

Исследование влияния размеров рецептивного поля на результат обучения свёрточных нейронных сетей

Р.Р. Габбасов¹, Р.А. Парингер^{1,2}

¹Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

²Институт систем обработки изображений РАН - филиал ФНИЦ «Кристаллография и фотоника» РАН, Молодогвардейская 151, Самара, Россия, 443001

Аннотация. Свёрточные нейронные сети (СНС) успешно применяются для решения многих задач, таких как распознавание цифр и объектов. В данной работе мы изучаем размер рецептивного поля глубоких свёрточных нейронных сетей, в частности, проверяем идею «избыточного» рецептивного поля. Мы проводим ряд экспериментов на двух распространенных моделях СНС — VGG16 и ResNet18 — с целью изучения влияния размера рецептивного поля на время обучения, точность и производительность СНС. Мы проводим эксперименты с использованием фреймворка MakiFlow на наборе данных CALTECH256. Результаты экспериментов показали, что оптимизация нейронных сетей путём уменьшения размера рецептивного поля позволяет добиться ускорения обучения сети на 5-7% с сохранением точности работы сети.

1. Введение

На сегодняшний день СНС широко используются для решения задач компьютерного зрения. Нужно заметить, что в этой области они значительно превосходят по эффективности полносвязные нейронные сети. Применение последних к такому типу задач оказывается неэффективным по следующим причинам [1]:

1) большая размерность входного изображения предполагает наличие в каждом слое полносвязной нейронной сети огромного количества синаптических связей, вследствие чего возрастает размер нейронной сети, и, соответственно, увеличивается время обучения сети;

2) полносвязная нейронная сеть чувствительна к изменениям масштаба изображения, ракурса камеры и другим геометрическим трансформациям входного изображения, инвариантность к искажениям подобного вида у неё отсутствует;

3) полносвязная нейронная сеть игнорирует топологию входных данных: входные компоненты могут быть представлены в любом фиксированном порядке без влияния на результат обучения; изображения же, однако, имеют четкую двумерную структуру, значения соседних пикселей на любом локальном участке сильно коррелируют между собой.

Поэтому для решения задач, связанных с изображениями, вместо полносвязных нейронных сетей лучше подходят свёрточные нейронные сети, поскольку последние в ходе своей работы способны выделять и комбинировать локальные признаки, обладая притом меньшим числом обучаемых параметров и частичной устойчивостью к геометрическим искажениям входного изображения.

Свёрточная нейронная сеть – это специальная архитектура нейронной сети, созданная для работы с изображениями и предложенная Яном Лекуном в 1988 году [2]. Структурно она воспроизводит особенности строения зрительной коры мозга [3], в которой есть так называемые простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определённого набора простых клеток.

Ключевым компонентом свёрточной нейронной сети является свёрточный слой. Каждый такой слой производит операцию свёртки на входном изображении. В сущности, данная операция представляет собой поэлементное умножение весов ядра свёртки (матрицы весов небольшого размера) на соразмерный ему участок изображения. Результат умножения суммируется, формируя значение свёртки. Данное ядро применяет свёртку ко всему входному изображению, перемещаясь вдоль него подобно скользящему окну с определенным шагом. Результирующие значения свёрток формируют так называемую карту признаков, которая в дальнейшем обрабатывается схожим образом, но другими свёрточными слоями. На одном свёрточном слое действуют несколько ядер свёртки, каждое из которых формирует свою карту признаков. Каждое ядро свёртки «ответственно» за выделение какого-либо одного признака.

Визуализация работы двух ядер свёртки приведена на рисунке 1. На вход подается одна карта признаков, после работы двух свёрточных фильтров получают две карты признаков.

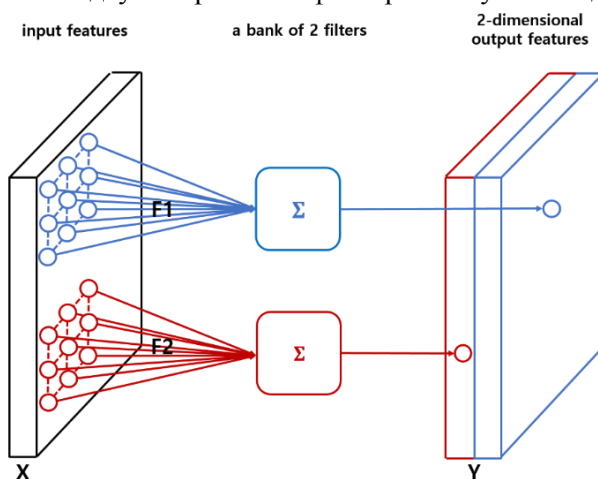


Рисунок 1. Визуализация работы двух свёрточных фильтров.

