COMS 4705-HW2 Emotion Classification with Neural Networks

Jing Qian (jq2282@columbia.edu)

Dear Instructor, I want to use **one late day** for this homework. Thank you!

1. Dense Network - Forward

In this problem, A_{in} has two columns, which could be considered as a batch with two inputs. So each column should be input to $W_1 A_{in} + b_1$ and get its corresponding Z_1. Here we could perform the matrix transformation on the batch with a little modification on the bias term: changing b_1 to B_1, which is a matrix having the same column number with A_{in} and each column of B_1 is b_1. Similarly, we modify b_{out} to B_{out}.

$$Z_1 = W_1 A_{in} + B_1 = \begin{bmatrix} 1 & -1 & 2 & 3 & 0 \\ 4 & 0 & -1 & 1 & 3 \\ 2 & 1 & 3 & -5 & -4 \\ 4 & -3 & 2 & 1 & -3 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 3 & 4 \\ 5 & 3 \\ 1 & 1 \\ 4 & 2 \end{bmatrix} + \begin{bmatrix} -1 & -1 \\ 2 & 2 \\ -4 & -4 \\ 3 & 3 \end{bmatrix} = \begin{bmatrix} 11 & 5 \\ 18 & 10 \\ -3 & -2 \\ 1 & -4 \end{bmatrix}.$$

To get A_1, we perform *relu* function on each element of Z_1 as following:

$$A_1 = f_1(Z_1) = \begin{bmatrix} f_1(11) & f_1(5) \\ f_1(18) & f_1(10) \\ f_1(-3) & f_1(-2) \\ f_1(1) & f_1(-4) \end{bmatrix} = \begin{bmatrix} 11 & 5 \\ 18 & 10 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Similarly, we could get Z_{out} and A_{out}:

$$Z_{out} = W_{out} A_1 + B_{out} = \begin{bmatrix} 2 & -2 & -1 & 3 \\ -2 & 1 & -5 & 4 \end{bmatrix} \begin{bmatrix} 11 & 5 \\ 18 & 10 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 12 & 12 \\ 3 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 3 \end{bmatrix},$$

$$A_{out} = f_{out}(Z_{out}) = \begin{bmatrix} f_{out}(1) & f_{out}(2) \\ f_{out}(3) & f_{out}(3) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 3 \end{bmatrix}.$$

2. Backpropagation

2.1. For $i = 1, \cdots, 7$, write the formula to calculate $\frac{\partial Loss}{\partial x_i}$.

According to the expression of fs, we could get the following differential and partial differential formulas:

$$
\begin{aligned}
\frac{\partial x_5}{\partial x_1} =& \frac{df_5}{dx_1} = \frac{\exp(-x_1)}{(1+\exp(-x_1))^2}, \\
\frac{\partial x_6}{\partial x_2} =& \frac{\partial f_6}{\partial x_2} = a + c * x_3, \\
\frac{\partial x_6}{\partial x_3} =& \frac{\partial f_6}{\partial x_3} = b + c * x_2, \\
\frac{\partial x_7}{\partial x_4} =& \frac{df_7}{dx_4} = 2 * x_4, \\
\frac{\partial x_8}{\partial x_5} =& \frac{\partial f_8}{\partial x_5} = -\frac{\exp(x_5+x_6)}{(\sum_{i=5}^{7}\exp(x_i))^2}, \\
\frac{\partial x_8}{\partial x_6} =& \frac{\partial f_8}{\partial x_6} = \frac{\exp(x_6)(\exp(x_5)+\exp(x_7))}{(\sum_{i=5}^{7}\exp(x_i))^2}, \\
\frac{\partial x_8}{\partial x_7} =& \frac{\partial f_8}{\partial x_7} = -\frac{\exp(x_7+x_6)}{(\sum_{i=5}^{7}\exp(x_i))^2}.
\end{aligned}
$$

If there is no function relationship between two units, their partial differential term equals to zero. In other words:

$$
\frac{\partial x_5}{\partial x_2} = \frac{\partial x_5}{\partial x_3} = \frac{\partial x_5}{\partial x_4} = \frac{\partial x_6}{\partial x_1} = \frac{\partial x_6}{\partial x_4} = \frac{\partial x_7}{\partial x_1} = \frac{\partial x_7}{\partial x_2} = \frac{\partial x_7}{\partial x_3} = 0.
$$

