Python을 활용한 이미지 분석

3. 심층 신경망 훈련

기상청

기상청

심층 신경망 훈련

01 옵티마이저

02 모델 컴파일

03 드롭아웃

1

옵티마이저

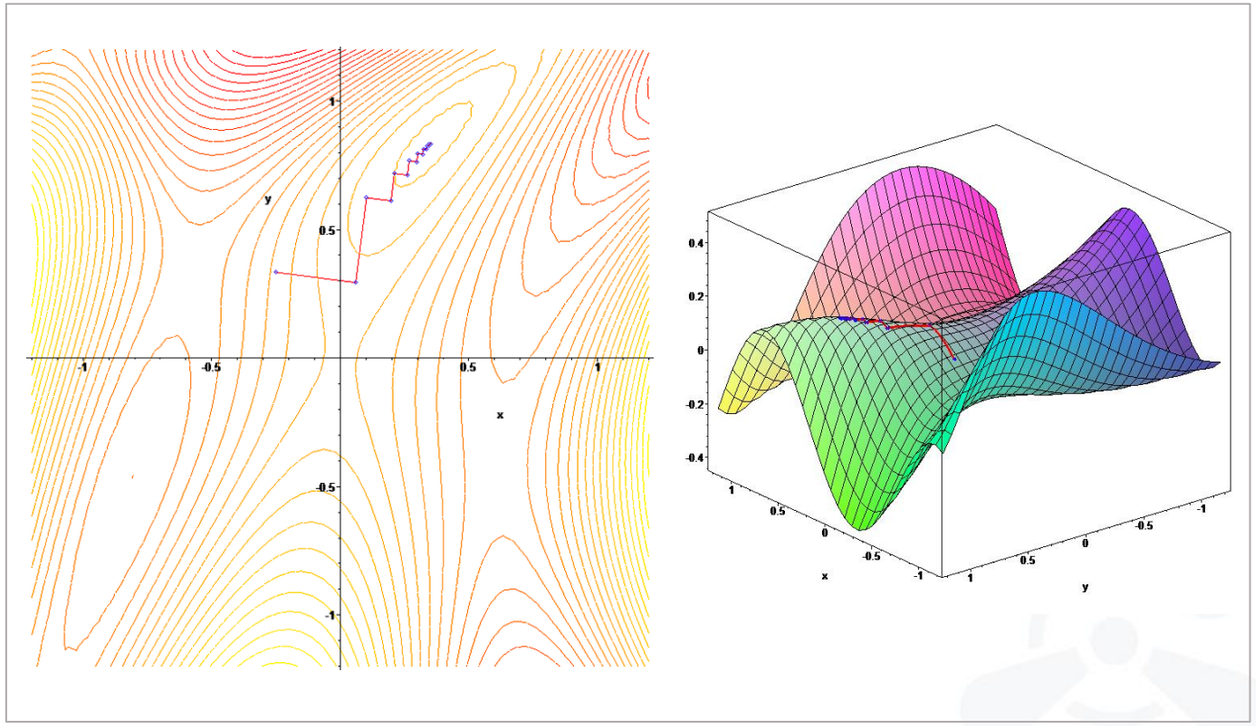
기상청

경사 하강법(Gradient Descent)

개념

경사 하강법 뉴럴넷이 가중치 파라미터를 최적화하는 방법

- 학습률과 손실함수의 기울기를 통해 가중치를 업데이트함
- 오차가 작은 방향으로 학습을 진행함
- Loss Function의 현 가중치에서의 기울기를 구해서 오차를 줄이는 방향으로 학습함

