

dropout과 dropconnect

동국대학교 컴퓨터공학전공 김관우, 서혜민

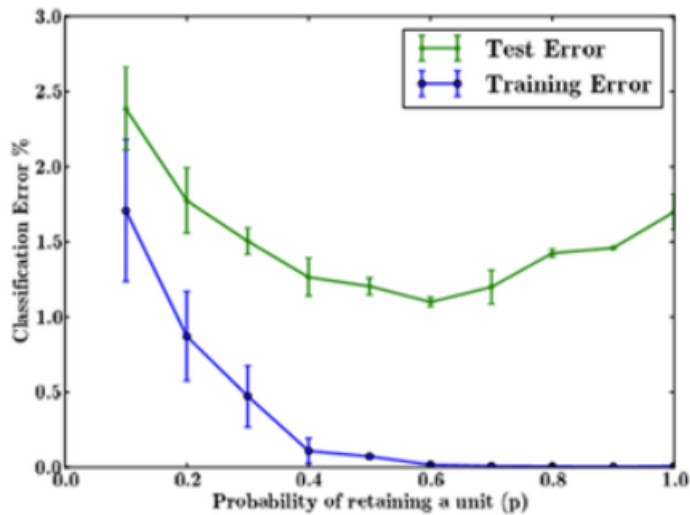
1. Dropout

dropout을 적용하면 음성, 영상, 텍스트 등 응용 분야에 관계없이 신경망의 성능을 개선시킬 수 있다.

뉴런들을 무작위로 생략 시키면서 학습을 수행하므로 파라미터들이 서로 동화 되는 것을 막을 수 있어 좀 더 의미 있는 특징들을 추출할 수 있다.

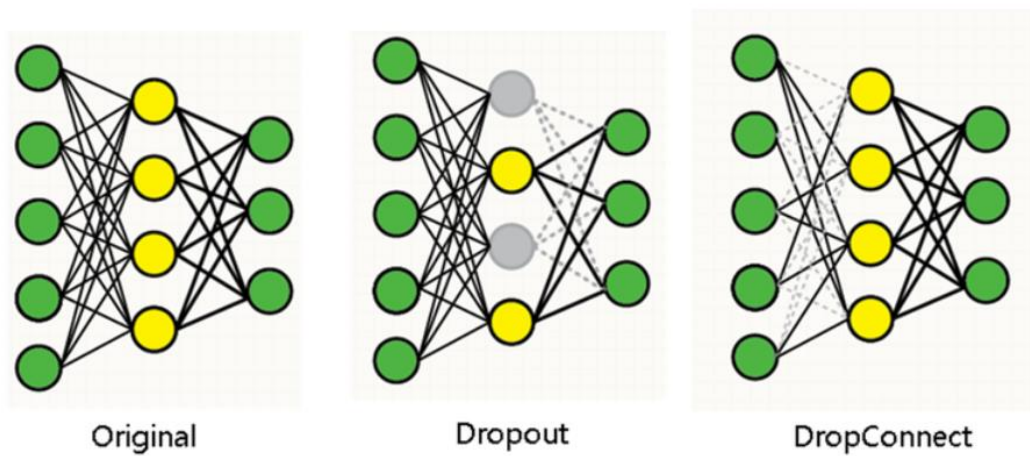
드랍아웃의 파라미터로는 p 값이 들어간다. $\rightarrow \text{Dropout}(p)$

p 의 값은 존속할 확률 변수를 의미한다. 아래 그래프는 이러한 p 의 값을 변화시키며 테스트를 수행한 것이다.



위의 그래프를 보면 p 의 값이 0.6일 때 error가 가장 적고 성능이 좋은 것을 알 수 있다. 하지만 0.6과 0.5는 거의 유사한 값이기 때문에 계산의 편의를 위해 hidden layer에서의 Dropout 확률 p 를 0.5로 정하는 것이 가장 효과적이다.

2. Dropconnect



Dropout은 뉴런을 생략하는 것이고 Dropconnect는 connection을 생략하는 것이다. 즉 Dropconnect는 결과적으로 weight를 생략하는 것이며 노드는 남아있다.

- 기존의 표준 신경망 출력 식

$$S_i(I) = \sum_{j=1}^n w_{ij} I_j \quad \text{for } i = 1, \dots, k$$

i 는 출력 노드의 위치가 되고 j 는 현재 레이어의 노드들이 된다. I 는 인풋 값이고, w 는 연결된 가중치의 값이 되므로 위와 같은 식으로 신경망의 출력을 계산할 수 있다.

- Dropout 적용 신경망 출력 식

$$S_i(I) = \sum_{j=1}^n w_{ij} \delta_j I_j \quad \text{for } i = 1, \dots, k$$

기존 표준 신경망 출력식에 베르누이 랜덤 변수가 적용된 것이 Dropout 신경망의 출력 식이다.

- Dropconnect 적용 신경망 출력 식

$$S_i(I) = \sum_{j=1}^n \delta_{ij} w_{ij} I_j \quad \text{for } i = 1, \dots, k$$

Dropout과 다른 점은 베르누이 랜덤에 ij 가 붙었다는 점이다.

** 베르누이 랜덤 변수

베르누이 랜덤 변수는 확률 결과를 0 또는 1로 바꾼 것을 의미한다. 즉 Dropout은 베르누이 랜덤 변수를 노드에 적용해 노드 자체를 생략하는 효과를 가지고, Dropconnect는 베르누이 랜덤 변수를 가중치에 적용해 connection을 생략하는 효과를 가진다.

3. 논문 작성 방향

- no drop, dropout, dropconnect를 각각 적용한 결과를 비교하여 성능을 비교해 볼 예정
- dropout 적용

```
dense_layer1 = Dense(units=256, activation='relu')(clf_output)
dense_layer1 = Dropout(0.5)(dense_layer1)
out = Dense(1, activation='sigmoid')(dense_layer1)
```

- dropconnect 적용

```
dense_layer1 = Dense(units=256, activation='relu')(clf_output)
dense_layer1 = DropConnectDense(units=256, prob=0.5, activation='relu', use_bias=True)(dense_layer1)
out = Dense(1, activation='sigmoid')(dense_layer1)
```

4. 문제점

- 현재 사용중인 컴퓨터의 사양 문제로 학습의 batch_size를 참고 논문과 같은 수치로 맞춰 학습이 불가능
- 따라서 epochs = 8, batch_size = 10으로 설정하여 세 가지 경우에 대해 모델을 생성하고 성능을 비교할 예정

dropout + max_norm을 같이 사용하여 정규화 하면 성능 향상에 큰 도움이 된다.