

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

На правах рукописи
УДК 519.254

Бахтеев Олег Юрьевич

ПОСЛЕДОВАТЕЛЬНОЕ ПОРОЖДЕНИЕ 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	4
Глава 1. Постановка задачи	8
Обзор	9
1.1. Постановка задачи	9
1.2. Метаоптимизация	11
1.2.1. Теоретические основания метаобучения	11
1.2.2. Метаоптимизация: learning to learn	12
1.2.3. Перебор структур	14
1.2.4. Обучение с подкреплением	14
1.3. Адаптивное изменение структуры	16
1.4. Байесовские методы порождения и выбора моделей	18
1.4.1. Автоматическое определение релевантности параметров	18
1.4.2. Суррогаты	19
1.4.3. Адаптивное изменение структуры	20
1.4.4. Порождающие модели	20
1.4.5. Состязательные модели	21
1.5. Способы прогнозирования графовых структур	21
1.6. Эвристические и прикладные методы	22
1.6.1. Эвристические методы	22
1.6.2. Структуры сетей специального вида	22
Глава 2. Выбор модели с использованием вариационного вывода	23
Выбор модели с использованием вариационного вывода	24
2.1. Методы получения вариационной оценки правдоподобия	27
2.1.1. Аппроксимация нормальным распределением	27
2.1.2. Аппроксимация с использованием градиентного метода	28
2.1.3. Аппроксимация с использованием динамики Ланжевена	32
Глава 3. Оптимизация гиперпараметров в задаче выбора модели	33
3.1. Градиентные методы оптимизации гиперпараметров	35
Глава 4. Выбор субоптимальной структуры модели	37
Глава 5. Анализ прикладных задач порождения и выбора моделей глубокого обучения	37
5.0.1. Выбор модели автокодировщика (Попова)	37
5.0.2. Evidence (АиТ)	40
5.0.3. Оптимизация гиперпараметров	45
5.0.4. Модели парофраза (Смердов)	47

5.0.5. Прореживание модели (Грабовой)	50
Список иллюстраций	55
Список таблиц	56

Введение

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

Под сложностью модели понимается *минимальная длина описания* [?], 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* [?]. Для получения оценки правдоподобия используются вариационные методы получения оценки правдоподобия [?], 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 [?, ?]. Поэтому задача выбора моделей глубокого обучения включает в себя выбор стратегии построения модели, эффективной по вычислительным ресурсам. В работе [?] приводятся теоретические оценки построения нейросетей с использованием , при которых построение модели производится итеративно последовательным увеличением числа нейронов в сети. В работе [?] 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 [?]. Альтернативой данным алгоритмам построения моделей являются методы, основанные на прореживании сетей глубокого обучения [?, ?, ?], т.е. последовательного удаления параметров, не дающих существенного прироста качества модели. В работах [?, ?] рассматривается послойное построение модели с отдельным критерием оптимизации для каждого слоя. В работах [?, ?, ?] предлагается декомпозиция модели на порождающую и разделяющую, оптимизируемых последовательно. В работе [?] 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 [?, ?, ?, ?] предлагается метод автоматического построения сверточной сети с использованием обучения с подкреплением. В [?] 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 [?] и автокодировщики [?]. В работе [?] рассматриваются некоторые типы регуляризации автокодировщиков, позволяющие формально рассматривать данные модели как порождающие модели с использованием байесового вывода. В работе [?] также рассматриваются ре-

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

В качестве критерия выбора модели в ряде работ [?, ?, ?, ?, ?, ?, ?] выступает правдоподобие модели. В работах [?, ?, ?, ?, ?] рассматривается проблема выбора модели и оценки гиперпараметров в задачах регрессии. Альтернативным критерием выбора модели является минимальная длина описания [?], являющаяся показателем статистической сложности модели и заданной выборки. В работе [?] рассматривается перечень критериев сложности моделей глубокого обучения и их взаимосвязь. В работе [?] в качестве критерия сложности модели выступает показатель нелинейности, характеризуемый степенью полинома Чебышева, аппроксимирующего функцию. В работе [?] 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 [?] 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 [?], в которой исследуется устойчивость классификации модели под действием шума.

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

Альтернативным методом выбора модели является выбор модели на основе скользящего контроля [?, ?]. Проблемой такого подхода является возможная высокая вычислительная сложность [?, ?]. В работах [?, ?] 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 [?, ?]. В работе [?] 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 [?, ?] рассматриваются градиентные методы оптимизации гиперпараметров, позволяющие оптимизировать большое количество гиперпараметров одновременно. В работе [?] предлагается метод оптимизации гиперпараметров с использованием градиентного спуска с моментом, в качестве оптимизируемого функционала рассматривается ошибка на валидационной части выборки. В работе [?] предлагается метод аппроксимации градиента функции потерь по гиперпараметрам, позволяющий использовать градиентные методы в задаче оптимизации гиперпараметров на больших выборках. В работе [?] предлагается упрощенный метод оптимизации гиперпараметров с градиентным спуском: вместо всей истории обновлений параметров для оптимизации используется только последнее обновление. В работе [?] рассматривается задача оптимизации параметров градиентного спуска с использованием нижней вариационной оценки интеграла правдоподобия.

Цели работы.

1. Исследовать методы построения моделей глубокого обучения.
2. Предложить критерии оптимальной и субоптимальной сложности модели глубокого обучения.
3. Предложить метод построения модели субоптимальной сложности.
4. Разработать алгоритм построения модели и провести вычислительный эксперимент для сравнения различных подходов к решению задачи автоматического построения моделей глубокого обучения.

Методы исследования. Для достижения поставленных целей используются методы вариационного байесовского вывода [?, ?, ?]. Рассматриваются графовое представление нейронной сети [?, ?]. Для получения вариационных оценок правдоподобия модели используется метод, основанный на градиентном спуске [?, ?]. В качестве метода получения модели субоптимальной сложности используется метод Automatic Relevance Determination [?, ?] с использованием градиентных методов оптимизации гиперпараметров [?, ?, ?, ?].

Основные положения, выносимые на защиту.

1. Предложен метод критерий и субоптимальной сложности модели глубокого обучения.
2. Разработан алгоритм построения модели глубокого обучения субоптимальной сложности.
3. Предложены методы оптимизации параметров и гиперпараметров модели.

4. Предложен обобщенный метод выбора модели глубокого обучения.
5. 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 данной диссертационной 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 иерархической классификации коллекций тезисов конференции и коллекций сайтов индустриального сектора; публикациями результатов исследования в рецензируемых научных изданиях, в том числе рекомендованных ВАК. Результаты работы докладывались и обсуждались на следующих научных конференциях.

1. TODO

Работа поддержана грантами Российского фонда фундаментальных исследований и Министерства образования и науки РФ.

1. 16-37-00488, Российский фонд фундаментальных исследований в рамках гранта “Разработка алгоритмов построения сетей глубокого обучения как суперпозиций универсальных моделей”.

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

1. TODO

Личный вклад. Все приведенные результаты, кроме отдельно оговоренных случаев, получены диссертантом 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 X наименований. Основной текст занимает Y 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.

В 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 субоптимальной сложности для заданной выборки для задачи классификации и регрессии. Работа данного комплекса анализируется на N выборках. Результаты, полученные с помощью предложенных методов, сравниваются с результатами известных алгоритмов.

Глава 1

Постановка задачи

Обзор

1.1. Постановка задачи

Задача выбора структуры модели является одной из базовых в области интеллектуального анализа данных. Проблему выбора структуры модели глубокого обучения можно сформулировать следующим образом: решается задача классификации или регрессии на заданной выборке \mathfrak{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 нелинейных функций. Под структурой модели понимается значения структурных параметров модели, т.е. параметров модели, характеризующий вид итоговой суперпозиции.

Формализуем описанную выше задачу. Задана выборка

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

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. Далее будем полагать, что пары объект (\mathbf{x}, y) являются реализацией некоторой случайно величины и порождены независимо.

Определение 1. Моделью глубокого обучения \mathbf{f} назовем дифференцируемую по параметрам функцию:

$$\mathbf{f}(\mathbf{x}, \mathbf{W}) : \mathbb{R}^n \rightarrow \mathbb{Y},$$

где \mathbf{W} — вектор параметров функции \mathbf{f} .

TODO: дальше идет определение структуры. Здесь тоже надо?

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

Смежной задачей к задаче выбора структуры модели является задача корректного представления структуры сети или параметризация сети глубокого обучения. Одним из возможных представлений структуры модели 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 достаточно общим и коррелирует с походом, описанным в [?], а также в библиотеках типа TensorFlow, Caffe, Teano, Torch, 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 то же время, в ряде работ, посвященных байесовской оптимизации [?, ?, ?], модель рассматривается как “черный

ящик”, имеющий ограниченный набор операций типа “произвести оптимизацию параметров” и “предсказать значение зависимой переменной по независимой переменной”. Подход, описанный в данных работах, также коррелирует с библиотеками машинного обучения, такими как Weka, RapidMiner или sklearn, в которых модель машинного обучения рассматривается как “черный ящик”.

Определение 2. Пусть задан граф V, E . Пусть для каждого ребра $(i, j) \in E$ определен вектор базовых функций $\mathbf{g}^{i,j}$. Граф V, E называется семейством моделей \mathfrak{F} , если функция, задаваемая рекурсивно как

$$f_j(\mathbf{x}) = \sum_{i \in \text{Adj}(v_j)} \langle \boldsymbol{\gamma}^{i,j}, \mathbf{g}^{i,j} \rangle (\mathbf{f}_k(\mathbf{x})), \quad \mathbf{f}_0(\mathbf{x}) = \mathbf{x}$$

является моделью при любых значениях векторов $\boldsymbol{\gamma}^{j,k}$.

Определение 3. Параметрами модели \mathbf{f} из семейства моделей \mathfrak{F} назовем конкатенацию векторов параметров моделей $\{\mathbf{f}_j\}_{j=0}^{|V|}$, $\mathbf{W} \in \mathbb{R}^d$.

Определение 4. Структурой модели $\boldsymbol{\Gamma}$ назовем конкатенацию векторов $\boldsymbol{\gamma}^{j,k}$.

Будем полагать, что для параметров модели \mathbf{W} и структуры $\boldsymbol{\Gamma}$ задано некоторое априорное распределение $p(\mathbf{W}, \boldsymbol{\Gamma})$.

Определение 5. Гиперпараметрами модели $\mathbf{h} \in \mathbb{R}^h$ назовем параметры распределения $p(\mathbf{W}, \boldsymbol{\Gamma})$.

Определение 6. Аппроксимирующим распределением назовем некоторое параметрическое приближение $q(\boldsymbol{\theta})$ апостериорного распределения параметров и структуры $p(\mathbf{W}, \boldsymbol{\Gamma} | \mathbf{X}, \mathbf{y}, \mathbf{h})$.

Определение 7. Оптимизируемыми параметрами модели $\boldsymbol{\theta} \in \mathbb{R}^u$ назовем параметры аппроксимирующего распределения q .

Определение 8. Пусть задано аппроксимирующее распределения q . Функцией потерь $L(\boldsymbol{\theta}, \mathbf{h})$ для модели \mathbf{f} назовем дифференцируемую функцию, характеризующую качество модели на обучающей выборке при параметрах модели, получаемых из распределения q .

В качестве функции L может выступать правдоподобие и апостериорная вероятность параметров модели на обучающей выборке.

Определение 9. Пусть задано аппроксимирующее распределения q и функция потерь L . Функцией валидации $Q(\mathbf{h}, \boldsymbol{\theta})$ для модели \mathbf{f} назовем дифференцируемую функцию, характеризующую качество модели при векторе $\boldsymbol{\theta}$, заданном неявно.

