

# Обучение нейросетевых каскадных классификаторов <sup>1</sup>

Л.М. Тепляков, С.А. Гладилин, Е.А. Швец, Д.П. Николаев

*Институт проблем передачи информации, Российская академия наук, Москва, Россия  
email: teplyakov@iitp.ru, gladilin@iitp.ru, shvets@iitp.ru, dimonstr@iitp.ru*

Поступила в редколлегию 13.11.2018

**Аннотация**—Искусственные нейронные сети (ИНС) позволяют решить многие задачи компьютерного зрения с качеством, недостижимым для других методов. В то же время вычислительная сложность “глубоких” ИНС значительно ограничивает области их применимости. В данной работе мы предлагаем модель нейросетевого классификатора, имеющего каскадную архитектуру. Такая модель позволяет классифицировать простые примеры, не производя полного объема вычислений, за счет чего снижается среднее время классификации входного примера.

Кроме того, в работе впервые предлагается метод совместного обучения нескольких уровней каскадного нейросетевого классификатора. Для этого используется совместная функция потерь, которая включает член, отвечающий за сложность полученной модели, и позволяет регулировать соотношение скорости и точности построенной модели в процессе обучения.

Модель обучена на наборе данных CIFAR-10 предложенным методом. Показано, что результирующая модель является Парето-улучшением (по точности и быстродействию) модели, обученной традиционным способом.

**КЛЮЧЕВЫЕ СЛОВА:** сверточные нейронные сети, каскадные классификаторы, ускорение нейронных сетей, обучение с учетом сложности.

## 1. ВВЕДЕНИЕ

Современные исследования демонстрируют, что использование искусственных нейронных сетей (ИНС) позволяет решить многие актуальные проблемы компьютерного зрения с качеством, недостижимым для других методов (например, именно с помощью нейросетевых моделей достигается высочайшее качество классификации [1], детектирования [2], сегментации [3]). С точки зрения феноменологического моделирования возможностей ИНС (нахождения границ их применения и точности) такой результат впечатляет. Однако при внедрении ИНС в промышленные системы компьютерного зрения важнейшую роль играет вычислительная сложность модели. Беспилотные летательные или наземные аппараты, мобильные устройства, “умные” камеры зачастую должны решать задачи компьютерного зрения в реальном времени при наличии лишь маломощного вычислителя. В таких случаях невозможно использование “глубоких” ИНС, производящих сотни и тысячи миллионов операций [4, 5] для обработки одного изображения.

Несмотря на очевидную важность вычислительной сложности построенной модели, многие работы оценивают построенные нейросетевые модели исключительно по точности их ответов. Следствием такого подхода является усложнение архитектур ИНС и повышение их вычислительной сложности; а поэтому и их непригодность для решения многих промышленных задач,

<sup>1</sup> Исследование выполнено при частичной финансовой поддержке РФФИ в рамках научных проектов №17-29-03514 и №16-07-01167.

требующих исполнения в реальном времени на маломощном вычислителе. Например, широко используемая [5, 6] нейронная сеть VGG [4] осуществляет около 15 млрд. перемножений при обработке одного изображения. Чтобы время обработки полноразмерного изображения такой сетью не превысило 1 секунды, требуется мощный графический вычислитель (GPU). Обычно такое оборудование доступно на облачных сервисах обработки данных; но и такие сервисы заинтересованы в ускорении используемых моделей, поскольку это позволит снизить затраты на оборудование и электроэнергию.

Тренд на повышение точности ИНС в ущерб вычислительной сложности возник при первоначальном рывке развития ИНС, когда этот инструмент был новым, а в промышленных системах зачастую использовались традиционные методы компьютерного зрения. Однако в последние годы развитие робототехники, технологии самоходных автомобилей [7, 8], автоматических средств контроля безопасности способствовало широкому внедрению ИНС, и проблема обучения эффективных, но быстродействующих ИНС привлекла внимание исследователей [5, 9, 10, 11, 12].

Необходимость решать классические задачи компьютерного зрения (например, детектировать объекты [13, 15, 16], сегментировать [17, 18, 19] и классифицировать [20, 21, 22] изображения) возникла задолго до повсеместного распространения ИНС. При этом решение этих задач осуществлялось на вычислителях, значительно уступающих современным в мощности. Некоторые из методов, которые позволили добиться необходимого качества и скорости на таких маломощных вычислителях, вышли из широкого употребления после распространения ИНС, несмотря на свою эффективность.

Одним из таких методов является каскадный принцип построения распознающей или классифицирующей системы, нацеленный на понижение среднего времени работы системы при сохранении ее среднего качества за счет быстрого принятия решения для простых примеров.

