

# Возможности для реализации системы автоматного обучения

В. А. Бирюкова<sup>1</sup>, Г. В. Боков<sup>2</sup>, А. С. Дробышев<sup>3</sup>, Г. В. Калачев<sup>4</sup>,  
В. С. Половников<sup>5</sup>, Д. В. Ронжин<sup>6</sup>, А. А. Часовских<sup>7</sup>

---

<sup>1</sup>*Бирюкова Вероника Андреевна* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: biryukovaveronika@mail.ru

Biryukova Veronika Andreevna — Postgraduate, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

<sup>2</sup>*Боков Григорий Владимирович* — кандидат физико-математических наук, доцент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: bokov@intsys.msu.ru

Bokov Grigoriy Vladimirovich — Candidate of Physical and Mathematical Science, Associate Professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

<sup>3</sup>*Дробышев Александр Сергеевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: drobyshev.sanya@yandex.ru

Drobyshev Alexander Sergeevich — Postgraduate, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

<sup>4</sup>*Калачев Глеб Вячеславович* — кандидат физико-математических наук, младший научный сотрудник каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: gleb.kalachev@yandex.ru

Kalachev Gleb Vyacheslavovich — Candidate of Physical and Mathematical Science, Junior Researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

<sup>5</sup>*Половников Владимир Сергеевич* — кандидат физико-математических наук, научный сотрудник каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: pvser@mail.ru

Polovnikov Vladimir Sergeevich — Candidate of Physical and Mathematical Science, Researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

<sup>6</sup>*Ронжин Дмитрий Владимирович* — кандидат физико-математических наук, младший научный сотрудник каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: d\_rongin@mail.ru

Ronzhin Dmitry Vladimirovich — Candidate of Physical and Mathematical Science, Junior Researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

<sup>7</sup>*Часовских Анатолий Александрович* — доктор физико-математических наук доцент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: chasovskikh@mail.ru.

Chasovskikh Anatoly Alexandrovich — Doctor of Physical and Mathematical Sciences, Associate Professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

Приведены результаты анализа некоторых доступных источников, на основе которого была реализована автоматная модель обучения искусственных нейронных сетей.

**Ключевые слова:** распознавание изображений, искусственная нейронная сеть, автоматизированное машинное обучение, автоматная модель обучения, гиперпараметры.

В этой заметке приведены результаты анализа доступных источников, на основе которого была реализована автоматная модель обучения искусственных нейронных сетей, представленная в работе [1].

## 1. Введение.

Для эффективного обучения искусственных нейронных сетей (ИНС) следует решить ряд задач: подобрать данные для обучения и тестирования, определить архитектуру ИНС, выполнить предобработку данных, задать метрики эффективности получаемых нейросетевых решений, выбрать параметры обучения. Только после этого обучение ИНС может быть начато. Неудовлетворительные результаты тестирования обученной модели ИНС приводят к изменению начальных установок, перечисленных выше, а затем, к возобновлению процесса обучения. Количество подходов к обучению ИНС зависит, естественно, от ряда причин, но, как правило, процесс обучения занимает значительное время у разработчика, особенно у не имеющего соответствующего опыта.

Возможность обучения ИНС распознаванию образов во многом предопределяется наличием в открытом доступе баз данных (БД). Эти базы, если и не позволяют перейти к обучению ИНС поставленной прикладной задаче, то, как правило, помогают получить предварительно обученную сеть, способную выделять необходимые признаки из обрабатываемых данных. Второй этап обучения предварительно обученной ИНС выполняется, как правило, на собранной пользователем специализированной БД. Мы приведем здесь краткое описание некоторых известных БД для анализа изображений и реализованный нами подход к унификации разметок, используемых в разных БД.

Для оптимизации процесса обучения ИНС используются методы AutoML, получившие в настоящее время широкое распространение. В этой работе мы приводим некоторые из этих методов, рассмотренных в [2].

Задание правильных гиперпараметров ИНС и алгоритма ее обучения является важнейшим условием получения удовлетворительного обучения модели, редко получается у не искушенного разработчика. Естественным желанием поэтому является использование автоматического

задания гиперпараметров. В этой работе приведены некоторые подходы к решению этой задачи. Для реализации собственной системы, как уже было сказано в [1], мы использовали градиентный метод поиска по сетке и метод «лучших практик».

