

RNN_기본 (Recurrent Neural Network)

2018년 12월 5일 수요일 오후 4:48

1. RNN의 원리와 사용 분야

RNN에서 R은 **Recurrent**라는 단어로 반복적인, 되풀이되는 등의 뜻입니다.

RNN은 **반복적인 데이터 - 순차적인 데이터(Sequential data)**를 학습하는데 특화되어 발전한 인공지능의 한 방식입니다.



그렇다면 어떤 데이터들이 RNN을 적용할 수 있을까요?

=> Language

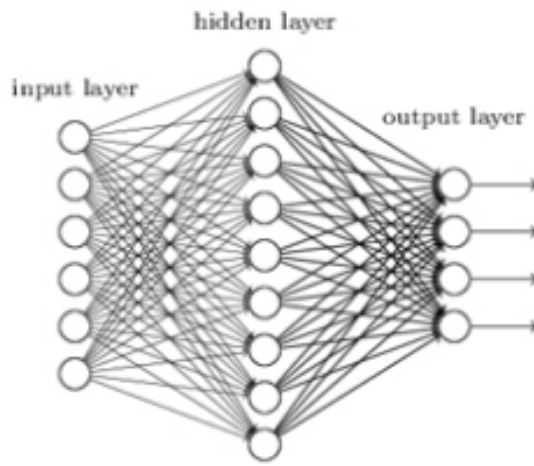
대표적으로 언어(Language)입니다. 단어 다음에 단어가 올 것 임이 확실한 데이터이므로 이를 이용해 음성인식, 단어의 의미 판단, 대화 등등을 할 수 있습니다.

(자연어 처리—Natural Language Processing)

또한 영상, 소리 또한 순차적인 데이터 이므로 동영상 분류, 음악 장르 분류등을 할 수 있습니다.

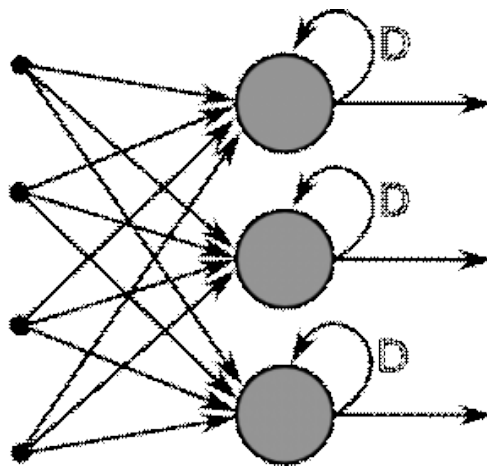
** RNN의 구조

기존의 인공지능망은 이와 같이 각 층의 뉴런이 연결되어 있는 구조입니다.



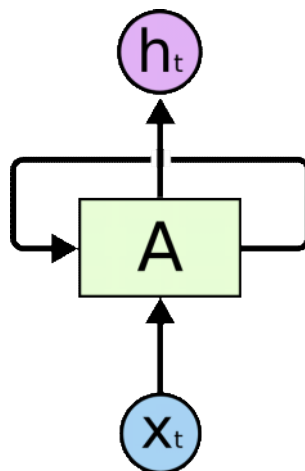
[기존의 인공신경망]

이에 추가적으로 RNN은 아래와 같은 방식을 사용합니다.



이렇게 과거 자신의 정보(가중치)를 기억하고 이를 학습에 반영합니다.

위의 layer를 한개의 box로 취급하여 단순화 시키면

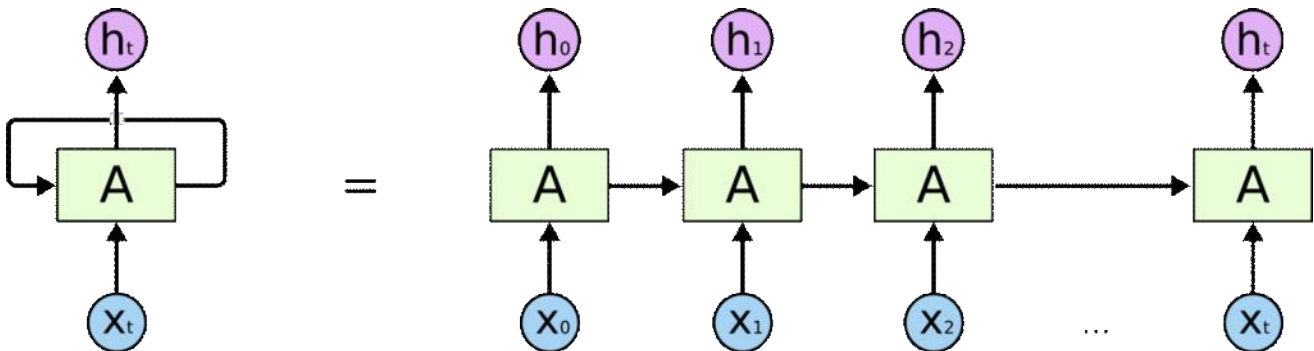


이와 같이 표현할 수 있습니다. 이런 RNN의 이전 작업을 현재 작업과 연결할 수 있다는 것이 큰 의미를 갖습니다.

