

Нейромережі GAN у створенні нових моделей

Т.А. Самоліук

Інститут кібернетики імені В.М. Глушкова НАН України, 03187, м. Київ, проспект Академіка Глушкова, 40, tamara.samoliuk@ukr.net

T. Samolyuk

GAN NEURO NETWORKS IN THE CREATION OF NEW MODELS

Abstract. *The technologies of generative competitive networks are considered, the means and resources for their study and use are given. The methodical application of GAN to create a model and tool for learning a convolution network is investigated. The block diagram and algorithm of GAN neural network for creation of a new model of training of convolutional neural networks for character recognition (figures) is presented. The description of work of the developed neural network GAN is given. A wide range of applications of GAN neural networks in various applications is considered. Problems in their training are investigated. Attention is drawn to the popularity and demonstration of interesting GAN results in a wide variety of applications. The moments of many unsolved problems in their learning are presented. It is emphasized that solving these problems will enhance our ability to virtualize and model natural processes - a task that can only be solved by having a large amount of data that is not always openly available and available for study. The use of small and well-understood datasets means smaller models can be developed and trained quickly, allowing you to focus on the model architecture and the image generation process itself. With the GAN Neural Network, you can build a dataset to train the neural network, which saves time in developing it. Deep learning systems have got a really great ability to detect text, image and video templates. But programs that create realistic images, natural sentences and paragraphs, or high quality translations have proven inaccessible. Generative competing networks or GANs offer a promising solution to these challenges by connecting two competing neural networks - one that generates content and one that rejects low-quality samples.*

Key words: *GAN, convolution network, deep learning.*

Анотація. *Розглянуто технології генеративних змагальних мереж, наведено засоби та ресурси для їх вивчення та використання. Досліджено методичне застосування GAN для створення моделі та інструменту навчання згорткової мережі.*

Ключові слова: *GAN, згорткові мережі, глибоке навчання.*

Аннотация. *Рассмотрено технологии генеративных состязательных сетей, приведены способы и ресурсы для их изучения и применения.*

Ключевые слова: *GAN, сверточные сети, глубокое обучение.*

Вступ. Генеративні змагальні мережі (GAN – Generative Adversarial Nets) це засоби для генеративного моделювання з використанням методів глибокого навчання, таких як в глибоких згорткових нейронних мережах.

Генеративне моделювання – це некероване навчальне завдання в машинному навчанні, яке передбачає автоматичне виявлення і вивчення закономірностей вхідних даних таким чином, що модель може бути використана для створення або виведення нових моделей, які можна було б отримати з оригінального набору даних.

GAN здатні реалістично генерувати нові людські обличчя, а також змінювати зображення, розфарбовуючи фотографії, зістарювати обличчя, використовуючи високу роздільну здатність.

GAN – дуже молода галузь, перша стаття про такі мережі з'явилась в 2014 році [1]. На цю ж тему нині публікуються величезна кількість статей та заявок.

В даній статті розглянуто ресурси, які можна використовувати для вивчення генеративних змагальних мереж, а також наведено розробку програми, в якій GAN створює модель засобів для навчання згорткової мережі.

Основна частина. Генеративна змагальна мережа (GAN) – це тип архітектури нейронної мережі для генеративного моделювання. Існує багато застосувань GAN, пов'язаних із створенням нових моделей:

- створення нових анімаційних персон;
- створення нових логотипів;
- генерація нових покемонів;
- створення моделей нового одягу;
- створення зразків нових меблів.

GAN може використовуватися для завдань обробки зображень, фотографій і відео. Взагалі, це називається перетворенням зображень, наприклад:

- перетворення фото літа в зиму;
- перетворення ескізу на фотографію;

- перетворення фотографій удень на нічний час.