В общем случае задача выбора структуры модели и параметров модели становится как двухуровневая задача оптимизации:

$$\mathbf{h}^* = \arg \max_{\mathbf{h} \in \mathbb{R}^h} Q(\mathbf{h}), \quad (1.2)$$

где T — оператор оптимизации, решающий задачу оптимизации:

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

Заметим, что частным случаем выбора структуры глубокой сети является выбор обобщенно-линейных моделей, т.к. отдельные слои нейросети можно рассматривать как обобщенно-линейные модели. Задачу выбора обобщенно-линейной модели можно рассматривать как задачу выбора признаков, методы решения которой делятся на три группы [?]:

1. Фильтрационные методы. Основной особенностью данных методов является то, что такие методы не используют какой-либо информации о модели, а отсекают признаки только на основе статистических показателей.
2. Оберточные методы — методы, анализирующие подмножества признаков. Такие методы выбирают не признаки, а подмножества признаков, что позволяет учесть корреляция признаков.
3. Методы погружения проводят оптимизацию моделей и выбор признаков в единой процедуре, являясь комбинацией предыдущих типов отбора признаков.

1.2. Метаоптимизация

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

1. могут оценивать и сравнивать методы оптимизации моделей
2. оценивать возможные декомпозиции процесса оптимизации моделей
3. на основе полученных оценок предлагать оптимальные стратегии оптимизации моделей и отвергать неоптимальные стратегии.

1.2.1. Теоретические основания метаобучения

В работе [?] рассматривается задача построения порождающих моделей, предлагается критерий для послойного обучения порождающих моделей:

$$L = \max_{\theta} ???$$

Определение 10. Сэмплированием назовем порождение нескольких экземпляров модели из заданного аппроксимирующего распределения q .

В работе [?] рассматриваются подходы к сэмплированию моделей глубокого обучения. Предлагается формализация пространства поиска и формальное описание элементов пространства моделей:

(Concat

```
(Conv2D [32, 64] [3, 5] [1])
(MaybeSwap BatchNormalization ReLU)
(Optional (Dropout [0.5, 0.9]))
(Affine [10]))
```

Figure 1. A simple search space with 24 different models. See Figure 2 for a path through the search space.

1.2.2. Метаоптимизация: learning to learn

В работе [?] предлагается подход к адаптивному изменению структуры сети, основанный на обучении с подкреплением. Предлагается параметризация модели нейросети, включающая в себя модифицирующие и анализирующие выходы, позволяющие модифицировать параметры модели:

$$net_{y_k}(1) = 0, \quad \forall t \geq 1 : \quad x_k(t) \leftarrow environment,$$

$$y_k(t) = f_{y_k}(net_{y_k}(t)),$$

$$\forall t > 1 : \quad net_{y_k}(t) = \sum_l w_{y_k l}(t-1)l(t-1), \quad (7)$$

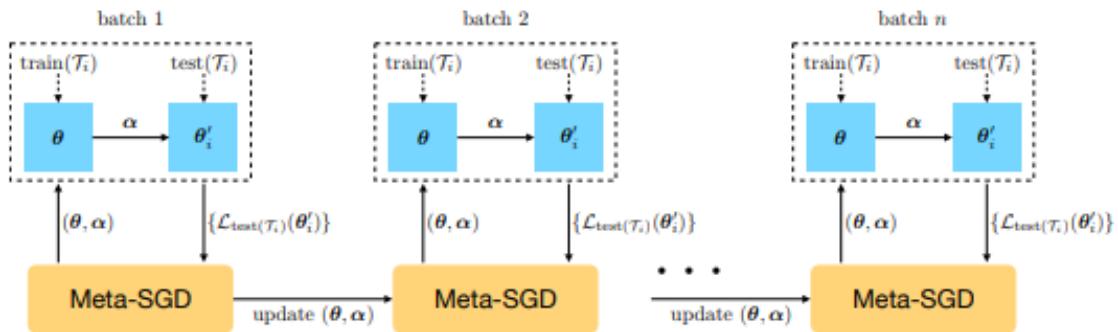
$$\forall t \geq 1 : \quad w_{ij}(t+1) = w_{ij}(t) + \Delta(t) g[\|adr(w_{ij}) - mod(t)\|^2] \quad (8)$$

$$val(1) = 0, \quad \forall t \geq 1 : \quad val(t+1) =$$

$$= \sum_{i,j} g[\|ana(t) - adr(w_{ij})\|^2]w_{ij}(t). \quad (9)$$

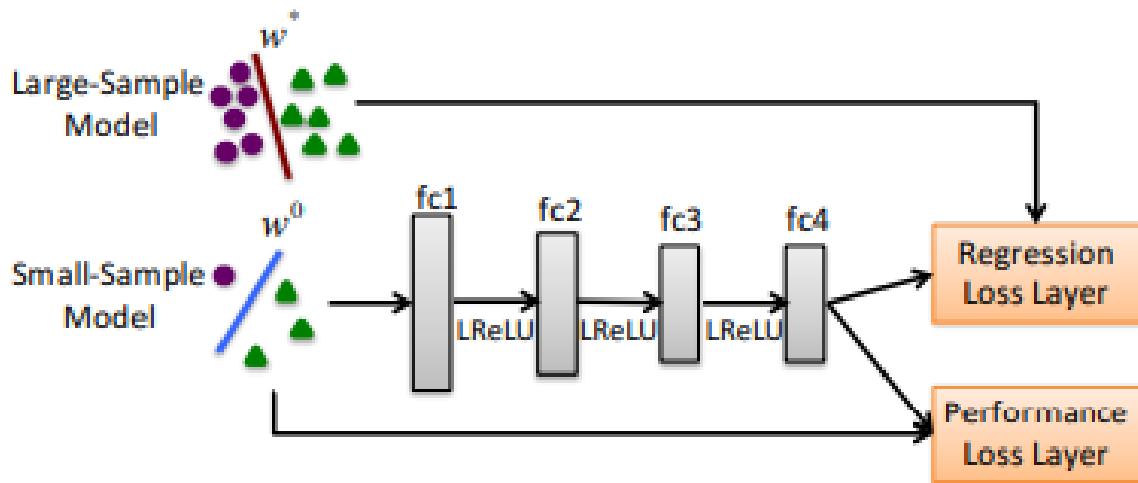
Предлагается продолжение подхода, позволяющая рекуррентно продолжать анализ модели и порождать мета-мета-...анализ.

В работе [?] рассматривается оптимизация метапараметров (шага градиентного спуска и начального распределения параметров) с использованием обучения с подкреплением. На каждой итерации сэмплируется подвыборка, по которой проводится оптимизация данных метапараметров:

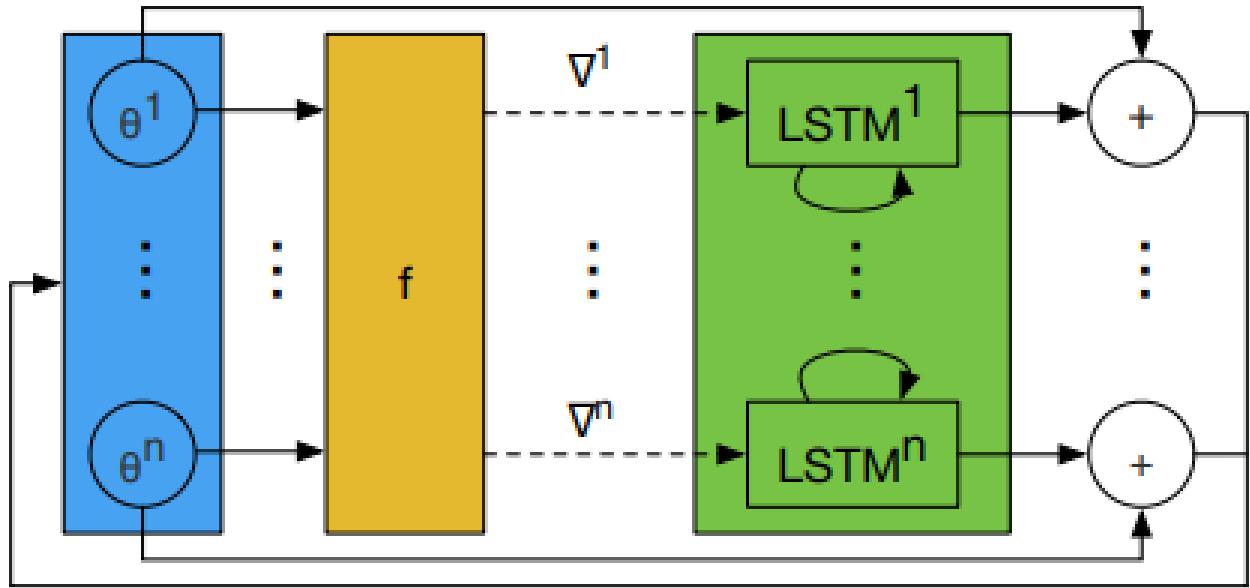


В работе [?] рассматривается задача восстановления параметров модели по параметрам слабо обученной модели:

$$L(\Theta) = \sum_{j=1}^J \left\{ \frac{1}{2} \left\| \mathbf{w}_j^* - T(\mathbf{w}_j^0, \Theta) \right\|_2^2 + \lambda \sum_{i=1}^{M+N} \left[1 - y_i^j \left(T(\mathbf{w}_j^0, \Theta)^T \mathbf{x}_i^j \right) \right]_+ \right\}. \quad (1)$$



В работе [?] рассматривается оптимизация метапараметров оптимизации с помощью LSTM, которая выступает альтернативе аналитических алгоритмов, таких как Adam или AdaGrad. LSTM имеет небольшое количество параметров, т.к. для каждого метапараметра используется своя копия модели LSTM с одинаковыми параметрами для каждой копии:



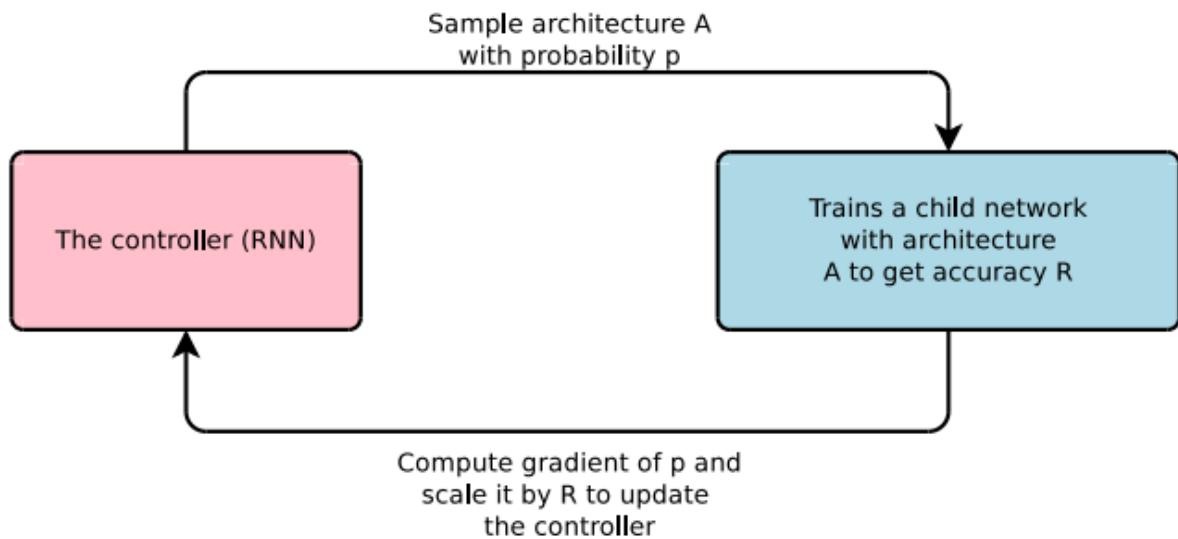
1.2.3. Перебор структур

В работе [?] рассматривается задача порождения сверточных нейронных сетей. Предлагается проводить поиск структуры сети по восходящему по сложности порядку: начиная от сетей с одним блоком и наращивая блоки. В силу высокой вычислительной сложности данного подхода, вместо построения модели, предлагается обучить рекуррентную нейросеть, которая предсказывает качество модели по заданным блокам.

