

Глава 1

Постановка задачи последовательного выбора моделей

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

Формализуем описанную выше задачу.

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

Задана простая выборка

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

состоящая из множества объектов

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

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

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

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

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

Определение 3. Пусть задан ациклический граф (V, E) , такой что

1. для каждого ребра $(j, k) \in E$: вектор базовых дифференцируемых функций $\mathbf{g}^{j,k} = [\mathbf{g}_0^{j,k}, \dots, \mathbf{g}_{K^{j,k}-1}^{j,k}]$ мощности $K^{j,k}$;
2. для каждой вершины $v \in V$: дифференцируемая функция агрегации \mathbf{agg}_v .

3. Функция $\mathbf{f} = \mathbf{f}_{|V|-1}$, задаваемая по правилу

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

$$v_k \in \{1, \dots, |V| - 1\}, \quad \mathbf{f}_0(\mathbf{x}) = \mathbf{x}$$

и являющаяся функцией из признакового пространства \mathbb{X} в пространство меток \mathbb{Y} при значениях векторов, $\boldsymbol{\gamma}^{j,k} \in [0, 1]^{K^{j,k}}$.

Граф (V, E) со множеством векторов базовых функций $\{\mathbf{g}^{j,k}, (j, k) \in E\}$ и функций агрегаций $\{\mathbf{agg}_v, v \in V\}$ назовем *параметрическим семейством моделей* \mathfrak{F} .

Примером функций агрегации выступают функции суммы и конкатенации векторов.

Определение 4. Функции $\mathbf{f}_0, \dots, \mathbf{f}_{|V|-1}$ из (1.2) назовем *слоями или подмоделями* модели \mathbf{f} .

Утверждение 1. Для любого значения $\boldsymbol{\gamma}^{j,k} \in [0, 1]^{K^{j,k}}$ функция $\mathbf{f} \in \mathfrak{F}$ является моделью.

Доказательство. Утверждение следует непосредственно из определения: по условию утверждения для любого $\boldsymbol{\gamma}^{j,k} \in [0, 1]^{K^{j,k}}$ функция является дифференцируемой функцией из признакового пространства \mathbb{X} в пространство меток \mathbb{Y} , что соответствует определению модели. \square

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

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

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

$$\begin{aligned} \mathbf{f} = \gamma_0^{1,2} \mathbf{softmax}(\gamma_0^{0,1} \mathbf{Conv}(\mathbf{x}, c_0, c_1)(\mathbf{x}) + \\ + \gamma_1^{0,1} \mathbf{Conv}(\mathbf{x}, 1, c_1) \circ \mathbf{Conv}(\mathbf{x}, c_0, c_2)(\mathbf{x}) \mathbf{w}_0^{1,2}). \end{aligned}$$

Определение 5. *Параметрами* модели \mathbf{f} из параметрического 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 $\mathbf{w}_l^{j,k}$.

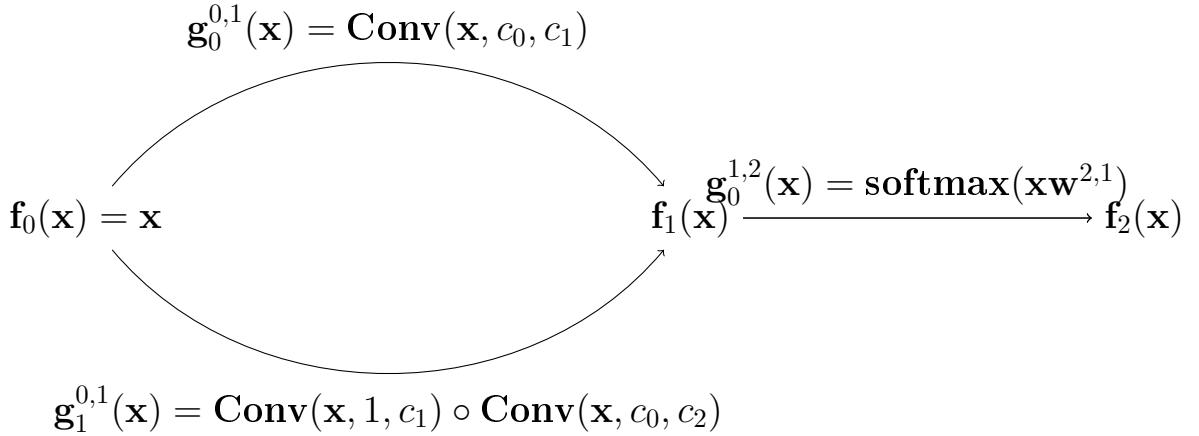


Рис. 1.1. Пример параметрического семейства моделей глубокого обучения: семейство описывает сверточную нейронную сеть.

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

Определение 7. *Параметризацией* множества моделей M назовем параметрическое 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 (1.2).

Предложенное определение параметризации не противоречит определению параметризации глубоких моделей в других работах. В [35] под параметризацией понимается представление матрицы параметров модели с использованием аппроксимации низкоранговыми матрицами. В [64] под параметризацией модели глубокого обучения понимается выбор графа, позволяющего описать структуру заданной модели глубокого обучения.

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

- 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.
- Структурные параметры лежат внутри булевого куба: $\gamma \in [0, 1]^{K^{j,k}}$. Релаксированная версия предыдущих ограничений, позволяющая проводить градиентную оптимизацию для структурных параметров.
- 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

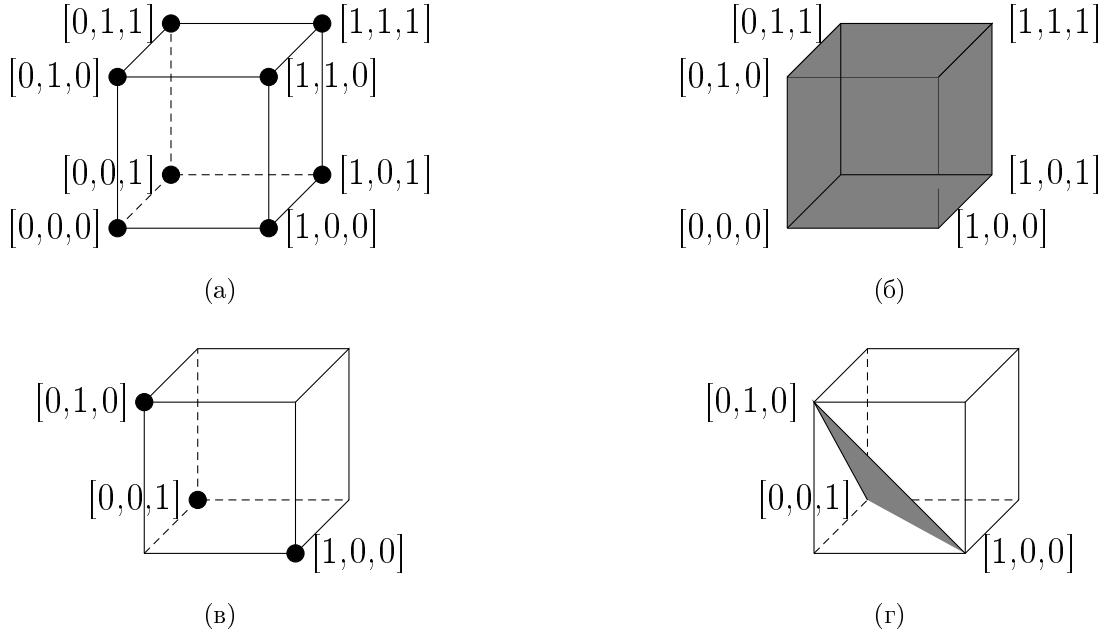


Рис. 1.2. Примеры ограничений для одного структурного параметра $\gamma^{j,k}, K^{j,k} = 3$.

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.

итоговую модель. Примером параметрического семейства моделей, требующим такое ограничение является семейство полносвязанных нейронных сетей с одним скрытым слоем и двумя значениями количества нейронов на скрытом слое. Схема семейства представлена на Рис. 1.5. Данное семейство можно представить как семейство с двумя базовыми функциями вида $\mathbf{g} = \sigma(\mathbf{w}^\top \mathbf{x})$, 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 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. Значение структурных параметров $\gamma^{j,k}$ интерпретируются как вклад каждой компоненты вектора базовых функций $\mathbf{g}^{j,k}$ в итоговую модель.

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

Для дальнейшей постановки задачи введем понятие вероятностной модели,

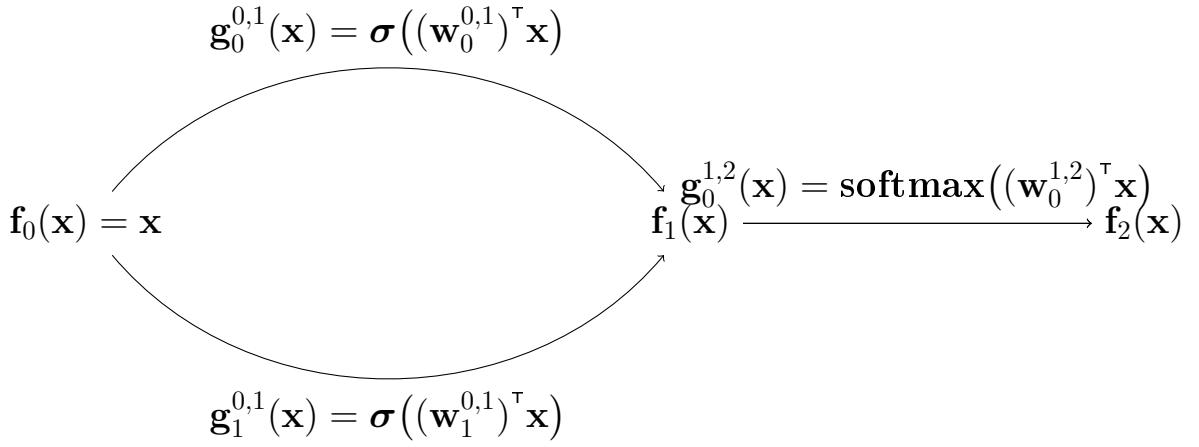


Рис. 1.3. Пример параметрического семейства моделей глубокого обучения: семейство описывает многослойную полносвязанную нейронную сеть с одним скрытым слоем и нелинейной функцией активации σ .

и связанных с ним определений. Будем полагать, что для параметров модели \mathbf{w} и структуры $\boldsymbol{\Gamma}$ задано распределение $p(\mathbf{w}, \boldsymbol{\Gamma} | \mathbf{h}, \boldsymbol{\lambda})$, соответствующее предположениям о распределении структуры и параметров.

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

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

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

где \mathbb{W} — множество значений параметров модели, \mathbb{H} — множество значений гиперпараметров, \mathbb{A} — множество значений метапараметров. Формальное определение последних будет дано далее.

Одной из 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

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

Определение 11. Вероятностной моделью глубокого обучения назовем совместное распределение вида

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

Определение 12. *Функцией правдоподобия выборки* назовем величину

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

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

Определение 13. *Обоснованностью модели* назовем величину

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

Получение значений апостериорного распределения и обоснованности модели сетей глубокого обучения является вычислительно сложной процедурой. Для получения оценок на данные величины используют методы, такие как аппроксимация Лапласа [29] и вариационная нижняя оценка [39]. В данной работе в качестве метода получения оценок обоснованности модели выступает вариационная нижняя оценка.

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

Определение 15. *Вариационными параметрами* модели $\boldsymbol{\theta} \in \Theta$ назовем параметры вариационного распределения $q(\mathbf{w}, \boldsymbol{\Gamma}|\boldsymbol{\theta})$.

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

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

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

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

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

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

$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}). \quad (1.6)$$

Определение 18. *Задачей выбора модели* \mathbf{f} назовем двухуровневую задачу оптимизации (1.5), (1.6).

Рассмотрим для примера базовый вариант выбора модели с применением функций q, L, Q .

Пример 1. Положим, что задано разбиение выборки на обучающую $\mathfrak{D}_{\text{train}}$ и валидационную $\mathfrak{D}_{\text{valid}}$ 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 $p(\mathbf{w}, \boldsymbol{\Gamma} | \mathbf{h}, \boldsymbol{\lambda})$. Положим в качестве функции $L(\boldsymbol{\theta} | \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})$ логарифм апостериорной вероятности модели:

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

Положим в качестве функции $Q(\mathbf{h} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\lambda})$ логарифм правдоподобия выборки при условии параметров \mathbf{w} и структуры $\boldsymbol{\Gamma}$:

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

Оптимизация параметров и структуры производится по обучающей выборке. Гиперпараметры \mathbf{h} выступают в качестве регуляризатора, чья оптимизация производится по валидационной выборке. Подобная оптимизация позволяет предотвратить переобучение модели [49].

Частным случаем задачи выбора структуры глубокой сети является выбор обобщенно-линейных моделей. Отдельные слои полносвязанных нейросетей являются обобщенно-линейными моделями. Задачу выбора обобщено-линейной моделей сводится к задаче выбора признаков, методы решения которой делятся на три группы [65]:

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

1.1. 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 *обоснованность модели* (1.4), являющееся байесовской интерпретацией *минимальной длины описания* [1], т.е. минимальное количество информации, которое требуется передать о модели и о выборке:

$$\text{MDL}(\mathbf{y}, \mathbf{f}) = \text{Len}(\mathbf{y}|\mathbf{w}^*, \mathbf{f}) + \text{COMP}(\mathbf{f}), \quad (1.7)$$

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} [1].

В общем случае правдоподобие модели является трудновычислимым. Для получения оценки правдоподобия используются вариационные методы получения оценки правдоподобия [2], основанные на аппроксимации неизвестного другим заданным распределением. Под субоптимальной сложностью понимается вариационная оценка правдоподобия модели. Альтернативной величиной, характеризующей сложность модели, выступает радемахеровская сложность (1.14). Данная величина используется как критерий для продолжения итеративного построения модели в [16].

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

Одним из методов получения приближенного значения правдоподобия модели является вариационный метод получения нижней оценки правдоподобия [2]. В работе [38] рассматривается стохастическая версия вариационного метода. В [39] рассматривается алгоритм получения вариационной нижней оценки правдоподобия для оптимизации гиперпараметров моделей глубокого обучения. В работе [40] рассматривается взаимосвязь градиентных методов получения ва-

риационной нижней оценки интеграла с методом Монте-Карло. В [41] рассматривается стохастический градиентный спуск в качестве оператора, порождающего распределение, аппроксимирующее апостериорное распределение параметров модели. В работе отмечается, что стохастический градиентный спуск не оптимизирует вариационную оценку правдоподобия, а приближает ее только до некоторого числа итераций оптимизации. Схожий подход рассматривается в работе [42], где также рассматривается стохастический градиентный спуск в качестве оператора, порождающего апостериорное распределение параметров. В работе [43] предлагается модификация стохастического градиентного спуска, аппроксимирующая апостериорное распределение.

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

1.2. 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 [8] предлагается удалять неинформативные параметры модели. Для этого находится точка оптимума $\boldsymbol{\theta}^*$ функции L , и производится разложение функции L в ряд Тейлора в окрестности $\boldsymbol{\theta}^*$:

$$L(\boldsymbol{\theta}^* + \Delta\boldsymbol{\theta} | \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}) - L(\boldsymbol{\theta}^* | \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}) = \frac{1}{2} \Delta\boldsymbol{\theta}^\top \mathbf{H} \Delta\boldsymbol{\theta} + o(\|\Delta\boldsymbol{\theta}\|^3), \quad (1.8)$$

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 (1.8) 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

$$\text{saliency}(\theta_i) = \frac{\theta_i^2}{2(H^{-1})_{i,i}}.$$

Данная функция характеризует информативность параметра.

В [66] было предложено развитие данного метода. В отличие от [8] не вводится предположений о диагональности гессиана функции ошибок, поэтому удаление неинформативных параметров модели производится точнее. Для получения оценок гессиана и его обратной матрицы применяется итеративный алгоритм.

Алгоритмы компрессии параметров модели. В [67, 68, 10] предлагаются методы компрессии параметров сетей глубокого обучения. Основным отличием задачи прореживания от задачи компрессии выступает эксплуатационное требование: если прореживание используется для получения оптимальной и наиболее устойчивой модели, то компрессия производится для уменьшения потребляемых вычислительных ресурсов при сохранении основных эксплуатационных характеристик исходной модели [68]. В [10] предлагается итеративное использование регуляризации типа DropOut [69] для прореживания модели. В [67, 68] 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 [68] 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 параметрических семействах моделей. Такой подход позволяет учитывать при выборе моделей не только эксплуатационные критерии качества модели, такие как точность итоговой модели и количество параметров в ней, но и некоторые статистические характеристики модели.

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

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

$$\rho = \exp\left(-\frac{\mu_j^2}{2\sigma_j^2}\right), \quad (1.9)$$

где μ_j, σ_j — среднее и дисперсия аппроксимирующего распределения q для па-

раметра w_j .

Идея данного метода была развита в [70], где также используются вариационные методы. В отличие от [39], в [70] 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:

$$p(\mathbf{w}|\mathbf{h}) \propto \prod_{\omega_i \in \Omega} \frac{1}{|\mathbf{h}_i|} \prod_{w \in \omega_i} \mathcal{N}(w|\mathbf{0}, \mathbf{h}_i^2),$$

где \mathbf{h}_i — гиперпараметр, соответствующий группе ω_i .

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:

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

где $h \in \mathbf{h}$ — гиперпараметр.

1.3. 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 [71, 72] рассматривается задача оптимизации гиперпараметров стохастическими методами. В [71] проводится сравнение случайного поиска значений гиперпараметров с переборным алгоритмом. В [72] 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 $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}' = T(\boldsymbol{\theta}|L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}), \quad (1.10)$$

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

Метапараметры соответствуют параметрам оптимизации, т.е. параметрам, которые не подлежат оптимизации в ходе задачи выбора модели.

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

Частным случаем оператора оптимизации является оператор стохастического спуска:

$$T(\boldsymbol{\theta}|L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}) = \boldsymbol{\theta} - \lambda_{\text{lr}} \nabla(-L(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})), \quad (1.11)$$

где λ_{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} :

$$\mathbf{h} = T(\mathbf{h}|Q, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\lambda}). \quad (1.12)$$

В случае, если для решения задачи (1.6) применяется несколько шагов оператора оптимизации (1.10), $\boldsymbol{\theta}^*$ рассматривается как рекурсивная функция от начального приближения вариационных параметров $\boldsymbol{\theta}^0$ и вектора гиперпараметров \mathbf{h} :

$$\boldsymbol{\theta}^* = T \circ \dots \circ T(\boldsymbol{\theta}|L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}) = \boldsymbol{\theta}^*(\boldsymbol{\theta}^0, \mathbf{h}). \quad (1.13)$$

Решение задачи оптимизации (1.12) при (1.13) является вычислительно сложным, поэтому применяются методы, аппроксимирующие применение градиентных методов при (1.13).

В [73] рассматривается оптимизация гиперпараметров градиентными методами для квадратичной функции потерь. В [49] в качестве оператора оптимизации гиперпараметров выступает метод градиентного спуска с моментом. Показано, что использование момента значительно снижает количество вычислительных ресурсов, требуемых для проведения оптимизации. В [74] предлагается аппроксимация градиентного метода, использующая предположение о линейности функции (1.13) от начального приближения $\boldsymbol{\theta}^0$. В [75] предлагается использовать численные методы для приближенного вычисления оператора оптимизации гиперпараметров. В [52] в качестве аппроксимации (1.13) предлагается рассматривать только последний шаг оптимизации:

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

где η — число шагов оптимизации.

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

В работе [76] предлагается моделировать качество модели Q (1.4) гауссовым процессом, параметрами которого выступают гиперпараметры исходной модели.

Одна из основных проблем использования гауссова процесса как суррогатной модели — кубическая сложность оптимизации. В работе [77] предлагается использовать случайные подпространства гиперпараметров для ускоренной оптимизации. В работе [78] предлагается комбинация из множества гауссовых моделей и линейной модели, позволяющая модели нелинейные зависимости гиперпараметров, а также существенно сократить сложность оптимизации.

В работе [79] предлагается рассматривать RBF-модель для аппроксимации качества Q исходной модели, что позволяет ускорить процесс оптимизации суррогатной модели. В [80] рассматривается глубокая нейронная сеть в качестве

суррогатной функции. Вместо интеграла правдоподобия (1.4), который оценивается в случае использования гауссового процесса в качестве суррогата, используется максимум апостериорной вероятности (1.3).

Одним из параметров гауссовых процессов является функция ядра гауссового процесса, полностью определяющая процесс в случае нулевого среднего. В работе [81] 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 [82] 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 числовых 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 .

1.4. 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 [83], а также в библиотеках типа TensorFlow [84], Theano [85], Pytorch [86], 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 [80, 79, 76], модель рассматривается как черный ящик, над которым производится ограниченный набор операций типа “произвести оптимизацию параметров” и “предсказать значение зависимой переменной по независимой пе-

ременной и параметрам модели". Подход, описанный в данных работах, также коррелирует с библиотеками машинного обучения, такими как Weka [87], RapidMiner [88] или sklearn [89], в которых модель машинного обучения рассматривается как черный ящик.

