М. С. Потанин, К. О. Вайсер, В. А. Жолобов, В. В. Стрижов, Оптимизация структуры сетей глубокого обучения, *Информ. и её примен.*, 2020, том 14, выпуск 4, 55–62

DOI: 10.14357/19922264200408

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2020. Т. 14. Вып. 4. С. 55–62

ОПТИМИЗАЦИЯ СТРУКТУРЫ СЕТЕЙ ГЛУБОКОГО ОБУЧЕНИЯ*

М. С. Потанин[1], К. О. Вайсер[2], В. А. Жолобов[3], В. В. Стрижов[4]

Аннотация: Исследуется проблема выбора оптимальной структуры модели. Моделью служит суперпозиция обобщенных линейных моделей, элементами которой являются линейная регрессия, логистическая регрессия, метод главных компонент, автоэнкодер и нейросеть. Под структурой модели понимаются значения структурных параметров модели, задающих вид итоговой суперпозиции. Исследуется свойства алгоритма выбора структуры модели. Исследуется зависимость точности, сложности и устойчивости модели от способа задания структуры. Создан алгоритм выбора оптимальной структуры нейронной сети. Проведен вычислительный эксперимент с использованием реальных и синтетических данных. В результате эксперимента существенно снижена структурная сложность моделей с сохранением точности аппроксимации.

Ключевые слова: выбор моделей; линейные модели; автокодировщик; нейронные сети; структура; генетический алгоритм

DOI: 10.14357/19922264200408

1 Введение

Решается задача аппроксимации выборки нейронными сетями. Нейронная сеть служит универсальной моделью [1, 2], так как приближает произвольную непрерывную функцию многих переменных с любой точностью. Нейрон, или однослойная нейронная сеть, представляет собой суперпозицию двух функций — функции активации и линейной комбинации признаков объекта. Но однослойные сети применимы только для линейно разделимых выборок. Для аппроксимации выборок общего вида требуется универсальная модель, оптимизация структуры которой и исследуется в данной работе.

Теорема 1 (Колмогоров, 1961) в [3] утверждает, что функция от n аргументов представима в виде комбинации $n(2n+1)$ функций одного аргумента. Какими именно должны быть функции σ_i и g_{ij}, не указывается. Теорема об универсальной аппроксимации 2 (Цыбенко, 1989) в [3] утверждает, что искусственная нейронная сеть прямой связи, в которой связи не образуют циклов, с одним скрытым слоем аппроксимирует любую непрерывную функцию многих переменных с любой точностью. Однако затруднительно выбрать такую структуру нейронной сети, чтобы размеры скрытого слоя не были велики. В теореме 3 (Ханин, 2017) оценивается оптимальная размерность скрытых слоев и обосновывается возможность замены нейронной сети с функциями активации ReLU [3] с входным слоем размерности n и одним скрытым слоем размерности k на эквивалентную с глубиной $k+2$ и размерностями скрытых слоев $n+2$. Эти три теоремы и определяют исследуемую структуру суперпозиций сети глубокого обучения.

Исследуется зависимость ошибки от суперпозиции автокодировщиков [4] и многослойной нейронной сети. Ошибка состоит из двух слагаемых: ошибки восстановления элементов выборки после кодирования и восстановления зависимых переменных. Слагаемые используют одни и те же признаки объектов, которые являются независимыми переменными, но разные зависимые переменные. Для автокодировщика зависимые переменные — это сами признаки объекта, для нейронной сети, следующей за ним, зависимая переменная — ответ y на объекте. Точка разделения — это место в суперпозиции, где автокодировщик, имеющий оптимальные параметры, передает преобразован-

*Работа выполнена при поддержке РФФИ (проекты 19-07-1155, 19-07-0885) и правительства РФ (соглашение 05.Y09.21.0018). Настоящая статья содержит результаты проекта «Статистические методы машинного обучения», выполняемого в рамках реализации Программы Центра компетенций Национальной технологической инициативы «Центр хранения и анализа больших данных», поддерживаемого Министерством науки и высшего образования Российской Федерации по договору МГУ имени М. В. Ломоносова с Фондом поддержки проектов Национальной технологической инициативы от 11.12.2018 № 13/1251/2018.

[1]Московский физико-технический институт, mark.potanin@phystech.edu

[2]Московский физико-технический институт, vajser.ko@phystech.edu

[3]Московский физико-технический институт, zholobov.va@phystech.edu

[4]Вычислительный центр имени А. А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук; Московский физико-технический институт, strijov@ccas.ru

ный вектор признаков в нейросеть. Необходимо найти оптимальное расположение разделения автокодировщика и сети, которое минимизирует ошибку аппроксимации выборки. Под структурой такой модели понимаются величины, задающие вид итоговой суперпозиции, т. е. число слоев автокодировщика и нейросети, а также число нейронов в слоях. Процедура минимизации ошибки аппроксимации выборки следующая: сначала максимизируется точность реконструкции кодировщиков, затем оптимизируются параметры нейросети. Вместе с точностью оптимизируется сложность модели. Под сложностью понимается структурная сложность модели — число параметров модели.

