

# Машинное обучение

## Аугментация обучающей выборки «на лету» для обучения нейронных сетей\*

А.В. ГАЙЕР<sup>I,II</sup>, А.В. ШЕШКУС<sup>II</sup>, Ю.С. ЧЕРНЫШОВА<sup>II,III</sup>

<sup>I</sup> Национальный исследовательский технологический университет «МИСиС», г. Москва, Россия

<sup>II</sup> ООО «Смарт Энджинс Сервис», Москва, Россия

<sup>III</sup> Институт системного анализа Федерального исследовательского центра «Информатика и управление» Российской академии наук, Москва, Россия

**Аннотация.** В работе исследуется аугментация «на лету» – метод увеличения репрезентативности обучающей выборки прямо во время обучения искусственных нейронных сетей. Рассматриваются наиболее частые преобразования с сохранением меток, которые будут полезны в большинстве практических задач. Ввиду ограничений существующих систем для аугментации «на лету», среди которых добавление новых преобразований и увеличение времени обучения, была разработана собственная система аугментации. Проведены эксперименты на датасете MNIST с разработанной системой аугментации «на лету», которые продемонстрировали эффективность ее работы – на архитектуре SimpleNet(310K) удалось превзойти актуальный лучший результат, уменьшив ошибку распознавания с 0.28 до 0.25%.

**Ключевые слова:** машинное обучение, искусственные нейронные сети, «онлайн» аугментация, аугментация «на лету».

**DOI:** 10.14357/20790279180517

### Введение

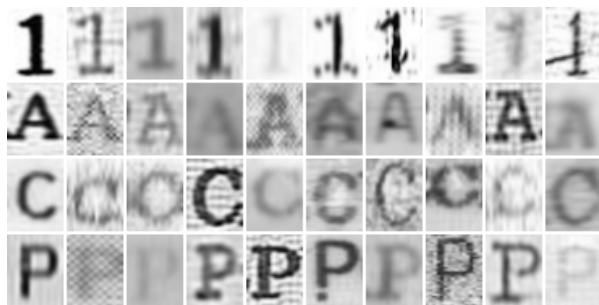
Использование искусственных нейронных сетей стало повсеместным явлением в задачах распознавания образов [1, 2]. Обучающие выборки для таких сетей содержат тысячи [3, 4] и даже миллионы [5] изображений. В зависимости от задачи, минимальное количество необходимых для обучения данных может сильно различаться, но в общем случае справедливо утверждение: чем их больше, тем лучше. Однако на практике зачастую довольно сложно собрать даже минимальное число данных, особенно если задача крайне специфичная [6]. Вторая серьезная проблема в создании натуральных датасетов – это необходимость разметки, присвоения меток объектам на изображениях. Этот процесс самый трудоемкий и долгий, так как он исключает полную автоматизацию процесса – не-

обходима проверка созданной разметки человеком на корректность.

Для обучения нейросетей важно, чтобы обучающая выборка была достаточно сбалансирована и репрезентативна [7]. На рис. 1 приведены примеры изображений из задачи распознавания символов – в первом столбце так называемые «идеальные» изображения, в остальных – деформированные, но встречающиеся на практике. Обученный только на близких к идеалу изображениях классификатор, не видевший все возможные деформации, с большой долей вероятности будет ошибаться на сильно искаженных изображениях.

Использование же недостаточного количества данных может привести к переобучению модели. Для решения этой проблемы на уровне модели используют регуляризацию или пытаются уменьшить количество параметров. На уровне работы с данными чаще всего используется аугментация.

\* Работа выполнена при частичной поддержке РФФИ, гранты 17-29-03236 и 17-29-07093.



**Рис. 1.** Примеры изображений из задачи распознавания символов

В работе [8] дается следующее определение аугментации: «искусственное увеличение обучающей выборки с применением преобразований, сохраняющих метки». Однако отметим, что некоторые методы аугментации подразумевают изменение метки, как пример – “mixup” аугментация [9], заключающаяся в смешивании информации двух сэмплов и ответов к ним. Преобразования, как правило, выбираются исходя из специфики поставленной задачи.

