Научная статья
Original article
УДК 004.8

ПРИМЕНЕНИЕ АУГМЕНТАЦИИ ДЛЯ УЛУЧШЕНИЯ КАЧЕСТВА КЛАССИФИКАЦИИ

THE USE OF AUGMENTATION TO IMPROVE THE QUALITY OF CLASSIFICATION

Афанасьев Дмитрий Юрьевич, студент 2 курса магистратуры по направлению подготовки «Информатика и вычислительная техника» (Искусственный интеллект), ФГБОУ ВО «Иркутский национальный исследовательский технический университет» (664074 Россия, г. Иркутск, ул. Лермонтова, д. 83)

Afanasyev Dmitry Yurievich, 2nd year Master's student in the field of study «Informatics and Computer Engineering» (Artificial Intelligence), Federal State Budget Educational Institution of Higher Education «Irkutsk National Research Technical University» (83 Lermontova st., Irkutsk, 664074 Russia)

Аннотация. В статье рассмотрено применение различных методов аугментации и сравнение их эффективности для улучшения качества классификации на наборе данных CIFAR-10 [1]. Применяются наиболее распространенные методы аугментации, такие как добавление шума, размытие, масштабирование, поворот, вращение, смещение, случайное стирание. Представлены графики точности и потерь при обучении на исходном и измененном различными аугментациями наборе данных, а также структура модели сверточной нейронной сети. Проведено сравнение эффективности используемых методов

аугментации. Практическая значимость проведенного исследования заключается в том, что результаты и материалы данной работы могут использоваться при выборе наиболее эффективных методов аугментации для создания дополнительных обучающих данных из имеющихся.

Annotation. The article discusses the application of various augmentation methods and the comparison of their effectiveness to improve the quality of classification on the CIFAR-10 dataset [1]. The most common augmentation methods are used, such as adding noise, blurring, scaling, rotation, rotation, displacement, random erasure. Graphs of accuracy and losses during training on the original and modified by various augmentations data set, as well as the structure of the convolutional neural network model are presented. The efficiency of the augmentation methods used is compared. The practical significance of the conducted research lies in the fact that the results and materials of this work can be used when choosing the most effective augmentation methods to create additional training data from the available ones.

Ключевые слова: аугментация, переобучение, несбалансированные выборки, классификация, компьютерное зрение, преобразования изображений, глубокое обучение, нейронные сети.

Keywords: augmentation, overfitting, unbalanced samples, classification, computer vision, image transformations, deep learning, neural networks.

Введение

На сегодняшний день существует множество областей, в которых применяют компьютерное зрение и сверточные нейронные сети. Одной из наиболее распространённых проблем при обучении нейронных сетей является небольшой объем данных. Проблема отсутствия достаточного количества данных заключается в том, что модель может не изучить все закономерности на основе имеющихся данных, а значит будет работать не так эффективно.

Но зачастую получение большого количества реальных данных – это тяжелый и дорогостоящий процесс. В таких случаях будет полезно использовать аугментацию – искусственное увеличение количества образов на основе преобразований при которых сохраняются выходные метки. В процессе

аугментации искусственно создаются разные версии объектов из реального набора для увеличения его объема, все новые данные создаются на основе уже имеющихся.

Эффект переобучения

Эффект переобучения – это разница в точности модели при оценке обучающих данных в сравнении с тестовыми данными. Такие модели показывают на тестовой выборке значительно меньшую точность, чем на обучающей выборке. При отсутствии переобучения модели ошибка на тестовых данных уменьшается совместно с ошибкой на обучающей выборке. Если ошибка обучения уменьшается, а ошибка в тестовых данных увеличивается, то такой сценарий характерен для переобучения.

Один из способов обнаружить переобучение – это построить график ошибок точности обучения и потерь для каждой эпохи во время обучения.

Аугментация данных – очень мощный способ борьбы с переобучением, она полезна при работе с несбалансированными выборками, когда количество данных одного класса значительно преобладают над данными другого класса. А также в случаях с недостаточным разнообразием данных [2].

Создание и обучение модели нейронной сети

Создадим нейронную сеть, чтобы классифицировать данные по одному из десяти классов. Первый вариант нейронной сети не будет использовать аугментацию данных. Последующие варианты будут использовать один из методов аугментации при обучении.

Модель представляет сверточную нейронную сеть типа VGG с 6 сверточными слоями и одним полносвязным слоем перед выходом [3]. Модель сверточной нейронной сети представлена на рисунке 1.