В данном исследовании для выбора оптимальной структуры используется генетический алгоритм. Задается множество случайных начальных значений структурных параметров. Затем вычисляется значение функции ошибки аппроксимации, которое характеризует качество модели в наборе. Согласно этой функции выбираются модели, которые обмениваются структурными параметрами, образуя новую структуру. Многократное повторение этой операции позволяет получить оптимальную структуру модели.

Алгоритмы прореживания OBD (optimal brain damage) [5] и OBS (optimal brain surgeon) [6] используют производные второго порядка функции ошибки по параметрам для выбора удаляемых параметров. В [7] авторы предлагают новый метод прореживания для глубоких нейронных сетей. Параметры каждого слоя независимо прореживаются на основе производных второго порядка функции послойной ошибки по соответствующим параметрам. В [8] используются производные первого порядка для снижения сложности сверточных нейронных сетей. Использование [5] позволило в [9] уменьшить число структурных параметров рекуррентной нейронной сети на 60% и снизить ошибку на валидационной выборке на 30% по сравнению с исходной моделью. Автоматизированные методы поиска нейросетевой архитектуры [10] являются частью парадигмы автоматического машинного обучения [11]. Система поиска получает на вход набор данных и тип решаемой задачи. Результат — оптимизированная архитектура нейронной сети.

2 Постановка задачи выбора модели

Задана выборка $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}^1$, $i = 1, \ldots, m$, где $\mathbf{x}$ — описание объекта, вектор из n элементов признаков; y — зависимая переменная. Моделью называется отображение $f : (\mathbf{x}, \mathbf{w}) \mapsto y$. Требуется построить аппроксимирующую модель $f(\mathbf{x})$ вида:

$$f = \sigma_k \circ \underset{1\times 1}{\mathbf{w}_k^{\mathsf{T}}} \boldsymbol{\sigma}_{k-1} \circ \mathbf{W}_{k-1}\boldsymbol{\sigma}_{k-2} \circ \cdots \cdots \circ \underset{n_2\times 1}{\mathbf{W}_2}\boldsymbol{\sigma}_1 \circ \underset{n_1\times n}{\mathbf{W}_1}\underset{n\times 1}{\mathbf{x}}. \quad (1)$$

Эта модель рассматривается как суперпозиция линейной модели, глубокой нейросети и автоэнкодера. Рассмотрим различные модели как частные случаи (1). Линейная, или логистическая, регрессия и один нейрон имеют вид $f(\mathbf{x}, \mathbf{w}) = \boldsymbol{\sigma}(\mathbf{w}^{\mathsf{T}}\mathbf{x})$, где $\boldsymbol{\sigma}$ — функция активации, непрерывная монотонная дифференцируемая функция (2); $\mathbf{w}$ — вектор параметров; $\mathbf{x}$ — объект, вектор с присоединенным элементом единица, соответствующим аддитивному параметру w_0. При использовании линейной функции активации получаем линейную регрессию $f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{x}$.

Такую функцию активации обозначим $\boldsymbol{\sigma} = \mathrm{id}$. При использовании сигмоидной функции активации получаем модель логистической регрессии:

$$f(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^{\mathsf{T}}\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^{\mathsf{T}}\mathbf{x})}. \quad (2)$$

Двухслойная нейронная сеть, состоящая из линейной комбинации нейронов, однослойных нейронных сетей:

$$f(\mathbf{x}, \mathbf{w}) = \sigma^{(2)}\left(\sum_{i=1}^{n_2} w_i^{(2)}\sigma^{(1)}\left(\sum_{j=1}^{n} w_{ij}^{(1)}x_j + w_{i0}^{(1)}\right) + w_0^{(2)}\right) = \sigma \circ \mathbf{w}^{\mathsf{T}}\boldsymbol{\sigma} \circ \mathbf{W}\mathbf{x}.$$

Метод главных компонент. Модель допускает вращения признакового пространства, т. е. объекты преобразуются только с помощью поворотов, $\mathbf{h} = \mathbf{W}\mathbf{x}$, где $\mathbf{W}$ — матрица поворота. Она ортогональна: $\mathbf{W}\mathbf{W}^{\mathsf{T}} = \mathbf{I}_n$. Полученное пространство образов $\mathbf{h}$ называется скрытым. Происходит преобразование без потерь.

