AI

과적합

장혜선

2021/08/10

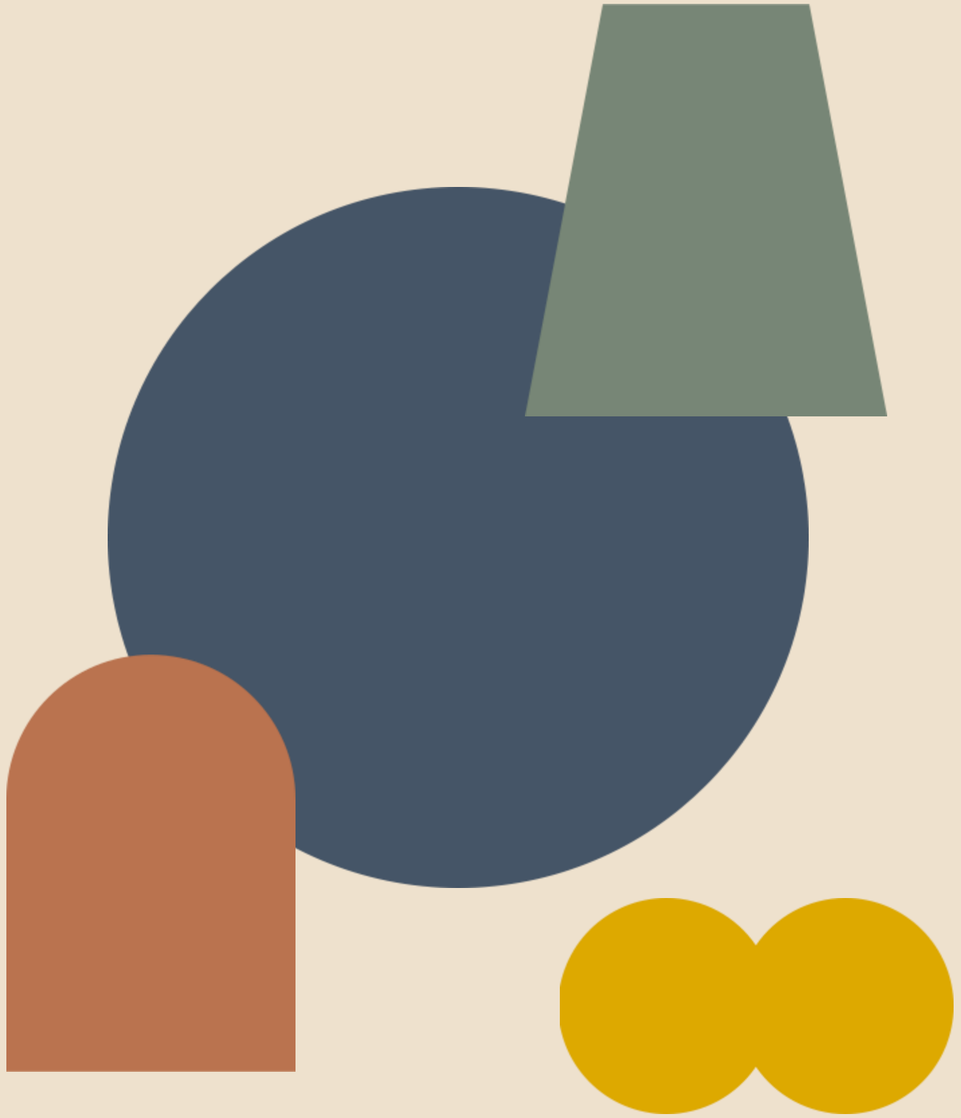
목차

Contents

01

과적합

과적합이란?



