МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
Высшего образования
«Нижегородский государственный университет им. Н.И. Лобачевского»
Национальный исследовательский университет
Институт информационных технологий, математики и механики Кафедра математического обеспечения и суперкомпьютерных технологий

Отчёт по практике

Тема :

Generative adversarial networks for Text-to-Image Synthesis

Выполнил:
студент гр. 381706-1
Митягина Д.С.
Научный руководитель:
Профессор, доктор технических наук
Турлапов В.Е.

Нижний Новгород 2019

Содержание

1 Введение

Одной из самых сложных проблем в мире Computer Vision является синтез высококачественных изображений из текстовых описаний. Без сомнения, это интересно и полезно, но современные системы искусственного интеллекта далеки от этой цели.

Генерация изображений имеет множество возможных применений в будущем, когда технологии будут готовы для коммерческого применения. Люди смогут создавать схему расположения мебели для своего дома, просто описывая ее на компьютере, а не тратя много часов на поиск нужного дизайна.
Создатели контента смогут творить в более тесном сотрудничестве с машиной, используя естественный язык. Кроме того, данная тема может стать более интересной, если ее развить. А именно, перевести задачу из генерации 2d изображения в построение 3d сцены и даже, возможно, формирование видео материалов с использование полученных сцен.

В данном отчете будут
- рассмотрены уже существующие решения задачи генерации изображений по текстовому описанию
- разобраны основные составляющие части программной реализации этих решений
- описаны основные положения из теории, лежащие в основе выбранных методов
- предъявлены результаты некоторых проведенных экспериментов

2 Постановка задачи

1. Исследовать различные публикации на поставленную тему.
2. Изучить средства и методы решения задачи синтеза изображений из их словесного описания.
3. Начать проведение практических экспериментов.

3 Необходимый теоретический минимум

1. Граф сцены

Граф сцены представляет структуру, которая содержит логическое и зачастую (но не обязательно) пространственное представление графической сцены. Определение графа сцены нечёткое, поскольку программисты, осуществляющие его реализацию в приложениях, — и, в частности, в индустрии разработки игр — берут базовые принципы и адаптируют их для применения в конкретных приложениях. Это означает, что нет договорённости о том, каким должен быть граф сцены.

Граф сцены представляет собой набор узлов такой структуры, как граф или дерево. Узел дерева может иметь множество потомков, но зачастую только одного предка, причём действие предка распространяется на все его дочерние узлы; эффект действия, выполненного над группой, автоматически распространяется на все её элементы. Во многих программах ассоциирование матрицы преобразования на уровне любой группы и умножение таких матриц представляет собой эффективный и естественный способ обработки таких действий. Общей особенностью, к примеру, является способность группировать связанные формы/объекты в составной объект, который можно перемещать, трансформировать, выбирать и т. д. так же просто, как и одиночный объект.

2. Свёрточная нейронная сеть

Идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв (англ. convolution layers) и субдискретизирующих слоёв. Структура сети — однонаправленная (без обратных связей), принципиально многослойная. Для обучения используются стандартные методы, чаще всего метод обратного распространения ошибки. Функция активации нейронов (передаточная функция) — любая, по выбору исследователя.

3. Генеративно-состязательная сеть (GAN)

Алгоритм машинного обучения без учителя, построенный на комбинации из двух нейронных сетей, одна из которых (сеть G) генерирует образцыа другая (сеть D) старается отличить правильные («подлинные») образцы от неправильных. Так как сети G и D имеют противоположные цели — создать образцы и отбраковать образцы — между ними возникает Антагонистическая игра. Генеративно-состязательную сеть описал Ян Гудфеллоу из компании Google в 2014 году.

Использование этой техники позволяет в частности генерировать фотографии, которые человеческим глазом воспринимаются как натуральные изображения.

4. Архитектура нейронной сети

1. Входные узлы (входной слой): вычислений в этих слоях нет, они просто передают информацию следующему слою.
2. Скрытые узлы (скрытый слой): в скрытых слоях выполняется промежуточная обработка или вычисления, после чего происходит перенос весов с входного слоя на следующий слой.
3. Выходные узлы (выходной слой): здесь мы наконец используем функцию активации.
4. Соединения и веса: сеть состоит из соединений, каждое соединение передает выход нейрона i на вход нейрона j. В этом смысле i является предшественником j, а j является преемником i. Каждому соединению присваивается вес $W_{i,\,j}$.
5. Функция активации: функция активации узла определяет выходные данные этого узла с учетом входных данных или набора входных данных.
6. Правило обучения. Правило обучения - это правило или алгоритм, который изменяет параметры нейронной сети для того, чтобы данный вход в сеть создавал предпочтительный результат. Этот процесс обучения обычно сводится к изменению весов и порогов.