Деякі приклади зміни зображень включають:

- автоматичне старіння фотографій облич;
- автоматичне розфарбовування чорно-білих фотографій;
- автоматичне збільшення роздільної здатності фотографій;
- автоматична передача стилю (наприклад, застосуйте стиль малювання до фотографій);
- автоматичне фарбування зображення (наприклад, заповнення затемнених частин зображення).

GAN можуть також використовуватися для створення послідовностей зображень або відео і використовуються на таких завданнях, як автоматичне передбачення послідовностей відео-кадрів і навіть сценаріїв для використання в навчальних моделях. Крім обробки зображень, методика може бути використана в цілому для збільшення даних, коли при підготовці моделі можуть бути створені повністю нові вірогідні зразки.

Можливості GAN, наприклад, CycleGAN в частині перенесення (підготовки) навчальних даних для близьких доменів застосування, може бути застосовано для трансферного навчання роботів схожим технологічним процесам. Це далеко не весь список застосувань GAN навіть з відкритих джерел, і він постійно розширюється.

GAN є генеративною моделлю, яка навчається з використанням двох нейронних мереж. Одна модель називається моделлю «генератор» або «генеративна мережа», яка навчається генерувати нові вірогідні зразки. Інша модель називається «дискримінатором» або «дискримінаційною мережею» і вчиться диференціювати генеровані приклади з реальними прикладами.

Більшість досліджень і додатків GAN зосереджені на області комп'ютерного зору.

Причинами цього є великі успіхи моделей глибокого навчання протягом останніх 5 – 7 років, таких, наприклад, як Convolutional Neural Networks (CNNs) в області комп'ютерного зору і досягнення сучасних результатів на складних завданнях, таких як виявлення та розпізнавання облич.

Канонічний приклад GAN полягає у генерації нових реалістичних фотографій, найбільш

вражаюче продемонстрованих на прикладі генерації фото реалістичних облич.

Додаткові приклади застосувань GAN приведено в наступних джерелах:

- Gans-awesome-додатки: список GAN додатків і демонстрацій [2];
- деякі цікаві застосування GAN, 2018 [3].

Ознайомитися з GAN, їх роботою та програмами можна переглядаючи відеопрезентації [4].

Цікавою є оновлена версія GAN з використанням сучасних конфігураційних та навчальних практик для згорткових нейронних мереж, які називаються генеративними змагальними мережами глибокої згортки або DCGAN [5].

Приведено висвітлення GAN в сучасних книгах про глибоке навчання [6].

Навчання GAN за своєю суттю нестабільне та схильне до збоїв, які можна подолати, застосувавши евристику найкращих практик у проектуванні, конфігурації та навчанні моделі GAN.

В GAN навчаються дві нейромережі одночасно (рисунок). Генеративна мережа породжує підробку, а дискримінаційна нейромережа визначає наскільки генерований зразок реалістичний, порівнюючи його з реальним зразком. Генеративна і дискримінаційна нейромережі поводять себе як противники в деякій грі, звідси й назва змагальні нейромережі.

Зазначимо, що обидві нейромережі навчаються по черзі, а цільовою функцією служить функція втрат, яка оптимізується методом градієнтного спуску.

Генеративна модель навчається підробляти, а дискримінаційна розпізнавати підробки. Дискримінаційна мережа (як звичайно, стандартна згорткова мережа) намагається класифікувати зображення як справжнє або генероване.

Дискримінаційна мережа приймає на вході зображення і робить свій висновок про його справжність – в загальному випадку значення, близьке до 1, означає «справжній», а близьке до 0 – підробка.