В [90] 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 [91] рассматриваются подходы к порождению моделей глубокого обучения. Предлагается формализация пространства поиска и формальное описание элементов пространства моделей. Приведем пример описания параметрического семейства моделей, соответствующего схеме из Рис. 1.1 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)) .
```

Прогнозирование графовых структур. В работе [92] предлагается метод прогнозирования графовой структуры на основе линейного программирования. Предлагается свести проблему поиска графовой структуры к комбинаторной проблеме. В работе [93] 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 методов порождения структур моделей глубокого обучения выступает стохастическое порождение структур. Данный тип порождения предполагает, что структуры порождаются случайно в соответствие вариационным распределением, заданным на структурах $q_{\Gamma}(\Gamma|\theta_{\Gamma})$. Затем выбирается одна, либо несколько наилучших структур с учетом валидационной функции Q или внешних, возможно недифференцируемых, критериев качества. Итоговая модель получается путем оптимизации параметров модели при выбранной структуре Γ . Заметим, что в ряде работ, одновременно порождается не только структура модели, но и итоговые параметры.

В работе [94] рассматривается порождение моделей, оптимизируемых без учителя. Модель представляется многослойным перцептроном вида:

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

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 индийских буфетов. Процесс индийского буфета 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 [95] предлагается метод выбора модели сверточной нейронной сети. Используется функция потерь, основанная на аппроксимации априорного распределения процесса индийского буфета для каждой базовой функции \mathbf{g}_j , являющейся j -м отображением объектов:

$$L = \sum_{\mathbf{x} \in \mathbf{X}} \left\| \mathbf{x} - \sum_{j=0}^{K-1} \mathbf{w}^j * \mathbf{g}_j(\mathbf{x}) \right\|_2^2 + \lambda^2 K,$$

где K — параметр, отвечающий за количество фильтров, λ — метапараметр алгоритма, знаком $*$ обозначается операция свертки.

В работе [96] предлагается ввести априорное распределение Бернулли на структурные параметры $\gamma^{j,k} \in \Gamma$.

В [97] 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 [5] приводятся теоретические оценки построения нейросетей с использованием жадных стратегий, при которых построение модели производится итеративно последовательным увеличением числа нейронов в сети. В работе [6] предлагается жадная стратегия выбора модели нейросети с использованием релевантных априорных распределений, т.е. параметрических распределений, оптимизация параметров которых позволяет удалить часть параметров из модели. Данный метод был к задаче построения модели метода релевантных векторов [7].

В работах [11, 12] рассматривается послойное построение модели с отдельным критерием оптимизации для каждого слоя. В работах [13, 14, 15] предлагается декомпозиция модели на порождающую и разделяющую, оптимизируемых последовательно.

В работах [98, 16] предлагается наращивание моделей, основанное на бустинге. Рассматривается задача построения нейросетевых моделей специального типа:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}_{|V|-1} \circ \mathbf{f}_{|V|-2} \circ \dots \mathbf{f}_0(\mathbf{x}), \quad \mathbf{f}_{i+1}(\mathbf{x}) = \sigma(\mathbf{f}_i(\mathbf{x})) + \mathbf{f}_i(\mathbf{x}),$$

приводится параметризация модели, позволяющая рассматривать декомпозицию модели на слабые классификаторы. В [16] рассматривается задача выбора полносвязной нейронной сети для задачи бинарной классификации, $R = 2$.

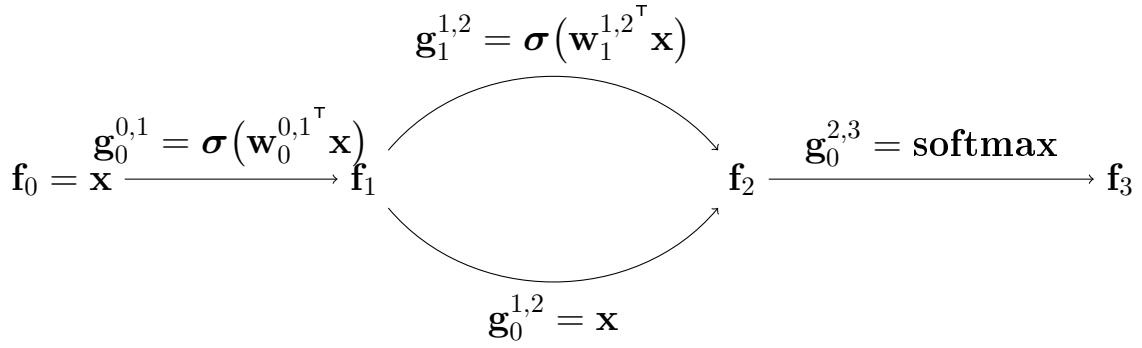


Рис. 1.4. Пример итерации алгоритма AdaNet [16]. Рассматриваются две альтернативные модели: модель с углублением сети (соответствует занулению функции \mathbf{f}_2 с использованием базовой функции $\mathbf{g}_1^{1,2}$) и модель с расширением сети (соответствует базовой функции $\mathbf{g}_0^{1,2}$).

В качестве функции агрегации для подмодели \mathbf{f}_3 выступает конкатенация: $\text{agg}_2 = \text{concat}$.

На каждом шаге построения выбирается одно из двух расширений модели, каждое из которых рассматривается как слабый классификатор: сделать модель шире или сделать модель глубже. Пример работы AdaNet представлен на Рис. 1.4. Построение модели заканчивается при условии снижении радемахеровской сложности:

$$\mathfrak{R} = \frac{1}{m} \mathbb{E}_{b_1, \dots, b_m} \sup_{\mathbf{w}} \sum_{i=1}^m b_i [y_i \neq \arg \max_c f[c](\mathbf{x}, \mathbf{w})], \quad (1.14)$$

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

В работе [99] 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 предсказывает качество модели по заданным подмоделям, и на основе данного предсказания выбрать наилучшую модель.

В работе [100] предлагается метод анализа структуры сети на основе линейных классификаторов, построенных на промежуточных слоях нейросети. Схожий метод был предложен в [101], где классификаторы на промежуточных уровнях используются для уменьшения вычислений при выполнении вывода и предсказаний. Промежуточные классификаторы работают как решающий список.

В работе [102] предлагается инкрементальный метод оптимизации нейросети. На первом этапе модель декомпозируется на несколько подмоделей, при

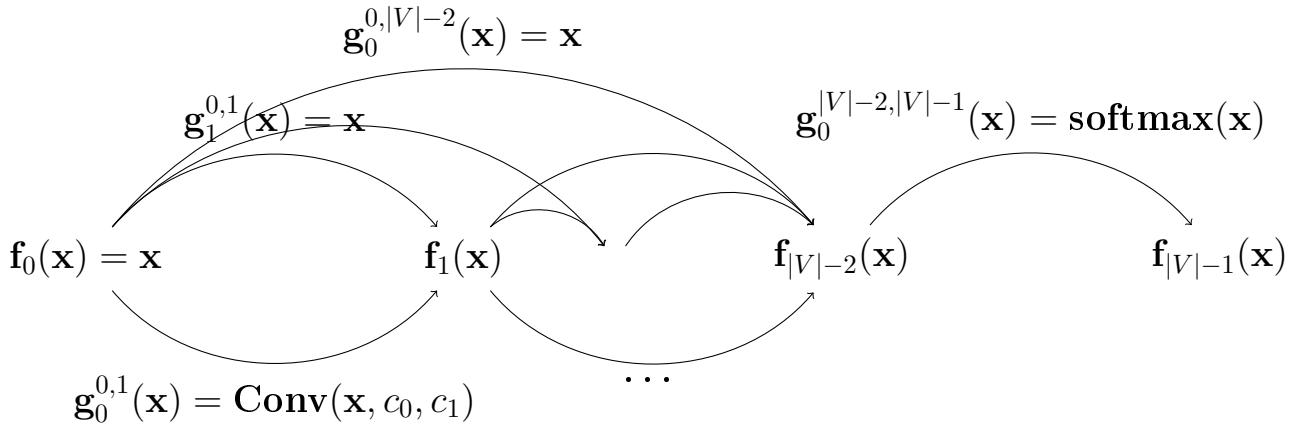


Рис. 1.5. Пример параметрического семейства моделей глубокого обучения, описываемый в [17]. Каждая подмодель \mathbf{f}_j является линейной комбинацией базовых функций: свертки и результата работы предыдущих подмоделей (англ. skip-connection).

которой модель последовательностью слоев $\mathbf{f}_1, \dots, \mathbf{f}_{|V|}$. Проводится последовательная оптимизация моделей вида:

- 1) $\mathbf{f} = \mathbf{f}_{|V|-1}(\mathbf{x})$;
- 2) $\mathbf{f} = \mathbf{f}_{|V|-2} \circ \mathbf{f}_{|V|}(\mathbf{x})$;
- 3) ...
- 4) $\mathbf{f} = \mathbf{f}_0 \circ \dots \circ \mathbf{f}_{|V|-1}(\mathbf{x})$.

Оптимизация структуры модели на основе обучения с подкреплением. В [17] предлагается итеративная схема выбора архитектуры сверточной нейросети с использованием обучения с подкреплением. Распределение структур и параметров $q(\mathbf{w}, \mathbf{\Gamma} | \boldsymbol{\theta})$ 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.

В работе [18] предлагается алгоритм построения регрессионной модели для оценки финального качества модели и ранней остановки оптимизации моделей. Он позволяет существенно ускорить поиск моделей, представленный в [17]. В [20] рассматривается задача переноса архитектуры нейросети, чья структура была выбрана по выборке, меньшей мощности. Как и в [17] предлагается метод параметризации сверточной нейронной сети в виде графа. Предложенная параметризация позволяет задать более мощное параметрическое семейство моделей, чем в [17]. Модель представляется в виде последовательности суперпозиций подмоделей, называемых клетками (англ. normal cell и reduction cell). Каждая из этих клеток содержит следующее множество нелинейных операций \mathbf{g} , состоящее из тождественной операции $\mathbf{g}(\mathbf{x}) = \mathbf{x}$, а также множество сверток

с фиксированным количеством каналов и размером фильтров и функций субдискретизации или пулинга. Алгоритм выбора структуры модели рекуррентной сетью выглядит следующим образом на шаге j :

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

В отличие от предыдущих работ, в работе [19] предлагается подход к инкрементальному обучению нейросети, основанном на модификации модели, полученной на предыдущем шаге. Рассматривается две операции над нейросетью: расширение и углубление сети.

В работах [103, 104, 105] рассматриваются методы деформации нейросетей. В работе [105] предлагается метод оптимального разделения нейросети на несколько независимых сетей для уменьшения количества связей и, как следствие, уменьшения сложности оптимизации модели. В работе [103] предлагается метод сохранения результатов оптимизации нейросети при построении новой более глубокой или широкой нейросети. В работе [104] рассматривается задача расширения сверточной нейросети, нейросеть рассматривается как граф.

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

1.5. 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 [106], 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 [107] предлагается подход к адаптивному изменению параметров сети. В качестве оператора оптимизации параметров рассматривается величина:

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

где \mathbf{f}_{mod} — функция, определяющая номер параметра из $\boldsymbol{\theta}$, подлежащего оптимизации, а $\mathbf{f}_{\text{optim}}$ — величина изменения параметра. В [107] 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}_{mod} , \mathbf{f}_{ana} , $\mathbf{f}_{\text{optim}}$.

В работе [108] рассматривается оптимизация метапараметров (шага градиентного спуска λ_{lr} и начального распределения параметров $\boldsymbol{\theta}^0$). Рассматривается задача оптимизации параметров модели в случае, когда количество примеров невелико. Для этого проводится оптимизация параметров оператора оптимизации, который выглядит следующим образом:

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

где векторы $\boldsymbol{\theta}^0$ и $\boldsymbol{\lambda}$ являются метапараметрами 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 [109] 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 , действующего на параметры $\boldsymbol{\theta}^0$, где $\boldsymbol{\theta}^0$ — параметры модели, оптимизированной на небольшой выборке. Предлагается функция оптимизации:

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

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

В работе [110] рассматривается оптимизация метапараметров оператора оптимизации с помощью модели долгой краткосрочной памяти LSTM, которая выступает альтернативе аналитических алгоритмов, таких как Adam [111] или AdaGrad [112]. 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:

$$Q(\mathbf{h}|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \sum_{t=1}^{\eta} L(\boldsymbol{\theta}^t|\mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}),$$

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

1.6. Выбор структур моделей специального вида

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

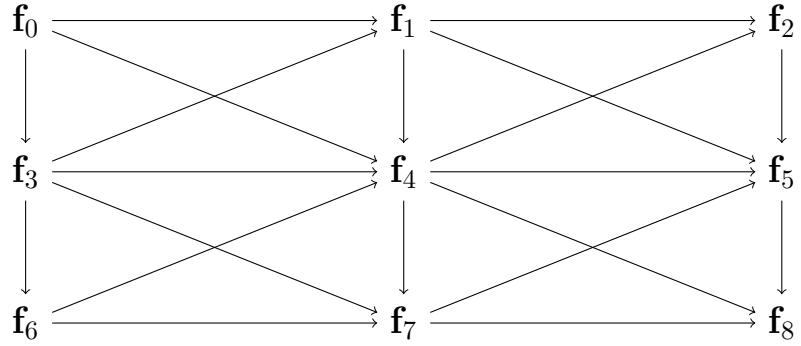


Рис. 1.6. Пример суперсети. Каждый путь из подмодели f_0 в конечную модель f_8 задает модель глубокого обучения.

В работе [113] рассматривается оптимизация моделей нейросетей с бинарной функцией активацией. Задача оптимизации сводится к задаче mixed integer программирования, которая решается методами выпуклого анализа. В работе [114] 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 [115, 116] рассматривается выбор архитектуры сети с использованием *суперсетей*: связанных между собой подмоделей, образующих граф, каждый путь из нулевой вершины в последнюю которого определяет модель глубокого обучения. Пример графа, описывающего суперсеть представлен на Рис. 1.6. В работе [116] рассматриваются стохастические суперсети, позволяющие выбрать структуру нейросети за ограниченное время оптимизации. Схожий подход был предложен в работе [115], 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 ($R = 0$).

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

автокодировщика и ступенчатых сетей [27] для случая построения многослойных порождающих моделей.

В ряде работ [117, 118, 119, 120, 121] рассматривается подход к построению порождающих моделей глубокого обучения, при котором каждая подмодель \mathbf{f}_i приближает распределение некоторой случайной величины \mathbf{z}_i , которая влияет на маргинальное распределение $p(\mathbf{X}, \mathbf{y}) = \int_{\mathbf{z}_0, \dots, \mathbf{z}_{|V|-1}} p(\mathbf{X}, \mathbf{y} | \mathbf{z}_0, \dots, \mathbf{z}_{|V|-1}) p(\mathbf{z}_1, \dots, \mathbf{z}_{|V|}) d\mathbf{z}_0 \dots d\mathbf{z}_{|V|-1}$. Подобный подход позволяет использовать вероятностную интерпретацию для каждой отдельной подмодели.

В работе [117] 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 [118], авторы рассматривают использование процесса сломанной трости в вариационном автокодировщике, тем самым получая модель со стохастической размерностью скрытой переменной. В [119] рассматривается смесь автокодировщиков, где смесь моделируется процессом Дирихле.

В работе [120] 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 [121] рассматривается задача построения порождающих моделей с дискретными значениями скрытой переменной \mathbf{z} , предлагается критерий для послойного обучения порождающих моделей:

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

где q — аппроксимирующее распределение для случайной величины \mathbf{z} , i пробегает все значения переменной \mathbf{z} .

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

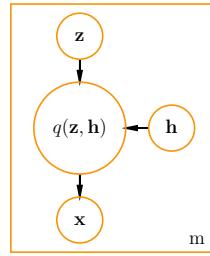


Рис. 1.7. Схема порождения вектора объектов \mathbf{X} , представленная в [53].

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

Глава 2

Выбор модели с использованием вариационного вывода

В данной главе рассматривается задача выбора моделей глубокого обучения субоптимальной сложности. Под сложностью модели понимается обоснованность модели (1.4). Под субоптимальной сложностью понимается приближенная оценка обоснованности модели, полученная с использованием вариационных методов. Вводятся вероятностные предположения о распределении параметров. На основе байесовского вывода предлагается функция обоснованности модели. Для получения оценки обоснованности применяются вариационные методы с использованием градиентных алгоритмов оптимизации. Проводится вычислительный эксперимент на нескольких выборках.

В работе предлагается метод получения вариационной нижней оценки обоснованности модели с использованием модифицированного алгоритма стохастического градиентного спуска. Модификация заключается в добавлении шумовой компоненты. Эта компонента позволяет получить более точные оценки обоснованности 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 обоснованности обучающей выборки и изменении значений параметров с его учетом. Приводится сравнение с алгоритмом получения вариационной нижней оценки, представленном в [39]. Рассматриваются следующие модификации базового алгоритма: оптимизация с кросс-валидацией с использованием и без использования регуляризации модели, алгоритм получения вариационной оценки обоснованности модели с применением нормального распределения, алгоритм получения вариационной оценки обоснованности с использованием стохастического градиентного спуска, алгоритм получения вариационной оценки обоснованности с использованием стохастической динамики Ланжевена. Данные алгоритмы решают следующие проблемы оптимизации моделей градиентным спуском: оптимизация модели с меньшими затратами вычислительных ресурсов, быстрая сходимость оптимизации, контроль переобучения и выбор наиболее адекватной модели. Под переобучением понимается потеря обобщающей способности модели с увеличением правдоподобия обучающей выборки [28]. Переобучение характерно для моделей с большим количеством параметров, сопоставимым с мощностью обучающей выборки, что встречается в случае выбора моделей глубокого обучения [3, ?]. Также алгоритмы имеют дальнейшую возможность применения к градиентным алгоритмам оптимизации гиперпараметров, описанным в [49].

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

метров модели.

2.1. Постановка задачи оптимизации обоснованности моделей

Определим понятие статистической сложности модели. Сложностью модели будем называть *обоснованность модели* (1.4). Пусть задано множество моделей M , для которых, возможно, не задан график параметрического семейства моделей. Для каждой модели $\mathbf{f} \in M$ заданы различные значения гиперпараметров \mathbf{h} . Рассмотрим два подхода к сравнению моделей:

1. Модели \mathbf{f} описываются общим параметрическим семейством моделей глубокого обучения \mathfrak{F} , т.е. имеют общий график описания моделей (V, E) , общее пространство параметров \mathbb{W} и пространство структур Γ . При таком подходе сравнение сложности различных моделей является адекватным, т.к. они определены на общем пространстве структур Γ и параметров \mathbb{W} . Возможно сравнение не только обоснованности модели, но и распределения на структуре Γ .
2. Модели \mathbf{f} не описываются общим параметрическим семейством. В данном случае напрямую сравнить распределение структур Γ нельзя.

В данном разделе рассматривается второй вариант. Будем полагать, что структура модели Γ для вероятностной модели глубокого обучения \mathbf{f} определена однозначно и метапараметры $\boldsymbol{\lambda}$ определены однозначно:

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

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

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

Заметим, что основная часть данной главы также применим и в случае, когда модели описываются общим параметрическим семейством моделей глубокого обучения. В данном случае вместо интеграла (2.1) предлагается использовать интеграл (1.4), учитывающий вероятностные предположения о структуре модели.

Определение 21. Модель \mathbf{f} назовем оптимальной среди моделей M , если достигается максимум интеграла (2.1).

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

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

Пример 2. Рассмотрим задачу линейной регрессии:

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

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

