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

Введение

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

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

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

В качестве порождающих моделей в сетях глубокого обучения выступают ограниченные машины Больцмана [3] и автокодировщики [17]. В работе [18] рассматриваются некоторые типы регуляризации автокодировщиков, позволяющие формально рассматривать данные модели как порождающие модели с использованием байесового вывода. В работе [19] также рассматриваются регуляризованные автокодировщики и свойства оценок их правдоподобия. В работе [20] предлагается обобщение автокодировщика с использованием вариационного байесовского вывода [2]. В работе [21] рассматриваются модификации вариационного автокодировщика и ступенчатых сетей (англ. ladder network) [22]

для случая построения многослойных порождающих моделей.

В качестве критерия выбора модели в ряде работ [23, 2, 24, 25, 26, 27] выступает правдоподобие модели. В работах [24, 25, 26, 27] рассматривается проблема выбора модели и оценки гиперапараметров в задачах регрессии. Альтернативным критерием выбора модели является минимальная длина описания [1], являющаяся показателем статистической сложности модели и заданной выборки. В работе [28] рассматривается перечень критериев сложности моделей глубокого обучения и их взаимосвязь. В работе [29] в качестве критерия сложности модели выступает показатель нелинейности, характеризуемый степенью полинома Чебышева, аппроксимирующего функцию. В работе [30] 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 [31] 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 [32], в которой исследуется устойчивость классификации модели под действием шума.

Одним из методов получения приближенного значения интеграла правдоподобия является вариационный метод получения нижней оценки интеграла [2]. В работе [33] рассматривается стохастическая версия вариационного метода. В работе [34] рассматривается алгоритм получения вариационной нижней оценки правдоподобия для оптимизации гиперпараметров моделей глубокого обучения. В работе [35] рассматривается получение вариационной нижней оценки интеграла с использованием модификации методов Монте-Карло. В работе [36] рассматривается стохастический градиентный спуск в качестве оператора, порождающего распределение, аппроксимирующее апостериорное распределение параметров модели. Схожий подход рассматривается в работе [37], где также рассматривается стохастический градиентный спуск в качестве оператора, порождающего апостериорное распределение параметров. В работе [38] предлагается модификация стохастического градиентного спуска, аппроксимирующая апостериорное распределение.

Альтернативным методом выбора модели является выбор модели на основе скользящего контроля [39, 24]. Проблемой такого подхода является возможная высокая вычислительная сложность [40, 41]. В работах [42, 43] 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 [23, 2]. В работе [24] рассматривается оптимизация гиперпарамтров с использованием метода скользящего контроля и методов оптимизации интеграла правдодобия моделей, отмечается низкая скорость схо-

димости гиперпараметров при использовании метода скользящего контроля. В ряде работ [44, 45] рассматриваются градиентные методы оптимизации гиперпараметров, позволяющие оптимизировать большое количество гиперпараметров одновременно. В работе [44] предлагается метод оптимизации гиперпараметров с использованием градиентного спуска с моментом, в качестве оптимизируемого функционала рассматривается ошибка на валидационной части выборки. В работе [46] предлагается метод аппроксимации градиента функции потерь по гиперпараметрам, позволяющий использовать градиентные методы в задаче оптимизации гиперпараметров на больших выборках. В работе [47] предлагается упрощенный метод оптимизации гиперпараметров с градиентным спуском: вместо всей истории обновлений параметров для оптимизации используется только последнее обновление. В работе [37] 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 оптимльной 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 [23, 2, 36]. Рассматриваются суперпозиции порождающей и разделяющей моделей [13, 14, 15]. Для получения оценок вариационных оценок правдоподобия модели используется метод, основанный на градиентном спуске [37, 36]. В качестве метода получения модели субоптимальной сложности используется метод Automatic Relevance Determination [23, 48] с использоваением градиентных методов оптимизации гиперпараметров [44, 45, 47, 46].

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 глубого обучения. Предложены критерии субоптимльной 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. Всероссийская конференция “Интеллектуализация обработки информации” ММРО-17, 2016 [**?**].
2. 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.

Во второй главе предлагается алгоритм построения модели глубокого обучения.

В третьей главе рассматриваются методы оценки параметров модели. Предлагаются критерии прореживания параметров.

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

Глава 1
Постановка задачи

Постановка задачи

Глава 2
Обзор

Обзор

Проблему выбора структуры модели глубокого обучения можно сформулировать следующим образом: решается задача классификации или регрессии на заданной выборке. 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 нелинейный функций. Под структурой модели понимается значения структурных параметров модели, т.е. параметров модели, характеризующий вид итоговой суперпозиции.

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”.

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.

В данном обзоре методы порождения и выбора обобщенно-линейных моделей не рассматривается в силу общности рассматриваемой задачи.

2.1. Постановка задачи

Задана выборка

$$\mathfrak{D} = \{(\mathbf{x}_i, y_i)\}, i = 1, \ldots, m, \tag{2.1}$$

состоящая из множества пар «объект-метка»,

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

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

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

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

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

Для каждой модели определена функция правдоподобия $p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f})$, где $\mathbf{x}$ — строка матрицы $\mathbf{X}$, $\mathbf{y}$ — вектор меток зависимой переменной y.

Будем полагать, что множество рассматриваемых моделей задается некоторой функцией $\mathfrak{F}(\beta)$. Для каждой модели $\mathbf{f}$ из конечного множества моделей $\mathfrak{F}(\beta)$ задано априорное непрерывное распределение параметров $p(\mathbf{w}|\alpha)$.

Пусть задана дифференцируемая функция Q, определяющая качество модели.

Определение Модель классификации $\hat{\mathbf{f}}$ назовем оптимальной среди моделей $\mathfrak{F}$, если достигается максимум качества Q:

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f} \in \mathfrak{F}} Q(\mathbf{f}).$$

Определение Назовем оператором оптимизации алгоритм T выбора вектора параметров $\mathbf{w}'$ по предыдущему значению параметров модели $\mathbf{w}$:

$$\mathbf{w}' = T(\mathbf{w}, \gamma),$$

где γ — вектор параметров оптимизации.

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

$$\mathbf{f} = \arg\min_{\mathbf{f}' \in \mathfrak{F}} \min_{\gamma, \alpha} Q, \tag{2.3}$$

$$\mathbf{w} = T(\mathbf{w}, \gamma).$$

В дальнейшем будем использовать следующие наименования для групп параметров, участвующих в задаче оптимизации (4.4):

1. $\mathbf{w}$ — множество параметров модели.
2. α — множество гиперпараметров модели.
3. β — множество структурных параметров.
4. γ — метапараметры.

2.2. Метаоптимизация

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

2.2.1. Теоретические основания метаобучения

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

$$\mathcal{U}_{\mathcal{D}}(\theta_I) := \max_{Q} \mathbb{E}_{\mathbf{x}\sim P_{\mathcal{D}}} \left[\log \sum_{\mathbf{h}} P_{\theta_I}(\mathbf{x}|\mathbf{h}) Q(\mathbf{h}) \right]$$

В работе [?] рассматриваются подходы к сэмплированию моделей глубокого обучения. Под *сэмплированием* понимается порождение нескольких экземпляров модели из заданного распределения для дальнейшего выбора наилучшей модели. 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.

