chapter 6 - 게이트가 추가된 RNN

- LSTM이나 GRU에는 '게이트'라는 구조가 더해져 있는데, 이 게이트 덕분에 시계열 데이터의 장기 의존 관계를 학습할 수 있습니다.
- 특히 LSTM의 구조를 시간을 들여 차분히 살펴보고, 이 구조가 '장기 기억'을 가능하게 하는 매커니즘을 이해해봅니다.

6.1 RNN의 문제점

6.1.1 RNN 복습

- RNN 계층은 시계열 데이터인 x_t를 입력하면 h_t를 출력합니다. 이 h_t는 RNN 계층의 [은닉 상태]라고 하여, 과거 정보를 저장합니다.

6.1.2 기울기 소실 또는 기울기 폭발

- [그림 6-4]에서와 같이 정답 레이블이 'Tom'임을 학습할 때 중요한 것이 바로 RNN 계층의 존재입니다. RNN 계층이 과거 방향으로 '의미 있는 기울기'를 전달함으로써 시간 방향의 의존 관계를 학습할 수 있는 것이죠. 이때 기울기는 (원래대로라면) 학습해야 할 의미가 있는 정보가 들어 있고, 그것을 과거로 전달함으로써 장기 의존 관계를 학습합니다. 하지만 만약 이 기울기가 중간에 사그라들면 가중치 매개변수는 전혀 갱신되지 않게 됩니다. 즉, 장기 의존 관계를 학습할 수 없게 됩니다. 안타깝지만, 현재의 단순한 RNN 계층에서는 시간을 거슬러 올라갈수록 기울기가 작아지거나(기울기 소실) 혹은 커질 수 있으며(기울기 폭발), 대부분 둘 중 하나의 운명을 걷게 됩니다.

6.1.3 기울기 소실과 기울기 폭발의 원인

- [그림 6-8]에서 보듯 기울기의 크기는 시간에 비례해 지수적으로 증가함을 알 수 있습니다. 이것이 바로 [기울기 폭발]입니다. 이러한 기울기 폭발이 일어나면 결국 오버플로를 일으켜 NaN 같은 값을 발생시키죠. 따라서 신경망 학습을 제대로 수행할 수 없게 됩니다.
- [그림 6-9]에서 알 수 있듯이, 이번에는 기울기가 지수적으로 감소합니다. 이것이 [기울기 소실]입니다.
- 왜 이런 지수적인 변화가 일어났을까요? 물론 행렬 Wh를 T번 반복해서 '곱'했기 때문입니다. 만약 Wh가 스칼라라면 이야기는 단순해지는데, Wh가 1보다 크면 지수적으로 증가하고, 1보다 작으면 지수적으로 감소합니다.
- 그럼 Wh가 스칼라가 아니라 행렬이라면 어떨까요? 이 경우, 행렬의 '특잇값'이 척도가 됩니다. 행렬의 특잇값이란, 간단히 말하면 데이터가 얼마나 퍼져 있는지를 나타냅니다. 이 특잇값의 값이 1보다 큰지 여부를 보면 기울기 크기가 어떻게 번할지 예측할 수 있습니다.
- 특잇값의 최댓값이 1봐 크면 지수적으로 증가하고, 1보다 작으면 지수적으로 감소할 가능성이 높다고 예측할 수 있습니다. 그럴 가능성이 높을 뿐, 특잇값이 1보다 크다고 해서 기울기가 반드시 폭발하는 것은 아닙니다.

6.1.4 기울기 폭발 대책

- [기울기 폭발]의 대책으로는 전통적인 기법이 있습니다. 바로 [기울기 클리핑]이라는 기법이죠.
- [기울기 클리핑]은 신경망에서 사용되는 모든 매개변수에 대한 기울기를 하나로 처리하고, threshold라는 문턱값을 초과하면, 기울기의 L2 노름으로 나눠서 기울기를 수정하는 것입니다.