При использовании аугментации возникает проблема выбора – во сколько раз увеличивать выборку и какими именно преобразованиями, чтобы полученные данные были как можно более репрезентативными и полно выражающими генеральную совокупность данных.

Наряду с обычной аугментацией обучающей выборки встречается метод аугментации «на лету», как например в работе [10]. На каждой итерации обучения к исходной выборке применяется различный набор преобразований со случайными параметрами.

Структура данной работы следующая. В разделе 1 дается конкретизация исследуемой области и обзор ключевых работ в ней. В разделе 2 ставится математическая и содержательная постановка задачи аугментации на лету. В разделе 3 приводится наше решение задачи с указанием конкретных используемых преобразований, а также краткое описание используемого датасета. В разделе 4 описывается ход экспериментов по обучению нейросети на изначальных и аугментированных данных, а также полученные результаты.

## 1. Обзор существующих работ

Все преобразования, используемые в аугментации, можно выделить в две категории. Первые опираются на информацию о процессе обучения модели и ее свойства [9, 11]. Вторые используют априорные знания о данных для конкретной зада-

чи. В данной работе мы будем рассматривать только вторую группу.

Типы аугментации соответствуют следующим группам: яркостные, геометрические, зашумления, аддитивные (наложение двух изображений). Постановка задачи аугментации и подробное описание типов преобразований изложены в работе [12] Выбор преобразований обычно происходит на основе возможных деформаций на изображениях. Так, в задаче распознавания символов, снятых на цифровую камеру [13], наиболее актуальны сглаживание, проективные искажения, яркостные изменения, добавление гауссового шума, повороты. При распознавании симметричных объектов можно использовать отражения по какой-либо оси.

Аугментация очень хорошо себя показывает в случае, если обучение происходит на искусственно сгенерированных данных. Данное исследование было отражено в работе [14]. Деформации “идеальных” созданных изображений делают их более реалистичными и увеличивают общую вариативность данных.

Помимо выше упомянутых “классических” видов преобразований в последнее время предпринимаются попытки использовать для аугментации генеративно-сопоставительные сети (generative adversarial networks, GANs) [15]. Так, в работе [16] исследуется влияние различных техник аугментации на качество обученных сетей, сравниваются традиционные преобразования с результатами работы аугментирующих генеративно-сопоставительных сетей. По утверждению авторов, на текущий момент GAN’ы и аугментация нейронными сетями уступают классическим преобразованиям.

Ввиду востребованности аугментации в практических задачах, многие современные фреймворки для обучения нейронных сетей имеют слой для аугментации входных данных. Например, совсем недавно NVIDIA анонсировала собственную библиотеку DALI для аугментации изображений на GPU [17], поддерживаемая TensorFlow и PyTorch. Также ведутся разработки отдельных независимых систем, подходящих для встраивания в некоторые фреймворки, например Augmentor, подходящий для Keras и PyTorch [18]. Упомянутые инструменты можно использовать для аугментации “на лету”. Однако набор преобразований в них ограничен, и добавление новых типов преобразований не всегда возможно. Также есть ограничения по используемой платформе – например Augmentor можно использовать, если фреймворк для обучения сетей использует python. Последний немаловажный технический фактор – в случае использования в архитектуре

сети отдельных слоев для аугментации время выполнения одной итерации увеличивается.

На текущий момент в работах, связанных с аугментацией, не наблюдается четкого разделения на аугментацию обучающей выборки до начала обучения и онлайн-аугментацию. В нашей работе мы исследуем аугментацию “на лету” так как считаем, что она имеет неоспоримые преимущества в сравнении с предварительным раздутием обучающей выборки.