인간은 판단을 할 때 지식을 기반으로 하지만 상황에 맞게 생각합니다. 대화를 할 때 문맥을 이해하고 말하는 것이 바로 그 예입니다. 이렇게 RNN은 상황에 맞는 결과를 낼 수 있다는 의미를 갖습니다.

** 학습 방법

인공 신경망과 다르게 RNN은 순환 구조이므로 hidden layer의 데이터를 저장하고 있어서 좀더 직관적으로 아래와 같이 layer가 펼쳐져있다고 생각할 수 도 있습니다.



하나의 순환은 여러개로 펼쳐져있다고 생각할 수 있습니다.

일반적인 인공신경망과 비슷하게 Gradient Descent와 backpropagation을 이용해 학습을 하는데 시간의 흐름에 따른 작업이기 때문에 backpropagation을 확장한 BPTT(Back-Propagation Through Time)을 사용해서 학습을 합니다.

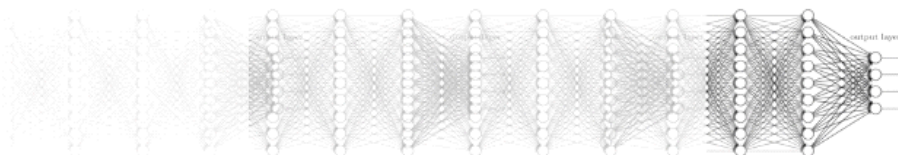
시간을 거슬러 올라가면서 backpropagation이 적용되는 구조이고 이것으로 인해 문제가 발생합니다.

바로 **Vanishing Gradients Problem** 입니다.

RNN이 시간을 거슬러 올라가면서 학습을 하는데 과거로 올라가면 올라갈수록 gradient 값이 계산이 잘 되지 않습니다. 그 이유는 gradient 계산이 곱셈 연산으로 이루어져 있기 때문입니다.

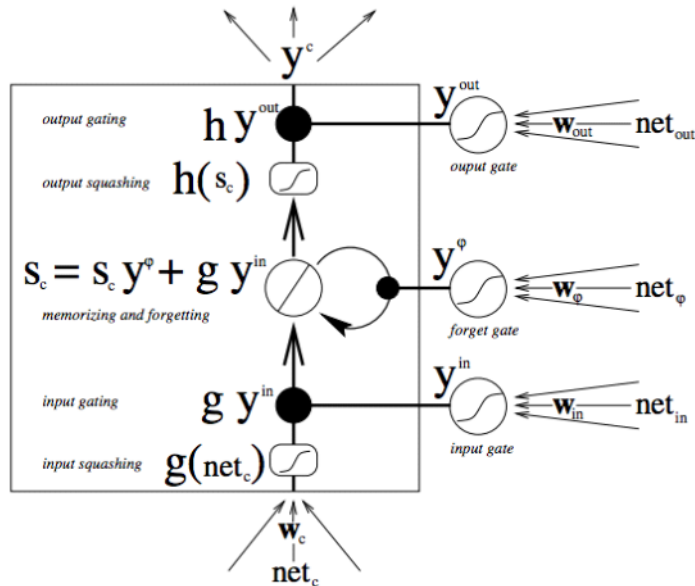
1보다 큰 값들을 계속해서 곱하면 발산하겠지만 이것은 최대 값을 지정해주면 해결됩니다. 1보다 작은 값들을 계속해서 곱하면 0으로 수렴해 사라져 버립니다. gradient가 사라져 버리는 것은 의미있는 값을 전달할 수 없다는 것이므로 문제가 됩니다.

Vanishing gradient (NN winter2: 1986-2006)



이렇게 점점 학습의 의미가 사라집니다. 긴 기간의 의미를 파악하지 못하고 짧은 기간 만이 유의미 해지므로 기억력이 좋지 못한 모델이 됩니다.

이를 해결하기 위해 등장한 **LSTM(Long-Short-Term Memory Units)** 은 엄청난 성능을 보여줍니다.



LSTM의 구조. (어려워 보인다)

이것은 매우 복잡한 원리로 작동하지만 큰 의미만 정리하면 입력, 과거값, 출력에 각각 가중치를 두고 학습하는 것입니다.
(입력/망각/출력 게이트)

이들의 연산은 곱셈이 아닌 더하기 연산으로 구성되어 있으므로 Vanishing Gradient Problem을 피해갈 수 있습니다.

실제로 매우 향상된 성능을 보여준다고 합니다. 최근 Google 번역의 성능이 굉장히 좋아졌는데 이는 이런 RNN을 도입했기 때문
입니다.