

Дата поступления рукописи в редакцию: 07.06.2026 г.

DOI: 10.24412/2076-1503-2026-6-936-946

Дата принятия рукописи в печать: 29.06.2026 г.

СЕМЕНОВ Александр Александрович,
научный сотрудник ФКУ НПО «СТУС» МВД России,
e-mail: mail@law-books.ru

ГОНЧАР Владимир Владимирович,
Доцент кафедры информационных технологий и
организации расследования киберпреступлений
Московская академия Следственного комитета
Российской Федерации имени А.Я. Сухарева,
e-mail: vg0778@bk.ru

РОЛЬ ДАТАСЕТОВ И ЭКСПЕРТНОГО ВЫБОРА МЕТОДОВ В ОБУЧЕНИИ НЕЙРОСЕТЕЙ

Аннотация. В работе рассматривается значимость датасета и экспертных знаний в процессе обучения нейросетей. Показано, что качество и структура данных во многом определяют возможность решения задачи, точность модели и характер ошибок. Описываются критерии оценки данных, используемые специалистами по машинному обучению, а также проводится сопоставление процессов обучения нейросетей и человека. Делается акцент на том, что выбор подхода к обучению выходит за рамки настройки алгоритмов оптимизации и требует глубокого понимания природы данных.

Ключевые слова: машинное обучение, нейросети, датасет, обучение моделей, качество данных, компьютерное зрение, экспертные знания, градиентный спуск, анализ данных, обобщающая способность, ошибки модели.

SEменов Aleksandr Aleksandrovich,
Chief Researcher, Federal State Budgetary Institution "STIS"
of the Ministry of Internal Affairs of the Russian Federation

GONCHAR Vladimir Vladimirovich,
Associate Professor,
Department of Information Technology and Cybercrime
Investigation Moscow Academy of the Investigative Committee
of the Russian Federation named after A. Ya. Sukharev

THE ROLE OF DATASETS AND EXPERT CHOICE OF METHODS IN TRAINING NEURAL NETWORKS

Annotation. This paper examines the importance of datasets and expert knowledge in training neural networks. It is shown that the quality and structure of data largely determine the feasibility of solving a problem, the accuracy of a model, and the nature of errors. The paper describes the data evaluation criteria used by machine learning specialists and compares the training processes of neural networks and humans. It emphasizes that the choice of training approach goes beyond tuning optimization algorithms and requires a deep understanding of the nature of the data.

Key words: machine learning, neural networks, datasets, model training, data quality, computer vision, expert knowledge, gradient descent, data analysis, generalization ability, model errors.

Сущность знаний специалиста по машинному обучению заключается в знании того, каким способом, при каких обстоятельствах и какие нейросети лучше обучать. При этом многообразие приемов и методик не ограничивается только модификациями

градиентного спуска. Рассмотрим некоторые другие аспекты.

Как было сказано ранее, ключевую роль при обучении нейросетей играет датасет. Именно он определяет, решается ли задача принципиально или нет. Опытный специалист по машинному обу-

чению по датасету может предсказать, обучится ли нейросеть, с какой точностью она будет работать и даже какие она будет делать ошибки, и все это еще до того, как определены конкретные архитектура нейросети, настройки градиентного спуска и т. д. Попробуем приблизительно сформулировать те критерии, которые он при этом использует (далее в главе будут приводиться примеры для сверточных нейросетей и задач компьютерного зрения, однако это лишь примеры, соображения излагаются в общем для любых типов данных).

Понятно, что нейросеть может проанализировать только ту информацию, которая есть во входных данных. Отсюда следует, что датасет должен состоять из таких единиц данных, каждая из которых содержит в себе всю необходимую для решения задачи информацию. Например, модель автомобиля вполне можно распознать по ее изо-

бражению, соответственно, нейросеть для распознавания модели автомобиля по изображению обучить можно. Обратим внимание на то, что человек почти никогда не обучается на однотипных данных, зачастую они смешаны, и в первом случае процесс занимает гораздо дольше. Например, человек учится говорить по разговорам других людей, и этот процесс занимает долгие месяцы. Если уже умеющий говорить ребенок видит новый для себя объект (например, автобус) и спрашивает у мамы, что это такое, она дает ему некоторую устную информацию, дополнительную к визуальной, что позволяет ему по единственному примеру усвоить знание и впредь его успешно применять. При машинном обучении нейросети нет возможности применить этот же прием, поэтому приходится компенсировать качество обучения количеством.

Как правило, для обучения нейросети некоторому знанию нужно в несколько десятков или сотен раз больше данных, чем для обучения тому же знанию человека.