### 2. Постановка задачи

Пусть имеется обучающая выборка  $X^{train}$  из  $N$  изображений одного размера и множество меток  $Y^{train}$  к ним:

$$X^{train} = \{x_1, x_2, \dots, x_N\}.$$

Рассмотрим процесс обучения некоторой модели  $M$  в дискретные моменты времени  $T = \{t_1, t_2, \dots, t_K\}$ . Пусть  $\Phi$  – вектор параметров, а  $L$  есть функция потерь. Качество обученной модели будем оценивать некоторым способом на тестовой выборке изображений  $X^{test}$  и ответов к ним  $Y^{test}$ . Обозначим наилучшее полученное качество обученной сети на тесте как:

$$Q(M, X^{train}, Y^{train}) = \min_{r_{t_i}} (M_{t_i}(X^{train}, Y^{train}, X^{test}, Y^{test}, \Phi_{t_i}, L) = r_{t_i})$$

$$r_{t_i} \in \mathbb{R}; i = \{1, 2, \dots, K\}.$$

Необходимо задать такое множество преобразований  $F = \{f_1, f_2, \dots, f_D\}$ , где

$$f_i = f_i(x_j) = \tilde{x}_j; Y(x_j) = Y(\tilde{x}_j),$$

выбирая из которого случайное подмножество  $F' \subseteq F$  и применяя преобразования из него к  $X^{train}$  (обозначим за  $F'_{t_i}[X^{train}]$ ):

$$Q(M, F'_{t_i}[X^{train}], Y^{train}),$$

$$< Q(M, X^{train}, Y^{train}),$$

$$i = \{1, 2, \dots, K\}.$$

Причем вероятность для отдельно взятой функции  $f_j$  быть выбранной в подмножество  $F'_{t_i}$ , содержащего только уникальные функции, равна  $p(f_j \in F'_{t_i}) = \frac{1}{D}$ .

Содержательно же необходимо разработать систему аугментации “на лету”, позволяющую раздувать обучающую выборку указанными преобразованиями (примененными случайным образом) прямо во время обучения нейронной сети. Система

должна эффективно обрабатывать данные, не увеличивая время обучения модели.

### 3. Предлагаемое решение (алгоритм)

Разработанная система принимает набор изображений и преобразует его согласно заданным параметрам, среди которых список возможных преобразований. Применение трансформаций к каждой картинке происходит в случайном количестве и порядке. В качестве преобразований были рассмотрены наиболее часто встречающиеся искажения в практических задачах:

1. Проективные преобразования (рис. 2). Поддерживаются 2 режима. В первом – каждая вершина прямоугольника изображения смещается внутрь. Во втором – две случайно выбранные вершины смещаются, преобразуя исходный прямоугольник в трапецию, параллелограмм или произвольный выпуклый четырехугольник;
2. Обрезание части изображения и растягивание до исходного размера;
3. Отражения по горизонтали или вертикали;
4. Гауссово размытие;
5. Нанесение линии;
6. Циклический сдвиг;
7. Поворот на некоторый угол;
8. Добавление гауссового шума;
9. Линейное изменение яркости. Может осуществляться и к части изображения;
10. Изменение яркости случайной монотонной функцией (рис. 3). Было предложено в работе [12];
11. Масштабирование вниз с последующим вверх. Вносит интерполяционный эффект на изображение. Актуально для изображений в низком разрешении, которые были увеличены до размера, требуемого на входе обрабатывающей системы;
12. Эрозия/Дилатация. В задаче распознавания символов, если известно что фон на всех изображениях светлый/темный, позволяет добиться “жирности” символа.

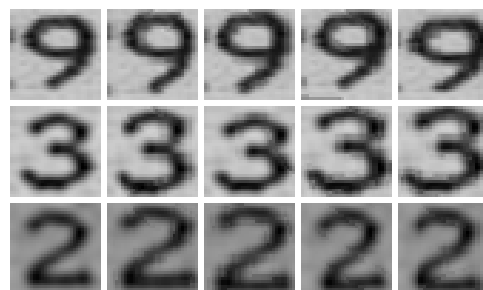
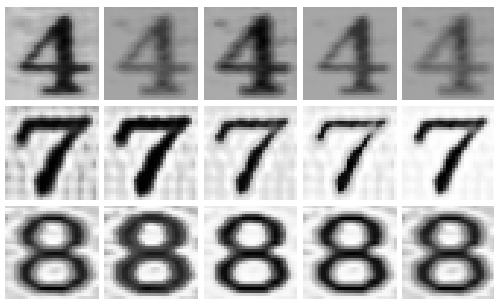


