
ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

*МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ
МАШИН, КОМПЛЕКСОВ И КОМПЬЮТЕРНЫХ
СЕТЕЙ (05.13.11)*

УДК 004.032.26

DOI: 10.24160/1993-6982-2020-2-90-105

Анализ функций потерь при обучении сверточных нейронных сетей с оптимизатором Adam для классификации изображений

А.Н. Апарнев, О.В. Бартеньев

Рассмотрена задача анализа и определения функций потерь, используемых при обучении сверточных нейронных сетей (СНС) и обеспечивающих наибольшую точность классификации изображений. Проанализированы функции потерь, реализованные на основе библиотек машинного обучения Keras и TensorFlow, позволяющих проектировать, обучать и использовать СНС на языке Python. В качестве источников изображений для обучения и оценки СНС использованы наборы данных MNIST, EMNIST и CIFAR10. На указанных наборах обучаются СНС с различной архитектурой, для каждой из которых сохраняется история ее обучения (сведения о точности классификации изображений по обучающей и оценочной выборках после каждой эпохи обучения). Затем полученные результаты обобщаются. Наибольшая достигнутая точность классификации изображений позволяет обосновать целесообразность использования при обучении СНС соответствующих функций потерь.

Ключевые слова: нейронная сеть, функция потерь, набор данных, история обучения, точность классификации.

Для цитирования: Апарнев А.Н., Бартеньев О.В. Анализ функций потерь при обучении сверточных нейронных сетей с оптимизатором Adam для классификации изображений // Вестник МЭИ. 2020. № 2. С. 90—105. DOI: 10.24160/1993-6982-2020-2-90-105.

Analyzing the Loss Functions in Training Convolutional Neural Networks with the Adam Optimizer for Classification of Images

A.N. Aparnev, O.V. Barten'ev

The problem of analyzing and determining the loss functions used in training convolutional neural networks (CNN), which provide the highest accuracy of image classification, is considered. The loss functions implemented on the basis of the Keras and TensorFlow machine learning libraries, using which CNNs can be designed, trained, and applied in the Python language, are analyzed. The MNIST, EMNIST and CIFAR10 datasets are used as image sources for CNN training and validation purposes. These data sets are used to train CNNs having different architectures, for each of which the history of its training is preserved (information about the accuracy of image classification on training and validation samples after each training epoch). The evaluation results are then generalized. The highest achieved accuracy of image classification makes it possible to justify the expediency of using the relevant loss functions in training the CNNs.

Key words: neural network, loss function, data set, training history, classification accuracy.

For citation: Aparnev A.N., Barten'ev O.V. Analyzing the Loss Functions in Training Convolutional Neural Networks with the Adam Optimizer for Classification of Images. Bulletin of MPEI. 2020;2:90—105. (in Russian). DOI: 10.24160/1993-6982-2020-2-90-105.

Введение

В настоящее время для классификации и распознавания изображений широко используются сверточные нейронные сети (СНС). Первой сверточной сетью был Neocognitron Кунихико Фукусимы, появившийся в 1979 — 1980 гг. [1, 2]. В современном виде сверточные сети введены в обращение в конце 1980-х гг. группой Я. Лекуна [3 — 5].

Основная особенность СНС — наличие сверточных слоев. Активное применение искусственных нейронных сетей этого типа стимулировано развитием инструментальных программных средств, к которым, прежде всего, следует отнести библиотеки машинного обучения Keras [6] и TensorFlow [7], позволяющие удобно и просто в компактном виде (набором слоев) описывать, проектировать, обучать и использовать СНС на языке Python для решения различных прикладных задач.

С использованием библиотеки Keras отдельный слой СНС формируется в результате вызова конструктора (метода) соответствующего класса: создание одного сверточного слоя СНС сводится к передаче конструктору класса «Conv2D» 15 параметров [8], а при создании полносвязного слоя нейронов требуется передача конструктору класса «Dense» 10 параметров [9]. При этом большинство параметров можно задать по умолчанию, и лишь отдельные из них (число фильтров (*filters*) и размер окна свертки (*kernel_size*) для «Conv2D») определить в явном виде.

Такой подход позволяет быстро собрать первоначальный вариант СНС, возможности которой затем можно итерационно улучшать, меняя заданные по умолчанию значения остальных параметров.

Другим значительным стимулом к активному использованию СНС для классификации и распознавания изображений стало создание открытых наборов данных для удобного формирования обучающих и тестовых выборок. Так, в настоящей работе в качестве источников изображений для обучения и тестирования СНС использованы наборы данных MNIST, EMNIST-letters (далее — просто EMNIST) и CIFAR10 [10 — 12].

Одним из параметров СНС, от выбора которого существенно зависит точность классификации изображений, является функция потерь (ФП), используемая при обучении СНС [13].

При классификации изображений преимущественно используется ФП типа «перекрестная энтропия» [13 — 15]. Для решения таких задач в [16] предлагается использовать в качестве ФП «расстояния Коши–Шварца» («Cauchy–Schwarz Divergence»), а также «квадрата верхней границы» («Squared Hinge»). Однако выводы носят, в большей степени, предположительный (*Seems to be*) характер, что обусловлено выполнением экспериментов на небольшом числе сравнительно простых нейросетевых моделей, и нет уверенности в

том, что эти результаты могут быть распространены на действительно глубокие СНС.

Результаты поиска ФП, обеспечивающих наибольшую точность классификации изображений (такие ФП называют эффективными), представленные в [17], показывают, что для разных СНС эффективными являются разные ФП. На наборах данных MNIST и EMNIST в разных СНС эффективными оказались 15, а на CIFAR10 — 10 ФП из 16 примененных. Кроме того, для одной СНС могут быть эффективными несколько ФП. Это означает, что при проектировании СНС следует прежде оценить эффективность большого числа ФП и только после этого выбрать лучшую ФП из числа эффективных. Очевидный способ решения этой задачи — полный перебор всех ФП. Высокая трудоемкость указанного подхода делает актуальной проблему сокращения списка перебираемых ФП без ущерба качеству обучения СНС.

На примере решения задачи классификации изображений проведены экспериментальные исследования использования различных функций потерь при обучении СНС, реализованных в библиотеках Keras и TensorFlow. Сформулирована схема поиска ФП, эффективных при обучении проектируемой СНС, выработаны рекомендации по выбору лучшей ФП из числа эффективных.

Постановка задачи

Цель работы — поиск альтернативы полному перебору, который, согласно [17], следует выполнять при определении ФП, эффективных в проектируемой СНС.

Для достижения цели, во-первых, проведено обучение различных СНС с использованием ФП, реализованных в библиотеке Keras. Результаты обучения сохранены в виде историй обучения.

Далее решены следующие задачи.

Проверена возможность переноса результатов поиска ФП, эффективных в одной СНС, на другие СНС. Поскольку в общем случае для разных СНС эффективны различные ФП [17], то задача решена следующим образом: в множестве используемых СНС выделены группы СНС, такие, что СНС одной группы отличаются между собой одной характеристикой, например числом слоев Dropout. По историям обучения для каждой СНС выбранной группы сформированы списки эффективных ФП, которые затем проверены на предмет совпадения. Иными словами, исследована возможность переноса опыта обучения одной СНС на другую в пределах заданной группы.

Изучено, как будет меняться список ФП, эффективных в некотором множестве СНС, при расширении этого множества.

Проверена возможность переноса результатов поиска ФП, эффективных на одном наборе данных, на другой набор данных.

Результаты решения поставленных задач использованы при разработке схемы поиска эффективных ФП для вновь проектируемой СНС.

Учитывая, что в одной СНС могут быть эффективными несколько ФП, проанализирована задача выбора лучшей ФП из числа эффективных.

Приведены оценки возможных потерь в точности классификации при отказе от поиска эффективных ФП и применения взамен ФП «перекрестная энтропия».

Таким образом, последовательно решены задачи:

- обучения различных СНС с разными ФП для классификации изображений и фиксации их историй обучения;
- исследования возможностей переноса результатов поиска эффективных ФП в другие условия обучения;
- оценки потерь в точности классификации при отказе от поиска лучшей ФП в пользу «перекрестной энтропии».

Базовая структура используемой сверточной нейронной сети

На рисунке 1 представлена базовая структура СНС (для ее отображения использован метод «*plot_model*» библиотеки «*keras.utils*» [2]), являющаяся структурой известной СНС типа «LeNet-5» [3]. Перед именами слоев указаны их идентификаторы.

Сверточные слои нейронов СНС предназначены для выделения признаков. Слои нейронов подвыборки снижают размерность обрабатываемых данных. Два полносвязных слоя с последовательно уменьшающимся числом нейронов совмещают выделенные признаки.

