tf.contrib.layers.input_from_feature_columns

```
input_from_feature_columns(
    columns_to_tensors,
    feature_columns,
    weight_collections=None,
    trainable=True,
    scope=None
)
```

Defined in `tensorflow/contrib/layers/python/layers/feature_column_ops.py`.

See the guide: Layers (contrib) > Feature columns

A tf.contrib.layers style input layer builder based on FeatureColumns.

Generally a single example in training data is described with feature columns. At the first layer of the model, this column oriented data should be converted to a single tensor. Each feature column needs a different kind of operation during this conversion. For example sparse features need a totally different handling than continuous features.

Example:

```
  # Building model for training
  columns_to_tensor = tf.parse_example(...)
  first_layer = input_from_feature_columns(
      columns_to_tensors=columns_to_tensor,
      feature_columns=feature_columns)
  second_layer = fully_connected(inputs=first_layer, ...)
  ...
```

where feature_columns can be defined as follows:

```
  sparse_feature = sparse_column_with_hash_bucket(
      column_name="sparse_col", ...)
  sparse_feature_emb = embedding_column(sparse_id_column=sparse_feature, ...)
  real_valued_feature = real_valued_column(...)
  real_valued_buckets = bucketized_column(
      source_column=real_valued_feature, ...)

  feature_columns=[sparse_feature_emb, real_valued_buckets]
```

Args:

- `columns_to_tensors`: A mapping from feature column to tensors. 'string' key means a base feature (not-transformed). It can have FeatureColumn as a key too. That means that FeatureColumn is already transformed by input pipeline.
- `feature_columns`: A set containing all the feature columns. All items in the set should be instances of classes derived by FeatureColumn.
- `weight_collections`: List of graph collections to which weights are added.
- `trainable`: If `True` also add variables to the graph collection `GraphKeys.TRAINABLE_VARIABLES` (see tf.Variable).
- `scope`: Optional scope for variable_scope.

Returns:

A Tensor which can be consumed by hidden layers in the neural network.

Raises:

- `ValueError`: if FeatureColumn cannot be consumed by a neural network.

Last updated November 2, 2017.