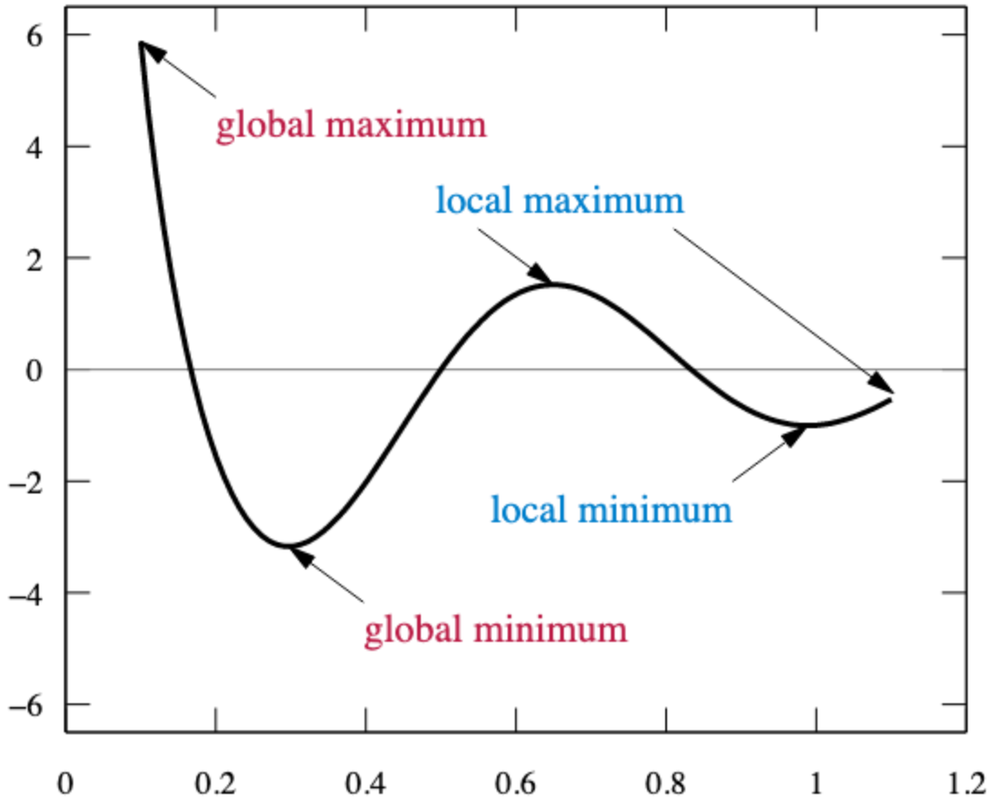


[출처 : 위키피디아. https://en.wikipedia.org]

기상청

경사 하강법(Gradient Descent)

한계



[출처 : 위키피디아. https://en.wikipedia.org]

기상청

경사 하강법(Gradient Descent)

한계

01 학습과정에서 Parameter의 최적값을 찾기 어려운 문제

02 Local Minima 문제

- 에러를 최소화 시키는 최적값을 찾을 때 학습 도중 Local Minima를 최적값으로 인식하여 더이상 학습을 진행하지 않는 문제

개선

01 경사 하강법을 비롯한 최적화 함수를 통해 개선

기상청

경사 하강법(Gradient Descent)

