Homework 8

Writing Assignment

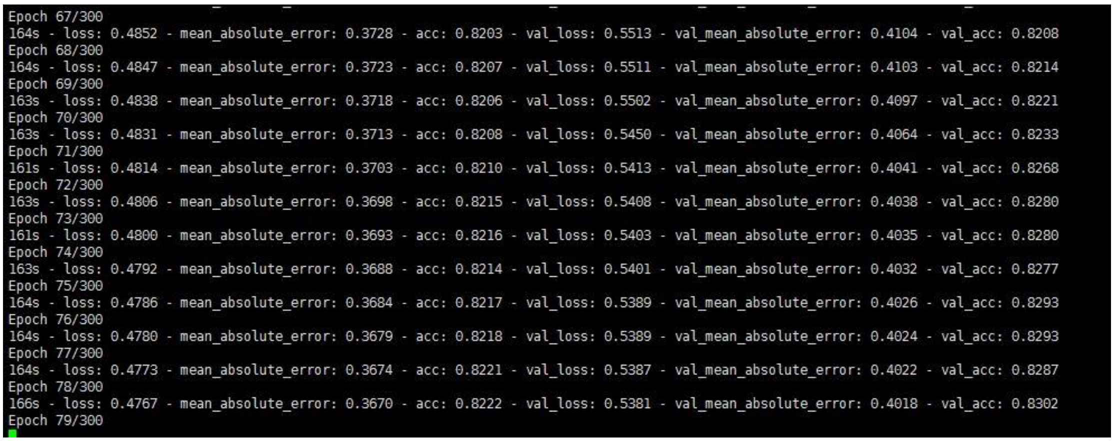
1. A:

(1) No. Considering the current model and algorithm, when the number of layers reaches a certain number, the effect of increasing the number of layers will become less and less obvious. But more problems (calculation force, gradient, activation function, etc.) need to be solved or it just leads to overfitting. This is where new approaches are needed to further improve. So only increasing the number of layers won't get you better results.

(2) An experiment from *https://zhuanlan.zhihu.com/p/26049468?ivk_sa=1024320u* :

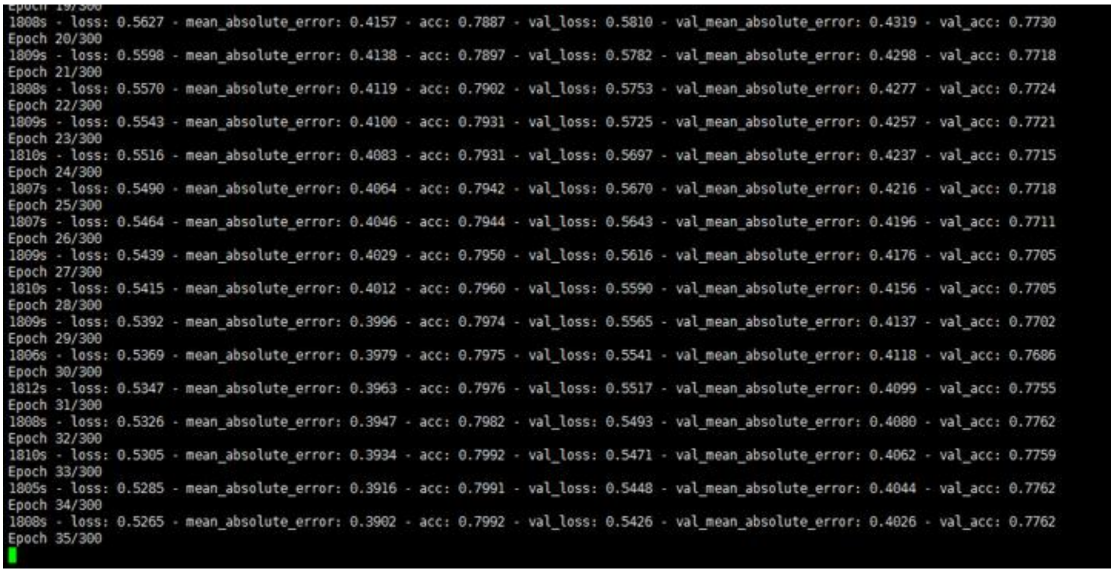
Experiment on heavy rain forecast (prediction of future heavy rain given a number of radar images and historical heavy rain data):

