РАЗВИТИЕ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ СВЁРТОЧНЫХ НЕЙРОСЕТЕЙ ДЛЯ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ

БЫЧКОВ Александр Григорьевич
аспирант
ФГБОУ ВО «Сибирский государственный индустриальный университет»
г. Новокузнецк, Россия

В работе рассматриваются история развития архитектур свёрточных нейронных сетей и математические методы, используемые для подсчета ее значений. Приведены основные составные части сети, влияющие на результат. Показано сравнение точности распознавания образов при разных архитектурах.
Ключевые слова: свёрточные нейронные сети, распознавание образов, transfer learning, точность работы.

Введение. Машинное обучение (англ. machine learning, ML) – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численные методы, методы оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме. Искусственный интеллект сыграл колоссальную роль в преодолении разрыва между возможностями людей и машин. Как исследователи, так и энтузиасты работают над многочисленными аспектами этой области, добиваясь удивительных результатов. Одним из них является компьютерное зрение. Примером машинного обучения в компьютерном зрении являются свёрточные нейронные сети [1].

В данной статье приведен обзор и сравнительный анализ известных архитектур, их преимущества, история развития и способы использования на практике.

История дальнейшего развития базовой структуры свёрточных сетей. В плане архитектур одной из самых первых была сеть LeNet 1998 г. (рисунок 1).

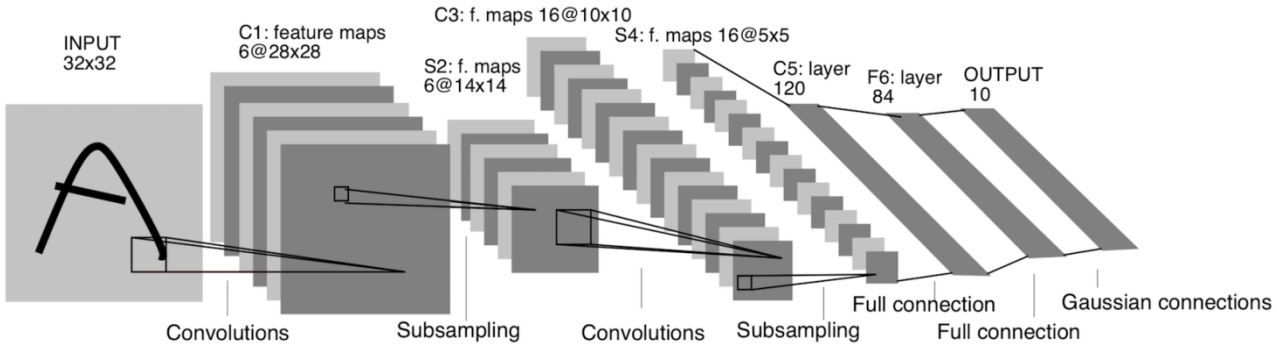

Рисунок 1. **Схема LeNet**

Эта архитектура была придумана Яном Лекуном (оттого и такое название) в 1989 г. как продолжение модели неокогнитрона (neocognitron) [2]. Модель свёрточной сети состоит из трех типов слоев: свёрточные (convolutional) слои, субдискретизирующие (subsampling, подвыборка) слои и слои «обычной» нейронной сети – перцептрона. Первые два типа слоев (convolutional, subsampling), чередуясь между собой, формируют входной вектор признаков для многослойного перцептрона. Сеть можно обучать с помощью градиентных методов.

Изначально Лекун использовал эту систему для распознавания отдельных цифр почтовых индексов. Сейчас подобная архитектура приме-

няется в основном для обучения студентов и пробы сил на датасете MNIST. Все современные сети обычно проверяют на наборе IMAGENET, который содержит 1 000 000 изображений, принадлежащих 1 000 классов [3].

Пунктирной линией показан уровень человеческого восприятия: около 5% неверно распознанных изображений. Как видно из графика, современные архитектуры свёрточных нейросетей намного превышают человеческий глаз. Стоит отметить, что эти 5% показаны весьма условно, так как для выявления такой оценки один сотрудник компании ImageNet обучался месяц и показал в финальном тестировании примерно такой результат [4].

