

tf.feature_column.embedding_column

```
embedding_column(
    categorical_column,
    dimension,
    combiner='mean',
    initializer=None,
    ckpt_to_load_from=None,
    tensor_name_in_ckpt=None,
    max_norm=None,
    trainable=True
)
```

Defined in [tensorflow/python/feature_column/feature_column.py](#).

_DenseColumn that converts from sparse, categorical input.

Use this when your inputs are sparse, but you want to convert them to a dense representation (e.g., to feed to a DNN).

Inputs must be a **_CategoricalColumn** created by any of the **categorical_column_*** function. Here is an example embedding of an identity column for a DNN model:

```
video_id = categorical_column_with_identity(
    key='video_id', num_buckets=1000000, default_value=0)
columns = [embedding_column(video_id, 9),...]
features = tf.parse_example(..., features=make_parse_example_spec(columns))
dense_tensor = input_layer(features, columns)
```

Args:

- categorical_column**: A **_CategoricalColumn** created by a **categorical_column_with_*** function. This column produces the sparse IDs that are inputs to the embedding lookup.
- dimension**: An integer specifying dimension of the embedding, must be > 0.
- combiner**: A string specifying how to reduce if there are multiple entries in a single row. Currently 'mean', 'sqnrn' and 'sum' are supported, with 'mean' the default. 'sqnrn' often achieves good accuracy, in particular with bag-of-words columns. Each of this can be thought as example level normalizations on the column. For more information, see **tf.embedding_lookup_sparse**.
- initializer**: A variable initializer function to be used in embedding variable initialization. If not specified, defaults to **tf.truncated_normal_initializer** with mean **0.0** and standard deviation **1/sqrt(dimension)**.
- ckpt_to_load_from**: String representing checkpoint name/pattern from which to restore column weights. Required if **tensor_name_in_ckpt** is not **None**.
- tensor_name_in_ckpt**: Name of the **Tensor** in **ckpt_to_load_from** from which to restore the column weights. Required if **ckpt_to_load_from** is not **None**.
- max_norm**: If not **None**, embedding values are l2-normalized to this value.
- trainable**: Whether or not the embedding is trainable. Default is True.

Returns:

`_DenseColumn` that converts from sparse input.

Raises:

- `ValueError`: if `dimension` not > 0.
- `ValueError`: if exactly one of `ckpt_to_load_from` and `tensor_name_in_ckpt` is specified.
- `ValueError`: if `initializer` is specified and is not callable.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

Support

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

English

[Terms](#) | [Privacy](#)