Если все же обучать нейросеть на небольшом датасете и достичь идеального качества работы нейросети на данных из датасета, может создаться впечатление, что размер датасета оказался достаточен. Почти всегда это ложное впечатление,

в действительности нейросеть не научилась выявлять какие-то признаки или закономерности, а просто «запомнила» все данные из датасета.

Переобучение – явление, заключающееся в том, что при относительно малом и однообразном датасете нейросеть научилась корректно работать на данных из датасета, но при этом плохо работает на любых других данных.

Чтобы убедиться в том, что нейросеть действительно обобщила информацию из датасета, а не просто запомнила ее, на время обучения из

датасета извлекают часть данных, на которых потом позже тестируют нейросеть.

Тестовая /валидационная выборка – часть датасета, которую используют для проверки работоспособности нейросети и не используют во время обучения. В некоторых сложных задачах могут возникать несколько уровней тестирования нейросети, в связи с чем иногда понятиям «тестовая» и «валидационная» выборка могут придавать разное значение.

Обучающая выборка – часть датасета, которую используют во время обучения, но не используют при проверке.

Test/validation – стадия проверки качества работы нейросети.

Train – стадия обучения нейросети, т. е. часть процесса машинного обучения без учета стадии проверки качества.

Хорошим признаком качественного датасета является интуитивно понятное человеку разделение объектов на классы. Например, если человек увидит датасет с изображениями автобусов и самолетов, он легко догадается, что от нейросети требуется отличать объекты этих классов. Если

же это будут автобусы и троллейбусы, то ему потребуется некоторая внимательность, чтобы до этого догадаться. Еще сложнее будет догадаться, какая задача ставится перед нейросетью, если в датасете будут изображения автобусов разных моделей и т. д. Чем понятнее человеку разделе-

ние объектов на классы, тем меньше может быть число единиц данных в датасете.

Например, на рисунке 1 приведены две группы фотографий домов. При этом даже без подсказок сразу бросается в глаза, что слева сгруппированы дома с плоской крышей, а справа – с наклонной. Если в датасете все классы настолько же легко отличимы друг от друга, это верный признак того, что нейросеть обучить будет достаточно просто. В более сложном случае тре-

буется, чтобы человек хотя бы гипотетически мог полностью освоить навык, которому хотят обучить нейросеть, по всему датасету. Например, если требуется диагностировать некоторое заболевание, при этом известно, что на всей планете есть только один врач, который умеет это делать по рентгеновскому снимку, то при достаточном размере обучающего датасета нейросеть также сможет освоить этот навык.



Рисунок 1. Группы фотографий домов.

Если нельзя себе даже теоретически представить человека, который за бесконечное время выучит определенный навык по датасету, то и нейросеть этому навыку обучить не удастся.

Представим себе теперь, что мы хотим обучить нейросеть для распознавания боевых кораблей середины XX в. В реальной жизни моряки распознавали свои и вражеские корабли по характерным признакам: числу палуб, труб, орудийных башен, мачт и т. д., т. е. распознавали зрительно характерные признаки и обращались мысленно в некоторую таблицу в своей памяти. Обращаем

внимание, что это не отменяет возможности помнить каждый корабль зрительно таким, какой он есть. Аналогично, обучить нейросеть для распознавания кораблей можно двумя способами: обучить ее непосредственно распознавать корабли или обучить ее распознавать характерные элементы, после чего подсчитывать их без нейросети и сравнивать с табличными данными (рисунк 2).

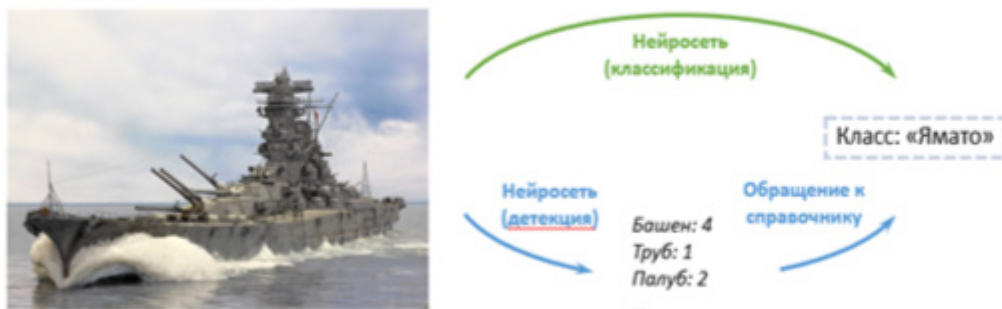


Рисунок 2. Нейросеть для распознавания кораблей.

