



Рисунок 5 – Архитектура порождающей модели для восстановления изображений

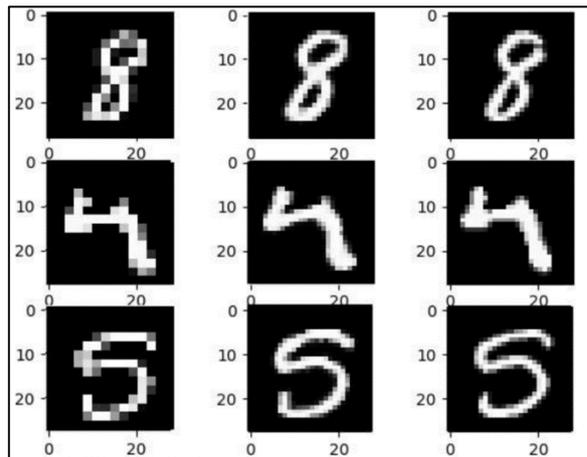


Рисунок 6 – Визуальное сравнение оригинального изображения (справа), изображения низкого разрешения (слева) и обработанного изображения (центр)

Литература

1. Generative Adversarial Nets [Электронный ресурс] // Neural Information Processing Systems (NIPS). – <https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf> (дата обращения: 03.04.2019).
2. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks [Электронный ресурс] // arXiv.org. – <https://arxiv.org/abs/1511.06434> (дата обращения: 07.04.2019).
3. The MNIST Database [Электронный ресурс] // Yann LeCun Blog. – <http://yann.lecun.com/exdb/mnist/> (дата обращения: 22.04.2019).
4. Semantically Decomposing the Latent Spaces of Generative Adversarial Networks [Электронный ресурс] // arXiv.org. – <https://arxiv.org/pdf/1705.07904.pdf> (дата обращения: 10.05.2019).

Р.Р. Бадрутдинов

ДИФФЕРЕНЦИРУЕМЫЙ ПОИСК АРХИТЕКТУР СВЁРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

(Самарский университет)

Поиск архитектур нейронных сетей (Neural Architecture Search, NAS) — процесс автоматического проектирования искусственных нейронных сетей. Подходы, разработанные в рамках этой парадигмы, показали результаты, не уступающие ручному подбору параметров моделей машинного обучения, а в



некоторых случаях даже лучшие. Методы NAS можно классифицировать по следующим признакам:

- Пространство поиска означает множество алгоритмов, среди которых производится поиск.
- Стратегия поиска определяет алгоритм поиска оптимальной архитектуры в пространстве поиска.
- Стратегии оценки производительности. Поскольку определение качества конкретной архитектуры требует обучения модели и последующей её валидации, вопрос объёма вычислений и, соответственно, времени поиска становится особенно важным.

Существует много подходов к поиску архитектур нейронных сетей, среди которых:

1. Дискретный поиск. Перебор таких параметров как количество слоёв и нейронов. Поскольку, в общем случае, множество архитектур счётно, требуется задать граничные условия перебора. Один из способов это сделать — применить ограничения, вычисленные в рамках теории Вапника-Червоненкиса [1].
2. Обучение с подкреплением [2]. В этом подходе каждая архитектура представляется программным агентом, который получает обратную связь посредством размера награды, которая часто связана с производительностью сети. Цель обучения — максимизация награды. Недостатком данного метода является высокая вычислительная сложность алгоритмов, поскольку на каждом шаге поиска осуществляется обучение и валидация модели-кандидата.
3. Эволюционные алгоритмы [3]. Мутациями здесь являются изменения архитектуры сети: добавление или удаление слоя, изменение типа слоя, изменение параметров слоя и т. д.
4. Методы непрерывной оптимизации архитектур нейронных сетей. Относительно новый подход, состоящий в построении отображения множества архитектур на непрерывное пространство с последующим поиском оптимальной архитектуры с применением градиентного алгоритма оптимизации в этом гладком пространстве. Показывает блестящие результаты при сравнительно небольшом объёме вычислений.

