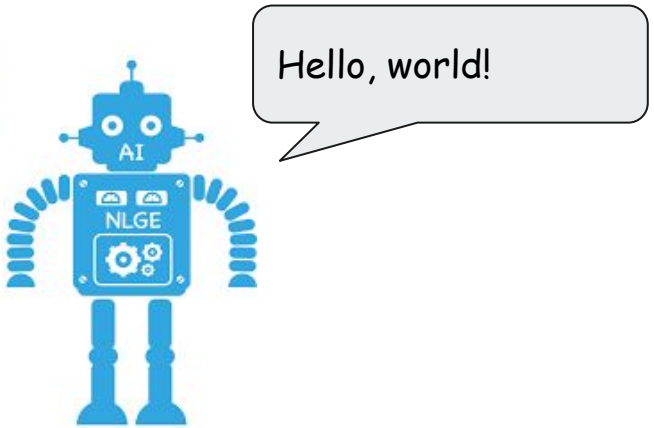
Neural Models Exploration in Natural Language Generation

Team: Yifeng Liu, Lixue Zhang, Wen Cui, Sha Tong, Ran Xu



Outline

- Motivation
- Dataset
- Two Models
  - RNN Model
  - WGAN
- Evaluation
- Discussion

Motivation

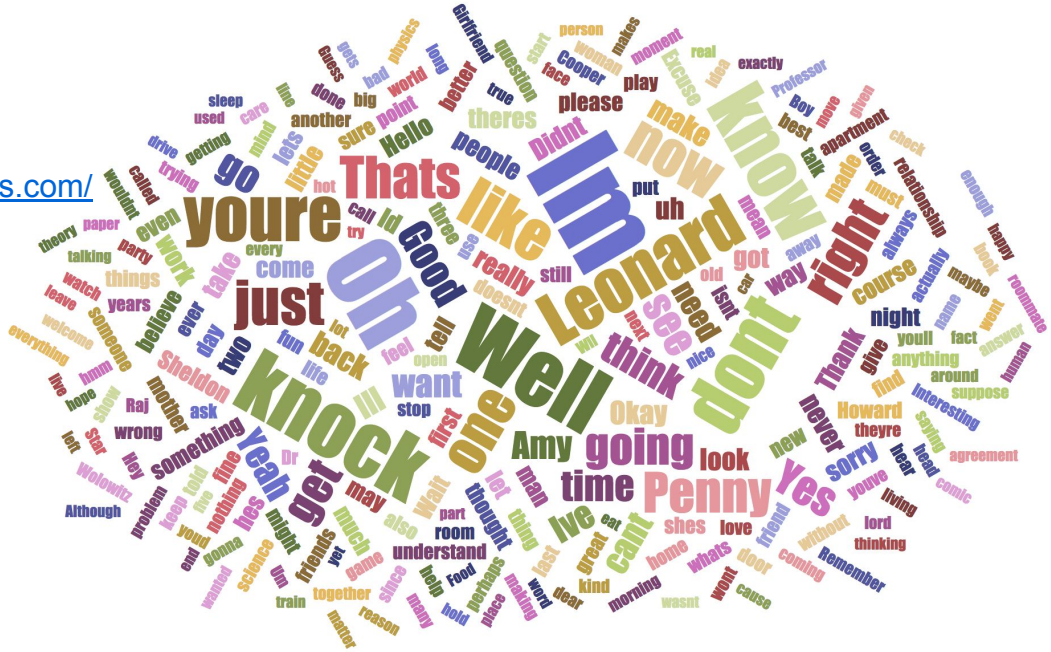
In this project, we implemented two different models for natural language generation. We first implement current state-of-art model which is utilizing Recurrent Neural Networks. And in the meanwhile, we also experiment Generative Adversarial Networks. We want to compare their performance in the aspect of the quality of the text generated. We expected some difficulty in training GANs due to the nature complexity of text process.

