MT Marathon 2016 – Written Exam With Solutions

Please answer the questions directly on this paper. Note that the answers will be checked by some other participant.

The final exam score is a complex number – some questions from the exam were not discussed on the slides, but are important NN topics. These questions are written in ***italics*** and are awarded with imaginary points.

You can get a hint for any question (for example if you are unsure how to compute softmax or how to derivate it) – just raise your hand and I will come to you. You will be awarded only 50% of points for such questions.

1) Compute BLEU (up to bigrams) and TER of the following translations.

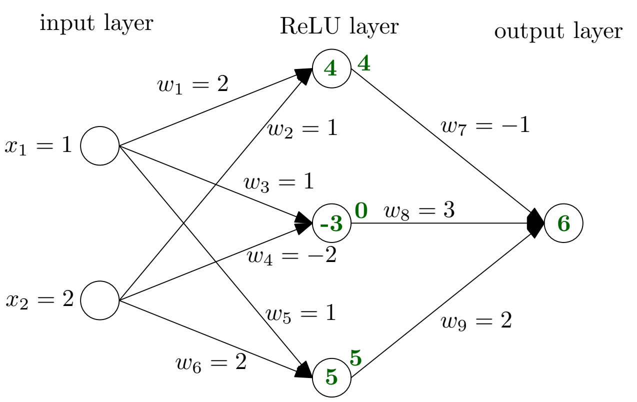
Human Translation: *the cat jumped on the table*

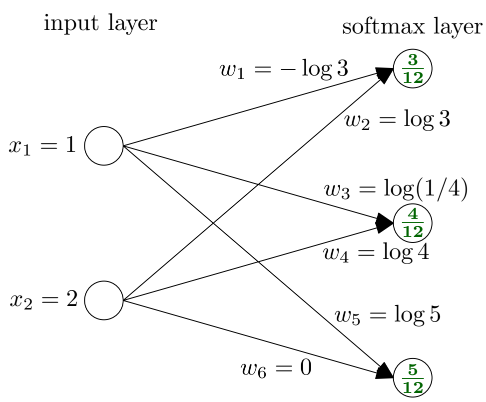
Machine Translation: *the table the the cat*

$$\begin{aligned}\text{BLEU} &= \textit{brevity penalty} \cdot \exp(\textit{mean of clipped log probability of unigrams and bigrams}) \\ &= e^{1-\frac{6}{5}} \cdot e^{\frac{1}{2}\log(\frac{4}{5})+\frac{1}{2}\log(\frac{2}{5})} = \sqrt{\frac{2}{5}}e^{-\frac{1}{5}}\end{aligned}$$

$$\begin{aligned}\text{TER} &= \textit{word edit distance normalized by length of HT} \\ &= \frac{4}{5}\end{aligned}$$

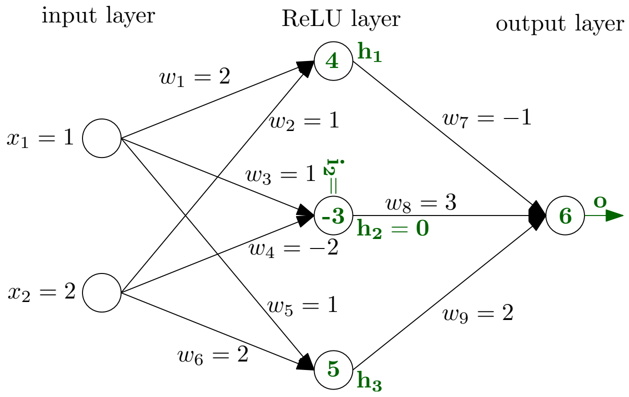
2) Compute the output of the following neural networks.





3) Compute derivatives of the networks with respect to all weights and inputs.

For the first network, the gold output is 3, and the loss function is MSE, i.e., $(output - gold)^2$.



$$\frac{\partial L}{\partial o} = 2(output - gold) = 6$$

$$\frac{\partial L}{\partial w_7} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial w_7} = \frac{\partial L}{\partial o} h_1 = 24$$

$$\frac{\partial L}{\partial w_8} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial w_8} = \frac{\partial L}{\partial o} h_2 = 0$$

$$\frac{\partial L}{\partial w_9} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial w_9} = \frac{\partial L}{\partial o} h_3 = 30$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial i_1}\frac{\partial i_1}{\partial w_1} = \frac{\partial L}{\partial i_1} x_1 = -6$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial i_1}\frac{\partial i_1}{\partial w_2} = \frac{\partial L}{\partial i_1} x_2 = -12$$