## 2. Некоторые известные базы изображений и унификация разметки.

Разработанная нами система автоматного обучения ИНС распознаванию визуальных образов использует накопленные массивы изображений, примеры которых мы приведем далее. Начнем с наиболее часто встречающихся в статьях, являющихся эталонными, универсальными:

- ImageNet [3] Используется для решения задачи классификации объектов. Содержит по 1300 изображений объектов из 1000 классов. Свободно распространяются сети, прошедшие предварительное обучение на этой БД. Ввиду большого количества классов, обучение на ImageNet позволяет сети выделять значительный объем признаков.
- PascalVOC [4] Содержит изображения объектов из 20 классов. Изображения размечены для решения задач детектирования и сегментации, для сегментации разметка достаточно.
- COCO [5] Универсальная база данных для решения задач распознавания визуальных образов объектов из 80 классов. Каждое изображение содержит, как правило, несколько объектов. Изображения размечены для решения задач классификации, детектирования, сегментации, поиска ключевых точек. Далее приведем примеры БД изображений, специализированных по тематике решаемой задачи или ориентированной на определенный устоявшийся алгоритм решения некоторой задачи:
- Kaggle Cat and Dog images for Classification Содержит 25000 изображений кошек и собак для решения задачи бинарной классификации заданных классов объектов.
- KITTI [6] База данных сегментации вида из автомобиля, построения карты глубины и детектирования машин двумерными и трехмерными охватывающими прямоугольниками и параллелепипедами соответственно. База может использоваться для обучения системы управления самодвижущим автомобилем.

- [7] Содержит изображения городских ландшафтов и карты их сегментации. Может использоваться для решения задач интеллектуального управления транспортным потоком.
- FDDB: Face Detection Data Set and Benchmark [8], Wider Face [9] Эти базы снимков, содержащие изображения, которые используются для обучения ИНС обнаруживать лица. Вторая из них включает 32 203 изображения, которые содержат 393 703 фрагментов с лицами.
- Menpo Dataset [10] База данных с изображениями лиц с возможностью выделения на лицах ключевых точек. Эта БД может использоваться для решения задачи выравнивания изображения лица, как предварительного этапа их распознавания.  
Наконец, приведем пример базы, используемой для решения специализированных задач детектирования и классификации.
- Russian Traffic Sign [11] Содержит 179138 размеченных снимков, 156 классов объектов, 15630 изображений объектов (для классификации), 104358 включений объектов (для детектирования).

Для реализации системы автоматного обучения нами были отобраны 3 БД: ImageNet, COCO и Kaggle Kat and Dog images for Classification.

ИНС принимает на вход изображение, выдаёт результат в зависимости от задачи 1. Классификация – номер класса объекта, изображенного на данной снимке 2. Сегментация – маску, то есть изображение того же разрешения, что и входное, с разделением на классы или сущности в зависимости от вида сегментации. 3. Детектирование – перечень охватывающих прямоугольников или многоугольников с номером класса для каждого из них. 4. Ключевые точки – последовательность координат для N ключевых точек для каждого вхождения объекта.

Следует отметить, что существуют архитектуры, позволяющие решать сразу несколько из перечисленных задач. К ним относится, например, Mask RCNN, которая может быть обучена для решения сразу всех четырех перечисленных задач.

Для обучения и тестирования ИНС требуются файлы разметки (аннотации), и они могут быть в различных форматах. Поэтому в разработанной нами системе эти файлы приводятся к единому формату, используемому в COCO.

### 3. Некоторые методы мета-обучения.

Ряд публикаций посвящен обзору существующих систем AutoML и описанию подходов для создания подобных систем. Для обзора нами использована работа [2].

