Conditioning in InfoGAN

Gaurav Didwania
160050020

Nitish Joshi
160070017

Ashish Mittal
160050022

Aman Jain
160050034

Abstract

In this project we analyze InfoGAN [1] under additional conditioning on some variables both theorotically as well as empirically. We conducted primary experiments based on Recurrent Conditional GAN (RCGAN) on synthetic time series data and show learnt disentangled features.

1 Introduction

Unsupervised learning can be described as the general problem of extracting value from unlabelled data which exists in vast quantities. A popular framework for unsupervised learning is that of representation learning whose goal is to use unlabelled data to learn a representation that exposes important semantic features as easily decodable factors. While unsupervised learning is ill-posed because the relevant downstream tasks are unknown at training time, a disentangled representation, one which explicitly represents the salient attributes of a data instance, should be helpful for the relevant but unknown tasks. For example, in a MNIST dataset, the salient attributes can be the thickness of the stroke and the orientation of the digits. A good generative model automatically tends to learn the disentangled features of the data. The most prominent generative models used are Variational Auto Encoders (VAE) [2] and Generative Adversarial Networks [3]. GANs have mostly been used to generate image data but in our paper we try to use GANs to generate time series data based on an input condition and also try to learn some disentangled features of output image.

2 Project Goal

In this project we try to modify the Recurrent Conditional GAN (RCGAN) [4] model to condition the output series on a set of input points such that the output function is enveloped by the input function(Here we fix the input function to be a particular family f(x) = a*exp(-bx). Further we want to add latent codes to input and try to learn the disentangled features of the output series by maximizing the mutual information among the latent codes and output as done in InfoGAN paper.

3 Related Work

There exists a large body of work on unsupervised learning. Since its introduction in 2014, there exist various line of works based on GAN. Though GANs were originally successful with image data, more recent works have extended there use in various other fields like sequence prediction for natural language generation and time series data.

Conditional GANs [5] condition the model on additional information and therefore allow us to direct the data generation process. This approach has been mainly used for image generation tasks. Recently, Conditional GAN architectures have been also used in natural language processing, including translation and dialogue generation, where none of them uses an RNN as the preferred choice for the discriminator and RL approach is used to train the models due to the discrete nature of the data.

Then in 2016, Information Maximizing Generative Adversarial Networks(Info GAN) was introduced which learns both continuous and discrete latent factors using maximization of mutual information among latent codes and generator data. This GAN required no supervision of any kind and learned interpretable and disentangled features on the dataset.

The use of GAN for generating real valued continuous data has been done in 2017 where they generated time series data with recurrent conditional GAN. This model has various medical uses like generating synthetic medical data for research purpose. The model uses RNN (LSTM) for both generator and discriminator.

4 Approach

We initially aimed to model text generation with GAN based on some other conditioning text sequence, along with learning some disentangled features of generated text sequence using latent variables. But we encountered certain issues as follows:

- Since the generation of tokens is via sampling over discrete tokens, end-to-end training is very difficult since it requires passing the gradient update from discriminator to generator during backpropogation. The approach usually taken to tackle this [6] relies on using Mote-Carlo based reinforcement learning methods.
- The above methods are very sensitive to hyperparameters and our known to not work well on text data. We tried SeqGAN based approaches on sythetic data (using an oracle generator) but it hinders the primary aim to learn to interpretable features for generation.

Therefore we chose to instead work on real valued time series data which also requires sequence prediction and apply the logic of conditional GAN and InfoGAN models in time series data generation.

4.1 Variational Mutual Information under Conditioning

Before we incorporate the idea of conditioning in InfoGAN, we theoretically verify that all the equations and learning objectives hold true under the additional conditional variable. Let c denote the extra latent codes in InfoGAN, y the conditioning variable and $V_y(G, D)$ be the conditional-GAN objective. Then our new objective function can be written as:

$$\min_G \max_D V_I(D, G) = V_y(D, G) - \lambda I(c; G(z, c|y))$$

Since $I(c; G(z, c|y))$ is hard to optimize directly, we can obtain a lower bound using variational information maximization:

$$\begin{aligned}
I(c; G(z, c|y)) &= H(c) - H(c|G(z, c|y)) \\
&= \mathbb{E}_{x\sim G(z,c|y)}\left[\mathbb{E}_{c'\sim P(c|x,y)}\left[\log P(c^{'}|x, y)\right]\right] + H(c) \\
&= \mathbb{E}_{x\sim G(z,c|y)}\left[\mathbb{E}_{c'\sim P(c|x,y)}\left[\log Q(c^{'}|x, y) + D_{KL}(P(.|x, y), Q(.|x, y))\right]\right] + H(c) \\
&\geq \mathbb{E}_{x\sim G(z,c|y)}\left[\mathbb{E}_{c'\sim P(c|x,y)}\left[\log Q(c^{'}|x, y))\right]\right] + H(c)
\end{aligned}$$

where the last inequality follows from the fact that KL-divergence is non-negative. We now maximimze the lower bound as given in the InfoGAN paper where we use a neural network $Q(c|x, y)$ to estimate the posterior distribution on c.

