NLP Project

Eyal Gutflaish , Maor Gaon, meni Sadi

February 15, 2018

1 ARCHITECTURE

- Let a question $q = [q_1, \ldots, q_n]$, a paragraph $p = [p_1, \ldots, p_m], m >> n$.
- Pad the paragraph Note we will have to choose a maximum length of a question and a paragraph, while adding a mask to the paragraph, i.e. we will have $[w_1, \ldots, w_m, bin]$ for each paragraph while the extra words will be 0.

  Let m be the maximal length of a paragraph.

  Let $Q(x) \in R^m$ be a 1-hot vector where $P[x] = 1$
- Run a GRU over the question – $h_q = GRU(q)$.
- Run a GRU over the context paragraph – $h_p = GRU(p, initialState = h_q)$
- $start = \text{argmax}_i \, Softmax(W_1 h_p + b_1), b_1 \in R^m$
- $end = \text{argmax}_i \, Softmax(W_2 h_p + b_2), b_2 \in R^m$
- Optimize according to (only on training):

$$L_1 = CrossEntropy(Q(start), Softmax(W_1 h_p + b_1)) \quad (1)$$
$$L_2 = CrossEntropy(Q(end), Softmax(W_1 h_p + b_1)) \quad (2)$$
$$Loss = L_1 + L_2 \quad (3)$$

- return $P[start, end]$

2 STUFF MICHAEL MENTIONED HE WANTS

- Move the TF code to keras.
- Add soft attention – if so you can visualize the confusion matrix and see what words from the paragraph were helpful in predicting.

- What memory does the RNN really have - we can know that be seeing what neurons "light up". E.G. far neurons would get a very gradients, hence would not be updated – see "Vanishing gradient problem". Visualizations should be a good idea.

  See Visualizing neurons for a nice video about this

- Show contusion matrix of attention after adding attention (if adding attention)
- Notice the data set is pretty large so in order to run it fully you would need a server from Michael with a GPU. But I guess we can experiment with a small portion of the data set
- Maor - Do your regular code optimization stuff. Feel free to change class/variable names.
- Work in small batches (see the config.py for the model's parameters) – If toy want to run the all dateset ask Michael for a server
- If You decide to improve the basic model please do an alternate model class. Do not use the qa_model class. Just inherit@override or copy paste. It took me a pretty lone time to make this base line work properly. Call it model_x or something like that.

3 RUN THE MODEL

- Use config.py for changing the parameters of the model
- go to the MAIN notebook for running the model. Or just run it from any IDE – "python train"
- To change the model it'self change qa_model.py
- Notice your credits in the assignment – every one should do it's relative share. E.g. I have 2 points credit, Maor has 4 points credit and meni has x points credit. I have spent at least a weak (9h a day) in this, so Maor it means you should put 2 weeks :) and mani x*3.5 weeks.
- Enjoy