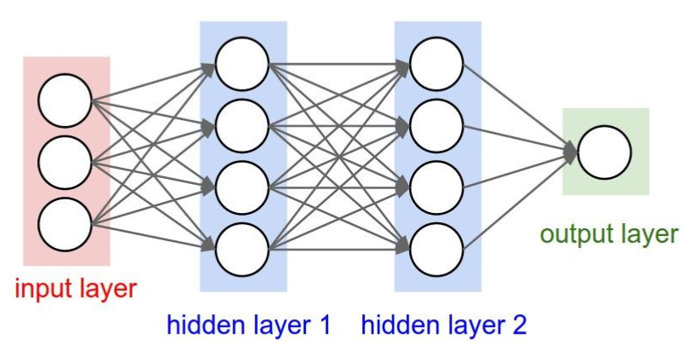
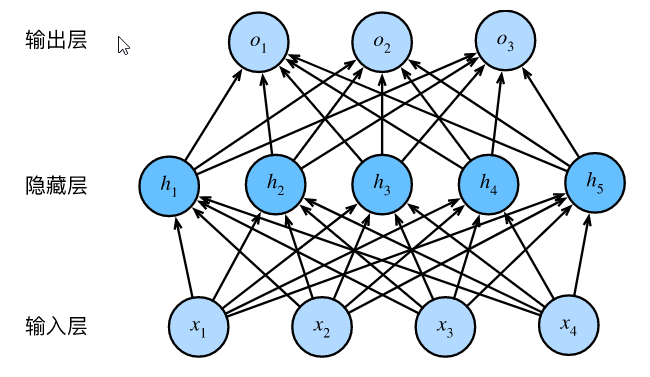
1 多层感知器也叫全连接神经网络，网络结构不包括（）

https://www.biaodianfu.com/imdb-sentiment-analysis-full-connection.html





答案: 多层感知器也叫全连接神经网络，网络结构包括 输入层 隐藏层 输出层

不包括反馈层

2 Keras有两种类型的模型，包括（）

答案：<https://blog.csdn.net/shentanyue/article/details/82318233>

Keras有两种类型的模型，[序贯模型（Sequential）](https://blog.csdn.net/shentanyue/article/details/82318095)和[函数式模型（Model）](https://blog.csdn.net/shentanyue/article/details/82318162)，函数式模型应用更为广泛，序贯模型是函数式模型的一种特殊情况。

两类模型有一些方法是相同的：

* model.summary()：打印出模型概况，它实际调用的是keras.utils.print\_summary
* model.get\_config():返回包含模型配置信息的Python字典。模型也可以从它的config信息中重构回去

3 Jupyter Notebook的本质是一个（）应用程序，便于创建和共享文学化程序文档，支持实时代码，数学方程和可视化。

答案: jupyter notebook(也叫IPython notebook)是一个交互式笔记本。  
Jupyter Notebook 的本质是一个 Web 应用程序，便于创建和共享文学化程序文档，支持实时代码，数学方程，可视化和 markdown。  
有很多集成库内包含jupyter notebook，例如sagemath，anaconda  
本文使用anaconda自带的jupyter notebook。

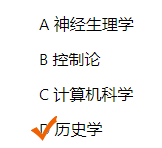
4 简述jupyter notebook的 特点

https://blog.csdn.net/laziji/article/details/106179787

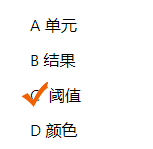
1. 编程时具有**语法高亮**、*缩进*、*tab补全*的功能。
2. 可直接通过浏览器运行代码，同时在代码块下方展示运行结果。
3. 以富媒体格式展示计算结果。富媒体格式包括：HTML，LaTeX，PNG，SVG等。
4. 对代码编写说明文档或语句时，支持Markdown语法。
5. 支持使用LaTeX编写数学性说明。

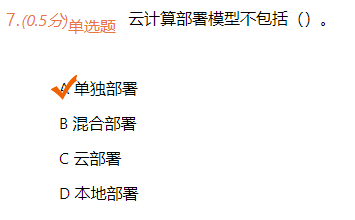
5 人工智能，是一门由（）等多种学科相互渗透而发展起来的综合性新学科，下面哪种不属于该范畴？

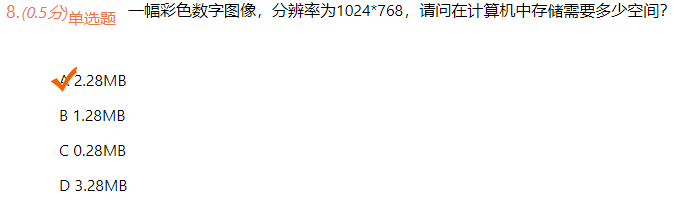
答案: 它是由计算机科学、控制论、信息论、神经生理学、哲学、语言学等多种学科相互渗透而发展起来的

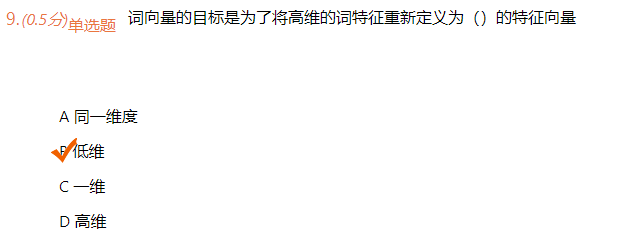


6 感知器接受每个感知元（神经元）传输过来的数据，当数据到达某个（阈值）的时候，就会产生对应的行为。

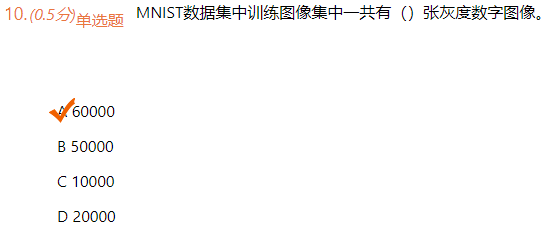


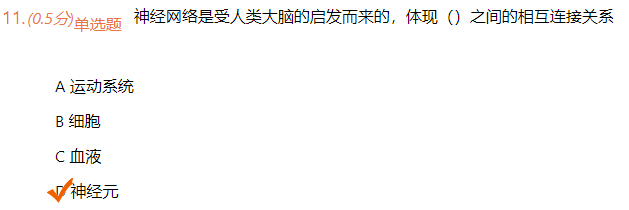


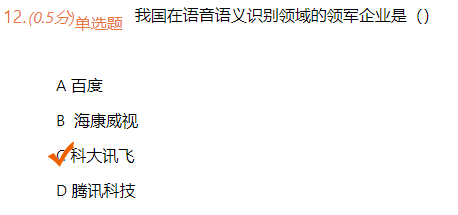




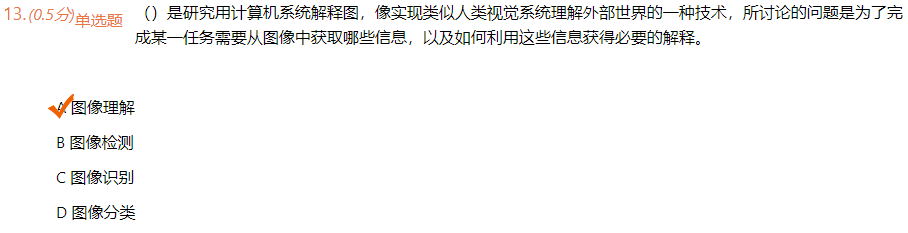
答案:词向量(Word Embedding)(text vectors) Word embedding是一种非常现代的用向量表示单词的方式。它的目标是为了将高维的词特征重新定义为低维的特征向量



