Re: [질문]TimeDistributed 레이어의 역할이 무엇인가요?

보낸 사람 Won Joon Yoo <ukairia777@gmail.com> (2020.12.27 01:46)
받는 사람 김종범

안녕하세요. 종범님.

제가 강의 시에 TimeDistributed 레이어는 크게 신경쓰지 말라는 식으로 설명을 했던 것 같은데
그래서 혼란을 드린 것 같습니다. 사실 자연어 처리에서는 주로 RNN 계열의 신경망을 쓰니까
일반적으로는 맞는 말이기는 한데 CNN을 사용할 때는 주의해야 합니다.

제가 이걸 감안하지 못하고 설명드린 것 같은데 제 실수입니다.

여튼 메일로나마 설명을 해드리면 TimeDistributed는 조금 쉽게 설명하면
문장 길이만큼 각각 실행한다라는 의미 정도로 해석하시면 될 것 같습니다.

예를 들어 LSTM이 returns_sequences=True인 경우에 Dense에다가 TimeDistributed()를 써주는 것을 볼 수 있는데요.

```
output = Bidirectional(LSTM(50, return_sequences=True, dropout=0.50, recurrent_dropout=0.25))(output)
output = TimeDistributed(Dense(tag_size, activation='softmax'))(output)
```

위 의미는 LSTM의 매 시점마다 Dense가 각각 실행된다는 의미입니다.
그래서 옛날에는 LSTM에서 return_sequences=True를 해준 경우에는 TimeDistributed()를 써줬어야 했습니다.

그러나 요즘에는 더 이상 LSTM 이후의 Dense에도 TimeDistributed()를 굳이 쓰지 않습니다.
즉, LSTM -> Dense로 가는 구간에서는 생략하더라도 모델이 정상 동작합니다.

실제로 아래 코드에서 제가 굵은 글씨로 표시한 부분을 보시면 됩니다.

```
==================================

# 단어 임베딩
words_input = Input(shape=(None,),dtype='int32',name='words_input')
words = Embedding(input_dim = vocab_size, output_dim = 64)(words_input)

# char 임베딩
character_input = Input(shape=(None, max_len_char,),name='char_input')
embed_char_out = TimeDistributed(Embedding(len(char_to_index), 30, embeddings_initializer=RandomUniform(minval=-0.5, maxval=0.5)), name='char_embeddin
dropout = Dropout(0.5)(embed_char_out)

# char 임베딩에 대해서는 Conv1D 수행
conv1d_out= TimeDistributed(Conv1D(kernel_size=3, filters=30, padding='same',activation='tanh', strides=1))(dropout)
maxpool_out=TimeDistributed(MaxPooling1D(max_len_char))(conv1d_out)
char = TimeDistributed(Flatten())(maxpool_out)
char = Dropout(0.5)(char)

# char 임베딩을 Conv1D 수행한 뒤에 단어 임베딩과 연결
output = concatenate([words, char])

# 연결한 벡터를 가지고 문장의 길이만큼 LSTM을 수행
output = Bidirectional(LSTM(50, return_sequences=True, dropout=0.50, recurrent_dropout=0.25))(output)

# 출력층
output = TimeDistributed(Dense(tag_size, activation='softmax'))(output)  <== 이 부분에서 TimeDistributed를 제거해도 무방합니다.

model = Model(inputs=[words_input, character_input], outputs=[output])
model.compile(loss='categorical_crossentropy', optimizer='nadam',  metrics=['acc'])
model.summary()

 ==================================
```

이는 LSTM에서 Dense로 가는 구간에서는 LSTM이 many-to-many 문제일 때
return_sequences가 True인 경우에 어차피 매 시점마다 출력을 내놓는 것을 상식적으로 알 수 있기 때문에
이제는 코드 상에서 제외가 가능하도록 한 것입니다.

하지만 Conv1D의 경우에는 조금 얘기가 다릅니다.

단어 단위의 Embedding을 거친 경우에는 Embedding 후의 출력이 (배치 크기, 문장 길이, 임베딩 사이즈) 이렇게 3차원입니다.
하지만 글자 단위 Embedding을 거친 경우에는 Embedding 후의 출력이 (배치 크기, 문장 길이, 단어 길이, 임베딩 사이즈) 이렇게 4차원입니다.