Contents

광물 예측

광석과 일반 돌에 음파 탐지기를 쏜 후 결과를 정리한 데이터로
둘을 구분하는 모델

학습셋

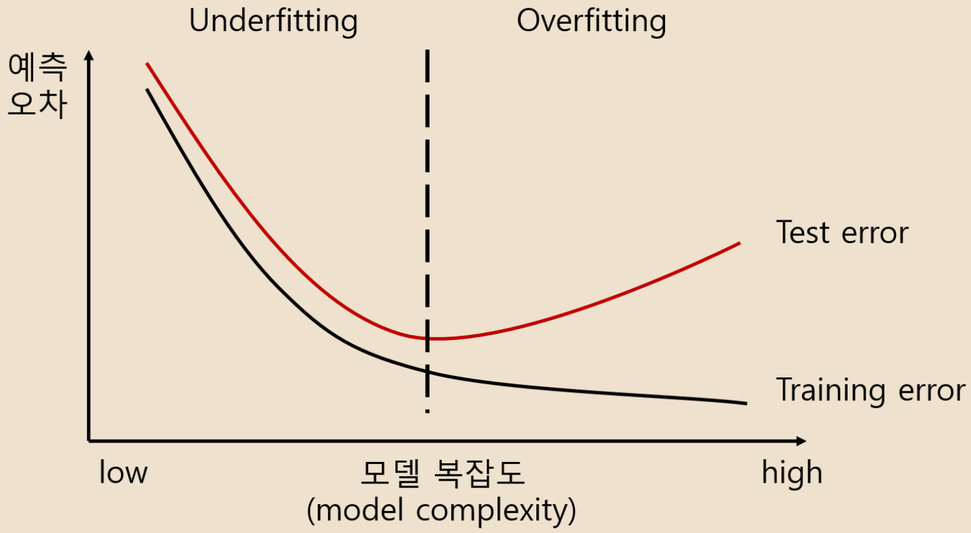
Accuracy: 1.0000

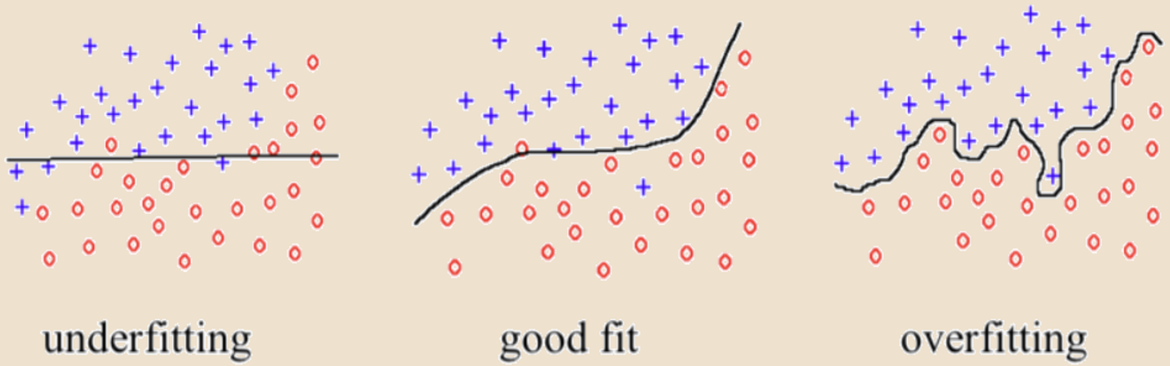
학습셋/테스트

Accuracy: 0.7937

과적합

모델이 학습 데이터셋 안에서는 일정 수준 이상의 예측 정확도를 보이지만,
새로운 데이터에 적용하면 잘 맞지 않는 것





발생 원인

1
- **Key word1**
  모델이 너무 복잡할 때

2
- **Key word2**
  변수가 지나치게 많을 때

3
- **Key word3**
  데이터에 오류가 있을 때

4
