

RNN과 LSTM

연준모

1. 순환 신경망 RNN
2. RNN 기울기 소멸
3. RNN 기울기 폭주
4. LSTM
5. LSTM수식으로 열어보기
6. Numpy로 LSTM 구현하기
+ 부록 수식 미분해보기

1. 순환 신경망 RNN 특징?

순서가 있는 데이터, 즉 시퀀스 데이터를 처리하기 위한 RNN

만일, 다음 문장에 순서가 없다면?

A 가 B 를 죽였다.

A, B 중 누가 누굴 죽인건지 알 수가 없다

텍스트 처럼 순서가 중요한 경우에 사용한다.

기억을 통해서 순서를 알게 된다

기억 : Hidden State (h_t)

이 RNN Cell을 여러 번 돌린 후 다음 레이어로 데이터를 넘기는 것이 특징

1. 순환 신경망 RNN 특징?

기본 특성

시퀀스 데이터 처리:

순서가 있는 데이터(텍스트 등)를 처리에 특화된 신경망

은닉상태

이전 입력을 기억하는 역할. 현재 입력과 함께 처리

RNN Cell의 반복적 적용

동일한 파라미터로 여러 번 적용하여 시퀀스 전체 처리

활성화 함수

주로 tanh 또는 ReLU등으로, 입력과 이전 은닉 상태의 자동합에 적용

1. 순환 신경망 RNN 특징?

ONECLICK AI

작동원리

입력처리

시퀀스의 각 요소가 순차적으로 처리

상태 업데이트

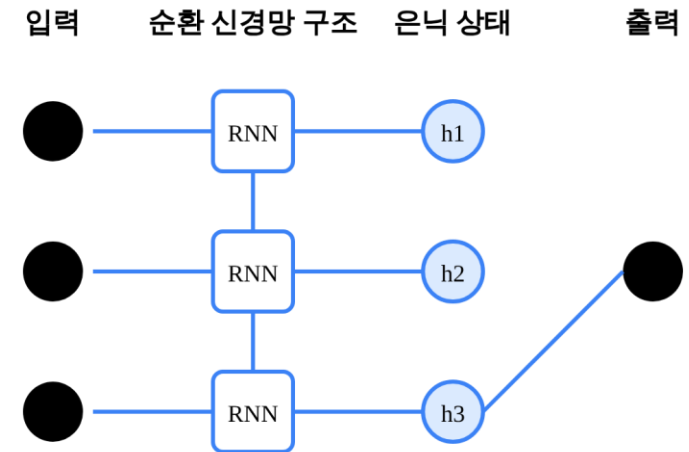
현재 입력과 이전 상태를 결합하여 새로운 상태 생성

출력 생성

내부 상태를 기반으로 출력 생성

학습 방식

모든 시점의 입력 - 출력 쌍을 동시에 학습



1. 순환 신경망 RNN RNN Cell?

ONECLICK AI

1. 입력 계층

현재 시점에서의 입력 x_t

현재 시간 단계인 t 에서 모델에 들어오는 데이터.

이전 시점의 은닉 상태 h_{t-1}

바로 이전 시간 단계인 $t-1$ 에서 계산된 결과값

이 은닉 상태는 과거의 정보를 요약해서 가지고 있으며, RNN의 기억 역할을 한다

쉬운 이해를 위해, t 의 수 만큼 RNN Cell이 반복 실행된다고 생각하자

1. 순환 신경망 RNN RNN Cell?

ONECLICK AI

2. 은닉 계층

가중치 행렬 W_x W_h

W_x 현재 입력인 x_t 에 곱해지는 가중치

W_h 이전 은닉 상태인 h_{t-1} 에 곱해지는 가중치

편향 b

현재 시점의 은닉상태 h_t

이전 은닉상태 h_{t-1} 와 현재 입력 x_t 을 조합하여 새로운 은닉상태 h_t 를 만든다

$$h_t = f(W_x \cdot x_t + W_h \cdot h_{t-1} + b)$$

여기서, f 는 활성화 함수

1. 순환 신경망 RNN RNN Cell?

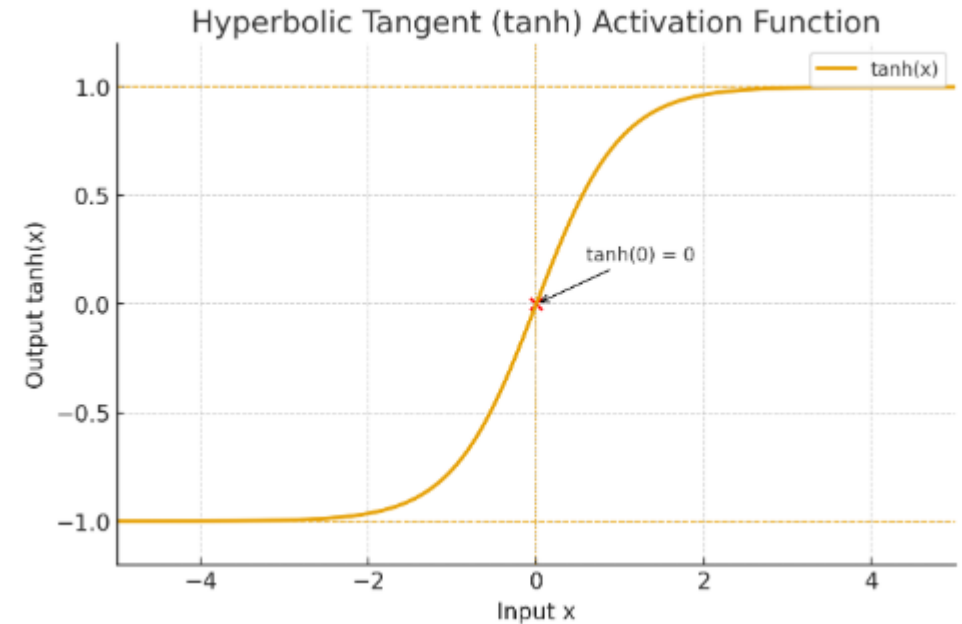
ONECLICK AI

3. 활성화 함수

하이퍼볼릭 탄젠트 tanh

계산값을 -1과 1 사이로 압축하여 다음 레이어로 전달
정보의 양 조절 + 기울기 소실문제를 완화

$$h_t = \tanh(W_x \cdot x_t + W_h \cdot h_{t-1} + b)$$



1. 순환 신경망 RNN RNN Cell?

ONECLICK AI

4. 출력계층

최종 출력 y_t

현재 시점의 은닉 상태 h_t 를 받아 최종출력값을 계산

출력층에도 별도의 가중치, 편향이 사용되며,

필요에 따라 softmax와 같은 활성화 함수를 거치기도 한다.

$$h_t = \tanh(W_x \cdot x_t + W_h \cdot h_{t-1} + b)$$

2. RNN 기울기 소멸

ONECLICK AI

1. 다음과 같이 생각하고 시작하자

Op	설명	입력 크기 (벡터 차원)	은닉 상태 크기	출력 크기
Input	시퀀스 데이터 ($t = 1, t = 2$)	2		
RNN Cell	순환 연산을 통해 은닉 상태 업데이트	2	2	
Dense	최종 은닉 상태를 출력 차원으로 변환		2	2
Softmax	최종 결과를 확률로 변환			2

