

1~2장

1) 초기 가중치 값은 어떻게 설정될까요?

2) 전통적인 프로그래밍 방식과 머신러닝이 다른점은 무엇일까요?

3) 다음중 알은 학습 방법이며 지각에 관련된 문제에 어려움을 겪으나 고차원매핑과 내적연산을 쉽게 만들어주는 기능을 하는것은?

1.SVM

2.결정트리

3.CUDA

4.랜덤포레스트

5.역전파 알고리즘

4) 이 최적화 방법은 AdaGrad의 단점인 학습률을 일괄적으로 처리하는 방식을 수정하기 위해 만들어진 방법이며 기울기를 제공한 값을 누적할 때 누적한 값을 지수평균으로 바꿔주는 방법으로 기울기 크기 조정 속도를 조정한다 이 방법의 이름은?

1) 텐서의 3가지 핵심 속성끼리 알맞게 짝 지어 보세요.

보기

1. 축의 개수(랭크) 2. 크기 3. 데이터 타입

a. shape b. ndim c. dtype

2) 256x256크기의 RGB컬러 이미지에 대한 128개의 배치를 4D텐서 크기로 서술시오.

3~4장

1. 짧은 뉴스 기사를 46가지 토픽 중 하나로 분류하고자 한다. 이 모델의 output이 다음과 같을 때 마지막 layer의 활성화 함수는 무엇인가?

```
predictions = model.predict(x_test)
```

```
predictions[0]
```

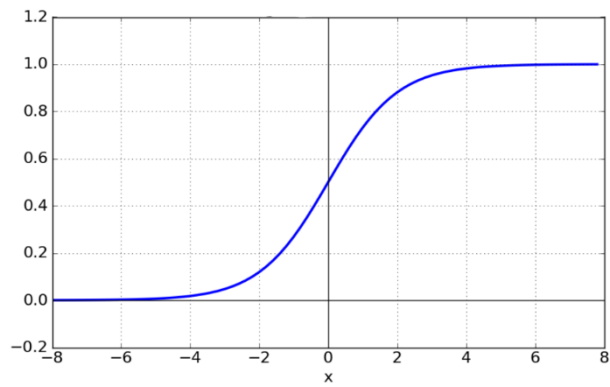
```
array([1.1406273e-06, 8.1942210e-07, 7.8634557e-08, 9.6189708e-01,  
       3.7535086e-02, 2.0714195e-08, 1.5888102e-07, 5.9379153e-07,  
       9.2927199e-05, 1.4421595e-07, 3.5828316e-06, 4.5982274e-06,  
       2.4158712e-06, 3.0957010e-06, 6.5731246e-07, 2.9040924e-08,  
       1.3646432e-04, 5.9918966e-06, 7.5117566e-07, 1.8801076e-05,  
       2.6808772e-04, 8.3087207e-06, 3.0766259e-08, 1.0177722e-06,  
       4.3600226e-09, 6.7802131e-07, 1.2540254e-09, 1.0339551e-07,  
       3.0626563e-08, 3.9220899e-06, 2.6182427e-06, 1.7596884e-08,  
       1.3069395e-07, 1.7569450e-08, 7.0352070e-07, 3.7651744e-06,  
       1.0276561e-06, 3.4851456e-07, 4.6788795e-09, 3.9794268e-06,  
       2.6921694e-07, 4.6970001e-07, 2.2856882e-07, 2.6603728e-09,  
       3.6867907e-13, 7.3737716e-08], dtype=float32)
```

2. K겹 교차 검증은 K번만큼 검증 데이터셋과 훈련 데이터셋을 변경해가면서 거의 모든 데이터에 대해 검증을 하는 방법이다. 이러한 방법을 사용하는 이유(장점)는 무엇인가?
3. 다음은 원소의 개수가 3인 숫자 리스트 a를 원-핫 인코딩한 결과이다. 리스트 a의 원래 모습을 나타내시오.

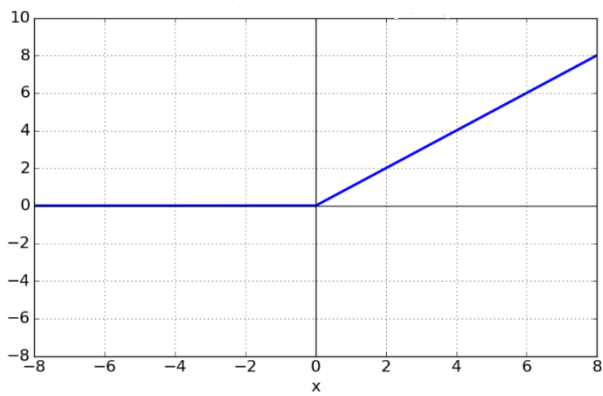
```
[[0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]]
```

4. 다음 중 시그모이드(sigmoid)함수의 그래프를 고르시오.

