

Проблема выбора структуры модели является фундаментальной в области машинного обучения интеллектуального анализа данных. Проблема выбора структуры модели глубокого обучения формулируется следующим образом: решается задача классификации или регрессии на заданной или пополняемой выборке \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 по параметрам нелинейный функций. Под структурой модели понимается значения структурных параметров модели, т.е. величин, задающих вид итоговой суперпозиции.

Формализуем описанную выше задачу.

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

Задана простая выборка

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

состоящая из множества объектов

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

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

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

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

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

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

$$\mathbf{f}_k(\mathbf{x}) = \mathbf{agg}_k (\{\langle \boldsymbol{\gamma}^{j,k}, \mathbf{g}^{j,k} \rangle (\mathbf{f}_j(\mathbf{x})) | j \in \text{Adj}(v_k)\}), \quad \mathbf{f}_0(\mathbf{x}) = \mathbf{x} \quad (2)$$

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

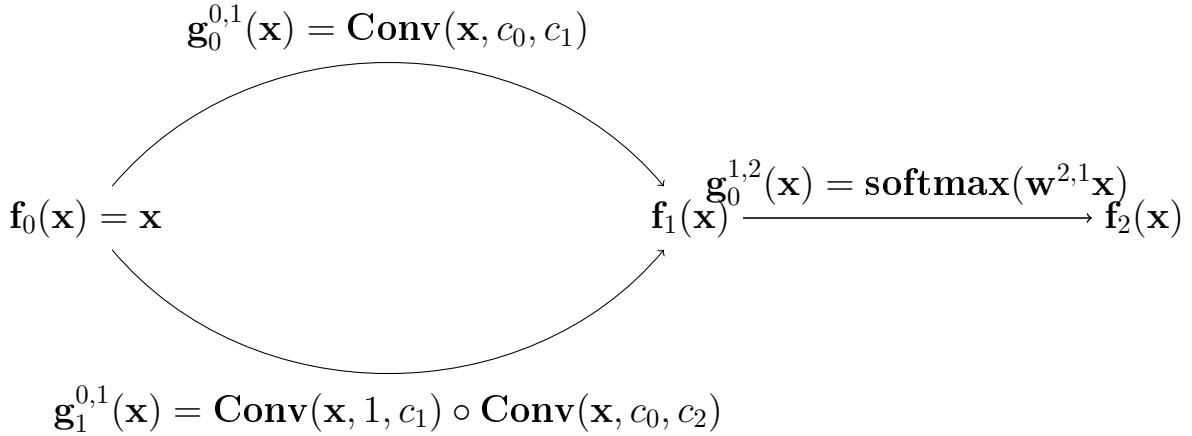


Рис. 1. Пример семейства моделей глубокого обучения: семейство описывает сверточную нейронную сеть.

Примером функций агрегации выступают функции суммы и конкатенации векторов.

Определение 4. Функции $\mathbf{f}_1, \dots, \mathbf{f}_{|V|}$ из (2) назовем называть *слоями или подмоделями* модели \mathbf{f} .

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

$$\mathbf{f} = \text{agg}_2 \left(\left\{ \gamma_0^{1,2} \mathbf{g}_0^{1,2} \left(\text{agg}_1 \left(\left\{ \gamma_0^{0,1} \mathbf{g}_0^{0,1}(\mathbf{x}), \gamma_1^{0,1} \mathbf{g}_1^{0,1}(\mathbf{x}) \right\} \right) \right) \right\} \right).$$

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

$$\mathbf{f} = \gamma_0^{1,2} \text{softmax} \left(\gamma_0^{0,1} \text{Conv}(\mathbf{x}, c_0, c_1)(\mathbf{x}) + \gamma_1^{0,1} \text{Conv}(\mathbf{x}, 1, c_1) \circ \text{Conv}(\mathbf{x}, c_0, c_2)(\mathbf{x}) \right).$$

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

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

Определение 7. *Параметризацией* множества моделей M назовем семейство моделей \mathfrak{F} , такое что для каждой модели $\mathbf{f} \in M$ существуют значение структуры модели Γ при котором функция \mathbf{f} совпадает с функцией (2).

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

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

1. Структурные параметры лежат на вершинах булевого куба: $\gamma_{j,k} \in \{0, 1\}^{K^{j,k}}$. Структурные параметры $\gamma_{j,k}$ интерпретируются как параметр включения или выключения компонент вектора базовых функций $\mathbf{g}^{j,k}$ в итоговую модель.
2. Структурные параметры лежат внутри булевого куба: $\gamma \in [0, 1]^{K^{j,k}}$. Релаксированная версия предыдущих ограничений, позволяющая проводить градиентную оптимизацию для структурных параметров.
3. Структурные параметры лежат на вершинах симплекса: $\gamma_{j,k} \in \bar{\Delta}^{K^{j,k}-1}$. Каждый вектор структурных параметров $\gamma_{j,k}$ имеет только одну ненулевую компоненту, определяющую какая из базовых функций $\mathbf{g}^{j,k}$ войдет в итоговую модель. Примером семейства моделей, требующим такое ограничение является семейство полно связанных нейронных сетей с одним скрытым слоем и двумя значениями количества нейронов на скрытом слое. Схема семейства представлена на Рис. 5. Данное семейство можно представить как семейство с двумя базовыми функциями вида $\mathbf{g} = \sigma(\mathbf{w}\mathbf{x})$, где матрицы параметров каждой из функций $\mathbf{g}^{1,1}, \mathbf{g}^{1,2}$ имеют фиксированное число нулевых столбцов. Количество этих столбцов определяет размерность итогового скрытого пространства (невырожденного?) или числа нейронов на скрытом слое.
4. Структурные параметры лежат внутри симплекса: $\gamma_{j,k} \in \Delta^{K^{j,k}-1}$. Релаксированная версия предыдущих ограничений, позволяющая проводить градиентную оптимизацию для структурных параметров. Значения структурных параметров $\gamma_{j,k}$ интерпретируются как вклад каждой компоненты вектора базовых функций $\mathbf{g}^{j,k}$ в итоговую модель.

Пример, иллюстрирующий представленные выше ограничения, изображен на Рис. 2. В данной работе рассматривается случай, когда на структурные параметры наложено ограничение 4. Данные ограничения позволяют решать задачу выбора модели как для семейства моделей типа многослойных полно связанных нейронных сетей, так и для более сложных семейств [1].

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

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

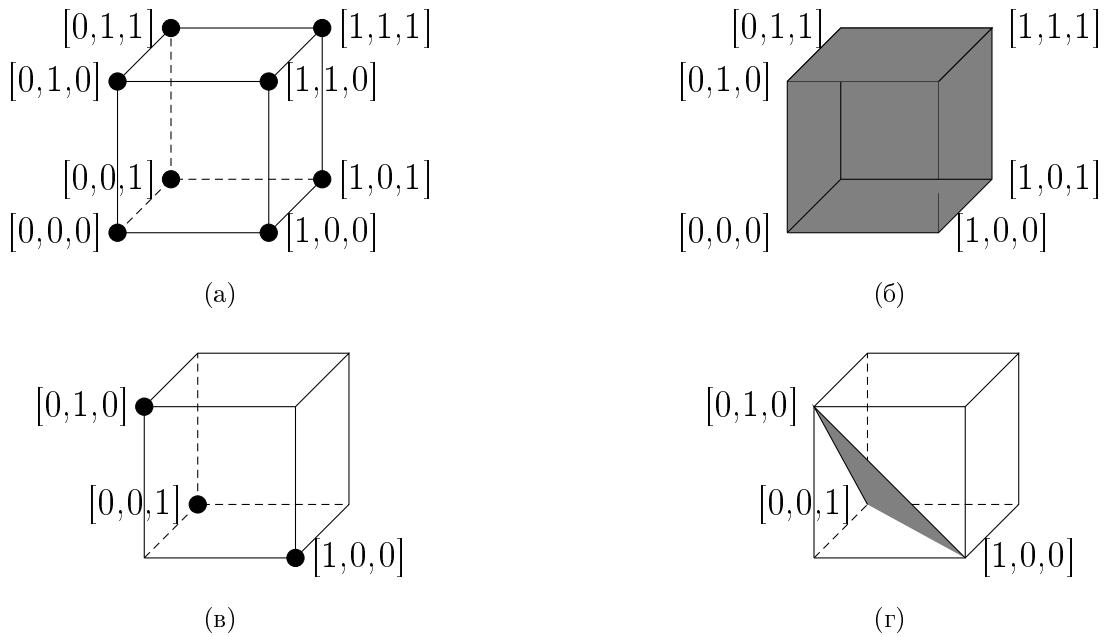


Рис. 2. Примеры ограничений для одного структурного параметра γ , $|\gamma| = 3$.
 а) структурный параметр лежит на вершинах куба, б) структурный параметр лежит внутри куба, в) структурный параметр лежит на вершинах симплекса, г) структурный параметр лежит внутри симплекса.

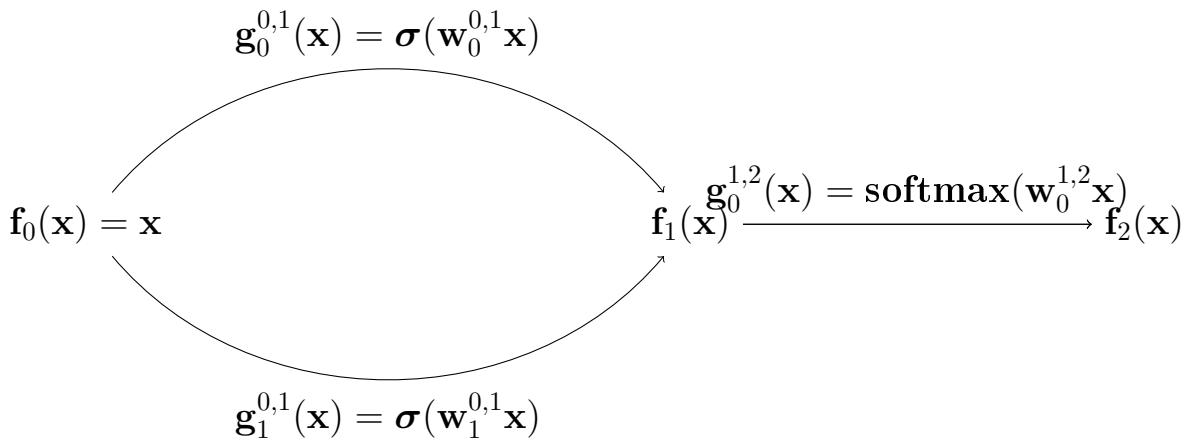


Рис. 3. Пример семейства моделей глубокого обучения: семейство описывает многослойную полно связную нейронную сеть с одним скрытым слоем и нелинейной функцией активации σ .

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

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

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

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

Определение 10. *Апостериорным распределением* назовем распределение вида

$$p(\mathbf{w}, \boldsymbol{\Gamma} | \mathbf{y}, \mathbf{X}, \mathbf{h}) = \frac{p(\mathbf{y} | \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{X}, \mathbf{h}) p(\mathbf{w}, \boldsymbol{\Gamma} | \mathbf{h})}{p(\mathbf{y} | \mathbf{X})} \propto p(\mathbf{y} | \mathbf{w}, \boldsymbol{\Gamma}, \mathbf{X}, \mathbf{h}) p(\mathbf{w}, \boldsymbol{\Gamma} | \mathbf{h}). \quad (3)$$

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

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

Определение 12. *Функцией правдоподобия выборки* назовем величину

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

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

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

Определение 13. *Правдоподобием модели* назовем величину

$$p(y | \mathbf{X}, \mathbf{h}) = \int_{\mathbf{w}, \boldsymbol{\Gamma}} p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma}) p(\mathbf{w}, \boldsymbol{\Gamma} | \mathbf{h}) d\mathbf{w} d\boldsymbol{\Gamma}. \quad (4)$$

Получение значений апостериорного распределения и правдоподобия модели сетей глубокого обучения является вычислительно сложной процедурой. Для получения оценок на данные величины используют методы, такие как аппроксимация Лапласа [2] и вариационная нижняя оценка [3]. В данной работе в качестве метода получения оценок правдоподобия модели выступает вариационное распределение.