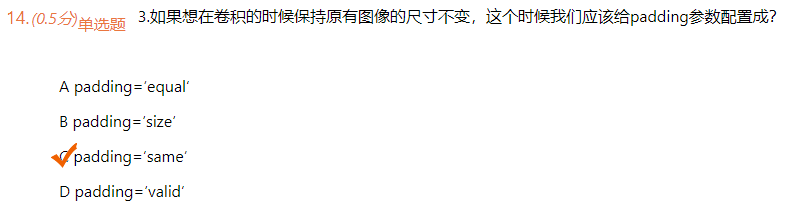


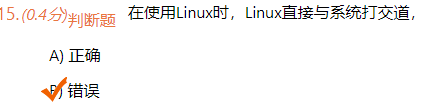


答案: 我国在语音语义识别领域的领军企业是（科大讯飞）



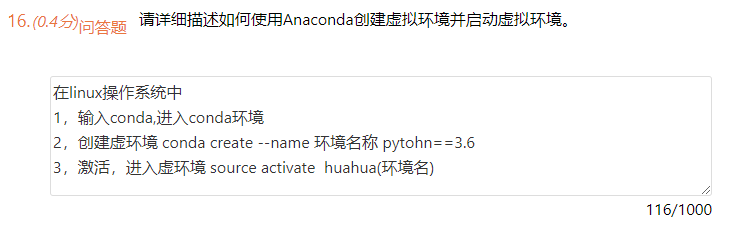
（图像理解）是研究用计算机系统解释图，像实现类似人类视觉系统理解外部世界的一种技术，所讨论的问题是为了完成某一任务需要从图像中获取哪些信息，以及如何利用这些信息获得必要的解释。

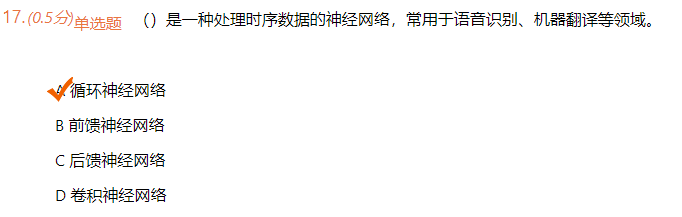


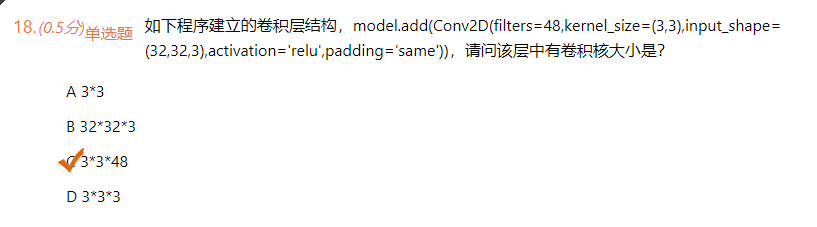


<https://blog.csdn.net/qq_42176520/article/details/81324886>

通常我们在使用 Linux 时，并不是直接与系统打交道，而是通过一个叫做 Shell 的中间程序来完成的，在图形界面下为了实现让我们在一个窗口中完成用户输入和显示输出，Linux 系统还提供了一个叫做终端模拟器的程序（Terminal）







选D



784+1 \* 1000 785000

1000+1 \*428 428428

428+1 \*4 1716

From keras.models import Sequential

From keras.layers import Dense,Dropout

Model=Sequential()

Model.add(Dense(units=1000,input\_dim=784, Kernel\_initializer=’normal’,activation=’relu’))

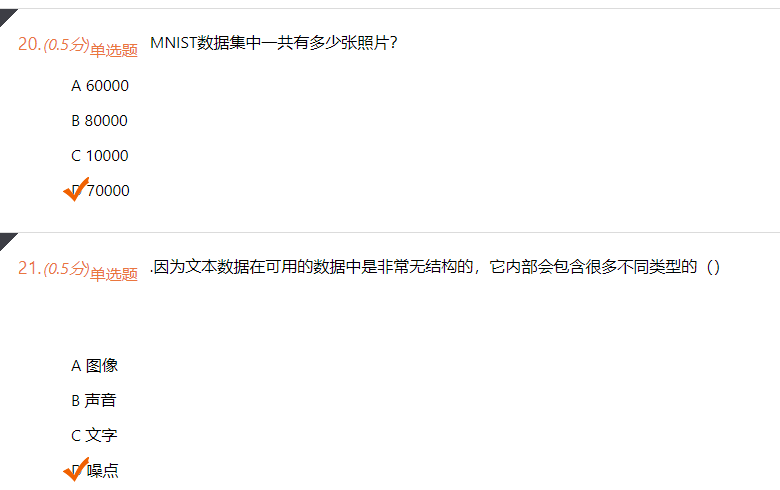
Model.droppout((0.5))

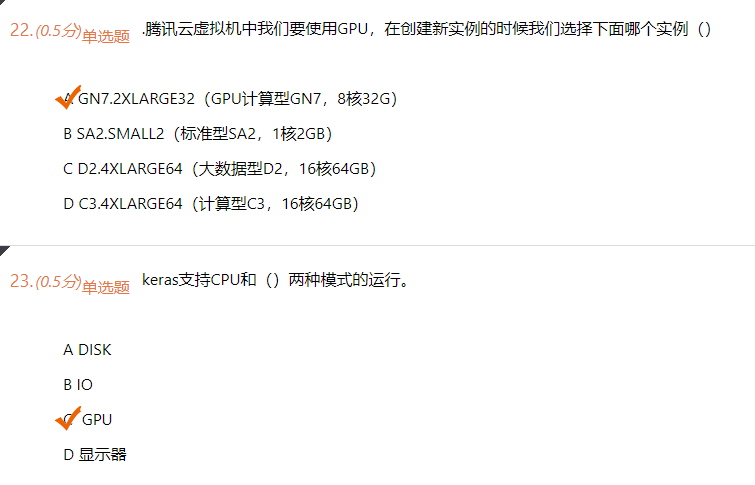
Model.add(Dense(units=428, Kernel\_initializer=’normal’,activation=’relu’))