С математической точки зрения всякая система AutoML может быть формализована следующим образом: Пусть  $T$  – множество всех задач, которые потенциально могут быть решены при помощи инструментов машинного обучения. Пусть  $\Theta$  – множество всех конфигураций для всех алгоритмов машинного обучения, причем  $\Theta$  описывает пространство параметров произвольной структуры такое, что всякий алгоритм машинного обучения полностью характеризуется некоторым элементом из  $\Theta$  (в т.ч. гиперпараметры алгоритма – количество слоев нейронной сети, размеры обучающей выборки, параметры оптимизационных алгоритмов и т.п.). Пусть также имеется множество  $P$  всех скалярных результатов измерения эффективности решения некоторой задачи из множества  $T$  алгоритмом, описанным параметрами элемента из  $\Theta$ , т.е.  $P_{(i,j)} = P(\theta_i, t_j)$ ,  $\theta_i \in \Theta$ ,  $t_j \in T$ . Скалярные значения определяются в соответствии с установленной метрикой и по определенной процедуре измерения, которые определяются заранее. Пусть задано множество измерений производительности  $P_{new}$  для некоторой новой задачи  $t_{new}$  для некоторых алгоритмов машинного обучения:  $P_{new} = P_{\{i, new\}}$ . Система AutoML, по существу, представляет собой некоторую модель  $L$  которая осуществляет рекомендацию для новой задачи  $t_{new}$  некоторого оптимального набора параметров  $\Theta^*$  исходя из данных, полученных по множеству  $P_{new}$ .

Особый интерес представляют рекомендательные системы, которые не зависят от задачи. В случае подобных систем можно считать  $P_{new} = \emptyset$ , и в ходе создания рекомендательной системы для такого случая обучается функция, которая выдает набор предпочтительных конфигураций  $f : \Theta \times T \rightarrow \theta_k^*$ ,  $k = 1 \dots K$ , причем среди  $\{\theta_k^*\}$  выбирается лучший алгоритм для использования или дальнейшей оптимизации. Множество  $\{\theta_k^*\}$  может быть отранжировано не только по качеству, но и по скорости обучения.

При решении задач создания AutoML систем отдельным образом выделяются Байесовские подходы. Так, для обучения  $f$  во многих работах используется моделирование Гауссовским процессом, либо совместно для всех задач, либо для каждой задачи отдельно (так называемые суррогатные модели), с дальнейшей комбинацией задач (например, при помощи взвешенной нормализованной суммы).

Подходы, связанные с обучением Гауссовских процессов для создания рекомендательных систем при подборе гиперпараметров, несмотря на их универсальность, отличаются сложностью обучения и большими

вычислительными затратами. В силу данного обстоятельства нередко производится предварительная оптимизация пространства поиска гиперпараметров с целью ускорения обучения модели для рекомендаций. Так, к примеру, для ускорения работы можно ограничивать размерность пространства  $\Theta$ :

- Выбирая некоторое число атрибутов, которые дают наибольший разброс на заданной задаче, либо на всех задачах совместно
- Выбирая некоторое число атрибутов, которые дают наибольший прирост качества (по мере оценивания), при фиксированных прочих атрибутах
- Выбирая некоторое число атрибутов, которые дают наименьшее ухудшение качества (по мере оценивания), при фиксированных прочих атрибутах.

В качестве широко используемых компонент, применяемых в системах AutoML могут быть упомянуты следующие.

- Active Testing.

Данный подход использует метрики для оценки близости конфигураций, на основании чего в ходе поиска оптимальной конфигурации для задачи выделяются ближайшие к заданной известные задачи. В качестве примера может быть рассмотрена метрика *relative landmarks*:  $RL_{(a,b,j)} = P_{(a,j)} - P_{(b,j)}$ , для конфигураций  $\theta_a$ ,  $\theta_b$  и задачи  $t_j$ . Идея данного подхода может быть описана следующей циклической процедурой:

- 1) выбираем некоторую «лучшую» конфигурацию,
- 2) выбираем «лучшую» конфигурацию на «близких» задачах по мере RL,
- 3) обновляем RL,
- 4) повторяем 1)-3) до удовлетворения некоторому критерию остановки.

- Суррогатные модели.

Данный подход заключается в обучении модели для каждой известной ранее задачи, на основании проделанных ранее замеров эффективности: строятся модели  $s_j(\theta_i) = P_{(i,j)}$  для всех  $t_j$  используя  $P$ . Далее, после обучения суррогатных моделей для всякой новой задачи выбирается ансамбль моделей, которые обучены для близких к данной задачам. Близость задач может оцениваться с использованием некоторой метрики, как упомянуто выше. Однако, можно

использовать и активный подход для сравнения. Например, задачи можно считать близкими, если суррогатная модель задачи  $t_j$  генерирует близкие предсказания для задачи  $t_{new}$ .

- Использование кривых обучения.

