1. 用于2D向量的shape是fully connected layer，为layer\_desnse，3D的话，layer\_lstm，4D的话就是layer\_conv\_2d
2. Keras使用步骤：
3. 定义训练数据，包括输入tensor和目标tensor
4. 定义神经网络层，然后map你的输入和target
5. 选择损失函数和优化器，还有metrics如accuracy
6. 迭代，并用fit来使用你的模型
7. shuffle数据看如果是只有训练集和测试机，那直接indices <- sample(1:nrow(), size = 0.8\*nrow())，如果是k-fold的话，indices <- sample(…)，fold <- cut(indices, breaks= ..,labels = FALSE)
8. 避免过拟合问题，可以通过，增加训练数据，减少神经网络数量，加入正则惩罚项以及dropout方法
9. universal workflow of machine learning
10. 对你的问题要做出一个定义，你的输入数据是什么？你想要预测什么？你要解决什么问题？分类，标量回归？矢量回归？
11. 选择度量模型成功的方法，这会引导你选择不同的损失函数。
12. Deciding on an evaluation protocol.使用hold-out方法，或者是k fold cross-validation
13. 对数据要记得预处理，神经网络对输入数据比较敏感，所以要归一化，转换成tensors
14. 建立模型打败baseline，注意我们做出的两个假设，一是给定输入我们可以预测输出，二是数据包含的信息足够我们在输入和输出间建立联系。三个key choices：
15. 最后一层的activation，分类会使用sigmoid，线性就不需要等等
16. 损失函数，IMDB用的是binary-crossentropy，回归用的是mse
17. Optimizer：包括用哪种，rmsprop还是adam，学习率是多少，自适应的学习率默认会给
18. 检查是否出现overfit以及underfit
19. 添加层数
20. 添加层中神经元
21. 多epochs训练
22. 正则化以及超参数的tuning