Для активации признаков используется слой активации. Его работа заключается в пропускании всех значений из карты признаков через нелинейную функцию активации. Этот слой обычно логически объединяют со слоем свёртки, считая, что функция активации встроена в слой свёртки. В качестве функции активации могут использоваться традиционные гиперболический тангенс или сигмоида, но, начиная с 2000 годов, себя зарекомендовала функция ReLU [4], определяемая как $f(x) = \max(0, x)$, и в настоящее время она является наиболее часто используемой функцией активации в данном типе нейронных сетей.

Для взятия наиболее значимых признаков, а также уменьшения пространственных размеров карт признаков используется слой подвыборки. В данном слое скользящее окно, обычно, размера 2×2 (эти значения могут отличаться) с шагом, равным своему размеру, (здесь – с шагом в 2 пикселя) проходит по всей карте признаков, последовательно выбирая наибольшие значения в захваченном участке (либо считая среднее из них). В результате получается карта признаков размером меньше исходной (для приведенных здесь значений — в 2 раза).

Эти слои могут быть скомбинированы в различном порядке, поэтому существует множество моделей свёрточных нейронных сетей. Общим в них является то, что свёрточные слои чередуются со слоями подвыборки. Этот факт и обеспечивает схожесть работы свёрточной нейронной сети с работой зрительной коры мозга.

На определённом этапе, когда размеры карт признаков достаточно малы, множество карт трансформируется в вектор, содержащий информацию о наиболее абстрактных признаках,

которые сети удалось выделить. Этот вектор в дальнейшем подаётся на вход полносвязной нейронной сети, например, для решения задачи классификации.

Пример типичной структуры свёрточной нейронной сети приведен на рисунке 2.

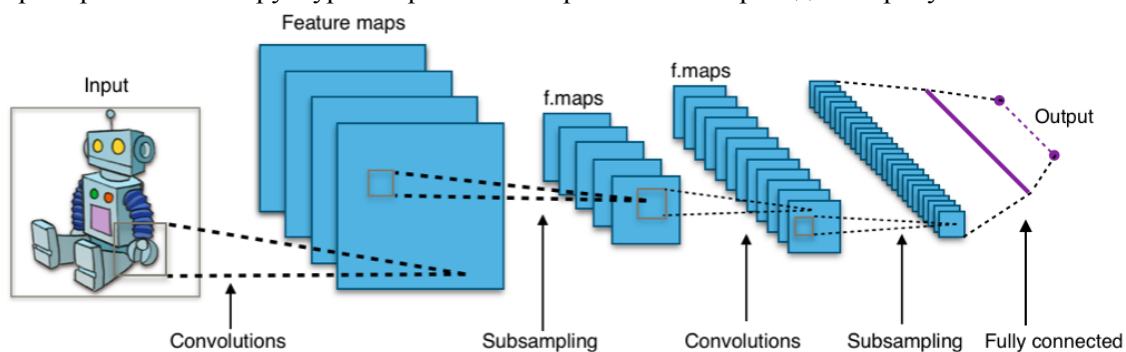


Рисунок 2. Типичная структура свёрточной нейронной сети.

Важным понятием, связанным со свёрточной нейронной сетью, является понятие *рецептивного поля*. Этим термином обозначается участок карты признаков, «воспринимаемый» свёрточным ядром на очередном слое сети, то есть та область, над которой проводится операция свёртки.

Можно заметить, что, в силу архитектуры сети, чем глубже внутри сети относительно её входа расположен свёрточный фильтр (свёрточное ядро), тем больше участок входного изображения, информация о котором «доходит» до него. Такой участок условимся называть *эффективным рецептивным полем* фильтра [5], а применительно к фильтрам на последнем слое свёрточной нейронной сети — *рецептивным полем всей сети*.

