Assignment 1 Q/A
CSE 538

1. Properly debugging in TF?

Question:

How should we be visually debugging inside of tensorflow? For example inside the loss function, if we wanted to evaluate our A or B values?

A = <some operations on true_w and inputs>

A = tf.Print(A, [A], message='DEBUG A: ')

B = <some operations>

return tf.substract(B,A)

I thought the way tf.Print() worked is that every time this function is evaluated (each training epoch) it would print out the evaluated A(second param of print) tensor. However when I run my code I do not see any messages. Is there another way I should be approaching this?

Answer:

What you described is a correct way to print the contents of a variable in a run.

Most likely, your code is not actually executed. To see the code is included during the graph build, add a regular print statement to make sure it is included in the graph construction.

2. Question regarding embedding_size

Question:

Is it possible to change the embedding_size for the word vectors for experimentation? When I changed, it threw an error because the pretrained model uses 128 size vectors.

Answer:

to try different embedding size, you would not be able to use the pre-trained model. (You can comment it out.) So, if you want to try different embedding size, you have to train from scratch.

3. Getting the Unigram probabilities for predicting words.

Question:

I was trying to extract the unigram probabilities for the labels vector from the unigram_prob vector using tf.embedding_lookup. But, I'm getting an error with respect to the shape of the unigram_prob vector. Am I doing something wrong here?

The error I'm getting is the following

ValueError: Shape must be at least rank 1 but is rank 0 for 'TargetUnigramProbs/Gather' (op: 'Gather') with input shapes: [], [?].

Answer:

unigram_prob is not a tensor. So you cannot use tf.embedding_lookup on it. It is a python list, so you might want to convert it to a tensor if you want to use it on tensorflow API func. Try using tf.gather() instead.

4. number of layers in our neural network

Question:

Does the neural net for our project have just one input layer and one output layer as we have found weights just once and not used rlu ?

Answer:

Of course you can see the skip-gram as a neural network. If so, it is a network with three layers, input, hidden(or called embedding layer in NLP) and output layer. So there are two big weights matrix, one is called center matrix(center vector, the v_c in the loss function) and another is outer matrix(context matrix, the u_w in the loss function). Also there is no nonlinear layers (except the softmax in output layer) between input and hidden layer or between hidden layer and output layer. So this is a big difference from common neural networks. In real neural networks, nonlinear layers are the biggest and most important magic. So this skip-gram model's structure just looks like a neural network.

Also, you can just see the skip-gram as this: this model is just find two big matrix- center and outer- to minimize the loss we designed. Then this becomes a simple optimize problem.

5. Cross entropy loss notes

Question:

Where can I find the notes on the cross entropy loss function?

Answer:

$$CrossEntropy(p, q) = -\sum_{x} p(x) log(q(x))$$

cross entropy loss = the negative of the log-likelihood function we saw in class.

6. Cross entropy loss

Question:

What should be the average cross entropy loss?

Answer:

My tests showed around 4.x

7. Question in loss function

Question:

I think the input of the 'cross_entropy_loss' function should be 'Input' and 'sm_weights', while 'Input' is a [Batch size, embedding size] matrix which is the output of the hidden layer and 'sm_weights' is a [Vocabulary size, embedding size] matrix which is the weights matrix of the output layer.

Answer:

Current argument signature is written so since it will be too slow for most of the students to compute against entire vocabulary for training. Of course, you are welcome to change the argument signature to implement the original version. If it is too slow, you can follow the instructions in the comment to implement the simpler/runnable version of it.

8. cross_entropy_loss arguments

Question:

Could you please elaborate on the arguments that are passed to the cross_entropy_loss function? I understand that vc and uo are the vectors corresponding to a context word and an outer word, but what is uw? Is it a collection of uo?

Answer:

Yes. u_o is the output vector of the current output (label) word, and u_w is the output vector of all label words in the batch. (Instead of using all vocabulary, we are only using the words in the current batch.)

Question:

How do we know which word is the current label word if all the function is receiving is both embedding matrices? Is there tensorflow magic going on?

It would seem that Uw is simply the true_w matrix as that is the batch size of all Uo. Although I don't see how it goes the other way

Answer:

(inputs[0],true_w[0]),...,(inputs[i],true_w[i[),...,(inputs[batch_size-1], true_w[batch_size-1[)
are the pairs of (context word embedding, output word embedding).
u_w is indeed from true_w matrix. You just sum over all vectors from true_w.