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

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

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

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

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

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

$$\mathbf{h}^* = \arg \min_{\mathbf{h} \in \mathbb{H}} Q(\mathbf{h}, \boldsymbol{\theta}^*, \mathbf{X}, \mathbf{y}), \quad (5)$$

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

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^u} L(\boldsymbol{\theta}, \mathbf{h}, \mathbf{X}, \mathbf{y}). \quad (6)$$

Определение 18. Выбором модели \mathbf{f} назовем решение двухуровневой задачи оптимизации (5).

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

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

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

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

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

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

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

производится по валидационной выборке. Подобная оптимизация позволяет предотвратить переобучение модели [4].

Частным случаем задачи выбора структуры глубокой сети является выбор обобщенно-линейных моделей. Отдельные слои полносвязанных нейросетей являются обобщенно-линейными модели. Задачу выбора обобщено-линейной моделей сводится к задаче выбора признаков, методы решения которой делятся на три группы [5]:

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

0.1. Критерии выбора модели глубокого обучения

В данном разделе рассматриваются различные критерии выбора моделей глубокого обучения, соответствующие функции валидации Q . В данной работе в качестве критерия выбора модели предлагается субоптимальная сложность модели. Под сложностью модели понимается *правдоподобие модели* (4), являющееся байесовской интерпретацией *минимальной длины описания* [6], т.е. минимальное количество информации, которое требуется передать о модели и о выборке:

$$\text{MDL}(\mathbf{y}, \mathbf{f}) = \text{Len}(\mathbf{y}|\mathbf{w}^*, \mathbf{f}) + \text{COMP}(\mathbf{f}), \quad (7)$$

где $\text{Len}(\mathbf{y}|\mathbf{w}^*, \mathbf{f})$ — *длина описания* матрицы \mathbf{y} с использованием модели \mathbf{f} и оценки вектора параметров \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{X} [6].

В общем случае правдоподобие модели является трудновычислимым. Для получения оценки правдоподобия используются вариационные методы получения оценки правдоподобия [7], основанные на аппроксимации неизвестного другим заданным распределением. Под субоптимальной сложностью понимается вариационная оценка правдоподобия модели. Альтернативной величиной, характеризующей сложность модели, выступает радемахеровская сложность (13). Данная величина используется как критерий для продолжения итеративного построения модели в [8].

В работе [9] рассматривается ряд критериев сложности моделей глубокого обучения и их взаимосвязь. В работе [10] в качестве критерия сложности модели выступает показатель нелинейности, характеризуемый степенью полинома

Чебышева, аппроксимирующую функцию. В работе [11] анализируется показатель избыточности параметров сети. Утверждается, что по небольшому набору параметров в глубокой сети с большим количеством избыточных параметров возможно спрогнозировать значения остальных. В работе [12] 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 [13], в которой исследуется устойчивость классификации модели под действием шума. В ряде работ [14, 7, 2, 15, 16, 17] в качестве критерия выбора модели выступает правдоподобие модели. В работах [2, 15, 16, 17] рассматривается проблема выбора модели и оценки гиперпараметров в задачах регрессии. Альтернативным критерием выбора модели является минимальная длина описания [6], являющаяся показателем статистической сложности модели и заданной выборки. В работе [6] рассматриваются различные модификации и интерпретации минимальной длины описания, в том числе связь с правдоподобием модели.

Одним из методов получения приближенного значения правдоподобия модели является вариационный метод получения нижней оценки правдоподобия [7]. В работе [18] рассматривается стохастическая версия вариационного метода. В [3] рассматривается алгоритм получения вариационной нижней оценки правдоподобия для оптимизации гиперпараметров моделей глубокого обучения. В работе [19] рассматривается взаимосвязь градиентных методов получения вариационной нижней оценки интеграла с методом Монте-Карло. В [20] рассматривается стохастический градиентный спуск в качестве оператора, порождающего распределение, аппроксимирующее апостериорное распределение параметров модели. В работе отмечается, что стохастический градиентный спуск не оптимизирует вариационную оценку правдоподобия, а приближает ее только до некоторого числа итераций оптимизации. Схожий подход рассматривается в работе [21], где также рассматривается стохастический градиентный спуск в качестве оператора, порождающего апостериорное распределение параметров. В работе [22] предлагается модификация стохастического градиентного спуска, аппроксимирующая апостериорное распределение.

Альтернативным методом выбора модели является выбор модели на основе скользящего контроля [23, 2]. Проблемой такого подхода является высокая вычислительная сложность [24, 25]. В работах [26, 27] 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.

0.2. Оптимизация параметров в задаче выбора структуры модели

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

Алгоритмы прореживания параметров модели. В [28] предлагается удалять неинформативные параметры модели. Для этого находится точка оптимума $\boldsymbol{\theta}^*$ функции L , и производится разложение функции L в ряд Тейлора в окрестности $\boldsymbol{\theta}^*$:

$$L(\boldsymbol{\theta}^* + \Delta\boldsymbol{\theta}) - L(\boldsymbol{\theta}^*) = \frac{1}{2}\Delta\boldsymbol{\theta}^\top \mathbf{H}\Delta\boldsymbol{\theta} + o(\|\Delta\boldsymbol{\theta}\|^3), \quad (8)$$

где \mathbf{H} — гессиан функции L . Связь между параметрами не учитывается, поэтому гессиан матрицы L является диагональным. Положим в качестве операции удаления параметра замену его значения на ноль. Выбор наиболее неинформативного параметра сводится к задаче условной минимизации (8) при условиях вида

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

В результате решения данной задачи минимизации каждому параметру определяется функция выпуклости

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

Данная функция характеризует информативность параметра.

В [29] было предложено развитие данного метода. В отличие от [28] не вводится предположений о диагональности гессиана функции ошибок, поэтому удаление неинформативных параметров модели производится точнее. Для получения оценок гессиана и его обратной матрицы применяется итеративный алгоритм.

Алгоритмы компрессии параметров модели. В [30, 31, 32] предлагаются методы компрессии параметров сетей глубокого обучения. Основным отличием задачи прореживания от задачи компрессии выступает эксплуатационное требование: если прореживание используется для получения оптимальной и наиболее устойчивой модели, то компрессия производится для уменьшения потребляемых вычислительных ресурсов при сохранении основных эксплуатационных характеристик исходной модели [31]. В [32] предлагается итеративное использование регуляризации типа DropOut [33] для прореживания модели. В [30, 31] используются методы снижения вычислительной точности представления параметров модели на основе кластеризации параметров \mathbf{w} модели: вместо значений параметров предлагается хранить идентификатор кластера, соответствующего параметру, что существенно снижает количество требуемой памяти. В [31] предлагается метод компрессии, основанный на кластеризации значений

параметров модели и представлении их в сжатом виде на основе кодов Хаффмана.

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

В работе [4] рассматривается задача оптимизации гиперпараметров. Авторы предлагают оптимизировать константы l_2 -регуляризации отдельно для каждого параметра модели, проводится параллель с методами автоматического определения релевантности параметров (англ. automatic relevance determination, ARD) [14]. Идея автоматического определения релевантности заключается в выборе оптимальных начений гиперпараметров \mathbf{h} с дальнейшим удалением неинформативных параметров. Неинформативными параметрами являются те параметры, которые с высокой вероятностью равны нулю относительно априорного или апостериорного распределения.

В работе [3] был предложен метод, основанный на получении вариационной нижней оценки правдоподобия модели. В качестве критерия информативности параметра выступает отношение вероятности нахождения параметра в пределах апостериорного распределения к вероятности равенства параметра нулю:

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

где μ_j, σ_j — среднее и дисперсия аппроксимирующего распределения q для параметра w_j .

Идея данного метода была развита в [34], где также используются вариационные методы. В отличие от [3], в [34] рассматривается ряд априорных распределений параметров, позволяющих прореживать модели более эффективно:

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

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

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

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

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

0.3. Оптимизация гиперпараметров модели

В данном разделе рассматриваются работы, посвященные методам оптимизации гиперпараметров. Методы, используемые для оптимизации гиперпараметров моделей глубокого обучения должны быть эффективными по вычислительным затратам в силу высокой вычислительной сложности оптимизации параметров модели. В [35, 36] рассматривается задача оптимизации гиперпараметров стохастическими методами. В [35] проводится сравнение случайного поиска значений гиперпараметров с переборным алгоритмом. В [36] производится сравнение случайного поиска и алгоритмов, основанных на вероятностных моделях.

Градиентные методы оптимизации гиперпараметров.

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

$$\boldsymbol{\theta}' = T(\boldsymbol{\theta}|L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\beta}), \quad (9)$$

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

Пример схожего описания оптимизации модели с использованием оператора оптимизации можно найти в [20].

Частным случаем оператора оптимизации является оператор стохастического спуска:

$$T(\boldsymbol{\theta}|L, \mathbf{X}, \mathbf{y}, \mathbf{h}, \boldsymbol{\beta}) = \boldsymbol{\theta} - \beta_{\text{lr}} \nabla L(\boldsymbol{\theta}, \mathbf{h}, \hat{\mathbf{X}}, \hat{\mathbf{y}}), \quad (10)$$

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

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

$$\mathbf{h} = T(\mathbf{h}|Q, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}^*, \boldsymbol{\beta}), \quad (11)$$

где $\boldsymbol{\theta}^*$ — вариационные параметры, полученные в ходе решения задачи оптимизации (6).

В случае, если для решения задачи (6) применяется несколько шагов оператора оптимизации (9), $\boldsymbol{\theta}^*$ рассматривается как рекурсивная функция от начального приближения вариационных параметров $\boldsymbol{\theta}^0$ и вектора гиперпараметров \mathbf{h} :

$$\boldsymbol{\theta}^* = T \circ \dots \circ T(\boldsymbol{\theta}^0|L, \mathbf{X}, \mathbf{y}, \mathbf{h}, \boldsymbol{\beta}) = \boldsymbol{\theta}^*(\boldsymbol{\theta}^0, \mathbf{h}). \quad (12)$$

Решение задачи оптимизации (11) при (12) является вычислительно сложным, поэтому применяются методы, аппроксимирующие применение градиентных методов при (12).

В [37] рассматривается оптимизация гиперпараметров градиентными методами для квадратичной функции потерь. В [4] в качестве оператора оптимизации гиперпараметров выступает метод градиентного спуска с моментом.

Показано, что использование момента значительно снижает количество вычислительных ресурсов, требуемых для проведения оптимизации. В [38] предлагается аппроксимация градиентного метода, использующая предположение о линейности функции (12) от начального приближения $\boldsymbol{\theta}^0$. В [39] предлагается использовать численные методы для приближенного вычисления оператора оптимизации гиперпараметров. В [40] в качестве аппроксимации (12) предлагается рассматривать только последний шаг оптимизации:

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

где η — число шагов оптимизации.

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

В работе [41] предлагается моделировать качество модели Q (4) гауссовым процессом, параметрами которого выступают гиперпараметры исходной модели.

