Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project

Project Plan

Predicting the spatial variation of COVID-19 infections using Generative Adversarial neural Networks

Li Yaqi

yaqi.li19@imperial.ac.uk

GitHub login: acse-yl7419

supervised by

Christopher Pain

Claire Heaney

Vinicius L S Silva

June 26, 2020

1 Introduction

In December 2019, the first case of Coronavirus disease 2019 (COVID-19) was found in Wuhan, Hubei province, China[1]. COVID-19 has posed a continuing threat to human health with its high transmission efficiency, severe infection consequences and unpredictable epidemic time. Until now, this disease has spread rapidly to every corner of the world and the total number of COVID-19 cases has reached 9,400,295 by 25 June 2020 according to European Centre for Disease Prevention and Control. This has greatly affected the economy, trade, travel, everyday life etc. At this time, using machine learning methods such as Generative Adversarial Networks (GANs) to predict COVID-19 seems to be very important.

Compartment models are often used for modelling of infectious diseases. (Grant, 2020). Here because COVID-19 has a long latent period when the individuals are in compartment E so SEIR is used to model the dynamics of epidemic disease. This model divides people into four groups: Susceptible (those who are vulnerable to infection), Exposed (those who have been infected but are unable to transmit the disease), Infectious (those who are infectious and may transmit the disease to others), Recovered (recovered individuals) (Prem et al., 2020). R0 is the reproductive number which represents propagation expectation or average outcome of this disease, simply speaking, it is the average number of infected persons that an infectious person can transmit the disease to[2].

GAN is an adversarial process in which the generator tries to generate new examples which want to fool the discriminator and the discriminator tries to distinguish whether the sample is fake or real (Goodfellow et al., 2014). In this case, both models use multilayer perceptrons. In addition the input of the generator is a normal Gaussian random noise and through training, and the output of the generator becomes more similar to real data. This process updates the discriminator parameters by gradient ascent and updates the generator parameters by gradient descent. This process has a global optimum when the probability distribution of real data equals to the probability distribution of the generated output (Cheng, Fang, Pain, & Navon, 2020). However, it can be seen that although GAN beats many other deep learning methods it still remain difficult to train (Arjovsky & Bottou, 2017). For example, the results are very dependent on the architectures of generator and discriminator, require the need for careful adjustment of hyperparameters and have possibility of model collapse.

Besides original GAN, there are many GAN variants such as DCGAN (Radford, Metz, & Chintala, 2015) and StyleGAN (Karras, Laine, & Aila, 2018). DCGAN almost uses the convolutional layers instead of fully connected layers. In addition, almost every layer in generator and discriminator uses batch normalization to accelerate the training and the discriminator utilizes LeakyReLU activation function instead of RELU to prevent gradient sparsity. As for StyleGAN this type of GAN cancels the random noise as the initial input for the generator and replaces it with a trainable constant vector. Another improvement for StyleGAN is to introduce Mapping Network consisting of eight fully connected layers to generate a style vector. Furthermore stochastic variation is introduced by adding Gaussian noise at each point for the generator and adaptive instance normalization (AdaIN) is incorporated into each block of the generator after the convolutional layers.

Wasserstein GAN (WGAN), an improved algorithm of GAN (Arjovsky, Chintala, & Bottou, 2017) is going to be implemented in this project. It has replaced Jensen-Shannon divergence in GAN with Earth-Mover distance in order to solve the problem of gradient vanishing and uses RMSProp optimizer instead of Adam which is based on momentum. Each time the discriminator's parameters are updated, their absolute values are truncated to no more than a fixed constant c, the clipping parameter. Here Wassertein Distance-1 is defined as the distance measurement between two probability distributions in other words it represents the minimum cost of transforming a distribution into

[1]https://en.wikipedia.org/wiki/Coronavirus_disease_2019 last accessed 26 June 2020.
[2]https://www.idmod.org/docs/hiv/model-seir.html last accessed 26 June 2020.

another. WGAN uses a weight clipping strategy to force the Lipschitz constraint in discriminator, and this can sometimes lead to difficulties with convergence. Therefore, another truncating strategy has been introduced - the gradient penalty - which directly has a penalty for the gradient norm of the discriminator with respect to its input (Gulrajani, Ahmed, Arjovsky, Dumoulin, & Courville, 2017).