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial i_2}\frac{\partial i_2}{\partial w_3} = \frac{\partial L}{\partial i_2} x_1 = 0$$

$$\frac{\partial L}{\partial w_4} = \frac{\partial L}{\partial i_2}\frac{\partial i_2}{\partial w_4} = \frac{\partial L}{\partial i_2} x_2 = 0$$

$$\frac{\partial L}{\partial w_5} = \frac{\partial L}{\partial i_3}\frac{\partial i_3}{\partial w_5} = \frac{\partial L}{\partial i_3} x_1 = 12$$

$$\frac{\partial L}{\partial w_6} = \frac{\partial L}{\partial i_3}\frac{\partial i_3}{\partial w_6} = \frac{\partial L}{\partial i_3} x_2 = 24$$

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial h_1} = \frac{\partial L}{\partial o} w_7 = -6$$

$$\frac{\partial L}{\partial h_2} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial h_2} = \frac{\partial L}{\partial o} w_8 = 18$$

$$\frac{\partial L}{\partial h_3} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial h_3} = \frac{\partial L}{\partial o} w_9 = 12$$

$$\frac{\partial L}{\partial i_1} = \frac{\partial L}{\partial h_1}\frac{\partial h_1}{\partial i_1} = \frac{\partial L}{\partial h_1} 1 = -6$$

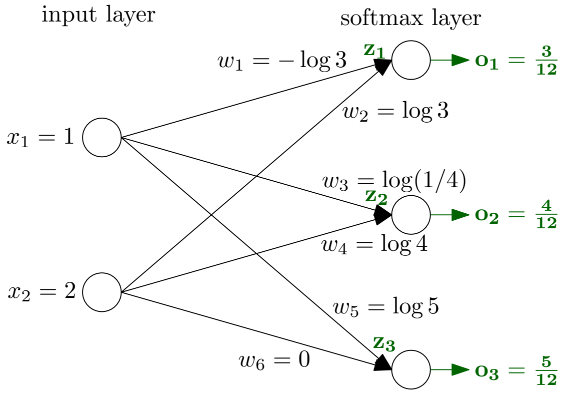
$$\frac{\partial L}{\partial i_2} = \frac{\partial L}{\partial h_2}\frac{\partial h_2}{\partial i_2} = \frac{\partial L}{\partial h_2} 0 = 0$$

$$\frac{\partial L}{\partial i_3} = \frac{\partial L}{\partial h_3}\frac{\partial h_3}{\partial i_3} = \frac{\partial L}{\partial h_3} 1 = 12$$

$$\frac{\partial L}{\partial x_1} = \sum_j \frac{\partial L}{\partial i_j}\frac{\partial i_j}{\partial x_1} = 0$$

$$\frac{\partial L}{\partial x_2} = \sum_j \frac{\partial L}{\partial i_j}\frac{\partial i_j}{\partial x_2} = 18$$

For the second network, the gold output is the distribution $(0, 0, 1)$ and the loss function is cross-entropy (which in the *single correct class* case is equal to log probability of the correct class).



Let $g = (g_1, g_2, g_3)$ be the gold output distribution. Then $L = -\sum_j g_j \log o_j$. We start by showing (the well known fact) that $\frac{\partial L}{\partial z_i} = o_i - g_i$.

$$\frac{\partial L}{\partial z_i} = \frac{\partial - \sum_j g_j \log o_j}{\partial z_i} = \frac{\partial - \sum_j g_j \log(e^{z_j}/\Sigma_k e^{z_k})}{\partial z_i}$$

$$= \frac{\partial - \sum_j (g_j z_j - g_j \log \Sigma_k e^{z_k})}{\partial z_i} = \frac{\partial \sum_j g_j \log \Sigma_k e^{z_k}}{\partial z_i} - \frac{\partial \sum_j g_j z_j}{\partial z_i}$$

$$= \sum_j (g_j \frac{1}{\Sigma_k e_{z_k}} e^{z_j}) - g_i = \sum_j (g_j o_i) - g_i = o_i - g_i$$

The rest is straightforward.

