CleverHans Documentation

Ian Goodfellow, Nicolas Papernot, Ryan Sheatsley

Jan 12, 2019

Contents

This documentation is auto-generated from the docstrings of modules of the current *master* branch of tensorflow/cleverhans.

To get started, we recommend reading the github readme. Afterwards, you can learn more by looking at the following modules:

CHAPTER 1

attacks module

CHAPTER 2

model module

The Model class and related functionality.

class cleverhans.model.**CallableModelWrapper**(*callable_fn*, *output_layer*)

Bases: *cleverhans.model.Model*

A wrapper that turns a callable into a valid Model

fprop(*x*, ***kwargs*)

Forward propagation to compute the model outputs. :param x: A symbolic representation of the network input :return: A dictionary mapping layer names to the symbolic

representation of their output.

class cleverhans.model.**Model**(*scope=None*, *nb_classes=None*, *hparams=None*, *needs_dummy_fprop=False*)

Bases: object

An abstract interface for model wrappers that exposes model symbols needed for making an attack. This abstraction removes the dependency on any specific neural network package (e.g. Keras) from the core code of CleverHans. It can also simplify exposing the hidden features of a model when a specific package does not directly expose them.

O_FEATURES = 'features'

O_LOGITS = 'logits'

O_PROBS = 'probs'

fprop(*x*, ***kwargs*)

Forward propagation to compute the model outputs. :param x: A symbolic representation of the network input :return: A dictionary mapping layer names to the symbolic

representation of their output.

get_layer(*x*, *layer*, ***kwargs*)

Return a layer output. :param x: tensor, the input to the network. :param layer: str, the name of the layer to compute. :param **kwargs: dict, extra optional params to pass to self.fprop. :return: the content of layer *layer*

get_layer_names()
Return the list of exposed layers for this model.

get_logits(*x*, ***kwargs*)

Parameters **x** – A symbolic representation (Tensor) of the network input

Returns A symbolic representation (Tensor) of the output logits

(i.e., the values fed as inputs to the softmax layer).

get_params()
Provides access to the model's parameters. :return: A list of all Variables defining the model parameters.

get_predicted_class(*x*, ***kwargs*)

Parameters **x** – A symbolic representation (Tensor) of the network input

Returns A symbolic representation (Tensor) of the predicted label

get_probs(*x*, ***kwargs*)

Parameters **x** – A symbolic representation (Tensor) of the network input

Returns A symbolic representation (Tensor) of the output

probabilities (i.e., the output values produced by the softmax layer).

make_input_placeholder()
Create and return a placeholder representing an input to the model.

This method should respect context managers (e.g. "with tf.device") and should not just return a reference to a single pre-created placeholder.

make_label_placeholder()
Create and return a placeholder representing class labels.

This method should respect context managers (e.g. "with tf.device") and should not just return a reference to a single pre-created placeholder.

make_params()
Create all Variables to be returned later by get_params. By default this is a no-op. Models that need their fprop to be called for their params to be created can set *needs_dummy_fprop=True* in the constructor.

exception cleverhans.model.**NoSuchLayerError**
Bases: ValueError

Raised when a layer that does not exist is requested.

cleverhans.model.**wrapper_warning**()
Issue a deprecation warning. Used in multiple places that implemented attacks by automatically wrapping a user-supplied callable with a CallableModelWrapper with output_layer="probs". Using "probs" as any part of the attack interface is dangerous. We can't just change output_layer to logits because: - that would be a silent interface change. We'd have no way of detecting

code that still means to use probs. Note that we can't just check whether the final output op is a softmax—for example, Inception puts a reshape after the softmax.

- automatically wrapping user-supplied callables with output_layer='logits' is even worse, see *wrapper_warning_logits*

Note: this function will be removed at the same time as the code that calls it.

cleverhans.model.**wrapper_warning_logits**()

Issue a deprecation warning. Used in multiple places that implemented attacks by automatically wrapping a user-supplied callable with a CallableModelWrapper with output_layer="logits". This is dangerous because it is under-the-hood automagic that the user may not realize has been invoked for them. If they pass a callable that actually outputs probs, the probs will be treated as logits, resulting in an incorrect cross-entropy loss and severe gradient masking.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

C

Index