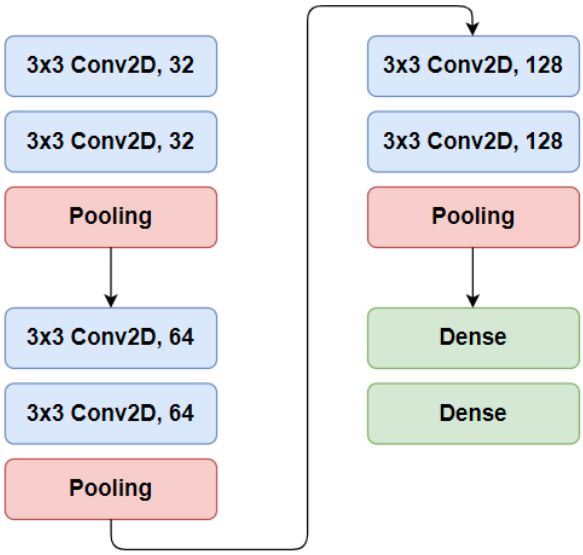

Рисунок 1 – Модель сверточной нейронной сети

Для обучения модели во всех случаях использовалось 50 эпох. После обучения модели получаем лучшую точность на обучающей выборке 100%, лучшую точность на проверочной выборке 71%. Графики точности и потерь представлены на рисунке 2.

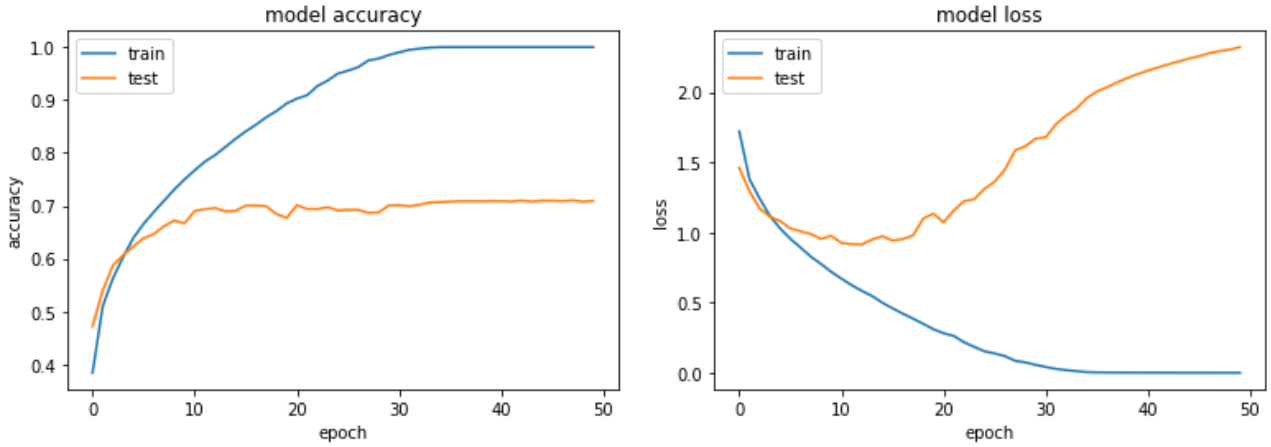

Рисунок 2 – Графики точности и потерь при обучении без аугментации

Разница между обучающими и тестовыми данными составила 29%, это произошло потому что модель переобучилась. Сверточная нейронная сеть начинает запоминать обучающие данные, по этой причине она с высокой точностью работает по данным из тренировочного набора, но когда модель начинает проверять точность на проверочной выборке, с которой она еще не сталкивалась, то результаты значительно отличаются.

По графику потерь можно увидеть, что после 12 эпохи значение функции потерь на обучающем наборе данных снижается, а значение функции потерь на тренировочном наборе возрастает, что является признаком переобучения нейронной сети.

Методы аугментации

Для реализации аугментаций использовались библиотеки Keras и Tensorflow которые имеют множество полезных функций. Особенно стоит отметить функцию ImageDataGenerator которая преобразует исходные изображения [4]. Вид преобразований зависит от аргументов, указанных в теле функции. Если среди аргументов ImageDataGenerator нет того, который подходит для реализации нужной аугментации можно создать свою функцию и использовать ее как аргумент.

Изображения с помощью ImageDataGenerator преобразуются каждую эпоху в качестве этапа предварительной обработки изображения, а затем используются для обучения модели. Благодаря этому способу мы избавляемся от необходимости обрабатывать все изображения и записывать их в каталог. Если аргументы для генератора подобраны правильно, то сгенерированные таким способом изображения сохраняют признаки и отличаются от исходных не полностью. Таким образом обученная модель будет более надежной и точной так как она обучается на разных вариантах одного изображения.