간단한 수식 풀이를 위해 4개의 레이어로만 구성

간단하게 하기 위해, RNN Cell 출력층에선 추가적인 연산 없이 바로 출력하는걸로 하자

2. RNN 기울기 소멸

ONECLICK AI

1. 다음과 같이 생각하고 시작하자

파라미터 설정

RNN 가중치 및 편향

$$w_{xh} = [[0.1, 0.2], [0.3, 0.4]]$$

$$w_{hh} = [[0.1, 0.1], [0.1, 0.1]] \quad 0 \text{보다 작은 값}$$

$$b_h = [0.1, 0.1]$$

Dense 가중치(w_{dense}), 편향(b_{dense})

$$w_{hy} = [[0.2, 0.3], [0.4, 0.5]]$$

$$b_y = [0.1, 0.2]$$

입력 데이터

$$x_1 = [1, 2] \quad x_2 = [2, 3]$$

초기 은닉 상태

$$h_0 = [0, 0]$$

정답 레이블

$$y_{true} = [1, 0]$$

학습율 : 0.1

2. RNN 기울기 소멸 1. 순전파

ONECLICK AI

✓ 입력 데이터가 신경망의 각 층을 통과하며 예측값을 계산하는 과정

순전파 1단계. RNN Cell $t = 1$ 계산

$$h_t = \tanh(W_x \cdot x_t + W_h \cdot h_{t-1} + b)$$

$$\begin{aligned} h &= \tanh(W \cdot x + W \cdot h + b) \\ &= \tanh([0.1, 0.2], [0.3, 0.4]] \cdot [1, 2] + [[0.1, 0.1], [0.1, 0.1]] \cdot [0, 0] + [0.1, 0.1]) \\ &= \tanh([0.5, 1.1] + [0, 0] + [0.1, 0.1]) \\ &= \tanh([0.6, 1.2]) \approx [0.537, 0.834] \end{aligned}$$

RNN Cell $t = 1$ 최종출력 $h_1 = [0.537, 0.834]$

2. RNN 기울기 소멸 1. 순전파

ONECLICK AI

순전파 2단계. RNN Cell t = 2 계산

$$h_t = \tanh(W_x \cdot x_t + W_h \cdot h_{t-1} + b)$$

$$\begin{aligned} h &= \tanh(W \cdot x + W \cdot h + b) \\ &= \tanh([0.1, 0.2], [0.3, 0.4]] \cdot [2, 3] + [[0.1, 0.1], [0.1, 0.1]] \cdot [0.537, 0.834] + [0.1, 0.1]) \\ &= \tanh([0.8, 1.8] + [0.137, 0.137] + [0.1, 0.1]) \\ &= \tanh([1.037, 2.037]) \approx [0.778, 0.966] \end{aligned}$$

RNN Cell t = 2 최종출력 $h_2 = [0.778, 0.966]$

2. RNN 기울기 소멸 1. 순전파

ONECLICK AI

순전파 3단계. Dense Layer 계산

$$\begin{aligned} Z_y &= W_{hy} \cdot h_t + b_y \\ &= [[0.2, 0.3], [0.4, 0.5]] \cdot [0.778, 0.966] + [0.1, 0.2] \\ &= [0.445, 0.794] + [0.1, 0.2] = [0.545, 0.994] \end{aligned}$$

2. RNN 기울기 소멸 1. 순전파

ONECLICK AI

순전파 4단계. Softmax Layer 계산

$$\hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$$e^{0.054} \approx 1.725, \quad e^{0.994} \approx 2.702$$

$$\hat{y} = \left[\frac{1.725}{1.725 + 2.702}, \frac{2.702}{1.725 + 2.702} \right] \approx [0.39, 0.61]$$

2. RNN 기울기 소멸 1. 순전파

ONECLICK AI

순전파 5단계 Cross Entropy Loss 계산

오차율 계산

$$L = - \sum_i y_i \log(\hat{y}_i)$$

$$Loss = -(1 \cdot \log(0.39) + 0 \cdot \log(0.61)) \approx 0.94$$

오차가 매우 높은 상황이다.

2. RNN 기울기 소멸 2. 역전파

ONECLICK AI

역전파 1단계 Dense Layer 기울기 계산

소프트맥스 + 크로스 엔트로피 손실 기울기

$$\frac{\partial L}{\partial Z_y} = \hat{y} - y_{\text{true}} = [0.39, 0.61] - [1, 0] = [-0.61, 0.61]$$

h_2 로 전달될 기울기($t = 2$ 에서의 기울기)

$$\frac{\partial L}{\partial h_2} = W_{hy}^T \cdot \frac{\partial L}{\partial Z_y} \approx [0.122, 0.122]$$

2. RNN 기울기 소멸 2. 역전파

ONECLICK AI

역전파 2단계 RNN Cell ($t = 2 \rightarrow t = 1$)기울기 계산

$t = 2$ 시점의 기울기와 W_{hh}^T 를 이용해서 $t = 1$ 로 전달될 기울기를 구한다

$$\frac{\partial L}{\partial h_1} = W_{hh}^T \cdot \left(\frac{\partial L}{\partial h_2} \cdot (1 - h_2^2) \right) \approx [[0.1, 0.1], [0.1, 0.1]]^T \cdot [0.048, 0.008] \approx [0.0056, 0.0056]$$

순전파에서 $h_2 = [0.778, 0.966]$, \tanh 의 미분은 $1 - \tanh^2(x)$

따라서, 미분값은 $[0.395, 0.067]$

여기에 미래에서 온 기울기인 $\frac{\partial L}{\partial h_2}$ 를 곱하면,

$$[0.122, 0.122] \cdot [0.395, 0.067] = [0.048, 0.008]$$

$$\begin{bmatrix} 0.1 & 0.1 \\ 0.1 & 0.1 \end{bmatrix} \cdot \begin{bmatrix} 0.048 \\ 0.008 \end{bmatrix} = \begin{bmatrix} (0.1 \times 0.048) + (0.1 \times 0.008) \\ (0.1 \times 0.048) + (0.1 \times 0.008) \end{bmatrix} = \begin{bmatrix} 0.0056 \\ 0.0056 \end{bmatrix}$$

2. RNN 기울기 소멸 3. 결과

ONECLICK AI

가중치 업데이트 기울기 소실

$t = 2$ 에서의 기울기 : $[0.122, 0.122]$

$t = 1$ 에서의 기울기 : $[0.048, 0.008]$

약 22배 감소

1 Epoch로 한 시점만 거슬러도 기울기가 크게 소실되었다

이로 인해 초기 시점 $t = 1$ 의 정보는 가중치 업데이트에 거의 반영되지 않아 학습이 제대로 이루어지지 않는다

이걸 **장기 의존성 문제**라 한다

3. RNN 기울기 폭발 1. 순전파

ONECLICK AI

같은 구조에서, 가중치 값만 더 커졌다.

파라미터 설정

RNN 가중치 및 편향

$$w_{xh} = [[0.1, 0.2], [0.3, 0.4]]$$

$$w_{hh} = [[2.0, 2.0], [2.0, 2.0]] \quad 1 \text{ 보다 큰 값}$$

$$b_h = [0.1, 0.1]$$

Dense 가중치(w_{dense}), 편향(b_{dense})

$$w_{hy} = [[0.2, 0.3], [0.4, 0.5]]$$

$$b_y = [0.1, 0.2]$$

입력 데이터

$$x_1 = [1, 2] \quad x_2 = [2, 3]$$

초기 은닉 상태

$$h_0 = [0, 0]$$

정답 레이블

$$y_{true} = [1, 0]$$

학습율 : 0.1

이번에는, 기울기 증폭을 명확하게 보기 위해 선형이라 가정하고 tanh 없다 생각하고 가겠다.

3. RNN 기울기 폭발 1. 순전파

ONECLICK AI

한번 해 봤으니까 빠르게 계산해보자

순전파 1단계. RNN Cell $t = 1$ 계산

$$h_1 = W_{xh} \cdot x_1^T + W_{hh} \cdot h_0^T + b_h$$

$$W_{xh} \cdot x_1^T = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1.1 \end{bmatrix}$$

$$W_{hh} \cdot h_0^T = \begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$h_1 = \begin{bmatrix} 0.5 \\ 1.1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 1.2 \end{bmatrix}$$

3. RNN 기울기 폭발 1. 순전파

ONECLICK AI

순전파 2단계. RNN Cell $t = 2$ 계산

$$h_2 = W_{xh} \cdot x_2^T + W_{hh} \cdot h_1^T + b_h$$

$$W_{xh} \cdot x_2^T = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 1.8 \end{bmatrix}$$

$$W_{hh} \cdot h_1^T = \begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix} \cdot \begin{bmatrix} 0.6 \\ 1.2 \end{bmatrix} = \begin{bmatrix} 3.6 \\ 3.6 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} 0.8 \\ 1.8 \end{bmatrix} + \begin{bmatrix} 3.6 \\ 3.6 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 4.5 \\ 5.5 \end{bmatrix}$$