2. Постановка задачи

Известные архитектуры свёрточных нейронных сетей чаще всего обучаются на больших кластерах. В настоящее время подобные вычислительные мощности пока что еще нельзя называть широко доступными. По этой причине является актуальной задача оптимизации архитектуры сети, как один из возможных вариантов решения этой проблемы.

Одним из способов оптимизации архитектуры сети теоретически может являться уменьшение рецептивного поля сети (в разумных пределах), так как оно соответствует уменьшению количества обучаемых параметров. В данной статье мы проверим эту мысль и ответим на вопрос, можно ли вообще говорить об уменьшении рецептивного поля как о процедуре, по крайней мере не ухудшающей результат обучения сети.

2.1. Идея избыточного рецептивного поля

В данной работе мы предлагаем идею того, что рецептивное поле всей сети может оказаться избыточным по отношению к размеру входного изображения, когда размер рецептивного поля сети в несколько раз превышает последний, и что уменьшение размера рецептивного поля ближе к размерам входного изображения не ухудшает точность работы сети, позволяя притом получить выигрыш во времени её обучения за счёт уменьшения количества обучаемых параметров.

3. Подсчёт размера рецептивного поля сети

Размер рецептивного поля сети обуславливается размерами фильтров, величиной, на которую сдвигается фильтр, а также количеством слоев и их взаимным расположением [6]. Изменяя эти параметры, можно добиться изменения размера рецептивного поля всей сети.

Приведем конкретные формулы для подсчёта размера эффективного рецептивного поля на очередном слое сети. Размер r эффективного рецептивного поля на текущем слое выражается через следующую рекуррентную формулу [7]:

$$r_{\text{out}} = r_{\text{in}} + (k - 1) * j_{\text{in}}, \quad (1)$$

в которой величина j связана рекуррентным соотношением:

$$j_{out} = j_{in} * s, \quad (2)$$

где k – размер фильтра, s – размер «шага» (stride) фильтра, индекс out означает значение величины для текущего слоя, in – для предыдущего. Стоит отметить, что эта формула применима не только для свёрточных слоев, но и для слоёв подвыборки.

4. Постановка эксперимента

4.1. Подготовка набора данных для обучения

В данной работе мы проверяем вышеизложенную идею для сетей, решающих задачу классификации. В качестве набора данных для обучения были взяты изображения из 10 классов датасета Caltech256 [8], содержащих равное количество изображений. Одна седьмая часть изображений из каждого класса была использована для составления тестовой выборки. Множество остальных изображений составило тренировочную выборку, и она была расширена с применением к изображениям поворотов до 60 градусов, гауссова шума с дисперсией до 20, отражений по вертикали или горизонтали примерно половины изображений. В результате тренировочная выборка содержала 9666 изображений, тестовая — 297.

4.2. Модели сетей, использованные для обучения

Для экспериментов были выбраны две популярные модели свёрточных нейронных сетей: VGG16 и ResNet18. Сети были построены и в дальнейшем обучены с использованием фреймворка MakiFlow [9].

4.2.1. VGG16

VGG16 — модель сверточной нейронной сети, предложенная К. Simonyan и А. Zisserman из Оксфордского университета в статье “Very Deep Convolutional Networks for Large-Scale Image Recognition” [10].

Структура сети представлена на рисунке 3.



Рисунок 3. Структура сети VGG16.

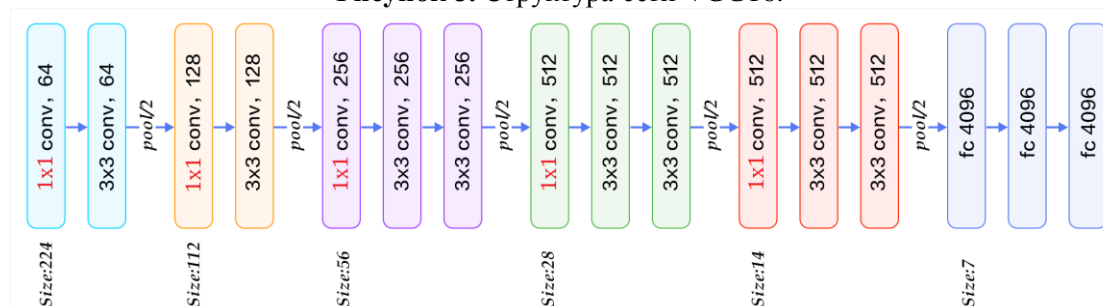


Рисунок 4. Структура модифицированной сети VGG16.

