Q1. Which of the following function cannot be used as activation function?

- max(0, x)
- 1 / (1 + e^x)
- 2*x - 1 [correct answer, activation function must be non-linear]
- (e^x - 1) / (e^x + 1)

Q2. Which of the following application should not use LSTM?

- Face detection in photos [correct answer, LSTM is for sequence data]
- Virtual assistant system, like Siri
- Surveillance system on road
- Document (text) understanding

Q3. Describe main difference between DNN, CNN and RNN.

广义上讲，CNN，RNN也都是DNN的一种，但是狭义上讲DNN指的是只有full connected layer的network，CNN是有convolutional layer和pooling layer的，RNN是有时间这个维度的，不是单个输入。

从应用上讲，只有fully connected layer的DNN其实不常用，不过fully connected layer本身非常有用，CNN和RNN中也会用到fully connected layer。CNN用于图像相关的应用，RNN用于序列类型的数据，比如语音，文本和视频。

Q4. What’s the benefit of designing a very deep network? What’s the potential problem it bring? Any method to solve it?

理论上网络深度越深，表达问题的复杂度可以越高，学习的能力越强。但是同时训练的难度，队训练数据量的要求以及训练时间都增高。Dropout，transfer learning都是对应的方法。

Q5. Coding problem

Implement a DNN:

1) Input layer has 10 nodes, it also means the input is a vector with length 10
2) Followed by a FC (fully connected) layer with 20 nodes, activation function ReLu
3) Followed by a FC layer with 15 nodes, activation function ReLu
4) Output layer is also a FC layer with 5 nodes, activation function Sigmoid

Also calculate how many parameters in the DNN need to be trained.

Codes: [红色是答案]

```
model = Sequential()
# TODO: FC layer with 20 nodes, Relu
model.add(Dense(20, input_shape=(10,)))
model.add(Activation(‘relu’))
# TODO: FC layer with 15 nodes, Relu
```

```
model.add(Dense(15))
model.add(Activation(‘relu’))
# TODO: FC layer with 5 nodes, Sigmoid
model.add(Dense(5))
model.add(Activation(‘sigmoid’))
```

#params = 10 * 20 + 20 + 20 * 15 + 15 + 15 * 5 + 5

Activation function 只是迷惑，因为activation function其实没有需要训练的参数。

Q6. Coding problem

Implement a CNN:

1) Takes a 32x32 RGB image as input
2) Followed by a Convolutional layer with 10 5*5 kernels, activation function relu
3) Followed by a 2x2 max pooling layer
4) Followed by a fully-connected layer with 10 nodes, activation function softmax

Also calculate how many parameters in the CNN need to be trained.

Codes: [红色是答案]

```
model = Sequential()
# convolutional layer with 10 5x5 kernels, activation function relu, input image is 32x32x3
model.add(Conv2D(10, kernel_size=(5, 5),
                 activation='relu',
                 input_shape=(32, 32, 3)))
# max pooling
model.add(MaxPooling2D(pool_size=(2, 2)))
# flatten before connected with fully-connected layer.
model.add(Flatten())  # 3D -> 1D
# fully connected layer with 10 nodes, softmax
model.add(Dense(num_category, activation='softmax'))
```

Calculate #params

1) Convolutional layer, each kernel has 5 * 5 * 3 + 1 = 76 params, 10 kernels in total 760
2) After conv layer, the image size change from 32x32x3 to 28x28x10, then after pooling layer it is 14x14x10
3) Flatten 14x14x10 to 1960d vector
4) Fully connected layer has 1960x10+10 = 19610 parameters

So in total 760 + 19610 = 20370 params to train