2.2.2. 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:

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

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

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

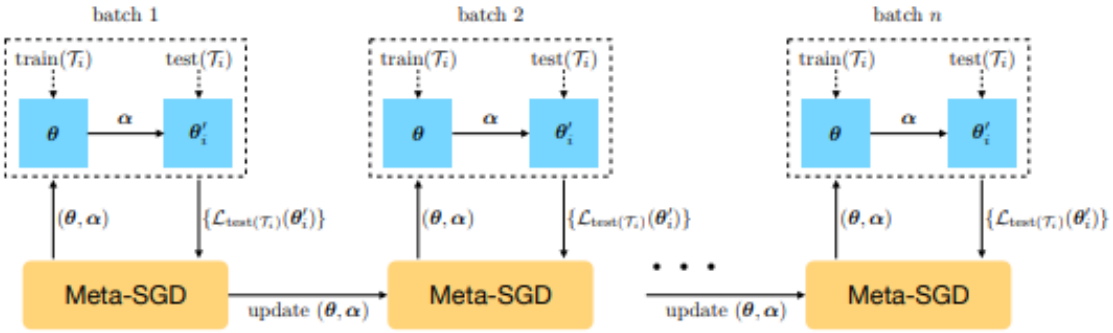
$$\forall t \geq 1: \quad w_{ij}(t+1) = w_{ij}(t) + \triangle(t)\, g[\, \|adr(w_{ij}) - mod(t)\|^2\,] \qquad (8)$$

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

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

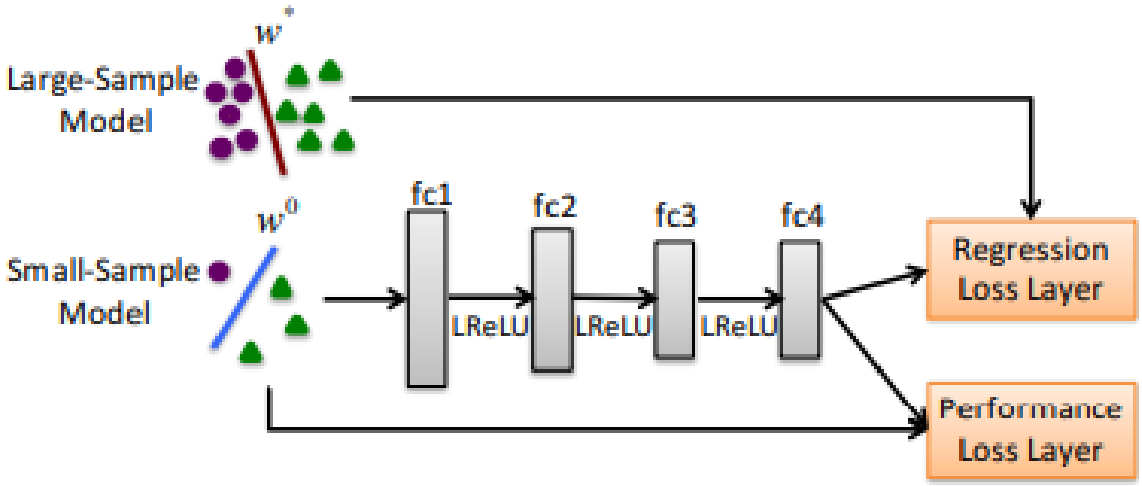
Предлагается продолжение подхода, позволяющая рекуррентно продолжать анализ модели и порождать мета-мета-...-анализ.

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:

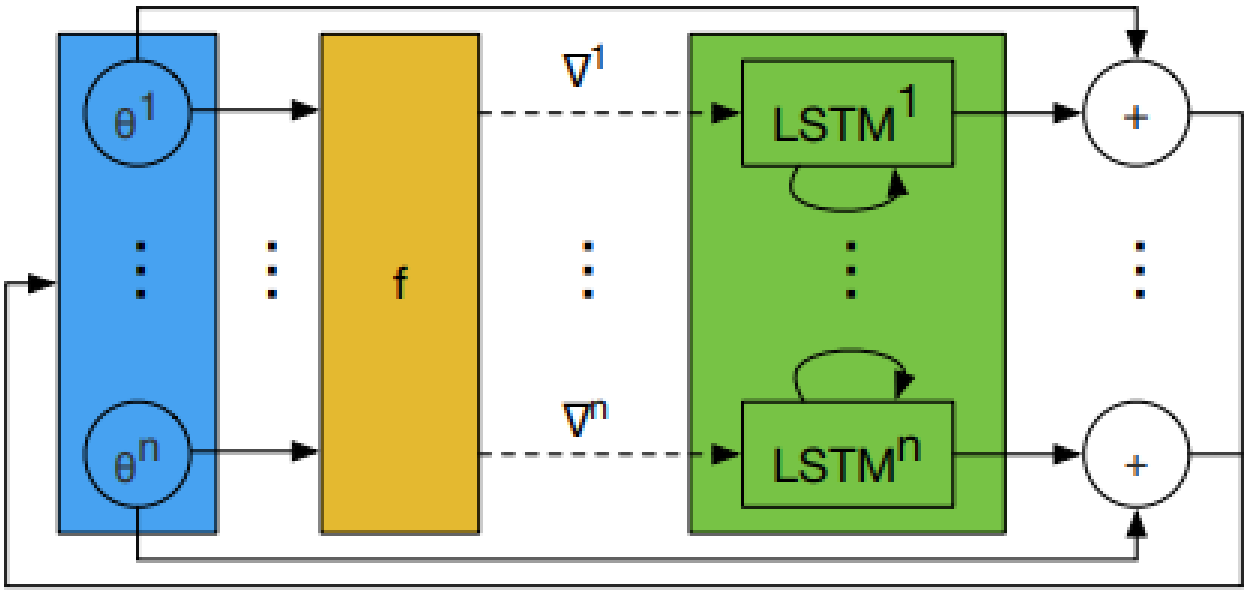


В работе [?] рассматривается задача восстановления параметров модели по параметрам слабо обученной модели:

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



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



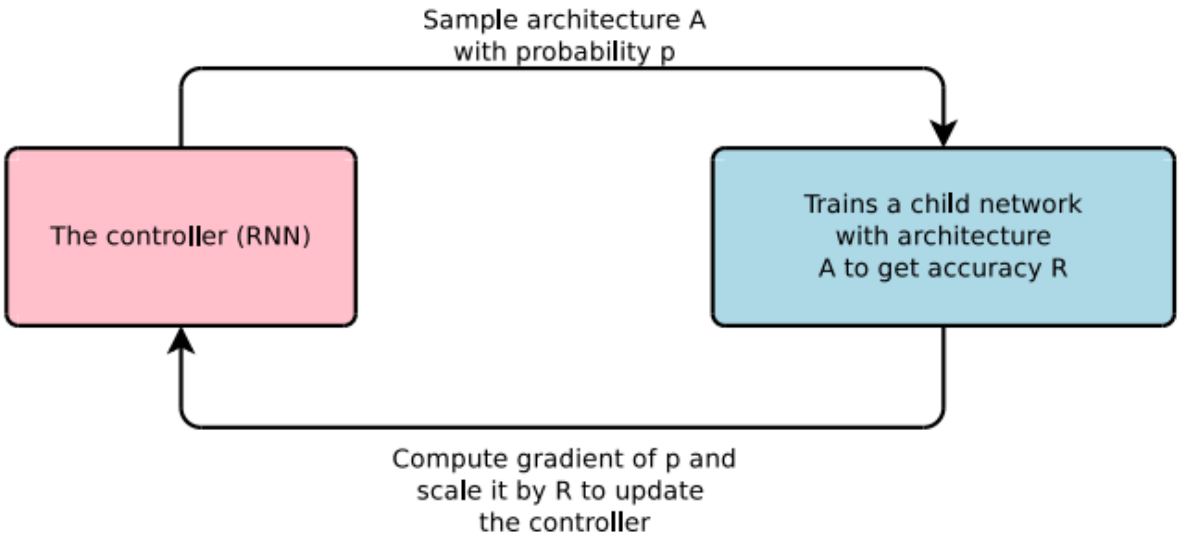
2.2.3. Перебор структур

В работе [?] рассматривается задача порождения сверточных нейронных сетей. Предлагается проводить поиск оптимальной структуры сети по восходящему по сложности порядку: начиная от сетей с одни 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.