3. RNN 기울기 폭발 1. 순전파

ONECLICK AI

순전파 3단계. Dense Layer 계산

$$Z_y = W_{hy} \cdot h_t + b_y$$

$$[[0.2, 0.3], [0.4, 0.5]] \cdot [[4.5], [5.5]] + [0.1, 0.2] = [2.65, 4.75]$$

순전파 4단계. Softmax 계산

$$\hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$$\hat{y} = \left[\frac{e^{2.65}}{e^{2.65} + e^{4.75}}, \frac{e^{4.75}}{e^{2.65} + e^{4.75}} \right] = [0.109, 0.891]$$

3. RNN 기울기 폭발 1. 순전파

ONECLICK AI

순전파 5단계. Cross Entropy 계산

$$L = - \sum_i y_i \log(\hat{y}_i)$$

$$L = -(1 \cdot \log(0.109) + 0 \cdot \log(0.891)) = 2.216$$

3. RNN 기울기 폭발 2. 역전파

ONECLICK AI

역전파 1단계. Dense Layer 기울기 계산

Softmax 에 대한 기울기

$$\frac{\partial L}{\partial logits} = \hat{y} - y_{true} = [0.109, 0.891] - [1, 0] = [-0.891, 0.891]$$

Dense Layer에 대한 기울기

$$\frac{\partial L}{\partial h_2} = W_{hy}^T \cdot \left(\frac{\partial L}{\partial logits} \right)^T = [[0.2, 0.4], [0.3, 0.5]] \cdot [[-0.891], [0.891]] = [[0.178], [0.178]]$$

3. RNN 기울기 폭발 2. 역전파

ONECLICK AI

역전파 2단계. RNN Cell $t = 2$ 기울기 계산

$h = 2 \rightarrow h = 1$ 에 대한 역전파

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial h_2} \cdot W_{hh}^T = [0.178, 0.178] \cdot \begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix} = [0.712, 0.712]$$

$h = 1 \rightarrow h = 0$ 에 대한 역전파

$$\frac{\partial L}{\partial h_0} = \frac{\partial L}{\partial h_1} \cdot W_{hh}^T = [0.712, 0.712] \cdot \begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix} = [2.848, 2.848]$$

4배 차이로 기울기 폭발

왜 RNN은 치매가 올까?

하나의 경로(은닉상태)를 통해 과거의 모든 정보다 계속해서 변환되며 흘러가기 때문이다.
이 과정에서 정보와 기울기가 계속해서 변질, 희석된다

이를 극복하기 위한 LSTM

셀 상태(Cell State, C_t):

정보가 거의 변질되지 않는 장기기억 보관용 셀

게이트(Gates):

어떤 정보를 올리고, 내리고, 유지할 지 결정하는 시스템
(Forget, Input, Output Gate)

4. LSTM 1. 순전파

RNN Cell vs. LSTM Cell 비교

구분	RNN Cell (단순 RNN)	LSTM Cell
핵심 아이디어	이전 시점의 은닉 상태를 현재 입력과 함께 사용하여 다음 은닉 상태를 만든다.	별도의 '장기 기억' 장치(셀 상태)를 두고, 게이트를 이용해 정보를 선별적으로 기억하고 잊는다.
주요 상태 벡터	은닉 상태 (Hidden State, h_t)	셀 상태 (Cell State, C_t) + 은닉 상태 (Hidden State, h_t)
내부 구조	하나의 tanh 활성화 함수를 포함한 간단한 순환 계층	**3개의 게이트 (망각, 입력, 출력)** 와 하나의 셀 상태
정보 흐름	과거의 모든 정보가 하나의 은닉 상태에 압축/혼합되어 정보 손실 가능성이 높다.	셀 상태를 통해 중요한 정보는 거의 변하지 않고 장기적으로 전달되며, 게이트가 정보의 흐름을 정교하게 제어한다.
기울기 문제	기울기 소실/폭주 문제에 매우 취약하여 장기 (Long-term) 의존성 학습이 어렵다.	셀 상태의 덧셈(+) 연산과 망각 게이트를 통해 기울기 소실 문제를 효과적으로 해결 한다.
계산 복잡도	상대적으로 파라미터 수가 적고 계산이 간단하다.	게이트마다 별도의 가중치를 가져 파라미터 수가 약 4배 많고 계산이 복잡하다.
핵심 수식	$h_t = \tanh(W_{hh}h_{t-1} + W_{hx}x_t + b)$	$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$ $h_t = o_t \odot \tanh(C_t)$

1. 셀 상태

C_t

정보를 장기간 기억하는 짬통

과거의 정보를 거의 그대로 기억하면서 게이트를 통해
어떤 정보를 버리고 어떤 정보를 새로 추가할지 결정한다

2. Gate 망각 게이트(Forget Gate)

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

h_{t-1} 이전 은닉 상태

x_t 현재 입력

이전 은닉 상태와 현재입력을 받아 이전 셀 상태 c_{t-1} 의 정보 중 어떤 것을 잊을지 결정한다

2. Gate 입력 게이트(Input Gate)

새로 들어온 정보 h_{t-1}, x_t 를 셀 상태에 얼마나 반영할 지 결정한다

1. 어떤 값을 업데이트 할 지 결정 : 시그모이드 통해 저장할 정보의 비율 정한다

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

2. 새로운 후보 정보 생성 : tanh 함수를 통해 새로운 후보값 \tilde{C}_t 을 만든다

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

3. 이 둘을 곱해서 셀 상태에 추가할 새로운 정보를 최종적으로 결정한다.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

3. Gate 출력 게이트(Output Gate)

업데이트된 셀 상태에서 어떤 정보를 현재 시점의 은닉상태에 내보낼지 결정한다

1. 출력할 부분 결정 : 시그모이드 함수로 출력할 정보의 비율 정한다

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

2. 최종 은닉상태 계산 : tanh 로 값 조정하고, 출력 게이트의 결과와 곱하여 최종 은닉상태 만든다

$$h_t = o_t \odot \tanh(C_t)$$

이것이 바로 다음 시점으로 전달되고, 현재 시점의 출력으로도 사용된다

LSTM Cell 작동 흐름 보기

1. 망각 : 이전 셀 상태에서 잊어야 할 정보를 망각 게이트를 통해 버린다

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

2. 입력 및 저장 : 입력 게이트를 통해 현재 정보로부터 셀 상태에 저장할 새로운 정보 결정

$$i_t \odot \tilde{C}_t$$

3. 셀 상태 업데이트 : 1번과 2번의 결과를 더하여 현재 시점의 셀 상태를 업데이트

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

4. 출력 : 출력 게이트를 통해 업데이트 된 셀 상태에서 어떤 정보를 쫓아 현재의 은닉 상태로 내보낼지 결정

$$h_t = o_t \odot \tanh(C_t)$$

5. LSTM 수식으로 열어보기

ONECLICK AI

1. 다음과 같이 생각하고 시작하자

Op	설명	입력 크기 (벡터 차원)	셀 상태 크 기	은닉 상태 크 기	출력 크 기
Input	시퀀스 데이터 ($t = 1, t = 2$)	2			
LSTM Cell	게이트와 셀 상태를 통해 은닉 상태 업데이트	2	2	2	
Dense	최종 은닉 상태를 출력 차원으로 변환			2	2
Softmax	최종 결과를 확률로 변환				2

간단히 계산해보자

5. LSTM 수식으로 열어보기

1. 다음과 같이 생각하고 시작하자

LSTM은 3개의 게이트와 셀 상태 업데이트를 위해 총 4세트의 w 와 b 를 가진다. 편의를 위해, 전부 다음과 같다고 하자

$$w = [[0.1, 0.2], [0.3, 0.4]] \quad b = [0.1, 0.1]$$

망각 게이트 : W_{xf}, W_{hf}, b_f

입력 게이트 : W_{xi}, W_{hi}, b_i

출력 게이트 : W_{xo}, W_{ho}, b_o

셀 상태 후보 : W_{xc}, W_{hc}, b_c

Dense 파라미터

$$W_{hy} = [0.2 \ 0.3; 0.4 \ 0.5]$$

$$b_y = [0.1 \ 0.2]$$

입력 데이터	$x_1 = [1, 2], \quad x_2 = [2, 3]$	정답 레이블	$y_{\text{true}} = [1, 0]$
--------	------------------------------------	--------	----------------------------