Изменение исходных изображений путем добавления шума является одним из способов увеличения количества изображений в наборе данных. Этот метод позволяет модели в будущем лучше работать с изображениями имеющими различные дефекты. Для реализации шума была написана функция, добавляющая случайные точки белого и черного цвета.

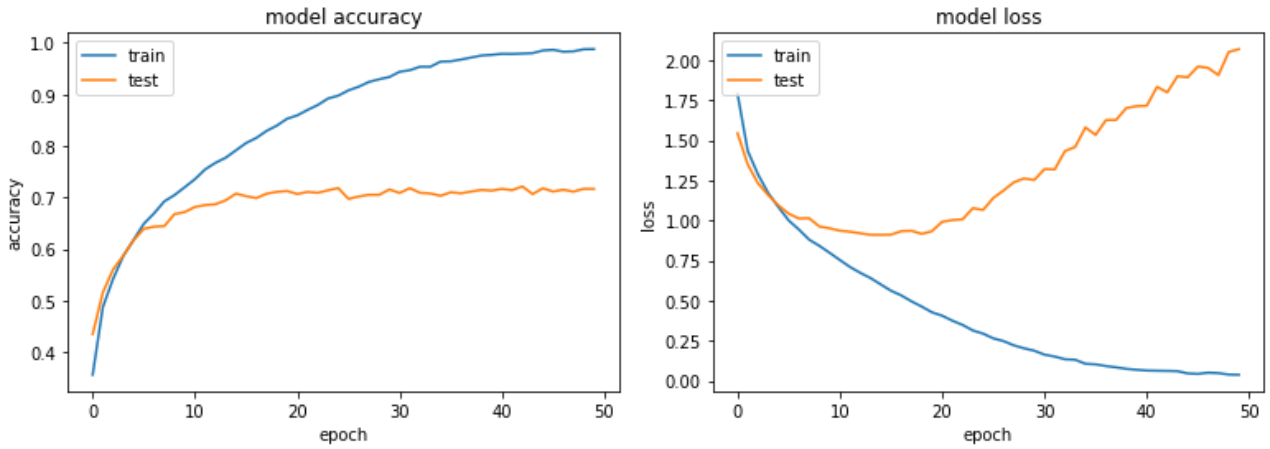

Рисунок 3 – Графики точности и потерь при обучении на наборе изображений с добавлением шума

На рисунке 3 видно, что добавление шума практически никак не повлияло на переобучение. Разница между точностью на обучающей и проверочной выборке все ещё велика. Функция потерь на тренировочном наборе данных после определенной эпохи начинает возрастать.

Ещё одна техника аугментации – размытие изображения. Размытие может сделать модель белее устойчивой к качеству используемых изображений. Применить размытие к изображением можно с помощью библиотеки imgaug, было использовано небольшое размытие по Гауссу с отклонением ядра от 0 до 0.5, так как размытие с большим отклонением ядра приводило к значительному уменьшению точности классификации.

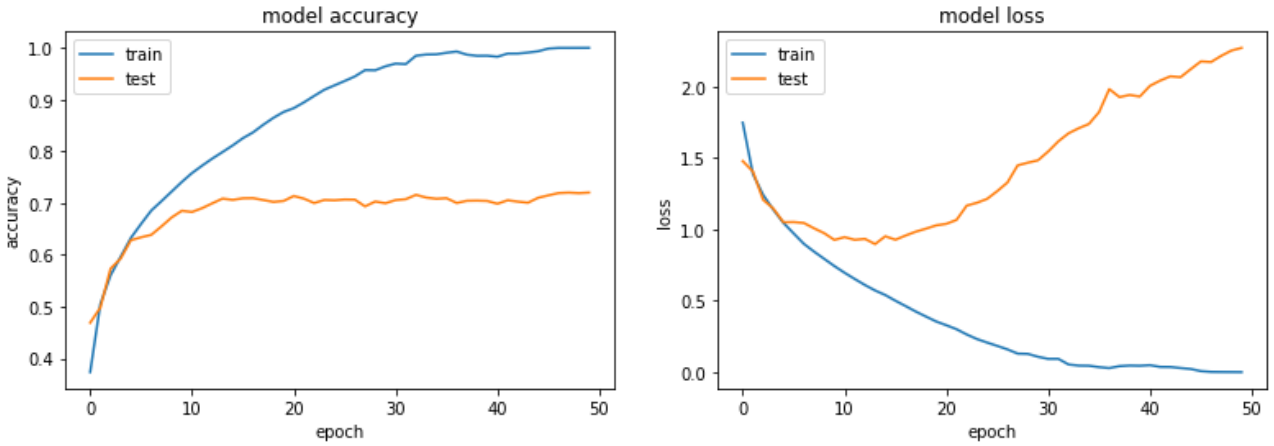

Рисунок 4 – Графики точности и потерь при обучении на наборе изображений с размытием