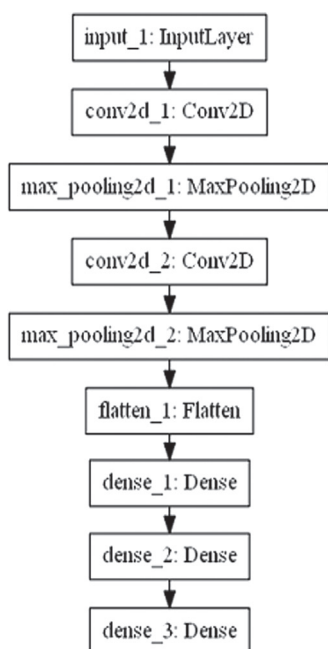


Рис. 1. Базовая структура СНС:

«InputLayer» — слой элементов входных данных; «Conv2D» — сверточный слой нейронов; «MaxPooling2D» — слой подвыборки; «Flatten» — слой, преобразующий многомерные данные в одномерные; «Dense» — полносвязный слой нейронов

Выходное значение выдает последний, классифицирующий слой СНС, число нейронов в котором равно числу классов изображений: 10 при использовании наборов данных MNIST и CIFAR10, 26 — в случае EMNIST.

Поскольку при обучении на наборе данных CIFAR10 СНС, построенные на основе базовой структуры, показали недостаточно высокую точность классификации изображений (не более 85,31%), то в рамках проводимых исследований дополнительно была использована СНС ResNet20v1, основанная на идее глубокого остаточного обучения [18, 19]. В работе [15] представлен программный код для формирования модели и обучения СНС ResNet20v1.

Используемые функции потерь

В таблице 1 даны ФП, используемые при обучении СНС. Первые 14 реализованы в библиотеке Keras [20], а остальные образованы на основе комбинирования типовых ФП из этой библиотеки.

Выражения для вычисления ФП библиотеки Keras и реализующие их программы указаны в [19].

Таблица 1

Функции потерь

| Функция потерь | Обозначение функции |
|--|---------------------|
| Средняя квадратическая ошибка (Mean Squared Error) | mse |
| Средняя абсолютная ошибка (Mean Absolute Error) | mae |
| Средняя абсолютная процентная ошибка (Mean Absolute Percentage Error) | mape |
| Средняя квадратическая логарифмическая ошибка (Mean Squared Logarithmic Error) | msle |
| Квадрат верхней границы (Squared Hinge) | sh |
| Верхняя граница (Hinge) | h |
| Категориальная верхняя граница (Categorical Hinge) | ch |
| Логарифм гиперболического косинуса (Logcosh) | lc |
| Категориальная перекрестная энтропия (Categorical Crossentropy) | cce |
| Разреженная категориальная перекрестная энтропия (Sparse Categorical Crossentropy) | scce |
| Бинарная перекрестная энтропия (Binary Crossentropy) | bce |
| Расстояние Кульбака-Лейблера (Kullback Leibler Divergence) | kld |
| Пуассон (Poisson) | pss |
| Косинусная близость (Cosine Proximity) | cp |
| cce + kld | ck |
| 17 * mse + kld | mk |

Описание наборов данных для обучения и оценки сверточных нейронных сетей

В качестве источников изображений для обучения и оценки СНС использованы наборы данных MNIST, EMNIST и CIFAR10. Набор данных MNIST содержит изображения рукописных цифр, выполненных в оттенках серого цвета [10], EMNIST — изображения рукописных букв английского алфавита также в оттенках серого цвета [11], CIFAR10 — цветные изображения десяти классов (самолеты, легковые автомобили, птицы, кошки, олени, собаки, лягушки, лошади, корабли, грузовики) [12]. Основные сведения о наборах данных приведены в табл. 2. Примеры изображений, а также программы загрузки наборов данных и воспроизведения изображений даны в [17].

Строковое представление сверточной нейронной сети

Строковое представление СНС вводится для ее компактного описания и содержит следующие обозначения:

- *CA_f*, *C_f* — сверточные слои с функцией активации *A* и без нее и размером выхода (числом фильтров), равным *f*;
- *R*, *L*, *S* — функции активации ReLU, Linear и Softmax, задаваемые как параметр слоя или как слой [21];
- *P*, *F* — слои подвыборки и Flatten;
- *DA_u*, *D_u* — полносвязные слои с функцией активации *A* и без нее и размером выхода (числом нейронов), равным *u*;
- *BN* — слой пакетной нормализации.

Пример. "c4: CR20-P-CR30-P-F-DR600-DL30-DS10".

В приведенной строке c4 — это имя СНС (сокращенное название СНС, используемое для ее идентификации).

В дальнейшем в строковом представлении СНС опустим обозначение последнего классифицирующего полносвязного слоя, поскольку он присутствует во всех СНС в неизменном виде: в случае MNIST и CIFAR10 — это DS10, а в случае EMNIST — DS26. По этой же причине в строковом представлении СНС отсутствует и обозначение входного слоя.

Часть используемых в эксперименте СНС содержит слои Dropout(*rate*) [9]. Присутствие слоя Dropout обозначается в строковом представлении СНС значением коэффициента прореживания *rate*.

Пример. В строке "c28: CR20-P-CR30-P-F-0.3-DR600-0.2-DL60-0.2" подстроки "0.3" и "0.2" — строковые представления значений параметра *rate* соответствующих Dropout-слоев.

Часть обучаемых СНС — сборки из двух (рис. 2) или трех СНС, построенных на основе базовой структуры (см. рис. 1).

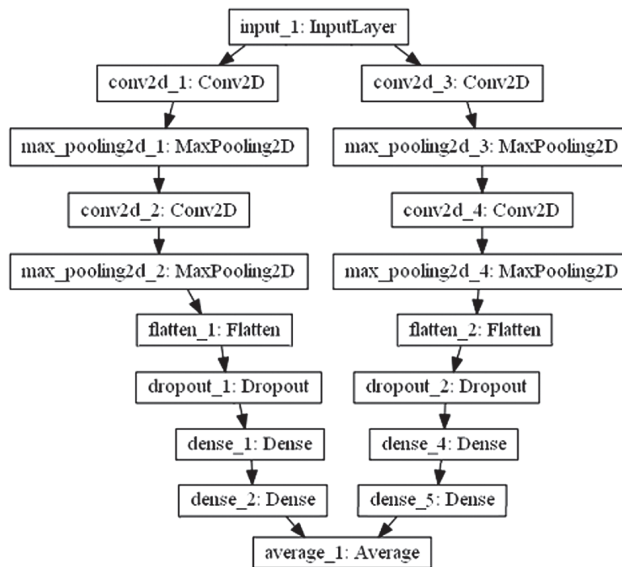


Рис. 2. Пример структуры сборочной нейронной сети

Пример обозначения: сборка СНС c4 с двумя ветвями имеет обозначение 2c4, а с тремя — 3c4.

В разработанной программе [17] СНС создается в результате разбора своего строкового представления.

Нейронные сети, используемые в эксперименте

Задействованные в эксперименте СНС — это (кроме ResNet20v1) модификации СНС с базовой структурой (см. рис. 1), получаемые за счет варьирования числом слоев Conv2D, Dense, Dropout и BatchNormalization, изменения значений параметров *filters*, *units* и *rate* слоев Conv2D, Dense и Dropout, а также применения сборок.

Описание работающих в эксперименте СНС приведено в табл. 3.

Поиск эффективных ФП выполнен с использованием СНС, представленных в списках обучения (табл. 4).

