情報基盤工学講座 小野田成晃

# 特許情報の自動生成

# 前回の課題

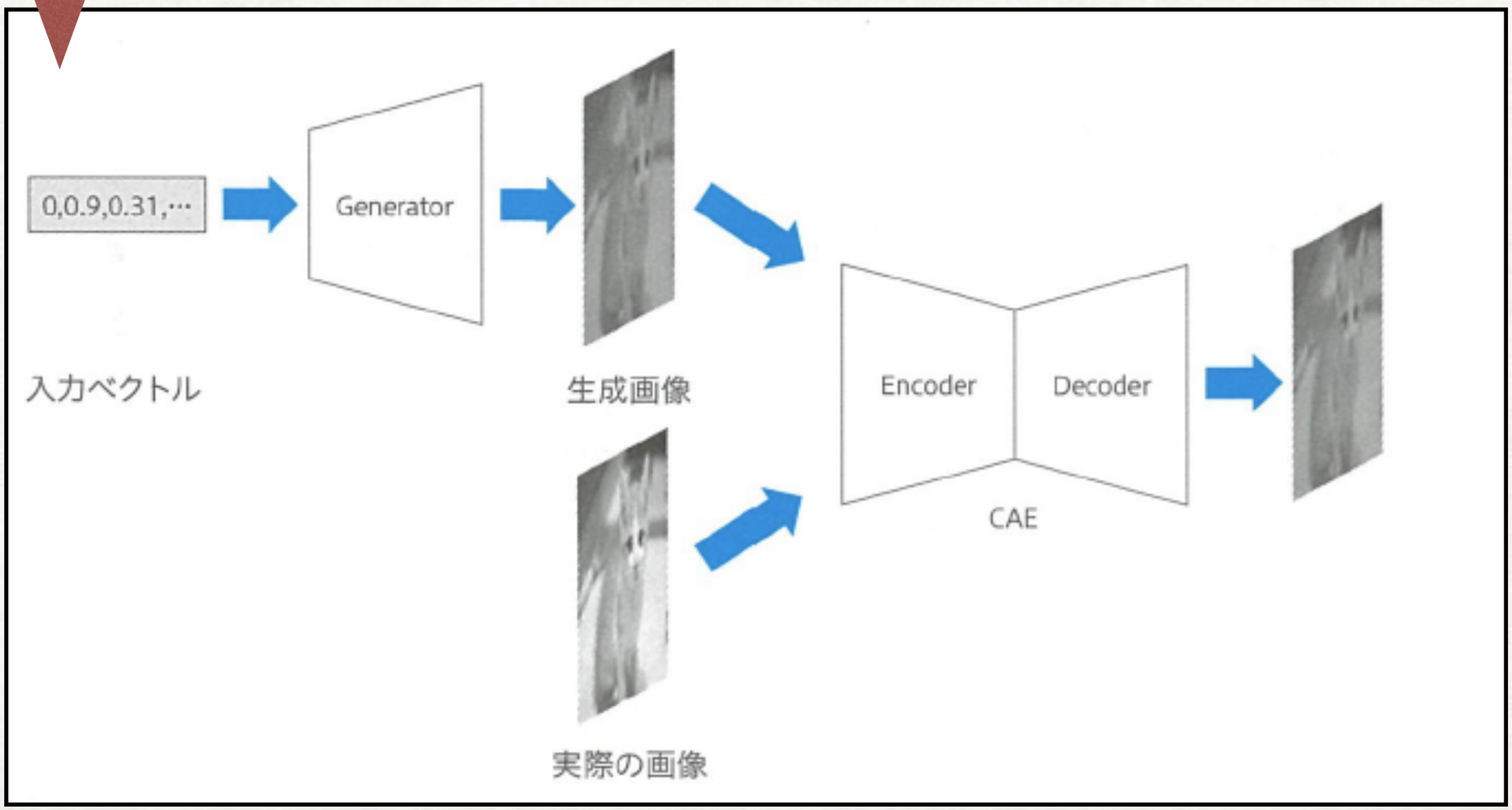
## DONEC QUIS NUNC

- seqGANの入力
- seqGANのエラー
- 系列変換モデル

# BEGANの入力

## QUIS NUNC

ここには乱数で初期化された
疑似画像特徴量が入る



# なぜGANではいけないのか？

## DONEC QUIS NUNC

- １）離散値の系列データを扱うのが困難。生成モデルGのパラメータの更新に識別モデルDの勾配を用いているが、"微小な"勾配により更新された生成モデルGに対応する出力値が離散のため存在しない可能性があるため

- ２）score/lossは系列データ全体に対してのみ考えられていて、部分的な系列データに対しては、現在と将来のscoreのバランスを考慮しなければならないので自明でない＝＞時系列を考慮できない