Одна из основных проблем использования гауссового процесса как суррогатной модели — кубическая сложность оптимизации. В работе [42] предлагается использовать случайные подпространства гиперпараметров для ускоренной оптимизации. В работе [43] предлагается комбинация из множества гауссовых моделей и линейной модели, позволяющая модели нелинейные зависимости гиперпараметров, а также существенно сократить сложность оптимизации.

В работе [44] предлагается рассматривать RBF-модель для аппроксимации качества Q исходной модели, что позволяет ускорить процесс оптимизации суррогатной модели. В [45] рассматривается глубокая нейронная сеть в качестве суррогатной функции. Вместо интеграла правдоподобия (4), который оценивается в случае использования гауссового процесса в качестве суррогата, используется максимум апостериорной вероятности (3).

Одним из параметров гауссовых процессов является функция ядра гауссового процесса, полностью определяющая процесс в случае нулевого среднего. В работе [46] предлагается функция ядра, определенная на графах:

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

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

В работе [47] рассматривается задача выбора структуры нейросети. Предлагается метод построения ковариационной функции для сравнения разнородных графов, соответствующих разным моделям нейронных сетей. Ковариационная функция основывается на метрике, заданной на некоторых характеристиках

$g(v)$ вершин, возможно не определенных для сравниваемых графов:

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

где λ_1, λ_2 — параметры функции d_v .

0.4. Порождение и выбор структуры модели глубокого обучения

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

Графовое представление структуры модели. Одним из возможных представлений структуры моделей глубокого обучения является графовое представление, в котором в качестве ребер графа выступают нелинейные функции, а в качестве вершин графа — представление выборки под действием соответствующих нелинейных функций. Данный подход к описанию модели является соответствует походам, описанным в [48], а также в библиотеках типа TensorFlow [49], Theano [50], Pytorch [51], в которых модель рассматривается как граф, ребрами которого выступают математические операции, а вершинами — результат их действия на выборку. В то же время, существуют и другие способы представления модели. В ряде работ, посвященных байесовской оптимизации [45, 44, 41], модель рассматривается как черный ящик, над которым производится ограниченный набор операций типа “произвести оптимизацию параметров” и “предсказать значение зависимой переменной по независимой переменной и параметрам модели”. Подход, описанный в данных работах, также коррелирует с библиотеками машинного обучения, такими как Weka [52], RapidMiner [53] или sklearn [54], в которых модель машинного обучения рассматривается как черный ящик.

В [55] представлен обзор по графовому описанию моделей глубокого обучения, предлагается метод формального описания графовых сетей (англ. Graph Network), являющийся обобщением предложенных ранее графовых описаний моделей.

В работе [56] рассматриваются подходы к порождению моделей глубокого обучения. Предлагается формализация пространства поиска и формальное описание элементов пространства моделей. Приведем пример описания семейства моделей, соответствующего схеме из Рис. 1 при условии, что структурные параметры γ имеют только одну ненулевую компоненту:

(Concat

```

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

```

TODO: в статье нет сотфмакса

Прогнозирование графовых структур. В работе [57] предлагается метод прогнозирования графовой структуры на основе линейного программирования. Предлагается свести проблему поиска графовой структуры к комбинаторной проблеме. В работе [58] предлагается метод прогнозирования структур деревьев, основанный на дважды-рекуррентных нейросетях (англ. doubly-reccurent), т.е. на сетях, отдельно прогнозирующих глубину и ширину уровней деревьев.

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

В работе [59] рассматривается порождение моделей, оптимизируемых без учителя. Модель представляется многослойным перцептроном вида:

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

где \mathbf{H}^i — бинарные матрицы, определяющие вклад каждого параметра из \mathbf{w}^i в итоговую модель, знаком \odot обозначается покомпонентное перемножение.

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

В работе [60] предлагается метод выбора модели сверточной нейронной сети. Используется функция потерь, основанная на аппроксимации априорного распределения процесса индийского буфета для каждой базовой функции \mathbf{g}_i , являющейся i -м отображением объектов:

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

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

В работе [61] предлагается ввести априорное распределение Бернулли на структурные параметры γ^i .

В [62] рассматривается задача выбора архитектуры с помощью большого количества параллельных запусков обучения моделей. Предлагаются критерии ранней остановки процедуры оптимизации обучения моделей.

Последовательный выбор структуры модели. В работе [63] приводятся теоретические оценки построения нейросетей с использованием жадных стратегий, при которых построение модели производится итеративно последовательным увеличением числа нейронов в сети. В работе [64] 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 [65].

В работах [66, 67] рассматривается послойное построение модели с отдельным критерием оптимизации для каждого слоя. В работах [68, 69, 70] предлагается декомпозиция модели на порождающую и разделяющую, оптимизируемых последовательно.

В работах [71, 8] предлагается наращивание моделей, основанное на бустинге. Рассматривается задача построения нейросетевых моделей специального типа:

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

приводится параметризация модели, позволяющая рассматривать декомпозицию модели на слабые классификаторы. В [8] рассматривается задача выбора полносвязной нейронной сети для задачи бинарной классификации, $Z = 2$. На каждом шаге построения выбирается одно из двух расширений модели, каждое из которых рассматривается как слабый классификатор: сделать модель шире или сделать модель глубже. Пример работы AdaNet представлен на Рис. 4. Построение модели заканчивается при условии снижении радемахеровской сложности:

$$\mathfrak{R} = \frac{1}{m} \mathsf{E}_{b_1, \dots, b_{|V|}} \sup_{\mathbf{w}} \sum_{i=1}^m b_i \arg \max_{c=\{0,1\}} f^c(\mathbf{x}_i, \mathbf{w}), \quad (13)$$

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

В работе [72] рассматривается задача порождения сверточных нейронных сетей. Предлагается проводить последовательный выбор структуры модели по восходящему числу параметров: начиная от сетей с одной подмоделью и итеративно увеличивая количество подмоделей. В силу высокой вычислительной сложности данного подхода, вместо последовательного порождения моделей, предлагается провести оптимизацию рекуррентной нейронной сети, которая

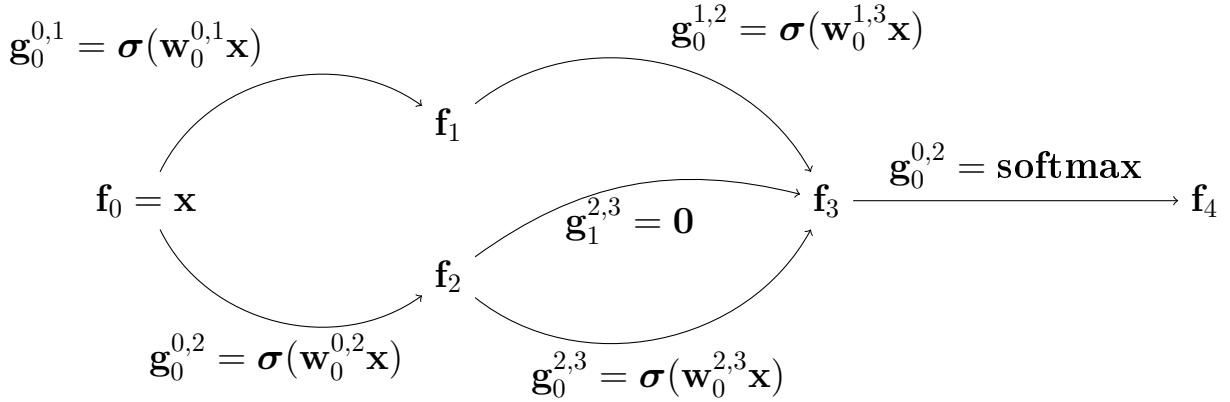


Рис. 4. Пример итерации алгоритма AdaNet [8]. Рассматриваются две альтернативные модели: модель с углублением сети (соответствует занулению функции f_2 с использованием базовой функции $g_1^{2,3} = \mathbf{0}$) и модель с расширением сети (соответствует базовой функции $g_0^{2,3}$).

В качестве функции агрегации для подмодели f_3 выступает конкатенация: $\text{agg}_3 = \text{concat}$.

предсказывает качество модели по заданным подмоделям, и на основе данного предсказания выбрать наилучшую модель.

В работе [73] предлагается метод анализа структуры сети на основе линейных классификаторов, построенных на промежуточных слоях нейросети. Схожий метод был предложен в [74], где классификаторы на промежуточных уровнях используются для уменьшения вычислений при выполнении вывода и предсказаний. Промежуточные классификаторы работают как решающий список.

В работе [75] предлагается инкрементальный метод оптимизации нейросети. На первом этапе модель декомпозируется на несколько подмоделей, при которой модель последовательностью слоев $f_1, \dots, f_{|V|}$. Проводится последовательная оптимизация моделей вида:

- 1) $f = f_{|V|}(x)$;
- 2) $f = f_{|V|-1} \circ f_{|V|}(x)$;
- 3) ...
- 4) $f = f_1 \circ \dots \circ f_{|V|}(x)$.

Оптимизация структуры модели на основе обучения с подкреплением. В [76] предлагается итеративная схема выбора архитектуры сверточной нейросети с использованием обучения с подкреплением. Распределение структур и параметров $q(\mathbf{w}, \Gamma)$ задается рекуррентной нейронной сетью, которая определяет значение параметров модели и наличие ребер с ненулевыми операциями между вершинами графов модели. Параметры рекуррентной нейронной сети оптимизируются на основе значения функции Q , получаемого на каждой итерации алгоритма.

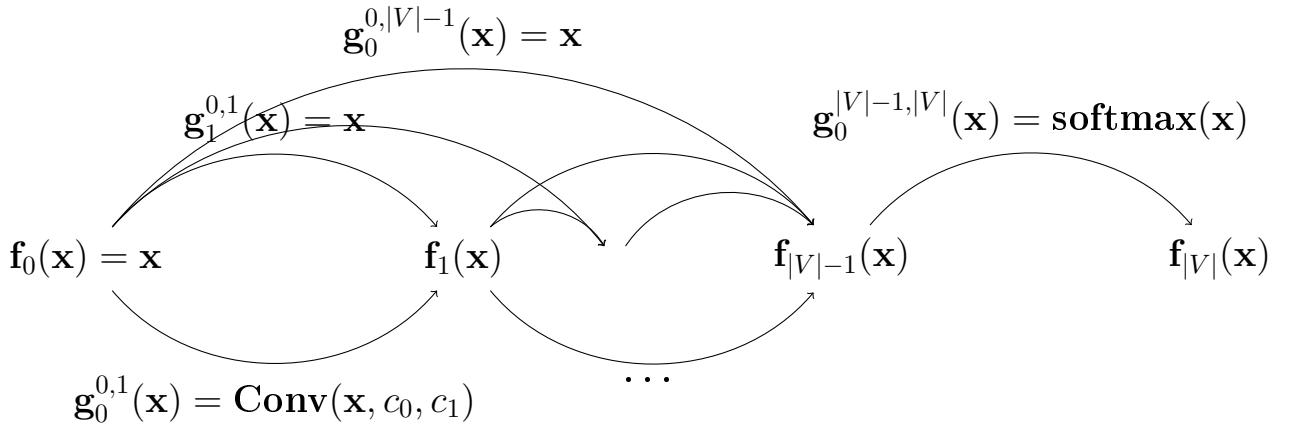


Рис. 5. Пример семейства моделей глубокого обучения, описываемый в [76]. Каждая подмодель \mathbf{f}_j является линейной комбинацией базовых функций: свертки и результата работы предыдущих подмоделей (англ. skip-connection).