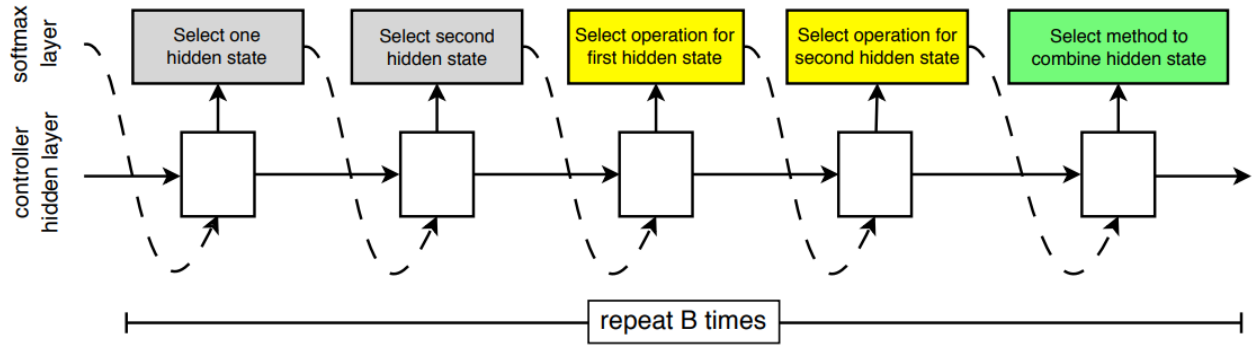
В работе [?] рассматривается задача выбора архитектуры с помощью большого количества параллельных запусков обучения моделей, предлагаются критерии ранней остановки оптимизации обучения моделей.

2.2.4. Обучение с подкреплением

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

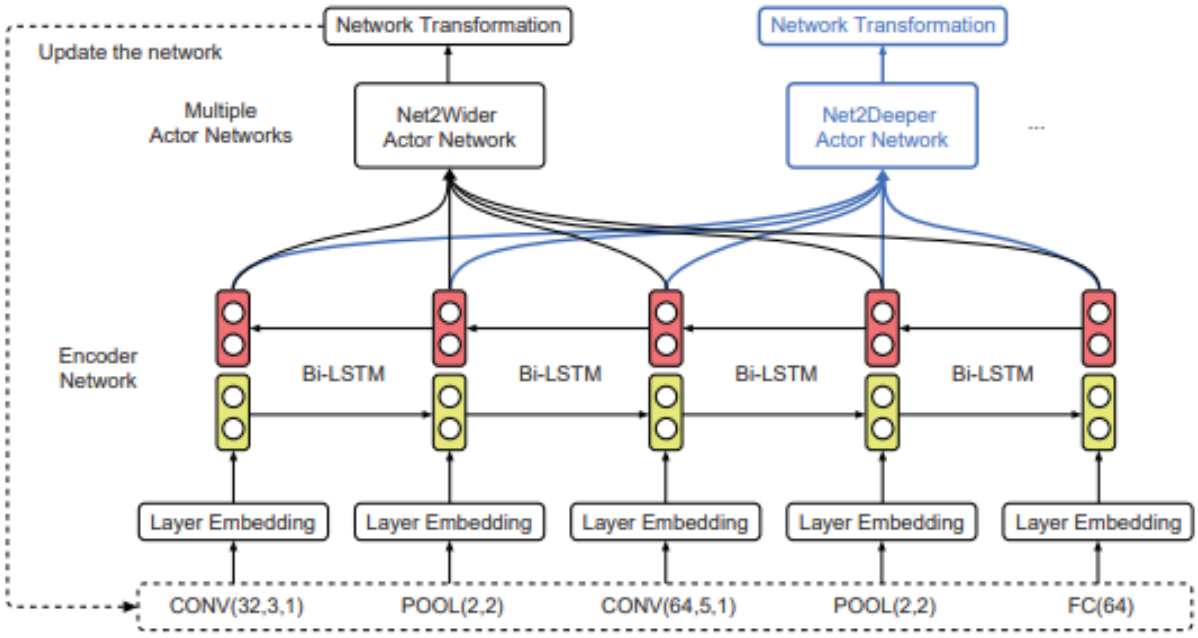


В работе [?] предлагается построение регрессионной модели для оценки финального качества модели и ранней остановки оптимизации моделей. Данный подход позволил существенно ускорить поиск моделей, представленный в работе [?]. В работе [?] рассматривается задача переноса архитектуры нейросети, обученной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



2.3. Адаптивное изменение структуры

В данном разделе собраны методы изменения структуры существующей модели.

Алгоритмы наращивания и прореживания параметров модели В работе [8] предлагается удалять неинформативные параметры модели, где в качестве показателя информативности выступает следующий фнуционал:

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

В работе [?] было предложено развитие данного метода. В данной работе, в отличие от [8] не вводится предположений о дигональности Гессиана функции ошибок, поэтому удаление неинформативных параметров модели производится точнее.

В работе [34] 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 равенста 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:

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

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

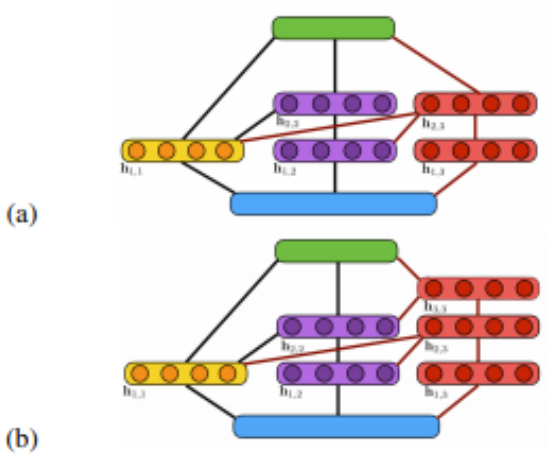
$$s \sim \mathcal{C}^+(0, \tau_0); \qquad \tilde{z}_i \sim \mathcal{C}^+(0, 1); \qquad \tilde{w}_{ij} \sim \mathcal{N}(0, 1); \qquad w_{ij} = \tilde{w}_{ij}\tilde{z}_i s,$$

Смежной темой к прореижванию 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 [10] предлагается итеративное использование регуляризации типа DropOut [?] для прореживания модели. В работах [?, ?] используются методы снижения вычислительной точности представления парамеров 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 [?, 16] предлагается наращивание моделей, основанное на бустинге. В работе рассматривается задача построения нейросетевых моделей специального типа:

$$f_{t+1} = \sigma(f_t) + f_t,$$

приводится параметризация модели, позволяющая рассматривать декомпозировать модель на слабые классификаторы. В работе [16] 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 радемахереовской сложности:

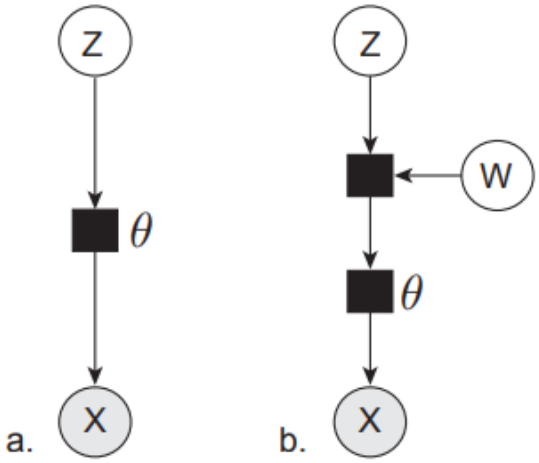
$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \frac{1}{m}\,\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{G}}\sum_{i=1}^{m}\sigma_i h(x_i)\right],$$

2.4. Байесовские методы порождения и выбора моделей

2.4.1. Автоматическое определение релевантности параметров

В работе [44] 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) [23].