Рассмотрим принцип работы каскадного алгоритма на примере широко известного детектора Виолы-Джонса [13]. Как правило, такой детектор последовательно анализирует примеры, полученные проходом скользящего окна по исходному изображению. При таком подходе количество входных примеров за единицу времени очень велико, но большинство из них не содержит целевого объекта, что делает особенно важным быстрое отсеивание негативных примеров.

Сам детектор Виолы-Джонса состоит из множества простых двуклассовых классификаторов (определяющих наличие целевого объекта на входном примере). Каждый из этих классификаторов, кроме последнего, имеет очень низкий показатель ложно-отрицательных срабатываний и предназначен для отсеивания примеров, которые являются “явно” отрицательными. Последний же уровень каскада обучен как сбалансированный классификатор. Поскольку входные примеры, отсеянные каким-либо из уровней каскада, немедленно признаются не содержащими объекта, то большинство негативных примеров будут отсеяны после нескольких уровней каскада. Это делает среднее время работы низким (худшее время работы достигается для примеров, дошедших до последнего уровня каскада).

Такая каскадная архитектура – эффективный метод снижения вычислительной сложности системы, и её использование позволяет компенсировать основной недостаток “глубоких” ИНС – их высокую вычислительную сложность [23, 2, 5]. Поэтому проблема “скрещивания” нейросетевых технологий и каскадного подхода в последние годы очень актуальна. Именно этой проблеме посвящена данная работа.

Глава 2 данной работы содержит обзор литературы; в главе 3 предлагается двухуровневая каскадную нейросетевую модель для многоклассовой классификации и алгоритм совместного обучения всех ИНС каскада, учитывающий среднюю сложность обученной системы. В главе 4

исследуются свойства предложенной модели на наборе изображений CIFAR10 [24], и обученная модель сравнивается с моделями, обученными существующими методами.

## 2. ОБЗОР ЛИТЕРАТУРЫ

Каскадный подход в классической работе Виолы-Джонса [13] заключается в быстром отсеивании негативных примеров последовательностью классификаторов с низким процентом ложно-положительных срабатываний. Самый простой способ включения ИНС в такую архитектуру заключается в замене последнего уровня каскада на нейросетевой классификатор [14].

Немного более сложным является замена классификаторов всех уровней каскада на искусственные нейронные сети. Этот принцип используют многие описанные в литературе каскадные нейросетевые модели [25, 26, 27]. Такие модели состоят из цепочки классифицирующих (на классы “объект” – “не объект”) ИНС. В зависимости от уверенности ответа ИНС  $i$ -ого уровня, входной пример либо отбрасывается, либо поступает на ИНС  $i + 1$ -ого уровня. Общим недостатком такой модели является их применимость только к задачам детекции или двуклассовой классификации.

Недостатком также является обработка изображения “с нуля” на каждом из уровней каскада (результаты вычислений на  $i$ -ом уровне не используются на  $(i + 1)$ -ом и последующих уровнях). Для борьбы с этой проблемой в работе [2] предлагается использовать на  $(i + 1)$ -ом слое признаки, генерируемые  $i$ -ым слоем (вместо исходного изображения). Каскадные свойства достигаются путем добавления ответвляющихся от основной ИНС классификаторов с малым числом параметров, которые могут классифицировать пример и остановить обработку входного примера.

Одним из наиболее впечатляющих результатов “скрещивания” нейросетевых технологий и каскадного принципа является нейросетевой детектор и классификатор “Faster R-CNN”, описанный в работе [5]. В этой нейросетевой модели изображение сначала обрабатывается сверточной нейронной сетью  $NN_{coarse}$ ; результат сканируется маленькой сверточной нейронной сетью (СНС)  $NN_{scan}$ , которая находит фиксированное число регионов, потенциально содержащих искомый объект. Затем регионы, потенциально содержащие объекты, обрабатываются более сложной СНС  $NN_{fine}$  для их классификации. Фактически, деление частей входного изображения, на те, которые следует обрабатывать, и те, которые следует отбросить, осуществляется с помощью сканирующей сети  $NN_{scan}$ .

В отличие от большинства описанных в литературе каскадных нейросетевых моделей, данная система осуществляет детектирование и классификацию объектов сразу нескольких классов. Однако компоненты этой системы тоже обучаются не в полном комплексе, и в процессе обучения не осуществляется оптимизация системы по быстродействию (число обрабатываемых  $NN_{fine}$  кусков изображения является фиксированным).

Обучение одной или нескольких ИНС каскада в отрыве от остальной системы является общим недостатком всех рассмотренных нами работ. Рассмотрим наиболее частую процедуру обучения, при которой каждая ИНС обучается индивидуально. Хотя задачей ИНС  $i$ -ого уровня является определение целесообразности передачи входного примера на следующий уровень каскада, ИНС  $i$ -ого уровня не обладает никакой информацией о  $(i + 1)$ -ом и последующих уровнях каскада. В данной статье мы показываем, что это приводит к неоптимальности итоговой каскадной модели.

