

Pull Requests

YANG Chengkai

37215175

Department of Electrical Engineering and Information System

M1

- Pull Requests I created
 - 2022-Flooding : Adaptive flooding variable value
 - 2022-NRI : CNN encoder, Visualization for encoder output
 - 2022-U-Net: Prevent exceeding memory

2022-Flooding : Adaptive flooding variable value

Flooding loss function

$$\tilde{J}(\theta) = |J(\theta) - b| + b, b \text{ is flooding constant}$$

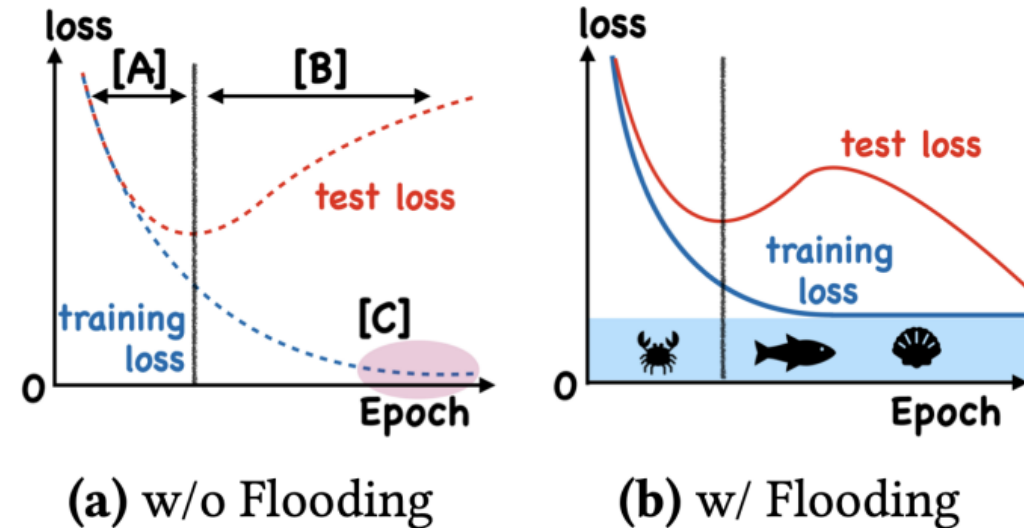
I implemented to decay b according to epoch number N

Example decay rule - Decay every 100 epochs:

if $N\%100==0$:

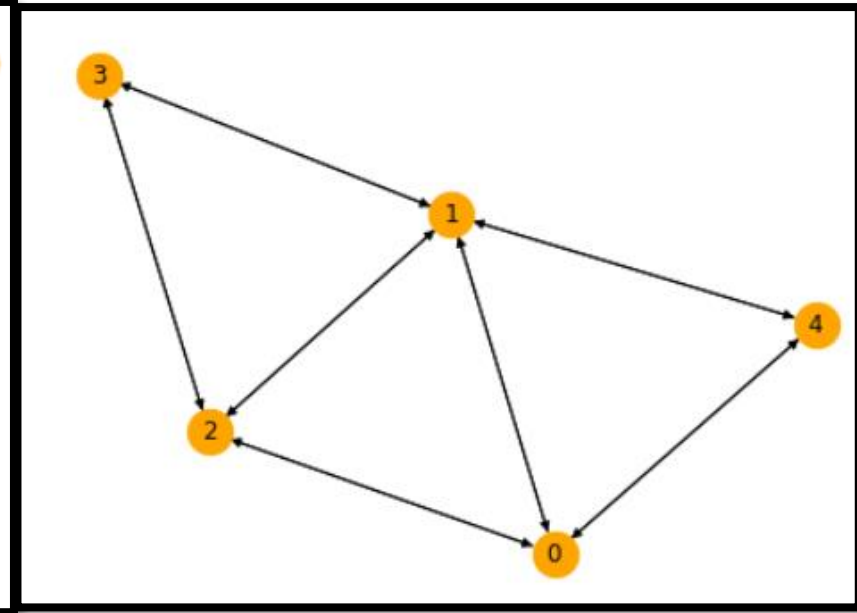
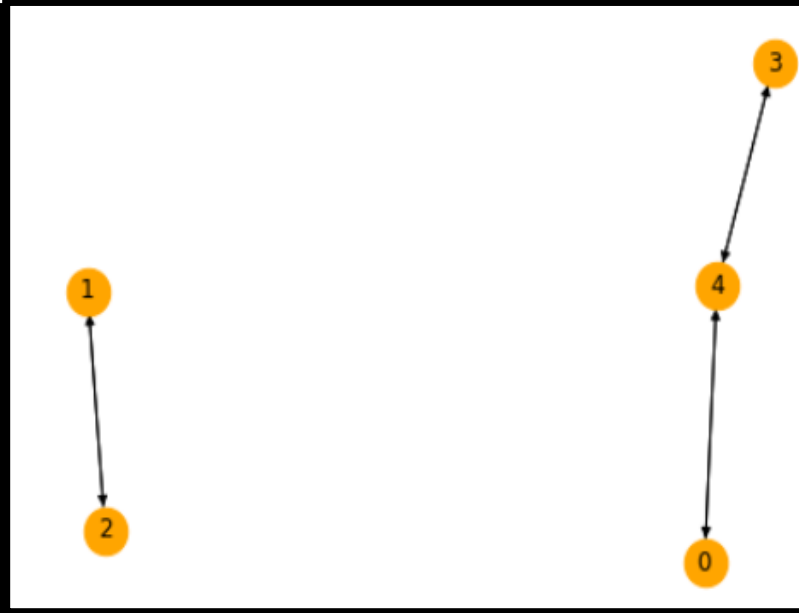
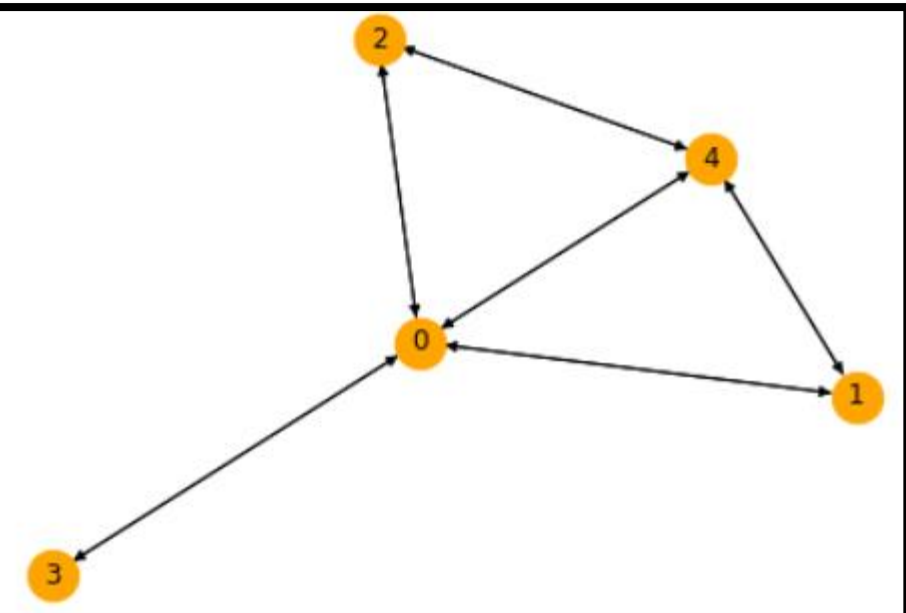
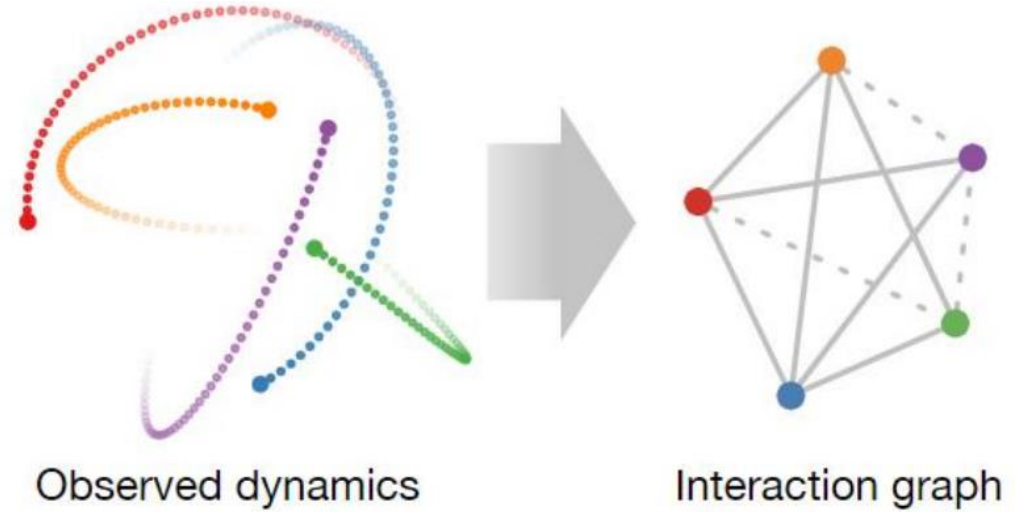
$$b = b - \lambda$$

