50.039 – Theory and Practice of Deep learning

Alex

Week 08

1 Task 1:

The following are the defining equations for a LSTM cell with a forget gate f_t,

$$
\begin{aligned}
i_t &= \sigma(W^i x_t + U^i h_{t-1}) \\
f_t &= \sigma(W^f x_t + U^f h_{t-1}) \\
o_t &= \sigma(W^o x_t + U^o h_{t-1}) \\
u_t &= \tanh(W^c x_t + U^c h_{t-1}) \\
c_t &= f_t \circ c_{t-1} + i_t \circ u_t \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
$$

The symbol $\circ$ denotes element-wise multiplication and $\sigma(x) = \frac{1}{1+e^{-x}}$

- is c_{t-1} as function of h_{t-1}? Give 3 sentences at most as answer why or why not.
- Write down the derivative $\frac{\partial h_t}{\partial h_{t-1}}$ (which is the equivalent of the term $\frac{\partial s_t}{\partial s_{t-1}}$ in the lecture) expressed as a function of $\frac{\partial i_t}{\partial h_{t-1}}$, $\frac{\partial f_t}{\partial h_{t-1}}$, $\frac{\partial o_t}{\partial h_{t-1}}$, $\frac{\partial u_t}{\partial h_{t-1}}$. You do not need to resolve the terms $\frac{\partial i_t}{\partial h_{t-1}}$, $\frac{\partial f_t}{\partial h_{t-1}}$, $\frac{\partial o_t}{\partial h_{t-1}}$, $\frac{\partial u_t}{\partial h_{t-1}}$.

  Note that for element-wise multiplications the product rule holds, even if you multiply two vectors.

  You do not need to answer this question: Do you understand why I do **not** want you to explicitly compute by hand $\frac{\partial h_t}{\partial h_{t-3}}$?
- Calculate $\sigma'(z)$, the derivative of the sigmoid (not all tasks can be hard, right?)
- Write down the derivative $\frac{\partial f_t}{\partial h_{t-1}}$ expressed as a function of σ' and other terms as required.

- how to activate a gate ? Why is this important?

  If an LSTM has a forget gate, then it is good practice to initialize it such that f_t is a vector of values close to 1 at the start of the training. This is done usually by adjusting the bias terms (, which we dropped here, though).

  Why one wants $f_t \approx 1$ at init? If $f_t \approx 1$, then at the beginning we have

  $$c_t = f_t \circ c_{t-1} + i_t \circ u_t \approx 1 \circ c_{t-1} + i_t \circ u_t = c_{t-1} + i_t \circ u_t,$$

  and thus $\frac{\partial c_t}{\partial c_{t-1}} \approx 1$, that is the gradient flows back through time unchallenged through the memory cell vectors c_t .

  Back to the question: how to activate a gate ... not using a bias ?

  Consider

  $$f_t = \sigma(W^f x_t + U^f h_{t-1})$$

  The output is a vector because

  $x_t \in \mathbb{R}^d$ is a vector but $W_f \in \mathbb{R}^{k \times d}$ is a matrix, $h_{t-1} \in \mathbb{R}^k$ is a vector but $W_f \in \mathbb{R}^{k \times k}$ is a matrix,

  so the output will be a vector $\in \mathbb{R}^k$. Lets look at one component of the vector:

  $$f_t(d) = \sigma(W^f(d) \cdot x_t + U^f(d) \cdot h_{t-1})$$

  where $W^f(d)$ is a $\mathbb{R}^{1 \times d}$ row or column vector and $U^f(d)$ is a $\mathbb{R}^{1 \times k}$ row or column vector. Thus $W^f(d) \cdot x_t$ is just an inner product of two vectors.

  Question: if $x_t = 0$, $U^f(d) \neq 0$, which vector h_{t-1} among all the vectors of euclidean length 1 maximized the values of $f_t(d)$?

  Speaking in math, find

  $$\begin{aligned} &\text{argmax}_{\{h_{t-1}:\|h_{t-1}\|_2=1\}} f_t(d) \\ =\ &\text{argmax}_{\{h_{t-1}:\|h_{t-1}\|_2=1\}} \sigma(W^f(d) \cdot x_t + U^f(d) \cdot h_{t-1}) \end{aligned}$$

  Think geometrically (its an inner product, and the euclidean norm is $v \cdot v = \|v\|_2^2$!!!) to find the solution, then it is easy.

- in the task above, does the argmax depend on the value of $W^f(d) \cdot x_t$? Does the max depend on it? Give at most 3 sentences justification

2 Task 2:

- You are given a 2-dimensional convolution with spatial size $(78, 84)$. When using a kernel of size $(5, 5)$ and stride 3 with padding of 2 , what will be the spatial size of the feature map which is the output of the convolution? Note that spatial size does not depend on the number of input or output channels.

- You are given a 2-dimensional convolution with spatial size $(64, 64)$. When using a kernel of size $(3, 5)$ and stride 2 with padding of 0 , what will be the spatial size of the feature map which is the output of the convolution? Note same that spatial size does not depend on the number of input or output channels.
- You are given a 1-dimensional convolution, when using a kernel of size 9 and stride 3 with padding 1. What spatial input size do you need to have, so that you have a spatial output size of 16?

How many trainable parameters are

- in a 2-D convolutional layer with input $(32, 19, 19)$, kernel size $(7, 7)$, stride 3, 64 kernel channels, no padding, no bias term?
- how many multiplications and how many additions are performed in this case above?
- in a 2-D convolutional layer with input $(512, 25, 25)$, kernel size $(1, 1)$, stride 1, 128 kernel channels, padding 2, no bias term?