В работе [77] предлагается алгоритм построения регрессионной модели для оценки финального качества модели и ранней остановки оптимизации моделей. Он позволяет существенно ускорить поиск моделей, представленный в [76]. В [78] рассматривается задача переноса архитектуры нейросети, чья структуры была выбрана по выборке, меньшей мощности. Как и в [76] предлагается метод параметризации сверточной нейронной сети в виде графа. Предложенная параметризация позволяет задать более мощное семейство моделей, чем в [76]. Модель представляется в виде последовательности суперпозиций подмоделей, называемых клетками (англ. normal cell и reduction cell). Каждая из этих клеток содержит следующее множество нелинейных операций \mathbf{g} , состоящее из тождественной операции $\mathbf{g}(\mathbf{x}) = \mathbf{x}$, а также множество сверток с фиксированным количеством каналов и размером фильтров и функций субдискретизации или пулинга. Алгоритм выбора структуры модели рекуррентной сетью выглядит следующим образом на шаге j :

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

В отличие от предыдущих работ, в работе [79] предлагается подход к инкрементальному обучению нейросети, основанном на модификации модели, полученной на предыдущем шаге. Рассматривается две операции над нейросетью: расширение и углубление сети.

В работах [80, 81, 82] рассматриваются методы деформации нейросетей. В работе [82] предлагается метод оптимального разделения нейросети на несколько независимых сетей для уменьшения количества связей и, как следствие, уменьшения сложности оптимизации модели. В работе [80] предлагается метод сохранения результатов оптимизации нейросети при построении новой более глубокой или широкой нейросети. В работе [81] рассматривается задача расширения сверточной нейросети, нейросеть рассматривается как граф.

В работе [1] используется представление модели из [78]. Вместо обучения с подкреплением используются градиентная оптимизация структуры и параметров, выполненная в единой процедуре.

0.5. Метаоптимизация моделей глубокого обучения

Задача выбора структуры модели тесно связана с раздел машинного обучения под названием *метаобучение* или *метаоптимизация*. Под метаобучением понимаются алгоритмы машинного обучения [83], которые:

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

В работе [84] предлагается подход к адаптивному изменению параметров сети. В качестве оператора оптимизации параметров рассматривается величина:

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

где \mathbf{f}_{mod} — функция, определяющая номер параметра из $\boldsymbol{\theta}$, подлежащего оптимизации, а $\mathbf{f}_{\text{optim}}$ — величина изменения параметра. В [84] также предлагается подмодель \mathbf{f}_{ana} , определяющая номер параметра, подлежащего дальнейшему анализу. Подход, описанный в данной работе, предполагает оптимизацию и анализ не только самой модели \mathbf{f} , но и дополнительных моделей $\mathbf{f}_{\text{mod}}, \mathbf{f}_{\text{ana}}, \mathbf{f}_{\text{optim}}$.

В работе [85] рассматривается оптимизация метапараметров (шага градиентного спуска β_{lr} и начального распределения параметров $\boldsymbol{\theta}^0$). Рассматривается задача оптимизации параметров модели в случае, когда количество примеров невелико. Для этого проводится оптимизация оператора оптимизации, который выглядит следующим образом:

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

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

В работе [86] рассматривается задача восстановления параметров модели по параметрам другой модели, чьи параметры были получены оптимизацией функции потерь на выборке меньшей мощности. Задачу можно рассматривать как задачу нахождения параметров некоторого оператора оптимизации $T : \boldsymbol{\theta}^0 \rightarrow \boldsymbol{\theta}$, где $\boldsymbol{\theta}^0$ — параметры модели, оптимизированной на небольшой выборке. Предлагается функция оптимизации:

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

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

В работе [87] рассматривается оптимизация метапараметров оператора оптимизации с помощью модели долгой краткосрочной памяти LSTM, которая выступает альтернативе аналитических алгоритмов, таких как Adam [88] или AdaGrad [89]. LSTM имеет небольшое число параметров, т.к. для каждого метапараметра используется свой экземпляр модели LSTM с одинаковыми параметрами для каждого экземпляра. Оптимизируемый функционал является суммой значений функции потерь L на нескольких шагах оптимизации:

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

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

0.6. Выбор структур моделей специального вида

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

В работе [90] рассматривается оптимизация моделей нейросетей с бинарной функцией активации. Задача оптимизации сводится к задаче mixed integer программирования, которая решается методами выпуклого анализа. В работе [91] 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 [92, 93] рассматривается выбор архитектуры сети с использованием *суперсетей*: связанных между собой подмоделей, образующих граф, каждый путь из нулевой вершины в последнюю которого определяет модель глубокого обучения. Пример графа, описывающего суперсеть представлен на Рис. 6. В работе [93] рассматриваются стохастические суперсети, позволяющие выбрать структуру нейросети за ограниченное время оптимизации. Схожий подход был

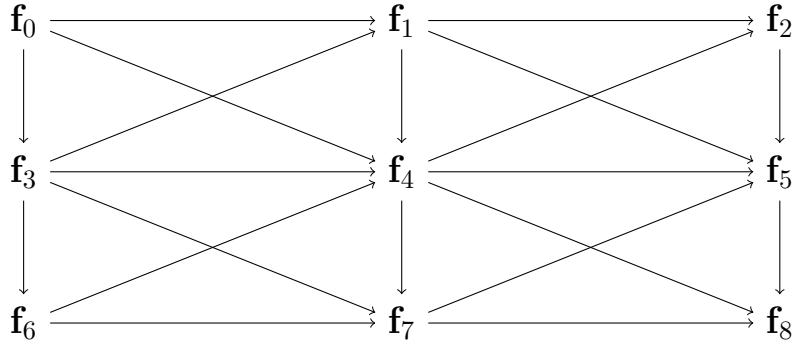


Рис. 6. Пример суперсети. Каждый путь из подмодели f_0 в конечную модель f_8 задает модель глубокого обучения.

предложен в работе [92], где предлагается использовать эволюционные алгоритмы для запоминания оптимальных подмоделей и переноса этих моделей в другие задачи.

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

В качестве порождающих моделей в сетях глубокого обучения выступают ограниченные машины Больцмана [94] и автокодировщики [95]. В работе [96] рассматриваются алгоритмы регуляризации автокодировщиков, позволяющих формально рассматривать данные модели как порождающие модели с использованием байесового вывода. В работе [97] рассматриваются регуляризованные автокодировщики и свойства оценок их правдоподобия. В работе [98] предлагается обобщение автокодировщика с использованием вариационного байесовского вывода [7]. В работе [99] рассматриваются модификации вариационного автокодировщика и ступенчатых сетей [100] для случая построения многослойных порождающих моделей.

В ряде работ [101, 102, 103, 104, 105] рассматривается подход к построению порождающих моделей глубокого обучения, при котором каждая подмодель f_i приближает распределение некоторой случайной величины \mathbf{z}_i , которая влияет на итоговое распределение $p(\mathbf{X}, \mathbf{y}) = \int_{\mathbf{z}_1, \dots, \mathbf{z}_{|V|}} p(\mathbf{X}, \mathbf{y} | \mathbf{z}_1, \dots, \mathbf{z}_{|V|}) p(\mathbf{z}_1, \dots, \mathbf{z}_{|V|}) d\mathbf{z}_1 \dots d\mathbf{z}_{|V|}$. Подобный подход позволяет использовать вероятностную интерпретацию для каждой отдельной подмодели.

В работе [101] рассматривается обобщение вариационного автокодировщика на случай более общих графических моделей. Предлагается проводить оптимизацию сложных графических моделей в единой процедуре. Для вывода предлагаются использовать нейронные сети. Другая модификация вариационного

автокодировщика представлена в работе [102], авторы рассматривают использование процесса сломанной трости в вариационном автокодировщике, тем самым получая модель со стохастической размерностью скрытой переменной. В [103] рассматривается смесь автокодировщиков, где смесь моделируется процессом Дирихле.

В работе [104] предлагается подход к оптимизации неизвестного распределения с помощью вариационного вывода. Предлагается решать задачу оптимизации итеративно, добавляя в модель новые компоненты вариационного распределения, проводится аналогия с бустингом.

В работе [105] рассматривается задача построения порождающих моделей с дискретными значениями скрытых переменных \mathbf{z}_i , предлагается критерий для послойного обучения порождающих моделей:

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

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

В работе [106] рассматривается метод ARD для снижения размерности скрытого пространства вариационных порождающих моделей. Скрытая переменная параметризуется как произведение некоторой случайной величины \mathbf{z} на вектор, отвечающий за релевантность каждой компоненты скрытой переменной. Схема порождения выборки \mathbf{X} представлена на Рис. 7.

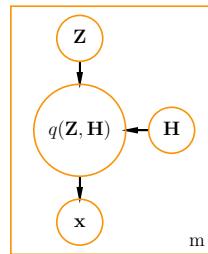


Рис. 7. Схема порождения вектора объектов \mathbf{X} , представленная в [106].

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

Глава 1

Выбор модели с использованием вариационного вывода

В данной главе рассматривается задача выбора моделей глубокого обучения субоптимальной сложности. Под сложностью модели понимается правдоподобие модели (4). Под субоптимальной сложностью понимается приближенная оценка правдоподобия модели, полученная с использованием вариационных методов. Вводятся вероятностные предположения о распределении параметров. На основе байесовского вывода предлагается функция правдоподобия модели. Для получения оценки правдоподобия применяются вариационные методы с использованием градиентных алгоритмов оптимизации. Проводится вычислительный эксперимент на нескольких выборках.

В данной работе предлагается метод получения вариационной нижней оценки правдоподобия модели с использованием модифицированного алгоритма стохастического градиентного спуска. Модификация заключается в добавлении шумовой компоненты. Эта компонента позволяет получить более точные оценки правдоподобия модели для сравнения моделей и выбора наиболее адекватной из них. Рассматривается ряд модификаций базового алгоритма. В качестве базового алгоритма выступает алгоритм оптимизации параметров модели с использованием стохастического градиентного спуска без контроля переобучения. Он заключается в итеративном вычислении градиента по параметрам от функции правдоподобия обучающей выборки и изменении значений параметров с его учетом. Приводится сравнение с алгоритмом получения вариационной нижней оценки, представленном в [3]. Рассматриваются следующие модификации базового алгоритма: оптимизация с кросс-валидацией с использованием и без использования регуляризации модели, алгоритм получения вариационной оценки правдоподобия модели с применением нормального распределения, алгоритм получения вариационной оценки правдоподобия с использованием стохастического градиентного спуска, алгоритм получения вариационной оценки правдоподобия с использованием стохастической динамики Ланжевена. Данные алгоритмы решают следующие проблемы оптимизации моделей градиентным спуском: оптимизация модели с меньшими затратами вычислительных ресурсов, быстрая сходимость оптимизации, контроль переобучения и выбор наиболее адекватной модели. Под переобучением понимается потеря обобщающей способности модели с увеличением правдоподобия обучающей выборки [14]. Переобучение характерно для моделей с большим количеством параметров, сопоставимым с мощностью обучающей выборки, что встречается в случае выбора моделей глубокого обучения [94, 107]. Также алгоритмы имеют дальнейшую возможность применения к градиентным алгоритмам оптимизации гиперпараметров, описанным в [4].

Свойства представленных в данной работе алгоритмов исследуются на выборках, на которых проверялась работа алгоритма вероятностного обратного распространения ошибок [108], где авторы акцентируются на оптимизации па-

раметров модели.