5 Experiments

Link to our Code. The base code can be found here.

5.1 MNIST

We first experimented with MNIST dataset to see if GAN is able to learn some disentangled features from the images and also condition the generation of digit based on an input condition. We conditioned the images by specifying even or odd but instead of giving a binomial condition variable, we provided the image of an even or odd number instead as the condition variable. We achieved some visible results where barring a few cases output was equivalent to input modulo 2. In first part of the experiment we tried to provide a binomial latent variable aiming to learn whether output is equal to input number or not. This did not give any insightful feature.
The base code for this model can be found here. The code is in Pytorch and we use Geforce rtx 2080 ti TITAN X (Pascal) GPU for training.

5.2 CIFAR

We experimented with both CIFAR-10 and CIFAR-100 to generate images with conditional GAN model but the model was unable to generate satisfactory images that could be classified by human eyes into specific classes to check if the condition variable is able to control the output image generation. We tried different models for GAN without notable improvement in image quality and also tried to interpolate the CIFAR image to get 64*64 images and then pass them as input to GAN.

5.3 LEGO

This dataset had objects with sharp edges. The trained GAN failed to generate sharp images for this dataset.

5.4 Time series data

We started with the following paper [4] as the base implementation and we tried to extend it by adding infogan loss and conditional inputs which was not used in this paper for time series data. Further we also used latent variables to learn disentangled features of the generated time series (in our case it is a sine function). In this the code is written in tensorflow. The base code can be found on this link. We have modified/written about 200-300 lines of code. Our code can be found here. We used two Nvidia-1080 Ti GPUs for our experiments.

5.4.1 Learning Disentangled Features without conditioning

We first started with a single latent variable but notable results were obtained with three latent variables. We expected the model to learn three visibly interpretable features of the time series data which are amplitude, frequency and phase of the wave. We tried two different cases:

- Case : All three variables were distributed uniformly in (0,1)
  This approach allows model to learn both the mapping from (0,1) to domain of feature the latent variable learns to represent. This gives more freedom to the model to map any latent variable to any disentangled feature.
  Here we find that only latent variable c2 learns (to some extent) disentangled representation of frequency but c1 and c3 failed to learn any interpretable parameter.
- Case : Each variable is distributed uniformly in its own expected domain
  Here we tried to restrict which latent variable learns which feature of the output wave. The following variables correspond to the given feature.
  - c1 -> amplitude : range [0.1,0.9]
  - c2 -> frequency : range [1,5]
  - c3 -> phase : range $[-\pi,\pi]$

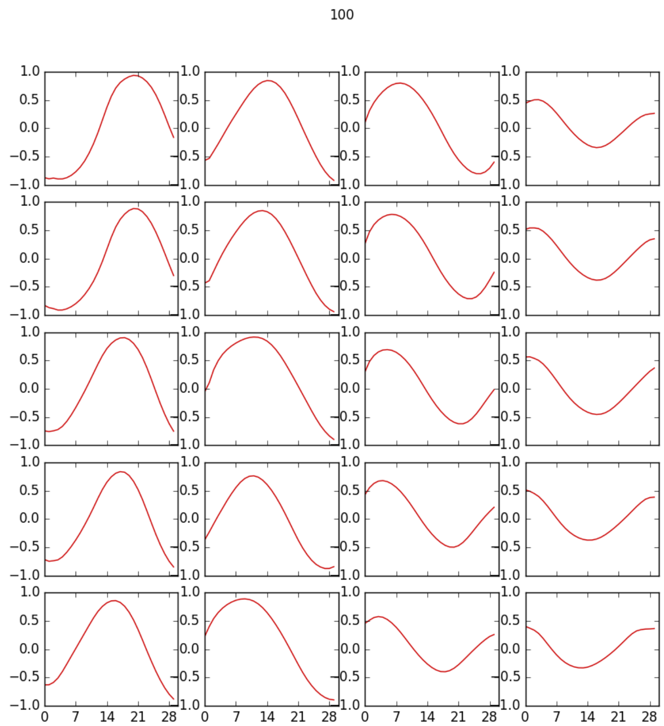

Figure 1: Images without conditioning generated with c1 = 0.1, c2 = 1.0, c3 $\in (-\pi, +\pi)$

