Data Free Model Extraction

Experiment

Based on the paper Data-Free Model Extraction https://arxiv.org/abs/2011.14779 by Truong et al which was for images in 10 classes, we use a generative model to generate examples of videos of 400 classes from which the student can learn. The student acts as a discriminator trying to get similar predictions as the predictions of the teacher when the generated image is given as input to both. Generator and Student are alternately learned. The backpropagation from the prediction loss (KLDiv or L1) was done by gradient approximation through the teacher.

Intuition

If we input random noise through the teacher, it gives mostly the same labels. We need a way to generate images that give different labels and are difficult to classify which is possible using a GAN based approach. The generator tries to generate images that will be difficult for the student to classify. This is encouraged by the L1 or KL Divergence loss function which the optimizer tries to maximize when learning the generator parameters. The student on the other hand tries to minimize the loss to better fit the teacher's predictions. Since the teacher is a blackbox, we cannot backpropagate through it. For that, we do gradient approximation using the Forward Differences method.

Methodology

We used the Data Free Model Extraction (DFME) paper's repository and modified it to work for videos. The Generator was replaced with Video Gan. We first conducted experiments before going to video experiments where we used the DFME with the Image Generator and tried to predict the class assuming it to be a single frame for the video for Swin-T and an image for the student MobileNet and tried to train for the 400 classes of Kinetics-400. We used various combinations of learning rate for student and generator and number of approximation points for the gradient approximator and changing the ratio between steps of generator and student

Conditional GAN

Experiment

Train a conditional GAN to combat the lack of diversity issues faced in the DFME experiment (most of the generated images were falling into very few classes, very high skew). This cGAN could later be used to:
generate a training set on which logits would be generated using the teacher. This training set would then be used to perform supervised learning.

Perform the DFME experiment but with much better data distribution

Intuition

Owing to the large number of classes in K400 and 600, achieving high diversity directly is difficult. Need some kind of loss mechanism to incentivise. Methods like incentivising entropy are likely to be difficult to learn from since batch sizes are small + number of classes always an issue. By using a cGAN we can directly train to generate images of each class.

Methodology

Used DFME paper's GeneratorC architecture with the embedding layer removed. The one-hot label is directly prepended to the noise vector z passed to the Generator. The output is then passed through Swin-T to get logits. A BCE loss is applied between logits and the one-hot (in the final, backprop run. Other losses tried with grad approx). Learning then happens through gradient approximation on the teacher (we actually used backprop in the run that worked). Code for this was modified from the DFME repo. Tried a huge range of LRs (something like 1e-1 to 1e-9) with grad approx, nothing worked. Tried a few LRs with backprop (1e-1, 1e-3, 1e-2, and 1e-7) and 1e-2 worked. Each were learning to some extent, categorically better than the grad approx runs.

Running Blackbox Model Extraction

1. Install all requirements ```pip install -q -r requirements.txt```
2. Download the weights of swin-t. URL: https://github.com/SwinTransformer/storage/releases/download/v1.0.4/swin_tiny_patch244_window877_kinetics400_1k.pth

Data Free Model Extraction

Modify the `config/params_dfme_swint.yaml` or `config/params_dfme_movinet.yaml` file to use the options as desired for Swin-T or MoViNet respectively and then ```python DFME/train.py --config path-to-config-yaml```
OR
Create a new config file to use the options as desired and then ```python DFME/train.py --config path-to-config-yaml```
NOTE: Make sure to properly set the model checkpoints path in the config.

Data Free Model Extraction with Conditional GAN

Pretraining cGAN

Modify the `config/params_pretrain_swint.yaml` or `config/params_pretrain_movinet.yaml` file to use the options as desired for Swin-T or MoViNet respectively and then ```python cGAN/pretrain_generator.py --config path-to-config-yaml```
OR
Create a new config file to use the options as desired and then ```python cGAN/pretrain_generator.py --config path-to-config-yaml```
NOTE: Make sure to properly set the path of datasets and model checkpoint path in the config.

Training Threat Model with Single Frame

Modify the `config/params_cgan_swint.yaml` or `config/params_cgan_movinet.yaml` file to use the options as desired for Swin-T or MoViNet respectively and then ```python cGAN/train.py --config path-to-config-yaml```
OR
Create a new config file to use the options as desired and then ```python cGAN/train.py --config path-to-config-yaml```
NOTE: Make sure to properly set the path of datasets and model checkpoint path in the config.