- Language Generation
- Learning Characteristic

Dataset

Crawled S1-S10 from https://bigbangtrans.wordpress.com/

```
Scene: Leonard and Penny's bedroom.
Sheldon: Leonard? Leonard?
Leonard: What?
Sheldon: You realize you and I could become brothers.
Leonard: We're not gonna be brothers. We're not gonna be
stepbrothers. Go to sleep.
Sheldon: I hope you're right. 'Cause a grown man living
with his brother and his brother's wife is weird.
Leonard: Go to sleep.
...
```



Character	Sheldon	Leonard	Penny	Howard	Raj	Amy	Bernadette	Rest
Dialogues	10949	9251	7276	5552	4181	3079	2437	8123

Reference: Image generated from word cloud https://www.jasondavies.com/wordcloud/

RNN: text generation theory basis.

Why RNN?

Natural language is inherently sequential, such as human speech, reading novels, magazine and papers, just to mention some. Recurrent Neural Networks (RNNs) are a family of neural networks designed specifically for sequential data processing. Thus, it is suitable for text understanding, representation, and generation.

Word by word ? or Character by Character ?

We can either regard each word as a unit or split them into characters? Both of those two methods exist now, below are their comparisons from some articals.

1> Word-based displays higher accuracy and lower computational cost than char-based solution. It's hard for char-based solution to capture long-short term memory.

2> One hot problem with word-based solution. When there are too manys words in your training dataset, the one hot vector would be really long. Sometimes, it would be really hard for you to train the model.

RNN: text generation theory basis.

RNN language model.
We can now start formalizing our ideas, let's consider a sentence S composed by T words, such that

$$S = (w_1, w_2, \ldots, w_T)$$

At the same time, each symbol Wi is from a vocabulary V which contains all the possible words,

$$V = \{v_1, v_2, \ldots, v_{|V|}\}$$

If we want to compute the probability of a sentence, we can use the chain rule to get

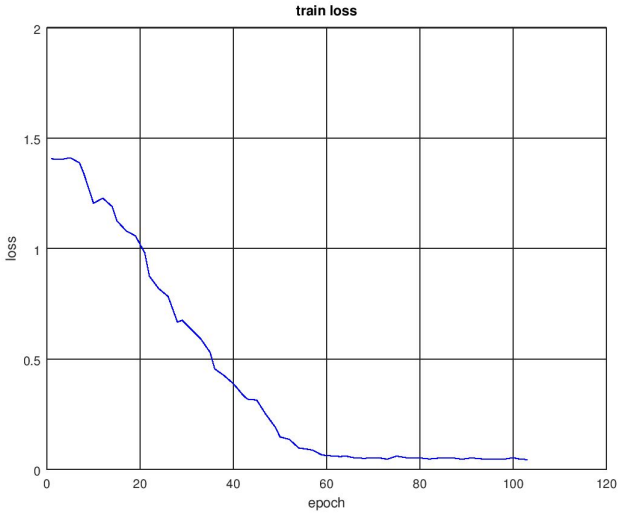
$$p(S) = p(w_1, w_2, \ldots, w_T) = p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_2, w_1) \cdots p(w_T|w_{T-1}, w_{T-2}, \ldots, w_1)$$

Loss Function.
The loss for a given sentence is the negative log probability the model assigns to the correct output.

$$L(\mathbf{x}) = -\sum_t log\, p_{model}(w_t = x_{t+1}) = -\sum_t log\, \mathbf{o}_t[x_{t+1}]$$

RNN: word-based lstm

Input rnn size: 512
Layers: 2
Basic cells: basic lstm cell
Embidding: trained 512 x 14837
Number of epoch: 100 currently. It is still running.



