций полученное распределение также слабо аппроксимирует апостериорное распределение. При приближении к точке экстремума снижается вариационная оценка

Рис. 2.2. Псевдокод алгоритма получения вариационной нижней оценки правдоподобия модели с использованием градиентного спуска

**Require:**  $\mathbf{X}, \mathbf{y}, p(\mathbf{w}|\mathbf{f})$ ;

**Require:** критерий останова  $M$ , начальное распределение параметров  $q_0$ , количество точек мультистарта  $r$ , функция потерь  $L$ , ее первая и вторая производные;

**Ensure:**  $\log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{f})$ ;

1: **for**  $g = 1, \dots, r$  **do**

2:    $\mathbf{w}^g \sim q_0$ ;

3:    $\mathbf{S} = \mathbf{S}(q_0)$ ;

4: **while** не достигнут критерий останова  $M$  **do**

5:   **for**  $g = 1, \dots, r$  **do**

6:      $\mathbf{w}^g = \mathbf{w}^g - \nabla L$ ;

7:      $\mathbf{r}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ;

8:      $\mathbf{r}_1 = \mathbf{r}_0 - \gamma \mathbf{r}_0^\top \nabla \nabla L(\mathbf{w}^g, \mathbf{y}, \mathbf{X})$ ;

9:      $\mathbf{r}_2 = \mathbf{r}_1 - \gamma \mathbf{r}_1^\top \nabla \nabla L(\mathbf{w}^g, \mathbf{y}, \mathbf{X})$ ;

10:     $\mathbf{S}^g = \mathbf{r}_0^\top (-2\mathbf{r}_0 + 3\mathbf{r}_1 - \mathbf{r}_2)$ ;

11:     $\mathbf{S} = \frac{1}{r} \sum_{g=1}^r \mathbf{S}^g$ ;

12:     $\hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}) = \frac{1}{r} \sum_{g=1}^r p(\mathbf{y}|\mathbf{X}, \mathbf{w}^g, \mathbf{f})$ ;

13:     $\hat{p}(\mathbf{w}|\mathbf{f}) = \frac{1}{r} \sum_{g=1}^r p(\mathbf{w}^g|\mathbf{f})$ ;

14:     $\log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{f}) = \log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}) + \log \hat{p}(\mathbf{w}|\mathbf{f})$ ;

правдоподобия модели, что интерпретируется как возможное начало переобучения [41]. Таким образом, снижение оценки (2.20) можно использовать как критерий остановки оптимизации модели для снижения эффекта переобучения.

На рис. 2.1 представлена аппроксимация распределения  $p(\mathbf{w}|\mathbf{Y}, \mathbf{X}, \mathbf{f})$  различными методами: *a*) нормальным распределением с диагональной матрицей ковариаций, *b*) с помощью градиентного спуска, *c*) с помощью стохастической динамики Ланжевена. Точками отмечены параметры модели  $\mathbf{f}$ , полученные в ходе нескольких запусков оптимизации и являющиеся реализациами случайной величины с распределением  $q(\mathbf{w})$ . Нормальное распределение слабо аппроксирует распределение  $p(\mathbf{w}|\mathbf{Y}, \mathbf{X}, \mathbf{f})$  в силу диагональности матрицы ковариаций. Распределение, полученное с помощью градиентного спуска, слабо аппроксирует распределение  $p(\mathbf{w}|\mathbf{Y}, \mathbf{X}, \mathbf{f})$ , так как сходится к моде.

### Аппроксимация с использованием динамики Ланжевена

Для достижения нижней оценки интеграла (2.10), более близкой к реальному значению логарифма интеграла (2.3), чем оценка с использованием градиентного спуска, предлагается использовать стохастическую динамику Ланжевена [43]. Стохастическая динамика Ланжевена представляет собой вариант стохастического градиентного спуска с добавлением гауссового шума:

$$T(\mathbf{w}) = \mathbf{w} - \gamma \nabla L - \frac{m}{\hat{m}} \log p(\hat{\mathbf{y}}|\hat{\mathbf{X}}, \mathbf{w}, \mathbf{f}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \frac{\gamma}{2} \mathbf{I}), \quad (2.22)$$

где  $\hat{\mathbf{X}}$  — псевдослучайная подвыборка,  $\hat{\mathbf{y}}$  — соответствующие метки,  $\hat{m}$  — размер подвыборки. Длина шага оптимизации  $\gamma$  удовлетворяет условиям, гарантирующим сходимость алгоритма в стандартных ситуациях [43]:

$$\sum_{\tau=1}^{\infty} \gamma_{\tau} = \infty, \quad \sum_{\tau=1}^{\infty} \gamma_{\tau}^2 < \infty.$$

Для оценки энтропии с учетом шума  $\boldsymbol{\varepsilon}$  предлагается использовать следующее неравенство [123, 124]:

$$\hat{S}(q^{\tau}(\mathbf{w})) \geq \frac{1}{2} u \log \left( \exp \left( \frac{2S(q^{\tau}(\mathbf{w}))}{u} \right) + \exp \left( \frac{2S(\boldsymbol{\varepsilon})}{u} \right) \right),$$

где  $\tau$  — текущий шаг оптимизации,  $S(\mathcal{N}(0, \frac{\gamma}{2}))$  — энтропия нормального распределения,  $\hat{S}(q^{\tau}(\mathbf{w}))$  — энтропия распределения  $q^{\tau}$  с учетом добавленного шума  $\boldsymbol{\varepsilon}$ .

В отличие от стохастического градиентного спуска стохастическая динамика Ланжевена сходится к апостериорному распределению параметров  $p(\mathbf{w}|\mathfrak{D}, \mathbf{f})$  [43, 125]. График аппроксимации апостериорного распределения с использованием динамики Ланжевена представлен на рис. 2.1, *c*. При одинаковом количестве итераций динамика Ланжевена продолжает аппроксимировать апостериорное распределение, в то время как градиентный спуск сходится к моде

распределения. Как видно из графика, алгоритм, основанный на стохастической динамике Ланжевена, способен давать более точную вариационную оценку правдоподобия (2.10). В то же время алгоритм более требователен к настройке параметров оптимизации [126]: “*быстро изменяющаяся кривизна [траекторий параметров модели] делает методы стохастической градиентной динамики Ланжевена по умолчанию неэффективными*”.

### 2.3. Анализ методов выбора моделей

Для анализа свойств предложенного критерия субоптимальности в задачах регрессии и классификации, а также методов получения нижних оценок правдоподобия модели в задачах выбора моделей был проведен ряд вычислительных экспериментов на выборках Boston Housing, Protein Structure, а также на небольшой подвыборке YearPredictionMSD (далее — Boston, Protein и MSD) [127] и подвыборке изображений рукописных цифр MNIST [128].

Для выборок Boston, Protein и MSD была рассмотрена задача регрессии

$$\mathbf{y} = \mathbf{f}(\mathbf{X}, \mathbf{w}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \mathbf{f} \in \mathfrak{F}.$$

В качестве множества моделей  $\mathfrak{F}$  были рассмотрены нейросети с одним скрытым слоем и softplus-функцией активации:

$$\mathbf{f}(\mathbf{w}, \mathbf{X}) = \text{softplus}(\mathbf{X}\mathbf{W}_1)\mathbf{W}_2, \quad (2.23)$$

где  $\mathbf{W}_1 \in \mathbb{R}^{n \times n_1}$  — матрица параметров скрытого слоя нейросети,  $\mathbf{W}_2 \in \mathbb{R}^{n_1 \times 1}$  — матрица параметров выходного слоя нейросети,  $\text{softplus}(\mathbf{X}) = \log(1+\exp(\mathbf{X}))$ .

Для выборки Boston также было рассмотрено множество моделей с тремя скрытыми слоями, построенных аналогично однослоевой модели (2.23). Размер каждого слоя равнялся 50.

Для выборки MNIST была рассмотрена задача бинарной классификации: из выборки были взяты только объекты, соответствующие цифрам 7 и 9. Размерность выборки была понижена с 784 до 50 методом главных компонент аналогично [129]. Для анализа моделей, полученных в случае высокой вероятности переобучения, из обучающей выборки были взяты первые 500 объектов. В качестве модели рассматривалась нейросеть с тремя скрытыми слоями

$$\mathbf{f}(\mathbf{w}, \mathbf{X}) = \sigma(\text{softplus}(\text{softplus}(\text{softplus}(\mathbf{X}\mathbf{W}_1)\mathbf{W}_2)\mathbf{W}_3)\mathbf{W}_4),$$

где  $\sigma(\mathbf{X}) = (1+\exp(-\mathbf{X}))^{-1}$  — сигмоида,  $\mathbf{W}_1, \dots, \mathbf{W}_4$  — параметры нейросети.

Во всех экспериментах исходная выборка  $\mathfrak{D}$  разбивалась на обучающую и контрольную подвыборки:  $\mathfrak{D} = \mathfrak{D}_{\text{train}} \sqcup \mathfrak{D}_{\text{test}}$ .

Оптимизация параметров производилась на подвыборке  $\mathfrak{D}_{\text{train}}$ . Для контроля переобучения некоторых алгоритмов из обучающей выборки  $\mathfrak{D}_{\text{train}}$  формировалась валидационная выборка  $\mathfrak{D}_{\text{valid}}$ , на которой не проводилась оптимизация параметров модели. Мощность валидационной выборки  $\mathfrak{D}_{\text{valid}}$  составляла

0,1 мощности обучающей выборки  $\mathfrak{D}_{\text{train}}$ , объекты для валидационной выборки выбирались случайным образом независимо для каждого старта алгоритма. Качество полученных моделей проверялось на подвыборке  $\mathfrak{D}_{\text{test}}$ . Критерием качества модели выступали среднеквадратичное отклонение вектора  $\mathbf{y}$  от вектора  $\mathbf{f}(\mathbf{w}, \mathbf{X})$  (RMSE) в случае задачи регрессии и доля верно предсказанных меток класса (Accuracy) в задаче классификации, а также соответствующие критерии при возмущении элементов выборки:

$$\text{RMSE}_\sigma = \text{RMSE}(\mathbf{f}(\mathbf{w}, \mathbf{X} + \boldsymbol{\varepsilon}), \mathbf{y}), \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma \mathbf{I}). \quad (2.24)$$

Были рассмотрены шесть алгоритмов.

1. Базовый алгоритм: оптимизация параметров без валидации и ранней остановки. Оптимизация проводилась с использованием стохастического градиентного спуска (2.21). Для данного алгоритма априорное распределение  $p(\mathbf{w}|\mathbf{f})$  не использовалось.
2. Алгоритм с валидацией. Для контроля переобучения во время оптимизации качество модели оценивалось на валидационной выборке  $\mathfrak{D}_{\text{valid}}$ . Для данного алгоритма априорное распределение также не использовалось.
3. Алгоритм с валидацией и введенным априорным распределением. В качестве априорного распределения рассматривается распределение вида  $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \alpha \mathbf{I})$ , где  $\alpha$  — дисперсия.
4. Нахождение вариационной нижней оценки с использованием стохастического градиентного спуска.
5. Нахождение вариационной нижней оценки с использованием стохастической динамики Ланжеяна.
6. Нахождение вариационной нижней оценки с аппроксимацией нормальным распределением (2.13).

Параметры модели выбирались из точек мультистарта (алгоритмы 1—5) или порождались из распределения  $\hat{q}$  (алгоритм 6). Количество точек мультистарта:  $r = 10$  для задач регрессии и  $r = 25$  для задачи классификации. Для алгоритмов 2—6 применялась ранняя остановка: каждые  $\tau_{\text{val}}$  итераций производилась оценка внутреннего критерия качества модели. В качестве критерия остановки применялось следующее условие: значение внутреннего критерия качества не улучшалось  $3\tau_{\text{val}}$  итераций. Для разных алгоритмов внутренним критерием качества выступали различные величины:

1. функция потерь  $L$  (2.16) на валидационной выборке  $\mathfrak{D}_{\text{valid}}$  для алгоритмов 2, 3,
2. вариационная нижняя оценка правдоподобия (3.11) на обучающей выборке  $\mathfrak{D}_{\text{train}}$  для алгоритмов 4, 5, 6.

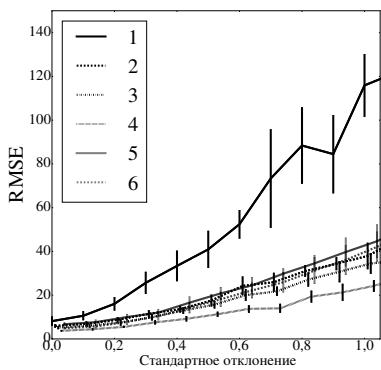
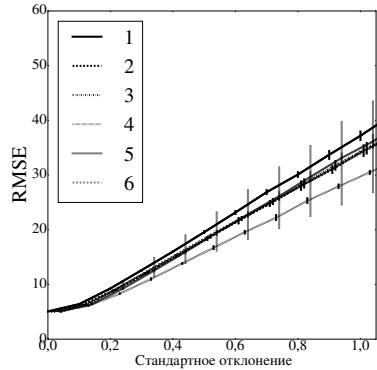
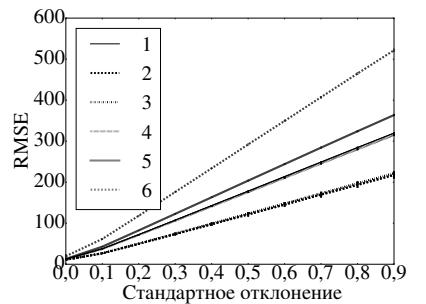
Для каждой модели назначались различные значения параметра  $\alpha$  ( $\alpha \in \{10, \dots, 10^9\}$ ) и длины шага оптимизации  $\gamma$ , отбирались наилучшие модели.

Таблица 2.1. Описание выборок для экспериментов по выбору моделей

Выборка $\mathfrak{D}$	Интервал валидации, $\tau_{\text{val}}$	Количество объектов, $m$	Количество признаков, $n$	Размер подвыборки, $\hat{m}$	Размер скрытого слоя, $n_1$
Boston Housing	100	506	13	$\hat{m} = m$	50
Protein	1000	45000	9	$\hat{m} = 200$	100
MSD	1000	5000	91	$\hat{m} = 50$	100
MNIST	100	500	50	$\hat{m} = 100$	50

Таблица 2.2. Результаты эксперимента по выбору моделей

Выборка $\mathfrak{D}$	Алгоритмы					
	1	2	3	4	5	6
Результаты, RMSE/Accuracy						
Boston, один слой	$8,1 \pm 2,0$	$5,9 \pm 0,7$	$5,2 \pm 0,6$	<b><math>3,7 \pm 0,2</math></b>	$6,7 \pm 0,7$	$5,0 \pm 0,4$
Boston, 3 слоя	$7,1 \pm 1,3$	$4,3 \pm 0,1$	$4,4 \pm 0,4$	<b><math>3,2 \pm 0,06</math></b>	$4,6 \pm 0,4$	$6,8 \pm 1,6$
Protein	$5,1 \pm 0,0$	$5,1 \pm 0,0$	$5,1 \pm 0,0$	$5,1 \pm 0,0$	$5,1 \pm 0,0$	<b><math>5,0 \pm 0,1</math></b>
MSD	$12,2 \pm 0,0$	<b><math>10,9 \pm 0,1</math></b>	<b><math>10,9 \pm 0,1</math></b>	$12,2 \pm 0,0$	$12,9 \pm 0,0$	$19,6 \pm 3,6$
MNIST	$0,985 \pm 0,002$	$0,984 \pm 0,002$	<b><math>0,986 \pm 0,002</math></b>	$0,914 \pm 0,005$	$0,979 \pm 0,003$	$0,971 \pm 0,001$
Результаты, RMSE/Accuracy <sub>0,5</sub>						
Boston, один слой	$43,9 \pm 9,4$	$18,6 \pm 2,0$	$15,8 \pm 2,3$	<b><math>11,9 \pm 1,1</math></b>	$20,3 \pm 3,1$	$18,2 \pm 3,3$
Boston, 3 слоя	$23,4 \pm 4,9$	$18,7 \pm 2,8$	$18,3 \pm 3,0$	<b><math>9,0 \pm 0,7</math></b>	$14,5 \pm 2,6$	$15,2 \pm 2,7$
Protein	$19,5 \pm 0,3$	$18,5 \pm 0,5$	$18,6 \pm 0,3$	<b><math>16,7 \pm 0,3</math></b>	$19,3 \pm 0,6$	$19,7 \pm 3,7$
MSD	$178,3 \pm 0,8$	<b><math>121,3 \pm 4,5</math></b>	$123,7 \pm 2,5$	$175,8 \pm 1,0$	$203,8 \pm 1,4$	$292,0 \pm 2,0$
MNIST	$0,931 \pm 0,004$	$0,929 \pm 0,006$	<b><math>0,934 \pm 0,007</math></b>	$0,857 \pm 0,007$	$0,919 \pm 0,008$	$0,916 \pm 0,004$
Результаты, RMSE/Accuracy <sub>1,0</sub>						
Boston, один слой	$120,9 \pm 33,4$	$42,5 \pm 6,3$	$32,5 \pm 6,0$	<b><math>25,7 \pm 3,2</math></b>	$42,4 \pm 5,7$	$41,3 \pm 6,3$
Boston, 3 слоя	$46,1 \pm 15,8$	$40,5 \pm 5,3$	$38,6 \pm 8,0$	<b><math>16,5 \pm 2,5</math></b>	$30,4 \pm 7,9$	$26,2 \pm 6,9$
Protein	$37,0 \pm 0,8$	$34,4 \pm 1,1$	$35,0 \pm 1,0$	<b><math>30,6 \pm 0,6</math></b>	$36,6 \pm 1,1$	$35,0 \pm 8,1$
MSD	$319,6 \pm 1,4$	<b><math>217,5 \pm 8,2</math></b>	$221,9 \pm 4,2$	$314,8 \pm 1,8$	$363,7 \pm 1,9$	$521,6 \pm 3,1$
MNIST	<b><math>0,814 \pm 0,010</math></b>	$0,808 \pm 0,010$	$0,812 \pm 0,008$	$0,772 \pm 0,010$	$0,802 \pm 0,009$	$0,800 \pm 0,009$
Сходимость алгоритмов, тыс. итераций						
Boston, один слой	25	25	25	14	10	27
Boston, 3 слоя	25	4	9	10	1	6
Protein	60	40	80	40	75	85
MSD	250	330	335	250	460	120
MNIST	1	6	3	13	3	25

Рис. 2.2. *a*Рис. 2.2. *б*Рис. 2.2. *в*Рис. 2.3. Возмущение выборки для однослойных нейросетей: *а*) Boston Housing, *б*) Protein, *в*) MSD.

Описание эксперимента представлено в табл. 1. Результаты экспериментов представлены в табл. 2. На рис. 2.3 представлен график зависимости  $RMSE_{\sigma}$  от параметра  $\sigma$  для однослойных моделей.

Модели имеют достаточно большое число параметров, поэтому в ходе оптимизации параметров может произойти переобучение. На выборке Boston Housing базовый алгоритм (1) показал наихудший результат в силу переобучения, при этом алгоритм 4 показал лучший результат по сравнению с алгоритмами 2 и 3. В данном случае использование вариационной оценки предпочтительнее алгоритмов, основанных на кросс-валидации. На выборке Protein все алгоритмы показали схожие результаты. На выборке MSD алгоритмы 4, 5, 6 показали худший результат в сравнении с алгоритмами, использующими валидационную подвыборку. Наихудший результат показал алгоритм 6, что говорит о значительном отличии апостериорного распределения параметров (2.4) от нормального.

Алгоритм 6 показал низкое качество (2.24) при возмущении объектов выборки в большинстве экспериментов. В трех экспериментах наилучшие показатели по данному критерию показал алгоритм 4. Заметим, что алгоритм 5, являющийся модификацией алгоритма 4, показал худшие результаты как по  $RMSE$ , так и по  $RMSE$  при возмущении объектов выборки. На выборке MNIST алгоритм 4 показал результаты значительно хуже остальных алгоритмов. В целом результаты по данному алгоритму схожи с результатами, описанными в [41]: в отличие от алгоритма 5 алгоритм 4, основанный на стохастическом градиентном спуске, дает заниженную оценку правдоподобия при приближении параметров к точке экстремума. Алгоритм 5, основанный на динамике Ланжевена, также показал худшее время сходимости на выборках MSD и Protein. Возможным дальнейшим улучшением качества этого алгоритма является введение дополнительной корректирующей матрицы, обеспечивающей лучшее время сходления параметров

к апостериорному распределению параметров [43].

Программное обеспечение для проведения экспериментов и проверки результатов находится в [130].

## Глава 3

### Оптимизация гиперпараметров в задаче выбора модели

Решается задача оптимизации гиперпараметров модели глубокого обучения. Для оптимизации гиперпараметров модели предлагаются алгоритмы, основанные на градиентном спуске. Так как сложность рассматриваемых алгоритмов сопоставима со сложностью оптимизации параметров модели, предлагаются проводить оптимизацию параметров и гиперпараметров в единой процедуре. Для выбора адекватных значений гиперпараметров вводятся вероятностные предположения о распределении параметров. В качестве оптимизируемой функции выступает байесовское правдоподобие модели и кросс-валидация. Для получения оценки правдоподобия используются вариационные методы. Проводится вычислительный эксперимент на нескольких выборках.