- **Key word4**
  편향된 부분만 가졌을 때

복잡도

변수

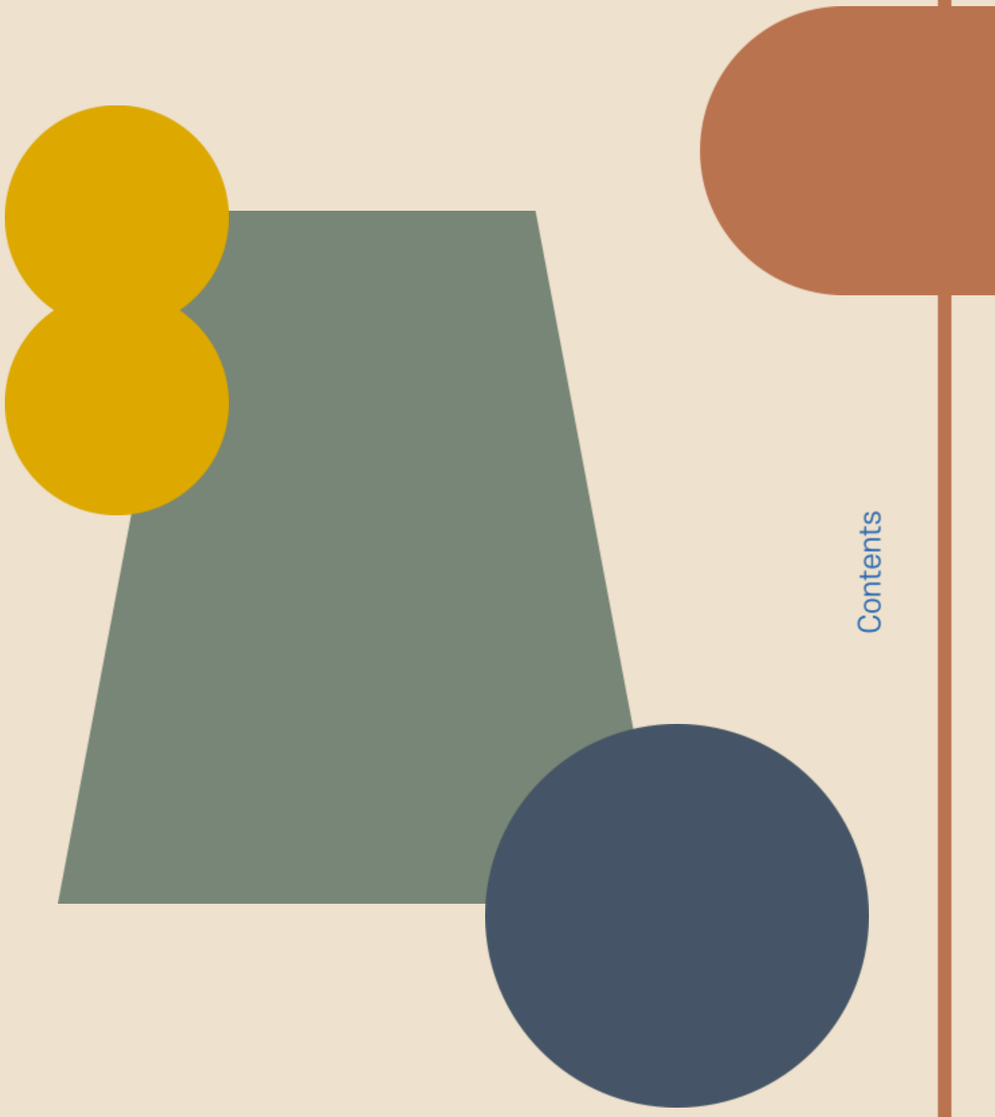
오류

편향

02

대응방안

과적합 방지 방안



대응방안1

데이터의 양 늘리기

- 데이터 양이 적을 때 => 특정 패턴, 노이즈까지 암기 => 과적합 발생 확률 ↑
- 데이터의 양 ↑ => 모델은 데이터의 일반적인 패턴 학습
- 만약, 데이터 양이 적다면? 데이터 증강을 함

대응방안2

모델 복잡도 줄이기

모델의 복잡도 : 은닉층, 매개변수 수

은닉층 수의 변화	학습셋의 예측률	테스트셋의 예측률
0	79.3	73.1
2	96.2	85.7
3	98.1	87.6
6	99.4	89.3
12	99.8	90.4
24	100 …	89.2

