УДК 004.021 DOI: 10.18698/2541-8009-2018-10-380

ИССЛЕДОВАНИЕ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНОЙ СЕТИ

А.Э. Айрапетов jk3000@yandex.ru
SPIN-код: 9821-1302

А.А. Коваленко annak0v@yandex.ru
SPIN-код: 9763-0772

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Алгоритмы генеративно-состязательных нейронных сетей являются сравнительно молодыми и одними из наиболее перспективных в семействе алгоритмов машинного обучения без учителя. В настоящее время такие нейронные сети применяются для генерации фотореалистичных изображений на основе предоставленных ей примеров. Однако потенциал данной подгруппы алгоритмов раскрыт не полностью и позволяет увеличить правдоподобность генерируемых изображений. В работе описаны особенности структуры генеративно-состязательной нейронной сети, выполнен анализ ее точности и производительности, а также приведены решения, позволяющие оптимизировать существующие алгоритм обучения и топологию нейронной сети.

Ключевые слова

Генеративно-состязательная сеть, машинное обучение, нейронные сети, обучение без учителя, генератор, дискриминатор, перцептрон, автокодировщик

Поступила в редакцию 19.09.2018

Введение. Генеративно-состязательная сеть — алгоритм машинного обучения, работа которого строится на основе двух «соперничающих» нейронных сетей. Принцип алгоритма заключается в том, что одна из этих сетей *G*, называемая генератором, пытается сгенерировать определенные образцы (например, изображения, видео или любые другие данные, на генерацию которых она запрограммирована), а другая сеть *D*, называемая дискриминатором, старается решить, является ли представленный ей образец настоящим или сгенерированным. Задачей генератора *G* является производство таких образцов, которые дискриминатор *D* сочтет настоящими, в то время как задача дискриминатора противоположна — он должен отбраковать сгенерированные образцы [1]. Таким образом, между генератором и дискриминатором возникает антагонистическая игра, в результате которой обе нейронные сети обучаются без учителя: генератор в итоге обучается генерировать образцы, по правдоподобности конкурирующие с настоящими, а дискриминатор — качественно выполнять проверку этих образцов [2].

Структура классической генеративно-состязательной сети на примере изображений в качестве образцов представлена на рис. 1.

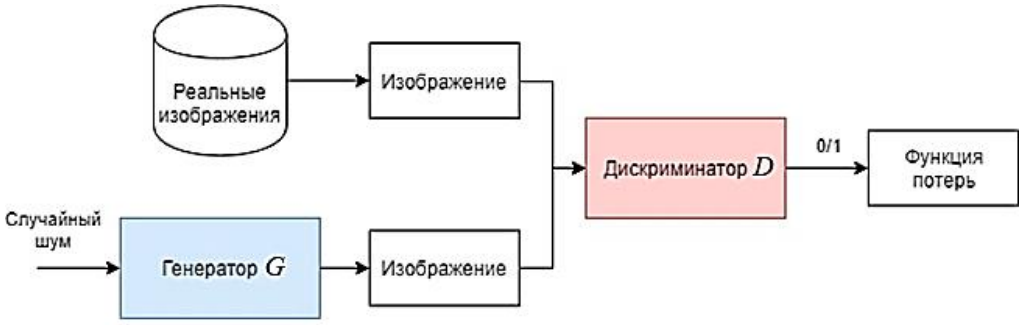

Рис. 1. Классическая генеративно-состязательная сеть

Здесь генератор производит изображения на основе случайного шума, форма которого регулируется в процессе обучения, а дискриминатор, получая на входе изображения, на выходе должен вывести информацию о том, настоящее изображение или нет (1 — настоящее, 0 — поддельное) [3].

Анализ алгоритма обучения. Согласно схеме, приведенной на рис. 1, дискриминатор D должен быть предварительно обучен с помощью реальных изображений, которым присвоена метка 1 («настоящее»). Второй этап работы алгоритма заключается в создании искусственных изображений генератором G на основе случайного шума, форма которого должна меняться в процессе обучения.