Таблица 2

Характеристика наборов данных

| Набор данных | Число классов | Число изображений | | | Размер изображения в пикселах |
|--------------|---------------|-------------------|------------|--------|-------------------------------|
| | | для обучения | для оценки | всего | |
| MNIST | 10 | 60000 | 10000 | 70000 | 28×28 |
| EMNIST | 26 | 124800 | 20800 | 145600 | 28×28 |
| CIFAR10 | 10 | 50000 | 10000 | 60000 | 32×32 |

Таблица 3

СНС, использованные в эксперименте

| Имя | Строковое представление | Число параметров | | |
|--------|--|------------------|-----------|-----------|
| | | MNIST | EMNIST | CIFAR10 |
| c1 | CR32-P-CR64-P-F-0.35-DR1024-0.2-DL30 | 3'276'724 | — | 4'245'780 |
| c3 | CR32-P-CR64-P-F-0.35-DR1024-0.2-DL52 | — | 3'300'342 | — |
| c9 | CR20-P-CR30-P-F-DR600-DL16 | 902'356 | — | — |
| c11 | CR20-P-CR30-P-F-0.25-DR600-DL16 | то же | | |
| c12 | CR20-P-CR30-P-F-0.25-DR600-0.2-DL16 | 902'356 | — | — |
| 2c12 | Сборка с двумя ветвями СНС c12 | 1'804'712 | 1'805'256 | — |
| 3c12 | Сборка с тремя ветвями СНС c12 | 2'707'068 | — | — |
| 2c12bn | Сборка СНС c12bn: C20-BN-R-P-C30-BN-R-P-F-0.25-DR600-BN-R-0.2-D16-BN-L | 1'807'376 | — | — |
| c13 | CR20-P-CR30-P-F-0.25-DR600-DL30 | 910'910 | — | — |
| c14 | CR20-P-CR30-P-F-0.25-DR600-0.2-DL30 | — | — | 1'176'930 |
| 2c14 | Сборка с двумя ветвями СНС c14 | — | — | 2'353'860 |
| 3c14 | Сборка с тремя ветвями СНС c14 | — | — | 3'539'790 |
| c15 | CR32-P-CR64-P-F-0.25-DR800-DL16 | 2'555'962 | — | 3'309'978 |
| c16 | CR32-P-CR64-P-F-0.25-DR800-0.2-DL16 | то же | | |
| c17 | CR32-P-CR64-P-F-0.25-DR800-DL30 | — | — | 3'321'332 |
| c18 | CR32-P-CR64-P-F-0.25-DR800-0.25-DL30 | то же | | |
| c19 | CR10-P-CR15-P-F-0.25-DR300 | — | — | 292'955 |
| c20 | CR10-P-CR15-P-F-0.3-DR300 | — | 231'211 | — |
| 2c20 | Сборка с двумя ветвями СНС c20 | | 462'422 | — |
| 3c20 | Сборка с тремя ветвями СНС c20 | | 693'633 | — |
| c24 | CR20-P-CR30-P-F-0.3-DR600-0.2-DL52-0.2 | — | 949'026 | — |
| 2c24 | Сборка с двумя ветвями СНС c24 | | 1'898'052 | — |
| c25 | CR20-P-CR30-P-F-DR600-DL60 | — | 930'216 | — |
| c26 | CR20-P-CR30-P-F-0.3-DR600-DL60 | то же | | |
| c27 | CR20-P-CR30-P-F-0.3-DR600-0.2-DL60 | то же | | |
| 2c27 | Сборка с двумя ветвями СНС c27 | | 1'860'432 | — |
| c28 | CR20-P-CR30-P-F-0.3-DR600-0.2-DL60-0.2 | — | 930'216 | — |
| 2c28 | Сборка с двумя ветвями СНС c28 | | 1'860'432 | — |
| 2c28bn | Сборка СНС c28bn: C20-BN-R-P-C30-BN-R-P-F-0.3-D600-BN-R-0.2-DL60-BN-R-0.2 | — | 1'863'272 | — |
| c33 | CR16-CR16-P-0.2-CR32-CR32-P-0.25-F-0.25-DR512-0.3-DL32-0.2 | — | — | 1'082'490 |
| c34bn | C32-BN-R-C32-BN-R-P-0.25-C64-BN-R-C64-BN-R-P-0.25-F-D512-BN-R-0.5 | 1'728'074 | 1'736'282 | 2'169'770 |
| r_n | ResNet20v1 | — | — | 273'066 |

Таблица 4

Списки обучения

| Набор данных | Имя СНС, входящей в список обучения |
|--------------|---|
| MNIST | c1, c9, c11, c12, c13, c15, c16, c34bn, 2c12, 2c12bn, 3c12 |
| EMNIST | c3, c24, c28, c34bn, 2c12, 2c20, 2c24, 2c27, 2c28, 2c28bn, 3c12 |
| CIFAR10 | c1, c14, c15, c16, c17, c18, c33, c34bn, 2c14, 3c14, r_n |

Отчет об обучении указанных в табл. 3. СНС имеется в [17].

Параметры нейронных сетей

Приведенные ниже значения параметров используются во всех СНС и их сборках.

- Во всех сверточных слоях:
 - функция активации ReLU;
 - размер окна фильтра $kernel_size = 3$ — если CIFAR10, $kernel_size = 4$ — в противном случае;
 - шаг окна фильтра $strides = 1$;
 - способ заполнения $padding = 'same'$.
- Во всех слоях подвыборки:
 - размер окна подвыборки $pool_size = 2$;
 - шаг окна подвыборки $strides = 2$;
 - способ заполнения $padding = 'same'$.
- Во всех слоях Dropout параметр $seed = 348$.
- Во всех слоях Conv2D и Dense:
 - инициализатор весов $k_init = RandomNormal(seed = 348)$;
 - $use_bias = True$ (к выходам добавляется смещение);
 - $bias_initializer = k_init$ (инициализация вектора смещения);
 - $kernel_initializer = k_init$ (инициализация тензора весов).
- Первый полносвязный слой СНС снабжен функцией активации ReLU.
- В случае трех полносвязных слоев второй слой имеет функцию активации Linear.
- Выходной слой всех СНС является полносвязным с функцией активации Softmax и числом нейронов, равным числу классов в наборе данных.
- Использован оптимизатор Adam с заданными по умолчанию аргументами (алгоритм оптимизации Adam [22] — модификация адаптивного алгоритма оптимизации Adagrad, но применяет в отличие от последнего сглаженные версии среднего и среднеквадратичного градиентов).
- Размер пакета обучения $batch_size = 256$.
- Число эпох обучения $epochs = 120$.
- Ранняя остановка: $EarlyStopping(monitor = 'loss', patience = 9)$.
- Выходы ветвей сборки усредняются в результате использования слоя Average [23].

Замечание. Применение ранней остановки с параметрами $monitor = 'loss'$ и $patience = 9$ означает, что обучение будет прервано, если в течение девяти подряд идущих эпох не снижались потери на обучающих данных.

Особенности применения ResNet20v1 [15].

- В слоях Conv2D использован инициализатор he_normal .
- Размер пакета обучения $batch_size = 128$.
- Число эпох обучения $epochs = 200$.
- Использована аугментация данных: СНС обучается на данных, сгенерированных ImageDataGenerator

[24], выполняющим случайные горизонтальный и вертикальный сдвиги изображений обучающей выборки и случайный поворот изображений на 180° относительно оси y , проходящей через центр изображения.

Используемые понятия

Введем следующие понятия:

Список ФП — содержит имена ФП, используемых для обучения СНС.

Эпизод обучения — обучение СНС с одной ФП на заданном наборе данных.

Цикл обучения — реализация эпизодов обучения со всеми ФП из списка ФП.

Серия обучения — выполнение циклов обучения всех СНС из списка обучения (см. табл. 4), т. е. обучение всех СНС со всеми ФП на взятом наборе данных.

Частное решение — результат обучения СНС с заданной ФП после каждой эпохи. Для каждого частного решения вычисляются $acc_{чр}$ и $val_acc_{чр}$ — значения точности классификации обучающих и оценочных данных, наблюдаемые после завершения эпохи обучения.

История обучения — два текстовых файла, формируемых в процессе выполнения эпизода обучения. Один файл содержит значения $acc_{чр}$, второй — $val_acc_{чр}$, вычисляемые после получения частного решения. Число записей в каждом файле истории обучения СНС равно числу эпох обучения.

Критерий выбора решения — показатель, по значению которого из множества решений выбирается лучшее. В работе использован традиционный критерий выбора решения val_acc — точность классификации изображений оценочных данных [13, 14]. В качестве лучшего выбирается решение с максимальным значением критерия.

ФП-решение — лучшее частное решение, определяемое по истории обучения. Описание ФП-решения задается в виде следующего списка:

[Имя СНС, Имя ФП, $val_acc_{фп}$, Имя набора данных],

где $val_acc_{фп}$ — максимальное значение $val_acc_{чр}$ в истории обучения.

Пример. Список [2c28, pss, 94.89, EMNIST] означает, что после обучения СНС 2c28 на EMNIST с ФП pss имеем $val_acc_{фп} = 94.89$.

Замечания. Здесь и далее точность указана в процентах.

ЦО-решение — лучшее ФП-решение цикла обучения. Определяется по историям обучения.

Пример. В цикле обучения СНС 2c28 на EMNIST лучшим ФП-решением является [2c28, 1c, 95.03] (в описании ФП-решения «Имя набора данных» опускается, если оно указано в контексте).

СО-решение — лучшее ФП-решение серии обучения, определяемое по историям обучения.

Пример. В серии обучения на EMNIST лучшим ФП-решением считается [c34bn, msle, 95.43].

