



A Study of Data Augmentation: SeqGAN

Matéria: Tópicos avançados em Inteligência Computacional

Autor: Matheus Muniz Damasco

Professores: Heder Soares e Alex Borges Vieira





Motivação

Com o avanço das LLM's e da área de NLP, técnicas de Data Augmentation se tornam cada vez mais essenciais, pois os dados que temos disponíveis para treino de LLM's já estão próximos ao limite (Algumas pessoas dizem que eles já acabaram). Dentro das técnicas de Data Augmentation achei interessante o uso de GAN's para Data Augmentation mesmo que as GAN's enfrentem problemas com dados discretos que é o caso de dados textuais. Nesse estudo focaremos na SeqGAN, que pelo meu estudo foi a primeira GAN feita para gerar sequências de tokens discretos.



Problemas

Dificuldade com Dados Discretos:

As GAN's foram originalmente criadas para gerar dados contínuos, como imagens. No entanto, ao serem aplicadas a dados discretos, como sequências de tokens (ex.: texto), a abordagem tradicional não funciona bem. Isso acontece porque as GAN's geram dados aleatórios e depois ajustam esse valor de forma contínua, o que não é aplicável a tokens discretos, já que não há uma correspondência direta para essas pequenas mudanças no vocabulário limitado (não podemos fazer ajustes contínuos em palavras como fazemos com números, pois o vocabulário de palavras é finito).

Avaliação de Sequências:

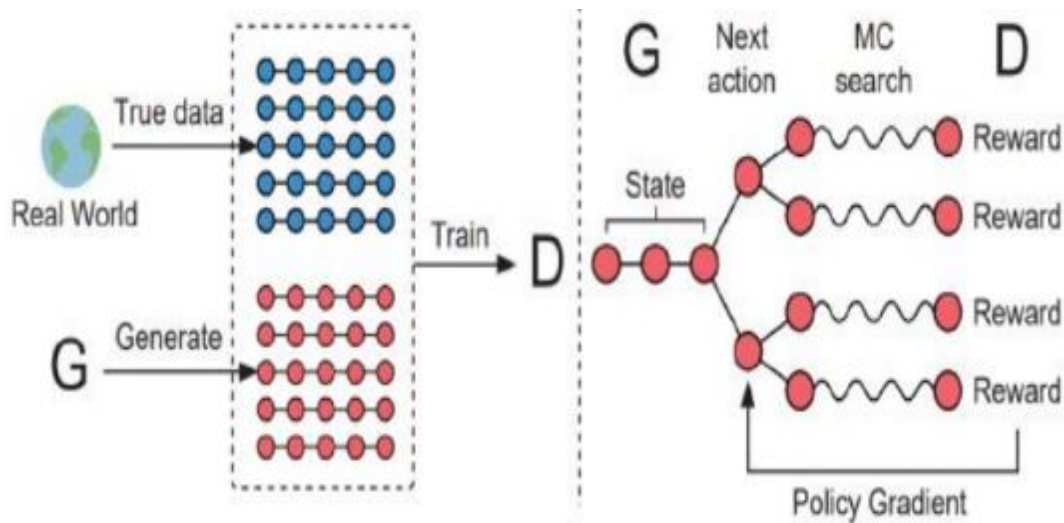
As GAN's geralmente avaliam a qualidade (pontuação/perda) de uma sequência apenas quando ela está completamente gerada. No entanto, durante a criação de uma sequência, a rede tem dificuldade em avaliar se a parte gerada até o momento está boa e como isso pode impactar a qualidade final do texto. Isso torna difícil balancear a avaliação entre a qualidade atual da sequência e seu potencial para gerar um bom texto no futuro, dificultando a avaliação intermediária durante o processo de geração.