1.1. Постановка задачи оптимизации правдоподобия моделей

Определим понятие статистической сложности модели. Сложностью модели будем называть *правдоподобие модели* (4). Пусть задано множество моделей M , для которых, возможно, не определена общая параметризация. Для каждой модели $\mathbf{f} \in M$ заданы различные значения гиперпараметров \mathbf{h} . Рассмотрим два подхода к сравнению сложностей моделей:

1. Модели \mathbf{f} принадлежат одному семейству \mathfrak{F} . При таком подходе сравнение сложности различных моделей является адекватным, т.к. они определены на общем пространстве структур Γ и параметров \mathbb{W} . Недостатком такого подхода является сложность вычисления правдоподобия модели в случае, когда структура Γ определена однозначно, что может противоречить введенным вероятностным предположениям о структуре модели.
2. Модели \mathbf{f} рассматриваются независимо от общей параметризации. Недостатком такого подхода является возможная некорректность сравнения моделей с заведомо различными структурами моделей, сильно отличающимися по количеству параметров. Возможным решение данного недостатка является введение дополнительного штрафа за большое количество параметров в модели [14].

В данном разделе рассматривается второй вариант. Будем полагать, что структура модели Γ для вероятностной модели глубокого обучения \mathbf{f} определена однозначно. Тем не менее, основная часть данной главы также применима и ко второму варианту.

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

$$p(\mathbf{y}|\mathbf{X}, \mathbf{h}) = \int_{\mathbf{w} \in \mathbb{W}} p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{h})p(\mathbf{w}|\mathbf{h})d\mathbf{w}. \quad (1.1)$$

Определение 21. Модель классификации \mathbf{f} назовем оптимальной среди моделей M , если достигается максимум интеграла (1.1).

Требуется найти оптимальную модель \mathbf{f} из заданного множества моделей M , а также значения ее параметров \mathbf{w} , доставляющие максимум апостериорной вероятности

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{h})p(\mathbf{w}|\mathbf{h})}{p(\mathbf{y}|\mathbf{X}, \mathbf{h})}. \quad (1.2)$$

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

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{h}) = (2\pi)^{-\frac{m}{2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})\right), \quad (1.3)$$

априорное распределение параметров модели имеет вид

$$p(\mathbf{w}|\mathbf{h}) = (2\pi)^{-\frac{n}{2}} |\mathbf{A}|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{w}^\top \mathbf{A} \mathbf{w}\right). \quad (1.4)$$

Правдоподобие модели (4) в этом примере вычисляется аналитически [109]:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{h}) = (2\pi)^{-\frac{m}{2}} |\mathbf{A}|^{\frac{1}{2}} |\mathbf{H}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})^\top (\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})\right) \exp\left(-\frac{1}{2}\hat{\mathbf{w}}^\top \mathbf{A} \hat{\mathbf{w}}\right), \quad (1.5)$$

где $\hat{\mathbf{w}}$ — значение наиболее вероятных (3) 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:

$$\mathbf{f} = \mathbf{f}_{|V|}(\mathbf{f}_{|V|-1}(\dots \mathbf{f}_1(\mathbf{x}))),$$

$\mathbf{f}_1, \dots, \mathbf{f}_{|V|}$ — дифференцируемые функции, $\mathbf{f}_{|V|}$ — многомерная логистическая функция:

$$\mathbf{f}_{|V|} = \frac{\mathbf{f}_{|V|-1}(\dots \mathbf{f}_1(\mathbf{x}))}{\sum_{r=1}^Z \exp(f_{|V|-1}^r(\dots \mathbf{f}_1(\mathbf{x})))},$$

где $f_{|V|-1}^r$ — r -я компонента функции $\mathbf{f}_{|V|-1}$. Компонента r вектора $\mathbf{f}_{|V|}$ определяет вероятность принадлежности объекта \mathbf{x} к классу r . Логарифм правдоподобия зависимой переменной аналогично (1.3) имеет вид

$$\log p(y|\mathbf{x}, \mathbf{w}, \mathbf{h}) = \log f_{|V|}^y(\mathbf{f}_{|V|-1}(\dots \mathbf{f}_1(\mathbf{x}))).$$

Данная модель описывает многослойную сеть, аналогичную моделям семейства, представленного на Рис. 5.

Интеграл правдоподобия (1.1) 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 (1.1), будем рассматривать его нижнюю оценку, полученную при помощи неравенства Йенсена [7]. Получим нижнюю оценку логарифма правдоподобия модели, используя неравенство

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{h}) = \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{h})}{q(\mathbf{w})} d\mathbf{w} + D_{KL}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})) \geq \quad (1.6)$$

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

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

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

Определение 22. Вариационной оценкой логарифма правдоподобия модели (1.1) $\log p(\mathbf{y} | \mathbf{X}, \mathbf{h})$ 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 (4) 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 (1.6) является нижней, поэтому может давать некорректные оценки для правдоподобия (1.1). Для того, чтобы оценить величину этой ошибки, докажем следующее утверждение.

Теорема 1 ([7]). 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 (4) эквивалентна минимизации расстояния Кульбака–Лейблера между распределением $q(\mathbf{w}) \in \mathfrak{Q}$ и апостериорным распределением параметров $p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \mathbf{h})$:

$$\hat{q} = \arg \max_{q \in Q} \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w} | \mathbf{X}, \mathbf{h})}{q(\mathbf{w})} d\mathbf{w} \Leftrightarrow \hat{q} = \arg \min_{q \in Q} D_{\text{KL}}(q(\mathbf{w}) || p(\mathbf{w} | \mathbf{y}, \mathbf{X}, \mathbf{h})), \quad (1.7)$$

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

Доказательство. Доказательство непосредственно следует из (1.6). Вычитая из обеих частей равенства $D_{KL}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h}))$, получим

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

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 (1.1) сводится к поиску распределения \hat{q} , аппроксимирующего распределение $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})$ наилучшим образом.

Определение 23. Пусть задано множество распределений \mathfrak{Q} . Модель \mathbf{f} назовем субоптимальной на множестве моделей M , если модель доставляет максимум нижней вариационной оценке интеграла (1.7)

$$\max_{q \in \mathfrak{Q}} \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{h})}{q(\mathbf{w})} d\mathbf{w}. \quad (1.8)$$

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

Вариационная оценка (1.6) интерпретируется как оценка сложности модели по принципу минимальной длины описания (7), где первое слагаемое определяет количество информации для описания выборки, а второе слагаемое — длину описания самой модели [3].

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

1.2. Методы получения вариационной оценки правдоподобия

Ниже представлены методы получения вариационных нижних оценок (1.8) правдоподобия (4). В первом параграфе рассматривается метод, основанный на аппроксимации апостериорного распределения $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})$ (3) многомерным гауссовым распределением с диагональной матрицей ковариаций. В последующих параграфах рассматриваются методы, основанные на различных модификациях стохастического градиентного спуска.

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

$$q \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}), \quad , \boldsymbol{\theta} = [\boldsymbol{\mu}_q, \text{diag}(\mathbf{A}_q^{-1})] \quad (1.9)$$

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

Пусть априорное распределение $p(\mathbf{w}|\mathbf{h})$ (1.4) параметров модели задано как нормальное:

$$p(\mathbf{w}|\mathbf{h}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{A}^{-1}), \quad \mathbf{h} = \text{diag}(\mathbf{A}_q^{-1}),$$

Тогда оптимизация (1.7) имеет вид

$$\int_{\mathbf{w}} q(\mathbf{w}) \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{h}) d\mathbf{w} - D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{h})) \rightarrow \max_{\mathbf{A}_q, \boldsymbol{\mu}_q}, \quad (1.10)$$

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

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

В качестве приближенного значения интеграла

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

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 (1.10) имеет вид

$$\begin{aligned} \mathbf{f} &= \arg \max_{\mathbf{A}_q, \boldsymbol{\mu}_q} \sum_{i=1}^m \log p(y_i|\mathbf{x}_i, \mathbf{w}_i) - D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{h})) = \\ &= \arg \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathbf{h}, \mathbf{X}, \mathbf{y}). \end{aligned} \quad (1.11)$$

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

$$y \sim \mathcal{N}(\mathbf{w}, \mathbf{B}^{-1}), \quad (1.12)$$

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

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

График аппроксимации распределения параметров представлен на рис. 1.1, а. Как видно из графика, с использованием метода (1.11) 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 $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})$.

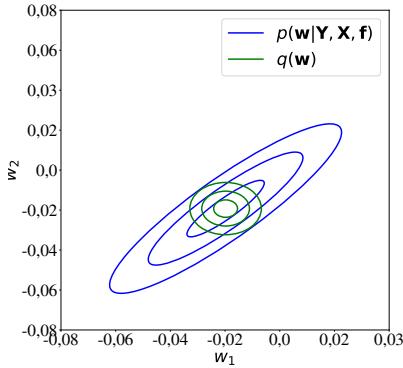
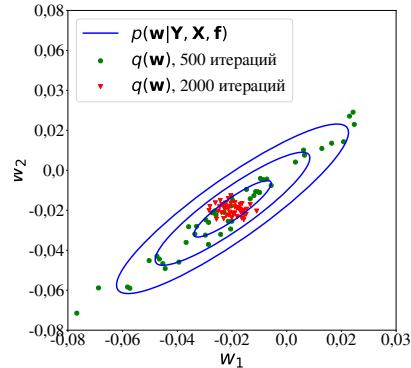
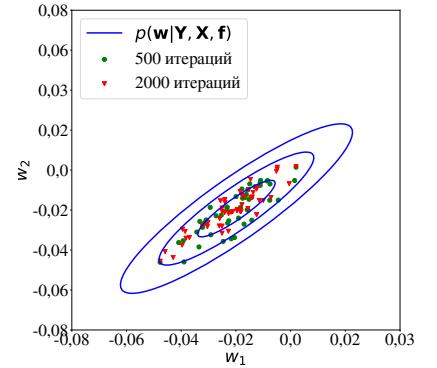
Рис. 1.0. *a*Рис. 1.0. *б*Рис. 1.0. *в*

Рис. 1.1. Аппроксимация распределения *a*) нормальным распределением, *б*) распределением, полученным с помощью градиентного спуска, *в*) с использованием стохастической динамики Ланжевена.

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 (1.6)

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{h}) \geq \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{h})}{q(\mathbf{w})} d\mathbf{w} = \mathbb{E}_{q(\mathbf{w})} (\log p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{h})) - S(q(\mathbf{w})), \quad (1.13)$$

где S — энтропия распределения:

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

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

$\mathbb{E}_{q(\mathbf{w})} (\log p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{h}))$ — 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 [110], 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 (1.17) S и оценок правдоподобия (1.13).

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

$$L(\boldsymbol{\theta}, \mathbf{h}, \mathbf{X}, \mathbf{y}) = -\sum_{l=1}^r \log p(\mathbf{y}, \mathbf{w}^l | \mathbf{X}, \mathbf{h}) \rightarrow \min, \quad \boldsymbol{\theta} = [\mathbf{w}^1, \dots, \mathbf{w}^r],$$

где r — число оптимизаций,

$$\log p(\mathbf{y}, \mathbf{w}^l | \mathbf{X}, \mathbf{h}) = -\sum_{i=1}^m \log p(y_i, \mathbf{w}^l | \mathbf{x}_i, \mathbf{h}) = -\log p(\mathbf{w}^l | \mathbf{h}) - \sum_{i=1}^m \log p(y_i | \mathbf{x}_i, \mathbf{w}^l, \mathbf{h}). \quad (1.14)$$

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 дальнейшего описания метода введем понятие оператора градиентного спуска, являющегося частным случаем оператора оптимизации (9).

