**교육일지**

**교육 제목 : 딥러닝**

**교육 장소 : YGL C6 강의실**

**교육 일시 : 2021/11/02**

**keras로 RNN 구현하기**

**RNN 층을 추가하는 코드**

**model.add(SimpleRNN(hidden\_size)) #가장 간단한 형태**

**# 추가 인자를 사용할 때**

**model.add(SimpleRNN(hidden\_size, input\_shape = (timesteps, input\_dim))**

**# 다른 표기**

**model.add(SimpleRNN(hidden\_size, input\_length=M, input\_dim=N))**

**from tensorflow.keras.models import Sequential**

**from tensorflow.keras.layers import SimpleRNN**

**model = Sequential()**

**model.add(SimpleRNN(3, input\_shape=(2,10)))**

**# model.add(SimpleRNN(3, input\_length=2, input\_dim=10)) 동일**

**model.summary()**

**model= Sequential()**

**model.add(SimpleRNN(3, batch\_input\_shape=(8, 2, 10)))**

**model.summary()**

**model= Sequential()**

**model.add(SimpleRNN(3, batch\_input\_shape=(8, 2, 10), return\_sequences=True))**

**model.summary()**

**파이썬으로 RNN구현하기**

**hidden\_state = 0 # 초기 은닉 상태를 0으로 초기화**

**for input\_t in input\_length: # 각 시점마다 입력을 받는다.**

**output\_t = tanh(input\_t, hidden\_state\_t) # 각 시점에 대해서 입력과 은닉 상태를 가지고 연산**

**hidden\_state\_t = output\_t # 계산 결과는 현재 시점의 은닉 상태과 된다.**

**import numpy as np**

**timesteps = 10**

**input\_dim = 4**

**hidden\_size = 8**

**# 입력에 해당되는 2D 텐서**

**inputs = np.random.random((timesteps, input\_dim))**

**# 초기 은닉 상태는 0벡터로 초기화**

**hidden\_state\_t = np.zeros((hidden\_size,))**

**print(hidden\_state\_t) # 8의 크기를 가지는 hidden state. 현재는 초기 hidden state로 모든 차원이 0의 값을 가짐**

**Wx = np.random.random((hidden\_size, input\_dim)) #(8,4) 2D텐서 생성 입력에 대한 가중치**

**Wh = np.random.random((hidden\_size, hidden\_size)) # (8,8)크기의 2D텐서 생성. hidden state에 대한 가중치**

**b = np.random.random((hidden\_size,)) #(8,)크기의 1D텐서 생성. 편향(bias)**

**print(np.shape(Wx)) # hidden state x 입력의 차원**

**print(np.shape(Wh)) # hidden state x hidden state size**

**print(np.shape(b)) # hidden state size**

**total\_hidden\_states= []**

**# 메모리 셀 동작**

**for input\_t in inputs: # 각 시점에 따라서 입력값이 입력이 됨.**

**output\_t = np.tanh(np.dot(Wx, input\_t) + np.dot(Wh, hidden\_state\_t)+b)**

**# Wx\* wt + Wh \* Ht-1 + b**

**total\_hidden\_states.append(list(output\_t)) #각 시점의 은닉 상태의 값을 계속해서 축적**

**print(np.shape(total\_hidden\_states)) # 각 시점 t별 메모리 셀의 출력의 크기는 (timestep, output\_dim)**

**hidden\_state\_t = output\_t**

**total\_hidden\_state = np.stack(total\_hidden\_states, axis=0)**

**# (timesteps, output\_dim)**

**print(total\_hidden\_states)**

**더 깊은 RNN**

**model = Sequential()**

**model.add(SimpleRNN(hidden\_size, input\_length =10, input\_dim =5, return\_sequences=True))**

**model.add(SimpleRNN(hidden\_size, return\_sequences = True))**

**model.summary()**

**BiLSTM**

**from tensorflow.keras.layers import Bidirectional**

**timesteps = 10**

**input\_dim = 5**

**model = Sequential()**

**model.add(Bidirectional(SimpleRNN(hidden\_size, return\_sequences=True, input\_shape=(timesteps, input\_dim))))**

**model = Sequential()**

**model.add(Bidirectional(SimpleRNN(hidden\_size, return\_sequences=True, input\_shape=(timesteps, input\_dim))))**

**model.add(Bidirectional(SimpleRNN(hidden\_size, return\_sequences=True)))**

**model.add(Bidirectional(SimpleRNN(hidden\_size, return\_sequences=True)))**

**model.add(Bidirectional(SimpleRNN(hidden\_size, return\_sequences=True)))**

**- Embedding을 사용하며, 단어 집합(Vocabulary)의 크기가 5,000이고 임베딩 벡터의 차원은 100입니다.**

**- 은닉층에서는 Simple RNN을 사용하며, 은닉 상태의 크기는 128입니다.**

**- 훈련에 사용하는 모든 샘플의 길이는 30으로 가정합니다.**

**- 이진 분류를 수행하는 모델로, 출력층의 뉴런은 1개로 시그모이드 함수를 사용합니다.**

**- 은닉층은 1개입니다.**

**## GRU 코드**

**model.add(GRU(hidden\_size, input\_shape=(timesteps, input\_dim)))**