tf.contrib.framework.load_variable_slot_initializer

```
load_variable_slot_initializer(
    ckpt_path,
    old_tensor_name,
    primary_partition_info,
    new_row_vocab_size,
    new_col_vocab_size,
    old_row_vocab_file=None,
    new_row_vocab_file=None,
    old_col_vocab_file=None,
    new_col_vocab_file=None,
    num_row_oov_buckets=0,
    num_col_oov_buckets=0,
    initializer=None,
    max_rows_in_memory=-1
)
```

Defined in `tensorflow/contrib/framework/python/ops/checkpoint_ops.py`.

Loads pre-trained multi-class slots for linear models from checkpoint.

Wrapper around `load_and_remap_matrix_initializer()` specialized for loading multi-class slots (such as optimizer accumulators) and remapping them according to the provided vocab files. See docs for `load_and_remap_matrix_initializer()` for more details. Takes in a `variable_scope._PartitionInfo` representing the slot's primary `Variable`'s partitioning. This is necessary since accumulator `Variable` creation ignores primary scoping and partitioning information.

Args:

- `ckpt_path`: Path to the TensorFlow checkpoint (version 2, `TensorBundle`) from which the old matrix `Tensor` will be loaded.
- `old_tensor_name`: Name of the 2-D `Tensor` to load from checkpoint.
- `primary_partition_info`: A `variable_scope._PartitionInfo` containing this slot's primary `Variable`'s partitioning information. This is used to calculate the offset and override the partition_info passed to the call to _initialize.
- `new_row_vocab_size`: `int` specifying the number of entries in `new_row_vocab_file`. If no row remapping is needed (no row vocab provided), this should be equal to the number of rows to load from the old matrix (which can theoretically be smaller than the number of rows in the old matrix).
- `new_col_vocab_size`: `int` specifying the number of entries in `new_col_vocab_file`. If no column remapping is needed (no column vocab provided), this should be equal to the number of columns in the old matrix.
- `old_row_vocab_file`: A scalar `Tensor` of type `string` containing the path to the old row vocabulary file. Can be None, which represents no remapping on the row axis.
- `new_row_vocab_file`: A scalar `Tensor` of type `string` containing the path to the new row vocabulary file. Can be None, which represents no remapping on the row axis.
- `old_col_vocab_file`: A scalar `Tensor` of type `string` containing the path to the old column vocabulary file. Can be None, which represents no remapping on the column axis.
- `new_col_vocab_file`: A scalar `Tensor` of type `string` containing the path to the new column vocabulary file. Can

be None, which represents no remapping on the column axis.

- `num_row_oov_buckets`: `int` specifying the number of out-of-vocabulary rows to append. Must be >= 0.
- `num_col_oov_buckets`: `int` specifying the number of out-of-vocabulary columns to append. Must be >= 0.
- `initializer`: Initializer function to initialize missing values. Accepts a 1-D tensor as the arg to specify the shape of the returned tensor. If `None`, defaults to using `zeros_initializer()`.
- `max_rows_in_memory`: `int` specifying the maximum number of rows to load from the checkpoint at once. If less than or equal to 0, the entire matrix will be loaded into memory. Setting this arg trades increased disk reads for lower memory usage.

Returns:

A variable initializer function that should be used to initialize a (potentially partitioned) `Variable` whose complete shape is `[new_row_vocab_size + num_row_oov_buckets, new_col_vocab_size + num_col_oov_buckets]`.

Raises:

- `TypeError`: If `initializer` is specified but not callable.