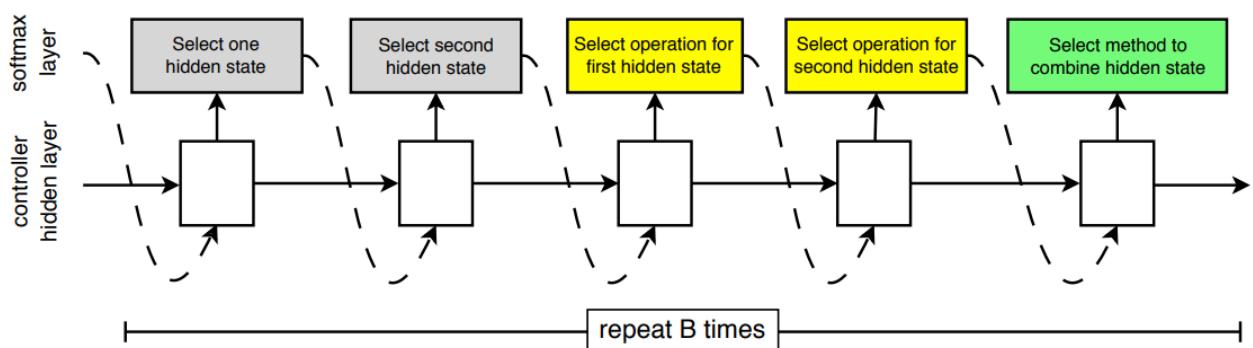
В работе [?] рассматривается задача выбора архитектуры с помощью большого количества параллельных запусков обучения моделей, предлагаются критерии ранней остановки оптимизации обучения моделей.

1.2.4. Обучение с подкреплением

В работе [?] представлена схема выбора архитектуры сверточной нейросети с использованием обучения с подкреплением. В качестве актора (контроллера) выступает рекуррентная нейронная сеть.

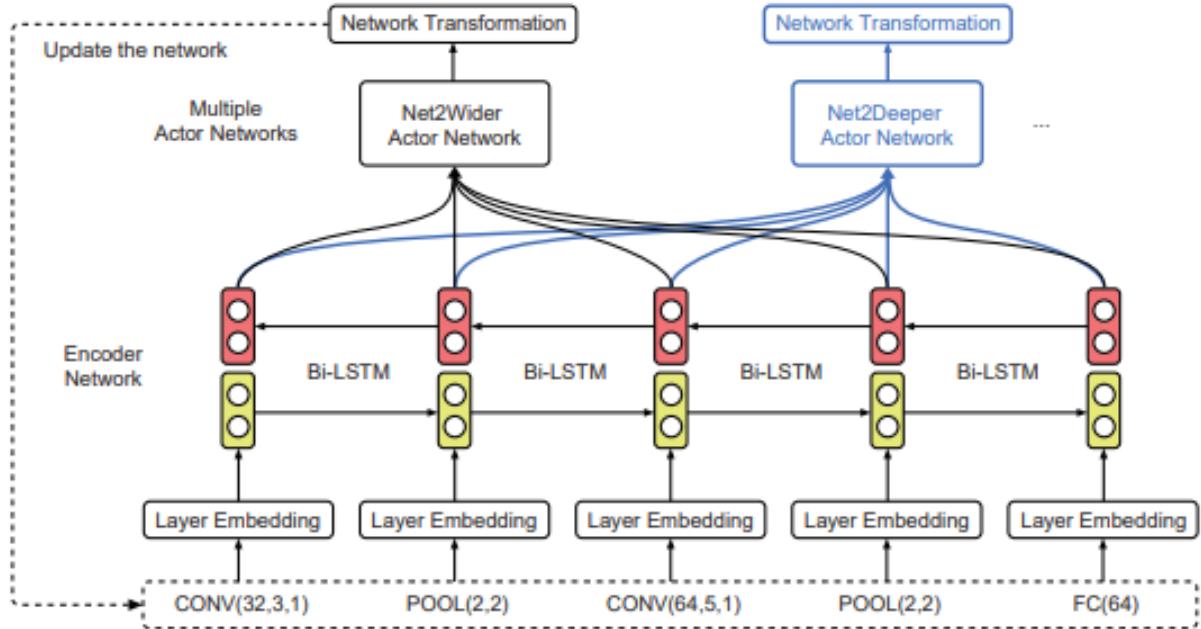


В работе [?] предлагается построение регрессионной модели для оценки финального качества модели и ранней остановки оптимизации моделей. Данный подход позволил существенно ускорить поиск моделей, представленный в работе [?]. В работе [?] рассматривается задача переноса архитектуры нейросети, обученной на более простой выборки, на более сложную. Также предлагается параметризация пространства поиска, более делательное, чем в [?]:



В отличие от предыдущих работ, в работе [?] предлагается подход к инкрементальному обучению нейросети, основанном на модификации модели, полученной на предыдущем шаге. Рассматривается две операции над нейросетью:

- Расширение сети
- Углубление сети



1.3. Адаптивное изменение структуры

В данном разделе собраны методы изменения структуры существующей модели.

Алгоритмы наращивания и прореживания параметров модели В работе [?] предлагается удалять неинформативные параметры модели, где в качестве показателя информативности выступает следующий функционал:

$$\delta E = \sum_i g_i \delta u_i + \frac{1}{2} \sum_i h_{ii} \delta u_i^2 + \frac{1}{2} \sum_{i \neq j} h_{ij} \delta u_i \delta u_j + O(\|\boldsymbol{\delta}\|^3) \quad (1)$$

В работе [?] было предложено развитие данного метода. В данной работе, в отличие от [?] не вводится предположений о диагональности Гессиана функции ошибок, поэтому удаление неинформативных параметров модели производится точнее.

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

$$\exp\left(-\frac{\mu_i^2}{2\sigma_i^2}\right) > \gamma \implies \left|\frac{\mu_i}{\sigma_i}\right| < \lambda$$

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

- Нормальное распределение с лог-равномерным распределением дисперсии, независимой для каждого нейрона:

$$p(\mathbf{W}, \mathbf{z}) \propto \prod_i^A \frac{1}{|z_i|} \prod_{ij}^{A,B} \mathcal{N}(w_{ij} | 0, z_i^2),$$

- Произведение двух половинных распределений Коши: одно ответственно за отдельный параметр, другое — за общее распределение параметров:

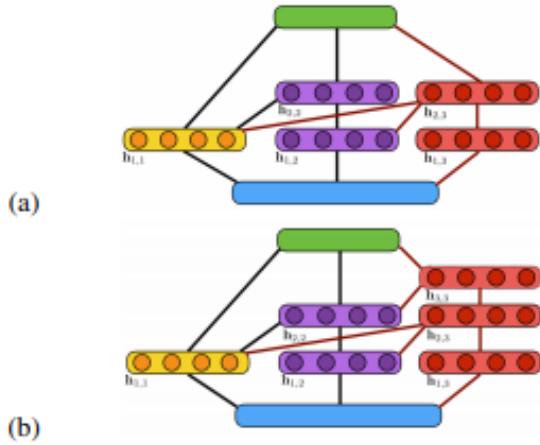
$$s \sim \mathcal{C}^+(0, \tau_0); \quad \tilde{z}_i \sim \mathcal{C}^+(0, 1); \quad \tilde{w}_{ij} \sim \mathcal{N}(0, 1); \quad w_{ij} = \tilde{w}_{ij} \tilde{z}_i s,$$

Смежной темой к прореживанию моделей выступает компрессия нейросетей. Основным отличием задачи прореживания и компрессии выступает эксплуатационное требование: если прореживание используется для получения оптимальной и наиболее устойчивой модели, то компрессия часто производится для сохранения памяти и основных эксплуатационных характеристик исходной модели (?). В работе [?] предлагается итеративное использование регуляризации типа DropOut [?] для прореживания модели. В работах [?, ?] используются методы снижения вычислительной точности представления параметров модели на основе кластеризации весов. В работе [?] предлагается метод компрессии, основанный на кластеризации значений параметров модели и представлении их в сжатом виде на основе кодов Хаффмана.

В работах [?, ?] предлагается наращивание моделей, основанное на бустинге. В работе рассматривается задача построения нейросетевых моделей специального типа:

$$\mathbf{f}_{t+1} = \sigma(\mathbf{f}_t) + \mathbf{f}_t,$$

приводится параметризация модели, позволяющая рассматривать декомпозицию модели на слабые классификаторы. В работе [?] на каждом шаге построения выбирается одно из двух расширений модели, каждое из которых рассматривается как слабый классификатор: 1. Сделать модель шире 2. Сделать модель глубже



Построение модели заканчивается при условии снижении радемахеровской сложности:

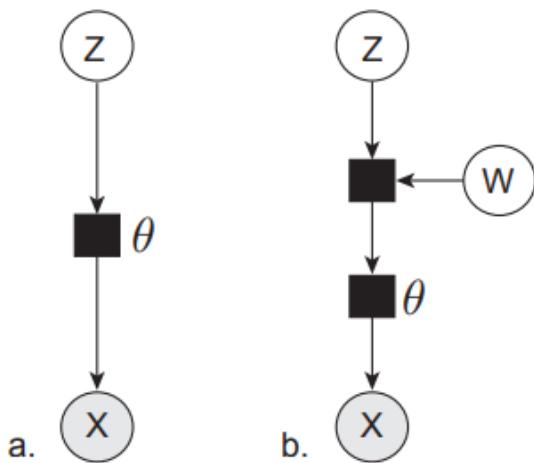
$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \frac{1}{m} \mathbb{E} \left[\sup_{h \in \mathcal{G}} \sum_{i=1}^m \sigma_i h(x_i) \right],$$

1.4. Байесовские методы порождения и выбора моделей

1.4.1. Автоматическое определение релевантности параметров

В работе [?] рассматривается задача оптимизации гиперпараметров. Авторы предлагают оптимизировать константы l_2 -регуляризации отдельно для каждого параметра модели, проводится параллель с методами автоматического определения релевантности параметров (ARD) [?].

В работе [?] 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:



1.4.2. Суррогаты

В работе [?] предлагаются моделировать качество модели гауссовым процессом, параметрами которого выступают гиперпараметры исходной модели. Модель, аппроксимирующая качество исходной модели, называется суррогатом.

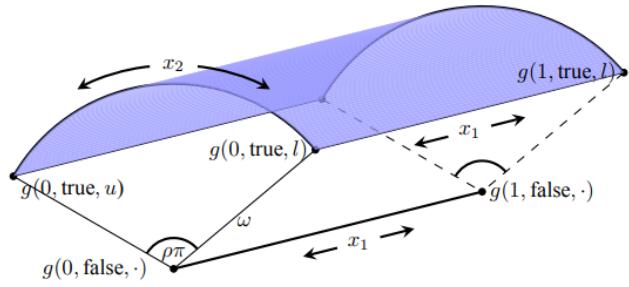
Одна из основных проблем использования гауссового процесса как суррогатной модели — кубическая сложность оптимизации. В работе [?] предлагаются использовать случайные подпространства гиперпараметров для ускоренной оптимизации. В работе [?] предлагается комбинация из множества гауссовых моделей и линейной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

В работе [?] предлагаются рассматривать RBF-модель для аппроксимации качества исходной модели, что позволяет ускорить процесс оптимизации суррогатной модели. В [?] рассматривается глубокая нейронная сеть в качестве суррогатной функции. Вместо интеграла правдоподобия, который оценивается в случае использования гауссового процесса в качестве суррогата, используется максимум апостериорной вероятности.

Важным параметром гауссовых процессов является функция ядра гауссово-го процесса, полностью определяющая процесс в случае нулевого среднего. В работе [?] предлагается функция ядра, определенная на графах:

$$k(x, y) = r(d(x, y)),$$

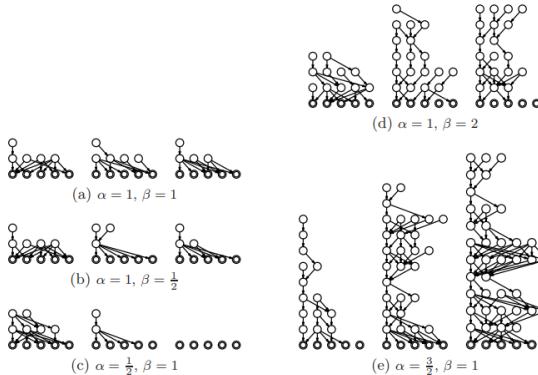
где d — геодезическое расстояние между вершинами графа, r — некоторая вещественная функция (наверно положительно определенная, но это не указано явно в статье). В работе [?] рассматривается задача выбора структуры нейросети, предлагается ядро специального вида, позволяющее учитывать только те гиперпараметры, которые есть в обеих сравниваемых моделях: к примеру, для двуслойной и трехслойной нейросети будут учитываться гиперпараметры, отвечающие только за первые два слоя.