Стоит отметить, что в нашей реализации этой модели сети на каждом слое была использована пакетная нормализация (batch normalization) [11]. Приведем значение размера рецептивного поля этой сети. Посчитанное по формулам (1) и (2), оно равно 212 (то есть рецептивное поле сети — квадрат 212x212).

4.2.2. Модифицированная VGG16

С целью уменьшения рецептивного поля сети были уменьшены размеры свёрточных ядер на первом слое каждого блока свёрточных слоёв с 3x3 до 1x1, так, как показано на рисунке 4.

Значение размера рецептивного поля модифицированной сети, посчитанное по формулам (1) и (2), равно 150 (то есть поле — квадрат 150x150).

4.2.3. ResNet18

Структура сети ResNet18 [12] представлена на рисунке 5.

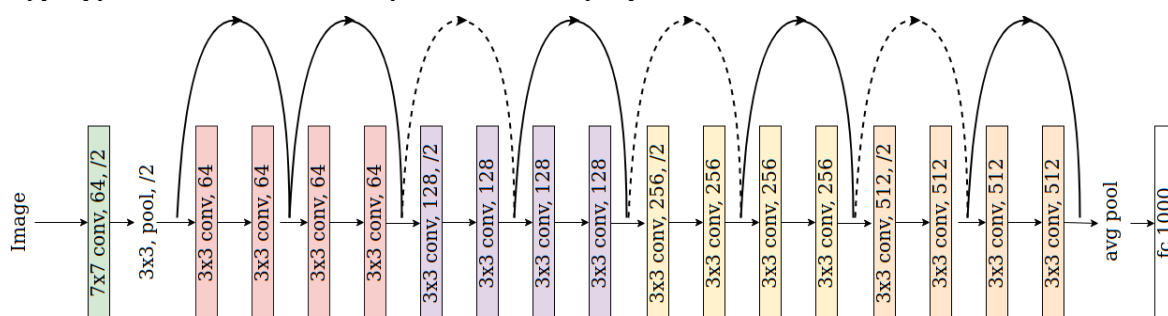


Рисунок 5. Структура сети ResNet18.

Приведем значение размера рецептивного поля этой сети. Посчитанное по формулам (1) и (2), оно равно 435 (то есть поле — квадрат 435x435).

4.2.4. Модифицированная ResNet18

С целью уменьшения рецептивного поля сети были уменьшены размеры некоторых свёрточных ядер с 3x3 до 1x1, так, как показано на рисунке 6.

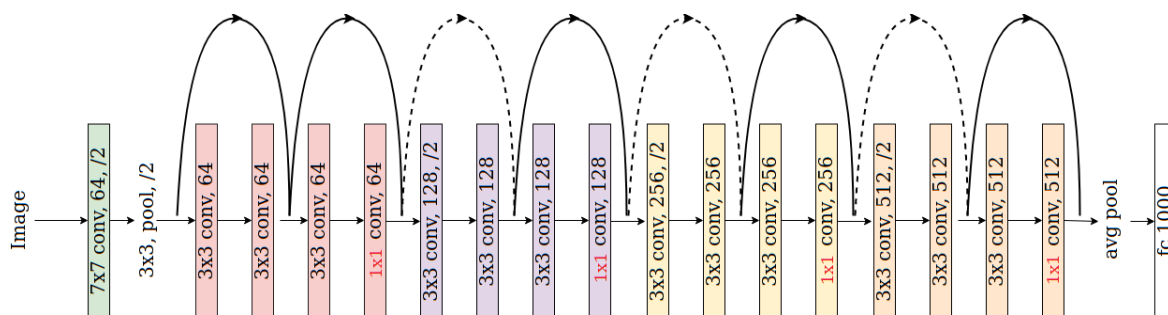


Рисунок 6. Структура модифицированной сети ResNet18.

Значение размера рецептивного поля модифицированной сети, посчитанное по формулам (1) и (2), равно 315 (поле — квадрат 315x315). тренировочная выборка содержала 9666 изображений, тестовая — 297.