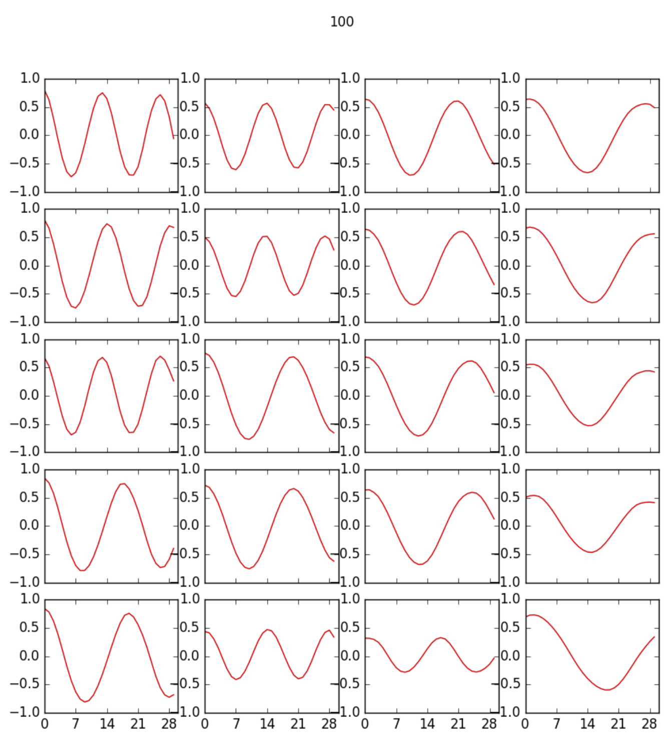

Figure 2: Images without conditioning generated with c1 = 0.1, c2 $\in (1.0, 5.0)$, c3 = $\pi/2$

5.4.2 Learning Disentangled Features with conditioning

Here we introduce a new input variable which is a series of points representing a wave of the form $Ae^{-\alpha x}$ where A and α are parameters. A lies uniformly in range [0.1,0.9] and α belongs to range [1,5]. We pass the conditioned inputs to both the discriminator and the generator where we concatenated conditioned inputs with the noise variable z in the generator and with the input x in the discriminator respectively. The output series is a multiplication of sine wave with the above conditioning wave, i.e the conditioning wave must form an envelope of the generated sine wave. The mathematical form of the generated wave is $Ae^{-\alpha x}\sin{(2\pi f x + p)}$, where **f** is frequency and **p** is phase.

Here too, we used three latent variables as above with different distribution for each variable. Though in this case amplitude is fixed given the condition wave and thus only two latent variables c2 and c3 need to be analyzed.