При удалении нескольких строк оптимальной [12] матрицы $\mathbf{W}$, например их число $u < n$, полученный вектор $\mathbf{h}$ имеет размер $u \times 1$. Получается проекция $\mathbf{h}$ вектора $\mathbf{x}$. Согласно теореме С. Р. Рао [12], первые u главных компонент восстанавливают $\mathbf{h}$ оптимальным способом, $\mathbf{r}(\mathbf{x}) = \mathbf{W}^{\mathsf{T}}\mathbf{h}$.

Автокодировщик $\mathbf{h}$ — это монотонное нелинейное отображение входного вектора свободных переменных $\mathbf{x} \in \mathbb{R}^n$ в скрытое представление $\mathbf{h} \in \mathbb{R}^u$ вида:

$$\mathbf{h}(\mathbf{x}) = \boldsymbol{\sigma}(\underset{u\times n}{\mathbf{W}}\mathbf{x} + \mathbf{b}).$$

В случае $\boldsymbol{\sigma} = \mathbf{id}$ и ортогональной матрицы $\mathbf{W}$ автокодировщик тождествен методу главных компонент. Скрытое представление $\mathbf{h}$ реконструирует вектор $\mathbf{x}$ линейно:

$$\underset{n\times u}{\mathbf{r}(\mathbf{x})} = \underset{n\times u}{\mathbf{W}'}\mathbf{h} + \mathbf{w}_0'.$$

3 Задача выбора оптимальной структуры модели

Решается задача выбора оптимальной структуры модели

$$f = \sigma_k \circ \boldsymbol{\Gamma}_k \otimes \underset{1\times 1}{\mathbf{w}_k^{\mathsf{T}}}\boldsymbol{\sigma}_{k-1} \circ \boldsymbol{\Gamma}_{k-1} \otimes \mathbf{W}_{k-1}\boldsymbol{\sigma}_{k-2} \circ \cdots$$
$$\cdots \circ \boldsymbol{\Gamma}_2 \otimes \underset{n_2\times 1}{\mathbf{W}_2\boldsymbol{\sigma}_1} \circ \boldsymbol{\Gamma}_1 \otimes \underset{n_1\times n^{n\times 1}}{\mathbf{W}_1\ \mathbf{x}}, \quad (3)$$

где $\boldsymbol{\Gamma}$ — матрица, задающая структуру модели; $\otimes$ — адамарово произведение, определяемое как поэлементное умножение. Если элемент $\gamma \in \{0, 1\}$ матрицы $\boldsymbol{\Gamma}$ равен нулю, то соответствующий элемент матрицы параметров $\mathbf{W}$ обнуляется и не участвует в работе модели. Множество индексов, соответствующих ненулевым элементам матрицы $\boldsymbol{\Gamma}$, обозначается $\mathcal{A}$. Требуется найти такое подмножество индексов $\mathcal{A}^*$, которое доставляет минимум функции

$$\mathcal{A}^* = \arg\min_{\mathcal{A}\subseteq\mathcal{I}} S(f_{\mathcal{A}}|\mathbf{w}^*, \mathfrak{D}_{\mathcal{C}}) \quad (4)$$

на разбиении выборки $\mathfrak{D}$, определенной множеством индексов $\mathcal{C}$. Здесь $\mathcal{I} = \mathcal{C} \sqcup \mathcal{L}$ — все индексы всех матриц $\boldsymbol{\Gamma}$. Таким образом, требуется снизить число признаков и повысить устойчивость модели. При этом параметры $\mathbf{w}^*$ модели доставляют минимум ошибки

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} S(\mathbf{w}|\mathfrak{D}_{\mathcal{L}}, f_{\mathcal{A}}) \quad (5)$$

на разбиении выборки, определенной множеством $\mathcal{L}$. Процедура разбиения описана в вычислительном эксперименте.

Генетический алгоритм. Для решения задачи оптимизации структуры (4) используется генетический алгоритм. Структура нейронной сети (3) включает в себя k слоев, l-й слой содержит N_l нейронов, $\sum_{l=1}^{k} N_l = L$. Каждому слою соответствует матрица $\boldsymbol{\Gamma}_l \in \{0, 1\}^{N_l}$. Это означает, что параметры, которые умножаются поэлементно на ноль, не будут учитываться. Составляется вектор

$$\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_L] = \mathbf{vec}[\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2, \ldots, \boldsymbol{\Gamma}_k],$$

соответствующий (3). Процедура оптимизации структуры:

1. Задается множество начальных значений $\mathfrak{G} = \{\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_R\}$, где случайным образом задаются элементы бинарного вектора $\boldsymbol{\gamma}$.
2. Для каждого $\boldsymbol{\gamma}_\iota \in\in \mathfrak{G}$ вычисляется значение функции ошибки S (7) (см. разд. 4).
3. Для каждого $\boldsymbol{\gamma}_\iota$ оценивается вероятность выбора его как структуры для скрещивания с помощью функции

$$P_\iota = \frac{1/S_\iota}{\sum_{\iota=1} 1/S_\iota}.$$

   Выбирается пара структур $\boldsymbol{\gamma}_p, \boldsymbol{\gamma}_q$ с максимальной вероятностью.
4. Выбирается случайный индекс точки разделения $\nu \in \{1, \ldots, L-1\}$.
5. Структуры разделяются на две части, происходит обмен элементами, следующими за ν:

$$[\gamma_{p,1}, \ldots, \gamma_{p,\nu}, \gamma_{q,\nu+1}, \ldots, \gamma_{q,L}] \to \boldsymbol{\gamma}_p',$$
$$[\gamma_{q,1}, \ldots, \gamma_{q,\nu}, \gamma_{p,\nu+1}, \ldots, \gamma_{p,L}] \to \boldsymbol{\gamma}_q'.$$

6. Выбираются случайные номера $\eta_1, \ldots, \eta_Q \in \in \{1, \ldots, L\}$.
7. У векторов $\boldsymbol{\gamma}_p', \boldsymbol{\gamma}_q'$ инвертируются позиции с номерами $\eta_1, \ldots, \eta_Q$.
8. Пункты 4–8 повторяются $R/2$ раз. Множество $\mathfrak{G}$ содержит на каждой итерации R структур, которым соответствует наименьшее ошибка.

Здесь R и Q — фиксированные параметры алгоритма. В эксперименте производится настройка $\boldsymbol{\Gamma}$ по частям, т. е. алгоритм запускается отдельно для каждого слоя. Результатом работы становится вектор, нулевые элементы которого соответствуют нейронам, исключаемым из структуры.

4 Функция ошибки и критерии качества модели

Для оптимизации структуры предлагается использовать композитную функцию ошибки (7). Она состоит из двух слагаемых. Первое слагаемое соответствует точности восстановления зависимой переменной. Второе слагаемое — это точность реконструкции независимой переменной автокодировщиком. Задача (5) представляет собой задачу минимизации функции S. Она включает слагаемые (7) и (9) для оптимизации параметров модели (1):

$$f = \sigma_k \circ \underset{1\times 1}{\mathbf{\Gamma}_k \mathbf{w}_k^{\mathsf{T}}} \boldsymbol{\sigma}_{k-1} \circ \mathbf{\Gamma}_{k-1}\mathbf{W}_{k-1}\boldsymbol{\sigma}_{k-2} \circ \cdots \underbrace{\cdots \circ \underset{n_2\times 1}{\mathbf{\Gamma}_2}\mathbf{W}_2\boldsymbol{\sigma}_1 \circ \underset{n_1\times n}{\mathbf{\Gamma}_1}\mathbf{W}_1 \underset{n\times 1}{\mathbf{x}}}_{E_{\mathbf{x}}}. \quad (6)$$

S

Первое слагаемое $E_{\mathbf{x}}$ — это функция ошибки реконструкции объекта стеком автокодировщиков. Второе слагаемое S — это функция ошибки нейросети. При выборе моделей используются три вида критериев качества: точность, устойчивость и сложность.

Точность. В задаче восстановления регрессии функция ошибки имеет вид:

$$S = \sum_{i\in\mathcal{I}} \big(y_i - f(\mathbf{x}_i)\big)^2. \quad (7)$$

При включении в модель (1) метода главных компонент или автокодировщика метки объектов не используются. Функция ошибки штрафует невязки восстановленного объекта:

$$E_{\mathbf{x}} = \sum_{i\in\mathcal{I}} \|\mathbf{x}_i - \mathbf{r}(\mathbf{x}_i)\|_2^2, \quad (8)$$

где $\mathbf{r}(\mathbf{x})$ — линейная реконструкция объекта $\mathbf{x}$. Функция (8) с аддитивной регуляризацией:

$$E_{\mathbf{x}} = \sum_{i=1}^{m} \|\mathbf{r}(\mathbf{x}_i, \mathbf{W}_{\mathrm{AE}}) - \mathbf{x}_i\|^2 + \lambda^2 \|\mathbf{W}\|^2_{\mathrm{Frobenius}}, \quad (9)$$

где m — число элементов в обучающей выборке. Параметры автокодировщика $\mathbf{W}_{\mathrm{AE}} = \{\mathbf{W}', \mathbf{W}, \mathbf{b}', \mathbf{b}\}$ оптимизированы таким образом (8), чтобы приблизить реконструкцию $\mathbf{r}(\mathbf{x})$ к исходному вектору $\mathbf{x}$.