Conv1D의 경우에도 일반적으로 3차원을 입력으로 받습니다. 그래서 단어 단위 Embedding을 거친 후에는 전혀 문제가 안 됩니다.
CNN을 이용한 텍스트 분류 예제를 생각해보시면, Embedding 후에 바로 Conv1D를 연결하는 것을 보실 수 있을 것입니다.
Ex)

```
model = Sequential()
model.add(Embedding(vocab_size, 256))
model.add(Conv1D(256, 3, padding='valid', activation='relu'))
```

하지만 글자 단위 임베딩을 수행한 위의 개체명 인식 코드의

경우에는 Conv1D에서 혼란이 발생하는데, Conv1D는 3D 텐서를 입력으로 받아야하는데
위의 경우 4D 텐서가 입력이 되니까 말이 안 된다고 주장하는 것입니다.

에러 경고 문구가 3D 텐서를 기대하였으나 4D 텐서가 들어온다는 게 바로 그 의미입니다.
(ValueError: Input 0 is incompatible with layer conv1d_2: expected ndim=3, found ndim=4)

여기에 TimeDistributed를 쓰게되면 TimeDistributed는 각 문장 길이만큼 Conv1D를 동작시키게 되면서
저희가 예상하는대로 동작을 하게 됩니다.

즉, 위에서는 써주시는 게 맞습니다.
제 설명이 이해가 잘 안 되면 다시 메일부탁드립니다.

감사합니다.

2020년 12월 26일 (토) 오후 2:25, 김종범 <6363@hdc-dvp.com>님이 작성:

> BiLSTM-CNN 구조로 ner dataset 푸는 문제에서
>
> dropout.shape은 (None, None, 15, 30)일 때
>
> TimeDsitributed(Conv1())(dropout) 하면 에러가 안 나는데
>
> 그냥 Conv1d()(dropout) 하면 차원이 맞지 않는다는 문제가 발생합니다.

```
Conv1D(kernel_size=3, filters=30, padding='same',activation='tanh', strides=1)
(dropout).shape
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-58-1a577d8683ac> in <module>
----> 1 Conv1D(kernel_size=3, filters=30, padding='same',activation='tanh', str
ides=1)(dropout).shape

~\Anaconda3\envs\tf2.3\lib\site-packages\keras\engine\base_layer.py in __c
all__(self, inputs, **kwargs)
    412                 # Raise exceptions in case the input is not compatib
le
    413                 # with the input_spec specified in the layer constru
ctor.
--> 414                 self.assert_input_compatibility(inputs)
    415
    416                 # Collect input shapes to build layer.

~\Anaconda3\envs\tf2.3\lib\site-packages\keras\engine\base_layer.py in ass
ert_input_compatibility(self, inputs)
    309                                     self.name + ': expected ndim=' +
    310                                     str(spec.ndim) + ', found ndim=' +
--> 311                                     str(K.ndim(x)))
    312             if spec.max_ndim is not None:
    313                 ndim = K.ndim(x)

ValueError: Input 0 is incompatible with layer conv1d_2: expected ndim=3, found nd
im=4
```

```
TimeDistributed(Conv1D(kernel_size=3, filters=30,
padding='same',activation='tanh', strides=1))(dropout).shape
```

```
TensorShape([Dimension(None), Dimension(None), Dimension(15), Dimension(30)])
```

> TimeDistributed 레이어는 이제는 사용하지 않는다 이런 식으로 알고 있었는데 그게 아닌가요? ㅠ
>
> 김종범
> 디지털혁신팀 / 매니저
>
> 서울특별시 용산구 한강대로23길 55
> **Mobile** 010-4442-6684
> **Email** 6363@hdc-dvp.com
>
> HDC 현대산업개발

주고받은 메일 5

Won Joon Yoo	[받은 메일함] Re: Re: [질문]TimeDistributed 레이어의 역할이 무엇인가요?	2020.12.27
Won Joon Yoo	[받은 메일함] Re: Re: [질문]TimeDistributed 레이어의 역할이 무엇인가요?	2020.12.27
김종범	[보낸 메일함] RE: Re: [질문]TimeDistributed 레이어의 역할이 무엇인가요?	2020.12.27
Won Joon Yoo	[받은 메일함] Re: [질문]TimeDistributed 레이어의 역할이 무엇인가요?	2020.12.27
김종범	[보낸 메일함] [질문]TimeDistributed 레이어의 역할이 무엇인가요?	2020.12.26