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

априорное распределение параметров модели имеет вид

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

обоснованность модели (1.4) в этом примере вычисляется аналитически [?]:

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

где $\hat{\mathbf{w}}$ — значение наиболее вероятных (1.3) 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:

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

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

Пример 3. Рассмотрим задачу классификации, в которой модель — нейросеть с softmax-слоем на выходе:

$$\mathbf{f} = \mathbf{f}_{|V|-1}(\mathbf{f}_{|V|-2}(\dots \mathbf{f}_0(\mathbf{x}))), \quad (2.6)$$

$\mathbf{f}_0, \dots, \mathbf{f}_{|V|-1}$ — дифференцируемые функции, $\mathbf{f}_{|V|-1}$ — многомерная логистическая функция:

$$\mathbf{f}_{|V|-1} = \frac{\mathbf{f}_{|V|-2}(\dots \mathbf{f}_1(\mathbf{x}))}{\sum_{c=1}^R \exp(f[c]_{|V|-2}(\dots \mathbf{f}_1(\mathbf{x})))}, \quad (2.7)$$

где $f_{|V|-2}[c]$ — c -я компонента функции $\mathbf{f}_{|V|-2}$. Компонента c вектора $\mathbf{f}_{|V|-1}$ определяет вероятность принадлежности объекта \mathbf{x} к классу c . Логарифм правдоподобия зависимой переменной аналогично (2.3) имеет вид

$$\log p(y|\mathbf{x}, \mathbf{w}) = \log f_{|V|-1}^y(\mathbf{f}_{|V|-2}(\dots \mathbf{f}_1(\mathbf{x}))).$$

Данная модель описывает многослойную сеть, аналогичную моделям семейства, представленного на Рис. 1.5.

Интеграл обоснованности (2.1) модели является трудновычислимым для данного семейства моделей. Одним из методов вычисления приближенного значения обоснованности является получение вариационной оценки обоснованности.

В качестве функции, приближающей логарифм интеграла (2.1), будем рассматривать его нижнюю оценку, полученную при помощи неравенства Йенсена [2]. Получим нижнюю оценку логарифма обоснованности модели, используя неравенство

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

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

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

Определение 22. Вариационной оценкой логарифма обоснованности модели (2.1) $\log p(\mathbf{y}|\mathbf{X}, \mathbf{h})$ называется оценка $\hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{h})$, 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 (1.4) к оптимизации вида

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

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

Оценка (2.8) является нижней, поэтому может давать некорректные оценки для обоснованности (2.1). Для того, чтобы оценить величину этой ошибки, докажем следующее утверждение.

Теорема 1 ([2]). 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 (1.4) эквивалентна минимизации расстояния Кульбака–Лейблера между распределением $q(\mathbf{w}) \in \mathfrak{Q}$ и апостериорным распределением параметров $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})$:

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

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

Доказательство. Доказательство непосредственно следует из (2.8). Вычитая из обеих частей равенства $D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h}))$, получим

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

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

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

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 (2.9)

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

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

Вариационная оценка (2.8) интерпретируется как оценка сложности модели по принципу минимальной длины описания (1.7), где первое слагаемое определяет количество информации для описания выборки, а второе слагаемое — длину описания самой модели [39].

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

2.2. Методы получения вариационной оценки обоснованности

Ниже представлены методы получения вариационных нижних оценок (2.10) обоснованности (1.4). В первом параграфе рассматривается метод, основанный

на аппроксимации апостериорного распределения $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})$ (1.3) 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:

$$q \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}), \quad \boldsymbol{\theta} = [\boldsymbol{\mu}_q, \text{diag}(\mathbf{A}_q^{-1})] \quad (2.11)$$

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

Пусть априорное распределение $p(\mathbf{w}|\mathbf{h})$ (3.1) 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 (2.9) имеет вид

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

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

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

В качестве приближенного значения интеграла

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

предлагается использовать формулу

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

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

Итоговая функция оптимизации (2.12) имеет вид

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

Пример 4. Пусть задана выборка \mathfrak{D} , в которой переменная y не зависит от \mathbf{x} :

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

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

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

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

Рис. 2.0. а

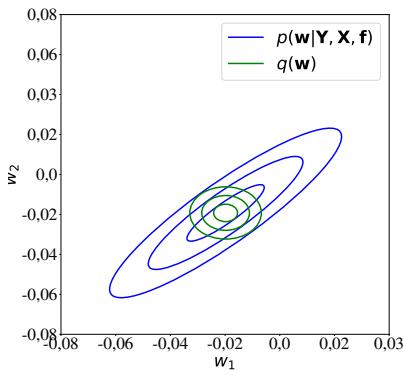


Рис. 2.0. б

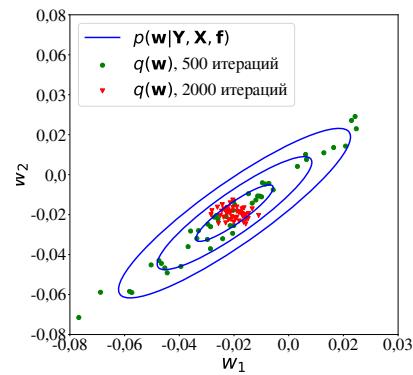


Рис. 2.0. в

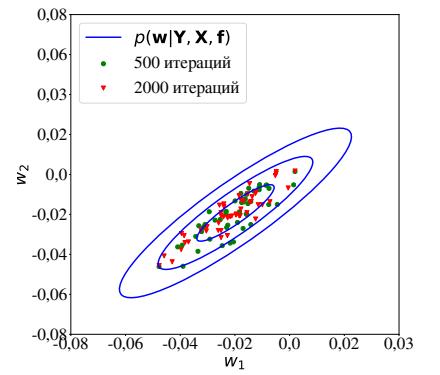


Рис. 2.1. Аппроксимация распределения а) нормальным распределением, б) распределением, полученным с помощью градиентного спуска, в) с использованием стохастической динамики Ланжевена.

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

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 (2.8)

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

где S — энтропия распределения:

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

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

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

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

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

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,

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

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

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

Для дальнейшего описания метода введем понятие оператора градиентного спуска, являющегося частным случаем оператора оптимизации (1.10).

Определение 24. Оператором градиентного спуска назовем оператор оптимизации вида

$$T(\boldsymbol{\theta} | L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}) = \boldsymbol{\theta} - \lambda_{\text{lr}} \nabla (-L(\boldsymbol{\theta} | \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})), \quad (2.17)$$

где λ_{lr} — длина шага градиентного спуска.

В данной главе будем рассматривать распределения, полученные из нескольких точек старта оптимизации параметров \mathbf{w} модели \mathbf{f} . Для простоты записи будем использовать запись $L(\mathbf{w})$ как эквивалентную форму записи $L(\boldsymbol{\theta} | \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})$ для $\boldsymbol{\theta} = [\mathbf{w}]^T$, и $T(\mathbf{w})$ как эквивалентную форму записи $T(\boldsymbol{\theta} | L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})$.

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

Теорема 2. Пусть T — оператор градиентного спуска, L — функция потерь, градиент ∇L которой имеет константу Липшица C_L . Пусть $\boldsymbol{\theta} = [\mathbf{w}^1, \dots, \mathbf{w}^r]^T$ — начальные приближения оптимизации модели, где r — число начальных приближений. Пусть λ_{lr} — длина шага градиентного спуска, такая что

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

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

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

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

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

Предварительно приведем две леммы, требуемые для доказательства теоремы.

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

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

$$S(q'(\mathbf{w})) - S(q(\mathbf{w})) = \int_{\mathbf{w}} q'(\mathbf{w}) \log \left| \frac{\partial T(\mathbf{w})}{\partial \mathbf{w}} \right| d\mathbf{w}. \quad (2.20)$$

Доказательство. Рассмотрим очередной шаг оптимизации. При $\lambda_{lr} < \frac{1}{C}$ оператор градиентного спуска T является диффеоморфизмом, а значит, и биекцией, справедлива формула (2.20). По усиленному закону больших чисел

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

где \approx означает сходимость почти наверное. Логарифм якобиана $\log \left| \frac{\partial T(\mathbf{w}'^l)}{\partial \mathbf{w}} \right|$ оператора T запишем как

$$\log \left| \frac{\partial T(\mathbf{w}'^l)}{\partial \mathbf{w}} \right| = \log |\mathbf{I} - \lambda_{lr} \mathbf{H}| = \sum_{i=1}^{|\mathbb{W}|} \log (1 - \lambda_{lr} \lambda_i), \quad (2.21)$$

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

При $(\lambda_{\text{lr}} \lambda_i)^2 \leq (\lambda_{\text{lr}} \lambda_{\max})^2 < 1$ выражение (2.21) раскладывается в ряд Тейлора:

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

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

Получим итоговую формулу для оценки обоснованности модели.

Теорема 3. Оценка (2.15) на шаге оптимизации τ представима в виде

$$\begin{aligned} \log \hat{p}(\mathbf{y} | \mathbf{X}, \mathbf{h}) &\approx \frac{1}{r} \sum_{g=1}^r L(\mathbf{w}_\tau^l | \mathbf{X}, \mathbf{y}) + \\ &+ S(q^0(\mathbf{w})) + \frac{1}{r} \sum_{b=1}^r \sum_{l=1}^r (-\lambda_{\text{lr}} \text{Tr}[\mathbf{H}(\mathbf{w}_b^l)] - \lambda_{\text{lr}}^2 \text{Tr}[\mathbf{H}(\mathbf{w}_b^l) \mathbf{H}(\mathbf{w}_b^l)]) \end{aligned} \quad (2.22)$$

с точностью до слагаемых вида $o_{\lambda_{\text{lr}}^2 \rightarrow 0}(1)$, 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:

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

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

$$S(q^b(\mathbf{w})) - S(q^{b-1}(\mathbf{w})) \approx \frac{1}{r} \sum_{l=1}^r (-\lambda_{\text{lr}} \text{Tr}[\mathbf{H}(\mathbf{w}_b^l)] - \lambda_{\text{lr}}^2 \text{Tr}[\mathbf{H}(\mathbf{w}_b^l) \mathbf{H}(\mathbf{w}_b^l)]). \quad (2.23)$$

Формула (2.22) получается подстановкой в выражение (2.15) суммы выражений вида (2.23), а также начальной энтропии $S(q^0(\mathbf{w}))$. \square

В [41] предлагается алгоритм приближенного вычисления для выражения, находящегося под знаком суммы в (2.22):

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

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

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

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: $\boldsymbol{\theta} = T(\boldsymbol{\theta}|L, \mathbf{X}, \mathbf{y}, \mathbf{h}, \lambda_{lr})$, где $\boldsymbol{\theta} = [\mathbf{w}_1, \dots, \mathbf{w}_r]^\top$;
- 6: **для** $l = 1, \dots, r$
- 7: $\mathbf{r}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
- 8: $\mathbf{r}_1 = \mathbf{r}_0 - \lambda_{lr} \mathbf{r}_0^\top \nabla \nabla (-L(\mathbf{w}^l|\mathbf{y}, \mathbf{X}))$;
- 9: $\mathbf{r}_2 = \mathbf{r}_1 - \lambda_{lr} \mathbf{r}_1^\top \nabla \nabla (-L(\mathbf{w}^l|\mathbf{y}, \mathbf{X}))$;
- 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: $\hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{h}) = \frac{1}{r} \sum_{l=1}^r p(\mathbf{y}|\mathbf{X}, \mathbf{w}^l)$;
- 13: $\hat{p}(\mathbf{w}|\mathbf{h}) = \frac{1}{r} \sum_{l=1}^r p(\mathbf{w}^l|\mathbf{h})$;
- 14: $\log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{h}) = \log \hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathbf{h}) + \log \hat{p}(\mathbf{w}|\mathbf{h})$;

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

Заметим, что при приближении параметров модели к точке экстремума оценка обоснованности 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 обоснованности 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 (1.11), оптимизирующий параметры $\mathbf{w}^1, \dots, \mathbf{w}^r$ по некоторой случайной подвыборке $\hat{\mathbf{X}}, \hat{\mathbf{y}}$, одинаковой для каждой точки старта $\mathbf{w}^1, \dots, \mathbf{w}^r$:

$$T(\boldsymbol{\theta}|L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}) = \boldsymbol{\theta} - \lambda_{lr} \nabla (-L(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})),$$

где λ_{lr} — шаг градиентного спуска, $\hat{\mathbf{y}}, \hat{\mathbf{X}}$ — случайная подвыборка заданной мощности выборки \mathfrak{D} , где $\hat{\mathbf{X}}$ — случайная подвыборка выборки \mathbf{X} , одинаковая для всех точек мультистарта, $\hat{\mathbf{y}}$ — соответствующие метки классов,

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

Как и версия алгоритма с использованием градиентного спуска (2.17), ос-

новной проблемой модифицированного алгоритма оценки интеграла (2.10) является грубость аппроксимации исходного распределения $p(\mathbf{w}|\mathbf{f}, \mathcal{D})$.

Рассмотрим пример (2.14). График аппроксимации распределения $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})$ представлен на рис. 2.1, б. Как видно из графика, градиентный спуск сходится к mode распределения. При небольшом количестве итераций полученное распределение также слабо аппроксимирует апостериорное распределение. При приближении к точке экстремума снижается вариационная оценка обоснованности модели, что интерпретируется как возможное начало переобучения [41]. Таким образом, снижение оценки (2.22) можно использовать как критерий остановки оптимизации модели для снижения эффекта переобучения.

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

Аппроксимация с использованием динамики Ланжевена

Для достижения нижней оценки интеграла (2.10), более близкой к реальному значению логарифма интеграла (1.4), чем оценка с использованием градиентного спуска, предлагается использовать стохастическую динамику Ланжевена [43]. Стохастическая динамика Ланжевена представляет собой вариант стохастического градиентного спуска с добавлением гауссового шума:

$$T(\boldsymbol{\theta}|L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}) = \boldsymbol{\theta} - \frac{m}{\hat{m}} \lambda_{lr} \nabla (-L(\boldsymbol{\theta}|\mathbf{h}, \hat{\mathbf{X}}, \hat{\mathbf{y}})) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \frac{\lambda_{lr}}{2} \mathbf{I}), \quad (2.24)$$

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

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

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

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

где τ — текущий шаг оптимизации, $S(\mathcal{N}(0, \frac{\lambda_{lr}}{2}))$ — 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 $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})$ [43, ?]. График аппроксимации апостериорного распределения с использованием динамики Ланжевена представлен на рис. 2.1,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 обоснованности (2.10). В то же время алгоритм более требователен к настройке параметров оптимизации [?]: “быстро изменяющаяся кривизна [траекторий параметров модели] делает методы стохастической градиентной динамики Ланжевена по умолчанию неэффективными”.

2.3. Анализ методов выбора моделей

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

Для выборок Boston, Protein и MSD была рассмотрена задача регрессии

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

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

$$\mathbf{f}(\mathbf{w}, \mathbf{X}) = (\mathbf{w}_0^{1,2})^\top \text{softplus}((\mathbf{w}_0^{0,1})^\top \mathbf{X}), \quad (2.25)$$

где $\mathbf{w}_0^{0,1} \in \mathbb{R}^{n \times n_1}$ — матрица параметров скрытого слоя нейросети, $\mathbf{w}_0^{1,2} \in \mathbb{R}^{n_1 \times 1}$ — матрица параметров выходного слоя нейросети, $\text{softplus}(\mathbf{X}) = \log(1 + \exp(\mathbf{X}))$.

Для выборки Boston также было рассмотрено множество моделей с тремя скрытыми слоями, построенных аналогично однослоиной модели (2.25). Размер каждого слоя равнялся 50.

Для выборки MNIST была рассмотрена задача бинарной классификации: из выборки были взяты только объекты, соответствующие цифрам 7 и 9. Размерность выборки была понижена с 784 до 50 методом главных компонент аналогично [?]. Для анализа моделей, полученных в случае высокой вероятности

переобучения, из обучающей выборки были взяты первые 500 объектов. В качестве модели рассматривалась нейросеть с тремя скрытыми слоями и функцией активации вида:

$$\sigma(\mathbf{X}) = (1 + \exp(-\mathbf{X}))^{-1}.$$

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

Оптимизация параметров производилась на подвыборке $\mathfrak{D}_{\text{train}}$. Для контроля переобучения некоторых алгоритмов из обучающей выборки $\mathfrak{D}_{\text{train}}$ формировалась валидационная выборка $\mathfrak{D}_{\text{valid}}$, на которой не проводилась оптимизация параметров модели. Мощность валидационной выборки $\mathfrak{D}_{\text{valid}}$ составляла 0,1 мощности обучающей выборки $\mathfrak{D}_{\text{train}}$, объекты для валидационной выборки выбирались случайным образом независимо для каждого старта алгоритма. Качество полученных моделей проверялось на подвыборке $\mathfrak{D}_{\text{test}}$. Критерием качества модели выступали среднеквадратичное отклонение вектора \mathbf{y} от вектора $\mathbf{f}(\mathbf{w}, \mathbf{X})$ (RMSE) в случае задачи регрессии и доля верно предсказанных меток класса (Accuracy) в задаче классификации, а также соответствующие критерии при возмущении элементов выборки:

$$\text{RMSE}_\sigma = \text{RMSE}(\mathbf{f}(\mathbf{w}, \mathbf{X} + \boldsymbol{\varepsilon}), \mathbf{y}), \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma \mathbf{I}). \quad (2.26)$$

Были рассмотрены шесть алгоритмов.

1. Базовый алгоритм: оптимизация параметров без валидации и ранней остановки. Оптимизация проводилась с использованием стохастического градиентного спуска (2.17). 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 Ланжеяна.
6. Нахождение вариационной нижней оценки с аппроксимацией нормальным распределением (2.13).

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

не улучшалось $3\tau_{\text{val}}$ итераций. Для разных алгоритмов внутренним критерием качества выступали различные величины:

- 1) функция потерь L (2.16) на валидационной выборке $\mathfrak{D}_{\text{valid}}$ для алгоритмов 2, 3;
- 2) вариационная нижняя оценка обоснованности (2.8) 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 λ_{lr} , отбирались наилучшие модели.

Описание эксперимента представлено в табл. 1. Результаты экспериментов представлены в табл. 2. На рис. 2.3 представлен график зависимости RMSE_σ от параметра σ для однослойных моделей.

Рис. 2.2. *a*

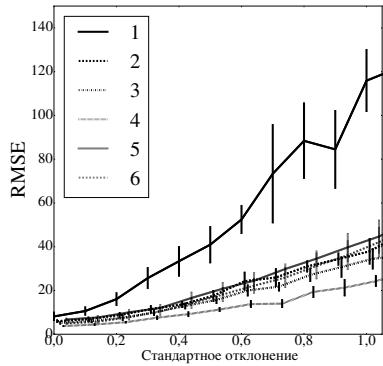


Рис. 2.2. *б*

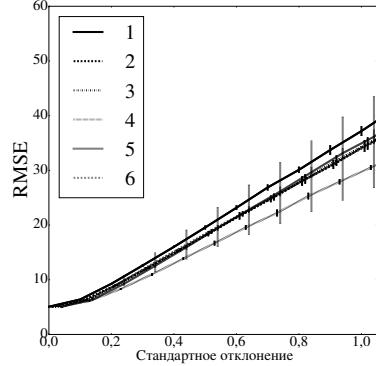


Рис. 2.2. *в*

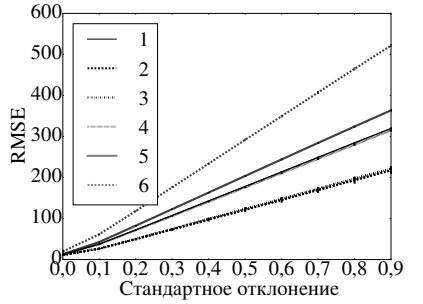


Рис. 2.3. Возмущение выборки для однослойных нейросетей: *a*) Boston Housing, *б*) Protein, *в*) MSD.

Таблица 2.1. 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

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

Таблица 2.2. Результаты эксперимента по выбору моделей

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

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 (1.3) от нормального.

Алгоритм 6 показал низкое качество (2.26) при возмущении объектов выборки в большинстве экспериментов. В трех экспериментах наилучшие показатели по данному критерию показал алгоритм 4. Заметим, что алгоритм 5, являющийся модификацией алгоритма 4, показал худшие результаты как по RMSE, так и по RMSE при возмущении объектов выборки. На выборке MNIST алгоритм 4 показал результаты значительно хуже остальных алгоритмов. В целом результаты по данному алгоритму схожи с результатами, описанными в [41]: в отличие от алгоритма 5 алгоритм 4, основанный на стохастическом градиентном спуске, дает заниженную оценку обоснованности 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 [43].

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

Глава 3

Оптимизация гиперпараметров в задаче выбора модели

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

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

В данной работе сравниваются градиентные методы оптимизации гиперпараметров. Основным достоинством подобных алгоритмов является их возможность одновременной оптимизации значительного числа гиперпараметров. В качестве базового алгоритма выступает алгоритм выбора гиперпараметров модели с использованием случайного поиска. В работах[74, 75, 52] в качестве целевой функции потерь рассматривается потеря на валидационной подвыборке с L_2 регуляризацией. В данной работе рассматривается общая задача оптимизации гиперпараметров. Рассматривающиеся алгоритмы и целевые функции потерь реализованы и представлены в качестве библиотеки для оптимизации гиперпараметров моделей [?]. Основным теоретическим вкладом данной главы является анализ рассматриваемых алгоритмов оптимизации гиперпараметров при использовании функции потерь общего вида, а также исследование качества и устойчивости итоговых моделей в случае использования кросс-валидации и вариационной оценки обоснованности. В экспериментальной части в качестве критерия выбора модели выступают вариационная нижняя оценка обоснованности модели и ошибка на валидационной части выборки. В отличие от [75], где также производится сравнение алгоритмов оптимизации гиперпараметров, в данной работе исследуется поведение алгоритмов на выборках большой мощности, таких как WISDM [?] и MNIST [?]. Численные эксперименты показывают, что при значительном количестве гиперпараметров, сопоставимым с количеством параметров модели, рассматриваемые алгоритмы предпочтительнее

стохастических.

3.1. Постановка задачи оптимизации гиперпараметров моделей

Пусть задана дифференцируемая по параметрам модель, приближающая зависимую переменную y :

$$\mathbf{f}(\mathbf{w}, \mathbf{x}) : \mathbb{W} \times \mathbb{X} \rightarrow \mathbb{Y}, \quad \mathbf{w} \in \mathbb{W}.$$

Как и в предыдущем разделе, будем полагать, что структура модели Γ для вероятностной модели глубокого обучения \mathbf{f} определена однозначно и метапараметры $\boldsymbol{\lambda}$ определены однозначно:

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

Пусть априорное распределение параметров имеет вид

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{A}^{-1}), \quad (3.1)$$

где $\mathbf{A}^{-1} = \text{diag}[\alpha_1, \dots, \alpha_u]^{-1}$ — матрица ковариаций диагонального вида.

Задача оптимизации гиперпараметров зависит как от критерия выбора модели, так и от метода оптимизации параметров модели. Проиллюстрируем задачу оптимизации гиперпараметров *двусвязанным байесовским выводом*.

Пример 5. Для дальнейшей формализации задачи в общем виде переобозначим

$$\boldsymbol{\theta} = \mathbf{w}, \quad \mathbf{h} = [\alpha_1, \dots, \alpha_u], \quad (3.2)$$

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

На *первом уровне* байесовского вывода производится оптимизация параметров (1.3) модели \mathbf{f} по заданной выборке \mathfrak{D} :

$$\boldsymbol{\theta}^* = \arg \max L(\boldsymbol{\theta} | \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}) = p(\mathbf{w} | \mathbf{X}, \mathbf{y}, \mathbf{h}) = \frac{p(\mathbf{y} | \mathbf{X}, \mathbf{w})p(\mathbf{w} | \mathbf{h})}{p(\mathbf{y} | \mathbf{X}, \mathbf{h})}. \quad (3.3)$$

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

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

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

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

$$p(\mathbf{y} | \mathbf{X}, \mathbf{h}) = \int_{\mathbf{w} \in \mathbb{R}^u} p(\mathbf{y} | \mathbf{X}, \mathbf{w})p(\mathbf{w} | \mathbf{h}) = Q(\mathbf{h} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\lambda}) \rightarrow \max_{\mathbf{h} \in \mathbb{H}}. \quad (3.4)$$

Как и в общей задаче (1.5), (3.11), требуется найти параметры $\boldsymbol{\theta}^*$ и гиперпараметры \mathbf{h}^* модели, доставляющие максимум следующей функции:

$$\mathbf{h}^* = \arg \max_{\mathbf{h} \in \mathbb{H}} Q(\mathbf{h} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\lambda}), \quad (3.5)$$

$$\boldsymbol{\theta}^*(\mathbf{h}) = \arg \max_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta} | \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}), \quad (3.6)$$

где L, Q — функции потерь и валидации (см. Опр. 16, 17).

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

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

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

Вариационным параметрам $\boldsymbol{\theta}$ модели \mathbf{f} соответствует вектор параметров модели:

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

Кросс-валидация. Разобьем выборку \mathfrak{D} случайно на K равных частей:

$$\mathfrak{D} = \mathfrak{D}_1 \sqcup \cdots \sqcup \mathfrak{D}_k, \mathfrak{D}_k = \{\mathbf{X}_k, \mathbf{y}_k\}, \quad k = 1, \dots, K.$$

Запустим K оптимизаций модели, каждую на своей части выборки. Положим $\boldsymbol{\theta} = [\mathbf{w}_1, \dots, \mathbf{w}_K]$, где $\mathbf{w}_1, \dots, \mathbf{w}_K$ — параметры модели при оптимизациях $1, \dots, K$.

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

$$L = \frac{1}{K} \sum_{k=1}^K \left(\frac{K}{K-1} \log p(\mathbf{y}_k | \mathbf{X}_k, \mathbf{w}_k) + \log p(\mathbf{w}_k | \mathbf{h}) \right). \quad (3.7)$$

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

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

где операция « $\mathbf{X} \setminus \mathbf{X}_k$ » определяется как взятие описаний всех объектов \mathbf{X} за исключением описаний объектов из \mathbf{X}_k .

Вариационная оценка обоснованности. Положим $L = Q$, равной вариационной оценке обоснованности модели:

$$\log p(\mathbf{y} | \mathbf{X}, \mathbf{h}) \geq -D_{KL}(q(\mathbf{w}) || p(\mathbf{w} | \mathbf{h})) + \int_{\mathbf{w}} q(\mathbf{w}) \log p(\mathbf{y} | \mathbf{X}, \mathbf{w}) d\mathbf{w} \approx \quad (3.8)$$

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

где q — нормальное распределение с диагональной матрицей ковариаций:

$$q \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}), \quad (3.9)$$

где $\mathbf{A}_q = \text{diag}[\alpha_1^q, \dots, \alpha_u^q]^{-1}$ — диагональная матрица ковариаций, $\boldsymbol{\mu}_q$ — вектор средних компонент, где u — общее количество параметров. Расстояние D_{KL} между двумя гауссовыми величинами задается 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 вариационных параметров $\boldsymbol{\theta}$ выступают параметры распределения q :

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

3.2. Градиентные методы оптимизации гиперпараметров

В данном разделе приводится описание рассматриваемых градиентных методов. Краткая характеристика и основные преимущества каждого из представленных методов отражены в Табл. 3.1, 3.2.

Алгоритм	Тип алгоритма	Преимущества алгоритма	Недостатки алгоритма
Случайный поиск	стохастический	простота реализации	Алгоритм неэффективен при большом количестве гиперпараметров (проклятие размерности)
Жадный алгоритм [52]	градиентный	Возможность одновременной оптимизации параметров и гиперпараметров	Жадность алгоритма
HOAG [75]	градиентный	Быстрая сходимость	Алгоритм требует сложных настройкам параметров
DrMAD [74]	градиентный	Алгоритм учитывает алгоритм оптимизации параметров модели и его параметры	Алгоритм страдает от проблем неустойчивости градиентного спуска (градиентный взрыв и затухание); Алгоритм работает в очень жестких предположениях.

Таблица 3.1. Преимущества и недостатки рассматриваемых алгоритмов

Алгоритм	Тип	Сложность итерации оптимизации	Предположения
Случайный поиск	стохастический	$O(\eta \Theta \cdot \hat{\mathcal{D}})$	-
Жадный алгоритм [52]	градиентный	$O(\eta \Theta \cdot \mathbb{H} \cdot \hat{\mathcal{D}})$	$\mathbf{H}(\boldsymbol{\theta}) = \mathbf{I}$
HOAG [75]	градиентный	$O(\eta \Theta \cdot \mathbb{H} \cdot \hat{\mathcal{D}} + o)$, где o — сложность решения системы линейных уравнений	первая производная Q и вторая производная L являются липшицевыми функциями $\det \mathbf{H} \neq 0$;
DrMAD [74]	градиентный	$O(\eta \Theta \cdot \mathbb{H} \cdot \hat{\mathcal{D}})$	Траектория оптимизации вариационных параметров $\boldsymbol{\theta} = \boldsymbol{\theta}^0, \dots, \boldsymbol{\theta}^\eta$ линейна

Таблица 3.2. Сложность и предположения для различных алгоритмов оптимизации гиперпараметров

Рассмотрим случай, когда оптимизация (3.6) параметров $\boldsymbol{\theta}$ производится с использованием градиентных методов. Пусть задан оператор стохастического градиентного спуска T (1.11), оптимизирующий вариационные параметры $\boldsymbol{\theta}$. Пусть оператор T производит η шагов оптимизации:

$$\boldsymbol{\theta}^* = T \circ T \circ \dots \circ T(\boldsymbol{\theta}^0 | L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}) = T^\eta(\boldsymbol{\theta}^0 | L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}), \quad (3.10)$$

где

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

λ_{lr} — длина шага градиентного спуска, $\boldsymbol{\theta}^0$ — начальное значение параметров $\boldsymbol{\theta}$, $\hat{\mathbf{y}}, \hat{\mathbf{X}}$ — случайная подвыборка заданной мощности выборки \mathcal{D} . В данной работе в качестве опреатора оптимизации параметров модели выступает стохастический градиентный спуск (1.11).

Перепишем задачу оптимизации (3.5), (3.6) в следующем виде

$$\mathbf{h}^* = \arg \max_{\mathbf{h} \in \mathbb{H}} Q(T^\eta(\boldsymbol{\theta}^0 | L, \mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})), \quad (3.11)$$

где $\boldsymbol{\theta}^0$ — начальное значение параметров $\boldsymbol{\theta}$. В дальнейшем для удобства будем применять сокращенные формы $Q(\mathbf{h} | \boldsymbol{\theta})$, $L(\boldsymbol{\theta} | \mathbf{h})$, $T(\boldsymbol{\theta}, \mathbf{h})$.

Оптимизационную задачу (3.11) предлагается решать с использованием градиентного спуска. Вычисление градиента от функции $Q(T^\eta(\boldsymbol{\theta}_0 | \mathbf{h}))$ по гиперпараметрам \mathbf{h} является вычислительно сложным в силу внутренней процедуры оптимизации $\boldsymbol{\theta}^* = T(\boldsymbol{\theta}_0, \mathbf{h})$. Общая схема оптимизации гиперпараметров представлена следующим образом:

1. От 1 до l :
2. Инициализировать параметры $\boldsymbol{\theta}$ при условии гиперпараметров \mathbf{h} .

3. Приближенно решить задачу оптимизации (3.11) и получить новый вектор параметров \mathbf{h}'
4. $\mathbf{h} = \mathbf{h}'$.

Здесь l — число итераций оптимизации гиперпараметров. Рассмотрим методы приближенного решения данной задачи оптимизации. Псевдокод общего алгоритма оптимизации гиперпараметров приведен на Рис. 3.1.

Жадный алгоритм. В качестве правила обновления вектора гиперпараметров \mathbf{h} на каждом шаге оптимизации (3.10) выступает градиентный спуск с учетом обновления параметров $\boldsymbol{\theta}$ на данном шаге:

$$\mathbf{h}' = \mathbf{h} - \lambda_{\text{lr}}^{\mathbf{h}} \nabla_{\mathbf{h}} (-Q(\mathbf{h}, T(\boldsymbol{\theta}^0, \mathbf{h}))) = \mathbf{h} - \lambda_{\text{lr}}^{\mathbf{h}} \nabla_{\mathbf{h}} (-Q(\mathbf{h}, \boldsymbol{\theta} - \lambda_{\text{lr}} \nabla (-L(\boldsymbol{\theta}|\mathbf{h})))), \quad (3.12)$$

где $\lambda_{\text{lr}}^{\mathbf{h}}$ — длина шага оптимизации гиперпараметров. Псевдокод жадного алгоритма оптимизации гиперпараметров приведен на Рис. 3.2.

Алгоритм HOAG. Предлагается получить приближенное значения градиента гиперпараметров $\nabla_{\mathbf{h}} Q(\mathbf{h}, T^{\eta}(\boldsymbol{\theta}^0))$ на основе следующей формулы:

$$\nabla_{\mathbf{h}} Q(T^{\eta}(\boldsymbol{\theta}^0, \mathbf{h})) = \nabla_{\mathbf{h}} Q(\mathbf{h}|\boldsymbol{\theta}) - (\nabla_{\boldsymbol{\theta}, \mathbf{h}}^2 (-L(\boldsymbol{\theta}|\mathbf{h})))^T \mathbf{H}(\boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\theta}} Q(\mathbf{h}|\boldsymbol{\theta}),$$

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

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

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

Итоговое правило обновления:

$$\mathbf{h}' = \mathbf{h} + \lambda_{\text{lr}}^{\mathbf{h}} \hat{\nabla}_{\mathbf{h}} Q. \quad (3.13)$$

В данной работе для приближенного решения шага 2 алгоритма HOAG используется стохастический градиентный спуск в силу сложности вычисления гессиана $\mathbf{H}(\boldsymbol{\theta})$. Псевдокод алгоритма HOAG приведен на Рис. 3.3.

Алгоритм DrMad. Для получения градиента от оптимизируемой функции Q как от функции от начальных параметров $\boldsymbol{\theta}^0$ предлагается пошагово восстановить η шагов оптимизации $T(\boldsymbol{\theta}^0)$ в обратном порядке аналогично методу обратного распространения ошибок. Для упрощения данной процедуры вводится предположение, что траектория изменения параметров $\boldsymbol{\theta}$ линейна:

$$\boldsymbol{\theta}^{\tau} = \boldsymbol{\theta}^0 + \frac{\tau}{\eta} (T(\boldsymbol{\theta}) - \boldsymbol{\theta}^0). \quad (3.14)$$

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

Вход: $\mathbf{X}, \mathbf{y}, p(\mathbf{w}|\mathbf{h}), \boldsymbol{\theta}^0$;

Вход: количество итераций l градиентной оптимизации гиперпараметров;

Вход: количество итераций η оптимизации вариационных параметров;

Выход: оптимальные значения параметров \mathbf{h} ;

- 1: **для** $r = 1, \dots, l$
- 2: провести оптимизацию вариационных параметров;
- 3: Приближенно решить задачу оптимизации (3.11) и получить новый вектор параметров \mathbf{h}'
- 4: $\mathbf{h} = \mathbf{h}'$;
- 5: **вернуть** \mathbf{h} .

Рис. 3.1. Псевдокод общего алгоритма оптимизации гиперпараметров.

1. Провести η шагов оптимизации: $\boldsymbol{\theta} = T(\boldsymbol{\theta}_0, \mathbf{h})$.
2. Положим $\hat{\nabla} \mathbf{h} = \nabla_{\mathbf{h}} Q(\mathbf{h}, \boldsymbol{\theta})$.
3. Положим $d\mathbf{v} = \mathbf{0}$.
4. Для $\tau = \eta \dots 1$ повторить:
5. Вычислить значения параметров $\boldsymbol{\theta}^\tau$ (3.14).
6. $d\mathbf{v} = \lambda_{lr} \hat{\nabla}_{\boldsymbol{\theta}}$.
7. $\hat{\nabla} \mathbf{h} = \hat{\nabla} \mathbf{h} - d\mathbf{v} \nabla_{\mathbf{h}} \nabla_{\boldsymbol{\theta}} Q$.
8. $\hat{\nabla} \boldsymbol{\theta} = \hat{\nabla} \boldsymbol{\theta} - d\mathbf{v} \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} Q$.

Итоговое правило обновления гиперпараметров аналогично (3.13). Псевдокод общего алгоритма оптимизации гиперпараметров приведен на Рис. 3.1.

В работе [74] отмечается неустойчивость алгоритма при высоких значениях длины шага градиентного спуска λ_{lr} . Поэтому вместо исходного правила (3.14) в данной работе первые 5% значений параметров не рассматриваются, а также учитывается только каждый τ_k шаг оптимизации:

$$\boldsymbol{\theta}^\tau = \boldsymbol{\theta}^{\tau_0} + \frac{\tau}{\eta} (T(\boldsymbol{\theta}) - \boldsymbol{\theta}^{\tau_0}), \quad \tau \in \{\tau_0, \dots, \eta\}, \tau \bmod \tau_k = 0, \quad (3.15)$$

где $\tau_0 = [0.05 \cdot \eta]$.