В случае, если накоплена информация о процессе обучения различных моделей, можно использовать эти данные для построения предсказательных систем. Например, обучение разделяется на шаги  $r_t$  (например, добавляя одинаковое количество новых данных для обучения), и вычисляются значения  $P(\theta_i, t_j, r_t) = P_{(i,j,t)}$ , тем самым задается кривую обучения алгоритма  $\theta_i$  на задаче  $t_j$ . Проводя оценку задачи  $t_{new}$  на алгоритме  $\theta_i$  можно останавливать обучение на некотором шаге, и (возможно учитывая другие оценки близости  $t_{new}$  и остальных  $t_i$ ) делать предсказание по частично построенной кривой, насколько хорошо данный алгоритм отработает на полной базе данных. Таким образом поиск потенциально хороших конфигураций может быть ускорен.

- Выбор конфигурации по ближайшим соседям.

При заданной мере близости задач, выбор конфигурации может быть сведен к выбору  $k$  ближайших задач (алгоритмом kNN), и, соответственно,  $k$  наилучших конфигураций для этих задач, например при помощи суррогатных моделей.

- Использование мета-атрибутов.

Различные задачи (частично характеризуемые базами данных, на которых данные задачи обучаются и тестируются) обладают определенными характерными параметрами, которые называются мета-атрибутами. Примерами таких атрибутов могут служить статистические характеристики баз данных, такие как среднее значение и разброс определенной характеристики, объемы баз данных, количество пропущенных атрибутов. Одним из подходов при обучении AutoML систем является определение близости задач через близость мета-атрибутов по некоторой метрике, для выбора наилучших конфигураций среди близких задач.

Следует отметить, что при вычислении мета-атрибутов обязательна их нормализация, а иногда требуется уменьшение размерности задачи.

- Обучение мета-атрибутам

Вместо непосредственного вычисления мета-атрибутов для задачи возможен вариант обучения модели, для построения новых мета-атрибутов. Идея подхода заключается в следующем. Генерируется

набор бинарных мета-атрибутов  $m_{(j,a,b)}$ , который показывает будет ли задача  $t_j$  лучше решена при помощи  $\theta_a$  чем  $\theta_b$ . Для создания таких атрибутов обучается модель, принимающая пару  $(a, b)$  и задачу  $t_j$ , для обучения используется набор  $P$  и множество мета-атрибутов, которые могут быть вычислены вручную.

- Предсказание оценки по задаче и конфигурации.

При использовании данного подхода строится регрессия (линейная, SVM) и по наборам мета-атрибутов и конфигурации  $\theta_i$  предсказывается для задачи её оценки на конфигурации  $\theta_i$ , в т.ч. скорость обучения. Результаты предсказания используются для поиска хороших конфигураций в дальнейшей оптимизации.

- Рекомендации по последовательному использованию конфигураций (pipeline).

Бывает возможным использование модели обучения с подкреплением (reinforcement-learning) для предсказания какие конфигурации использовать после применения выбранной ранее конфигурации, для дальнейшего улучшения качества.

В качестве примеров систем AutoML, реализованных в виде программных библиотек, которые используются на практике могут быть упомянуты следующие системы: Auto-SKLearn [2] и AutoKeras [12].

## 4. Методы оптимизации подбора гиперпараметров.

В работе [13] «Практическое руководство по классификации опорных векторов» для получения оптимального набора гиперпараметров предлагается выполнить перебор по решетке их значений. Этот метод высоко затратен по времени и другим ресурсам, но легко параллелизуем.

Имплементация есть, например, в библиотеке sklearn (scikit-learn) [14] языка программирования Python, в библиотеке Keras этого же языка для этого подхода есть методы или инструменты.

С целью сокращения времени оптимизации гиперпараметров может быть использован метод случайного поиска по сетке их значений [15]. Метод является менее надежным, чем полный перебор, но может быть результативным при использовать дополнительных знаний о распределении возможных значений гиперпараметров или если задача оптимизации имеет низкую внутреннюю размерность.

Известны подходы, связанные с Байесовской оптимизацией. Для этого на первом этапе изучается поведение функции, оценивающей качество распознавания, значения которой определяются на ряде наборов значений гиперпараметров. Затем отбрасываются заведомо плохие их комбинации и выполняется обучение на наиболее перспективных их наборах. Метод при равных ограничениях по времени, как правило, позволяет получить лучшие результаты, чем перебор по решетке и случайный поиск.

