HW 2 Solutions

Andrei A Simion

February 28, 2023

1 Negative Sampling for CBOW

In class we looked at the Skip-Gram and CBOW models and we looked at Negative Sampling. In the Skip-Gram model, we want to predict the the outside words from the center word. Negative Sampling removed the softmax dependency, which is expensive. The upshot is for a (w_c, w_o) pair we have

$$P(w_o|w_c) = \frac{\exp b_{w_o}^{\intercal} a_{w_c}}{\sum_{j=1}^{|V|} \exp b_{w_j}^{\intercal} a_{w_c}}$$

and replace this by

$$P(w_o|w_c) = (\frac{1}{1 + \exp -b_{w_o}^{\intercal} a_{w_c}}) E_{w_k \sim P(w)} [\prod_{k=1}^{K} \frac{1}{1 + \exp b_{w_k}^{\intercal} a_{w_c}}]$$

You can consider the expectation by: "Draw K random samples from the set V, with probability $P(w)$". For CBOW, we want to predict the inner word from the words around it. Thus, if $m = 1$, for example, we have

$$P(w_c|w_{c-1}, w_{c+1}) = \frac{\exp b_{w_c}^{\intercal} a_{avg}}{\sum_{j=1}^{|V|} \exp b_{w_j}^{\intercal} a_{avg}}$$

In this case, a_{avg} is the average a vector of the words w_{c-1}, w_{c+1}. The first goal is to submit what the objective for Negative Sampling would look like for CBOW. I.e., for the above example, what would it look like? Please submit a formula with justification. Your next goal is to take the notebook I give you and, using the hints and the notebook for Skip-Gram in class, implement the Negative Sampling Approach for CBOW. Can you print out the associated vectors for the validation words? Are they related, in turn, to each validation word. *See notebook.*

2 Mathematical Problems

Below are some mathematical drills.

Problem 1 Consider the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$. What is the derivate of $\sigma(x)$ in terms of $\sigma(x)$. You need to get the derivative and then simplify a bit. Do the same for hyperbolic tangent, $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. *They should get the gradient of the sigmoid is $\sigma(x)(1-\sigma(x))$. For tanh, the gradient is $1 - tanh(x)^2$.*

Problem 2 Assume you do CBOW and Skip-Gram with negative sampling. Assume $m = 1$. Which method, on average, will get more training samples? Suppose there are 3 sentences with 7, 8, and 11 tokens. How many training sampling (positive training samples), will each method get. Draw a picture of a sentence with token counts and think about the number of samples each method gives. This is why Skip-Gram is used more often. It is more "sample efficient": you get more training data. *The window is $m = 1$. Consider Skip-Gram first. For the sentence with 7 words we fix the center word and move it, we should get 2 training examples for each word except the two at the ends, this gives 5*2 + 2 = 12. Similarly, for others we get 12 + 2 and 18 + 2. This gives a total of 12 + 14 + 20 = 46. For CBOW, we need two context words around each center word, so words in the ends can't be used. This gives 5 + 6 + 9 = 20 positive examples.*

Problem 3 Assume you have input $a_0 = x$, and you set $z_0 = w^{[1]}a_0 + b^{[1]}$, then $a_1 = \sigma(z_0)$, then $z_1 = w^{[2]}a_1 + b^{[2]}$ and finally $a_2 = \sigma(z_1)$. Assume that the loss is $l = -log(a_2)$. What is the derivative of l with respect to each of the 4 parameters $w^{[1],[2]}$ and $b^{[1],[2]}$ (4 derivatives - express in terms of a and z and other parameters if necessary)? What happens if z_0 is very large to the derivative $\frac{da_1}{dz_0}$? How would this affect learning for $w^{[1]}$ and $b^{[1]}$. Everything here is a scalar, 1 dimensional. It's like you have 1 training sample and you are doing *Stochastic* gradient descent: batch size = 1. *The idea is to just use the chain rule and bunch of times. We have* $\frac{\partial l}{\partial w^{[2]}} = \frac{\partial l}{\partial a_2}\frac{\partial a_2}{\partial z_1}\frac{\partial z_1}{\partial w^{[2]}} = (-1/a_2)(\sigma(z_1)(1-\sigma(z_1)))a_1$ *and* $\frac{\partial l}{\partial b^{[2]}} = \frac{\partial l}{\partial a_2}\frac{\partial a_2}{\partial z_1}\frac{\partial z_1}{\partial b^{[2]}} = (-1/a_2)(\sigma(z_1)(1-\sigma(z_1)))$. *We also have* $\frac{\partial l}{\partial w^{[2]}} = \frac{\partial l}{\partial a_2}\frac{\partial a_2}{\partial z_1}\frac{\partial z_1}{\partial a_1}\frac{\partial a_1}{\partial z_0}\frac{\partial z_0}{\partial w^{[1]}} = (-1/a_2)(\sigma(z_1)(1-\sigma(z_1)))(w^{[2]})(\sigma(z_0)(1-\sigma(z_0)))x$ *and* $\frac{\partial l}{\partial b^{[2]}} = \frac{\partial l}{\partial a_2}\frac{\partial a_2}{\partial z_1}\frac{\partial z_1}{\partial a_1}\frac{\partial a_1}{\partial z_0}\frac{\partial z_0}{\partial w^{[1]}} = (-1/a_2)(\sigma(z_1)(1-\sigma(z_1)))(w^{[2]})(\sigma(z_0)(1-$

$\sigma(z_0)))$. *The idea is to realize that the derivatives* $\frac{da_1}{dz_0}$ *and* $\frac{da_2}{dz_1}$ *could be zero if* z_0 *or* z_1 *are too extreme. This prevents learning, since the gradients are zero and SGD will have a 0 update.*