tf.feature_column.weighted_categorical_column

```
weighted_categorical_column(
    categorical_column,
    weight_feature_key,
    dtype=tf.float32
)
```

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

Applies weight values to a `_CategoricalColumn`.

Use this when each of your sparse inputs has both an ID and a value. For example, if you're representing text documents as a collection of word frequencies, you can provide 2 parallel sparse input features ('terms' and 'frequencies' below).

Example:

Input `tf.Example` objects:

```
[
  features {
    feature {
      key: "terms"
      value {bytes_list {value: "very" value: "model"}}
    }
    feature {
      key: "frequencies"
      value {float_list {value: 0.3 value: 0.1}}
    }
  },
  features {
    feature {
      key: "terms"
      value {bytes_list {value: "when" value: "course" value: "human"}}
    }
    feature {
      key: "frequencies"
      value {float_list {value: 0.4 value: 0.1 value: 0.2}}
    }
  }
]
```

```
categorical_column = categorical_column_with_hash_bucket(
    column_name='terms', hash_bucket_size=1000)
weighted_column = weighted_categorical_column(
    categorical_column=categorical_column, weight_feature_key='frequencies')
columns = [weighted_column, ...]
features = tf.parse_example(..., features=make_parse_example_spec(columns))
linear_prediction, _, _ = linear_model(features, columns)
```

This assumes the input dictionary contains a `SparseTensor` for key 'terms', and a `SparseTensor` for key 'frequencies'. These 2 tensors must have the same indices and dense shape.

Args:

- `categorical_column`: A `_CategoricalColumn` created by `categorical_column_with_*` functions.
- `weight_feature_key`: String key for weight values.
- `dtype`: Type of weights, such as `tf.float32`. Only float and integer weights are supported.

Returns:

A `_CategoricalColumn` composed of two sparse features: one represents id, the other represents weight (value) of the id feature in that example.

Raises:

- `ValueError`: if `dtype` is not convertible to float.

Last updated November 2, 2017.