Data assimilation (DA) is a method of predicting the medium and long-term development trend of practical problems to produce an optimal estimate based on its core, that is short-term field measured data onto corrected numerical model. In this project, data assimilation is needed to get the information about infection state at a point in time in order to predict the future and the past of that point. For example, GAN may choose to generate 9 time levels along one day and can predict the 9th time level after data assimilating the first 8 time levels.

2 Project Plan

2.1 Description

2.1.1 Prepare for the data

The first step is to be familiar with basic knowledge such as GANs and data assimilation. In the data collected from the SEIR compartmental model, there are 3888 time levels and there are 8 people groups (susceptible, exposed, infectious, recovered, and for each of these one group is at home and another is mobile). There is a 10×10 mesh representing the spatial variation of people in part of a very simple city. As a result there are 800 variables at each time step which would be too many to easily train a network. So the data is compressed, by using singular value decomposition (Wall, Rechtsteiner, & Rocha, 2003), from 800 to 15 variables.

2.1.2 Train the WGAN

At this step, in order to implement Wasserstein GAN both network model need to be implemented in functions using multiple layers of neural networks. Then train generator model using randomly generated noise and train discriminator model using real data and output data from the generator. Later calculate the loss function as well as applying optimizer to proceed parameter updates in each epoch. Through training, the trainable parameters of the generator change and the output data from the generator would be more and more similar to real data.

2.1.3 Predict using WGAN model

Load the generator model after the training step and the discriminator model is no longer needed. Data assimilation is combined with WGAN to predict one point or multiple points.

Data assimilation:

- One time step assimilation: first process the compressed real data, which means at each time point selecting 9 time points every 10 time steps after this time point, so now each time point can have 9 consecutive time levels. Meanwhile the generator produces a set of fake data. Then select first 8 time levels to calculate the mean square error. Next use the optimizer to update the original noise thousands of epochs. Consequently, predict the 9th time level.
- A couple of months assimilation: first opt for the last time level of generated fake data from updated noise in the first step then combine it with the first 8 time level real data so now 9

time level mixed data is produced. However the last 8 time level combined data is used as our new real data to train our new time step prediction. Repeat this process several times.

Through these predictions, it is possible to have a general idea of how the virus spreads. Moreover we can also plot the people distribution in each group in future months to have a visualized result.