Определение 24. Оператором градиентного спуска назовем оператор оптимизации вида

$$T(\boldsymbol{\theta} | L, \mathbf{X}, \mathbf{y}, \mathbf{h}, \beta_{\text{lr}}) = \boldsymbol{\theta} - \beta_{\text{lr}} \nabla L(\boldsymbol{\theta}, \mathbf{h}, \mathbf{X}, \mathbf{y}), \quad (1.15)$$

где β_{lr} — 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 [20].

Теорема 2. 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 β_{lr} — длина шага градиентного спуска, такая что

$$\beta_{\text{lr}} < \frac{1}{C_L}, \quad \beta_{\text{lr}} < \left(\max_{g \in \{1, \dots, r\}} \lambda_{\max}(\mathbf{H}(\mathbf{w}^l)) \right)^{-1}, \quad (1.16)$$

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

При выполнении неравенств (1.16) разность энтропий распределений $q'(\mathbf{w}), q(\mathbf{w})$ на смежных шагах почти наверное сходится к следующему выражению:

$$\mathsf{S}(q'(\mathbf{w})) - \mathsf{S}(q(\mathbf{w})) \approx \frac{1}{r} \sum_{l=1}^r (-\beta_{\text{lr}} \text{Tr}[\mathbf{H}(\mathbf{w}^l)] - \beta_{\text{lr}} \text{Tr}[\mathbf{H}(\mathbf{w}^l) \mathbf{H}(\mathbf{w}^l)]) + o_{\beta_{\text{lr}}^2 \rightarrow 0}(1), \quad (1.17)$$

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

Доказательство. Предварительно приведем две леммы, требуемые для доказательства теоремы.

Лемма 1 ([111]). Пусть T — оператор градиентного спуска, L — дважды дифференцируемая функция потерь, градиент ∇L которой имеет константу Липшица C_L . Пусть для длины шага β_{lr} выполнено неравенство $\beta_{lr} < \frac{1}{C_L}$. Тогда T является диффеоморфизмом.

Лемма 2 ([112]). 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

$$S(q'(\mathbf{w})) - S(q(\mathbf{w})) = \int_{\mathbf{w}} q'(\mathbf{w}) \log \left| \frac{\partial T(\mathbf{w})}{\partial \mathbf{w}} \right| d\mathbf{w}. \quad (1.18)$$

Рассмотрим очередной шаг оптимизации. При $\beta_{lr} < \frac{1}{C}$ оператор градиентного спуска T является диффеоморфизмом, а значит, и биекцией, справедлива формула (1.18). По усиленному закону больших чисел

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

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

$$\log \left| \frac{\partial T(\mathbf{w}'^l)}{\partial \mathbf{w}} \right| = \log |\mathbf{I} - \beta_{lr} \mathbf{H}| = \sum_{i=1}^u \log (1 - \beta_{lr} \lambda_i), \quad (1.19)$$

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

При $(\beta_{lr} \lambda_i)^2 \leq (\beta_{lr} \lambda_{\max})^2 < 1$ выражение (1.19) раскладывается в ряд Тейлора:

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

Просуммировав полученные выражения для каждой точки мультистарта и вынеся $o_{\beta_{lr}^2 \rightarrow 0}(1)$ за скобки, получим выражение (1.17), что и требовалось доказать. \square

Получим итоговую формулу для оценки правдоподобия модели.

Теорема 3. Оценка (1.13) на шаге оптимизации τ представима в виде

$$\log \hat{p}(\mathbf{y} | \mathbf{X}, \mathbf{h}) \approx \frac{1}{r} \sum_{g=1}^r L(\mathbf{w}_\tau^l, \mathbf{X}, \mathbf{y}) + \quad (1.20)$$

$$+S(q^0(\mathbf{w})) + \frac{1}{r} \sum_{b=1}^r \sum_{l=1}^r (-\beta_{lr} \text{Tr}[\mathbf{H}(\mathbf{w}_b^l)] - \beta_{lr}^2 \text{Tr}[\mathbf{H}(\mathbf{w}_b^l) \mathbf{H}(\mathbf{w}_b^l)])$$

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

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

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

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

$$S(q^b(\mathbf{w})) - S(q^{b-1}(\mathbf{w})) \approx \frac{1}{r} \sum_{l=1}^r (-\beta_{lr} \text{Tr}[\mathbf{H}(\mathbf{w}_b^l)] - \beta_{lr}^2 \text{Tr}[\mathbf{H}(\mathbf{w}_b^l) \mathbf{H}(\mathbf{w}_b^l)]). \quad (1.21)$$

Формула (1.20) получается подстановкой в выражение (1.13) суммы выражений вида (1.21), а также начальной энтропии $S(q^0(\mathbf{w}))$. \square

В [20] предлагается алгоритм приближенного вычисления для выражения, находящегося под знаком суммы в (1.20):

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

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

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

Заметим, что при приближении параметров модели к точке экстремума оценка правдоподобия устремляется в минус бесконечность в силу постоянно убывающей энтропии. Таким образом, чем ближе градиентный метод приближает параметры модели к точке экстремума, тем менее точной становится оценка правдоподобия модели. Один из методов борьбы с данной проблемой представлен в следующих параграфах.

Модификация алгоритма оптимизации модели.

В качестве оператора T предлагается использовать псевдослучайный стохастический градиентный спуск, т.е. градиентный спуск (10), оптимизирующий параметры $\mathbf{w}^1, \dots, \mathbf{w}^r$ по некоторой случайной подвыборке $\hat{\mathbf{X}}, \hat{\mathbf{y}}$, одинаковой для каждой точки старта $\mathbf{w}^1, \dots, \mathbf{w}^r$:

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

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

Рис. 1.2. Псевдокод алгоритма получения вариационной нижней оценки правдоподобия модели с использованием градиентного спуска

Вход: $\mathbf{X}, \mathbf{y}, p(\mathbf{w}|\mathbf{h})$;

Вход: критерий останова Stop, начальное распределение параметров q^0 , количество точек мультистарта r , функция потерь L , ее первая и вторая производные;

Выход: $\log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{h})$;

- 1: **для** $l = 1, \dots, r$
- 2: $\mathbf{w}^l \sim q^0$;
- 3: $\mathbf{S} = \mathbf{S}(q^0)$;
- 4: **пока** не достигнут критерий останова Stop
- 5: $\boldsymbol{\theta} = T(\boldsymbol{\theta})$;
- 6: **для** $l = 1, \dots, r$
- 7: $\mathbf{r}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
- 8: $\mathbf{r}_1 = \mathbf{r}_0 - \beta_{\text{lr}} \mathbf{r}_0^\top \nabla \nabla L(\mathbf{w}^l, \mathbf{y}, \mathbf{X})$;
- 9: $\mathbf{r}_2 = \mathbf{r}_1 - \beta_{\text{lr}} \mathbf{r}_1^\top \nabla \nabla L(\mathbf{w}^l, \mathbf{y}, \mathbf{X})$;
- 10: $\mathbf{S}^l = \mathbf{r}_0^\top (-2\mathbf{r}_0 + 3\mathbf{r}_1 - \mathbf{r}_2)$;
- 11: $\mathbf{S} = \frac{1}{r} \sum_{l=1}^r \mathbf{S}^l$;
- 12: $\hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{h}) = \frac{1}{r} \sum_{l=1}^r p(\mathbf{y}|\mathbf{X}, \mathbf{w}^l, \mathbf{h})$;
- 13: $\hat{p}(\mathbf{w}|\mathbf{h}) = \frac{1}{r} \sum_{l=1}^r p(\mathbf{w}^l|\mathbf{h})$;
- 14: $\log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{h}) = \log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{h}) + \log \hat{p}(\mathbf{w}|\mathbf{h})$;

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 (1.15), основной проблемой модифицированного алгоритма оценки интеграла (1.8) является грубость аппроксимации исходного распределения $p(\mathbf{w}|\mathbf{f}, \mathfrak{D})$.

Рассмотрим пример (1.12). График аппроксимации распределения $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})$ представлен на рис. 1.1,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 [20]. Таким образом, снижение оценки (1.20) можно использовать как критерий остановки оптимизации модели для снижения эффекта переобучения.

На рис. 1.1 представлена аппроксимация распределения $p(\mathbf{w}|\mathbf{Y}, \mathbf{X}, \mathbf{h})$ различными методами: *a*) 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.

Аппроксимация с использованием динамики Ланжевена

Для достижения нижней оценки интеграла (1.8), более близкой к реальному значению логарифма интеграла (4), чем оценка с использованием градиентного спуска, предлагается использовать стохастическую динамику Ланжевена [22]. Стохастическая динамика Ланжевена представляет собой вариант стохастического градиентного спуска с добавлением гауссового шума:

$$T(\mathbf{w}) = \mathbf{w} - \beta_{\text{lr}} \nabla L - \frac{m}{\hat{m}} \log p(\hat{\mathbf{y}}|\hat{\mathbf{X}}, \mathbf{w}, \mathbf{h}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \frac{\beta_{\text{lr}}}{2} \mathbf{I}), \quad (1.22)$$

где $\hat{\mathbf{X}}$ — псевдослучайная подвыборка, $\hat{\mathbf{y}}$ — соответствующие метки, \hat{m} — размер подвыборки. Длина шага оптимизации β_{lr} удовлетворяет условиям, гарантирующим сходимость алгоритма в стандартных ситуациях [22]:

$$\sum_{\tau=1}^{\infty} \beta_{\text{lr}}^{\tau} = \infty, \quad \sum_{\tau=1}^{\infty} (\beta_{\text{lr}}^{\tau})^2 < \infty.$$

Для оценки энтропии с учетом шума $\boldsymbol{\varepsilon}$ предлагается использовать следую-

щее неравенство [112, 113]:

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

где τ — текущий шаг оптимизации, $S(\mathcal{N}(0, \frac{\beta_{\text{lr}}}{2}))$ — 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 $p(\mathbf{w}|\mathfrak{D}, \mathbf{h})$ [22, 114]. График аппроксимации апостериорного распределения с использованием динамики Ланжевена представлен на рис. 1.1, в. При одинаковом количестве итераций динамика Ланжевена продолжает аппроксимировать апостериорное распределение, в то время как градиентный спуск сходится к модели распределения. Как видно из графика, алгоритм, основанный на стохастической динамике Ланжевена, способен давать более точную вариационную оценку правдоподобия (1.8). В то же время алгоритм более требователен к настройке параметров оптимизации [115]: “быстро изменяющаяся кривизна [траекторий параметров модели] делает методы стохастической градиентной динамики Ланжевена по умолчанию неэффективными”.

1.3. Анализ методов выбора моделей

Для анализа свойств предложенного критерия субоптимальности в задачах регрессии и классификации, а также методов получения нижних оценок правдоподобия модели в задачах выбора моделей был проведен ряд вычислительных экспериментов на выборках Boston Housing, Protein Structure, а также на небольшой подвыборке YearPredictionMSD (далее — Boston, Protein и MSD) [116] и подвыборке изображений рукописных цифр MNIST [117].

Для выборок Boston, Protein и MSD была рассмотрена задача регрессии

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

В качестве множества моделей M были рассмотрены нейросети с одним скрытым слоем и softplus-функцией активации:

$$\mathbf{f}(\mathbf{w}, \mathbf{X}) = \text{softplus}(\mathbf{X}\mathbf{W}_1)\mathbf{W}_2, \tag{1.23}$$

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

Для выборки Boston также было рассмотрено множество моделей с тремя скрытыми слоями, построенных аналогично однослойной модели (1.23). Размер каждого слоя равнялся 50.