Иллюстративный пример поведения представленных методов представлен на Рис. 3.5. Жадный алгоритм, соответствующий красной линии на графике, оптимизирует гиперпараметры во время процедуры оптимизация вариационных параметров $\boldsymbol{\theta}$. Алгоритм HOAG оптимизирует гиперпараметры после каждой процедуры оптимизации вариационных параметров. Алгоритм DrMAD использует линеаризованную аппроксимацию траектории оптимизации вариационных параметров $\boldsymbol{\theta}$.

Вход: $\mathbf{X}, \mathbf{y}, p(\mathbf{w}|\mathbf{h}), \boldsymbol{\theta}^0$;

Вход: количество итераций l градиентной оптимизации гиперпараметров;

Вход: количество итераций η оптимизации вариационных параметров;

Выход: оптимальные значения параметров \mathbf{h} ;

- 1: **для** $r = 1, \dots, l$
- 2: **для** $\tau = 1, \dots, \eta$
- 3: проводить шаг оптимизации вариационных параметров;
- 4: обновить гиперпараметры (3.12);
- 5: $\mathbf{h} = \mathbf{h}'$.
- 6: **вернуть** \mathbf{h} .

Рис. 3.2. Псевдокод жадного алгоритма оптимизации гиперпараметров.

Вход: $\mathbf{X}, \mathbf{y}, p(\mathbf{w}|\mathbf{h}), \boldsymbol{\theta}^0$;

Вход: количество итераций l градиентной оптимизации гиперпараметров;

Вход: количество итераций η оптимизации вариационных параметров;

Выход: оптимальные значения параметров \mathbf{h} ;

- 1: **для** $r = 1, \dots, l$
- 2: проводить оптимизацию вариационных параметров;
- 3: Решить линейную систему для вектора $\boldsymbol{\lambda}$: $\mathbf{H}(\boldsymbol{\theta})\boldsymbol{\lambda} = \nabla_{\boldsymbol{\theta}}Q(\mathbf{h}, \boldsymbol{\theta})$.
- 4: $\hat{\nabla}_{\mathbf{h}}Q = \nabla_{\mathbf{h}}Q(\mathbf{h}|\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}, \mathbf{h}}L(\boldsymbol{\theta}|\mathbf{h})^T\boldsymbol{\lambda}$;
- 5: $\mathbf{h} = \mathbf{h} + \lambda_{lr}\hat{\nabla}_{\mathbf{h}}Q$.
- 6: **вернуть** \mathbf{h} .

Рис. 3.3. Псевдокод алгоритма НОАГ.

Вход: $\mathbf{X}, \mathbf{y}, p(\mathbf{w}|\mathbf{h}), \boldsymbol{\theta}^0$;

Вход: количество итераций l градиентной оптимизации гиперпараметров;

Вход: количество итераций η оптимизации вариационных параметров;

Выход: оптимальные значения параметров \mathbf{h} ;

- 1: **для** $r = 1, \dots, l$
- 2: $\boldsymbol{\theta} = T(\boldsymbol{\theta}_0, \mathbf{h})$.
- 3: $\hat{\nabla} \mathbf{h} = \nabla_{\mathbf{h}} Q(\mathbf{h}|\boldsymbol{\theta})$.
- 4: $d\mathbf{v} = \mathbf{0}$.
- 5: **для** $\tau = \eta, \dots, 1$
- 6: Вычислить значения параметров $\boldsymbol{\theta}^\tau$ (3.14).
- 7: $d\mathbf{v} = \lambda_{\text{lr}} \hat{\nabla}_{\boldsymbol{\theta}}$.
- 8: $\hat{\nabla} \mathbf{h} = \hat{\nabla} \mathbf{h} + d\mathbf{v} \nabla_{\mathbf{h}} \nabla_{\boldsymbol{\theta}} Q$.
- 9: $\hat{\nabla} \boldsymbol{\theta} = \hat{\nabla} \boldsymbol{\theta} + d\mathbf{v} \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} Q$.
- 10: $\mathbf{h} = \mathbf{h} - \hat{\nabla} \mathbf{h}$.
- 11: **вернуть** \mathbf{h} .

Рис. 3.4. Псевдокод алгоритма DrMAD.

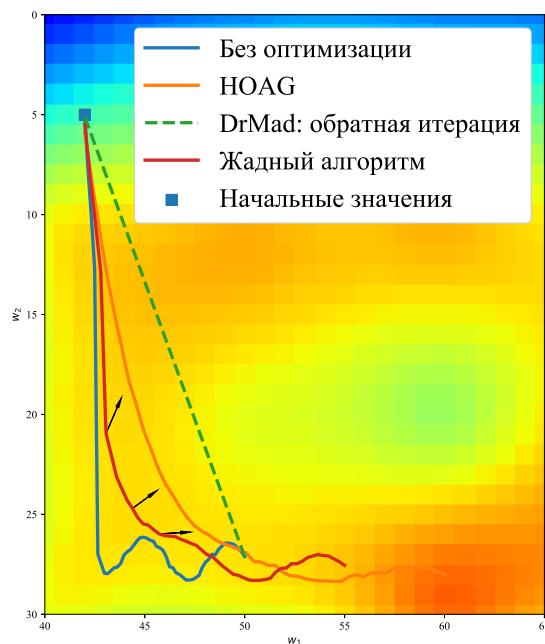


Рис. 3.5. Иллюстративный пример действия операторов оптимизации на гиперпараметры. Интенсивность цвета графика соответствует значения функции валидации Q .

3.3. Анализ алгоритмов оптимизации гиперпараметров

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

WISDM [?], а также на синтетических данных. Рассматривались задачи классификации и регрессии. В случае задач регрессии рассматриваемые модели \mathbf{f} возвращали скалярные значения:

$$\mathbb{Y} \subset R, \quad \mathbf{f}(\mathbf{w}, \mathbf{x}) = f(\mathbf{w}, \mathbf{x}).$$

Рассматривались следующие критерии качества:

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

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

где Q^0 – значение функции Q до начала оптимизации гиперпараметров.

3. Внешний критерий качества моделей E :

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

в случае задачи регрессии

$$E = \text{Accuracy} = 1 - \frac{1}{m} \sum_{i=1}^m [\arg \max_{r \in \{1, \dots, R\}} f^r(\mathbf{x}_i, \mathbf{w}) \neq y_i]$$

в случае задачи классификации.

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

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

$$E_\sigma = \text{Accuracy}_\sigma = 1 - \frac{1}{m} \sum_{i=1}^m [\arg \max_{r \in \{1, \dots, R\}} f^r(\mathbf{x}_i, \mathbf{w} + \boldsymbol{\varepsilon}) \neq y_i], \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma \mathbf{I}).$$

В качестве улучшаемого алгоритма рассматривался случайный поиск параметров с количеством итераций поиска, совпадающих с количеством итераций оптимизации гиперпараметров l : количество итераций $l = 50$ для синтетической выборки и выборки WISDM, $l = 25$ для выборки MNIST. В качестве функций Q и L рассматривались функции кросс-валидации (3.7) с $k = 4$ и вариационная оценка обоснованности (3.8).

На всех выборках гиперпараметры инициализировались случайно из равномерного распределения:

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

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

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

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

где $a_{\min} = -2.5, b_{\max} = 10.5$ для синтетической выборки и $a_{\min} = -5, b_{\max} = 11$ для выборок WISDM и MNIST. Калибровка значения λ_{lr} проводилась на небольшом количестве итераций оптимизаций гиперпараметров l : $l = 50$ для синтетической выборки, $l = 10$ для выборки WISDM $l = 5$ для выборки MNIST. В случае, если алгоритмы показывали неустойчивую работу непосредственно во время запуска эксперимента (взрыв градиента или численное переполнение), то длина шага $\lambda_{\text{lr}}^{\mathbf{h}}$ понижалась. Для алгоритма DrMad параметр τ_k , отвечающий за количество рассматриваемых шагов оптимизации был установлен как $\tau_k = 1$ для синтетической выборки и выборки WISDM, $\tau_k = 10$ для выборки MNIST.

Синтетическая выборка. Синтетические данные были порождены из выборки с одним признаком, $\mathbf{X} \in \mathbb{R}^{m \times 1}$. Порождения выборки происходило по следующему правилу:

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

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

Было проведено 5 запусков для каждого алгоритма. Графики итоговых полиномов представлены на Рис. 3.6. Как видно из графиков, с использованием вариационной оценки удалось получить полиномы, близкие к линейным моделям. Подобные модели показывают наилучшее значение обоснованности в силу слабого переобучения и хорошего качества на тестовой выборке.

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

Рассматривалась нейросеть с 10 нейронами на скрытом слое:

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

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

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

Графики сходимости алгоритмов, а также качества полученных моделей представлены на Рис. 3.7. Как видно из графиков, градиентные алгоритмы DrMad и HOAG показывают значительно худший результат по сравнению с жадным алгоритмом оптимизации. Случайный поиск показывает достаточно хорошие результаты в случае небольшого числа оптимизируемых гиперпараметров \mathbf{h} . В случае, когда в качестве функции Q используется вариационная нижняя оценка обоснованности (3.8) и количество гиперпараметров велико, эффективно работающими алгоритмами оказалась жадная оптимизация и HOAG. HOAG имеет большее время сходимости и требует более сложных вычислений в процессе оптимизации.

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

Графики сходимости алгоритмов, а также качества полученных моделей представлены на Рис. 3.8. Как видно из графиков, модели, достигающие наилучшей оценки обоснованности, имеют наихудшее итоговое качество, но более устойчивы к возмущению параметров модели. Для дополнительного анализа данной проблемы были проведены эксперименты по оптимизации моделей на выборке с добавленным шумом и использованием значений гиперпараметров \mathbf{h} , полученных ранее:

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

где $\hat{\sigma}$ варьировалась в отрезке от 0 до 0.5. График зависимости качества моделей от значения $\hat{\sigma}$ приведен на Рис. 3.9. Гиперпараметры, достигающие наибольших значений вариационной оценки (3.8) менее подвержены шуму в обучающей выборке, что можно интерпретировать как меньшую подверженность к переобучению.

Как можно видеть по результатам экспериментов, градиентные методы показывают лучший результат, чем случай поиск в случае большого количества гиперпараметров. Наилучшие результаты были получены жадным поиском. Алгоритм DrMad, показавший результаты хуже, чем жадный алгоритм и HOAG, является упрощенной версией алгоритма, представленного в [74]. Данный алгоритм позволяет проводить оптимизацию не только гиперпараметров, но параметров алгоритма оптимизации T . Поэтому возможным развитием метода DrMad является получение оптимальных значений параметров оптимизации.

Алгоритм	L, Q	$Q(\mathbf{h} \boldsymbol{\theta})$	Convergence	E	$E_{0.25}$	$E_{0.5}$
<i>Синтетическая выборка</i>						
Случайный поиск	(3.7)	-171.6	26.2 \pm 20.0	1.367	1.410	1.555
Greedy	(3.7)	-172.5	30.0 ± 24.5	1.421	1.439	1.536
DrMAD	(3.7)	-174.1	40.2 ± 16.1	1.403	1.424	1.512
HOAG	(3.7)	-174.7	29.4 ± 24.0	1.432	1.463	1.553
Случайный поиск	(3.8)	-63.5	32.4 ± 18.7	1.368	1.426	1.546
Greedy	(3.8)	-25.5	1.2 \pm 0.4	1.161	1.174	1.193
DrMAD	(3.8)	-25.1	10.6 ± 0.8	1.157	1.163	1.184
HOAG	(3.8)	-25.8	10.8 ± 1.5	1.141	1.149	1.177
<i>WISDM</i>						
Случайный поиск	(3.7)	-1086661.1	22.0 ± 19.3	0.660	0.670	0.690
Greedy	(3.7)	-1086707.1	15.4 \pm 17.2	0.707	0.723	0.769
DrMAD	(3.7)	-1086708.2	29.2 ± 8.0	0.694	0.708	0.742
HOAG	(3.7)	-1086733.5	28.2 ± 7.13	0.701	0.724	0.753
Random search	(3.8)	-35420.4	14.4 ± 7.8	0.732	0.755	0.785
Greedy	(3.8)	-3552.9	1.0 \pm 0.0	0.702	0.730	0.767
DrMAD	(3.8)	-26091.4	50.0 ± 0.0	0.729	0.753	0.816
HOAG	(3.8)	-16566.6	49.0 ± 0.0	0.733	0.755	0.801
<i>MNIST</i>						
Random search	(3.7)	-3236.4	7.8 ± 1.9	0.981	0.966	0.866
Greedy	(3.7)	-3416.7	10.8 ± 10.4	0.979	0.962	0.860
DrMAD	(3.7)	-3469.0	17.0 ± 5.6	0.982	0.962	0.831
HOAG	(3.7)	-3748.6	8.6 \pm 7.3	0.980	0.961	0.853
Random search	(3.8)	-1304556.4	14.2 ± 5.7	0.982	0.943	0.814
Greedy	(3.8)	-11136.2	1.0 \pm 0.0	0.977	0.952	0.884
DrMAD	(3.8)	-1305432.9	24.6 ± 0.5	0.982	0.941	0.813
HOAG	(3.8)	-280061.6	24.0 ± 0.0	0.981	0.943	0.819

Таблица 3.3. Результаты эксперимента по оптимизации гиперпараметров.

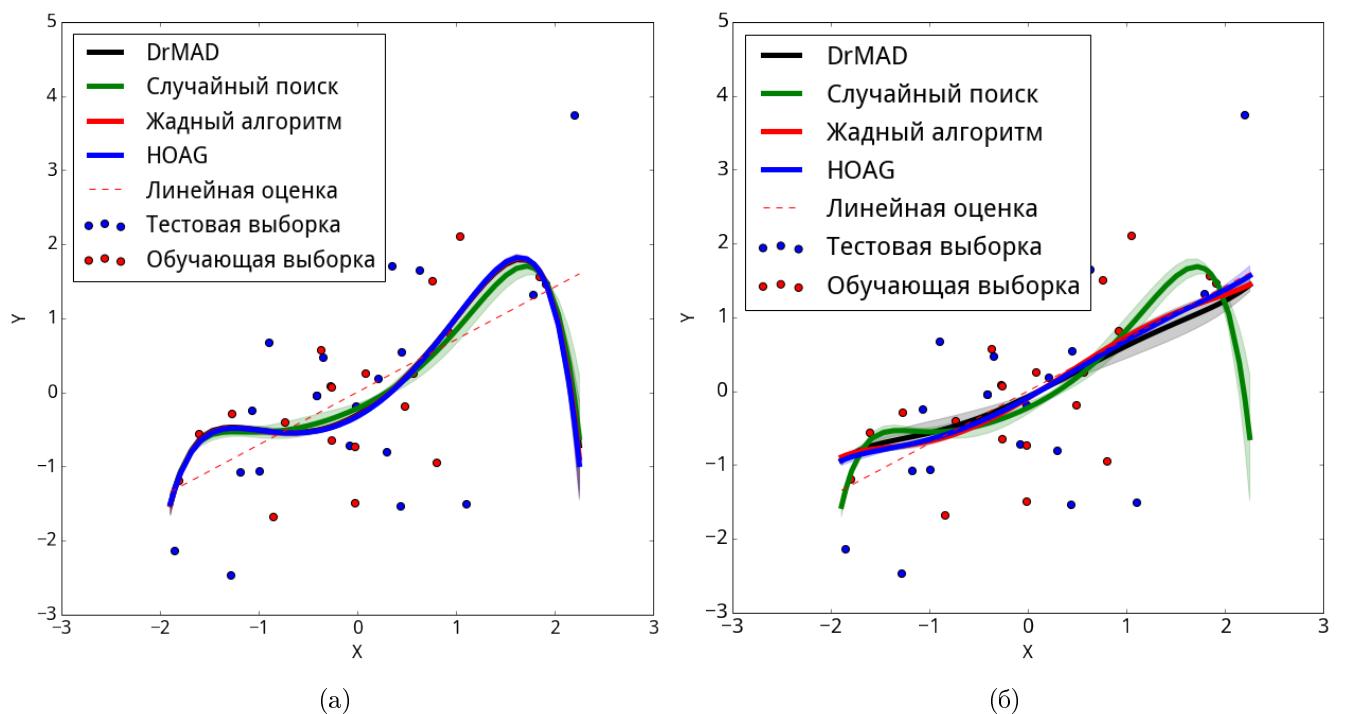


Рис. 3.6. Итоговые модели для синтетической выборки: а) с использованием кросс-валидации, б) с использованием вариационной оценки обоснованности модели.

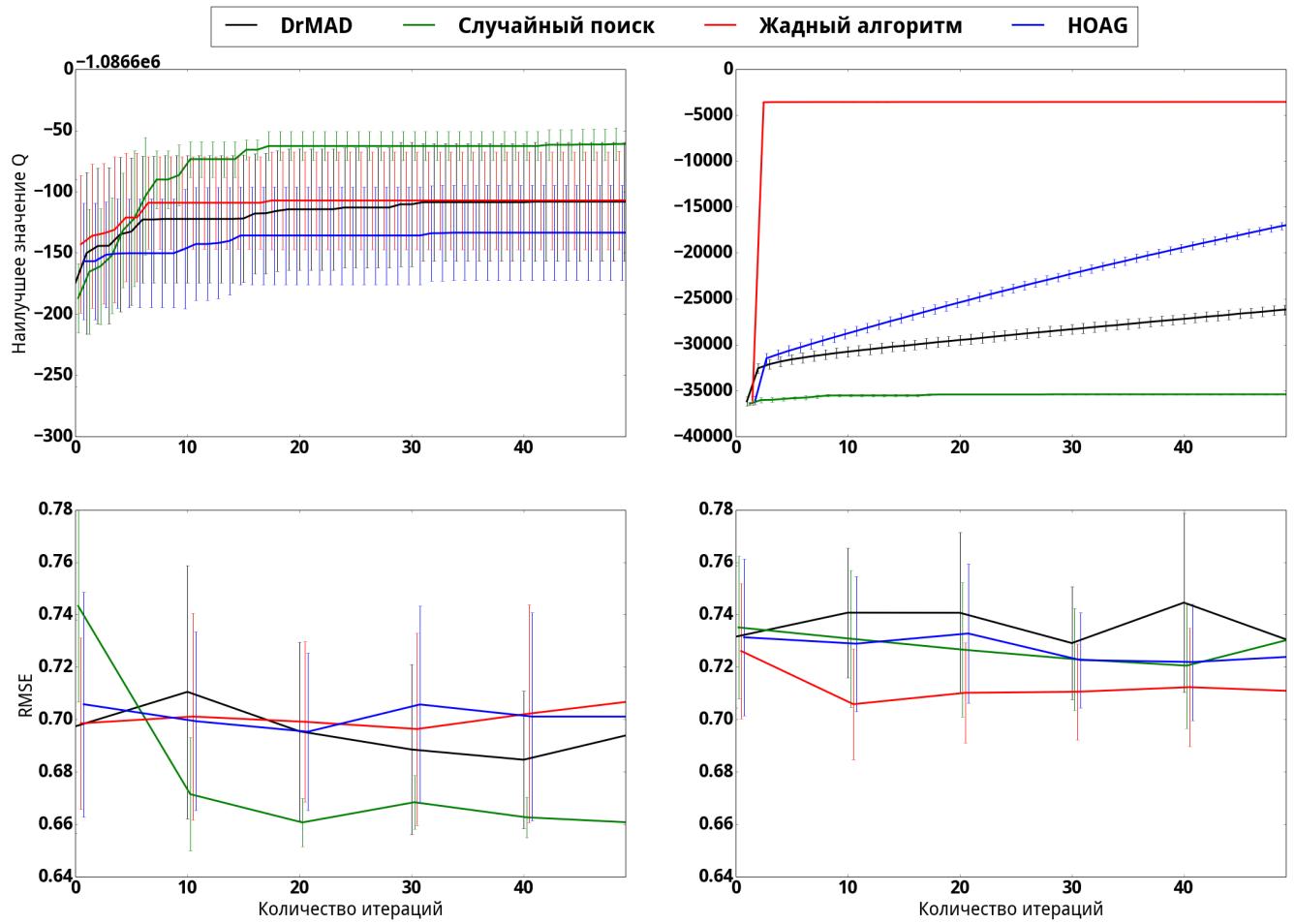


Рис. 3.7. WISDM, наилучшее значение функции Q и RMSE для кросс-валидации (слева) и вариационной оценки обоснованности модели (справа).

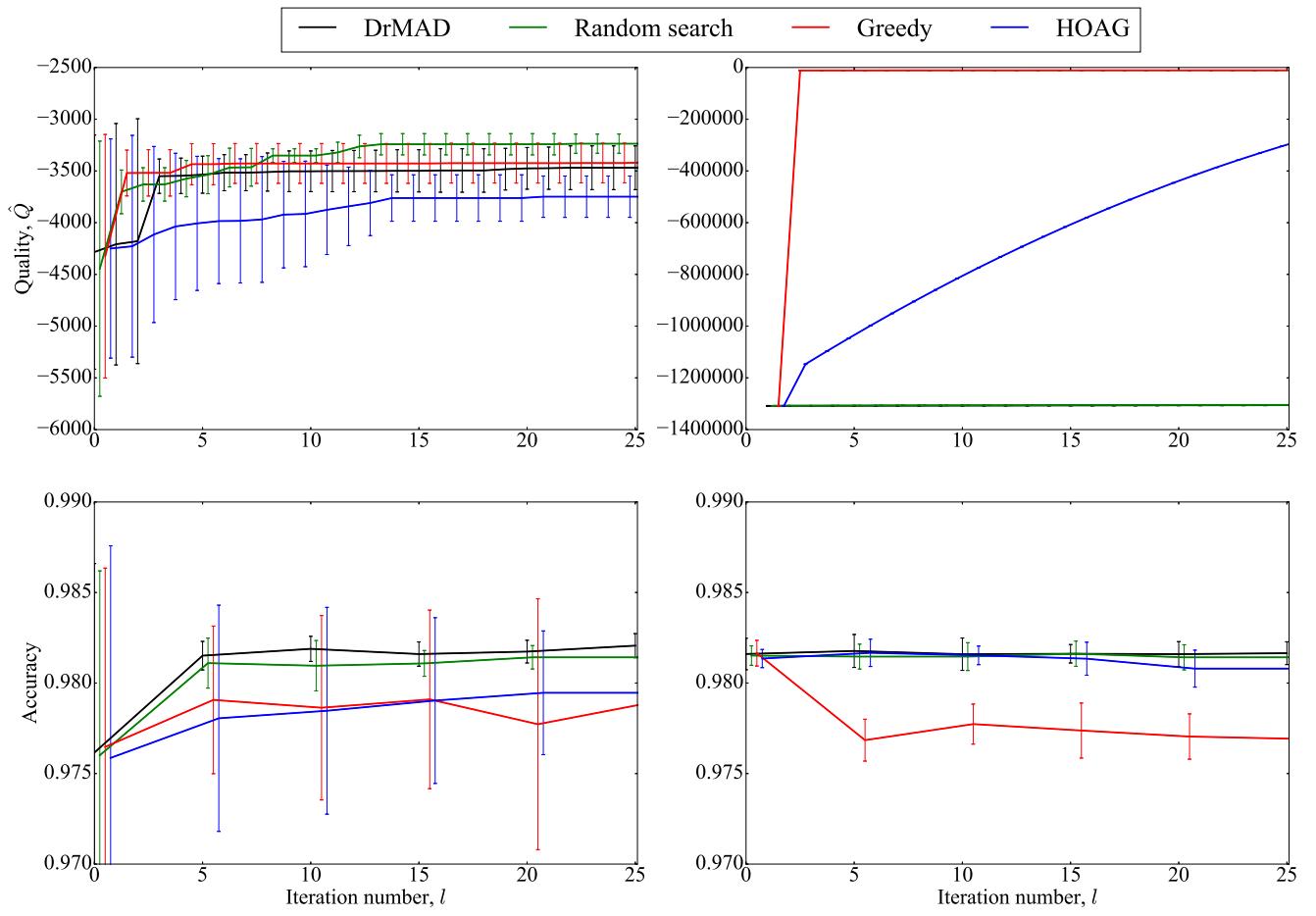


Рис. 3.8. MNIST, наилучшее значение функции Q и RMSE для кросс-валидации (слева) и вариационной оценки обоснованности модели (справа).

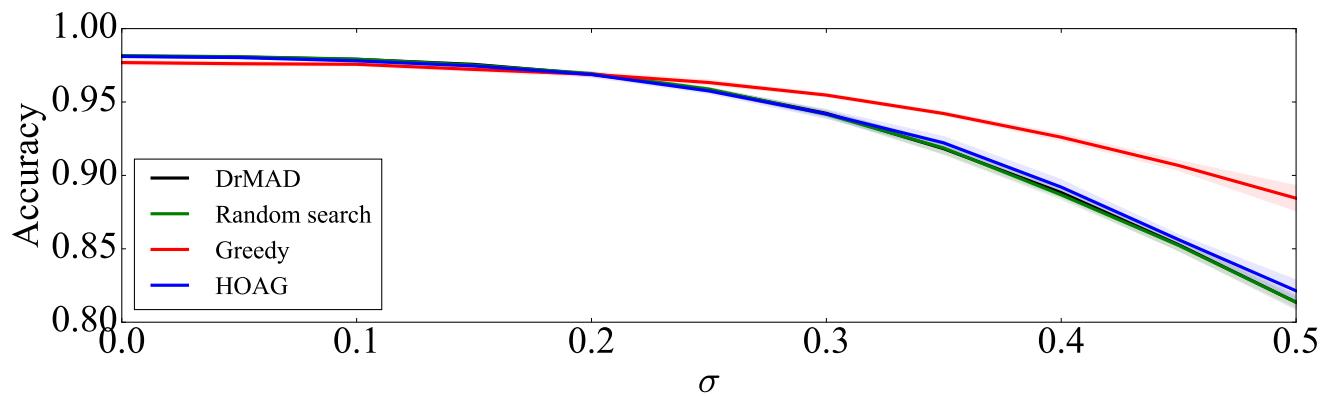


Рис. 3.9. MNIST, точность классификации на тестовой выборке при добавлении шума в обучающую выборку. Гиперпараметры были оптимизированы с использованием вариационной оценки обоснованности модели.

Глава 4

Анализ прикладных задач порождения и выбора моделей глубокого обучения

В данной главе анализируются свойства предложенных моделей и рекомендации по их использованию. Качество моделей, полученных с использованием предложенных методов сравнивается с качеством известных методов.

4.1. Выбор модели классификации временных рядов

В данном разделе рассматривается задача построения сети глубокого обучения для классификации временных рядов. Под временным рядом понимается реализация некоторого случайного процесса. Работы [?, ?, ?] посвящены классификации временных рядов с использованием методов глубокого обучения. В работе [?] для классификации временных рядов используются рекуррентные нейронные сети. В работе [?] рассматриваются различные суперпозиции ограниченной машины Больцмана, автокодировщика и двуслойной нейронной сети. Исследуется суперпозиция, состоящая из ограниченной машины Больцмана, автокодировщика и двуслойной нейронной сети [22]. Работа [?] посвящена рекуррентной модификации модели ограниченной машины Больцмана для классификации временных рядов.

В данном разделе решается прикладная задача классификации временных рядов. В качестве данных для вычислительного эксперимента используются данные с акселерометров мобильных телефонов [?]. Для решения задачи оптимизации используется алгоритм обратного распространения ошибок с послойным предобучением сети и дальнейшей настройкой параметров всех слоев [?].

Постановка задачи. Рассматривается задача классификации. Моделью классификации \mathbf{f} выступает суперпозиция подмоделей, аналогичная (2.6):

$$\mathbf{f}(\mathbf{W}, \mathbf{x}) = \mathbf{f}_1(\mathbf{f}_2(\dots \mathbf{f}_{|V|}(\mathbf{x}))) : \mathbb{R}^n \rightarrow [0, 1]^R, \quad (4.1)$$

где $\mathbf{f}_v, v \in \{1, \dots, |V|\}$ — модели, параметрическое семейство вектор-функций; \mathbf{W} — вектор параметров моделей; c -ю компоненту $\mathbf{f}(\mathbf{x}, \mathbf{W})[c]$ вектора $\mathbf{f}(\mathbf{x}, \mathbf{W})$ будем интерпретировать как вероятность отнесения объекта \mathbf{x}_i к классу с меткой c (2.7).

Требуется минимизировать функцию ошибки L на обучающей выборке \mathfrak{D} , где L — сумма отрицательных логарифмов правдоподобия по всем объектам выборки

$$\mathbf{W}^* = \arg \min_{\mathbf{W}} L(\mathbf{W} | \mathfrak{D}),$$

где

$$L(\mathbf{W} | \mathfrak{D}) = - \sum_{i=1}^m \sum_{=1}^R [y_i =] \log p(y_i = | \mathbf{x}_i, \mathbf{W}).$$

Структура сети глубокого обучения. Для решения задачи предлагаются использовать суперпозицию, состоящую из трех компонент: ограниченной машины Больцмана, автокодировщика и двуслойной нейросети с softmax-классификатором. Модель (4.1) в данном случае выглядит следующим образом:

$$\mathbf{f}(\mathbf{W}, \mathbf{x}) = \mathbf{f}_{\text{SM}} \circ \mathbf{f}_{\text{AE}} \circ \mathbf{f}_{\text{RBM}}(\mathbf{x}),$$

где $\mathbf{f}_{\text{RBM}}, \mathbf{f}_{\text{AE}}, \mathbf{f}_{\text{SM}}$ — модели ограниченной машины Больцмана, автокодировщика и двуслойной нейронной сети соответственно.

Ограниченнная машина Больцмана. Ограниченнная машина Больцмана представляет собой двудольный граф, где первая доля соответствует объекту \mathbf{x} , а вторая доля — бинарному вектору \mathbf{h} длины n' . Рассмотрим случай, когда вектор \mathbf{x} принимает бинарные значения. Определим смещенную оценку логарифма совместного распределения объекта \mathbf{x} и скрытого вектора \mathbf{h} следующим образом:

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{x}^T \cdot \mathbf{b}_{\text{vis}} - \mathbf{h}^T \mathbf{b}_{\text{hid}} - \mathbf{h}^T \mathbf{W}_{\text{RBM}} \mathbf{x}, \quad (4.2)$$

где $\mathbf{b}_{\text{vis}}, \mathbf{b}_{\text{hid}}, \mathbf{W}_{\text{RBM}}$ — параметры модели.

Пусть совместное распределение пары векторов \mathbf{x}, \mathbf{h} задано следующим образом:

$$p(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{x}, \mathbf{h})),$$

где Z — нормировочный коэффициент:

$$Z = \sum_{\mathbf{x} \in \{0,1\}^n} \sum_{\mathbf{h} \in \{0,1\}^{n'}} \exp(-E(\mathbf{x}, \mathbf{h})).$$

Функция вероятности вектора \mathbf{x} есть сумма вероятностей по всем скрытым состояниям вектора \mathbf{h} :

$$p(\mathbf{x}) = \sum_{\mathbf{h} \in \{0,1\}^{n'}} p(\mathbf{x}, \mathbf{h}).$$

Определим элемент суперпозиции (4.1):

$$\mathbf{f}_{\text{RBM}}(\mathbf{x}) = \mathbb{E}(\mathbf{h}|\mathbf{x}). \quad (4.3)$$

Параметры модели (4.3) оптимизируются следующим образом:

$$\mathbf{W}_{\text{RBM}}^*, \mathbf{b}_{\text{vis}}^*, \mathbf{b}_{\text{hid}}^* = \arg \max_{\mathbf{W}_{\text{RBM}}, \mathbf{b}_{\text{vis}}, \mathbf{b}_{\text{hid}}} p(\mathbf{X}, [\mathbf{W}_{\text{RBM}}, \mathbf{b}_{\text{vis}}, \mathbf{b}_{\text{hid}}]) = \prod_{i=1}^m \sum_{\mathbf{h} \in \{0,1\}^{n'}} \frac{1}{Z} \exp(-E(\mathbf{x}_i, \mathbf{h})) \quad (4.4)$$

В данной работе используется модифицированная версия ограниченной машины Больцмана, позволяющая работать с небинарными входными данными [?]. В этой модификации функция E (4.2) задается следующим образом:

$$E(\mathbf{x}, \mathbf{h}) = \frac{(\mathbf{x} - \mathbf{b}_{\text{vis}})^2}{2\hat{\sigma}^2} - \mathbf{h}^T \cdot \mathbf{b}_{\text{hid}} - \frac{\mathbf{h}^T}{\hat{\sigma}} \mathbf{W}_{\text{RBM}} \mathbf{x},$$

где $\hat{\sigma}$ — эмпирическая оценка среднеквадратичного отклонения по выборке \mathbf{X} , деление производится покомпонентно. Для решения задачи оптимизации (4.4) используется алгоритм, описанный в [3].

Автокодировщик. Автокодировщик предназначен для снижения размерности исходного пространства признаков. Автокодировщик представляет собой суперпозицию кодирующего и декодирующего блока:

$$\mathbf{f}'_{AE} = \mathbf{f}_{enc}(\mathbf{f}_{dec}(\mathbf{x})),$$

где

$$\mathbf{f}_{enc}(\mathbf{x}) = \boldsymbol{\sigma}(\mathbf{W}_{enc}\mathbf{x} + \mathbf{b}_{enc}) \text{ — кодирующий блок,}$$

$$\mathbf{f}_{dec} \circ \mathbf{f}_{enc}(\mathbf{x}) = \boldsymbol{\sigma}(\mathbf{W}_{dec}\mathbf{f}_{enc}(\mathbf{x}) + \mathbf{b}_{dec}) \text{ — декодирующий блок,}$$

$$\boldsymbol{\sigma}(\mathbf{x}) = (1 + \exp(-\mathbf{x}))^{-1} \text{ — сигмоидная функция,}$$

$\mathbf{W}_{enc}, \mathbf{W}_{dec}, \mathbf{b}_{enc}, \mathbf{b}_{dec}$ — параметры модели.

Введем дополнительное ограничение на матрицы $\mathbf{W}_{enc}, \mathbf{W}_{dec}$:

$$\mathbf{W}_{enc} = \mathbf{W}_{dec}^T.$$

Параметры $\mathbf{W}_{enc}, \mathbf{W}_{dec}, \mathbf{b}_{enc}, \mathbf{b}_{dec}$ оптимизируются так, чтобы по вектору \mathbf{x} получить восстановленный вектор \mathbf{f}'_{AE} , близкий к исходному \mathbf{x} :

