

«МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(национальный исследовательский университет)
ФИЗТЕХ-ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
КАФЕДРА «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ»

Гребенькова Ольга Сергеевна

Порождение моделей заданной сложности с использованием байесовских гиперсетей

03.03.01 — Прикладные математика и физика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Научный руководитель:
к. ф.-м. н. Бахтеев Олег Юрьевич

Москва
2021

1 Введение

В работе рассматривается задача оптимизации модели глубокого обучения. Под моделью глубокого обучения понимается суперпозиция дифференцируемых по параметрам функций. Построение модели заданной сложности — одна из фундаментальных проблем глубокого обучения, так как по построению данное семейство моделей имеет избыточное число параметров [1]. В работе предполагается, что сложность модели задана заранее. Под сложностью модели понимается ее обоснованность.

Предлагаемый метод заключается в представлении параметров модели глубокого обучения в виде гиперсети. Гиперсеть — модель, которая задает параметры модели. На вход такой модели подается информация о структуре модели, а в результате работы порождается вектор параметров входной модели. Другими словами, гиперсеть — это отображение из множества переменных, отвечающих за сложность модели, во множество параметров модели. В работе [2] рассмотрены статистические и динамические гиперсети для порождения весов сверточных и рекуррентных моделей соответственно.

В данной работе используется байесовский подход. Вводятся вероятностные предположения о параметрах моделей глубокого обучения. В работах [1, 3] предлагается использовать в качестве функции ошибки для оптимизации параметров модели глубокого обучения минимальную длину описания. Минимальная длина описания — минимальный объем информации, который требуется для передачи информации о модели и о выборке. Также в работах [1, 4] получены виды аппроксимаций для обоснованности модели для различных классов априорных распределений параметров.

Альтернативным подходом к построению модели заданной сложности выступает задача порождения и выбора оптимальной структуры моделей глубокого обучения. В работе [5] рассматривается возможность порождения широкого класса сверточных моделей как ча-

стей обобщенной модели, которая называется «фабрикой» (англ. fabric). Данная структура позволяют обойти процесс оптимизации параметров и проверки качества одиночных моделей. В работах [6, 7] представлены подходы к решению задачи выбора структуры нейросети с использованием дифференцируемых алгоритмов — стохастического (англ. Stochastic Neural Architecture Search — SNAS) и дифференцируемого (англ. Differentiable Neural Architecture Search — DNAS) методов поиска нейронных архитектур. Особенность работы [7] заключается в решении задачи выбора архитектуры модели, удовлетворяющей эксплуатационным требованиям: быстродействию на различных типах процессоров.

В данной работе исследуется поведение обобщенной функции обособленности модели. Исследуется влияние априорного распределения на сложность модели. Для контроля сложности предлагается рассматривать задачу оптимизации параметров гиперсети. Данная модель порождает модели наперед заданной сложности с меньшими вычислительными затратами, чем в случае оптимизации модели, получаемой напрямую. Работа схожа с работой [8], где также исследовалась возможность получения гиперсети для предсказания наилучших гиперпараметров оптимизации модели. Вычислительный эксперимент проводился на выборке рукописных цифр MNIST [9].

2 Постановка задачи

Задана выборка:

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

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

$$\mathbf{f}(\mathbf{x}, \mathbf{w}) : \mathbb{R}^m \times \mathbb{R}^n \longrightarrow \mathbb{R}^Y,$$

где $\mathbf{w} \in \mathbb{R}^n$ — пространство параметров модели. Пусть задано априорное распределение вектора параметров в пространстве \mathbb{R}^n :

$$p(\mathbf{w}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{A}_{\text{pr}}^{-1}),$$

где $\boldsymbol{\mu}$, $\mathbf{A}_{\text{pr}}^{-1}$ — вектор средних и матрица ковариации априорного распределения. Тогда

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$$

является апостериорным распределением вектора параметров \mathbf{w} при заданной выборке \mathcal{D} . Пусть также задано вариационное распределение

$$q(\mathbf{w}) \sim \mathcal{N}(\mathbf{m}, \mathbf{A}_{\text{ps}}^{-1}).$$

Здесь \mathbf{m} , $\mathbf{A}_{\text{ps}}^{-1}$ — вектор средних и матрица ковариации, аппроксимирующее неизвестное апостериорное распределение $p(\mathbf{w}|\mathcal{D})$.