Сюда же можно отнести алгоритмы, основанные на обучении с подкреплением (reinforcement learning, RL). Исследование метода содержится в [16]. В библиотеке github можно найти реализацию метода.

Следует отметить градиентные методы. Их использование связано с конкретным алгоритмом обучения, для которого можно вычислить градиенты гиперпараметров и воспользоваться методом градиентного спуска [17]. В своих исследованиях В.С. Половников с учениками использовал оптимизацию гиперпараметра dropout, включив дополнительный параметр в функцию потерь и оптимизировав его вместе с остальными весами.

Градиентный метод, изложенный в статье [18], позволяет оптимизировать следующие гиперпараметры: размер пакета, расписание скорости обучения, распределение весов при инициализации, схемы регуляризации, данные для обучения, архитектуры нейронных сетей. Для этого используется алгоритм стохастического градиентного спуска (stochastic gradient descent, SGD) со слагаемым, содержащим параметр момента, затем находятся формулы для вычисления градиентов. Для получения точных оценок требуется работать с гессианами, но, используя методы оптимизации, удастся сократить время их подсчета. Таким образом удастся оптимизировать скорость обучения и величину момента. Оптимизация размера пакета осуществляется по методу Hamiltonian Monte Carlo с использованием статьи [19].

Веса нейронной сети, согласно рассматриваемому алгоритму, инициализируются случайно, а оптимизируются вариации этих весов, аналогично подходу, использованному в работе [20]. Регуляризация сводится к взвешенной добавке L2, различным параметрам соответствует различные коэффициенты. Авторы показывают, что, как минимум, для логистической регрессии на MNIST это осмысленно. Предложен алгоритм для оптимизации данных. Путем объединения градиентов посредством преобразований данных авторы могут вычислять градиенты цели валидации в отношении процедур предварительной обработки, взвешивания или увеличения данных.

Оптимизация архитектуры нейронной сети используется в методе из статьи [36], который через множители включает те или иные ветви. В процесс обучения чередуется изменение параметров и этих множителей,

а потом по порогу убираются ветви с низкими коэффициентами. В зависимости от набора удаленных ветвей получаются различные архитектуры нейронных сетей. Метод хорош, но сильно ограничен по пространству поиска (последовательная блочная структура).

Перспективным представляется подход, автором которого является Roger Grosse, связанный с одновременной оптимизации параметров и гиперпараметров. Этому подходу посвящены статьи [21]– [23] и другие исследования.

Следует упомянуть также эволюционные методы, в частности, генетические алгоритмы, основанные на понятии популяции. Оптимизация здесь осуществляется по схеме: вносится мутация (в данном случае, меняются гиперпараметры алгоритма обучения) и после анализа результатов обучения с новыми гиперпараметрами, сравниваем, если новый вариант нейронной сети работает лучше, то дальше развиваем его дальше. Отметим методы, основанные на раннем останове. При этом, например, осуществляется параллельный поиск по сетке, но периодически отбрасываются самые отстающие наборы значений гиперпараметров, а также методы, основанные на схеме моделирования отжига simulated annealing) [24].

Для реализации системы автоматного обучения мы использовали поиск по решетке значений выбранных гиперпараметров, но не полный перебор их значений, а, используя выбор наиболее «перспективных» направлений перемещения по ней.

Исследование выполнено при поддержке Министерства науки и высшего образования Российской Федерации (грант № 075-15-2020-801)

## Список литературы

- [1] Бирюкова В.А., “Автоматный подход для оптимизации работы системы обучения нейронных сетей”, *Интеллектуальные системы. Теория и приложения*, **25:4** (2021), 71–78.
- [2] Hutter F., Kotthoff L., Vanschoren J., *Automated Machine Learning*, «Springer», Friburg, Germany, 2019, 223 с.
- [3] Deng J., Dong W., Socher R., Li L.-J., Li K. and Fei-Fei L., “ImageNet: A Large-Scale Hierarchical Image Database”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, 248–255
- [4] Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A., “The PASCAL Visual Object Classes Challenge: A Retrospective”, *International Journal of Computer Vision*, **111:1** (2015), 98–136