При раздельном обучении ИНС каскада также невозможно непосредственно учитывать вычислительную сложность системы в процессе оптимизации. В данной работе мы предлагаем способ одновременного обучения всех ИНС каскада с использованием функции потерь, зависящей от оценки текущей средней вычислительной сложности системы. Это позволяет динамически изменять соотношение между точностью предсказания и вычислительной сложностью

в процессе обучения. Идея использования оценки сложности системы в функции потерь была предложена также в [28]; однако предложенный в этой работе метод не может быть применен при использовании ИНС в качестве классификаторов.

Подытоживая написанное выше: описанные в литературе каскадные нейросетевые модели зачастую обладают следующими недостатками:

1. Модели, полученные прямой адаптацией алгоритма Виолы-Джонса, пригодны только для детекции или бинарной (но не многоклассовой) классификации
2. При обучении каждая ИНС обучается независимо; хотя ИНС  $i$ -ого уровня непосредственно влияет на то, какие примеры будут подаваться на следующие уровни каскада, в процессе обучения она не обладает никакой информацией о  $(i + 1)$ -ом и последующих уровнях каскада. [25, 26].
3. В процессе обучения невозможен явный контроль желаемой средней скорости работы каскада. Функция потерь, используемая для обучения системы, не включает в себя слагаемые, зависящие от средней сложности обученного каскада; следовательно, при обучении не происходит непосредственной оптимизации итоговой скорости системы.
4. Результаты вычислений на первых  $i$  уровнях каскада не используются для вычислений на  $(i + 1)$ -ом и последующих уровнях.

В данной работе мы предлагаем очень простое развитие модели последовательных ИНС [25, 26] на случай многоклассовой классификации: вместо отбрасывания примера, каждая ИНС каскада осуществляет полноценную классификацию; если уверенность ИНС в ответе (элемент вектора выходного слоя, соответствующий выбранному классу) меньше порога, классификацию осуществляет следующий слой нейронной сети. Кроме того, мы предлагаем процесс совместного обучения всех ИНС каскада, решающий 2 и 3 проблемы приведенного выше списка.

### 3. ОПИСАНИЕ МОДЕЛИ

В задачах классификации больших разнородных наборов изображений наилучшее качество как правило достигается “глубокими” СНС [29, 30]. Прямое уменьшение числа или размера слоев СНС обычно снижает качество распознавания [4].

Однако и “легкие” модели правильно классифицируют большую часть изображений. Если, например, сложная и простая модели имеют точности 95% и 75% соответственно, то запускать тяжелую ИНС имеет смысл только в случае, когда легкая модель не может дать корректный ответ. Для реализации такого механизма, однако, необходимо уметь оценивать достоверность предсказания легкой ИНС. Непрямым индикатором достоверности предсказания является небинаризованный выход сети, соответствующий интересующему классу. Взаимосвязь достоверности предсказания и небинаризованного выхода ИНС исследована в работе [31]; показано, что эти величины коррелируют для СНС простых архитектур. Однако при совместном обучении всего каскада можно сделать зависимость более явной.

Рассмотрим каскад из двух ИНС. В режиме эксплуатации, если небинаризованное значение выхода первой сети каскада меньше некоторого порога, финальный ответ системы будет зависеть от результата работы второй ИНС, а время работы системы будет увеличено. В этом случае логично при обучении системы штрафовать первую ИНС системы за ошибку в ее ответе только если ее небинаризованный выход больше порога; а иначе, начислять штраф за увеличенную сложность каскада (и возможно, штрафовать вторую сеть за неправильный ответ). На этой идее и основана предлагаемая нами каскадная модель для многоклассовой классификации и метод ее обучения. Схема предлагаемой модели приведена на рис. 1.

В данной работе для простоты рассмотрим каскад, состоящий из двух ИНС. Определим термины, необходимые для формального описания предлагаемой модели. Пусть СНС  $NN$