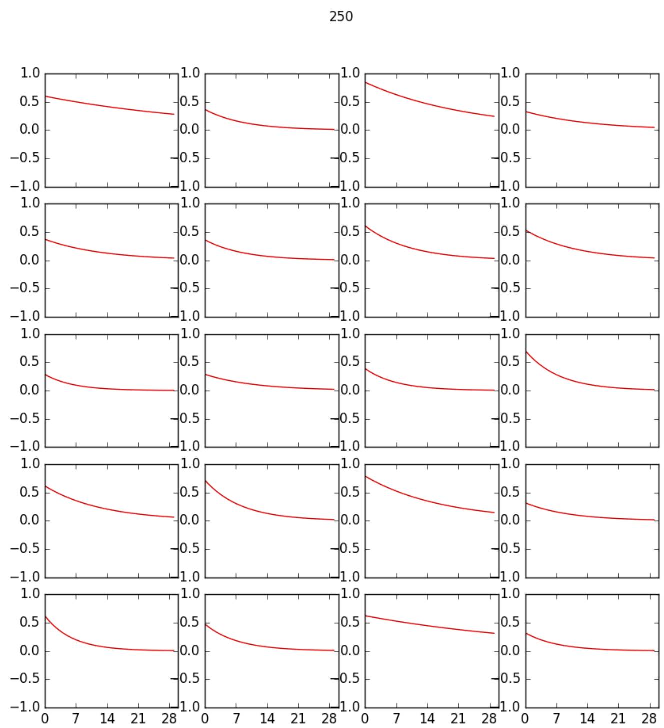

Figure 3: Conditioning images used to generate the images in Figure 4

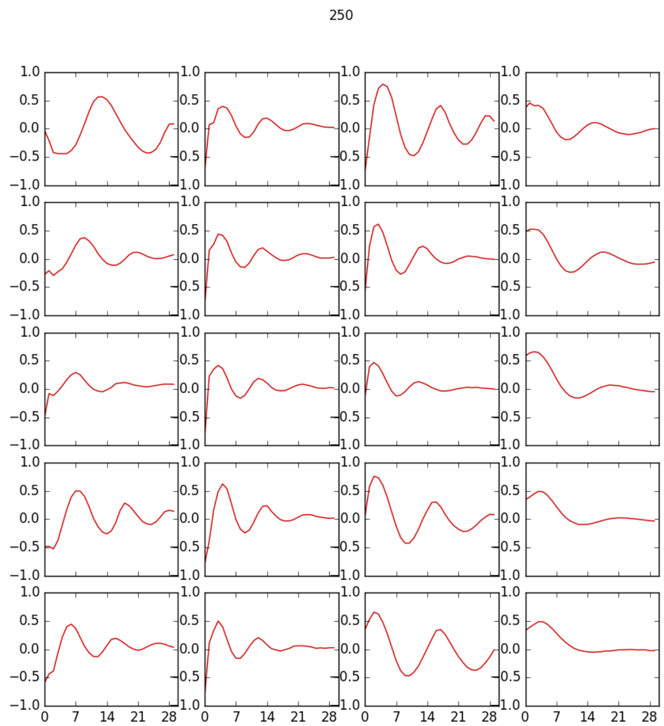

Figure 4: Images with conditioning generated with c1 = 0.9, c2 = 5 ,c3 ∈ ($-\pi$,$+\pi$)

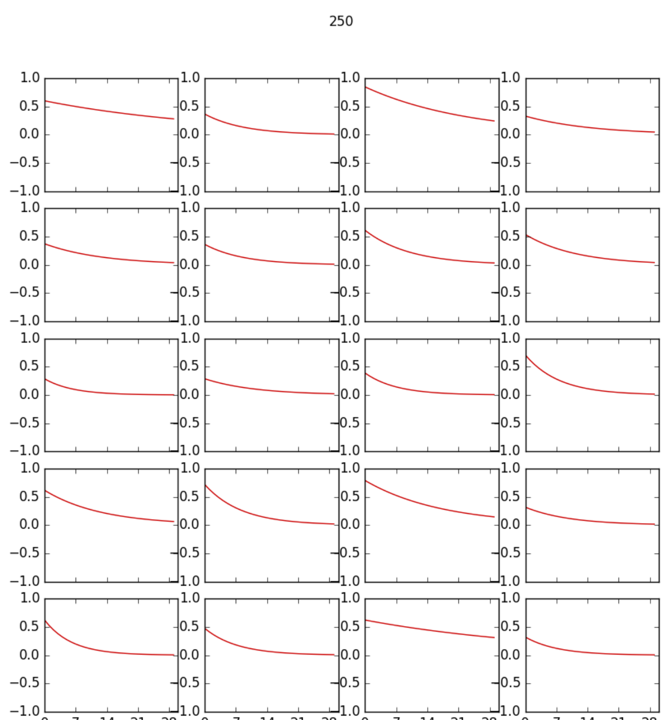

Figure 5: Conditioning images used to generate the images in Figure 6

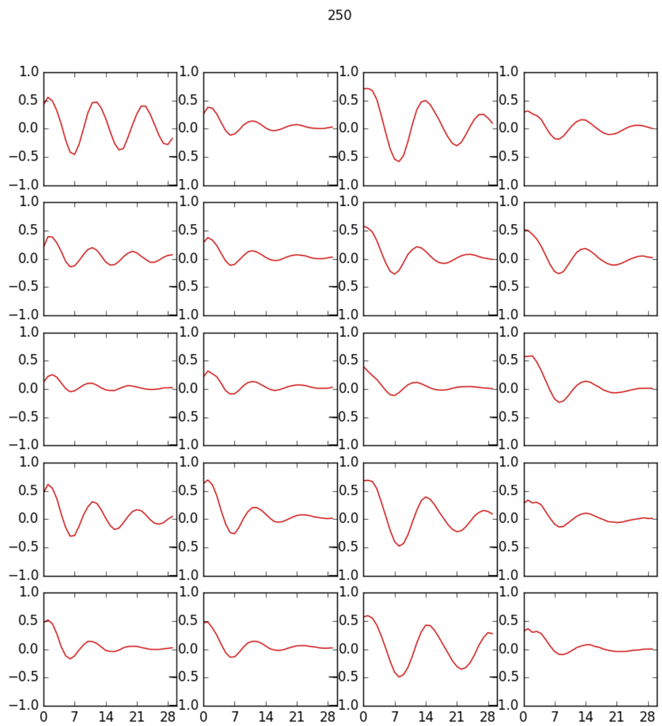

Figure 6: Image with conditioning generated with c1 = 0.9, c2 $\in (1.0, 5.0)$, c3 = $\pi/2$

6 Effort

- Percentage of time spent in different parts:
  - MNIST 10
  - CIFAR and LEGO 20
  - Time Series 70
- Most challenging part was deciding and implementing the conditioning variable in time series data.
- Equal Contribution of all team members.

References

[1] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

[2] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[4] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs. *arXiv e-prints*, 2017.

[5] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[6] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017.