DI504 Foundations of Deep Learning

02/02/2022

online (19:00-24:00)

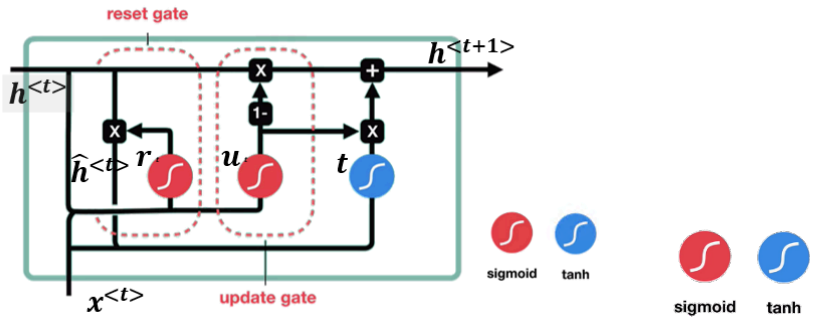
- This is an online midterm exam with four questions. You are to upload the exam to ODTUClass tonight until 23:59. Please do not email your answers.
- Passing the work of others off as your own is a breach of academic ethics and also of the University's disciplinary rules.
- A simple photo of your sheet or word editing program outputs are both acceptable for your hand-written answers.
- Good luck. Please contact me by email, if you have any questions. I will answer them as fast as I can.

DI504 Foundations of Deep Learning
The Final

Fall 21/22

Duration: 5 hours 02/02/2022

Question 2 (30 Points): Please find below a "Gated Recurrent Unit" (GRU).



At a given time t the weights of the given GRU is as given below:
Wr = [1 1 0 0 0; 1 0 0 1 1; 1 1 0 0 0], $b_r = -1.0$
Wu = [1 0 1 0 1; 1 0 1 0 1; 1 0 1 0 1], $b_u = +1.5$
Wt = [1 1 1 1 1; 0 0 0 0 0; 1 1 1 1 1], $b_t = +0.4$

And the input and the hidden state variables as such:
$x^{<t>} = [+1.82 \quad -0.21]^{\mathrm{T}}$, $h^{<t>} = [+0.36 \quad -1.45 \quad +0.23]^{\mathrm{T}}$

a) **(10 Points)** Find the value of the intermediate of state vector $\hat{h}^{<t>}$ at the output of the "**reset gate**".
b) **(10 Points)** Find the value of the next state $h^{<t+1>}$.
c) **(10 Points)** What exactly happens at time <t> at this gate? Please explain your reasoning.

You may need your calculators. Please remember that the operations inside the GRU are element-wise multiplications/additions.

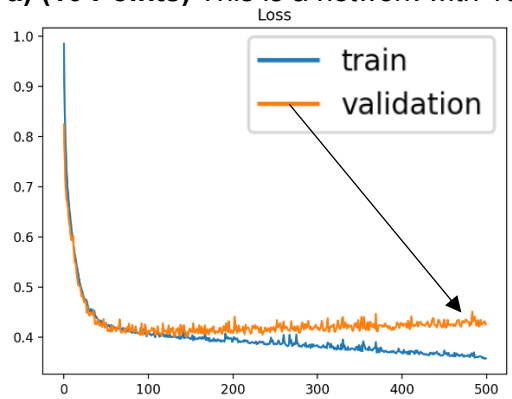
For the order of the x and h vectors when concatenated, use the formula we saw in class:

$$\sigma\left(\mathbf{W}^{inp1}_{\mathrm{l}\times(\mathrm{k}+\mathrm{n})} \cdot \begin{bmatrix} \mathbf{x}_{\mathrm{n}\times 1}(t_0) \\ \mathbf{a}_{\mathrm{k}\times 1}(t_0) \end{bmatrix} + \mathbf{b}^{inp1}_{\mathrm{l}\times 1}\right)$$