В работе [48] 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:



2.4.2. Суррогаты

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

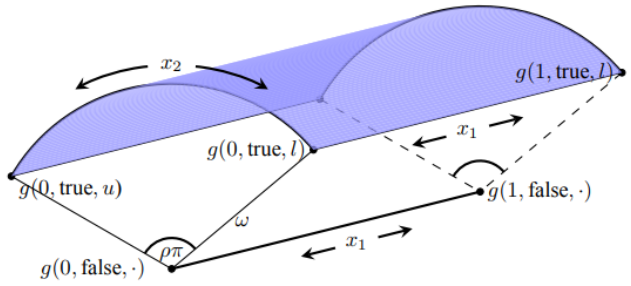
Одна из основных проблем использования гауссового процесса как суррогатной модели — кубическая сложность оптимизации. В работе [?] предлагается использовать случайные подпространства гиперпараметров для ускоренной оптимизации. В работе [?] предлагается кобминация из множества гауссовых моделей и линейной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

В работе [?] предлагается 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 веротяности.

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

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

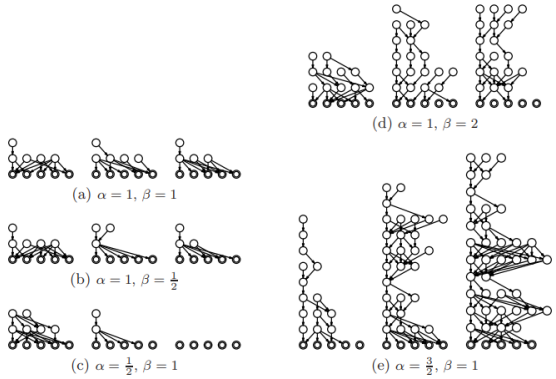
где d — геодезическое расстояние между вершинами графа, r — некоторая вещественная функция (наверно положительно определенная, но это не указано явно в статье). В работе [?] рассматривается задач выбора структуры нейросети, предлагается ядро специального вида, позволюящее учитывать только те гиперпапараметры,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



2.4.3. 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:

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

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

2.4.4. Порождающие модели

В работе [13] было предложено обобщение вариационного автокодировщика на случай частичного обучения: итоговая модель вариационного автокодировщика является порождающей моделью, учитывающий 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.

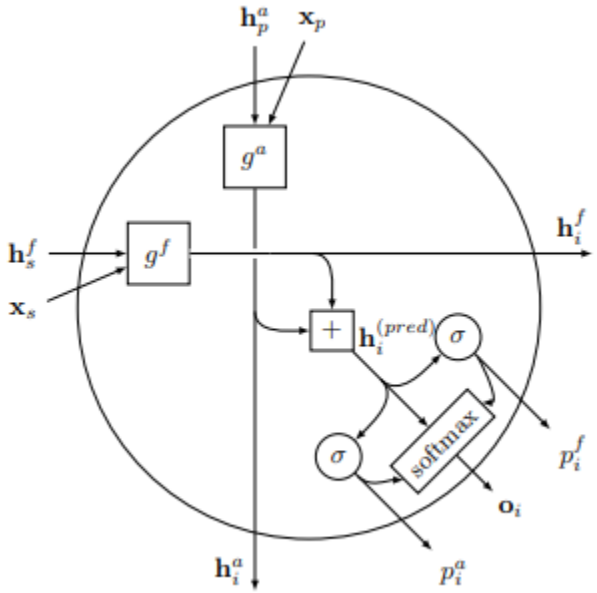
2.4.5. Состязательные модели

2.5. 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-reccurent), т.е. на сетях, отдельно предсказывающих глубину и ширину уровней деревьев.



2.6. Эвристические и прикладные методы

2.6.1. 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.

2.6.2. Структуры сетей специального вида

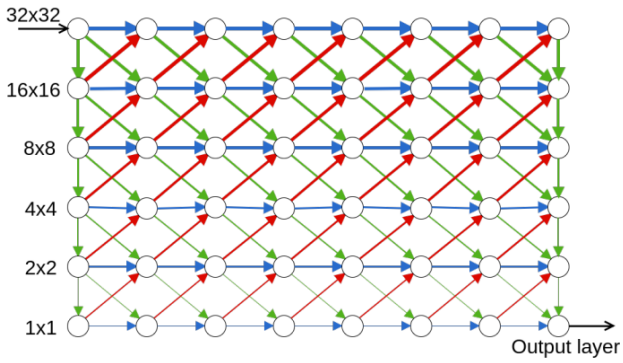
В данном разделе представлены работы по поиску оптимальной структуры сети, описывающие частные случаи поиска оптимальных моделей со структурами специального вида.

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 энергитических 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 суперсет, позволяющие выбрать структуру нейросети за органиченное 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 3
Выбор модели с использованием вариационного вывода

Выбор модели с использованием вариационного вывода

Задана выборка

$$\mathfrak{D} = \{(\mathbf{x}_i, y_i)\}, i = 1, \ldots, m, \tag{3.1}$$

состоящая из множества пар «объект-метка»

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

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

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

$$\mathbf{f}(\mathbf{w}, \mathbf{X}) = \mathbf{f}_1(\mathbf{f}_2(\ldots \mathbf{f}_K(\mathbf{w}, \mathbf{X}))) : \mathbb{R}^{m \times n} \to \mathbb{Y}^m, \tag{3.2}$$

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

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

Сложностью модели $\mathbf{f}$ назовем правдоподобие модели:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{f}) = \int_{\mathbf{w} \in \mathbb{R}^u} p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}) p(\mathbf{w}|\mathbf{f}) d\mathbf{w}. \tag{3.3}$$

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

Модель классификации $\mathbf{f}$ назовем оптимальной среди моделей $\mathfrak{F}$, если достигается максимум интеграла (3.3).

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

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}) p(\mathbf{w}|\mathbf{f})}{p(\mathbf{y}|\mathbf{X}, \mathbf{f})}. \tag{3.4}$$

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

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{f}) = (2\pi)^{-\frac{m}{2}} \exp\Big(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\mathsf{T}(\mathbf{y} - \mathbf{X}\mathbf{w})\Big), \tag{3.5}$$

априорное распределение параметров модели имеет вид

$$p(\mathbf{w}|\mathbf{f}) = (2\pi)^{-\frac{n}{2}}|\mathbf{A}|^{\frac{1}{2}}\exp(-\frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{A}\mathbf{w}). \tag{3.6}$$

Правдоподобие модели (3.3) в этом примере вычисляется аналитически [?]:

$$p(\mathbf{y}|\mathbf{X},\mathbf{f}) = (2\pi)^{-\frac{m}{2}}|\mathbf{A}|^{\frac{1}{2}}|\mathbf{H}|^{-\frac{1}{2}}\exp\Big(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\hat{\mathbf{w}})^\mathsf{T}(\mathbf{y}-\mathbf{X}\hat{\mathbf{w}})\Big)\exp\Big(-\frac{1}{2}\hat{\mathbf{w}}^\mathsf{T}\mathbf{A}\hat{\mathbf{w}}\Big), \tag{3.7}$$

где $\hat{\mathbf{w}}$ — значение наиболее вероятных (3.4) параметров модели:

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

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

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

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

В качестве функции, приближающей логарифм интеграла (3.3), будем рассматривать его нижнюю оценку, полученную при помощи неравенства Йенсена [2]. Получим нижнюю оценку логарифма правдоподобия модели, используя неравенство

$$\log p(\mathbf{y}|\mathbf{X},\mathbf{f}) = \int_\mathbf{w} q(\mathbf{w})\log\frac{p(\mathbf{y},\mathbf{w}|\mathbf{X},\mathbf{f})}{q(\mathbf{w})}d\mathbf{w} + \mathrm{D}_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{y},\mathbf{X},\mathbf{f})\big) \geq \tag{3.8}$$

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

$$= -\mathrm{D}_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f})\big) + \int_\mathbf{w} q(\mathbf{w})\log p(\mathbf{y}|\mathbf{X},\mathbf{w},\mathbf{f})d\mathbf{w},$$

где $\mathrm{D}_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f})\big)$ — расстояние Кульбака–Лейблера между двумя распределениями:

$$\mathrm{D}_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f})\big) = -\int_\mathbf{w} q(\mathbf{w})\log\frac{p(\mathbf{w}|\mathbf{f})}{q(\mathbf{w})}d\mathbf{w},$$

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

Вариационной оценкой логарифма правдоподобия модели (3.3) $\log p(\mathbf{y}|\mathbf{X},\mathbf{f})$ называется оценка $\log\hat{p}(\mathbf{y}|\mathbf{X},\mathbf{f})$, полученная аппроксимацией неизвестного апостериорного распределения $p(\mathbf{w}|\mathbf{y},\mathbf{X},\mathbf{f})$ заданным распределением $q(\mathbf{w})$.