초기 상태	$h_0 = [0, 0], \quad c_0 = [0, 0]$	학습률	$\eta = 0.1$
-------	------------------------------------	-----	--------------

5. LSTM 수식으로 열어보기 1. 순전파

ONECLICK AI

✓ 수식으로 열어보자

순전파 1단계. LSTM Cell t = 1 계산

$$\text{입력 : } x_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, h_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, c_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

1. Forget Gate : 과거의 정보를 얼마나 잊을까(Sigmoid)

$$z_{f1} = W_x x_1 + W_h h_0 + b = \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1.1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 1.2 \end{pmatrix}$$

$$f_1 = \sigma(z_{f1}) = \sigma \begin{pmatrix} 0.6 \\ 1.2 \end{pmatrix} = \begin{pmatrix} 0.6457 \\ 0.7685 \end{pmatrix}$$

2. Input Gate : 새로운 정보를 얼마나, 어떻게 추가할까?

$$z_{i1} = W_x x_1 + W_h h_0 + b = \begin{pmatrix} 0.6 \\ 1.2 \end{pmatrix}$$

$$i_1 = \sigma(z_{i1}) = \sigma \begin{pmatrix} 0.6 \\ 1.2 \end{pmatrix} = \begin{pmatrix} 0.6457 \\ 0.7685 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 1. 순전파

ONECLICK AI

✓ 수식으로 열어보자

순전파 1단계. LSTM Cell t = 1 계산

3. Cell State Update : 셀 상태 후보

$$z_{c1} = W_x x_1 + W_h h_0 + b = \begin{pmatrix} 0.6 \\ 1.2 \end{pmatrix}$$

$$\tilde{c}_1 = \tanh(z_{c1}) = \tanh \begin{pmatrix} 0.6 \\ 1.2 \end{pmatrix} = \begin{pmatrix} 0.5370 \\ 0.8337 \end{pmatrix}$$

4. Cell State Update : 새로운 정보를 얼마나, 어떻게 추가할까?

$$c_1 = f_1 \odot c_0 + i_1 \odot \tilde{c}_1 = \begin{pmatrix} 0.6457 \\ 0.7685 \end{pmatrix} \odot \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.6457 \\ 0.7685 \end{pmatrix} \odot \begin{pmatrix} 0.5370 \\ 0.8337 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.3468 \\ 0.6409 \end{pmatrix} = \begin{pmatrix} 0.3468 \\ 0.6409 \end{pmatrix}$$

5. Output Gate : 어떤 정보를 출력할지

$$z_{o1} = W_x x_1 + W_h h_0 + b = \begin{pmatrix} 0.6 \\ 1.2 \end{pmatrix}$$

$$o_1 = \sigma(z_{o1}) = \sigma \begin{pmatrix} 0.6 \\ 1.2 \end{pmatrix} = \begin{pmatrix} 0.6457 \\ 0.7685 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 1. 순전파

ONECLICK AI

✓ 수식으로 열어보자

순전파 1단계. LSTM Cell t = 1 계산

6. Hidden State : 은닉 상태 업데이트

$$h_1 = o_1 \odot \tanh(c_1) = \begin{pmatrix} 0.6457 \\ 0.7685 \end{pmatrix} \odot \tanh \begin{pmatrix} 0.3468 \\ 0.6409 \end{pmatrix} = \begin{pmatrix} 0.6457 \\ 0.7685 \end{pmatrix} \odot \begin{pmatrix} 0.3333 \\ 0.5657 \end{pmatrix} = \begin{pmatrix} 0.2152 \\ 0.4347 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 1. 순전파

ONECLICK AI

✓ 수식으로 열어보자

순전파 2단계. LSTM Cell t = 2 계산

입력 : $x_2 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, h_1 = \begin{pmatrix} 0.2152 \\ 0.4347 \end{pmatrix}, c_1 = \begin{pmatrix} 0.3468 \\ 0.6409 \end{pmatrix}$

1. Forget Gate : 과거의 정보를 얼마나 잊을까(Sigmoid)

$$\begin{aligned} z_{f2} &= W_x x_2 + W_h h_1 + b = \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix} \begin{pmatrix} 0.2152 \\ 0.4347 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} \\ &= \begin{pmatrix} 0.8 \\ 1.8 \end{pmatrix} + \begin{pmatrix} 0.1085 \\ 0.2384 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 1.0085 \\ 2.1384 \end{pmatrix} \end{aligned}$$

$$f_2 = \sigma(z_{f2}) = \sigma \begin{pmatrix} 1.0085 \\ 2.1384 \end{pmatrix} = \begin{pmatrix} 0.7327 \\ 0.8945 \end{pmatrix}$$

2. Input Gate : 새로운 정보를 얼마나, 어떻게 추가할까?

$$z_{i2} = z_{f2} = \begin{pmatrix} 1.0085 \\ 2.1384 \end{pmatrix}$$

$$i_2 = \sigma(z_{i2}) = \begin{pmatrix} 0.7327 \\ 0.8945 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 1. 순전파

ONECLICK AI

✓ 수식으로 열어보자

순전파 2단계. LSTM Cell t = 2 계산

3. Cell State Update : 셀 상태 후보

$$z_{c2} = z_{f2} = \begin{pmatrix} 1.0085 \\ 2.1384 \end{pmatrix}$$

$$\tilde{c}_2 = \tanh(z_{c2}) = \tanh \begin{pmatrix} 1.0085 \\ 2.1384 \end{pmatrix} = \begin{pmatrix} 0.7651 \\ 0.9729 \end{pmatrix}$$

4. Cell State Update : 새로운 정보를 얼마나, 어떻게 추가할까?

$$c_2 = f_2 \odot c_1 + i_2 \odot \tilde{c}_2 = \begin{pmatrix} 0.7327 \\ 0.8945 \end{pmatrix} \odot \begin{pmatrix} 0.3468 \\ 0.6409 \end{pmatrix} + \begin{pmatrix} 0.7327 \\ 0.8945 \end{pmatrix} \odot \begin{pmatrix} 0.7651 \\ 0.9729 \end{pmatrix} = \begin{pmatrix} 0.2541 \\ 0.5732 \end{pmatrix} + \begin{pmatrix} 0.5606 \\ 0.8711 \end{pmatrix} = \begin{pmatrix} 0.8147 \\ 1.4443 \end{pmatrix}$$

5. Output Gate : 어떤 정보를 출력할지

$$z_{o2} = z_{f2} = \begin{pmatrix} 1.0085 \\ 2.1384 \end{pmatrix}$$

$$o_2 = \sigma(z_{o2}) = \begin{pmatrix} 0.7327 \\ 0.8945 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 1. 순전파

ONECLICK AI

✓ 수식으로 열어보자

순전파 2단계. LSTM Cell t = 2 계산

6. Hidden State : 은닉 상태 업데이트

$$h_2 = o_2 \odot \tanh(c_2) = \begin{pmatrix} 0.7327 \\ 0.8945 \end{pmatrix} \odot \tanh \begin{pmatrix} 0.8147 \\ 1.4443 \end{pmatrix} = \begin{pmatrix} 0.7327 \\ 0.8945 \end{pmatrix} \odot \begin{pmatrix} 0.6723 \\ 0.8943 \end{pmatrix} = \begin{pmatrix} 0.4926 \\ 0.8001 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 1. 순전파

ONECLICK AI

✓ 수식으로 열어보자

순전파 3단계. Dense Layer 계산

가중치 + 편향 더해준다

$$\begin{aligned} y_{\text{logit}} &= W_{hy}h_2 + b_y = \begin{pmatrix} 0.2 & 0.3 \\ 0.4 & 0.5 \end{pmatrix} \begin{pmatrix} 0.4926 \\ 0.8001 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix} \\ &= \begin{pmatrix} 0.0985 + 0.2400 \\ 0.1970 + 0.4001 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix} = \begin{pmatrix} 0.3385 \\ 0.5971 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix} = \begin{pmatrix} 0.4385 \\ 0.7971 \end{pmatrix} \end{aligned}$$