4 Обзор источников

На данную тему существует сравнительно немного литературы, что дает понять ее новизну и необходимость дальнейшего ее изучения.
Несомненно, одним из наиболее важных аспектов работы являются данные, использованные для обучения моделей.
Каждая статья отличается своим подходом к решению данной проблемы, но в большинстве своем они содержат в качестве составной части генеративно-состязательную сеть и различные ее модификации(ObjGAN, StackGAN, StackGAN++, AttnGAN и так далее).
Каждый из существующих методов дает отличные от других результаты. Если попытаться обобщить, то можно сказать, что есть алгоритмы, прекрасно выполняющие свою задачу при несложной структуре сцены и маленьком количестве объектов, но показывающие плохие результаты при усложнении одного из критериев. Есть авторы, попытавшиеся расширить границы возможностей GAN в данной области. В данном отчете будут рассмотрены примеры из обеих выделенных мною категорий.
А теперь уделим вниание данным, на которых обучались сети. Как говорил профессор Эндрю Ын в своих курсах: «В деле машинного обучения достигает успеха не тот, у кого есть наилучший алгоритм, а тот, у кого есть наилучшие данные». Чаще всего используются стандартные наборы данных(Data set) с фотографиями и текстовым описанием интересующих объектов(в большей части статьей такими объектами являются птицы, цветы, иногда и лица людей).
Далее приведен более подробный анализ двух конкретных статей :
- Image Generation from Scene Graphs, Justin Johnson, Agrim Gupta, Li Fei-Fei;
- AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks, Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, Xiaodong He

5 Разбор статей

5.1 Статья 1 : Image Generation from Scene Graphs

Justin Johnson, Agrim Gupta, Li Fei-Fei

Большинство существующих методов дают потрясающие результаты на ограниченных доменах, таких как описания птиц или цветов, но не могут точно воспроизвести сложные предложения со многими объектами и отношениями.
Для преодоления этого ограничения в данной статье авторы предлагают метод генерирования изображений из графов сцен, позволяющих явно рассуждать об объектах и их отношениях. Представленная в публикации модель использует свертку графа для обработки входных графов, вычисляет макет сцены, прогнозируя границы Bounding Box (BBox, ограничивающий параллелепипед) и маски сегментации для объектов, и преобразует макет в изображение с каскадным уточнением сети.

Основы метода :
Предложение представляет собой линейную структуру, в которой одно слово следует за другим; однако, информация, передаваемая сложным предложением, часто может быть более явно представлена в виде графа сцены объектов и их отношений.

- Для обработки входных данных графа сцены используется сеть свертки графа.
- Для создания изображения, которое соответствует макету применяется cascaded refinement network (CRN), которая обрабатывает макет.
- Чтобы убедиться, что сгенерированные изображения реалистичны и содержат необходимые объекты разумно применить генеративно-состязательные сети, работающие на патчах изображений и сгенерированных объектах.
- Сквозной процесс обучения можно разделить на два основных компонента :
Учебный компонент - первый этап, на котором машина записывает все параметры, выполняемые оператором (через Сверточные нейронные сети (CNN)).
Компонент логического вывода возможен тогда, когда машина действует на основе ранее полученного опыта от компонента обучения сквозного процесса обучения.

