# 1. Аннотация работы

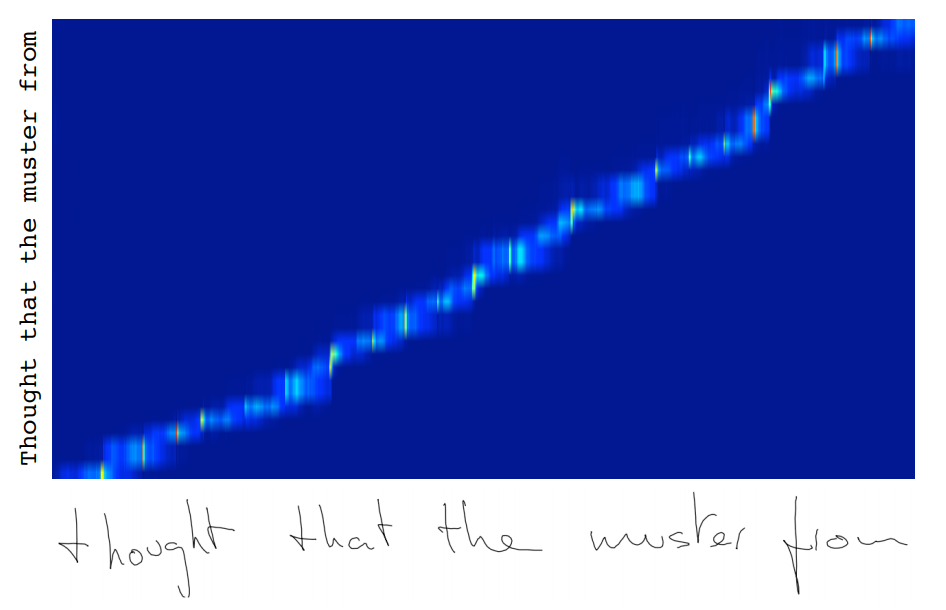
В работе представлена модификация существующей рекуррентной нейронной сети для генерации рукописного текста на основе некоторой входной последовательности символов. Эта модификация заключается в использовании дискриминатора для обучения вместо функции потерь. Эксперименты показали, что использование дискриминатора для обучения не является эффективным.

# 2. П**остановка задачи исследовани**я

Целью данной работы является исследование применимости Generative Adversarial Networks к задаче генерации рукописного текста на примере добавления дискриминатора к существующему генератору (Scribe).

# 3. О**бзор публикаций и известных решений**

Базой для работы является проект [Scribe](https://greydanus.github.io/2016/08/21/handwriting/), в котором описан генератор рукописного текста, основанный на статье [“Generating Sequences With Recurrent Neural Networks”](https://arxiv.org/abs/1308.0850) (Graves, 2014). В данной статье представлена сеть на основе ячеек LSTM. Особенностью сети является применение механизма внимания (attention mechanism), который сеть использует для того, чтобы определить, какой участок текста генерируется в настоящий момент.



Пример работы attention mechanism для генерации строки

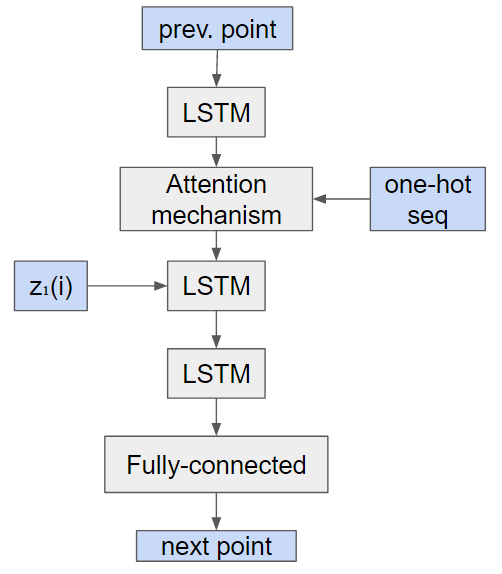
В “[Language Generation with Recurrent Generative Adversarial Networks without Pre-training](https://arxiv.org/abs/1706.01399)” (Press et al., 2017) описана рекуррентная сеть для генерации текста, использующая GAN для обучения. Также представлены несколько расширений для процесса обучения:

* Curriculum Learning (CL) - обучение начинается с генерации коротких предложений и длина постепенно наращивается.
* Variable Length (VL) - во время обучения генератор создает все последовательности длиной от 1 до l вместо одной последовательности длины l.
* Teacher Helping (TH) - для генерации последовательность длины i генератору на вход подается последовательность реальных данных длиной i-1 и генератор предсказывает только одну (последнюю) точку.

В статье “[C-RNN-GAN: Continuous recurrent neural networks with adversarial training](https://arxiv.org/abs/1611.09904)” (Morgen, 2016) описана генерация музыки с помощью LSTM и GAN. Особенностью вляется “заморозка” обучения дискриминатора в том случае, если дискриминатор становится слишком сильным, то есть имеет слишком низкое значение функции потерь (70% от потерь G)

# 4. О**писание разработанного прототипа**

Мы практически не меняли генератор Scribe и используем ту же структуру. Изменения заключаются в том, что последний слой предсказывает конкретную точку вместо распределения, а также в добавлении шума z(i) во второй слой LSTM.



