torchbnn Documentation

Release v1.1

harrykim

Dec 07, 2020

Contents:

CHAPTER 1

Modules

1.1 Bayes Module

class torchbnn.modules.module.**BayesModule**

Applies Bayesian Module Currently this module is not being used as base of bayesian modules because it has not many utilies yet, However, it can be used in the near future for convenience.

freeze()

Sets the module in freezed mode. This has effect on bayesian modules. It will fix epsilons, e.g. weight_eps, bias_eps. Thus, bayesian neural networks will return same results with same inputs.

unfreeze()

Sets the module in unfreezed mode. This has effect on bayesian modules. It will unfix epsilons, e.g. weight_eps, bias_eps. Thus, bayesian neural networks will return different results even if same inputs are given.

1.2 Bayes Linear

class torchbnn.modules.linear.**BayesLinear**(*prior_mu*, *prior_sigma*, *in_features*, *out_features*, *bias=True*)

Applies Bayesian Linear

Parameters

- **prior_mu** (*Float*) – mean of prior normal distribution.
- **prior_sigma** (*Float*) – sigma of prior normal distribution.

Note: other arguments are following linear of pytorch 1.2.0.

https://github.com/pytorch/pytorch/blob/master/torch/nn/modules/linear.py

extra_repr()
Overriden.

forward(*input*)
Overriden.

1.3 Bayes Conv

class torchbnn.modules.conv.**BayesConv2d**(*prior_mu*, *prior_sigma*, *in_channels*, *out_channels*, *kernel_size*, *stride=1*, *padding=0*, *dilation=1*, *groups=1*, *bias=True*, *padding_mode='zeros'*)

Applies Bayesian Convolution for 2D inputs

Parameters

- **prior_mu** (*Float*) – mean of prior normal distribution.
- **prior_sigma** (*Float*) – sigma of prior normal distribution.

Note: other arguments are following conv of pytorch 1.2.0.

https://github.com/pytorch/pytorch/blob/master/torch/nn/modules/conv.py

forward(*input*)
Overriden.

1.4 Bayes Batchnorm

class torchbnn.modules.batchnorm.**BayesBatchNorm2d**(*prior_mu*, *prior_sigma*, *num_features*, *eps=1e-05*, *momentum=0.1*, *affine=True*, *track_running_stats=True*)

Applies Bayesian Batch Normalization over a 2D input

Parameters

- **prior_mu** (*Float*) – mean of prior normal distribution.
- **prior_sigma** (*Float*) – sigma of prior normal distribution.

Note: other arguments are following batchnorm of pytorch 1.2.0.

https://github.com/pytorch/pytorch/blob/master/torch/nn/modules/batchnorm.py

1.5 BKLLoss

class torchbnn.modules.loss.**BKLLoss**(*reduction='mean'*, *last_layer_only=False*)
Loss for calculating KL divergence of baysian neural network model.

Parameters

- **reduction** (*string, optional*) – Specifies the reduction to apply to the output: `'mean'`: the sum of the output will be divided by the number of elements of the output. `'sum'`: the output will be summed.
- **last_layer_only** (*Bool*) – True for return only the last layer's KL divergence.

forward(*model*)

Parameters **model** (*nn.Module*) – a model to be calculated for KL-divergence.

CHAPTER 2

Utils

2.1 Freeze Model

torchbnn.utils.freeze_model.**freeze**(*module*)
Methods for freezing bayesian-model.

Parameters **model** (*nn.Module*) – a model to be freezed.

torchbnn.utils.freeze_model.**unfreeze**(*module*)
Methods for unfreezing bayesian-model.

Parameters **model** (*nn.Module*) – a model to be unfreezed.

CHAPTER 3

Functional

3.1 Bayesian KL Loss

torchbnn.functional.**bayesian_kl_loss**(*model*, *reduction='mean'*, *last_layer_only=False*)

An method for calculating KL divergence of whole layers in the model.

Parameters

- **model** (*nn.Module*) – a model to be calculated for KL-divergence.
- **reduction** (*string, optional*) – Specifies the reduction to apply to the output: `'mean'`: the sum of the output will be divided by the number of elements of the output. `'sum'`: the output will be summed.
- **last_layer_only** (*Bool*) – True for return only the last layer's KL divergence.

Python Module Index

t

Index