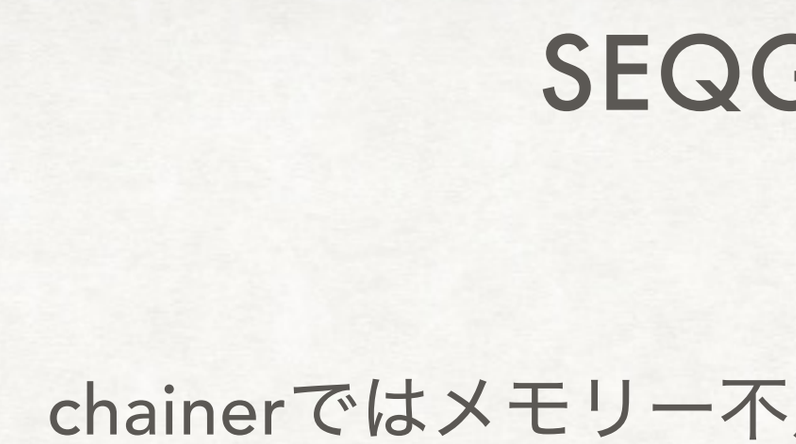
# SEQGANのエラー

## DONEC QUIS NUNC

- 用意されたソースがpython2.7で旧バージョンのchainerだったのでソースを変換した
- ＝＞しかしout of memory がでてきてエポック100以上で実行できない状況になる
- やはり一から実装する必要がでてきた

# SEQGANの実装について

- chainerではメモリー不足のエラーを掃くのでtensorflowかpytroch







- 特許文書データを分かち書きせずに入力（まずは要約のみ）
- そこでランダムなベクトルで生成した疑似特許文章と正しい特許文章を交互に学習
- 従来研究と異なるのは強化学習部の報酬の与え方に特許の価値を加味する必要がある

0 1 0 0 0 0
0 0 0 0 1 0
0 0 1 0 0 0
0 0 0 0 1 0

初期値は適当な乱数

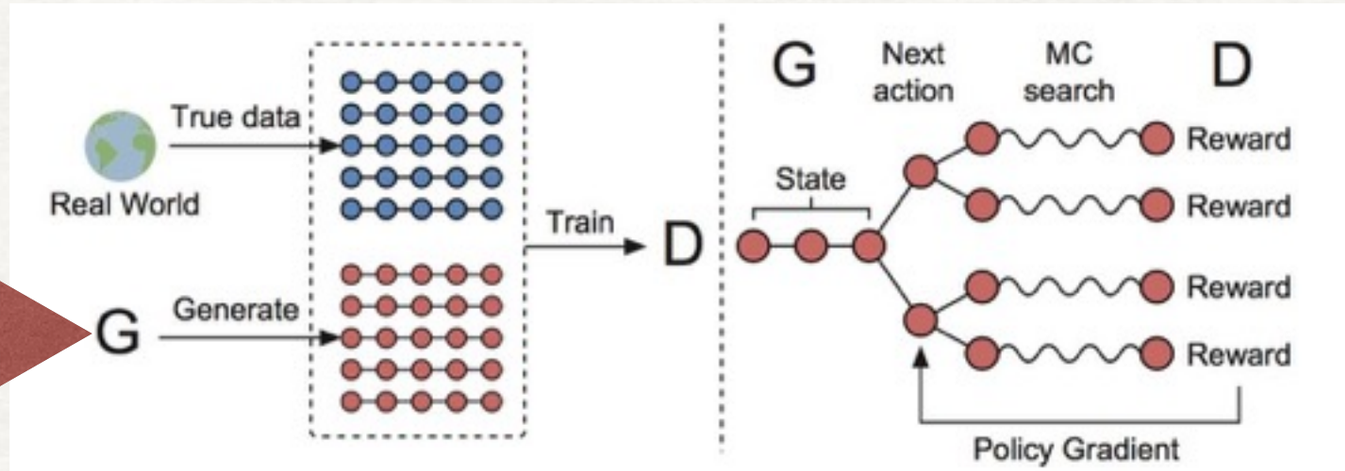



Figure 1: The illustration of SeqGAN. Left: $D$ is trained over the real data and the generated data by $G$. Right: $G$ is trained by policy gradient where the final reward signal is provided by $D$ and is passed back to the intermediate action value via Monte Carlo search.

# アナロジー

- 結局現状のモデルはNNと制御アルゴリズム（強化学習）を組み合わせたものが主流であった
- 実際seqGANは最尤法で最適化した言語モデルで作成したテキストより高い評価をだしている
- 最終的にはseqGANとは違うNN言語モデルを生成する必要があるが勉強のためまずはseqGANで再現したい
- ＝＞検証時の比較にも使えるため

# おわりに

- seqGANを動画で説明する講座があったので
- それを実装しながら閲覧する


- １０月までに特許版実装