Для модели \mathbf{f} и соответствующего ей вектора параметров \mathbf{w} определим логарифмическую функцию правдоподобия выборки:

$$\mathcal{L}_{\mathcal{D}}(\mathcal{D}|\mathbf{w}) = \log p(\mathcal{D}|\mathbf{w}). \quad (1)$$

Оптимальные значения \mathbf{w} находятся из максимизации $\mathcal{L}(\mathcal{D})$ — логарифма обоснованности модели:

$$\mathcal{L}(\mathcal{D}) = \log p(\mathcal{D}) = \log \int_{\mathbf{w} \in \mathbb{R}^n} p(\mathcal{D}|\mathbf{w})p(\mathbf{w})d\mathbf{w}. \quad (2)$$

Так как вычисление интеграла (2) относится к вычислительно сложным задачам, рассмотрим вариационный подход к решению задачи. Оценим интеграл (2):

$$\mathcal{L}(\mathcal{D}) = \log p(\mathcal{D}) = \int_{\mathbf{w} \in \mathbb{R}^n} q(\mathbf{w}) \log \frac{p(\mathcal{D}, \mathbf{w})}{q(\mathbf{w})} d\mathbf{w} - \int_{\mathbf{w} \in \mathbb{R}^n} q(\mathbf{w}) \log \frac{p(\mathbf{w}|\mathcal{D})}{q(\mathbf{w})} d\mathbf{w} \geq$$

$$\begin{aligned}
&\geq \int_{\mathbf{w} \in \mathbb{R}^n} q(\mathbf{w}) \log \frac{p(\mathfrak{D}, \mathbf{w})}{q(\mathbf{w})} d\mathbf{w} = \int_{\mathbf{w} \in \mathbb{R}^n} q(\mathbf{w}) \log \frac{p(\mathbf{w})}{q(\mathbf{w})} d\mathbf{w} + \int_{\mathbf{w} \in \mathbb{R}^n} q(\mathbf{w}) \log p(\mathfrak{D}|\mathbf{w}) d\mathbf{w} = \\
&= \mathcal{L}_{\mathbf{w}}(\mathfrak{D}, \mathbf{w}) + \mathcal{L}_E(\mathfrak{D}). \tag{3}
\end{aligned}$$

Первое слагаемое формулы (3) — это различие между апостериорным и априорным распределением параметров, задающее сложность распределением параметров относительно априорных предположений. Оно определяется расстоянием Кульбака–Лейблера, то есть расстоянием между вариационным распределением $q(\mathbf{w})$ и априорным распределением $p(\mathbf{w})$:

$$\mathcal{L}_{\mathbf{w}}(\mathfrak{D}, \mathbf{w}) = -D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w})).$$

Второе слагаемое формулы (3) представляет собой математическое ожидание правдоподобия выборки $\mathcal{L}_{\mathfrak{D}}(\mathfrak{D}|\mathbf{w})$:

$$\mathcal{L}_E = \mathbb{E}_{q(\mathbf{w})} \mathcal{L}_{\mathfrak{D}}(\mathfrak{D}|\mathbf{w}).$$

Обобщенная обоснованность модели — это один из показателей сложности модели [1]. Рассматривается задача оптимизации параметров модели по обобщенной функции обоснованности \mathfrak{L} :

$$\mathfrak{L}(\lambda) = \lambda \mathcal{L}_{\mathbf{w}}(\mathfrak{D}, \mathbf{w}) - \mathcal{L}_E(\mathfrak{D}), \tag{4}$$

где параметр сложности λ контролирует влияние априорного распределения на выбор итоговой модели.

Введем множество допустимых значений параметра сложности $\lambda \in \Lambda \subset \mathbb{R}^+$. Требуется найти такое отображение $\mathfrak{G} : \Lambda \rightarrow \mathbb{R}^n$, при котором для произвольного значения параметра сложности $\lambda \in \Lambda$ параметры доставляли бы максимум следующей функции:

$$\mathfrak{G}(\lambda) = \arg \max_{\mathbf{w} \in \mathbb{R}^n} (\log p(\mathfrak{D}|\mathbf{w}) - \lambda D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}))). \tag{5}$$

3 Построение гиперсети для контроля сложности модели

Решение задачи оптимизации (5) для произвольного значения $\lambda \in \Lambda$ относится к вычислительно сложным задачам. В данной работе для ее решения предлагается использовать гиперсеть.

