tf.feature_column.categorical_column_with_identity

```
categorical_column_with_identity(
    key,
    num_buckets,
    default_value=None
)
```

Defined in `tensorflow/python/feature_column/feature_column.py`.

A `_CategoricalColumn` that returns identity values.

Use this when your inputs are integers in the range `[0, num_buckets)`, and you want to use the input value itself as the categorical ID. Values outside this range will result in `default_value` if specified, otherwise it will fail.

Typically, this is used for contiguous ranges of integer indexes, but it doesn't have to be. This might be inefficient, however, if many of IDs are unused. Consider `categorical_column_with_hash_bucket` in that case.

For input dictionary `features`, `features[key]` is either `Tensor` or `SparseTensor`. If `Tensor`, missing values can be represented by `-1` for int and `''` for string. Note that these values are independent of the `default_value` argument.

In the following examples, each input in the range `[0, 1000000)` is assigned the same value. All other inputs are assigned `default_value` 0. Note that a literal 0 in inputs will result in the same default ID.

Linear model:

```
video_id = categorical_column_with_identity(
    key='video_id', num_buckets=1000000, default_value=0)
columns = [video_id, ...]
features = tf.parse_example(..., features=make_parse_example_spec(columns))
linear_prediction, _, _ = linear_model(features, columns)
```

Embedding for a DNN model:

```
columns = [embedding_column(video_id, 9),...]
features = tf.parse_example(..., features=make_parse_example_spec(columns))
dense_tensor = input_layer(features, columns)
```

Args:

- `key`: A unique string identifying the input feature. It is used as the column name and the dictionary key for feature parsing configs, feature `Tensor` objects, and feature columns.
- `num_buckets`: Range of inputs and outputs is `[0, num_buckets)`.
- `default_value`: If `None`, this column's graph operations will fail for out-of-range inputs. Otherwise, this value must be in the range `[0, num_buckets)`, and will replace inputs in that range.

Returns:

A `_CategoricalColumn` that returns identity values.

Raises:

- `ValueError`: if `num_buckets` is less than one.
- `ValueError`: if `default_value` is not in range `[0, num_buckets)`.