Как видно на рисунке 4 размытие незначительно повлияло на модель, скорее всего это связано с тем, что изображения в наборе данных имеют низкое разрешение.

В процессе масштабирования изображение может увеличиваться или сжиматься по горизонтали и вертикали. Этот метод делает обучение модели более эффективным, потому что на обучающих данных интересующие нас объекты не всегда могут иметь одинаковый размер и полностью находится в кадре [5]. Для выполнения масштабирования был использован аргумент zoom_range функции ImageDataGenerator. В данном аргументе указывается диапазон случайного масштабирования изображения. Для аугментации было выбрано случайное масштабирование от 0 до 50%, результаты можно увидеть на рисунке 5.

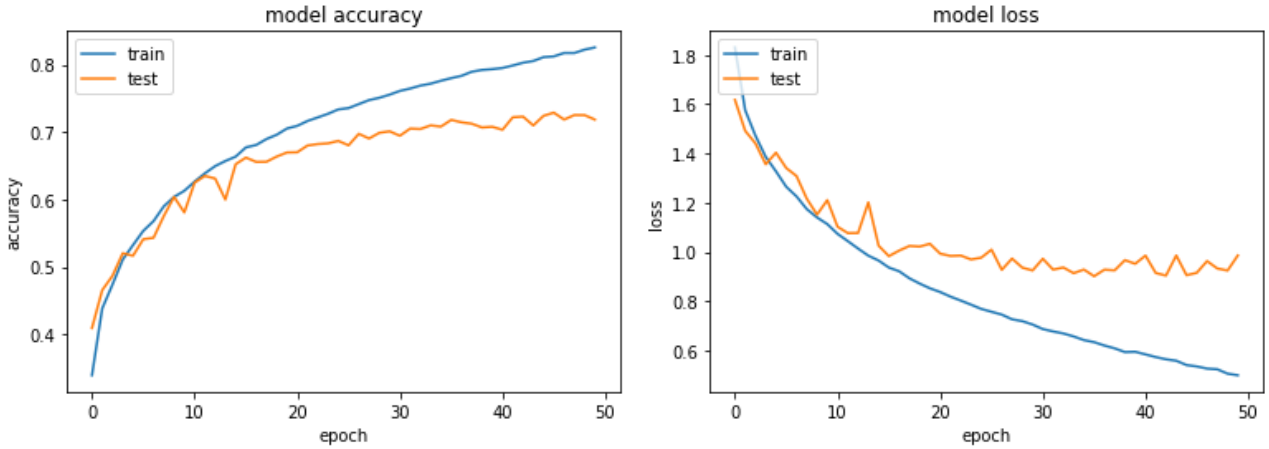

Рисунок 5 – Графики точности и потерь при обучении на наборе изображений с измененным масштабом

По полученным результатам можно сделать вывод что масштабирование не убирает переобучение полностью, но заметно его снижает.

Изображения могут поворачиваться по горизонтали или вертикали при этом меняется расположение пикселей, но сохраняются все особенности изображения. На практике поворот по горизонтальной оси гораздо более распространен, чем по вертикальной. Целесообразность применения горизонтального и вертикального поворота зависят от набора изображений, в некоторых ситуациях это преобразование не сохраняет метку.

Повороты изображения можно реализовать с помощью аргументов horizontal_flip и vertical_flip функции ImageDataGenerator. Которые реализуют

случайные горизонтальные и вертикальные повороты. Для аугментации набора данных использовались только горизонтальные повороты.