В работе решается задача оптимизации гиперпараметров моделей глубокого обучения. Под *моделью* понимается суперпозиция функций, решающая задачу классификации или регрессии. Под *гиперпараметрами* модели понимается параметры распределения параметров модели.

Одна из проблем построения моделей глубокого обучения — большое число параметров модели [3], которое достигает нескольких миллионов, а оптимизация модели достигает десятков дней [119]. Задача выбора модели глубокого обучения включает в себя выбор стратегии построения модели, эффективной по вычислительным ресурсам. Проблема оптимизации параметров модели глубокого обучения является вычислительно сложной в силу невыпуклости оптимизируемой функции потерь. Поэтому задача поиска параметров оптимизации является важной, и нахождение оптимальных гиперпараметров сильно влияет на итоговое качество модели.

В данной работе сравниваются градиентные методы оптимизации гиперпараметров. Основным достоинством подобных алгоритмов является их возможность одновременной оптимизации щащительного количества гиперпараметров. В качестве базового алгоритма выступает выбор гиперпараметров модели с использованием случайного поиска. В работах [70, 71, 52] в качестве целевой функции потерь рассматривается потеря на валидационной подвыборке с  $L_2$  регуляризацией. В данной работе рассматривается общая задача оптимизации гиперпараметров. Рассматриваемые алгоритмы и целевые функции потерь реализованы и представлены в качестве библиотеки для оптимизации гиперпараметров моделей [131]. Основным теоретическим вкладом данной работы является анализ рассматриваемых алгоритмов оптимизации гиперпараметров при использовании функции потерь общего вида, а также исследование качества и устойчивости итоговых моделей в случае использования кросс-валидации и вариационной оценки правдоподобия. В экспериментальной части в качестве критерия выбора модели выступают вариационная нижняя оценка правдоподобия модели и ошибка на валидационной части выборки. В отличие от [71], где также производится сравнение алгоритмов оптимизации гиперпараметров,

Algorithm	Type	Optimization iteration complexity	Correctness suppositions
Random search	stochastic	$O(\eta s  \hat{\mathcal{D}} )$	-
Greedy [52]	gradient-based	$O(\eta(s+h) \hat{\mathcal{D}} )$	$\mathbf{H}(\boldsymbol{\theta}) = \mathbf{I}$
HOAG [71]	gradient-based	$O(\eta s  \hat{\mathcal{D}}  + h^2  \hat{\mathcal{D}}  + o)$ , where $o$ is a complexity of linear equation solution	first derivatives of $Q$ and second derivatives of $L$ are Lipschitz functions; $\det \mathbf{H} \neq 0$ ;
DrMAD [70]	gradient-based	$O(\eta s  \hat{\mathcal{D}} )$	Parameter trajectory $\boldsymbol{\theta} = \boldsymbol{\theta}_0, \dots, \boldsymbol{\theta}_\eta$ is linear

Таблица 3.1. Complexity and correctness of the analyzed algorithms

в данной работе исследуется поведение алгоритмов на выборках большой мощности, таких как WISDM [132] и MNIST [128]. Численные эксперименты показывают, что при значительном количестве гиперпараметров, сопоставимым с количеством параметров модели, рассматриваемые алгоритмы предпочтительнее стохастических.

В работах [133, 134] предлагаются стратегии выбора гиперпараметров модели, основанные на случайном выборе параметров. Другим методом, представленным в литературе [135, 136], является обучение вероятностных моделей для предсказания гиперпараметров. В работе [49] отмечается, что данный метод нахождения оптимальных гиперпараметров является неэффективными в случае, когда число гиперпараметров велико.

В работах [49, 50, 70, 71, 52] предлагаются методы оптимизации гиперпараметров, основанные на градиентных алгоритмах оптимизации: восстанавливается вся история изменения параметров в ходе оптимизации, в качестве функции для оптимизации гиперпараметров рассматривается функция потерь от конечного значения параметров, которое выражается через начальное значение параметров. Данная процедура является неэффективной по памяти, т.к. для хранения всей истории оптимизации параметров требуется большое количество памяти. В работе [52] предлагается жадный вариант градиентной оптимизации гиперпараметров. В работе [49] рассматривается оптимизация параметров с моментом, позволяющая эффективно хранить историю параметров в памяти. В работе [70] предлагается метод, рассматривающий траекторию оптимизации параметров как линейную, что также позволяет эффективно хранить историю параметров. В работе [71] рассматривается аппроксимация градиента оптимизируемой функции.

Для решения заданной рассматриваемой задачи требуется выбрать критерий выбора модели [28, 2]. В качестве критерия выбора модели в ряде работ [28, 2, 29, 30, 32] выступает правдоподобие модели. В работах [29, 30, 32]

Алгоритм	Тип алгоритма	Преимущества алгоритма	Недостатки алгоритма
Случайный поиск	стохастический	простота реализации	алгоритм неэффективен при большом количестве гиперпараметров (проклятие размерности)
Жадный алгоритм [52]	градиентный	Возможность одновременной оптимизации параметров и гиперпараметров	Жадность алгоритма
HOAG [71]	градиентный	Быстрая сходимость	Алгоритм требует сложных настроек параметров
DrMAD [70]	градиентный	Алгоритм учитывает алгоритм оптимизации параметров модели и его параметры	Алгоритм страдает от проблем неустойчивости градиентного спуска (градиентный взрыв и затухание); Алгоритм работает в очень жестких предположениях.

Таблица 3.2. Преимущества и недостатки рассматриваемых алгоритмов

рассматривается проблема выбора модели и оценки гиперпараметров в задачах регрессии. Одним из методов получения приближенного значения интеграла правдоподобия является вариационный метод получения нижней оценки интеграла [2]. В работе [38] рассматривается стохастическая версия вариационного метода. В работе [40] рассматривается взаимосвязь градиентных методов получения вариационной нижней оценки интеграла с методом Монте-Карло. Альтернативным критерием выбора модели является минимальная длина описания [1], являющаяся показателем статистической сложности модели и заданной выборки. В работе [39] рассматривается алгоритм получения вариационной нижней оценки правдоподобия для оптимизации гиперпараметров моделей глубокого обучения, проводится связь между правдоподобием модели и минимальной длиной описания.

Альтернативным методом выбора модели является выбор модели на основе скользящего контроля [44, 29]. Проблемой такого подхода является возможная высокая вычислительная сложность [45, 46]. В работах [47, 48] рассматривается проблема смещения оценок качества модели и гиперпараметров, получаемых при использовании  $k$ -fold метода скользящего контроля, при котором выборка делится на  $k$ -частей с обучением на  $k - 1$  части и валидацией результата на оставшейся части выборки.

### 3.1. Постановка задачи оптимизации гиперпараметров моделей

Задана выборка

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}, i = 1, \dots, m, \quad (3.1)$$

состоящая из множества пар «объект-метка»,

$$\mathbf{x}_i \in \mathbf{X} \subset \mathbb{R}^n, \quad y_i \in \mathbf{y} \subset \mathbb{Y},$$

где  $\mathbf{X}$  — матрица объектов,  $\mathbf{y}$  — вектор меток зависимой переменной  $y$ . Метка  $y$  объекта  $\mathbf{x}$  принадлежит либо конечному множеству:  $y \in \mathbb{Y} = \{1, \dots, Z\}$  в случае задачи классификации, где  $Z$  — число классов, либо некоторому подмножеству вещественных чисел  $y \in \mathbb{R}$  в случае задачи регрессии.

Задана дифференцируемая по параметрам модель, приближающая зависимую переменную  $y$ :

$$f : \mathbb{R}^n \rightarrow \mathbb{Y}, \quad \mathbf{w} \in \mathbb{R}^u.$$

Рассмотрим модель  $f$  и ее вероятностную интерпретацию для случая задач регрессии и классификации.

**Регрессия.** Положим, что зависимая переменная распределена нормально:

$$\mathbf{y} = \mathcal{N}(\mathbf{f}, \mathbf{I}), \quad (3.2)$$

где  $\mathbf{f} = \mathbf{f}(\mathbf{w}, \mathbf{X})$ .

Определим правдоподобие выборки  $p(\mathbf{y}|\mathbf{X}, \mathbf{w})$ :

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = -\frac{m}{2} \log(2\pi) - \frac{1}{2} (\mathbf{y} - \mathbf{f}(\mathbf{w}, \mathbf{X}))^\top \mathbf{I} (\mathbf{y} - \mathbf{f}(\mathbf{w}, \mathbf{X})).$$

**Классификация.** В случае двуклассовой классификации положим, что зависимая переменная распределена биномиально:

$$\mathbf{y} = \mathcal{B}(1 - \mathbf{f}, \mathbf{f}), \quad (3.3)$$

где вектор-функция  $\mathbf{f}$  задает вероятность принадлежности объектов  $\mathbf{X}$  к первому классу. В случае многоклассовой классификации зависимая переменная распределена мультиномиально,  $r$ -я компонента  $\mathbf{f}$  задает вероятность принадлежности классу  $r$ . Тогда правдоподобие выборки задается как

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \sum_{\mathbf{x}, y \in \mathbf{X}, \mathbf{x}} \sum_{r=1}^Z [y = r] \log f_r(\mathbf{w}, \mathbf{x}),$$

где  $f_r$  —  $r$ -я компонента функции  $\mathbf{f}$ .

Для задач классификации (3.3) и регрессии (3.2) задано параметрическое априорное распределение  $p(\mathbf{w}|\mathbf{A})$  вида:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{A}^{-1}), \quad (3.4)$$

где  $\mathbf{A}^{-1} = \text{diag}[\alpha_1, \dots, \alpha_u]^{-1}$  — матрица ковариаций диагонального вида. Гипотезы (3.2), (3.3), (3.4) не противоречат друг другу в силу неограниченности нормального распределения [137].

Задача оптимизации гиперпараметров зависит как от критерия выбора модели, так и от метода оптимизации параметров модели. Проиллюстрируем задачу оптимизации гиперпараметров *двусвязным байесовским выводом*. Для дальнейшей формализации задачи в общем виде введем переобозначение:

$$\boldsymbol{\theta} = \mathbf{w}, \quad \mathbf{h} = [\alpha_1, \dots, \alpha_u], \quad (3.5)$$

где  $\boldsymbol{\theta}$  — множество оптимизируемых параметров модели,  $\mathbf{h}$  — множество гиперпараметров модели.

На *первом уровне* байесовского вывода производится оптимизация параметров модели  $f$  по заданной выборке  $\mathfrak{D}$ :

$$\hat{\boldsymbol{\theta}} = \arg \max (-L(\boldsymbol{\theta}, \mathbf{h})) = p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \mathbf{A}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\mathbf{A})}{p(\mathbf{y}|\mathbf{X}, \mathbf{A})}. \quad (3.6)$$

На *втором уровне* производится оптимизация апостериорного распределения гиперпараметров  $\mathbf{h}$ :

$$p(\mathbf{A}|\mathbf{X}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{X}, \mathbf{A})p(\mathbf{A}),$$

где знак « $\propto$ » означает равенство с точностью до нормирующего множителя.

Полагая распределение параметров  $p(\mathbf{A})$  равномерным на некоторой большой окрестности, получим задачу оптимизации гиперпараметров:

$$Q(\boldsymbol{\theta}, \mathbf{h}) = p(\mathbf{y}|\mathbf{X}, \mathbf{A}) = \int_{\mathbf{w} \in \mathbb{R}^u} p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\mathbf{A}) \rightarrow \max_{[\alpha_1, \dots, \alpha_u] \in \mathbb{R}^n}. \quad (3.7)$$

Сформулируем задачу оптимизации гиперпараметров в общем виде. Обозначим за  $\mathbf{h} \in \mathbb{R}^h$  вектор гиперпараметров модели (3.5). Обозначим за  $\boldsymbol{\theta} \in \mathbb{R}^s$  множество всех оптимизируемых параметров (3.5). Пусть задана дифференцируемая функция потерь  $L(\boldsymbol{\theta}, \mathbf{h})$ , по которой производится оптимизация функции  $f$  (3.6). Пусть также задана дифференцируемая функция  $Q(\boldsymbol{\theta}, \mathbf{h})$ , определяющая итоговое качество модели  $f$  и приближающая интеграл (3.7).

Требуется найти параметры  $\hat{\boldsymbol{\theta}}$  и гиперпараметры  $\hat{\mathbf{h}}$  модели, доставляющие минимум следующему функционалу:

$$\hat{\mathbf{h}} = \arg \max_{\mathbf{h} \in \mathbb{R}^h} Q(\hat{\boldsymbol{\theta}}(\mathbf{h}), \mathbf{h}), \quad (3.8)$$

$$\hat{\boldsymbol{\theta}}(\mathbf{h}) = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^s} L(\boldsymbol{\theta}, \mathbf{h}). \quad (3.9)$$

Рассмотрим вид переменной  $\boldsymbol{\theta}$  и функций  $L, Q$  для различных методов выбора модели и оптимизации ее параметров.

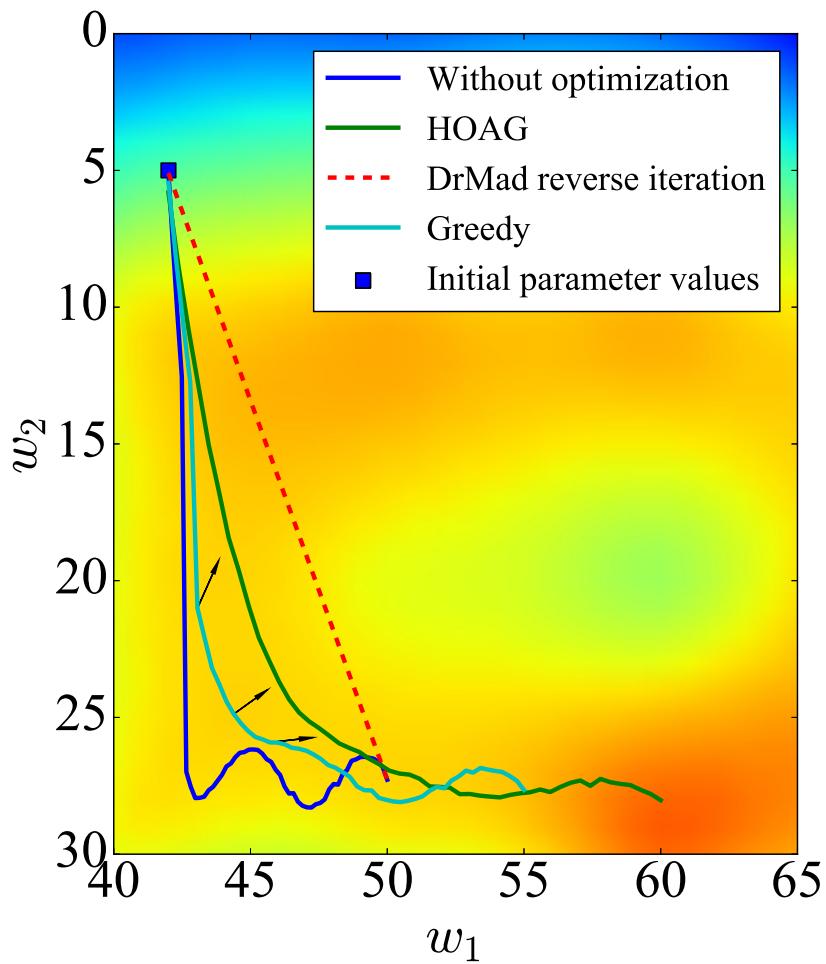


Рис. 3.1. An example of parameter update trajectories. The color displays the value of the validation function  $Q$ . Greedy algorithm optimizes hyperparameter during the parameter optimization, therefore it has the light blue trajectory between the optimized green trajectory of HOAG algorithm and the dark blue trajectory of parameters without hyperparameter optimization. DrMAD uses a linearized dashed parameter trajectory during hyperparameter optimization procedure.

**Базовый метод** Пусть оптимизация параметров и гиперпараметров производится по всей выборке  $\mathfrak{D}$  по одной и той же функции:

$$L(\boldsymbol{\theta}, \mathbf{h}) = Q(\boldsymbol{\theta}) = \log p(\mathbf{y}, \mathbf{w} | \mathbf{X}, \mathbf{A}) = \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}) + \log p(\mathbf{w} | \mathbf{A})$$

Вспомогательная переменная  $\boldsymbol{\theta}$ , по которой производится оптимизация модели  $f$ , соответствует параметрам модели:

$$\boldsymbol{\theta} = \mathbf{w}.$$

**Кросс-валидация** Разобьем выборку  $\mathfrak{D}$  на  $k$  равных частей:

$$\mathfrak{D} = \mathfrak{D}_1 \sqcup \cdots \sqcup \mathfrak{D}_k.$$

Запустим  $k$  оптимизаций модели, каждую на своей части выборки. Положим  $\boldsymbol{\theta} = [\mathbf{w}_1, \dots, \mathbf{w}_k]$ , где  $\mathbf{w}_1, \dots, \mathbf{w}_k$  — параметры модели при оптимизации  $k$ .

Положим функцию  $L$  равной среднему значению минус логарифма апостериорной вероятности по всем  $k - 1$  разбиениям  $\mathfrak{D}$ :

$$L(\boldsymbol{\theta}, \mathbf{h}) = -\frac{1}{k} \sum_{q=1}^k \left( \frac{k}{k-1} \log p(\mathbf{y} \setminus \mathbf{y}_q | \mathbf{X} \setminus \mathbf{X}_q, \mathbf{w}_q) + \log p(\mathbf{w}_q | \mathbf{A}) \right). \quad (3.10)$$

Положим функцию  $Q$  равной среднему значению правдоподобия выборки по частям выборки  $\mathfrak{D}_q$ , на которых не проходила оптимизация параметров:

$$Q(\boldsymbol{\theta}, \mathbf{h}) = \frac{1}{k} \sum_{q=1}^k k \log p(\mathbf{y}_q | \mathbf{X}_q, \mathbf{w}_q).$$

**Вариационная оценка правдоподобия** Положим  $L = -Q$ , равной вариационной оценке правдоподобия модели:

$$\begin{aligned} \log p(\mathbf{y} | \mathbf{X}, \mathbf{A}) &\geq -D_{\text{KL}}(q(\mathbf{w}) || p(\mathbf{w} | \mathbf{A})) + \int_{\mathbf{w}} q(\mathbf{w}) \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \mathbf{A}) d\mathbf{w} \approx \quad (3.11) \\ &\approx \sum_{i=1}^m \log p(y_i | \mathbf{x}_i, \mathbf{w}_i) - D_{\text{KL}}(q(\mathbf{w}) || p(\mathbf{w} | \mathbf{A})) = -L(\boldsymbol{\theta}, \mathbf{h}) = Q(\boldsymbol{\theta}), \end{aligned}$$

где  $q$  — нормальное распределение с диагональной матрицей ковариаций:

$$q \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}), \quad (3.12)$$

где  $\mathbf{A}_q = \text{diag}[\alpha_1^q, \dots, \alpha_u^q]^{-1}$  — диагональная матрица ковариаций,  $\boldsymbol{\mu}_q$  — вектор средних компонент. Расстояние  $D_{\text{KL}}$  между двумя гауссовыми величинами задается как

$$D_{\text{KL}}(q(\mathbf{w}) || p(\mathbf{w} | \mathbf{f})) = \frac{1}{2} (\text{Tr}[\mathbf{A} \mathbf{A}_q^{-1}] + (\boldsymbol{\mu} - \boldsymbol{\mu}_q)^T \mathbf{A} (\boldsymbol{\mu} - \boldsymbol{\mu}_q) - u + \ln |\mathbf{A}^{-1}| - \ln |\mathbf{A}_q^{-1}|).$$

В качестве оптимизируемых параметров  $\boldsymbol{\theta}$  выступают параметры распределения  $q$ :

$$\boldsymbol{\theta} = [\alpha_1, \dots, \alpha_u, \mu_1, \dots, \mu_u].$$

### 3.2. Градиентные методы оптимизации гиперпараметров

Рассмотрим случай, когда оптимизация (3.9) параметров  $\boldsymbol{\theta}$  производится с использованием градиентных методов.

**Определение.** Назовем оператором оптимизации алгоритм  $T$  выбора вектора параметров  $\boldsymbol{\theta}'$  по параметрам предыдущего шага  $\boldsymbol{\theta}$ :

$$\boldsymbol{\theta}' = T(\boldsymbol{\theta}, \mathbf{h}).$$

Рассмотрим оператор градиентного спуска, производящий  $\eta$  шагов оптимизации:

$$\hat{\boldsymbol{\theta}} = T \circ T \circ \dots \circ T(\boldsymbol{\theta}_0, \mathbf{h}) = T^\eta(\boldsymbol{\theta}_0, \mathbf{h}), \quad (3.13)$$

где

$$T(\boldsymbol{\theta}, \mathbf{h}) = \boldsymbol{\theta} - \gamma \nabla L(\boldsymbol{\theta}, \mathbf{h}),$$

$\gamma$  — длина шага градиентного спуска,  $\boldsymbol{\theta}_0$  — начальное значение параметров  $\boldsymbol{\theta}$ . В данной работе в качестве опреатора оптимизации параметров модели выступает стохастический градиентный спуск:

$$T(\boldsymbol{\theta}, \mathbf{h})_{\text{SGD}} = \boldsymbol{\theta} - \gamma \nabla L(\boldsymbol{\theta}, \mathbf{h})|_{\mathfrak{D}=\hat{\mathfrak{D}}},$$

где  $\hat{\mathfrak{D}}$  — случайная подвыборка исходной выборки  $\mathfrak{D}$ .

