Theory Exercise

This theoretical task is going to be about Generative Adversarial Networks, which are affectionately known as GAN. As you might know already that GANs have created a lot of buzz in the ML and DL community lately. But most of us don't know what is the real reason behind this hype. Are GANs really useful? Is it a novel idea? Does it even work? Well, these are questions that you will hopefully be able to answer by the end of this exercise.

Task 1

Read and understand the idea behind GANs and try to analyze the hype around it.

The following are some useful links (not all of them are mandatory but are highly recommended):

1. A well written blog describing GANs http://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them
2. Ian Goodfellow talking about GANs at NIPS 2016 https://www.youtube.com/watch?v=AJVyzd0rqdc
3. Recent papers/developments on GANs: http://gkalliatakis.com/blog/delving-deep-into-gans

The video should be enough for you to answer the questions.

Question 1

State the objective function of a GAN (the one proposed by Goodfellow et al.)

Question 2

Describe each term in the objective function of the GAN

Question 3 (optional)

Can you reformulate the objective function of GAN in terms of Categorical Cross-Entropy? Justify your answer.

Question 4

Describe the Generator and Discriminator. Be as formal as possible.

Question 5

What are the problems that GANs face while training. Describe them if any.

Question 6

How can you evaluate GANs? Please provide objective answers.

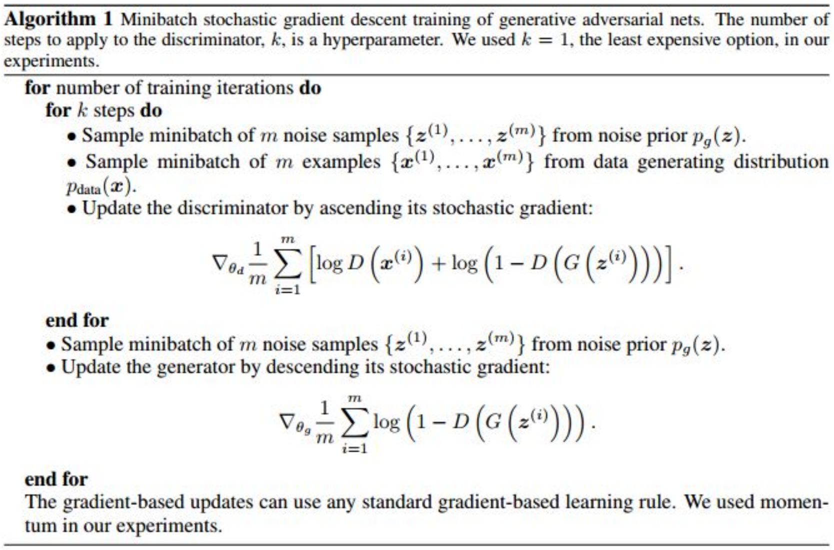
Practical Exercise

Task 1:

1.1. Implement a Vanilla GAN as described by Ian Goodfellow in his first paper : https://arxiv.org/abs/1406.2661 to generate adversarial MNIST images.

You can take help from here : https://wiseodd.github.io/techblog/2016/09/17/gan-tensorflow/

Run the code with the original loss as described by Ian Goodfellow and observe the results :



Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

```
for number of training iterations do
    for $k$ steps do
        • Sample minibatch of $m$ noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
        • Sample minibatch of $m$ examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
        • Update the discriminator by ascending its stochastic gradient:
            $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right].$
    end for
    • Sample minibatch of $m$ noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
    • Update the generator by descending its stochastic gradient:
        $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right).$
end for
```

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

1.2. Run the same code with a different loss functions : **Logistic loss** as described in Brandon Amos blog (http://bamos.github.io/2016/08/09/deep-completion/) and compare the results with above (1.1.)

1.3. Run the code for **20k iterations** and then for **100k iterations** and observe the results in both cases. How/why is the output different for both the cases ? Try to find a suitable reason for both .

Also, try to find how the results can be improved.

Check out this video from Soumith Chintala to describing how to train and improve GANs : https://www.youtube.com/watch?v=myGAju4L7O8

1.4. **OPTIONAL**: Using the concept of conditional GANs, try to generate a particular output class image : (8 digit for example in case of MNIST).

HINT: Just change 2 lines of code in previous code

Task 2:

2.1. Train a CNN MNIST model and create adversarial images to classify 4s as 9s. (**VDL Experts**: Use the model already trained in Exercise 1)

HINT: Take help from Jasoni Carter Github : https://github.com/jasonicarter/MNIST-adversarial-images

Try to reason how these adversarial images are produced.

2.2. Now use a random noise image to classify it as 9 and observe the results. Compare the results with 2.1.

2.3. Use a zero image (matrix with all values 0) and classify it as 9. Observe the activations and the results and compare the result with above observations.

Optional Task:

3. **OPTIONAL FOR EXPERTS**: Try other GAN methods like DCGANs with other datasets like Celeb dataset and observe the results.

https://github.com/jasonicarter/MNIST-adversarial-images

Acknowledgement

We would like to thank **Saurabh Varshneya** and **Kumar Shridhar** for making this very interesting exercise on GANs. For any queries, contact: **Saurabh Varshneya** (varshney@rhrk.uni-kl.de)