Цель генератора G заключается в генерации таких изображений, чтобы дискриминатор как можно хуже их различил (принял за настоящие), и может быть описана следующим выражением:

$$\min_G \log\{1 - D[G(z)]\}, \qquad (*)$$

где D — функция дискриминатора; G — функция генератора; z — случайный шум.

Цель дискриминатора D, математически описанная с помощью следующего выражения, противоположна — он должен отличить подделку с как можно более высокой вероятностью:

$$\max_D \{\log D(x_t) + \log[1 - D(x_f)]\},$$

где x_t — настоящее изображение; $x_f = G(z)$ — сгенерированное изображение.

Таким образом, алгоритм представляет собой минимаксную игру для двух нейронных сетей [4]:

$$\min_G \max_D \{\log D(x_t) + \log[1 - D(G(z))]\}.$$

В классическом алгоритме информация от генератора к дискриминатору передается путем прямого прохода по топологии, а от дискриминатора к генератору — путем обратного. Данный алгоритм можно усовершенствовать, подавая результат работы дискриминатора напрямую на вход генератора, как представлено на рис. 2 [5].

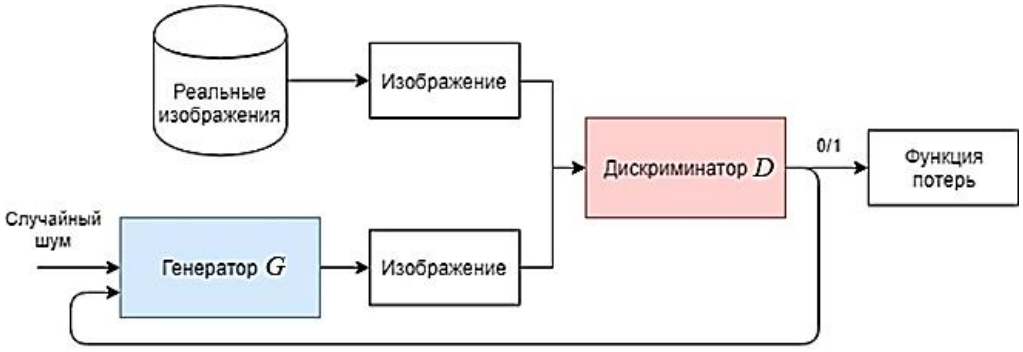

Рис. 2. Генеративно-состязательная сеть с прямым проходом

Однако при таком подходе возникает проблема на начальных этапах обучения модели — дискриминатор, обученный на реальных изображениях, будет с легкостью отличать сгенерированные изображения, в результате чего его функция $D(G(z))$, а соответственно и функция логарифма $\log\{1 - D[G(z)]\}$, будут колебаться рядом с нулевым значением, в результате чего генератор не будет получать никакой полезной информации и не будет обучаться. Данную проблему можно решить, выразив цель генератора из выражения (*), заменив поиск минимума поиском максимума [6]:

$$\max_G \log[D(G(z))].$$

Данное решение порождает еще одно условие — на любом шаге обучения дискриминатор должен показывать точность, лежащую в некоторых оптимальных положительных границах. Этого можно добиться, применив следующий алгоритм обучения:

1) осуществление нескольких итераций обучения дискриминатора на выборке из реальных и сгенерированных изображений при фиксированном генераторе;

2) осуществление одной итерации обучения генератора на сгенерированном изображении и реакции фиксированного дискриминатора на него.

Этот подход позволит держать точность дискриминатора на оптимальном для данного шага обучения уровне и постепенно обучать генератор [7]. Особенностью подхода является то, что генератор учится создавать похожие на настоящие изображения, ни разу не увидев настоящего изображения, и обучается только на основе информации, получаемой от дискриминатора.

Анализ структуры генеративно-состязательной сети. Главным недостатком описанной генеративно-состязательной сети является то, что данная модель не умеет извлекать признаки из получаемых настоящих изображений. Это способствует тому, что при переносе образца, который учится создавать генератор, из одного домена в другой (например, при изменении фона изображения) система не распознает, что изменен был лишь контекст, и примет это за другой образец [8]. К примеру, такой недостаток делает данную структуру уязвимой к шуму на фотографиях.