Then we could calculate $\frac{\partial Loss}{\partial x_i}$ according to Chain Rule as following:

$$
\begin{aligned}
\frac{\partial Loss}{\partial x_7} =& \frac{\partial Loss}{\partial x_8}\frac{\partial x_8}{\partial x_7} = \frac{\partial Loss}{\partial x_8}(-\frac{\exp(x_7+x_6)}{(\sum_{i=5}^{7}\exp(x_i))^2}) = -\frac{\partial Loss}{\partial x_8}\frac{\exp(x_7+x_6)}{(\sum_{i=5}^{7}\exp(x_i))^2}, \\
\frac{\partial Loss}{\partial x_6} =& \frac{\partial Loss}{\partial x_8}\frac{\partial x_8}{\partial x_6} = \frac{\partial Loss}{\partial x_8}\frac{\exp(x_6)(\exp(x_5)+\exp(x_7))}{(\sum_{i=5}^{7}\exp(x_i))^2}, \\
\frac{\partial Loss}{\partial x_5} =& \frac{\partial Loss}{\partial x_8}\frac{\partial x_8}{\partial x_5} = \frac{\partial Loss}{\partial x_8}(-\frac{\exp(x_5+x_6)}{(\sum_{i=5}^{7}\exp(x_i))^2}) = -\frac{\partial Loss}{\partial x_8}\frac{\exp(x_5+x_6)}{(\sum_{i=5}^{7}\exp(x_i))^2}, \\
\frac{\partial Loss}{\partial x_4} =& \frac{\partial Loss}{\partial x_8}\sum_{j=5}^{7}\frac{\partial x_8}{\partial x_j}\frac{\partial x_j}{\partial x_4} = \frac{\partial Loss}{\partial x_8}\frac{\partial x_8}{\partial x_7}\frac{\partial x_7}{\partial x_4} = -\frac{\partial Loss}{\partial x_8}\frac{\exp(x_7+x_6)}{(\sum_{i=5}^{7}\exp(x_i))^2}(2 * x_4), \\
\frac{\partial Loss}{\partial x_3} =& \frac{\partial Loss}{\partial x_8}\sum_{j=5}^{7}\frac{\partial x_8}{\partial x_j}\frac{\partial x_j}{\partial x_3} = \frac{\partial Loss}{\partial x_8}\frac{\partial x_8}{\partial x_6}\frac{\partial x_6}{\partial x_3} = \frac{\partial Loss}{\partial x_8}\frac{\exp(x_6)(\exp(x_5)+\exp(x_7))}{(\sum_{i=5}^{7}\exp(x_i))^2}(b + c * x_2), \\
\frac{\partial Loss}{\partial x_2} =& \frac{\partial Loss}{\partial x_8}\sum_{j=5}^{7}\frac{\partial x_8}{\partial x_j}\frac{\partial x_j}{\partial x_2} = \frac{\partial Loss}{\partial x_8}\frac{\partial x_8}{\partial x_6}\frac{\partial x_6}{\partial x_2} = \frac{\partial Loss}{\partial x_8}\frac{\exp(x_6)(\exp(x_5)+\exp(x_7))}{(\sum_{i=5}^{7}\exp(x_i))^2}(a + c * x_3), \\
\frac{\partial Loss}{\partial x_1} =& \frac{\partial Loss}{\partial x_8}\sum_{j=5}^{7}\frac{\partial x_8}{\partial x_j}\frac{\partial x_j}{\partial x_1} = \frac{\partial Loss}{\partial x_8}\frac{\partial x_8}{\partial x_5}\frac{\partial x_5}{\partial x_1} = -\frac{\partial Loss}{\partial x_8}\frac{\exp(x_5+x_6)}{(\sum_{i=5}^{7}\exp(x_i))^2}\frac{\exp(-x_1)}{(1+\exp(-x_1))^2}.
\end{aligned}
$$

2.2. Calculate $\frac{\partial Loss}{\partial a}, \frac{\partial Loss}{\partial b}, \frac{\partial Loss}{\partial c}, \frac{\partial Loss}{\partial d}$ and update the learned a, b, c, d.

Similar to the partial differential formula in 2.1., we could get the following fomula:

$$\begin{aligned}
\frac{\partial x_6}{\partial a} =& \frac{\partial f_6}{\partial a} = x_2, \\
\frac{\partial x_6}{\partial b} =& \frac{\partial f_6}{\partial b} = x_3, \\
\frac{\partial x_6}{\partial c} =& \frac{\partial f_6}{\partial c} = x_2 * x_3, \\
\frac{\partial x_7}{\partial d} =& \frac{\partial f_7}{\partial d} = 1,
\end{aligned}$$

Also, due to similar reasons in 2.1, $\frac{\partial x_5}{\partial a} = \frac{\partial x_5}{\partial b} = \frac{\partial x_5}{\partial c} = \frac{\partial x_5}{\partial d} = \frac{\partial x_6}{\partial d} = \frac{\partial x_7}{\partial a} = \frac{\partial x_7}{\partial b} = \frac{\partial x_7}{\partial c} = 0$.