Процедура оптимизации параметров композитной функции (6):

(1) оптимизируются параметры модели согласно (9);

(2) заданные параметры фиксируются;

(3) оптимизируются параметры согласно (7).

Сложность — это число параметров модели, $\sum_{l=1}^{k} \|\mathbf{\Gamma}_k\|_1 \to \min$.

Устойчивость — это минимум дисперсии функции ошибки (7): $\mathsf{D}(S) \to \min$. При вычислении устойчивости выборка считается фиксированной и изменение устойчвости считается зависящим только от структуры и параметров модели.

5 Вычислительный эксперимент

Исследуется процедура оптимизации структуры нейросети с сохранением качества аппроксимации. Структура оптимизируется с помощью генетического алгоритма. Цель вычислительного эксперимента состоит в определении оптимальной позиции разделения автокодировщиков и нейронной сети, а также исследовании зависимости точности, сложности и устойчивости модели от способа задания структуры. Исходный код находится на Github [3].

Наборы данных. Качество предложенного подхода к построению модели оценивается на нескольких реальных наборах данных и одном синтетическом наборе. Выборки взяты из открытого репозитория данных для машинного обучения [13]. Описание всех выборок представлено в таблице. Синтетический набор данных состоит из признаков с различными свойствами ортогональности и коррелированности друг с другом и с целевой переменной. Процедура генерации синтетических данных описана в работе [14]. Возможны следующие конфигурации синтетических данных: неполный и скоррелированный; адекватный и случайный; адекватный и избыточный; адекватный и скоррелированный.

Каждый набор данных разбивается на три части. Обучающая выборка — 60% от исходного набора. На этой выборке модель тренируется и фиксируются значения параметров. Валидационная выборка — 20% от исходного набора. На этой выборке применяется генетический алгоритм, который ищет оптимальную структуру. Тестовая выборка — 20% от исходного набора. Она никак не участвует

Результат применения генетического алгоритма для прореживания сети

Выборка $\mathfrak{D}$	m	n	Ошибка сети с прореживанием	Ошибка сети без прореживания	Сложность без прореживания	Сложность после прореживания
Credit Card	30000	35	0,3204 ± 0,0032	0,2681 ± 0,0034	68	25
Protein	45730	9	4,4968 ± 0,0238	4,4968 ± 0,0238	16	1
Airbnb	10498	16	135,0773 ± 0,5909	33,9163 ± 0,5978	32	12
Wine quality	4898	11	0,5818 ± 0,0147	0,5941 ± 0,0149	20	4
Synthetic, 10^{-3}	2000	30	0,3005 ± 0,0081	0,303 ± 0,0079	60	12

в оптимизации структуры модели. Эта выборка используется только для контроля качества — сравнение модели исходной и оптимизированной структуры, а также сравнение с другими алгоритмами прореживания сетей. Решается задача восстановления регрессии, т. е. зависимой переменной служит $y \in \mathbb{R}$. В процессе работы были рассмотрены два подхода к решению задачи и, соответственно, две структуры нейронной сети.

Первый подход. Автокодировщик преобразует входные векторы $\mathbf{x}$, которые затем подаются на вход полносвязной нейронной сети.

Второй подход. Оптимизируются параметры автокодировщика, его параметры фиксируются и предпоследний слой соединяется с полносвязной нейросетью, оптимизируются параметры модели. Предпоследний слой содержит меньшее число параметров по сравнению с размерностью входного пространства признаков. Число парамеров определяет число нейронов в этом слое, т. е. полносвязная сеть получает на вход скрытое представление исходной независимой переменной.

Параметры в обеих сетях инициализированы нормальным распределением с нулевым средним и смещением $\sqrt{(1/2)(N_{\text{in}} + N_{\text{out}})}$, где N_{in} — число входных признаков; N_{out} — число признаков на выходе слоя. Такая инициализация параметров была предложена в [15]. Она выбрана экспериментальным путем как показывающая наилучший результат точности аппроксимации. Каждая из сетей обучалась в течение 500 итераций обновления параметров, и размер пакета обучения равен 128. В качестве функции активации в слоях автокодировщика используется $\mathrm{Relu}(x) = \max(0, x)$, для полносвязной сети тоже Relu, но в последнем слое id.

