CS224N

Lecture 3: Word Window Classification, Neural Networks, and Matrix Calculus

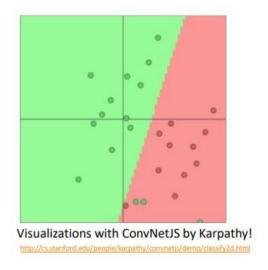
1. Classification

_

$$\{x_{i}, y_{i}\}_{i=1}^{N}$$

training dataset을 1부터 N까지 inputs과 outputs에 대해 training dataset을 가지고 있다. NLP에서 input은 단어나, 문장, 파일을 의미하며, output은 words나 다른 것일 수도 있다.

전통적인 Classification은 softmax나 logistic regression을 이용해서 class를 구분하는 linear boundary를 결정했다.



<Softmax classifier>

$$p(y|x) = \frac{\exp(f_y)}{\sum_{c=1}^{C} \exp(f_c)} = \operatorname{softmax}(f_y)$$

Training with softmax and cross-entropy loss

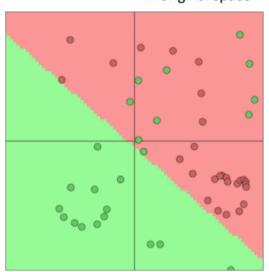
$$-\log p(y|x) = -\log \left(\frac{\exp(f_y)}{\sum_{c=1}^{C} \exp(f_c)}\right)$$

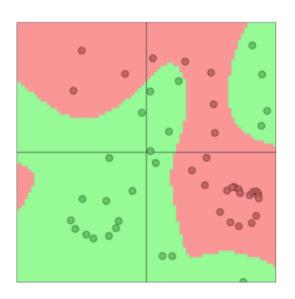
값을 학습할 때, 올바른 y값을 예측하도록 확률을 극대화하거나, negative log값을 최소화하도록 학습을 하게 된다.

2. Neural networks

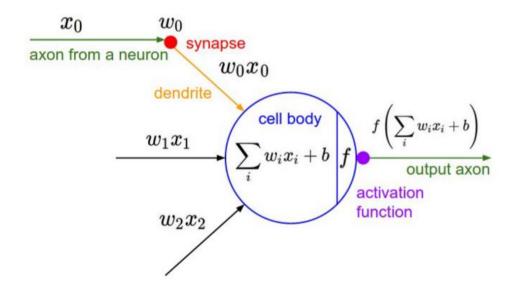
- decision boundary

In original space





non-linear boundary(오른쪽 그림)를 설정함으로써 분류의 정확도를 더 높일 수 있다.



- 3. Window Classification
- Named Entity Recognition (NER)

글에서 특정 카테고리에 속하는 단어가 언급되는지 추정한다.

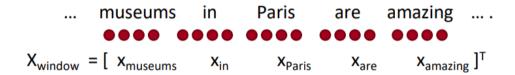
(카테고리 예시: org, loc, per 등등)

분류기를 실행함으로써 클래스를 할당한다.

그러나, 고유명사의 경우 분류의 기준이 모호할 수 있다는 문제점과, 문맥의 정확한 파악이 여전히 어렵다는 점을 고려하면 NER의 정확도는 떨어진다.

Window classification은 중심단어와 주변단어를 함께 분류 문제에 활용하는 방법이다.

• Example: Classify "Paris" in the context of this sentence with window length 2:



- Resulting vector $x_{window} = x \in R^{5d}$, a column vector!
- Softmax (Logistic)

$$s = U^T f(Wx + b) \qquad score(x) = U^T a \in \mathbb{R}$$

$$x \in \mathbb{R}^{20 \times 1}, W \in \mathbb{R}^{8 \times 20}, U \in \mathbb{R}^{8 \times 1}$$

$$s = U^T a$$

$$a = f(z)$$

$$z = Wx + b$$

$$\mathbf{x}_{\mathsf{window}} = [\mathbf{x}_{\mathsf{museums}}, \mathbf{x}_{\mathsf{in}}, \mathbf{x}_{\mathsf{park}}, \mathbf{x}_{\mathsf{are}}, \mathbf{x}_{\mathsf{areaning}}]$$

- Max-margin loss
- 5. Matrix calculus
- Gradient descent algorithm
- Jacobian matrix