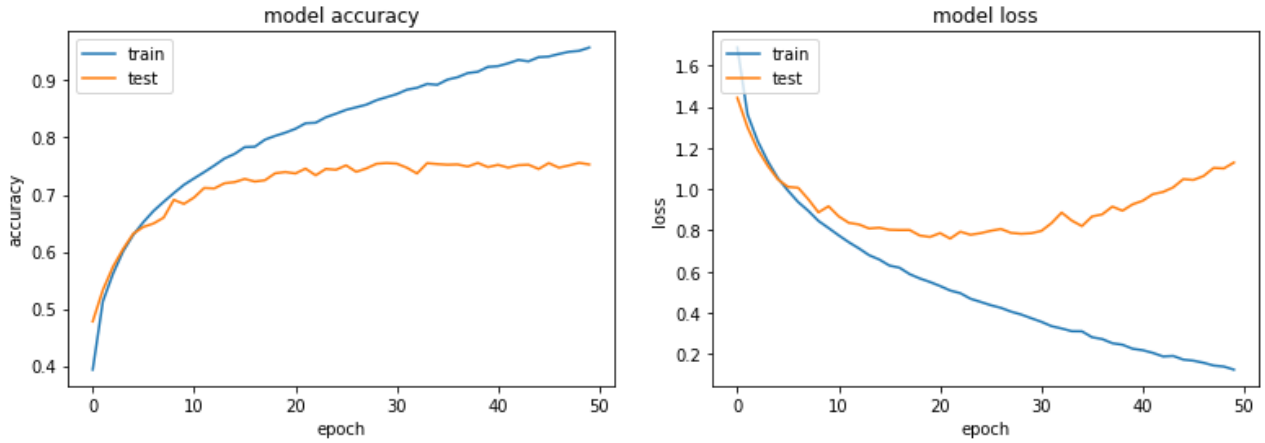

Рисунок 6 – Графики точности и потерь при обучении на наборе изображений со случайными горизонтальными поворотами

На графиках рисунка 6 видно, что модель продолжает переобучаться, но повороты немного улучшили обучение по сравнению с такими методами как размытие и шум.

Одним из наиболее часто используемых методов аугментации является вращение изображения. Изображение может поворачиваться вправо или влево по оси на угол от 1 до 359 градусов при этом информация на изображении остается прежней [6]. Применение вращения безопасно не для всех наборов изображений, целесообразно применять небольшие вращения от 1 до 20 градусов, так как по мере увеличения угла метка данных может не сохраняться.

Выполнить вращения можно используя аргумент rotation_range функции ImageDataGenerator. Данному аргументу для выполнения случайных вращения необходимо указать диапазон градусов. Были выбраны два различных диапазона значений, в первом случае 70 градусов – графики представлены на рисунке 7, во втором 20 градусов – рисунок 8.

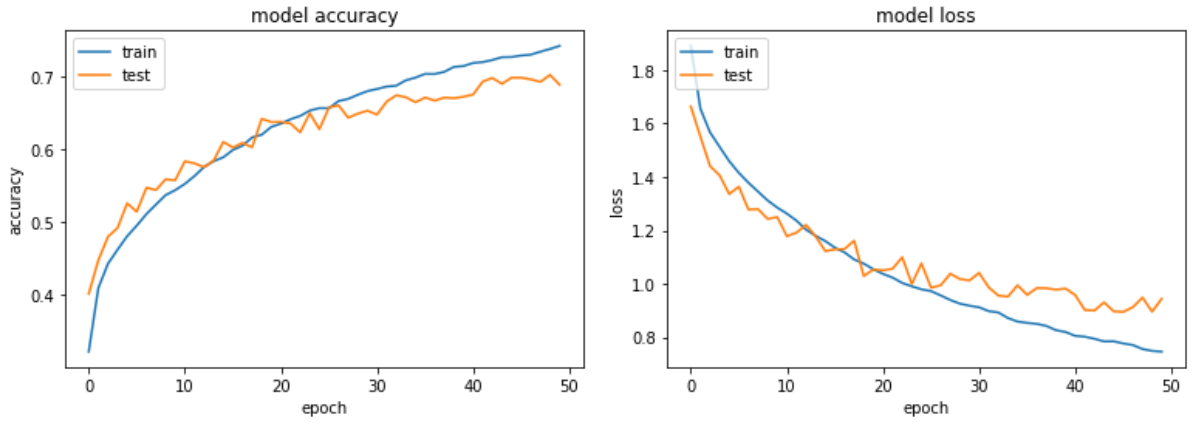

Рисунок 7 – Графики точности и потерь при обучении на наборе изображений со случайными вращениями 70 градусов

При 70 градусах точность классификации на обучающей и проверочной выборке немного отличается. Потери даже после 50 эпох продолжают медленно снижаться. Так как точность модели продолжает расти, а потери снижаться возможно улучшение модели за счет более длительного обучения. Можно сделать вывод, что аугментации методом вращения являются полезным инструментом в борьбе с переобучением.