가중치 업데이트 값 구하기

가중치(Weight) 업데이트 =

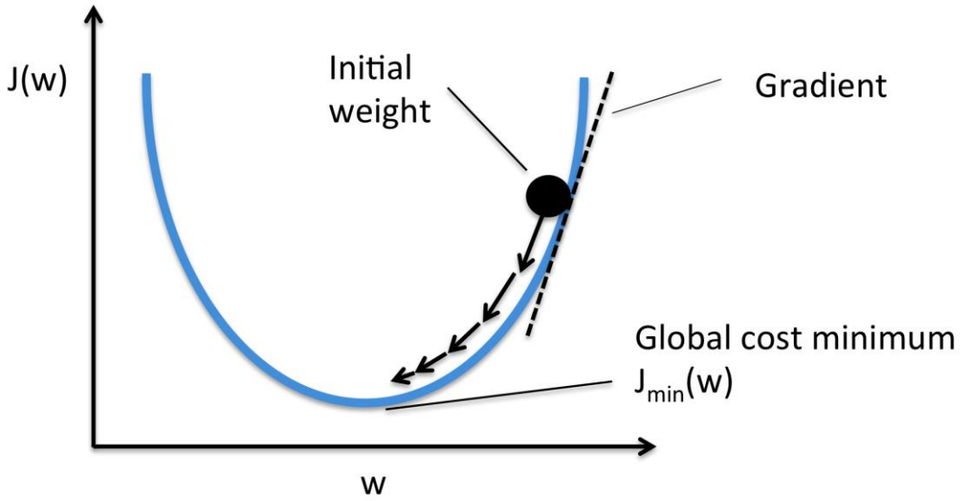
에러를 낮추는 방향 (Descent) × 현 지점의 기울기 (Gradient) × 학습률 (Learning Rate)

기상청

경사 하강법(Gradient Descent)

기울기(Gradient)

- 기울기가 0인 지점의 비용이 최소인 지점으로, 미분의 결과로 얻어지는 기울기가 0임
- 기울기가 0인 지점을 기준으로 왼쪽과 오른쪽의 기울기가 서로 다른 방향임
- 미분의 결과가 0이 될 때까지 반복적으로 미분을 실행하면서 직선의 기울기를 수정하면 비용이 최소가 되는 지점을 찾을 수 있음