Перепишем задачу оптимизации (3.8), (3.9) в следующем виде

$$\hat{\mathbf{h}} = \arg \max_{\mathbf{h} \in \mathbb{R}^h} Q(T^\eta(\boldsymbol{\theta}_0, \mathbf{h})), \quad (3.14)$$

где  $\boldsymbol{\theta}_0$  — начальное значение параметров  $\boldsymbol{\theta}$ .

Оптимизационную задачу (4.5) предлагается решать с использованием градиентного спуска. Вычисление градиента от функции  $Q(T^\eta(\boldsymbol{\theta}_0, \mathbf{h}))$  по гиперпараметрам  $\mathbf{h}$  является вычислительно сложным в силу внутренней процедуры оптимизации  $T(\boldsymbol{\theta}_0, \mathbf{h})$ . Общая схема оптимизации гиперпараметров представлена следующим образом:

1. От 1 до  $l$ :
2. Инициализировать параметры  $\boldsymbol{\theta}$  при условии гиперпараметров  $\mathbf{h}$ .
3. Приближенно решить задачу оптимизации (4.5) и получить новый вектор параметров  $\mathbf{h}'$
4.  $\mathbf{h} = \mathbf{h}'$ .

где  $l$  — количество итераций оптимизации гиперпараметров. Рассмотрим методы приближенного решения данной задачи оптимизации.

**Жадный алгоритм** В качестве правила обновления вектора гиперпараметров  $\mathbf{h}$  на каждом шаге оптимизации (3.13) выступает градиентный спуск с учетом обновления параметров  $\boldsymbol{\theta}$  на данном шаге:

$$\mathbf{h}' = \mathbf{h} - \gamma_{\mathbf{h}} \nabla_{\mathbf{h}} Q(T(\boldsymbol{\theta}, \mathbf{h}), \mathbf{h}) = \mathbf{h} - \gamma_{\mathbf{h}} \nabla_{\mathbf{h}} Q(\boldsymbol{\theta} - \gamma \nabla L(\boldsymbol{\theta}, \mathbf{h}), \mathbf{h}),$$

где  $\gamma_{\mathbf{h}}$  — длина шага оптимизации гиперпараметров.

**Алгоритм HOAG** Предлагается получить приближенное значения градиента гиперпараметров  $\nabla_{\mathbf{h}} Q(T^{\eta}(\boldsymbol{\theta}_0, \mathbf{h}))$  на основе следующей формулы:

$$\nabla_{\mathbf{h}} Q(T^{\eta}(\boldsymbol{\theta}_0, \mathbf{h})) = \nabla_{\mathbf{h}} Q(\boldsymbol{\theta}, \mathbf{h}) - (\nabla_{\boldsymbol{\theta}, \mathbf{h}}^2 L(\boldsymbol{\theta}, \mathbf{h}))^T \mathbf{H}(\boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \mathbf{h}),$$

где  $\mathbf{H}$  — гессиан функции  $L$  по параметрам  $\boldsymbol{\theta}$ .

Процедура получения приближенного значения градиента гиперпараметров  $\nabla_{\mathbf{h}} Q$  производится итеративно:

1. Провести  $\eta$  шагов оптимизации:  $\boldsymbol{\theta} = T(\boldsymbol{\theta}_0, \mathbf{h})$ .
2. Решить линейную систему для вектора  $\boldsymbol{\lambda}$ :  $\mathbf{H}(\boldsymbol{\theta})\boldsymbol{\lambda} = \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \mathbf{h})$ .
3. Приближенное значение градиентов гиперпараметра вычисляется как:  

$$\hat{\nabla}_{\mathbf{h}} Q = \nabla_{\mathbf{h}} Q(\boldsymbol{\theta}, \mathbf{h}) - \nabla_{\boldsymbol{\theta}, \mathbf{h}} L(\boldsymbol{\theta}, \mathbf{h})^T \boldsymbol{\lambda}.$$

Итоговое правило обновления:

$$\mathbf{h}' = \mathbf{h} - \gamma_{\mathbf{h}} \hat{\nabla}_{\mathbf{h}} Q. \quad (3.15)$$

В данной работе для приближенного решения шага 2 алгоритма HOAG используется стохастический градиентный спуск в силу сложности вычисления гессиана  $\mathbf{H}(\boldsymbol{\theta})$ .

### Алгоритм DrMad

Для получения градиента от оптимизируемой функции  $Q$  как от функции от начальных параметров  $\boldsymbol{\theta}_0$  предлагается пошагово восстановить  $\eta$  шагов оптимизации  $T(\boldsymbol{\theta}_0)$  в обратном порядке аналогично методу обратного распространения ошибок. Для упрощения данной процедуры вводится предположение, что траектория изменения параметров  $\boldsymbol{\theta}$  линейна:

$$\boldsymbol{\theta}^{\tau} = \boldsymbol{\theta}_0 + \frac{\tau}{\eta} T(\boldsymbol{\theta}). \quad (3.16)$$

Алгоритм вычисления приближенного значения градиента  $\nabla_{\mathbf{h}}$  является частным случаем алгоритма обратного распространения ошибки и представим в следующем виде:

1. Провести  $\eta$  шагов оптимизации:  $\boldsymbol{\theta} = T(\boldsymbol{\theta}_0, \mathbf{h})$ .
2. Положим  $\hat{\nabla}_{\mathbf{h}} = \nabla_{\mathbf{h}} Q(\boldsymbol{\theta}, \mathbf{h})$ .
3. Положим  $d\mathbf{v} = \mathbf{0}$ .

4. Для  $\tau = \eta \dots 1$  повторить:
5. Вычислить значения параметров  $\boldsymbol{\theta}^\tau$  (3.16).
6.  $d\mathbf{v} = \gamma \hat{\nabla}_{\boldsymbol{\theta}}$ .
7.  $\hat{\nabla}\mathbf{h} = \hat{\nabla}\mathbf{h} - d\mathbf{v} \nabla_{\mathbf{h}} \nabla_{\boldsymbol{\theta}} Q$ .
8.  $\hat{\nabla}\boldsymbol{\theta} = \hat{\nabla}\boldsymbol{\theta} - d\mathbf{v} \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} Q$ .

Итоговое правило обновления гиперпараметров аналогично (3.15). В работе [70] отмечается неустойчивость алгоритма при высоких значениях длины шага градиентного спуска  $\gamma$ . Поэтому вместо исходного правила (3.16) в данной работе первые 5% значений параметров не рассматриваются, а также учитывается только каждый  $\tau_k$  шаг оптимизации:

$$\boldsymbol{\theta}^\tau = \boldsymbol{\theta}_{\tau_0} + \frac{\tau}{\eta} T(\boldsymbol{\theta}), \quad \tau \in \{\tau_0, \dots, \eta\}, \tau \bmod \tau_k = 0, \quad (3.17)$$

где  $\tau_0 = [0.05 \cdot \eta]$ .

### 3.3. Анализ алгоритмов оптимизации гиперпараметров

Для анализа рассматриваемых алгоритмов оптимизации гиперпараметров был проведен ряд вычислительных экспериментов на выборках MNIST [128], WISDM [132], а также на синтетических данных.

Рассматривались следующие критерии качества:

1. Наилучшее значение  $\hat{Q} = \max_{j \in \{1, \dots, l\}} Q^j$ .
2. Среднее число итераций алгоритма для сходимости. Под данным показателем понимается число шагов оптимизации гиперпараметров, при котором ошибка  $Q$  изменяется не более чем на 1% от своего наилучшего значения:

$$\arg \min_j : \frac{Q^j - Q^0}{\hat{Q} - Q^0} \geq 0.99,$$

где  $Q^0$  — значение функции  $Q$  до начала оптимизации гиперпараметров.

3. Внешний критерий качества моделей  $E$ :

$$E = \text{RMSE} = \left( \frac{1}{m} \sum_1^m (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2 \right)^{\frac{1}{2}}$$

в случае задачи регрессии,

$$E = \text{Accuracy} = 1 - \frac{1}{m} \sum_1^m [f(\mathbf{x}_i, \mathbf{w}) \neq y_i]$$

в случае задачи классификации.

Алгоритм	Тип алгоритма	Сложность работы одной итерации $O(\eta s  \hat{\mathcal{D}} )$	Предположения для корректности
Случайный поиск	стохастический	$O(\eta s  \hat{\mathcal{D}} )$	-
Жадный алгоритм [52]	градиентный	$O(\eta(s+h) \hat{\mathcal{D}} )$	$\mathbf{H}(\boldsymbol{\theta}) = \mathbf{I}$
HOAG [71]	градиентный	$O(\eta s  \hat{\mathcal{D}}  + h^2  \hat{\mathcal{D}}  + o)$ , где $o$ — время решения уравнения пункте 3	первые производные $Q$ и вторые производные $L$ — липшицевы; $\det \mathbf{H} \neq 0$ ;
DrMAD [70]	градиентный	$O(\eta s  \hat{\mathcal{D}} )$	Траектория оптимизации параметров $\boldsymbol{\theta} = \boldsymbol{\theta}_0, \dots, \boldsymbol{\theta}_\eta$ — линейная

Таблица 3.3. Основные свойства рассматриваемых алгоритмов

4. Внешний критерий качества моделей  $E_\sigma$  при возмущении параметров модели:

$$E_\sigma = \text{RMSE}_\sigma = \left( \frac{1}{m} \sum_1^m (f(\mathbf{x}_i, \mathbf{w} + \boldsymbol{\varepsilon} - \mathbf{y}_i))^2 \right)^{\frac{1}{2}}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma \mathbf{I}).$$

В качестве улучшаемого алгоритма рассматривался случайный поиск параметров с количеством итераций поиска, совпадающих с количеством итераций оптимизации гиперпараметров  $l$ :  $l = 50$  для синтетической выборки и выборки WISDM,  $l = 25$  для выборки MNIST. Рассматриваемые алгоритмы представлены в Табл. 3.3. Пример поведения траекторий параметров под действием алгоритмов приведен на Рис. 3.1. В качестве функций  $Q$  и  $L$  рассматривались функции кросс-валидации (3.10) с  $k = 4$  и вариационной оценки правдоподобия (3.11).

На всех выборках гиперпараметры инициализировались случайно из равномерного распределения:

$$\mathbf{h} \sim \mathcal{U}(a, b)^h,$$

где  $a = -2, b = 10$  для синтетической выборки и  $a = -4, b = 10$  для выборок WISDM и MNIST.

Длина градиентного шага  $\gamma_{\mathbf{h}}$  подбиралась для каждого алгоритма из сетки значений вида  $\{r \cdot 10^s, s \leq 1, r \in \{1, 25, 50, 75\}\}$  таким образом, чтобы итоговое значение гиперпараметров  $\mathbf{h}$  удовлетворяло следующему правилу:

$$a_{\min} \leq \min(\mathbf{h}), \quad \max(\mathbf{h}) \leq b_{\max},$$

где  $a_{\min} = -2.5, b_{\max} = 10.5$  для синтетической выборки и  $a_{\min} = -5, b_{\max} = 11$  для выборок WISDM и MNIST. Калибровка значения  $\gamma$  проводилась на небольшом количестве итераций оптимизаций гиперпараметров  $l$ :  $l = 50$  для

синтетической выборки,  $l = 10$  для выборки WISDM  $l = 5$  для выборки MNIST. В случае, если алгоритмы показывали неустойчивую работу непосредственно во время запуска эксперимента (взрыв градиента или численное переполнение), то длина шага  $\gamma_h$  понижалась. Для алгоритма DrMad параметр  $\tau_k$ , отвечающий за количество рассматриваемых шагов оптимизации был установлен как  $\tau_k = 1$  для синтетической выборки и выборки WISDM,  $\tau_k = 10$  для выборки MNIST.

**Синтетическая выборка** Синтетические данные были порождены по следующему правилу:

$$\mathbf{y} = \mathbf{X} + \boldsymbol{\varepsilon}, \quad \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

где  $m = 40$ ,  $n = 1$ . В качестве модели  $\mathbf{f}$  выступает регрессия с признаками  $\{\mathbf{X}^0, \dots, \mathbf{X}^9, \sin(\mathbf{X}), \cos(\mathbf{X})\}$ .

Было проведено 5 запусков для каждого алгоритма. Графики итоговых полиномов представлены на Рис. 3.2. Как видно из графиков, с использованием вариационной оценки удалось получить полиномы, близкие к линейным моделям. Подобные модели показывают наилучшее правдоподобие в силу слабого переобучения и хорошего качества на тестовой выборке.

**WISDM** Выборка WISDM состоит из набора записей акселерометра. Каждой записи соответствуют три координаты по осям акселерометра. В качестве набора объектов рассматривалось наборы из 199 последовательных записей акселерометра. В качестве набора меток рассматривалась евклидовая норма соответствующих 200-х записей акселерометра.

Рассматривалась нейросеть с 10 нейронами на скрытом слое:

$$\mathbf{f} = \mathbf{W}_2 \cdot \text{RELU}(\mathbf{W}_1 \mathbf{X} + \mathbf{b}_1) + \mathbf{b}_2,$$

где  $\mathbf{W}_1, \mathbf{b}_1$  — параметры первого слоя нейросети,  $\mathbf{W}_2, \mathbf{b}_2$  — параметры второго слоя нейросети,

$$\text{RELU}(\mathbf{x}) = \max(\mathbf{0}, \mathbf{x}).$$

Графики сходимости алгоритмов, а также качества полученных моделей представлены на Рис. 3.3. Как видно из графиков, градиентные алгоритмы DrMad и HOAG показывают значительно худший результат по сравнению с жадным алгоритмом оптимизации. Случайный поиск показывает достаточно хорошие результаты в случае небольшого числа оптимизируемых гиперпараметров  $\mathbf{h}$ . В случае, когда в качестве функции  $Q$  используется вариационная нижняя оценка правдоподобия (3.11) и количество гиперпараметров велико, эффективно работающими алгоритмами оказалась жадная оптимизация и HOAG. HOAG имеет большее время сходимости и требует более сложных вычислений в процессе оптимизации.

**MNIST** Выборка MNIST состоит из множества изображений рукописных цифр. Рассматривалась нейросеть с 300 нейронами на скрытом слое.

Графики сходимости алгоритмов, а также качества полученных моделей представлены на Рис. 3.4. Как видно из графиков, модели, достигающие наилучшей оценки правдоподобия, имеют наихудшее итоговое качество, но более

устойчивы к возмущению параметров модели. Для дополнительного анализа данной проблемы были проведены эксперименты по оптимизации моделей на выборке с добавленным шумом с использованием значений гиперпараметров  $\mathbf{h}$ , полученных ранее:

$$\hat{\mathfrak{D}} = \mathfrak{D} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \hat{\sigma}\mathbf{I}),$$

где  $\hat{\sigma}$  варьировалась в отрезке от 0 до 0.5. График зависимости качества моделей от значения  $\hat{\sigma}$  приведен на Рис. 3.5. Гиперпараметры, достигающие наибольших значений вариационной оценки (3.11) менее подвержены шуму в обучающей выборке, что можно интерпретировать как меньшую подверженность к переобучению.

Как можно видеть по результатам экспериментов, градиентные методы показывают лучший результат, чем случай поиск в случае большого количества гиперпараметров. Наилучшие результаты были получены жадным поиском. Алгоритм DrMad, показавший результаты хуже, чем жадный алгоритм и НОАГ, является упрощенной версией алгоритма, представленного в [70]. Данный алгоритм позволяет проводить оптимизацию не только гиперпараметров, но параметров алгоритма оптимизации  $T$ . Поэтому возможным развитием метода DrMad является получение оптимальных значений параметров оптимизации.

Algorithm	$L, Q$	$Q(\theta, \mathbf{A})$	Convergence	E	$E_{0.25}$	$E_{0.5}$
<i>Synthetic</i>						
Random search	(3.10)	<b>-171.6</b>	<b>26.2</b> $\pm$ <b>20.0</b>	<b>1.367</b>	<b>1.410</b>	<b>1.555</b>
Greedy	(3.10)	-172.5	30.0 $\pm$ 24.5	1.421	1.439	1.536
DrMAD	(3.10)	-174.1	40.2 $\pm$ 16.1	1.403	1.424	<b>1.512</b>
HOAG	(3.10)	-174.7	29.4 $\pm$ 24.0	<b>1.432</b>	1.463	1.553
Random Search	(3.11)	-63.5	32.4 $\pm$ 18.7	1.368	1.426	1.546
Greedy	(3.11)	-25.5	<b>1.2</b> $\pm$ <b>0.4</b>	1.161	1.174	1.193
DrMAD	(3.11)	<b>-25.1</b>	10.6 $\pm$ 0.8	1.157	1.163	1.184
HOAG	(3.11)	-25.8	10.8 $\pm$ 1.5	<b>1.141</b>	<b>1.149</b>	<b>1.177</b>
<i>WISDM</i>						
Random search	(3.10)	<b>-1086661.1</b>	22.0 $\pm$ 19.3	<b>0.660</b>	<b>0.670</b>	<b>0.690</b>
Greedy	(3.10)	-1086707.1	<b>15.4</b> $\pm$ <b>17.2</b>	0.707	0.723	0.769
DrMAD	(3.10)	-1086708.2	29.2 $\pm$ 8.0	0.694	0.708	0.742
HOAG	(3.10)	-1086733.5	28.2 $\pm$ 7.13	0.701	0.724	0.753
Random search	(3.11)	-35420.4	14.4 $\pm$ 7.8	0.732	0.755	0.785
Greedy	(3.11)	<b>-3552.9</b>	<b>1.0</b> $\pm$ <b>0.0</b>	<b>0.702</b>	<b>0.730</b>	<b>0.767</b>
DrMAD	(3.11)	-26091.4	50.0 $\pm$ 0.0	0.729	0.753	0.816
HOAG	(3.11)	-16566.6	49.0 $\pm$ 0.0	0.733	0.755	0.801
<i>MNIST</i>						
Random search	(3.10)	-3236.4	7.8 $\pm$ 1.9	0.981	<b>0.966</b>	<b>0.866</b>
Greedy	(3.10)	<b>-3416.7</b>	10.8 $\pm$ 10.4	0.979	0.962	0.860
DrMAD	(3.10)	-3469.0	17.0 $\pm$ 5.6	<b>0.982</b>	0.962	0.831
HOAG	(3.10)	-3748.6	<b>8.6</b> $\pm$ <b>7.3</b>	0.980	0.961	0.853
Random search	(3.11)	-1304556.4	14.2 $\pm$ 5.7	<b>0.982</b>	0.943	0.814
Greedy	(3.11)	<b>-11136.2</b>	<b>1.0</b> $\pm$ <b>0.0</b>	0.977	<b>0.952</b>	<b>0.884</b>
DrMAD	(3.11)	-1305432.9	24.6 $\pm$ 0.5	<b>0.982</b>	0.941	0.813
HOAG	(3.11)	-280061.6	24.0 $\pm$ 0.0	0.981	0.943	0.819

Таблица 3.4. Experiment results

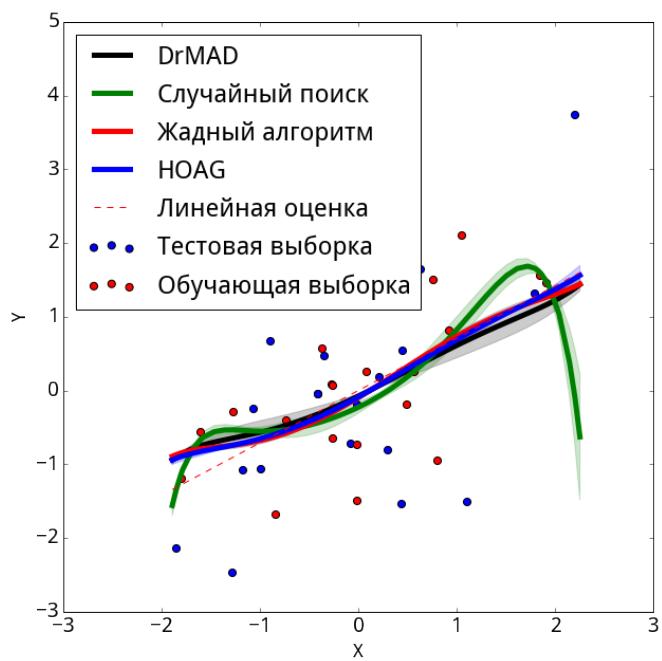
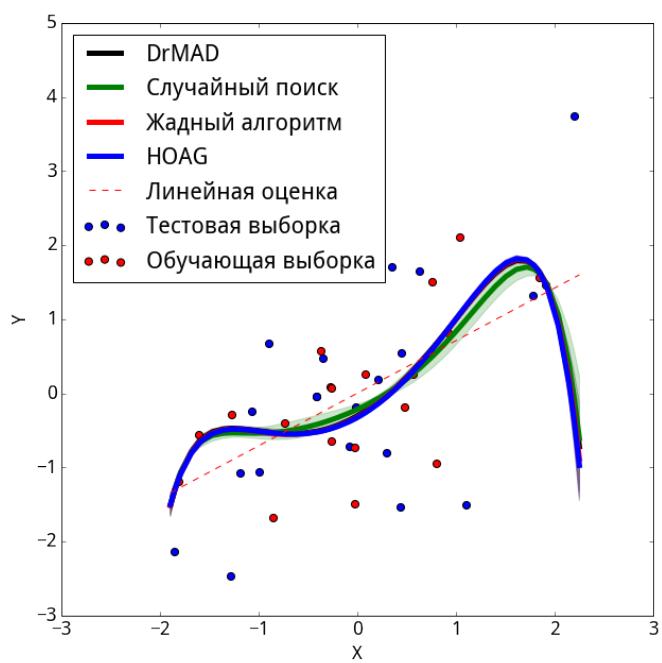