Пусть задано множество параметров Λ , контролирующих сложность модели. Гиперсеть — это параметрическое отображение из множества Λ во множество параметров модели:

$$\mathbf{G} : \Lambda \times \mathbb{R}^u \rightarrow \mathbb{R}^n,$$

где \mathbb{R}^u — множество допустимых параметров гиперсети. Рассмотрены два вида гиперсетей: 1) с отображением во множество матриц низкого ранга и 2) с линейной аппроксимацией. Пусть \mathbf{f} — функция, переводящаяся λ в скрытый слой; $\mathbf{U}_1, \mathbf{U}_2$ — матрицы, переводящие из скрытого слоя в нужную размерность, их конкатенация принадлежит пространству параметров гиперсети: $[\mathbf{U}_1, \mathbf{U}_2] = \mathbf{U} \in \mathbb{R}^u$; \mathbf{B}_1 — матрица, не зависящая от λ . Тогда первая реализация гиперсети имеет вид

$$\mathbf{G}_{\text{lowrank}}(\lambda) = (\mathbf{f}(\lambda)\mathbf{U}_1)^\top (\mathbf{f}(\lambda)\mathbf{U}_2) + \mathbf{B}_1, \quad (6)$$

где параметр λ — случайное число, реализуемое для каждой подвыборки при оптимизации параметров. Вторая реализация имеет вид

$$\mathbf{G}_{\text{linear}}(\lambda) = \lambda \mathbf{b}_2 + \mathbf{b}_3, \quad (7)$$

где $\mathbf{b}_2, \mathbf{b}_3$ — константы, не зависящие от λ .

Для аппроксимации оптимизационной задачи (5) предлагается оптимизировать параметры гиперсети $\mathbf{U} \in \mathbb{R}^u$ по случайно порожденным значениям параметра сложности $\lambda \in \Lambda$:

$$\mathbb{E}_{\lambda \sim P(\lambda)} (\log p(\mathcal{D}|\mathbf{w}) - \lambda D_{\text{KL}}(q(\mathbf{w})||p(\mathbf{w}))) \rightarrow \max_{\mathbf{U} \in \mathbb{R}^u}, \quad (8)$$

где $P(\lambda)$ — некоторое распределение на множестве Λ . В данной работе в качестве распределения использовалось следующее: $\log \lambda \sim \mathcal{U}(-1, 2)$, где \mathcal{U} — равномерное распределение.

4 Вычислительный эксперимент

Для анализа свойств обобщенной задачи оптимизации (8) и предложенного метода построения гиперсети был проведен вычислительный эксперимент на выборке рукописных цифр MNIST [9]. Проведено сравнение следующих методов:

- а) построения модели напрямую без использования гиперсети (5);
- б) построения модели напрямую без использования гиперсети (5) с оптимизацией за одну эпоху;
- в) построение с использованием гиперсети (6);
- г) построение с использованием гиперсети (6) с дообучением итоговой модели за одну эпоху;
- д) построение с использованием гиперсети (7);
- е) построение с использованием гиперсети (7) с дообучением итоговой модели за одну эпоху.

Второй метод позволяет определить, насколько использование гиперсетей (6), (7) с дообучением за одну эпоху эффективно для оптимизации параметров модели в сравнении с оптимизацией модели напрямую. Для каждой из моделей проводилось пять запусков, результаты которых усреднялись.

Для каждой модели проводилось прореживание параметров с применением подхода, описанного в [1]. Критерием удаления пара-

метров выступала относительная плотность модели

$$\rho(w_i) \propto \exp\left(-\frac{\mu_i^2}{2\sigma_i^2}\right).$$

Рассматривались следующие критерии качества модели:

1. Точность классификации

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

где m — длина тестовой выборки.