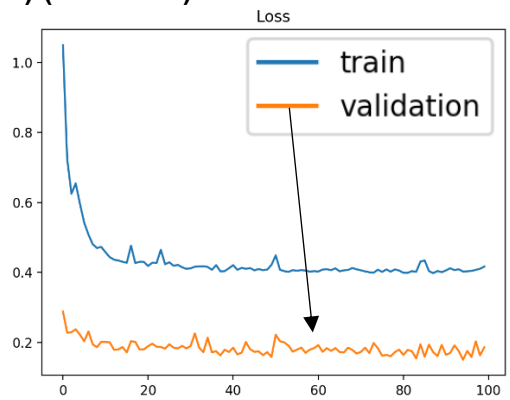
Question 1 (30 Points): Below you will find 3 different learning curves, from 3 different experiments. Imagine, while you are babysitting your training, you have obtained these curves. Speculate on the curves, name any obvious phenomenon you observe and clearly indicate what your next action(s) should be. For each case, some details may be provided, please read them carefully.
Since not all detail on the architecture of the network or the dataset are provided for the cases below, while speculating on how to proceed, please support your cases with proper conditions, such as:

- If my dataset is small, I would **change**…
- If I am transfer learning from a deep network, I would **first analyze** …

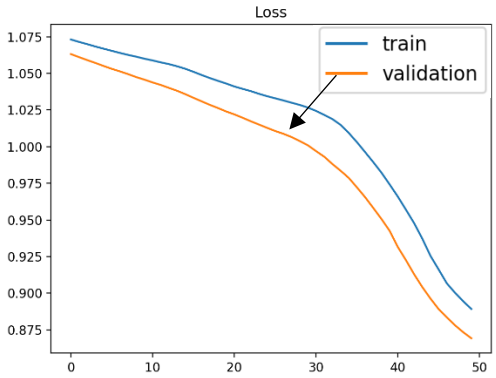
a) (10 Points) This is a network with 10 convolutional and 2 fully connected layers.



b) (10 Points) No information about the network or the dataset.



c) (10 Points) No information about the network or the dataset.

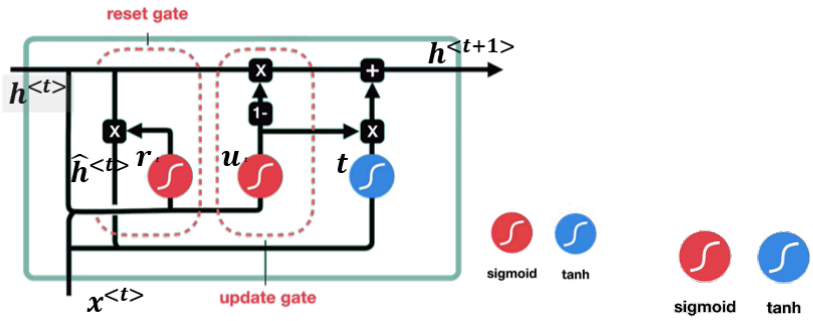


DI504 Foundations of Deep Learning
The Final

Fall 21/22

Duration: 5 hours 02/02/2022

Question 2 (30 Points): Please find below a "Gated Recurrent Unit" (GRU).



At a given time t the weights of the given GRU is as given below:
Wr = [1 1 0 0 0; 1 0 0 1 1; 1 1 0 0 0], $b_r = -1.0$
Wu = [1 0 1 0 1; 1 0 1 0 1; 1 0 1 0 1], $b_u = +1.5$
Wt = [1 1 1 1 1; 0 0 0 0 0; 1 1 1 1 1], $b_t = +0.4$

And the input and the hidden state variables as such:
$x^{<t>} = [+1.82 \quad -0.21]^{\mathrm{T}}$, $h^{<t>} = [+0.36 \quad -1.45 \quad +0.23]^{\mathrm{T}}$

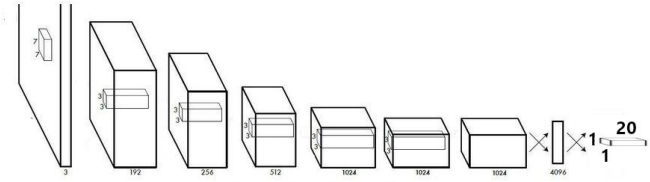
a) **(10 Points)** Find the value of the intermediate of state vector $\hat{h}^{<t>}$ at the output of the "**reset gate**".
b) **(10 Points)** Find the value of the next state $h^{<t+1>}$.
c) **(10 Points)** What exactly happens at time <t> at this gate? Please explain your reasoning.

You may need your calculators. Please remember that the operations inside the GRU are element-wise multiplications/additions.