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial w_1} = \frac{\partial L}{\partial z_1} x_1 = \frac{3}{12} \qquad \frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial w_2} = \frac{\partial L}{\partial z_1} x_2 = \frac{6}{12}$$

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial w_3} = \frac{\partial L}{\partial z_2} x_1 = \frac{4}{12} \qquad \frac{\partial L}{\partial w_4} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial w_4} = \frac{\partial L}{\partial z_2} x_2 = \frac{8}{12}$$

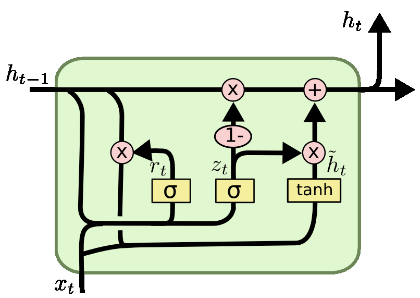
$$\frac{\partial L}{\partial w_5} = \frac{\partial L}{\partial z_3}\frac{\partial z_3}{\partial w_5} = \frac{\partial L}{\partial z_3} x_1 = \frac{5}{12} \qquad \frac{\partial L}{\partial w_6} = \frac{\partial L}{\partial z_3}\frac{\partial z_3}{\partial w_6} = \frac{\partial L}{\partial z_3} x_2 = \frac{10}{12}$$

$$\frac{\partial L}{\partial x_1} = \sum_j \frac{\partial L}{\partial z_j}\frac{\partial z_j}{\partial x_1} = \sum_j \frac{\partial L}{\partial z_j} w_{2j-1} = \frac{-3\log 3 - 4\log 4 + 5\log 5}{12}$$

$$\frac{\partial L}{\partial x_2} = \sum_j \frac{\partial L}{\partial z_j}\frac{\partial z_j}{\partial x_2} = \sum_j \frac{\partial L}{\partial z_j} w_{2j} = \frac{3\log 3 + 4\log 4}{12}$$

4) Write explicit GRU equations.

Compute h_t using h_{t-1} and x, using the GRU sketch from the slides. Include all weight matrices and bias terms.



$$r_t = \sigma(W_r x + U_r h_{t-1} + b_r)$$
$$z_t = \sigma(W_z x + U_z h_{t-1} + b_z)$$
$$\tilde{h}_t = \sigma(W_h x + U_h(r_t \odot h_{t-1}) + b_h)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

5) Explain (with formulas) how can we handle the exploding gradient problem.

We can perform *gradient clipping*, either using the norm of the whole gradient (if $||g||_2 > c$, then $g \leftarrow \frac{c}{||g||_2} g$), or element-wise ($\forall i$, if $g_i > c$ then $g_i \leftarrow c$).

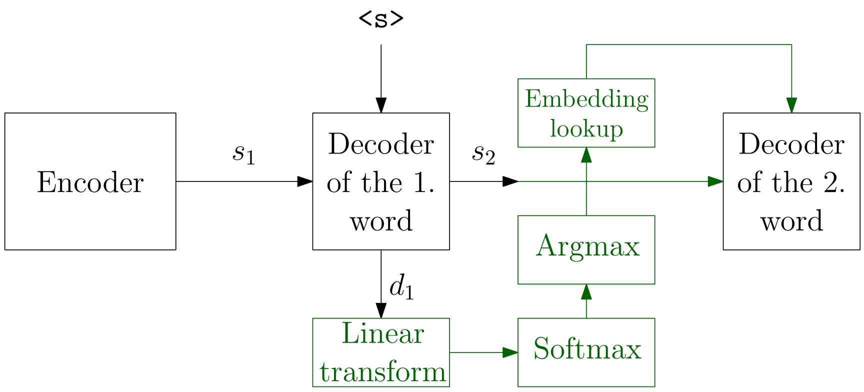
6) What is vanishing gradient problem in plain RNNs? Try explaining why it occurs.

The problem is that in plain RNNs the gradient can vanish (become close to zero) during backpropagation.

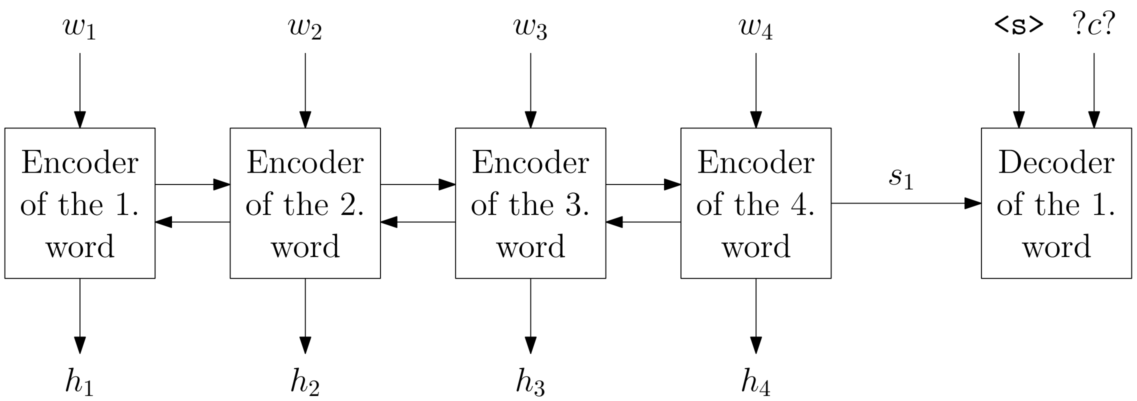
The problem is caused by the fact that plain RNNs tend to perform nearly the same transformation to the state which is passed through the RNN. If we approximate the derivation of the transformation by its Jacobian matrix J, and if we assume it is the same in all steps, the gradient after the last i steps can be approximated as $J_i g$. And if the largest eigenvalue of J is less than one (which may really be the case), the gradient gets smaller and smaller.