Будем рассматривать задачу нахождения вариационной оценки как задачу оптимизации. Пусть задано множество распределений $Q = \{q(\mathbf{w})\}$. Сведем

задачу нахождения наиболее близкой вариационной нижней оценки интеграла (3.3) к оптимизации вида

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

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

Оценка (4.7) является нижней, поэтому может давать некорректные оценки для правдоподобия (3.3). Для того, чтобы оценить величину этой ошибки, докажем следующее утверждение.

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

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

логарифма интеграла (3.3) эквивалентна минимизации расстояния Кульбака–Лейблера между распределением $q(\mathbf{w}) \in Q$ и апостериорным распределением параметров $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})$:

$$\hat{q} = \arg\max_{q \in Q} \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w} \Leftrightarrow \hat{q} = \arg\min_{q \in Q} \mathrm{D}_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})\big), \tag{3.9}$$

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

Таким образом, задача нахождения вариационной оценки, близкой к значению интеграла (3.3) сводится к поиску распределения $\hat{q}$, аппроксимирующего распределение $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})$ наилучшим образом. Доказательство утверждения 1 см. в Приложении.

Модель $\mathbf{f}$ назовем субоптимальной на множестве моделей $\mathfrak{F}$ по множеству распределений Q, если модель доставляет максимум нижней вариационной оценке интеграла (4.10)

$$\max_{q \in Q} \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w}. \tag{3.10}$$

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

Вариационная оценка (4.7) интерпретируется как оценка сложности модели по принципу минимальной длины описания [1], где первое слагаемое определяет

количество информации для описания выборки, а второе слагаемое — длину описания самой модели [34].

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

3.1. Методы получения вариационной оценки правдоподобия

Ниже представлены методы получения вариационных нижних оценок (3.10) правдоподобия (3.3). В первом подразделе рассматривается метод, основанный на аппроксимации апостериорного распределения $p(\mathbf{w}|\mathbf{y},\mathbf{X},\mathbf{f})$ (3.4) многомерным гауссовым распределением с диагональной матрицей ковариаций. В последующих разделах рассматриваются методы, основанные на различных модификациях стохастического градиентного спуска.

3.1.1. Аппроксимация нормальным распределением

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

$$q \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}), \tag{3.11}$$

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

Пусть априорное распределение $p(\mathbf{w}|\mathbf{f})$ (3.6) параметров модели задано как нормальное:

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

Тогда оптимизация (4.10) имеет вид

$$\int_{\mathbf{w}} q(\mathbf{w}) \log p(\mathbf{y}|\mathbf{X},\mathbf{w},\mathbf{f}) d\mathbf{w} - D_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f})\big) \to \max_{\mathbf{A}_q,\boldsymbol{\mu}_q}, \tag{3.12}$$

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

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

В качестве приближенного значения интеграла

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

предлагается использовать формулу

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

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

Итоговая функция оптимизации (3.12) имеет вид

$$\mathbf{f} = \arg\max_{\mathbf{A}_q, \boldsymbol{\mu}_q} \sum_{i=1}^{m} \log p(y_i|\mathbf{x}_i, \mathbf{w}_i) - D_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{f})\big). \tag{3.13}$$

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

$$y \sim \mathcal{N}(\mathbf{w}, \mathbf{B}^{-1}), \tag{3.14}$$

$$\mathbf{B}^{-1} = \begin{pmatrix} 2 & 1,8 \\ 1,8 & 2 \end{pmatrix},$$

$$p(\mathbf{w}|\mathbf{f}) = \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

График аппроксимации распределения параметров представлен на рис. **??**,*а*. Как видно из графика, с использованием метода (3.13) получено грубое приближение апостериорного распределения $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{f})$, что может существенно занизить оценку правдоподобия модели.

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

3.1.2. Аппроксимация с использованием градиентного метода

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

Представим неравенство (4.7)

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{f}) \geq \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})}{q(\mathbf{w})} d\mathbf{w} = \mathsf{E}_{q(\mathbf{w})}\big(\log p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{f})\big) - \mathsf{S}\big(q(\mathbf{w})\big), \tag{3.15}$$

где S — энтропия распределения:

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

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

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

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

Оценка распределений производится при оптимизации параметров. Оптимизация выполняется в режиме мультистарта [?], т.е. при запуске оптимизации параметров модели из нескольких разных начальных приближений. Основная проблема такого подхода — вычисление энтропии S распределений $q(\mathbf{w}) \in Q$. Ниже представлен метод получения оценок энтропии (3.19) S и оценок правдоподобия (3.15).

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

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

где r — число оптимизаций, L — оптимизируемая функция потерь

$$L = -\sum_{i=1}^{m} \log p(y_i, \mathbf{w}|\mathbf{x}_i, \mathbf{f}) = -\log\, p(\mathbf{w}|\mathbf{f}) - \sum_{i=1}^{m} \log p(y_i|\mathbf{x}_i, \mathbf{w}, \mathbf{f}). \tag{3.16}$$

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

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

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

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:

$$T(\mathbf{w}) = \mathbf{w} - \gamma \nabla L(\mathbf{w}, \mathbf{y}, \mathbf{X}), \tag{3.17}$$

где γ — длина шага градиентного спуска.

Пусть значения $\mathbf{w}^1, \ldots, \mathbf{w}^r$ — 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 $\mathbf{w}^1, \ldots, \mathbf{w}^r$. Под действием оператора T распределение параметров $\mathbf{w}_1, \ldots, \mathbf{w}_r$ изменяется. Для учета энтропии распределений, полученных в ходе оптимизации, формализуем метод, представленный в [36].

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

$$\gamma < \frac{1}{C_L}, \quad \gamma < \Big(\max_{g \in \{1,\ldots,r\}} \lambda_{\max}(\mathbf{H}(\mathbf{w}^g))\Big)^{-1}, \tag{3.18}$$

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

При выполнении неравенств (3.18) разность энтропий распределений $q'(\mathbf{w}), q(\mathbf{w})$ на смежных шагах почти наверное сходится к следующему выражению:

$$\mathsf{S}\big(q'(\mathbf{w})\big) - \mathsf{S}\big(q(\mathbf{w})\big) \approx \frac{1}{r}\sum_{g=1}^{r}\big(-\gamma\mathrm{Tr}[\mathbf{H}(\mathbf{w}'^g)] - \gamma\mathrm{Tr}[\mathbf{H}(\mathbf{w}'^g)\mathbf{H}(\mathbf{w}'^g)]\big) + o_{\gamma^2\to 0}(1), \tag{3.19}$$

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

Получим итоговую формулу для оценки правдоподобия модели.