Серьезный прорыв произошел с появлением архитектуры AlexNet в 2012 г.

Данная сеть имела примерно 60 000 000 параметров. Именно на этой архитектуре впервые производилось обучение на GPU (видеокартах). В ходе разработки возникало множество ошибок, связанных с тем, что в те времена видеокарты не были подготовлены к таким вычислениям. Собственно, именно из-за этого пришлось дробить сеть на две части, что видно на рисунке 8. Именно сеть AlexNet стала поворотной точкой, после которой начал проявляться активный интерес к Deep Learning [5]. Эта сеть была разработана А. Крыжевским совместно с И. Суцкевером и Дж. Хинтоном. Необходимо отметить, что и до AlexNet были сети на GPU (к примеру, сеть от К. Челлапиллы в 2006 г.). AlexNet на наборе IMAGENET показал ошибку в ~15%.

Еще одним важным моментом в истории свёрточных нейросетей является архитектура VGG от 2014 г. Схема этой сети приведена на рисунке 2.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Рисунок 2. **Описание схемы сети VGG**

Было предложено несколько вариантов VGG, как показано на рисунке 2, которые различаются количеством весов, в некоторых случаях количеством слоев и ядром свертки. Всего у нее было около 140 000 000 параметров. Основной причиной такого пристального внимания служит то, что все слои очень схожи, почти как «кирпичи», что позволяет использовать снова и снова те же блоки в других моделях. При этом архитектура является очень простой в освоении и понимании, что также дает ей большие возможности для модификации под свои нужды [5]. VGG показал ошибку на наборе IMAGENET в ~6%.

Другой важной точкой является ResNet (Residual Connections Network) 2015 г. Одна из основных проблем VGG и ей подобных сетей состояла в следующем. Чем дальше, чем более глубокие сети применяются (с большим количеством слоев), тем большая выразительная емкость у этих сетей. То есть, они начали распознавать все более и более высокоуровневые признаки. Однако выяснилось, что чем больше сеть, например, в 56 слоев, как приведено на рисунке 2, тем хуже она тренируется. Это не было связано напрямую с переобучением (сеть показывает удовлетворительный результат на тренировочном наборе, но плохой результат на валидационном) – такие сети тренируются плохо, – из-за чего установлено, что 56-слойная сеть тренируется хуже, чем 20-слойная, хотя, казалось бы, чем больше выразительная сила, тем лучше должна тренироваться сеть. Проблема состояла в том, что в начале тренировки все слои весов заполняются случайными числами [6]. И если возникает ситуация, что какой-то из слоев не успел натренироваться, то он испортит показатели всем слоям, следующим за ним. Более того, во время обратного прохода от него пойдут некорректные градиенты в дальнейшие слои, что еще сильнее замедлит обучение. И чем больше слоев в сети, тем больше вероятность возникновения такого результата, из-за чего сети с большим количеством слоев показывают себя хуже.

В ходе решения этой проблемы возникла такая идея: необходимо упрощать тренировку. Было предложено не обучать каждый слой с нуля, а все, что пошло на вход слоя, передавать дальше. Единственным условием является возможность скорректировать данные значения. То есть, слой предсказывает не напрямую выход, а его поправку (residual). Отсюда и название ResNet [6]. Слои в данных сетях вычисляют не все изображение на выходе, они отдают не весь выход. Вход «течет» на выход, а слой может его лишь поправить. Если раньше на входе был x, а на выходе F(x), то теперь на входе x, а на выходе F(x) + x.

Оказалось, что такое изменение дает возможность обучать куда более глубокие сети, чем раньше. Обычно берется архитектура VGG, в нее добавляется множество свёрточных слоев и после этого еще добавляются residual connections.

В итоге на практике получилась возможность тренировать сети глубиной до сотни слоев и больше. Была даже попытка обучить сеть глубиной в 1000 слоев, и все равно в итоге сеть смогла обучиться, пусть и эффективность данной модели получилась не слишком высокой [6].