В первом случае понадобится много примеров изображений для каждого корабля, а во втором случае достаточно, чтобы точными были только табличные данные, обучить нейросеть можно будет на изображениях любых кораблей, обладающих перечисленными элементами. Соот-

ветственно, во втором случае данных может быть меньше, при этом требования к их сбору ниже. Другими словами, чем сложнее постановка задачи непосредственно для нейросети, тем выше требования к датасету. На практике же часто возможности по сбору датасета ограничены.

Желательно ставить перед нейросетью как можно более простую частную задачу, это позволит снизить требования к датасету, использовать имеющийся датасет максимально и, таким образом, максимизировать точность решения задачи.

Представим себе теперь, что мы обучили нейросеть распознавать автобусы и трамваи на фото. Можно догадаться: отличает одних от других она по наличию проводов и рельс. Если такой нейросети передать изображение троллейбуса, то неизвестно, решит ли она, что это автобус, потому что не едет по рельсам, или что это трамвай,

потому что держится за провода (рисунок 3). Зато можно смело предполагать: так как троллейбусы явно обладают всеми теми признаками, которые нейросеть должна выучить для определения автобусов и трамваев, то она их тоже будет по крайней мере относить к одной из двух категорий.

Если практическая задача требует распознавания некоторых классов объектов, то желательно включать в датасет не только эти классы, но также и другие, которые обладают схожими признаками и могут быть легко перепутаны с классами, представляющими интерес, во время инференса.



Рисунок 3. Фотографии автобуса, троллейбуса и трамвая для нейросети.

Важно помнить, что нейросеть также поведет себя почти непредсказуемо на любых данных, которые, наоборот, совсем не похожи ни на какие данные из датасета. Обратим внимание, что человек способен при обучении выделять общее и отбрасывать детали. Например, ребенок, которому впервые показали автобус, легко распознает

его в любых других условиях. Нейросети, напротив, сами по себе такой способностью не обладают. Например, нейросеть, обученная распознавать автобусы на изображениях, может не работать ночью, если в датасете не было ночных фото, не работать зимой, если в датасете не было фото автобусов зимой, и т. д.

Аугментация (данных) – модификация данных датасета с целью повышения качества датасета. Аугментацию не следует путать с использованием искусственно созданных данных. Аугментация сама по себе не может обеспечить качество датасета, однако позволяет обеспечить его разнообразие по некоторым признакам.

В каждой отдельной задаче таких свойств, которые следует учесть в датасете, масса, и человеку достаточно сложно предсказать их все сразу именно потому, что он привык абстрагировать общее от деталей. По этой причине на практике часто возникает ситуация, когда нейросеть, обу-

ченная на большом датасете, не работает из-за того, что в нем не была учтена какая-нибудь, на первый взгляд, незначительная мелочь (например, особенности объектива камеры, угол падения солнечного света и т. д.).

Данные в датасете должны быть разнообразны настолько же, насколько будут разнообразны данные при инференсе нейросети, при этом желательно, чтобы все возможные проявления разнообразия по каждому признаку были представлены в датасете приблизительно в одном и том же количестве.

Например, если от нейросети требуется, чтобы она работала в любое время суток, будет недостаточно добавить в датасет из 10000 дневных фото 10 ночных. Это же будет касаться других возможных особенностей изображений: качества изображений, времени года и т. д.

Представим себе теперь, что мы собираем датасет для распознавания самолетов на видео с камеры, стоящей на земле и поднятой вверх.

У слоев нейросети ориентация задана однозначно, т. е. есть верх и низ, лево и право, хотя самолеты могут лететь под самыми разными углами. Чтобы нейросеть работала во всех случаях, датасет должен содержать варианты расположения самолетов на снимке под разными углами. Однако, в отличие от предыдущих приме-

ров, для обеспечения такого разнообразия в датасете не нужно делать дополнительные снимки, достаточно повернуть уже существующие.

Какими бы качествами ни обладал датасет, всегда найдется изображение, по-своему уникальное и отличающееся от любого другого в датасете, на котором признаки, анализируемые нейросетью, будут выражены лишь частично. Чтобы смоделировать такую ситуацию во время обучения, перед некоторыми из последних слоев (как правило, полносвязных) на каждом шаге обучения случайным образом обнуляют определенную долю значений карты признаков, в результате чего нейросеть вынуждена учиться использовать запасные признаки и становится более устойчивой.

Дропаут /dropout – слой нейросети, включаемый только на этапе обучения, выполняющий случайное обнуление части значений карты признаков и предназначенный для увеличения устойчивости нейросети к незнакомым данным.