For the order of the x and h vectors when concatenated, use the formula we saw in class:

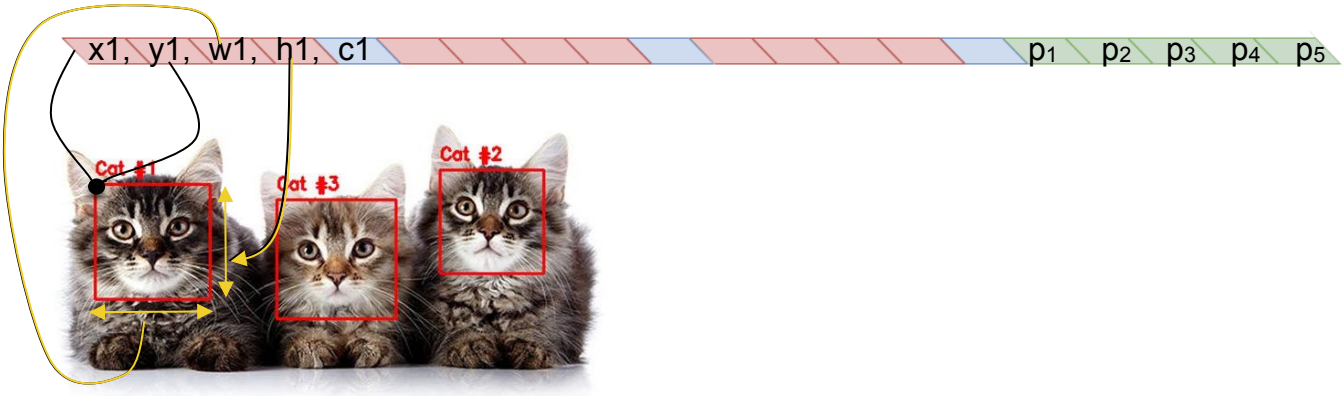
$$\sigma\left(\mathbf{W}^{inp1}_{\mathrm{l}\times(\mathrm{k}+\mathrm{n})} \cdot \begin{bmatrix} \mathbf{x}_{\mathrm{n}\times 1}(t_0) \\ \mathbf{a}_{\mathrm{k}\times 1}(t_0) \end{bmatrix} + \mathbf{b}^{inp1}_{\mathrm{l}\times 1}\right)$$

Question 3 (20 Points): Please find a modified version of YOLO v1 with reduced number of object output and class categories. This version, let's call it "miniYOLO", can only output at most 3 boxes out of 5 categories.



miniYOLO has a single output cell 1x1x20, in which first 15 values are the **<u>normalized</u>** bounding box locations (red cells: x_i, y_i, width, height) and the confidence values (blue cells) for the 3 output, where the last 5 (green cells) are the class probabilities. Because of its architecture, miniYOLO, can output up to 3 boxes, which must be of the same category, decided by the maximum class probability (among the last 5 value of 1x1x20).

The classes are (in order): *dog*, *cat*, *elephant*, *bird*, *bronteroc*.



Given an output vector:

$$y = \begin{bmatrix} 0.20 & 0.23 & 0.47 & 0.51 & 0.34 & 0.11 & 0.06 & 0.33 & 0.79 & 0.89 & \cdots \\ \cdots & 0.72 & 0.66 & 0.28 & 0.34 & 0.61 & 0.13 & 0.04 & 0.21 & 0.23 & 0.39 \end{bmatrix}$$

Please find the detected objects, assuming that the confidence threshold is **<u>0.6</u>** (i.e. only detections higher than this confidence value are valid). Please draw your results (roughly) on the image, with position values and class names.

Question 4 (20 Points): Consider the categorical input with index values given below:

0. Apple –> one-hot vector would be: $[1\ 0\ 0\ 0\ 0]^T$
1. Mango
2. Pear
3. Tomato
4. Potato

You are given an Embedding layer with an embedding dimension of "2". The embedding layer treats the input values as one-hot vectors with the provided index order (similarly to the nn.Embedding class).

Given the Embedding weight matrix W of size 2 x K (K=5 being the vocabulary size)

$$\mathbf{W} = \begin{bmatrix} -0.81 & -0.13 & -0.63 & +0.38 & +0.92 \\ -0.91 & -0.43 & -0.83 & +0.54 & +0.67 \end{bmatrix}$$

a) (10 Points) Draw the 2D embedding space showing each input category location?

b) (10 Points) If this embedding layer output is to be fed to a classification network, what kind of a problem can this embedding be possibly trained/designed for? Please speculate.