Sample output:

sheldon: oh, listen to them. now howard
sheldon: hysterical. now we might do how it
sheldon: oh, very problems.(he opens an arm)
sheldon: it doesn’t matter. that’s my spot.
sheldon: it might never be had that pizza was been an lives and then he would
sheldon: of course it’s gone, i’d be going to go to the same science
sheldon: don’t worry, don’t be fooled.
sheldon: it might be not my winter figure. the
sheldon: don’t worry enough as a lack of the mind and you’re only myself with

RNN: char-based lstm

Difference from word-based solution.
Total chars: 72
Lstm: 2 layers, 128 input
Epochs: 10
Sequence length: 9

Sample Output:

sheldon: hello, the tinet an accidence that substage of start the scientist? stay.
sheldon: why not entitled new of science.
sheldon: i don’t take you hear it.
sheldon: i can’t be an astroied the what sastent the startic new night.
sheldon: oh, there’s in the on start the exploto becusely something the she a think the in a tending start th
sheldon: it’s an explodicted hat to space to be more part to go back to the pretty.
sheldon: i’m sorry, if you didn’t want to pretent of your conite science for me point sure i be leonard of th

Wasserstein Generative Adversarial Networks (WGAN)

- Traditional GAN [1]
  - GANs are trained to minimize the distance between the generated and true data distributions. Jensen-Shannon divergence was used as this distance metric.

$$JSD(\mathbb{P}_r \| \mathbb{P}_g) = \frac{1}{2} KL(\mathbb{P}_r \| \mathbb{P}_A) + \frac{1}{2} KL(\mathbb{P}_g \| \mathbb{P}_A)$$

  - Training is not stable: G's updates get worse as the D gets better
  - Objective function is not continuous with respect to G parameters [2]

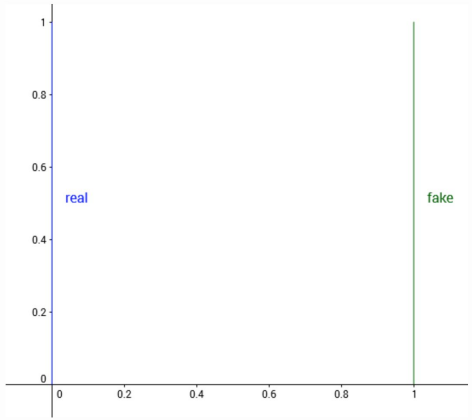
- Improved WGAN
  - Using Wasserstein metric (Earth-Mover distance) [3][4]

$$W(P_r, P_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_\theta}[f(x)]$$

Generator loss: $L_G = -\mathbb{E}_{\tilde{x} \sim \mathbb{P}_g}[D(\tilde{x})]$

Discriminator loss: $L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}.$

Why WGAN? - Continuous gradient



Real and fake distribution when $\theta = 1$

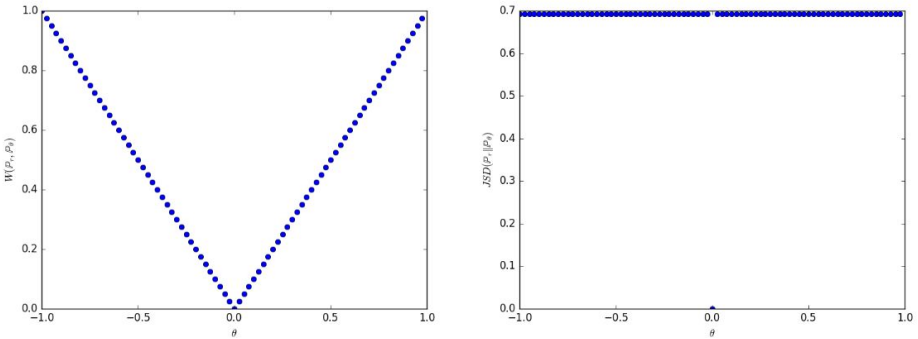


Figure 1: These plots show $\rho(\mathbb{P}_\theta, \mathbb{P}_0)$ as a function of θ when ρ is the EM distance (left plot) or the JS divergence (right plot). The EM plot is continuous and provides a usable gradient everywhere. The JS plot is not continuous and does not provide a usable gradient.