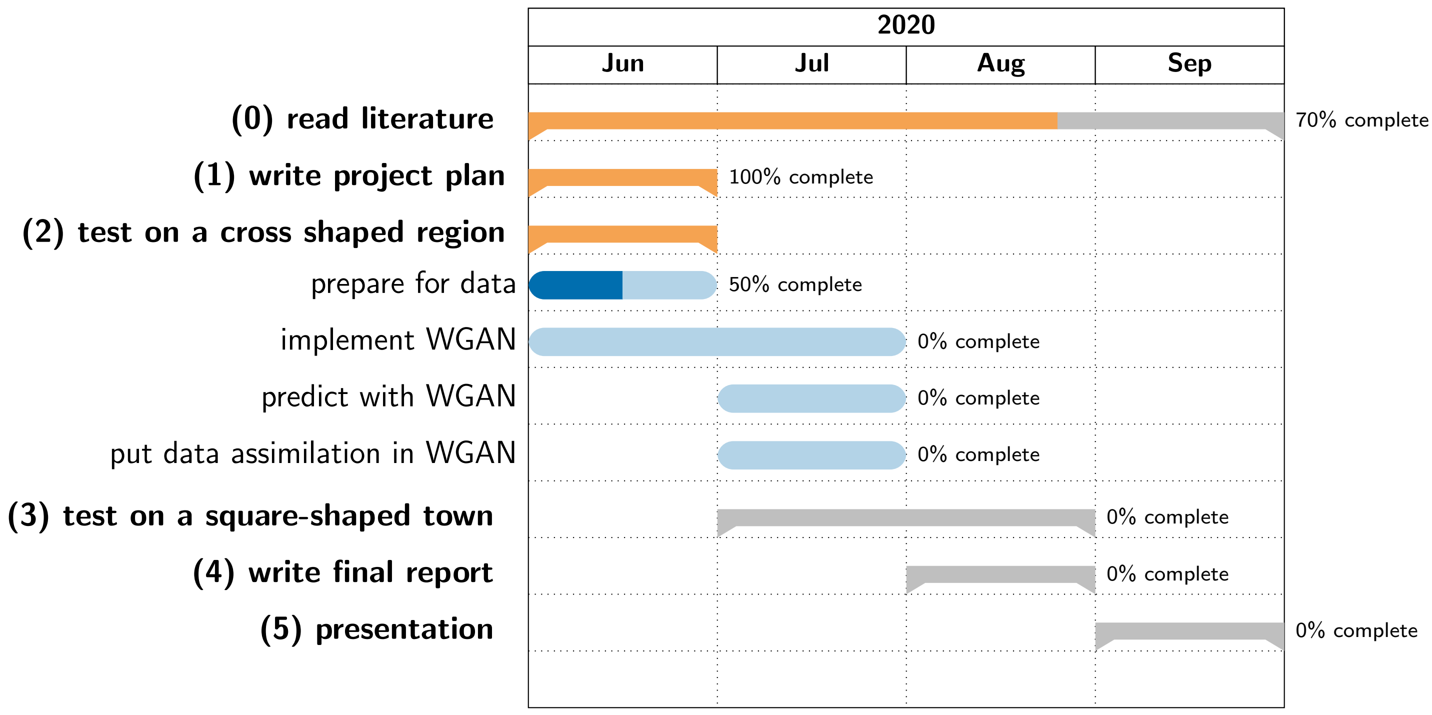
2.1.4 Larger cases

The above procedures are first tested on a very small case that is a cross shaped region. In this project, we also want to apply our model to a larger square-shaped region with more features. For example, in the square-shaped town, it has shops, school, offices and numerous flats which are placed in a random way. It is able to see people movements between homes and other places in this region and predict the total number of people in each group versus time.

2.2 Objectives

The aim of this project is to take results from a compartmental model, which makes predictions on how people move from one compartment or group to another (e.g. there could be a susceptible group etc), and to try to learn this behavior with an adversarial network to predict the distribution of people within the groups in the future. In this case, the compartmental model has been combined with a model which describes the spatial spread of the virus, so the results show spatial variation. As well as predicting the spread of the virus, this project will try to assimilate data to discover the initial condition corresponding to the initial data, and to see how the disease could be controlled.

2.3 Progress to date and schedule



References

Arjovsky, M., & Bottou, L. (2017). *Towards Principled Methods for Training Generative Adversarial Networks.*

Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein GAN.*

Cheng, M., Fang, F., Pain, C., & Navon, I. (2020). Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network. *Computer Methods in Applied Mechanics and Engineering*, *365*, 113000. Retrieved from http://www.sciencedirect.com/science/article/pii/S0045782520301845 doi: https://doi.org/10.1016/j.cma.2020.113000

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27* (pp. 2672–2680). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

Grant, A. (2020). *Dynamics of COVID-19 epidemics: SEIR models underestimate peak infection rates and overestimate epidemic duration.* medRxiv. Retrieved from https://doi.org/10.1101/2020.04.02.20050674 doi: 10.1101/2020.04.02.20050674

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). *Improved Training of Wasserstein GANs.*

Karras, T., Laine, S., & Aila, T. (2018). *A Style-Based Generator Architecture for Generative Adversarial Networks.*

Prem, K., Liu, Y., Russell, T. W., Kucharski, A. J., Eggo, R. M., Davies, N., ... Klepac, P. (2020). The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study. *The Lancet Public Health*, *5*(5), e261 - e270. Retrieved from http://www.sciencedirect.com/science/article/pii/S2468266720300736 doi: https://doi.org/10.1016/S2468-2667(20)30073-6

Radford, A., Metz, L., & Chintala, S. (2015). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.*

Wall, M. E., Rechtsteiner, A., & Rocha, L. M. (2003). Singular Value Decomposition and Principal Component Analysis. In D. P. Berrar, W. Dubitzky, & M. Granzow (Eds.), *A practical approach to microarray data analysis* (pp. 91–109). Boston, MA: Springer US. Retrieved from https://doi.org/10.1007/0-306-47815-3_5 doi: 10.1007/0-306-47815-3_5