$$\mathbf{W}_{enc}^*, \mathbf{W}_{dec}^*, \mathbf{b}_{enc}^*, \mathbf{b}_{dec}^* = \arg \min_{\mathbf{W}_{enc}, \mathbf{W}_{dec}, \mathbf{b}_{enc}, \mathbf{b}_{dec}} \frac{1}{m} \sum_{i=1}^m \|\mathbf{f}'_{AE}(\mathbf{x}_i) - \mathbf{x}_i\|_2^2. \quad (4.5)$$

Декодирующий блок \mathbf{f}_{dec} требуется только для решения задачи оптимизации (4.5) и не используется в суперпозиции (4.1). Таким образом, элемент суперпозиции (4.1) определен как

$$\mathbf{f}_{AE} = \mathbf{f}_{enc}(\mathbf{x}).$$

Двухслойная нейросеть. Двухслойная сеть представляет собой логистическую вектор-функцию:

$$\mathbf{f}_{hid}(\mathbf{x}) = \mathbf{W}_2^T \tanh(\mathbf{W}_1^T \mathbf{x}), \quad (4.6)$$

$$\mathbf{f}_{SM}(\mathbf{x}) = \frac{\exp(\mathbf{f}_{hid}(\mathbf{x}))}{\|\exp(\mathbf{f}_{hid}(\mathbf{x}))\|_1},$$

где c -я компонента $f_{SM}(\mathbf{x})[c]$ вектора $\mathbf{f}_{SM}(\mathbf{x})$ интерпретируется как вероятность принадлежности объекта \mathbf{x} классу c . Итоговая функция классификации (4.1) ставит в соответствие объекту \mathbf{x} метку класса y , где y — класс, к которому принадлежит \mathbf{x} с наибольшей вероятностью:

$$f(\mathbf{W}, \mathbf{x})[c] = \begin{cases} 1, & \text{если } c = \arg \max_{c'} f_{SM}(\mathbf{f}_{AE}(\mathbf{f}_{RBM}(\mathbf{x}))[c'], \\ 0 & \text{иначе.} \end{cases}$$

Здесь \mathbf{f}_{AE} , \mathbf{f}_{RBM} — автокодировщик (4.5) и ограниченная машина Больцмана (4.4) соответственно, $f_{\text{SM}}(\mathbf{x})[c]$ — c -я компонента вектора \mathbf{f}_{SM} , $f(\mathbf{W}, \mathbf{x})[c]$ — c -я компонента вектор-функции \mathbf{f} .

Итоговая задача оптимизации выглядит следующим образом:

$$\boldsymbol{\theta}^* = \arg \min \sum_{i=1}^m \sum_{c=1}^R [y_i = c] \log(f_{\text{SM}}(\mathbf{f}_{\text{AE}}(\mathbf{f}_{\text{RBM}}(\mathbf{x}_i)))[c]),$$

где $\boldsymbol{\theta}^* = [\mathbf{W}_{\text{RBM}}^*, \mathbf{b}_{\text{vis}}^*, \mathbf{b}_{\text{hid}}^*, \mathbf{W}_{\text{enc}}^*, \mathbf{W}_{\text{dec}}^*, \mathbf{b}_{\text{enc}}^*, \mathbf{b}_{\text{dec}}^*, \mathbf{W}_2^*, \mathbf{W}_1^*]$ — параметры ограниченной машины Больцмана (4.4), автокодировщика (4.5) и двуслойной сети (4.6).

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

Основные эксперименты — исследование зависимости ошибки классификации от числа параметров и размера выборки — были проведены как с использованием инструментария на базе библиотеки Theano, так и с использованием инструментария на языке Matlab. Для оценки качества классификации была проведена процедура скользящего контроля [44] при соотношении числа объектов обучающей и контрольной выборки 3:1. Число нейронов на каждом слое задавалось из соотношения 10:6:3. При проведении процедуры скользящего контроля для каждого отсчета количества нейронов было произведено пять запусков. В эксперименте с использованием инструментария на базе Theano при обучении двуслойной нейронной сети проводился мультистарт [?], т. е. одновременный запуск обучения сети с восемью разными стартовыми значениями параметров для предотвращения возможного застревания алгоритма обучения в локальном минимуме. При оценке качества классификации выбиралась модель с наилучшими результатами. График зависимости ошибки классификации от числа используемых нейронов изображен на рис. 4.1.

Для оценки зависимости качества классификации от размера обучающей выборки была проведена кросс-валидация с фиксированным количеством объектов в обучающей выборке (25% исходной выборки) и переменным размером обучающей выборки. Число нейронов было установлено как 364:224:112. При проведении процедуры скользящего контроля для каждого отсчета было произведено пять запусков. График зависимости ошибки классификации от размера обучающей выборки представлен на рис. 4.2.

Для исследования скорости оптимизации нейросети в зависимости от кон-

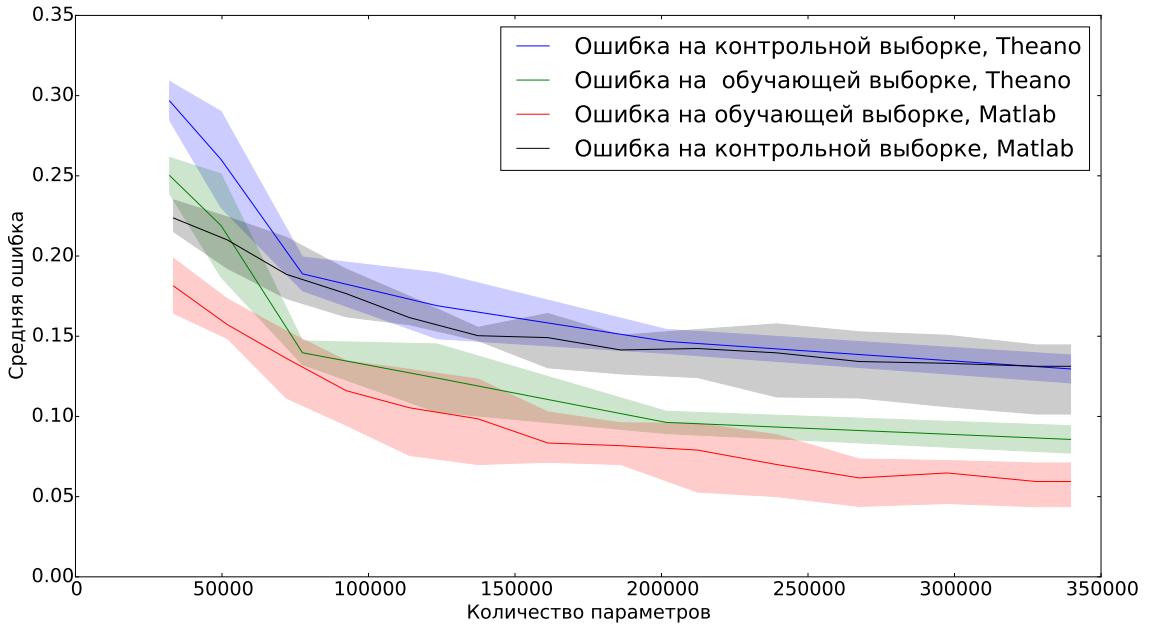


Рис. 4.1. Зависимость ошибки от числа нейронов

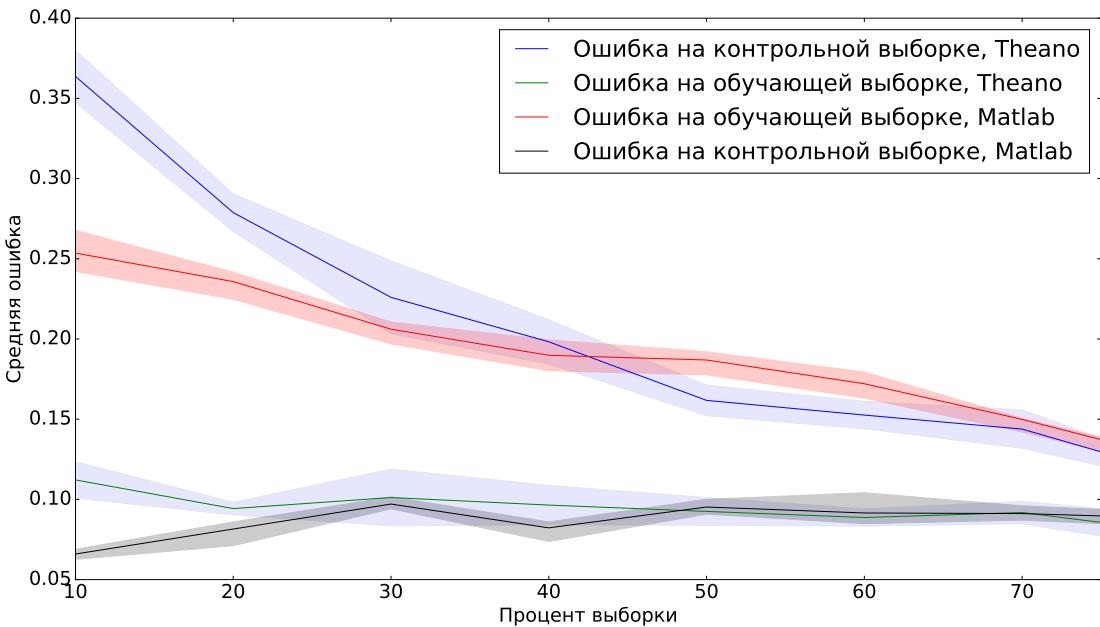


Рис. 4.2. Зависимость ошибки от размера обучающей выборки

фигурации Theano был сделан следующий эксперимент: проводилось обучение двуслойной нейросети на основе подсчитанных заранее параметров ограниченной машины Больцмана (4.4) и автокодировщика (4.5). Обучение проходило за 100 итераций. При обучении алгоритм запускался параллельно с r разными стартовыми позициями, $r \in \{1, \dots, 4\}$. Число нейронов было установлено как

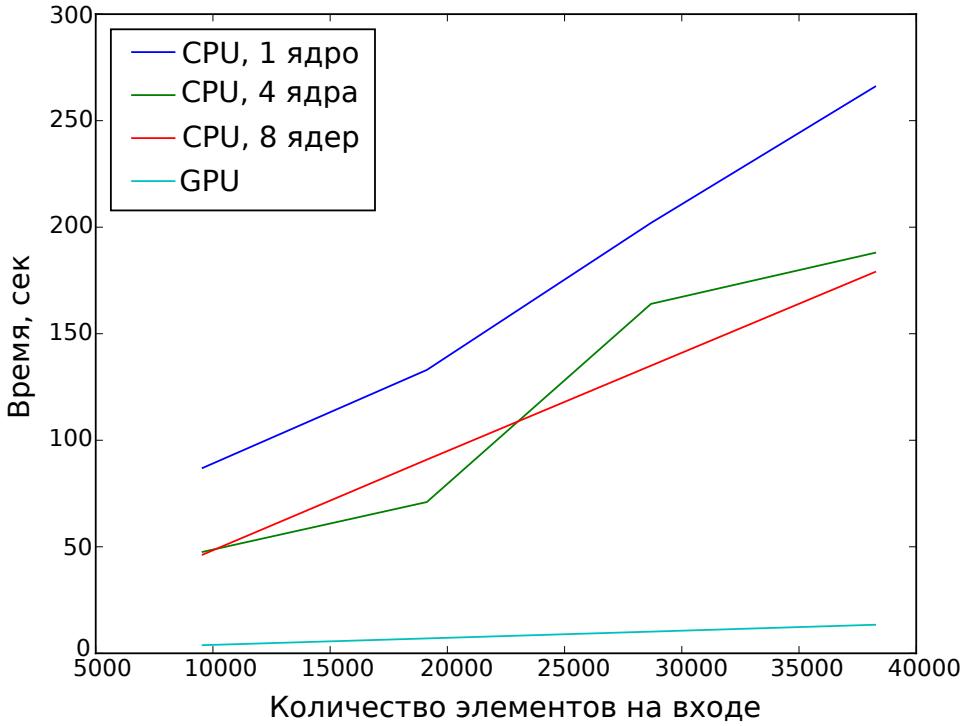


Рис. 4.3. Результаты эксперимента по исследованию скорости процесса обучения

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

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

Результаты эксперимента приведены на рис. 4.3. Как видно из графика, вычисление с использованием CUDA показывает значительное ускорение по сравнению с вычислением на центральном процессоре.

4.2. Выбор модели обнаружения перефраза в тексте

В данном разделе решается задача выбора оптимальной нейросетевой модели из класса рекуррентных нейронных сетей. Рекуррентной нейросетью называется нейросеть со связью между нейронами одного слоя. В качестве критерия оптимальности используется нижняя оценка правдоподобия модели.

Для построения модели рекуррентной сети рассматривается модель из [?], решающая задачу определения сходства предложений. Модель принимает на вход векторизованные представления слов. Векторизация выполняется с помощью алгоритма GloVe [?], основанного на факторизации матрицы слов-контекстов и использовании весовой функции для уменьшения значимости редких слов. Альтернативой этому алгоритму выступает линейная модель

Word2vec, комбинирующая в себе Continuous Bag-of-Words, skip-gram, negative sampling [?]. Несмотря на разные подходы к проблеме, GloVe и Word2vec оптимизируют схожие функционалы. Упрощенной линейной моделью Word2vec, предназначенней для классификации документов, является fastText — метод, работающий на символьных n -граммах [?].

Для решения задач, связанных с обработкой естественного языка применяются модели рекуррентных сетей [?]. Обобщением рекуррентных моделей являются рекурсивные модели автокодировщиков, агрегирующие информацию от входного текста не рекуррентно, а по дереву синтаксического разбора.

Предлагается подход, основанный на получении вариационной нижней оценки правдоподобия модели. Предлагаемый подход сравнивается с методом удаления параметров Optimal Brain Damage, базирующимся на анализе функции ошибки (1.8).

Вычислительный эксперимент проводится на выборке размеченных пар предложений SemEval 2015. Для каждой пары предложений из корпуса дана экспертная оценка их семантической близости. Требуется построить модель, оценивающую семантическую близость двух предложений. Проблема рассматривается как задача многоклассовой классификации, аналогично [?]. Критерием качества служит F_1 -мера, учитывая как полноту, так и точность предсказаний. В качестве базовой модели рассматривается пара соединенных рекуррентных сетей с общим вектором параметров и softmax-классификатором на выходе.

Постановка задачи. Для построения выборки используем набор пар предложений SemEval 2015 [?]. Каждому слову сопоставим вектор размерности n . Обозначим через l число слов в самом длинном предложении. Предложения длины, меньше l , дополним нулевыми векторами. Построим выборку

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

где $\mathbf{x}_i = [\mathbf{x}_i^1, \mathbf{x}_i^2]$ — пары последовательностей векторов слов, соответствующих i -й паре предложений, $\mathbf{x}_i^1, \mathbf{x}_i^2 \in \mathbb{R}^{n \times l}$; $y_i \in \mathbb{Y} = \{0, \dots, R\}$ — экспертная оценка семантической близости.

Требуется построить модель $f(\mathbf{w}) : \mathbb{R}^{n \times l} \times \mathbb{R}^{n \times l} \rightarrow \mathbb{Y}$, сопоставляющую паре предложений \mathbf{x}_i^1 и \mathbf{x}_i^2 класс семантической близости, где $\mathbf{w} \in \mathbb{W} \subseteq \mathbb{R}^s$ — пространство параметров модели. Искомая модель выбирается из множества M рекуррентных нейронных сетей с функцией активации \tanh . Модель

$$f(\mathbf{w}) : \mathbb{R}^{n \times l} \times \mathbb{R}^{n \times l} \rightarrow \mathbb{Y}$$

принадлежит искомому множеству моделей M , если существуют такие матрицы перехода $\mathbf{W}_1 \in \mathbb{R}^{n \times m}$, $\mathbf{W}_2 \in \mathbb{R}^{n \times n}$, $\mathbf{W}_3 \in \mathbb{R}^{(Z \times 2n)}$ и вектор смещения $\mathbf{b} \in \mathbb{R}^n$, что для j -х элементов $\mathbf{x}_{ij}^1, \mathbf{x}_{ij}^2 \in \mathbb{R}^m$ последовательностей \mathbf{x}_i^1 и \mathbf{x}_i^2 определены

векторы скрытого слоя $\mathbf{h}_{ij}^1, \mathbf{h}_{ij}^2 \in \mathbb{R}^n$:

$$\mathbf{h}_{ij}^1 = \tanh(\mathbf{W}_1 \mathbf{x}_{ij}^1 + \mathbf{W}_2 \mathbf{h}_{i,j-1}^1 + \mathbf{b}), \quad (4.7)$$

$$\mathbf{h}_{ij}^2 = \tanh(\mathbf{W}_1 \mathbf{x}_{ij}^2 + \mathbf{W}_2 \mathbf{h}_{i,j-1}^2 + \mathbf{b}). \quad (4.8)$$

Для определения класса семантической близости используются последние значения скрытого слоя \mathbf{h}_{il}^1 и \mathbf{h}_{il}^2 , объединенные в один вектор. После l -й итерации пару предложений будем относить к классу с наибольшим значением, полученным после l -й итерации, $j = 1, \dots, l$:

$$y = \arg \max_{c \in \{1, \dots, R\}} \left(\mathbf{W}_3 \begin{bmatrix} \mathbf{h}_{il}^1 \\ \mathbf{h}_{il}^2 \end{bmatrix} \right) [c], \quad (4.9)$$

где $(\cdot)[c]$ — c -я компонента вектора.

В качестве оптимизируемой функции потерь L выступает вариационная оценка правдоподобия модели (2.8):

$$L = - \sum_{i=1}^m \log p(y_i | \mathbf{x}_i, \hat{\mathbf{W}}) + D_{\text{KL}}(q(\mathbf{W}) || p(\mathbf{W} | \mathbf{h})), \quad \hat{\mathbf{W}} \sim q, \quad (4.10)$$

где q — вариационное распределение, аппроксимирующее неизвестное апостериорное распределение параметров. В качестве вариационного распределения выберем нормальное распределение:

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

где $\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}$ — вектор средних и матрица ковариации. Априорное распределение $p(\mathbf{w} | \mathbf{h})$ вектора параметров \mathbf{w} будем считать нормальным с параметрами $\boldsymbol{\mu}$ и \mathbf{A} :

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

где $\boldsymbol{\mu}$ — вектор средних, \mathbf{A}^{-1} — матрица ковариаций.

Рассмотрим частные случаи вида матриц ковариаций \mathbf{A}_q^{-1} и \mathbf{A}^{-1} . Так как априори нет предпочтений при выборе параметров, то априорное распределение для всех параметров считаем одинаковым, т. е. вектор средних $\boldsymbol{\mu} = \mu$, матрица ковариаций скалярна: $\mathbf{A}^{-1} = \alpha \mathbf{I}$.

Априорное распределение уточняется после каждого шага оптимизации вариационных параметров. Алгоритм решения оптимизационной задачи заключается в выполнении градиентного шага при заданном априорном распределении, вычислении апостериорного распределения и аппроксимации нового априорного распределения полученным апостериорным.

Рассмотрим различные виды матрицы ковариаций \mathbf{A}_q^{-1} вариационного распределения q .

1. Матрица ковариаций скалярна: $\mathbf{A}_q^{-1} = \alpha_q \mathbf{I}$. В этом случае дивергенция выглядит следующим образом:

$$D_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}) || \mathcal{N}(\boldsymbol{\mu}, \mathbf{A}^{-1})) = \sum_{j=1}^{|\mathbf{W}|} \left(\log \frac{\alpha}{\alpha_q} + \frac{(\mu - \boldsymbol{\mu}_q[j])^2 + \alpha_q^2 + \alpha^2}{2\alpha^2} \right),$$

где $\boldsymbol{\mu}_q[j]$ — i -я компонента вектора $\boldsymbol{\mu}_q$.

По значениям параметров α_q и $\boldsymbol{\mu}_q$ вариационного распределения вычислим оптимальные параметры априорного. Из условия

$$\frac{\partial}{\partial \mu} D_{\text{KL}} = \sum_{j=1}^{|\mathbf{W}|} \frac{\mu - \boldsymbol{\mu}_q[j]}{\alpha^2} = 0$$

получаем выражения для μ на следующей итерации

$$\mu' = \frac{1}{|\mathbf{W}|} \sum_{j=1}^{|\mathbf{W}|} \boldsymbol{\mu}_q[j].$$

Аналогично

$$\frac{\partial}{\partial \alpha^2} D_{\text{KL}} = \sum_{j=1}^{|\mathbf{W}|} \frac{1}{2\alpha^2} - \frac{(\mu - \boldsymbol{\mu}_q[j])^2 + \alpha_q^2}{2\alpha^4} = 0 \Rightarrow \hat{\alpha}^2 = \frac{1}{|\mathbf{W}|} \sum_{i=1}^{|\mathbf{W}|} (\mu - \boldsymbol{\mu}_q[j])^2 + \alpha_q^2.$$

2. Матрица ковариаций диагональна: $\mathbf{A}_q^{-1} = \text{diag}(\boldsymbol{\alpha}_q^2)$.

В этом случае

$$D_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q^{-1}) || \mathcal{N}(\boldsymbol{\mu}, \mathbf{A}^{-1})) = \sum_{j=1}^{|\mathbf{W}|} \left(\log \frac{\alpha}{\boldsymbol{\alpha}_q[j]} + \frac{(\mu - \boldsymbol{\mu}_q[j])^2 + \boldsymbol{\alpha}_q[j]^2 + \alpha^2}{2\alpha^2} \right).$$

Значения параметров априорного распределения для следующей итерации вычисляются следующим образом:

$$\text{из } \frac{\partial}{\partial \mu} D_{\text{KL}} = \sum_{j=1}^{|\mathbf{W}|} \frac{\mu - \boldsymbol{\mu}_q[j]}{\alpha^2} = 0 \text{ получаем } \hat{\mu} = \frac{1}{|\mathbf{W}|} \sum_{j=1}^{|\mathbf{W}|} \boldsymbol{\mu}_q[j],$$

$$\text{из } \frac{\partial}{\partial \alpha^2} D_{\text{KL}} = \sum_{j=1}^{|\mathbf{W}|} \frac{1}{2\alpha^2} - \frac{(\mu - \boldsymbol{\mu}_q[j])^2 + \boldsymbol{\alpha}_q[j]^2}{2\alpha^4} = 0$$

получаем

$$\hat{\alpha}^2 = \frac{1}{|\mathbf{W}|} \sum_{j=1}^{|\mathbf{W}|} (\mu - \boldsymbol{\mu}_q[j])^2 + \boldsymbol{\alpha}_q[j]^2.$$

Оптимизация параметров сводится к следующему алгоритму:

1. Инициализировать $\boldsymbol{\alpha}_q = \mathbf{1}$, $\boldsymbol{\mu}_q = \mathbf{0}$, $\mu = 0$, $\alpha^2 = 1$.

Повторять:

2. Сделать градиентный шаг (2.17) по вариационным параметрам $\boldsymbol{\theta} = [\boldsymbol{\mu}, \boldsymbol{\alpha}_q]$.
3. Обновить параметры априорного распределения.
4. **Пока** значение L не стабилизируется.

Удаление нерелевантных параметров Введем множество индексов активных параметров модели $\mathcal{A} = \{j | w_j \neq 0\}$. Для увеличения правдоподобия модели предлагается уменьшить число активных параметров $|\mathcal{A}|$. Для удаления выберем параметры, имеющие наибольшую плотность вариационной вероятности ρ в нуле. Если апостериорная матрица ковариаций скалярна, то

$$\rho_j = \exp\left(-\frac{\boldsymbol{\mu}_q[j]^2}{\boldsymbol{\alpha}_q[j]^2}\right). \quad (4.11)$$

Чем больше ρ , тем меньше $|\frac{\boldsymbol{\mu}_q[j]}{\boldsymbol{\alpha}_q[j]}|$, поэтому удаляются параметры со значением $|\frac{\boldsymbol{\mu}_q[j]}{\boldsymbol{\alpha}_q[j]}| < \lambda$, где λ — пороговое значение. Варьируя пороговое значение λ , выбираем оптимальное число неудаленных параметров. Для диагонального вида матрицы ковариаций критерий удаления параметров записывается как $|\frac{\boldsymbol{\mu}_q[j]}{\boldsymbol{\alpha}_q[j]}| < \lambda$.

Вычислительный эксперимент. Цель эксперимента — проверка работоспособности предложенного алгоритма и сравнение результатов с ранее полученными. В качестве данных использовалась выборка SemEval 2015, состоящая из 8331 пары схожих и несхожих предложений. Слова преобразовывались в векторы размерности 50 при помощи алгоритма GloVe [?].

Для базовых алгоритмов тренировочная, валидационная и тестовая выборки составили 70%, 15% и 15% соответственно. Для рекуррентной нейронной сети, полученной вариационным методом, валидационная выборка отсутствовала, а тренировочная и тестовая выборки составили 85% и 15% соответственно. Критерием качества была выбрана F_1 -мера. В качестве базовых алгоритмов использовались линейная регрессия, метод ближайших соседей, решающее дерево и модификация метода опорных векторов SVC. Базовые алгоритмы взяты из библиотеки sklearn [89]. Дополнительно были построены рекуррентная нейросеть с одним скрытым слоем [?] и нейросеть с одним скрытым слоем и вариационной оптимизацией параметров.

На рис. 4.5а и 4.5б представлена зависимость оценки правдоподобия L (4.12) от параметра λ . Для обоих случаев существует оптимальное значение λ , минимизирующее L ; модели с таким параметром будут оптимальными. На рис. 4.5в, 4.5г, 4.5д и 4.5е отображены зависимости качества модели от λ и доли

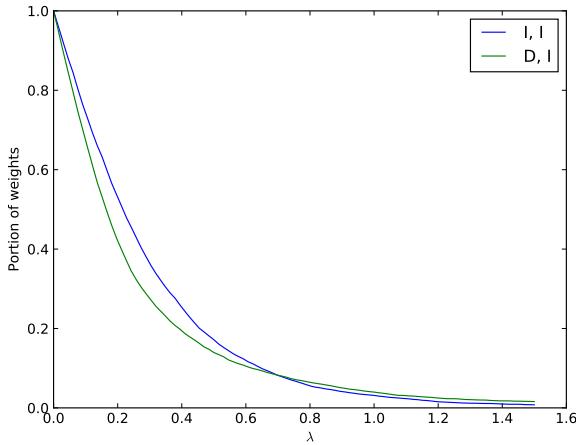


Рис. 4.4. Доля неудаленных параметров сети в зависимости от порогового значения λ для скалярного (I) и диагонального (D) вида апостериорной матрицы ковариаций.

выброшенных параметров. Видно, что даже при удалении большинства параметров из сети качество предсказаний меняется несущественно, что говорит о слишком большом числе параметров исходной модели.

Из рис. 4.4 видно, что при малых λ из сети с диагональной апостериорной матрицей ковариаций удаляется больше весов, а при больших λ — меньше, что говорит о лучшем отборе параметров такой моделью.

Таблица 4.1. Результаты вычислительного эксперимента

Модель	F_1 , валидация	F_1 , тест
Логистическая регрессия	0,286	0,286
SVC	0,290	0,290
Дерево решений	0,316	0,316
KNN	0,322	0,322
Рекуррентная модель	0,393	0,362
Рекуррентная модель с вариационным распределением, $\mathbf{A} = \alpha \mathbf{I}, \mathbf{A}_q = \alpha_q \mathbf{I}$	—	0,311
Рекуррентная модель с вариационным распределением, $\mathbf{A} = \alpha \mathbf{I}, \mathbf{A}_q$ диагональная	—	0,330

4.3. Определение релевантности параметров модели глубокого обучения

В данном разделе решается задача выбора субоптимальной структуры нейронной сети. Предлагается удалять наименее релевантные параметры модели.

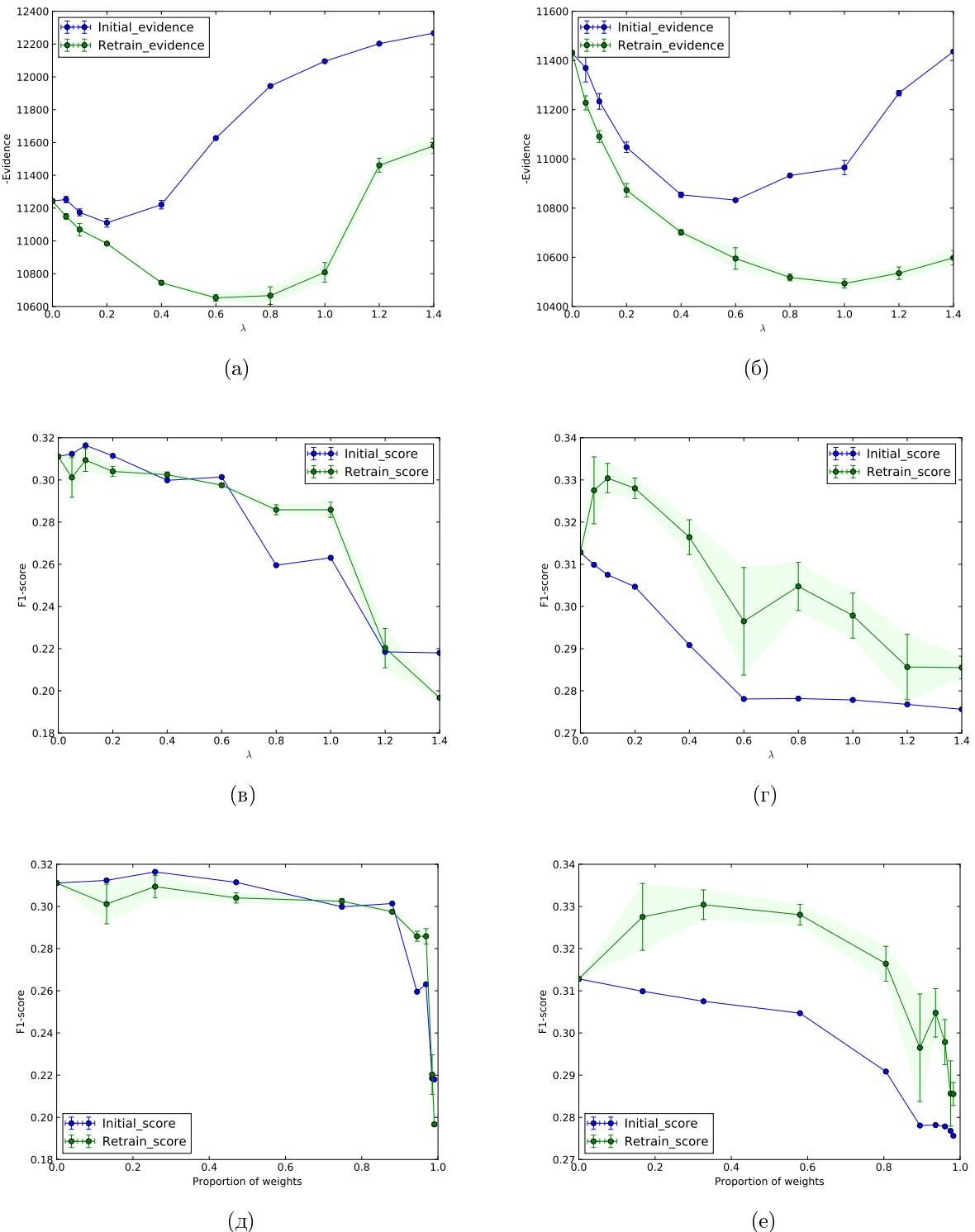


Рис. 4.5. Зависимость нижней оценки правдоподобия модели и F1-меры от λ для скалярной (а, б, в) и диагональной (г, д, е) матриц.

