Convolutional neural networks for text classification

March 23, 2019; due March 29, 2019 (11:59 pm)

In this homework you will learn how to build a simple convolutional neural networks (1 convolution layer with max pooling + 1 activation layer) from scratch, and use the model to solve text classification problem. As optional, you also have a chance to build real life CNN models using Keras + Tensorflow and use it to challenge the model you build from scratch.

The first part of homework should be done **independently**.

Math preliminaries

1. What is the form of sigmoid function $\sigma(z)$? Show that $\sigma'(z) = \sigma(z)[1 - \sigma(z)]$.
2. Another popular activation function is $tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$, show that $tanh'(z) = 1 - g^2(z)$.
3. For a single variable single layer perceptron with sigmoid activation function (equivalent to LR) and loss function defined as:

   $$\hat{y}_i = \sigma(w_1 x_i + w_0)$$

   $$L(w_0,\ w_1) \ = \sum_i y_i lg(\hat{y}_i) + (1 - y_i) lg(1 - \hat{y}_i)$$

   Show that:

   $$\frac{\partial L}{\partial w_1} = \sum_i (y_i - \hat{y}_i) x_i$$

   $$\frac{\partial L}{\partial w_0} = \sum_i (y_i - \hat{y}_i)$$

4. For column vectors x and w, and matrix M, define the gradient operator $\nabla_x = (\frac{\partial}{\partial x_0}, \frac{\partial}{\partial x_1}, ..., \frac{\partial}{\partial x_n})^T$, show that:

   $$\nabla_x (w^T x) = w$$
   $$\nabla_x (x^T w) = w$$
   $$\nabla_x (x^T M x) = 2Mx$$

5. Let's expand Q3 to a more general case. Suppose there is a single layer perceptron with multiple variables (each x_i is a vector):

   $$\hat{y}_i = \sigma(w^T x_i\ )$$

   $$L(w\ ) \ = \sum_i y_i lg(\hat{y}_i) + (1 - y_i) lg(1 - \hat{y}_i)$$

   Show that:

   $$\nabla_w L(w) = \sum_i (y_i - \hat{y}_i) x_i$$

(hint: use the notation defined in Q4)

6. In a CNN illustrated as Fig 1, suppose the loss function is:

$$L(U,\ w) = y\, lg(\hat{y}) + (1-y) lg(1-\hat{y})$$

From the conclusion in Q5, we can get that:

$$\nabla_w L(U,w) = (y-\hat{y})h$$

Can you calculate $\nabla_{u_i} L(U,w)$ using similar techniques?

This part of homework can be done either **independently** or in a small group with up to 3 people.

CNN

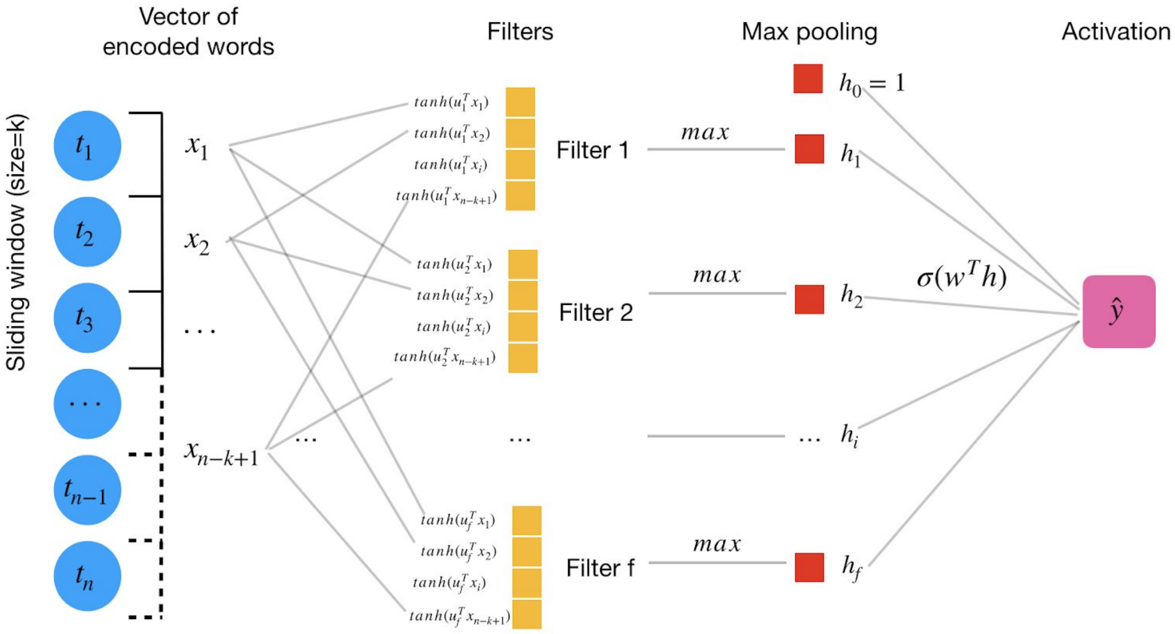

Fig 1. Schema of the 1 layer CNN

Follow the instruction in the notebook, and implement the missing code to build the CNN classifier from scratch. Note that the training might be very slow. Consider reducing the training data size and vocabulary size for testing your code. Ask questions in Piazza if you get blocked.

Hint: In this CNN, words should be one-hot encoded, but we actually numerically encoded it in the code. This is a sparse trick we did to boost the efficiency, try to understand how it works.

Deliverables (zip them all)

- pdf version of your final notebook
- Use the best model you trained, generate the prediction for test.txt, name the output file prediction.csv (Be careful: the best model in your training set might not be the best model for the test set).
- After you finished the run, does the model perform better than the bag of words model you built last week? What do you think that contributes to the difference in performance?
- HW1_writeup.pdf: summarize the method you used and report their performance. If you worked on the optional task, add the discussion. Add a short essay discussing the biggest challenges you encounter during this assignment and what you have learnt.