①

a) $V_2 = \begin{bmatrix} -0.165 & 0.245 & -0.059 \\ -0.196 & -0.032 & 0.209 \end{bmatrix}$

$S_2 = \begin{bmatrix} 0.0106 & 0.0699 & .1099 \end{bmatrix}$

$S_2 = \begin{bmatrix} 0.0106 & 0.070 & 0.110 \\ 0.141 & 0.170 & 0.021 \end{bmatrix}$

$W_2 = \begin{bmatrix} -0.182 & 0.308 & 0.674 \\ 0.502 & -0.858 & -0.934 \end{bmatrix}$

False

(3) a) $h_t = o_t \odot \tanh(c_t)$

$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$

$x_t = 0$

$= \sigma[U_o h_{t-1} + b_o]$

$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$ — (3)

$f_t = \sigma x[U_f h_{t-1} + b_f]$ $x_t = 0$ — (1)

$i_t = \sigma[U_i h_{t-1} + b_i]$, $x_t = 0$ — (2)

$\tilde{c}_t = \tanh(U_c h_{t-1} + b_c)$ — (4)

Putting (1), (2), (4) in (3)

$c_t = \sigma[U_f h_{t-1} + b_f] \odot c_{t-1} + \sigma[U_i h_{t-1} + b_i] \odot \tanh[U_c h_{t-1} + b_c]$

$h_t = \sigma[U_o h_{t-1} + b_o] \odot \{\sigma[U_f h_{t-1} + b_f] \odot c_{t-1} + \sigma[U_i h_{t-1} + b_i] \odot \tanh[U_c h_{t-1} + b_c]\}$

so $h_t \neq h_{t-1}$,

Qualitatively due to input zero also, network may decide to forget something from long term memory as even input to the long term Memory

True

3) b) By equation, the propagation of gradient-

$$\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial o_t} \times \frac{\partial o_t}{\partial h_{t-1}} + \frac{\partial h_t}{\partial c_t} \times \frac{\partial c_t}{\partial F_t} \times \frac{\partial F_t}{\partial h_{t-1}}$$

$$+ \cdots$$

$$\frac{\partial c_t}{\partial c_{t-1}} = F_t$$

In the long term part of the LSTM, cell $\partial c_t / \partial c_{t-1}$ becomes smaller when F_t is small.

By the concept of forget gate, if the previous input is to forgotten $F(t)$ should be small, so error does not affect previous states.

c) **True**

i_t, F_t = output of sigmoid [$\because$ sigmoid ≤ 1,] so non-negative.

for o_t, output gate also sigmoid output.

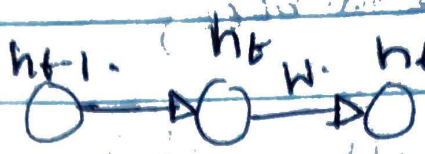
Sigmoid bounded between 0,1 so non-negative output

d) **False**

F_t, i_t, o_t are independent of each other.

F_t, i_t, o_t can be seen as individual probabaty of forgetting data, adding data and output data, but they do not sum to 1.

(2) a)



$$\frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+1}} \times \frac{\partial h_{t+1}}{\partial h_t}$$

~~$h_{t+1} = act(\quad z_t) \cdot W$~~

$$h_{t+1} = act(\underbrace{h_t W}_{z_t})$$

z_t as per question.

$$\frac{\partial h_{t+1}}{\partial h_t} = \sigma'(z_t) W^T$$

$$\boxed{\nabla h_t = \nabla h_{t+1} \times \sigma'(z_t) \times W^T}$$

Dimension che

$h_t = 64.$

$W = 64 \times 64$

(64×3) ×)

(×64

(64×1) ×

64×32

(32×)32×64

b) $\sigma'(z) = \gamma_4$ $\quad \nabla h_t = \nabla h_{t+1} \times \gamma_4 \times w.$

if $\boxed{w > 4}$ $\nabla h_t > 1$, so gradients would explode.

if $\boxed{w < 4}$, $\nabla h_t < 1$, so the gradients will vanish.