В данной работе рассмотрен один из наиболее современных методов поиска оптимальной архитектуры нейронных сетей в дифференцируемом пространстве — дифференцируемый метод поиска архитектур (Differentiable ARchiTecture Search, DARTS) [4]. Найденная архитектура протестирована на наборе данных MNIST [5]. Основная идея метода DARTS заключается в релаксации задачи дискретной оптимизации (поиска оптимальной архитектуры нейронной сети) — к непрерывной. Поиск в полученном гладком пространстве осуществляется посредством алгоритма градиентного спуска, при этом происходит минимизация количества ошибок на тестовой выборке.

Поиск параметров осуществляется для одной клетки сети. Клетка — ориентированный ациклический граф, состоящий из N последовательно соединён-



ных узлов. Полученная в результате клетка может быть соединена последовательно с другими клетками для получения свёрточной нейронной сети. Схема оптимизируемой клетки показана на рисунке 1.

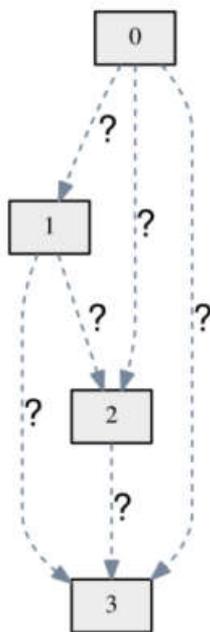


Рисунок 1 — Схема оптимизируемой клетки свёрточной нейронной сети

В оригинальной работе предлагается предположить, что каждая клетка имеет два входных узла и один выходной узел. Задача поиска оптимальной архитектуры, таким образом, заключается в поиске набора операций O , соединяющих узлы в клетке.

Релаксация в линейном программировании — задача замены задачи дискретной оптимизации на задачу непрерывной оптимизации [6].

Для того, чтобы сделать пространство поиска непрерывным, категориальная задача поиска релаксируется к мягкому максимуму от всех возможных вариантов выбора в соответствии с формулой:

$$\bar{o}^{(i,j)} = \sum_o \exp \frac{(\alpha_o^{i,j})}{\sum_o \alpha_o^{(i,j)}} o(x)$$

Смесь операций для ребра (i, j) задаётся вектором $\alpha^{(i,j)}$. Операция поиска оптимальной операции сводится к нахождению O . Для получения дискретной архитектуры достаточно каждую операцию \bar{o} заменить на $o^{(i,j)} = \operatorname{argmax}_o \alpha^{(i,j)}$.

После релаксации задачу можно сформулировать как совместное обучение гиперпараметров α и параметров нейронной сети w посредством оптимизации функции погрешности на множестве тестирования.

Обозначим L_{train} и L_{val} функцию погрешности на обучающей выборке и выборке для тестирования, соответственно. Понятно, что они зависят не только от архитектуры α , но и от параметров сети w . Задачу поиска, таким образом, можно выразить как поиск $\alpha^* = \operatorname{argmin}_\alpha L_{val}(\alpha, w^*), w^* = \operatorname{argmin}_w L_{train}(\alpha^*, w)$. Эту задачу можно записать как задачу двухуровневой оптимизации, где α - это внешняя переменная, а w - внутренняя:



$$\min_{\alpha} L_{val}(w^*(\alpha), \alpha)$$

$$w^*(\alpha) = \operatorname{argmin}_{\alpha} L_{train}(w, \alpha)$$

Выполнение градиентного спуска в пространстве архитектур накладно ввиду дорогой внутренней оптимизации параметров, поэтому была предложена следующая аппроксимация:

$$\nabla_{\alpha} L_{val}(w^*(\alpha), \alpha) \approx \nabla_{\alpha} L_{val}(w - \xi \nabla_w L_{train}(w, \alpha), \alpha)$$

