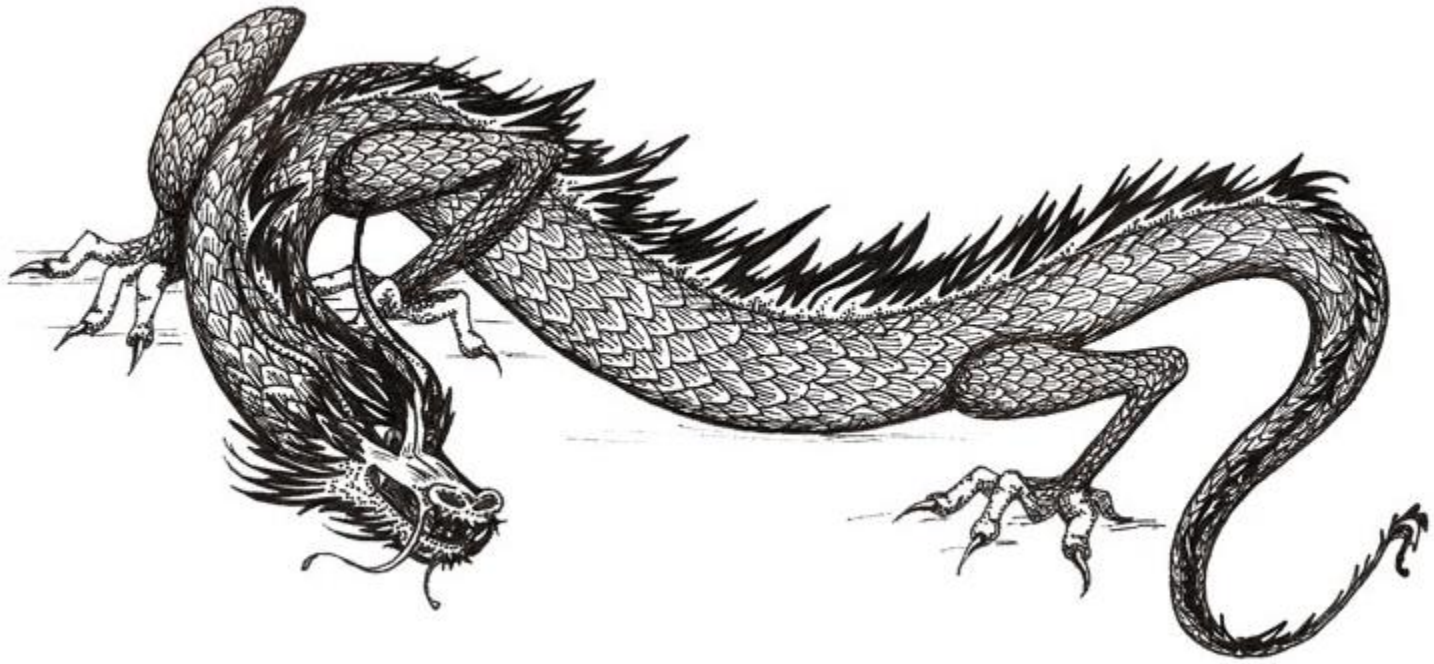

自定义网络

主讲：龙良曲

Outline

- keras.Sequential
- keras.layers.Layer
- keras.Model

keras.Sequential

```
network = Sequential([layers.Dense(256, activation='relu'),
                      layers.Dense(128, activation='relu'),
                      layers.Dense(64, activation='relu'),
                      layers.Dense(32, activation='relu'),
                      layers.Dense(10)])
network.build(input_shape=(None, 28*28))
network.summary()
```

- model.trainable_variables
- model.call()

Layer/Model

- Inherit from keras.layers.Layer keras.Model

- __init__
- call
- Model: compile/fit/evaluate

自定义层

```python
class MyDense(layers.Layer):

    def __init__(self, inp_dim, outp_dim):
        super(MyDense, self).__init__()

        self.kernel = self.add_variable('w', [inp_dim, outp_dim])
        self.bias = self.add_variable('b', [outp_dim])

    def call(self, inputs, training=None):

        out = inputs @ self.kernel + self.bias

        return out
```

自定义网络

```python
class MyModel(keras.Model):
    def __init__(self):
        super(MyModel, self).__init__()
        self.fc1 = MyDense(28*28, 256)
        self.fc2 = MyDense(256, 128)
        self.fc3 = MyDense(128, 64)
        self.fc4 = MyDense(64, 32)
        self.fc5 = MyDense(32, 10)

    def call(self, inputs, training=None):
        x = self.fc1(inputs)
        x = tf.nn.relu(x)
        x = self.fc2(x)
        x = tf.nn.relu(x)
        x = self.fc3(x)
        x = tf.nn.relu(x)
        x = self.fc4(x)
        x = tf.nn.relu(x)
        x = self.fc5(x)
        return x
```


JUST
DO
IT.

下一课时

模型加载与保存

Thank You.