Сравнительный анализ сетей и дополнительные способы повышения их эффективности. На рисунке 3 приведены все упомянутые архитектуры с их подвидами. По горизонтальной шкале показано то, насколько они вычислительно затратны. По вертикали приведен их лучший результат. Размер круга соответствует количеству входных параметров.

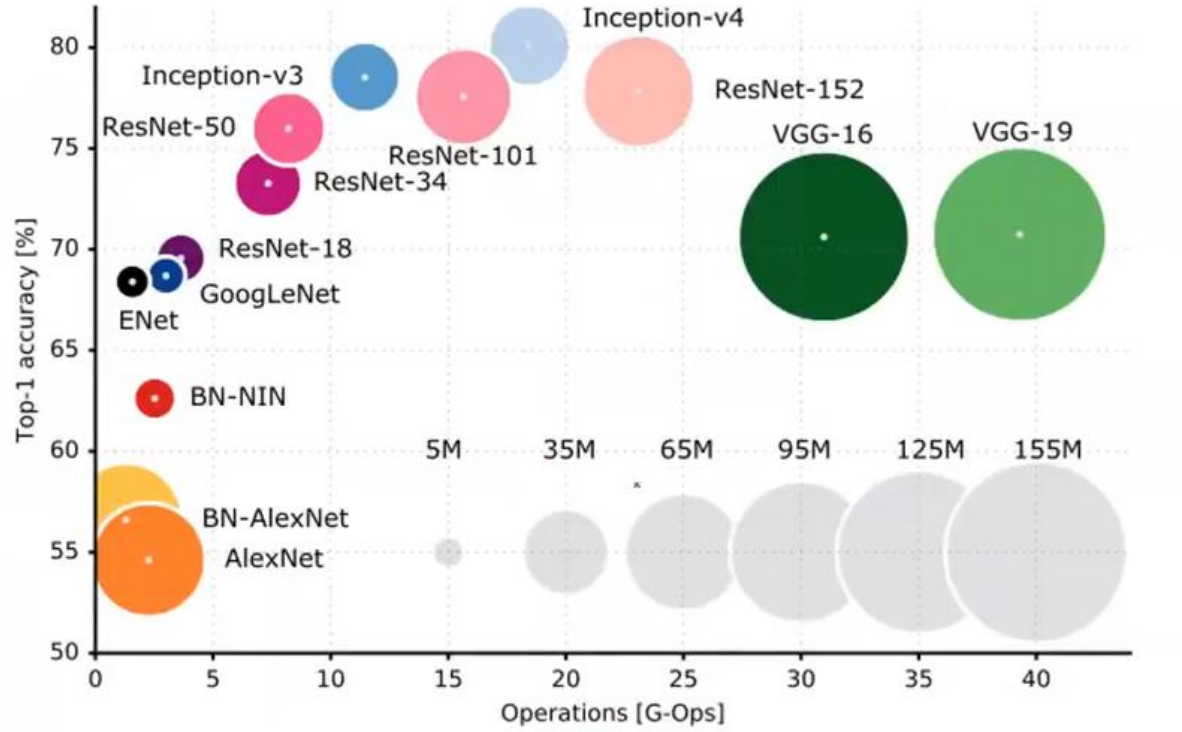

Рисунок 3. **Сравнительный график эффективности различных сетей**

Следующим важным моментом является подготовка изображений для обучения и использование различных полезных приемов. В ходе решения задачи может возникнуть такая ситуация, когда для обучения дано слишком малое число изображений (~10 – 100). Обучать с нуля на таком числе исходных данных невозможно (будет огромное переобучение). Для решения такой проблемы используется Transfer Learning: берется уже натренированная на схожей задаче сеть, все ее слои «замораживаются» (веса делаются неизменными) за исключением последнего. Этот последний слой, на котором и происходит выдача нейроном результата, меняется и обучается на желаемом наборе с использованием уже готовых весов замороженных слоев. То есть до этого сеть училась извлекать признаки из данных, что отражено в замороженных слоях, а теперь сеть должна научиться интерпретировать эти признаки [7; 8].