Представим себе теперь, что нам нужно обучить некоторую нейросеть, заведомо работающую только с яркими изображениями. В цифровом RGB представлении белый цвет кодируется значениями [255, 255, 255], соответственно, от ярких изображений можно ожидать, что все значения цветности во всех пикселях для них будут большими, например, больше 200. Тогда каждый вес w первого сверточного слоя нейросети будет во время вычислений умножаться на значения в диапазоне от 200 до 255. Эту ситуацию можно сравнить с такой, в которой от вас бы потребовалось взвесить килограмм яблок, но при условии, что к каждому при взвешивании привязана килограммовая гиря. Понятно, что было бы проще эти гири просто отвязать. Также и в данном случае, нейросети будет трудно учиться из-за того, что возника-

ющие при вычислении сверточного слоя слагаемые для разных данных почти одинаковы, так как содержат одинаковый для них всех «мусор». Для решения этой проблемы поступают так: подсчитывают средние по датасету значения (в англоязычной литературе их называют mean) для каждого из трех цветов, а перед подачей в нейросеть (как при обучении, так и в инференсе) вычитают из каждого входного значения соответствующее среднее. Так, если представить, что в приведенном примере средние значения цветности будут равны [233, 235, 230], то диапазоны значений, передаваемых в нейросеть, будут для трех цветов соответственно от -33 до 22, от -35 до 20 и от -30 до 25. В этом случае нейросети учиться гораздо проще. Такого рода прием называют нормализацией.

Нормализация данных – приведение данных к некоторой шкале, в которой их удобнее использовать.

Нормализацию можно использовать не только применительно к входным данным, но также и к промежуточным значениям, вычисляемым в нейросети. Применение нормализации убирает случайные скачки промежуточных значений между слоями нейросети, другими словами, входные значения для каждого слоя становятся как бы более предсказуемыми, что позволяет каждому слою учиться самостоятельно, а не переучиваться полностью каждый раз, когда прогресс обучения предыдущего слоя полностью изменил диапазон выходных значений. Так, наиболее распространенной нормализацией, применяемой в сверточных нейросетях, является батч-нормализация [5]. Ее реализуют в виде самостоятель-

ного типа слоя с двумя обучаемыми весами, который располагается до или после каждого сверточного слоя. Особенность тут заключается в том, что нормализацию выполняют не по всему датасету или одному изображению, а по батчу или части батча на каждом шаге градиентного спуска. В более простом случае нормализацию можно выполнять просто среди всех однотипных значений. Например, простейшая нормализация значений карты признаков в сверточной нейросети может заключаться в приведении значений в каждом канале независимо к диапазону от -1 до 1. Псевдокод, реализующий вычисления прямого хода такого слоя нормализации, приведен на рисунке 4.

```
for (l=0;l<L;l++) {
  s = 0.0;
  min = 9999999999.0;
  max = -9999999999.0;
  for (y=0;y<Y;y++)
    for (x=0;x<X;x++) {
      if (IN[y][x][l] > max)
        IN[y][x][l] = max;
      if (IN[y][x][l] < min)
        IN[y][x][l] = min;
      s = s + IN[y][x][l];
    }
  s = s / (X*Y);
  for (y=0;y<Y;y++)
    for (x=0;x<X;x++)
      OUT[y][x][l] = (IN[y][x][l] - s) / (0.5*(max - min));
}
```

Рисунок 4. Псевдокод, реализующий вычисления прямого хода слоя нормализации.

Представим себе теперь начало процесса обучения со всеми перечисленными выше приемами. Создадим необученную, пустую нейросеть, заполнив все значения ее весов нулями. Легко заметить, что в сверточной нейросети, созданной таким образом, при первом прямом ходе на всех внутренних слоях будут получаться только нулевые результаты, в том числе нулю будет равен градиент лосса, в результате чего процесс обучения вовсе не стартует. Таким образом, начальные значения весов нужно задать хоть какими-то, чтобы первый посчитанный градиент не был нуле-

вым. Мы ожидаем, что в процессе обучения разные веса сверточных слоев обретут разные значения и будут выражать разные признаки.

Однако если в начальный момент времени значения двух фильтров одного сверточного слоя заданы одинаково, то соответствующие значения градиентов, приходящиеся на эти фильтры, будут одинаковыми, процесс обучения для двух фильтров будет происходить одинаково, в результате они попросту будут дублировать друг друга. Чтобы этого не произошло, разные фильтры одного слоя нужно инициализировать разными значениями.

Инициализация нейросети – придание весам нейросети начальных значений перед машинным обучением.

Инициализация нейросети определяет начальный толчок, который получит процесс градиентного спуска, ведь она определяет точку, где начинается градиентный спуск и, следовательно, может определить конечную точку, куда этот градиентный спуск придет. Поэтому существует масса разных подходов, в том числе очень нетривиальных, к инициализации нейросетей. Однако на практике значения весов нейросетей чаще всего инициализируются попросту случайным образом в определенном диапазоне.