1.4.3. Адаптивное изменение структуры

В работе [?] рассматривается порождение unsupervised-моделей с использованием расширения процесса Индийского Буфета:

$$p(K^{(m+1)} = k | K^{(m)}, \alpha, \beta) = \frac{1}{k!} \exp \left\{ -\lambda(K^{(m)}; \alpha, \beta) \right\} \lambda(K^{(m)}; \alpha, \beta)^k,$$



В работе [?] предлагается упрощенная модель Индийского Буфета:

$$-\log p(x, W, z) \sim \sum_{i=1}^N \|x_i - Wz_i\|_2^2 + \lambda^2 K$$

В работе [?] предлагается параметризация структуры модели с использованием Бернулиевских величин: каждая величина отвечает за включение или выключение слоя сети.

1.4.4. Порождающие модели

В работе [?] было предложено обобщение вариационного автокодировщика на случай частичного обучения: итоговая модель вариационного автокодировщика является порождающей моделью, учитывающей метки объектов.

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

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

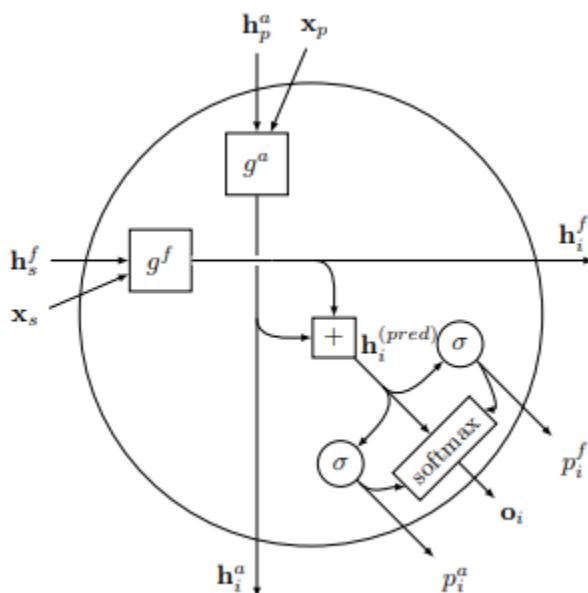
1.4.5. Состязательные модели

1.5. Способы прогнозирования графовых структур

В разделе собраны ключевые работы по порождению графовых моделей.

В работе [?] предлагается метод прогнозирования графовой структуры на основе линейного программирования. Предлагается свести проблему поиска графовой структуры к комбинаторной проблеме.

В работе [?] предлагается метод прогнозирования структур деревьев, основанный на дважды-рекуррентных нейросетях (doubly-recurrent), т.е. на сетях, отдельно предсказывающих глубину и ширину уровней деревьев.



1.6. Эвристические и прикладные методы

1.6.1. Эвристические методы

В работе [?] предлагается метод анализа структуры сети на основе линейных классификаторов, построенных на промежуточных слоях нейросети. Схожий метод был предложен в [?], где классификаторы на промежуточных уровнях используются для уменьшения вычислений при выполнении вывода и предсказаний. Промежуточные классификаторы работают как решающий список <http://www.eecs.harvard.edu/htk/publication/2016-icpr-teerapittayanon-mcdanel-kung.pdf>

В работе [?] предлагается инкрементальный метод построения нейросети: на каждом этапе построения в модель добавляются новые слои. Для улучшения качества модели, слои добавляются в начало модели, и затем проходят оптимизацию.

1.6.2. Структуры сетей специального вида

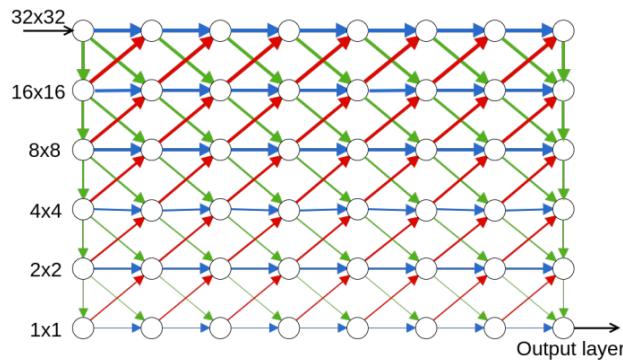
В данном разделе представлены работы по поиску оптимальной структуры сети, описывающие частные случаи поиска оптимальных моделей со структурами специального вида.

В работе [?] рассматривается оптимизация моделей нейросетей с бинарной функцией активацией. Задача оптимизации сводится к задаче mixed integer программирования, которая решается методами выпуклого анализа.

SKIP-сети, нужно ли писать? ResNet?

В работе [?] 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 [?, ?] рассматривается выбор архитектуры сети с использованием *суперсетей*: больших связанных между собой сетей, образующих граф, пути в котором определяют итоговую архитектуру нейросети. В работе [?] рассматриваются стохастические суперсети, позволяющие выбрать структуру нейросети за ограниченное время оптимизации. Схожий подход был предложен в работе [?], где предлагается использовать эволюционные алгоритмы для запоминания оптимальных подмоделей и переноса этих моделей в другие задачи.



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

Глава 2

Выбор модели с использованием вариационного вывода

Выбор модели с использованием вариационного вывода

Определение 11. Сложностью модели \mathbf{f} назовем правдоподобие модели:

$$Q = p(\mathbf{y}|\mathbf{X}, \mathbf{h}) = \int_{\mathbf{W} \in \mathbb{R}^{|\mathbf{W}|}, \Gamma \in \mathbb{R}^{|\Gamma|}} p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \Gamma, \mathbf{h}) p(\mathbf{W}, \Gamma|\mathbf{h}) d\mathbf{W}. \quad (2.1)$$

В данной главе будем полагать, что для каждой модели $\mathbf{f} \in \mathfrak{F}$ структура Γ фиксированна и определена однозначно. Модели $\mathbf{f} \in \mathfrak{F}$ имеют различные размерности d соответствующих векторов параметров \mathbf{W} .

Модель классификации \mathbf{f} назовем оптимальной среди моделей \mathfrak{F} , если достигается максимум интеграла (2.1).

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

$$p(\mathbf{W}|\mathbf{y}, \mathbf{X}, \mathbf{h}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \mathbf{h})p(\mathbf{W}|\mathbf{h})}{p(\mathbf{y}|\mathbf{X}, \mathbf{h})}. \quad (2.2)$$

Пример 1. Рассмотрим задачу линейной регрессии:

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

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

$$p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \mathbf{h}) = (2\pi)^{-\frac{m}{2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{W})^\top(\mathbf{y} - \mathbf{X}\mathbf{W})\right), \quad (2.3)$$

априорное распределение параметров модели имеет вид

$$p(\mathbf{W}|\mathbf{A}) = (2\pi)^{-\frac{n}{2}} |\mathbf{A}|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{W}^\top \mathbf{A} \mathbf{W}\right), \quad , \mathbf{h} = \text{diag}(\mathbf{A}). \quad (2.4)$$

Правдоподобие модели (2.1) в этом примере вычисляется аналитически [?]:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{h}) = (2\pi)^{-\frac{m}{2}} |\mathbf{A}|^{\frac{1}{2}} |\mathbf{H}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\mathbf{W}})^\top(\mathbf{y} - \mathbf{X}\hat{\mathbf{W}})\right) \exp\left(-\frac{1}{2}\hat{\mathbf{W}}^\top \mathbf{A} \hat{\mathbf{W}}\right), \quad (2.5)$$

где $\hat{\mathbf{W}}$ — значение наиболее вероятных (2.2) параметров модели:

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

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

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

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

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

$$\mathbf{f} = \mathbf{f}_{\text{SM}}(\mathbf{f}_1(\dots \mathbf{f}_K(\mathbf{x}))),$$

$\mathbf{f}_1, \dots, \mathbf{h}_K$ — дифференцируемые функции, \mathbf{f}_{SM} — многомерная логистическая функция:

$$\mathbf{f}_{\text{SM}} = \frac{\mathbf{f}_1(\dots \mathbf{f}_K(\mathbf{x}))}{\sum_{r=1}^Z \exp(f_{r,1}(\dots \mathbf{f}_K(\mathbf{x})))},$$

где $f_{r,1}$ — r -я компонента функции \mathbf{f}_1 . Компонента r вектора \mathbf{f}_{SM} определяет вероятность принадлежности объекта \mathbf{x} к классу r . Логарифм правдоподобия зависимой переменной аналогично (2.3) имеет вид

$$\log p(y|\mathbf{x}, \mathbf{W}, \mathbf{h}) = \log \hat{f}_{\hat{r}, \text{SM}}(\mathbf{f}_1(\dots \mathbf{f}_K(\mathbf{x}))),$$

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 (2.1) 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 (2.1), будем рассматривать его нижнюю оценку, полученную при помощи неравенства Йенсена [?]. Получим нижнюю оценку логарифма правдоподобия модели, используя неравенство

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X}, \mathbf{h}) &= \int_{\mathbf{W}} q(\mathbf{W}) \log \frac{p(\mathbf{y}, \mathbf{W}|\mathbf{X}, \mathbf{h})}{q(\mathbf{W})} d\mathbf{W} + D_{\text{KL}}(q(\mathbf{W})||p(\mathbf{W}|\mathbf{y}, \mathbf{X}, \mathbf{h})) \geq \\ &\geq \int_{\mathbf{W}} q(\mathbf{W}) \log \frac{p(\mathbf{y}, \mathbf{W}|\mathbf{X}, \mathbf{h})}{q(\mathbf{W})} d\mathbf{W} = \\ &= -D_{\text{KL}}(q(\mathbf{W})||p(\mathbf{W}|\mathbf{h})) + \int_{\mathbf{W}} q(\mathbf{W}) \log p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \mathbf{h}) d\mathbf{W}, \end{aligned} \tag{2.6}$$

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

$$D_{\text{KL}}(q(\mathbf{W})||p(\mathbf{W}|\mathbf{h})) = - \int_{\mathbf{W}} q(\mathbf{W}) \log \frac{p(\mathbf{W}|\mathbf{h})}{q(\mathbf{W})} d\mathbf{W},$$

$$p(\mathbf{y}, \mathbf{W}|\mathbf{X}, \mathbf{h}) = p(\mathbf{y}|\mathbf{X}, \mathbf{h})p(\mathbf{W}|\mathbf{h}).$$

Определение 12. Пусть задано аппроксимирующее распределение q . Вариационной оценкой логарифма правдоподобия модели (2.1) $\log p(\mathbf{y}|\mathbf{X}, \mathbf{h})$ называется оценка $\log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{h})$, полученная аппроксимацией неизвестного апостериорного распределения $p(\mathbf{W}|\mathbf{y}, \mathbf{X}, \mathbf{h})$ заданным распределением $q(\mathbf{W})$.

Будем рассматривать задачу нахождения вариационной оценки как задачу оптимизации. Пусть задано множество распределений $\mathfrak{Q} = \{q(\mathbf{W})\}$. Сведем задачу нахождения наиболее близкой вариационной нижней оценки интеграла (2.1) к оптимизации вида

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

В данной работе в качестве множества \mathfrak{Q} рассматривается нормальное распределение и распределение параметров, неявно получаемое оптимизацией градиентными методами.

Оценка (3.5) является нижней, поэтому может давать некорректные оценки для правдоподобия (2.1). Для того, чтобы оценить величину этой ошибки, докажем следующее утверждение.

