tf.layers.dense

```
dense(
    inputs,
    units,
    activation=None,
    use_bias=True,
    kernel_initializer=None,
    bias_initializer=tf.zeros_initializer(),
    kernel_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    kernel_constraint=None,
    bias_constraint=None,
    trainable=True,
    name=None,
    reuse=None
)
```

Defined in `tensorflow/python/layers/core.py`.

Functional interface for the densely-connected layer.

This layer implements the operation: `outputs = activation(inputs.kernel + bias)` Where `activation` is the activation function passed as the `activation` argument (if not `None`), `kernel` is a weights matrix created by the layer, and `bias` is a bias vector created by the layer (only if `use_bias` is `True`).

★ **Note:** if the `inputs` tensor has a rank greater than 2, then it is flattened prior to the initial matrix multiply by `kernel`.

Arguments:

- `inputs`: Tensor input.
- `units`: Integer or Long, dimensionality of the output space.
- `activation`: Activation function (callable). Set it to None to maintain a linear activation.
- `use_bias`: Boolean, whether the layer uses a bias.
- `kernel_initializer`: Initializer function for the weight matrix. If `None` (default), weights are initialized using the default initializer used by `tf.get_variable`.
- `bias_initializer`: Initializer function for the bias.
- `kernel_regularizer`: Regularizer function for the weight matrix.
- `bias_regularizer`: Regularizer function for the bias.
- `activity_regularizer`: Regularizer function for the output.
- `kernel_constraint`: An optional projection function to be applied to the kernel after being updated by an `Optimizer` (e.g. used to implement norm constraints or value constraints for layer weights). The function must take as input the unprojected variable and must return the projected variable (which must have the same shape). Constraints are not safe to use when doing asynchronous distributed training.
- `bias_constraint`: An optional projection function to be applied to the bias after being updated by an `Optimizer`.
- `trainable`: Boolean, if `True` also add variables to the graph collection `GraphKeys.TRAINABLE_VARIABLES` (see

`tf.Variable`).

- `name`: String, the name of the layer.
- `reuse`: Boolean, whether to reuse the weights of a previous layer by the same name.

Returns:

Output tensor.