Solução da SeqGAN

Para resolver os dois problemas mencionados, a SeqGAN considera o procedimento de geração de sequência como um processo de tomada de decisão sequencial. O modelo gerador é tratado como um agente de aprendizado por reforço (RL), onde o estado são os tokens gerados até o momento e a ação é o próximo token a ser gerado. Um discriminador é usado para avaliar a sequência e fornecer feedback, orientando o aprendizado do modelo gerador. Para superar o problema de que o gradiente não pode ser transmitido de volta para o modelo gerador quando a saída é discreta, tratamos o modelo gerador como uma política estocástica parametrizada. No policy gradient, é utilizada a busca Monte Carlo para aproximar o valor de ação-estado. O modelo gerador é treinado diretamente via policy gradient, o que resolve naturalmente a dificuldade de diferenciação para dados discretos em uma GAN convencional.

Arquitetura da SeqGAN



Arquitetura do SeqGAN.

Esquerda: D é treinado sobre os dados reais e os dados gerados por G.

Direita: G é treinado pelo Policy Gradient onde o sinal de recompensa final é fornecido por D e é passado de volta para o valor de ação intermediário via Monte Carlo Search.

Dataset

- COCO Image Captions: O COCO (Common Objects in Context) é um dataset em grande escala utilizado em pesquisa de visão computacional, focado em detecção, segmentação e legendagem de objetos. Ele é amplamente usado para treinar e avaliar modelos de última geração, oferecendo uma grande variedade de imagens e anotações, sendo bem organizado e fácil de usar para pesquisadores.

A man wearing a shirt with paint splatters



A baby calf looks up over at the adult cow in the middle of the pasture.



A group of people stand on skis and snowboards.



The huge twin engine airliner has red, blue, and orange paint.





Resultados dos Textos Gerados

Exemplos da amostra de treinamento

- # 0 Exemplo: A street in front of an airport with people going in the entrance .
- # 1 Exemplo: Blue and green bird sitting on the ground next to plants .
- # 2 Exemplo: A racer is sitting on a horse in the gates .
- # 3 Exemplo: A man skiing a jump , while in the background there is snow melting off the slopes .
- # 4 Exemplo: A man that is holding a knife and a pot with broccoli .

Exemplos de textos gerados com base em MLE

- # 0 Exemplo: The train traveling down the track on town track just set out of the park .
- # 1 Exemplo: The fence sitting inside the bathroom surrounded by trees and tube .
- # 2 Exemplo: A train moving tracks next to a wooden size platform in the country in a dimly environment .
- # 3 Exemplo: A woman on the tennis court preparing to hit with a bat .
- # 4 Exemplo: A cat is sitting on a train along with many other riders ' s figurine .

Exemplos de textos gerados com base no SeqGAN

- # 0 Exemplo: The street scene filled with broccoli all and a box of dessert .
- # 1 Exemplo: A tabby cat sitting on top of a table full of tongs .
- # 2 Exemplo: A young girl flying three kite in the kitchen to look .
- # 3 Exemplo: A line of people playing a game with their square ' s leg in the water .
- # 4 Exemplo: A lone dog on the couch against a toilet window above brush .



Resultados com base na métrica BLEU

Avaliação dos exemplos da amostra de treinamento com base no BLEU

```
--- ---  
2 -gram BLEU score : 0.5499117864293881  
--- ---  
3 -gram BLEU score : 0.43699142793369117  
--- ---  
4 -gram BLEU score : 0.3173224414085977  
--- ---  
5 -gram BLEU score : 0.21083373415979104
```

Avaliação da qualidade da geração de texto baseada em BLEU após treinamento com MLE

```
--- ---  
2 -gram BLEU score : 0.5226959797212712  
--- ---  
3 -gram BLEU score : 0.3826083717484991  
--- ---  
4 -gram BLEU score : 0.24742168834243194  
--- ---  
5 -gram BLEU score : 0.1487098461666625
```

Avaliação da qualidade da geração de texto baseada em BLEU após treinamento com SeqGAN

```
--- ---  
2 -gram BLEU score : 0.5124317667785041  
--- ---  
3 -gram BLEU score : 0.3824308100492639  
--- ---  
4 -gram BLEU score : 0.26082058408289294  
--- ---  
5 -gram BLEU score : 0.16414649785572483
```




Refêrencias

- [SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient](#)
- [Dataset: COCO Image Captions](#)
- [Github: Generation-of-short-texts-SeqGAN](#)
- [My Code: PyTorch SeqGAN](#)