Для выборки MNIST была рассмотрена задача бинарной классификации: из выборки были взяты только объекты, соответствующие цифрам 7 и 9. Размерность выборки была понижена с 784 до 50 методом главных компонент аналогично [118]. 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

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

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 $f(\mathbf{w}, \mathbf{X})$ (RMSE) в случае задачи регрессии и доля верно предсказанных меток класса (Accuracy) в задаче классификации, а также соответствующие критерии при возмущении элементов выборки:

$$\text{RMSE}_\sigma = \text{RMSE}(f(\mathbf{w}, \mathbf{X} + \boldsymbol{\varepsilon}), \mathbf{y}), \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma \mathbf{I}). \quad (1.24)$$

Были рассмотрены шесть алгоритмов.

1. Базовый алгоритм: оптимизация параметров без валидации и ранней остановки. Оптимизация проводилась с использованием стохастического градиентного спуска (1.15). Для данного алгоритма априорное распределение $p(\mathbf{w}|\mathbf{h})$ 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 (1.11).

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 (1.14) на валидационной выборке $\mathfrak{D}_{\text{valid}}$ для алгоритмов 2, 3,
2. вариационная нижняя оценка правдоподобия (1.6) 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 β_{lr} , отбирались наилучшие модели.

Описание эксперимента представлено в табл. 1. Результаты экспериментов представлены в табл. 2. На рис. 1.3 представлен график зависимости RMSE_σ от параметра σ для однослойных моделей.

Рис. 1.2. *a*

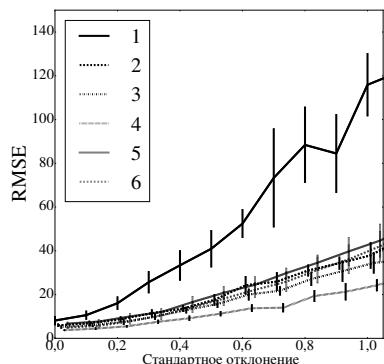


Рис. 1.2. *б*

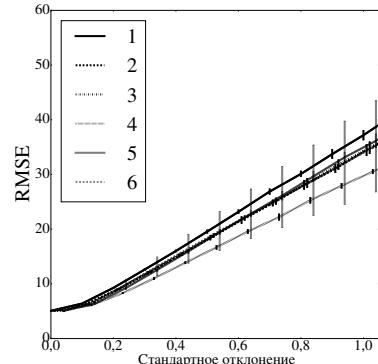


Рис. 1.2. *в*

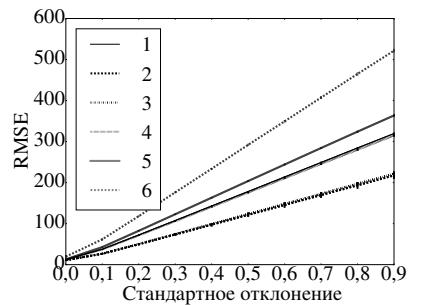


Рис. 1.3. Возмущение выборки для однослойных нейросетей: *a*) Boston Housing, *б*) Protein, *в*) MSD.

Таблица 1.1. Описание выборок для экспериментов по выбору моделей

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 1.2. Результаты эксперимента по выбору моделей

Выборка \mathfrak{D}	Алгоритмы					
	1	2	3	4	5	6
Результаты, RMSE/Accuracy						
Boston, один слой	$8,1 \pm 2,0$	$5,9 \pm 0,7$	$5,2 \pm 0,6$	$3,7 \pm 0,2$	$6,7 \pm 0,7$	$5,0 \pm 0,4$
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

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 (3) от нормального.

Алгоритм 6 показал низкое качество (1.24) при возмущении объектов выборки в большинстве экспериментов. В трех экспериментах наилучшие показатели по данному критерию показал алгоритм 4. Заметим, что алгоритм 5, являющийся модификацией алгоритма 4, показал худшие результаты как по RMSE, так и по RMSE при возмущении объектов выборки. На выборке MNIST алгоритм 4 показал результаты значительно хуже остальных алгоритмов. В целом результаты по данному алгоритму схожи с результатами, описанными в [20]: 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 сходимости параметров к апостериорному распределению параметров [22].

Программное обеспечение для проведения экспериментов и проверки результатов находится в [119].

Список основных обозначений

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

Список иллюстраций

1	Пример семейства моделей глубокого обучения: семейство описывает сверточную нейронную сеть.	2
2	Примеры ограничений для одного структурного параметра $\gamma, \gamma = 3$. а) структурный параметр лежит на вершинах куба, б) структурный параметр лежит внутри куба, в) структурный параметр лежит на вершинах симплекса, г) структурный параметр лежит внутри симплекса.	4
3	Пример семейства моделей глубокого обучения: семейство описывает многослойную полносвязную нейронную сеть с одним скрытым слоем и нелинейной функцией активации σ	4
4	Пример итерации алгоритма AdaNet [8]. Рассматриваются две альтернативные модели: модель с углублением сети (соответствует занулению функции f_2 с использованием базовой функции $g_1^{2,3}$) и модель с расширением сети (соответствует базовой функции $g_0^{2,3}$).	16
5	Пример семейства моделей глубокого обучения, описываемый в [76]. Каждая подмодель f_j является линейной комбинацией базовых функций: свертки и результата работы предыдущих подмоделей (англ. skip-connection).	17
6	Пример суперсети. Каждый путь из подмодели f_0 в конечную модель f_8 задает модель глубокого обучения.	20
7	Схема порождения вектора объектов \mathbf{X} , представленная в [106].	21
1.1	Аппроксимация распределения а) нормальным распределением, б) распределением, полученным с помощью градиентного спуска, в) с использованием стохастической динамики Ланжевена.	28
1.2	Псевдокод алгоритма получения вариационной нижней оценки правдоподобия модели с использованием градиентного спуска	32
1.3	Возмущение выборки для однослойных нейросетей: а) Boston Housing, б) Protein, в) MSD.	36

Список таблиц

1.1	Описание выборок для экспериментов по выбору моделей	36
1.2	Результаты эксперимента по выбору моделей	37

Список использованных источников

1. *Liu Hanxiao, Simonyan Karen, Yang Yiming.* Darts: Differentiable architecture search // arXiv preprint arXiv:1806.09055. — 2018.
2. *Токмакова А. А., Стрижов В. В.* Оценивание гиперпараметров линейных и регрессионных моделей при отборе шумовых и коррелирующих признаков // Информатика и её применение. — 2012. — Т. 6(4). — С. 66–75. http://strijov.com/papers/Tokmakova2011HyperParJournal_Preprint.pdf.
3. *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>.
4. *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>.
5. *Li Jundong, Liu Huan.* Challenges of feature selection for big data analytics // IEEE Intelligent Systems. — 2017. — Vol. 32, no. 2. — Pp. 9–15.
6. *Grünwald Peter.* A Tutorial Introduction to the Minimum Description Length Principle // Advances in Minimum Description Length: Theory and Applications. — MIT Press, 2005.
7. *Bishop Christopher M.* Pattern Recognition and Machine Learning (Information Science and Statistics). — Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
8. AdaNet: Adaptive Structural Learning of Artificial Neural Networks / Corinna Cortes, Xavier Gonzalvo, Vitaly Kuznetsov et al. // International Conference on Machine Learning. — 2017. — Pp. 874–883.
9. *Перекрестенко Д.О.* Анализ структурной и статистической сложности суперпозиции нейронных сетей. — 2014. <http://sourceforge.net/p/mlalgorithms/code/HEAD/tree/Group074/Perekrestenko2014Complexity.pdf>
10. *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>.
11. 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.
12. *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>.

13. 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>.
14. MacKay David J. C. Information Theory, Inference & Learning Algorithms. — New York, NY, USA: Cambridge University Press, 2002.
15. 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.
16. 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.
17. 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>.
18. 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>.
19. 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>.
20. 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>.
21. Mandt Stephan, Hoffman Matthew D, Blei David M. Continuous-Time Limit of Stochastic Gradient Descent Revisited.
22. 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.
23. 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>.
24. 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>.

25. 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>.
26. 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>.
27. 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>.
28. Cun Yann Le, Denker John S., Solla Sara A. Optimal Brain Damage // Advances in Neural Information Processing Systems. — Morgan Kaufmann, 1990. — Pp. 598–605.
29. Hassibi Babak, Stork David G, Wolff Gregory J. Optimal brain surgeon and general network pruning // Neural Networks, 1993., IEEE International Conference on / IEEE. — 1993. — Pp. 293–299.
30. Incremental network quantization: Towards lossless cnns with low-precision weights / Aojun Zhou, Anbang Yao, Yiwen Guo et al. // *arXiv preprint arXiv:1702.03044*. — 2017.
31. Han Song, Mao Huizi, Dally William J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding // *arXiv preprint arXiv:1510.00149*. — 2015.
32. 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>.
33. Dropout: A simple way to prevent neural networks from overfitting / Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky et al. // *The Journal of Machine Learning Research*. — 2014. — Vol. 15, no. 1. — Pp. 1929–1958.
34. Louizos Christos, Ullrich Karen, Welling Max. Bayesian compression for deep learning // Advances in Neural Information Processing Systems. — 2017. — Pp. 3290–3300.
35. Bergstra James, Bengio Yoshua. Random search for hyper-parameter optimization // *Journal of Machine Learning Research*. — 2012. — Vol. 13, no. Feb. — Pp. 281–305.
36. Algorithms for hyper-parameter optimization / James S Bergstra, Rémi Bardenet, Yoshua Bengio, Balázs Kégl // Advances in Neural Information Processing Systems. — 2011. — Pp. 2546–2554.

37. *Bengio Yoshua*. Gradient-based optimization of hyperparameters // *Neural computation*. — 2000. — Vol. 12, no. 8. — Pp. 1889–1900.
38. DrMAD: Distilling Reverse-Mode Automatic Differentiation for Optimizing Hyperparameters of Deep Neural Networks / Jie Fu, Hongyin Luo, Jiashi Feng et al. // *arXiv preprint arXiv:1601.00917*. — 2016.
39. *Pedregosa Fabian*. Hyperparameter optimization with approximate gradient // Proceedings of the 33rd International Conference on Machine Learning. — 2016.
40. 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.
41. *Snoek Jasper, Larochelle Hugo, Adams Ryan P.* Practical bayesian optimization of machine learning algorithms // Advances in neural information processing systems. — 2012. — Pp. 2951–2959.
42. Bayesian Optimization in High Dimensions via Random Embeddings. / Ziyu Wang, Masrour Zoghi, Frank Hutter et al. // IJCAI. — 2013. — Pp. 1778–1784.
43. Bayesian Optimization with Tree-structured Dependencies / Rodolphe Jenatton, Cedric Archambeau, Javier González, Matthias Seeger // International Conference on Machine Learning. — 2017. — Pp. 1655–1664.
44. Hyperparameter optimization of deep neural networks using non-probabilistic RBF surrogate model / Ilija Ilievski, Taimoor Akhtar, Jiashi Feng, Christine Annette Shoemaker // *arXiv preprint arXiv:1607.08316*. — 2016.
45. Scalable Bayesian Optimization Using Deep Neural Networks / Jasper Snoek, Oren Rippel, Kevin Swersky et al. // Proceedings of the 32nd International Conference on Machine Learning / Ed. by Francis Bach, David Blei. — Vol. 37 of *Proceedings of Machine Learning Research*. — Lille, France: PMLR, 2015. — 07–09 Jul. — Pp. 2171–2180. <http://proceedings.mlr.press/v37/snoek15.html>.
46. Structure Optimization for Deep Multimodal Fusion Networks using Graph-Induced Kernels / Dhanesh Ramachandram, Michal Lisicki, Timothy J Shields et al. // *arXiv preprint arXiv:1707.00750*. — 2017.
47. Raiders of the lost architecture: Kernels for Bayesian optimization in conditional parameter spaces / Kevin Swersky, David Duvenaud, Jasper Snoek et al. // *arXiv preprint arXiv:1409.4011*. — 2014.
48. *Воронцов Константин Вячеславович*. Локальные базисы в алгебраическом подходе к проблеме распознавания: Ph.D. thesis. — Graz, 1999.
49. *Abadi Martín, Agarwal Ashish, Barham Paul et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. — 2015. — Software available from tensorflow.org. <http://tensorflow.org/>.

