cs224n

Lecture 3

목표

다 이해해버릴테다!

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)}$$

???

Stochastic gradients with word vectors!

- But in each window, we only have at most $2m + 1$ words, so $\nabla_\theta J_t(\theta)$ is very sparse!

$$\nabla_\theta J_t(\theta) = \begin{bmatrix} 0 \\ \vdots \\ \nabla_{v_{like}} \\ \vdots \\ 0 \\ \nabla_{u_I} \\ \vdots \\ \nabla_{u_{learning}} \\ \vdots \end{bmatrix} \in \mathbb{R}^{2dV}$$

???

cs224n

Lecture 3

cs224n

Lecture 2..

현재까지 공부한 것

용래형 블로그
Cs224n - Lecture 2, Lecture3
Gradient
여러 구글링

현재까지 이해한 내용(?)

One hot encoding
Sparse, dense vector
Word2vec-SG, CBOW
Training methods-Hierarchical softmax,
Negative sampling
Negative log likelihood!!!!!!!
등등….

정리!!!!

- 특정 단어를 가르키는 One hot vector에 W(dense-representation)를 product하면 그 단어의 dense vector가 나옴.
- 구한 vector에 W의 Transpose를 product하면 다른 단어들과의 내적값들로 이루어진 vector가 나옴.
- 구한 vector를 softmax를 사용하여 확률로 반환하면 이 확률이 context word와 같이 쓰일 확률임.
- 같이 쓰이는 단어의 확률을 높이도록 parameter(단어들의 dense vector)을 조정하여야 함.
- 학습방법은 Grandient Descent를 이용해 Cost Fucntion줄이도록!

궁금한 내용

Center word의 왼쪽, 오른쪽 이런식으로 여러가지의 확률 분포를 갖는 게 아니라,
context word
에 대한 오직 하나의 확률 분포를 갖는다

궁금한 내용

U_o 가 context word vectors 이면
여러 개의 word를 어떻게 표현될까?

궁금한 내용

Note: Every word has two vectors! Makes it simpler!

- 모델에서의 모든 parameter의 set을 하나의 긴 vector인θ로 정의한다.
- skip-gram model에서는 d차원의 dimension을 가지는 v개의 word vectors의 matrix일 것이다.
- 이때 matrix의 길이가 $2dV$인 이유는 하나의 word vector가 center word로 쓰일 수도 context word로 쓰일 수도 있기 때문이다.

objective function의 값을 최소화하기 위해 gradient를 이용한다. 이때 partial derivatives를 하는데 다음과 같은 과정을 거친다. $\frac{\partial}{\partial v_c} log \frac{exp(u_o^T v_c)}{\sum_{w=1}^{V} exp(u_w^T v_c)} = \frac{\partial}{\partial v_c}(logexp(u_o^T v_c) - log \sum_{w=1}^{V} exp(u_w^T v_c)) \cdots$

$= u_o - \sum_{x=1}^{V} p(x|c) u_x$

- 이때 u_o는 실제 output context word vector이고 $p(x|c)$는 softmax값이다. 각 단어의 softmax값으로 weighted sum*을 하고 있는 것이다. 즉, softmax값이 크다면 해당 단어가 context로 올 확률이 크므로 예측한 context vector의 값이 거의 그대로 반영될 것이고, softmax값이 작다면 반대이므로 거의 반영되지 않을 것이다.
- 예를 들면 $p(dog|bark)$=1 이라고 하면 u_{dog}는 그대로 합에 반영될 것이고 $p(cat|bark)$= 0 이라고 하면 u_{cat}은 합에 거의 반영되지 않을 것이다.

  *모두 균일하게 더하는 게 아니라, 각각의 값에 가중치를 줘서 중요한 것은 더 크게 반영되도록 하는 방법이다.
- u_x에 대해 좀 더 설명하자면, u_x는 model이 예측한 context vector이다.

궁금한 내용

Gradient Descent

- To minimize $J(\theta)$ over the full batch (the entire training data) would require us to compute gradients for all windows

- Updates would be for each element of θ :

$$\theta_j^{new} = \theta_j^{old} - \alpha \frac{\partial}{\partial \theta_j^{old}} J(\theta)$$

- With step size α
- In matrix notation for all parameters:

$$\theta^{new} = \theta^{old} - \alpha \frac{\partial}{\partial \theta^{old}} J(\theta)$$

$$\theta^{new} = \theta^{old} - \alpha \nabla_\theta J(\theta)$$

Stan

궁금한 내용

Stochastic gradients with word vectors!

- But in each window, we only have at most $2m + 1$ words, so $\nabla_\theta J_t(\theta)$ is very sparse!

$$\nabla_\theta J_t(\theta) = \begin{bmatrix} 0 \\ \vdots \\ \nabla_{v_{like}} \\ \vdots \\ 0 \\ \nabla_{u_I} \\ \vdots \\ \nabla_{u_{learning}} \\ \vdots \end{bmatrix} \in \mathbb{R}^{2dV}$$

돌아오는 주 할 것들..

- Github 작성법
- Lecture 3
- 모두의 딥러닝
  등등...

목표

대부분 이해...?