2. Стабильность модели S : отношение точности исходной модели к точности модели с прореживанием 90% параметров. Данная величина показывает, насколько качество модели стабильно относительно удаления значительного числа параметров.
3. Число обновлений параметров модели. Этот показатель определяется как число всех обновлений значений параметров модели за все эпохи и характеризует сложность итоговой оптимизации.
4. Обобщенная обоснованность модели \mathfrak{L} (4).

Была рассмотрена нейросеть состоящая из двух слоев: 100 и 10 нейронов соответственно, где второй слой отвечает за функцию softmax:

$$\mathbf{f}(\mathbf{x}, \mathbf{w}) = \text{softmax}(\mathbf{w}_2^\top \text{ReLU}(\mathbf{w}_1^\top \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2),$$

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

$$\text{softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_{j=1}^k \exp(x_j)}, \quad i = 1, \dots, k,$$

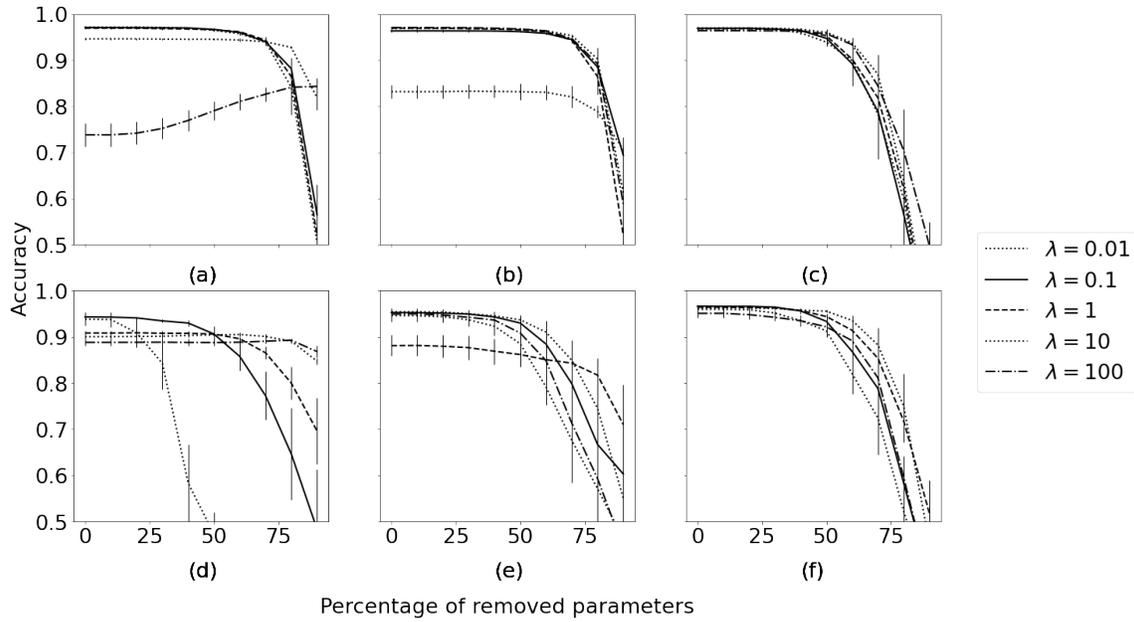


Рис. 1: График зависимости точности классификации от процента удаленных параметров для: а) модели без использования гиперсети (5); б) модели без использования гиперсети (5) с оптимизацией за одну эпоху; в) модели с использованием гиперсети (6); г) модели с использованием гиперсети (6) с дообучением итоговой модели за одну эпоху; д) модели с использованием гиперсети (7); е) модели с использованием гиперсети (7) с дообучением итоговой модели за одну эпоху

$$\mathbf{ReLU}(\mathbf{x}) = \mathbf{max}(0, \mathbf{x}). \quad (9)$$

Нейросеть запускалась для разных значений параметра сложности $\lambda \in \Lambda$.

На рис. 1а показано, как меняется точность классификации при удалении параметров указанным методом. Из графика видно, что вариационный метод позволяет удалить $\approx 60\%$ параметров при $\lambda = 0, 1; 1$ и $\approx 80\%$ параметров при $\lambda = 10; 100$ без значительной потери точности классификации. При дальнейшем удалении качество для всех значений снижается. При больших значениях $\lambda > 100$ получается избыточно упрощенная модель, которая содержит малое число параметров. Таким образом, удаление параметров нейросети при данном значении λ слабо влияет на точность классификации. Однако изначальная точность невысока.