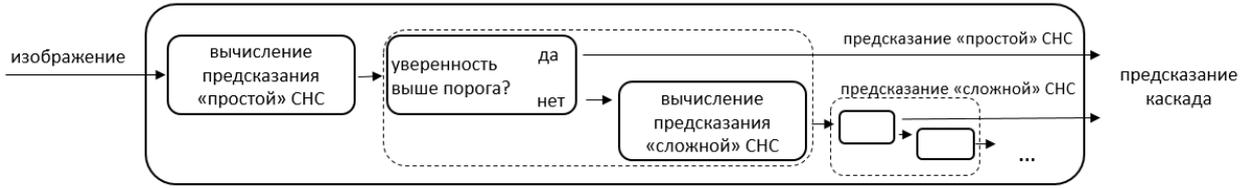


Рис. 1. Схема каскада

решает задачу классификации на  $n$  классов. Будем называть  $\vec{c} = NN(\theta, I)$  – предсказанием ( $\vec{c} \in \mathbb{R}^n, \sum_i c_i = 1, c_i \geq 0$ )  $NN$  для изображения  $I$ , где  $\theta$  – параметры  $NN$ ;  $p_{NN,\theta}(I) = \max_i c_i$  – уверенность СНС в предсказании;  $y_{NN,\theta}^*(I) = \operatorname{argmax}_i c_i$  – предсказанным классом. Для набора данных (пар изображение – класс)  $\{(I_0, y_0), \dots, (I_{m-1}, y_{m-1})\}$ ,  $y_i = 0, \dots, n - 1$  точность определяется как доля правильных ответов:  $\#\{y_{NN,\theta}^*(I_i) = y_i \mid i = 0, \dots, m - 1\} / m$ .

Пусть каскад состоит из двух СНС –  $NN_1$  (простая СНС) и  $NN_2$  (сложная СНС). Тогда предсказание каскада  $Y$ :

$$Y = \alpha y_{NN_1,\theta_1}^*(I) + (1 - \alpha) y_{NN_2,\theta_2}^*(I) \tag{1}$$

где  $\alpha = \mathbb{1}(P_{NN_1,\theta_1}(I) > \tau)$  определяет используемую СНС,  $\tau$  – порог уверенности.

В данном случае зависимость ответа сети может зависеть от входного изображения  $I$  скачкообразно (если уверенность  $P_{NN_1,\theta_1}(I)$  сети равна  $\tau$ ). Функция потерь зависит от ответа сети, поэтому нам необходимо создать гладкое приближение функции потерь и ответа ИНС. Для этого заменим скачкообразную функцию  $\alpha(P_{NN_1,\theta_1}(I))$  на её приближение  $\alpha^*$ .

В качестве функции потерь мы используем кросс-энтропию – логарифм уверенности, соответствующей правильному ответу:

$$L(NN(\theta, I), y) = -\log NN(\theta, I)_y \tag{2}$$

При обучении мы находим оптимальные параметры обеих нейросетей совместно. Функция потерь, гладко аппроксимирующая ошибку каскада, запишется в виде:

$$L_{pred} = \alpha^* L(NN_1(\theta_1, I), y) + (\alpha^* - 1) L(NN_2(\theta_2, I), y) \tag{3}$$

где:

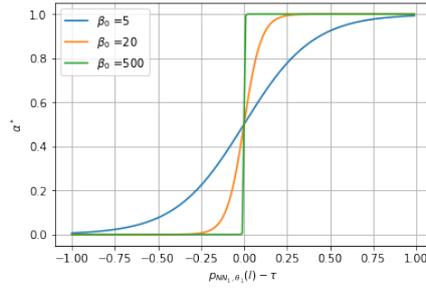
$$\alpha^* = [1 + \exp(-(\beta_0 + \beta_1 \cdot epoch)(p_{NN_1,\theta_1}(I) - \tau))]^{-1} \tag{4}$$

Где  $epoch$  – номер эпохи обучения.

Рассмотрим функцию  $\alpha^*$ . Она является дифференцируемым приближением  $\alpha$ . Коэффициенты  $\beta_0, \beta_1$  задают скорость приближения  $\alpha^*$  к ступенчатой функции в процессе обучения.

В рассмотренной функции потерь не учитывается вычислительная сложность компонент, а значит, системе выгодно всегда использовать сложную СНС, которая может добиться высокого качества, тогда как простая СНС должна выдавать низкую уверенность для всех изображений. Для учета вычислительной сложности мы добавляем член  $L_{time} = (1 - \alpha^*)p$ , где  $p$  – штраф за вычисление  $NN_2$ . При высокой уверенности первой СНС в предсказании  $L_{time}$  близка к нулю. Итоговая функция потерь:

$$L = L_{pred} + L_{time} = \alpha^* L(NN_1(\theta_1, I), y) + (\alpha^* - 1) L(NN_2(\theta_2, I), y) + (1 - \alpha^*)p \tag{5}$$



**Рис. 2.** Поведение  $\alpha^*$  при  $\beta_1 = 0$  и различных  $\beta_0$

## 4. ЭКСПЕРИМЕНТЫ

### 4.1. Архитектуры сетей

В работе рассматривался каскад из двух СНС, архитектуры которых представлены в табл. 1. В качестве функции активации скрытых слоев использовалась ELU [32]:

$$ELU(x) = \begin{cases} x, & x > 0 \\ e^x - 1, & x \leq 0; \end{cases}$$

а выходного слоя – softmax. При обучении мы использовали dropout [33] с вероятностью 0.2, 0.3, 0.4 после первого, второго и третьего max-pooling слоев, соответственно.

