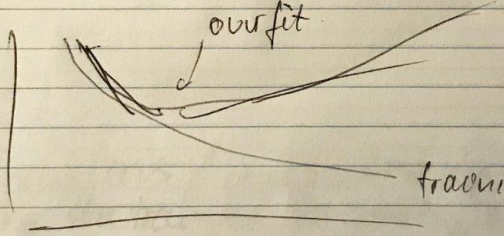
THOMAS NOO
final
5/16 (2021)

① The content of training set should contains the following features:
a) Time of the date, conditions of the cloud, temperature, windspeed, wind direction (southeast/west)
b) # of input = 5.
# of output = 5.
c) # of hidden layers = 3. (Start with 3, then tune hyperparameters) dense layers
d) ↳ Activation function = ReLu. (neurons)
e) - use Softmax to get the probabilistic/statistic
- use pretraining to freeze model, and then use it output
(another model)

② a) overfitting is happening. The model cannot generalize well

b)



Because there is a gap b/w training loss and test loss

c) -Add regularization to prevent overfitting:
- Add dropout.
- Use ensemble method (dropout | bagging, boosting)
- use early stopping
- Pretraining
- use shared weights

(4) e) Add padding,

$(F_6 - 1)/2 = (8-1)/2 = 7/2 = 1$

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

THOMAS NGO

9/17/21

(5) b) $I_g = 6, \quad F_g = 3$

$I_g - F_g + 1 = 6 - 3 + 1 = 4$. (feature map/output size)

The feature map's size = 4×4.

a) $3(-1) + (0)(0) + (6)(0) + 8(1) + 6(-1) + (1)(0) +$

$9(1) + 8(1) + 9(-1) = \boxed{7}$

The input volume in the upper-left corner.

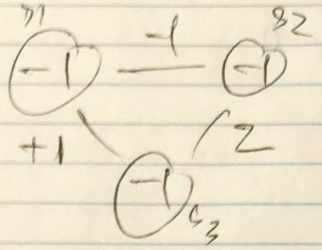
c) Add padding to keep the size of the output feature map to be constant value. (all layers)

d) $\begin{vmatrix} 3 & 0 & 6 \\ 8 & 6 & 1 \\ 9 & 8 & 9 \end{vmatrix}$ ↪ max(image) = 9. (upper left)

$\begin{vmatrix} 0 & 6 & 5 \\ 8 & 6 & 9 \\ 7 & 0 & 6 \end{vmatrix}$ ↪ ~~max (image)~~

$\begin{vmatrix} 7 & 2 & 9 \\ 1 & 2 & 6 \\ 5 & 1 & 8 \end{vmatrix}$ max (upper right) = $\boxed{9}$ (upper right)

(4) a) $\vec{s} = (-1, -1, -1)$



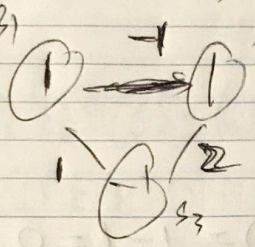
$s_1 = (-1)(-1) + (+1)(-1) = 1 - 1 = 0 \leq 0.$

$\Rightarrow$ Do Nothing $\Rightarrow$ network is stabilized.
Since it's the same value as the last run.

THOMAS NGO

5/17/21

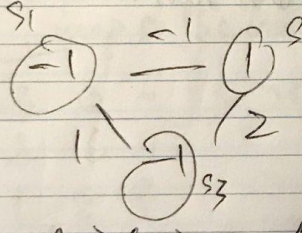
(4) (a) $\vec{w} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & -1 & +1 \\ -1 & 0 & 2 \\ +1 & 2 & 0 \end{pmatrix} \end{array}$



$s_1 = (-1)(1) + 1(-1) = -2 < 0 \Rightarrow$

flip s_1 to -1

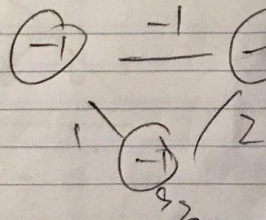
b)



$s_2 = (-1)(-1) + (2)(-1) = 1 + (-2) = -1 < 0.$

$\Rightarrow$ flip s_2 to -1.

c)



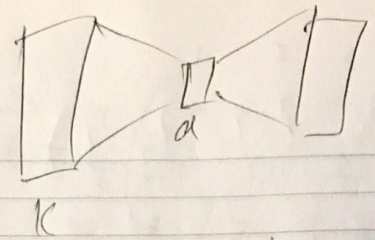
$s_3 = (-1)(1) + (2)(-1) = -1 - 2 = -3 < 0$

$\Rightarrow$ keep $s_3 = -1$.

THOMAS NGO

9/17/2021

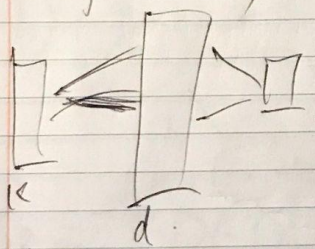
3) a)



if $K < d$, undercomplete → model finds the most important features and discard irrelevant features. (compress)

b) the output will be something similar to the input. for example, if we feed in corrupted image to this autoencoder, the output will be some image similar to the input, but the image will be blurry.

c) if $K > d$, the representation of hidden layer will be sparse. (i.e, a lot zero features) (sparse representation)



d) - Add penalties / constraints in hidden layers when performing the training to create a reconstructed useful representation on the output

- Add bias to hidden layers

- only allow top-k activations in hidden layers to be non-zero.

(2) b) How to prove it? (overfitting)

plug in different the same test instance, if the models returns different predictions & because the model only memorizes some specific datapoints, it doesn't learn the pattern of the true model.