A bit complicated in Tensorflow. I used callback to extract current epoch number and pass it to the loss function



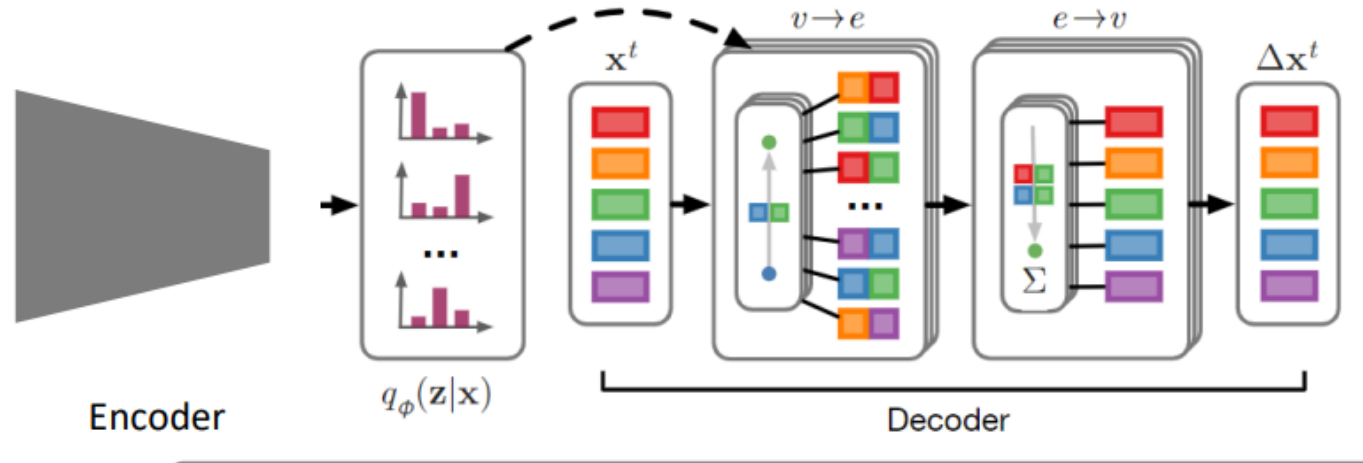
2022-NRI : Visualization for encoder output

- Visualize the latent space, which is probability of edge existence for each edge
- Constructed directed graph on the learned latent space, using networkx



2022-NRI : CNN encoder

- Implemented Encoder using convolutional neural network
- CNN extracts information from edges



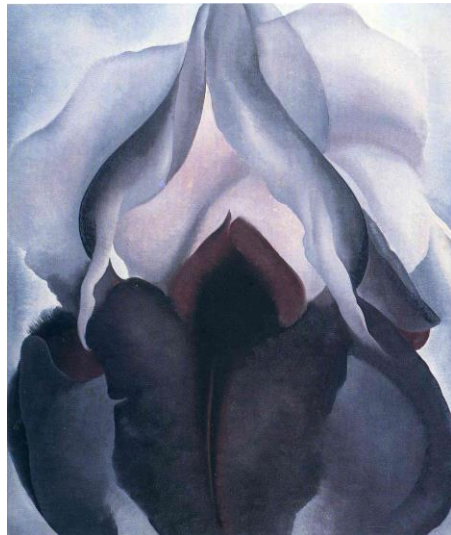
2022-U-Net: Prevent exceeding memory

- Encountered “allocation exceeds 10 of system memory”
- Solution: Feed test images separately to the model

- Pull Requests I received and merged for 2022-AdaIN-Pytorch
 - Color Control by MasaTate
 - Demo app by Naba

Merged: Color Control by MasaTate

- Goal: Preserve color for style transfer
- Solution: Before style transfer, use *Linear Histogram Matching* to match the color of style image to content image
- Preserve color for basic style transfer, style interpolation, and video style transfer (unstable)



Without color control



With color control

Merged: Demo App by Naba

- Goal: Create UI to use AdaIN Style Transfer more easily
- Solution: Created a Streamlit app, employed locally and online by Huggingface
 - Support preserving color
 - Support changing alpha
 - <https://huggingface.co/spaces/subatomicseer/2022-AdaIN-pytorch-Demo>