Вспомним теперь, что в сверточной нейросети каждый сверточный слой извлекает признаки определенного уровня. На первом сверточном слое это первичные признаки, которые могут быть вполне понятные человеку (как на рис. 5.5) и, следовательно, для которых человек может даже сам придумать начальные значения (так, кстати, сделали разработчики AlexNet). Представим себе теперь, что мы обучили нейросеть для распознавания породы кошек и нам нужна нейросеть для распознавания породы рысей.

Эмбэддинг – термин, которым называют вектор числовых значений. Формально под этот термин подходит почти все, что угодно, на практике же обычно этим термином пользуются, когда хотят подчеркнуть какое-то особое значение, ассоциацию вектора с какими-то другими данными, передачу вектора из одной сущности в другую и т. д.

Можно предположить, что первичные признаки у двух нейросетей будут очень похожи, поэтому несколько первых слоев второй нейросети можно инициализировать попросту значениями весов уже обученной первой нейросети, если первая нейросеть уже хорошо работает. При этом, как мы замечали ранее, чем ближе слой нейросети к концу, тем охотнее его веса меняются при градиентном спуске, т. е. быстрее всех учатся слои, содержащие признаки более высокого порядка. Отсюда следует, что даже если мы проинициали-

зируем веса всех слоев второй нейросети значениями первой нейросети, то мы не собьем процесс обучения с верного пути, так как чем меньше признаки первой нейросети подходят для задачи второй нейросети, тем вероятно ближе они находятся к концу нейросети и тем быстрее они будут меняться. Другими словами, для получения второй нейросети мы можем попросту взять первую нейросеть и продолжить учить ее на новом для нее датасете.

Файн-тюнинг /fine-tuning – процесс, при котором нейросеть, обученную на одном датасете, для решения конкретной задачи продолжают учить на совершенно другом датасете. Иногда процесс обучения на новом датасете называют «дообучением».

Файн-тюнинг часто применяется на практике, чаще всего берут нейросеть, обученную на гигантском датасете с разнообразными данными (например, из миллионов изображений), и дообучают ее на значительно меньшем датасете, который относится к конкретной прикладной задаче.

При этом в некоторых случаях (чаще всего – для экономии времени) несколько начальных слоев нейросети воспринимают как фиксированную математическую операцию, а обучение выполняют только для нескольких последних слоев.

Замораживание весов – прием обучения, при котором определенные веса фиксируются и воспринимаются не как обучаемые веса, а как постоянные значения. Фактически для замораживания весов достаточно просто не изменять их во время градиентного спуска.

Файн-тюнинг и замораживание весов являются частными примерами более общего приема: вся нейросеть или ее часть могут быть использо-

ваны в качестве части другой нейросети, при подготовке обучающих данных, в качестве математической операции в составе лосса и т. д.

Например, распространено использование конструкции из двух нейросетей, одной из них ставят задачу преобразовать входные данные в некоторое числовое представление (грубо говоря,

«закодировать» входные данные), а другой – использовать это числовое представление для генерации выходного результата (грубо говоря, «расшифровать»).

Например, в задаче идентификации человека по лицу эмбэddингом называют набор числовых значений, выдаваемый нейросетью и сверяемый с базой данных лиц.

Энкодер – часть нейросети или нейросеть, выполняющая преобразование входных данных в некоторый эмбэddинг.

Декодер – часть нейросети или нейросеть, выполняющая преобразование некоторых эмбэddингов в выходные данные.

Автоэнкодер – нейросеть, состоящая из последовательности энкодера и декодера, которой ставится задача воспроизвести на выходе те же данные, что были ею получены на входе.

Например, нейросеть, решающую задачу сегментации, можно рассматривать как комбинацию энкодера и декодера, но не как автоэнкодер. Сами по себе эти термины не несут практической пользы, но удобны для пояснения устройства и

порядка обучения нейросетей. Рассмотрим пример. Положим, мы хотим создать нейросеть, которая будет обрабатывать фотографию и преобразовывать ее так, чтобы из нее получился рисунок известного художника (*рисунок 5*).



Рисунок 5. Преобразование фотографии нейросетью.

Такую нейросеть можно обучить, например, следующим путем (*рисунок 6*).

- 1) обучим нейросеть, умеющую распознавать художника, нарисовавшего картину;
- 2) «отломаем» у нейросети последний слой, заморозим веса и используем ее в качестве энкодера (при этом получится, что выдаваемые нейросетью эмбэddинги кодируют стиль художника), добавим к ней еще один энкодер, выходы обеих нейросетей передадим на вход некоторому декодеру и получившуюся конструкцию обучим как автоэнкодер;

- 3) во время инференса будет подставляться пользовательское изображение на вход второму энкодеру, а на вход первому будет подставляться картина известного художника.