- [5] Lin T.-Y. , Maire M., Belongie S., Bourdev L., Ross G., James H., Perona P., Ramanan D., Zitnick C. L., Dollár P., “Microsoft COCO: Common Objects in Context”, *ArXiv*, **abs/ 1405.0312**, 2014
- [6] Geiger A., Lenz P., Stiller C. and Urtasun R., “Vision meets Robotics: The KITTI Dataset”, *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012
- [7] Cordts M., Omran M., Ramos S., Rehfeld T., Enzweiler M., Benenson R., Franke U., Roth S., Schiele B., “The Cityscapes Dataset for Semantic Urban Scene Understanding”, *ArXiv*, **abs/ 1604.01685**, 2014
- [8] Jain V. and Learned-Miller E., *FDDB: A Benchmark for Face Detection in Unconstrained Settings*, University of Massachusetts, Amherst, 2014, 11 c.
- [9] Yang, Shuo and Luo, Ping and Loy, Chen Change and Tang, Xiaoou, “WIDER FACE: A Face Detection Benchmark”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016
- [10] Deng J., and Roussos A., Chrysos G., Ververas E., Kotsia I., Shen J., Zafeiriou S., “The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking”, *International Journal of Computer Vision*, **127**:6 (2019), 599–624
- [11] Шахуро В.И., Коношин А.С., “Российская база изображений автодорожных знаков”, *Компьютерная оптика*, **40**:2 (2016), 294–300
- [12] Jin Haifeng and Song Qingquan and Hu, Xia, “An Efficient Neural Architecture Search System”, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, 1946–1956
- [13] Chin-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin, *A practical guide to support vector classification*, National Taiwan University, Taipei, 2010, 16 c.
- [14] Aurelien Geron, *Hands-On Mashine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*, O’Reilly Media, Inc., Sebastopol, California, 2019, 856 c.
- [15] James Bergstra, Yoshua Bengio, “Random Search for Hyper-Parameter Optimization”, *Machine Learning Research*, **13** (2012), 281–305
- [16] Bergstra, J., Yamins, D., Cox, D. D., “Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision

Architectures.”, *TProc. of the 30th International Conference on Machine Learning (ICML 2013)*, 2013

- [17] Larsen J., Hansen L.K., Svarer C., Ohlson B.O.M., “Design and regularization of neural networks: the optimal use of a validation set”, *IEEE Signal Processing Society Workshop*, 1996, 62–71
- [18] Maclaurin D., Duvenaud D., Adams R. P., “Gradient-based Hyperparameter Optimization through Reversible Learning”, *ArXiv, abs/ 1502.03492*, 2015
- [19] Salimans T., Kingma D.P., Welling M., “Markov chain Monte Carlo and variational inference: Bridging the gap”, *ArXiv, abs/ 1410.6460*, 2014
- [20] Chen Zhu, Renkun Ni, Zheng Xu, Kezhi Kong, W. Ronny Huang, Tom Goldstein, “GradInit: Learning to Initialize Neural Networks for Stable and Efficient Training”, *ArXiv, abs/ 2102.08098*, 2021
- [21] MacKay M., Vicol P., Lorraine J., Duvenaud D., Grosse R., “Self-Tuning Networks: Bilevel Optimization of Hyperparameters using Structured Best-Response Functions”, *ArXiv, abs/ 1903.03088*, 2019
- [22] Bae J., Grosse R., “Delta-STN: Efficient Bilevel Optimization for Neural Networks using Structured Response Jacobians”, *ArXiv, abs/ 2010.13514*, 2020
- [23] Franceschi L., Frasconi P., Salso S., Grazzi R., Pontil M., “Bilevel Programming for Hyperparameter Optimization and Meta-Learning”, *ArXiv, abs/ 1806.04910*, 2018
- [24] Correia A. H.C., Worrall D. E., Bondesan R., “Neural Simulated Annealing”, *ArXiv, abs/ 2203.02201*, 2022

### **Opportunities for the implementation of automatic learning system**

**Biryukova V.A., Bokov G.V., Drobyshev A.S., Kalachev G.V.,  
Polovnikov V.S., Ronzhin D.V., Chasovskikh A.A.**

The results of the analysis of some available sources are presented, on the basis of which an automatic learning model for artificial neural networks was implemented.

*Keywords:* image recognition, artificial neural network, automated machine learning, automatic learning model, hyperparameters.