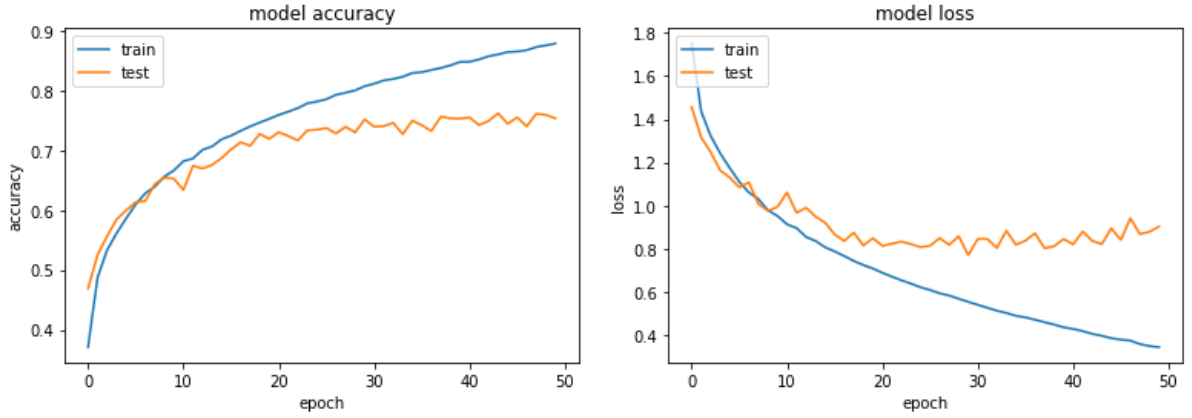

Рисунок 8 – Графики точности и потерь при обучении на наборе изображений со случайными вращениями 20 градусов

При 20 градусах эффект переобучения снизился по сравнению с базовой моделью, но использование вращение на большие углы дает гораздо более сильный эффект.

Смещение – это геометрическое преобразование, которое сопоставляет положение каждого пикселя в изображении с новым местоположением. Таким образом, изображение смещается вдоль оси X или Y или по обоим направлениям. Смещая изображения, можно изменять положение объектов и, следовательно,

придать модели больше разнообразия, что очень полезно поскольку в большинстве случаев объекты могут располагаться практически в любом месте изображения.

Смещения можно выполнить используя аргументы width_shift_range и height_shift_range функции ImageDataGenerator реализующие горизонтальные и вертикальные сдвиги изображения. В качестве значений аргументов можно использовать процент от общей ширины и/или высоты изображения. Использовались значения аргументов width_shift_range = 0.1 и height_shift_range = 0.1 при которых изображение будет произвольно перемещаться по горизонтали и вертикали между диапазоном от -10% до +10% ширины и высоты всего изображения. На рисунке 9 представлены графики точности и потерь при обучении.

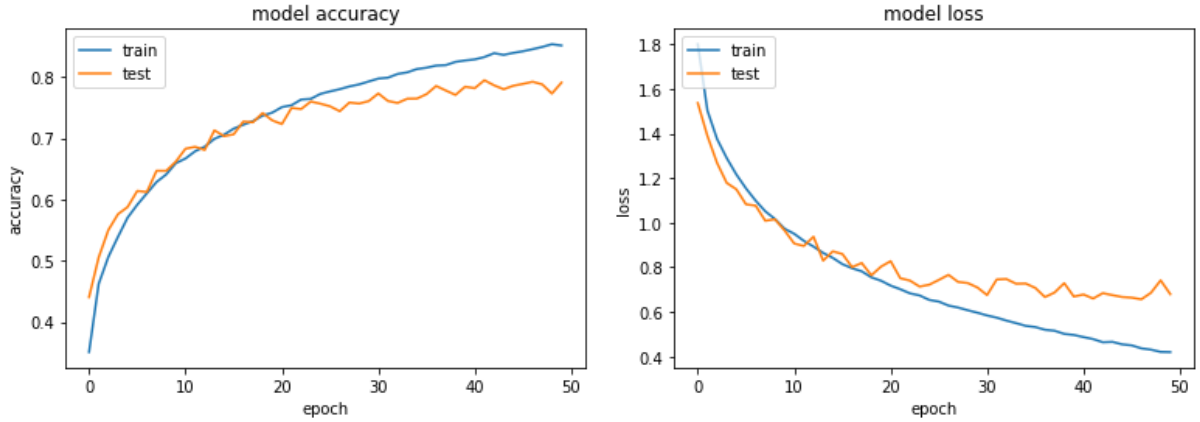

Рисунок 9 – Графики точности и потерь при обучении на наборе изображений со смещениями от -10% до +10% по горизонтали и вертикали

Потери на тренировочном наборе продолжают медленно снижаться на всех эпохах, а точность увеличиваться. Возможно при увеличении длительности обучения получиться добиться лучшего результата. Можно сделать вывод что аугментация методом смещения помогает уменьшить эффект переобучения.