В описанной комбинации наибольшую сложность представляет обучение автоэнкодера на втором шаге. Один из популярных способов эффективного обучения автоэнкодеров заключается в одновременном обучении двух нейросетей с противоположными целями, использующими общий лосс.

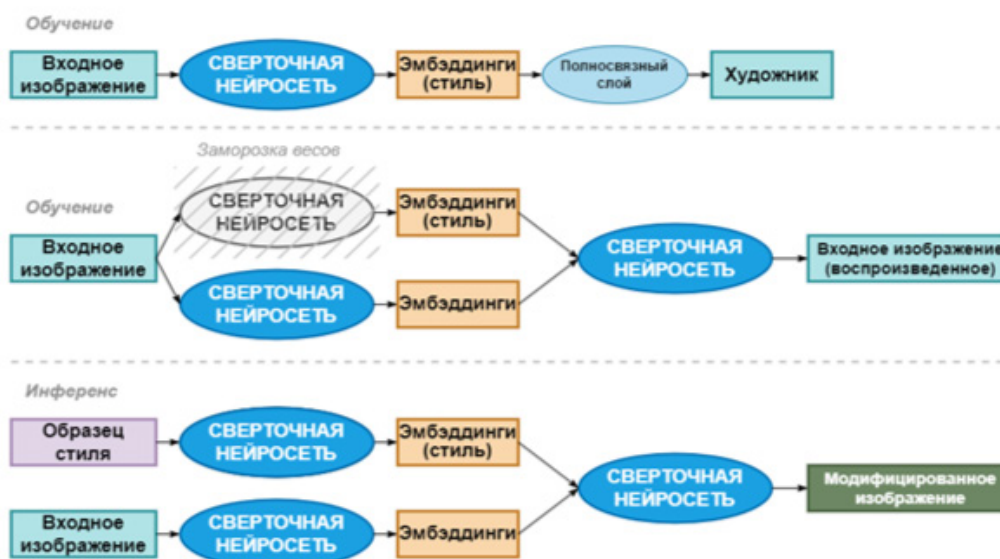


Рисунок 6. Обучение нейросети.

GAN /Generative adversarial networks/ГАНы/состязующиеся нейросети – пара нейросетей – генератор и дискриминатор, – и подход к их обучению, при котором эти нейросети обучаются совместно, при этом задача генератора – генерировать данные, а дискриминатора – определять, являются ли данные сгенерированными генератором или нет. Нейросети используют один и тот же лосс, оценивающий, удалось ли генератору обмануть дискриминатор, с разным знаком. Генератор стремится «обмануть» дискриминатор, т. е. создать набор данных, не похожий на произведенный нейросетью, а дискриминатор, наоборот, стремится научиться отличать произведенные генератором данные от любых других.

В информационных источниках часто в контексте ГАНов упоминают классический раздел математики, называемый математическими играми, и сам процесс обучения ГАНов называют минимаксной игрой. Эта отсылка справедлива, однако с практической точки зрения она совершенно бесполезна.

Методика обучения GAN широко применяется для обучения генеративных нейросетей, при этом чаще всего полезным результатом обучения является только генератор, дискриминатор может быть отброшен. Так, в области компьютерного зрения с помощью ГАНов успешно решаются задачи улучшения качества изображений, переноса части свойств изображения на другое изображение и т. д.

В сложных случаях, когда требуется решение с помощью нейросетей задачи, подразумевающей анализ и генерацию данных разных категорий, например, совместный анализ текстов и картинок (обычно говорят «данные разных модальностей» или «мультимодальные данные»), чаще такую систему выстраивают по частям, т. е. составляют ее из нескольких нейросетей, обучаемых отдельно, обеспечивая стыковку между форматами используемых эмбеддингов, лоссов, разметок, и т. д.