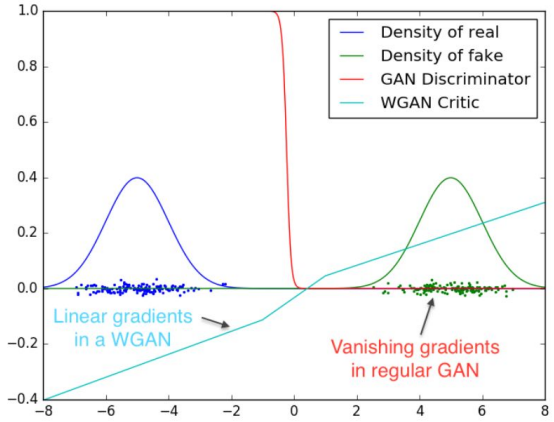
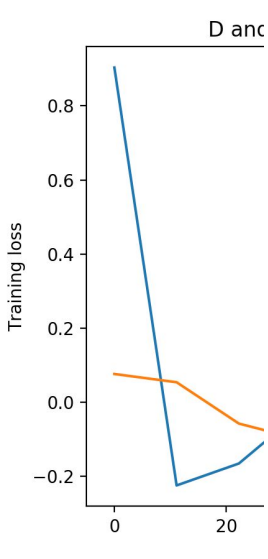
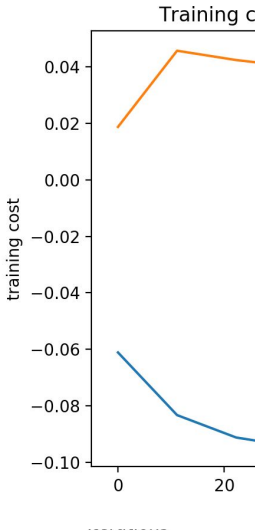


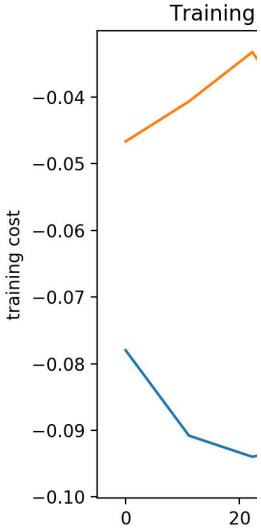
Figure 2: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the traditional GAN discriminator saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

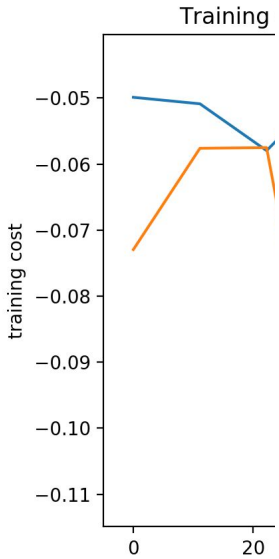
WGAN Results

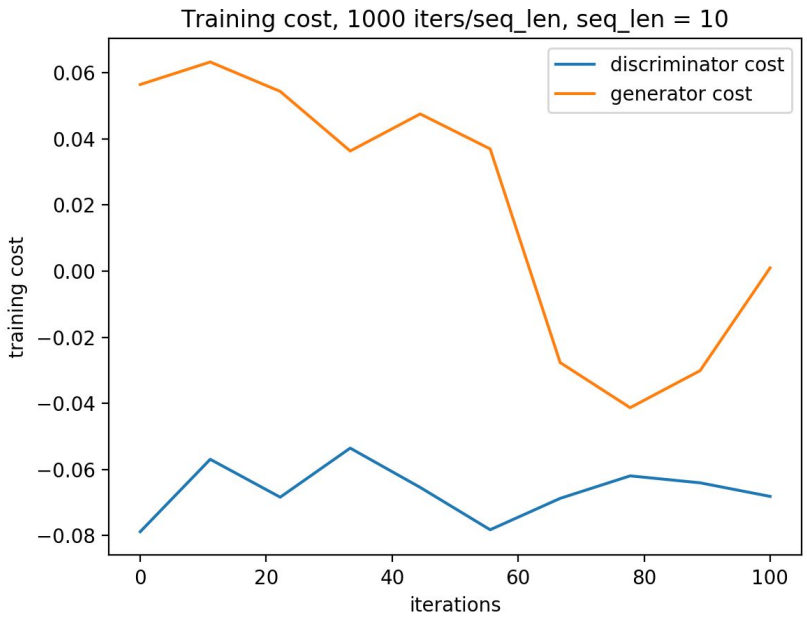
- Setup: Char-level model, GRU 1 layer 512 hidden states for G and D, GAN iteration: 50, iterations/seq = 100, seq_length = 80
- https://github.com/amirbar/rnn.wgan [5]