6.2 기울기 소실과 LSTM

- RNN 학습에서는 기울기 소실도 큰 문제입니다. 그리고 이 문제를 해결하려면 RNN 계층의 아키텍처를 근본부터 뜯어고쳐야 합니다. 여기서 등장하는 것이, 이번 장의 핵심 주제인 '게이트가 추가된 RNN'입니다. 대표적으론, LSTM과 GRU가 있습니다

6.2.1 LSTM의 인터페이스

- [그림 6-11]에서 보듯 LSTM 계층의 인터페이스에는 c라는 경로가 있다는 차이가 있습니다. 이 c를 기억 셀이라 하며, LSTM 전용의 기억 메커니즘입니다.
- 기억 셀의 특징은 데이터를 자기 자신으로만(LSTM 계층 내에서만) 주고받는다는 것입니다. 즉, LSTM 계층 내에서만 완결되고, 다른 계층으로는 출력하지 않습니다. 반면, LSTM의 은닉층 상태 h는 RNN 계층과 마찬가지로 다른 계층으로 (위쪽으로) 출력됩니다.
- 기억 셀 c는 외부에서는 보이지 않습니다.

6.2.2 LSTM 계층 조립하기

- [그림 6-12]처럼 현재의 기억 셀 c_i는 3개의 입력으로부터 '어떤 계산'을 수행하여 구할 수 있습니다. 여기서 핵심은 갱신된 c_i를 사용해 은닉 상태 h_t를 계산한다는 것입니다. 또한 이 계산은 $h_t = \tanh(c_i)$인데, 이는 c_i의 각 요소에 tanh 함수를 적용한다는 뜻입니다.
- [게이트]는 데이터의 흐름을 제어합니다.
- LSTM에서 사용하는 [게이트]는 1) 열기 / 닫기 뿐만 아니라, 2) 어느 정도 열지를 조절할 수 있습니다. 다시 말해, 다음 단계로 흘려보낼 물의 양을 제어합니다. '어느 정도'를 '열림 상태'라 부릅니다. [그림 6-14]처럼 게이트의 열림 상태틀 0.0 ~ 1.0 사이의 실수로 나타납니다.

6.2.3 output 게이트

- 이번 절에서는 $\tanh(c_i)$에 게이트를 적용하는 걸 생각해보죠. 즉, $\tanh(c_i)$의 각 원소에 대해 '그것이 다음 시각의 은닉 상태에 얼마나 중요한가'를 조정합니다. 한편, 이 게이트는 다음 은닉 상태 h_t의 출력을 담당하는 게이트이므로 output 게이트(출력 게이트)라고 합니다.
- [그림 6-15]와 같은 output 게이트에서 수행하는 [식 6.1]의 계산을 'σ'로 표기하겠습니다. 그리고 σ의 출력을 O라고 하면, h_t는 O와 $\tanh(c_i)$의 곱으로 계산됩니다. 여기서 말하는 '곱'이란 원소별 곱이며, 이것을 [아다마르 곱]이라고 합니다. 이상이 LSTM의 output 게이트입니다.
- 따라서 (주로) 게이트에서는 시그모이드 함수가, 실질적인 '정보'를 지니는 데이터에는 tanh 함수가 활성화 함수로 사용됩니다.

6.2.4 forget 게이트

- 그러면 c_{t-1}의 기억 중에서 불필요한 기억을 잊게 해주는 게이트를 추가해볼까요? 이를 [forget 게이트(망각 게이트)]라 부르도록 하죠. forget 게이트를 LSTM 계층에 추가하면 계산 그래프가 [그림 6-16]처럼 됩니다.

6.2.5 새로운 기억 셀

- forget 게이트를 거치면서 이전 시각의 기억 셀로부터 잊어야 할 기억이 삭제되었습니다. 그런데 이 상태로는 기억 셀이 잊는 것밖에 하지 못하겠네요.
- 그래서 새로 기억해야 할 정보를 기억셀에 추가해야 합니다. 그러기 위해서 [그림 6-17]과 같이 tanh 노드를 추가합니다.
- [그림 6-17]에서 보듯 tanh 노드가 계산한 결과가 이전 시각의 기억 셀 c_{t-1}에 더해집니다. 기억 셀에 새로운 '정보'가 추가된 것이죠. 이 tanh 노드는 '게이트'가 아니며, 새로운 '정보'를 기억 셀에 추가하는 것이 목적입니다. 따라서 활성화 함수로는 시그모이드 함수가 아닌 tanh 함수가 사용됩니다.
- 여기에서는 기억 셀에 추가하는 새로운 기억을 g로 표기했습니다. 이 g가 이전 시각의 기억 셀인 c_{t-1}에 더해짐으로써 새로운 기억이 생겨납니다.