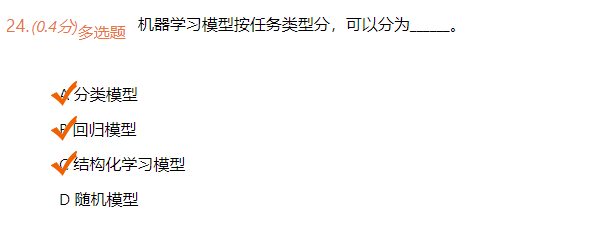
Model.droppout((0.25))

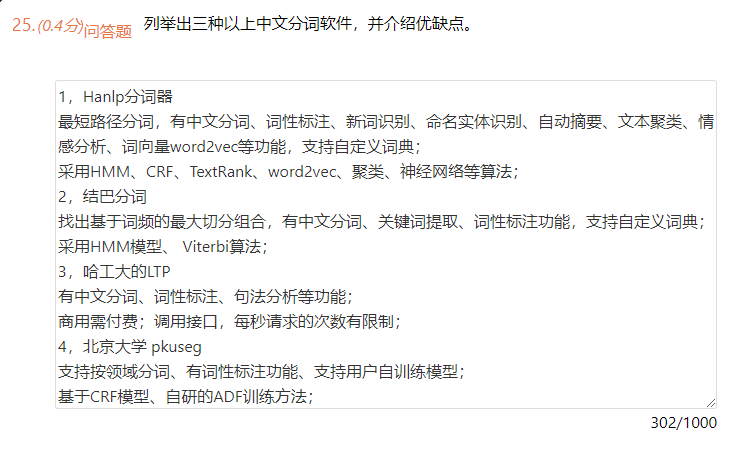
Model.add(Dense(units=4, Kernel\_initializer=’normal’,activation=’softmax’))

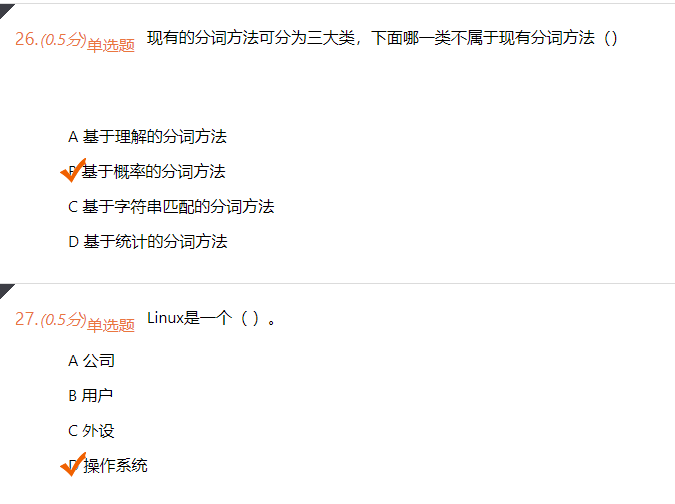
Print(model.summary())











1 将资本投入变成可变收入

2 大范围规模经济的优势

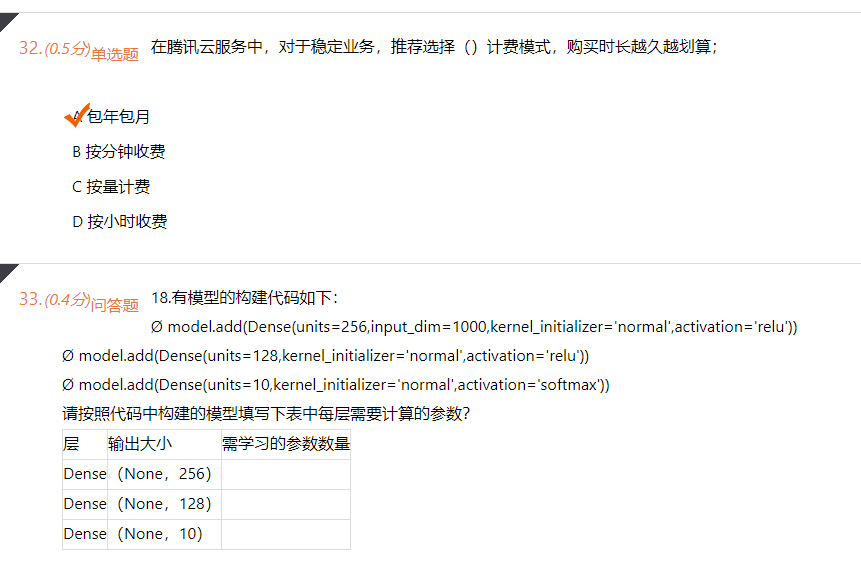
3 无需预估容量

4 增加速度和灵活性

5 无需为运行和维护数据中心额外投资

6 数分钟内实现全球化部署





1000+1 \* 256 256256

256+1 \*128 32896

128+1 \*10 1290

Kernel\_initializer=’normal’



选 C



From keras.models import Sequential

From keras.layers import Dense

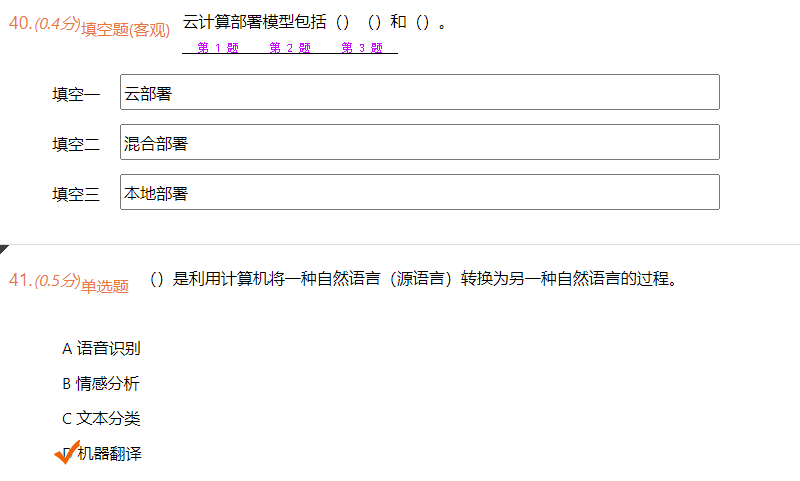
Model=Sequential()

Model.add(Dense(units=256,input\_dim=784,kernel\_initializer=’normal’,activation=’relu’))

Model.add(Dense(units=10,kernel\_initializer=’normal’,activation=’softmax’))

Print(model.summary())



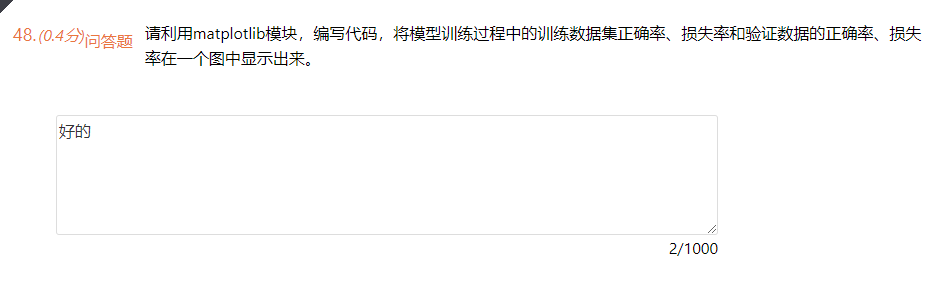






44选 C





Import matplotlib.pyplot as plt

Def show\_train(train\_history,train,validation):