Лучшая СНС списка обучения — СНС лучшего ФП-решения серии обучения.

Среднее ФП-решение серии обучения определяется по ФП-решениям серии обучения с заданной ФП. Описание среднего ФП-решения задается в виде следующего списка:

[Имя ФП, $val_acc_{\text{ФП}_C}$, Имя набора данных],

где $val_acc_{\text{ФП}_C}$ — среднее значение $val_acc_{\text{ФП}}$ всех ФП-решений серии обучения с заданной ФП.

Пример. Применение *msle* в серии обучения на EMNIST обеспечивает следующие ФП-решения: [c34bn, 95.43], [2c28bn, 95.25], [2c24, 94.91], [2c28, 94.9], [2c27, 94.85], [c24, 94.82], [2c12, 94.75], [3c12, 94.67], [c28, 94.67], [2c20, 94.65], [c3, 94.6]. Среднее значение критерия в приведенном списке равно 94.86, поэтому среднее ФП-решение с *msle* на EMNIST имеет следующее описание: [*msle*, 94.86, EMNIST].

Приоритетное решение — ФП-решение с $val_acc_{\text{ФП}} > val_acc_{\text{ФП}_C_max}$, где $val_acc_{\text{ФП}_C_max}$ — максимальное значение $val_acc_{\text{ФП}_C}$ в серии обучения.

Пример. После выполнения серии обучения на EMNIST по историям обучения найдено лучшее среднее ФП-решение [lc, 94.88]. Следовательно, ФП-решение [2c24, mk, 94.92] — приоритетное, а [2c28, sh, 94.83] таковым не является.

Список приоритетных решений — список, содержащий описания приоритетных решений.

Измерение значения критерия — выполнение эпизода обучения с последующим поиском $val_acc_{\text{ФП}}$.

Погрешность измерения значения критерия — погрешность, определяемая по результатам нескольких измерений $val_acc_{\text{ФП}}$.

Лучшая функция потерь цикла обучения (серии обучения, среднего ФП-решения) — ФП, обеспечившая лучшее ФП-решение в цикле обучения (серии обучения, средних ФП-решениях).

Схема эксперимента

Сбор данных, по которым можно судить об эффективности ФП, перечисленных в табл. 1, осуществляется в результате выполнения трех серий обучения по следующей схеме (на каждом наборе проводится одна серия).

1. Задание набора данных.
2. Выбор СНС из списка обучения набора.
3. Реализация цикла обучения выбранной СНС.
4. Повтор пунктов 2, 3 для каждой СНС списка обучения (каждая СНС выбирается в списке обучения единожды, порядок выбора значения не имеет).

Цикл обучения состоит из шестнадцати эпизодов (по числу используемых ФП). При реализации эпизода обучения сохраняется история обучения СНС. Та-

ким образом, после завершения цикла обучения имеем шестнадцать историй обучения (по числу применяемых ФП). После завершения серии обучения для обработки доступно $16nN$ историй обучения, где nN — число имен СНС в списке обучения.

Замечание. Все сохраненные истории обучения доступны в [17].

Значение $val_acc_{\text{ФП}}$, получаемое после завершения эпизода обучения и используемое для оценки качества обучения СНС, можно в условиях проводимого эксперимента употребить и для оценки эффективности ФП, поскольку в цикле обучения единственным изменяемым параметром СНС является ФП. При этом ФП считаются одинаково эффективными, если в полученных ФП-решениях $val_acc_{\text{ФП}}$ различаются на величину, меньшую погрешности измерения значения критерия.

Погрешность измерения значения критерия

При неоднократном обучении одной и той же СНС получаются решения с различными значениями $val_acc_{\text{ФП}}$. Это объясняется тем, что некоторые методы, например инициализаторы весов, используют генератор случайных чисел. Кроме того, при обучении при параллельном выполнении операций не гарантирован их порядок. Поэтому из-за ограниченной точности вычислений даже сложение нескольких чисел может дать разные результаты при разных порядках сложения [25].

Таким образом, выполнив некоторое число измерений значений критерия, можно вычислить Δ — погрешность измерения $val_acc_{\text{ФП}}$.

Наличие такой погрешности позволяет считать ФП одинаково эффективными, если они обеспечивают решения с значениями критерия, отличающимися на величину, не превышающую Δ .

Ввиду наличия погрешности измерения значения критерия в работе для каждой СНС выполняется поиск списка эффективных ФП. В него входят ФП, обеспечившая решение с наибольшим значением критерия ($val_acc_{\text{ФП}_max}$), и все ФП, обеспечившие решение с $val_acc_{\text{ФП}}$, таким, что $val_acc_{\text{ФП}_max} - val_acc_{\text{ФП}} \leq \Delta$.

Оценки Δ представлены в [17]. Они, как и точность классификации, даны в процентах и равны 0,024% в случае MNIST; 0,048% — в случае EMNIST и 0,217% — в случае CIFAR10.

Задачи обработки историй обучения

Одна из задач настоящей работы состоит в том, чтобы выяснить, можно ли переносить опыт поиска эффективных ФП в одной СНС на другую. Для достижения поставленной цели выделим группы СНС, такие, что СНС одной группы отличаются между собой одной характеристикой, например числом слоев Dropout. Затем изучение возможности переноса опыта поиска эффективных ФП в одной СНС на другую выполним в пределах группы. Положительный результат, если он будет выявлен, позволит существенно снизить трудоемкость проектирования СНС: при обучении мо-

дифицированной СНС, например, с добавленным словом Dropout можно будет использовать ФП, эффективность которых подтверждена при обучении СНС без этого слоя. Так, если изначально проверяются 16 ФП, а число выявленных эффективных ФП равно двум, то трудоемкость проектирования модифицированной модели уменьшается в 8 раз.

Аналогична и задача изучения возможности переноса опыта поиска ФП, эффективных на наборе данных, на другой набор данных. Ее наличие и информация об ФП, эффективных, например, на MNIST, позволит использовать эти ФП, например, на EMNIST.

Предварительно для достижения целей в результате обработки историй обучения формируются следующие списки:

- эффективных ФП цикла обучения;
- эффективных ФП серии обучения.

Информация об эффективных ФП цикла обучения позволит выяснить возможность переноса результатов поиска эффективных ФП в пределах одной из описанных ниже групп СНС.

Нейронные сети, задействованные в эксперименте (см. табл. 3), различаются структурой и такими параметрами, как размеры выходов слоев, коэффициенты прореживания слоев Dropout и используемыми ФП. Прочие параметры не меняются.

Выделим следующие группы СНС (при описании группы берутся только характеристики, различающие СНС, прочие характеристики СНС группы совпадают):

- различающиеся числом слоев Dropout (c25, c26, c27, c28);
- различающиеся значением параметра *rate* слоя Dropout (c19, c20);
- различающиеся числом выходов слоев (c12, c14, c16, c18);
- различающиеся числом ветвей (c12, 2c12, 3c12);
- с пакетной нормализацией и без нее (2c12bn, 2c12).

Имея в виду приведенные группы СНС, сформулируем следующую задачу обработки накопленных в эксперименте историй обучения: можно ли переносить результаты поиска ФП, эффективных в одной СНС заданной группы, на другую СНС той же группы (набор данных фиксирован).

Информация об эффективных ФП серии обучения позволяет решить задачи:

- переноса результатов поиска эффективных ФП на одном наборе данных на другой;
- прогноза, какие ФП окажутся эффективными для вновь обучаемой СНС.

Полученные при решении этих задач результаты используются при разработке схемы поиска ФП, эффективных в обучаемой СНС.

Формирование списка эффективных функций потерь цикла обучения

Список эффективных ФП цикла обучения формируется по следующему алгоритму.

Входные данные: имя набора данных, имя СНС, для которой выполняется цикл обучения, истории обучения СНС.

Выходные данные: ФП_ЦО — список эффективных ФП цикла обучения.

1. ФП_ЦО = []

2. Для Имя ФП из Списка ФП:

По истории обучения СНС с функцией ФП найти $val_acc_{ФП}$

Добавить в ФП_ЦО список [Имя ФП, $val_acc_{ФП}$].

3. Упорядочить ФП_ЦО по убыванию $val_acc_{ФП}$

4. Оставить ФП_ЦО только эффективные ФП, то есть ФП, для которых выполнено следующее условие:

$val_acc_{ФП_max} - val_acc_{ФП} \leq \Delta$, где $val_acc_{ФП_max}$ — значение критерия первой ФП списка; Δ — погрешность измерения $val_acc_{ФП}$.

Пример. ФП_ЦО на MNIST для СНС c9 ($\Delta = 0,024$).