6.2.6 input 게이트

- 마지막으로 [그림 6-17]의 g에 게이트를 하나 추가할 생각입니다. 여기에서 새롭게 추가하는 게이트를 input 게이트라고 하겠습니다.
- input 게이트는 g의 각 원소가 새로 추가되는 정보로써의 가치가 얼마나 큰지를 판단합니다. 새 정보를 무비판적으로 수용하는 게 아니라, 적절히 취사선택하는 것이 이 게이트의 역할입니다.
- 다른 관점에서 보면, input 게이트에 의해 가중된 정보가 새로 추가되는 셈입니다.
- input 게이트의 도출 결과를 g의 원소별 곱 결과를 기억 셀에 새로 추가합니다. 이상이 LSTM 안에서 이뤄지는 처리입니다.

6.2.7 LSTM의 기울기 흐름

- LSTM의 구조는 설명했습니다만, 이것이 어떤 원리로 기울기 소실을 없애주는 걸까요? 그 원리는 기억 셀 c의 역전파에 주목하면 보입니다.
- 앞에서 본 RNN의 역전파에서는 똑같은 가중치 행렬을 사용하여 '행렬 곱'을 반복했고, 그래서 기울기 소실이 일어났습니다. 반면, 이번 LSTM의 역전파에서는 '행렬 곱'이 아닌 '원소별 곱'이 이뤄지고, 매 시각 다른 게이트 값을 이용해 원소별 곱을 계산합니다. 이처럼 매번 새로운 게이트 값을 이용하므로 곱셈의 효과가 누적되지 않아 기울기 소실이 일어나지 않는 것입니다.
- [그림 6-19]의 'x' 노드의 계산은 forget 게이트가 제어합니다 (그리고 매 시각 다른 게이트 값을 출력합니다). 그리고 forget 게이트가 '잊어야 한다'고 판단한 기억 셀의 원소에 대해서는 그 기울기가 작아지는 것이죠. 한편, forget 게이트가 '잊어서는 안 된다'고 판단한 원소에 대해서는 그 기울기가 약화되지 않은 채로 과거 방향으로 전해집니다. 따라서 기억 셀의 기울기가 (오래 기억해야 할 정보일 경우) 소실 없이 전파되리라 기대할 수 있습니다.

- 따라서 기억 셀이 장기 의존 관계를 유지(학습)하리라 기대할 수 있습니다.

6.5 RNNLM 추가 개선

6.5.1 LSTM 계층 다층화

- RNNLM으로 정확한 모델을 만들고자 한다면 많은 경우 LSTM 계층을 깊게 쌓아(계층을 여러게 쌓아) 효과를 볼 수 있습니다.
- 그렇다면 몇 증이나 쌓아야 할까요? 물론 그건 하이퍼파라미터에 관한 문제입니다. 쌓는 층 수는 하이퍼파라미터이므로 처리할 문제의 복잡도나 준비된 학습 데이터의 양에 따라 적절하게 결정해야 하지요.

6.5.2 드롭아웃에 의한 과적합 억제

- LSTM을 겹겹이 쌓은 시스템은 종종 [과적합]을 일으킵니다. 더 나쁜 소식은, 불행하게도 RNN은 일반적인 피드포워드 신경망보다 쉽게 과적합을 일으키다는 것입니다.
- 앞에서 보앗듯이, 과적합을 억제하는 전통적인 방법은, 1) 훈련 데이터 양 늘리기와 2) 모델의 복잡도 줄이기가 있습니다. 그 외에는 모델의 복잡도에 패널티를 주는 [정규화]도 효과적입니다. 예컨대 L2 정규화는 가중치가 너무 커지면 패널티를 부과합니다.
- 또, 드롭아웃처럼 훈련 시 계층 내의 뉴런 몇 개를 무작위로 무시하고 학습하는 방법도 일종의 정규화라고 볼 수 있습니다.
- 드롭아웃 계층을 어디에 삽입해야 할까요?