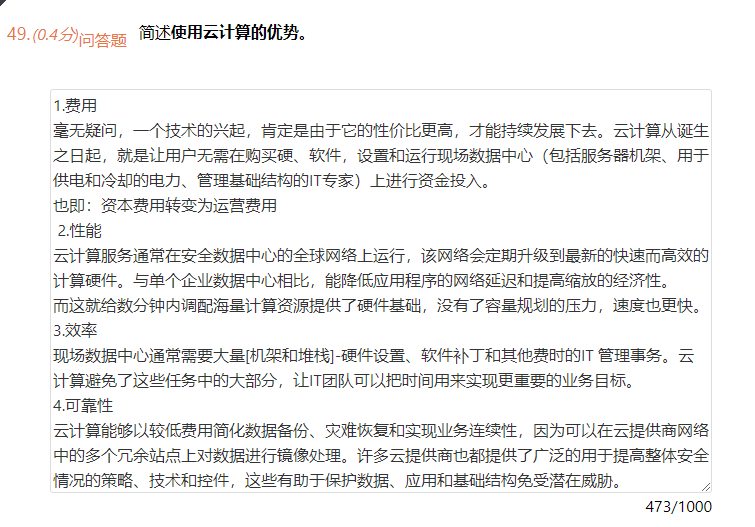
Plt.plot(train\_history.history[train])

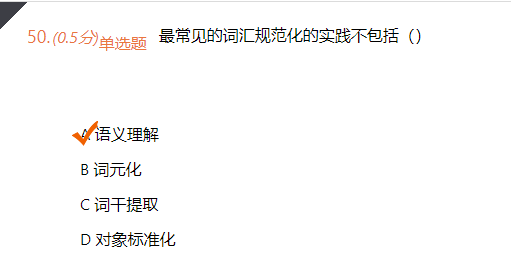
Plt.plot(train\_history.history[validation])

Plt.show()

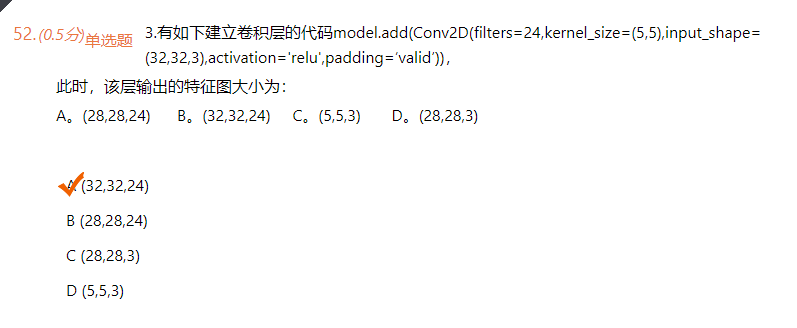
Show\_train(train\_history,’acc’,’val\_acc’)

Show\_train(train\_history,’loss’,’val\_loss’)