После п. 3 алгоритма:

ФП_ЦО = [[bce, 99.55], [sccce, 99.45], [kld, 99.44], ..., [mare, 99.26]]

$val_acc_{ФП_max} = 99.55$.

После п. 4 алгоритма:

ФП_ЦО = [[bce, 99.55]]

Список ФП_ЦО содержит ФП, эффективные в заданной СНС.

Перенос опыта использования функции потерь в другие условия обучения нейронной сети

Оценка возможности переноса результатов поиска эффективных ФП в цикле обучения одной СНС на другую выполнена в результате сравнения списков ФП_ЦО, сформированных для СНС одной группы.

Рассмотрим случай применения ФП в СНС, отличающихся только числом слоев Dropout (такие СНС имеют одинаковое число весов). Укажем в табл. 5 для каждой СНС эффективные ФП.

Если эффективных ФП более одной, то в последнем столбце этой и последующих таблиц указывается диапазон изменения $val_acc_{ФП}$.

Из имеющихся в табл. 5 данных следует, что при изменении в СНС числа Dropout-слоев при прочих равных условиях в общем случае меняются и ФП, обеспечивающие лучшее качество обучения.

В таблице 6 приведены списки эффективных ФП в СНС, обученных на EMNIST, с одинаковой структурой, но разными значениями параметра *rate* слоя Dropout.

Достигнутые результаты говорят о том, что при изменении значения параметра *rate* при прочих равных условиях поиск эффективных ФП следует выполнить заново.

В таблице 7 даны списки эффективных ФП в СНС, обученных на MNIST, с одинаковой структурой, но разным числом выходов в используемых слоях.

Ясно, что в данном случае перенос опыта применения ФП для обучения одной СНС на другую недопустим.

Таблица 5

ФП, эффективные в СНС с различным числом Dropout-слоев

| Нейронная сеть | Эффективные ФП | $val_acc_{\text{ФП}}$ |
|---|--------------------|------------------------|
| MNIST | | |
| c9: CR20-P-CR30-P-F-DR600-DL16 | bce | 99.55 |
| c11: CR20-P-CR30-P-F-0,25-DR600-DL16 | kld, mae | 99.51...99.53 |
| c12: CR20-P-CR30-P-F-0,25-DR600-0,2-DL16 | cce, scce, cp | 99.53...99.55 |
| EMNIST | | |
| c25: CR20-P-CR30-P-F-DR600-DL60 | lc | 93.91 |
| c26: CR20-P-CR30-P-F-0,3-DR600-DL60 | mse, scce, msle | 94.45...94.49 |
| c27: CR20-P-CR30-P-F-0,3-DR600-0,2-DL60 | lc | 94.72 |
| c28: CR20-P-CR30-P-F-0,3-DR600-0,2-DL60-0,2 | mse | 94.85 |
| CIFAR10 | | |
| c17: CR32-P-CR64-P-F-0,25-DR800-DL30 | scce, kld | 74.70...74.89 |
| c18: CR32-P-CR64-P-F-0,25-DR800-0,2-DL30 | scce, kld, pss, mk | 75.41...75.61 |

Таблица 6

ФП, эффективные в СНС с разными значениями параметра rate

| Нейронная сеть | Эффективные ФП | $val_acc_{\text{ФП}}$ |
|---------------------------------|----------------|------------------------|
| c19: CR10-P-CR15-P-F-0,25-DR300 | lc, msle | 94.38...94.42 |
| c20: CR10-P-CR15-P-F-0,3-DR300 | sh, mse | 94.46...94.49 |

Таблица 7

ФП, эффективные в СНС одной структуры, но с разным числом выходов в сверточных и/или полносвязных слоях

| Нейронная сеть | Эффективные ФП | $val_acc_{\text{ФП}}$ |
|--|---------------------------------|------------------------|
| c12: CR20-P-CR30-P-F-0,25-DR600-0,2-DL16 | cce, scce, cp | 99.53...99.55 |
| c14: CR20-P-CR30-P-F-0,25-DR600-0,2-DL30 | cp, kld, ck, msle, sh, bce, cce | 99.50...99.52 |
| c16: CR32-P-CR64-P-F-0,25-DR800-0,2-DL16 | kld, scce | 99.51...99.53 |
| c18: CR32-P-CR64-P-F-0,25-DR800-0,2-DL30 | msle, scce, kld, mk, cp, cce | 99.45...99.47 |

В таблице 8 показаны списки эффективных ФП в СНС c12 и c20, а также в сборках из этих СНС с двумя и тремя ветвями. В данном случае при переходе от линейной СНС к сборочной меняется состав эффективных ФП.

В таблице 9 приведены списки эффективных ФП в СНС 2c12 и 2c28, а также в таких же СНС, но с пакетной нормализацией.

Вывод очевиден: после введения слоя пакетной нормализации поиск эффективных ФП следует повторить.

Таким образом, в пределах одного набора данных не удастся найти ни одного случая, когда можно перенести результаты поиска эффективных ФП в одной СНС на другую.

Формирование списков эффективных функций потерь серии обучения

При обработке историй обучения формируются три списка ФП, эффективных в серии обучения:

- ФП_{СО} — содержит эффективные ФП, отобранные по $val_acc_{\text{ФП}_C}$ СО-решений серии обучения;
- ФП_{СПР} — включает эффективные ФП, найденные по списку приоритетных решений в серии обучения;
- ФП_{ЦО} — содержит эффективные ФП цикла обучения, обеспечившего лучшее ФП-решение в серии обучения.

Общее представление об эффективности ФП в сериях обучения можно получить, анализируя данные табл. 10, в которой для каждой ФП указаны среднее значение ($val_acc_{\text{ФП}_C}$) и диапазон изменения $val_acc_{\text{ФП}}$ в серии обучения. Записи отсортированы по убыванию $val_acc_{\text{ФП}_C}$.

Список ФП_{СО} включает ФП, для которых выполняется следующее условие:

$$val_acc_{\text{ФП}_C \text{ max}} - val_acc_{\text{ФП}_C} \leq \Delta,$$

где $val_acc_{\text{ФП}_C \text{ max}}$ — наибольшее значение $val_acc_{\text{ФП}_C}$ в серии обучения; Δ — погрешность измерения $val_acc_{\text{ФП}}$.

Таблица 8

ФП, эффективные в СНС c12 и c20 и сборках на их основе

| Имя СНС | Эффективные ФП | $val_acc_{ФП}$ |
|---------|--------------------------------------|-----------------|
| MNIST | | |
| c12 | cce, scce, cp | 99.53...99.55 |
| 2c12 | ck, msle, cp, bce, kld, mse | 99.56...99.58 |
| 3c12 | cce, ck, kld, cp, pss, h, scce, msle | 99.55...99.57 |
| EMNIST | | |
| c20 | sh, mse | 94.46...94.49 |
| 2c20 | sh, lc | 94.71...94.74 |
| 3c20 | cp, msle, lc, pss | 94.69...94.73 |

Таблица 9

ФП, эффективные в СНС 2c12, 2c12bn, 2c28 и 2c28bn

| Имя СНС | Эффективные ФП | $val_acc_{ФП}$ |
|---------|-----------------------------|-----------------|
| MNIST | | |
| 2c12 | ck, msle, cp, bce, kld, mse | 99.56...99.58 |
| 2c12bn | mapc, sh, h | 99.62...99.64 |
| EMNIST | | |
| 2c28 | lc | 95.03 |
| 2c28bn | msle, h, scce, mae | 95.21...95.25 |

Пример. В случае CIFAR10 ($\Delta = 0,217$) в ФП_CO окажутся следующие ФП: bce, cce, pss, cp, scce, mk.

Список ФП_CO формируется по средним значениям $val_acc_{ФП}$, которые могут серьезно отличаться от максимальных значений этого показателя. Например, в случае CIFAR10 и все эта разница равна $91.30 - 77.38 = 13.92$.

Для ФП, включенных в список ФП_СПР, составленному по списку приоритетных решений, это различие меньше, поскольку в СПР попадает описание ФП-решения ФП данного списка $val_acc_{ФП} > val_acc_{ФП_C_max}$.

Список приоритетных решений (СПР) формируется по следующему алгоритму:

Входные данные: имя набора данных, список и истории обучения набора данных, $val_acc_{ФП_C_max}$.

Выходные данные: СПР.

1. СПР = []

2. Для Имя СНС из Списка Обучения:

Для Имя ФП из Списка ФП: // В списке 16 имен ФП

Найти по историям обучения $val_acc_{ФП}$.

Если $val_acc_{ФП} > val_acc_{ФП_C_max}$:

Добавить в СПР список [Имя СНС, Имя ФП, $val_acc_{ФП}$].