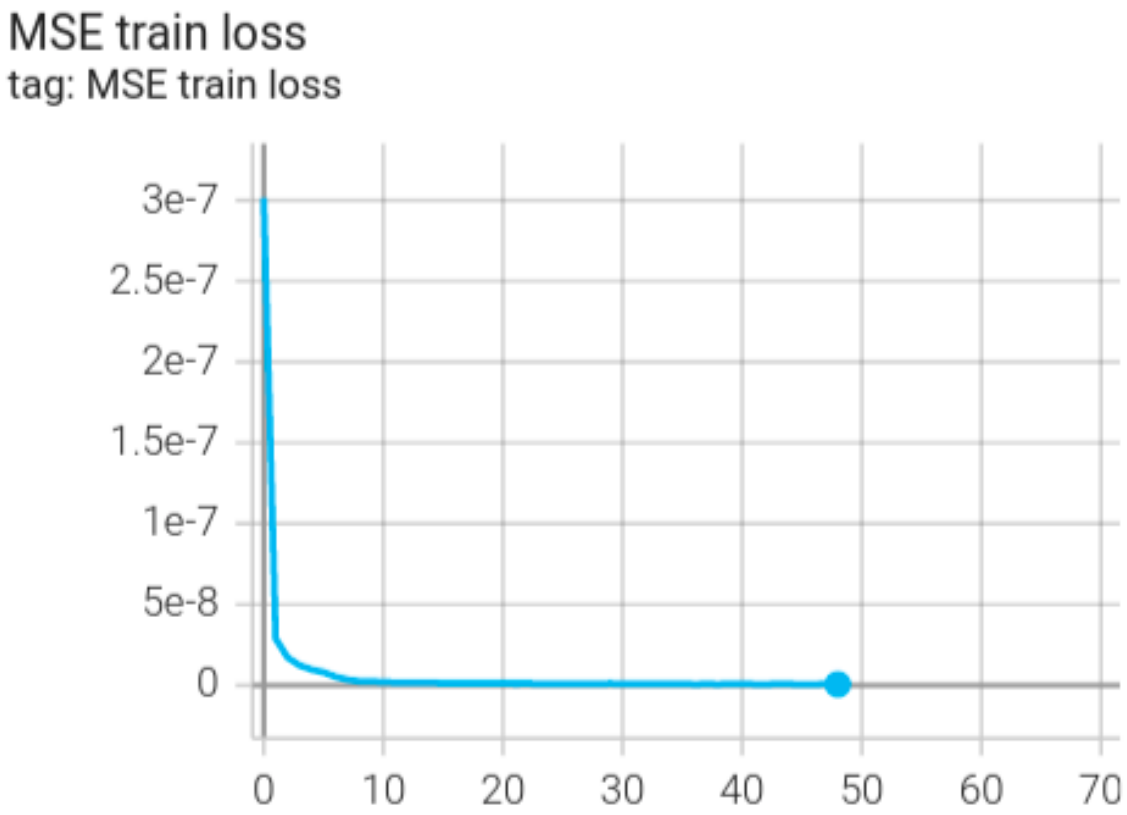
$$\boxed{d = 4}$$

Programming Exercises

Problem 1

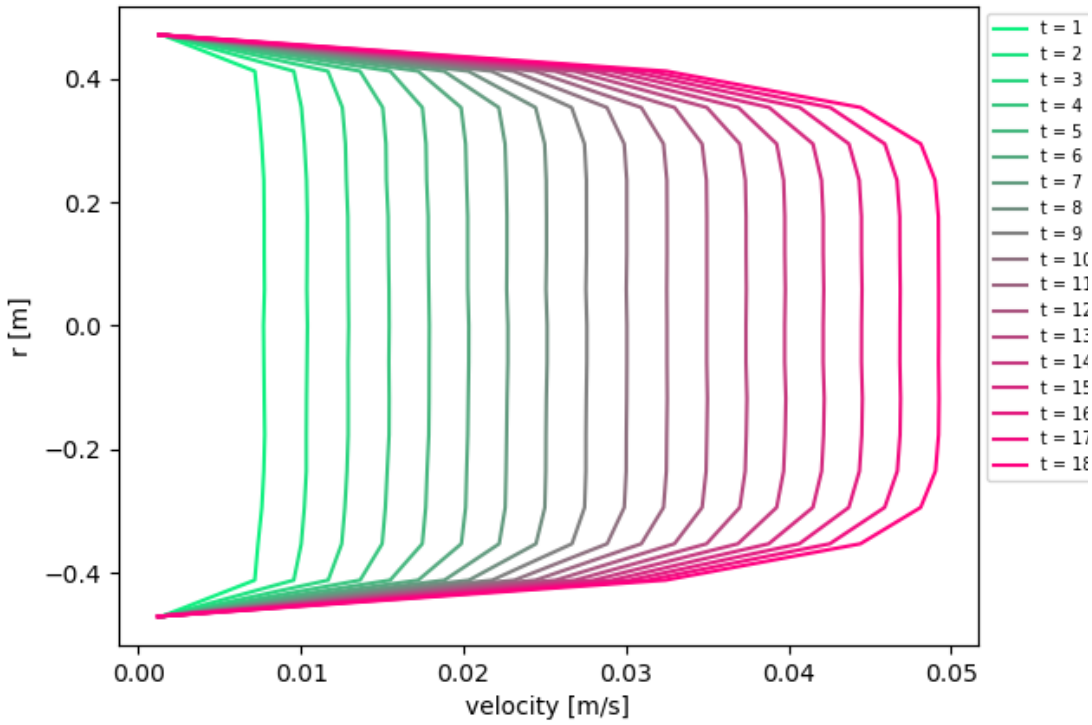
Network structure:

- Three lstm layers, of the following features
  - 1st lstm, Input size 17, hidden size 128
  - 2nd lstm, Input size 128, hidden size 256
  - 3rd lstm, Input size 256, hidden size 512
- One MLP of the following feature
  - Input size 512, output size 17
- Number of trainable parameters: 83985
- Number of iterations: 50
- Learning rate = 0.001
- Optimizer: Adam
- Train loss criterion: MSELoss
- Train loss vs batch number
- Batch size: 256
- Scheduler: Step LR with 0.1 gamma, and 25 epochs step size
  - Train MSELoss vs epoch

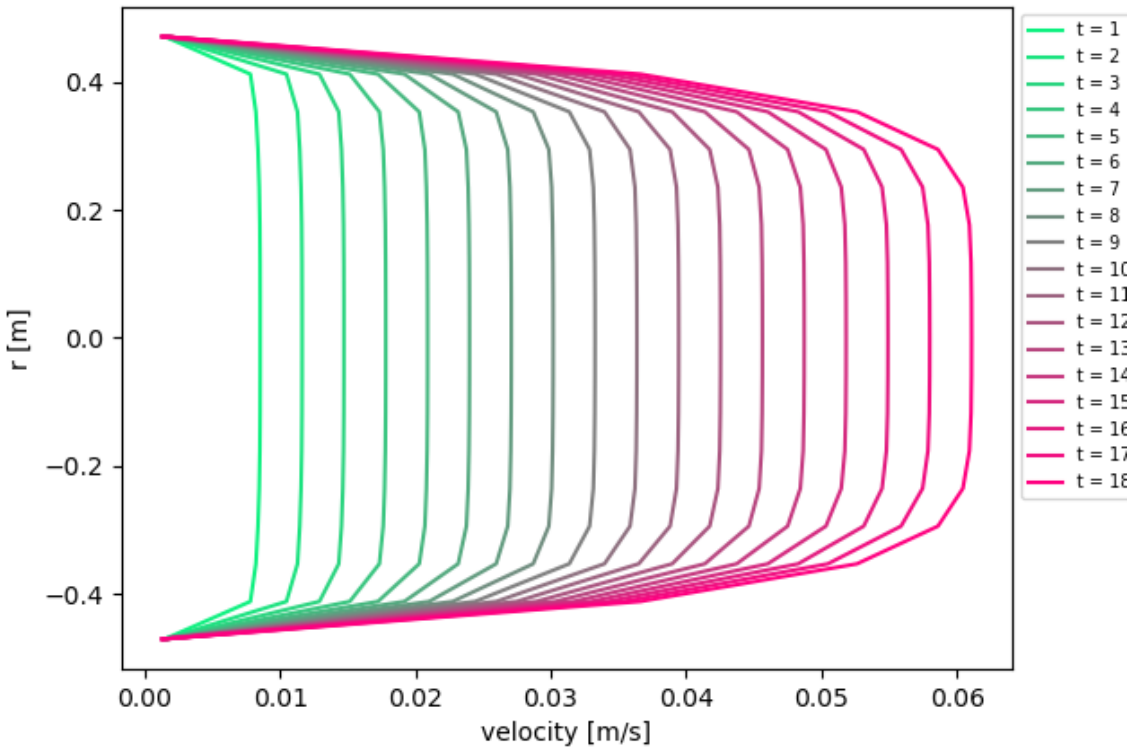


  - Final test MSELoss:**0.00028252945048734546**
  - Final test L1Loss: **0.06915978819597512**

Results diagram



Prediction



Ground truth