WGAN Training Problems

- D is strong and G learns slowly

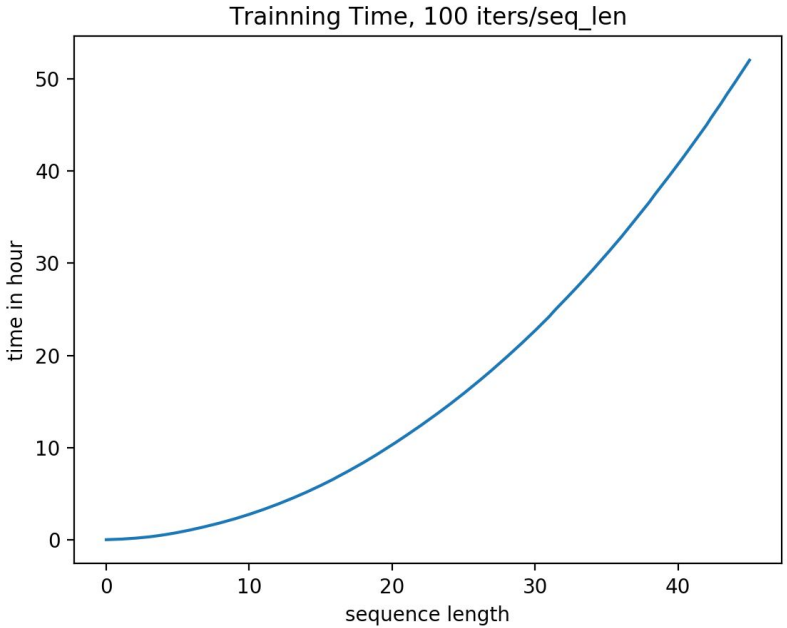
Real sample

An ac||
Oh, t||
Okay,||
Your ||
A hus||

Generated sample:

Tahns||
Nt.n ||
unkunkunkunkS||
unk Welu||
unkunk Buh||

- How many iterations per char ???
  Takes time to train the critic converge

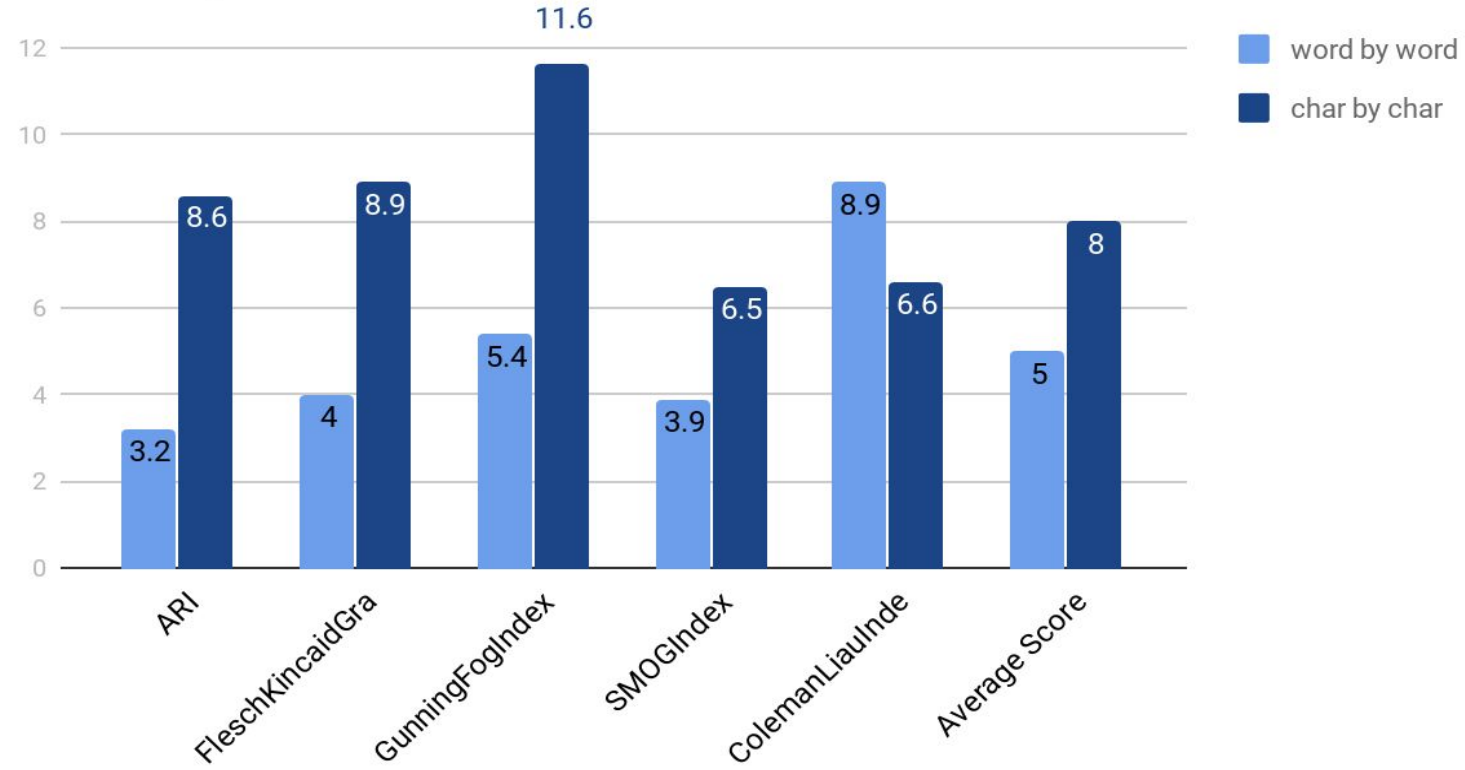


Evaluation

Readability describes the ease with which a document can be read. There exist many different tests to calculate readability. The test output is an approximate representation of the U.S grade level /years of education needed to comprehend a text.

Automated Readability Index (ARI)	$4.71\left(\frac{\text{characters}}{\text{words}}\right)+0.5\left(\frac{\text{words}}{\text{sentences}}\right)-21.43$
Flesch Reading Ease	$206.835-1.015\left(\frac{\text{total words}}{\text{total sentences}}\right)-84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$
FleschKincaid Grade Level	$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right)+11.8\left(\frac{\text{total syllables}}{\text{total words}}\right)-15.59$
Gunning Fog Index	$0.4\left[\left(\frac{\text{words}}{\text{sentences}}\right)+100\left(\frac{\text{complex words}}{\text{words}}\right)\right]$
SMOG index	$\text{grade}=1.0430\sqrt{\text{number of polysyllables}\times\frac{30}{\text{number of sentences}}}+3.1291$
Coleman-Liau Index	$CLI=0.0588L-0.296S-15.8$

Readability Score



Discussion

- LSTM is current state-of-the-art approach for language generation, it is efficient compared to GAN

- GAN has achieved state-of-the-art results in image generation
- We expect GAN performs better because when training the data, generate cost is taken into account, however, GAN computational cost is too high.
- If we can have more time to train the data, we are expecting to see better result of GAN.
- Future work: word based GAN

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie , Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozairy , Aaron Courville, Yoshua Bengioz, Generative Adversarial Nets

[2] Martin Arjovsky and L´eon Bottou. Towards principled methods for training generative adversarial networks. January 2017, arXiv:1701.04862v1

[3] Martin Arjovsky, Soumith Chintala, and L´eon Bottou, Wasserstein GAN

[4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, Improved Training of Wasserstein GAN

[5] Ofir Press, Amir Bar, Ben Bogin, Jonathan Berant, Lior Wolf "Language Generation with Recurrent Generative Adversarial Networks without Pre-training" January 2017, rxiv.org/abs/1706.01399