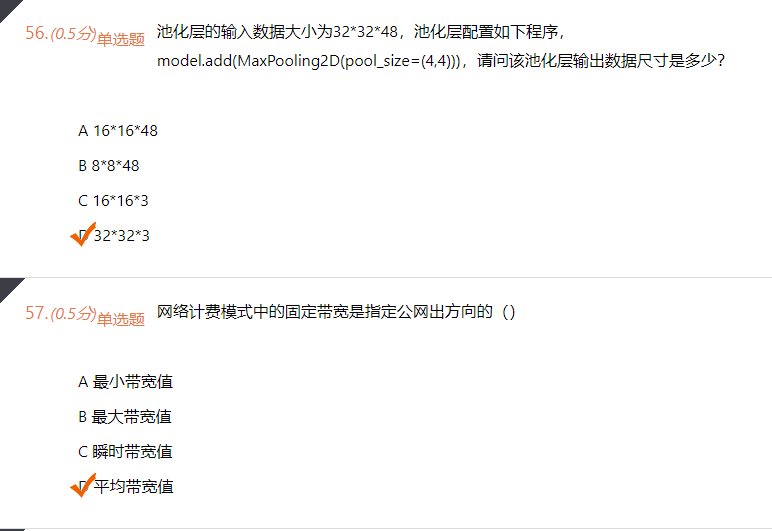




A



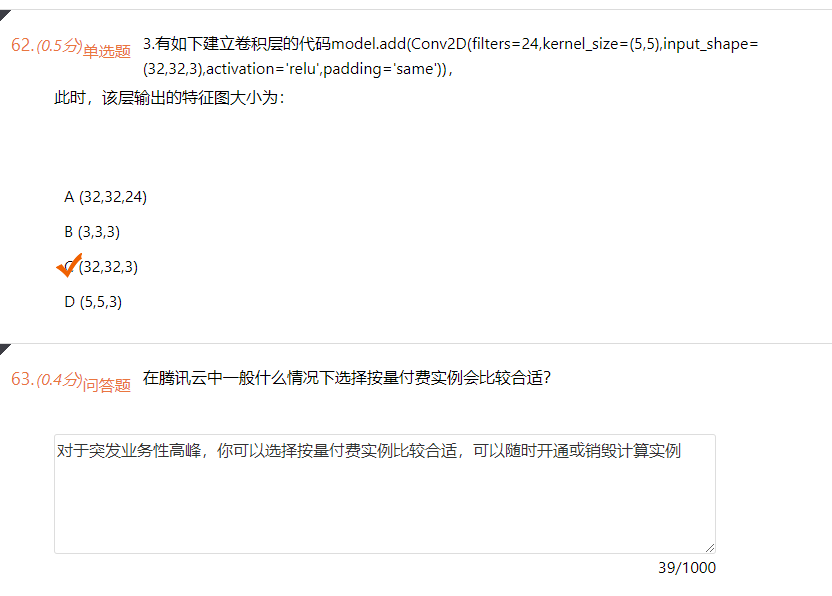




56 选 B







62 选A











73选A



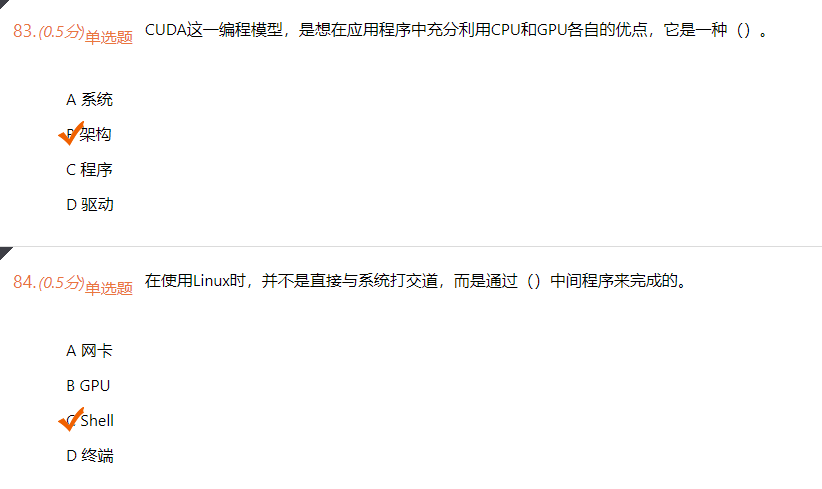


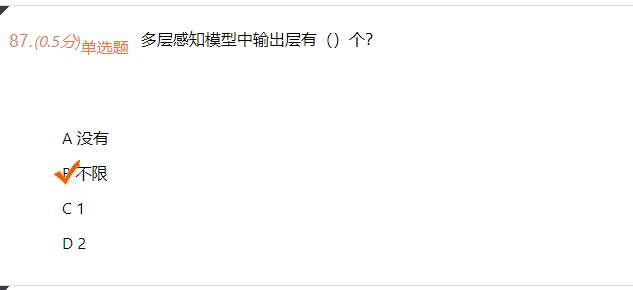
78选 CD



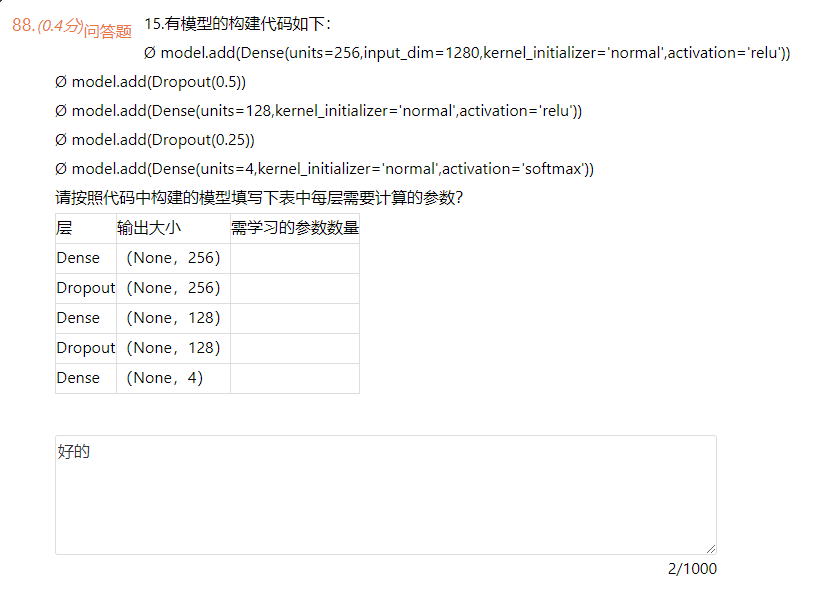


81选 204800KB=640\*320\*8/8

‘



87xuan c



327936

0

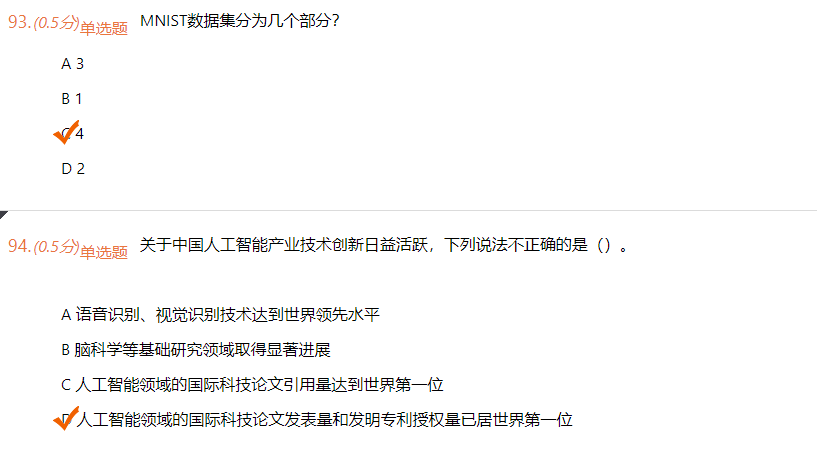