Описанный недостаток можно устранить, подойдя к основной идее генеративно-состязательной сети с другой стороны и представив топологию, как показано на рис. 3 [9].

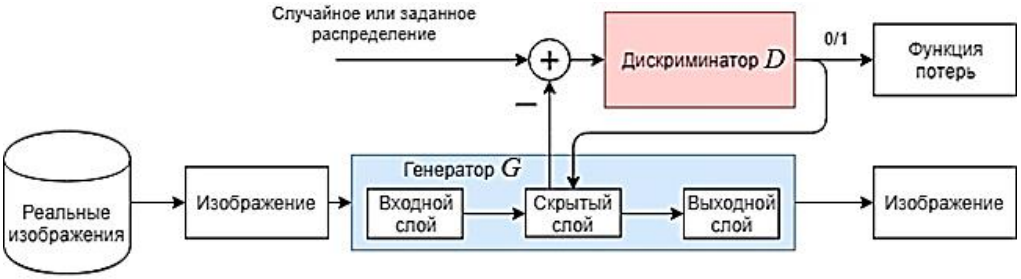

Рис. 3. Состязательный автокодировщик

В данной структуре генератор по сути является автокодировщиком. Данная структура нейронной сети представляет собой обычный перцептрон с одним скрытым слоем с той лишь поправкой, что выходной слой имеет столько же нейронов, сколько и входной, а размерность скрытого слоя, наоборот, ограничена. Принцип работы автокодировщика заключается в том, что выходной результат такой нейронной сети должен быть близок к исходным подаваемым данным, а из-за вычислительного ограничения скрытого слоя в ходе своих вычислений нейронная сеть вынуждена искать обобщения поступающих на вход данных и выполнять их сжатие [9].

Теперь настоящие изображения поступают на вход генератора, скрытый слой которого подключен к входу дискриминатора. Также на вход дискриминатора подается некоторое распределение шума, которое может быть как заданным аналитически, так и случайным. Задачей дискриминатора в данном случае является различение изображений, генерируемых кодером (входным слоем генератора), и изображений из распределения, а также подстраивание параметров кодера под заданное случайное или аналитически описанное распределение [10]. Такая структура имеет две функции потерь — одна для качества данных (соответствия выходных данных автокодировщика входным), а вторая — для генеративно-состязательной сети. Процесс обучения в данном случае будет выглядеть следующим образом:

1) итерация обучения автокодировщика, в ходе которой обновляются параметры генератора для того, чтобы выход сети соответствовал входу;

2) итерации обучения генеративно-состязательной сети, описанные выше:

– несколько итераций обучения дискриминатора на основе настоящих и сгенерированных изображений при фиксированном генераторе;

– одна итерация обучения генератора (корректировка скрытого слоя автокодировщика) при фиксированном дискриминаторе.

В результате такого обучения представленная сеть будет проецировать образы из набора настоящих изображений на заданное распределение, а также для

любой величины из заданного распределения находить адекватный образ, основанный на предоставленном наборе изображений.

Выводы. Предложенная структура позволяет выделять именно признаки (например, стиль написания и вид буквы для текста), присущие предоставленным настоящим образцам, и на их основе генерировать искусственные образцы, похожие на настоящие. Данный подход можно усовершенствовать, применив, к примеру, более сложную топологию для выделения нескольких разных признаков изображений, что позволит генерировать искусственные изображения, варьируя выделенные признаки и создавая различные их комбинации для определенных образов.