Рис. 3.2. Resulting models for the synthetic dataset: a — cross-validation, b — evidence lower bound

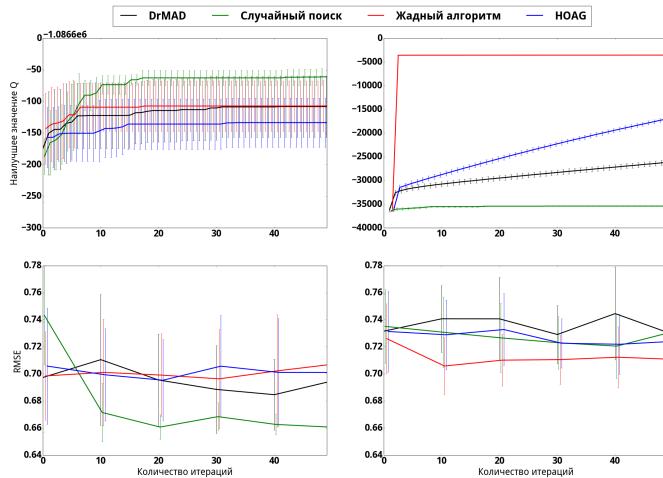


Рис. 3.3. WISDM, best validation value  $\hat{Q}$  and RMSE for cross-validation (left) and evidence lower bound (right)

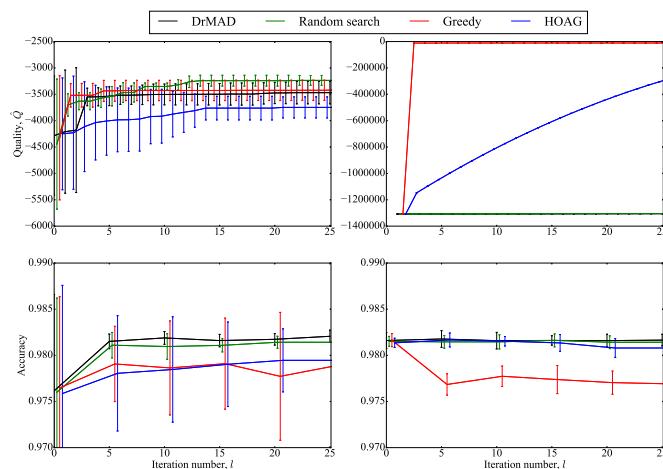


Рис. 3.4. MNIST, best validation value  $\hat{Q}$  and Accuracy for cross-validation (left) and evidence lower bound (right)

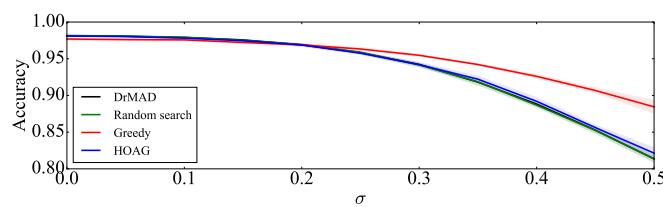


Рис. 3.5. MNIST, Model accuracy with noise in the Test dataset. The hyperparameters were optimized with evidence lower bound criterion.

## Глава 4

### Выбор субоптимальной структуры модели

В данной главе рассматривается задача выбора структуры модели глубокого обучения. Требуется предложить метод выбора модели субоптимальной сложности, позволяющий проводить выбор модели в нескольких режимах (ELBO, AddDel, полный перебор, оптимизация без регуляризации и с регуляризацией).

Решается задача нахождения оптимальной структуры. Предлагается ввести предположения о распределениях параметров и структуры модели. Проводится градиентная оптимизация параметров и гиперпараметров модели на основе байесовского вариационного вывода. В качестве оптимизируемой функции для гиперпараметров модели предлагается обобщенная функция правдоподобия. Показано, что данная функция позволяет проводить оптимизацию несколькими алгоритмами: последовательным добавлением и удалением параметров, полным перебором, а также максимизацией нижней оценки правдоподобия модели. Решается двухуровневая задача оптимизации: на первом уровне проводится оптимизация нижней оценки правдоподобия модели по вариационным параметрам модели. На втором уровне проводится оптимизация гиперпараметров модели.

#### 4.1. Постановка задачи выбора структуры модели

Задана выборка

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}, i = 1, \dots, m, \quad (4.1)$$

состоящая из множества пар «объект-метка»

$$\mathbf{x}_i \in \mathbf{X} \subset \mathbb{R}^n, \quad y_i \in \mathbf{Y} \subset \mathbb{Y}.$$

Метка  $y$  объекта  $\mathbf{x}$  принадлежит либо множеству:  $y \in \mathbb{Y} = \{1, \dots, Z\}$  в случае задачи классификации, где  $Z$  — число классов, либо некоторому подмножеству вещественных чисел  $y \in \mathbb{Y} \subseteq \mathbb{R}$  в случае задачи регрессии. Положим, что пары объект  $(\mathbf{x}, y)$  являются реализацией некоторой случайно величины и порождены независимо.

Пусть задано семейство моделей глубокого обучения  $\mathfrak{F}$ . Пусть значения каждого структурного параметра  $\boldsymbol{\gamma}^{i,j}$  лежат на симплексе  $\Delta^{K^{i,j}-1}$ . Пусть для каждого структурного параметра  $\boldsymbol{\gamma}^{i,j} \in \Gamma$  определено параметрическое априорное распределение  $p(\boldsymbol{\gamma}^{i,j}), \mathbf{m}^{i,j}, c_{\text{temp}})$ , где где  $\mathbf{m}^{i,j}$  — параметр средних,  $c_{\text{temp}}$  — температура (или концентрация) распределения. Перечислим свойства, которыми должно обладать данное распределение:

1.  $p(\boldsymbol{\gamma}^{i,j})$  является непрерывным на симплексе  $\Delta^{K^{i,j}-1}$ .
2. При устремлении температуры к бесконечности распределение сходится к равномерному:  $\lim_{c_{\text{temp}} \rightarrow \infty} p(\boldsymbol{\gamma}^{i,j}), \mathbf{m}^{i,j}, c_{\text{temp}}) = \mathcal{U}(\Delta^{K^{i,j}-1})$ .
3. При устремлении температуры к нулю распределение сходится к сингулярному распределению следующего вида:  $\lim_{c_{\text{temp}} \rightarrow 0} p(\boldsymbol{\gamma}_k^{i,j}) = m_k^{i,j}$ .

**Утверждение 3.** Перечисленными свойствами обладают распределения Дирихле и Гумбель-софтмакс.

Обозначим через  $S$  сюръективное отношение между параметром модели  $w \in \mathbf{W}$  и весами  $\gamma$  базовых функцией  $\mathbf{g}$ , определенное по следующему правилу:  
Если  $w \in \mathbf{W}$  является параметром функции  $\mathbf{g}_k^{i,j}$ , где  $(i, j) \in E$ , то  $S(w) = \gamma_k^{i,j}$ .  
Априорное распределение параметров зададим следующим образом:

$$\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{A}^{-1} \otimes S(\mathbf{W})).$$

где  $\mathbf{A}$  — диагональная матрица с положительными элементами на диагонали.  
Пусть также определено правдоподобие выборки  $p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma})$ .

Задача выбора структуры модели предполагает поиск значений гиперпараметров модели  $\mathbf{A}, \mathbf{m}$  доставляющих максимум правдоподобия модели:

$$\arg \max_{\mathbf{A}, \mathbf{m}} p(\mathbf{y}|\mathbf{X}, \mathbf{A}, \mathbf{m}, c_{\text{temp}}), \quad (4.2)$$

а также соответствующие параметры и структуру модели:

$$\arg \max_{\mathbf{W}, \boldsymbol{\Gamma}} p(\mathbf{W}, \boldsymbol{\Gamma}|\mathbf{X}, \mathbf{y}, \mathbf{A}, \mathbf{m}, c_{\text{temp}}). \quad (4.3)$$

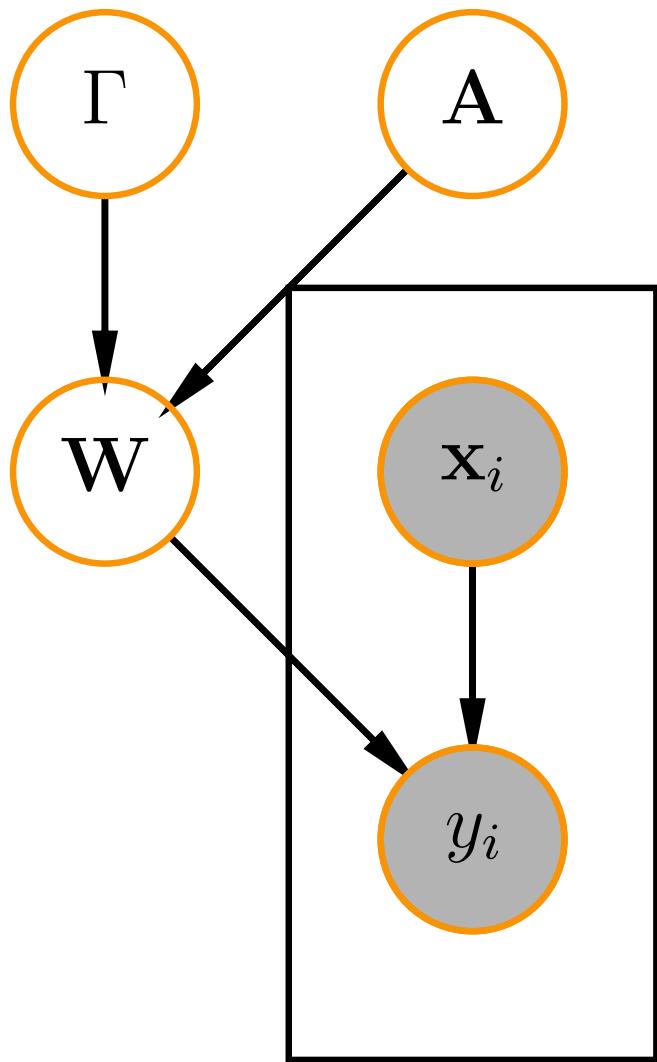


Рис. 4.1. Plate notation

TODO: схема

Для определения возможных значений структуры  $\Gamma$  введем следующие обозначения. Обозначим через  $\Delta(\Gamma)$  множество точек, лежащих внутри произведения симплексов, определенных на структурных параметрах

$$\Delta(\Gamma) = \prod_{i,j \in E} \Delta^{K^{i,j}-1}.$$

Обозначим через  $\bar{\Delta}(\Gamma)$  только те точки  $\Delta(\Gamma)$ , которые принадлежат вершинам соответствующих симплексов.

Докажем теорему об оптимальности решения задачи (4.3), лежащего на вершинах симплексов  $\times_{(i,j) \in E} \Delta^{K^{i,j}-1}$ .

### Теорема.

Пусть  $\Gamma_1$  и  $\Gamma_2$  — реализации  $\Gamma$ , такие что:

- $\Gamma_1 \in \bar{\Delta}(\Gamma)$ .
- $\Gamma_2 \notin \bar{\Delta}(\Gamma)$ .

Тогда для любых положительно определенных матриц  $\mathbf{A}_1$  и  $\mathbf{A}_2$  и векторов  $\mathbf{m}_1, \mathbf{m}_2, \min(\mathbf{m}_1) > 0$  справедлива следующее отношение апостериорных вероятностей:

$$\lim_{c_{\text{temp}} \rightarrow 0} \frac{p(\Gamma_2, \mathbf{W}_2 | \mathbf{y}, \mathbf{X}, \mathbf{A}_1, \mathbf{m}_2, c_{\text{temp}})}{p(\Gamma_1, \mathbf{W} | \mathbf{y}, \mathbf{X}, \mathbf{A}_1, \mathbf{m}_1, c_{\text{temp}})} = \infty.$$

### Доказательство.

По свойству априорного распределения структурных параметров

$$p\left(\lim_{c_{\text{temp}} \rightarrow 0} \gamma_k^{i,j} = 1\right) = m_k^{i,j}.$$

Тогда:

$$p\left(\lim_{c_{\text{temp}} \rightarrow 0} \boldsymbol{\gamma}^{i,j} \in \bar{\Delta}^{K^{i,j}}\right) = 1.$$

Тогда апостериорная вероятность  $\Gamma$  : в пределе равняется нулю, если структура не лежит на произведении вершин симплекса.

$$p(\Gamma_2, \mathbf{W} + 2 | \mathbf{y}, \mathbf{X}, \mathbf{A}_2, \mathbf{m}_2, c_{\text{temp}}) \propto p(\Gamma)p(\mathbf{y} | \Gamma, \mathbf{W}, \mathbf{X}, \mathbf{A}_1, \mathbf{m}) \rightarrow 0,$$

$$p(\Gamma_1, \mathbf{W}_1 | \mathbf{y}, \mathbf{W}, \mathbf{X}, \mathbf{A}_1, \mathbf{m}_1, c_{\text{temp}}) \propto p(\Gamma)p(\mathbf{y} | \Gamma, \mathbf{W}, \mathbf{X}, \mathbf{A}_1, \mathbf{m}) \rightarrow C,$$

где  $C$  — константа, большая нуля, т.к.  $\min(\mathbf{m}_1) > 0$ . что и требовалось доказать.

### Решение задачи с помощью вариационного вывода

В общем виде получение значения интеграла (2.3) является вычислительно сложной процедурой. В качестве приближенного значения интеграла используется вариационную верхнюю оценку правдоподобия модели.

Будем приближать неизвестное апостериорное распределение апостериорные распределение параметрическим распределением  $q$  с параметрами  $\boldsymbol{\theta}$ . Разница между верхней оценкой (3.11) и правдоподобием модели (2.3) определяется дивергенцией между вариационным распределение  $q$  и апостериорным распределением  $p(\mathbf{W}, \Gamma | \mathbf{y}, \mathbf{X}, \mathbf{A}, \mathbf{m}, c_{\text{temp}})$ .

Зададим вариационное распределение  $q$  следующим образом. Факторизуем  $q$  на два распределения:

$$q = q_{\mathbf{W}} q_{\Gamma} : q_{\mathbf{W}} \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}), \quad q_{\Gamma} = \prod_{(i,j) \in E} q_{\gamma}^{i,j}, \quad q_{\gamma} \sim \mathcal{GS}(\mathbf{m}_q^{i,j}, c_q).$$

Обозначим за  $\mathbf{m}_q$  конкатенацию всех векторов средних  $\mathbf{m}_q^{i,j}$ .

Для получения значения  $\log_q p(\mathbf{y}|\mathbf{X}, \mathbf{A}, \mathbf{m}, c_{\text{temp}})$  используя следующие методы сэмплирования:

$$\begin{aligned} \mathbb{E}_q \log p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c_{\text{temp}}) &\approx \frac{1}{N} \sum_{u=1}^N \log p(\mathbf{y}|\mathbf{X}, \hat{\mathbf{W}}_u, \hat{\boldsymbol{\Gamma}}_u, \mathbf{A}, \mathbf{m}, c_{\text{temp}}), \\ D_{KL}(q||p(\mathbf{W}, \boldsymbol{\Gamma}|\mathbf{A}, \mathbf{m}, c_{\text{temp}})) &= D_{KL}(q_{\boldsymbol{\Gamma}}||p(\boldsymbol{\Gamma}|\mathbf{m}, c_{\text{temp}}) + D_{KL}(q_{\mathbf{W}||p(\mathbf{W}|\mathbf{A})} \approx \\ &\quad \frac{1}{N} \sum_{u=1}^N (\log q_{\boldsymbol{\Gamma}}(\hat{\boldsymbol{\Gamma}}_u) - \log p(\hat{\boldsymbol{\Gamma}}_u) + 0.5(\text{tr}(\hat{\mathbf{S}}_u(\mathbf{W})\mathbf{A}\mathbf{A}_q^{-1}) + \\ &\quad + \boldsymbol{\mu}^T \hat{\mathbf{S}}_u(\mathbf{W})\mathbf{A}\boldsymbol{\mu} - |\mathbf{W}| + \log \det \hat{\mathbf{S}}_u(\mathbf{W})\mathbf{A} - \log \det \mathbf{A}_q)), \end{aligned}$$

где  $N$  — количество реализаций случайных величин,  $\hat{\boldsymbol{\Gamma}}_u, \hat{\mathbf{W}}_u$  — реализации случайных величин,  $\hat{\mathbf{S}}_u(\mathbf{W})$  — соответствие между параметрами и реализацией весов базовых функций.

Сэмплирование происходит следующим образом:

$$\begin{aligned} \hat{\mathbf{W}} &= \boldsymbol{\mu} + \varepsilon \mathbf{A}_q, \quad \varepsilon \in \mathcal{N}(\mathbf{0}, \mathbf{1}), \\ \hat{\boldsymbol{\gamma}}_k &= \frac{\exp((\log m_k + a_k)/c)}{\sum_{i=1}^K (\log m_i + a_i)/c}, \quad \mathbf{a} \in -\log(\log(\mathcal{U}(0, 1)^K)). \end{aligned}$$

Численную оценку, полученную описанным выше способом будет обозначать как

$$\hat{\log}_q p(\mathbf{y}|\mathbf{X}, \mathbf{A}, \mathbf{m}, c_{\text{temp}}) = \hat{\mathbb{E}}_q \log p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c_{\text{temp}}) - \hat{D}_{KL}(q||p(\mathbf{W}, \boldsymbol{\Gamma}|\mathbf{A}, \mathbf{m}, c_{\text{temp}}))$$

TODO: plate-notation.

Докажем теорему о дискретности задачи оптимизации вариационной оценки в предельном случае.

**Теорема.** Пусть  $c = c_{\text{temp}}$ . Для любых значений ковариационных матриц  $\mathbf{A}, \mathbf{A}_q$ , любого вектора  $\boldsymbol{\mu}_q$  существуют такие точки  $\mathbf{m}_q^1 \in \times_{(i,j) \in E} \bar{\Delta}^{K^{i,j}-1}, \mathbf{m}^1 \in \times_{(i,j) \in E} \bar{\Delta}^{K^{i,j}-1}$  на вершинах симплексов структуры  $\boldsymbol{\Gamma}$ , что для любой точки  $\mathbf{m}_q^2 \in \times_{(i,j) \in E} \Delta^{K^{i,j}-1}$  и  $\mathbf{m}^2 \in \times_{(i,j) \in E} \Delta^{K^{i,j}-1}$  внутри симплексов справедливо выражение:

$$\lim_{c_{\text{temp}} \rightarrow 0} \frac{\log \hat{p}_{q_{\mathbf{W}} q_{\boldsymbol{\Gamma}}^2}(\mathbf{y}|\mathbf{X})}{\log \hat{p}_{q_{\mathbf{W}} q_{\boldsymbol{\Gamma}}}(\mathbf{y}|\mathbf{X})} \geq 1, \quad \text{где } q_{\boldsymbol{\Gamma}}^1 = \max_c q_{\boldsymbol{\Gamma}}(\mathbf{m}_q^1, c).$$

**Доказательство.** По свойству предельного распределения  $\mathcal{GS}$  задача сводится к задаче с сингулярным распределением на структурах. Расписав  $\log_q p(\mathbf{y}|\mathbf{X}, \mathbf{A}, \mathbf{m}, c_{\text{temp}})$  через двойную сумму находим максимальный элемент.

## 4.2. Обобщенная постановка задачи

Определим основные величины, которые характеризуют сложность модели.