### 4.2. Используемые данные

В экспериментах использовался открытый набор данных CIFAR10 (см. рис. 3), состоящий из 50.000 обучающих и 10.000 тестовых цветных изображений 10 классов<sup>1</sup> размера 32x32.



**Рис. 3.** Используемые изображения. Слева – случайные изображения из набора CIFAR10, справа – случайные искажения изображения.

Перед поступлением в ИНС изображения были поканально нормализованы. По обучающим изображениям для каждого канала  $i \in \{R, G, B\}$  было рассчитано среднее  $\mu_i$  и дисперсия  $\sigma_i$ ; затем обучающие и тестовые изображения попиксельно были линейно преобразованы по закону:  $x_i^* = (x_i - \mu_i)/\sigma_i$ .

Обучающие данные искажались при подаче на обучение (рис. 3) при помощи следующих искажений:

<sup>1</sup> Используемые классы: самолет, автомобиль, птица, кошка, олень, собака, лягушка, лошадь, корабль, грузовик.

1. Зеркальное отображение по горизонтали с вероятностью 50%
2. Поворот на случайный угол  $[-5^0, 5^0]$
3. Горизонтальный сдвиг на случайное число пикселей  $n \in \{-5, \dots, 5\}$
4. Вертикальный сдвиг на случайное число пикселей  $n \in \{-5, \dots, 5\}$

Пикселям за границами исходного изображения присваивалась интенсивность ближайших пикселей границы изображения.

### 4.3. Обучение

Во всех экспериментах для обучения использовался метод оптимизации Adam [34] с параметрами по умолчанию, размер мини-батча составил 1024 примера. Была использована  $L_2$  регуляризация (по параметрам сети) с весом  $10^{-4}$ .

Для остановки и контроля скорости обучения анализировалась зависимость  $L_{val}(epoch)$  точности предсказаний сети на тестовом наборе от номера эпохи обучения. Если от момента достижения глобального минимума  $L_{val}(epoch)$  проходило 30 эпох без улучшения точности, скорость обучения уменьшалась в 2 раза, если проходило 50 эпох – обучение прекращалось.

Input $32 \times 32 \times 3$	Input $32 \times 32 \times 3$
Conv $3 \times 3 \times 32$	Conv $3 \times 3 \times 32$
Max-Pool	Conv $3 \times 3 \times 32$
Conv $3 \times 3 \times 64$	Max-Pool
Max-Pool	Conv $3 \times 3 \times 64$
fc 10	Conv $3 \times 3 \times 64$
	Max-Pool
	Conv $3 \times 3 \times 128$
	Conv $3 \times 3 \times 128$
	Max-Pool
	fc 10

**Таблица 1.** Архитектуры сетей. Слева – первая СНС каскада, справа – вторая.

### 4.4. Раздельное обучение

Простой метод обучения используемого нами каскада – раздельное обучение. Мы обучили каждую СНС каскада классифицировать набор CIFAR10, первая СНС достигла 70.58% точности на тестовом наборе, вторая – 90.54%. Затем из этих СНС мы сформировали каскад и для каждого порога уверенности вычислили долю ошибок каскада и долю вызовов второй СНС каскада. Результаты представлены на рис. 4.

Доля вызовов второй СНС однозначно определяет среднее время работы каскада:

$$T = t_{NN_1} + r * t_{NN_2}$$

где:  $T$  – среднее время работы каскада,  $t_{NN_1}$ ,  $t_{NN_2}$  – среднее время работы первой и второй СНС соответственно,  $r$  – доля вызовов второй СНС.

### 4.5. Сравнение с раздельным обучением

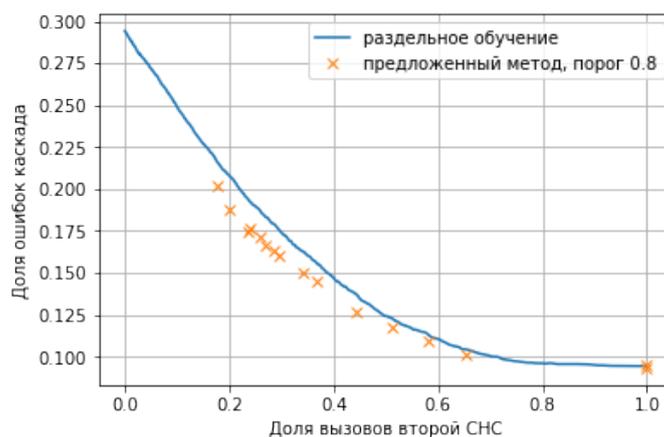
Для сравнения с раздельным обучением мы обучали каскад предложенным методом с различными штрафами, пороге  $\tau = 0.8$ , параметрами кривизны  $\beta_0 = 1$ ,  $\beta_1 = 0.1$ . Выбор порога описан в следующем разделе. Результаты эксперимента представлены на рис. 4, 5 и в табл. 2.