Список литературы:

- [1] Chen Zhang, Joohee Kim Video object detection with two-path convolutional LSTM pyramid. 2016. – URL: https://www.researchgate.net/publication/343702579_Video_Object_Detection_With_Two-Path_Convolutional_LSTM_Pyramid/fulltext/5f3b265892851cd302013482/Video-Object-Detection-With-Two-Path-Convolutional-LSTM-Pyramid.pdf (дата обращения: 16.01.2026).
- [2] Hanson A., Koutilya PNVR, Sanjukta Krishnagopal, Larry Davis Bidirectional Convolutional LSTM for the Detection of Violence in Videos. 2018. – URL: https://openaccess.thecvf.com/content_eccv_2018_workshops/w10/html/Hanson_Bidirectional_Convolutional_LSTM_or_the_Detection_of_Violence_in_Videos_ECCVW_2018_paper.html (дата обращения: 15.01.2026).
- [3] Howard A.G., Menglong Zhu, Bo Chen, Kalenichenko D. MobileNets: Efficient Convolutional

- Neural Networks for Mobile Vision Applications. 2017. – URL: <https://arxiv.org/pdf/1704.04861/> (дата обращения: 11.01.2026).
- [4] Iandola F.N., Han S., Moskewicz M.W., Ashraf K., Dally W.J., Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size Recognition. 2016. Published as a conference paper at ICLR. 2015. – URL: https://www.researchgate.net/publication/301878495_SqueezeNet_AlexNet-level_accuracy_with_50x_fewer_parameters_and_05MB_model_size (дата обращения: 10.01.2026).
- [5] Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015. – URL: <https://arxiv.org/pdf/1502.03167/> (дата обращения: 13.01.2026).
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Deep Residual Learning for Image Recognition. 2015. – URL: <https://arxiv.org/pdf/1512.03385> (дата обращения: 11.01.2026).
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Spatial pyramid pooling in deep convolutional networks for visual recognition. 2015. – URL: <https://arxiv.org/pdf/1406.4729/> (дата обращения: 11.01.2026).
- [8] Krizhevsky I., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. 2012. – URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d-6b76c8436e-924a68c45b-Paper.pdf (дата обращения: 12.01.2026).
- [9] Lavin A., Scott Gray Fast Algorithms for Convolutional Neural Networks. 2015. – URL: <https://arxiv.org/abs/1509.09308> (дата обращения: 11.01.2026).
- [10] Mikolov T., Kai Chen, Corrado G., Jeffrey Dean Efficient Estimation of Word Representations in Vector Space. 2013. – URL: <https://arxiv.org/abs/1301.3781> (дата обращения: 15.01.2026).
- [11] Norman P. Jouppi, Cliff Young, Nishant Patil and others In-Datcenter Performance Analysis of a Tensor Processing Unit. 2017. – URL: <https://arxiv.org/pdf/1704.04760> (дата обращения: 11.01.2026).
- [12] Ronneberger O., Fischer Ph., Brox Th. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015. – URL: <https://arxiv.org/pdf/1505.04597/> (дата обращения: 13.01.2026).
- [13] Szegedy C., Wei Liu, Chapel Hill, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich Going Deeper with Convolutions. 2014. – URL: <https://arxiv.org/pdf/1409.4842> (дата обращения: 12.01.2026).
- [14] Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2015. Published as a conference paper at ICLR. 2015. – URL: <https://arxiv.org/pdf/1409.1556> (дата обращения: 10.01.2026).
- [15] Wei Liu, Anguelov D., Erhan D., Szegedy C., Reed S., Cheng-Yang Fu, Berg A.C. SSD: Single Shot MultiBox Detector. 2016. – URL: <https://arxiv.org/pdf/1512.02325> (дата обращения: 11.01.2026).
- [16] Официальный сайт ONNX. – URL: <https://onnx.ai/>.
- [17] Основы технологий искусственного интеллекта: учебное пособие/ А.А. Семенов, В.В. Гончар, А.С. Эрдниев. – Московский университет МВД России имени В.Я. Кикотя, 2025. – 190с.
- [18] Финансовый мониторинг: учебное пособие / В.И. Глотов, А.У. Альбеков, О.Н. Тисен и др. – КноРус, 2022. – 198 с.
- [19] Тисен О.Н. Противодействие современным приемам и способам вербовки в международные террористические организации // Российская юстиция. 2018, №9. С. 51-54.
- [20] Овчинский А.С., Аветисян К.Р. Аудиальные воздействия на участников массовых мероприятий как объект административно-правового регулирования// Вестник Московского университета МВД. 2015. № 1-. С. 293-297.
- [21] Зиборов О.В., Аветисян К.Р. Комплексный подход при противоборстве организации массовых беспорядков// Вестник Московского университета МВД. 2023. № 7. С. 100-104.