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

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

логарифма интеграла (2.1) эквивалентна минимизации расстояния Кульбака–Лейблера между распределением $q(\mathbf{W}) \in \mathfrak{Q}$ и апостериорным распределением параметров $p(\mathbf{W}|\mathbf{y}, \mathbf{X}, \mathbf{h})$:

$$\hat{q} = \arg \max_{q \in \mathfrak{Q}} \int_{\mathbf{W}} q(\mathbf{W}) \log \frac{p(\mathbf{y}, \mathbf{W}|\mathbf{X}, \mathbf{h})}{q(\mathbf{W})} d\mathbf{W} \Leftrightarrow \hat{q} = \arg \min_{q \in \mathfrak{Q}} D_{\text{KL}}(q(\mathbf{W}) || p(\mathbf{W}|\mathbf{y}, \mathbf{X}, \mathbf{h})), \quad (2.7)$$

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

Доказательство. Доказательство непосредственно следует из (3.5). Вычитая из обеих частей равенства $D_{\text{KL}}(q(\mathbf{w}) || p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h}))$, получим

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

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

Таким образом, задача нахождения вариационной оценки, близкой к значению интеграла (2.1) сводится к поиску распределения \hat{q} , аппроксимирующего распределение $p(\mathbf{W}|\mathbf{y}, \mathbf{X}, \mathbf{h})$ наилучшим образом.

Модель \mathbf{f} назовем субоптимальной на множестве моделей \mathfrak{F} по множеству распределений \mathfrak{Q} , если модель доставляет максимум нижней вариационной оценке интеграла (3.8)

$$\max_{q \in \mathfrak{Q}} \int_{\mathbf{W}} q(\mathbf{W}) \log \frac{p(\mathbf{y}, \mathbf{W} | \mathbf{X}, \mathbf{h})}{q(\mathbf{W})} d\mathbf{W}. \quad (2.8)$$

Определение 13. Субоптимальность модели может быть также названа вариационной оптимальностью модели или LB-оптимальностью (*Lower Bound — нижняя граница*) модели.

Вариационная оценка (3.5) интерпретируется как оценка сложности модели по принципу минимальной длины описания [?], где первое слагаемое определяет количество информации для описания выборки, а второе слагаемое — длину описания самой модели [?].

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

где $\text{Len}(\mathbf{y} | \hat{\mathbf{W}}, \mathbf{h})$ — *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 [?].

В данной работе решается задача выбора субоптимальной модели при различных заданных множествах \mathfrak{Q} .

2.1. Методы получения вариационной оценки правдоподобия

Ниже представлены методы получения вариационных нижних оценок (2.8) правдоподобия (2.1). В первом подразделе рассматривается метод, основанный на аппроксимации апостериорного распределения $p(\mathbf{W} | \mathbf{y}, \mathbf{X}, \mathbf{h})$ (2.2) многомерным гауссовым распределением с диагональной матрицей ковариаций. В последующих разделах рассматриваются методы, основанные на различных модификациях стохастического градиентного спуска.

2.1.1. Аппроксимация нормальным распределением

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

$$q \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}), \quad (2.9)$$

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

Тогда оптимизация (3.8) имеет вид

$$\int_{\mathbf{W}} q(\mathbf{W}) \log p(\mathbf{y} | \mathbf{X}, \mathbf{W}, \mathbf{h}) d\mathbf{W} - D_{\text{KL}}(q(\mathbf{W}) || p(\mathbf{W} | \mathbf{f})) \rightarrow \max_{\mathbf{A}_q, \boldsymbol{\mu}_q}, \quad (2.10)$$

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

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

В качестве приближенного значения интеграла

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

предлагается использовать формулу

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

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

Итоговая функция оптимизации (2.10) имеет вид

$$\mathbf{f} = \arg \max_{\mathbf{A}_q, \boldsymbol{\mu}_q} \sum_{i=1}^m \log p(y_i|\mathbf{x}_i, \mathbf{W}_i) - D_{\text{KL}}(q(\mathbf{W})||p(\mathbf{W}|\mathbf{f})). \quad (2.11)$$

Пример 3. Пусть задана выборка \mathfrak{D} , в которой переменная y не зависит от \mathbf{x} :

$$y \sim \mathcal{N}(\mathbf{W}, \mathbf{B}^{-1}), \quad (2.12)$$

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

График аппроксимации распределения параметров представлен на рис. 5.4, а. Как видно из графика, с использованием метода (2.11) получено грубое приближение апостериорного распределения $p(\mathbf{W}|\mathbf{y}, \mathbf{X}, \mathbf{h})$, что может существенно занизить оценку правдоподобия модели.

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

2.1.2. Аппроксимация с использованием градиентного метода

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

Представим неравенство (3.5)

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{h}) \geq \int_{\mathbf{W}} q(\mathbf{W}) \log \frac{p(\mathbf{y}, \mathbf{W}|\mathbf{X}, \mathbf{h})}{q(\mathbf{W})} d\mathbf{W} = \mathbb{E}_{q(\mathbf{w})} (\log p(\mathbf{y}, \mathbf{W}|\mathbf{X}, \mathbf{h})) - S(q(\mathbf{w})), \quad (2.13)$$

где S — энтропия распределения:

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

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

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

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

Оценка распределений производится при оптимизации параметров. Оптимизация выполняется в режиме мультистарта [?], т.е. при запуске оптимизации параметров модели из нескольких разных начальных приближений. Основная проблема такого подхода — вычисление энтропии S распределений $q(\mathbf{W}) \in \mathfrak{Q}$. Ниже представлен метод получения оценок энтропии (2.17) S и оценок правдоподобия (2.13).

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

$$L = - \sum_{i=1}^m \log p(y_i, \mathbf{W}|\mathbf{x}_i, \mathbf{h}) = -\log p(\mathbf{W}|\mathbf{f}) - \sum_{i=1}^m \log p(y_i|\mathbf{x}_i, \mathbf{W}, \mathbf{h}). \quad (2.14)$$

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 14. 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:

$$T(\mathbf{W}) = \mathbf{W} - \gamma \nabla L(\mathbf{W}, \mathbf{y}, \mathbf{X}), \quad (2.15)$$

где γ — 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 [?].

Теорема 1. Пусть T — оператор градиентного спуска, L — функция потерь, градиент ∇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 γ — длина шага градиентного спуска, такая что

$$\gamma < \frac{1}{C_L}, \quad \gamma < \left(\max_{g \in \{1, \dots, r\}} \lambda_{\max}(\mathbf{H}(\mathbf{W}^g)) \right)^{-1}, \quad (2.16)$$

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

При выполнении неравенств (2.16) разность энтропий распределений $q'(\mathbf{W}), q(\mathbf{W})$ на смежных шагах почти наверное сходится к следующему выражению:

$$S(q'(\mathbf{W})) - S(q(\mathbf{W})) \approx \frac{1}{r} \sum_{g=1}^r (-\gamma \text{Tr}[\mathbf{H}(\mathbf{W}'^g)] - \gamma \text{Tr}[\mathbf{H}(\mathbf{W}'^g)\mathbf{H}(\mathbf{W}'^g)]) + o_{\gamma^2 \rightarrow 0}(1), \quad (2.17)$$

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

Предварительно приведем две леммы [?, ?], требуемые для доказательства теоремы.

Лемма 1. Пусть T — оператор градиентного спуска, L — дважды дифференцируемая функция потерь, градиент ∇L которой имеет константу Липшица C_L . Пусть для длины шага γ 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

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

Доказательство. 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

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

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 λ_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 (2.17), что и требовалось доказать. \square

Получим итоговую формулу для оценки правдоподобия модели. Оценка (2.13) на шаге оптимизации τ представима в виде

$$\log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{h}) \approx \frac{1}{r} \sum_{g=1}^r L(\mathbf{W}_\tau^g, \mathbf{X}, \mathbf{y}) + \mathsf{S}(q^0(\mathbf{W})) + \frac{1}{r} \sum_{b=1}^\tau \sum_{g=1}^r (-\gamma \text{Tr}[\mathbf{H}(\mathbf{W}_b^g)] - \gamma^2 \text{Tr}[\mathbf{H}(\mathbf{W}_b^g) \mathbf{H}(\mathbf{W}_b^g)]) \quad (2.18)$$

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.

В [?] предлагается алгоритм приближенного вычисления для выражения, находящегося под знаком суммы в (2.18):

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

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 постоянного 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 далее.

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 $\mathbf{W}^1, \dots, \mathbf{W}^r$:

$$T(\mathbf{W}) = \mathbf{W} - \frac{m}{\hat{m}} \gamma \nabla L(\mathbf{W}, \hat{\mathbf{y}}, \hat{\mathbf{X}}), \quad (2.19)$$

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 (2.19), основной проблемой модифицированного алгоритма оценки интеграла (2.8) является грубость аппроксимации исходного распределения $p(\mathbf{W}|\mathbf{f}, \mathfrak{D})$.

Рассмотрим пример 2 (2.12). График аппроксимации распределения $p(\mathbf{W}|\mathbf{y}, \mathbf{X}, \mathbf{h})$ представлен на рис. 5.4, б. Как видно из графика, градиентный спуск сходится к mode распределения. При небольшом количестве итераций полученное распределение также слабо аппроксирует апостериорное распределение. При приближении к точке экстремума снижается вариационная оценка правдоподобия модели, что интерпретируется как возможное начало переобучения [?]. Таким образом, снижение оценки (2.18) можно использовать как критерий остановки оптимизации модели для снижения эффекта переобучения.

На рис. 5.4 представлена аппроксимация распределения $p(\mathbf{W}|\mathbf{Y}, \mathbf{X}, \mathbf{h})$ различными методами: а) нормальным распределением с диагональной матрицей ковариаций, б) с помощью градиентного спуска, в) с помощью стохастической динамики Ланжевена. Точками отмечены параметры модели \mathbf{f} , полученные в ходе нескольких запусков оптимизации и являющиеся реализациями случайной величины с распределением $q(\mathbf{W})$. Нормальное распределение слабо аппроксирует распределение $p(\mathbf{W}|\mathbf{Y}, \mathbf{X}, \mathbf{h})$ в силу диагональности матрицы ковариаций. Распределение, полученное с помощью градиентного спуска, слабо аппроксирует распределение $p(\mathbf{W}|\mathbf{Y}, \mathbf{X}, \mathbf{h})$, так как сходится к mode.

2.1.3. Аппроксимация с использованием динамики Ланжевена

Для достижения нижней оценки интеграла (2.8), более близкой к реальному значению логарифма интеграла (2.1), чем оценка с использованием градиентного спуска, предлагается использовать стохастическую динамику Ланжевена [?]. Стохастическая динамика Ланжевена представляет собой вариант стохастического градиентного спуска с добавлением гауссового шума:

$$T(\mathbf{W}) = \mathbf{W} - \gamma \nabla L - \frac{m}{\hat{m}} \log p(\hat{\mathbf{y}}|\hat{\mathbf{X}}, \mathbf{W}, \mathbf{h}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \frac{\gamma}{2} \mathbf{I}), \quad (2.20)$$

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

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

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

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

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

В отличие от стохастического градиентного спуска стохастическая динамика Ланжевена сходится к апостериорному распределению параметров $p(\mathbf{W}|\mathfrak{D}, \mathbf{h})$ [?, ?]. График аппроксимации апостериорного распределения с использованием динамики Ланжевена представлен на рис. 5.4, б. 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 (2.8). В то же время алгоритм более требователен к настройке параметров оптимизации [?]: “быстро изменяющаяся кривизна [траекторий параметров модели] делает методы стохастической градиентной динамики Ланжевена по умолчанию неэффективными”.

Глава 3

Оптимизация гиперпараметров в задаче выбора модели

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 положим:

$$\boldsymbol{\theta} = \mathbf{W}, \quad \mathbf{h} = \text{diag}(\mathbf{A}) = [\alpha_1, \dots, \alpha_u]. \quad (3.1)$$

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

$$\hat{\boldsymbol{\theta}} = \arg \max (-L(\boldsymbol{\theta}, \mathbf{h})) = p(\mathbf{W}|\mathbf{X}, \mathbf{y}, \mathbf{A}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{W})p(\mathbf{W}|\mathbf{A})}{p(\mathbf{y}|\mathbf{X}, \mathbf{A})}. \quad (3.2)$$

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:

$$Q(\boldsymbol{\theta}, \mathbf{h}) = p(\mathbf{y}|\mathbf{X}, \mathbf{A}) = \int_{\mathbf{W} \in \mathbb{R}^u} p(\mathbf{y}|\mathbf{X}, \mathbf{W})p(\mathbf{W}|\mathbf{A}) \rightarrow \max_{[\alpha_1, \dots, \alpha_u] \in \mathbb{R}^n}. \quad (3.3)$$

В общем виде задача оптимизации гиперпараметров сводится к двухуровневой задаче оптимизации (3.8). Рассмотрим вид переменной $\boldsymbol{\theta}$ и функций L, Q для различных методов выбора модели и оптимизации ее параметров.