Оценка (3.15) на шаге оптимизации τ представима в виде

$$\log\hat{p}(\mathbf{y}|\mathbf{X},\mathbf{f}) \approx \frac{1}{r}\sum_{g=1}^{r} L(\mathbf{w}_\tau^g,\mathbf{X},\mathbf{y}) + \mathsf{S}\big(q^0(\mathbf{w})\big) + \frac{1}{r}\sum_{b=1}^{\tau}\sum_{g=1}^{r}\big(-\gamma\mathrm{Tr}[\mathbf{H}(\mathbf{w}_b^g)] - \gamma^2\mathrm{Tr}[\mathbf{H}(\mathbf{w}_b^g)\mathbf{H}(\mathbf{w}_b^g)] \tag{3.20}$$

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

В [36] предлагается алгоритм приближенного вычисления для выражения, находящегося под знаком суммы в (3.20):

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

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

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

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

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

$$T(\mathbf{w}) = \mathbf{w} - \frac{m}{\hat{m}}\gamma\nabla L(\mathbf{w},\hat{\mathbf{y}},\hat{\mathbf{X}}), \tag{3.21}$$

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

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

Как и версия алгоритма с использованием градиентного спуска (3.21), основной проблемой модифицированного алгоритма оценки интеграла (3.10) является грубость аппроксимации исходного распределения $p(\mathbf{w}|\mathbf{f},\mathfrak{D})$.

Рассмотрим пример 2 (3.14). График аппроксимации распределения $p(\mathbf{w}|\mathbf{y},\mathbf{X},\mathbf{f})$ представлен на рис. **??**,*б*. Как видно из графика, градиентный спуск сходится к моде распределения. При небольшом количестве итераций полученное распределение также слабо аппроксимирует апостериорное распределение. При приближении к точке экстремума снижается вариационная оценка правдоподобия модели, что интерпретируется как возможное начало переобучения [36]. Таким образом, снижение оценки (3.20) можно использовать как критерий остановки оптимизации модели для снижения эффекта переобучения.

На рис. **??** представлена аппроксимация распределения $p(\mathbf{w}|\mathbf{Y},\mathbf{X},\mathbf{f})$ 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 аппроксимирует распределение $p(\mathbf{w}|\mathbf{Y},\mathbf{X},\mathbf{f})$ в силу диагональности матрицы ковариаций. Распределение, полученное с помощью градиентного спуска, слабо аппроксимирует распределение $p(\mathbf{w}|\mathbf{Y},\mathbf{X},\mathbf{f})$, так как сходится к моде.

3.1.3. Аппроксимация с использованием динамики Ланжевена

Для достижения нижней оценки интеграла (3.10), более близкой к реальному значению логарифма интеграла (3.3), чем оценка с использованием градиентного спуска, предлагается использовать стохастическую динамику Ланжевена [38]. Стохастическая динамика Ланжевена представляет собой вариант стохастического градиентного спуска с добавлением гауссового шума:

$$T(\mathbf{w}) = \mathbf{w} - \gamma\nabla L - \frac{m}{\hat{m}}\log p(\hat{\mathbf{y}}|\hat{\mathbf{X}},\mathbf{w},\mathbf{f}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0},\frac{\gamma}{2}\mathbf{I}), \tag{3.22}$$

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 [38]:

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

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

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

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

В отличие от стохастического градиентного спуска стохастическая динамика Ланжевена сходится к апостериорному распределению параметров $p(\mathbf{w}|\mathfrak{D}, \mathbf{f})$ [38, **?**]. График аппроксимации апостериорного распределения с использованием динамики Ланжевена представлен на рис. **??**,*в*. 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 (3.10). 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 4
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 в общем виде введем переобозначение:

$$\boldsymbol{\theta} = \mathbf{w}, \quad \mathbf{h} = [\alpha_1, \ldots, \alpha_u], \tag{4.1}$$

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

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

$$\hat{\boldsymbol{\theta}} = \arg\max\big(-L(\boldsymbol{\theta}, \mathbf{h})\big) = p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \mathbf{A}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\mathbf{A})}{p(\mathbf{y}|\mathbf{X}, \mathbf{A})}. \tag{4.2}$$

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:

$$Q(\boldsymbol{\theta}, \mathbf{h}) = p(\mathbf{y}|\mathbf{X}, \mathbf{A}) = \int_{\mathbf{w}\in\mathbb{R}^u} p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\mathbf{A}) \to \max_{[\alpha_1,\ldots,\alpha_u]\in\mathbb{R}^n}. \tag{4.3}$$

Сформулируем задачу оптимизации гиперпараметров в общем виде. Обозначим за $\mathbf{h} \in \mathbb{R}^h$ вектор гиперпараметров модели (4.1). Обозначим за $\boldsymbol{\theta} \in \mathbb{R}^s$

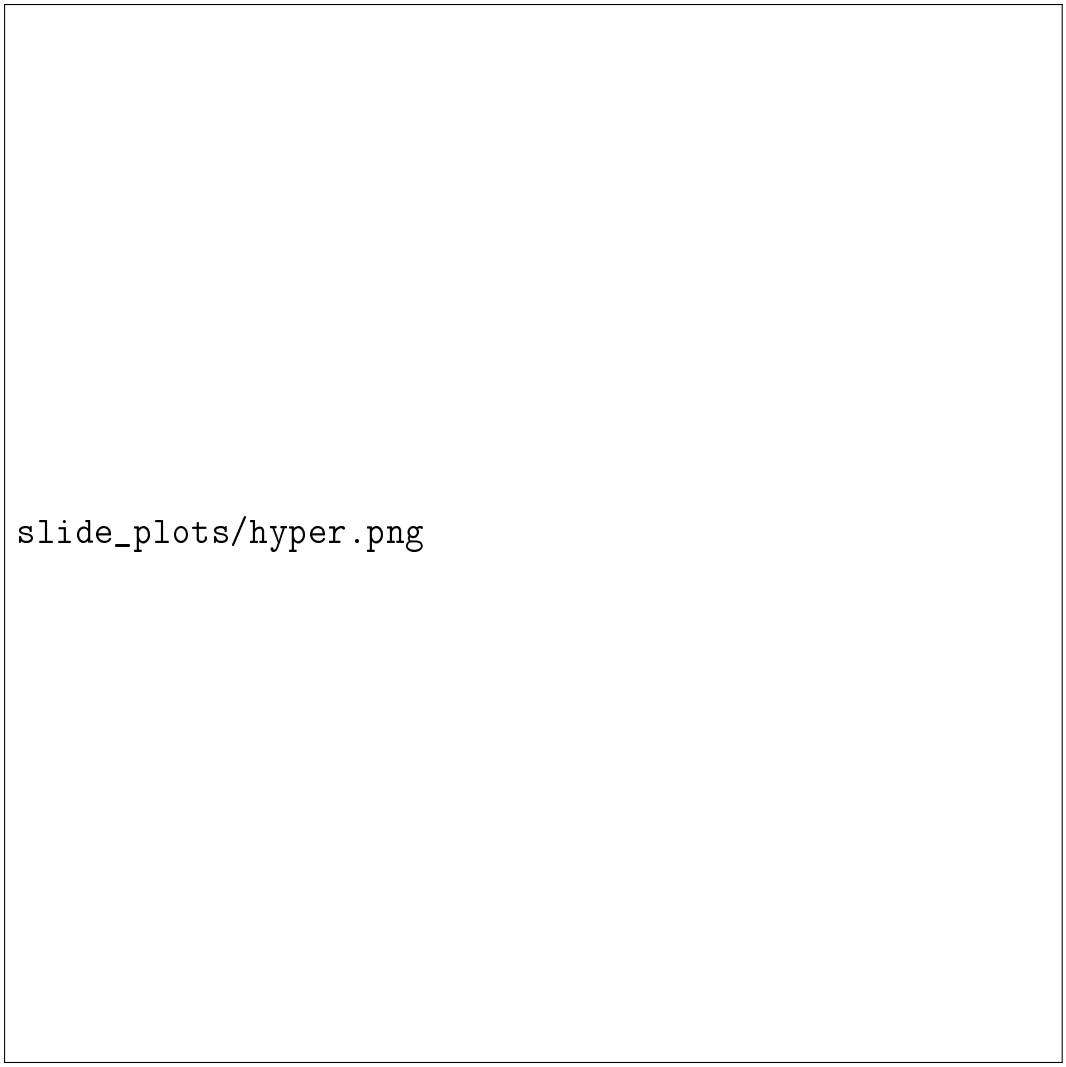


Рис. 4.1. Зависимость правдодобия модели от значения гиперпараметра α. TODO: переделать

множество всех оптимизируемых параметров (4.1). Пусть задана дифференцируемая функция потерь $L(\boldsymbol{\theta}, \mathbf{h})$, по которой производится оптимизация функции f (4.2). Пусть также задана дифференцируемая функция $Q(\boldsymbol{\theta}, \mathbf{h})$, определяющая итоговое качество модели f и приближающая интеграл (4.3).

