ADV-TRAIN : ADV-Train is Deep Vision TRaining And INference Framework

Release Alpha 1.0

Deepak Ravikumar Tatachar and Sangamesh Kodge

Aug 16, 2020

CONTENTS:

This is a framework built on top of pytorch to make machine learning training and inference tasks easier. Along with that it also enables easy dataset and network instantiations, visualize boundaries and more. This was created at the Nanoelectronics Research Laboratory at Purdue.

CHAPTER

ONE

INDICES AND TABLES

- genindex
- modindex

CHAPTER

TWO

MODULES

class advtrain.utils.normalize.**denormalize**(*mean*, *std*, *img_dimensions*, *device='cpu'*)

De-normalizes the input and provides a backpropable de-normalization function

Parameters

- **mean** (*list*) – list of the mean for each channel
- **std** (*list*) – list of the std for each channel
- **img_dimensions** (*list*) – list [channels,h,w]
- **device** (*str*) – cpu/cuda

Returns Returns an object of normalize

class advtrain.utils.normalize.**normalize**(*mean*, *std*, *img_dimensions*, *device='cpu'*)

Normalizes the input and provides a backpropable normalization function. It perfroms the channel wise normalization using mean and standard deviation

Parameters

- **mean** (*list*) – list of the mean for each channel
- **std** (*list*) – list of the std for each channel
- **img_dimensions** (*list*) – list [channels,h,w]
- **device** (*str*) – cpu/cuda

Returns Returns an object of denormalize

advtrain.instantiate_model.**instantiate_model**(*dataset='cifar10'*, *num_classes=10*, *input_quant='FP'*, *arch='resnet'*, *dorefa=False*, *abit=32*, *wbit=32*, *qin=False*, *qout=False*, *suffix=''*, *load=False*, *torch_weights=False*, *device='cpu'*)

Initializes/load network with random weight/saved and return auto generated model name 'dataset_arch_suffix.ckpt'

Parameters

- **dataset** – mnists/cifar10/cifar100/imagenet/tinyimagenet/simple dataset the netwoek is trained on. Used in model name
- **num_classes** – number of classes in dataset.
- **arch** – resnet/vgg/lenet5/basicnet/slpconv model architecture the network to be instantiated with
- **suffix** – str appended to the model name

- **load** – boolean variable to indicate load pretrained model from ./pretrained/dataset/
- **torch_weights** – boolean variable to indicate load weight from torchvision for imagenet dataset

Returns models with desired weight (pretrained / random) model_name : str 'dataset_arch_suffix.ckpt' used to save/load model in ./pretrained/dataset

Return type model

class advtrain.utils.preprocess.**preprocess**

This class consists of a forward pass and backward pass function for preprocessing layer.

back_approx(*x*)

This function forward propogates through the preprocessing layer :param x: Input to be back propagated

Returns The backward propagated input

forward(*x*)

This function forward propogates through the preprocessing layer Args:

Returns:

class advtrain.attack_framework.multi_lib_attacks.**attack_wrapper**(*model*, *device*, ***params*)

This is a LAL (Library Abstraction Layer) Wrapper around foolbox, advertorch and custom attack implementation

Example of attack_params {

'lib': 'foolbox', 'attack':'pgd', 'iterations': 40, 'epsilon': 0.01, 'stepsize': 0.01, 'bpda': True, 'random': True 'preprocess': <pointer to quantize or hafltone or FP>, 'custom_norm_func': <pointer to normalize>, 'targeted': True,

}

Example of dataset_params {

'mean': [0.5, 0.5, 0.5], 'std': [0.5, 0.5, 0.5], 'num_classes': 10

}

PYTHON MODULE INDEX

a

INDEX