Аугментация методом случайного стирания выбирает на изображении прямоугольную область и стирает ее, это заставляет модель обращать внимание на все изображение, а не только на его часть. Например, если модель не может распознать объект потому что его часть находится в тени или символы на изображении имеют нечеткие части, в этом случае применения случайного стирания на обучающей выборке может улучшить модель [7].

Для выполнения случайных стираний была написана функция, заменяющая случайный участок изображения прямоугольником случайной формы черного цвета, высота и ширина которого не может быть больше чем 20% от высоты и ширины изображения, результаты можно увидеть на рисунке 10.

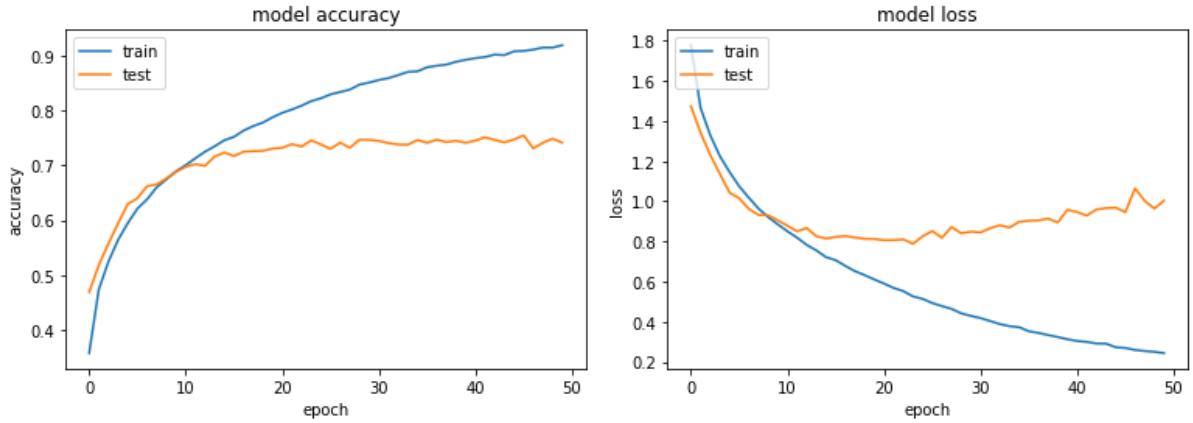

Рисунок 10 – Графики точности и потерь при обучении на наборе изображений со случайными стираниями

Модель продолжает переобучаться, но эффект переобучения меньше чем у таких методов аугментации как размытие и шум.

Сравнение результатов различных методов аугментации

Чтобы наглядно сравнить результаты различных методов аугментации постоим таблицу 1 в которой укажем основные показатели модели нейронной сети указывающие на переобучение, такие как:

- A – лучшая точность на проверочной выборке;
- Dif – разница между обучающей и тренировочной выборкой – чем она меньше, тем меньшим эффектом переобучения обладает модель;
- Le – эпоха, с которой начинают увеличиваться потери на тренировочном наборе данных – чем позже начинают увеличиваться потери, тем лучше, при полном отсутствии переобучения они всегда будут уменьшаться вместе с потерями на обучающем наборе;
- Ae – эпоха, с которой перестает увеличиваться точность на тренировочном наборе – чем позже перестает увеличиваться точность, тем лучше, при полном отсутствии переобучения тренировочная точность будет совпадать с обучающей точность.

Таблица 1 – Изменение основных показателей в зависимости от используемого метода аугментации

Метод аугментации		A	Dif	Le	Ae
Вращение	70	74%	4%	-	-
	20	76%	12%	38	44
Смещение		79%	6%	-	-
Изменение масштаба		73%	9%	43	46
Случайное стирание		75%	17%	22	42
Поворот		75%	20%	22	33
Добавление шума		72%	24%	14	43
Размытие		72%	27%	14	46
Без аугментации		71%	29%	12	37

Заключение

Из исследуемых методов аугментации лучше всего показали себе аугментации основанные на вращении изображения при этом угол вращения составлял до 70 градусов, при меньших углах вращения эффективность падает. Применение данного вида аугментации на наборе данных CIFAR-10 по сравнению с обучением без аугментации позволило повысить точность классификации на тренировочном наборе данных на 3%, снизить разницу между обучающей и тренировочной точностью до 4% что является хорошим результатом, а также избавиться от увеличения потерь и уменьшения точности со временем, это позволит в дальнейшем улучшить модель путем более длительного обучения.