**Определение** Параметрической сложностью  $C_{\mathbf{W}}$  модели назовем наименьшую дивергенцию вариационного распределения при условии заданного априорного распределения параметров:

$$C_{\mathbf{W}} = \arg \min_{\mathbf{A}} D_{\text{KL}}(q|p(\mathbf{W}, \boldsymbol{\Gamma}|\mathbf{A}, \mathbf{m}, \mathbf{c}_{\text{temp}})).$$

**Определение** Структурной сложностью  $C_{\boldsymbol{\Gamma}}$  модели назовем энтропию распределения структуры:

$$C_{\boldsymbol{\Gamma}} = -\mathbb{E}_{q_{\boldsymbol{\Gamma}}} \log q_{\boldsymbol{\Gamma}},$$

где  $q_{\boldsymbol{\Gamma}}$  — вариационное распределение структуры модели.

В силу многоэкстремальности задачи (4.5), оптимизация параметров вариационных распределений  $\boldsymbol{\theta}$  и априорных распределений  $\mathbf{h}$  должна позволять не только находить локальный оптимум задачи (4.5), но и использовать ряд эвристических алгоритмов, таких как снижение и наращивание сложности модели. Сформулируем основные требования к оптимизационной задаче и оптимизируемым функционалам:

1. Оптимизируемые функции  $L, Q$  должны быть дифференцируемы.
2. Распределение параметров модели, являющееся решением задачи оптимизации должно доставлять максимум апостериорной вероятности в некоторой окрестности.
3. Степень регуляризации структуры  $\boldsymbol{\Gamma}$  и параметров  $\mathbf{W}$  должна быть контролируемой.
4. Решение задачи оптимизации должно являться локально-оптимальным для вариационной оценки (3.11).
5. Оптимизация должна позволять варьировать параметрическую сложность модели  $C_{\mathbf{W}}$ .
6. Оптимизация должна позволять варьировать структурную сложность  $\boldsymbol{\Gamma}$  модели.
7. Оптимизация должна позволять проводить полный перебор структуры  $\boldsymbol{\Gamma}$ .

Сформулируем задачу как двухуровневую задачу оптимизации. обозначим через  $\boldsymbol{\theta}$  оптимизируемые на первом уровне величины. обозначим через  $\mathbf{h}$  величины, оптимизируемые на втором уровне. Положим  $\boldsymbol{\theta}$  равным параметрам распределений  $q_{\mathbf{W}}, q_{\boldsymbol{\Gamma}} : \boldsymbol{\theta} = [\boldsymbol{\mu}_q, \mathbf{A}_q, \mathbf{m}_q, c]^T$ . Положим  $\mathbf{h} = [\mathbf{A}, \mathbf{m}]$ .

обозначим через  $L$  функцию потерь:

$$L = c_{\text{reg}} \mathbb{E}_q \log p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c_{\text{temp}}) - D_{\text{KL}}(q_{\boldsymbol{\Gamma}}||p(\boldsymbol{\Gamma})) - D_{\text{KL}}(q_{\mathbf{W}}||p(\mathbf{w})), \quad (4.4)$$

где  $c_{\text{reg}}$  — коэффициент регуляризации регуляризации структуры  $\boldsymbol{\Gamma}$  и параметров  $\mathbf{W}$  априорным распределением.

**Лемма.** Пусть  $\mathbf{A}_q$  фиксирована и близка к нулю,  $c_{\text{reg}} = 1$ . Тогда максимизация  $L$  эквивалентна оптимизации апостериорной вероятности параметров при  $c \rightarrow 0$ .

**Доказательство.**  $L = \mathbb{E}_q \log p(\mathbf{y} | \mathbf{X}, \mathbf{W}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c_{\text{temp}}) - D_{KL}(q || p(\mathbf{W}, \boldsymbol{\Gamma} | \mathbf{A}, \mathbf{m}, c_{\text{temp}}))$ . Полагая ковариационную матрицу близкой к нулю

$$\mathbb{E}_q \log p(\mathbf{y} | \mathbf{X}, \mathbf{W}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c_{\text{temp}}) \approx \log p(\mathbf{y} | \mathbf{X}, \boldsymbol{\mu}_q, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c_{\text{temp}})$$

$$D_{KL}(q || p(\mathbf{W}, \boldsymbol{\Gamma} | \mathbf{A}, \mathbf{m}, c_{\text{temp}})) = \frac{1}{N} \sum_{u=1}^N (\log q_{\boldsymbol{\Gamma}}(\hat{\boldsymbol{\Gamma}}_u) - \log p(\hat{\boldsymbol{\Gamma}}_u) + 0.5(\boldsymbol{\mu}^T \hat{\mathbf{S}}_u(\mathbf{W}) \mathbf{A} \boldsymbol{\mu} - |\mathbf{W}| + \log \det \hat{\mathbf{S}}_u(\mathbf{W}) \mathbf{A})).$$

Следующая теорема говорит о том, что варьируя  $c_{\text{reg}}$  мы проводим оптимизацию, ассимптотически аналогичную оптимизации выборки из того же распределения, но другой мощности.

**Теорема.** Пусть  $c_{\text{reg}} > 0$ ,  $c_{\text{reg}} m \in \mathbb{N}$ . Тогда функция  $L$  сходится почти наверно к вариационной нижней оценке правдоподобия для произвольной подвыборки  $\hat{\mathfrak{D}}$  мощностью  $m_0 = \frac{m}{c_{\text{reg}}}$ , поделенной на данную константу.

**Доказательство.** Рассмотрим произвольную подвыборку  $\hat{\mathfrak{D}}$  мощностью  $m_0$ . Нижняя оценка правдоподобия модели для подвыборки имеет вид:

$$\mathbb{E}_{q_w, q_\gamma} \log p(\hat{\mathbf{y}} | \hat{\mathbf{X}}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) - D_{KL}(q_\gamma || p(\boldsymbol{\Gamma})) - D_{KL}(q_w || p(\mathbf{w})).$$

$$\log p(\hat{\mathbf{y}} | \hat{\mathbf{X}}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) = \sum_i \log p(\hat{\mathbf{y}}_i | \hat{\mathbf{x}}_i, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) \xrightarrow[m \rightarrow \infty]{\text{П.Н.}} m_0 \mathbb{E} \log p(\mathbf{y} | \mathbf{x}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c)$$

Таким образом, ассимптотическая формула вариационной нижней оценки правдоподобия для подвыборки мощностью  $m_0$  выглядит следующим образом:

$$m_0 \mathbb{E} \log p(\mathbf{y} | \mathbf{x}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) - D_{KL}(q_\gamma || p(\boldsymbol{\Gamma})) - D_{KL}(q_w || p(\mathbf{w})).$$

Домножив на выражение на  $\frac{m}{m_0}$  получаем ассимптотику для  $L$ , что и требовалось доказать.

Пусть  $Q$  — валидационная функция:

$$Q = c_{\text{train}} \mathbb{E}_q \log p(\mathbf{y} | \mathbf{X}, \mathbf{W}, \boldsymbol{\Gamma}, \mathbf{A}^{-1}, c_{\text{prior}}) - c_{\text{prior}} D_{KL}(p(\mathbf{W}, \boldsymbol{\Gamma} | \mathbf{A}^{-1}, \mathbf{m}, c_{\text{temp}}) || q(\mathbf{W}, \boldsymbol{\Gamma})) - c_{\text{comb}} \sum_{p' \in \mathbf{P}} D_{KL}(\boldsymbol{\Gamma} | p') \rightarrow \max,$$

где  $\mathbf{P}$  — множество (возможно пустое) распределений на структуре модели,  $c_{\text{prior}}$  — коэффициент регуляризации параметрической сложности модели,  $c_{\text{comb}}$  — коэффициент перебора.

Сформулируем задачу поиска оптимальной модели как двухуровневую задачу.

$$\hat{\mathbf{h}} = \arg \max_{\mathbf{h} \in \mathbb{R}^h} Q(T^\eta(\boldsymbol{\theta}_0, \mathbf{h})), \quad (4.5)$$

где  $T$  — оператор оптимизации, решающий задачу оптимизации:

$$L(T^\eta(\boldsymbol{\theta}_0, \mathbf{h})) \rightarrow \max.$$

**Теорема.** Пусть  $D_{KL}(q_w | p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \mathbf{A}, \mathbf{m}, c)) = 0, D_{KL}(q_\gamma | p(\Gamma | \mathbf{y}, \mathbf{X}, \mathbf{A}, \mathbf{m}, c)) = 0$ , пусть  $c_{\text{prior}} = 1, c_{\text{reg}} = 1, c_{\text{comb}} = 0$ . Тогда оптимизация (4.5) эквивалентна оптимизации (2.3).

**Доказательство.** При соблюдении условий теоремы оптимизация вариационной оценки эквивалента оптимизации правдоподобия модели. При  $c_{\text{prior}} = 1, c_{\text{reg}} = 1, c_{\text{comb}} = 0$ , функция  $Q$  становится равной вариационной нижней оценке. Таким образом, двухуровневая оптимизация становится эквивалентной оптимизации правдоподобия модели по  $\mathbf{A}, \mathbf{m}$ , что и требовалось доказать.

### 4.3. Анализ предложенного метода выбора структуры модели

обозначим через  $F(c_{\text{reg}}, c_{\text{train}}, c_{\text{prior}}, c_{\text{comb}}, \mathbf{P}, c_{\text{temp}})$  множество экстремумов функции  $L$  при решении задачи двухуровневой оптимизации.

**Теорема.** Пусть  $\mathbf{f}_1 \in F(1, 1, c_{\text{prior}}^1, 0, \emptyset, c_{\text{temp}}), \mathbf{f}_2 \in F(1, 1, c_{\text{prior}}^2, 0, \emptyset, c_{\text{temp}})$ ,  $c_{\text{prior}}^1 < c_{\text{prior}}^2$ .

Пусть вариационные параметры моделей  $\mathbf{f}_1$  и  $\mathbf{f}_2$  лежат в области  $\mathbf{U}$ , в которой соответствующие функции  $L$  и  $Q$  являются локально-выпуклыми.

Тогда модель  $\mathbf{f}_1$  имеет параметрическую сложность, не меньшую чем у  $\mathbf{f}_2$ .

$$C_{\text{param}}(\mathbf{f}_1) \geq C_{\text{param}}(\mathbf{f}_2).$$

**Доказательство.** обозначим через  $q^1, q^2$  — вариационные распределения моделей  $\mathbf{f}_1, \mathbf{f}_2$ ,  $p^1, p^2$  — априорные распределения моделей.

Отсюда справедливы следующие неравенства (по единственности точек экстремума  $L, Q$ ):

$$\mathsf{E}_{q^1} \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) - c_{\text{prior}}^1 D_{KL}(q^1 || p^1) - \mathsf{E}_{q^2} \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) + c_{\text{prior}}^1 D_{KL}(q^1 || p^1)$$

$$\mathsf{E}_{q^2} \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) - c_{\text{prior}}^2 D_{KL}(q^2 || p^2) - \mathsf{E}_{q^1} \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) + c_{\text{prior}}^2 D_{KL}(q^2 || p^2)$$

Складывая неравенства получим:

$$D_{KL}(q^1 || p^1) \geq D_{KL}(q^2 || p^2),$$

$$\mathsf{E}_{q^2} \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) \leq \mathsf{E}_{q^1} \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c).$$

С учетом полученных неравенств распишем доказываемое утверждение:

$$\begin{aligned} & \max_p (-D_{\text{KL}}(q^1 || p)) - \max_p (-D_{\text{KL}}(q^2 || p^2)) = \\ & \max_p (-c_{\text{prior}}^2 D_{\text{KL}}(q^1 || p) + \mathsf{E}_{q^1} \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) - \mathsf{E}_{q^1} \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c)) - \\ & - \max_p (-c_{\text{prior}}^2 D_{\text{KL}}(q^2 || p) + \mathsf{E}_{q^2} \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c) + \mathsf{E}_{q^2} \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{A}, \mathbf{m}, c)) \end{aligned}$$

ЧТО И Т.Д.

**Теорема.** Пусть  $\mathbf{f} \in F(1, 1, c_{\text{prior}}, 0, \emptyset, c_{\text{temp}})$ . При устремлении  $c_{\text{prior}}$  к бесконечности параметрическая сложность модели  $\mathbf{f}$  устремляется к нулю (или существует?):

$$\lim_{c_{\text{prior}} \rightarrow \infty} C_{\text{param}}(\mathbf{f}) = 0.$$

### Доказательство

В пределе:  $Q = D_{KL}$ .

Минимум достигается при совпадении параметров распределений:  $tu = 0$ .

Докажем существование решения  $L$ , которое удовлетворяет этому.

Рассмотрим значение  $L$  при  $A \rightarrow 0$ . Два случая: либо конечное значение, либо бесконечное.

Таким образом, калибруя  $A$  получаем значения, близкие к нулю.

Рассмотрим последовательность. Тогда  $\liminf - > 0$ .

Доказано.

**Теорема** Пусть для каждого ребра  $(i, j)$  семейства моделей  $\mathfrak{F}$  априорное распределение

$$p(\gamma_{i,j}) = \lim_{c_{\text{temp}} \rightarrow 0} \mathcal{GS}(c_{\text{temp}}).$$

Пусть  $c_{\text{reg}} > 0, c_{\text{train}} > 0, c_{\text{prior}} > 0$ . Пусть  $\mathbf{f} \in F(c_{\text{reg}}, c_{\text{train}}, c_{\text{prior}}, 0, \emptyset, c_{\text{temp}})$ . Тогда структурная сложность модели  $\mathbf{f}$  равняется нулю.

$$C_{\text{struct}}(\mathbf{f}) = 0.$$

**Доказательство 1.** Доказываем, что гипер-концентрация будет лежать на вершине

2. У нас получается, что  $D_{KL}$  будет конечным только в случае совпадения. (???)
3. Итого, получили.

**Теорема** Пусть  $\mathbf{f}_1 \in F(c_{\text{reg}}, c_{\text{train}}, c_{\text{prior}}, 0, \emptyset, c_{\text{temp}}^1), \mathbf{f}_2 \in \lim_{c_{\text{temp}}^2 \rightarrow \infty} F(c_{\text{reg}}, c_{\text{train}}, c_{\text{prior}}, 0, \emptyset, c_{\text{temp}}^2)$ . Пусть вариационные параметры моделей  $f_1$  и  $f_2$  лежат в области  $U$ , в которой соответствующие функции  $L$  и  $Q$  являются локально-выпуклыми. Тогда разница структурных сложностей моделей ограничена выражением:

$$C_{\text{struct}}(\mathbf{f}_1) - C_{\text{struct}}(\mathbf{f}_2) \leq \mathbb{E}_q^1 \log p(\mathbf{y} | \mathbf{X}, \mathbf{W}, \boldsymbol{\Gamma}, \mathbf{A}^{-1}, c_{\text{temp}}^1) - \mathbb{E}_q^2 \log p(\mathbf{y} | \mathbf{X}, \mathbf{W}, \boldsymbol{\Gamma}, \mathbf{A}^{-1}).$$

**Доказательство 0.** Доказываем равномерную сходимость.

1. расписываем неравенства вида:  $L_1 - DKL(q_1 | p_1) < L_2 - DKL(q_2 | p_1)$
2. Замечаем, что при стремлении к бесконечности гумбель превращается в равномерное
3. выражаем все в равномерном
4. замечаем, что  $D_{KL} = \text{Entropy} + \text{const}$  для равномерного

**Утверждение (очень предварительно).** Изменение  $c$  позволяет избежать ухода в локальный минимум.

**Утверждение (очень предварительно).** Изменение  $c_2$  позволяет избежать ухода в локальный минимум.

**Утверждение (очень предварительно).** Взаимосвязь структуры и параметров в prior позволяет получить «хорошие» модели.

**Утверждение (предварительно).** Пусть  $c_1 = c_2 = c_3 = 0$ . Пусть  $q_w \sim \mathcal{N}(\mathbf{0}, \sigma)$ ,  $\sigma \sim 0$ . Тогда оптимизация эквивалентна обычной оптимизации параметров с  $l_2$ -регуляризацией.

## Глава 5

### Анализ прикладных задач порождения и выбора моделей глубокого обучения

### Анализ прикладных задач порождения и выбора моделей глубокого обучения

#### 5.1. Выбор модели автокодировщика (Попова)

В качестве данных для проведения вычислительного эксперимента использовались данные WISDM [132], представляющие собой набор записей акселерометра мобильного телефона. Каждой записи соответствуют три координаты по осям акселерометра. Набор данных содержит записи движений для 6 классов переменной длины. При проведении вычислительного эксперимента из каждой записи использовались первые 200 сегментов. Т. к. выборка не сбалансирована, в нее добавлялись повторы записей классов, содержащих количество записей, меньшее чем у большего класса.

Основные эксперименты — исследование зависимости ошибки классификации от числа параметров и размера выборки — были проведены как с использованием инструментария на базе библиотеки Theano, так и с использованием инструментария на языке Matlab. Для оценки качества классификации была проведена процедура скользящего контроля [?] при соотношении числа объектов обучающей и контрольной выборки 3:1. Число нейронов на каждом слое задавалось из соотношения 10:6:3. При проведении процедуры скользящего контроля для каждого отсчета количества нейронов было произведено пять запусков. В эксперименте с использованием инструментария на базе Theano при обучении двухслойной нейронной сети проводился мультистарт [121], т. е. одновременный запуск обучения сети с 8 разными стартовыми значениями параметров для предотвращения возможного застревания алгоритма обучения в локальном минимуме. При оценке качества классификации выбиралась модель с наилучшими результатами. График зависимости ошибки классификации от числа используемых нейронов изображен на рис. 5.1.

Для оценки зависимости качества классификации от размера обучающей выборки была проведена кроссвалидация с фиксированным количеством объектов в обучающей выборке (25% исходной выборки) и переменным размером обучающей выборки. Число нейронов было установлено как 364:224:112. При проведении процедуры скользящего контроля для каждого отсчета было произведено пять запусков. График зависимости ошибки классификации от размера обучающей выборки представлен на рис. 5.2.

Для исследования скорости работы процесса обучения нейросети в зависимости от конфигурации Theano был сделан следующий эксперимент: проводилось обучение двухслойной нейросети на основе подсчитанных заранее парамет-

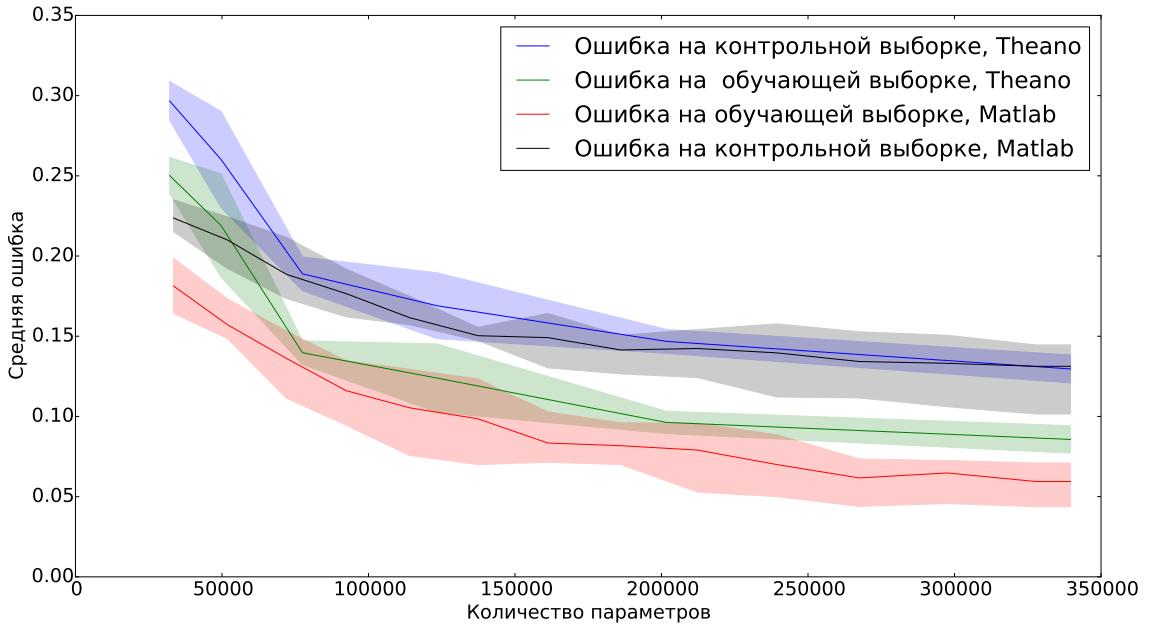


Рис. 5.1. Зависимость ошибки от числа нейронов

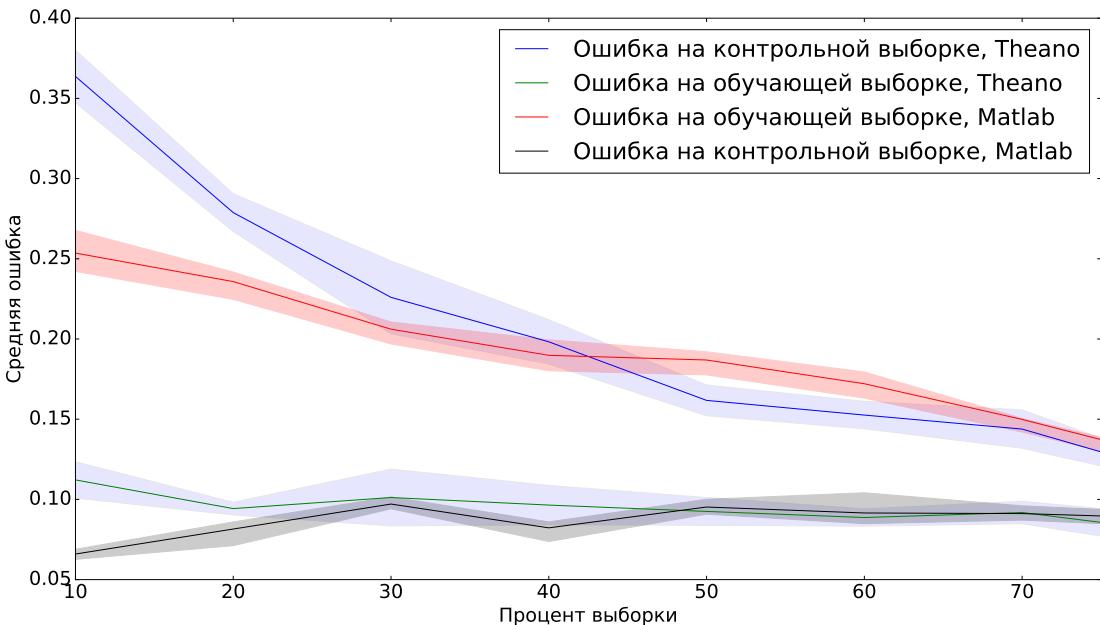


Рис. 5.2. Зависимость ошибки от размера обучающей выборки

ров ограниченной машины Больцмана (??) и автокодировщика (??). Обучение проходило за 100 итераций. При обучении алгоритм запускался параллельно с  $n$  разными стартовыми позициями,  $n \in \{1, \dots, 4\}$ . Число нейронов было установлено как 300:200:100. Запуск осуществлялся со следующими конфигурациями Theano:

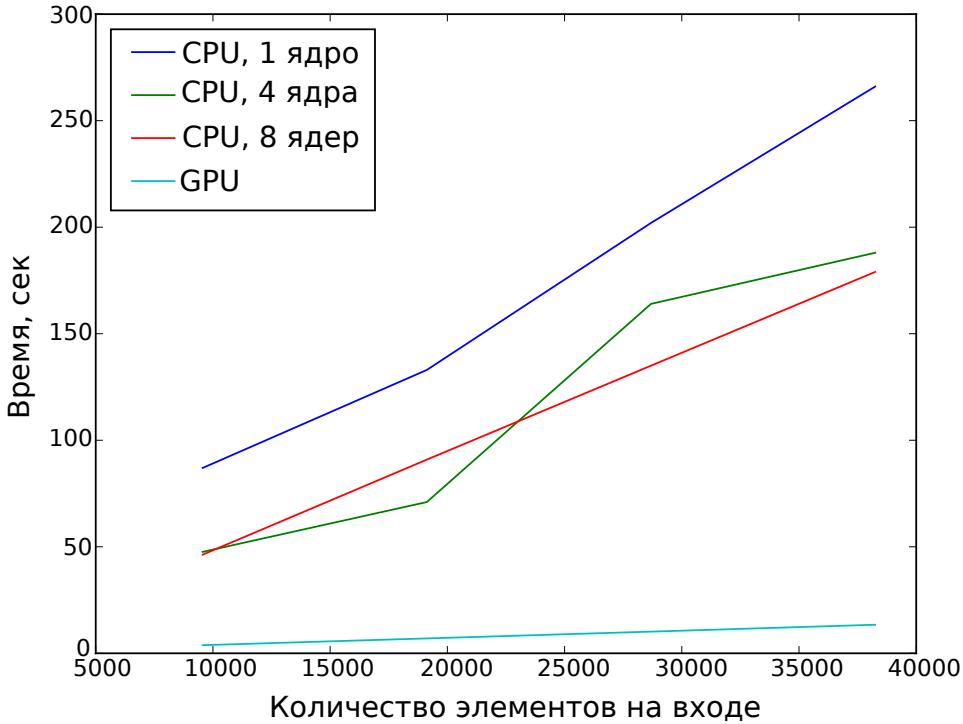


Рис. 5.3. Результаты эксперимента по исследованию скорости процесса обучения

- вычисление на центральном процессоре, задействовано одно ядро;
- вычисление на центральном процессоре, задействовано четыре ядра;
- вычисление на центральном процессоре, задействовано восемь ядер;
- вычисление на графическом процессоре.

Результаты эксперимента приведены на рис. 5.3. Как видно из графика, вычисление с использованием CUDA показывает значительное ускорение по сравнению с вычислением на центральном процессоре.

## 5.2. Модели парафраза (Смердов)

Цель эксперимента — проверка работоспособности предложенного алгоритма и сравнение результатов с ранее полученными. В качестве данных использовалась выборка SemEval 2015, состоящая из 8331 пары схожих и несхожих предложений. Слова преобразовывались в векторы размерности 50 при помощи алгоритма GloVe [?]. Для базовых алгоритмов тренировочная, валидационная и тестовая выборки составили 70%, 15% и 15% соответственно. Для рекуррентной нейронной сети, полученной вариационным методом, валидационная выборка отсутствовала, а тренировочная и тестовая выборки составили 85% и 15% соответственно. Критерием качества была выбрана F1-мера. В качестве базовых алгоритмов использовались линейная регрессия, метод ближайших соседей, решающее дерево и модификация метода опорных векторов SVC. Базовые ал-

горитмы взяты из библиотеки sklearn. Дополнительно были построены рекуррентная нейросеть с одним скрытым слоем [?] и нейросеть с одним скрытым слоем и вариационной оптимизацией параметров [?, ?].

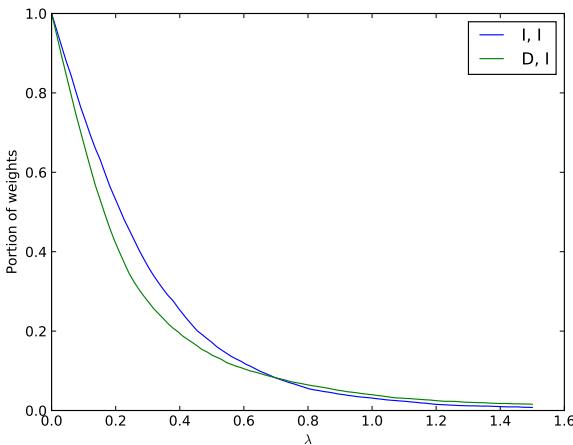


Рис. 5.4. Доля неудаленных параметров сети в зависимости от порогового значения  $\lambda$  для скалярного ( $I$ ) и диагонального ( $D$ ) вида апостериорной матрицы ковариаций

На рис. ?? и ?? представлена зависимость оценки правдоподобия  $L$  (??) от параметра  $\lambda$ . Для обоих случаев существует оптимальное значение  $\lambda$ , минимизирующее  $L$ ; модели с таким параметром будут оптимальными. На рис. ??, ??, ?? и ?? отображены зависимости качества модели от  $\lambda$  и доли выброшенных параметров. Видно, что даже при удалении большинства параметров из сети качество предсказаний меняется несущественно, что говорит о слишком большом числе параметров исходной модели.

Из рис. 5.4 видно, что при малых  $\lambda$  из сети с диагональной апостериорной матрицей ковариаций удаляется больше весов, а при больших  $\lambda$  — меньше, что говорит о лучшем отборе параметров такой моделью.

### 5.3. Прореживание модели (Грабовой)

Для анализа свойств предложенного алгоритма и сравнения его с существующими был проведен вычислительный эксперимент в котором параметры нейросети удалялись методами, которые были описаны в разделах 3.1—3.3 и методом Белсли.

В качестве данных использовались три выборки. Выборки Wine [?] и Boston Housing [?] — это реальные данные. Синтетические данные сгенерированы таким образом чтобы параметры сети были мультиколлинеарными. Генерация данных состояла из двух этапов. На первом этапе генерировался вектор параметров  $\mathbf{w}_{\text{synthetic}}$ :

$$\mathbf{w}_{\text{synthetic}} \sim \mathcal{N}(\mathbf{m}_{\text{synthetic}}, \mathbf{A}_{\text{synthetic}}), \quad (5.1)$$

$$\text{где } \mathbf{m}_{\text{synthetic}} = \begin{bmatrix} 1.0 \\ 0.0025 \\ \dots \\ 0.0025 \end{bmatrix}, \quad \mathbf{A}_{\text{synthetic}} = \begin{bmatrix} 1.0 & 10^{-3} & \dots & 10^{-3} & 10^{-3} \\ 10^{-3} & 1.0 & \dots & 0.95 & 0.95 \\ \dots & \dots & \dots & \dots & \dots \\ 10^{-3} & 0.95 & \dots & 0.95 & 1.0 \end{bmatrix}.$$

На втором этапе генерировалась выборка  $\mathfrak{D}_{\text{synthetic}}$ :

$$\mathfrak{D}_{\text{synthetic}} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \sim \mathcal{N}(\mathbf{1}, \mathbf{I}), y_i = x_{i0}, i = 1 \dots 10000\}. \quad (5.2)$$

В приведенном выше векторе параметров  $\mathbf{w}_{\text{synthetic}}$  для выборки  $\mathfrak{D}_{\text{synthetic}}$ , наиболее релевантным является первый параметр, а все остальные параметры являются нерелевантными. Матрица ковариации была выбрана таким образом, чтобы все нерелевантные параметры были зависимы и метод Белсли был максимально эффективен.

Таблица 5.1. Описание выборок

Выборка	Тип задачи	Размер выборки	Число признаков
Wine	классификация	178	13
Boston Housing	регрессия	506	13
Synthetic data	регрессия	10000	100

Для алгоритмов тренировочная и тестовая выборки составили 80% и 20% соответственно. Критерием качества прореживания служит процент параметров нейросети, удаление которого не влечет значимой потери качества прогноза. Также критерием качества служит устойчивость нейросети к зашумленности данных.

Качеством прогноза  $R_{\text{cl}}$  модели для задачи классификации является точность прогноза модели:

$$R_{\text{cl}} = \frac{\sum_{(\mathbf{x}, y) \in \mathfrak{D}} [f(\mathbf{x}, \mathbf{w}) = y]}{|\mathfrak{D}|}, \quad (5.3)$$

Качеством прогноза  $R_{\text{rg}}$  модели для задачи регрессии является среднеквадратическое отклонение результата модели от точного:

$$R_{\text{rg}} = \frac{\sum_{(\mathbf{x}, y) \in \mathfrak{D}} (f(\mathbf{x}, \mathbf{w}) - y)^2}{|\mathfrak{D}|}, \quad (5.4)$$

**Wine.** Рассмотрим нейронную сеть с 13 нейронами на входе, 13 нейронами в скрытом слое и 3 нейронами на выходе.

На рис. 5.5 показано как меняется точность прогноза  $R_{\text{cl}}$  при удалении параметров указанными методами. Из графика видно, что метод оптимального прореживания, вариационный метод и метод Белсли позволяют удалить  $\approx 80\%$  параметров и качество всех этих методов падает при удалении  $\approx 90\%$  параметров нейросети.

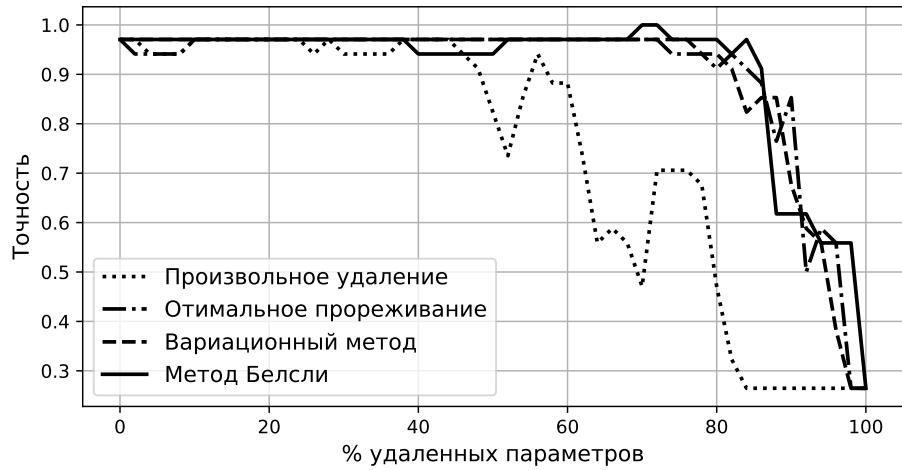


Рис. 5.5. Качество прогноза при удаление параметров на выборке Wine

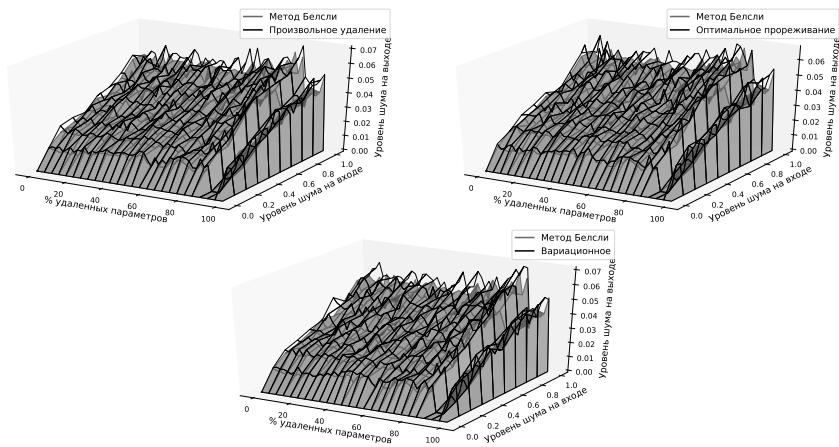


Рис. 5.6. Влияние шума в начальных данных на шум выхода нейросети на выборке Wine: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод

На рис. 5.6 показаны поверхности изменения уровня шума ответов нейросети при изменении процента удаленных параметров и уровня шума входных данных для разных методов прореживания. На графиках показано, что при удалении параметров нейросети методом Белсли шум меньше, чем при удалении параметров другими методами, на это указывает то что поверхность которая соответствует методу Белсли ниже других поверхностей.

**Boston Housing.** Рассмотрим нейронную сеть с 13 нейронами на входе, 39 нейронами в скрытом слое и одним нейроном на выходе.

На рис. 5.7 показано как меняется среднеквадратическое отклонение прогноза  $R_{rg}$  от точного ответа при удалении параметров указанными методами. График показывает, что метод Белсли является более эффективным, чем другие методы, так-как позволяет удалить больше параметров нейросети без потери качества.

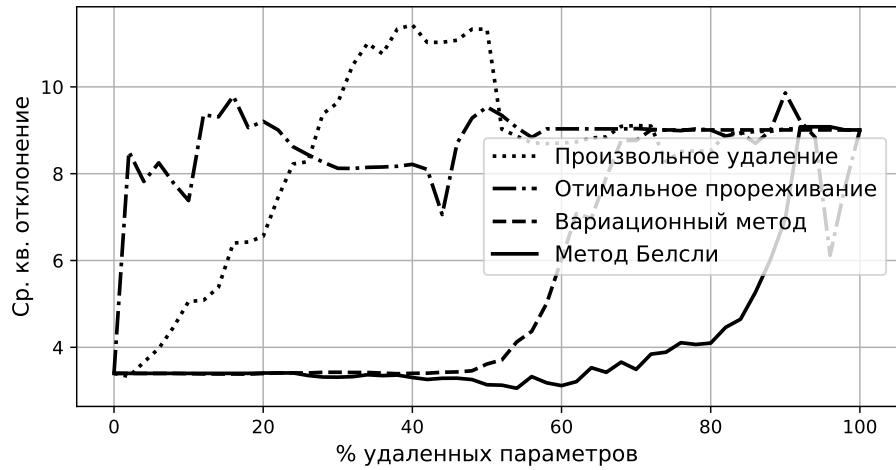


Рис. 5.7. Качество прогноза при удаление параметров на выборке Boston

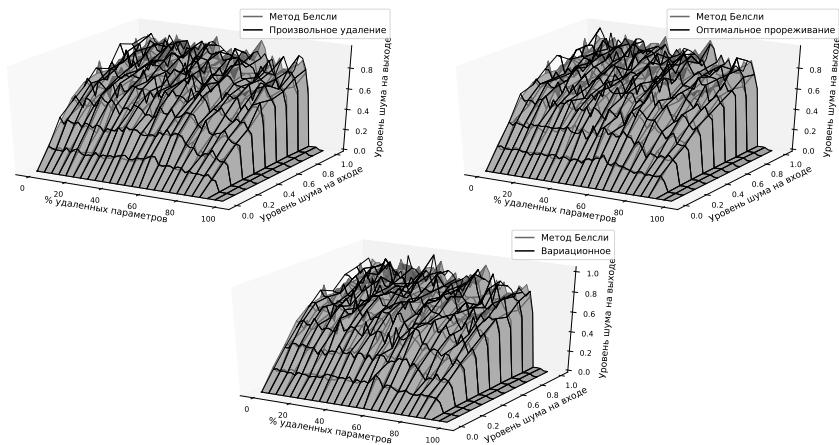


Рис. 5.8. Влияние шума в начальных данных на шум выхода нейросети на выборке Boston: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод

На рис. 5.8 показаны поверхности изменения уровня шума ответов нейросети при изменении процента удаленных параметров и уровня шума входных данных для разных методов прореживания. График показывает, что уровень шума всех методов одинаковый, так как поверхности всех методов находятся на одном уровне.

**Синтетические данные.** Рассмотрим нейронную сеть с 100 нейронами на входе и одним нейроном на выходе.

На рис. 5.9 показано как меняется среднеквадратическое отклонение прогноза от  $R_{rg}$  точного ответа при удалении параметров указанными методами. График показывает, что удаление параметров методом Белсли является более эффективным чем другие методы прореживания, так как качество прогноза нейросети улучшается при удалении шумовых параметров.

На рис. 5.10 показаны поверхности изменения уровня шума ответов нейро-

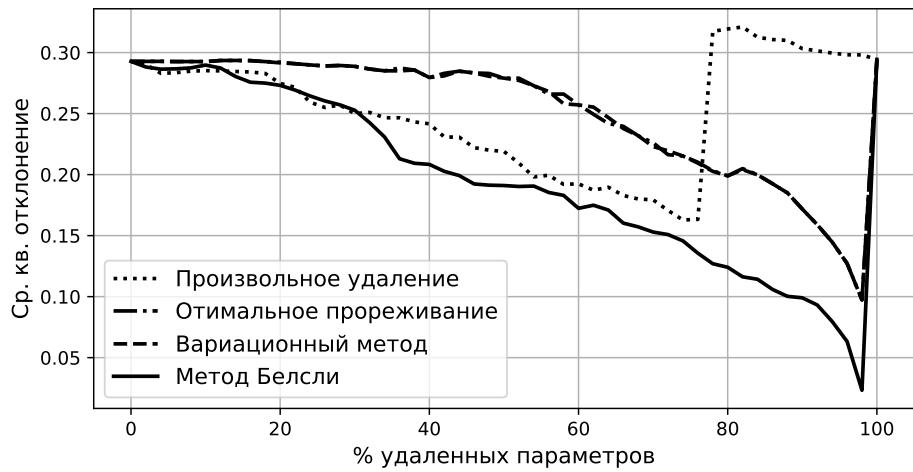


Рис. 5.9. Качество прогноза при удаление параметров на синтетической выборке

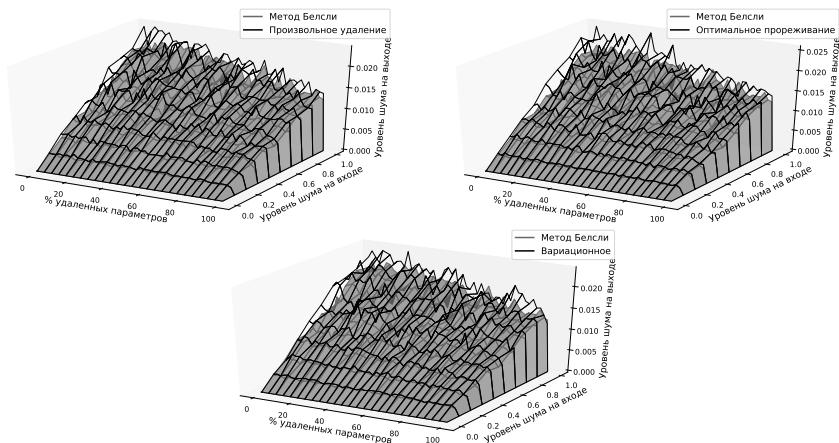


Рис. 5.10. Влияние шума в начальных данных на шум выхода нейросети на выборке Boston: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод

сети при изменении процента удаленных параметров и уровня шума входных данных для разных методов прореживания. На графиках показано, что при удалении параметров нейросети методом Белсли шум меньше, чем при удалении параметров другими методами, так как поверхность которая соответствует методу Белсли ниже других поверхностей.