Важлива нова ідея – зворотнє розповсюдження через дискримінатор і генератор з метою корекції параметрів генератора таким

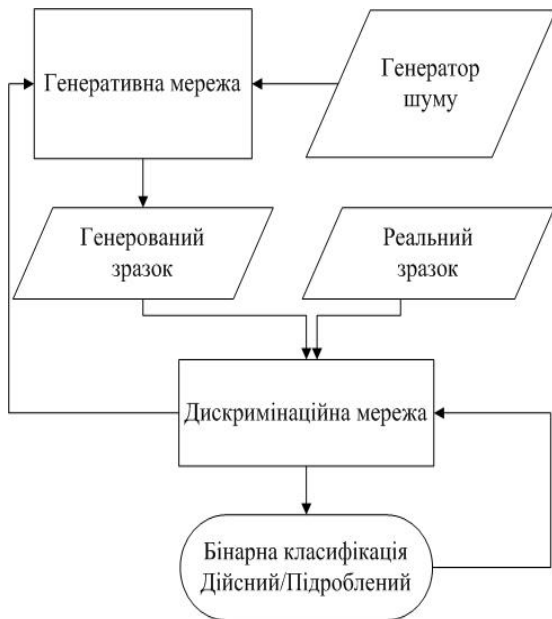


РИСУНОК. Схема генеративної змагальної мережі (GAN)

чином, щоб генератор зміг навчитися, як успішніше обманути дискримінатор.

Моделі генераторів та дискримінаторів, які використовуються в архітектурі GAN, можна визначити просто та безпосередньо в бібліотеці глибокого навчання Keras.

Дискримінаційна модель навчається, як і будь-яка інша модель глибокого навчання бінарної класифікації. Модель генератора навчається за допомогою дискримінаційної моделі в композиційній архітектурі моделі.

Застосування GAN для розширення (аугментації) навчального набору MNIST. На основі вище наведеного матеріалу розроблено GAN для розширення (аугментації) навчального набору MNIST [7].

Набір даних MNIST є аббревіатурою, що означає модифікований набір даних Національного інституту стандартів і технологій. Це набір даних із 70 000 невеликих квадратних зображень у градаціях сірого 28 x 28 пікселів, які містять рукописні однозначні цифри від 0 до 9. Завдання полягає у створенні зображення рукописних цифр від 0 до 9 включно і побудові нової навчальної вибірки в MNIST.

Бібліотека Keras надає доступ до набору даних MNIST за допомогою функції `mnist.load_dataset()`. Вона повертає два кортежі, один з вхідними та вихідними елементами для

стандартного навчального набору даних, а інший з елементами введення та виведення для стандартного тестового набору даних.

Генератор генерує зображення цифр та створює нову вибірку для навчання нейромережі.

Зокрема, модель генератора навчається генерувати нові правдоподібні цифри від 0 до 9, використовуючи дискримінатор, який намагається порівняти реальні зображення з навчального набору MNIST і нові зображення, що виводяться моделлю генератора.

Модель генератора відповідає за створення нових, підроблених, але правдоподібних зображень рукописних цифр.

Він робить це, приймаючи точку від генератора шуму як вхідні дані і виводячи квадратні зображення розміром 28 x 28 пікселів у градаціях сірого. Генератор шуму або латентний простір є довільно визначеним векторним простором гауссово-розподілених значень, наприклад 100-розмірним. Модель генератора буде вводити значення прихованих точок i , у свою чергу, прихованого простору поки, наприкінці навчання, латентний векторний простір не стане стисненим представленням вхідного простору MNIST-зображення, і генератор перетворить його на правдоподібні MNIST-зображення.

- Входи генератора: точка в прихованому просторі, наприклад, 100-елементний вектор гауссівських випадкових чисел.

- Виходи генератора: двовимірне квадратне зображення у градаціях сірого 28 x 28 пікселів із значеннями в $[0, 1]$.

Розробка моделі генератора вимагає перетворення вектора з прихованого простору, з 100 розмірами до 2D масиву розміром 28 x 28 або 784 значеннями.

Першим кроком є створення нових точок у прихованому просторі. Ми викликаємо функцію `randn() NumPy` для генерації масивів випадкових чисел, витягнутих із стандартного гауссового. Потім масив випадкових чисел може бути змінений у зразки, тобто n рядків з 100 елементами на рядок. Функція `generate_latent_points()` реалізує це і генерує необхідну кількість точок у прихованому просторі, яка може бути використана як вхідна інформація для моделі генератора.

