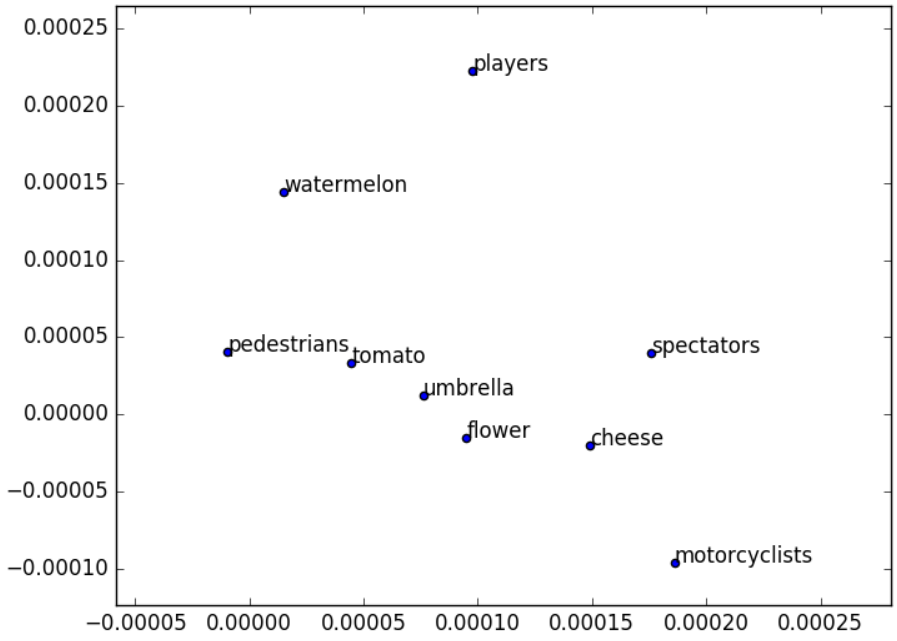
CSC321 Assignment 2

Zikun Chen
1001117882

February 22, 2017

Part 1 - Visualize Word Embedding



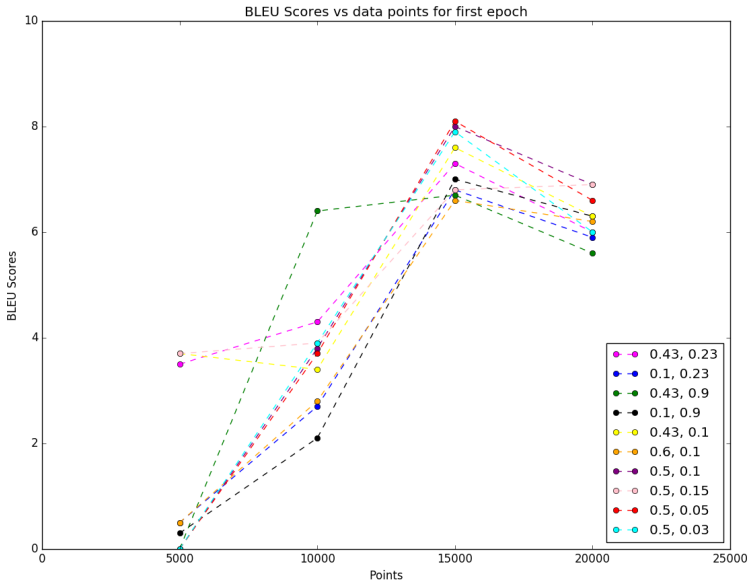
In this subset of words shown above, I chose specifically some plurals that describe people and other singular nouns describing objects like food. I found that the second group form cluster in the middle, and the other nouns describing people like motorcyclists fall on the outside of the parameter. This shows that the representation vectors of similar words are closer to each other in the high dimensional space.

Cosine Similarity:

motorcyclists & vehicles: 0.371195102169
motorcyclists & pedestrians: 0.402447608851
motorcyclists & watermelon: -0.00990755672344
motorcyclists & elephant: 0.0887606891413
football & soccer: 0.737345239006
big & large: 0.748236941148
orange & banana: 0.493180895151
Pedestrians & sleeping: -0.0249091618303

As we can see from the above cosine similarity computations, similar words have a higher cosine similarity value, meaning that their representation vectors are close in the high dimension word representation space. For example, {motorcyclists, pedestrians} are very much related from our intuition and {motorcyclists , elephant} have nothing to do with each other. Thus, the pairs get a higher and a lower similarity value respectively.

Part 4 - Train and Evaluate Model



Learning Rate and Momentum:

From the above graph, we can see that the red plot with learning rate of 0.5 and momentum of 0.05 work the best on the validation, getting a score of 8.1 at 15000 data points after 15 minutes of training.

This is a higher learning rate than the default 0.43 and a lower momentum

than the default 0.23. The reason why we want a very low momentum is that initially we have randomly initialized weights that have very bad predictive power, and we do not want them to influence the progress of our gradient descent. Therefore, as the coefficient in front of the previous change of weights, the momentum should be very low initially so that the learning rate dominates. And the learning rate determines how much of an influence the steepest downhill direction has on the update. If it is large at the start of the training, we will quickly adjust in the downhill direction, which will make the validation score better.

However, we do not want too large of a learning rate initially either. For example, comparing the the orange plot with learning rate 0.6 and the purple plot with learning rate 0.5, we can see that when the learning rate is too large, we get a bad validation score because we are taking too big of a step in the downhill direction so that we actually surpass the minimum to the other side on the error surface, causing the update to diverge.

Later on during the training, we should update the hyper-parameters so that the learning rate slows down and the momentum increases to deal with problems like narrow ravines at the bottom of the error surface.

In this particular model, the initial choices of 0.5 learning rate and 0.05 momentum works quite well on the validations.

The BLEU score of this model is 7.0 on the test set.

Beam Width:

Beam width allows the algorithm to only store the important words in memory at each level.When beam width is increased, there are more candidate words at each level to choose from, and more combination of words to store in the memory, some of which can be irrelevant to our goal of finding the most probable caption at the end. So we only want the highest few combinations and words at each level based on conditional probability. If we increase the beam width to some level (maybe the whole vocabulary), the algorithm can run out of memory before we reach the most probable sentence.

However, if we set the beam width to be too low, we can lose some hidden better choices. For example if the beam width is one, and we pick the highest probability word A in the first level, and second word B has highest probability conditioned on A. But there might be another word D in second level nothing being picked up that has a higher probability conditioned on the word C if C is the first word.

Therefore, there is a trade-off in choosing the right beam width between efficiency and generative power.

Captions:

Bad captions occur when the limited beam width does not result in picking up the highest word combinations. And sometimes a stop sign is picked

up as a high probability in the middle of the caption even though it should not have been stopped.

And good captions occur when the final word combinations after beam search successfully contain the most probable caption based on conditional probabilities, which achieved our goal of beam searching the most probable word combination without sacrificing too much of memory and run-time complexity.