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:

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

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} :

$$L(\boldsymbol{\theta}, \mathbf{h}) = -\frac{1}{k} \sum_{q=1}^k \left(\frac{k}{k-1} \log p(\mathbf{y} \setminus \mathbf{y}_q | \mathbf{X} \setminus \mathbf{X}_q, \mathbf{W}_q) + \log p(\mathbf{W}_q | \mathbf{A}) \right). \quad (3.4)$$

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:

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X}, \mathbf{A}) &\geq -D_{\text{KL}}(q(\mathbf{W})||p(\mathbf{W}|\mathbf{A})) + \int_{\mathbf{W}} q(\mathbf{W}) \log p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \mathbf{A}) d\mathbf{W} \approx \\ &\approx \sum_{i=1}^m \log p(y_i|\mathbf{x}_i, \mathbf{W}_i) - D_{\text{KL}}(q(\mathbf{W})||p(\mathbf{W}|\mathbf{A})) = -L(\boldsymbol{\theta}, \mathbf{h}) = Q(\boldsymbol{\theta}), \end{aligned} \quad (3.5)$$

где q — нормальное распределение с диагональной матрицей ковариаций:

$$q \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}), \quad (3.6)$$

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_{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 :

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

3.1. Градиентные методы оптимизации гиперпараметров

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

Рассмотрим оператор градиентного спуска, производящий η шагов оптимизации:

$$\hat{\boldsymbol{\theta}} = T \circ T \circ \dots \circ T(\boldsymbol{\theta}_0, \mathbf{h}) = T^\eta(\boldsymbol{\theta}_0, \mathbf{h}), \quad (3.7)$$

где

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

γ — 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 (??), (??) в следующем виде

$$\hat{\mathbf{h}} = \arg \max_{\mathbf{h} \in \mathbb{R}^h} Q(T^\eta(\boldsymbol{\theta}_0, \mathbf{h})), \quad (3.8)$$

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

Оптимизационную задачу (3.8) 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 (3.8) 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 (3.7) 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 η 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:

$$\mathbf{h}' = \mathbf{h} - \gamma_{\mathbf{h}} \hat{\nabla}_{\mathbf{h}} Q. \quad (3.9)$$

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 η 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:

$$\boldsymbol{\theta}^\tau = \boldsymbol{\theta}_0 + \frac{\tau}{\eta} T(\boldsymbol{\theta}). \quad (3.10)$$

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

1. Провести η 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 $\boldsymbol{\theta}^\tau$ (3.10).
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 (3.9). В работе [?] отмечается неустойчивость алгоритма при высоких значениях длины шага градиентного спуска γ . Поэтому вместо исходного правила (3.10) в данной работе первые 5% значений параметров не рассматриваются, а также учитывается только каждый τ_k шаг оптимизации:

$$\boldsymbol{\theta}^\tau = \boldsymbol{\theta}_{\tau_0} + \frac{\tau}{\eta} T(\boldsymbol{\theta}), \quad \tau \in \{\tau_0, \dots, \eta\}, \tau \bmod \tau_k = 0, \quad (3.11)$$

где $\tau_0 = [0.05 \cdot \eta]$.

Глава 4

Выбор субоптимальной структуры модели

Глава 5

Анализ прикладных задач порождения и выбора моделей глубокого обучения

5.0.1. Выбор модели автокодировщика (Попова)

В качестве данных для проведения вычислительного эксперимента использовались данные WISDM [?], представляющие собой набор записей акселерометра мобильного телефона. Каждой записи соответствуют три координаты по осям акселерометра. Набор данных содержит записи движений для 6 классов

Алгоритм	Тип алгоритма	Сложность работы одной итерации	Предположения для корректности
Случайный поиск	стохастический	$O(\eta s \hat{\mathcal{D}})$	-
Жадный алгоритм [?]	градиентный	$O(\eta(s+h) \hat{\mathcal{D}})$	$\mathbf{H}(\boldsymbol{\theta}) = \mathbf{I}$
HOAG [?]	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 [?]	градиентный	$O(\eta s \hat{\mathcal{D}})$	Траектория оптимизации параметров $\boldsymbol{\theta} = \boldsymbol{\theta}_0, \dots, \boldsymbol{\theta}_\eta$ — линейная

Таблица 3.1. Основные свойства рассматриваемых алгоритмов

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 [?], 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 зависи-

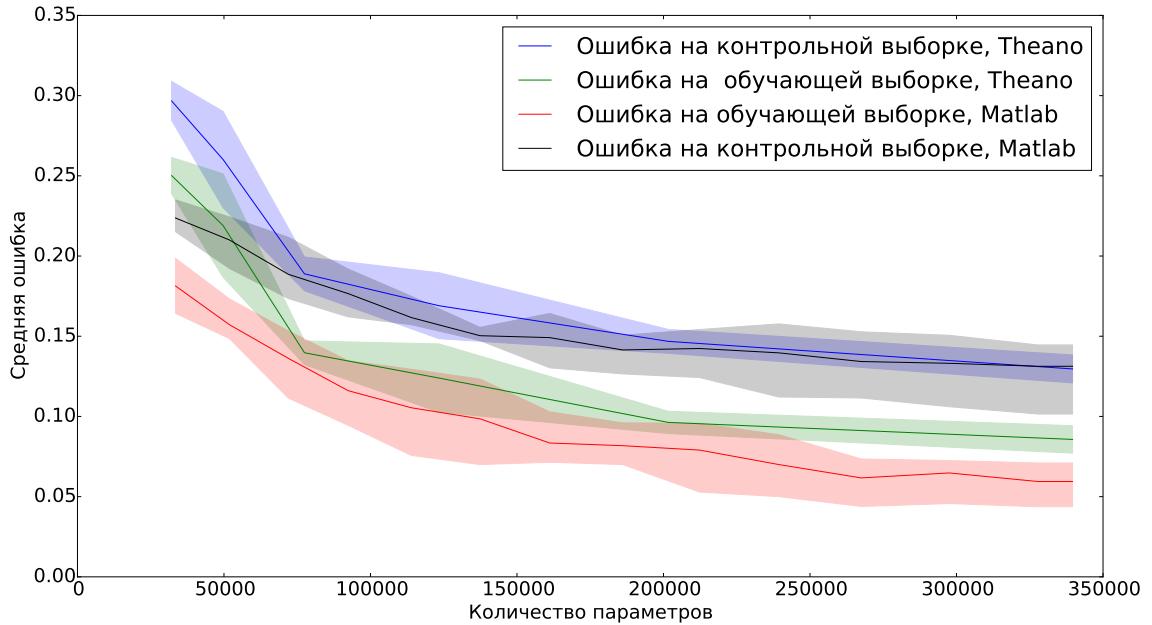


Рис. 5.1. Зависимость ошибки от числа нейронов

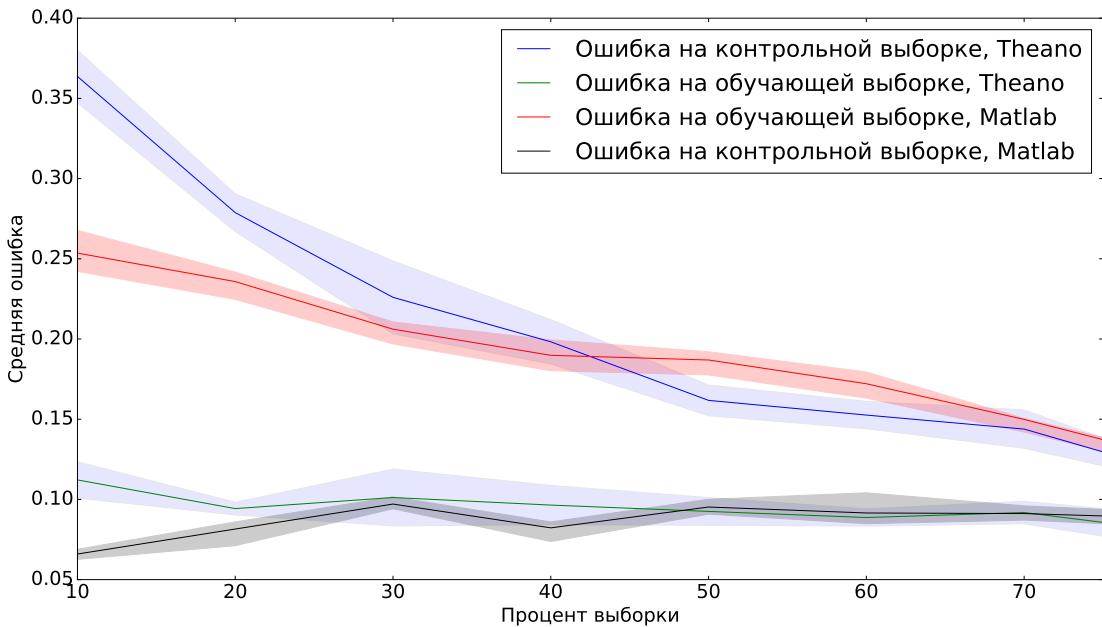


Рис. 5.2. Зависимость ошибки от размера обучающей выборки

мости от конфигурации Theano был сделан следующий эксперимент: проводилось обучение двухслойной нейросети на основе подсчитанных заранее параметров 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 установ-

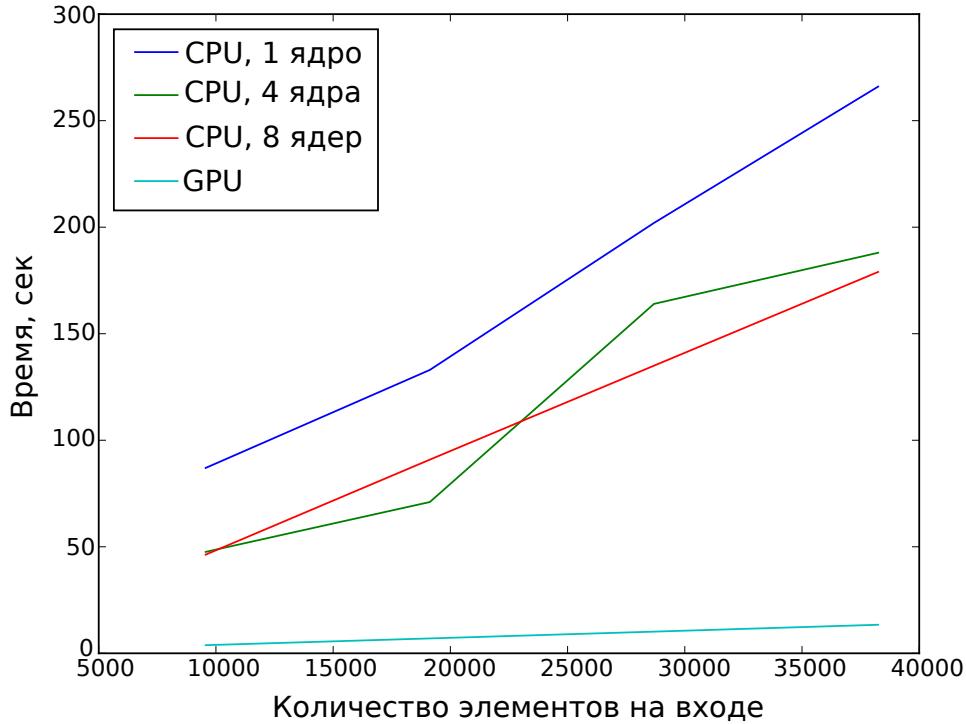


Рис. 5.3. Результаты эксперимента по исследованию скорости процесса обучения

лено как 300:200:100. Запуск осуществлялся со следующими конфигурациями Theano:

- 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.

5.0.2. Evidence (АиТ)

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) [?] и подвыборке изображений рукописных цифр MNIST [?].