9. Getting nan for cross-entropy loss

Question:

After implementing cross - entropy, for steps greater than 4000, I am getting the average loss as nan. Is it because the loss function has not been implemented correctly? Or are we supposed to change the parameters like batch-size??

Answer:

It should work without changing parameters. Try to debug your loss function by checking the intermediate values, looking for the variables/tensors that become 0. (and why)

Question:

Is it advisable to replace Nan values encountered in the tensors with a 0?

Answer:

Not really. Mainly because it is likely that you have an issue with your implementation.

10. NCE loss initial value

Question:

What would be the initial value of nce loss, as in for the first iteration? I'm getting a large number, as in 3 digits. Is it a direct sum over the logs or a mean of the logs for the sample data?

Answer:

Since we have pre-trained model, if the loss is not decreasing after the first step, not converges to 1.x, there is something wrong.
The loss displayed is a mean.

Question:

If the loss is slightly negative, then the positive term i.e log of probability for the correct word is just slightly greater than the log of the summed probability for the negative words, hence a positive term in the brackets. Is that theoretically not possible ?

Answer:

You are summing the positive term and the summed negative term, which can only be negative or 0. (because they are log of values in [0-1)] Then, you negate it, producing a positive value.

11. Question in NCE loss function

Question:

I'm getting loss as nan for learning_rate=1 in the GradientDescentOptimizer. Should my implemented loss function run properly with learning_rate=1 also(for nce_loss). I have changed my learning_Rate and num_steps to different values for which I'm not seeing nan values.

Answer:

Yes it should run properly with learning_rate=1.0. In general, if you see "nan" there's something wrong.

Maybe, try to add a very small number (1e-10 for example) when you take a log, to avoid passing 0 as its input. This might help.

12. Need clarification regarding NCE loss

Question:

Can you please explain how are we supposed to use the samples for the NCE loss.
Is this required to take a values out of the given vocabulary?
Also please confirm where to use the weights in NCE calculation?

Answer:
The samples are the negative samples. You want to compute the probabilities of these negative samples, given a context word wc.
nce_weights is the output matrix. It is used to compute the score. uo is the output vector taken from nce_weights matrix.

Question:
I am wondering if we can use the NCE loss function of tensorflow, directly? I'm assuming that's not the case?

Answer:
Don't use the NCE function of tensorflow. (If you do, you are giving up the points for NCE loss function implementation.)

Question:
I am very confused as to how to use labels to get negative samples

Answer:
inputs: context word vectors
labels: output words that are paired with each context words // word ids
samples: negative samples //word ids

13. Problem with NCE loss equation

Question:
As per the below equation of calculation NCE loss if I am considering the final summation(over all the data from batch) I am getting a loss value like "40000". But If I am ignoring the final summation then I am getting the loss value as "1.5". Please let me know which one to follow.

Answer:
You don't have to do the summation over the batch since the code that calls the loss function actually take care of it.

14. Regarding NCE justification task.

Question:

What is expected to be written for justification of the NCE? Are we supposed to write a summary of what we understand of NCE loss method. Do we have to use equations from the paper to explain our justification? Can you please tell us what is expected in this task?

Answer:
Basically, explain NCE loss. Equations would be necessary to explain NCE. "No more than one page" means, do not write it too long, but also implies that I am expecting one or two sentences long rough description.

15. Regarding weights variable in NCE loss function

Question:
What is the difference between weights and embedding?
In cross_entropy function, we send the embedding of inputs and labels. Why aren't we doing the same for NCE loss?

Answer:
Actually we are doing the same thing for cross entropy and nce:
- for cross entropy, inputs is from "embedding" and true_w is from "sm_weights".
- for nce, inputs is also from "embedding" and it passes "weights" and "biases".

For both cross-entropy and nce, we need two matrices: One to get embeddings from vocabulary, and the other to predict the target words.

I think the diagrams from this blog post explains it clearly.
https://towardsdatascience.com/learn-word2vec-by-implementing-it-in-tensorflow-45641adaf2ac

16. Pre-trained model

Question:
Would you tell us how the pre-trained model is trained? 200000 iteration with nce or cross entropy?

Answer:
It is trained using NCE loss for around 4M steps.

17. Regarding word analogy

Question:
While computing the similarity is there any particular type of similarity that we are supposed to use?

