1. 딥러닝과 머신러닝의 차이는 무엇인가? (2강 참조)

머신러닝은 주어진 데이터를 사람이 인위적으로 특징을 추출한 뒤 데이터를 통해 컴퓨터를 학습시킵니다. 딥러닝은 사람이 특징을 추출하지 않아도 데이터만 제공하면 컴퓨터가 인공신경망을 통해 알아서 특징을 만들어낸다. 따라서 사람이 문제에 대한 지식을 몰라도 문제해결능력을 갖춘다.

딥러닝은 뇌 신경망을 본따 만든 머신러닝 알고리즘을 통해 학습한다. 이 신경망은 층층이 깊게 이루어져 있다.

딥러닝은 머신러닝보다 속도가 느리며, 복잡하고, 계산량이 많다.

2. RNN의 문제점을 해결하기 위해 LSTM이 개발되었다. RNN의 문제점은 무엇이며 어떻게 해결한 것인가? (2강 참조)

RNN은 시게열 데이터를 가지고 학습하며 과거의 정보를 모두 기억한다. 따라서 시퀀스가 길 때 weight가 1보다 크면 gradient exploding 문제가 발생하고 1보다 작으면 gradient vanishing 현상이 발생한다. LSTM은 이 문제를 해결하기 위해 cell state와 3개의 gate를 이용해 다음 셀 상태를 업데이트하고 전달한다. , forget 게이트는 이전 셀 상태에서 어떤 정보를 삭제해야 할지 결정합니다. 둘째로, input 게이트는 현재 입력값과 이전 상태에서 새로운 정보를 얼마나 추가할지를 결정합니다. 마지막으로, output 게이트는 다음 숨겨진 상태의 값과 새로운 셀 상태를 결정합니다.

3. LSTM의 구조와 핵심기능에 대해서 설명하라. (2-3강 참조)

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=winddori2002&logNo=221992543837>

cell state와 3개의 게이트가 있다.

Forget gate: 과거 정보를 기억할지 잊을지를 결정한다. 시그모이드 함수에 Input data를 넣어 연산을 진행한다. 시그모이드 함수는0~1사이의 값을 출력한다. 0에 가까울수록 정보를 잊은것이다.

Input gate: 현재 정보를 잊기/기억하기' 를 결정한다. 시그모이드 함수와 tanh함수에 Input data를 넣어 연산을 진행한다. 시그모이드 함수는 0에 가까울수록 많은 정보를 삭제하는 것을 의미하고 tanh함수는 -1~1 사이의 값을 갖는데 나중에 현재 정보를 cell state에 얼마나 더할지를 결정하는 역할을 한다.

두 게이트에서 나온 출력결과를 이용해 과거에서 유지할 정보 + 현재에서 유지할 정보를 현재 시점의 cell state에 update한다.

Output gate: 시그모이드 함수에 input 데이터를 입력하고 업데이트된 cell state에 tanh함수를 적용하여 두개를 곱해 output으로 내보낸다.

4. LSTM을 훈련하기 위해서는 Backpropagation through time(BPTT)기능을 이용한다. BPTT의 동작원리나 목적에 대해서 설명하라.

BPTT 동작원리: timestep에서 예측한 출력결과와 실제출력결과를 비교하여 오차를 계산한다. 편미분과 chain rule을 이용해 역방향으로 전파되어 가중치를 계산하여 업데이트 함으로써 예측결과와 실제결과의 오차를 줄여나간다.

목적: 이 과정을 반복하여 오차가 최소화되는 가중치를 찾아내고, 이를 이용하여 새로운 데이터를 정확하게 예측하는 것이 목적입니다. 이 과정에서 시간이 지남에 따라서 기울기가 점점 작아지는 문제가 발생할 수 있습니다. 이러한 문제를 해결하기 위해서 LSTM(Long Short-Term Memory)과 같은 RNN의 변형 알고리즘이 개발되었습니다.