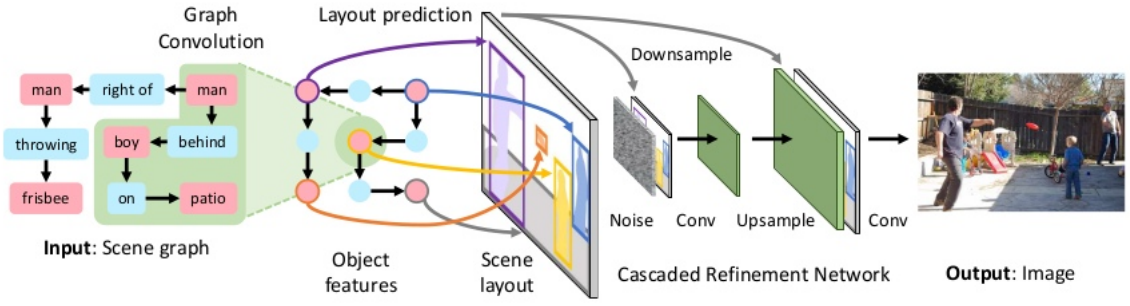

*Рис. 1 - Схема, описывающая метод

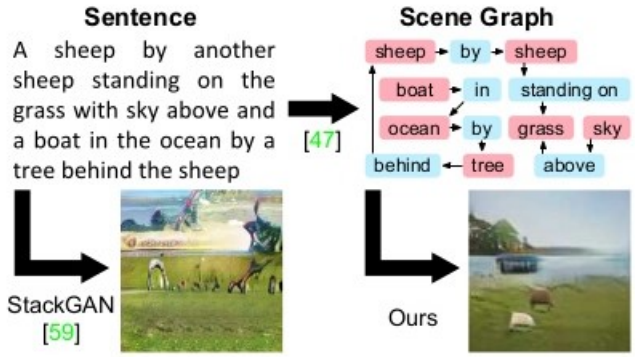

*Рис. 2 - Генерация графа сцены по предложению

Входом для описанной модели является граф сцены, описывающий объекты и связи между ними. Дан набор категорий объектов C и набор категории R, граф сцены - это граф (O, E), где O = o_1,. , , , o_n - множество объектов с каждым o $i \in C$, и $E \subseteq O \times R \times O$- множество направленных ребер вида (o_i, r, o_j) где o_i, $o_j \in O$и$r \in R$.

Конкретно, заданные входные векторы v_i, v_r $\in \mathrm{R}^D$ для всех объектов $o_i \in O$ и ребра (o_i, r, o_j) $\in E$, вычисляем вывод векторы для v_i', v_r' $\in \mathrm{R}^D_{out}$ для всех узлов и ребер с использованием три функции g s, g p и g o, которые принимают в качестве входных данных тройку векторов (v_i, v_r, v_j) для ребра и вывод новых векторов для субъекта o_i, предиката r и объекта o_j соответственно. Чтобы вычислить выходные векторы v_r' для ребер, мы просто устанавливаем $v_r = g_p\ (v_i, v_r, v_j)$.
Аналогично использовалось g_o для вычисления набора кандидатов V_o^i для всех ребер, оканчивающихся на o_i. В частности,
$V_s^i = g_s\ (v_i, v_r, v_j)$: $(o_i, r, o_j) \in E$.
$V_o^i = g_o\ (v_j, v_r, v_i)$: $(o_j, r, o_i \in E$.
Выходной вектор для объекта o i затем вычисляется как $v_i' = h\ (V_s^i \bigcup V_o^i)$, где h - симметричная функция, которая объединяет входной набор векторов в один выходной вектор.

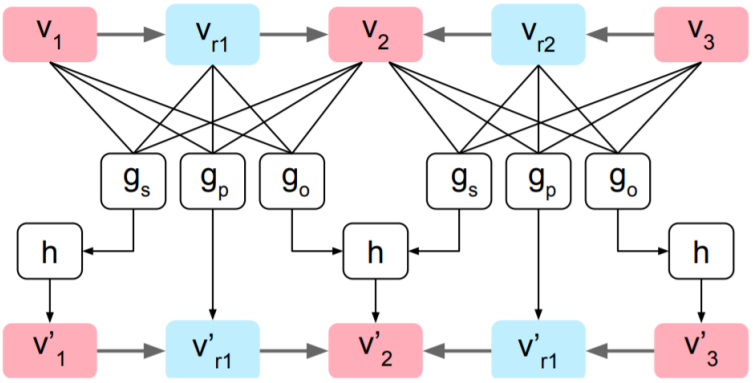

*Рис. 3 - Вычислительный граф

Вычислительный граф, иллюстрирующий один граф сверточного слоя. Граф состоит из трех объектов o_1, o_2 и o_3 и два ребра (o_1, r_1, o_2) и (o_3, r_2, o_2). Вдоль каждого края три входные вектора передаются в функции g_s, g_p и g_o; g_p напрямую вычисляет выходной вектор для ребра, а g_s и g_o вычисляют векторы-кандидаты, которые подаются к симметричной функции пула h для вычисления выходных векторов для объектов.