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

$$\hat{\mathbf{h}} = \arg\max_{\mathbf{h}\in\mathbb{R}^h} Q(\hat{\boldsymbol{\theta}}(\mathbf{h}), \mathbf{h}), \tag{4.4}$$

$$\hat{\boldsymbol{\theta}}(\mathbf{h}) = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^s} L(\boldsymbol{\theta}, \mathbf{h}). \tag{4.5}$$

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:

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

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

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

$$L(\boldsymbol{\theta}, \mathbf{h}) = -\frac{1}{k}\sum_{q=1}^{k}\Big(\frac{k}{k-1}\log p(\mathbf{y} \setminus \mathbf{y}_q|\mathbf{X} \setminus \mathbf{X}_q, \mathbf{w}_q) + \log p(\mathbf{w}_q|\mathbf{A})\Big). \tag{4.6}$$

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:

$$\log\, p(\mathbf{y}|\mathbf{X}, \mathbf{A}) \geq -\mathrm{D}_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{A})\big) + \int_{\mathbf{w}} q(\mathbf{w})\log\, p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{A})d\mathbf{w} \approx \tag{4.7}$$

$$\approx \sum_{i=1}^{m} \log p(y_i|\mathbf{x}_i, \mathbf{w}_i) - D_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{A})\big) = -L(\boldsymbol{\theta}, \mathbf{h}) = Q(\boldsymbol{\theta}),$$

где q — нормальное распределение с диагональной матрицей ковариаций:

$$q \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}), \tag{4.8}$$

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

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

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

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

4.1. Градиентные методы оптимизации гиперпараметров

Рассмотрим случай, когда оптимизация (4.5) параметров $\boldsymbol{\theta}$ производится с использованием градиентных методов.

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

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

Рассмотрим оператор градиентного спуска, производящий η шагов оптимизации:

$$\hat{\boldsymbol{\theta}} = T \circ T \circ \cdots \circ T(\boldsymbol{\theta}_0, \mathbf{h}) = T^{\eta}(\boldsymbol{\theta}_0, \mathbf{h}), \tag{4.9}$$

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:

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

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

Перепишем задачу оптимизации (4.4), (4.5) в следующем виде

$$\hat{\mathbf{h}} = \arg\max_{\mathbf{h}\in\mathbb{R}^h} Q(T^{\eta}(\boldsymbol{\theta}_0, \mathbf{h})), \tag{4.10}$$

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

Оптимизационную задачу (4.10) 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 парметры $\boldsymbol{\theta}$ при условии гиперпараметров $\mathbf{h}$.
3. Приближенно решить задачу оптимизации (4.10) 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 (4.9) выступает градиентный спуск с учетом обновления параметров $\boldsymbol{\theta}$ на данном шаге:

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

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

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

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

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:

$$\mathbf{h}' = \mathbf{h} - \gamma_{\mathbf{h}} \hat{\nabla}_{\mathbf{h}} Q. \tag{4.11}$$

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 $\boldsymbol{\theta}_0$ предлагаетя 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:

$$\boldsymbol{\theta}^{\tau} = \boldsymbol{\theta}_0 + \frac{\tau}{\eta} T(\boldsymbol{\theta}). \tag{4.12}$$

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 $\boldsymbol{\theta}^\tau$(4.12).
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 (4.11). В работе [**?**] отмечается неустойчивость алгоритма при высоких значениях длины шага градиентного спуска γ. Поэтому вместо исходного правила (4.12) в данной работе первые 5% значений параметров не рассматриваются, а также учитывается только каждый τ_k шаг оптимизации:

$$\boldsymbol{\theta}^\tau = \boldsymbol{\theta}_{\tau_0} + \frac{\tau}{\eta} T(\boldsymbol{\theta}), \quad \tau \in \{\tau_0, \dots, \eta\}, \tau \bmod \tau_k = 0, \tag{4.13}$$

где $\tau_0 = [0.05 \cdot \eta]$.

Глава 5
Анализ прикладных задач порождения и выбора моделей глубокого обучения

Заключение

Список иллюстраций

Список таблиц

Список использованных источников

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

D. D. Lee et al. — Curran Associates, Inc., 2015. — Pp. 1135–1143. http://papers.nips.cc/paper/5784-learning-both-weights-and-connections-for-efficient-neural-network.pdf.

11. Greedy Layer-Wise Training of Deep Networks / Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle // Advances in Neural Information Processing Systems 19 / Ed. by B. Schölkopf, J. C. Platt, T. Hoffman. — MIT Press, 2007. — Pp. 153–160. http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf.
12. *Hinton Geoffrey E., Osindero Simon, Teh Yee-Whye.* A Fast Learning Algorithm for Deep Belief Nets // *Neural Comput.* — 2006. — Июль. — Vol. 18, no. 7. — Pp. 1527–1554. http://dx.doi.org/10.1162/neco.2006.18.7.1527.
13. Semi-supervised Learning with Deep Generative Models / Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, Max Welling // Advances in Neural Information Processing Systems 27 / Ed. by Z. Ghahramani, M. Welling, C. Cortes et al. — Curran Associates, Inc., 2014. — Pp. 3581–3589. http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf.
14. *Li Yi, Shapiro L. O., Bilmes J. A.* A generative/discriminative learning algorithm for image classification // Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. — Vol. 2. — 2005. — Oct. — Pp. 1605–1612 Vol. 2.
15. *J. Lasserre.* Hybrid of generative and discriminative methods for machine learning: Ph.D. thesis / University of Cambridge. — 2008.
16. AdaNet: Adaptive Structural Learning of Artificial Neural Networks / Corinna Cortes, Xavi Gonzalvo, Vitaly Kuznetsov et al. // *CoRR.* — 2016. — Vol. abs/1607.01097. http://arxiv.org/abs/1607.01097.
17. *Cho Kyunghyun.* Foundations and Advances in Deep Learning: G5 Artikkeliväitöskirja. — Aalto University; Aalto-yliopisto, 2014. — P. 277. http://urn.fi/URN:ISBN:978-952-60-5575-6.
18. *Alain Guillaume, Bengio Yoshua.* What regularized auto-encoders learn from the data-generating distribution // *Journal of Machine Learning Research.* — 2014. — Vol. 15, no. 1. — Pp. 3563–3593. http://dl.acm.org/citation.cfm?id=2750359.
19. *Kamyshanska Hanna, Memisevic Roland.* On autoencoder scoring // Proceedings of the 30th International Conference on Machine Learning (ICML-13) / Ed. by Sanjoy Dasgupta, David Mcallester. — Vol. 28. — JMLR Workshop and Conference Proceedings, 2013. — Май. — Pp. 720–728. http://jmlr.org/proceedings/papers/v28/kamyshanska13.pdf.
20. *D. Kingma M. Welling.* Auto-Encoding Variational Bayes // Proceedings of the International Conference on Learning Representations (ICLR). — 2014.
21. How to Train Deep Variational Autoencoders and Probabilistic Ladder Networks. / Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe

et al. // *CoRR*. — 2016. — Vol. abs/1602.02282. http://dblp.uni-trier.de/db/journals/corr/corr1602.html#SonderbyRMSW16.