Рис.2. Примеры проективных преобразований (в первом столбце оригинальные изображения)



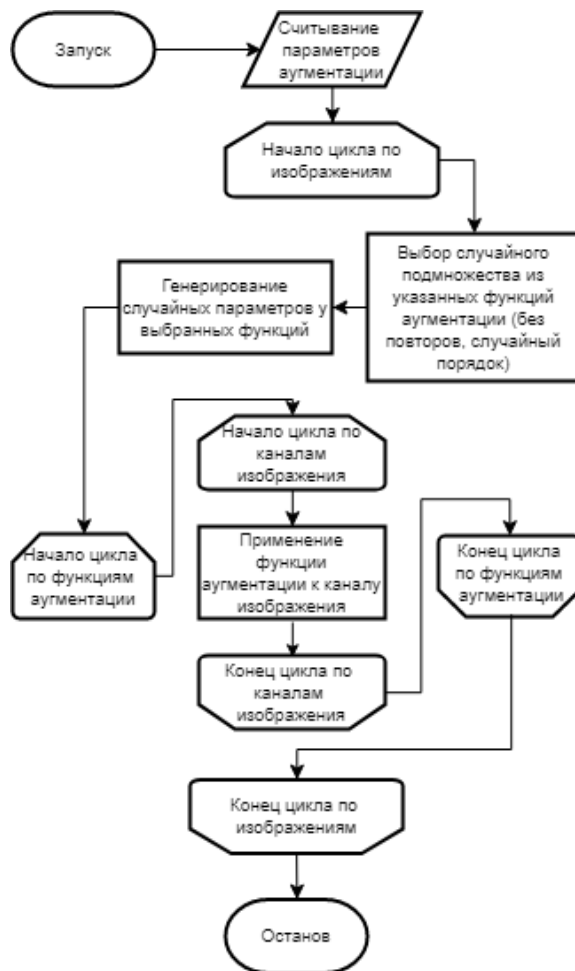
**Рис.3.** Примеры яркостной аугментации случайной монотонной функцией (в первом столбце оригинальные изображения)

Алгоритм аугментации следующий. Получив на вход изображение, из списка указанных преобразований выбирается случайное число преобразований (без повторов). Список перемешивается и применяется последовательно к изображению со случайными параметрами (рис. 4). Обработка каналов происходит независимо с одинаковыми параметрами. Все случайные величины имеют равномерное распределение. Блок-схема алгоритма приведена на рис. 5.

Особое внимание было уделено эффективности данной модели. Необходимо было добиться того, чтобы ее использование не замедляло процесс обучения, в отличие от существующих решений. Это было достигнуто за счет некоторого распараллеливания задачи: пока текущий батч с данными используется для непосредственного обучения на GPU устройстве, следующий батч в этот момент аугментируется на CPU. На практике предложенный подход не приводил к увеличению времени обучения – во всех опробованных практических задачах время аугментации батча на CPU было меньше, чем обучение на нем на GPU. Таким образом, время обучения остается практически на том же уровне, как если бы аугментация не использовалась.



**Рис.4.** Примеры изображений с применением совокупности преобразований (в первом столбце оригинальные изображения)



**Рис.5.** Блок-схема алгоритма аугментации

**Описание датасета.** В качестве данных был использован датасет MNIST [3], содержащий 70000 монохромных изображений рукописных цифр (10 классов) в разрешении 28x28, из которых 10000 предполагаются для использования в качестве теста.