  1) 첫 번째 후보는 [그림 6-32]와 같이 LSTM 계층의 시계열 방향으로 삽입하는 것입니다.

  : LSTM - Dropout - LSTM - Dropout - LSTM - ...

  : 하지만 정답을 먼저 알려드리자면, 이는 좋은 방법이 아닙니다.

  - RNN에서 시계열 방향으로 드롭아웃을 넣어버리면 (학습 시) 시간이 흐름에 따라 정보가 사라질 수 있습니다. 즉, 흐르는 시간에 비례해 드롭아웃에 의한 노이즈가 축적됩니다. 노이즈 축적을 고려하면, 시간축 방향으로의 드롭아웃은 그만두는 편이 좋을 것입니다. 그렇다면 [그림 6-33]처럼 드롭아웃 계층을 깊이 방향(상하 방향)으로 삽입하는 방안을 생각해보죠.

  - 이렇게 구성하면 시간 방향(좌우 방향)으로 아무리 진행해도 정보를 잃지 않습니다. 드롭아웃이 시간축과는 독립적으로 깊이 방향(상하 방향)에만 영향을 주는 것이죠.

- 지금까지 이야기한 것처럼, '일반적인 드롭아웃'은 시간 방향에는 적합하지 않습니다. 그러나 최근 연구에서는 RNN의 시간 방향 정규화를 목표로 하는 방법이 다양하게 제안되고 있습니다. 예컨대 문헌 [36]에서는 [변형 드롭아웃]을 제안했고, 시간 방향으로 적용하는 데 성공했습니다.
- [변형 드롭아웃]은 깊이 방향은 물론 시간 방향에도 이용할 수 있어서 언어 모델의 정확도를 한층 더 향상시킬 수 있습니다. 그 구조는 [그림 6-34]와 같은데, 같은 계층에 속한 드롭아웃들은 같은 [마스크]를 공유합니다. 여기서 말하는 [마스크]란 데이터의 [통과/차단]을 결정하는 **이진 형태의 무작위 패턴**을 가리킵니다.
- [그림 6-34]에서 보듯 같은 계층의 드롭아웃끼리 마스크를 공유함으로써 마스크가 '고정'됩니다. 그 결과 정보를 잃게 되는 방법도 '고정'되므로, 일반적인 드롭아웃 때와 달리 정ㅈ보가 지수적으로 손실되는 사태를 피할 수 있습니다.

6.5.3 가중치 공유

- 언어 모델을 개선하는 아주 간단한 트릭 중 [가중치 공유]가 있습니다. weight tryint을 직역하면 '가중치를 연결한다'이지만, 실질적으로는 [그림 6-35]에서 보듯 가중치를 공유하는 효과를 줍니다.
- [그림 6-35]처럼 Embedding 계층의 가중치와 Affine 계층의 가중치를 연결하는 기법이 가중치 기법입니다. 두 계층이 가중치를 공유함으로써 학습하는 매개변수 수가 크게 줄어드는 동시에 정확도도 향상되는 일석이조의 기술입니다.
- Embedding 계층의 가중치를 전치하여 Affine 계층의 가중치로 설정하기만 하면 됩니다.
- 게다가 매개변수 수가 줄어든다 함은 과적합이 억제되는 혜택으로 이어질 수 있습니다.

6.6 정리

- 단순한 RNN의 학습에서는 기울기 소실과 기울기 폭발이 문제가 된다.
- 기울기 폭발에는 기울기 클리핑, 기울기 소실에는 게이트가 추가된 RNN(LSTM과 GRU 등)이 효과적이다.
- LSTM에는 input 게이트, forget 게이트, output 게이트 등 3개의 게이트가 있다.
- 게이트에는 전용 가중치가 있으며, 시그모이드 함수를 사용하여 0.1 ~ 1.0 사이의 실수를 출력한다.
- 언어 모델 개선에는 LSTM 계층 다층화, 드롭아웃, 가중치 공유 등의 기법이 효과적이다.
- RNN의 정규화는 중요한 주제이며, 드롭아웃 기반의 다양한 기법이 제안되고 있다.