Для обеих моделей с использованием гиперсетей использовался оптимизатор SGD. Обучение проводилось на протяжении 25 эпох. В качестве $\mathbf{A}_{\text{pr}}^{-1}$ используется $\mathbf{diag}(0,1)$. Для первой модели (6) был рассмотрен случай с 50 нейронами в скрытом слое и с функций активации ReLU (9). При обучении второй модели (7) каждая подвыборка проходила процесс оптимизации с пятью разными значениями сэмплируемой λ . Прореживание нейросетей запускалось для разных значений параметра сложности $\lambda \in \Lambda$.

Таблица 1: Точность и число обновлений параметров моделей

Модель	Accuracy ($\lambda = 0, 1$)	Accuracy ($\lambda = 1$)	Accuracy ($\lambda = 10$)	Accuracy ($\lambda = 100$)	Число обновлений параметров
а) Без гиперсети	0,87686	0,901570	0,8674	0,73031	7 458 976 000
б) Без гиперсети с дообучением	0,81682	0,81986	0,83477	0,78924	298 359 040
в) Гиперсеть (6)	0,87719	0,87666	0,87466	0,83124	4 165 376 600
г) Гиперсеть (6) с дообучением	0,90262	0,90167	0,88876	0,83902	4 239 966 360
д) Гиперсеть (7)	0,900679	0,90021	0,89218	0,53857	3 729 488 000
е) Гиперсеть (7) с дообучением	0,90104	0,91456	0,89538	0,80627	3 804 077 760

На рис. 1в показано, как меняется точность классификации при удалении параметров указанным методом для модели с низкоранго-

вой аппроксимацией. Как видно из графика, средняя точность классификации относительно базового эксперимента для малых значений $\lambda \in [0, 1; 10]$ понизилась. Также сильно увеличилось отклонение от среднего. При этом для всех значений $\lambda \in \Lambda$ получили более стабильную модель: точность классификации меньше зависит от удаления параметров. Однако наблюдаем большую потерю точности при удалении более 80% .

На рис. 1д показано, как меняется точность классификации при удалении параметров указанным методом для модели с линейной аппроксимацией. Линейная модель показала еще более стабильные результаты относительно предыдущих экспериментов. При этом точность классификации для небольших значений λ , $\lambda \in [0,1; 10]$ остается постоянной при удалении до 60% процентов параметров и равной $\approx 90\%$. Отклонения от среднего незначительные для небольших значений λ , $\lambda \in [0,1; 10]$. Далее данные модели были дообучены независимо от гиперсети в течение одной эпохи и эксперимент с прореживанием был запущен еще раз.

Таблица 2: Стабильность моделей

Модель	Стабильность $S(\lambda = 0, 1)$	Стабильность $S(\lambda = 1)$	Стабильность $S(\lambda = 10)$	Стабильность $S(\lambda = 100)$
а) Без гиперсети	2,209571	1,476841	1,194326	1,165526
б) Без гиперсети с дообучением	1,518516	1,537249	1,473905	1,362817
в) Гиперсеть (6)	1,208003	1,203938	1,177115	1,125796
г) Гиперсеть (6) с дообучением	1,20265	1,201760	1,186034	1,156385
д) Гиперсеть (7)	1,206345	1,205434	1,208482	1,216405
е) Гиперсеть (7) с дообучением	1,281112	1,287463	1,255834	1,289508

На рис. 1г показано, как меняется точность у дообученной модели с низкоранговой аппроксимацией при удалении параметров. Как видно из графика, после обучения точность классификации увеличилась, уменьшилось отклонение от среднего. Стабильность модели осталась прежней, и для всех значений λ точность значительно падает при удалении более 80% параметров. Для значения $\lambda = 100$ модель показала улучшение в точности классификации и большую

стабильность относительно предыдущей версии модели.