4.3. Процесс обучения

Описанные ниже параметры обучения были подобраны в результате тестов и наблюдений за изменением точности и значений функций потерь в процессе обучения. На приведенных ниже параметрах обучения каждая модель сети была затем обучена пять раз, и результаты обучения для каждой из них были потом усреднены.

4.3.1. VGG16

Каждое изображение, поступавшее на вход сети, было приведено к размеру 128x128. Таким образом, размер рецептивного поля сети после модификации оказывался ближе к размеру входного изображения, чем до неё. Обе сети были обучены с помощью адаптивного

оптимизатора Adam [13] при следующих параметрах: learning rate = 0,001; gamma = 0,5; количество эпох — 10; функция потерь — Focal Loss.

4.3.2. ResNet18

Каждое изображение, поступавшее на вход сети, было приведено к размеру 256x256. Таким образом, размер рецептивного поля сети после модификации оказывался ближе к размеру входного изображения, чем до неё. Обе сети были обучены с помощью оптимизатора RMSProp при следующих параметрах: learning rate = 0,0005; gamma — 2 для оригинальной, 1 для модифицированной; количество эпох — 20; функция потерь — Focal Loss.

5. Результаты экспериментов

5.1. VGG16

Графики значений функции потерь и функции ошибки сети представлены ниже на рисунках 7 и 8.

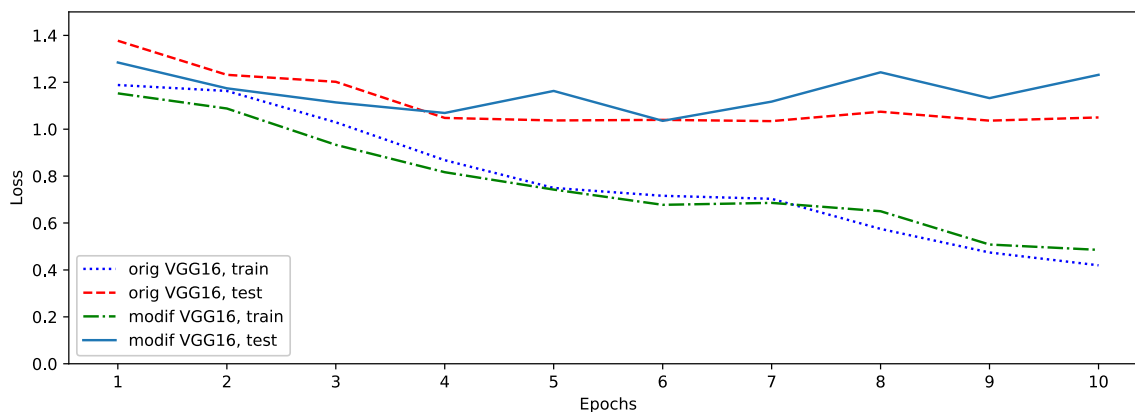


Рисунок 7. График функции потерь для VGG16.

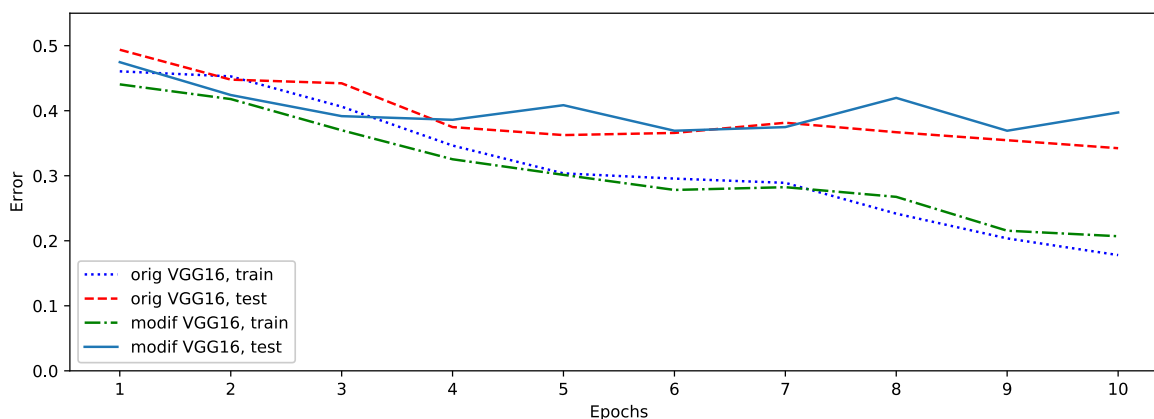


Рисунок 8. График функции ошибки для VGG16.

