stAdv Documentation

Release 0.2.1

Beranger Dumont

Nov 03, 2018

Contents:

CHAPTER 1

stadv package

1.1 stadv.layers module

stadv.layers.**flow_st**(*images*, *flows*, *data_format='NHWC'*)

Flow-based spatial transformation of images. See Eq. (1) in Xiao et al. (arXiv:1801.02612).

Parameters

- **images** (*tf.Tensor*) – images of shape *(B, H, W, C)* or *(B, C, H, W)* depending on *data_format*.
- **flows** (*tf.Tensor*) – flows of shape *(B, 2, H, W)*, where the second dimension indicates the dimension on which the pixel shift is applied.
- **data_format** (*str*) – `'NHWC'` or `'NCHW'` depending on the format of the input images and the desired output.

Returns *tf.Tensor* of the same shape and type as *images*.

1.2 stadv.losses module

stadv.losses.**adv_loss**(*unscaled_logits*, *targets*, *kappa=None*)

Computes the adversarial loss. It was first suggested by Carlini and Wagner (arXiv:1608.04644). See also Eq. (3) in Xiao et al. (arXiv:1801.02612).

Parameters

- **unscaled_logits** (*tf.Tensor*) – logits of shape *(B, K)*, where *K* is the number of input classes.
- **targets** (*tf.Tensor*) – *1-D* integer-encoded targets of length *B* with value corresponding to the class ID.
- **kappa** (*tf.Tensor*) – confidence parameter, see Carlini and Wagner (arXiv:1608.04644). Defaults to 0.

Returns 1-D *tf.Tensor* of length *B* of the same type as *unscaled_logits*.

`stadv.losses.`**`flow_loss`**(*flows*, *padding_mode='SYMMETRIC'*, *epsilon=1e-08*)

Computes the flow loss designed to "enforce the locally smooth spatial transformation perturbation". See Eq. (4) in Xiao et al. (arXiv:1801.02612).

Parameters

- **`flows`** (*`tf.Tensor`*) – flows of shape *(B, 2, H, W)*, where the second dimension indicates the dimension on which the pixel shift is applied.
- **`padding_mode`** (*`str`*) – how to perform padding of the boundaries of the images. The value should be compatible with the *mode* argument of `tf.pad`. Expected values are:
  - `'SYMMETRIC'`: symmetric padding so as to not penalize a significant flow at the boundary of the images;
  - `'CONSTANT'`: 0-padding of the boundaries so as to enforce a small flow at the boundary of the images.
- **`epsilon`** (*`float`*) – small value added to the argument of `tf.sqrt` to prevent NaN gradients when the argument is zero.

Returns 1-D *tf.Tensor* of length *B* of the same type as *flows*.

1.3 stadv.optimization module

`stadv.optimization.`**`lbfgs`**(*loss*, *flows*, *flows_x0*, *feed_dict=None*, *grad_op=None*, *fmin_l_bfgs_b_extra_kwargs=None*, *sess=None*)

Optimize a given loss with (SciPy's external) L-BFGS-B optimizer. It can be used to solve the optimization problem of Eq. (2) in Xiao et al. (arXiv:1801.02612). See the documentation on scipy.optimize.fmin_l_bfgs_b for reference on the optimizer.

Parameters

- **`loss`** (*`tf.Tensor`*) – loss (can be of any shape).
- **`flows`** (*`tf.Tensor`*) – flows of shape *(B, 2, H, W)*, where the second dimension indicates the dimension on which the pixel shift is applied.
- **`flows_x0`** (*`np.ndarray`*) – Initial guess for the flows. If the input is not of type *np.ndarray*, it will be converted as such if possible.
- **`feed_dict`** (*`dict`*) – feed dictionary to the `tf.run` operation (for everything which might be needed to execute the graph beyond the input flows).
- **`grad_op`** (*`tf.Tensor`*) – gradient of the loss with respect to the flows. If not provided it will be computed from the input and added to the graph.
- **`fmin_l_bfgs_b_extra_kwargs`** (*`dict`*) – extra arguments to `scipy.optimize.fmin_l_bfgs_b` (e.g. for modifying the stopping condition).
- **`sess`** (*`tf.Session`*) – session within which the graph should be executed. If not provided a new session will be started.

Returns *Dictionary* with keys `'flows'` (*np.ndarray*, estimated flows of the minimum), `'loss'` (*float*, value of loss at the minimum), and `'info'` (*dict*, information summary as returned by `scipy.optimize.fmin_l_bfgs_b`).

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

Index