순전파 4단계. softmax 계산

활성화 함수 계산

$$y_{\text{pred}} = \text{softmax}(y_{\text{logit}}) = \text{softmax} \begin{pmatrix} 0.4385 \\ 0.7971 \end{pmatrix}$$

$$e^{0.4385} \approx, e^{0.7971} \approx 2.2191$$

$$\text{Sum} = 1.5504 + 2.2191 = 3.7695$$

$$y_{\text{pred}} = \begin{pmatrix} 1.5504/3.7695 \\ 2.2191/3.7695 \end{pmatrix} = \begin{pmatrix} 0.4113 \\ 0.5887 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 1. 순전파

ONECLICK AI

✓ 수식으로 열어보자

순전파 5단계. Cross Entropy Layer 계산

손실량 구해준다

$$L = -(y_{true,1} \log(y_{pred,1}) + y_{true,2} \log(y_{pred,2}))$$

$$L = -(1 \cdot \log(0.4113) + 0 \cdot \log(0.5887)) = -(-0.8884) = 0.8884$$

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 1단계. Softmax + Cross Entropy Layer 기울기 계산

기울기 구해준다

1. 손실함수 L 을 소프트맥스의 입력값에 대해 미분 Loss w.r.t. Logits ($\frac{\partial L}{\partial y_{logit}}$)

$$\frac{\partial L}{\partial y_{logit}} = y_{pred} - y_{true} = \begin{pmatrix} 0.4113 \\ 0.5887 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -0.5887 \\ 0.5887 \end{pmatrix}$$

2. 손실함수 L 을 Dense Layer의 가중치에 대해 미분 Loss w.r.t. W_{hy} ($\frac{\partial L}{\partial W_{hy}}$)

$$\frac{\partial L}{\partial W_{hy}} = \frac{\partial L}{\partial y_{logit}} h_2^T = \begin{pmatrix} -0.5887 \\ 0.5887 \end{pmatrix} (0.4926 \quad 0.8001) = \begin{pmatrix} -0.2899 & -0.4710 \\ 0.2899 & 0.4710 \end{pmatrix}$$

3. 손실함수 L 을 Dense Layer의 편향에 대해 미분 Loss w.r.t. b_y ($\frac{\partial L}{\partial b_y}$)

$$\frac{\partial L}{\partial b_y} = \frac{\partial L}{\partial y_{logit}} = (-0.5887 \quad 0.5887)$$

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 1단계. Softmax + Cross Entropy Layer + DenseLayer 기울기 계산
기울기 구해준다

4. 손실함수 L 을 LSTM Cell의 히든값 h_2 에 대해 미분 Loss w.r.t. h_2 ($\frac{\partial L}{\partial h_2}$)

$$\frac{\partial L}{\partial h_2} = W_{hy}^T \frac{\partial L}{\partial y_{logit}} = \begin{pmatrix} 0.2 & 0.4 \\ 0.3 & 0.5 \end{pmatrix} \begin{pmatrix} -0.5887 \\ 0.5887 \end{pmatrix} = \begin{pmatrix} 0.1177 \\ 0.1177 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 2 기울기 계산

초기 기울기

Dense Layer로부터 역전파된 t = 2의 최종출력 h_2 에 대한 오차신호

$$\partial h_2 = \frac{\partial L}{\partial h_2} = \begin{pmatrix} 0.1177 \\ 0.1177 \end{pmatrix}, \partial c_{next} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad t = 2 \text{ 가 마지막 시점이므로, 다음 시점 } t = 3 \text{의 셀 상태는 없다 따라서 } 0, 0$$

1. 출력 게이트 기울기 ∂o_2 계산 최종 출력 h_2의 오차 ∂h_2 에 출력 게이트 o_2가 얼마나 기여했는지 계산

$$\partial o_2 = \partial h_2 \odot \tanh(c_2) = \begin{pmatrix} 0.1177 \\ 0.1177 \end{pmatrix} \odot \begin{pmatrix} 0.6723 \\ 0.8943 \end{pmatrix} = \begin{pmatrix} 0.0791 \\ 0.1053 \end{pmatrix} \quad h_2 \text{를 } o_2 \text{에 대해 미분하면 } \partial \tanh(c_2) \text{가 남는다 따라서 식이 저래 나온거다}$$

$$\partial z_{o_2} = \partial o_2 \odot o_2(1 - o_2) = \begin{pmatrix} 0.0791 \\ 0.1053 \end{pmatrix} \odot \begin{pmatrix} 0.7327(1 - 0.7327) \\ 0.8945(1 - 0.8945) \end{pmatrix} = \begin{pmatrix} 0.0156 \\ 0.0099 \end{pmatrix}$$

순전파 복기 : $h_2 = o_2 \odot \tanh(c_2)$

o_2는 시그모이드 결과. 시그모이드 미분하면 $\odot o_2(1 - o_2)$ 이다

∂o_2 에 미분 값 곱해 시그모이드 활성화 함수 1 ~ 7 전부 연쇄법칙 쓴다
이전 값인 z_{o_2} 그래디언트 구한다

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 2 기울기 계산

2. 셀 상태의 기울기 ∂c_2 계산

$$\partial c_2 = \partial c_{next} + \partial h_2 \odot o_2 \odot (1 - \tanh^2(c_2)) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.1177 \\ 0.1177 \end{pmatrix} \odot \begin{pmatrix} 0.7327 \\ 0.8945 \end{pmatrix} \odot (1 - (0.6723^2 0.8943^2)) = \begin{pmatrix} 0.0473 \\ 0.0210 \end{pmatrix}$$

셀 상태 c_2 에 대한 총 오차 신호 계산. 다음 셀 상태 c_3 (여기선 없음), 현재 출력 h_2 에 오차영향 미친다

순전파 복기 : $h_2 = o_2 \odot \tanh(c_2)$

t = 2 셀 상태에서, c_2 에 대한 그래디언트는 다음 두 그래디언트의 합이다

∂c_{next} (t = 3의 그래디언트) h_2 로부터 온 그래디언트(h_2 를 c_2 에 대해 미분하면 $o_2 \odot (1 - \tanh^2(c_2))$ 이다. 여기에, ∂h_2 를 곱하면 된다.

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 2 기울기 계산

3. 망각 게이트 기울기 ∂f_2 계산

$$\partial f_2 = \partial c_2 \odot c_1 = \begin{pmatrix} 0.0473 \\ 0.0210 \end{pmatrix} \odot \begin{pmatrix} 0.3468 \\ 0.6409 \end{pmatrix} = \begin{pmatrix} 0.0164 \\ 0.0135 \end{pmatrix}$$

$$\partial z_{f_2} = \partial f_2 \odot f_2(1 - f_2) = \begin{pmatrix} 0.0164 \\ 0.0135 \end{pmatrix} \odot \begin{pmatrix} 0.7327(1 - 0.7327) \\ 0.8945(1 - 0.8945) \end{pmatrix} = \begin{pmatrix} 0.0032 \\ 0.0013 \end{pmatrix}$$

셀 상태오차 ∂c_2 에 망각게이트 f_2 가 얼마나 기여했는지 계산

순전파 복기 : $c_2 = f_2 \odot c_1 + i_2 \odot \tilde{c}_2$

c_2 를 f_2 에 대해 미분하면 이전 셀 상태인 c_1 이 남는다. 따라서, ∂c_2 에 c_1 을 곱해서 f_2 의 그래디언트를 구한다.

$$\frac{\partial L}{\partial f_2} = \frac{\partial L}{\partial c_2} \cdot \frac{\partial c_2}{\partial f_2} = \partial c_2 \odot c_1$$

∂z_{f_2} 는, 출력 게이트와 마찬가지로, 시그모이드 함수의 역행이므로 미분값 $f_2(1 - f_2)$ 를 곱한다

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 2 기울기 계산

4. 입력 게이트 기울기 ∂i_2 계산

$$\partial i_2 = \partial c_2 \odot \tilde{c}_2 = \begin{pmatrix} 0.0473 \\ 0.0210 \end{pmatrix} \odot \begin{pmatrix} 0.7651 \\ 0.9729 \end{pmatrix} = \begin{pmatrix} 0.0362 \\ 0.0204 \end{pmatrix}$$

$$\partial z_{i2} = \partial i_2 \odot i_2(1 - i_2) = \begin{pmatrix} 0.0362 \\ 0.0204 \end{pmatrix} \odot \begin{pmatrix} 0.7327(1 - 0.7327) \\ 0.8945(1 - 0.8945) \end{pmatrix} = \begin{pmatrix} 0.0071 \\ 0.0019 \end{pmatrix}$$