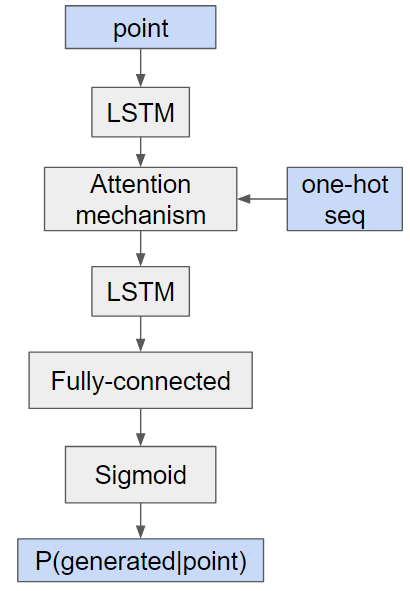
Структура генератора

Дискриминатор имеет похожую структуру, с той разницей, что в нем используется два слоя LSTM вместо трех, нет шума и на последнем слое используется сигмоида.

Функции потерь генератора и дискриминатора задаются как:

g_loss.gif

d_loss.gif

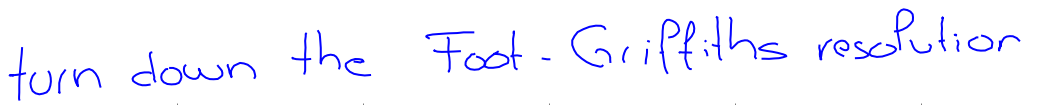


Структура дискриминатора

# 5. И**спользуемы**е **данны**е

Для генерации использовался набор [IAM Handwriting Database](http://www.fki.inf.unibe.ch/databases/iam-handwriting-database). В нем содержится 13353 строки с текстом от 657 авторов.

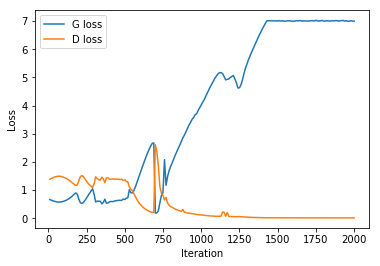
Примеры (взяты из описания Scribe):



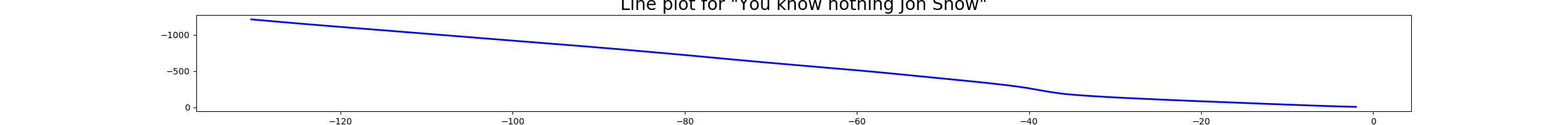
Каждая строка описывается набором точек из 3-х параметров: x1, x2, eos (end of stroke). x1 и x2 являются координатами, а eos - бинарным значением 0 или 1, где 1 означает конец штриха.

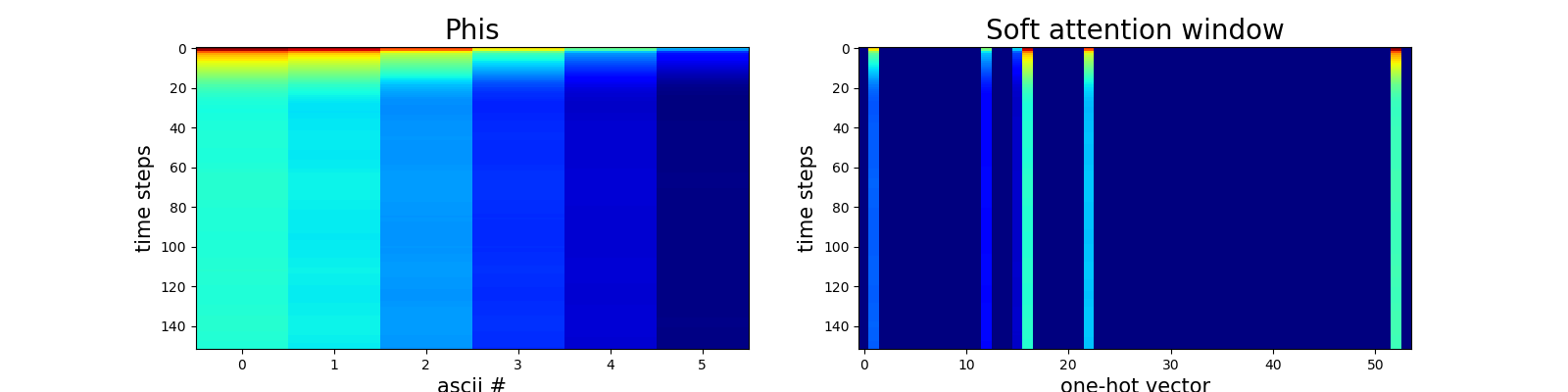
# 6. О**писание экспериментов и полученные результаты**

Полученные результаты свидетельствуют о том, что дискриминатор не предоставляет данные, которые позволяют генератору эффективно обучаться. По функции потерь видно, что начиная с некоторого момента генератор уже не способен создавать “правдоподобные” последовательности и это приводит к тому, что ошибка генератора уже после 500 итераций только увеличивается.



В свою очередь это приводит к тому, что генератор не выучивает нужные параметры распределения. Вывод после двух тысяч итераций выглядит приблизительно так:





По графикам, описывающим поведение механизма внимания можно судить, что модель не может корректно определить, какую часть текста он создает в данный момент.

# 7. Ч**то делать дальше**

Основная задача чтобы начать получать удчаные результаты - сделать генератор более мощным. Этого можно попытаться достичь с помощью (Press et al., 2017). Однако введение Curriculum Learning потребует изменения работы механизма внимания, так как в существующей имплементации он не может работать менее, чем с 25 точками.