3. Упорядочить СПР по убыванию $val_acc_{ФП}$.

Пример. СПР на MNIST; $val_acc_{ФП_C_max} = 99.52$ (см. табл. 10):

[c34bn, msle, 99.71], [c34bn, pss, 99.7], ...,
 [2c12bn, mapc, 99.64], [2c12bn, sh, 99.63], ...,
 [2c12, mk, 99.53], [2c12, mapc, 99.53].

Список ФП_СПР формируется по списку СПР по следующему алгоритму:

Входные данные: имя набора данных, СПР и истории обучения.

Выходные данные: список ФП_СПР.

1. ФП_СПР = []

2. Для Имя_ФП из Списка ФП:

// Среднее значение $val_acc_{ФП}$ для текущей ФП, найденное по СПР

$val_acc_{СПРФП} = 0$

$N = 0$ // Число присутствий ФП в СПР

Для ПР из СПР: // ПР[1] — содержит имя ФП

Если ПР[1]=Имя_ФП://ПР[2]— содержит $val_acc_{ФП}$

$val_acc_{СПРФП} = val_acc_{СПРФП} + ПР[2]$

$N = N + 1$

Если $N > 0$:

$val_acc_{СПРФП} = val_acc_{СПРФП} / N$

Добавить в ФП_СПР список [Имя ФП, N , $val_acc_{СПРФП}$].

4. Упорядочить ФП_СПР по убыванию N , $val_acc_{СПРФП}$.

5. Оставить в ФП_СПР списки с наибольшим N .

Пример. ФП_СПР на MNIST:

После п. 4:

[[cce, 6, 99.58], [kld, 6, 99.56], [scce, 5, 99.59], ...,
 [ch, 2, 99.64]]

После п. 5:

[[cce, 6, 99.58], [kld, 6, 99.56]].

Таблица 10

Результаты серий обучения

| Набор данных | ФП | $val_acc_{ФП_C}$ | Диапазон изменения $val_acc_{ФП}$ |
|--------------|-------|--------------------|------------------------------------|
| MNIST | pss | 99.52 | 99.41...99.70 |
| | scce | 99.52 | 99.38...99.68 |
| | kld | 99.52 | 99.38...99.66 |
| | msle | 99.51 | 99.36...99.71 |
| | sh | 99.51 | 99.36...99.70 |
| | cce | 99.51 | 99.28...99.68 |
| | cp | 99.51 | 99.41...99.68 |
| | mk | 99.51 | 99.41...99.67 |
| | bce | 99.51 | 99.43...99.66 |
| | mse | 99.51 | 99.38...99.65 |
| | ck | 99.50 | 99.30...99.67 |
| | lc | 99.49 | 99.30...99.68 |
| | mape | 99.47 | 99.26...99.67 |
| | mae | 99.46 | 99.26...99.68 |
| | h | 99.45 | 99.30...99.68 |
| ch | 99.44 | 99.30...99.67 | |
| EMNIST | lc | 94.88 | 94.65...95.41 |
| | msle | 94.86 | 94.60...95.43 |
| | mse | 94.86 | 94.60...95.42 |
| | cp | 94.81 | 94.59...95.42 |
| | cce | 94.81 | 94.59...95.37 |
| | ck | 94.81 | 94.50...95.36 |
| | kld | 94.81 | 94.60...95.35 |
| | scce | 94.79 | 94.50...95.35 |
| | sh | 94.78 | 94.00...95.38 |
| | mk | 94.77 | 94.01...95.37 |
| | bce | 94.76 | 94.47...95.41 |
| | pss | 94.75 | 93.95...95.31 |
| | ch | 94.17 | 93.12...95.39 |
| | mae | 92.56 | 89.80...95.38 |
| | mape | 91.86 | 85.93...95.30 |
| h | 91.59 | 87.97...95.40 | |
| CIFAR10 | bce | 77.38 | 73.56...91.30 |
| | cce | 77.27 | 74.10...90.70 |
| | pss | 77.25 | 73.57...91.57 |
| | cp | 77.24 | 73.34...90.72 |
| | scce | 77.23 | 73.13...90.98 |
| | mk | 77.17 | 73.88...90.59 |
| | kld | 77.07 | 73.38...90.81 |
| | ck | 76.99 | 73.71...90.23 |
| | mse | 76.79 | 72.39...90.08 |
| | sh | 76.04 | 69.98...90.14 |
| | lc | 75.91 | 65.75...89.15 |
| | msle | 75.64 | 66.93...89.11 |
| | mae | 74.51 | 67.64...85.09 |
| | mape | 73.90 | 68.19...84.31 |
| | h | 72.10 | 57.22...84.32 |
| ch | 57.74 | 18.68...88.81 | |

Таким образом, после выполнения п. 5 в ФП_СПР остаются ФП, чаще приводящие к получению ФП-решения с $val_acc_{ФП} > val_acc_{ФП_С_max}$. При этом для заданной ФП разница между $val_acc_{СПРФП}$ и максимальным $val_acc_{ФП}$ будет меньше, чем аналогичное значение в случае списка ФП_СО.

Пример. ФП_СПР на CIFAR10: [[bce, 88.13], [pss, 87.97]].

ФП bce: $val_acc_{СПРФП} = 88,13$; $val_acc_{ФП_С_max} = 91,30$;
 $91,30 - 88,13 = 3,17$.

ФП pss: $val_acc_{СПРФП} = 87,97$; $val_acc_{ФП_С_max} = 91,37$;
 $91,37 - 87,97 = 3,40$.

Список ФП_ЦО содержит третью версию перечня ФП, эффективных в серии обучения. ФП_ЦО формируется по историям обучения, сохраненным при реализации цикла обучения, обеспечившего лучшее ФП-решение в серии.

Пример. Найдено лучшее ФП-решение: [r_n, pss, 91.37, CIFAR10]. $\Delta = 0,217$. По историям обучения r_n на CIFAR10 получен следующий список ФП_ЦО: [[pss, 91.37], [bce, 91.30]].

Назовем ФП_СО, ФП_СПР и ФП_ЦО списками эффективности.

Списки эффективности формируются для каждой серии обучения. ФП, включенные в списки эффективности, сформированные для серии обучения, реализованной на одном из наборов данных, наиболее часто эффективны в циклах обучения этой серии и поэтому

рассматриваются в работе как ФП, эффективные на наборе данных.

Анализ списков эффективности

Используем списки эффективности для поиска ответов на следующие вопросы:

1. Можно ли по имеющимся ФП_СО и/или ФП_СПР прогнозировать, какие ФП будут эффективны во вновь обучаемой СНС?

2. Совпадают ли списки ФП_ЦО, полученные для заданной СНС, на разных наборах данных?

3. Можно ли считать, что ФП, эффективные на одном наборе данных, будут эффективны и на другом?

Для этого в табл. 11 представлены следующие списки эффективных ФП:

- ФП_СО и ФП_СПР, построенные по сериям обучения с полными списками обучения (см. табл. 4);
- ФП_СО и ФП_СПР, созданные по сериям обучения со списками обучения без лучшей СНС (с34bn — в случаях MNIST и EMNIST; r_n — в случае CIFAR10);
- ФП_ЦО, образованные по циклам обучения лучших на каждом наборе данных СНС;
- ФП_ЦО для с34bn на CIFAR10.

В таблице 11 в столбце диапазонов критерия указаны диапазоны изменения $val_acc_{ФП_С}$ или $val_acc_{СПРФП}$ или $val_acc_{ФП}$ для приводимых ФП (в зависимости от вида списка), а в столбце примечаний — имя СНС, по циклу обучения которой сформирован ФП_ЦО, а так-

Таблица 11

Списки эффективности на разных наборах данных

| Вид списка | Эффективные ФП | Диапазон критерия | Примечание |
|------------|---|-------------------|------------|
| MNIST | | | |
| ФП_СО | bce, scce, pss, cp, kld, sh, mse, cce, msle, mk, ck | 99.48...99.50 | Без с34bn |
| | pss, scce, kld, msle, sh, cce, cp, mk, bce, mse, ck | 99.50...99.52 | Есть с34bn |
| ФП_СПР | kld | 99.54 | Без с34bn |
| | cce, kld | 99.56...99.58 | Есть с34bn |
| ФП_ЦО | msle, pss, sh | 99.70...99.71 | с34bn |
| EMNIST | | | |
| ФП_СО | lc, msle, mse | 94.81...94.82 | Без с34bn |
| | lc, msle, mse | 94.86...94.88 | Есть с34bn |
| ФП_СПР | msle, mse | 94.93...94.95 | Без с34bn |
| | lc, msle, mse | 95.10...95.13 | Есть с34bn |
| ФП_ЦО | msle, cp, mse, bce, lc, h, ch | 95.39...95.43 | с34bn |
| CIFAR10 | | | |
| ФП_СО | bce, cce, cp, scce, pss, mk | 75.82...75.99 | Без r_n |
| | bce, cce, cp, pss, scce, mk | 77.23...77.38 | Есть r_n |
| ФП_СПР | cce, scce, pss, kld, cp | 80.26...80.46 | Без r_n |
| | bce, pss | 87.97...88.13 | Есть r_n |
| ФП_ЦО | pss, bce | 91.30...91.37 | r_n |
| ФП_ЦО | bce, ch | 84.92...84.94 | с34bn |

же сведения по наличию в списке обучения лучшей на наборе данных СНС (с34bn или r_n).