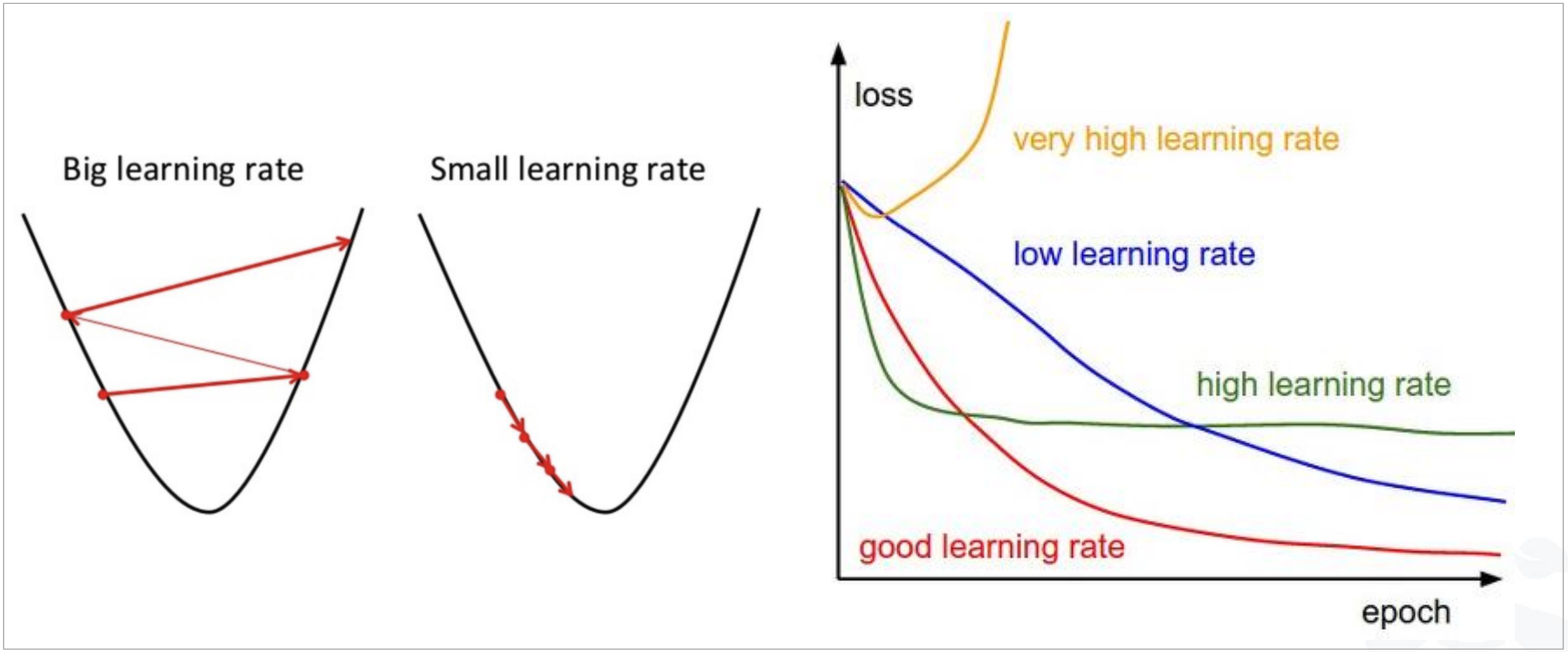


[출처 : MLXTEND. http://rasbt.github.io/mlxtend]

1. 옵티마이저

경사 하강법(Gradient Descent)

학습률(Learning Rate, 보폭)



기상청

옵티마이저의 종류

경사 하강법(Gradient Descent, GD)

- 최적의 값을 찾아가는 것은 정확하지만, 너무 느림

확률적 경사 하강법(Stochastic Gradient Descent, SGD)

- 데이터 샘플을 무작위로 추출하여 일부만 경사 하강법에 사용해 학습 속도를 개선함
- 빠르지만 최적의 값을 찾아가는 방향이 뒤죽박죽임
- 한 스텝 나아가기 위한 사이즈를 정하기가 어려움

1. 옵티마이저

기상청

옵티마이저의 종류

- 방향과 스텝 사이즈를 고려하는 새로운 옵티마이저

방향성			Momentum, NAG
스텝사이즈			Adagrad, RMSPop, AdaDelta
방향성	+	스텝사이즈	Adam, Nadam

1. 옵티마이저

기상청

옵티마이저의 종류

모멘텀(Momentum)

- 손실 함수의 기울기가 0인 여러 지점 중 한 지점에서 멈추지 않고 계속 최솟값을 찾도록 관성 개념을 추가함

RMSProp(Root Mean Square Propagation)

- 손실 함수의 기울기에 따라 학습률을 조절함

아담(Adam)

- RMSProp과 모멘텀의 기능을 결합한 것
- 손실 함수의 기울기에 따라 학습률을 조절하면서 관성을 이용해 지역 최솟값에 갇히지 않고 전체 최솟값을 찾음

1. 옵티마이저

기상청

실습

예제

문제 상황	실습 코드
텐서플로에 내장된 옵티마이저를 확인	dir(tf.keras.optimizers)

예시화면

```
dir(tf.keras.optimizers)
```

```
['Adadelta',
 'Adagrad',
 'Adam',
 'Adamax',
 'Ftrl',
 'Nadam',
 'Optimizer',
 'RMSprop',
 'SGD',
```

2

모델 컴파일

2. 모델 컴파일

기상청

model.compile()

옵티마이저 (Optimizer)

- 데이터와 손실 함수를 바탕으로 모델의 업데이트 방법을 결정함
- GD, SGD, Momentum, RMSProp, Adam 등이 있음

평가지표 (Metrics)

- 훈련 단계와 테스트 단계를 모니터링하기 위해 사용함
- 분류(Accuracy), 회귀(MAE, MSE)에 따라 적절한 평가지표를 사용함

2. 모델 컴파일

기상청

model.compile()

손실 함수(Loss Function)

- 훈련하는 동안 모델의 오차를 측정
- 모델의 학습이 올바른 방향으로 향하도록 함수를 최소화

분류의 손실 함수

바이너리(예측할 값의 종류가 둘 중 하나)

- binary_crossentropy

멀티클래스(예측할 값의 종류가 2개 이상)