5.2. ResNet18

Графики значений функции потерь и функции ошибки сети представлены ниже на рисунках 9 и 10.

5.3. Анализ результатов

Полученные результаты свидетельствуют о том, что оптимизация нейронных сетей за счёт уменьшения рецептивного поля позволяет добиться ускорения обучения сети с сохранением точности. Действительно, произведенное описанным выше образом уменьшение рецептивного поля сети сохранило уровень точности работы сети, причём выигрыш по времени (то, на

сколько быстрее в среднем проходила очередная эпоха обучения) составил 7,48% для VGG16 и 5,06% для ResNet18. Это связано с уменьшением количества обучаемых параметров.

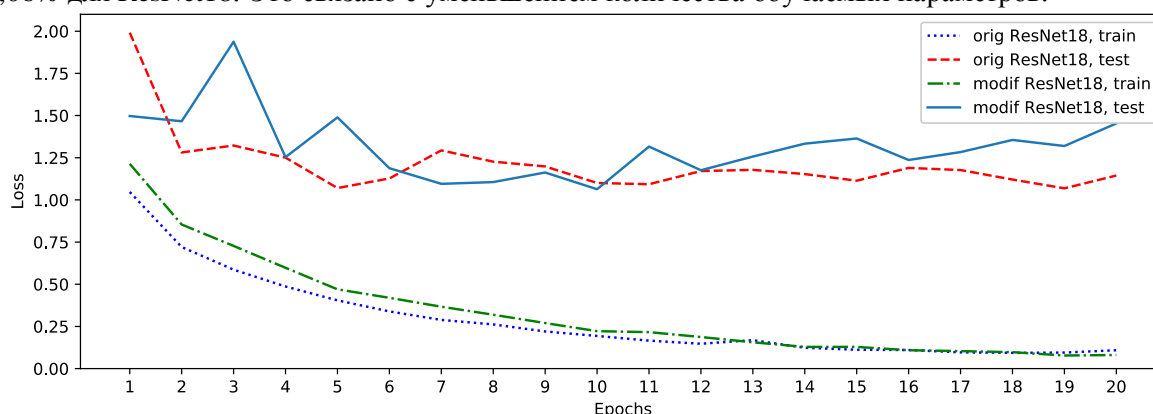


Рисунок 9. График функции потерь для ResNet18.

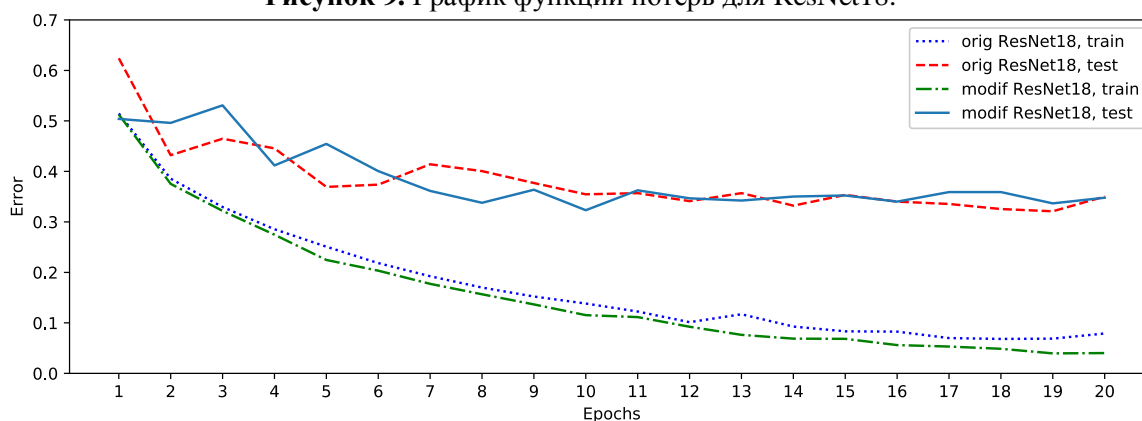


Рисунок 10. График функции ошибки для ResNet18.

