lecture 3 preview

1. Classification review/introduction

- input data X : word, sentence, document…(d dimention vector)
- class Y : sentiment, named entities
- 벡터공간의 word vector들을 그룹화하는 learned line(=classifier)
- multiply Xi by estimated weight vector, that estimated weight vector will then go into a classification decision.
- softmax classifier - 2차원 평면에서는 선이지만, 사실은 d차원에 벡터공간에 존재하는 plane이다.
- minimize the log probability of the wrong class = maximize the probability of the correct class

<cross entropy bakground>

- p : true probability disribution
- q : computed model probability
- go through the classes - whats the probability of the true model?

2. Neural networks introduction

<traditional ML optimization>

- basic classifier : gradient update on loss function→ move around W→ moving the classifier(the line)

<neural network classifier>

- softmax(logistic regression) alone not very powerful
- softmax gives only linear decision boundaries

- neural network classifier ; non-linear classifier, complex decision boudaries
- word representations are also parameters of model→ change the representations of words to allow our classifiers to do better
- simulaneously changing the weight, representation of words. optimizing both.

<artificial neuron>

- a neuron can be a binary logistic regression unit
- neural network = running several logistic regessions at the same time
- minimize cross entropy loss

3. Named Enity Recognition

: 어떤 이름을 의미하는 단어를 보고 그 단어가 어떤 유형인지를 인식하는 것. 기업이름, 사람이름 등.

why might NER be hard?

- hard to work out boundaries of entity
- hard to know if something is an entity
- hard to know class of unknown/novel entity

4. Binary True vs corrupted word window classification

→build classifiers for language that work inside a context

- 한 단어임에도 맥락에 따라서 의미가 달라지기도 한다. ex) to sanction, to seed(auto-synonyms)

<window classification>

- idea : classify a word in its context window of neighboring words
- big vector of word window

<binary classification with unnormalized scores>

- assume we want to classify whether the center word is a Location

- go over all positions in a corpus, location entity in the center = return high score
- compute a window’s scoe with a layer neuron net : the middle layer(extra layer) learns non linear interactions btw the input word vectors

5. Marix calculus introduction

<computing gradients by hand>

- matrix calculus : fully vectorized gradients
    - much faster and more useful than non-vectorized gradients

<gradients>

- one output, many inputs? → jacobian matrix : 편미분의 조합을 나타낸 행렬

<jacobian matrix>

- mxn matrix : n input, m output