- categorical_crossentropy
  - ✓ one-hot형태의 클래스 예 : [0, 1, 0, 0]
- sparse_categorical_crossentropy
  - ✓ 정답값이 0, 1, 2, 3, 4와 같은 형태일 때

2. 모델 컴파일

기상청

model.compile()

회귀의 손실 함수

MSE	MAE
평균제곱오차	평균절대오차

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n}$$

2. 모델 컴파일

기상청

실습

예제

문제 상황	실습 코드
모델 컴파일	model.compile()

예시화면

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

3

드롭아웃

기상청

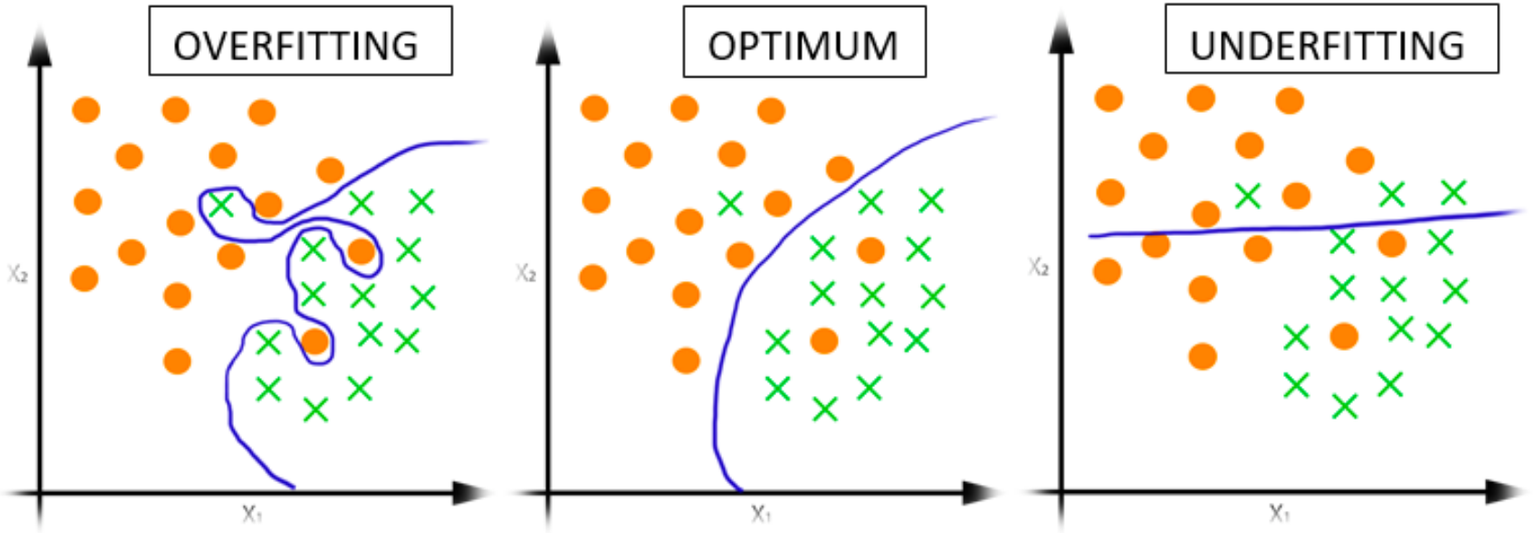
과소적합(Underfitting)과 과적합(Overfitting)

과소적합

- 데이터가 너무 적거나 학습이 잘 되지 않는 상황

과적합

- 훈련 세트에서는 좋은 성능을 내지만 실제 예측 세트에서는 좋은 성능을 내지 못함



[출처 : https://machinelearningmedium.com/2017/09/08/overfitting-and-regularization/]