Литература

[1] GAN — генеративные состязательные сети.
URL: http://robocraft.ru/blog/machinelearning/3693.html (дата обращения 10.05.2018).
[2] GAN: a beginner's guide to generative adversarial networks.
URL: https://deeplearning4j.org/generative-adversarial-network (дата обращения 15.05.2018).
[3] Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair Sh., Courville A., Bengio Yo. *Generative adversarial networks*. Universite de Montreal, 2014, 9 p.
[4] Generative adversarial networks. URL: https://habr.com/post/352794 (дата обращения 17.05.2018).
[5] Ganin Ya., Lempitsky V. Unsupervised domain adaptation by backpropagation.
URL: https://arxiv.org/pdf/1409.7495.pdf (дата обращения 15.05.2018).
[6] Gitman I. Adversarial networks.
URL: http://www.machinelearning.ru/wiki/images/4/4f/Gan.pdf (дата обращения 20.05.2018).
[7] How to train a GAN? Tips and tricks to make GANs work.
URL: https://github.com/soumith/ganhacks (дата обращения 21.05.2018).
[8] Mirza M., Osindero S. Conditional generative adversarial nets.
URL: https://arxiv.org/pdf/1411.1784.pdf (дата обращения 15.05.2018).
[9] A beginner's guide to deep autoencoders.
URL: https://deeplearning4j.org/deepautoencoder (дата обращения 28.05.18).
[10] [10] Makhzani A., Shlens J., Jaitly N., Goodfellow I., Frey B. Adversarial autoencoders.
URL: https://arxiv.org/pdf/1511.05644.pdf (дата обращения 15.05.2018).

Айрапетов Алексей Эдуардович — студент кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Коваленко Анна Александровна — студентка кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Боровик Ирина Геннадьевна, старший преподаватель кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

INVESTIGATION OF GENERALLY-STANDING NETWORK

A.E. Ayrapetov jk3000@yandex.ru
SPIN-code: 9821-1302

A.A. Kovalenko annak0v@yandex.ru
SPIN-code: 9763-0772

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The article deals with the algorithms of generative-adversarial neural networks, which are relatively young and one of the most promising in the family of machine learning algorithms without a teacher. Currently, such neural networks are used to generate photorealistic images based on the examples provided to it. However, the potential of this subgroup of algorithms is not fully disclosed and allows increasing the plausibility of generated images. The paper describes the features of the structure of the generative-adversarial neural network, analyzes its accuracy and performance, and also provides solutions that allow optimizing the existing learning algorithm and the topology of the neural network.

Keywords

Generative-adversarial network, machine learning, neural networks, training without a teacher, generator, discriminator, perceptron, autocoder

Received 19.09.2018

References

[1] GAN — generativnye sostyazatel'nye seti [GAN — generative adversarial networks]. Available at: http://robocraft.ru/blog/machinelearning/3693.html (accessed 10 May 2018).
[2] GAN: a beginner's guide to generative adversarial networks. Available at: https://deeplearning4j.org/generative-adversarial-network (accessed 15 May 2018).
[3] Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair Sh., Courville A., Bengio Yo. Generative adversarial networks. Universite de Montreal, 2014, 9 p.
[4] Generative adversarial networks. Available at: https://habr.com/post/352794 (accessed 17 May 2018).
[5] Ganin Ya., Lempitsky V. Unsupervised domain adaptation by backpropagation. Available at: https://arxiv.org/pdf/1409.7495.pdf (accessed 15 May 2018).
[6] Gitman I. Adversarial networks. Available at: http://www.machinelearning.ru/wiki/images/4/4f/Gan.pdf (accessed 20 May 2018).
[7] How to train a GAN? Tips and tricks to make GANs work. Available at: https://github.com/soumith/ganhacks (accessed 21 May 2018).
[8] Mirza M., Osindero S. Conditional generative adversarial nets. Available at: https://arxiv.org/pdf/1411.1784.pdf (accessed 15 May 2018).
[9] A beginner's guide to deep autoencoders. Available at: https://deeplearning4j.org/deepautoencoder (accessed 28 May 2018).
[10] Makhzani A., Shlens J., Jaitly N., Goodfellow I., Frey B. Adversarial autoencoders. Available at: https://arxiv.org/pdf/1511.05644.pdf (accessed 15 May 2018).

Ayrapetov A.E. — Bachelor's Degree student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Kovalenko A.A. — Bachelor's Degree student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — I.G. Borovik, Assist. Professor, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.