## Заключение

В работе были предложены критерии оптимальной и субоптимальной сложности моделей глубокого обучения. Предложен алгоритм выбора субоптимальной модели, основанный на получении вариационной нижней оценки правдоподобия модели. Был предложен метод получения оценки, основанный на стохастическом градиентном спуске, позволяющий проводить выбор модели и оптимизацию модели единообразно. Исследованы свойства стохастического градиентного спуска, а также оценок правдоподобия, полученных с его использованием. Работа представленного алгоритма проиллюстрирована рядом выборок. Вычислительный эксперимент продемонстрировал значимое влияние априорного распределения на апостериорное распределение параметров модели. В силу многоэкстремальности оптимизируемых функций получение аналитических оценок для гиперпараметров модели является вычислительно сложным. В дальнейшем планируется исследовать применение предложенных алгоритмов для оптимизации гиперпараметров градиентными методами, представленными в [49].

The experiments showed that each algorithm can perform effectively and therefore the appropriate hyperparameter optimization method should rely on the amount of hyperparameters and the specific of the problem.

When dealing with small amount of hyperparameters the random search showed the best results since the search procedure can be employed more effectively than gradient-based optimization in low-dimensional hyperparameters space. For the high-dimensional hyperparameter space both the HOAG and greed algorithms showed good performance. The HOAG algorithm is more preferable if the model optimization problem is expensive. On the other hand we can schedule greed hyperparameter optimization to make it less expensive as in [52].

The DrMad algorithm showed rather poor results on the MNIST and WISDM datasets. Perhaps it is because of high learning rate  $\gamma$  we used in experiments. The large value of the learning rate can make the DrMad algorithm unstable. Two improvements can be proposed. We can use more stable optimization like Adam or AdaGrad for both parameter and hyperparameter optimization. The second improvement was proposed to develop in [70]: we can use more complicated parameter trajectory approximation to make it more similar to the original parameter trajectory. Opposing to the HOAG and greedy algorithms, the DrMad optimization has prerequisites for optimization not only hyperparameters but also the metaparameters, i.e. the parameters of the optimization procedure. The opportunity of such optimization using reversed differentiation was shown in [70].

The other interesting aspect of our experiments is the relation between the model error (RMSE or Accuracy) and the value of validation loss  $Q$ . The models obtained by the evidence lower bound showed higher errors than the models obtained using cross-validation on the MNIST and WISDM datasets. Nevertheless these models also showed greater stability when the noise was added to the Test datasets. The evidence

lower bound showed significantly better results on the synthetic dataset, when the amount of object in the Train dataset is small. Therefore we can conclude that the evidence lower bound usage is preferable when the model tend to overfit or when the cross-validation usage is too computationally expensive. In [39] note that the evidence lower bound optimization required more iterations for the convergence. In our experiments we used the same number of iterations both for the cross-validation and evidence lower bound. The more accurate iteration number calibration can improve the final quality of these models.

The paper analyzed the gradient-based hyperparameter optimization algorithms. We adapted the analyzed algorithms for general validation functions and evaluated their performance on the MNIST and WISDM datasets. Two model selection criteria were compared: the cross-validation and evidence lower bound.

The experiments showed that the gradient-based algorithms are effective when the number of hyperparameters is large. The results showed that models obtained using evidence lower bound have higher error rater than models obtained using cross-validation, but they are also more stable when the test dataset contain a lot of noise.

The authors implemenetd the algorithms as a toolbox available at [131]. The toolbox is developed in Python using Theano [81] and Numpy [138] libraries.

In the future we are planning to develop the analyzed algorithm and to extend gradient-based algorithms to optimize not only hyperparameters, but also the parameters of the model optimization. The other object of our research will be the difference between the cross-validation and evidence lower bound and the theoretical aspects of their properties for the models with large amount of parameters.

## **Заключение**

## Список основных обозначений

- $\mathbf{x}_i$  — вектор признакового описания  $i$ -го объекта  
 $y_i$  — метка  $i$ -го объекта  
 $\mathfrak{D}$  — выборка  
 $\mathbf{X}$  — матрица, содержащая признаковое описание объектов выборки  
 $\mathbf{y}$  — вектор меток объектов выборки  
 $m$  — количество объектов в выборке  
 $n$  — количество признаков в признаковом описании объекта  
 $\mathbb{X}$  — признаковое пространство объектов  
 $\mathbb{Y}$  — множество меток объектов  
 $Z$  — множество классов в задаче классификации  
 $(V, E)$  — граф со множеством вершин  $V$  и множеством ребер  $E$   
 $\mathbf{g}^{j,k}$  — вектор базовых функций для ребра  $(j, k)$   
 $K^{j,k}$  — мощность вектора базовых функций для ребра  $(j, k)$   
 $\text{agg}_v$  — функция агрегации для вершины  $v$ .  $\boldsymbol{\gamma}^{j,k}$  — структурный параметр для ребра  $(j, k)$   
 $\Delta^K$  — симплекс на  $K$  вершинах  
 $\bar{\Delta}^K$  — множество вершин симплекса на  $K$  вершинах  
 $\mathfrak{F}$  — семейство моделей  
 $\mathbf{W}$  — параметры модели  
 $\mathbb{W}$  — пространство параметров модели  
 $\Gamma$  — структура модели  
 $\leq_{\neq}$  — множество значений структуры модели  
 $\mathbf{h}$  — гиперпараметры модели  
 $\mathbb{H}$  — пространство гиперпараметров модели  
 $p(\mathbf{W}, \Gamma | \mathbf{h})$  — априорное распределение параметров и структуры модели  
 $p(\mathbf{W}, \Gamma | \mathbf{y}, \mathbf{X}, \mathbf{h})$  — апостериорное распределение параметров и структуры модели  
 $p(y, \mathbf{W}, \Gamma | \mathbf{x}, \mathbf{h})$  — вероятностная модель глубокого обучения  
 $p(y | \mathbf{X}, \mathbf{W}, \Gamma)$  — правдоподобие выборки  
 $p(y | \mathbf{X}, \mathbf{h})$  — правдоподобие модели  
 $q(\mathbf{W}, \Gamma)$  — аппроксимирующее распределение  
 $\boldsymbol{\theta} \in \mathbb{R}^u$  — оптимизируемые параметры модели  
 $L(\boldsymbol{\theta}, \mathbf{h}, \mathbf{X}, \mathbf{y})$  — функция потерь  
 $Q(\mathbf{h}, \boldsymbol{\theta}, \mathbf{X}, \mathbf{y})$  — валидационная функция  
 $T(\boldsymbol{\theta} | L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\beta})$  — оператор оптимизации  
 $\boldsymbol{\beta}$  — вектор метапараметров

## Список иллюстраций

1.1	Пример семейства моделей глубокого обучения: семейство описывает сверточную нейронную сеть. . . . .	11
1.2	Примеры ограничений для одного структурного параметра $\gamma,  \gamma  = 3$ . а) структурный параметр лежит на вершинах куба, б) структурный параметр лежит внутри куба, в) структурный параметр лежит на вершинах симплекса, г) структурный параметр лежит внутри симплекса. . . . .	13
1.3	Пример семейства моделей глубокого обучения: семейство описывает многослойную полносвязную нейронную сеть с одним скрытым слоем и нелинейной функцией активации $\sigma$ . . . . .	13
1.4	Пример итерации алгоритма AdaNet [16]. Рассматриваются две альтернативные модели: модель с углублением сети (соответствует занулению функции $f_2$ с использованием базовой функции $g_1^{2,3}$ ) и модель с расширением сети (соответствует базовой функции $g_0^{2,3}$ ). . . . .	23
1.5	Пример семейства моделей глубокого обучения, описываемый в [17]. Каждая подмодель $f_j$ является линейной комбинацией базовых функций: свертки и результата работы предыдущих подмоделей (англ. skip-connection). . . . .	24
1.6	Пример суперсети. Каждый путь из подмодели $f_0$ в конечную модель $f_8$ задает модель глубокого обучения. . . . .	27
1.7	Схема порождения вектора объектов $\mathbf{X}$ , представленная в [53]. . . . .	28
2.1	Аппроксимация распределения а) нормальным распределением, б) распределением, полученным с помощью градиентного спуска, в) с использованием стохастической динамики Ланжевена. . . . .	38
2.2	Псевдокод алгоритма получения вариационной нижней оценки правдоподобия модели с использованием градиентного спуска . . . . .	42
2.3	Возмущение выборки для однослойных нейросетей: а) Boston Housing, б) Protein, в) MSD. . . . .	47
3.1	An example of parameter update trajectories. The color displays the value of the validation function $Q$ . Greedy algorithm optimizes hyperparameter during the parameter optimization, therefore it has the light blue trajectory between the optimized green trajectory of HOAG algorithm and the dark blue trajectory of parameters without hyperparameter optimization. DrMAD uses a linearized dashed parameter trajectory during hyperparameter optimization procedure. . . . .	54
3.2	Resulting models for the synthetic dataset: а — cross-validation, б — evidence lower bound . . . . .	63

3.3	WISDM, best validation value $\hat{Q}$ and RMSE for cross-validation (left) and evidence lower bound (right) . . . . .	64
3.4	MNIST, best validation value $\hat{Q}$ and Accuracy for cross-validation (left) and evidence lower bound (right) . . . . .	64
3.5	MNIST, Model accuracy with noise in the Test dataset. The hyperparameters were optimized with evidence lower bound criterion.	64
4.1	Plate notation . . . . .	67
5.1	Зависимость ошибки от числа нейронов . . . . .	77
5.2	Зависимость ошибки от размера обучающей выборки . . . . .	77
5.3	Результаты эксперимента по исследованию скорости процесса обучения . . . . .	78
5.4	Доля неудаленных параметров сети в зависимости от порогового значения $\lambda$ для скалярного ( $I$ ) и диагонального ( $D$ ) вида апостериорной матрицы ковариаций . . . . .	79
5.5	Качество прогноза при удаление параметров на выборке Wine . .	81
5.6	Влияние шума в начальных данных на шум выхода нейросети на выборке Wine: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод . . . . .	81
5.7	Качество прогноза при удаление параметров на выборке Boston .	82
5.8	Влияние шума в начальных данных на шум выхода нейросети на выборке Boston: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод . . . . .	82
5.9	Качество прогноза при удаление параметров на синтетической выборке . . . . .	83
5.10	Влияние шума в начальных данных на шум выхода нейросети на выборке Boston: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод . . . . .	83

## **Список таблиц**

2.1	Описание выборок для экспериментов по выбору моделей . . . . .	46
2.2	Результаты эксперимента по выбору моделей . . . . .	46
3.1	Complexity and correctness of the analyzed algorithms . . . . .	50
3.2	Преимущества и недостатки рассматриваемых алгоритмов . . . . .	51
3.3	Основные свойства рассматриваемых алгоритмов . . . . .	59
3.4	Experiment results . . . . .	62
5.1	Описание выборок . . . . .	80

## Список использованных источников

1. *Grünwald Peter.* A Tutorial Introduction to the Minimum Description Length Principle // Advances in Minimum Description Length: Theory and Applications. — MIT Press, 2005.
2. *Bishop Christopher M.* Pattern Recognition and Machine Learning (Information Science and Statistics). — Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
3. *Salakhutdinov Ruslan, Hinton Geoffrey E.* Learning a Nonlinear Embedding by Preserving Class Neighbourhood Structure // Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07) / Ed. by Marina Meila, Xiaotong Shen. — Vol. 2. — Journal of Machine Learning Research - Proceedings Track, 2007. — Pp. 412–419. <http://jmlr.csail.mit.edu/proceedings/papers/v2/salakhutdinov07a/salakhutdinov07a.pdf>.
4. On the importance of initialization and momentum in deep learning / Ilya Sutskever, James Martens, George E. Dahl, Geoffrey E. Hinton // Proceedings of the 30th International Conference on Machine Learning (ICML-13) / Ed. by Sanjoy Dasgupta, David Mcallester. — Vol. 28. — JMLR Workshop and Conference Proceedings, 2013. — Май. — Pp. 1139–1147. <http://jmlr.org/proceedings/papers/v28/sutskever13.pdf>.
5. Approximation and learning by greedy algorithms / Andrew R. Barron, Albert Cohen, Wolfgang Dahmen, Ronald A. DeVore // *Ann. Statist.* — 2008. — 02. — Vol. 36, no. 1. — Pp. 64–94. <http://dx.doi.org/10.1214/009053607000000631>.
6. *Tzikas Dimitris, Likas Aristidis.* An Incremental Bayesian Approach for Training Multilayer Perceptrons // Artificial Neural Networks – ICANN 2010: 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings, Part I / Ed. by Konstantinos Diamantaras, Wlodek Duch, Lazaros S. Iliadis. — Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. — Pp. 87–96. [http://dx.doi.org/10.1007/978-3-642-15819-3\\_12](http://dx.doi.org/10.1007/978-3-642-15819-3_12).
7. *Tipping Michael E.* Sparse Bayesian Learning and the Relevance Vector Machine // *J. Mach. Learn. Res.* — 2001. — Сентябрь. — Vol. 1. — Pp. 211–244. <http://dx.doi.org/10.1162/15324430152748236>.
8. *Cun Yann Le, Denker John S., Solla Sara A.* Optimal Brain Damage // Advances in Neural Information Processing Systems. — Morgan Kaufmann, 1990. — Pp. 598–605.
9. *Попова М. С., Стрижов В. В.* Выбор оптимальной модели классификации физической активности по измерениям акселерометра // Информатика и ее применение. — 2015. — Т. 9(1). — С. 79–89. <http://strijov.com/papers/Popova2014OptimalModelSelection.pdf>.
10. Learning both Weights and Connections for Efficient Neural Network / Song Han, Jeff Pool, John Tran, William Dally // Advances in Neural Information Processing Systems 28 / Ed. by C. Cortes,

- N. D. Lawrence, D. D. Lee et al. — Curran Associates, Inc., 2015. — Pp. 1135–1143. <http://papers.nips.cc/paper/5784-learning-both-weights-and-connections-for-efficient-neural-network.pdf>.
11. Greedy Layer-Wise Training of Deep Networks / Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle // Advances in Neural Information Processing Systems 19 / Ed. by B. Schölkopf, J. C. Platt, T. Hoffman. — MIT Press, 2007. — Pp. 153–160. <http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf>.
  12. Hinton Geoffrey E., Osindero Simon, Teh Yee-Whye. A Fast Learning Algorithm for Deep Belief Nets // *Neural Comput.* — 2006. — Июль. — Vol. 18, no. 7. — Pp. 1527–1554. <http://dx.doi.org/10.1162/neco.2006.18.7.1527>.
  13. Semi-supervised Learning with Deep Generative Models / Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, Max Welling // Advances in Neural Information Processing Systems 27 / Ed. by Z. Ghahramani, M. Welling, C. Cortes et al. — Curran Associates, Inc., 2014. — Pp. 3581–3589. <http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf>.
  14. Li Yi, Shapiro L. O., Bilmes J. A. A generative/discriminative learning algorithm for image classification // Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. — Vol. 2. — 2005. — Oct. — Pp. 1605–1612 Vol. 2.
  15. J. Lasserre. Hybrid of generative and discriminative methods for machine learning: Ph.D. thesis / University of Cambridge. — 2008.
  16. AdaNet: Adaptive Structural Learning of Artificial Neural Networks / Corinna Cortes, Xavier Gonzalvo, Vitaly Kuznetsov et al. // International Conference on Machine Learning. — 2017. — Pp. 874–883.
  17. Zoph Barret, Le Quoc V. Neural architecture search with reinforcement learning // *arXiv preprint arXiv:1611.01578*. — 2016.
  18. Accelerating neural architecture search using performance prediction / Bowen Baker, Otkrist Gupta, Ramesh Raskar, Nikhil Naik // *CoRR, abs/1705.10823*. — 2017.
  19. Efficient Architecture Search by Network Transformation / Han Cai, Tianyao Chen, Weinan Zhang et al. — 2018.
  20. Learning transferable architectures for scalable image recognition / Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V Le // *arXiv preprint arXiv:1707.07012*. — 2017.
  21. Liu Hanxiao, Simonyan Karen, Yang Yiming. Darts: Differentiable architecture search // *arXiv preprint arXiv:1806.09055*. — 2018.
  22. Cho Kyunghyun. Foundations and Advances in Deep Learning: G5 Artikkeliväitöskirja. — Aalto University; Aalto-yliopisto, 2014. — P. 277. <http://urn.fi/URN:ISBN:978-952-60-5575-6>.

23. *Alain Guillaume, Bengio Yoshua.* What regularized auto-encoders learn from the data-generating distribution // *Journal of Machine Learning Research*. — 2014. — Vol. 15, no. 1. — Pp. 3563–3593. <http://dl.acm.org/citation.cfm?id=2750359>.
24. *Kamyshanska Hanna, Memisevic Roland.* On autoencoder scoring // Proceedings of the 30th International Conference on Machine Learning (ICML-13) / Ed. by Sanjoy Dasgupta, David Mcallester. — Vol. 28. — JMLR Workshop and Conference Proceedings, 2013. — Май. — Pp. 720–728. <http://jmlr.org/proceedings/papers/v28/kamyshanska13.pdf>.
25. *D. Kingma M. Welling.* Auto-Encoding Variational Bayes // Proceedings of the International Conference on Learning Representations (ICLR). — 2014.
26. How to Train Deep Variational Autoencoders and Probabilistic Ladder Networks. / Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe et al. // *CoRR*. — 2016. — Vol. abs/1602.02282. <http://dblp.uni-trier.de/db/journals/corr/corr1602.html#SonderbyRMSW16>.
27. Semi-Supervised Learning with Ladder Network. / Antti Rasmus, Harri Valpola, Mikko Honkala et al. // *CoRR*. — 2015. — Vol. abs/1507.02672. <http://dblp.uni-trier.de/db/journals/corr/corr1507.html#RasmusVHBR15>.
28. *MacKay David J. C.* Information Theory, Inference & Learning Algorithms. — New York, NY, USA: Cambridge University Press, 2002.
29. *Токмакова А. А., Стрижсов В. В.* Оценивание гиперпараметров линейных и регрессионных моделей при отборе шумовых и коррелирующих признаков // *Информатика и её применение*. — 2012. — Т. 6(4). — С. 66–75. [http://strijov.com/papers/Tokmakova2011HyperParJournal\\_Preprint.pdf](http://strijov.com/papers/Tokmakova2011HyperParJournal_Preprint.pdf).
30. *Зайцев А. А., Стрижсов В. В., Токмакова А. А.* Оценка гиперпараметров регрессионных моделей методом максимального правдоподобия // *Информационные технологии*. — 2013. — Vol. 2. — Pp. 11–15. [http://strijov.com/papers/ZaytsevStrijovTokmakova2012Likelihood\\_Preprint.pdf](http://strijov.com/papers/ZaytsevStrijovTokmakova2012Likelihood_Preprint.pdf).
31. *Strijov V., Weber Gerhard-Wilhelm.* NONLINEAR REGRESSION MODEL GENERATION USING HYPERPARAMETERS OPTIMIZATION: Preprint 2009-21. — Middle East Technical University, 06800 Ankara, Turkey: Institute of Applied Mathematics, 2009. — Октябрь. — Preprint No. 149.
32. *Стрижсов В. В.* Порождение и выбор моделей в задачах регрессии и классификации: Ph.D. thesis / Вычислительный центр РАН. — 2014. <http://strijov.com/papers/Strijov2015ModelSelectionRu.pdf>.
33. *Перекрестенко Д. О.* Анализ структурной и статистической сложности суперпозиции нейронных сетей. — 2014. <http://sourceforge.net/p/mlalgorithms/code/HEAD/tree/Group074/Perekrestenko2014Complexity.pdf>.
34. *Vladislavleva E.* Other publications TiSEM: : Tilburg University, School of Economics and Management, 2008. <http://EconPapers.repec.org/RePEc:tiu:tiutis:65a72d10-6b09-443f-8cb9-88f3bb3bc31b>.