3. 드롭아웃

기상청

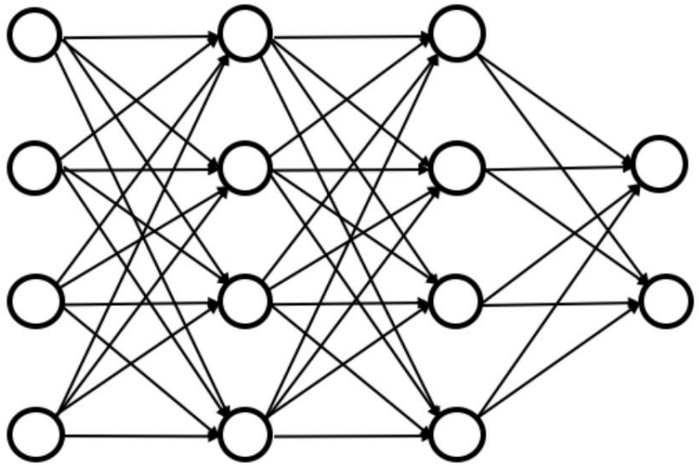
개념

드롭아웃

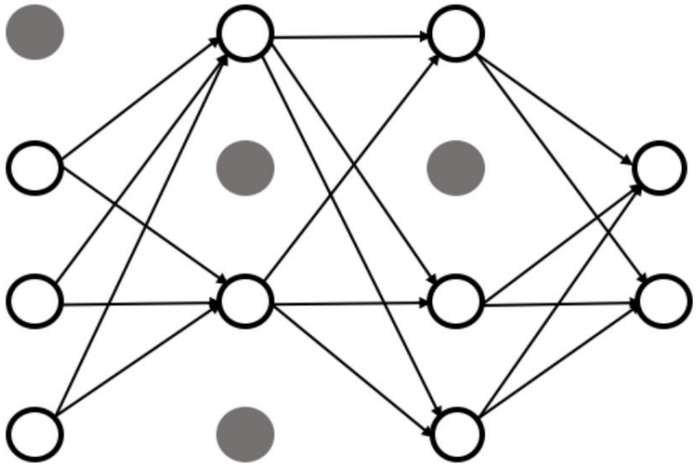
- 모델이 특정 정보에 의존하지 않도록 사용함
- 과적합 방지를 위해 사용함

▶ 모델을 학습시킬 때 지정된 비율만큼의 노드를 무작위로 누락함

[Standard Neural Network]



[Network Applying Dropout]



[출처 : https://arxiv.org/pdf/1809.05165.pdf]

3. 드롭아웃

기상청

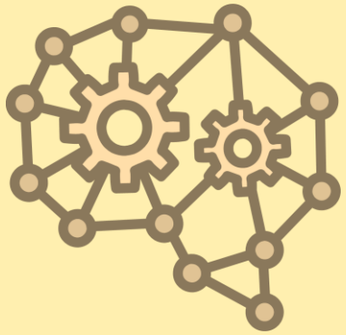
실습

예제

문제 상황	실습 코드
드롭아웃 층 추가	tf.keras.layers.Dropout(0.2)

예시화면

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])
```



KEY POINT

- 옵티마이저
  - 학습 속도, 방향, 스텝 사이즈를 통해 학습을 최적화
  - 방향성 : Momentum, NAG
  - 스텝사이즈 : Adagrad, RMSProp, AdaDelta
  - 방향성 + 스텝사이즈 : Adam, Nadam
- 모델 컴파일
  - 손실 함수(Loss Function)
  - 옵티마이저(Optimizer)
  - 평가지표(Metrics)
- 드롭아웃
  - 모델이 특정 정보에 의존하지 않도록 사용
  - 과적합 방지를 위해 사용
  - 모델을 학습시킬 때 지정된 비율만큼의 노드를 무작위로 누락

Python을 활용한 이미지 분석

3. 심층 신경망 훈련

"이번 시간을 모두 마치셨습니다.
수고하셨습니다."

기상청