32896

0

516





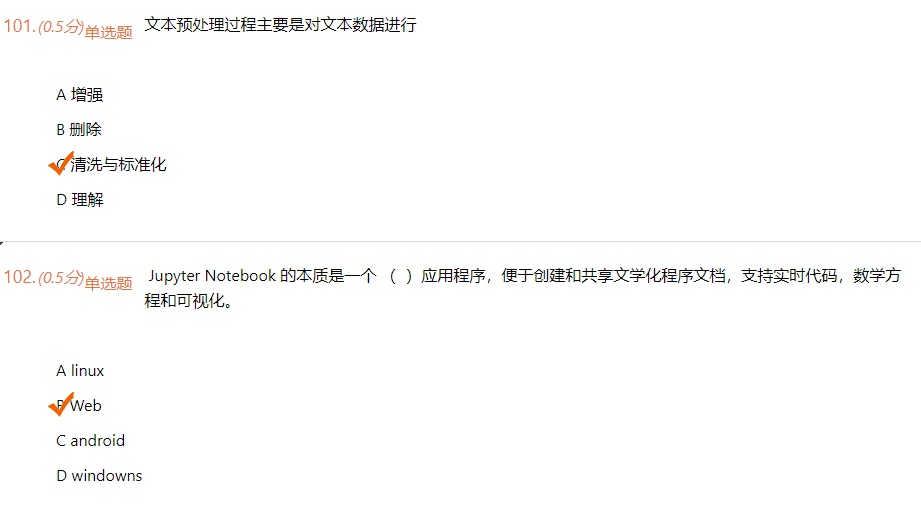








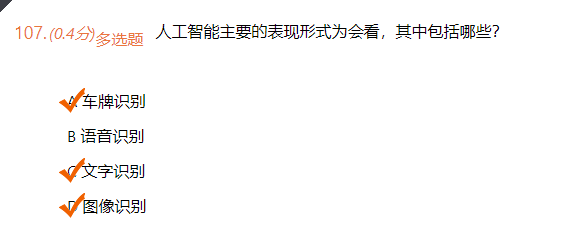
100 xuan B



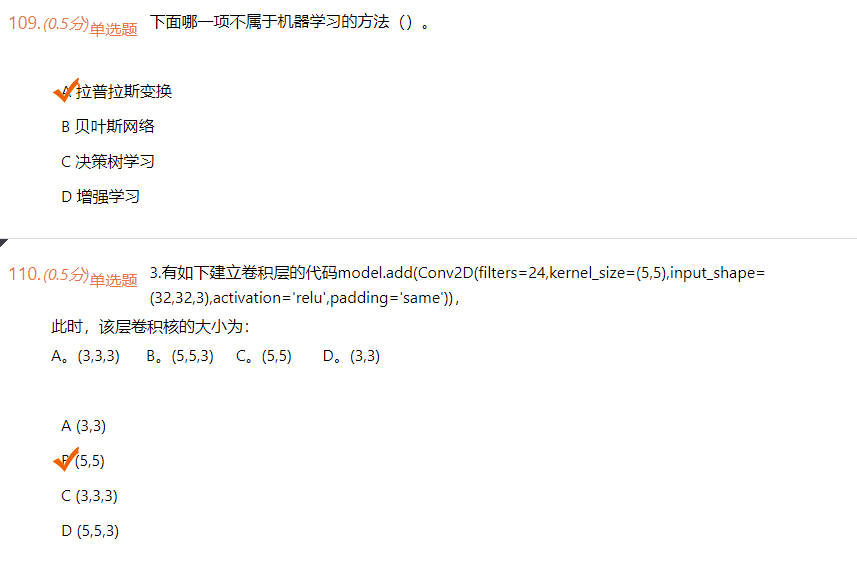


选 A 784

哦了。

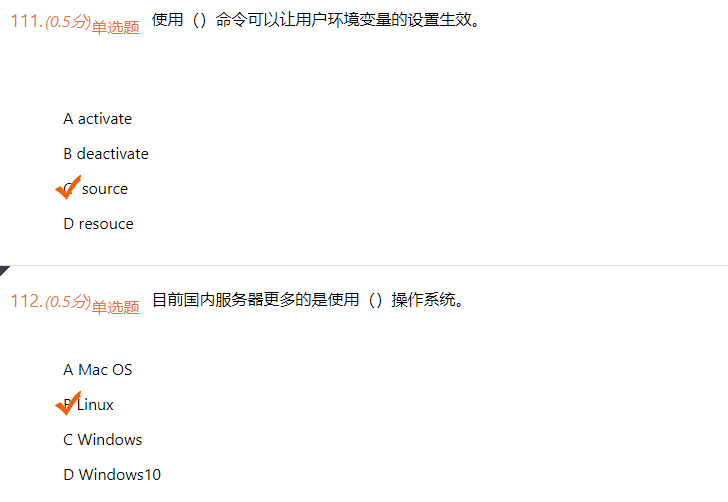






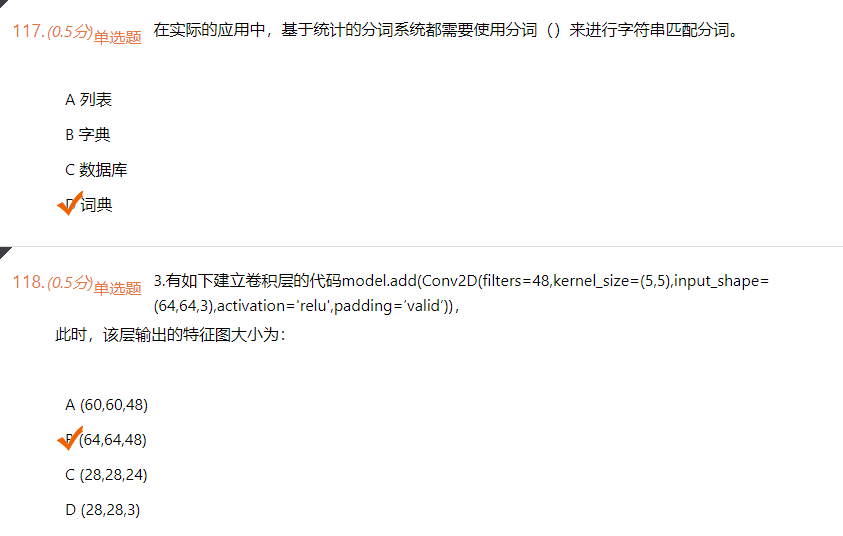
110选 D

109 xuan D









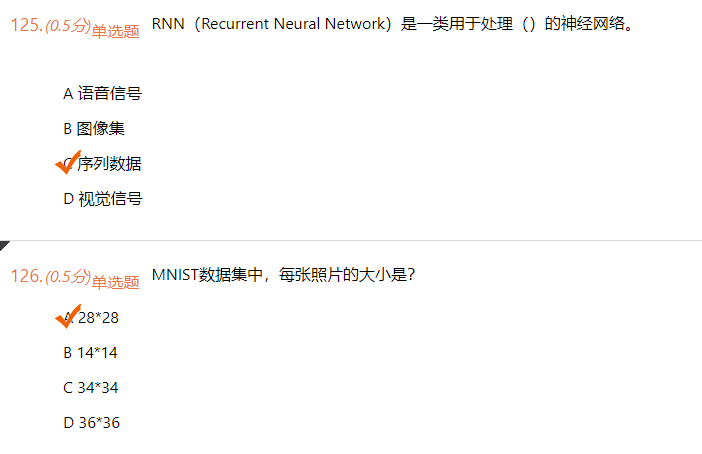




121 选 D