#### 4. Результаты

Для проведения экспериментов была взята архитектура сети из работы [19], в которой авторы показали, что высоких результатов в распознавании на известных датасетах можно достигнуть с довольно простой архитектурой и много меньшим числом параметров. SimpleNet содержит 13 сверточных слоев с размерами фильтров 3x3 и батч нормализациями после каждого. Из статьи была взята облегченная версия архитектуры, именуемая авторами как SimpleNet(310K) и содержащая 310 тыс. параметров. Мы обучили на данной архитектуре сеть на MNIST, и достигли указанной авторами

ми ошибки – 0,28%. После этого была подключена разработанная система аугментации на лету. В качестве преобразований были выбраны:

1. Проективные преобразования с фиксацией двух вершин, доля смещения – от 10 до 17% пикселей от ширины/высоты изображения;
2. Поворот на угол от 8 до 15 градусов. Знак поворота также выбирался случайно;
3. Добавление полностью черной линии поверх изображения. Использовалось для создания разрыва в непрерывно нарисованных символах;
4. Изменение яркости случайной монотонной функцией;
5. Обрезание и растягивание до изначального размера, до 11% пикселей от ширины/высоты изображения.



**Рис. 6.** Примеры полученных изображений из MNIST в результате аугментации (в первом столбце оригинальные изображения)

На каждой эпохе происходило 100% раздутие обучающей выборки, примеры получаемых изображений приведены на рис. 6. В результате нами было достигнуто 0,25% ошибок распознавания. Важно отметить, что подобный результат авторы достигали только на оригинальной архитектуре с 5,4 млн параметров.

В следующем эксперименте мы попробовали уменьшить число фильтров на каждом слое в 2 раза, уменьшив тем самым число параметров примерно в 4 раза. Без аугментации мы достигли того

же самого значения ошибки – 0,28%. С включенной аугментацией и преобразованиями, описанными выше, удалось добиться ошибки равной 0,26% (табл. 1).

### Заключение

Результатом данной работы стала модель аугментации данных на лету, т.е. преобразующая обучающие данные прямо во время обучения нейросети. Такой метод раздутия обладает следующими достоинствами. Во-первых, увеличивается вариативность данных – за время обучения модель увидит большее количество искаженных изображений, чем если бы сгенерировали фиксированное число данных заранее. Во-вторых, не происходит увеличения физического размера обучающей выборки – время обучения сети будет существенно ниже.

Наша система, в отличие от аналогичных инструментов в сторонних фреймворках, сохраняет время обучения с аугментацией на том же уровне, как если бы учили без нее. Это было достигнуто за счет некоторого распараллеливания процесса: пока текущий обучающий батч обрабатывается на GPU, следующий батч в этот момент аугментируется на CPU. Эффективность аугментации была подтверждена на датасете MNIST с архитектурой сети SimpleNet(310K). В результате была улучшена точность распознавания на тесте – с 0,28% ошибок распознавания удалось уменьшить до 0,25%. Близкий результат был получен и на архитектуре с меньшим в 4 раза количеством параметров – число ошибок уменьшилось с 0,28 до 0,26%.

### Литература

1. *Полевой Д.В.* Актуальные задачи создания систем массового ввода с использованием оптического распознавания для преобразования сложно структурированных бумажных документов в гибридных информационных системах // Системный анализ и информационные технологии. Тр. четвертой междунар. конф.

**Табл. 1**

Результаты обучения с аугментацией «на лету» и без

Датасет и архитектура	Значение ошибки без аугментации на лету	Значение ошибки с аугментацией на лету
MNIST, SimpleNet(310K)	0,28%	0,25%
MNIST, SimpleNet(310K), облегченная по числу параметров в 4 раза	0,28%	0,26%