Под релевантностью [39] подразумевается то, насколько параметр влияет на функцию ошибки. Малая релевантность указывает на то, что удаление этого параметра не влечет значимого изменения функции ошибки. Метод предполагает построение исходной избыточной сложности нейросети с большим числом избыточных параметров. Для определения релевантности параметров предлагаются оптимизировать параметры и гиперпараметры в единой процедуре. Для удаления параметров предлагается использовать метод Белсли [?].

Проверка и анализ метода проводится на выборке Boston Housing, Wine и синтетических данных. Результат сравнивается с моделью, полученной при помощи базовых алгоритмов.

Постановка задачи. Задана выборка

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

где $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{1, \dots, R\}$, R — число классов. Рассмотрим модель

$$\mathbf{f}(\mathbf{x}, \mathbf{W}) : \mathbb{R}^n \times \mathbb{W} \rightarrow [0, 1]^R,$$

$$\mathbf{f}(\mathbf{W}, \mathbf{x}) = \text{softmax}(\mathbf{f}_1(\mathbf{f}_2(\dots(\mathbf{f}_{|V|-1}(\mathbf{x}, \mathbf{W})).$$

Параметр w_j модели \mathbf{f} называется активным, если $w_j \neq 0$. Множество индексов активных параметров обозначим \mathcal{A} . Задано пространство активных параметров модели:

$$\mathbb{W}_{\mathcal{A}} = \{\mathbf{w} \in \mathbb{W} \mid w_j \neq 0, j \in \mathcal{A}\}.$$

Для модели \mathbf{f} с множеством индексов активных параметров \mathcal{A} и соответствующего ей вектора параметров $\mathbf{w} \in \mathbb{W}_{\mathcal{A}}$ определим логарифмическую функцию правдоподобия выборки (2.8):

$$L = - \sum_{i=1}^m \log p(y_i | \mathbf{x}_i, \hat{\mathbf{W}}) + D_{\text{KL}}(q(\mathbf{W}) || p(\mathbf{W} | \mathbf{h})), \quad \hat{\mathbf{W}} \sim q, \quad (4.12)$$

где q — вариационное распределение, аппроксимирующее неизвестное апостериорное распределение параметров.

Аналогично (4.12) будем проводить оптимизацию вариационной оценки правдоподобия модели, где в качестве вариационного распределения q и априорного распределения параметров $p(\mathbf{W} | \mathbf{h})$ выступает нормальное. Требуется найти множество активных параметров $\mathbb{W}_{\mathcal{A}}$, доставляющие минимум функции потерь L :

$$L \rightarrow \min_{\mathcal{A}, \mathbf{W} \in \mathbb{W}_{\mathcal{A}}} .$$

Случайное удаление. Метод случайного удаления заключается в том, что случайным образом удаляется некоторый параметр w_ξ из множества активных параметров сети. Индекс параметра ξ порождается из равномерного распределения:

$$\xi \sim \mathcal{U}(\mathcal{A})$$

Оптимальное прореживание. Метод оптимального прореживания [8] (1.8) использует вторую производную целевой функции. Нахождение очередного индекса элемента для удаления сводится к задаче оптимизации:

$$\xi = \arg \min \frac{w_\xi^2}{2\mathbf{H}[i, j]},$$

где ξ — индекс наименее релевантного, удаляемого параметра.

Удаление неинформативных параметров с помощью вариационного вывода. В работе [39] предлагается удалять параметры, которые имеют максимальное отношение плотности $p(\mathbf{w}|\mathcal{A})$ априорной вероятности в нуле к плотности вероятности априорной вероятности в математическом ожидании μ_j параметра w_j .

Для гауссовского распределения с диагональной матрицей ковариации получаем:

$$p(w_j|\mathbf{h}) = \frac{1}{\sqrt{2\alpha_q[j]^2}} \exp\left(-\frac{(w_j - \mu_j)^2}{2\alpha_q[j]^2}\right).$$

Разделим плотность вероятности в нуле к плотности в математическом ожидании

$$\frac{p(w_j = 0|\mathbf{h})}{p(w_j = \mu_j|\mathbf{h})} = \exp\left(-\frac{\mu_j^2}{2\alpha_j^2}\right),$$

и поставим следующую задачу оптимизации:

$$\xi = \arg \min_{j \in \mathcal{A}} \left| \frac{\mu_j}{\alpha_j} \right|,$$

где ξ — индекс наименее релевантного, удаляемого параметра.

Предлагаемый метод определения релевантности параметров нейросети. Предлагается метод основанный, на модификации метода Белсли. Пусть \mathbf{W} — вектор параметров, доставляющий минимум функционалу потерь L на множестве $\mathbb{W}_{\mathcal{A}}$, а \mathbf{A}_q соответствующая ему ковариационная матрица.

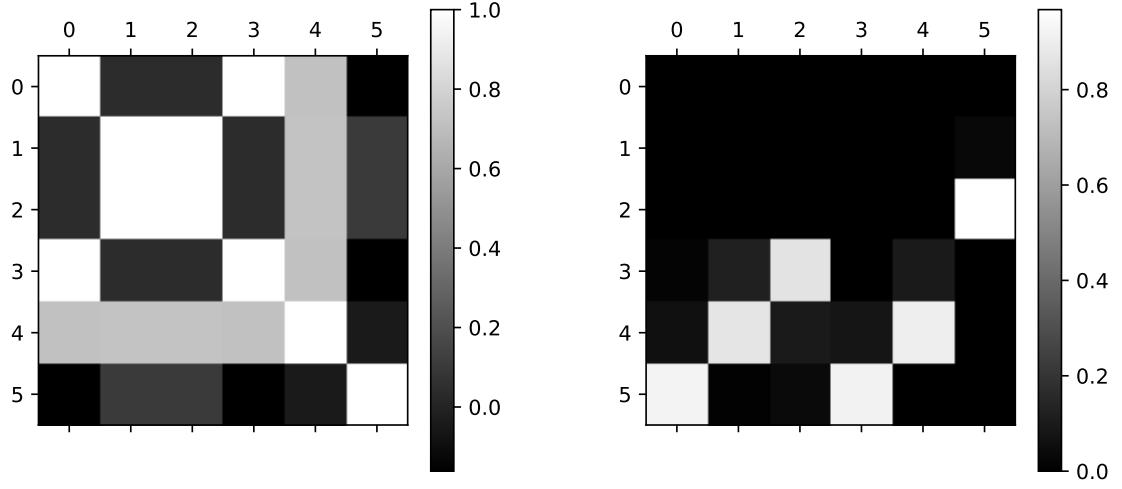
Выполним сингулярное разложение матрицы

$$\mathbf{A}_q = \mathbf{U}\Lambda\mathbf{V}^\top.$$

Индекс обусловленности η_j определим как отношение максимального элемента к j -му элементу матрицы Λ . Для нахождения мультикоррелирующих признаков требуется найти индекс ξ вида:

$$\xi = \arg \max_{j \in \mathcal{A}} \eta_j.$$

Дисперсионный долевой коэффициент $q_{i,j}$ определим как вклад j -го признака в дисперсию i -го элемента вектора параметра \mathbf{w} :



(a) Матрица ковариации

(б) Дисперсионные доли

Рис. 4.6. Иллюстрация метода Белсли

Таблица 4.2. Иллюстрация метода Белсли

η	q_1	q_2	q_3	q_4	q_5	q_6
1.0	$2 \cdot 10^{-17}$	$4 \cdot 10^{-17}$	$1 \cdot 10^{-16}$	$2 \cdot 10^{-17}$	$6 \cdot 10^{-17}$	$3 \cdot 10^{-4}$
1.5	$5 \cdot 10^{-17}$	$9 \cdot 10^{-17}$	$2 \cdot 10^{-16}$	$5 \cdot 10^{-17}$	$3 \cdot 10^{-20}$	$3 \cdot 10^{-2}$
3.3	$9 \cdot 10^{-18}$	$1 \cdot 10^{-17}$	$2 \cdot 10^{-17}$	$9 \cdot 10^{-18}$	$2 \cdot 10^{-19}$	$9 \cdot 10^{-1}$
$2 \cdot 10^{15}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$8 \cdot 10^{-1}$	$2 \cdot 10^{-3}$	$9 \cdot 10^{-2}$	$1 \cdot 10^{17}$
$8 \cdot 10^{15}$	$6 \cdot 10^{-2}$	$8 \cdot 10^{-1}$	$9 \cdot 10^{-2}$	$8 \cdot 10^{-2}$	$9 \cdot 10^{-1}$	$2 \cdot 10^{17}$
$1 \cdot 10^{16}$	$9 \cdot 10^{-1}$	$1 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$9 \cdot 10^{-1}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{-21}$

$$q_{i,j} = \frac{u_{i,j}^2 / \lambda_{j,j}}{\sum_{j=1}^n u_{i,j}^2 / \lambda_{j,j}}.$$

Большие значение дисперсионных долей указывают на наличие зависимости между параметрами. Находим долевые коэффициенты, которые вносят максимальный вклад в дисперсию параметра w_ξ :

$$\zeta = \arg \max_{j \in \mathcal{A}} q_{\xi,j}.$$

Параметр с индексом ζ определим как наименее релевантный параметр нейронной сети.

Проиллюстрируем принцип работы метода Белсли на примере. Рассмотрим

данные, порожденные следующим образом:

$$\mathbf{w} = \begin{bmatrix} \sin(x) \\ \cos(x) \\ 2+\cos(x) \\ 2+\sin(x) \\ \cos(x) + \sin(x) \\ x \end{bmatrix},$$

с матрицей ковариации на рис. 4.6.а, где $x \in \{0.0, 0.02, \dots, 20.0\}$.

В табл. 4.2 приведены индексы обусловленности и соответствующие им дисперсионные доли, которые также изображены на рис. 4.6.б. Согласно этим данным, максимальный индекс обусловленности $\eta_6 = 1.2 \cdot 10^{16}$. Ему соответствуют максимальные дисперсионные доли признаков с индексами 1 и 4, которые, как видно из построения выборки, коррелируют между собой.

Вычислительный эксперимент. Для анализа свойств предложенного алгоритма и сравнения его с существующими был проведен вычислительный эксперимент. В качестве данных использовались три выборки. Выборки Wine [?] и Boston Housing [?] — это реальные данные. Синтетические данные сгенерированы таким образом, чтобы параметры сети были мультикоррелируемые. Генерация данных состояла из двух этапов. На первом этапе генерировался вектор параметров $\mathbf{W}_{\text{synthetic}}$:

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

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

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

$$\mathfrak{D}_{\text{synthetic}} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \sim \mathcal{N}(\mathbf{1}, \mathbf{I}), y_i = x_{i,0}, i = 1 \dots 10000\}.$$

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

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

Таблица 4.3. Описание выборок

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

Качеством прогноза Accuracy модели для задачи классификации является точность прогноза модели:

$$\text{Accuracy} = \frac{\sum_{i=1}^m [f(\mathbf{x}_i, \mathbf{w}) = y_i]}{m},$$

Качеством прогноза RMSE модели для задачи регрессии является среднеквадратическое отклонение результата модели от точного:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^m (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2}{m}}.$$

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

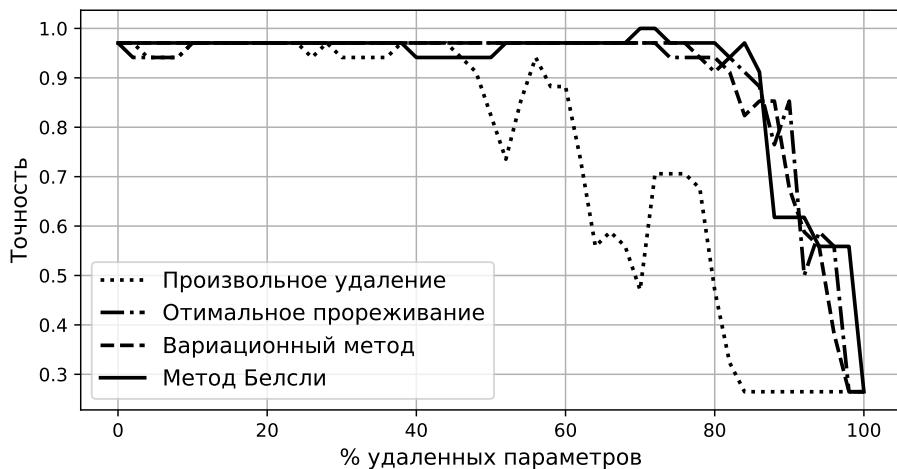


Рис. 4.7. Качество прогноза при удалении параметров на выборке Wine

На рис. 4.7 показано как меняется точность прогноза Accuracy при удалении параметров указанными методами. Из графика видно, что метод оптимального прореживания, вариационный метод и метод Белсли позволяют удалить $\approx 80\%$ параметров и качество всех этих методов падает при удалении $\approx 90\%$ параметров нейросети.

На рис. 4.8 показаны поверхности изменения уровня шума ответов нейросети при изменении процента удаленных параметров и уровня шума входных

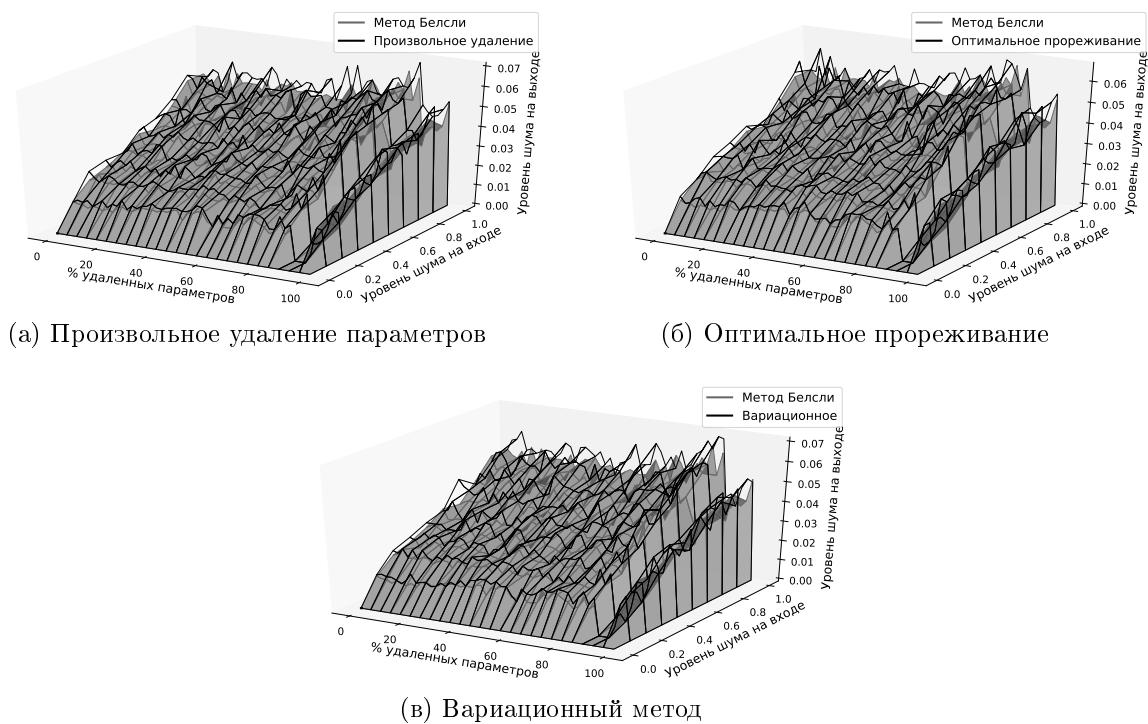


Рис. 4.8. Влияние шума в начальных данных на шум выхода нейросети на выборке Wine

данных для разных методов прореживания. На графиках показано, что при удалении параметров нейросети методом Белсли шум меньше, чем при удалении параметров другими методами, на это указывает то что поверхность которая соответствует методу Белсли ниже других поверхностей.

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

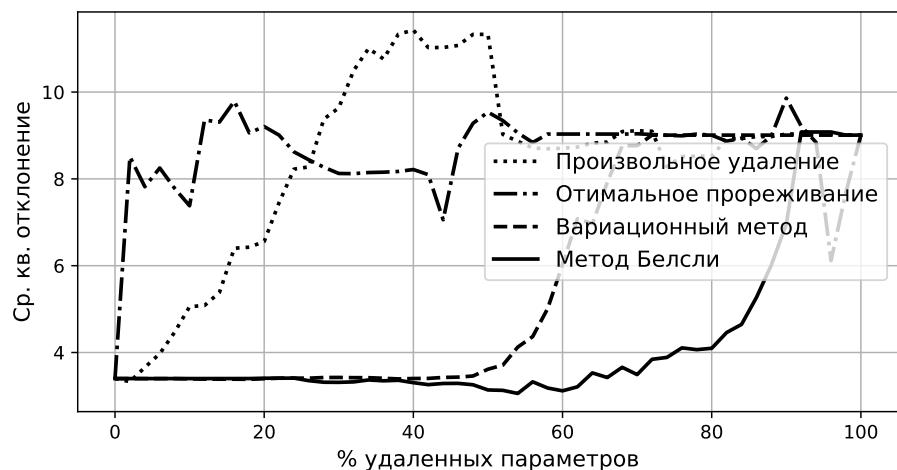


Рис. 4.9. Качество прогноза при удаление параметров на выборке Boston

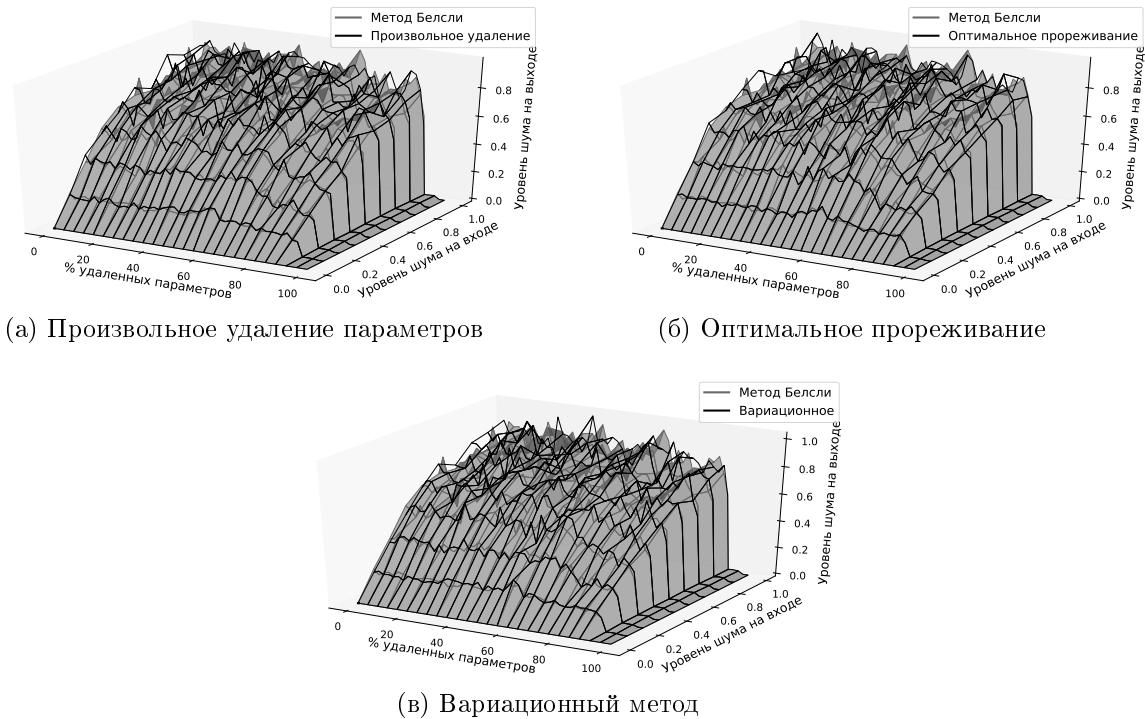


Рис. 4.10. Влияние шума в начальных данных на шум выхода нейросети на выборке Boston

На рис. 4.9 показано как меняется среднеквадратическое отклонение прогноза RMSE от точного ответа при удалении параметров указанными методами. График показывает, что метод Белсли является более эффективным, чем другие методы, т.к. позволяет удалить больше параметров нейросети без потери качества.

На рис. 4.10 показаны поверхности изменения уровня шума ответов нейросети при изменении процента удаленных параметров и уровня шума входных данных для разных методов прореживания. График показывает, что уровень шума всех методов одинаковый, так как поверхности всех методов находятся на одном уровне.

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

На рис. 4.11 показано как меняется среднеквадратическое отклонение прогноза от RMSE точного ответа при удалении параметров указанными методами. График показывает, что удаление параметров методом Белсли является более эффективным чем другие методы прореживания, т.к. качество прогноза нейросети улучшается при удалении шумовых параметров.

На рис. 4.12 показаны поверхности изменения уровня шума ответов нейросети при изменении процента удаленных параметров и уровня шума входных данных для разных методов прореживания. На графиках показано, что при удалении параметров нейросети методом Белсли шум меньше, чем при удалении параметров другими методами, т.к. поверхность которая соответствует

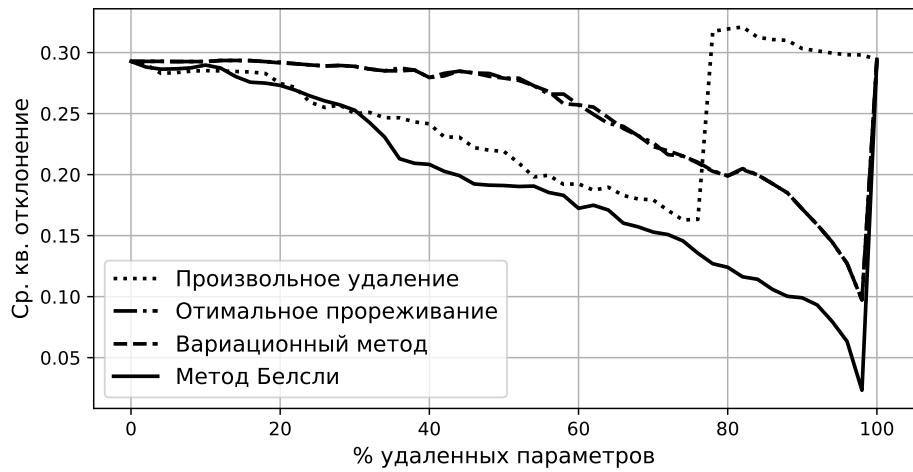


Рис. 4.11. Качество прогноза при удаление параметров на синтетической выборке

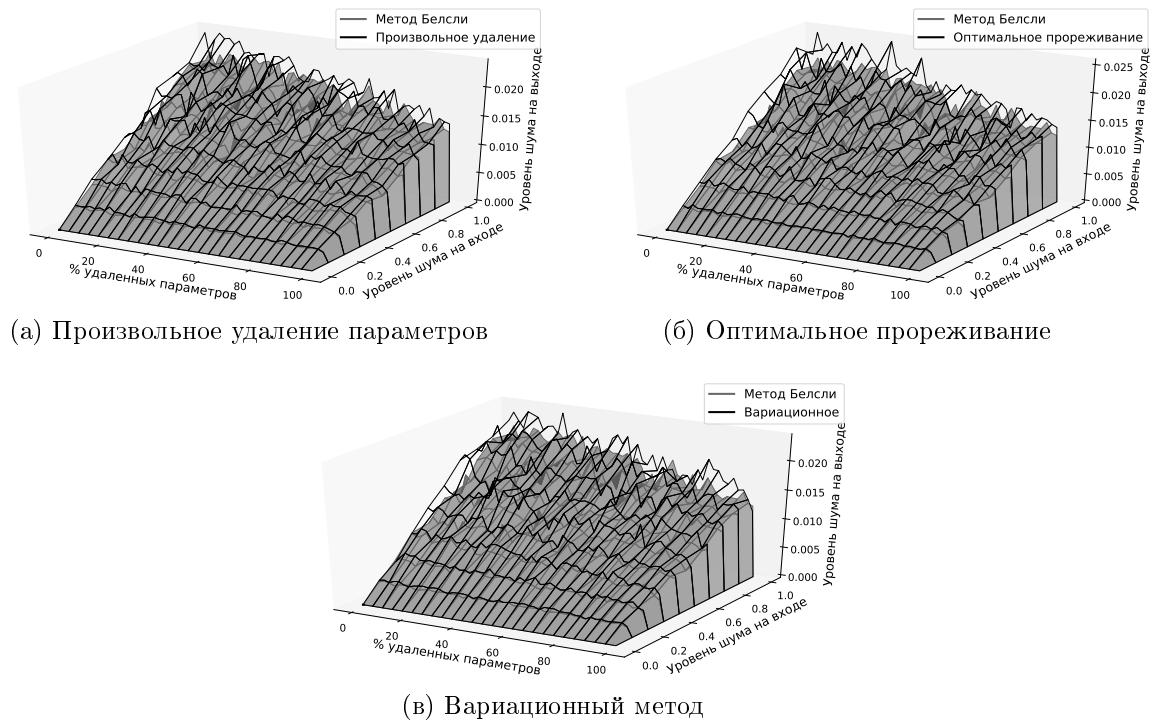


Рис. 4.12. Влияние шума в начальных данных на шум выхода нейросети на синтетической выборке

методу Белсли ниже других поверхностей.

Заключение

Основные результаты диссертационной работы заключаются в следующем.

В главе 1 введены основные понятия, поставлены задачи выбора модели глубокого обучения и проанализированы методы оптимизации параметров модели, методы оптимизации гиперпараметров, методы представления моделей глубокого обучения в графовом виде, методы оптимизации структурных параметров и метапараметров модели. Последние включают в себя как эвристические методы, так и методы, основанные на байесовском выводе и вероятностных предположениях о распределении параметров, гиперпараметров и метапараметров модели.

В главе 2 были предложены критерии оптимальной и субоптимальной сложности моделей глубокого обучения. Предложен алгоритм выбора субоптимальной модели, основанный на получении вариационной нижней оценки правдоподобия модели. Был предложен метод получения оценки, основанный на стохастическом градиентном спуске, позволяющий проводить выбор модели и оптимизацию модели единообразно. Исследованы свойства стохастического градиентного спуска, а также оценок правдоподобия, полученных с его использованием. Работа представленного алгоритма проиллюстрирована рядом выборок. Вычислительный эксперимент продемонстрировал значимое влияние априорного распределения на апостериорное распределение параметров модели.

В главе 3 были проанализированы градиентные методы оптимизации гиперпараметров. Предложено обобщение существующих методов на функции потерь и валидации общего вида. Было проведено сравнение двух критериев выбора модели: на основе кросс-валидации и на основе вариационной оценки правдоподобия модели. Эксперименты показали, что градиентные методы оптимизации гиперпараметров являются эффективными в случае, когда число гиперпараметров велико. Также эксперименты показали, что те модели, гиперпараметры и параметры которых были оптимизированы с использованием вариационной оценки правдоподобия модели, имеют меньшую точность классификации, чем те модели, чьи гиперпараметры и параметры были оптимизированы с использованием метода кросс-валидации. В то же время, первые модели оказались более робастными при доабвлении шума в выборку. Модели, чья оптимизация проводилась с использованием вариационной оценки правдоподобия, оказались значительно лучшими на синтетической выборке, когда число объектов в обучающей выборке мало по сравнению с числом параметров. Поэтому вариационная оценка правдоподобия более предпочтительна, когда вероятность переобучения моделей велика или когда проведение кросс-валидации вычислительно затратно.

В главе 4 был предложен обобщенный метод выбора структуры модели субоптимальной сложности. Формализовано понятие параметрической сложности для вероятностных моделей. Сформулированы требования к вариационным распределениям, введенным на структуре модели. Показано, что предложенный

метод выбора структуры модели обобщает такие методы выбора модели как оптимизация согласно критерию максимального правдоподобия, оптимизация вариационной оценки обоснованности модели, снижение и увеличение сложности модели, а также полный перебор.

В главе 5 проведен анализ свойств предложенных методов. Описан реализованный программный комплекс, позволяющий осуществлять выбор моделей глубокого обучения. Проведено сравнение предложенных алгоритмов с известными решениями. Предложенные алгоритмы показали более высокие результаты.

Список основных обозначений

- $\mathbf{x}_i \in \mathbf{X}$ — вектор признакового описания i -го объекта
 $y_i \in \mathbf{y}$ — метка i -го объекта
 \mathfrak{D} — выборка
 $\mathbf{X} \subset \mathbb{X}$ — матрица, содержащая признаковое описание объектов выборки
 $\mathbf{y} \subset \mathbb{Y}$ — вектор меток объектов выборки
 m — количество объектов в выборке
 n — количество признаков в признаковом описании объекта
 $\mathbb{X} = \mathbb{R}^m$ — признаковое пространство объектов
 \mathbb{Y} — множество меток объектов
 R — множество классов в задаче классификации
 r — число оптимизаций модели
 (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
 $\gamma^{j,k}$ — структурный параметр для ребра (j, k)
 Δ^K — симплекс на K вершинах
 $\bar{\Delta}^K$ — множество вершин симплекса на K вершинах
 \mathfrak{F} — параметрическое семейство моделей
 U — область определения оптимизационной задачи
 $\mathbf{w} \in \mathbb{W}$ — параметры модели
 \mathbb{W} — пространство параметров модели
 $U_{\mathbf{w}} \subset \mathbb{W}$ — область определения параметров модели
 $\Gamma \in \mathbb{G}$ — структура модели
 \mathbb{G} — множество значений структуры модели
 $U_{\Gamma} \subset \mathbb{G}$ — область определения параметров модели
 $\mathbf{h} \in \mathbb{H}$ — гиперпараметры модели
 \mathbb{H} — пространство гиперпараметров модели
 $U_{\mathbf{h}} \subset \mathbb{H}$ — область определения гиперпараметров
 $\boldsymbol{\theta} \in \Theta$ — параметры вариационного распределения
 Θ — пространство параметров вариационного распределения
 $U_{\boldsymbol{\theta}} \subset \Theta$ — область определения вариационных параметров модели
 $\boldsymbol{\theta}_{\mathbf{w}} \in \Theta_{\mathbf{w}}$ — параметры вариационного распределения, аппроксимирующего апостериорное распределение параметров модели
 $\Theta_{\mathbf{w}}$ — пространство параметров вариационного распределения, аппроксимирующего апостериорное распределение параметров модели
 $U_{\boldsymbol{\theta}_{\mathbf{w}}} \subset \Theta_{\mathbf{w}}$ — область определения параметров вариационного распределения, аппроксимирующего апостериорное распределение параметров модели
 $\boldsymbol{\theta}_{\Gamma} \in \Theta_{\Gamma}$ — параметры вариационного распределения, аппроксимирующего апостериорное распределение структуры модели
 Θ_{Γ} — пространство параметров вариационного распределения, аппроксимирующую-

щего апостериорное распределение структуры модели
 $U_{\theta_\Gamma} \subset \Theta_\Gamma$ — область определения параметров вариационного распределения, априорного апостериорное распределение структуры модели
 $\lambda \in \Lambda$ — вектор метапараметров
 Λ — пространство метапараметров
 $U_\lambda \subset \Lambda$ — область определения метапараметров
 $p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \boldsymbol{\Gamma})$ — правдоподобие выборки
 $p(\mathbf{w}, \boldsymbol{\Gamma}|\mathbf{h}, \boldsymbol{\lambda})$ — априорное распределение параметров и структуры модели
 $p(\mathbf{h}|\boldsymbol{\lambda})$ — распределение гиперпараметров модели
 $p(\boldsymbol{\Gamma}|\mathbf{h}, \boldsymbol{\lambda})$ — априорное распределение структуры модели
 $p(\mathbf{w}|\boldsymbol{\Gamma}, \mathbf{h}, \boldsymbol{\lambda})$ — априорное распределение параметров модели
 $p(\mathbf{w}, \boldsymbol{\Gamma}|\mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})$ — апостериорное распределение параметров и структуры модели