Single-layer CNN + LSTM:



```
Epoch 67/300
164s - loss: 0.4852 - mean_absolute_error: 0.3728 - acc: 0.8203 - val_loss: 0.5513 - val_mean_absolute_error: 0.4104 - val_acc: 0.8208
Epoch 68/300
164s - loss: 0.4847 - mean_absolute_error: 0.3723 - acc: 0.8207 - val_loss: 0.5511 - val_mean_absolute_error: 0.4103 - val_acc: 0.8214
Epoch 69/300
163s - loss: 0.4838 - mean_absolute_error: 0.3718 - acc: 0.8206 - val_loss: 0.5502 - val_mean_absolute_error: 0.4097 - val_acc: 0.8221
Epoch 70/300
163s - loss: 0.4831 - mean_absolute_error: 0.3713 - acc: 0.8208 - val_loss: 0.5450 - val_mean_absolute_error: 0.4064 - val_acc: 0.8233
Epoch 71/300
161s - loss: 0.4814 - mean_absolute_error: 0.3703 - acc: 0.8210 - val_loss: 0.5413 - val_mean_absolute_error: 0.4041 - val_acc: 0.8268
Epoch 72/300
163s - loss: 0.4806 - mean_absolute_error: 0.3698 - acc: 0.8215 - val_loss: 0.5408 - val_mean_absolute_error: 0.4038 - val_acc: 0.8280
Epoch 73/300
161s - loss: 0.4800 - mean_absolute_error: 0.3693 - acc: 0.8216 - val_loss: 0.5403 - val_mean_absolute_error: 0.4035 - val_acc: 0.8280
Epoch 74/300
163s - loss: 0.4792 - mean_absolute_error: 0.3688 - acc: 0.8214 - val_loss: 0.5401 - val_mean_absolute_error: 0.4032 - val_acc: 0.8277
Epoch 75/300
164s - loss: 0.4786 - mean_absolute_error: 0.3684 - acc: 0.8217 - val_loss: 0.5389 - val_mean_absolute_error: 0.4026 - val_acc: 0.8293
Epoch 76/300
164s - loss: 0.4780 - mean_absolute_error: 0.3679 - acc: 0.8218 - val_loss: 0.5389 - val_mean_absolute_error: 0.4024 - val_acc: 0.8293
Epoch 77/300
164s - loss: 0.4773 - mean_absolute_error: 0.3674 - acc: 0.8221 - val_loss: 0.5387 - val_mean_absolute_error: 0.4022 - val_acc: 0.8287
Epoch 78/300
166s - loss: 0.4767 - mean_absolute_error: 0.3670 - acc: 0.8222 - val_loss: 0.5381 - val_mean_absolute_error: 0.4018 - val_acc: 0.8302
Epoch 79/300
```

VGG19 + LSTM:

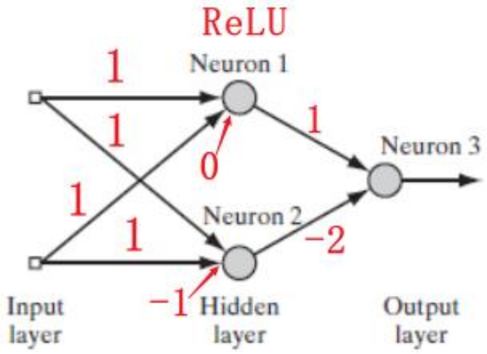


```
Epoch 19/300
1808s - loss: 0.5627 - mean_absolute_error: 0.4157 - acc: 0.7887 - val_loss: 0.5810 - val_mean_absolute_error: 0.4319 - val_acc: 0.7730
Epoch 20/300
1809s - loss: 0.5598 - mean_absolute_error: 0.4138 - acc: 0.7897 - val_loss: 0.5782 - val_mean_absolute_error: 0.4298 - val_acc: 0.7718
Epoch 21/300
1808s - loss: 0.5570 - mean_absolute_error: 0.4119 - acc: 0.7902 - val_loss: 0.5753 - val_mean_absolute_error: 0.4277 - val_acc: 0.7724
Epoch 22/300
1809s - loss: 0.5543 - mean_absolute_error: 0.4100 - acc: 0.7931 - val_loss: 0.5725 - val_mean_absolute_error: 0.4257 - val_acc: 0.7721
Epoch 23/300
1810s - loss: 0.5516 - mean_absolute_error: 0.4083 - acc: 0.7931 - val_loss: 0.5697 - val_mean_absolute_error: 0.4237 - val_acc: 0.7715
Epoch 24/300
1807s - loss: 0.5490 - mean_absolute_error: 0.4064 - acc: 0.7942 - val_loss: 0.5670 - val_mean_absolute_error: 0.4216 - val_acc: 0.7718
Epoch 25/300
1807s - loss: 0.5464 - mean_absolute_error: 0.4046 - acc: 0.7944 - val_loss: 0.5643 - val_mean_absolute_error: 0.4196 - val_acc: 0.7711
Epoch 26/300
1809s - loss: 0.5439 - mean_absolute_error: 0.4029 - acc: 0.7950 - val_loss: 0.5616 - val_mean_absolute_error: 0.4176 - val_acc: 0.7705
Epoch 27/300
1810s - loss: 0.5415 - mean_absolute_error: 0.4012 - acc: 0.7960 - val_loss: 0.5590 - val_mean_absolute_error: 0.4156 - val_acc: 0.7705
Epoch 28/300
1809s - loss: 0.5392 - mean_absolute_error: 0.3996 - acc: 0.7974 - val_loss: 0.5565 - val_mean_absolute_error: 0.4137 - val_acc: 0.7702
Epoch 29/300
1806s - loss: 0.5369 - mean_absolute_error: 0.3979 - acc: 0.7975 - val_loss: 0.5541 - val_mean_absolute_error: 0.4118 - val_acc: 0.7686
Epoch 30/300
1812s - loss: 0.5347 - mean_absolute_error: 0.3963 - acc: 0.7976 - val_loss: 0.5517 - val_mean_absolute_error: 0.4099 - val_acc: 0.7755
Epoch 31/300
1808s - loss: 0.5326 - mean_absolute_error: 0.3947 - acc: 0.7982 - val_loss: 0.5493 - val_mean_absolute_error: 0.4080 - val_acc: 0.7762
Epoch 32/300
1810s - loss: 0.5305 - mean_absolute_error: 0.3934 - acc: 0.7992 - val_loss: 0.5471 - val_mean_absolute_error: 0.4062 - val_acc: 0.7759
Epoch 33/300
1805s - loss: 0.5285 - mean_absolute_error: 0.3916 - acc: 0.7991 - val_loss: 0.5448 - val_mean_absolute_error: 0.4044 - val_acc: 0.7762
Epoch 34/300
1808s - loss: 0.5265 - mean_absolute_error: 0.3902 - acc: 0.7992 - val_loss: 0.5426 - val_mean_absolute_error: 0.4026 - val_acc: 0.7762
Epoch 35/300
```

In terms of accuracy on verification set: for the same data set, the accuracy of single-layer CNN is gradually improved, and it is obvious. VGG19 does not.
In terms of training time, the same LSTM, represented by VGG19 as a feature, takes 30 minutes (1800 seconds) to train an Epoch, while single-layer CNN only takes 180 seconds.
Obviously, single-layer CNN is better.

(3) We should consider the activation function, parameters in functions, the number of layers, the number of neurons in each layer, the use of other algorithm such as Softmax, the method of training the network and the memory the network needs.

2. A:



$W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$, $w = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$.

The whole structure is:

$$\mathrm{w}^T \max\{\mathbf{0}, \mathrm{W}^{\mathrm{T}}\mathbf{x} + \mathbf{b}\} = \mathrm{y}$$

Result:

Input	Output
$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	0
$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	1
$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	1
$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	0

3. A:

a. $y_1 = 4$, $y_2 = -2$.

b. $y_1 = 0$, $y_2 = 2$.