tf.contrib.learn.train

```
train(
    graph,
    output_dir,
    train_op,
    loss_op,
    global_step_tensor=None,
    init_op=None,
    init_feed_dict=None,
    init_fn=None,
    log_every_steps=10,
    supervisor_is_chief=True,
    supervisor_master='',
    supervisor_save_model_secs=600,
    keep_checkpoint_max=5,
    supervisor_save_summaries_steps=100,
    feed_fn=None,
    steps=None,
    fail_on_nan_loss=True,
    monitors=None,
    max_steps=None
)
```

Defined in `tensorflow/contrib/learn/python/learn/graph_actions.py`.

See the guide: Learn (contrib) > Graph actions

Train a model. (deprecated)

THIS FUNCTION IS DEPRECATED. It will be removed after 2017-02-15. Instructions for updating: graph_actions.py will be deleted. Use tf.train.* utilities instead. You can use learn/estimators/estimator.py as an example.

Given `graph`, a directory to write outputs to (`output_dir`), and some ops, run a training loop. The given `train_op` performs one step of training on the model. The `loss_op` represents the objective function of the training. It is expected to increment the `global_step_tensor`, a scalar integer tensor counting training steps. This function uses `Supervisor` to initialize the graph (from a checkpoint if one is available in `output_dir`), write summaries defined in the graph, and write regular checkpoints as defined by `supervisor_save_model_secs`.

Training continues until `global_step_tensor` evaluates to `max_steps`, or, if `fail_on_nan_loss`, until `loss_op` evaluates to `NaN`. In that case the program is terminated with exit code 1.

Args:

- `graph`: A graph to train. It is expected that this graph is not in use elsewhere.
- `output_dir`: A directory to write outputs to.
- `train_op`: An op that performs one training step when run.
- `loss_op`: A scalar loss tensor.
- `global_step_tensor`: A tensor representing the global step. If none is given, one is extracted from the graph using the same logic as in `Supervisor`.
- `init_op`: An op that initializes the graph. If `None`, use `Supervisor`'s default.

- `init_feed_dict` : A dictionary that maps `Tensor` objects to feed values. This feed dictionary will be used when `init_op` is evaluated.
- `init_fn` : Optional callable passed to Supervisor to initialize the model.
- `log_every_steps` : Output logs regularly. The logs contain timing data and the current loss.
- `supervisor_is_chief` : Whether the current process is the chief supervisor in charge of restoring the model and running standard services.
- `supervisor_master` : The master string to use when preparing the session.
- `supervisor_save_model_secs` : Save a checkpoint every `supervisor_save_model_secs` seconds when training.
- `keep_checkpoint_max` : The maximum number of recent checkpoint files to keep. As new files are created, older files are deleted. If None or 0, all checkpoint files are kept. This is simply passed as the max_to_keep arg to tf.train.Saver constructor.
- `supervisor_save_summaries_steps` : Save summaries every `supervisor_save_summaries_steps` seconds when training.
- `feed_fn` : A function that is called every iteration to produce a `feed_dict` passed to `session.run` calls. Optional.
- `steps` : Trains for this many steps (e.g. current global step + `steps`).
- `fail_on_nan_loss` : If true, raise `NanLossDuringTrainingError` if `loss_op` evaluates to `NaN` . If false, continue training as if nothing happened.
- `monitors` : List of `BaseMonitor` subclass instances. Used for callbacks inside the training loop.
- `max_steps` : Number of total steps for which to train model. If `None` , train forever. Two calls fit(steps=100) means 200 training iterations. On the other hand two calls of fit(max_steps=100) means, second call will not do any iteration since first call did all 100 steps.

Returns:

The final loss value.

Raises:

- `ValueError` : If `output_dir` , `train_op` , `loss_op` , or `global_step_tensor` is not provided. See `tf.contrib.framework.get_global_step` for how we look up the latter if not provided explicitly.
- `NanLossDuringTrainingError` : If `fail_on_nan_loss` is `True` , and loss ever evaluates to `NaN` .
- `ValueError` : If both `steps` and `max_steps` are not `None` .

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**