Week 1 : A Neural Probabilistic Language Model

120220210 고재현

2023년 3월 28일

1 Introduction

사람이 문장을 구성하는 단계를 생각해 보자. 중심이 되는 단어를 생각하고, 그 단어들을 조합하여 문장을 구성한다. 또는, 완성된 문장에서 특정 요소를 교체하여 문장을 완성하기도 한다. 이러한 언어 시퀀스의 구성 방식을 컴퓨터가 수행할 수 있는 반복 가능한 형태(Algorithmic)로 만드려면 어떻게 해야 할까?

2 related works

2.1 Probabilistic Language models

인간이 언어를 학습하는 과정을 문장의 요소 간, 혹은 단어의 요소 간의 관계를 확률적으로 표현하는 방법을 배우는 것이라고 한다면, 이러한 확률 관계를 잘 모델링 할 수 있다면 인간의 학습 과정을 재현할 수 있을 것이다. 이러한 관점에서 출발한 것이 바로 통계적 언어 모델(Probabilistic language modeling) 이다. 이는 문자, 단어, 문장 등의 언어 토큰 시퀀스가 주어졌을 때, 다음에 나올 토큰의 확률을 예측하는 기법이다. 수업 시간에는 주로 단어 단위의 시퀀스를 다루었으나, 목적에 따라 시퀀스의 단위는 달라질 수 있다. 이러한 확률 모델들 중에 가장 유명하고 간단한 형태인 N-gram model 을 다루어 본다.[1]

2.2 N-gram models

n-gram 모델은 통계적 언어 모델링에서 가장 기본적인 방법 중 하나로, 연속된 n 개의 단어를 하나의 단위로 취급하여 각 단어가 다음 단어로 등장할 확률을 추정하는 모델이다.

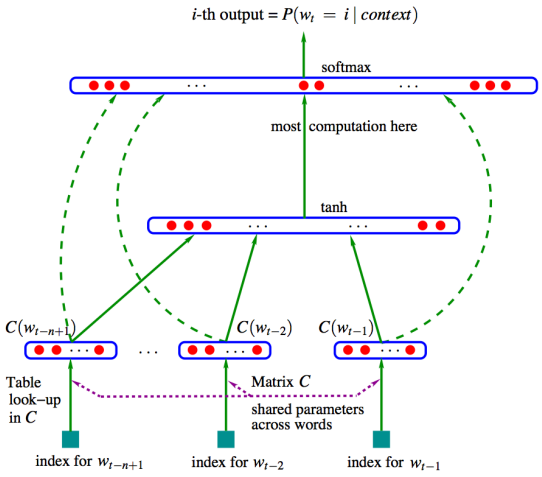
$$P(w_n|w_{n-1}, w_{n-2}, ..., w_1) = P(w_n|w_{n-1}, w_{n-2}, ..., w_{n-n_{gram}+1}) \tag{1}$$

예를 들어 bigram model을 생각하자. The cat sat on the mat 이라는 문장이 주어졌을 때, bigram model은 문장을 $P(The), P(cat|The), P(sat|cat), P(on|sat), P(the|on), P(mat|the)$ 로 분리된 조건부확률의 chain으로 보고, 각각의 확률을 corpus에서 빈도수를 통해 미리 계산해둔 뒤 이를 꺼내어 문장의 생성 확률을 계산한다. 이를 위해서는 말뭉치 내의 모든 단어의 빈도를 구하여 조건부 확률표를 구성해야 한다. 또한, 긴 범위의 문맥을 고려하기 어렵다는 단점이 있다.

[1] 논문에서 또한 다루었다.

3 Neural Probabilistic Language Model

Neural Probabilistic Language Model[1]은 N-gram model을 neural network로 구현한 모델이다.



〈그림 1〉 NPLM Architecture

3.1 forward propagation

먼저, 말뭉치 내의 각각의 단어를 one-hot encoding하여 벡터로 표현한다. 이를 $w_1, w_2, ..., w_n$ 이라고 하자. 이제 단어 두 개가 주어졌다고 하자. 예를 들어 '나는', '학교에' 이다. 여기에서 '간다' 라는 one-hot vector를 추론하는 것이 모델의 목표이다. 이를 수식으로 표현하면 다음과 같다.

$$P(w_t|w_{t-1}, w_{t-2}, ..., w_{t-n+1}) = \frac{\exp(y_{w_t})}{\sum_{i=1}^{n} \exp(y_i)} \tag{2}$$

단어들의 관계를 표현하기 위해 one-hot vector들을 열백터로 변환하는 행렬 C 를 정의한다. 이를 통해 $w_1, w_2, ..., w_n$ 사이의 관계를 반영하여 $Cw_1, Cw_2, ..., Cw_n$ 으로 표현할 수 있다. 이제 각각의 변환된 단어를 이어붙여 하나의 벡터로 표현하고, 이를 x_t 라고 하자. 이어붙이는 이유는 단어들의 정보를 잃어버리지 않고, 단일 백터로 만들어 2-layer neural network를 통과시키기 위함이다. 그리고 2-layer network를 통해 출력 단어 y를 예측한다. 모델은 단순한 2-layer perceptron이다.

$$y_{wt} = b + U \cdot \tanh(d + Hx_t) \tag{3}$$

출력을 softmax 함수를 통과시켜 확률로 변환한다.

3.2 bacpropagation

forward propagation의 출력과 실제 단어의 임베딩 값을 비교(negative log likelihood loss)하여 역전파를 통해 임베딩 변환 행렬 C 및 네트워크 파라미터를 학습한다.

3.3 result

결과는 perplexcity 측면에서 기존 언어 모델보다 강점을 보여주고 있다. 이 논문에 출판될 당시에는 연산능력 측면에서 한계가 있었으나, 현재는 GPU를 이용하여 재현하면 금새 구현 가능할 것이다.

	n	c	h	m	direct	mix	train.	valid.	test.
MLP1	5		50	60	yes	no	182	284	268
MLP2	5		50	60	yes	yes		275	257
MLP3	5		0	60	yes	no	201	327	310
MLP4	5		0	60	yes	yes		286	272
MLP5	5		50	30	yes	no	209	296	279
MLP6	5		50	30	yes	yes		273	259
MLP7	3		50	30	yes	no	210	309	293
MLP8	3		50	30	yes	yes		284	270
MLP9	5		100	30	no	no	175	280	276
MLP10	5		100	30	no	yes		265	**252**
Del. Int.	3						31	352	336
Kneser-Ney back-off	3							334	323
Kneser-Ney back-off	4							332	321
Kneser-Ney back-off	5							332	321
class-based back-off	3	150						348	334
class-based back-off	3	200						354	340
class-based back-off	3	500						326	**312**
class-based back-off	3	1000						335	319
class-based back-off	3	2000						343	326
class-based back-off	4	500						327	312
class-based back-off	5	500						327	312

Table 1: Comparative results on the Brown corpus. The deleted interpolation trigram has a test perplexity that is 33% above that of the neural network with the lowest validation perplexity. The difference is 24% in the case of the best n-gram (a class-based model with 500 word classes). n : order of the model. c : number of word classes in class-based n-grams. h : number of hidden units. m : number of word features for MLPs, number of classes for class-based n-grams. *direct*: whether there are direct connections from word features to outputs. *mix*: whether the output probabilities of the neural network are mixed with the output of the trigram (with a weight of 0.5 on each). The last three columns give perplexity on the training, validation and test sets.

〈그림 2〉 NPLM result

참고 문헌

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.