6. Заключение

В ходе проведенных экспериментов, мы наблюдали, что уменьшение рецептивного поля сети так, что его размер оказывается ближе к размеру входного изображения, (то есть избавление от избыточного рецептивного поля сети) практически не влияет на точность работы сети. В то же время происходит уменьшение количества обучаемых параметров, что дает выигрыш во времени обучения сети. Это позволяет говорить о описанной процедуре уменьшения рецептивного поля как об оптимизирующей процедуре, а также позволяет высказывать дальнейшие идеи о том, как совершать её наилучшим образом в зависимости от тех или иных параметров.

7. Благодарности

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (№ 19-29-01135).

8. Литература

- [1] LeCun, Y. Convolutional networks for images, speech, and time series / Y. LeCun, Y. Bengio // The handbook of brain theory and neural networks – The MIT Press, Cambridge, 1995. – P. 276-279.
- [2] LeCun, Y. Backpropagation applied to handwritten zip code recognition / Y. LeCun, B. Boser, J. S. Denker // Neural computation. – 1989. – Vol. 1(4). – P. 541-551.
- [3] Matsugu, M. Subject independent facial expression recognition with robust face detection using a convolutional neural network / M. Matsugu, K. Mori, Y. Mitari, Y. Kaneda // Neural Networks. – 2003. – Vol. 16(5-6). – P. 555-559.

- [4] Krizhevsky, A. ImageNet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G. E. Hinton // *Advances in neural information processing systems*. – 2012. – P. 1097-1105.
- [5] Le, H. What are the receptive, effective receptive, and projective fields of neurons in convolutional neural networks? / H. Le, A. Borji // *arXiv preprint arXiv:1705.07049*, 2017.
- [6] Dumoulin, V. A guide to convolution arithmetic for deep learning / V. Dumoulin, F. Visin // *arXiv preprint arXiv:1603.07285*, 2016.
- [7] A guide to receptive field arithmetic for Convolutional Neural Networks [Электронный ресурс]. – Режим доступа: <https://medium.com/mlreview/a-guide-to-receptive-field-arithmetic-for-convolutional-neural-networks-e0f514068807> (26.11.2019).
- [8] Griffin, G. Caltech-256 object category dataset / G. Griffin, A. Holub, P. Perona // *California Institute of Technology*, 2007.
- [9] Фреймворк MakiFlow [Электронный ресурс]. – Режим доступа: <https://github.com/MakiResearchTeam/MakiFlow> (27.11.2019).
- [10] Simonyan, K. Very deep convolutional networks for large-scale image recognition / K. Simonyan, A. Zisserman // *arXiv preprint arXiv:1409.1556*, 2014.
- [11] Ioffe, S. Batch normalization: Accelerating deep network training by reducing internal covariate shift / S. Ioffe, C. Szegedy // *arXiv preprint arXiv:1502.03167*, 2015.
- [12] He, K. Deep residual learning for image recognition / K. He, X. Zhang, S. Ren, J. Sun // *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. – P. 770-778.
- [13] Kingma, D.P. Adam: A method for stochastic optimization / D.P. Kingma, J. Ba // *arXiv preprint arXiv:1412.6980*, 2014.

Influence of the receptive field size on accuracy and performance of a convolutional neural network

R.R. Gabbasov¹, R.A. Paringer^{1,2}

¹Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

²Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS, Molodogvardejskaya street 151, Samara, Russia, 443001

Abstract. Convolutional neural networks (CNNs) have been successfully applied to many tasks such as digit and object recognition. In this paper we study the size of receptive field of deep convolutional neural networks, in particular, we check the idea of a “redundant” receptive field. We run a set of experiments on two common CNN models — VGG16 and ResNet18 — in order to explore the influence of receptive field size on CNN's training time, accuracy and performance. We run experiments using MakiFlow framework on CALTECH256 dataset. The experiments' results show that optimization of neural networks (NNs) by reducing the size of receptive field allows to reduce the NN's training time by 5-7% while maintaining the accuracy of the network.