- (Абзаково, Россия, 17-23 августа 2011 г.): в 2 т. Т.2. Челябинск: Изд-во Челяб. гос. ун-та, 2011. С. 192-195.
2. *Bulatov K.* Smart IDReader: Document Recognition in Video Stream / K. Bulatov, V. V. Arlazarov, T. Chernov, O. Slavin and D. Nikolaev // 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, 2017, pp. 39-44. doi: 10.1109/ICDAR.2017.347.
  3. *The mnist database of handwritten digits* [Электронный ресурс]. URL: <http://yann.lecun.com/exdb/mnist> (дата обращения 15.06.2018).
  4. *The CIFAR-10 dataset* [Электронный ресурс]. URL: <https://www.cs.toronto.edu/~kriz/cifar.html> (дата обращения 15.06.2018).
  5. *Russakovsky O.* ImageNet large scale visual recognition challenge / O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, L. Fei-Fei // International Journal of Computer Vision. – 2015. – Vol. 115, Issue 3. – P. 211-252.
  6. *Ronneberger O.* U-Net: Convolutional Networks for Biomedical Image Segmentation / Ronneberger O., Fischer P., Brox T. // In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham.
  7. *Weiss G.* The Effect of Class Distribution on Classifier Learning: An Empirical Study / Weiss G., Provost F. // Technical Report ML-TR-44, Department of Computer Science, Rutgers University. August 2, 2001.
  8. *Krizhevsky A.* Learning multiple layers of features from tiny images. Master’s thesis. Department of Computer Science. University of Toronto. 2009. 60 p.
  9. *Zhang H.* Mixup: Beyond Empirical Risk Minimization / Zhang H., Ciss M., Dauphin Y., Lopez D. // ArXiv e-prints [Электронный ресурс] – ArXiv:1710.09412 – 2017 – URL: <https://arxiv.org/abs/1710.09412> (дата обращения 15.06.2018).
  10. *Zagoruyko S.* Learning to Compare Image Patches via Convolutional Neural Networks / Zagoruyko S., Komodakis N. // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015.
  11. *Fawzi A.* Adaptive data augmentation for image classification / Fawzi A., Samulowitz H., Turaga D., Frossard P. // Image Processing (ICIP), 2016 IEEE International Conference on. 2016. С. 3688-3692.
  12. *Емельянов С.О.* Методы аугментации обучающих выборок в задачах классификации изображений / Емельянов С.О., Иванова А.А., Швец Е.А., Николаев Д.П. // Сенсорные системы. 2018. Т. 32. № 3 (в печати).
  13. *Арлазаров В.В.* Анализ особенностей использования стационарных и мобильных малоразмерных цифровых видео камер для распознавания документов / Арлазаров В.В., Жуковский А.Е., Кривцов В.Е., Николаев Д.П., Полевой Д.В. // Информационные технологии и вычислительные системы, №3. 2014. С. 71-81.
  14. *Chernyshova Y.* Generation method of synthetic training data for mobile OCR system / Chernyshova Y. Gayer A. Sheshkus A. // Proc. SPIE 10696, Tenth International Conference on Machine Vision (ICMV 2017), 106962G (13 April 2018); doi: 10.1117/12.2310119
  15. *Goodfellow I.* Generative adversarial nets / I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio // Proceedings of the 27th International Conference on Neural Information Processing Systems. – 2014. – Vol. 2. – P. 2672-2680.
  16. *Perez L.* The Effectiveness of Data Augmentation in Image Classification using Deep Learning / Perez L., Wang J. // ArXiv e-prints [Электронный ресурс] – ArXiv:1712.04621 – 2017 – URL: <https://arxiv.org/abs/1712.04621> (дата обращения 01.06.2018).
  17. *Announcing NVIDIA DALI and NVIDIA nvJPEG* [Электронный ресурс]. URL: <https://news.developer.nvidia.com/announcing-nvidia-dali-and-nvidia-nvjpeg/> (дата обращения 19.06.2018).
  18. *Bloice D.* Augmentor: An Image Augmentation Library for Machine Learning / Bloice D., Stocker C., Holzinger A. // ArXiv e-prints [Электронный ресурс] – ArXiv:1712.04621 – 2017 – URL: <https://arxiv.org/abs/1712.04621> (дата обращения 15.06.2018).
  19. *Hasanpour S.H.* Let’s keep it simple, Using simple architectures to outperform deeper and more complex architectures / Hasanpour S. H., Rouhani M., Mohsen F., Sabokrou M. // ArXiv e-prints [Электронный ресурс] – ArXiv:1608.06037 – 2016 – URL: <https://arxiv.org/abs/1608.06037> (дата обращения 25.05.2018).