Приведенные в табл. 11 данные можно интерпретировать следующим образом.

Добавление одной СНС в список обучения может изменить списки эффективности на наборе данных как по составу ФП, так и по порядку следования ФП в этом списке.

Списки ФП_CO и ФП_СПР, сформированные без присутствия лучшей СНС в списке обучения, нельзя использовать для прогнозирования списков эффективности вновь обучаемой СНС (с34bn — в случаях MNIST и EMNIST; r_n — в случае CIFAR10).

Списки эффективности для заданной СНС (с34bn), обученной на разных наборах данных, не совпадают.

Не следует переносить опыт применения ФП на одном наборе на другой. Так, в списки эффективности CIFAR10 не входят имена многих ФП, например mse и sh, имеющиеся в списках эффективности MNIST и EMNIST.

Схема формирования списка эффективных функций потерь

Приведенные результаты обработки историй обучения позволяют сделать следующие выводы:

- нельзя отдать предпочтение одной ФП при классификации изображений MNIST, EMNIST и CIFAR10;
- для СНС можно указать несколько эффективных ФП (см. табл. 5 — 9, 11);
- в разных наборах данных эффективны разные ФП (см. табл. 11);
- число ФП, эффективных в СНС, существенно меньше числа ФП, встроенных в библиотеку Keras;
- добавленные в список ФП функции sk и mk редко оказываются среди эффективных ФП (см. табл. 7) и могут быть исключены из списка ФП без ущерба качеству обучения;
- нельзя переносить опыт поиска ФП, эффективных в одной СНС, на другую: для каждой СНС поиск эффективных ФП следует проводить заново.

Поиск можно выполнить, придерживаясь следующей схемы.

1. Выбрать СНС.
2. Сформировать начальный список ФП, например, включив в него все ФП библиотеки Keras.
3. Выполнить обучение СНС с каждой ФП из текущего списка ФП, сохраняя истории обучения.
4. Провести несколько раз повторное обучение СНС с каждой ФП, сохраняя истории обучения (в [17] для определения погрешности измерения значения критерия СНС обучают трижды).
5. По результатам п.п. 3 и 4 определить Δ — погрешность измерения $val_acc_{ФП}$.
6. Используя полученное значение погрешности, сформировать по историям обучения список эффективных ФП.
7. Если точность классификации изображений оценочных данных, получаемая с применением ФП, неудовлетворительна, — сократить список ФП, удалив из него малоэффективные, и перейти к другой СНС.
8. Повторять п.п. 3 — 7 до достижения желаемого результата.
9. Выбрать ФП из списка эффективных ФП и провести обучения СНС с целью получения обученной модели СНС.

Замечание. Предусмотренные п. 7 сокращение списка ФП и замена одной СНС на другую разработчик выполняет, опираясь на собственный опыт.

Предусмотренный п. 9 выбор ФП из числа эффективных ФП можно сделать по такому показателю качества обучения, как «общее число ошибочно классифицированных изображений обучающих и оценочных данных»:

$$err_img = (1 - val_acc_{ФП})n_val + (1 - acc_{ФП})n_train,$$

где n_val, n_train — числа изображений в оценочных и обучающих данных; $acc_{ФП}$ — точность классификации изображений обучающих данных.

Пример выбора ФП приведен в табл. 12.

Таблица 12

Выбор ФП из числа эффективных

| Набор данных | НС | Эффективные ФП | $val_acc_{ФП}$ | $acc_{ФП}$ | err_img | Выбор |
|--------------|-------|----------------|-----------------|------------|------------|-------|
| MNIST | с34bn | msle | 99.71 | 99.875 | 104 | pss |
| | | pss | 99.70 | 99.910 | 84 | |
| | | sh | 99.70 | 99.885 | 99 | |
| EMNIST | с34bn | msle | 95.43 | 96.907 | 4811 | cp |
| | | cp | 95.42 | 96.974 | 4729 | |
| | | mse | 95.42 | 96.716 | 5051 | |
| | | bce | 95.41 | 96.513 | 5307 | |
| | | lc | 95.41 | 96.361 | 5496 | |
| CIFAR10 | r_n | pss | 91.37 | 97.998 | 1864 | bce |
| | | bce | 91.30 | 98.422 | 1659 | |

Таблица 13

Снижение $val_acc_{\text{ФП}}$ при замене ФП, обеспечивших обучение наибольшим значением критерия, на ссе

| Набор данных | НС | ФП | $val_acc_{\text{ФП_max}}$ | ФП | $val_acc_{\text{ФП}}$ | Снижение $val_acc_{\text{ФП}}$ |
|--------------|-------|------|-----------------------------|-----|------------------------|---------------------------------|
| MNIST | c34bn | msle | 99.71 | ссе | 99.68 | 0.03 |
| EMNIST | c34bn | msle | 95.43 | ссе | 95.37 | 0.06 |
| CIFAR10 | r_n | pss | 91.37 | ссе | 90.70 | 0.67 |

Отказ от поиска эффективных ФП в пользу применения перекрестной энтропии связан с некоторым снижением точности классификации оценочных данных (табл. 13).

Во всех приведенных в табл. 13 случаях снижение в точности классификации, вызванное применением ссе взамен ФП, обеспечившей обучение наибольшим значением $val_acc_{\text{ФП}}$, превышает погрешность измерения критерия.

Заключение

Для экспериментальной оценки эффективности ФП библиотеки Keras на MNIST, EMNIST и CIFAR10 для каждого набора составлены списки обучения из СНС, построенных на основе базовой структуры (см. рис. 1), и ResNet20v1. Применено строковое представление СНС и разработана программа на Python, создающая средствами библиотеки Keras СНС по ее строковому представлению и выполняющая эпизод обучения — обучение СНС с одной ФП. Результаты эпизода обуче-

ния сохраняются в виде истории. В эксперименте на каждом наборе данных реализована серия обучения, в результате чего накоплены истории обучения СНС из составленных списков (см. табл. 4).

Рассмотрены алгоритмы формирования по историям обучения списков ФП, эффективных как в отдельной СНС, так и на наборе данных. Критерий оценки эффективности — точность классификации изображений оценочных данных.

Обработка историй обучения показала, что в общем случае СНС можно одинаково эффективно обучать, применяя разные ФП. При этом не обнаружено возможности переноса результатов поиска ФП, эффективных в одной СНС, на другую: в каждой СНС поиск эффективных ФП следует выполнять заново.

Сформулирована схема поиска эффективных ФП; выбор ФП из числа эффективных может быть выполнен по общему числу ошибочно классифицированных изображений обучающих и оценочных данных.

Литература

1. Fukushima K. Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position — Neocognitron // Trans. IECE. 1979. V. J62-A(10). Pp. 658—665.
2. Fukushima K. Neocognitron: A Self-Organizing Neural Network for a Mechanism of Pattern Recognition Unaffected by Shift in Position // Biological Cybernetics. 1980. V. 36. No. 4. Pp. 193—202.
3. LeCun Y. e. a. Back-propagation Applied to Handwritten Zip Code Recognition // Neural Computation. 1989. V. 1. No. 4. Pp. 541—551.
4. LeCun Y. e. a. Gradient-based Learning Applied to Document Recognition // Proc. IEEE. 1998. V. 86. No. 11. Pp. 2278—2324.
5. LeCun Y. e. a. Handwritten Digit Recognition with a Back-Propagation Network // Advances in Neural Information Proc. Syst. 1990. V. 2. Pp. 396—404.
6. Keras: The Python Deep Learning library [Электрон. ресурс] <https://keras.io/> (дата обращения 01.05.2019).
7. TensorFlow. An Open Source Machine Learning Framework for Everyone [Электрон. ресурс] <https://www.tensorflow.org/> (дата обращения 01.05.2019).
8. Keras documentation. Convolutional Layers [Электрон. ресурс] <https://keras.io/layers/convolutional/> (дата обращения 01.05.2019).