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:

$$\mathbf{f}(\mathbf{w}, \mathbf{X}) = \text{softplus}(\mathbf{X}\mathbf{W}_1)\mathbf{W}_2, \quad (5.1)$$

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 однослоиной модели (5.1). 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 [?]. 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:

$$\text{RMSE}_\sigma = \text{RMSE}(\mathbf{f}(\mathbf{w}, \mathbf{X} + \boldsymbol{\varepsilon}), \mathbf{y}), \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma \mathbf{I}). \quad (5.2)$$

Были рассмотрены шесть алгоритмов.

1. Базовый алгоритм: оптимизация параметров без валидации и ранней остановки. Оптимизация проводилась с использованием стохастического градиентного спуска (2.19). 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 α — дисперсия.
4. Нахождение вариационной нижней оценки с использованием стохастического градиентного спуска.
5. Нахождение вариационной нижней оценки с использованием стохастической динамики Ланжевена.
6. Нахождение вариационной нижней оценки с аппроксимацией нормальным распределением (2.11).

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 τ_{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 (2.14) на валидационной выборке $\mathfrak{D}_{\text{valid}}$ для алгоритмов 2, 3,
2. вариационная нижняя оценка правдоподобия (3.5) на обучающей выборке $\mathfrak{D}_{\text{train}}$ для алгоритмов 4, 5, 6.

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

Рис. 5.3. *a*

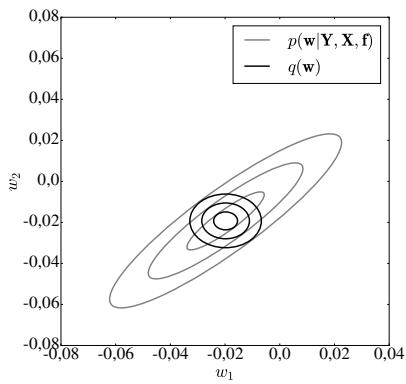


Рис. 5.3. *б*

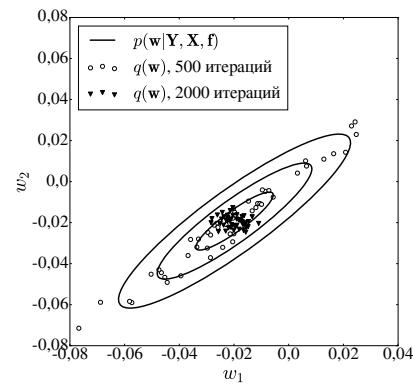


Рис. 5.3. *в*

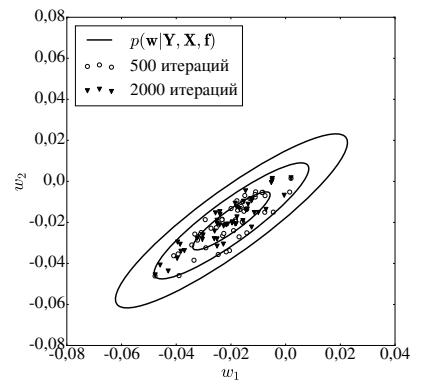


Рис. 5.4.

Описание эксперимента представлено в табл. 1. Результаты экспериментов представлены в табл. 2. На рис. 5.5 представлен график зависимости RMSE_σ от параметра σ для однослойных моделей.

Таблица 5.1. Описание выборок для экспериментов

Выборка \mathfrak{D}	Интервал валидации, τ_{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 5.2. Результаты эксперимента

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

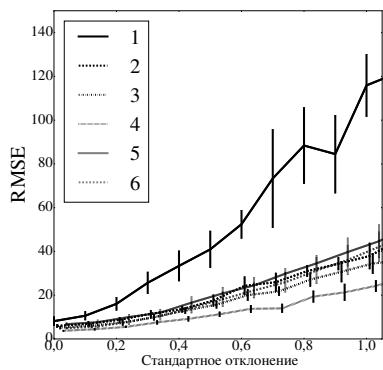
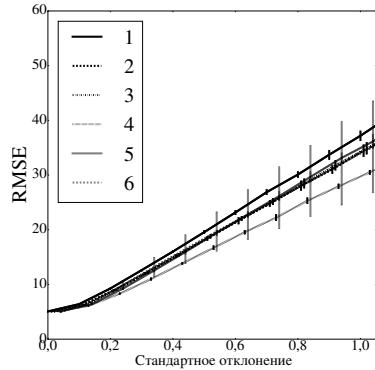
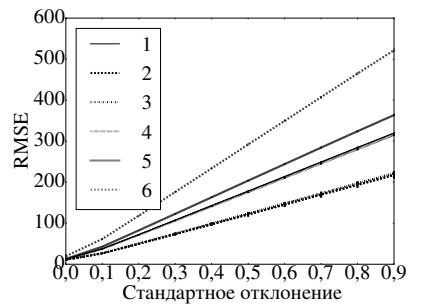
Рис. 5.4. *a*Рис. 5.4. *б*Рис. 5.4. *в*

Рис. 5.5.

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 (2.2) от нормального.

Алгоритм 6 показал низкое качество (5.2) 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 [?]: 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 [?].

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

5.0.3. Оптимизация гиперпараметров

Для анализа рассматриваемых алгоритмов оптимизации гиперпараметров был проведен ряд вычислительных экспериментов на выборках MNIST [?], WISDM [?], 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 :

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

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.

4. Внешний критерий качества моделей E_σ при возмущении параметров модели:

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

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. 3.1. Пример поведения траекторий параметров под действием алгоритмов приведен на Рис. ???. В качестве функций Q и L рассматривались функции кросс-валидации (3.4) с $k = 4$ и вариационной оценки правдоподобия (3.5).

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 γ 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 $\gamma_{\mathbf{h}}$ понижалась. Для алгоритма DrMad параметр τ_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. ???. 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. ??, ???. 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 (3.5) 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. ??, ??, ??, ?. 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 ... Гиперпараметры, достигающие наибольших значений вариационной оценки (3.5) 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 случайный поиск в случае большого количества гиперпараметров. Наилучшие результаты были получены жадным поиском. Алгоритм DrMad, показавший результаты хуже, чем жадный алгоритм и HOAG, является упрощенной версией алгоритма, представленного в [?]. Данный алгоритм позволяет проводить оптимизацию не только гиперпараметров, но и параметров алгоритма оптимизации T . Поэтому возможным развитием метода DrMad является получение оптимальных значений параметров оптимизации.

5.0.4. Модели парофраза (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

Алгоритм	L, Q	$Q(\theta, h)$	Сходимость	E	$E_{0.25}$	$E_{0.5}$
<i>Синтетическая выборка</i>						
Случайный поиск	(3.4)	-171.6	26.2 ± 20.0	1.367	?	?
Жадная оптимизация	(3.4)	-172.5	30.0 ± 24.5	1.421	?	?
DrMAD	(3.4)	-174.1	40.2 ± 16.1	1.403	?	?
HOAG	(3.4)	-174.7	29.4 ± 24.0	1.432	?	?
Случайный поиск	(3.5)	-63.5	32.4 ± 18.7	1.368	?	?
Жадная оптимизация	(3.5)	-25.5	1.2 ± 0.4	1.161	?	?
DrMAD	(3.5)	-25.1	10.6 ± 0.8	1.157	?	?
HOAG	(3.5)	-25.8	10.8 ± 1.5	1.141	?	?
<i>WISDM</i>						
Случайный поиск	(3.4)	-1086661.1	22.0 ± 19.3	0.660	?	?
Жадная оптимизация	(3.4)	-1086707.1	15.4 ± 17.2	0.707	?	?
DrMAD	(3.4)	-1086708.2	29.2 ± 8.0	0.694	?	?
HOAG	(3.4)	-1086733.5	28.2 ± 7.13	0.701	?	?
Случайный поиск	(3.5)	-35420.4	14.4 ± 7.8	0.732	?	?
Жадная оптимизация	(3.5)	-3552.9	1.0 ± 0.0	0.702	?	?
DrMAD	(3.5)	-26091.4	50.0 ± 0.0	0.729	?	?
HOAG	(3.5)	-16566.6	49.0 ± 0.0	0.733	?	?
<i>MNIST</i>						
Случайный поиск	(3.4)	-3305.1	13.3 ± 8.1	0.0179	?	?
Жадная оптимизация	(3.4)	-3416.7	13.8 ± 9.3	0.0193	?	?
DrMAD	(3.4)	?	?	?	?	?
HOAG	(3.4)	-3748.6	8.6 ± 7.3	0.0217	?	?
Случайный поиск	(3.5)	-1304556.4	14.2 ± 5.7	0.0187	?	?
Жадная оптимизация	(3.5)	-11136.2	7.8 ± 3.6	0.0231	?	?
DrMAD	(3.5)	?	?	?	?	?
HOAG	(3.5)	-280061.6	24.0 ± 0.0	0.0189	?	?

Таблица 5.3. Результаты экспериментов

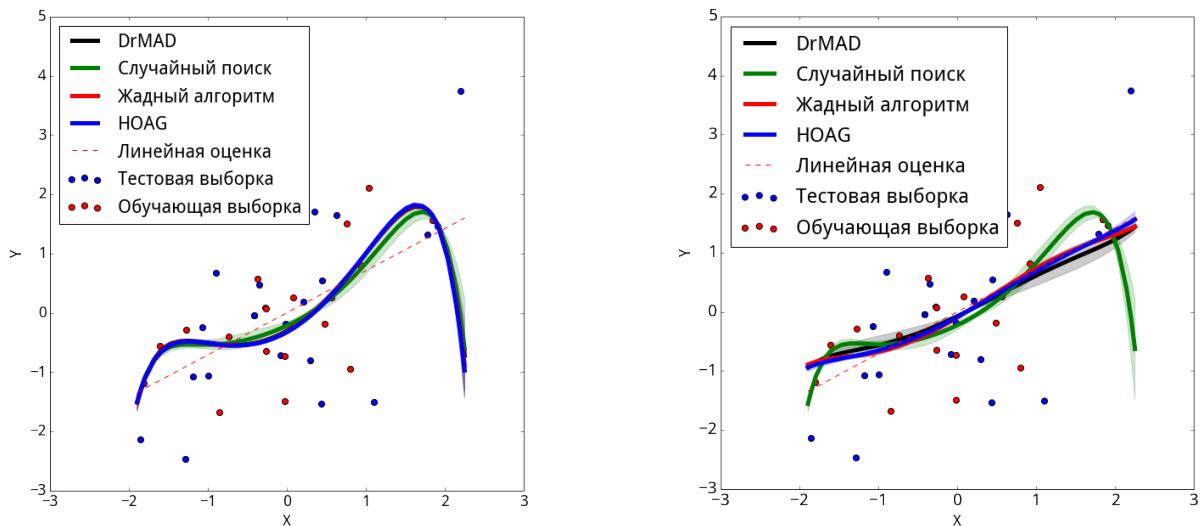


Рис. 5.6. Графики итоговых полиномов для синтетической выборки: а — кросс-валидация, б — вариационная оценка