Answer:
cosine similarity would be the go-to method.

Question:
Also, how do we define least and most illustrative pair? Should we take it as the maximum similar to any of the input training data or average it over all over input data.

Answer:
One way would be taking the average of the difference vectors of the example pairs, and find the most/least similar pair.

Question:
what is the answer file for the word_analogy_test.txt set?

Answer:
The answer file is what you generate after running the word analogy task. The answer file for word_analogy_test.txt is not given. So you cannot check the accuracy for the test file. Once you are confident with the model, you generate the answer file with that model and submit it with the assignment.

18. Constant Accuracy

Question:
I generated models (nce) using 4 different values of skip window but the accuracy of word analogy didn't even change a bit. Is it possible or there might be some issue with my code?

Answer:
I would expect at least a little bit of differences in the embeddings. But they still might generate the same answers.
Check the similarity scores in your word_analogy code to see if the different skip-window embeddings produce different similarity scores.

19. words:{first, american, would}

Question:
can you please explain what you mean by this :
Top 20 similar words according to your NCE and cross entropy model
for the words:{first, american, would}
Are these the nearest to words that get printed while running word2vec_basic.py?

Answer:
They can be obtained using the similar method as we print out those nearest words.
You only need to do it after you are done training, by loading the trained model.

Question:
Are we able to use the same similarity.eval() function to load the nearest words, as is done during training? Or are we supposed to compute these using the dictionary and embeddings lists available to us when we load the model using pickle?

Answer:
You would need to load the model and compute this separately.

Question:
Are we supposed to be running word_analogy.py on word_analogy_dev.txt for experiments then on word_analogy_test.txt for the best models? I don't see the set of words {first, american, would} in either or maybe I'm not clear on how to get the top twenty similar words?

Answer:
You would have to write another python script that loads the model, and look for the similar words for {first, american, would}

20. Regarding analogy task section in report.

Question:
Should we write the whole output from analogy task for 5 different configurations in analogy task section? Or just the accuracy and explanation is enough?

Answer:
Accuracy and explanation would be enough.

21. README Vs Report

Question:
I am not getting what different content will we write in the readme and the report. I think a lot of things will overlap. Could you please briefly explain what is expected in each file?

Answer:
Think of it as:

- Implementation details in the README
- And the rest in the report. (Mention implementation details here whenever necessary.)

22. Can we edit other parts of the file

Question:
Can we add a few things to other parts of the word2vec_basic.py file? other than the space provided to write the code for generating batch.
For example : Making some global variables in main

Answer:
You can. Just mark by commenting in the code when you added something extra. For example,

Adding some_global_var for XXX
some_global_var

Other Questions

If you didn't find your question here, you can ask it in Piazza, meet with TAs during office hours or email me: mbastan@cs.stony brook.edu

NOTES

- I believe many of you are wondering how many more steps should I train this? There is no hard rule for this. On top of the pre-trained embeddings, I trained about 20,000 steps more to see a little bit less than 2% performance improvement (in the word analogy test.) This takes about 20min on my laptop using only CPU.

- General tips when you work on tensor computations:
    - Break the whole list of operations into smaller ones. (Like the comments at "cross_entropy_loss" function.)
    - Write down the shapes of the tensors.

In case of failure to submit to blackboard

Below is only for the students who are experiencing failure at blackboard. If you can or already have submitted, ignore this.

FOLLOW BELOW STEPS AS IS. OTHERWISE I WON'T RECOGNIZE IT AS PROPER SUBMISSION.

1. Make a zip file with model files only.
Filename: <YOUR_SBU_ID>_models.zip

2. Upload this file to your Stony Brook CS/ Stony Brook Google Drive.
3. Get a sharable link of the file with either of the options
   - Anyone with the link
   - Anyone at Stony Brook University Dept. of Computer Science
   - Anyone at Stony Brook University Dept. of Computer Science with link

   * If you are using Stony Brook Google Drive, not CS one, then choose the one equivalent.

3. Create a file:

Filename: models_link.txt
The first line should be the link to the link from 3

4. Create a zip file with all required files except the model files, and "models_link.txt". (This file should be small enough for black board.)
Filename: <YOUR_SBU_ID>.zip

5. Upload the file at 4 to blackboard.

** Mind that I will be checking the last modified date of the files. If the date is later than due, point deductions will be applied to be fair with others.