대응방안3

가중치 규제 적용

- 복잡한 모델을 간단하게 해줌
- 가중치를 제한하면 모델이 몇 개의 데이터에 집착하지 않게됨
- 가중치에 어떠한 규제 값을 적용하여 예방

L1 규제

손실함수 + 가중치 절댓값

L1 norm

$$||w||_1 = \sum_{i=1}^{n} |w_i|$$

$$L = \underbrace{-(ylog(a) + (1-y)log(1-a))}_{\text{손실함수}} + \underbrace{a\sum_{i=1}^{n} |w_i|}_{\text{a X L1 norm}}$$

대응방안3

가중치 규제 적용

규제된 손실함수 값 = (손실 함수 + 규제값)

L2 규제

손실함수 + 가중치 제곱

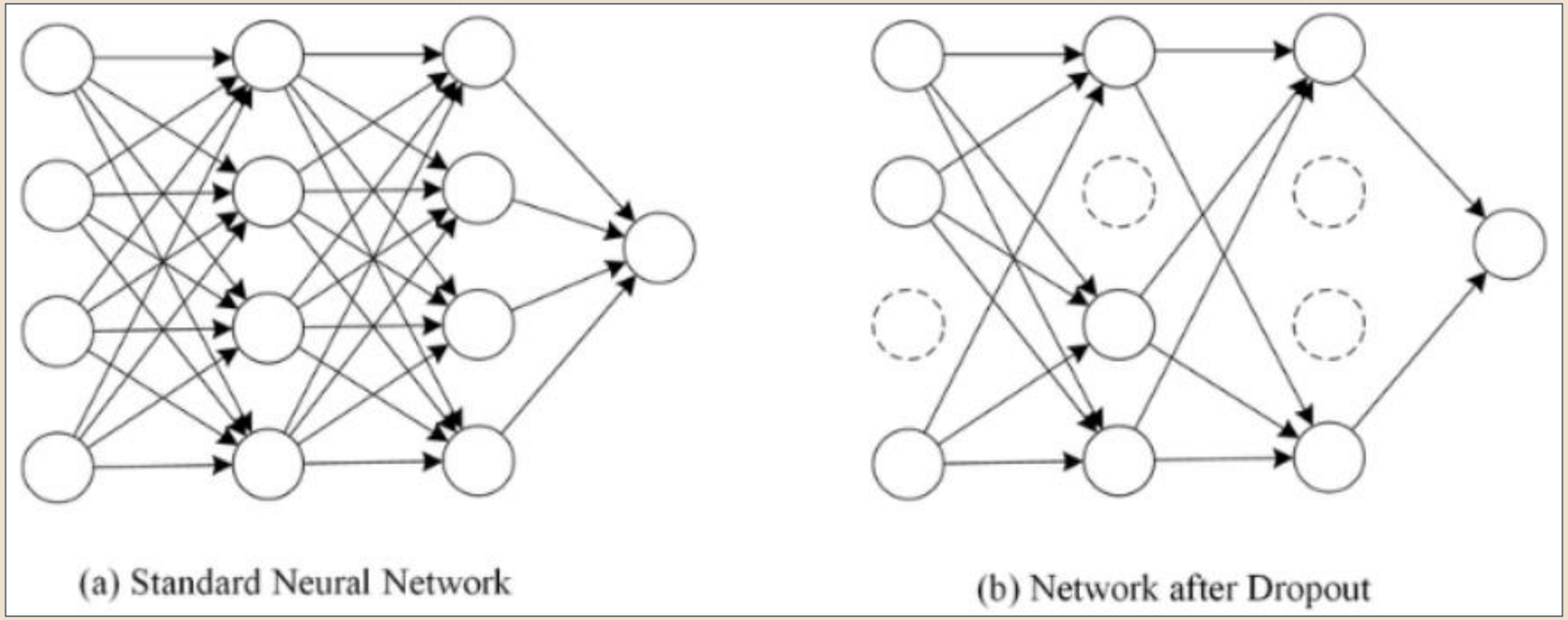
L2 norm

$$||w||_2 = \sqrt{\sum_{i=1}^{n} |w_i|^2}$$

$$L = \underbrace{-\left(y\log(a) + (1-y)\log(1-a)\right)}_{\text{손실함수}} + \underbrace{\frac{1}{2}a\sum_{i=1}^{n}|w_i|^2}_{\text{a X L2 norm}}$$