5. 딥러닝에서 예측의 정확도를 높이기 위해서는 참값과 예측값의 오차를 계산한다. 오차를 줄이기 위한 최적화 알고리즘, stochastic gradient decent(SGD)이 있다. SGD의 동작원리에 대해서 설명하라. (5강 참조)

가중치 1개를 랜덤하게 선택하여 기울기를 계산하고 경사하강법을 이용해 가중치를 업데이트 한다. 이러한 과정을 여러 번 반복하여 손실 함수를 최소화하는 모델 파라미터를 찾습니다.

6. LSTM의 입력데이터를 준비하는 방법으로는 정규화와 표준화 방법이다. 두 방식의 차이에 대해서 설명하시오.

정규화는 데이터를 0과 1사이의 값으로 변환하는 방법입니다. 이 방법은 데이터의 범위를 일정하게 만들어주어 모델이 학습을 더 잘할 수 있도록 도와줍니다.

표준화는 데이터를 평균이 0, 표준편차가 1이 되도록 변환하는 방법입니다. 이 방법은 데이터의 분포를 정규분포에 가깝게 만들어줍니다. 표준화된 데이터는 이상치(outlier)에 대한 영향을 줄이고, 모델이 더 안정적으로 학습을 할 수 있도록 도와줍니다.

7. Categorical data를 입력데이터로 준비하기 위해서는 One-Hot-Encoder함수를 이용한다. One-Hot-Encoder코드를 어떻게 자성하는지 샘플코드를 작성해 보시오. 라이브러리와 모듈로부터 호출하는 코드를 포함해야 함. (5강 참조)

텍스트이(가) 표시된 사진

자동 생성된 설명

8. LSTM의 입력데이터는 3D배열로 준비해야 한다. Input\_shape(samples, time steps, features) 중 sample, time steps, features에 대해서 각각을 설명하시오. (6강 참조)

Sample:데이터셋의 행의 수

Timestep:입력 시퀀스의 길이

feature : timestep에서 하나의 데이터를 표현하는데 사용되는 개별 변수

예시:

1.sample(1분반,2분반,3분반) 2.time step(순차,37개) 3.feature(1명의 점수)

9. LSTM의 output layer에서 활성함수(activation function)의 출력으로 결과값을 예측한다. 활성함수가 어떤 종류가 있는지 사용목적이 무엇인지 설명하시오. (6강 참조)

Linear 함수: 주로 회귀 문제에서 사용됩니다. 출력값이 입력값에 선형적으로 종속됩니다. 부동산 집값 예측(집크기,방개수)

Relu: 이 함수는 0보다 작은 입력값을 0으로 출력하고, 0보다 큰 입력값은 입력값 그대로 출력합니다. DNN에서 사용됨 손글씨 이미지를 분류하는 모델에서 ReLU 함수를 사용하면, sparse한 데이터인 이미지의 특성을 잘 반영하면서 분류 정확도를 높일 수 있습니다.

tanH: 실제 값 입력을 -1과 1 사이의 값으로 매핑합니다. 여기서 출력은-1과 1 사이에서 정규화됨 LSTM에서 사용됨

softmax 함수는 출력값을 다중 클래스 분류(classification) 문제에서 확률로 나타내주는 함수로, 각 클래스에 대한 확률의 합이 1이 되도록 정규화해줍니다. 따라서 softmax 함수는 주로 다중 클래스 분류 문제에서 출력값을 예측하는데 사용됩니다.

sigmoid 함수는 이진 분류(binary classification) 문제에서 출력값을 확률로 나타내주는 함수로, 0과 1사이의 값을 출력합니다. 이진 분류 문제에서는 하나의 클래스만을 고려하기 때문에 sigmoid 함수가 사용됩니다.

10. LSTM네트워크의 출력은 참값과 예측값의 오차를 계산하는 loss function의 함수가 있다. 3종류의 Loss function에 대해서 설명하시오.

각각의 함수의 사용 예시를 설명해드리겠습니다.