122 选A

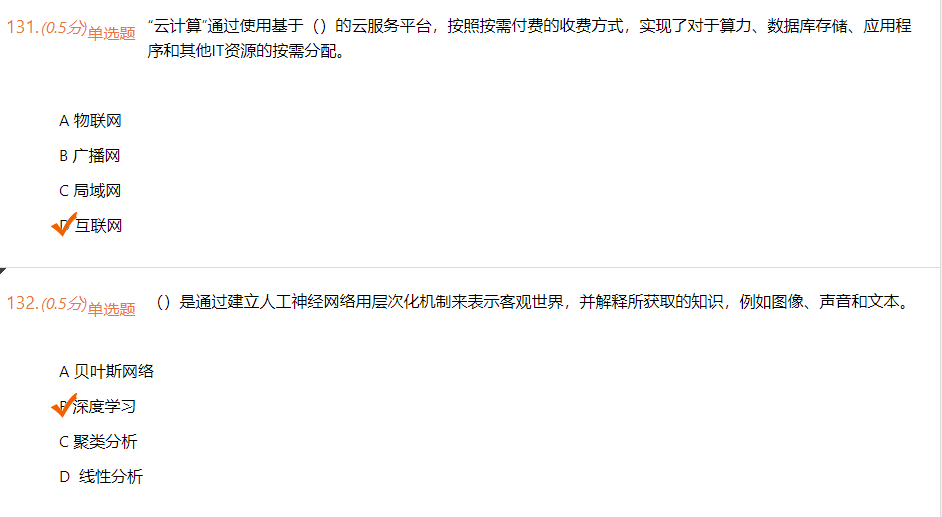






128 选 D

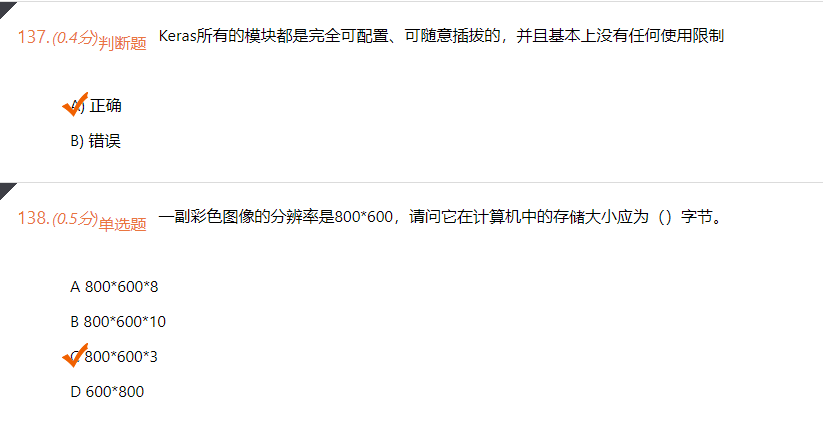




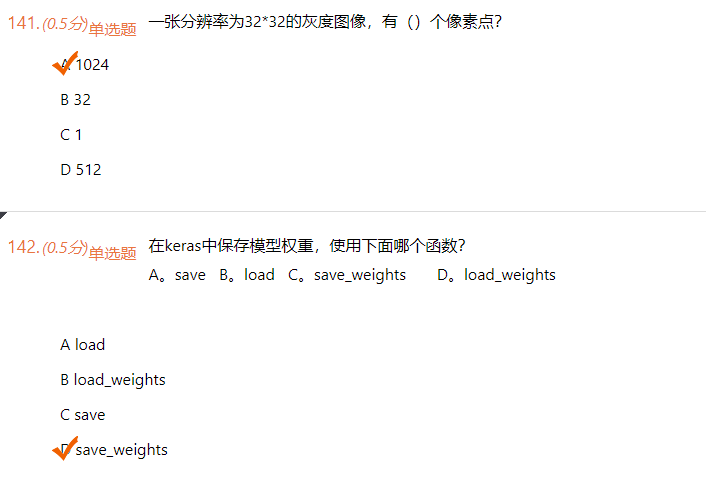


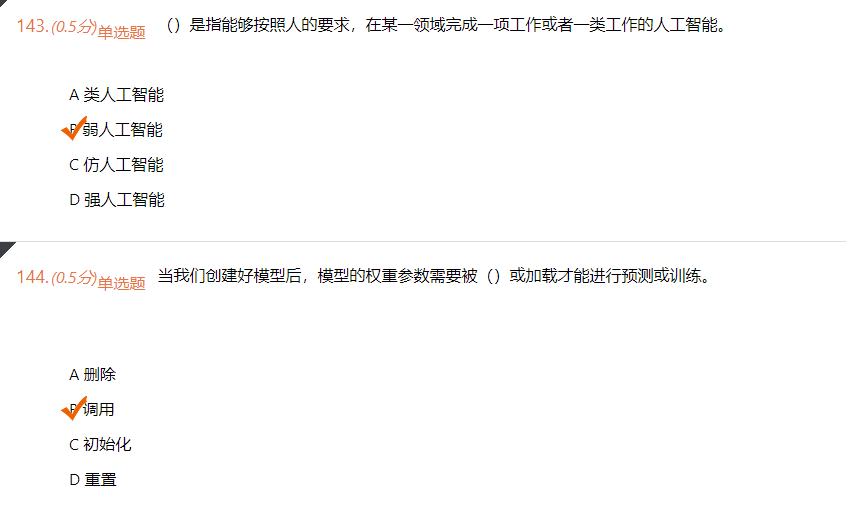


29









C

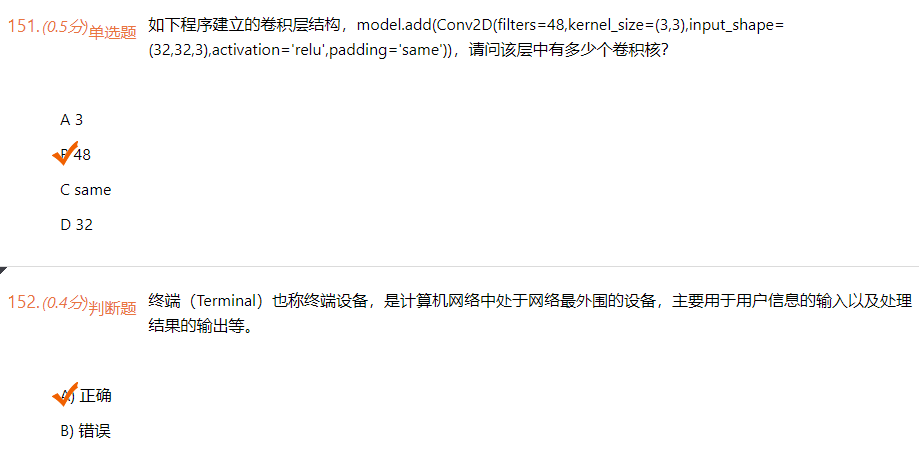


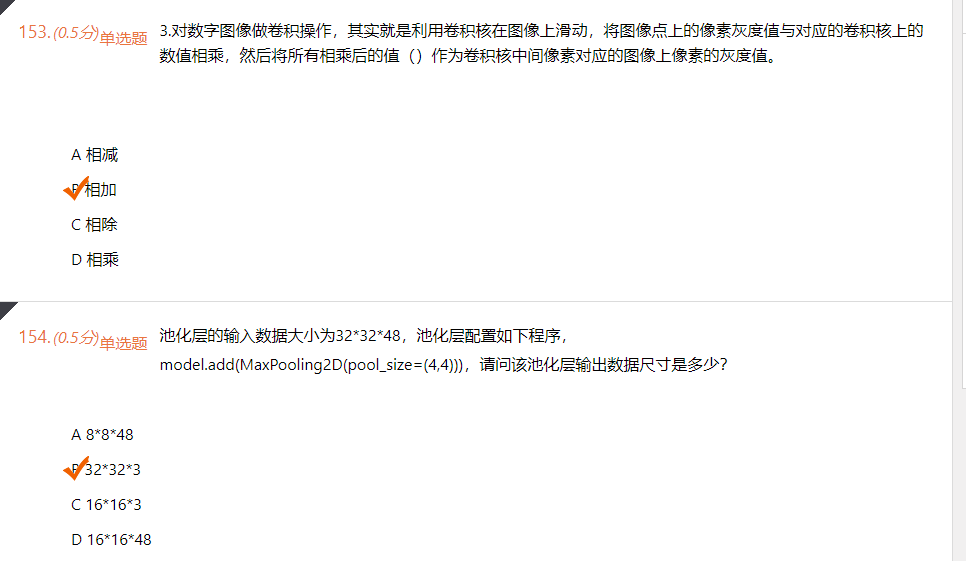




149 D

150 X = train\_image.reshape(1296).astype(‘float’)