References

1. Fukushima K. Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position — Neocognitron. Trans. IECE. 1979;J62-A(10): 658—665.
2. Fukushima K. Neocognitron: A Self-Organizing Neural Network for a Mechanism of Pattern Recognition Unaffected by Shift in Position. Biological Cybernetics. 1980; 36;4:193—202.
3. LeCun Y. e. a. Back-propagation Applied to Handwritten Zip Code Recognition. Neural Computation. 1989;1;4:541—551.
4. LeCun Y. e. a. Gradient-based Learning Applied to Document Recognition. Proc. IEEE. 1998;86;11:2278—2324.
5. LeCun Y. e. a. Handwritten Digit Recognition with a Back-Propagation Network. Advances in Neural Information Proc. Syst. 1990;2:396—404.
6. Keras: The Python Deep Learning library [Elektron. Resurs] <https://keras.io/> (Data Obrashcheniya 01.05.2019).
7. TensorFlow. An Open Source Machine Learning Framework for Everyone [Elektron. Resurs] <https://www.tensorflow.org/> (Data Obrashcheniya 01.05.2019).
8. Keras documentation. Convolutional Layers [Elektron. Resurs] <https://keras.io/layers/convolutional/> (Data Obrashcheniya 01.05.2019).

9. **Keras documentation.** Core Layers [Электрон. ресурс] <https://keras.io/layers/core/> (дата обращения 01.05.2019).
10. **LeCun Y., Cortes C., Burges C.** The MNIST Database of Handwritten Digits [Электрон. ресурс] <http://yann.lecun.com/exdb/mnist/> (дата обращения 01.05.2019).
11. **Cohen G. e. a.** EMNIST: an Extension of MNIST to Handwritten Letters [Электрон. ресурс] <https://arxiv.org/pdf/1702.05373.pdf> (дата обращения 01.05.2019).
12. **The CIFAR10 Dataset.** [Электрон. ресурс] <https://www.cs.toronto.edu/~kriz/cifar.html> (дата обращения 01.05.2019).
13. **Николенко С., Кадури А., Архангельская Е.** Глубокое обучение. СПб.: Питер, 2018.
14. **Джулли А., Пал С.** Библиотека Keras — инструмент глубокого обучения. М.: ДМК Пресс, 2018.
15. **Trains a ResNet on the CIFAR10 Dataset** [Электрон. ресурс] https://keras.io/examples/cifar10_resnet/ (дата обращения 01.05.2019).
16. **Janocha K., Czarnecki W.M.** On Loss Functions for Deep Neural Networks in Classification. Submitted on 18 Feb 2017. [Электрон. ресурс] <https://arxiv.org/abs/1702.05659> (дата обращения 01.05.2019).
17. **Бартеньев О.В.** Классификация изображений нейронной сетью [Электрон. ресурс] <http://100byte.ru/python/imgClasses/imgClasses.html> (дата обращения 01.05.2019).
18. **He K. e. a.** Deep Residual Learning for Image Recognition // Proc. 2016 CVPR, 2016. Pp. 770-778.
19. **Функции** потерь библиотеки Keras [Электрон. ресурс] <http://100byte.ru/python/loss/loss.html> (дата обращения 01.05.2019).
20. **Keras Documentation.** Built-in Loss Functions. [Электрон. ресурс] <https://github.com/keras-team/keras/blob/master/keras/losses.py> (дата обращения 01.05.2019).
21. **Keras Advanced Activations** [Электрон. ресурс] <https://keras.io/layers/advanced-activations/> (дата обращения 01.05.2019).
22. **Kingma D., Ba J.** Adam: A Method for Stochastic Optimization // arXiv, 2014. [Электрон. ресурс] <http://arxiv.org/abs/1412.6980> (дата обращения 01.05.2019).
23. **Keras documentation.** Merge Layers [Электрон. ресурс] <https://keras.io/layers/merge/> (дата обращения 01.05.2019).
24. **Keras Documentation.** ImageDataGenerator [Электрон. ресурс] <https://keras.io/preprocessing/image/> (дата обращения 01.05.2019).
25. **Keras documentation.** How can I Obtain Reproducible Results Using Keras During Development? [Электрон. ресурс] <https://keras.io/getting-started/faq/#how-can-i-obtain-reproducible-results-using-keras-during-development> (дата обращения 01.05.2019).
9. **Keras documentation.** Core Layers [Elektron. Resurs] <https://keras.io/layers/core/> (Data Obrashcheniya 01.05.2019).
10. **LeCun Y., Cortes C., Burges C.** The MNIST Database of Handwritten Digits [Elektron. Resurs] <http://yann.lecun.com/exdb/mnist/> (Data Obrashcheniya 01.05.2019).
11. **Cohen G. e. a.** EMNIST: an Extension of MNIST to Handwritten Letters [Elektron. Resurs] <https://arxiv.org/pdf/1702.05373.pdf> (Data Obrashcheniya 01.05.2019).
12. **The CIFAR10 Dataset.** [Elektron. Resurs] <https://www.cs.toronto.edu/~kriz/cifar.html> (Data Obrashcheniya 01.05.2019).
13. **Nikolenko S., Kadurin A., Arkhangel'skaya E.** Glubokoe Obuchenie. SPb.: Piter, 2018. (in Russian).
14. **Dzhulli A., Pal S.** Biblioteka Keras — Instrument Glubokogo Obucheniya. M.: DMK Press, 2018. (in Russian).
15. **Trains a ResNet on the CIFAR10 Dataset** [Elektron. Resurs] https://keras.io/examples/cifar10_resnet/ (Data Obrashcheniya 01.05.2019).
16. **Janocha K., Czarnecki W.M.** On Loss Functions for Deep Neural Networks in Classification. Submitted on 18 Feb 2017. [Elektron. Resurs] <https://arxiv.org/abs/1702.05659> (Data Obrashcheniya 01.05.2019).
17. **Barten'ev O.V.** Klassifikatsiya Izobrazheniy Neyronnoy Set'yu [Elektron. Resurs] <http://100byte.ru/python/imgClasses/imgClasses.html> (Data Obrashcheniya 01.05.2019). (in Russian).
18. **He K. e. a.** Deep Residual Learning for Image Recognition. Proc. 2016 CVPR, 2016:770-778.
19. **Funktsii Poter' Biblioteki Keras** [Elektron. Resurs] <http://100byte.ru/python/loss/loss.html> (Data Obrashcheniya 01.05.2019). (in Russian).
20. **Keras Documentation.** Built-in Loss Functions. [Elektron. Resurs] <https://github.com/keras-team/keras/blob/master/keras/losses.py> (Data Obrashcheniya 01.05.2019).
21. **Keras Advanced Activations** [Elektron. Resurs] <https://keras.io/layers/advanced-activations/> (Data Obrashcheniya 01.05.2019).
22. **Kingma D., Ba J.** Adam: A Method for Stochastic Optimization. arXiv, 2014. [Elektron. Resurs] <http://arxiv.org/abs/1412.6980> (Data Obrashcheniya 01.05.2019).
23. **Keras documentation.** Merge Layers [Elektron. Resurs] <https://keras.io/layers/merge/> (Data Obrashcheniya 01.05.2019).
24. **Keras Documentation.** ImageDataGenerator [Elektron. Resurs] <https://keras.io/preprocessing/image/> (Data Obrashcheniya 01.05.2019).
25. **Keras documentation.** How can I Obtain Reproducible Results Using Keras During Development? [Elektron. Resurs] <https://keras.io/getting-started/faq/#howcan-i-obtain-reproducible-results-using-keras-during-development> (Data Obrashcheniya 01.05.2019).

Сведения об авторах:

Апарнев Алексей Николаевич — аспирант кафедры прикладной математики и искусственного интеллекта НИУ «МЭИ», e-mail: apich238@gmail.com

Бареньев Олег Васильевич — кандидат технических наук, доцент кафедры прикладной математики и искусственного интеллекта НИУ «МЭИ», e-mail: mdf4@mail.ru

Information about authors:

Aparnev Aleksey N. — Ph.D.-student of Applied Mathematics and Artificial Intelligence Dept., NRU MPEI, e-mail: apich238@gmail.com

Barten'ev Oleg V. — Ph.D. (Techn.), Assistant Professor of Applied Mathematics and Artificial Intelligence Dept., NRU MPEI, e-mail: mdf4@mail.ru

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов

Conflict of interests: the authors declare no conflict of interest

Статья поступила в редакцию: 11.07.2018

The article received to the editor: 11.07.2018