Wine Review Generation with Generative Adversarial Networks

Brandon, Palmer, Nate, Ben

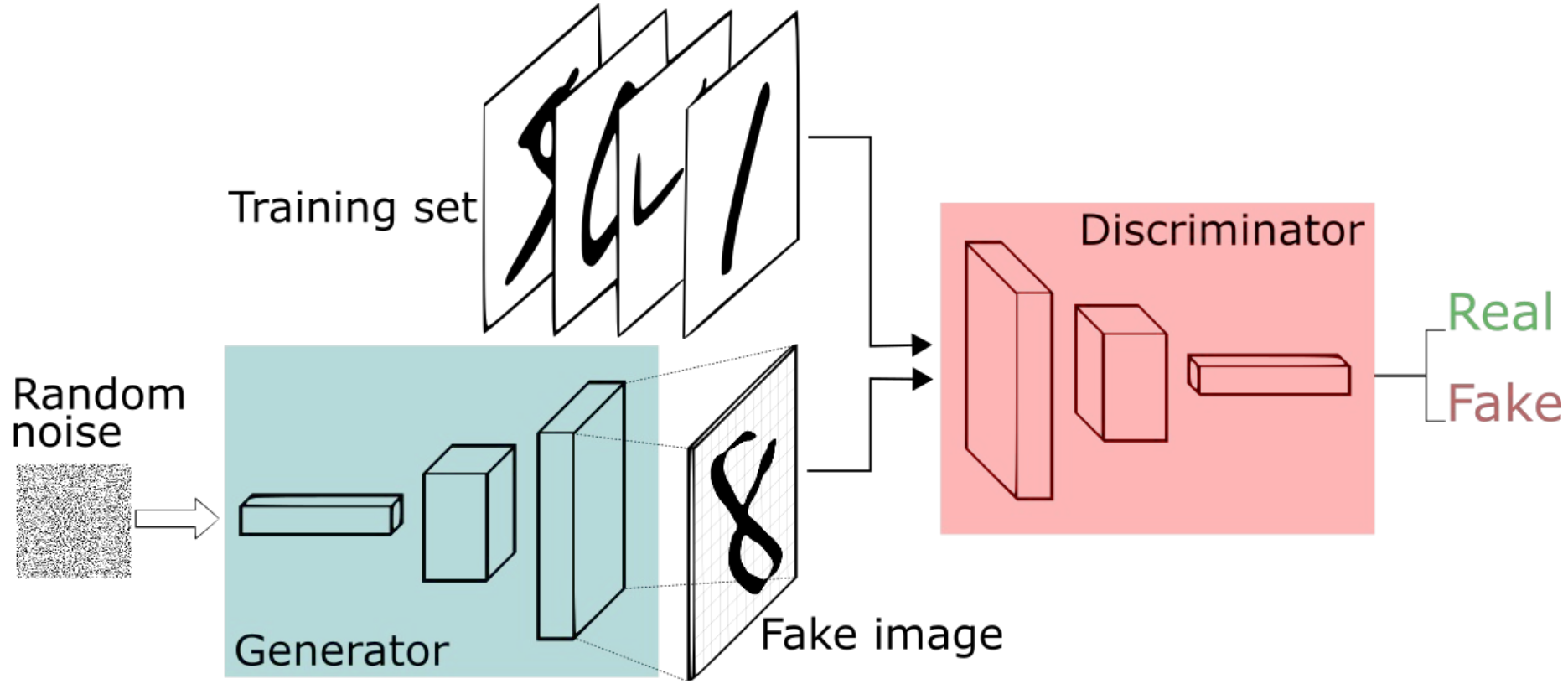
Why GANs?

- A generative adversarial network (GAN), introduced in 2014 by Ian J. Goodfellow, has been described as “the most interesting idea in the last ten years in machine learning”
- Since their inception, GANs have had remarkable success, producing extremely accurate results in the realm of image generation
- Bleeding edge of ML (**TODO**: elaborate)

Edmond de Belamy, 2018
Created by a GAN and sold for $432,500

How do GANs Work?

- A GAN consists of two contesting neural networks operating in a zero-sum game:
  - The **generative network** is trained to generate data with the objective of ‘fooling’ the discriminator
  - The **discriminative network** is trained to discriminate generator-produced data from real data
- The discriminator is initially trained on a known dataset
- The generator trains based on whether it succeeds in fooling the discriminator
- This cycle of generation and discrimination is repeated until the generative network can produce believable results


Training set
Discriminator
Real
Fake
Random noise
Generator
Fake image

Our Question

- GANs are primarily being applied to image generation, so we would like to see how they perform for natural language processing tasks
- In particular, can GANs be used to effectively generate text?
- We propose the construction of a GAN which can produce realistic, human-quality wine reviews
- **TODO**: Illustrative example of the problem?
- **TODO**: Formal definition of the problem?

What approaches have people taken in the past?

- Recurrent neural networks trained to generate text with GAN's using curriculum learning
  - Evaluation Metric was the proportion of word n-grams from generated sequences that also appear in a test set
  - Evaluation Metric: 87.5% with added extensions
- Sequence generation framework to generate new sentences which they call SeqGAN
  - Experiment Chinese Poem Generation
  - evaluation metric of humans judging the quality of the poems
  - Maximum Likelihood Model achieved a human score of 0.4165 while the SeqGAN achieved a human score of 0.5356
- RankGAN analyzes and ranks a collection of human-written and machine-written sentences by giving a reference group
  - Experiment Chinese Poem Generation
  - Compared to the models MLE and SeqGAN the RankGAN model achieved the highest BLEU-2 score of 0.812

Interesting Aspects of Project

- Using Keras to build the generator and discriminator
- No need to design a loss function since GAN will figure out it’s own evaluation rules
- Adding extensions to the GAN to improve performance
- GANs are similar to how humans learn!
    - **TODO**: Elaborate

The Data

- **TODO**: What kind of data is available for this problem?
  - Kaggle data
- **TODO**: Evaluation metric
  - Perplexity

Baselines

- **TODO**: Discuss performance of N-Gram baseline
  - Sample outputs
  - Perplexity scores

Our Model: Approach

- **TODO**: What did we implement for the published baseline
  - GAN model which uses a LSTM as the generator, a different LSTM as the discriminator, and WGAN-GP (wasserstein GAN, with a gradient penalty) as the objective
  - Source: Sai, et al. “Adversarial Generation of Natural Language.”

Our Model: Initial Performance

- **TODO**: Performance of models without extensions

Our Model: Extensions

- **TODO**: Which extensions did we implement?

Our Model: Performance with Extensions

- **TODO**: How did our model perform with extensions compared to before?

How does this topic relate to class?

- Text generation has been addressed before in class
  - NGram model
  - Neural Net language models
- GANs are just another way to generate text

What did we learn?

- How to use Keras
- How to train a generator and discriminator
- How to connect those two into a GAN structure
- How to clean a dataset
- How to add extensions to a GAN