7) Assume basic encoder-decoder model without attention. Write/draw the inputs used for decoding the second word during inference time.

Include all operations like tanh, softmax, linear transformations, biases, embeddings, ...



8) Assume encoder-decoder model with attention. Write how exactly is computed the attention c used when decoding the first word.



$$e_{1,j} = v^\top \tanh(W s_1 + U h_j + b)$$

$$\alpha_{1,j} = \frac{e^{e_{1j}}}{\sum_k e^{e_{1k}}}$$

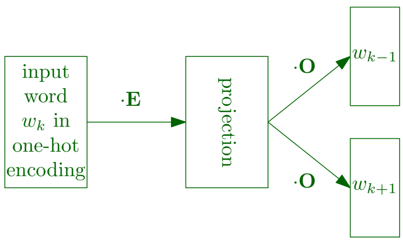
$$c = c_1 = \sum_j \alpha_{1,j} h_j$$

9) Compute 3 steps and then further 4 steps of BPE on the following dictionary.

Word	*Frequency*
t a l l </w>	2
t a l l e r </w>	1
t a l l e s t </w>	1
b e s t </w>	1

The dictionary after 3 steps: tall </w>, tall e r </w>, tall e s t </w>, b e s t </w>
The dictionary after 7 steps either: tall</w>, tall e r </w>, tall est</w>, b est</w>
or: tall</w>, talle r </w>, talle st</w>, b e st</w>

10) Explain the skip-gram model of Mikolov et al.
Consider the basic skip-gram model (which uses full softmax output) with context size 1. Draw the network used in the model, specify exactly how is the output computed (notably it should be clear what all parameters of the model are), and try writing the loss function.



Let $l(v_i, v_j) = E_{i,*}O_{*,j}$ is the unormalized output of the network for input word v_i and output word v_j.
The softmax-normalized output is then $s(v_i, v_j) = \frac{e^{l(v_i,v_j)}}{\sum_k e^{l(v_i,v_k)}}$.
The loss function given w_t on input is therefore $L(w_t) = \log(s(w_t, w_{t-1})) + \log(s(w_t, w_{t+1}))$.

In practice, the full softmax output is too slow. If you know negative sampling, try sketching out how it works.
Instead of normalizing the output words using softmax, let the output for word v_i be normalized using the sigmoid function (i.e., the network then computes probability estimate for every word independently). The loss then becomes $\sigma(l(w_t, w_{t-1})) - \sum_{\text{random k words } r_i} \sigma(l(w_t, r_i))$. The random negative samples are sampled according to *unigram probability*$^{3/4}$.

Recently, structures skip-gram (SSkip-gram) model is gaining popularity. If we tell you that it uses a separate output matrix for every context offset, try guessing how it is computed.

Instead of sharing the output matrix for every context offset (-1 and +1 in our case), we have separate output matrices O_{-1} and O_1. Therefore, the embeddings also encode offset of the context words. The structured skip-gram embeddings can be computed using for example `wang2vec`.

11) *List as many NN training algorithms you know apart from standard SGD.*

For example SGD with momentum, SGD with Nesterov momentum, AdaGrad, RMSProp, RMSProp with momentum, AdaDelta, Adam; but the list is much larger.

12) *Name as many NN regularization methods you know.*

For example weight decay, dropout, zoneout, BatchNorm, LayerNorm; but again, there are many many others.

13) *Why are minibatches regularly used in NN training? Write at least two reasons.*
One reason is speed: matrix-matrix multiplication is asymptotically faster than performing several individual matrix-verctor multiplications. Also, for GPUs, there is some slowdown for every operation executed (i.e., every network node evaluated), so using batches allow evaluating on multiple inputs. The other reason is accuracy: gradients of individual inputs are quite noisy, so using an averaged gradient of multiple inputs usually results in higher overall performance of the model.