Далі, ми можемо використовувати генеровані точки як вхідні дані для моделі генератора

для створення нових вибірок, а потім побудувати нові зображення рукописних цифр.

Оскільки модель не навчена, сформовані зображення є абсолютно випадковими значеннями пікселів в $[0, 1]$.

Після визначення моделі генератора, наступним кроком є навчання моделі.

Ваги в моделі генератора оновлюються на основі продуктивності моделі дискримінатора.

Коли дискримінатор хороший при виявленні підроблених зразків, генератор оновлюється більше, і коли модель дискримінатора є відносно слабкою або плутоною при виявленні підроблених зразків, модель генератора оновлюється менше.

Модель дискримінатора має взяти зразок зображення з нашого набору даних як вхідний та визначити вихідний прогноз класифікації щодо того, чи є зразок реальним або підробленим.

Це двійкова проблема класифікації:

- входи дискримінатора: зображення з одним каналом і розміром 28×28 пікселів;
- виходи дискримінатора: двійкова класифікація, ймовірність реальної вибірки (або підробки).

Функція `define_discriminator()` визначає модель дискримінатора і визначає розмір вхідного зображення. Модель дискримінатора має два згорткових шари з 64 фільтрами кожен, невеликий розмір ядра 3, і більше, ніж звичайний шар 2. Вхідний зразок є реальним або підробленим.

Модель підготовлена для мінімізації функції бінарної крос-ентропії, придатної для двійкової класифікації.

Використано кращі практики визначення моделі дискримінатора, наприклад, LeakyReLU замість ReLU, Dropout і версію стохастичного градієнтного спуску Adam із швидкістю навчання 0,0002, імпульсом 0,5.

Модель буде оновлюватися партіями, зокрема, з колекцією реальних зразків і колекцією генерованих вибірок.

На тренуванні епоха визначається як один прохід через весь навчальний набір. Функція `generate_real_samples()` приймає навчальний набір як аргумент і вибирає випадковий набір зображень; вона також повертає мітку класу для вибірки, зокрема, мітку класу 1, для позначення реальних зображень.

Джерело підроблених зображень генеруємо такими, що складаються з випадкових значень пікселів у діапазоні $[0, 1]$, подібно масштабованим реальним зображенням.

Функція `generate_fake_samples()` генерує зображення випадкових значень пікселів і пов'язану з ними мітку класу 0 для підробки.

Нарешті, потрібно навчити модель дискримінатора.

Це передбачає багаторазове вилучення зразків реальних зображень і зразків генерованих зображень і оновлення моделі для фіксованого числа ітерацій.

Модель навчиться швидко розрізняти реальні та фальшиві (випадково генеровані) зображення, отже, не потрібно буде багато партій, перш ніж вона навчиться розпізнавати відмінно.

Функція `train_discriminator()` реалізує навчання дискримінатора, використовуючи розмір партії в 256 зображень, де 128 є реальними і 128 є підробленими зображеннями.

На заключному етапі роботи GAN отримуємо побудовану вибірку для навчання мережі з розпізнавання знаків (цифр).

Мова програмування – Python 3.6. Середовище розробки – Spyder в Anaconda.

Наступним етапом проекту є застосування вибраного підходу створення вибірок для навчання нейромереж у різних галузях промисловості, транспорту, освіти, медицини, криміналістики. Наприклад, аналогічно можна проводити аугментацію навчальних вибірок складальних деталей для роботизованих технологічних ліній. Запропоновано застосування GAN для трансферного навчання роботів схожим технологічним процесам.

Висновки. GAN – це клас генеративних моделей, що належить до навчання без учителя.