Авторы генерируют реалистичные выходные изображения, обучая сеть генерации изображений e состязательно против пары дискриминаторных сетей D_{img} и D_{obj}. Дискриминатор D пытается классифицировать входные данные x как реальные или поддельные путем максимизации $L_{GAN} = \underset{x \sim p_{real}}{E} logD(x) + \underset{x \sim p_{fake}}{E} log(1 - D(x))$
гдех$\sim$ p_{fake} - выход из сети генерации f. В то же время, f пытается генерировать выходные

данные, которые будут обмануть дискриминатор путем минимизации L_{GAN}. D_{img} дискриминатор изображения на основе патча обеспечивает общий вид сгенерированных изображений реалистичен; он классифицирует регулярно расположенный, перекрывающийся набор изображений патчи как реальные или фальшивые. Дискриминатор объекта D_{obj} гарантирует, что каждый объект на изображении выглядит реалистичным. В дополнение к классификации каждого объекта как реальный или фальшивый, D_{obj} также гарантирует, что каждый объект распознаваем, используя вспомогательный классификатор, который предсказывает категория объекта; и D_{obj}, и F пытаются максимизировать вероятность того, что D_{obj} правильно классифицирует объекты.

5.2 Статья 2 : AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks

Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, Xiaodong He

В этой статье предложена Attentional Generative Adversarial Network (AttnGAN), которая обеспечивает сосредоточенное, многоэтапный синтез изображения из текста. Благодаря новой генеративной сети внимания AttnGAN можно синтезировать мелкозернистые детали в разных областях изображения, обращая внимание на соответствующие слова в описании на естественном языке. Предложенная Генеративная Состязательная Сеть (AttnGAN) имеет два новых компонента: генеративная сеть с вниманием (attentional generative network) и глубокая мультимодальная модель подобия внимания(the deep attentional multimodal similarity model).

Описание метода :
Современные GAN-модели для генерации текста в изображения обычно кодируют целое предложение в текстовое описание в одном векторе как условие для генерации изображения, но не хватает детальной информации на уровне слов. В этом разделе авторы статьи предлагают новую модель внимания, которая позволяет генеративной сети рисовать различные субрегионы изображения, основанные на отношении к этим субрегионам. Предложенная генеративная сеть внимания имеет m генераторов (G_0, G_1, ..., G_{m-1}), которые берут скрытые состояния (h_0, h_1, ..., h_{m-1}) в качестве входных данных и генерируют изображения малых и больших масштабов ($\hat{x^0}$, $\hat{x^1}$, ..., $\hat{x^{m-1}}$).
В частности, $h_0 = F_0$ (z, F^{ca} ($\overline{e}$));
$h_i = F_i$ (h_{i-1}, F_i^{attn}(e, h_{i-1})) для i = 1, 2, ..., m - 1;
$\hat{x^i} = G_i$ (h_i).

Здесь z - вектор шума, обычно выбираемый из стандартного нормального распределения, $\overline{e}$ является глобальным вектором предложения, а e является матрица векторов слов. F^{ca} представляет собой кондиционирование дополнение, которое преобразует вектор предложения e в вектор кондиционирования. F^{attn} - предлагаемая модель внимания на i-м этапе AttnGAN. F^{ca}, F_i^{attn}, F_i и G_i моделируются как нейронные сети.
Для создания реалистичных изображений с несколькими уровнями (т.е. уровень предложения и уровень слова) условий, определяется конечная целевая функция генеративной сети внимания:
$L = L_G + \lambda\ L_{DAMSM}$, where $L_G = \sum_{i<m1}^{i=0} L_{G_i}$

Модель глубокого внимательного мультимодального сходства DAMSM изучает две нейронные сети, которые отображают субрегионы изображения и слова предложения в общее семантическое пространство, таким образом, измеряет сходство изображения с текстом на уровне слова, чтобы вычислить мелкозернистые потери при генерации изображения.