1.MSE (Mean Squared Error) 회귀 문제에서 가장 많이 사용되는 손실 함수입니다.

참값과 예측값의 제곱 오차를 평균한 값으로, 제곱이 포함되어 있어 오차가 커질수록 더 큰 패널티를 부여합니다.

2. Categorical Cross-Entropy

Categorical Cross-Entropy는 다중 분류(multi-class classification) 문제에서 사용됩니다. 출력 값이 0과 1 사이의 확률값으로 표현되는 경우, 이 확률값과 실제값의 차이를 측정합니다.

Binary Cross-Entropy는 이진 분류(binary classification) 문제에서 사용됩니다. 출력 값이 0 또는 1인 이진 분류에서 사용되며, 참값과 예측값 사이의 오차를 계산합니다. Cross-Entropy는 참값과 예측값이 서로 다를 때, 큰 오차를 얻으므로 학습을 더욱 빠르게 수렴시키는 데에 효과적입니다.

11. LSTM모델을 훈련 중에 사용하는 epoch, batch변수들이 있다. 각각의 의미를 설명하시오. (6강 참조)

Epoch:하나의 데이터셋 전체 데이터 셋에 관하여 10번 훈련을 완료하면 10 epoch 완료했다고 함.

Batch\_size: 한 번에 처리되는 데이터의 양을 의미합니다. Batch size가 32면 32개의 데이터가 한번에 모델에 입력되어 예측값을 계산하고 이를 바탕으로 가중치가 업데이트 됩니다.

12. LSTM모델의 4종류에 대해서 설명하시오.(7강 참조)

One to One- 하나의 입력에 하나의 출력이 있는 형태 시계열 데이터에서 다음 실제값을 예측할 때 사용 . 예를 들어, 이미지 분류와 같이 하나의 입력 이미지에 대한 하나의 출력 레이블이 있는 경우에 사용됩니다. 3d데이터 필요없음

One to Many: 하나의 입력에 다중 출력이 있는 형태, 한 개의 이미지를 집어넣었을 때, 그 이미지에 대한 설명을 텍스트로 출력하는경우에 사용.

Many to One: 여러 개의 입력 시퀀스가 주어졌을 때 하나의 출력을 생성하는 모델입니다. 예를 들어, 텍스트 분류와 같이 여러 개의 문장으로 이루어진 문서를 입력으로 받아 이 문서가 어떤 카테고리에 속하는지 분류하는 경우에 사용됩니다.

Many to Many: 문서를 요약할 경우 사용됨

13. LSTM네트워크에서 RepeatVector레이어의 사용 목적에 대해서 설명하시오. (7강의 참조)

RepeatVector 레이어는 LSTM 네트워크에서 사용되며, 출력 시퀀스의 길이를 입력 시퀀스의 길이와 동일하게 만들어줍니다. 2D출력을 3D입력에 맞추는 용도로 사용 시퀀스가 가변 길이일 때 사용

14. LSTM모델을 구축하는 코드를 작성하시오. (6강의 참조)

텍스트이(가) 표시된 사진

자동 생성된 설명

For example, we can define an LSTM hidden layer with 2 memory cells followed by a Dense

output layer with 1 neuron as follows:

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense, RepeatVector, TimeDistributed

model = Sequential()

model.add(LSTM(2))

model.add(Dense(1))

15. Vanilla LSTM모델을 구축하는 코드를 작성하시오. (8강의 참조)

텍스트, 스크린샷, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Embedding, LSTM

length = 5

n\_features = 10

out\_index = 2

model = Sequential()

model.add(LSTM(25, input\_shape=(length, n\_features)))

model.add(Dense(n\_features, activation= 'softmax' ))

model.compile(loss= 'categorical\_crossentropy' , optimizer= 'adam' , metrics=[ 'acc' ])

print(model.summary())