Для вычисления ошибки (7) используются выходные значения полносвязной сети. Варьируется число промежуточных слоев автокодировщика и полносвязной сети от 1 до 5. Рассматривается декартово произведение двух множеств: $[1, 2, 3, 4, 5] \times$ $\times\, [1, 2, 3, 4, 5]$. Число нейронов в каждом скрытом слое одинаково для любой сети и равно десяти. Для каждой конфигурации считается ошибка (7). Качество оценивается на синтетическом наборе данных. Полученный результат представлен на рис. 1. Размер пузыря пропорционален полученной ошибке. Каждая конфигурация сети обучалась на наборе, полученном с помощью бутстреп-метода из данных, взятых для обучения. Число итераций процедуры бутстреп-метода равно 10. Ошибка считалась на отложенном наборе данных для тестирования. Видно, что при увеличении числа слоев полносвязной сети ошибка в основном падает. Минимальная ошибка достигается при конфигурации: четыре слоя автокодировщика и два слоя полносвязной сети для первого подхода; один слой автокодировщика и четыре слоя полносвязной сети для второго подхода. Данную конфигурацию возьмем для дальнейшего исследования параметров модели, используя, соответственно, второй подход.

На рис. 2 представлена дисперсия и значение ошибки (7) в зависимости от числа слоев в автоко-

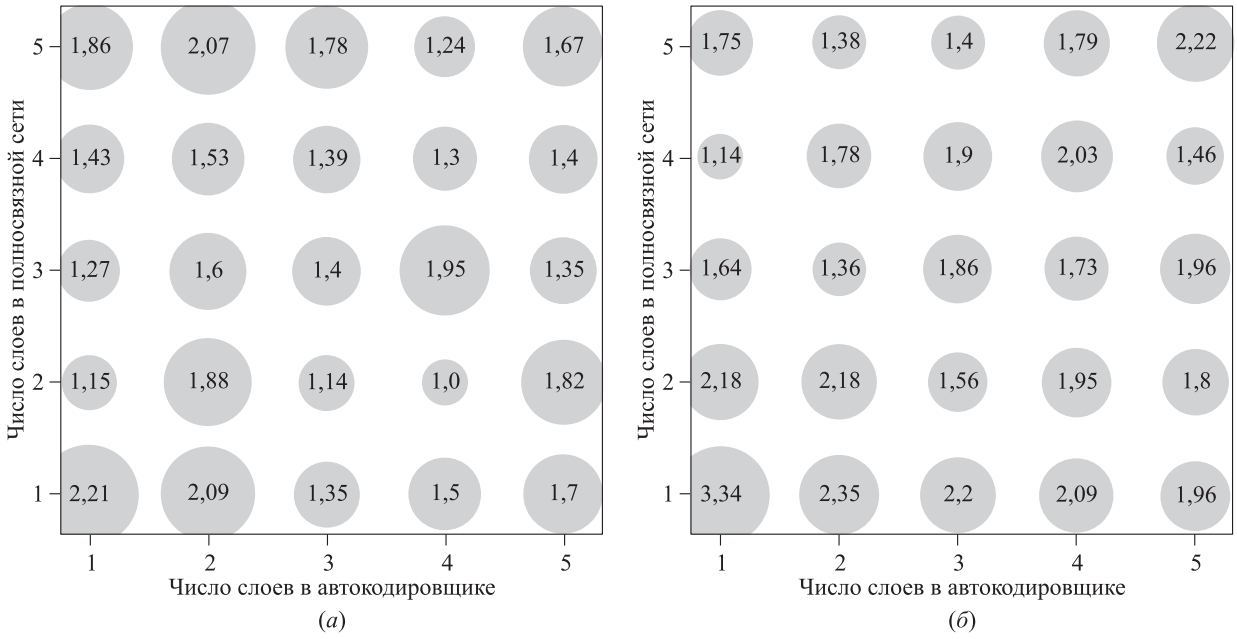

Рис. 1 Ошибка (7) в зависимости от конфигурации модели: (*а*) первый подход; (*б*) второй подход

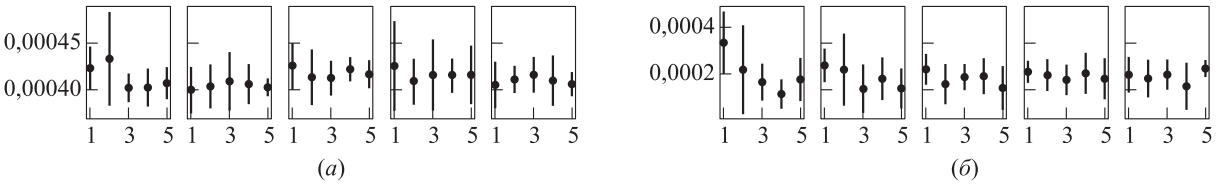

Рис. 2 Ошибка и ее дисперсия в зависимости от структуры модели: (*а*) первый подход; (*б*) второй подход. Порядковый номер графика и точки на графике соответствуют разному числу слоев соответствующей модели

дировщике и полносвязной сети. Дисперсия была получена с помощью десяти итераций бутстрепирования обучающей выборки для каждой конфигурации. С увеличением числа слоев в автокодировщике при использовании первого подхода снижаются ошибка и дисперсия ошибки.