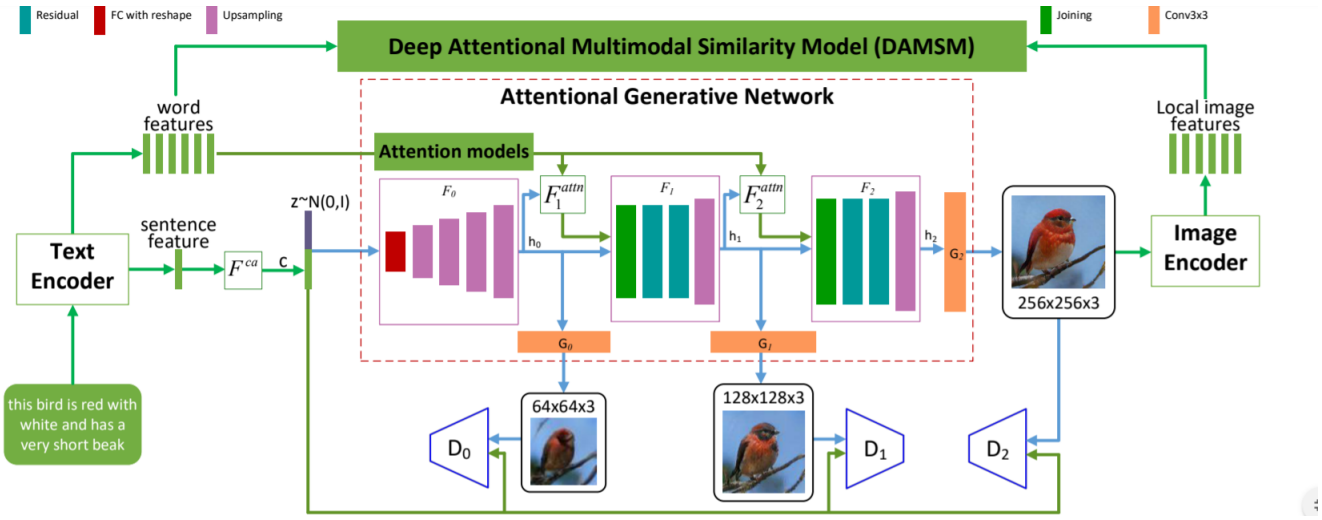

*Рис. 4 - Схема, описывающая метод

6 Результаты практики

Была предпринята попытка расширить теоретические знания в области нейронных сетей (в том числе применительно к задаче генерации изображений по текстовому описанию сцены). В результате работы был реализован простейший автокодировщик, являющийся составной частью одного из описанных методов.
Так же была реализована простая версия одной из частей конволюционной(сверточной) нейронной сети. Выбрана именно конволюционная сеть, т.к. она(ее модификация) применяется в рассматриваемой в курсовой работе статье.
Полученная модель дала следующие результаты :

```
In [15]: epoch_list = list(range(1, len(hist.history['acc']) + 1))
         plt.plot(epoch_list, hist.history['acc'], epoch_list, hist.history['val_acc'])
         plt.legend(("Training Accuracy", "Validation Accuracy"))
         plt.show()
```

*Рис. 5 - Точность участка Training против точности Validation. Простейший пример.

7 Заключение

В ходе проделанной работы были достигнуты поставленные цели.

1. Исследованы различные публикации на поставленную тему (2).
2. Изучены средства и методы решения задачи синтеза изображений из их словесного описания.
3. Проведены практические эксперименты.

В дальнейшем планирую рассмотреть следующую литературу:

1. Generative Adversarial Networks Cookbook by Josh Kalin, December 2018
2. Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network, Zizhao Zhang, Yuanpu Xie, Lin Yang

8 Список литературы

1. AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks, Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, Xiaodong He
2. Image Generation from Scene Graphs, Justin Johnson, Agrim Gupta, Li Fei-Fei
3. CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training, Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, Gang Hua
4. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas
5. Text-to-Image Synthesis Based on Machine Generated Captions, Marco Menardi, Alex Falcon, Saida S.Mohamed, Lorenzo Seidenari, Giuseppe Serra, Alberto Del Bimbo and Carlo Tasso
6. MirrorGAN: Learning Text-to-image Generation by Redescription Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao
7. Object-driven Text-to-Image Synthesis via Adversarial Training, Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, Jianfeng Gao