순전파 복기 : $c_2 = f_2 \odot c_1 + i_2 \odot \tilde{c}_2$

c_2 를 i_2 에 대해 미분하면 셀 상태 후보인 \tilde{c}_2 가 남는다

따라서, $\text{partial} \backslash c_2$ 에 \tilde{c}_2 를 곱한다

$$\frac{\partial L}{\partial i_2} = \frac{\partial L}{\partial c_2} \cdot \frac{\partial c_2}{\partial i_2} = \partial c_2 \odot \tilde{c}_2$$

∂z_{i2} 마찬가지로, 시그모이드 미분식에 넣어준다

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 2 기울기 계산

5. 셀 상태의 후보 기울기 $\partial \tilde{c}_2$ 계산

$$\partial \tilde{c}_2 = \partial c_2 \odot i_2 = \begin{pmatrix} 0.0473 \\ 0.0210 \end{pmatrix} \odot \begin{pmatrix} 0.7327 \\ 0.8945 \end{pmatrix} = \begin{pmatrix} 0.0347 \\ 0.0188 \end{pmatrix}$$

$$\partial z_{c2} = \partial \tilde{c}_2 \odot (1 - \tilde{c}_2^2) = \begin{pmatrix} 0.0347 \\ 0.0188 \end{pmatrix} \odot (1 - (0.7651^2 \ 0.9729^2)) = \begin{pmatrix} 0.0144 \\ 0.0010 \end{pmatrix}$$

순전파 복기 : $c_2 = f_2 \odot c_1 + i_2 \odot \tilde{c}_2$

c_2 를 \tilde{c}_2 에 대해 미분하면 입력 게이트 i_2 가 남는다

따라서 ∂c_2 에 i_2 를 곱한다

$$\frac{\partial L}{\partial \tilde{c}_2} = \frac{\partial L}{\partial c_2} \cdot \frac{\partial c_2}{\partial \tilde{c}_2} = \partial c_2 \odot i_2$$

\tilde{c}_2 는 \tanh 함수의 결과이므로, \tanh 의 미분값인 $(1 - \tilde{c}_2^2)$ 를 곱한다

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 2 기울기 계산

6. t=2 시점 파라미터 그래디언트

$$\partial z_2 = \partial z_{f2} + \partial z_{i2} + \partial z_{c2} + \partial z_{o2} = \begin{pmatrix} 0.0403 \\ 0.0141 \end{pmatrix} \quad \text{4개 게이트의 오차는 결국 하나의 선형 변환에서 비롯되었으므로, 각 게이트의 활성화 함수 이전 기울기를 모두 더해 만든다}$$

$$\frac{\partial L}{\partial W_x(2)} = \partial z_2 x_2^T = \begin{pmatrix} 0.0403 \\ 0.0141 \end{pmatrix} \begin{pmatrix} 2 & 3 \end{pmatrix} = \begin{pmatrix} 0.0806 & 0.1209 \\ 0.0282 & 0.0423 \end{pmatrix} \quad \partial z_2 \text{를 가중치 } W_x \text{에 대해 미분하면 입력값 } x_2 \text{가 나온다. 차원을 맞추기 위해 } \partial z_2 \text{와 } x_2^T \text{를 외적한다}$$

$$\frac{\partial L}{\partial W_h(2)} = \partial z_2 h_1^T = \begin{pmatrix} 0.0403 \\ 0.0141 \end{pmatrix} \begin{pmatrix} 0.2152 & 0.4347 \end{pmatrix} = \begin{pmatrix} 0.0087 & 0.0175 \\ 0.0030 & 0.0061 \end{pmatrix} \quad W_h \text{는 이전 은닉 상태 } h_1 \text{이 곱해졌으므로, } \partial z_2 \text{와 } h_1^T \text{를 외적한다}$$

$$\frac{\partial L}{\partial b(2)} = \partial z_2 = \begin{pmatrix} 0.0403 \\ 0.0141 \end{pmatrix} \quad \text{편향 } b \text{는 그대로 더해졌으므로 그래디언트는 } \partial z_2 \text{와 같다}$$

순전파 복기 : $W_x x_2 + W_h h_1 + b$

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 2 기울기 계산

7. 이전 시점으로 전달할 기울기 t=2의 오차 ∂z_2 에 이전 은닉 상태 h_1 이 얼마나 기여했는지 계산

$$\partial h_1 = W_h^T \partial z_2 = \begin{pmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{pmatrix} \begin{pmatrix} 0.0403 \\ 0.0141 \end{pmatrix} = \begin{pmatrix} 0.0083 \\ 0.0137 \end{pmatrix}$$

$$\partial c_1 = \partial c_2 \odot f_2 = \begin{pmatrix} 0.0473 \\ 0.0210 \end{pmatrix} \odot \begin{pmatrix} 0.7327 \\ 0.8945 \end{pmatrix} = \begin{pmatrix} 0.0347 \\ 0.0188 \end{pmatrix}$$

순전파 시 h_1 은 W_h 와 곱해졌으므로, 역전파 시에는 W_h^T 와 ∂z_2 를 곱하여 ∂h_1 을 구한다
이것이 t=1의 h_{next} 기울기가 된다

t=2의 셀 상태 오차 ∂c_2 에 이전 셀 상태 c_1 이 얼마나 기여했는지 계산한다

순전파 시 c_1 은 망각 게이트 f_2 와 곱해졌으므로,
역전파 시에는 ∂c_2 와 f_2 를 곱하여 ∂c_1 을 구한다.
이것이 t=1의 c_{next} 기울기가 된다

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 3단계. LSTM Layer t = 1 기울기 계산

초기 기울기

$$\partial c_{next} = \partial c_1$$

1. 출력 게이트 기울기 ∂o_1 계산

$$\partial o_1 = \partial h_{next} \odot \tanh(c_1) = \begin{pmatrix} 0.0083 \\ 0.0137 \end{pmatrix} \odot \begin{pmatrix} 0.3333 \\ 0.5657 \end{pmatrix} = \begin{pmatrix} 0.0028 \\ 0.0077 \end{pmatrix}$$

$$\partial z_{o1} = \partial o_1 \odot o_1(1 - o_1) = \begin{pmatrix} 0.0028 \\ 0.0077 \end{pmatrix} \odot \begin{pmatrix} 0.6457(1 - 0.6457) \\ 0.7685(1 - 0.7685) \end{pmatrix} = \begin{pmatrix} 0.0006 \\ 0.0014 \end{pmatrix}$$

2. 셀 상태 기울기 partial c_1 계산

$$\begin{aligned} \partial c_1 &= \partial c_{next} + \partial h_{next} \odot o_1 \odot (1 - \tanh^2(c_1)) = \begin{pmatrix} 0.0347 \\ 0.0188 \end{pmatrix} + \begin{pmatrix} 0.0083 \\ 0.0137 \end{pmatrix} \odot \begin{pmatrix} 0.6457 \\ 0.7685 \end{pmatrix} \odot (1 - (0.3333^2 0.5657^2)) \\ &= \begin{pmatrix} 0.0400 \\ 0.0259 \end{pmatrix} \end{aligned}$$

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 1 기울기 계산

3. 망각 게이트 기울기 ∂f_1 계산

$$\partial f_1 = \partial c_1 \odot c_0 = \begin{pmatrix} 0.0400 \\ 0.0259 \end{pmatrix} \odot \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\partial z_{f1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

4. 입력 게이트 기울기 ∂i_1 계산

$$\partial i_1 = \partial c_1 \odot \tilde{c}_1 = \begin{pmatrix} 0.0400 \\ 0.0259 \end{pmatrix} \odot \begin{pmatrix} 0.5370 \\ 0.8337 \end{pmatrix} = \begin{pmatrix} 0.0215 \\ 0.0216 \end{pmatrix}$$

$$\partial z_{i1} = \partial i_1 \odot i_1(1 - i_1) = \begin{pmatrix} 0.0215 \\ 0.0216 \end{pmatrix} \odot \begin{pmatrix} 0.6457(1 - 0.6457) \\ 0.7685(1 - 0.7685) \end{pmatrix} = \begin{pmatrix} 0.0049 \\ 0.0038 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 1 기울기 계산