Таблица 3: Обобщенная обоснованность модели

Модель	\mathcal{L} ($\lambda = 0, 1$)	\mathcal{L} ($\lambda = 1$)	\mathcal{L} ($\lambda = 10$)	\mathcal{L} ($\lambda = 100$)
а) Без гиперсети	-9035,229	-24338,234	-56679,427	-128928,671
б) Без гиперсети с дообучением	-32788,838	-51832,864	-186124,637	-696240,551
в) Гиперсеть (6)	-24566,315	-30949,930	-56720,932	-166657,021
г) Гиперсеть (6) с дообучением	-19994,677	-27220,746	-55508,397	-132758,414
д) Гиперсеть (7)	-24603,767	-28189,602	-58147,425	-177139,477
е) Гиперсеть (7) с дообучением	-20776,473	-26262,996	-57948,826	-134340,962

На рис. 1е показано, как меняется точность у дообученной модели с линейной аппроксимацией при удалении параметров. Как видно из графика, после обучения точность классификации увеличилась для всех значений λ , значительно увеличилось отклонение от среднего при удалении более чем 60% параметров. Понижилась стабильность модели, но она по-прежнему выше, чем в экспериментах с полноранговой моделью. Для значения $\lambda = 100$ модель показала улучшения в точности классификации.

На рис. 1б показано, как меняется точность у дообученной в течение одной эпохи модели без гиперсети при удалении параметров. Из графика видно, что одной эпохи недостаточно для значительного улучшения модели, построенной напрямую.

Общие результаты экспериментов представлены в табл. 1, 2, 3. Несмотря на незначительную потерю в качестве, гиперсеть позволяет получить схожие результаты в сравнении с обычными моделями, но при существенно меньших вычислительных затратах. Более того, по графикам видно, что модель сохраняет схожие свойства при прореживании.

5 Заключение

В работе рассматривалась задача оптимизации модели глубокого обучения с наперед заданной сложностью. Итоговый метод заключался в представлении модели глубокого обучения в виде гиперсети. Использовался байесовский подход. Были введены вероятностные предположения о параметрах моделей глубокого обучения. По результатам вычислительного эксперимента можно сделать вывод о том, что модели на основе гиперсети имеют меньшую точность классификации, чем обычные модели. Однако при использовании гиперсети снижаются вычислительные затраты и сохраняются свойства моделей при прореживании.

В дальнейшем планируется исследовать теоретические свойства гиперсетей, а также улучшить предложенные модели для построения сетей глубокого обучения с контролем сложности.

Список литературы

- [1] *Graves A.* Practical Variational Inference for Neural Networks // Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems (NIPS 2011). P. 2348–2356.
- [2] *Ha D., Dai A. M., Le Q. V.* HyperNetworks // arXiv.org, 2016. arXiv:1609.09106 [cs.LG]. P. 1–29.
- [3] *Kuznetsov M. P., Tokmakova A. A., Strijov V. V.* Analytic and Stochastic Methods of Structure Parameter Estimation // Informatica, 2016. Vol. 27. P. 607–624.
- [4] *Bakhteev O. Yu., Strijov V. V.* Deep Learning Model Selection of Suboptimal Complexity // Automatics and Remote Control, 2018. Vol. 79. P. 1474–1488.

- [5] *Saxena S., Verbeek J.* Convolutional Neural Networks // Advances in Neural Information Processing Systems 29: 30th Annual Conference on Neural Information Processing Systems (NIPS 2016). P. 4053–4061.
- [6] *Xie S., Zheng H., Liu C., Lin L.* SNAS: Stochastic Neural Architecture Search // arXiv.org, 2019. arXiv:1812.09926 [cs.LG]. P. 1–17.
- [7] *Wu B., Dai X., Zhang P., Wang Y., Sun F., Wu Y., Tian Y., Vajda P., Jia Y., Keutzer K.* FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search // 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — IEEE, 2019. P. 10726–10734.
- [8] *Lorraine J., Duvenaud D.* Stochastic Hyperparameter Optimization through Hypernetworks // arXiv.org, 2018. arXiv:1802.09419 [cs.LG]. P. 1–9.
- [9] *LeCun Y., Cortes C., Burges C.* The MNIST dataset of handwritten digits, 1998. <http://yann.lecun.com/exdb/mnist/index.html>.