Traffic congestion modelling with Generative Adversarial Networks

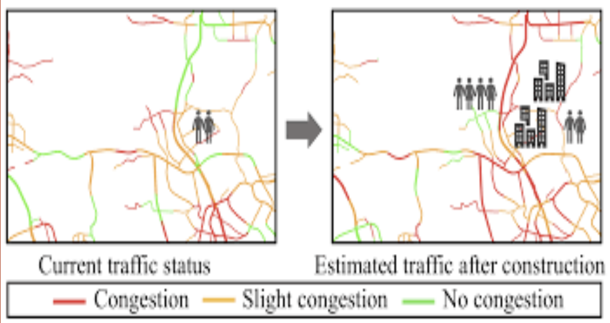
Andrew Smith, Vacation scholarship program

Supervisors: Dr. Joyce Zhang and Dr. Tingjin Chu

Background

When evaluating potential urban development plans a critical step prior to deployment is the estimation of the resultant traffic status.

Figure 1: Traffic estimation and evaluation in Vaughan, Canada



This traffic predication problem can be challenging to solve as:

- The plan may yield new travel demand patterns not captured in historical data.
- Traffic data tends to feature complex heterogeneous spatial-temporal dependencies.

Generative adversarial networks approach this task with an adversarial procedure formulated as a minimax game with loss function:

$$\min_G \max_D E_x[\log(D(x))] + E_z[1 - D(G(z))]$$

Here we simultaneously train two deep neural networks:

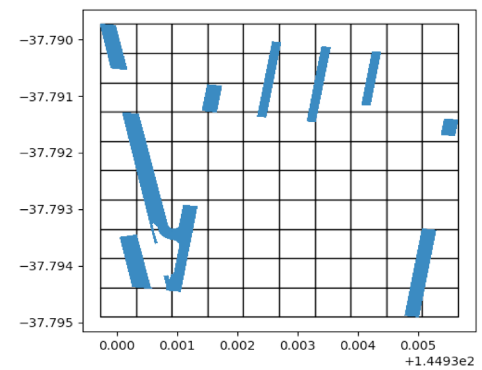
- The *generator* G that seeks to map noise to the data generating process.
- The *discriminator* D which measures the distance between the generated and real distributions.

Both models are trained until our generator produces traffic distribution samples that fool the discriminator – converging to a Nash equilibrium of the minimax game. We can generate conditional samples by feeding both generator and discriminator the conditioning information. My research applies the Curb-GAN [1] architecture and methodology to Melbourne traffic data - City of Melb. (2020). Traffic Count Vehicle Classification 2014-17

Methodology

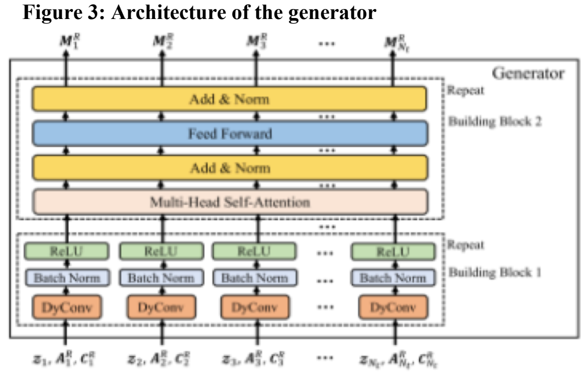
We first segment Melbourne into $I \times J$ equal length grid cells and spilt the city into overlapping target regions (R) of size $l \times l$.

Figure 2: Recorded road segments of an example target region



The dataset includes hourly vehicle counts which represent travel demand d_t^R and average speed M_t^R. Given regional historic traffic correlations A^R and an expected travel demand sequence $\widehat{d_1^R}, \ldots, \widehat{d_{N_t}^R}$ we aim to generate hourly consecutive traffic distributions $\widehat{M_1^R}, \ldots, \widehat{M_{N_t}^R}$.