35. Predicting Parameters in Deep Learning / Misha Denil, Babak Shakibi, Laurent Dinh et al. // Advances in Neural Information Processing Systems 26 / Ed. by C.j.c. Burges, L. Bottou, M. Welling et al. — 2013. — Pp. 2148–2156. [http://media.nips.cc/nipsbooks/nipspapers/paper\\_files/nips26/1053.pdf](http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/1053.pdf).
36. Xu Huan, Mannor Shie. Robustness and generalization // *Machine Learning*. — 2012. — Vol. 86, no. 3. — Pp. 391–423. <http://dx.doi.org/10.1007/s10994-011-5268-1>.
37. Intriguing properties of neural networks. / Christian Szegedy, Wojciech Zaremba, Ilya Sutskever et al. // *CoRR*. — 2013. — Vol. abs/1312.6199. <http://dblp.uni-trier.de/db/journals/corr/corr1312.html#SzegedyZSBEGF13>.
38. Stochastic Variational Inference / Matthew D. Hoffman, David M. Blei, Chong Wang, John Paisley // *J. Mach. Learn. Res.* — 2013. — Май. — Vol. 14, no. 1. — Pp. 1303–1347. <http://dl.acm.org/citation.cfm?id=2502581.2502622>.
39. Graves Alex. Practical Variational Inference for Neural Networks // Advances in Neural Information Processing Systems 24 / Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett et al. — Curran Associates, Inc., 2011. — Pp. 2348–2356. <http://papers.nips.cc/paper/4329-practical-variational-inference-for-neural-networks.pdf>.
40. Salimans Tim, Kingma Diederik P., Welling Max. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. // ICML / Ed. by Francis R. Bach, David M. Blei. — Vol. 37 of *JMLR Proceedings*. — JMLR.org, 2015. — Pp. 1218–1226. <http://dblp.uni-trier.de/db/conf/icml/icml2015.html#SalimansKW15>.
41. Maclaurin Dougal, Duvenaud David K., Adams Ryan P. Early Stopping is Nonparametric Variational Inference // *CoRR*. — 2015. — Vol. abs/1504.01344. <http://arxiv.org/abs/1504.01344>.
42. Mandt Stephan, Hoffman Matthew D, Blei David M. Continuous-Time Limit of Stochastic Gradient Descent Revisited.
43. Welling Max, Teh Yee Whye. Bayesian Learning via Stochastic Gradient Langevin Dynamics // Proceedings of the 28th International Conference on Machine Learning (ICML-11) / Ed. by Lise Getoor, Tobias Scheffer. — ICML '11. — New York, NY, USA: ACM, 2011. — June. — Pp. 681–688.
44. Arlot Sylvain, Celisse Alain. A survey of cross-validation procedures for model selection // *Statist. Surv.*. — 2010. — Vol. 4. — Pp. 40–79. <http://dx.doi.org/10.1214/09-SS054>.
45. Fast and Accurate Support Vector Machines on Large Scale Systems / Abhinav Vishnu, Jeyanthi Narasimhan, Lawrence Holder et al. // 2015 IEEE International Conference on Cluster Computing, CLUSTER 2015, Chicago, IL, USA, September 8-11, 2015. — 2015. — Pp. 110–119. <http://dx.doi.org/10.1109/CLUSTER.2015.26>.

46. Cross-validation pitfalls when selecting and assessing regression and classification models / Damjan Krstajic, Ljubomir J. Buturovic, David E. Leahy, Simon Thomas // *Journal of Cheminformatics*. — 2014. — Vol. 6, no. 1. — Pp. 1–15. <http://dx.doi.org/10.1186/1758-2946-6-10>.
47. Hornung Roman, Bernau Christoph, Truntzer Caroline et al. Full versus incomplete cross-validation: measuring the impact of imperfect separation between training and test sets in prediction error estimation. — 2014. <http://nbn-resolving.de/urn/resolver.pl?urn=nbn:de:bvb:19-epub-20682-6>.
48. Bengio Yoshua, Grandvalet Yves. No Unbiased Estimator of the Variance of K-Fold Cross-Validation // *J. Mach. Learn. Res.* — 2004. — Декабрь. — Vol. 5. — Pp. 1089–1105. <http://dl.acm.org/citation.cfm?id=1005332.1044695>.
49. Maclaurin Dougal, Duvenaud David, Adams Ryan. Gradient-based Hyperparameter Optimization through Reversible Learning // Proceedings of the 32nd International Conference on Machine Learning (ICML-15) / Ed. by David Blei, Francis Bach. — JMLR Workshop and Conference Proceedings, 2015. — Pp. 2113–2122. <http://jmlr.org/proceedings/papers/v37/maclaurin15.pdf>.
50. Domke Justin. Generic Methods for Optimization-Based Modeling. // AISTATS / Ed. by Neil D. Lawrence, Mark A. Girolami. — Vol. 22 of *JMLR Proceedings*. — JMLR.org, 2012. — Pp. 318–326. <http://dblp.uni-trier.de/db/journals/jmlr/jmlrp22.html#Domke12>.
51. Pedregosa Fabian. Hyperparameter optimization with approximate gradient // Proceedings of the 33nd International Conference on Machine Learning (ICML). — 2016. <http://jmlr.org/proceedings/papers/v48/pedregosa16.html>.
52. Scalable Gradient-Based Tuning of Continuous Regularization Hyperparameters / Jelena Luketina, Tapani Raiko, Mathias Berglund, Klaus Greff // Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 / Ed. by Maria-Florina Balcan, Kilian Q. Weinberger. — Vol. 48 of *JMLR Workshop and Conference Proceedings*. — JMLR.org, 2016. — Pp. 2952–2960.
53. Karaletsos Theofanis, Rätsch Gunnar. Automatic Relevance Determination For Deep Generative Models // *arXiv preprint arXiv:1505.07765*. — 2015.
54. A monolingual approach to detection of text reuse in Russian-English collection / Oleg Bakhteev, Rita Kuznetsova, Alexey Romanov, Anton Khritankov // Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), 2015 / IEEE. — 2015. — Pp. 3–10.
55. Бахтев Олег Юрьевич. Выбор модели глубокого обучения субоптимальной сложности с использованием вариационной оценки правдоподобия // Интеллектуализация обработки информации ИОИ-2016. — 2016. — Pp. 16–17.

56. Machine-Translated Text Detection in a Collection of Russian Scientific Papers / Alexey Romanov, Rita Kuznetsova, Oleg Bakhteev, Anton Khritankov // *Dialogue*. — 2016. — P. 2.
57. Бахтеев Олег Юрьевич. Градиентные методы оптимизации гиперпараметров моделей глубокого обучения // Всероссийская конференция ММРО-18. — 2017. — Рр. 10–11.
58. Бахтеев Олег Юрьевич, Кузнецова Маргарита Валерьевна. Детектирование переводных заимствований в текстах научных статей из журналов, входящих в РИНЦ // Всероссийская конференция ММРО-18. — 2017. — Рр. 128–129.
59. Бахтеев Олег Юрьевич. Выбор модели глубокого обучения субоптимальной сложности с использованием вариационной оценки правдоподобия // Интеллектуализация обработки информации ИОИ-2018. — 2016. — Рр. 16–17.
60. Бахтеев Олег Юрьевич, Стрижов Вадим Викторович. Выбор моделей глубокого обучения субоптимальной сложности // *Автоматика и телемеханика*. — 2018. — no. 8. — Pp. 129–147.
61. Бахтеев ОЮ. Восстановление панельной матрицы и ранжирующей модели по метризованной выборке в разнородных шкалах // *Машинное обучение и анализ данных*. — 2006. — Vol. 72, no. 7. — P. 1958.
62. Бахтеев ОЮ. Восстановление пропущенных значений в разнородных шкалах с большим числом пропусков // *Машинное обучение и анализ данных*. — 2015. — Vol. 1, no. 11. — Pp. 1484–1499.
63. Li Jundong, Liu Huan. Challenges of feature selection for big data analytics // *IEEE Intelligent Systems*. — 2017. — Vol. 32, no. 2. — Pp. 9–15.
64. Hassibi Babak, Stork David G, Wolff Gregory J. Optimal brain surgeon and general network pruning // Neural Networks, 1993., IEEE International Conference on / IEEE. — 1993. — Pp. 293–299.
65. Incremental network quantization: Towards lossless cnns with low-precision weights / Aojun Zhou, Anbang Yao, Yiwen Guo et al. // *arXiv preprint arXiv:1702.03044*. — 2017.
66. Han Song, Mao Huizi, Dally William J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding // *arXiv preprint arXiv:1510.00149*. — 2015.
67. Dropout: A simple way to prevent neural networks from overfitting / Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky et al. // *The Journal of Machine Learning Research*. — 2014. — Vol. 15, no. 1. — Pp. 1929–1958.
68. Louizos Christos, Ullrich Karen, Welling Max. Bayesian compression for deep learning // Advances in Neural Information Processing Systems. — 2017. — Pp. 3290–3300.
69. Bengio Yoshua. Gradient-based optimization of hyperparameters // *Neural computation*. — 2000. — Vol. 12, no. 8. — Pp. 1889–1900.

70. DrMAD: Distilling Reverse-Mode Automatic Differentiation for Optimizing Hyperparameters of Deep Neural Networks / Jie Fu, Hongyin Luo, Jiashi Feng et al. // *arXiv preprint arXiv:1601.00917*. — 2016.
71. *Pedregosa Fabian*. Hyperparameter optimization with approximate gradient // Proceedings of the 33rd International Conference on Machine Learning. — 2016.
72. *Snoek Jasper, Larochelle Hugo, Adams Ryan P.* Practical bayesian optimization of machine learning algorithms // Advances in neural information processing systems. — 2012. — Pp. 2951–2959.
73. Bayesian Optimization in High Dimensions via Random Embeddings. / Ziyu Wang, Masrour Zoghi, Frank Hutter et al. // IJCAI. — 2013. — Pp. 1778–1784.
74. Bayesian Optimization with Tree-structured Dependencies / Rodolphe Jenatton, Cedric Archambeau, Javier González, Matthias Seeger // International Conference on Machine Learning. — 2017. — Pp. 1655–1664.
75. Hyperparameter optimization of deep neural networks using non-probabilistic RBF surrogate model / Ilija Ilievski, Taimoor Akhtar, Jiashi Feng, Christine Annette Shoemaker // *arXiv preprint arXiv:1607.08316*. — 2016.
76. Scalable Bayesian Optimization Using Deep Neural Networks / Jasper Snoek, Oren Rippel, Kevin Swersky et al. // Proceedings of the 32nd International Conference on Machine Learning / Ed. by Francis Bach, David Blei. — Vol. 37 of *Proceedings of Machine Learning Research*. — Lille, France: PMLR, 2015. — 07–09 Jul. — Pp. 2171–2180. <http://proceedings.mlr.press/v37/snoek15.html>.
77. Structure Optimization for Deep Multimodal Fusion Networks using Graph-Induced Kernels / Dhanesh Ramachandram, Michal Lisicki, Timothy J Shields et al. // *arXiv preprint arXiv:1707.00750*. — 2017.
78. Raiders of the lost architecture: Kernels for Bayesian optimization in conditional parameter spaces / Kevin Swersky, David Duvenaud, Jasper Snoek et al. // *arXiv preprint arXiv:1409.4011*. — 2014.
79. *Воронцов Константин Вячеславович*. Локальные базисы в алгебраическом подходе к проблеме распознавания: Ph.D. thesis. — Graz, 1999.
80. *Abadi Martín, Agarwal Ashish, Barham Paul et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. — 2015. — Software available from tensorflow.org. <http://tensorflow.org/>.
81. *Theano Development Team*. Theano: A Python framework for fast computation of mathematical expressions // *arXiv e-prints*. — 2016. — may. — Vol. abs/1605.02688. <http://arxiv.org/abs/1605.02688>.
82. Automatic differentiation in PyTorch / Adam Paszke, Sam Gross, Soumith Chintala et al. — 2017.
83. *Eibe Frank, Hall MA, Witten IH*. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques" // *Morgan Kaufmann*. — 2016.

84. *Hofmann Markus, Klinkenberg Ralf.* RapidMiner: Data mining use cases and business analytics applications. — CRC Press, 2013.
85. Scikit-learn: Machine learning in Python / Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort et al. // *Journal of machine learning research*. — 2011. — Vol. 12, no. Oct. — Pp. 2825–2830.
86. Relational inductive biases, deep learning, and graph networks / Peter W Battaglia, Jessica B Hamrick, Victor Bapst et al. // *arXiv preprint arXiv:1806.01261*. — 2018.
87. *Negrinho Renato, Gordon Geoff.* Deeparchitect: Automatically designing and training deep architectures // *arXiv preprint arXiv:1704.08792*. — 2017.
88. Learning Bayesian network structure using LP relaxations / Tommi Jaakkola, David Sontag, Amir Globerson, Marina Meila // Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. — 2010. — Pp. 358–365.
89. *Alvarez-Melis David, Jaakkola Tommi S.* Tree-structured decoding with doubly-recurrent neural networks. — 2016.
90. *Adams Ryan, Wallach Hanna, Ghahramani Zoubin.* Learning the structure of deep sparse graphical models // Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. — 2010. — Pp. 1–8.
91. *Feng Jiashi, Darrell Trevor.* Learning the structure of deep convolutional networks // Proceedings of the IEEE international conference on computer vision. — 2015. — Pp. 2749–2757.
92. *Shirakawa Shinichi, Iwata Yasushi, Akimoto Youhei.* Dynamic Optimization of Neural Network Structures Using Probabilistic Modeling // *arXiv preprint arXiv:1801.07650*. — 2018.
93. Toward Optimal Run Racing: Application to Deep Learning Calibration / Olivier Bousquet, Sylvain Gelly, Karol Kurach et al. // *arXiv preprint arXiv:1706.03199*. — 2017.
94. Learning deep resnet blocks sequentially using boosting theory / Furong Huang, Jordan Ash, John Langford, Robert Schapire // *arXiv preprint arXiv:1706.04964*. — 2017.
95. Progressive neural architecture search / Chenxi Liu, Barret Zoph, Jonathon Shlens et al. // *arXiv preprint arXiv:1712.00559*. — 2017.
96. *Alain Guillaume, Bengio Yoshua.* Understanding intermediate layers using linear classifier probes // *arXiv preprint arXiv:1610.01644*. — 2016.
97. *Teerapittayanon Surat, McDanel Bradley, Kung HT.* Branchynet: Fast inference via early exiting from deep neural networks // Pattern Recognition (ICPR), 2016 23rd International Conference on / IEEE. — 2016. — Pp. 2464–2469.
98. Incremental Training of Deep Convolutional Neural Networks / R Istrate12, ACI Malossi, C Bekas, D Nikolopoulos.

99. *Chen Tianqi, Goodfellow Ian, Shlens Jonathon.* Net2net: Accelerating learning via knowledge transfer // *arXiv preprint arXiv:1511.05641*. — 2015.
100. Forward thinking: Building and training neural networks one layer at a time / Chris Hettinger, Tanner Christensen, Ben Ehlert et al. // *arXiv preprint arXiv:1706.02480*. — 2017.
101. *Miranda Conrado S, Von Zuben Fernando J.* Reducing the Training Time of Neural Networks by Partitioning // *arXiv preprint arXiv:1511.02954*. — 2015.
102. *Schmidhuber Juergen, Zhao Jieyu, Wiering MA.* Simple principles of metalearning // *Technical report IDSIA*. — 1996. — Vol. 69. — Pp. 1–23.
103. *Schmidhuber Jürgen.* A neural network that embeds its own meta-levels // *Neural Networks*, 1993., IEEE International Conference on / IEEE. — 1993. — Pp. 407–412.
104. Meta-SGD: Learning to Learn Quickly for Few Shot Learning / Zhenguo Li, Fengwei Zhou, Fei Chen, Hang Li // *arXiv preprint arXiv:1707.09835*. — 2017.
105. *Wang Yu-Xiong, Hebert Martial.* Learning to learn: Model regression networks for easy small sample learning // European Conference on Computer Vision / Springer. — 2016. — Pp. 616–634.
106. Learning to learn by gradient descent by gradient descent / Marcin Andrychowicz, Misha Denil, Sergio Gomez et al. // *Advances in Neural Information Processing Systems*. — 2016. — Pp. 3981–3989.
107. *Kinga D, Adam J Ba.* A method for stochastic optimization // International Conference on Learning Representations (ICLR). — Vol. 5. — 2015.
108. *Duchi John, Hazan Elad, Singer Yoram.* Adaptive subgradient methods for online learning and stochastic optimization // *Journal of Machine Learning Research*. — 2011. — Vol. 12, no. Jul. — Pp. 2121–2159.
109. *Friesen Abram L, Domingos Pedro.* Deep Learning as a Mixed Convex-Combinatorial Optimization Problem // *arXiv preprint arXiv:1710.11573*. — 2017.
110. *Kristiansen Gus, Gonzalvo Xavi.* EnergyNet: Energy-based Adaptive Structural Learning of Artificial Neural Network Architectures // *arXiv preprint arXiv:1711.03130*. — 2017.
111. Pathnet: Evolution channels gradient descent in super neural networks / Chrisantha Fernando, Dylan Banarse, Charles Blundell et al. // *arXiv preprint arXiv:1701.08734*. — 2017.
112. *Veniat Tom, Denoyer Ludovic.* Learning time-efficient deep architectures with budgeted super networks // *arXiv preprint arXiv:1706.00046*. — 2017.
113. Composing graphical models with neural networks for structured representations and fast inference / Matthew Johnson, David K Duvenaud, Alex Wiltschko et al. // *Advances in neural information processing systems*. — 2016. — Pp. 2946–2954.
114. *Nalisnick Eric, Smyth Padhraic.* Deep Generative Models with Stick-Breaking Priors // *arXiv preprint arXiv:1605.06197*. — 2016.

115. Abbasnejad M Ehsan, Dick Anthony, van den Hengel Anton. Infinite variational autoencoder for semi-supervised learning // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) / IEEE. — 2017. — Pp. 781–790.
116. Miller A. C., Foti N., Adams R. P. Variational Boosting: Iteratively Refining Posterior Approximations // ArXiv e-prints. — 2016. — nov.
117. Arnold Ludovic, Ollivier Yann. Layer-wise learning of deep generative models // arXiv preprint arXiv:1212.1524. — 2012.
118. Kuznetsov Mikhail, Tokmakova Aleksandra, Strijov Vadim. Analytic and stochastic methods of structure parameter estimation // Informatica. — 2016. — Vol. 27, no. 3. — Pp. 607–624.
119. Sutskever Ilya, Vinyals Oriol, Le Quoc V. Sequence to sequence learning with neural networks // Advances in neural information processing systems. — 2014. — Pp. 3104–3112.
120. Hernández-Lobato José Miguel, Adams Ryan. Probabilistic backpropagation for scalable learning of bayesian neural networks // International Conference on Machine Learning. — 2015. — Pp. 1861–1869.
121. Shang Yi, Wah B. W. Global optimization for neural network training // Computer. — 1996. — Mar. — Vol. 29, no. 3. — Pp. 45–54.
122. Gradient descent converges to minimizers / Jason D Lee, Max Simchowitz, Michael I Jordan, Benjamin Recht // University of California, Berkeley. — 2016. — Vol. 1050. — P. 16.
123. Dembo Amir, Cover Thomas M, Thomas Joy A. Information theoretic inequalities // Information Theory, IEEE Transactions on. — 1991. — Vol. 37, no. 6. — Pp. 1501–1518.
124. Nicholas Altieri, D. Duvenaud. Variational Inference with Gradient Flows. — Дата обращения: 15.05.2016. URL: <http://approximateinference.org/accepted/AltieriDuvenaud2015.pdf>.
125. Sato Issei, Nakagawa Hiroshi. Approximation analysis of stochastic gradient langevin dynamics by using fokker-planck equation and ito process // Proceedings of the 31st International Conference on Machine Learning (ICML-14). — 2014. — Pp. 982–990.
126. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks / Chunyuan Li, Changyou Chen, David E. Carlson, Lawrence Carin // Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. — 2016. — Pp. 1788–1794. <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11835>.
127. Lichman M. UCI Machine Learning Repository. — Дата обращения: 15.03.2017. URL: <http://archive.ics.uci.edu/ml>.
128. LeCun Yann. The MNIST database of handwritten digits // <http://yann.lecun.com/exdb/mnist/>. — 1998.

129. *Maclaurin Dougal, Adams Ryan P.* Firefly Monte Carlo: exact MCMC with subsets of data // Proceedings of the 24th International Conference on Artificial Intelligence / AAAI Press. — 2015. — Pp. 4289–4295.
130. Код вычислительного эксперимента. — Дата обращения: 15.03.2017. URL: [svn.code.sf.net/p/mlalgorithms/code/Group074/Bakhteev2016Evidence/](https://svn.code.sf.net/p/mlalgorithms/code/Group074/Bakhteev2016Evidence/).
131. <https://svn.code.sf.net/p/mlalgorithms/code/Group074/Bakhteev2017Hypergrad/code/>.
132. *Kwapisz Jennifer R, Weiss Gary M, Moore Samuel A.* Activity recognition using cell phone accelerometers // *ACM SigKDD Explorations Newsletter*. — 2011. — Vol. 12, no. 2. — Pp. 74–82.
133. *Bergstra James, Bengio Yoshua.* Random search for hyper-parameter optimization // *Journal of Machine Learning Research*. — 2012. — Vol. 13, no. Feb. — Pp. 281–305.
134. Algorithms for hyper-parameter optimization / James S Bergstra, Rémi Bardenet, Yoshua Bengio, Balázs Kégl // Advances in Neural Information Processing Systems. — 2011. — Pp. 2546–2554.
135. Scalable Bayesian Optimization Using Deep Neural Networks. / Jasper Snoek, Oren Rippel, Kevin Swersky et al. // ICML. — 2015. — Pp. 2171–2180.
136. *Hutter Frank, Hoos Holger H, Leyton-Brown Kevin.* Sequential model-based optimization for general algorithm configuration // International Conference on Learning and Intelligent Optimization / Springer. — 2011. — Pp. 507–523.
137. *Farcomeni Alessio.* Bayesian constrained variable selection // *Statistica Sinica*. — 2010. — Pp. 1043–1062.
138. *Oliphant Travis E.* A guide to NumPy. — Trelgol Publishing USA, 2006. — Vol. 1.