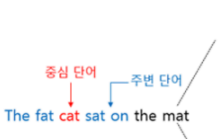
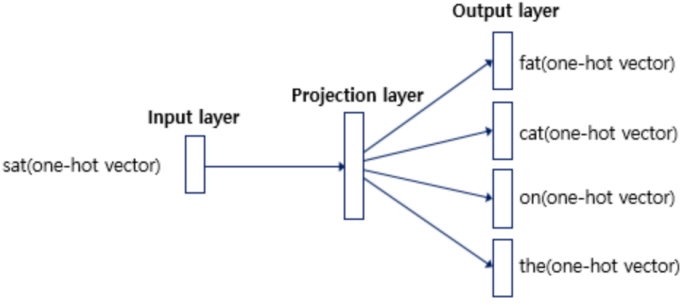
10강

Skit-gram

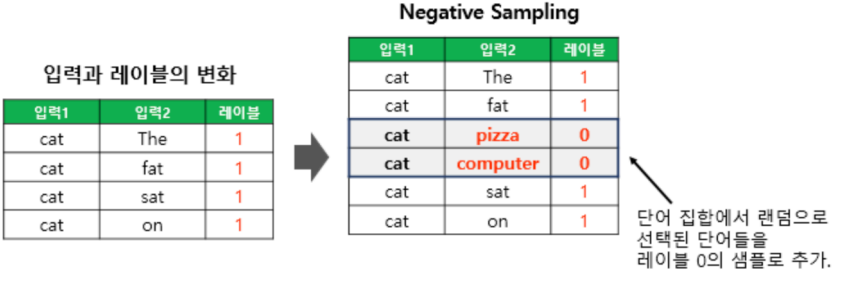


중심 단어	주변 단어
cat	The
cat	Fat
cat	sat
cat	on
sat	fat
sat	cat
sat	on
sat	the

Skit-gram 학습과정



Skip-Gram with Negative Sampling




중심 단어
주변 단어
The fat cat sat on the mat
The fat cat sat on the mat
입력
레이블
cat
The
cat
fat
cat
sat
cat
on
sat
fat
sat
cat
sat
on
sat
the
...
...
입력과 레이블의 변화
입력1
입력2
레이블
cat
The
1
cat
fat
1
cat
sat
1
cat
on
1
sat
fat
1
sat
cat
1
sat
on
1
sat
the
1
...
...
...

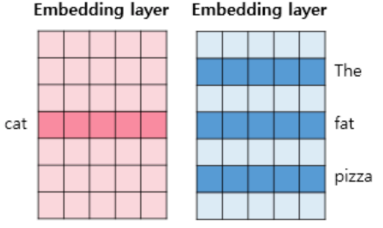
입력1	입력2	레이블
cat	The	1
cat	fat	1
cat	pizza	0
cat	computer	0
cat	sat	1
cat	on	1
cat	cute	1
cat	mighty	0
...	...	...

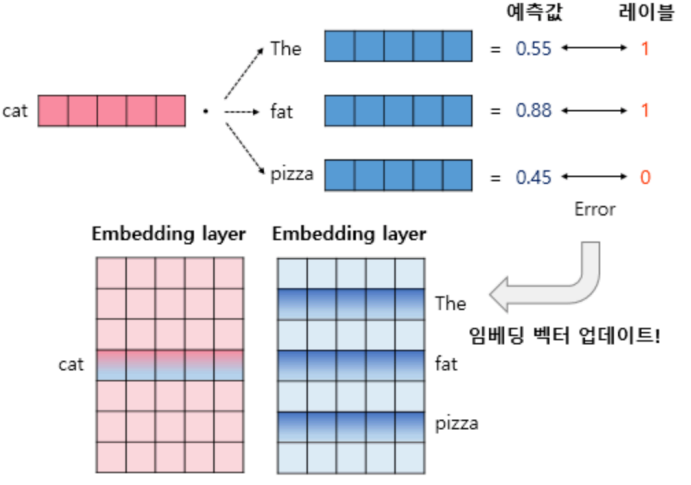
```
from tensorflow.keras.preprocessing.sequence import skipgrams
skip_grams = [skipgrams(sample, vocabulary_size=vocab_size, window_size=10) for sample in sequences[:10]]
```

```
pairs, labels = skip_grams[0][0], skip_grams[0][1]
for i in range(5):
    print("{:s}({:d}), {:s}({:d}) -> {:d}".format(
    idx2word[pairs[i][0]], pairs[i][0],
    idx2word[pairs[i][1]], pairs[i][1],
    labels[i]))
```

```
think(8), know(6) -> 1
signs(2475), believe(42) -> 1
christian(332), pointed(1421) -> 1
tongues(3115), tongues(3115) -> 1
church(352), armentrout(25793) -> 0
```

네거티브 샘플링 학습과정





```
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Embedding, Reshape, Activation, Input, Dot
from tensorflow.keras.utils import plot_model

embed_size = 50

def word2vec():
    target_inputs = Input(shape=(1,), dtype='int32')
    target_embedding = Embedding(vocab_size, embed_size)(target_inputs)

    context_inputs = Input(shape=(1,), dtype='int32')
    context_embedding = Embedding(vocab_size, embed_size)(context_inputs)

    dot_product = Dot(axes=2)([target_embedding, context_embedding])
    dot_product = Reshape((1,), input_shape=(1,1))(dot_product)
    output = Activation('sigmoid')(dot_product)

    model = Model(inputs=[target_inputs, context_inputs], outputs=output)
    model.compile(loss='binary_crossentropy', optimizer='adam')

    return model
```

```python
for epoch in range(1,11):
    loss = 0
    for _, elem in enumerate(skip_grams):
        first_elem = np.array(list(zip(*elem[0]))[0], dtype='int32')
        second_elem = np.array(list(zip(*elem[0]))[1], dtype='int32')
        labels = np.array(elem[1], dtype='int32')
        X = [first_elem, second_elem] # 학습데이터
        Y = labels # 정답
        loss += model.train_on_batch(X,Y) # 한번의 epoch에 여러 번 트레이닝

    print('Epoch : ', epoch, 'Loss : ', loss)
```

```
Epoch :  1 Loss :  1230.8024683147669
Epoch :  2 Loss :  913.956863515079
Epoch :  3 Loss :  804.8944275230169
Epoch :  4 Loss :  752.8729493767023
Epoch :  5 Loss :  721.9261713698506
Epoch :  6 Loss :  698.8518658801913
Epoch :  7 Loss :  677.1577545925975
Epoch :  8 Loss :  653.6212713383138
Epoch :  9 Loss :  626.5351788364351
Epoch :  10 Loss :  595.3925204891711
```