 $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \boldsymbol{\Gamma}, \mathbf{h}, \boldsymbol{\lambda})$ — апостериорное распределение структуры модели
 $p(\boldsymbol{\Gamma}|\mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})$ — апостериорное распределение структуры модели
 $p(\mathbf{h}|\mathbf{y}, \mathbf{X}, \boldsymbol{\lambda})$ — апостериорное распределение гиперпараметров
 $p(y, \mathbf{w}, \boldsymbol{\Gamma}|\mathbf{x}, \mathbf{h})$ — вероятностная модель глубокого обучения
 $p(\mathbf{y}|\mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})$ — обоснованность модели
 $q(\mathbf{w}, \boldsymbol{\Gamma}|\boldsymbol{\theta})$ — вариационное распределение параметров и структуры модели
 $q_{\mathbf{w}}(\mathbf{w}|\boldsymbol{\Gamma}, \boldsymbol{\theta}_{\mathbf{w}})$ — вариационное распределение структуры модели
 $q_{\boldsymbol{\Gamma}}(\boldsymbol{\Gamma}|\boldsymbol{\theta}_\Gamma)$ — вариационное распределение параметров модели
 $L(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda})$ — функция потерь
 $Q(\mathbf{h}|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\lambda})$ — валидационная функция
 $T(\boldsymbol{\theta}|L(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X}, \mathbf{h}, \boldsymbol{\lambda}))$ — оператор оптимизации
 \mathfrak{Q} — семейство вариационные распределений
 S — энтропия распределения
 M — множество моделей без общей параметризации
 $D_{\text{KL}}(p_1||p_2)$ — дивергенция Кульбака-Лейблера между распределениями p_1 и p_2
 \mathbf{A}^{-1} — матрица ковариаций параметров модели
 \mathbf{s} — конкатенация параметров концентрации на структуре модели

Список иллюстраций

1.1	Пример параметрического семейства моделей глубокого обучения: семейство описывает сверточную нейронную сеть.	3
1.2	Примеры ограничений для одного структурного параметра $\gamma^{j,k}, K^{j,k} = 3$. 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
1.3	Пример параметрического семейства моделей глубокого обучения: семейство описывает многослойную полносвязную нейронную сеть с одним скрытым слоем и нелинейной функцией активации σ	5
1.4	Пример итерации алгоритма AdaNet [16]. Рассматриваются две альтернативные модели: модель с углублением сети (соответствует занулению функции \mathbf{f}_2 с использованием базовой функции $\mathbf{g}_1^{1,2}$) и модель с расширением сети (соответствует базовой функции $\mathbf{g}_0^{1,2}$).	16
1.5	Пример параметрического семейства моделей глубокого обучения, описываемый в [17]. Каждая подмодель \mathbf{f}_j является линейной комбинацией базовых функций: свертки и результата работы предыдущих подмоделей (англ. skip-connection).	17
1.6	Пример суперсети. Каждый путь из подмодели \mathbf{f}_0 в конечную модель \mathbf{f}_8 задает модель глубокого обучения.	20
1.7	Схема порождения вектора объектов \mathbf{X} , представленная в [53].	22
2.1	Аппроксимация распределения а) нормальным распределением, б) распределением, полученным с помощью градиентного спуска, в) с использованием стохастической динамики Ланжевена.	29
2.2	Псевдокод алгоритма получения вариационной нижней оценки обоснованности модели с использованием градиентного спуска	33
2.3	Возмущение выборки для однослойных нейросетей: а) Boston Housing, б) Protein, в) MSD.	37
3.1	Псевдокод общего алгоритма оптимизации гиперпараметров.	46
3.2	Псевдокод жадного алгоритма оптимизации гиперпараметров.	47
3.3	Псевдокод алгоритма HOAG.	47
3.4	Псевдокод алгоритма DrMAD.	48
3.5	Иллюстративный пример действия операторов оптимизации на гиперпараметры. Интенсивность цвета графика соответствует значения функции валидации Q	48
3.6	Итоговые модели для синтетической выборки: а) с использованием кросс-валидации, б) с использованием вариационной оценки обоснованности модели.	53

3.7	WISDM, наилучшее значение функции Q и RMSE для кросс-валидации (слева) и вариационной оценки обоснованности модели (справа)	54
3.8	MNIST, наилучшее значение функции Q и RMSE для кросс-валидации (слева) и вариационной оценки обоснованности модели (справа)	55
3.9	MNIST, точность классификации на тестовой выборке при добавлении шума в обучающую выборку. Гиперпараметры был оптимизированы с использованием вариационной оценки обоснованности модели.	55
4.1	Зависимость ошибки от числа нейронов	60
4.2	Зависимость ошибки от размера обучающей выборки	60
4.3	Результаты эксперимента по исследованию скорости процесса обучения	61
4.4	Доля неудаленных параметров сети в зависимости от порогового значения λ для скалярного (I) и диагонального (D) вида апостериорной матрицы ковариаций.	66
4.5	Зависимость нижней оценки правдоподобия модели и F1-меры от λ	67
4.6	Илюстрация метода Белсли	70
4.7	Качество прогноза при удаление параметров на выборке Wine . .	72
4.8	Влияние шума в начальных данных на шум выхода нейросети на выборке Wine	73
4.9	Качество прогноза при удаление параметров на выборке Boston . .	73
4.10	Влияние шума в начальных данных на шум выхода нейросети на выборке Boston	74
4.11	Качество прогноза при удаление параметров на синтетической выборке	75
4.12	Влияние шума в начальных данных на шум выхода нейросети на синтетической выборке	75

Список таблиц

2.1	Описание выборок для экспериментов по выбору моделей	37
2.2	Результаты эксперимента по выбору моделей	38
3.1	Преимущества и недостатки рассматриваемых алгоритмов	43
3.2	Сложность и предположения для различных алгоритмов оптимизации гиперпараметров	44
3.3	Результаты эксперимента по оптимизации гиперпараметров.	52
4.1	Результаты вычислительного эксперимента	66
4.2	Иллюстрация метода Белсли	70
4.3	Описание выборок	72

Список использованных источников

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

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

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

35. Predicting Parameters in Deep Learning / Misha Denil, Babak Shakibi, Laurent Dinh et al. // Advances in Neural Information Processing Systems 26 / Ed. by C.j.c. Burges, L. Bottou, M. Welling et al. — 2013. — Pp. 2148–2156. http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/1053.pdf.
36. Xu Huan, Mannor Shie. Robustness and generalization // *Machine Learning*. — 2012. — Vol. 86, no. 3. — Pp. 391–423. <http://dx.doi.org/10.1007/s10994-011-5268-1>.
37. Intriguing properties of neural networks. / Christian Szegedy, Wojciech Zaremba, Ilya Sutskever et al. // *CoRR*. — 2013. — Vol. abs/1312.6199. <http://dblp.uni-trier.de/db/journals/corr/corr1312.html#SzegedyZSBE GF13>.
38. Stochastic Variational Inference / Matthew D. Hoffman, David M. Blei, Chong Wang, John Paisley // *J. Mach. Learn. Res.* — 2013. — Май. — Vol. 14, no. 1. — Pp. 1303–1347. <http://dl.acm.org/citation.cfm?id=2502581.2502622>.
39. Graves Alex. Practical Variational Inference for Neural Networks // Advances in Neural Information Processing Systems 24 / Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett et al. — Curran Associates, Inc., 2011. — Pp. 2348–2356. <http://papers.nips.cc/paper/4329-practical-variational-inference-for-neural-networks.pdf>.
40. Salimans Tim, Kingma Diederik P., Welling Max. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. // ICML / Ed. by Francis R. Bach, David M. Blei. — Vol. 37 of *JMLR Proceedings*. — JMLR.org, 2015. — Pp. 1218–1226. <http://dblp.uni-trier.de/db/conf/icml/icml2015.html#SalimansKW15>.
41. Maclaurin Dougal, Duvenaud David K., Adams Ryan P. Early Stopping is Nonparametric Variational Inference // *CoRR*. — 2015. — Vol. abs/1504.01344. <http://arxiv.org/abs/1504.01344>.
42. Mandt Stephan, Hoffman Matthew D, Blei David M. Continuous-Time Limit of Stochastic Gradient Descent Revisited.
43. Welling Max, Teh Yee Whye. Bayesian Learning via Stochastic Gradient Langevin Dynamics // Proceedings of the 28th International Conference on Machine Learning (ICML-11) / Ed. by Lise Getoor, Tobias Scheffer. — ICML '11. — New York, NY, USA: ACM, 2011. — June. — Pp. 681–688.
44. Arlot Sylvain, Celisse Alain. A survey of cross-validation procedures for model selection // *Statist. Surv.*. — 2010. — Vol. 4. — Pp. 40–79. <http://dx.doi.org/10.1214/09-SS054>.
45. Fast and Accurate Support Vector Machines on Large Scale Systems / Abhinav Vishnu, Jeyanthi Narasimhan, Lawrence Holder et al. // 2015 IEEE International Conference on Cluster Computing, CLUSTER 2015, Chicago, IL, USA, September 8-11, 2015. — 2015. — Pp. 110–119. <http://dx.doi.org/10.1109/CLUSTER.2015.26>.

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

56. Machine-Translated Text Detection in a Collection of Russian Scientific Papers / Alexey Romanov, Rita Kuznetsova, Oleg Bakhteev, Anton Khritankov // *Dialogue*. — 2016. — P. 2.
57. Бахтеев Олег Юрьевич. Градиентные методы оптимизации гиперпараметров моделей глубокого обучения // Всероссийская конференция ММРО-18. — 2017. — Рр. 10–11.
58. Бахтеев Олег Юрьевич, Кузнецова Маргарита Валерьевна. Детектирование переводных заимствований в текстах научных статей из журналов, входящих в РИНЦ // Всероссийская конференция ММРО-18. — 2017. — Рр. 128–129.
59. Бахтеев Олег Юрьевич. Выбо модели глубокого обучения субоптимальной сложности с использованием вариационной оценки правдоподобия // Интеллектуализация обработки информации ИОИ-2018. — 2016. — Рр. 16–17.
60. Бахтеев Олег Юрьевич, Стрижов Вадим Викторович. Выбор моделей глубокого обучения субоптимальной сложности // *Автоматика и телемеханика*. — 2018. — no. 8. — Pp. 129–147.
61. Bakhteev OY, Strijov VV. Comprehensive analysis of gradient-based hyperparameter optimization algorithms // *Annals of Operations Research*. — 2019. — Pp. 1–15.
62. Бахтеев ОЮ. Восстановление панельной матрицы и ранжирующей модели по метризованной выборке в разнородных шкалах // *Машинное обучение и анализ данных*. — 2006. — Vol. 72, no. 7. — P. 1958.
63. Бахтеев ОЮ. Восстановление пропущенных значений в разнородных шкалах с большим числом пропусков // *Машинное обучение и анализ данных*. — 2015. — Vol. 1, no. 11. — Pp. 1484–1499.
64. Learning deep generative models of graphs / Yujia Li, Oriol Vinyals, Chris Dyer et al. // *arXiv preprint arXiv:1803.03324*. — 2018.
65. Li Jundong, Liu Huan. Challenges of feature selection for big data analytics // *IEEE Intelligent Systems*. — 2017. — Vol. 32, no. 2. — Pp. 9–15.
66. 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.
67. Incremental network quantization: Towards lossless cnns with low-precision weights / Aojun Zhou, Anbang Yao, Yiwen Guo et al. // *arXiv preprint arXiv:1702.03044*. — 2017.
68. 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.
69. 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.

70. *Louizos Christos, Ullrich Karen, Welling Max.* Bayesian compression for deep learning // Advances in Neural Information Processing Systems. — 2017. — Pp. 3290–3300.
71. *Bergstra James, Bengio Yoshua.* Random search for hyper-parameter optimization // *Journal of Machine Learning Research*. — 2012. — Vol. 13, no. Feb. — Pp. 281–305.
72. Algorithms for hyper-parameter optimization / James S Bergstra, Rémi Bardenet, Yoshua Bengio, Balázs Kégl // Advances in Neural Information Processing Systems. — 2011. — Pp. 2546–2554.
73. *Bengio Yoshua.* Gradient-based optimization of hyperparameters // *Neural computation*. — 2000. — Vol. 12, no. 8. — Pp. 1889–1900.
74. 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.
75. *Pedregosa Fabian.* Hyperparameter optimization with approximate gradient // Proceedings of the 33rd International Conference on Machine Learning. — 2016.
76. *Snoek Jasper, Larochelle Hugo, Adams Ryan P.* Practical bayesian optimization of machine learning algorithms // Advances in neural information processing systems. — 2012. — Pp. 2951–2959.
77. Bayesian Optimization in High Dimensions via Random Embeddings. / Ziyu Wang, Masrour Zoghi, Frank Hutter et al. // IJCAI. — 2013. — Pp. 1778–1784.
78. Bayesian Optimization with Tree-structured Dependencies / Rodolphe Jenatton, Cedric Archambeau, Javier González, Matthias Seeger // International Conference on Machine Learning. — 2017. — Pp. 1655–1664.
79. 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.
80. 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>.
81. 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.
82. 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.
83. *Воронцов Константин Вячеславович.* Локальные базисы в алгебраическом подходе к проблеме распознавания: Ph.D. thesis. — Graz, 1999.

84. *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/>.
85. *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>.
86. Automatic differentiation in PyTorch / Adam Paszke, Sam Gross, Soumith Chintala et al. — 2017.
87. *Eibe Frank, Hall MA, Witten IH.* The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques // Morgan Kaufmann. — 2016.
88. *Hofmann Markus, Klinkenberg Ralf.* RapidMiner: Data mining use cases and business analytics applications. — CRC Press, 2013.
89. 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.
90. Relational inductive biases, deep learning, and graph networks / Peter W Battaglia, Jessica B Hamrick, Victor Bapst et al. // *arXiv preprint arXiv:1806.01261*. — 2018.
91. *Negrinho Renato, Gordon Geoff.* Deeparchitect: Automatically designing and training deep architectures // *arXiv preprint arXiv:1704.08792*. — 2017.
92. 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.
93. *Alvarez-Melis David, Jaakkola Tommi S.* Tree-structured decoding with doubly-recurrent neural networks. — 2016.
94. *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.
95. *Feng Jiashi, Darrell Trevor.* Learning the structure of deep convolutional networks // Proceedings of the IEEE international conference on computer vision. — 2015. — Pp. 2749–2757.
96. *Shirakawa Shinichi, Iwata Yasushi, Akimoto Youhei.* Dynamic Optimization of Neural Network Structures Using Probabilistic Modeling // *arXiv preprint arXiv:1801.07650*. — 2018.
97. Toward Optimal Run Racing: Application to Deep Learning Calibration / Olivier Bousquet, Sylvain Gelly, Karol Kurach et al. // *arXiv preprint arXiv:1706.03199*. — 2017.
98. Learning deep resnet blocks sequentially using boosting theory / Furong Huang, Jordan Ash, John Langford, Robert Schapire // *arXiv preprint arXiv:1706.04964*. — 2017.

99. Progressive neural architecture search / Chenxi Liu, Barret Zoph, Jonathon Shlens et al. // *arXiv preprint arXiv:1712.00559*. — 2017.
100. *Alain Guillaume, Bengio Yoshua*. Understanding intermediate layers using linear classifier probes // *arXiv preprint arXiv:1610.01644*. — 2016.
101. *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.
102. Incremental Training of Deep Convolutional Neural Networks / R Istrate12, ACI Malossi, C Bekas, D Nikolopoulos.
103. *Chen Tianqi, Goodfellow Ian, Shlens Jonathon*. Net2net: Accelerating learning via knowledge transfer // *arXiv preprint arXiv:1511.05641*. — 2015.
104. 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.
105. *Miranda Conrado S, Von Zuben Fernando J*. Reducing the Training Time of Neural Networks by Partitioning // *arXiv preprint arXiv:1511.02954*. — 2015.
106. *Schmidhuber Juergen, Zhao Jieyu, Wiering MA*. Simple principles of metalearning // *Technical report IDSIA*. — 1996. — Vol. 69. — Pp. 1–23.
107. *Schmidhuber Jürgen*. A neural network that embeds its own meta-levels // Neural Networks, 1993., IEEE International Conference on / IEEE. — 1993. — Pp. 407–412.
108. Meta-SGD: Learning to Learn Quickly for Few Shot Learning / Zhenguo Li, Fengwei Zhou, Fei Chen, Hang Li // *arXiv preprint arXiv:1707.09835*. — 2017.
109. *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.
110. 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.
111. *Kinga D, Adam J Ba*. A method for stochastic optimization // International Conference on Learning Representations (ICLR). — Vol. 5. — 2015.
112. *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.
113. *Friesen Abram L, Domingos Pedro*. Deep Learning as a Mixed Convex-Combinatorial Optimization Problem // *arXiv preprint arXiv:1710.11573*. — 2017.
114. *Kristiansen Gus, Gonzalvo Xavi*. EnergyNet: Energy-based Adaptive Structural Learning of Artificial Neural Network Architectures // *arXiv preprint arXiv:1711.03130*. — 2017.

115. Pathnet: Evolution channels gradient descent in super neural networks / Chrisantha Fernando, Dylan Banarse, Charles Blundell et al. // *arXiv preprint arXiv:1701.08734*. — 2017.
116. *Veniat Tom, Denoyer Ludovic*. Learning time-efficient deep architectures with budgeted super networks // *arXiv preprint arXiv:1706.00046*. — 2017.
117. 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.
118. *Nalisnick Eric, Smyth Padhraic*. Deep Generative Models with Stick-Breaking Priors // *arXiv preprint arXiv:1605.06197*. — 2016.
119. *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.
120. *Miller A. C., Foti N., Adams R. P.* Variational Boosting: Iteratively Refining Posterior Approximations // *ArXiv e-prints*. — 2016. — nov.
121. *Arnold Ludovic, Ollivier Yann*. Layer-wise learning of deep generative models // *arXiv preprint arXiv:1212.1524*. — 2012.
122. *Maddison Chris J, Mnih Andriy, Teh Yee Whye*. The concrete distribution: A continuous relaxation of discrete random variables // *arXiv preprint arXiv:1611.00712*. — 2016.
123. On some variance reduction properties of the reparameterization trick / Ming Xu, Matias Quiroz, Robert Kohn, Scott A Sisson // *arXiv preprint arXiv:1809.10330*. — 2018.
124. The Reparameterization Trick. <http://gregorygundersen.com/blog/2018/04/29/reparamete>
125. *Hinton Geoffrey, Van Camp Drew*. Keeping neural networks simple by minimizing the description length of the weights // in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory / Citeseer. — 1993.