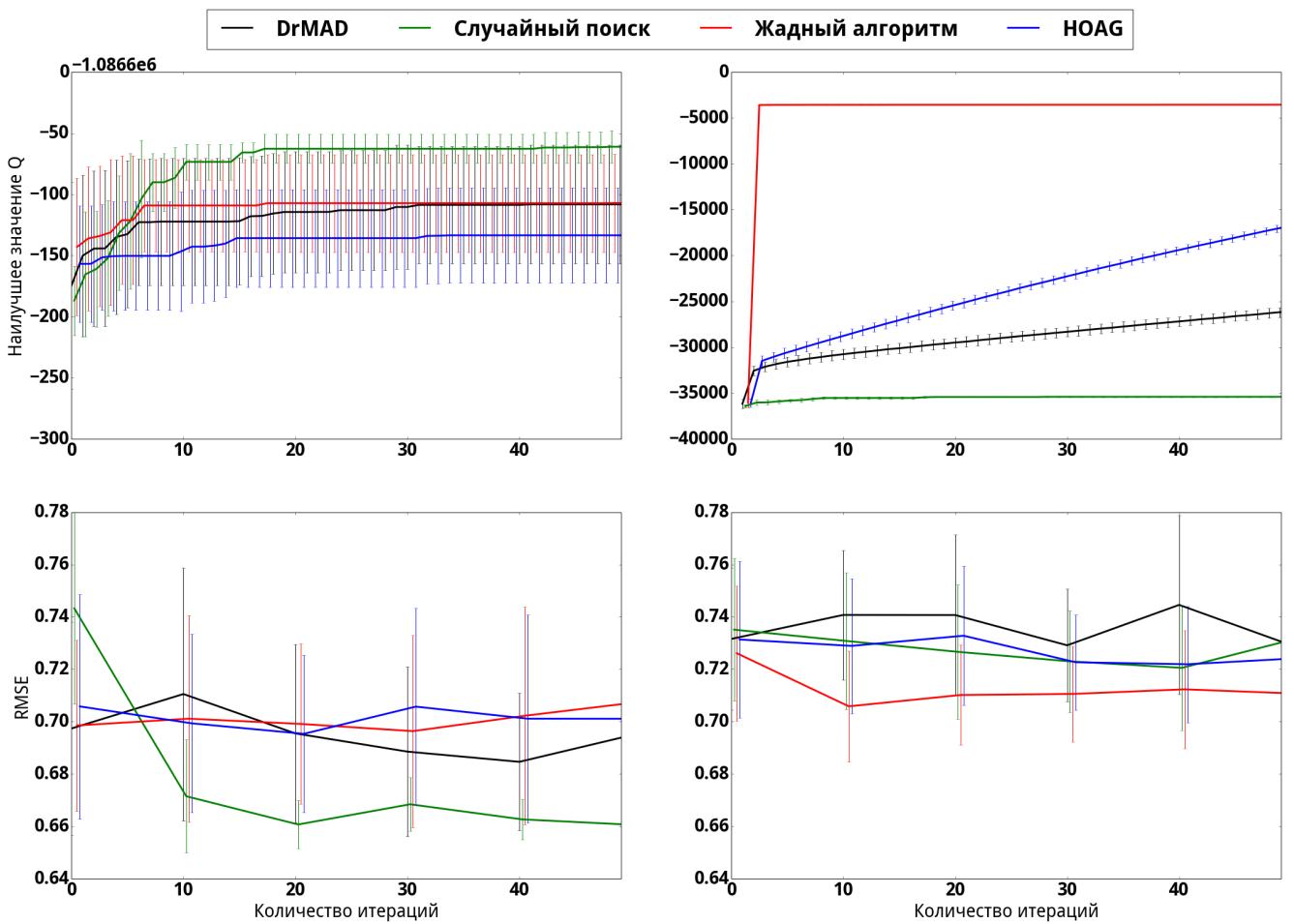


Рис. 5.7. Графики зависимости функции \hat{Q} и качества модели от количества итераций оптимизации для кросс-валидации (слева), вариационная оценка (справа)

алгоритмов использовались линейная регрессия, метод ближайших соседей, решающее дерево и модификация метода опорных векторов SVC. Базовые алгоритмы взяты из библиотеки sklearn. Дополнительно были построены рекуррентная нейросеть с одним скрытым слоем [?] и нейросеть с одним скрытым слоем и вариационной оптимизацией параметров [?, ?].

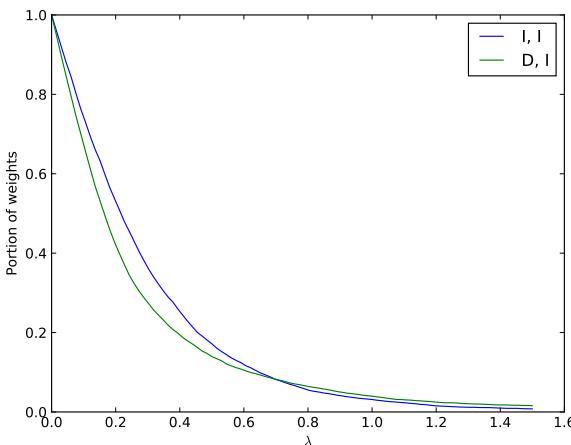


Рис. 5.8. Доля неудаленных параметров сети в зависимости от порогового значения λ для скалярного (I) и диагонального (D) вида апостериорной матрицы ковариаций

На рис. ?? и ?? представлена зависимость оценки правдоподобия L (??) от параметра λ . Для обоих случаев существует оптимальное значение λ , минимизирующее L ; модели с таким параметром будут оптимальными. На рис. ??, ??, ?? и ?? отображены зависимости качества модели от λ 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. 5.8 видно, что при малых λ из сети с диагональной апостериорной матрицей ковариаций удаляется больше весов, а при больших λ — меньше, что говорит о лучшем отборе параметров такой моделью.

5.0.5. 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 генерированы таким образом чтобы параметры сети были мультиколлинеарными. Генерация данных состояла из двух этапов. На первом этапе генерировался вектор

параметров $\mathbf{w}_{\text{synthetic}}$:

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

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

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 5.4. 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_{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_{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.

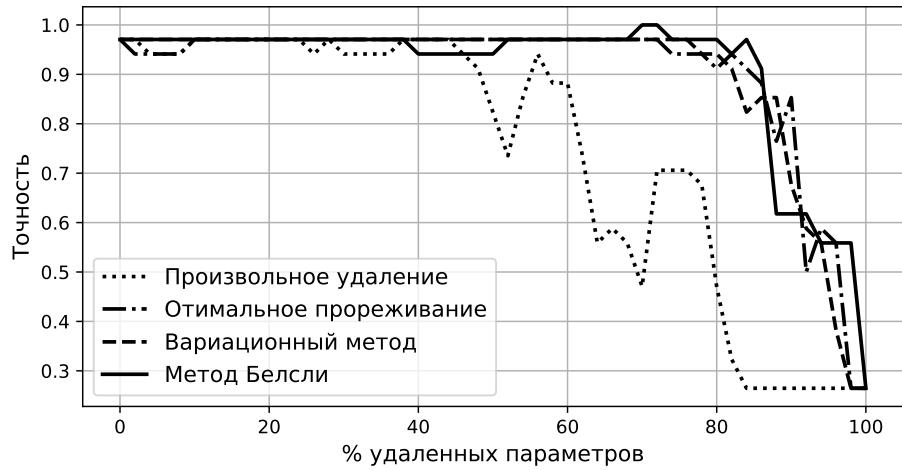


Рис. 5.9. Качество прогноза при удалении параметров на выборке Wine

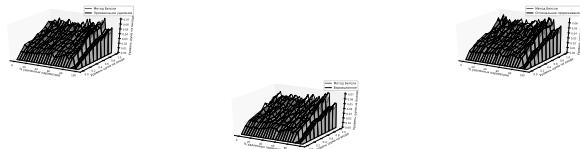


Рис. 5.10. Влияние шума в начальных данных на шум выхода нейросети на выборке Wine: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод

На рис. 5.9 показано как меняется точность прогноза R_{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.

На рис. 5.10 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. 5.11 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.

На рис. 5.12 показаны поверхности изменения уровня шума ответов нейро-

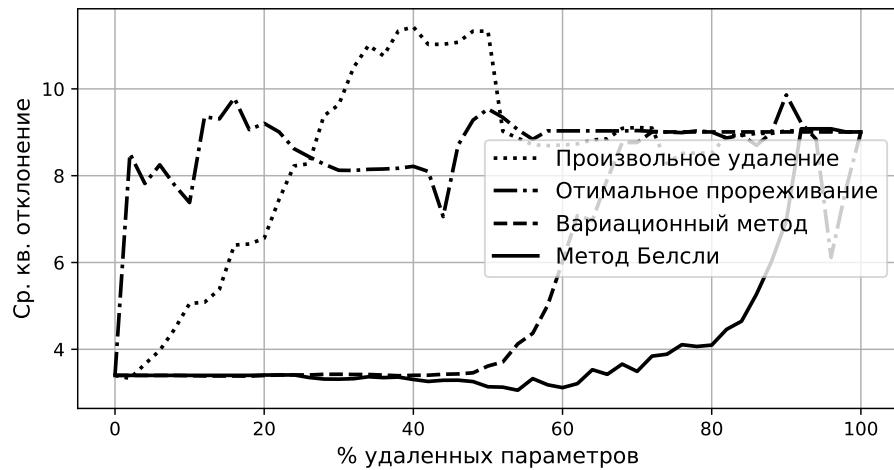


Рис. 5.11. Качество прогноза при удаление параметров на выборке Boston

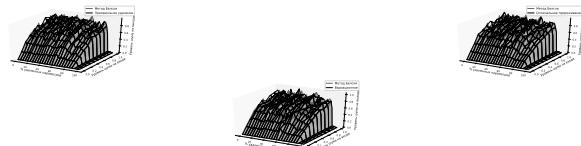


Рис. 5.12. Влияние шума в начальных данных на шум выхода нейросети на выборке Boston: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод

сети при изменении процента удаленных параметров и уровня шума входных данных для разных методов прореживания. 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. 5.13 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. 5.14 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, так-как поверхность которая соответствует методу Белсли ниже других поверхностей.

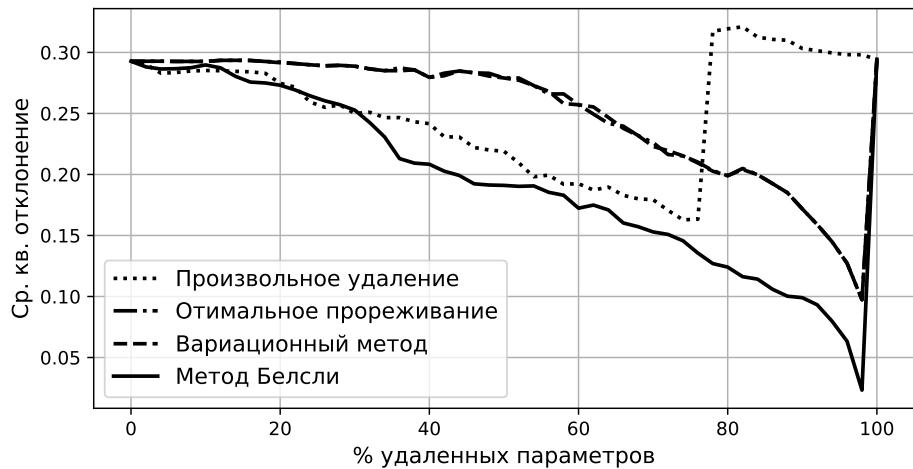


Рис. 5.13. Качество прогноза при удаление параметров на синтетической выборке

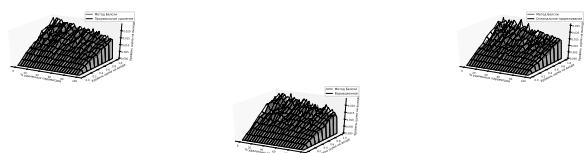


Рис. 5.14. Влияние шума в начальных данных на шум выхода нейросети на выборке Boston: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод

Список иллюстраций

5.1	Зависимость ошибки от числа нейронов	39
5.2	Зависимость ошибки от размера обучающей выборки	39
5.3	Результаты эксперимента по исследованию скорости процесса обучения	40
5.4	42
5.5	44
5.6	Графики итоговых полиномов для синтетической выборки: а — кросс-валидация, б — вариационная оценка	49
5.7	Графики зависимости функции \hat{Q} и качества модели от количества итераций оптимизации для кросс-валидации: кросс-валидация (слева), вариационная оценка (справа)	49
5.8	Доля неудаленных параметров сети в зависимости от порогового значения λ для скалярного (I) и диагонального (D) вида апостериорной матрицы ковариаций	50
5.9	Качество прогноза при удаление параметров на выборке Wine .	52
5.10	Влияние шума в начальных данных на шум выхода нейросети на выборке Wine: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод	52
5.11	Качество прогноза при удаление параметров на выборке Boston .	53
5.12	Влияние шума в начальных данных на шум выхода нейросети на выборке Boston: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод	53
5.13	Качество прогноза при удаление параметров на синтетической выборке	54
5.14	Влияние шума в начальных данных на шум выхода нейросети на выборке Boston: а — Произвольное удаление параметров, б — Оптимальное прореживание, в — Вариационный метод	54

Список таблиц

3.1	Основные свойства рассматриваемых алгоритмов	38
5.1	Описание выборок для экспериментов	43
5.2	Результаты эксперимента	43
5.3	Результаты экспериментов	48
5.4	Описание выборок	51