Штраф	дол.выз., %	дол. ошиб. каск., %	дол.ошиб. разд., %	абс. прир. %,	отн. прир. %
0.0	100.0	9.5	9.4	-0.1	-1.1
0.2	100.0	9.3	9.4	0.1	1.1
0.3	65.4	10.2	10.5	0.3	2.9
0.4	58.0	11.0	11.3	0.3	2.7
0.5	51.3	11.8	12.3	0.5	4.1
0.7	44.2	12.6	13.7	1.1	8.0
1.1	36.8	14.5	15.6	1.1	7.1
1.3	34.1	15.0	16.3	1.3	8.0
1.7	29.4	16.0	17.6	1.6	9.1
1.9	28.4	16.3	17.9	1.6	8.9
2.1	26.8	16.6	18.3	1.7	9.3
2.3	26.0	17.1	18.6	1.5	8.1
2.5	24.1	17.7	19.2	1.5	7.8
3.0	23.6	17.4	19.3	1.9	<b>9.8</b>
4.0	20.1	18.8	20.8	<b>2.0</b>	9.6
5.0	17.5	20.2	21.8	1.6	7.3

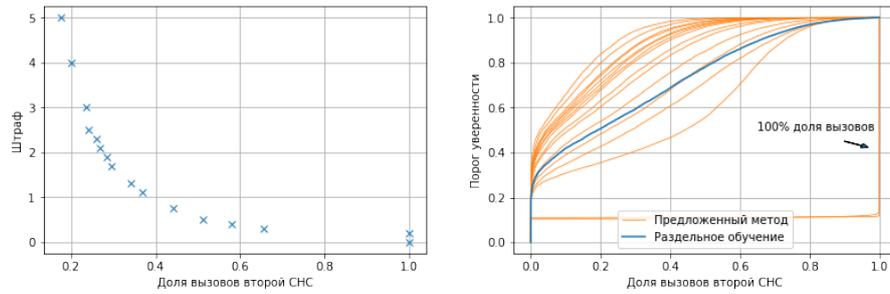
**Таблица 2.** Результаты обучения предложенным методом и сравнение с отдельным обучением по точности. Столбцы: штраф, доля вызовов второй СНС, доля ошибок каскада, доля ошибок при отдельном обучении при равной доле вызовов второй СНС, абсолютный прирост точности, относительный прирост точности

Предложенный метод обучения превосходит отдельное обучение по точности и скорости по Парето (за исключением каскада с нулевым штрафом) (рис. 4, табл. 2). Параметр  $p$  действительно играет роль штрафа – при росте  $p$  доля вызовов второй СНС уменьшается (рис. 5).

Из положительных свойств отдельного обучения можно отметить относительную простоту настройки под необходимую скорость, так как кривая зависимости доли вызовов от порога уверенности имеет широкий линейный участок (рис. 5). Для предложенного метода обучения зависимость доли вызовов от штрафа носит экспоненциальный характер (рис. 5). Кривые зависимости доли вызовов от порога уверенности для уже обученных каскадов представлены на рис. 5. Видно, что предложенный метод обучения при низких штрафах (0 и 0.2) сходится к “вырожденному” каскаду, в котором первая СНС всегда выдает уверенность 0.1.



**Рис. 4.** Сравнение методов обучения. Зависимость доли ошибок от скорости работы.



**Рис. 5.** Сравнение методов обучения. Слева – зависимость доли вызовов от штрафа для порога 0.8. Справа – зависимость доли вызовов от порога уверенности после обучения.

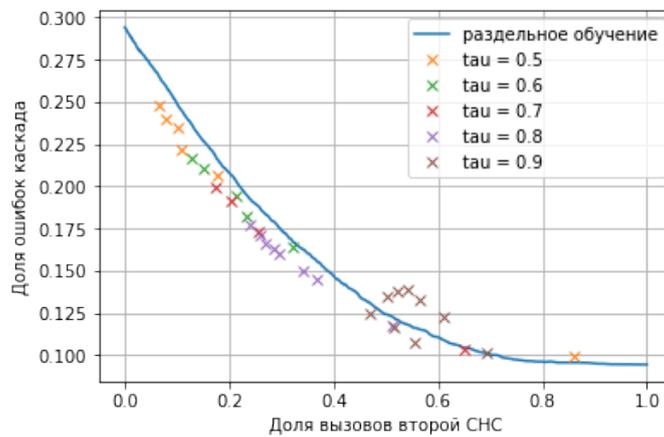
4.6. Оптимальные параметры обучения

Предложенный метод обучения обладает тремя параметрами: штрафом  $p$ , порогом  $\tau$  и параметрами зависимости кривизны (4) от эпохи  $\beta_0$  и  $\beta_1$ .