Здесь w - веса на данном шаге, ξ - скорость обучения для шага внутренней оптимизации. Идея состоит в том, чтобы выполнять приближенно внутреннюю оптимизацию за 1 шаг и избежать тем самым дорогого обучения параметров сети. Так же легко видеть, что если w уже находится в локальном оптимуме, то это выражение принимает вид $\nabla_{\alpha} L_{val}(w, \alpha)$. Для формирования дискретной архитектуры для каждого узла выбираются k сильнейших операций среди всех ненулевых найденных операций от предыдущих узлов. Сила операции определяется выражением:

$$\frac{\exp(\alpha_o^{(i,j)})}{\sum_{\delta} \exp(\alpha_{\delta}^{(i,j)})}$$

В данной работе анализируется применимость DARTS к классическому набору MNIST. MNIST — это банк из 70000 изображений рукописных цифр, предложенный в 1998 году Яном Лекуном. Изображения разбиты на 10 классов.

Для проведения экспериментов была выбрана облачная платформа Google Compute Cloud с большим количеством оперативной памяти и графическим процессором для ускорения вычислений. Конфигурация виртуальной машины для проведения экспериментов приведена в таблице 1.

Таблица 1. Конфигурация облачной виртуальной машины

Параметр	Значение
Количество процессоров	2
Объём RAM	13 ГБ
Модель графического процессора	Nvidia Tesla K80
Объём памяти графического процессора	12 ГБ

В ходе экспериментов проверялась эффективность поиска оптимальной архитектуры для сетей с количеством слоёв 1, 2 и 3.

Для каждой сети производилась оптимизация архитектуры, обучение методом обратного распространения ошибки, построенном на стохастическом градиентном спуске. Обучение производилось в 2 эпохи, набор данных для обучения разбивался на мини-наборы по 50 примеров.

Валидация модели производилась на наборе данных для тестирования. Целевым показателем считалась доля верных ответов сети.

Результаты экспериментов приведены в таблице 2.



Таблица 2. Результаты экспериментов

Количество слоёв	Доля верных ответов на наборе для тестирования
1	0,9797
2	0,9903
3	0,9859

Литература

1. Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. [Текст]/ В.Н. Вапник, А.Я. Червоненкис: – М.: Наука, 1974.
2. Обучение с подкреплением // Википедия. [2008—2019]. Дата обновления: 31.03.2019. URL: <https://ru.wikipedia.org/?oldid=98962763> (дата обращения: 01.05.2019).
3. Эволюционные алгоритмы // Википедия. [2008—2018]. Дата обновления: 24.10.2018. URL: <https://ru.wikipedia.org/?oldid=95788219> (дата обращения: 01.05.2019).
4. Hanxiao Liu, Karen Simonyan, Yiming Yang DARTS: Differentiable Architecture Search [Электронный ресурс] // arXiv.org. 2001. Дата обновления: 23.04.2019. URL: <https://arxiv.org/abs/1806.09055> (дата обращения: 01.05.2019)
5. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, Backpropagation Applied to Handwritten Zip Code Recognition, 1989
6. Linear programming relaxation // Википедия. [2006—2019]. Дата обновления: 09.02.2019. URL: https://en.wikipedia.org/w/index.php?title=Linear_programming_relaxation&oldid=882528878 (дата обращения: 19.05.2019).

Н.М. Барсков, И.М. Куликовских

ПРОГНОЗИРОВАНИЕ ВЗАИМОДЕЙСТВИЙ ПОЛЬЗОВАТЕЛЕЙ С РЕКЛАМНЫМ СОДЕРЖИМЫМ ИНТЕРНЕТ СТРАНИЦ

(Самарский университет)

В рамках данной работы был разработан и реализован алгоритм для прогнозирования взаимодействия пользователей с рекламными элементами интернет-страниц. Дана обзорная структура предложенного метода, оценки точности прогнозируемых величин.

Предложенный алгоритм представляет собой комбинацию метода наивного сезонного прогнозирования, модели прогнозирования Prophet и метода калибровки выборки.