Group Members:
- Neda Khanaki
- Thomas Ngo
- Miguel Gutierrez
- Tevin Vu

Final Group Review

Q1:
A: The training set content: [time stamp of the date, sky status(weather condition:sunny/cloudy/rainy…), high and low temperature, wind direction,wind speed]
B: 5 inputs and 5 outputs
C: 5 hidden layers (RNN or LSTM or GRU): one hidden layer for each input
D: input layers: one-hot encoded, output layers: softmax/Sigmoid
E: We can add them as 3 new inputs to the dataset or use bi-directional recurrent network

Q2:
a. Overfitting is happening. The model cannot generalize or predict the unseen data well.
b. How to prove it?
- There is a gap between training loss and test loss
- Besides, if we plug in the same test instance, the predictions are different every time we run the model. Then, it indicates that the model only memorizes some specific data points. In other words, it doesn't learn the pattern of the true model. As a result, the model is experiencing the overfitting.
- The difference between the accuracy percentage of the training set and test sample is the sign of overfitting.
c. We could add regularization or additionally we could add to/ increase the training data size. Increasing the training data size would hurt accuracy for the training data set but increase the accuracy on the test data. They could use penalty-based regularization, ensemble methods, do pretraining to find a good initialization, early stopping, continuation and curriculum methods, sharing parameters with domain-specific insights.

Q3:
A: if $k < d$, it's called undercomplete in autoencoder. The model finds the most important features and discards the irrelevant/unimportant features. Basically, the model is compressing the original input to learn the most important features.
B: The output will be something similar to the input. For example, if we feed in a corrupted image to this autoencoder, the output will be some images similar to the input image, but not entirely the same as the input. The output image could be blurry.
C: The representation of the hidden layers will be sparse. (sparse representation). There are a lot of zero features.
D:
- Add penalties/constraints in hidden layers when performing the training to create a reconstructed/useful representation on the output.
- We can add Sparsity constraints through Regularization (L1 / KL)
- Add bias to hidden layers

- Allow top-k activations in hidden layers to be non-zero

Q4:

a:s_1 = sign[1*(-1) + (-1)*1] = sign[-2] = -1 since -2 < 0 → change s_1 to -1

B: s^T=(-1,1,-1)

s_2 = sign[-1*(-1) + (-1)*2]= sign[-1] = -1 since -1 < 0 → change s2 to -1

C:s^T =(-1,-1,-1)

S_3 = sign[-1.1 + (-1).2] = sign[-3] = -1 since -3 < 0 → s3 is same as -1

d:s^T =(-1,-1,-1)

s_1 = sign[-1*(-1) + 1*(-1)] = sign[0] = -1 → s_1 state stay the same
s_2 = sign[-1*(-1) + 2*(-1)] = sign[-1] = -1 → s_2 state stay the same
S_3 = sign[-1*1 + 2*(-1)] = sign[-3] = -1 → s3 state stay the same
Seems like the network is stabilized so
s^T =(-1,-1,-1)

Q5

a. Upper-left number: 3*(-1) + 0*0 + 6*0 + 8*1 + 6*(-1) + 1*0 + 9*1 + 8*1 + 9*(-1)= 7

b:[(Length L - Filter F)/1 + 1]x[(B - Filter F)/1 + 1]= [(6-3)/1 + 1]x[(6-3)/1 + 1] = 4 x 4 → feature map: 4x4

c:Add the padding to keep the size of the output feature map to a constant values:

(F-1)/2 = (3-1)/2 = 1 for all the sizes.

d:For the upper-right corner: 9

E. Output activation for the upper right corner have been if the map were padded: 9

0	0	0	0	0	0	0	0
0	3	0	6	7	2	9	0
0	8	6	1	1	2	6	0
0	9	8	9	5	1	8	0
0	7	3	2	0	6	5	0
0	2	9	7	8	5	9	0
0	3	5	2	7	0	6	0
0	0	0	0	0	0	0	0