Нині вони дуже популярні і демонструють цікаві результати своєї роботи в самих різних областях застосування. На даний момент існує багато нерозв'язаних проблем в їх навчанні. Так моделі GAN можуть мати наступні проблеми:

- колапс режиму: генератор виробляє обмежені варіанти зразків;
- зменшений градієнт: дискримінатор стає досить успішним, коли градієнти зникають, а генератор нічого не дізнається;
- незбіжність: параметри моделі коливаються, дестабілізуються і ніколи не збігаються;

- незбалансованість між генератором і дискримінатором визиває перенавчання;

- дуже чутливі до гіперпараметрів.

Їх розв'язання підвищить нашу здатність віртуалізувати і моделювати природні процеси – завдання, яке ми можемо розв'язати лише маючи велику кількість даних, які не завжди знаходяться у відкритому доступі та доступні для вивчення. Удосконалення GAN, такі, як масштабування моделей та поступове зростання моделей, дозволяє створювати більші та якісні зображення.

Використання малих та добре зрозумілих наборів даних означає, що менші моделі можуть бути розроблені та навчені швидко, що дозволяє зосередитися на архітектурі моделі та самому процесі генерації зображень. Застосування нейромережі GAN для побудови набору даних зекономить час її розробки.

Системи глибокого навчання отримали дійсно чудову можливість виявлення шаблонів тексту, зображень та відео. Але програми, які створюють реалістичні зображення, правильні речення та абзаци, або переклади з високою якістю, виявились недоступними. Генеративні змагальні мережі або GAN пропонують перспективне рішення цих завдань, з'єднавши дві конкуруючі нейронні мережі – одну, яка генерує контент, а інша, яка відкидає зразки низької якості.

СПИСОК ЛІТЕРАТУРИ

1. Goodfellow I.J., et al, Generative adversarial networks. *Universite de Montreal*. 2014. 9 p.
2. Gans-awesome-applications: Curated list of awesome GAN applications and demos.
URL: <https://github.com/nashory/gans-awesome-applications> (дата звернення: 12.09.2019).
3. Some cool applications of GANs, 2018.

URL:https://medium.com/@jonathan_hui/gan-some-cool-applications-of-gans-4c9ecca35900 (дата звернення: 12.09.2019).

4. Goodfellow I.J. Generative Adversarial Networks, NIPS 2016 Tutorial.
URL:<https://www.youtube.com/watch?v=HGYYEUSm-0Q> (дата звернення: 12.09.2019).
5. Alec Radford, et al. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015. URL:(<https://arxiv.org/abs/1511.06434> (дата звернення: 12.09.2019)).
6. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. ООО «Питер Пресс», 2017. С. 353–372.
7. Набір даних MNIST
URL:(<http://yann.lecun.com/exdb/mnist/> (дата звернення: 10.09.2019)).

REFERENCES

1. Goodfellow I.J., et al. Generative adversarial networks. *Universite de Montreal*. 2014. 9 p.
2. Gans-awesome-applications: Curated list of awesome GAN applications and demos Available at: <https://github.com/nashory/gans-awesome-applications> (accessed 12 September 2019).
3. Some cool applications of GANs, 2018. Available at: https://medium.com/@jonathan_hui/gan-some-cool-applications-of-gans-c9ecca35900 (accessed 12 September 2019).
4. Goodfellow I.J. Generative Adversarial Networks, NIPS 2016 Tutorial. Available at: <https://www.youtube.com/watch?v=HGYYEUSm-0Q> (accessed 12 September 2019).
5. Alec Radford, et al. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015. Available at: <https://arxiv.org/abs/1511.06434> (accessed 12 September 2019).
6. Nikolenko S., Kadurin A., Arkhangelskaya E. Glubokoe obuchenie. Pogrugenie v mir neironnih setey. Peter Press LLC, 2017. P. 353–372.
7. Nabor dannih MNIST. Available at: <http://yann.lecun.com/exdb/mnist/>(accessed 12 September 2019).

Одержано 24.09.2019