REPORT – HW1

Hemasai Aishwarya Vijayakumar

SBUID: 112673842

Python Version: 3.7
TensorFlow Version: 1.14.0

A) HYPER PARAMETER TUNING

Default configurations gave an accuracy of **31 for cross entropy and 31.3 for NCE**

Observations:

- Learning rate reduced from 1.0 to 0.1- It is observed that the accuracy decreased when the learning rate is decreased.
  But whereas**,** when the learning rate is decreased to 0.5, the cross entropy accuracy increased to 31.1%. There is no much difference in the loss as well.
- Increasing batch_size ,decreasing skip_window and num_skips improved the accuracies of both the models
- Num_sampled value affects NCE more than the cross entropy model(as per configuration 2)
- Doubling the skip_window increases the cross entropy accuracy by 0.3% although the loss value almost remains the same. Whereas, NCE model's accuracy decreased by 0.6%

FIVE CONFIGURATIONS FOR CROSS ENTROPY AND NCE MODEL:

1) BEST CONFIGURATION:

CROSS ENTROPY	NCE
batch_size = 256, num _sampled = 128 **Accuracy: 31.6%**	batch_size = 256, num _sampled = 128 **Accuracy: 31.8%**

2) CONFIGURATION 2:

CROSS ENTROPY	NCE
num _sampled = 128 **Accuracy: 30.9%**	num _sampled = 128 **Accuracy: 29.4%**

3) CONFIGURATION 3:

CROSS ENTROPY	NCE
batch_size = 256, max_num_steps:=100001 **Accuracy: 31.1%**	batch_size = 256, max_num_steps:=100001 **Accuracy: 31%**

4) CONFIGURATION 4:

CROSS ENTROPY	NCE
max_num_steps:=100001 **Accuracy: 31.1%**	max_num_steps:=100001 **Accuracy: 31.2%**

5) CONFIGURATION 5:

CROSS ENTROPY	NCE
skip_window = 8 , num_skips = 16 , batch_size = 256, **Accuracy: 31.1%**	skip_window = 8 , num_skips = 16 , batch_size = 256, **Accuracy: 31.1%**

6) CONFIGURATION 6:

CROSS ENTROPY	NCE
batch_size = 256, **Accuracy: 31%**	batch_size = 256 **Accuracy: 31.1%**

a. There is an increase in accuracy on increasing the batch_size and num_sampled for both the models. Only increasing batch_size decreased NCE accuracy to 31.1% but cross entropy accuracy remained the same.
b. Doubling skip_window, num_skips and batch_size didn't really make a big impact on the accuracy of both the models.
c. Reducing the max_num_steps didn't impact much as well
d. Significant difference was noticed when num_sampled was doubled. Both the accuracies decreased.

B) TOP 20 SIMILAR WORDS

WORD	CROSS ENTROPY MODEL	NCE MODEL
First	last , second , original , next , best , same , following,	until ,during ,book , before ,time , most ,

	most ,entire , latter , third , another , final ,largest , gabby , current , before , bodega , protect ,outside	early ,being ,when , theory ,only ,where , work ,famous ,up , however ,later ,use , at ,law
American	German , British , English , French , Italian , Canadian , Russian , European, Irish, international, Ancient, Spanish, Christian, autres, reckless, unofficial, counterrevolutionaries, gallimard, non, Greek	would ,could ,proudhon , i ,free ,so , english , modern ,joseph ,labor , him ,using ,involved , law ,being ,about , theory ,within , rights ,word
Would	will ,could ,must ,did , can ,should ,last ,does , german ,may ,british , english ,might ,french , italian ,second ,original , do ,russian ,next	could ,i ,him ,american , proudhon ,labor,using,word, my ,free ,so ,involved , english ,modern , will, through ,did ,within rights ,involved

Similarities between these words:

The accuracy of Cross entropy and NCE are almost the same. But based on the word predictions it's quite evident that Cross entropy made more sensible predictions than NCE. For example, Predictions like last, final are antonyms of first. Whereas, original, best are synonyms of first. These are the predictions made by cross entropy model for 'first' which are related to it. Similarly, words like British, German, French are related to 'American'. Whereas, many of the words generated by nce model are noise and unrelated.

C) JUSTIFICATION BEHIND NCE METHOD LOSS

NCE Model allows to fit in models which are not normalized. NCE model is independent of vocabulary size , this is a very big advantage since you can calculate the same accuracy with very less computations in comparison to models like cross entropy. The logistic regression classifier is trained to be able to distinguish between the samples from the positive examples and negative examples (noise distribution).

The unigram distribution of training data is used as the noise distribution here. We assume that noise samples are k times more frequent that data samples.

$$P^h(D=1|w,\theta) = \frac{P_\theta^h(w)}{P_\theta^h(w) + kP_n(w)} = \sigma\left(\Delta s_\theta(w,h)\right),$$

Where, σ is the logistic function and $\Delta s_\theta(w,h) = s_\theta(w,h) - \log(kP_n(w))$ is the difference between probability distribution of positive samples target words and the negative ones. Here we are using a non-normalized model and obtaining a perfectly normalized one

This is the final loss equation that we get,

J(θ, Batch) = ∑ (wo..wc)∈Batch − [log P r(D = 1, wo|wc) + ∑ x∈V^k log(1 − P r(D = 1, wx|wc))
where,

P r(D = 1, wo|wc) = σ (s(wo, wc) − log [kP r(wo)])
P r(D = 1, wx|wc) = σ (s(wx, wc) − log [kP r(wx)])
σ(x) = 1/(1 + e^−x) and s(wo, wc) = (uc^T uo) + bo

For each target word vector, we apply sigmoid function on it and calculate the log Function of its difference with the k times the unigram probability. Bias is added to this and it keeps adjusting in each step. For each input word, we take k input words, compute their sum and add that to the probability distribution of each word and apply log on it.
Hence, the sum over k noise samples instead of a sum over the entire vocabulary makes NCE training time linear in the number of noise samples and independent of the vocabulary size.

$$\frac{\partial}{\partial\theta} J^{h,w}(\theta) = \left(1 - \sigma\left(\Delta s_\theta(w,h)\right)\right) \frac{\partial}{\partial\theta} \log P_\theta^h(w) - \sum_{i=1}^{k} \left[\sigma\left(\Delta s_\theta(x_i,h)\right) \frac{\partial}{\partial\theta} \log P_\theta^h(x_i)\right].$$

As we increase the number of noise samples k, the above mentioned estimate approaches the likelihood gradient of the normalized model, allowing us to trade off computation cost against the accuracy.

Reference:

https://www.cs.toronto.edu/~amnih/papers/wordreps.pdf