Perceptron Self Handout

Aliya Deri

October 6, 2014

1 Problem

You are running a generative model (in this case, PM generation) that builds a model from training data and creates $k$best lists from test data. You want to rerank those $k$best lists using a feature-based discriminative method.

2 Set up

1. Set aside some dev data. (80-10-10 should work.)
2. Like before, build your model on training data. Create $k$best lists by running that model on dev and test data.

   If there are multiple models, as in this case, run each of them and produce your final $k$best list by taking the union of the different $k$best lists and keeping the different model scores as different feature weights.
3. Define and calculate some features that will apply to each entry in the $k$best lists.

   Some examples:

   (a) the model score(s)–if there are multiple models
   (b) language model score
   (c) percent of component words kept in the output portmanteau
   (d) whether the portmanteau is a component word
   (e) length of the portmanteau

   Take the log of probabilities, but bound very large values! It may also be good to normalize.
4. Have some function that tells you how good the portmanteau hypothesis is. This can also be binary; either it is the best or it's not. In this case, I use edit distance. In any $k$best list, the best portmanteau is the one with the lowest edit distance–hopefully 0!

3 Variable Definition

This is adapted from: http://www.msr-waypoint.com/en-us/groups/speech/0100141.pdf.

- n component word pairs in the dev set
- n_i portmanteau hypotheses in each $k$best list
- $x_{i,j}$ is the j-th hypothesis portmanteau of the i-th dev set CW pair
- $x_{i,R}$ is the best hypothesis portmanteau for the ith CW pair.
- Similarly, $y_{i,j}$ refers to the test set
- D features
- Each portmanteau hypothesis has an associated feature value vector, $f_d(x_{i,j}) = [f_0(x_{i,j}), f_1(x_{i,j}..., f_D(x_{i,j}]$
- Similarly, there is a weight vector of length D, which we are learning.
- Since we are running the averaged perceptron, there is also a "sum of all weights seen so far" vector, which we call W. The average of this vector is what we use on test!
- We have discriminative function $g(x_{i,j}) = \sum_{k=0}^{D} w_k f_k(x_{i,j})$. That is, the weighted sum of all features. Higher is better.
- t is the number of iterations
- η is the learning step size.

4 Perceptron Algorithm

Set $w_0 = 1$ and all other weights w_{1toD} to 0.
Set all values of W to 0.
For $j = 1...t$ \\for all iterations
  For each $n_i, i = 1...n$ \\for all dev examples
    Choose the $x_{i,j}$ with the largest $g(x_{i,j})$ value
    (E.g., choose the hypothesis that the current weighted sum prefers).
    For each w_d
      $w_d = w_d + \eta(f_d(x_{i,R}) - f_d(x_{i,j}))$
      \\update based on the difference between gold and best reranked
      $W_d += w_d$ \\update the sum
$W_d = \frac{W_d}{(t \times n)}$ \\take the average

5 A few notes

Tomer Can scale the update based on the comparison function (e.g., edit distance). That is, if we have a large edit distance between the gold and the best reranked, we take a larger step.

Mathematically, if ϵ is the difference between the gold edit distance (the lowest in the list) and the best reranked edit distance, the update is:

$w_d = w_d + \eta * |\epsilon|(f_d(x_{i,R}) - f_d(x_{i,j}))$

Fei NLP people frequently make the mistake of using very few iterations! The number of iterations must be high, or the learned weights won't be high.

Aliya I use 1000 iterations (t) and .001 step size (η).

6 Post perceptron

- You can use the function $g(x)$, the learned weight vector W, and the feature functions to rerank the test set's $k$best lists.
- If this reranking happens in the middle of a pipeline of FSMs, you will need to rerank the $k$best lists by each hypothesis' g value, normalize the g values, and build a trie to create a FSA from the reranked weights.
- If you are reranking in the middle of a pipeline, how you normalize the g values can change how much the reranking affects the final output. Ways to increase the effect: Use higher exponent bases (10^g instead of e^g), or increase the exponentiation on the trie FSM itself.