При  $\beta_1 = 0$  полученный каскад уступает каскаду, обученному в предыдущем разделе (табл. 3) и может уступать в качестве раздельному обучению. Выбор оптимального  $\beta_1$  при  $\beta_1 > 0$  может ускорить обучение, однако не должен влиять на полученный оптимум, поэтому мы остановились на линейной зависимости с низкой скоростью роста  $\beta_0 = 1, \beta_1 = 0.1$ .

Для анализа зависимости поведения от параметров  $p$  и  $\tau$  мы обучили каскады с порогами  $\tau = 0.3, 0.4, \dots, 0.9$  и штрафами  $p = 0.3, 0.5, \dots, 2.7, 2.9$ . Полученные результаты представлены на рис. 6.

Каскады при пороге 0.9 непредсказуемо зависят от штрафа и при некоторых штрафах их свойства хуже каскадов, полученных раздельным обучением. Остальные значения порога задают разные параметризации доли вызовов от штрафа, и при этом каждый порог обеспечивает схожий прирост точности. Поэтому мы остановились на пороге 0.8.



**Рис. 6.** Зависимость каскада от параметров.

5. ЗАКЛЮЧЕНИЕ

В данной работе предложена нейросетевая каскадная модель для многоклассовой классификации изображений, состоящая из нескольких классифицирующих нейронных сетей; передача

$\beta_0$	дол.выз., %	дол. ошиб. каск., %	абс. прир. %
500	37	17	-1.6
20	31	17	-0.3
5	33	16	0.1

Таблица 3. Каскад при  $p = 1.75$ ,  $\tau = 0.8$ ,  $\beta_1 = 0$ 

входного примера на следующий уровень каскада осуществлялась, если небинаризованный ответ сети (соответствующий выбранному классу) был меньше порогового значения.

Кроме того, предложен метод совместного обучения всех уровней такой модели. Для этого пороговая функция потерь каскада была приближена гладкой функцией, а также в функцию потерь было включено слагаемое, оценивающее вычислительную сложность сети.

Разработанная модель обучена и протестирована на наборе изображений CIFAR10; показано, что каскад, обученный предложенным методом, демонстрирует более высокое качество классификации и большую скорость работы по сравнению с каскадом, компоненты которого обучены отдельно.

### СПИСОК ЛИТЕРАТУРЫ

1. Krizhevsky, A., Sutskever, I., Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012, pp. 1097–1105.
2. Yang F, Choi W, Lin Y. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2129–2137.
3. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation *International Conference on Medical image computing and computer-assisted intervention*, 2015, p. 234–241.
4. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition *arXiv preprint arXiv:1409.1556*, 2014.
5. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2129–2137.
6. Kruthiventi S.S., Ayush K., Babu R.V. Deepfix: A fully convolutional neural network for predicting human eye fixations *IEEE Transactions on Image Processing*, 2017, p. 4446–4456.
7. Yang, Q., Yoo S.J. Optimal UAV Path Planning: Sensing Data Acquisition Over IoT Sensor Networks Using Multi-Objective Bio-Inspired Algorithms *IEEE Access*, 2018, vol. 6 pp. 13671–13684.
8. Paden, B., Čáp, M., Yong, S.Z., Yershov, D. and Frazzoli, E.. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles* 2016, vol. 1 no. 1, pp.33-55.
9. Ilin D., Limonova E., Arlazarov V., Nikolaev D., Fast integer approximations in convolutional neural networks using layer-by-layer training *Ninth International Conference on Machine Vision (ICMV 2016)*, 2016, pp. 103410Q.
10. Rastegari M., Ordonez V., Redmon J., Farhadi A., Xnor-net: Imagenet classification using binary convolutional neural networks *European Conference on Computer Vision*, 2016, p. 525–542.
11. Лимонова Е. Е., Шешкус А. В., Николаев Д. П., Иванова А. А., Ильин Д. А., Арлазаров В. Л. Оптимизация быстродействия первых слоев глубоких сверточных нейронных сетей *Вестник РФФИ* 2016, № 4, стр. 84-96.
12. Limonova E.E., Sheshkus A.V., Ivanova A.A., Nikolaev D.P. Convolutional Neural Network Structure Transformations for Complexity Reduction and Speed Improvement *Pattern Recognition and Image Analysis* 2018, vol. 28, no 1, pp. 24-33.