**Гайер Александр Вячеславович.** ООО «Смарт Энджинс Сервис», г. Москва, Россия. Научный сотрудник-программист. Количество печатных работ: 2. Область научных интересов: глубокое обучение, компьютерное зрение. E-mail: gayer.alexandr@yandex.ru

**Шешкус Александр Владимирович.** ООО «Смарт Энджинс Сервис», г. Москва, Россия. Старший научный сотрудник-программист. Количество печатных работ: 18. Область научных интересов: глубокое обучение, компьютерное зрение, проективно-инвариантная сегментация изображений. E-mail: astdcall@gmail.com

**Чернышова Юлия Сергеевна.** ООО «Смарт Энджинс Сервис», г. Москва, Россия. Научный сотрудник-программист. Институт системного анализа Федерального исследовательского центра «Информатика и управление» Российской академии наук, г. Москва, Россия. Аспирантка. Количество печатных работ: 3. Область научных интересов: синтез искусственных данных для обучения, аугментация данных. E-mail: chernyshovayulia07@gmail.com

### Augmentation on the fly for the neural networks learning

A.V. Gayer<sup>I,II</sup>, A.V. Sheshkus<sup>II</sup>, Y.S. Chernyshova<sup>II,III</sup>

<sup>I</sup> National University of Science and Technology "MISIS", Moscow, Russia

<sup>II</sup> LLC "Smart Engines Service", Moscow, Russia

<sup>III</sup> Institute for Systems Analysis, Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia

**Abstract.** In this work, we research online augmentation – method of increasing the representativeness of the training set during the learning of artificial neural networks. We consider the most common transformations with preservation of labels, which will be useful in many practical tasks. Due the limitations of existing systems for online augmentation, among which the adding new transformation functions and increased learning time, we introduce a new effective augmentation system without any impact on learning time. Experiments on the MNIST dataset with the developed system have shown that we outperform the current best result at SimpleNet(310K) architecture of neural network, reducing the error rate from 0.28% to 0.25%..

**Keywords:** machine learning, artificial neural networks, online augmentation, augmentation on the fly, real-time augmentation.

**DOI:** 10.14357/20790279180517

### References

1. Polevoj D.V. Aktual'nye zadachi sozdaniya sistem massovogo vvoda s ispol'zovaniem opticheskogo raspoznavaniya dlya preobrazovaniya slozhno strukturirovannyh bumazhnyh dokumentov v gibridnyh informacionnyh sistemah // Conference on Systems analysis and information technologies SAIT-2011, V. 2, P. 192-195 (in Russian).
2. K. Bulatov. SmartIDReader: Document Recognition in Video Stream / K. Bulatov, V. V. Arlazarov, T. Chernov, O. Slavin and D. Nikolaev // 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, 2017, pp. 39-44. doi: 10.1109/ICDAR.2017.347.
3. The mnist database of handwritten digits [Electronical Resource]. URL: <http://yann.lecun.com/exdb/mnist> (date request 15.06.2018).
4. The CIFAR-10 dataset [Electronical Resource]. URL: <https://www.cs.toronto.edu/~kriz/cifar.html> (date request 15.06.2018).
5. Russakovsky O. ImageNet large scale visual recognition challenge / O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, L. Fei-Fei // International Journal of Computer Vision. – 2015. – Vol. 115, Issue 3. – P. 211-252.
6. Ronneberger O. U-Net: Convolutional Networks for Biomedical Image Segmentation / Ronneberger O., Fischer P., Brox T. // In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham.
7. Weiss G. The Effect of Class Distribution on Classifier Learning : An Empirical Study / Weiss G., Provost F. // Technical Report ML-TR-44, Department of Computer Science, Rutgers University. August 2, 2001.