5. 셀 상태 후보 기울기 $\partial \tilde{c}_1$ 계산

$$\partial \tilde{c}_1 = \partial c_1 \odot i_1 = \begin{pmatrix} 0.0400 \\ 0.0259 \end{pmatrix} \odot \begin{pmatrix} 0.6457 \\ 0.7685 \end{pmatrix} = \begin{pmatrix} 0.0258 \\ 0.0199 \end{pmatrix}$$

$$\partial z_{c1} = \partial \tilde{c}_1 \odot (1 - \tilde{c}_1^2) = \begin{pmatrix} 0.0258 \\ 0.0199 \end{pmatrix} \odot (1 - (0.5370^2 0.8337^2)) = \begin{pmatrix} 0.0184 \\ 0.0061 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 1 기울기 계산

6. t=1 시점 파라미터 그래디언트

$$\partial z_1 = \partial z_{f1} + \partial z_{i1} + \partial z_{c1} + \partial z_{o1} = \begin{pmatrix} 0.0239 \\ 0.0113 \end{pmatrix}$$

$$\frac{\partial L}{\partial W_x(1)} = \partial z_1 x_1^T = \begin{pmatrix} 0.0239 \\ 0.0113 \end{pmatrix} \begin{pmatrix} 1 & 2 \end{pmatrix} = \begin{pmatrix} 0.0239 & 0.0478 \\ 0.0113 & 0.0226 \end{pmatrix}$$

$$\frac{\partial L}{\partial W_h(1)} = \partial z_1 h_0^T = \begin{pmatrix} 0.0239 \\ 0.0113 \end{pmatrix} \begin{pmatrix} 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\frac{\partial L}{\partial b(1)} = \partial z_1 = \begin{pmatrix} 0.0239 \\ 0.0113 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 2. 역전파

ONECLICK AI

✓ 수식으로 열어보자

역전파 2단계. LSTM Layer t = 1 기울기 계산

최종 기울기 총합

$$\frac{\partial L}{\partial W_x} = \frac{\partial L}{\partial W_x(1)} + \frac{\partial L}{\partial W_x(2)} = \begin{pmatrix} 0.0239 & 0.0478 \\ 0.0113 & 0.0226 \end{pmatrix} + \begin{pmatrix} 0.0806 & 0.1209 \\ 0.0282 & 0.0423 \end{pmatrix} = \begin{pmatrix} 0.1045 & 0.1687 \\ 0.0395 & 0.0649 \end{pmatrix}$$

$$\frac{\partial L}{\partial W_h} = \frac{\partial L}{\partial W_h(1)} + \frac{\partial L}{\partial W_h(2)} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0.0087 & 0.0175 \\ 0.0030 & 0.0061 \end{pmatrix} = \begin{pmatrix} 0.0087 & 0.0175 \\ 0.0030 & 0.0061 \end{pmatrix}$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial b(1)} + \frac{\partial L}{\partial b(2)} = \begin{pmatrix} 0.0239 \\ 0.0113 \end{pmatrix} + \begin{pmatrix} 0.0403 \\ 0.0141 \end{pmatrix} = \begin{pmatrix} 0.0642 \\ 0.0254 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 3. 가중치 업데이트

ONECLICK AI

✓ 수식으로 열어보자

가중치 업데이트

학습율 : 0.1 $W_{new} = W_{old} - \eta \cdot \frac{\partial L}{\partial W}$

W_x 업데이트

$$W_{x,new} = \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix} - 0.1 \cdot \begin{pmatrix} 0.1045 & 0.1687 \\ 0.0395 & 0.0649 \end{pmatrix} = \begin{pmatrix} 0.0896 & 0.1831 \\ 0.2961 & 0.3935 \end{pmatrix}$$

W_h 업데이트

$$W_{h,new} = \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix} - 0.1 \cdot \begin{pmatrix} 0.0087 & 0.0175 \\ 0.0030 & 0.0061 \end{pmatrix} = \begin{pmatrix} 0.0991 & 0.1982 \\ 0.2997 & 0.3994 \end{pmatrix}$$

b 업데이트

$$b_{new} = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} - 0.1 \cdot \begin{pmatrix} 0.0642 \\ 0.0254 \end{pmatrix} = \begin{pmatrix} 0.0936 \\ 0.0975 \end{pmatrix}$$

5. LSTM 수식으로 열어보기 3. 가중치 업데이트

ONECLICK AI

✓ 수식으로 열어보자

가중치 업데이트

학습율 : 0.1 $W_{new} = W_{old} - \eta \cdot \frac{\partial L}{\partial W}$

W_{hy} 업데이트

$$W_{hy,new} = \begin{pmatrix} 0.2 & 0.3 \\ 0.4 & 0.5 \end{pmatrix} - 0.1 \cdot \begin{pmatrix} -0.2899 & -0.4710 \\ 0.2899 & 0.4710 \end{pmatrix} = \begin{pmatrix} 0.2290 & 0.3471 \\ 0.3710 & 0.4529 \end{pmatrix}$$

b_y 업데이트

$$b_{y,new} = \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix} - 0.1 \cdot \begin{pmatrix} -0.5887 \\ 0.5887 \end{pmatrix} = \begin{pmatrix} 0.1589 \\ 0.1411 \end{pmatrix}$$

6. Numpy로 LSTM 구현하기

ONECLICK AI

https://huggingface.co/OneclickAI/LSTM_GUE_test_Model

https://huggingface.co/gihakkk/LSTM_test

위 두 코드를 통해서 LSTM을 직접 돌려보자

미분 : 가중치 편향 등이 최종 결과의 오차에 얼마나 큰 책임이 있는지 측정하기 위함

기울기가 크다 -> 최종 오차에 큰 영향을 미쳤다 -> 값을 많이 수정해야 한다

기울기가 작다 -> 최종 오차에 작은 영향을 미쳤다 -> 값을 조금만 수정해야 한다

더 확실하게 알고가기 위해 모든 구간에 대해 직접 미분 해 보자

오늘은 Dense Layer, Softmax + Cross Entropy 이렇게 2 경우에 대해서만 해 보자

이 둘은 죽을 때 까지 써먹는 조합이다 알아두면 좋다

1. softmax + cross Entropy

$$\hat{y}_i = \frac{e^{z_i}}{\sum_k e^{z_k}} \quad L = - \sum_i y_i \log(\hat{y}_i)$$

손실 L을 z_j 에 대해 미분하려면, 중간 다리인 \hat{y}_i 를 거쳐야 한다

$$\frac{\partial L}{\partial z_j} = \sum_i \frac{\partial L}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z_j}$$

$\partial L / \partial \hat{y}_i$ 계산 : Cross-Entropy 식을 y^i 에 대해 미분합니다.

$$\frac{\partial L}{\partial \hat{y}_i} = -y_i \frac{1}{\hat{y}_i}$$

+ 부록 수식 미분해보기 | 1. softmax + crossEntropy

ONECLICK AI

1. softmax + cross Entropy

$\partial \hat{y}_i / \partial z_j$ 계산 : Softmax 함수를 미분. 두 경우로 나뉜다.

$$\frac{\partial L}{\partial z_j} = \sum_i \frac{\partial L}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z_j}$$

$$\hat{y}_i = \frac{e^{z_i}}{\sum_k e^{z_k}}$$

Case 1 : $i = j \rightarrow$ 몫의 미분법 $(u/v)' = (u'v - uv')/v^2$ 사용

$$\frac{\partial \hat{y}_i}{\partial z_i} = \frac{(e^{z_i})'(\sum_k e^{z_k}) - (e^{z_i})(\sum_k e^{z_k})'}{(\sum_k e^{z_k})^2} = \frac{e^{z_i}(\sum_k e^{z_k}) - e^{z_i}(e^{z_i})}{(\sum_k e^{z_k})^2}$$