22. Semi-Supervised Learning with Ladder Network. / Antti Rasmus, Harri Valpola, Mikko Honkala et al. // *CoRR*. — 2015. — Vol. abs/1507.02672. http://dblp.uni-trier.de/db/journals/corr/corr1507.html#RasmusVHBR15.
23. *MacKay David J. C.* Information Theory, Inference & Learning Algorithms. — New York, NY, USA: Cambridge University Press, 2002.
24. *Токмакова А. А., Стрижов В. В.* Оценивание гиперпараметров линейных и регрессионных моделей при отборе шумовых и коррелирующих признаков // *Информатика и её применения*. — 2012. — Т. 6(4). — С. 66–75. http://strijov.com/papers/Tokmakova2011HyperParJournal_Preprint.pdf.
25. *Зайцев А. А., Стрижов В. В., Токмакова А. А.* Оценка гиперпараметров регрессионных моделей методом максимального правдоподобия // *Информационные технологии*. — 2013. — Vol. 2. — Pp. 11–15. http://strijov.com/papers/ZaytsevStrijovTokmakova2012Likelihood_Preprint.pdf.
26. *Strijov V., Weber Gerhard-Wilhelm.* NONLINEAR REGRESSION MODEL GENERATION USING HYPERPARAMETERS OPTIMIZATION: Preprint 2009-21. — Middle East Technical University, 06800 Ankara, Turkey: Institute of Applied Mathematics, 2009. — Октябрь. — Preprint No. 149.
27. *Стрижов В. В.* Порождение и выбор моделей в задачах регрессии и классификации: Ph.D. thesis / Вычислительный центр РАН. — 2014. http://strijov.com/papers/Strijov2015ModelSelectionRu.pdf.
28. *Перекрестенко Д.О.* Анализ структурной и статистической сложности суперпозиции нейронных сетей. — 2014. http://sourceforge.net/p/mlalgorithms/code/HEAD/tree/Group074/Perekrestenko2014Complexi
29. *Vladislavleva E.* Other publications TiSEM: : Tilburg University, School of Economics and Management, 2008. http://EconPapers.repec.org/RePEc:tiu:tiutis:65a72d10-6b09-443f-8cb9-88f3bb3bc31b.
30. Predicting Parameters in Deep Learning / Misha Denil, Babak Shakibi, Laurent Dinh et al. // Advances in Neural Information Processing Systems 26 / Ed. by C.j.c. Burges, L. Bottou, M. Welling et al. — 2013. — Pp. 2148–2156. http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/1053.pdf.
31. *Xu Huan, Mannor Shie.* Robustness and generalization // *Machine Learning*. — 2012. — Vol. 86, no. 3. — Pp. 391–423. http://dx.doi.org/10.1007/s10994-011-5268-1.
32. Intriguing properties of neural networks. / Christian Szegedy, Wojciech Zaremba, Ilya Sutskever et al. // *CoRR*. — 2013. — Vol. abs/1312.6199. http://dblp.uni-trier.de/db/journals/corr/corr1312.html#SzegedyZSBEGF13.
33. Stochastic Variational Inference / Matthew D. Hoffman, David M. Blei, Chong Wang, John Paisley // *J. Mach. Learn. Res.* — 2013. — Май. — Vol. 14, no. 1. — Pp. 1303–1347. http://dl.acm.org/citation.cfm?id=2502581.2502622.

34. *Graves Alex.* Practical Variational Inference for Neural Networks // Advances in Neural Information Processing Systems 24 / Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett et al. — Curran Associates, Inc., 2011. — Pp. 2348–2356. http://papers.nips.cc/paper/4329-practical-variational-inference-for-neural-networks.pdf.
35. *Salimans Tim, Kingma Diederik P., Welling Max.* Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. // ICML / Ed. by Francis R. Bach, David M. Blei. — Vol. 37 of *JMLR Proceedings*. — JMLR.org, 2015. — Pp. 1218–1226. http://dblp.uni-trier.de/db/conf/icml/icml2015.html#SalimansKW15.
36. *Maclaurin Dougal, Duvenaud David K., Adams Ryan P.* Early Stopping is Nonparametric Variational Inference // *CoRR*. — 2015. — Vol. abs/1504.01344. http://arxiv.org/abs/1504.01344.
37. *Mandt Stephan, Hoffman Matthew D, Blei David M.* Continuous-Time Limit of Stochastic Gradient Descent Revisited.
38. *Welling Max, Teh Yee Whye.* Bayesian Learning via Stochastic Gradient Langevin Dynamics // Proceedings of the 28th International Conference on Machine Learning (ICML-11) / Ed. by Lise Getoor, Tobias Scheffer. — ICML '11. — New York, NY, USA: ACM, 2011. — June. — Pp. 681–688.
39. *Arlot Sylvain, Celisse Alain.* A survey of cross-validation procedures for model selection // *Statist. Surv.* — 2010. — Vol. 4. — Pp. 40–79. http://dx.doi.org/10.1214/09-SS054.
40. Fast and Accurate Support Vector Machines on Large Scale Systems / Abhinav Vishnu, Jeyanthi Narasimhan, Lawrence Holder et al. // 2015 IEEE International Conference on Cluster Computing, CLUSTER 2015, Chicago, IL, USA, September 8-11, 2015. — 2015. — Pp. 110–119. http://dx.doi.org/10.1109/CLUSTER.2015.26.
41. Cross-validation pitfalls when selecting and assessing regression and classification models / Damjan Krstajic, Ljubomir J. Buturovic, David E. Leahy, Simon Thomas // *Journal of Cheminformatics*. — 2014. — Vol. 6, no. 1. — Pp. 1–15. http://dx.doi.org/10.1186/1758-2946-6-10.
42. *Hornung Roman, Bernau Christoph, Truntzer Caroline et al.* Full versus incomplete cross-validation: measuring the impact of imperfect separation between training and test sets in prediction error estimation. — 2014. http://nbn-resolving.de/urn/resolver.pl?urn=nbn:de:bvb:19-epub-20682-6.
43. *Bengio Yoshua, Grandvalet Yves.* No Unbiased Estimator of the Variance of K-Fold Cross-Validation // *J. Mach. Learn. Res.* — 2004. — Декабрь. — Vol. 5. — Pp. 1089–1105. http://dl.acm.org/citation.cfm?id=1005332.1044695.
44. *Maclaurin Dougal, Duvenaud David, Adams Ryan.* Gradient-based Hyperparameter Optimization through Reversible Learning // Proceedings of the 32nd International Conference on Machine Learning (ICML-15) / Ed. by David Blei, Francis Bach. — JMLR Workshop and Conference Proceedings, 2015. — Pp. 2113–2122. http://jmlr.org/proceedings/papers/v37/maclaurin15.pdf.

45. *Domke Justin.* Generic Methods for Optimization-Based Modeling. // AISTATS / Ed. by Neil D. Lawrence, Mark A. Girolami. — Vol. 22 of *JMLR Proceedings.* — JMLR.org, 2012. — Pp. 318–326. http://dblp.uni-trier.de/db/journals/jmlr/jmlrp22.html#Domke12.
46. *Pedregosa Fabian.* Hyperparameter optimization with approximate gradient // Proceedings of the 33nd International Conference on Machine Learning (ICML). — 2016. http://jmlr.org/proceedings/papers/v48/pedregosa16.html.
47. Scalable Gradient-Based Tuning of Continuous Regularization Hyperparameters / Jelena Luketina, Tapani Raiko, Mathias Berglund, Klaus Greff // Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 / Ed. by Maria-Florina Balcan, Kilian Q. Weinberger. — Vol. 48 of *JMLR Workshop and Conference Proceedings.* — JMLR.org, 2016. — Pp. 2952–2960.
48. *Karaletsos Theofanis, Rätsch Gunnar.* Automatic Relevance Determination For Deep Generative Models // *arXiv preprint arXiv:1505.07765.* — 2015.