# Assignment5

May 24, 2020

## 1 Character-based convolutional encoder for NMT

(a) No, CNN architectures depends on the length of input since its filters will scan through the input, then the size of the output of the conv layer would depend on the length of input. And it affects the next layer's parameter shape.
(b) Padding size will be 1. Then $x'_{reshaped} \in \mathbb{R}^{m_{word}+2+(2*padding)}$. The minimum sentence size is 1. In this case we can scan it through at least 1 window only when there are 4 positions around it on each side. Considering that there are 2 tokens for start and for end, then the padding size should be $4-2-1=1$.
(c) It makes the model more flexible, which could both prefer lower level information($x_{conv\_out}$) and higher level information($x_{proj}$). For initialization, we should choose positive $b_{gate}$, which will lead to a larger value of $x_{gate}$ in order to let the model prefer to directly use the $x_{conv\_out}$ thus training easier.
(d) Pros of transformer: 1) Faster training by getting rid of recurrent structure. 2) Could capture higher level interaction between distant words.
(f) For verifying the correctness of the highway model, we tried a random $w_{proj}$ and $w_{gate}$:

$$[[-0.0484, -0.0198, -0.2165], [0.1328, -0.3303, -0.1018], [0.2238, 0.5419, 0.1360]]$$

$$[[0.2357, 0.0661, 0.2262], [0.5599, -0.2397, -0.0204], [0.1328, -0.0038, -0.0553]]$$

Bias $b_proj$ and $b_{gate}$: $[-0.1959, 0.0554, -0.0647]$ and $[-0.5109, -0.4980, -0.5195]$ With input:

$$[[0.1, 0.2, 0.3], [0.4, 0.5, 0.6]]$$

The output is:

$$[[0.0600, 0.1243, 0.2281], [0.2247, 0.3001, 0.5160]]$$

We copied the parameters into the *test_highway.py* file and computed the formula manually, and we got the same output.
(g) We choose to use a simple input: $torch.ones(3, 2, 10)$ which means the batch size is 3, padded word length is 2(for each word it only has 2 chars), and char embedding size is 10. The CNN output channel is set to be 3. Since the convolutional filter size is 5 and padding length is 1, the convolutional output is $\mathbb{R}^{3,3,(10+2*1-(5-1))} = \mathbb{R}^{3,3,8}$. So we should set max pooling layer's window size to be 8. After the maxpooling layer the final output is $\mathbb{R}^{3,3,1}$. And we checked the intermediate output are matching the calculated value.