Следующим важным моментом является подготовка изображений для обучения и использование различных полезных приемов. В ходе решения задачи может возникнуть такая ситуация, когда для обучения дано слишком малое число изображений (~10 – 100). Обучать с нуля на таком числе исходных данных невозможно (будет огромное переобучение). Для решения такой проблемы используется Transfer Learning: берется уже натренированная на схожей задаче сеть, все ее слои «замораживаются» (веса делаются неизменными) за исключением последнего. Этот последний слой, на котором и происходит выдача нейроном результата, меняется и обучается на желаемом наборе с использованием уже готовых весов замороженных слоев. То есть до этого сеть училась извлекать признаки из данных, что отражено в замороженных слоях, а теперь сеть должна научиться интерпретировать эти признаки [7; 8].

Иногда бывают ситуации, когда данных чуть больше (~1000+). В таких случаях можно не замораживать всю сеть. Замораживаются первые слои, а последующие вполне обучаются. Причем, эти слои обучаются с разной скоростью обучения. Скажем, одной из тактик является применять веса, близкие к последнему слою, с коэффициентом 0,1, а веса, лежащие между замороженными слоями и уменьшенными в 0,1 раз, применяются

с коэффициентом 0,01. Точные цифры и соотношение заморозки слоев определяются экспериментально в зависимости от задачи.

Основная причина, почему Transfer Learning и заморозка слоев работает, состоит в том, что по определению свёрточной сети первые слои отвечают за распознавание самых базовых признаков – черточек, точек, и прочих примитивных фигур [7]. Эти примитивы встречаются почти в любом изображении.

Выводы. Таким образом, в данной статье было рассмотрено текущее состояние области задач искусственного интеллекта. Их особенность состоит в том, что как только указанные задачи будут решены, они переходят в разряд задач обычного вычисления.

Показана история развития свёрточных нейросетей, их эволюция. Описаны методы, которые были использованы для повышения эффективности работы сетей, проведен сравнительный анализ сетей разной архитектуры.

Описана концепция Transfer Learning, которая позволяет использовать уже обученные на одной задаче сети для решения других задач.

СПИСОК ЛИТЕРАТУРЫ

1. *Комашинский В.И.* Нейронные сети и их применение в системах управления и связи. – М.: Горячая линия-Телеком, 2002. – 94 с.
2. *Krizhevsky A., Sutskever I., and Hinton G.E.* Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, 2012, P. 1097-1105.
3. *Kumar Chellapilla, Sid Puri, Patrice Simar* High Performance Convolutional Neural Networks for Document Processing». In Lorette, Guy (ed.). Tenth International Workshop on Frontiers in Handwriting Recognition. Suvisoft, 2006.
4. *Iglovikov V., Shvets A.* TernausNet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation, 2018, arXiv: 1801.05746.
5. *Santos J.* A heuristic approach to the multitask-multiprocessor assignment problem using the empty-slots method and rate monotonic scheduling // J. Santos, E. Ferro, J. Orozco, R. Cayssials J. of Real-Time Systems, 1997, Vol. 13 (2). P. 167-199.
6. SegNet: A deep convolutional encoder-decoder architecture for image segmentation // V. Badrinarayanan, A. Kendall, R. Cipolla. – arXiv: 1511.00561, 2015.
7. *Shelhammer L.E., Darrell T.* Fully convolutional networks for semantic segmentation // The IEEE Conf. On Computer Vision and Pattern Recognition (CVPR), 2015, P. 3431- 3440.
8. U-net: convolutional networks for biomedical image segmentation / O. Ronneberger, P. Fischer, T. Brox // Proc. Med. Image Comput. Comput.-Assisted Intevention., 2015, P. 234-241.

DEVELOPMENT AND COMPARATIVE ANALYSIS OF CONVOLUTIONAL NEURAL NETWORKS FOR IMAGE CLASSIFICATION

BYCHKOV Alexander Grigorievich
Postgraduate Student
Siberian State Industrial University
Novokuznetsk, Russia

The paper discusses the history of the development of convolutional neural network architectures and the mathematical methods used to calculate its values. The main components of the network that influence the result are given. A comparison of the accuracy of pattern recognition for different architectures is shown.
Keywords: convolutional neural networks, pattern recognition, transfer learning, performance accuracy.