대응방안4 드롭아웃(Dropout)

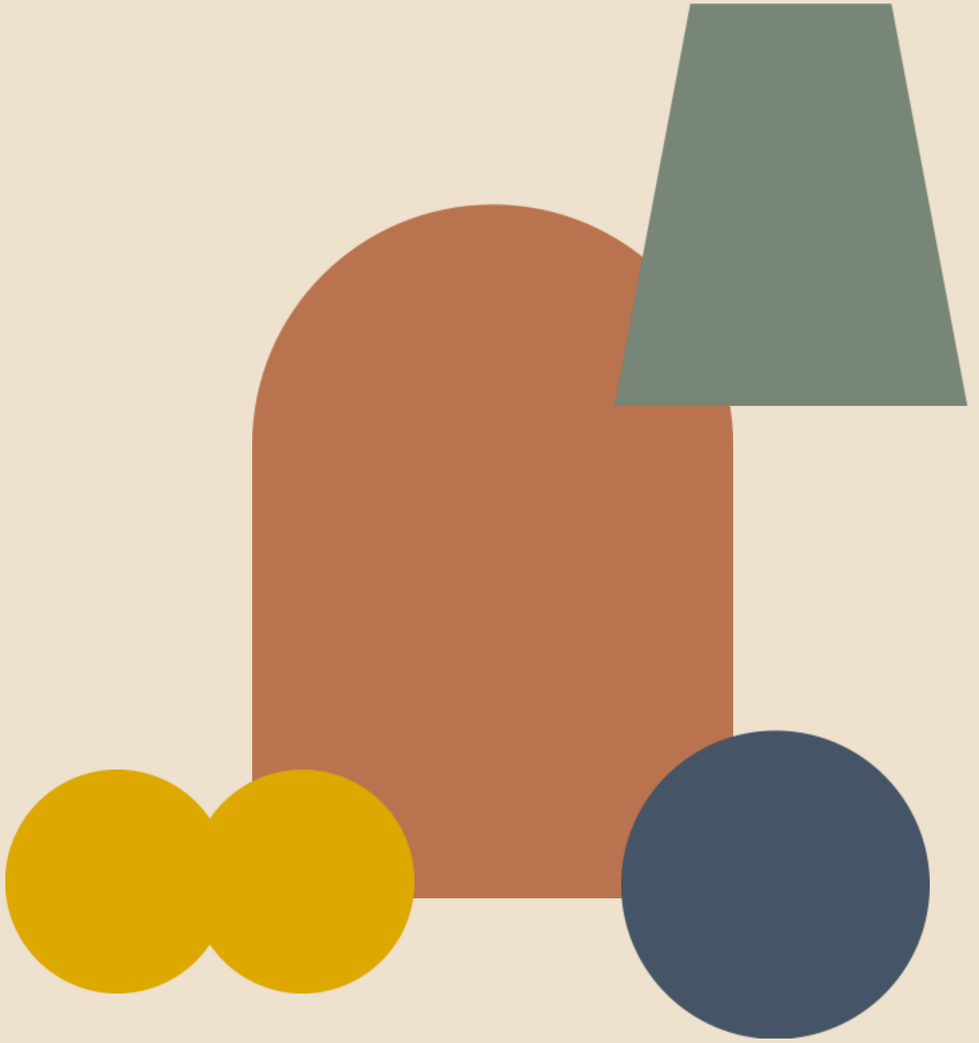
학습 과정에서 신경망 일부를 사용하지 않는 방법



03

학습중단

학습 자동 중단 시키기



학습 중단

정확도와 오차

모델의 학습 시간에 따른 정확도와 테스트 결과 확인하기

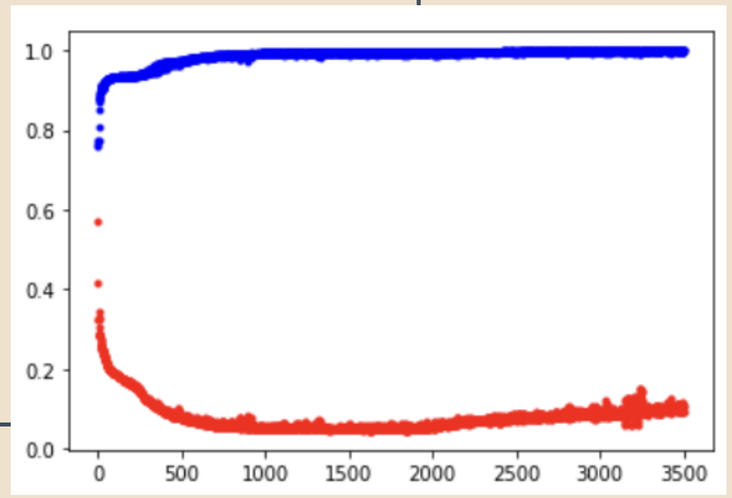
```
df = df_pre.sample(frac=0.15)


history = model.fit(X,Y,validation_split=0.33, epochs=3500, batch_size=500)

y_vloss=history.history['val_loss']
y_acc=history.history['accuracy']

x_len = numpy.arange(len(y_acc))
plt.plot(x_len, y_vloss, "o", c="red", markersize=3)
plt.plot(x_len, y_acc, "o", c="blue", markersize=3)

plt.show()
```



학습 중단

과적합으로 인해 테스트 셋의 정확도가 나빠지므로,
학습을 진행할 때 테스트셋의 오차가 줄지 않으면 학습을 멈추게 하는 것

레드와인/화이트와인 분류

학습 중단

레드와인과 화이트와인 분류 모델

설정

```
seed = 0
numpy.random.seed(seed)
tf.random.set_seed(seed)

df_pre = pd.read_csv('dataset/wine.csv', header=None)
df = df_pre.sample(frac=1)

dataset = df.values
X = dataset[:, 0:12]
Y = dataset[:, 12]
```

학습 중단

신경망

레드와인과 화이트와인 분류 모델

```
model = Sequential()
model.add(Dense(30, input_dim=12, activation='relu'))
model.add(Dense(12, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