We could get the values of x_5, x_6, x_7 according to the values of x_1, x_2, x_3, x_4 and a, b, c, d.

$$\begin{aligned}
x_5 =& f_5(x_1) = \frac{1}{1+\exp(-x_1)} = \frac{1}{1+\exp(-0)} = 1/2, \\
x_6 =& f_6(x_2, x_3) = a * x_2 + b * x_3 + c * x_2 * x_3 = 3 * 2 + 4 * (-1) + 2 * 2 * (-1) = -2, \\
x_7 =& (x_4)^2 + d = 2^2 + 2 = 6.
\end{aligned}$$

So we have:

$$\begin{aligned}
\frac{\partial Loss}{\partial a} =& \frac{\partial Loss}{\partial x_8} \sum_{j=5}^{7} \frac{\partial x_8}{\partial x_j} \frac{\partial x_j}{\partial a} = \frac{\partial Loss}{\partial x_8} \frac{\partial x_8}{\partial x_6} \frac{\partial x_6}{\partial a} = \frac{\partial Loss}{\partial x_8} \frac{\exp(x_6)(\exp(x_5)+\exp(x_7))}{(\sum_{i=5}^{7}\exp(x_i))^2} x_2 \\
=& 3 * \frac{\exp(-2)[\exp(1/2)+\exp(6)]}{(\exp(-2)+\exp(1/2)+\exp(6))^2} * 2 = 0.002, \\
\frac{\partial Loss}{\partial b} =& \frac{\partial Loss}{\partial x_8} \sum_{j=5}^{7} \frac{\partial x_8}{\partial x_j} \frac{\partial x_j}{\partial b} = \frac{\partial Loss}{\partial x_8} \frac{\partial x_8}{\partial x_6} \frac{\partial x_6}{\partial b} = \frac{\partial Loss}{\partial x_8} \frac{\exp(x_6)(\exp(x_5)+\exp(x_7))}{(\sum_{i=5}^{7}\exp(x_i))^2} x_3 \\
=& 3 * \frac{\exp(-2)[\exp(1/2)+\exp(6)]}{(\exp(-2)+\exp(1/2)+\exp(6))^2} * (-1) = -0.001, \\
\frac{\partial Loss}{\partial c} =& \frac{\partial Loss}{\partial x_8} \sum_{j=5}^{7} \frac{\partial x_8}{\partial x_j} \frac{\partial x_j}{\partial c} = \frac{\partial Loss}{\partial x_8} \frac{\partial x_8}{\partial x_6} \frac{\partial x_6}{\partial c} = \frac{\partial Loss}{\partial x_8} \frac{\exp(x_6)(\exp(x_5)+\exp(x_7))}{(\sum_{i=5}^{7}\exp(x_i))^2} * x_2 * x_3 \\
=& 3 * \frac{\exp(-2)[\exp(1/2)+\exp(6)]}{(\exp(-2)+\exp(1/2)+\exp(6))^2} * 2 * (-1) = -0.002, \\
\frac{\partial Loss}{\partial d} =& \frac{\partial Loss}{\partial x_8} \sum_{j=5}^{7} \frac{\partial x_8}{\partial x_j} \frac{\partial x_j}{\partial a} = \frac{\partial Loss}{\partial x_8} \frac{\partial x_8}{\partial x_7} \frac{\partial x_7}{\partial d} = \frac{\partial Loss}{\partial x_8} (-\frac{\exp(x_7+x_6)}{(\sum_{i=5}^{7}\exp(x_i))^2}) \\
=& -3 * \frac{\exp(-2+6)}{(\exp(-2)+\exp(1/2)+\exp(6))^2} = -0.001.
\end{aligned}$$

So the learned parameters a, b, c, d after updated are:

$$
\begin{aligned}
a \leftarrow & a - \eta \frac{\partial Loss}{\partial a} = 3 - 0.1 * 0.002 = 2.9998, \\
b \leftarrow & b - \eta \frac{\partial Loss}{\partial b} = 4 - 0.1 * (-0.001) = 4.0001, \\
c \leftarrow & c - \eta \frac{\partial Loss}{\partial c} = 2 - 0.1 * (-0.002) = 2.0002, \\
d \leftarrow & d - \eta \frac{\partial Loss}{\partial d} = 2 - 0.1 * (-0.001) = 2.0001.
\end{aligned}
$$

3. Coding Reflections

3.1. Extension 1: changes to the preprocessing of the data

The first extension I tried is to use tokenizers specifically for Tweets: TweetTokenizer from nltk package. I implemented it in the utils.py. I modified mainly in the function get_tokens() and also the class EmotionDataset, function make_vectors(), function vectorize_data. Moreover, in the main() function in hw2.py, I added the code to run extension1. Since the data we studied are tweets, it makes sense to use tokenizers specifically for Tweets. In fact, running on the same dense neural network, changing from original tokenizers to TweetTokenizer increased the F-score of test data by around 1.6%. So I think analyzing Tweets with tokenizers specifically for Tweets actually improve the model performance.

3.2. Extension 2: architecture change

The second extension I tried is flattening embeddings using average method other than sum in the dense network. I implemented it in the models.py as a new class ExperimentalNetwork. Moreover, in the main() function in hw2.py, I added the code to run extension2. I thought the original flattening embeddings with sum ignored the difference between long sentence and short sentence. A long sentence with relatively small word embeddings may have similar sum to a short sentence with relatively large word embeddings. So I tried the average method, i.e., dividing the sum by the unpadded sentence length. But in fact, changing from original flattening to average method decreased the F-score of test data by almost half. I guess maybe the sentence length is not so important in our classification task. Also, I notice there are both positive and negative values in the embeddings, which may cancel out and also weaken the effect of sentence length.