8. *Krizhevsky A.* Learning multiple layers of features from tiny images. Master’s thesis. Department of Computer Science. University of Toronto. 2009. 60 p.
9. *Zhang H.* Mixup: Beyond Empirical Risk Minimization / Zhang H., Ciss M., Dauphin Y., Lopez D. // ArXiv e-prints [Electronical Resource] – ArXiv:1710.09412 – 2017 – URL: <https://arxiv.org/abs/1710.09412> (date request 15.06.2018).
10. *Zagoruyko S.* Learning to Compare Image Patches via Convolutional Neural Networks / Zagoruyko S., Komodakis N. // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015.
11. *Fawzi A.* Adaptive data augmentation for image classification / Fawzi A., Samulowitz H., Turaga D., Frossard P. // Image Processing (ICIP), 2016 IEEE International Conference on. 2016. C. 3688-3692.
12. *Emelyanov S.O.* Methods of training augmentation in the task of image classification / Emelyanov S.O., Ivanova A.A., Shvets E.A., Nikolaev D.P. // Sensornye sistemy [Sensory systems]. 2018. V. 32(3) (in Russian).
13. *Arlazarov V.V. et al.* Analysis of features of the use of fixed and mobile small-sized digital video camera for OCR // Information Technologies and Computer Systems, 3/2014. P. 71-81 (in Russian).
14. *Chernyshova Y.* Generation method of synthetic training data for mobile OCR system / Chernyshova Y. Gayer A. Sheshkus A. // Proc. SPIE 10696, Tenth International Conference on Machine Vision (ICMV 2017), 106962G (13 April 2018); doi: 10.1117/12.2310119.
15. *Goodfellow I.* Generative adversarial nets / I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio // Proceedings of the 27th International Conference on Neural Information Processing Systems. – 2014. – Vol. 2. – P. 2672-2680.
16. *Perez L.* The Effectiveness of Data Augmentation in Image Classification using Deep Learning / Perez L., Wang J. // ArXiv e-prints [Electronical Resource] – ArXiv:1712.04621 – 2017 – URL: <https://arxiv.org/abs/1712.04621> (date request 01.06.2018).
17. *Announcing NVIDIA DALI and NVIDIA nvJPEG* [Electronical Resource]. URL: <https://news.developer.nvidia.com/announcing-nvidia-dali-and-nvidia-nvjpeg/> (date request 19.06.2018).
18. *Bloice D.* Augmentor: An Image Augmentation Library for Machine Learning / Bloice D., Stocker C., Holzinger A. // ArXiv e-prints [Electronical Resource] – ArXiv:1712.04621 – 2017 – URL: <https://arxiv.org/abs/1712.04621> (date request 15.06.2018).
19. *Hasanpour S.H.* Let’s keep it simple, Using simple architectures to outperform deeper and more complex architectures / Hasanpour S. H., Rouhani M., Mohsen F., Sabokrou M. // ArXiv e-prints [Electronical Resource] – ArXiv:1608.06037 – 2016 – URL: <https://arxiv.org/abs/1608.06037> (date request 25.05.2018).

**A.V. Gayer.** LLC “Smart Engines Service”, Prospekt 60-Letiya Oktyabrya, 9, Moscow, 117312, Russia.  
E-mail: [gayer.alexandr@yandex.ru](mailto:gayer.alexandr@yandex.ru)

**A.V. Sheshkus.** LLC “Smart Engines Service”, Prospekt 60-Letiya Oktyabrya, 9, Moscow, 117312, Russia.  
E-mail: [astdcall@gmail.com](mailto:astdcall@gmail.com)

**Y.S. Chernyshova.** Institute for Systems Analysis, Federal Research Center “Computer Science and Control” of Russian Academy of Sciences, prospekt 60-Letiya Oktyabrya, 9, Moscow, 117312, Russia.  
E-mail: [chernyshovayulia07@gmail.com](mailto:chernyshovayulia07@gmail.com)