50. *Theano Development Team*. Theano: A Python framework for fast computation of mathematical expressions // *arXiv e-prints*. — 2016. — may. — Vol. abs/1605.02688. <http://arxiv.org/abs/1605.02688>.
51. Automatic differentiation in PyTorch / Adam Paszke, Sam Gross, Soumith Chintala et al. — 2017.
52. *Eibe Frank, Hall MA, Witten IH*. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques" // *Morgan Kaufmann*. — 2016.
53. *Hofmann Markus, Klinkenberg Ralf*. RapidMiner: Data mining use cases and business analytics applications. — CRC Press, 2013.
54. Scikit-learn: Machine learning in Python / Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort et al. // *Journal of machine learning research*. — 2011. — Vol. 12, no. Oct. — Pp. 2825–2830.
55. Relational inductive biases, deep learning, and graph networks / Peter W Battaglia, Jessica B Hamrick, Victor Bapst et al. // *arXiv preprint arXiv:1806.01261*. — 2018.
56. *Negrinho Renato, Gordon Geoff*. Deeparchitect: Automatically designing and training deep architectures // *arXiv preprint arXiv:1704.08792*. — 2017.
57. Learning Bayesian network structure using LP relaxations / Tommi Jaakkola, David Sontag, Amir Globerson, Marina Meila // Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. — 2010. — Pp. 358–365.
58. *Alvarez-Melis David, Jaakkola Tommi S*. Tree-structured decoding with doubly-recurrent neural networks. — 2016.
59. *Adams Ryan, Wallach Hanna, Ghahramani Zoubin*. Learning the structure of deep sparse graphical models // Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. — 2010. — Pp. 1–8.
60. *Feng Jiashi, Darrell Trevor*. Learning the structure of deep convolutional networks // Proceedings of the IEEE international conference on computer vision. — 2015. — Pp. 2749–2757.
61. *Shirakawa Shinichi, Iwata Yasushi, Akimoto Youhei*. Dynamic Optimization of Neural Network Structures Using Probabilistic Modeling // *arXiv preprint arXiv:1801.07650*. — 2018.
62. Toward Optimal Run Racing: Application to Deep Learning Calibration / Olivier Bousquet, Sylvain Gelly, Karol Kurach et al. // *arXiv preprint arXiv:1706.03199*. — 2017.
63. 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>.

64. *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.
65. *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>.
66. 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>.
67. *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>.
68. 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>.
69. *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.
70. *J. Lasserre.* Hybrid of generative and discriminative methods for machine learning: Ph.D. thesis / University of Cambridge. — 2008.
71. Learning deep resnet blocks sequentially using boosting theory / Furong Huang, Jordan Ash, John Langford, Robert Schapire // *arXiv preprint arXiv:1706.04964*. — 2017.
72. Progressive neural architecture search / Chenxi Liu, Barret Zoph, Jonathon Shlens et al. // *arXiv preprint arXiv:1712.00559*. — 2017.
73. *Alain Guillaume, Bengio Yoshua.* Understanding intermediate layers using linear classifier probes // *arXiv preprint arXiv:1610.01644*. — 2016.
74. *Teerapittayanon Surat, McDanel Bradley, Kung HT.* Branchynet: Fast inference via early exiting from deep neural networks // Pattern Recognition (ICPR), 2016 23rd International Conference on / IEEE. — 2016. — Pp. 2464–2469.
75. Incremental Training of Deep Convolutional Neural Networks / R Istrate12, ACI Malossi, C Bekas, D Nikolopoulos.

76. *Zoph Barret, Le Quoc V.* Neural architecture search with reinforcement learning // *arXiv preprint arXiv:1611.01578*. — 2016.
77. Accelerating neural architecture search using performance prediction / Bowen Baker, Otkrist Gupta, Ramesh Raskar, Nikhil Naik // *CoRR, abs/1705.10823*. — 2017.
78. Learning transferable architectures for scalable image recognition / Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V Le // *arXiv preprint arXiv:1707.07012*. — 2017.
79. Efficient Architecture Search by Network Transformation / Han Cai, Tianyao Chen, Weinan Zhang et al. — 2018.
80. *Chen Tianqi, Goodfellow Ian, Shlens Jonathon.* Net2net: Accelerating learning via knowledge transfer // *arXiv preprint arXiv:1511.05641*. — 2015.
81. Forward thinking: Building and training neural networks one layer at a time / Chris Hettinger, Tanner Christensen, Ben Ehlert et al. // *arXiv preprint arXiv:1706.02480*. — 2017.
82. *Miranda Conrado S, Von Zuben Fernando J.* Reducing the Training Time of Neural Networks by Partitioning // *arXiv preprint arXiv:1511.02954*. — 2015.
83. *Schmidhuber Juergen, Zhao Jieyu, Wiering MA.* Simple principles of metalearning // *Technical report IDSIA*. — 1996. — Vol. 69. — Pp. 1–23.
84. *Schmidhuber Jürgen.* A neural network that embeds its own meta-levels // Neural Networks, 1993., IEEE International Conference on / IEEE. — 1993. — Pp. 407–412.
85. Meta-SGD: Learning to Learn Quickly for Few Shot Learning / Zhenguo Li, Fengwei Zhou, Fei Chen, Hang Li // *arXiv preprint arXiv:1707.09835*. — 2017.
86. *Wang Yu-Xiong, Hebert Martial.* Learning to learn: Model regression networks for easy small sample learning // European Conference on Computer Vision / Springer. — 2016. — Pp. 616–634.
87. Learning to learn by gradient descent by gradient descent / Marcin Andrychowicz, Misha Denil, Sergio Gomez et al. // Advances in Neural Information Processing Systems. — 2016. — Pp. 3981–3989.
88. *Kinga D, Adam J Ba.* A method for stochastic optimization // International Conference on Learning Representations (ICLR). — Vol. 5. — 2015.
89. *Duchi John, Hazan Elad, Singer Yoram.* Adaptive subgradient methods for online learning and stochastic optimization // *Journal of Machine Learning Research*. — 2011. — Vol. 12, no. Jul. — Pp. 2121–2159.
90. *Friesen Abram L, Domingos Pedro.* Deep Learning as a Mixed Convex-Combinatorial Optimization Problem // *arXiv preprint arXiv:1710.11573*. — 2017.
91. *Kristiansen Gus, Gonzalvo Xavi.* EnergyNet: Energy-based Adaptive Structural Learning of Artificial Neural Network Architectures // *arXiv preprint arXiv:1711.03130*. — 2017.

92. Pathnet: Evolution channels gradient descent in super neural networks / Chrisantha Fernando, Dylan Banarse, Charles Blundell et al. // *arXiv preprint arXiv:1701.08734*. — 2017.
93. *Veniat Tom, Denoyer Ludovic*. Learning time-efficient deep architectures with budgeted super networks // *arXiv preprint arXiv:1706.00046*. — 2017.
94. *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>.
95. *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>.
96. *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>.
97. *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>.
98. *D. Kingma M. Welling*. Auto-Encoding Variational Bayes // Proceedings of the International Conference on Learning Representations (ICLR). — 2014.
99. 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>.
100. 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>.
101. Composing graphical models with neural networks for structured representations and fast inference / Matthew Johnson, David K Duvenaud, Alex Wiltschko et al. // Advances in neural information processing systems. — 2016. — Pp. 2946–2954.
102. *Nalisnick Eric, Smyth Padhraic*. Deep Generative Models with Stick-Breaking Priors // *arXiv preprint arXiv:1605.06197*. — 2016.
103. *Abbasnejad M Ehsan, Dick Anthony, van den Hengel Anton*. Infinite variational autoencoder for semi-supervised learning // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) / IEEE. — 2017. — Pp. 781–790.

104. *Miller A. C., Foti N., Adams R. P.* Variational Boosting: Iteratively Refining Posterior Approximations // *ArXiv e-prints*. — 2016. — nov.
105. *Arnold Ludovic, Ollivier Yann.* Layer-wise learning of deep generative models // *arXiv preprint arXiv:1212.1524*. — 2012.
106. *Karaletsos Theofanis, Rätsch Gunnar.* Automatic Relevance Determination For Deep Generative Models // *arXiv preprint arXiv:1505.07765*. — 2015.
107. *Sutskever Ilya, Vinyals Oriol, Le Quoc V.* Sequence to sequence learning with neural networks // *Advances in neural information processing systems*. — 2014. — Pp. 3104–3112.
108. *Hernández-Lobato José Miguel, Adams Ryan.* Probabilistic backpropagation for scalable learning of bayesian neural networks // *International Conference on Machine Learning*. — 2015. — Pp. 1861–1869.
109. *Kuznetsov Mikhail, Tokmakova Aleksandra, Strijov Vadim.* Analytic and stochastic methods of structure parameter estimation // *Informatica*. — 2016. — Vol. 27, no. 3. — Pp. 607–624.
110. *Shang Yi, Wah B. W.* Global optimization for neural network training // *Computer*. — 1996. — Mar. — Vol. 29, no. 3. — Pp. 45–54.
111. Gradient descent converges to minimizers / Jason D Lee, Max Simchowitz, Michael I Jordan, Benjamin Recht // *University of California, Berkeley*. — 2016. — Vol. 1050. — P. 16.
112. *Dembo Amir, Cover Thomas M, Thomas Joy A.* Information theoretic inequalities // *Information Theory, IEEE Transactions on*. — 1991. — Vol. 37, no. 6. — Pp. 1501–1518.
113. *Nicholas Altieri, D. Duvenaud.* Variational Inference with Gradient Flows. — дата обращения: 15.05.2016. URL: <http://approximateinference.org/accepted/AltieriDuvenaud2015.pdf>.
114. *Sato Issei, Nakagawa Hiroshi.* Approximation analysis of stochastic gradient langevin dynamics by using fokker-planck equation and ito process // *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. — 2014. — Pp. 982–990.
115. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks / Chunyuan Li, Changyou Chen, David E. Carlson, Lawrence Carin // *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, February 12-17, 2016, Phoenix, Arizona, USA. — 2016. — Pp. 1788–1794. <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11835>.
116. *Lichman M.* UCI Machine Learning Repository. — Дата обращения: 15.03.2017. URL: <http://archive.ics.uci.edu/ml>.
117. *LeCun Yann.* The MNIST database of handwritten digits // <http://yann.lecun.com/exdb/mnist/>. — 1998.

118. *Maclaurin Dougal, Adams Ryan P.* Firefly Monte Carlo: exact MCMC with subsets of data // Proceedings of the 24th International Conference on Artificial Intelligence / AAAI Press. — 2015. — Pp. 4289–4295.
119. Код вычислительного эксперимента. — Дата обращения: 15.03.2017. URL: svn.code.sf.net/p/mlalgorithms/code/Group074/Bakhteev2016Evidence/.