13. Viola P., Jones M., Rapid object detection using a boosted cascade of simple features *Proceedings of the 2001 Computer Vision and Pattern Recognition*, 2001, p.1–1.
14. Kuznetsova E., Shvets E., Nikolaev D., Viola-Jones based hybrid framework for real-time object detection in multispectral images, *Proceedings SPIE. Eighth International Conference on Machine Vision (ICMV 2015)* 2015, vol. 9875, 98750N, pp. 1-6.
15. Schneiderman H., Kanade T., A statistical method for 3D object detection applied to faces and cars *Computer Vision and Pattern Recognition*, 2000 p. 746–751.
16. Schneiderman H., Kanade T., A statistical Real-time object detection for "smart" vehicles *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999 p. 87–93.
17. Felzenszwalb P.F., Huttenlocher D.P., Efficient graph-based image segmentation, *International journal of computer vision*, 2004, p. 167–181.
18. Shi J., Malik J., Normalized cuts and image segmentation, *IEEE Transactions on pattern analysis and machine intelligence*, 2000, p. 888–905.
19. Lim Y.W., Lee S.U., On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques, *Pattern recognition*, 1990, p. 935–952.
20. Haralick, E.M., Shanmugam K., Textural features for image classification, *IEEE Transactions on systems, man, and cybernetics*, 1973, p. 610–621.
21. Chapelle O., Haffner P., Vapnik V.N., Support vector machines for histogram-based image classification, *IEEE transactions on Neural Networks*, 1999, p. 1055–1064.
22. Kato Z., Zerubia J., Berthod M., Satellite image classification using a modified Metropolis dynamics *Acoustics, Speech, and Signal Processing*, 1992, p.573–576.
23. Гладилин, С. А., Николаев, Д. П., Полевой, Д. В. Соколова, Н. А. Исследование возможности повышения точности распознавания нейронной сети при фиксированном числе активных нейронов. *Информационные технологии и вычислительные системы*, 2016, № 1, С. 96–105.
24. Krizhevsky A., Vinod N., Geoffrey H., The CIFAR-10 dataset. <http://www.cs.toronto.edu/kriz/cifar.html>.
25. Li H., Lin Z., Shen X., Brandt J., Hua G., A convolutional neural network cascade for face detection *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, p. 5325–5334.
26. Kalinowski I., and Spitsyn V., Compact Convolutional Neural Network Cascade for Face Detection *arXiv preprint arXiv:1508.01292*, 2015.
27. Zhang K., Zhang Z., Li Z., Qiao Y., Joint face detection and alignment using multitask cascaded convolutional networks *IEEE Signal Processing Letters*, 2016, p. 1499–1503.
28. Cai Z., Saberian M., Vasconcelos N., Learning complexity-aware cascades for deep pedestrian detection *Proceedings of the IEEE International Conference on Computer Vision*, 2015, p. 3361–3369.
29. Graham B., Fractional max-pooling *arXiv preprint arXiv:1412.6071* (2014).
30. Springenberg J.T., Dosovitskiy A., Brox T., Riedmiller M., Striving for simplicity: The all convolutional net *arXiv preprint arXiv:1412.6806*, 2014.
31. Guo C., Pleiss G., Sun Y., Weinberger K.Q., On Calibration of Modern Neural Networks *International Conference on Machine Learning*, 2017, p. 446–4456.
32. Clevert D., Unterthiner T., Hochreiter S., Fast and accurate deep network learning by exponential linear units (elus) *arXiv preprint arXiv:1511.07289*, 2015.
33. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*. 2014, vol. 15 no. 1, 1929–1958.
34. Kingma D.P., Ba J., Adam: A method for stochastic optimization *arXiv preprint arXiv:1412.6980*, 2014.

35. He K., Zhang X., Ren S., Sun J., Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, p. 770–778.
36. Girshick R., Fast r-cnn, *Proceedings of the IEEE international conference on computer vision*, 2015, p. 1440–1448.
37. Zeng D., Zhao F., Ge S., Shen W., Fast Cascade Face Detection with Pyramid Network *Pattern Recognition Letters*, 2018.
38. Deng J., Dong W., Socher R., Li L., Li K., Li F., Imagenet: A large-scale hierarchical image database *Computer Vision and Pattern Recognition*, 2009, p. 248–255.

## Training of Neural Network-based Cascade Classifiers

Тепляков Л., Гладиллин С., Шветс Е., Николаев Д.

Deep Artificial Neural Networks (ANNs) achieve state-of-art performance in many computer vision tasks; however, their applicability in the industry is significantly hindered by their high computational complexity. In this paper we propose a model of ANN classifier with cascade architecture, which allows to lower the average computational complexity of the system by classifying simple input samples without performing full volume of calculations.

We propose a method for joint optimization of all ANNs of the cascade. We introduce joint loss function that contains a term responsible for the complexity of the model and allows to control the ratio of the precision and speed of the resulting system.

We train the model on CIFAR-10 dataset with the proposed method and show that the resulting model is a Pareto improvement (regarding to speed and precision) compared to the model trained in a traditional way.

**KEYWORDS:** Convolutional Neural Networks, cascades, Neural network optimization, fast neural networks, complexity-aware training.