Аугментации путем смещения изображения также показали хороший результат. Точность распознавания на тестовом наборе выросла на 8%, разница между точностью на обучающем и тренировочном наборе снизилась до 6% потери перестали увеличиваться, а точность снижаться со временем.

Использование аугментации методом масштабирование позволило увеличить точность на тренировочном наборе на 2%, снизить разницу между точностью на обучающем и тренировочном наборе на 20%, потери теперь начинают

увеличиваться на 31 эпоху позже, а точность уменьшаться на 9 эпох позже. Видно улучшение по уменьшению эффекта переобучения по сравнению с обучением без аугментации.

Применение аугментации способом случайных стираний увеличивает точность на тренировочном наборе на 4%, снижает разницу между обучающей и тренировочной точностью на 12%, а также на 10 эпох задерживает время начала увеличение потерь, точность продолжает увеличиваться на 5 эпох дольше.

Аугментация методом переворота по горизонтали увеличивает точность на 4%, уменьшает разницу между обучающей и тренировочной точностью на 9%, увеличение потерь начинается на 10 эпох позже.

Аугментации, использующие шум и размытие показали незначительные уменьшения эффекта переобучения, разница между обучающей и тренировочной точностью снизилась на 5% и 2% соответственно, потери начинают увеличиваться на 2 эпохи позже.

Литература

1. The CIFAR-10 dataset [Электронный ресурс]. – Режим доступа: https://www.cs.toronto.edu/~kriz/cifar.html. – (дата обращения: 10.03.2022).
2. Alexander Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I. Iglovikov, Alexandr A. Kalinin. Albumentations: fast and flexible image augmentations [Электронный ресурс] // CoRR. 2018. URL: https://arxiv.org/pdf/1809.06839.pdf (дата обращения: 10.03.2022).
3. Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition [Электронный ресурс] // ICLR. 2015. URL: https://arxiv.org/pdf/1409.1556.pdf (дата обращения: 11.03.2022).
4. Generate batches of tensor image data with real-time data augmentation [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator. – (дата обращения: 14.03.2022).
5. Yukang Chen, Yanwei Li, Tao Kong, Lu Qi, Ruihang Chu, Lei Li, Jiaya Jia. Scale-aware Automatic Augmentation for Object Detection // CVPR. 2021. P. 9563-9572.

6. Rowel Atienza. Data Augmentation for Scene Text Recognition // ICCVW. 2021. P. 1561-1570.
7. Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, Yi Yang. Random Erasing Data Augmentation // AAAI. 2020. P. 13001-13008.

References

1. The CIFAR-10 dataset [Electronic resource]. – Access mode: https://www.cs.toronto.edu/~kriz/cifar.html. – (date of access: 03/10/2022).
2. Alexander Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I. Iglovikov, Alexandr A. Kalinin. Albumentations: fast and flexible image augmentations [Electronic resource] // CoRR. 2018. URL: https://arxiv.org/pdf/1809.06839.pdf (date of access: 03/10/2022).
3. Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition [Electronic resource] // ICLR. 2015. URL: https://arxiv.org/pdf/1409.1556.pdf (date of access: 03/11/2022).
4. Generate batches of tensor image data with real-time data augmentation [Electronic resource]. – Access mode: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator. – (date of access: 03/14/2022).
5. Yukang Chen, Yanwei Li, Tao Kong, Lu Qi, Ruihang Chu, Lei Li, Jiaya Jia. Scale-aware Automatic Augmentation for Object Detection // CVPR. 2021. P. 9563-9572.
6. Rowel Atienza. Data Augmentation for Scene Text Recognition // ICCVW. 2021. P. 1561-1570.
7. Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, Yi Yang. Random Erasing Data Augmentation // AAAI. 2020. P. 13001-13008.

© Афанасьев Д.Ю., 2022 Научно-образовательный журнал для студентов и преподавателей «StudNet» №4/2022.

Для цитирования: Афанасьев Д.Ю. ПРИМЕНЕНИЕ АУГМЕНТАЦИИ ДЛЯ УЛУЧШЕНИЯ КАЧЕСТВА КЛАССИФИКАЦИИ// Научно-образовательный журнал для студентов и преподавателей «StudNet» №4/2022.