tf.contrib.training.evaluate_repeatedly

```
evaluate_repeatedly(
    checkpoint_dir,
    master='',
    scaffold=None,
    eval_ops=None,
    feed_dict=None,
    final_ops=None,
    final_ops_feed_dict=None,
    eval_interval_secs=60,
    hooks=None,
    config=None,
    max_number_of_evaluations=None,
    timeout=None,
    timeout_fn=None
)
```

Defined in `tensorflow/contrib/training/python/training/evaluation.py`.

Repeatedly searches for a checkpoint in `checkpoint_dir` and evaluates it.

During a single evaluation, the `eval_ops` is run until the session is interrupted or requested to finish. This is typically requested via a `tf.contrib.training.StopAfterNEvalsHook` which results in `eval_ops` running the requested number of times.

Optionally, a user can pass in `final_ops`, a single `Tensor`, a list of `Tensors` or a dictionary from names to `Tensors`. The `final_ops` is evaluated a single time after `eval_ops` has finished running and the fetched values of `final_ops` are returned. If `final_ops` is left as `None`, then `None` is returned.

One may also consider using a `tf.contrib.training.SummaryAtEndHook` to record summaries after the `eval_ops` have run. If `eval_ops` is `None`, the summaries run immediately after the model checkpoint has been restored.

Note that `evaluate_once` creates a local variable used to track the number of evaluations run via `tf.contrib.training.get_or_create_eval_step`. Consequently, if a custom local init op is provided via a `scaffold`, the caller should ensure that the local init op also initializes the eval step.

Args:

- `checkpoint_dir`: The directory where checkpoints are stored.
- `master`: The address of the TensorFlow master.
- `scaffold`: An tf.train.Scaffold instance for initializing variables and restoring variables. Note that `scaffold.init_fn` is used by the function to restore the checkpoint. If you supply a custom init_fn, then it must also take care of restoring the model from its checkpoint.
- `eval_ops`: A single `Tensor`, a list of `Tensors` or a dictionary of names to `Tensors`, which is run until the session is requested to stop, commonly done by a `tf.contrib.training.StopAfterNEvalsHook`.
- `feed_dict`: The feed dictionary to use when executing the `eval_ops`.
- `final_ops`: A single `Tensor`, a list of `Tensors` or a dictionary of names to `Tensors`.
- `final_ops_feed_dict`: A feed dictionary to use when evaluating `final_ops`.
- `eval_interval_secs`: The minimum number of seconds between evaluations.

- `hooks`: List of `tf.train.SessionRunHook` callbacks which are run inside the evaluation loop.
- `config`: An instance of `tf.ConfigProto` that will be used to configure the `Session`. If left as `None`, the default will be used.
- `max_number_of_evaluations`: The maximum times to run the evaluation. If left as `None`, then evaluation runs indefinitely.
- `timeout`: The maximum amount of time to wait between checkpoints. If left as `None`, then the process will wait indefinitely.
- `timeout_fn`: Optional function to call after a timeout. If the function returns True, then it means that no new checkpoints will be generated and the iterator will exit. The function is called with no arguments.

Returns:

The fetched values of `final_ops` or `None` if `final_ops` is `None`.