학습 중단

레드와인과 화이트와인 분류 모델

설정

```
early_stopping_callback = EarlyStopping(monitor='val_loss',
                                        patience=100)

model.fit(X, Y, validation_split=0.2,
          epochs=2000,
          batch_size=500,
          callbacks=[early_stopping_callback])
```

학습 중단

레드와인과 화이트와인 분류 모델

결과

```
Epoch 487/2000
11/11 [==============================] - 0s 5ms/step - loss: 0.0408 - accuracy: 0.9872 - val_loss: 0.0558 -
val_accuracy: 0.9838
Epoch 488/2000
11/11 [==============================] - 0s 4ms/step - loss: 0.0449 - accuracy: 0.9868 - val_loss: 0.0519 -

<중략>

Epoch 587/2000
11/11 [==============================] - 0s 5ms/step - loss: 0.0357 - accuracy: 0.9897 - val_loss: 0.0537 -
val_accuracy: 0.9862
Epoch 588/2000
11/11 [==============================] - 0s 5ms/step - loss: 0.0459 - accuracy: 0.9882 - val_loss: 0.0531 -
val_accuracy: 0.9862
204/204 [==============================] - 0s 1ms/step - loss: 0.0407 - accuracy: 0.9892

 정확도 : 0.9892
```

감사합니다.

장혜선

2021/08/10