Lecture3: Word Window Classification, Neural Networks, and Matrix Calculus

Lecture Plan

- Course information update
- Classification review/introduction
- Neural networks introduction.
- Named Entity Recognition
- Binary tree vs. corrupted word window classification
- Matrix calculus introduction

1. Course information update

Course information update

2. Classification setup and notation

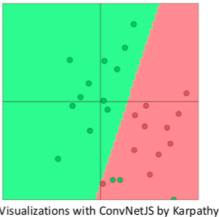
Generally we have a training dataset consisting of samples

$$\{x_i, y_i\}^N_{i=1}$$

- x_i are inputs, e.g. words (indices or vectors!), sentences, documents, etc.
 - Dimension d
- y_i are labels (one of C classes) we try to predict, for example:
 - classes: sentiment, named entities, buy/sell decision
 - other words
 - later: multi-word sequences
- 위와 같이 생긴 데이터를 Machine Learning/Deep Learning을 통해 학습시킨다.
- x_i 는 inputs (words, sentences, documents)을 의미하고 d 차원이다.
- y_i 는 labels 이고 우리가 예측하려고 하는 것이다.
 - o 여기서 label 은 sentiment, named entities, buy/sell decision등이 될 수 있다.

Classification intuition

- Training data: $\{x_i, y_i\}_{i=1}^N$
- Simple illustration case: -
 - Fixed 2D word vectors to classify
 - Using softmax/logistic regression
 - Linear decision boundary



Visualizations with ConvNetJS by Karpathy!

- **Traditional ML/Stats approach:** assume x_i are fixed, train (i.e., set) softmax/logistic regression weights $W \in \mathbb{R}^{C \times d}$ to determine a decision boundary (hyperplane) as in the picture
- **Method**: For each x, predict:

$$p(y|x) = \frac{\exp(W_y.x)}{\sum_{c=1}^{C} \exp(W_c.x)}$$

- p(y|x) 는 input이 주어졌을 때, output이 나올 확률이다.
- 전통적인 기법으로 sofmax/logistic regression을 사용할 수 있다.
- Linear하게 선을 그어 classification하는 방법이다.

4. Named Entity Recognition (NER)

NER 의 목적

- 문서의 특정 항목에 대한 언급 추적
- 질문 답변의 경우 답변은 일반적으로 이름이 지정된 항목이다.
- 얻고자하는 많은 정보들은 지정되 이름과 연관이 있다.
- 동일한 테크닉들이 다른 slot-filling classifcation(????)으로 확장 될 수 있다.

NER 이 Hard한 기술인 이유

- entity를 찾기가 어렵다.
- 특정단어가 이름인지 지명인지 알기가 어렵다.
- entity는 애매모호하고 context에 따라 달라진다.
 - o context에 따라 사람일 수도있고 특정 사물의 이름일 수도있고,,, 애매,,,하다.

5. Binary word window classification

In general, single words의 분류는 문맥상에서 애매하다.

동일한 표현이 반대의 의미로 인식되는 경우나 개체가 애매모호한 연결성을 가질 때...

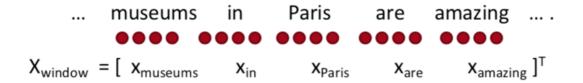
- Example : auto-antonyms:
 - "To sanction" can mean "to permit" or "to punish"
 - "To seed " can mean "to place seeds" or "to remove seeds"
- Example: resolving linking of ambiguous named entities
 - Paris -> Paris, France vs Paris Hilton vs Paris, Texas
 - Hathaway -> Bershire Hathaway vs Anne Hathaway

Window classification

- word classification을 할 때, 주위 단어들을 본다.
- EX)

Window classification: Softmax

- Train softmax classifier to classify a center word by taking concatenation of word vectors surrounding it in a window
- Example: Classify "Paris" in the context of this sentence with window length 2:



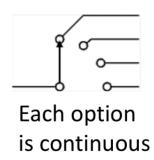
- Resulting vector $x_{window} = x \in R^{5d}$, a column vector!
- Idea: "Why didn't we make a big vector of a word window?"
- x_w 에는 5d만큼의 vector가 포함되어있다.
- window사이즈 벡터 내의 center인 paris가 장소인지 아닌지 classification
- ullet s = paris , s_c 는 score corrupt 즉, paris가 아닌 나머지 네개의 단어들의 score

The max-margin loss

- <u>Idea for training objective</u>: Make true window's score larger and corrupt window's score lower (until they're good enough)
- s = score(museums in Paris are amazing)
- s_c = score(Not all museums in Paris)
- Minimize

$$J = \max(0, 1 - s + s_c)$$

 This is not differentiable but it is continuous → we can use SGD.



36

- s = high score
- ullet s_c 는 corrupt할 것 근데 저 window 자체를 s 와 s_c로 두는건가?
- 일단 저 score(x)를 어떻게 구하는거징
- 일단 차이 1까지 즉, 일정한 변량 까지의 loss J는 괜찮다는의미,,,?
- 일단 보기에 SVM 처럼 보임