# tf.contrib.linear_optimizer.SDCAOptimizer

**Contents**

## Class **`SDCAOptimizer`**

Defined in `tensorflow/contrib/linear_optimizer/python/sdca_optimizer.py`.

Wrapper class for SDCA optimizer.

The wrapper is currently meant for use as an optimizer within a tf.learn Estimator.

Example usage:

```
real_feature_column = real_valued_column(...)
sparse_feature_column = sparse_column_with_hash_bucket(...)
sdca_optimizer = linear.SDCAOptimizer(example_id_column='example_id',
                                      num_loss_partitions=1,
                                      num_table_shards=1,
                                      symmetric_l2_regularization=2.0)
classifier = tf.contrib.learn.LinearClassifier(
    feature_columns=[real_feature_column, sparse_feature_column],
    weight_column_name=...,
    optimizer=sdca_optimizer)
classifier.fit(input_fn_train, steps=50)
classifier.evaluate(input_fn=input_fn_eval)
```

Here the expectation is that the `input_fn_*` functions passed to train and evaluate return a pair (dict, label_tensor) where dict has `example_id_column` as `key` whose value is a `Tensor` of shape [batch_size] and dtype string. num_loss_partitions defines the number of partitions of the global loss function and should be set to `(#concurrent train ops/per worker) x (#workers)`. Convergence of (global) loss is guaranteed if `num_loss_partitions` is larger or equal to the above product. Larger values for `num_loss_partitions` lead to slower convergence. The recommended value for `num_loss_partitions` in `tf.learn` (where currently there is one process per worker) is the number of workers running the train steps. It defaults to 1 (single machine). `num_table_shards` defines the number of shards for the internal state table, typically set to match the number of parameter servers for large data sets.

## Properties

### `example_id_column`

### `num_loss_partitions`

### `num_table_shards`

### `symmetric_l1_regularization`

### `symmetric_l2_regularization`

## Methods

### `__init__`

```
__init__(
    example_id_column,
    num_loss_partitions=1,
    num_table_shards=None,
    symmetric_l1_regularization=0.0,
    symmetric_l2_regularization=1.0
)
```

### `get_name`

```
get_name()
```

### `get_train_step`

```
get_train_step(
    columns_to_variables,
    weight_column_name,
    loss_type,
    features,
    targets,
    global_step
)
```

Returns the training operation of an SdcaModel optimizer.

*Last updated November 2, 2017.*