References:

- [1] Chen Zhang, Joohee Kim Video object detection with two-path convolutional LSTM pyramid. 2016. – URL: https://www.researchgate.net/publication/343702579_Video_Object_Detection_With_Two-Path_Convolutional_LSTM_Pyramid/fulltext/5f3b2658928-51cd302013482/Video-Object-Detection-With-Two-Path-Convolutional-LSTM-Pyramid.pdf (дата обращения: 16.01.2026).
- [2] Hanson A., Koutilya PNV, Sanjukta Krishnagopal, Larry Davis Bidirectional Convolutional LSTM for the Detection of Violence in Videos. 2018. – URL: https://openaccess.thecvf.com/content_eccv_2018_workshops/w10/html/Hanson_Bidirectional_Convolutional_LSTM_or_the_Detection_of_Violence_in_Videos_ECCVW_2018_paper.html (дата обращения: 15.01.2026).
- [3] Howard A.G., Menglong Zhu, Bo Chen, Kalenichenko D. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017. – URL: <https://arxiv.org/pdf/1704.04861/> (access date: 11.01.2026).
- [4] Iandola F.N., Han S., Moskewicz M.W., Ashraf K., Dally W.J., Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size Recognition. 2016. Published as a conference paper at ICLR. 2015. – URL: https://www.researchgate.net/publication/301878495_

SqueezeNet_AlexNet-level_accuracy_ with_50x_fewer_parameters_and_05MB_model_size (дата обращения: 10.01.2026).

[5] Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015. – URL: <https://arxiv.org/pdf/1502.03167/> (access date: 13.01.2026).

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Deep Residual Learning for Image Recognition. 2015. – URL: <https://arxiv.org/pdf/1512.03385> (access date: 11.01.2026).

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Spatial pyramid pooling in deep convolutional networks for visual recognition. 2015. - URL: <https://arxiv.org/pdf/1406.4729/> (access date: 11.01.2026).

[8] Krizhevsky I., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. 2012. - URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d-6b76c8436e_924a68c45b-Paper.pdf (accessed on: 12.01.2026).

[9] Lavin A., Scott Gray Fast Algorithms for Convolutional Neural Networks. 2015. - URL: <https://arxiv.org/abs/1509.09308> (access date: 11.01.2026).

[10] Mikolov T., Kai Chen, Corrado G., Jeffrey Dean Efficient Estimation of Word Representations in Vector Space. 2013. – URL: <https://arxiv.org/abs/1301.3781> (access date: 15.01.2026).

[11] Norman P. Jouppi, Cliff Young, Nishant Patil and others In-Datacenter Performance Analysis of a Tensor Processing Unit. 2017. – URL: <https://arxiv.org/pdf/1704.04760> (access date: 11.01.2026).

[12] Ronneberger O., Fischer Ph., Brox Th. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015. – URL: <https://arxiv.org/pdf/1505.04597/> (access date: 13.01.2026).

[13] Szegedy C., Wei Liu, Chapel Hill, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich Going Deeper with Convolutions. 2014. – URL: <https://arxiv.org/pdf/1409.4842> (accessed on: 12.01.2026)

[14] Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2015. Published as a conference paper at ICLR 2015. - URL: <https://arxiv.org/pdf/1409.1556> (access date: 10.01.2026).

[15] Wei Liu, Anguelov D., Erhan D., Szegedy C., Reed S., Cheng-Yang Fu, Berg A.C. SSD: Single Shot MultiBox Detector. 2016. – URL: <https://arxiv.org/pdf/1512.02325> (access date: 11.01.2026).

[16] ONNX official website. – URL: <https://onnx.ai/>.

[17] Fundamentals of artificial intelligence technologies: a textbook/A.A. Semenov, V.V. Gonchar, A.S. Erdniev. - Moscow University of the Ministry of Internal Affairs of Russia named after V.Ya. Kikotya, 2025. - 190s.

[18] Financial monitoring: training manual/V.I. Glotov, A.U. Albekov, O.N. Tisen et al. - KnoRus, 2022. - 198 s.

[19] Tisen O.N. Countering modern methods and methods of recruitment to international terrorist organizations//Russian Justice. 2018, №9. S. 51-54.

[20] Ovchinsky A.S., Avetisyan K.R. Audio effects on participants of mass events as an object of administrative and legal regulation//Bulletin of the Moscow University of the Ministry of Internal Affairs. 2015. № 1-. S. 293-297.

[21] Ziborov O.V., Avetisyan K.R. An integrated approach to confronting the organization of mass riots//Bulletin of the Moscow University of the Ministry of Internal Affairs. 2023. № 7. S. 100-104.





ЮРКОМПАНИ
www.law-books.ru

Юридическое издательство «ЮРКОМПАНИ»
издает научные журналы:

- Научно-правовой журнал «Образование и право», рекомендованный ВАК Министерства науки и высшего образования России (специальности 12.00.01, 12.00.02), выходит 1 раз в месяц.
- Научно-правовой журнал «Право и жизнь», рецензируемый (РИНЦ, E-Library), выходит 1 раз в 3 месяца.