CS224N PA4: Neural Networks for Named Entity Recognition

Daoying Lin (SUID 06090664) Patrick Manion (SUID)

1 System Design

1.1 Overview of System Implementation Details

1.2 Design Choices

2 Gradient

It can be shown that the expression for $\frac{\partial J(\theta)}{\partial L}$ is:

$$\frac{\partial J(\theta)}{\partial L} = W^T U^T (p_\theta - y) \odot tanh'(Wx + b^{(1)})$$

.

We also generalized the gradient expression for multiple layers of neural network, which is summarized next.

2.1 Gradient Derivation (With Extra Credit)

Let w^l_{jk} denote the weight for connecting the k^{th} neuron in the $(l-1)^{th}$ layer to the j^{th} neuron in the l^{th} layer; b^l_j denote the bias for the j^{th} neuron in the l^{th} layer; a^l_j denote the activation of the j^{th} neuron in the l^{th} layer; z^l_j denote the weighted input to the j^{th} neuron in the l^{th} layer; $h_l(.)$ denote the activation function for the weighted input $\mathbf{z}_l$. Note that $z^l_j = \sum_i w^l_{ji} a^{l-1}_i + b^l_j$ and $a^l_j = h_l(z^l_j)$. Let's define $\delta^l_j = \frac{\partial J}{\partial z^l_j}$, the error of neuron j in layer l. Then it can be easily derived that the following four equations are true for any backpropagation system:

$$\delta^L = \frac{\partial J}{\partial a^L} \odot h'_L(z^L) \tag{1a}$$

$$\delta^l = (W^{l+1})^T \delta^{l+1} \odot h'_l(z^l) \text{ for } l = 1, ..., L-1 \tag{1b}$$

$$\frac{\partial J}{\partial b^l_j} = \delta^l_j \text{ for } l = 1, ..., L \tag{1c}$$

$$\frac{\partial J}{\partial w^l_{jk}} = \delta^l_j (a^{l-1}_k)^T \text{ for } l = 1, ..., L \tag{1d}$$

where $h'_L(z^L) = p_\theta * (1 - p_\theta)$

For current system, we've three layers: input layer, hidden layer and output layer. The cost function is $J = -[ylna^L + (1-y)ln(1-a^L)]$. Using the above general system, we can obtain the following:

$$\delta^3 = p_\theta - y \tag{2a}$$

$$\delta^2 = (W^3)^T \delta^3 \odot tanh'(z^2) = U^T(p_\theta - y) \odot tanh'(Wx + b^{(1)}) \tag{2b}$$

$$\delta^1 = (W^2)^T \delta^2 \odot I'(x) = W^T \delta^2 = W^T U^T \delta^3 \odot tanh'(Wx + b^{(1)}) \tag{2c}$$

And

$$\frac{\partial J}{\partial U} = a^2 \delta^3 = tanh(Wx + b^{(1)})(p_\theta - y) \tag{3a}$$

$$\frac{\partial J}{\partial W} = a^1 \delta^2 = LU^T(p_\theta - y) \odot tanh'(Wx + b^{(1)}) \tag{3b}$$

$$\frac{\partial J}{\partial L} = a^0 \delta^1 = W^T U^T(p_\theta - y) \odot tanh'(Wx + b^{(1)}) \tag{3c}$$

$$\frac{\partial J}{\partial b^{(2)}} = \delta^3 = p_\theta - y \tag{3d}$$

$$\frac{\partial J}{\partial b^{(1)}} = \delta^2 = U^T(p_\theta - y) \odot tanh'(Wx + b^{(1)}) \tag{3e}$$

2.2 Gradient Check Results

3 Parameter Tuning

3.1 Learning Rate α

3.2 Max Iterations I

3.3 Regularization Value λ

3.4 Hidden Layer Size H

3.5 Window Size C

3.6 Fixed vs. Updated Word Vectors

3.7 Randomly Initialized vs. Pre-trained Word Vectors

4 Error Analysis

We analyzed the error by entity. For each entity, we presented two misclassification errors: False Positive, misclassification of non-PER word as PER and False Negative, misclassification of PER as any non-PER word. Here non-PER word includes LOC, ORG, MISC and O.

4.1 PER

- *False Positive* There are 129 cases where O is misclassified as PER. Among them, mostly are numbers (1, 2, 3, 53.98, 1,627, 1988, etc) and compound adjectives (ex-rebel, Lieutenant-Colonel, newly-signed, over-allotment, soft-spoken, etc.). These misclassification can be avoided if we could enforce some rules. For example, usually numbers won't be a person. There are 69 cases where ORG is misclassified as PER. Some examples are: Fed, Duke, Ford, Johnson, Kent, Jones, Lola, Magna. A lot of them are actually organization's that are named after their founders. These would be really hard for the algorithm to distinguish since those words are could be either cases.

  There are 38 cases where LOC is misclassified as PER. PATRICK.

  There are 25 cases where MISC is misclassified as PER. One very interesting example are the words "Michael" and "Collins". These two are actually human names. The reason the algorithm misclassified them is because there is a movie named "Michael Collins". For cases like this, it's going to be really hard to make the right prediction.

- *False Negative* The false negative is dominated by misclassifying PER as O, which are mostly non-standard English name. For example: Hondo, Inzamam-ul-Haq, Capelli, Djorkaeff, Hun, Wang, Donghai, Xiao, Sihanouk, etc. This type of error can be reduced by pre-training words on more general text that contains non-standard English name.

4.2 MISC

- *False Positive* The false positive is dominated by misclassification of O as MISC. Again, there are a lot of numbers (13, 40, 1988, etc) and compound adjectives (little-known, 88-year-old, army-backed, etc).

- *False Negative* The false negative is dominated by misclassifying MISC as O. A lot of them related to sports event like "ENGLISH COUNTY CHAMPIONSHIP", "U.S. Open Tennis Championship", "U.S. Amateur Championship", etc. While the algorithm often correctly classified other words as MISC but failed to classify championship correctly. This is a challenging problem since most of the time "championship" should be O. It's only referring to an entity when in phrase like this. Similarly, MISC is misclassified as LOC because usually those words are indeed location but when it appears in phrase like "the HONE KONG Open", "the Chicago PMI", "the Berlin Grand Prix", etc. These kinds of error may be reduced by sequence modeling or treating those phrases as if one word.

5 Extra Credit

5.1 Alternative Prepared Word Vectors

5.2 Deeper Neural Networks