분자만 분리해서 정리하면,

$$\frac{\partial \hat{y}_i}{\partial z_i} = \frac{e^{z_i}}{\sum_k e^{z_k}} - \left(\frac{e^{z_i}}{\sum_k e^{z_k}} \right)^2 = \hat{y}_i - \hat{y}_i^2 = \hat{y}_i(1 - \hat{y}_i)$$

+ 부록 수식 미분해보기 1. softmax + crossEntropy

ONECLICK AI

1. softmax + cross Entropy

$\partial \hat{y}_i / \partial z_j$ 계산 : Softmax 함수를 미분. 두 경우로 나뉜다.

$$\frac{\partial L}{\partial z_j} = \sum_i \frac{\partial L}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z_j} \quad \hat{y}_i = \frac{e^{z_i}}{\sum_k e^{z_k}}$$

Case 2 : $i \neq j$ 가 아닐 때 -> e^{z_i} 는 상수 취급 한다

$$\frac{\partial \hat{y}_i}{\partial z_j} = \frac{0 \cdot (\sum_k e^{z_k}) - e^{z_i}(e^{z_j})}{(\sum_k e^{z_k})^2} = -\frac{e^{z_i}}{\sum_k e^{z_k}} \cdot \frac{e^{z_j}}{\sum_k e^{z_k}} = -\hat{y}_i \hat{y}_j$$

1. softmax + cross Entropy

이제, 다시 하나로 합쳐주면,

$$\frac{\partial L}{\partial z_j} = \left(\frac{\partial L}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial z_j} \right) + \sum_{i \neq j} \left(\frac{\partial L}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z_j} \right)$$

$$\frac{\partial L}{\partial z_j} = \left(-\frac{y_j}{\hat{y}_j} \cdot \hat{y}_j(1 - \hat{y}_j) \right) + \sum_{i \neq j} \left(-\frac{y_i}{\hat{y}_i} \cdot (-\hat{y}_i \hat{y}_j) \right)$$

$$\frac{\partial L}{\partial z_j} = -y_j(1 - \hat{y}_j) + \sum_{i \neq j} y_i \hat{y}_j = -y_j + y_j \hat{y}_j + \sum_{i \neq j} y_i \hat{y}_j$$

\hat{y}_j 로 묶어주면,

$$\frac{\partial L}{\partial z_j} = -y_j + \hat{y}_j \left(y_j + \sum_{i \neq j} y_i \right) = -y_j + \hat{y}_j \left(\sum_i y_i \right)$$

1. softmax + cross Entropy

y 는 원-핫 벡터이므로 모든 요소의 합 $\sum_i y_i = 1$ 이다

$$\frac{\partial L}{\partial z_j} = -y_j + \hat{y}_j(1) = \hat{y}_j - y_j$$

벡터 형태로 일반화 하면

$$\frac{\partial L}{\partial z_y} = \hat{y} - y_{true}$$

2. Dense Layer

$$Z_y = W_{hy} \cdot h_t + b_y \quad \text{원소 단위로 보면} \quad z_i = \sum_j W_{ij} h_j + b_i$$

하나의 가중치 W_{ij} 에 대해 편미분 하면

$$\frac{\partial z_i}{\partial W_{ij}} = \frac{\partial}{\partial W_{ij}} (W_{i1}h_1 + W_{i2}h_2 + \cdots + W_{ij}h_j + \cdots) = h_j$$

입력값인 h_j 만 남는다 ? 망했네

하지만, 우리가 알고싶은건 $\frac{\partial L}{\partial W_{ij}}$

2. Dense Layer

연쇄 법칙 사용하면 다음과 같다

$$\frac{\partial L}{\partial W_{ij}} = \sum_k \frac{\partial L}{\partial z_k} \cdot \frac{\partial z_k}{\partial W_{ij}}$$

$\frac{\partial z_k}{\partial W_{ij}}$ 이놈은, $k = i$ 일 때만 h_j 이고, 아닐때는 0이 된다

따라서, 시그마의 합은 $k = i$ 일 때의 항 하나만 남게 된다

$$\frac{\partial L}{\partial W_{ij}} = \frac{\partial L}{\partial z_i} \cdot \frac{\partial z_i}{\partial W_{ij}} = \frac{\partial L}{\partial z_i} \cdot h_j$$

$\frac{\partial L}{\partial W_{ij}}$ 의 i, j 위치값은 $\frac{\partial L}{\partial z_y}$ 벡터의 i 번째 값 * $h_{out,2}$ 벡터의 j 번째 값

$$z_i = \sum_j W_{ij} h_j + b_i \text{ 를 } \frac{\partial L}{\partial W_{ij}} \text{로 미분하는데 } ij \text{ 말고 다른데 들어온다? 다 0 되버림}$$

2. Dense Layer

	h_1	h_2	h_3	h_4
$\frac{\partial L}{\partial z_1}$	A	B	C	D
$\frac{\partial L}{\partial z_2}$	E	F	G	H
$\frac{\partial L}{\partial z_3}$	I	J	K	L
$\frac{\partial L}{\partial z_4}$	M	N	O	P

$$\frac{\partial L}{\partial W_{ij}} = \frac{\partial L}{\partial z_i} \cdot \frac{\partial z_i}{\partial W_{ij}} = \frac{\partial L}{\partial z_i} \cdot h_j$$

$$\frac{\partial L}{\partial W_{ij}} \text{의 } i, j \text{ 위치값은 } \frac{\partial L}{\partial z_y} \text{ 벡터의 } i \text{ 번째 값} * h_{out,2} \text{ 벡터의 } j \text{ 번째 값}$$

따라서, 앞선 식이 두 벡터의 외적 결과와 **모든 위치에서 정확히 일치**

하지만, 지금 식에선 둘다 세로벡터 이므로, 왼쪽 표처럼 하기 위해 전치 박아준다

$$\frac{\partial L}{\partial W_{hy}} = \left(\frac{\partial L}{\partial z_y} \right)^T \cdot h_{out,2}$$

만일, $h_{out,2}$ 에 전치를 박는다면?

2. Dense Layer

$\frac{\partial z_y}{\partial h_{out,2}}$ 가 최종적으로 W_{hy}^T 와의 곱으로 표현되는지 이해해 보자

$$z_i = \sum_k W_{ik} h_k + b_i$$

이번엔, z_i 를 하나의 특정 입력 h_j 에 대해 편미분 하자

$$\frac{\partial z_i}{\partial h_j} = \frac{\partial}{\partial h_j} (W_{i1}h_1 + \dots + W_{ij}h_j + \dots) = W_{ij}$$

마찬가지로, 연쇄법칙 같겨주면

$$\frac{\partial L}{\partial h_j} = \sum_i \frac{\partial L}{\partial z_i} \cdot \frac{\partial z_i}{\partial h_j} \quad \frac{\partial z_i}{\partial h_j} \text{ 이거 대입해 주면,}$$

$$\frac{\partial L}{\partial h_j} = \sum_i \frac{\partial L}{\partial z_i} \cdot W_{ij}$$

2. Dense Layer

$$\frac{\partial L}{\partial h_j} = \sum_i \frac{\partial L}{\partial z_i} \cdot W_{ij}$$

위 식을 보면, 행렬과 벡터의 곱셈 형태이다. $\frac{\partial L}{\partial z_i}$ 세로 벡터랑 W_{ij} 행렬의 곱이다

W_{ij} 정확히는 전치행렬인 W_{hy}^T 이므로, 다음이 성립된다

$$\frac{\partial L}{\partial h_{out,2}} = W_{hy}^T \cdot \left(\frac{\partial L}{\partial z_y} \right)^T$$

<-여기서 전치를 박아야 차원 크기가 맞으면서 행렬 곱셈이 가능해 진다

+ 중요한 잡지식

Tensorflow, pytorch 등 모든 딥러닝 라이브러리에서 벡터는 기본으로 세로벡터다

따라서, 행렬 * 벡터는 무조건 차원 맞게 하기로 하고

통일을 위해서 순전파는 W, 역전파는 W^T 쓰기로 약속한거다

- RNN 알아보기
- 기울기 소멸, 폭주
- LSTM
- Softmax, Dense

감사합니다