С помощью описанных двух подходов получается оптимальная архитектура соединения автокодировщика и полносвязной сети. Далее применяется описанный ранее генетический алгоритм для прореживания сети и уменьшения ее сложности. Алгоритм применяется на нейронах каждого слоя сети. В таблице приведены результаты применения генетического алгоритма для исследуемых наборов данных, качество алгоритма оценивается по тестовой выборке. В качестве ошибки выступает $\mathrm{MAE} = (1/m)\sum_{i=1}^{m} |y_i - f(\mathbf{x}_i)|$, а в качестве сложности алгоритма выступает число ненулевых нейронов. Под нулевым нейроном понимается нейрон, все параметры которого равны нулю.

6 Заключение

В представленной работе исследованы два подхода к построению модели, состоящей из автокодировщика и нейронной сети и имеющей композитную функцию ошибки. Представлены подходы к поиску оптимальной точки разделения автокодировщика и нейронной сети. Исследовано применение генетического алгоритма для оптимизации структуры и снижения сложности. Работа предложенного алгоритма исследовалась на пяти различных наборах данных. Как показано в таблице, предложенный алгоритм выбора структуры существенно снижает сложность модели без потери качества аппроксимации.

Литература

1. *Cybenko G. V.* Approximation by superpositions of a sigmoidal function // Math. Control Signal., 1989. Vol. 2. Iss. 4. P. 303–314.
2. *Бахтеев О. Ю., Стрижов В. В.* Выбор моделей глубокого обучения субоптимальной сложности // Автоматика и телемеханика, 2018. Вып. 8. С. 129–147.
3. *Потанин М. С., Вайсер К. О., Жолобов В. А., Стрижов В. В.* Приложение к статье: вычислительный эксперимент по выбору универсальной модели и базовые теоремы, 2019. https://github.com/MarkPotanin/GeneticOpt.
4. *Hinton G. E., Salakhutdinov R. R.* Reducing the dimensionality of data with neural networks // Science, 2006. Vol. 313. Iss. 5786. P. 504–507.
5. *LeCun Y., Denker J. S., Solla S., Howard R. E., Jackel L. D.* Optimal brain damage // Adv. Neur. In., 1989. Vol. 2. P. 598–605.
6. *Hassibi B., Stork D. G.* Second order derivatives for network pruning: Optimal brain surgeon // Adv. Neur. In., 1992. Vol. 5. P. 164–171.
7. *Dong X., Chen S., Pan S.* Learning to prune deep neural networks via layer-wise optimal brain surgeon // Adv. Neur. In., 2017. Vol. 30. P. 4857–4867.
8. *Molchanov P., Tyree S., Karras T., Aila T., Kautz J.* Pruning convolutional neural networks for resource efficient transfer learning // ArXiv.org, 2016. ArXiv:1611.06440.
9. *Chaber P., Lawrynczuk M.* Pruning of recurrent neural models: An optimal brain damage approach // Nonlinear Dynam., 2018. Vol. 92. Iss. 2. P. 763.
10. *Elsken T., Metzen J. H., Hutter F.* Neural architecture search: A survey // ArXiv.org, 2018. ArXiv:1808.05377.
11. *Hutter F., Kotthoff L., Vanschoren J.* Automated machine learning – methods, systems, challenges. — Springer, 2019. 223 p.
12. *Pao C. P.* Линейные статистические методы и их применение / Пер. с англ. — М.: Наука, 1968. 548 с. (*Rao C. R.* Linear statistical inference and its application. — New York, NY, USA: Wiley, 1968. 548 p.)
13. UCI Machine Learning Repository, 2007. https://archive.ics.uci.edu/ml.
14. *Katrusa A. M., Strijov V. V.* Stress test procedure for feature selection algorithms // Chemometr. Intell. Lab., 2015. Vol. 142. P. 172–183.
15. *Glorot X., Bengio Y.* Understanding the difficulty of training deep feedforward neural networks // 13th Conference (International) on Artificial Intelligence and Statistics Proceedings. — Sardinia, Italy, 2010. P. 249–256.

Поступила в редакцию 02.12.19

DEEP LEARNING NEURAL NETWORK STRUCTURE OPTIMIZATION

M. S. Potanin[1], K. O. Vayser[1], V. A. Zholobov[1], and V. V. Strijov[1,2]

[1]Moscow Institute of Physics and Technology, 9 Institutskiy Per., Dolgoprudny, Moscow Region 141700, Russian Federation

[2]A. A. Dorodnicyn Computing Center, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 40 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The paper investigates the optimal model structure selection problem. The model is a superposition of generalized linear models. Its elements are linear regression, logistic regression, principal components analysis, autoencoder and neural network. The model structure refers to values of structural parameters that determine the form of final superposition. This paper analyzes the model structure selection method and investigates dependence of accuracy, complexity and stability of the model on it. The paper proposes an algorithm for selection of the neural network optimal structure. The proposed method was tested on real and synthetic data. The experiment resulted in significant structural complexity reduction of the model while maintaining accuracy of approximation.