## References

- [1] Biryukova V.A., “Optimization of neural network learning via system with automata approach”, *Intelligent Systems. Theory and Applications*, **25**:4 (2021), 71–78 (In Russian)
- [2] Hutter F., Kotthoff L., Vanschoren J., *Automated Machine Learning*, «Springer», Friburg, Germany, 2019, 223 c.
- [3] Deng J., Dong W., Socher R., Li L.-J., Li K. and Fei-Fei L., “ImageNet: A Large-Scale Hierarchical Image Database”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, 248–255
- [4] Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A., “The PASCAL Visual Object Classes Challenge: A Retrospective”, *International Journal of Computer Vision*, **111**:1 (2015), 98–136
- [5] Lin T.-Y. , Maire M., Belongie S., Bourdev L., Ross G., James H., Perona P., Ramanan D., Zitnick C. L., Dollár P., “Microsoft COCO: Common Objects in Context”, *ArXiv*, **abs/ 1405.0312**, 2014
- [6] Geiger A., Lenz P., Stiller C. and Urtasun R., “Vision meets Robotics: The KITTI Dataset”, *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012
- [7] Cordts M., Omran M., Ramos S., Rehfeld T., Enzweiler M., Benenson R., Franke U., Roth S., Schiele B., “The Cityscapes Dataset for Semantic Urban Scene Understanding”, *ArXiv*, **abs/ 1604.01685**, 2014
- [8] Jain V. and Learned-Miller E., *FDDB: A Benchmark for Face Detection in Unconstrained Settings*, University of Massachusetts, Amherst, 2014, 11 c.
- [9] Yang, Shuo and Luo, Ping and Loy, Chen Change and Tang, Xiaoou, “WIDER FACE: A Face Detection Benchmark”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016
- [10] Deng J., and Roussos A., Chrysos G., Ververas E., Kotsia I., Shen J., Zafeiriou S., “The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking”, *International Journal of Computer Vision*, **127**:6 (2019), 599–624
- [11] Shakhuro V.I., Konushin A.S., “Russian traffic sign images dataset”, *Computer Optics*, **40**:2 (2016), 294–300 (In Russian)

- [12] Jin Haifeng and Song Qingquan and Hu, Xia, “An Efficient Neural Architecture Search System”, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, 1946–1956
- [13] Chin-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin, *A practical guide to support vector classification*, National Taiwan University, Taipei, 2010, 16 c.
- [14] Aurelien Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*, O’Reilly Media, Inc., Sebastopol, California, 2019, 856 c.
- [15] James Bergstra, Yoshua Bengio, “Random Search for Hyper-Parameter Optimization”, *Machine Learning Research*, **13** (2012), 281–305
- [16] Bergstra, J., Yamins, D., Cox, D. D., “Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures.”, *TProc. of the 30th International Conference on Machine Learning (ICML 2013)*, 2013
- [17] Larsen J., Hansen L.K., Svarer C., Ohlson B.O.M., “Design and regularization of neural networks: the optimal use of a validation set”, *IEEE Signal Processing Society Workshop*, 1996, 62–71
- [18] Maclaurin D., Duvenaud D., Adams R. P., “Gradient-based Hyperparameter Optimization through Reversible Learning”, *ArXiv*, **abs/1502.03492**, 2015
- [19] Salimans T., Kingma D.P., Welling M., “Markov chain Monte Carlo and variational inference: Bridging the gap”, *ArXiv*, **abs/1410.6460**, 2014
- [20] Chen Zhu, Renkun Ni, Zheng Xu, Kezhi Kong, W. Ronny Huang, Tom Goldstein, “GradInit: Learning to Initialize Neural Networks for Stable and Efficient Training”, *ArXiv*, **abs/2102.08098**, 2021
- [21] MacKay M., Vicol P., Lorraine J., Duvenaud D., Grosse R., “Self-Tuning Networks: Bilevel Optimization of Hyperparameters using Structured Best-Response Functions”, *ArXiv*, **abs/1903.03088**, 2019
- [22] Bae J., Grosse R., “Delta-STN: Efficient Bilevel Optimization for Neural Networks using Structured Response Jacobians”, *ArXiv*, **abs/2010.13514**, 2020
- [23] Franceschi L., Frasconi P., Salso S., Grazi R., Pontil M., “Bilevel Programming for Hyperparameter Optimization and Meta-Learning”, *ArXiv*, **abs/1806.04910**, 2018

- [24] Correia A. H.C., Worrall D. E., Bondesan R., “Neural Simulated Annealing”, *ArXiv*, ***abs/2203.02201***, 2022