Figure 3: Architecture of the generator



In order to capture the spatial dependencies *dynamic convolutional layers* (DyConv) are applied in G and D with propagation rule:

$$H_i^R = \sigma(A^R H_{i-1}^R W_i)$$

Where W_i are the layers weights. This module is inspired by graph convolution models which define a spatial graph convolution as an aggregation of the features of each node with its neighbors [2]. The traffic correlation matrix A^R captures the graphical structure of traffic by composing cells into highly correlated components.

For the temporal dependencies *multi-head self attention mechanism* is applied in G and D:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{MHA}(Q,K,V) = \text{Concat}(\text{Attention}_i, \ldots, \text{Attention}_h)$$

Self attention relates different positions of the same sequence to enhance the important parts of input. Multi head attention consists of jointly attending to information from different representation subspaces to capture different types of relevance [3].

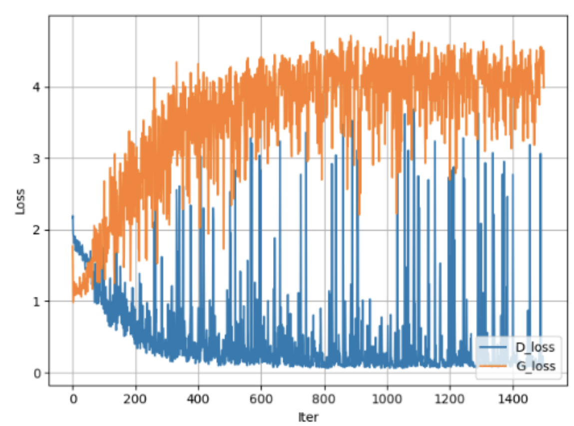
The model utilizes *dropout*: where during training we randomly drop units to simulate an ensemble of thinned networks. Dropout helps the model better generalize by eliminating complex co-adaptations between units.

Batch normalization is applied to enable faster and more stable training and consists of a normalizing step of each layers inputs.

The *Adam optimizer* is used for stochastic optimization and computes individual adaptive learning rates per parameter based on estimates of the 1st and 2nd gradient moments.

Results and conclusions

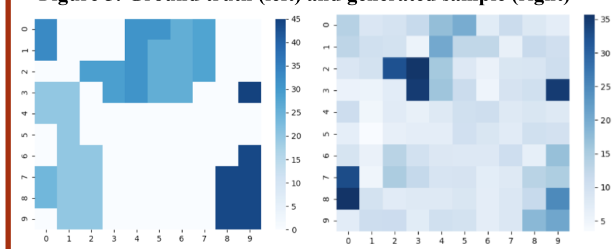
Figure 4: Loss function of D and G over 1500 epochs



Due to time and computational restraints the model was trained on a limited sample size. Deep learning models generally require a large training set due to the high number of parameters needing to be tuned.

Consequently we see high noise in the loss functions albeit with some evidence of convergence.

Figure 5: Ground truth (left) and generated sample (right)



Despite the small sample size qualitatively we see that the generator is able to produce reasonable looking samples.

From the Curb-GAN paper, using the mean absolute percentage error and rooted mean square error as evaluation metrics, the outlined architecture outperforms other competitive models.

Curb-GAN is capable of generating realistic conditional traffic distribution sequences enabling policy makers to estimate the impact of various travel demands on regional traffic status.

References

[1] Zhang, Y et al. (2020). Conference on Knowledge Discovery and Data Mining 842-852.

[2] Song, C et al, (2017). AAAI-20

[3] Vaswani, A et al. (2017). NIPS 2017

Acknowledgements and further information

I would like to thank my supervisors Dr. Joyce Zhang and Dr. Tingjin Chu for taking the time to guide me through this material and answer any questions I have. I would also like to thank the authors of the Curb-GAN paper and especially Yingxue Zhang for answering my queries on dataset formatting.

Feel free to contact at:

andsmith@student.unimelb.edu.au

https://github.com/and-smith/Melb-traffic-data-pre-processing