Keywords: model selection; linear models; autoencoders; neural networks; structure; genetic alghorithm

DOI: 10.14357/19922264200408

Acknowledgments

This research was supported by RFBR (projects 19-07-1155 and 19-07-0885) and by the Government of the Russian Federation (agreement 05.Y09.21.0018). This paper contains results of the project "Statistical methods of machine learning" which is carried out within the framework of the Program "Center of Big Data Storage and Analysis" of the National Technology Initiative Competence Center. It is supported by the Ministry of Science and Higher Education of the Russian Federation according to the agreement between the M. V. Lomonosov Moscow State University and the Foundation of Project Support of the National Technology Initiative from 11.12.2018 No. 13/1251/2018.

References

1. Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function. *Math. Control Signal.* 2(4):303–314.
2. Bakhteev, O. Yu, and V. V. Strijov. 2018. Deep learning model selection of suboptimal complexity. *Automat. Rem. Contr.* 79(8):1474–1488.
3. Potanin, M. S., K. O. Vajser, V. A. Zholobov, and V. V. Strijov. 2019. Prilozhenie k stat'e: vychislitel'nyy eksperiment po vyboru universal'noy modeli i bazovye teoremy [Appendix to the paper: Computational expirement and basic theorems]. Available at: https://github.com/MarkPotanin/GeneticOpt (accessed November 5, 2020).
4. Hinton, G. E., and R. R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507.
5. LeCun, Y., J. Denker, and S. Solla. 1989. Optimal brain damage. *Adv. Neur. In.* 2:598–605.
6. Hassibi, B., and D. G. Stork. 1992. Second order derivatives for network pruning: Optimal brain surgeon. *Adv. Neur. In.* 5:164–171.
7. Dong, X., S. Chen, and S. Pan. 2017. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Adv. Neur. In.* 30:4857–4867.
8. Molchanov, P., S. Tyree, T. Karras, T. Aila, and J. Kautz. 2016. Pruning convolutional neural networks for resource efficient transfer learning. arXiv:1611.06440 [cs.LG]. Available at: https://arxiv.org/abs/1611.06440 (accessed November 5, 2020).
9. Chaber, P., and M. Lawryńczuk. 2018. Pruning of recurrent neural models: an optimal brain damage approach. *Nonlinear Dynam.* 92(2):763–780.
10. Elsken, T., J. H. Metzen, and F. Hutter. 2018. Neural architecture search: A survey. arXiv:1808.05377 [stat.ML]. Available at: https://arxiv.org/abs/1808.05377 (accessed November 5, 2020).
11. Hutter, F., L. Kotthoff, and J. Vanschoren. 2019. *Automated machine learning-methods, systems, challenges*. Springer. 223 p.
12. Rao, C. R. 1973. *Linear statistical inference and its applications*. Vol. 2. New York, NY: Wiley. 548 p.
13. UCI Machine Learning Repository. Available at: http://archive.ics.uci.edu/ml (accessed November 5, 2020).
14. Katrutsa, A. M., and V. V. Strijov. 2015. Stress test procedure for feature selection algorithms. *Chemometr. Intell. Lab.* 142:172–183.
15. Glorot, X., and Yo. Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. *13th Conference (Internatioanl) on Artificial Intelligence and Statistics Proceedings.* 249—256.

Received December 2, 2019

Contributors

Potanin Mark St. (b. 1994) — PhD student, Moscow Institute of Physics and Technology, 9 Institutskiy Per., Dolgoprudny, Moscow Region 141701, Russian Federation; mark.potanin@phystech.edu

Vayser Kirill O. (b. 2000) — student, Moscow Institute of Physics and Technology, 9 Institutskiy Per., Dolgoprudny, Moscow Region 141701, Russian Federation; vajser.ko@phystech.edu

Zholobov Vladimir Al. (b. 1998) — student, Moscow Institute of Physics and Technology, 9 Institutskiy Per., Dolgoprudny, Moscow Region 141701, Russian Federation; zholobov.va@phystech.edu

Strijov Vadim V. (b. 1967) — Doctor of Science in physics and mathematics, leading scientist, A. A. Dorodnicyn Computing Center, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 40 Vavilov Str., Moscow 119333, Russian Federation; professor, Moscow Institute of Physics and Technology, 9 Institutskiy Per., Dolgoprudny, Moscow Region 141701, Russian Federation; strijov@phystech.edu