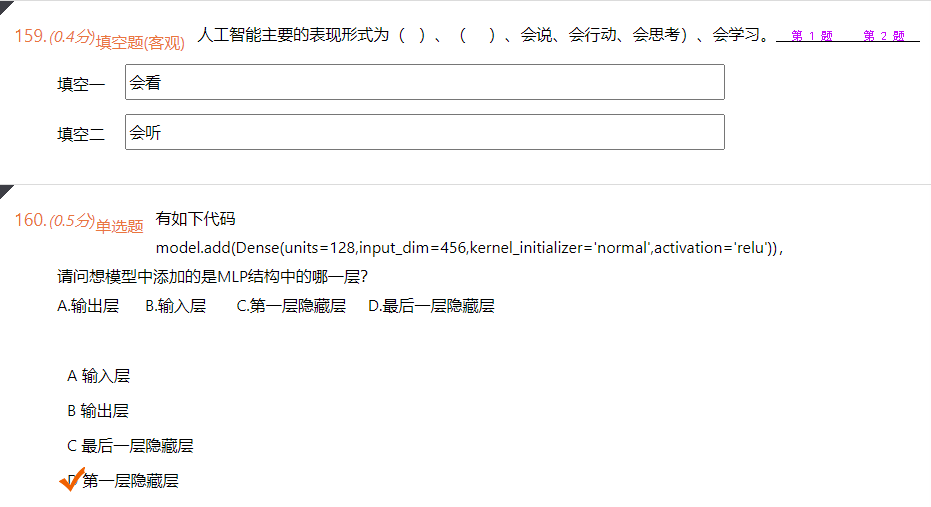




154 选A







160 选A







165 选A 输入层\*隐藏层+隐藏层\*输出层=150

权重参数 输入层+1 \*隐藏层 +



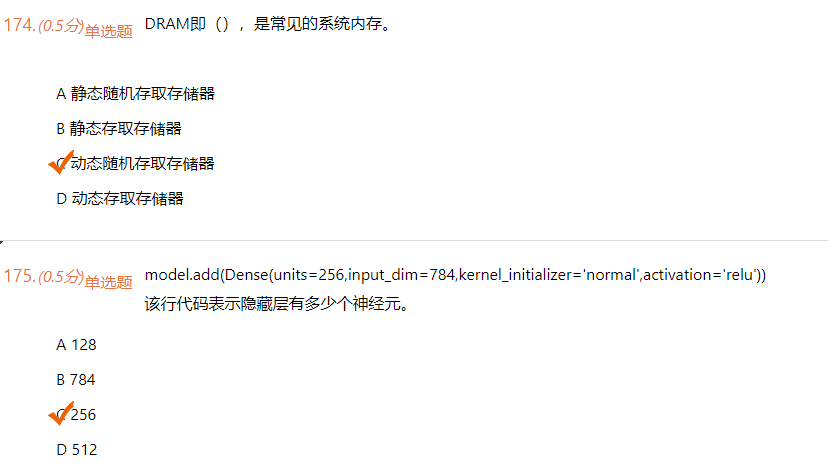


170 B

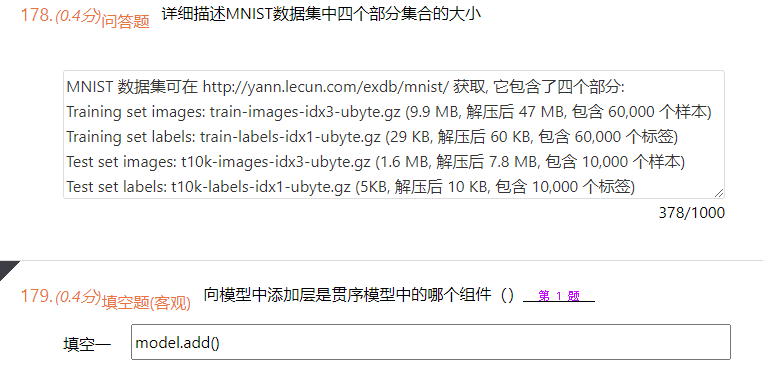








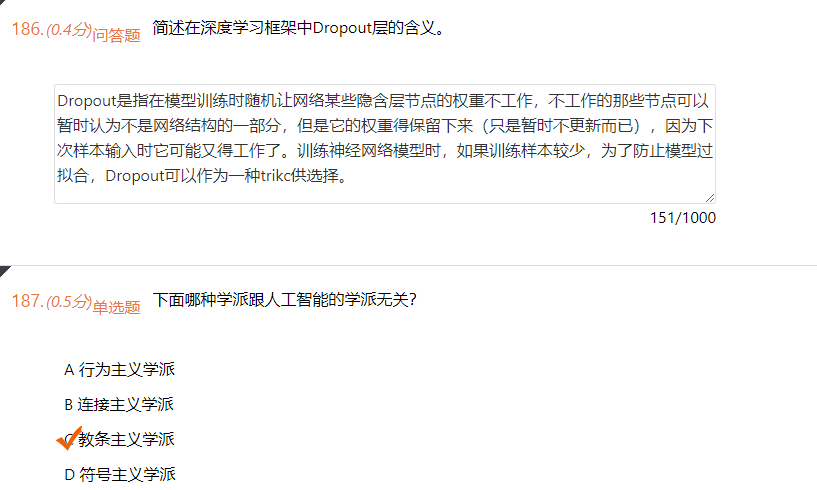










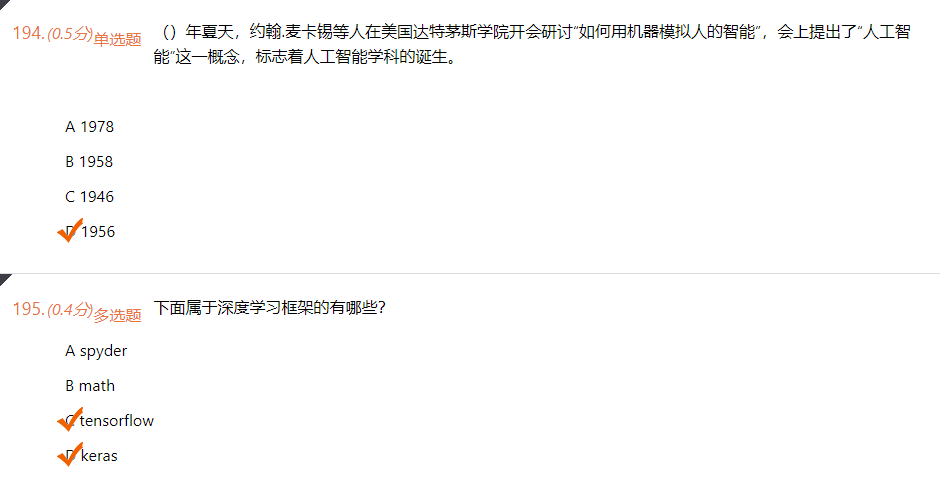








192 xuan D





200

10

2000



200 xuan B