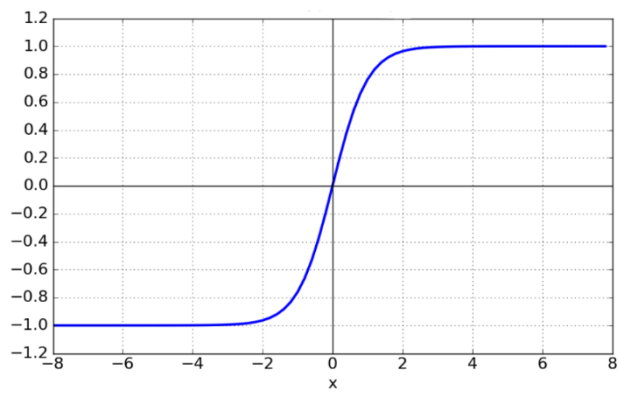
1)



2)



3)



5. 다음은 보스턴 주택 가격예측 모델의 일부 코드입니다. 이 코드가 회귀 문제임을 알 수 있는 코드를 한 줄을 적으시오.

```
from keras import models
from keras import layers

def build_model():
    model=models.Sequential()
    model.add(layers.Dense(64,activation='relu',
input_shape=(train_data.shape[1],)))
    model.add(layers.Dense(64, activation='relu'))
    model.add(layers.Dense(1))
    model.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])
    return model
```

6. 다음은 과대적합(Over-fitting)을 방지하는 방법에 대한 설명들이다. 옳지 않은 것을 모두 고르시오.

- 1.네트워크 크기축소법은 단순히 모델내 파라미터수를 줄이는 것이다.
- 2.가중치규제-L1은 가중치의 절댓값에 비례하는 비용이 손실함수값에 추가되어 가중치를 갱신하는 방법이다.
3. 가중치규제-L2은 가중치의 제곱에 비례하는 비용이 손실함수값에 추가되어 가중치를 갱신하는 방법이다.
4. 드롭아웃(Dropout)은 모델내 지정된 특성의 일부출력을 제외시키는 방법이다.
5. 단순히 학습데이터를 증가시키는 것은 해결방법이 아니다.

5~6장

5.1 최대풀링 연산의 문제점은?

5.2 문제 그럴듯한 이미지를 생성하도록 여러 가지 랜덤한 변환을 적용하여 샘플을 늘리는 방식은?

5.3절

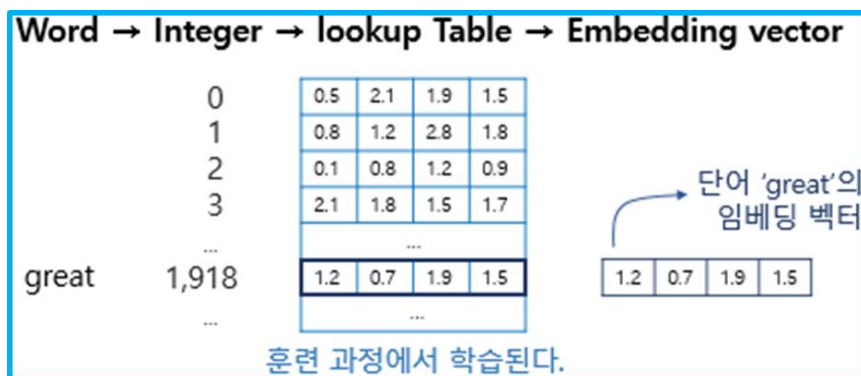
특성 추출에서 미세조정을 사용하는 방법은 ?

5.4절

컨브넷 학습 시각화를 통해 알 수 있는 점이 아닌 것은?

- ① 첫 번째 층은 초기 사진에 있는 거의 모든 정보가 유지된다.
- ② 상위 층으로 갈수록 점점 더 추상적으로 변하고 시각적으로 이해하기 어렵다.
- ③ 상위 층으로 갈수록 눈, 코, 입, 귀처럼 고수준 개념을 학습하기 시작한다.
- ④ 층이 깊어짐에 따라 비어 있는 필터가 줄어든다.
- ⑤ 비어 있는 필터는 학습된 패턴이 없다는 것을 의미한다.

6.1 'embedding 층을 사용하여 단어 임베딩 학습하기' 과정을 그림보고 간단하게 설명하시오.



6.2 문제 LSTM이 RNN보다 긴 시퀀스에서 더 잘 작동하는 이유는?

6.3 6.3절에 예제인 온도 예측 문제의 성능을 향상시키는 방법이 아닌 것은?

- ① 스택킹한 각 순환 층의 유닛 수를 조정한다.
- ② RMSprop 옵티마이저가 사용한 학습률을 조정한다.
- ③ GRU층 대신 LSTM 층을 사용한다.
- ④ 순환 층 위에 큰 완전 연결된 분류 층을 사용한다.

6.4

컨브넷을 사용한 시퀀스처리에 대한 내용으로 옳지 않은 것은?

- ① 1D 컨브넷은 특정 자연어 처리 같은 일부 문제에 RNN을 대신할 수 있는 빠른 모델이다.
- ② 1D 컨브넷은 비용이 적게 들고 RNN은 비용이 많이 든다.
- ③ CNN과 RNN은 연결에서 1D 컨브넷을 전결합 단계로 사용한다.
- ④ CNN은 시퀀스 길이를 줄이고 RNN이 처리할 유용한 표현을 추출한다.
- ⑤ 온도 예측문제에서 CNN과 RNN을 연결하는 것이 CNN만 쓰는 것보다 낫다.