

산업인공지능 개론

Home Work #3

퀴즈 문제 풀이

학과 : 산업인공지능학과

학번 : 2024254022

이름 : 정현일

2024.04.29.

Quiz

❖ 일반 신경망과 딥러닝 신경망에 대한 설명으로 옳지 않은 것은?

- ① 일반 신경망인 다층 퍼셉트론에서 은닉층의 개수는 2~3개 정도로 적다.
- ② 일반 신경망에서 특징 추출을 별도로 할 경우 개발자가 이를 처리해줘야 한다.
- ③ 딥러닝 신경망에서는 데이터에 대한 특징 추출 방법이 학습을 통해서 결정될 수 있다.
- ④ 기울기 소멸 문제는 계단 모양 활성화 함수를 사용하여 완화시킬 수 있다.

-> 기울기 소멸 문제를 완화하기 위해 ReLU 같은 비선형 활성화 함수를 사용

❖ 다음 신경망의 학습에 대한 설명으로 옳지 않은 것은?

- ① 시그모이드 함수의 미분값은 0.25이하인 양의 값이기 때문에, 이를 활성화 함수로 사용하면 기울기 소멸 문제가 발생할 수 있다.
- ② ReLU 함수의 출력값은 0이 될 수 없다.
- ③ 제이비어(Xavier) 기법을 사용할 때 가중치의 초기값은 층(layer)에 있는 노드 개수에 영향을 받는다.
- ④ 동일한 학습 알고리즘을 적용하더라도 신경망에 있는 가중치의 초기값은 성능에 영향을 크게 줄 수 있다.

-> ReLU함수는 입력값이 0 이하일 때는 0을 출력

❖ 딥러닝 모델의 성능을 향상시키기 위해 사용하는 기법은 아닌 것은?

- ① 데이터 확대
- ② 정규화
- ③ 배치 정규화
- ④ 특성 삭제 -> 특성의 삭제는 과적합 방지를 위한 방법인것 같음

❖ 일반신경망과 딥러닝 신경망에 대한 설명으로 옳지 않은 것은?

- ① 딥러닝 신경망은 층의 수가 많을수록 과적합에 취약하다.
- ② 일반 신경망은 과적합에 상대적으로 덜 취약하다.
- ③ 딥러닝 신경망은 자동으로 특징을 추출한다.
- ④ 일반 신경망은 깊은 구조를 가진다. -> 딥러닝 신경망이 깊은 구조를 가짐

❖ 일반 신경망과 딥러닝 신경망의 주요 차이점은?

- ① 사용되는 뉴런의 수
- ② 네트워크의 깊이. -> 여러개의 층으로 신경망을 이루고 있음
- ③ 활성화 함수의 종류
- ④ 그래디언트 소멸 문제의 발생

❖ 과적합에 대한 설명으로 옳지 않은 것은?

- ① 과적합 상태이면 테스트 데이터에 대한 성능이 학습 데이터에 대한 성능보다 좋다.
- ② 오차함수를 오차항과 모델 복잡도항으로 구성함으로써 과적합을 완화시킬 수 있다.
- ③ 학습할 때 일정 확률로 노드들을 무작위로 선택하여 해당 노드에 대한 연결선이 없는 것처럼 학습에 배제하는 드롭아웃은 과적합 해소에 도움이 된다.
- ④ 미니배치 단위로 가중치를 갱신하는 학습을 하면 과적합을 완화시키는데 도움이 될 수 있다.

-> 테스트 데이터나 실제 상황에서의 성능은 상대적으로 떨어질 수 있어서 틀림

❖ 다음 가중치 학습 기법에 대한 설명으로 옳지 않은 것은?

- ① 딥러닝 신경망의 학습 알고리즘은 기본적으로 경사 하강법에 기반한다.
- ② 모멘텀 사용 경사 하강법에서는 직전 시점의 가중치 갱신 정보를 일부 활용하여 가중치를 갱신한다.
- ③ Adadelata 알고리즘은 가중치별로 학습율에 다르게 적용될 수 있도록 한다.
- ④ 현재 알려진 가중치 학습 기법 중 ADAM이 가장 성능이 우수한 기법이다.

-> 모든 경우에서 가장 우수한 성능을 보일것 같지 않음

❖ 기울기 소멸 문제는 주로 어떤 모델에서 발생하는가?

- ① 얇은 신경망
- ② 깊은 신경망 -> 여러개의 층을 지나가면서 발생하므로 정답
- ③ 단일 계층 신경망
- ④ SVM

❖ 어떤 모델이 주로 복잡한 계층적 특징을 학습하는 데 적합한가?

- ① 얇은 신경망
- ② 깊은 신경망 -> 복잡한 계층은 깊은 신경망으로 학습하는게 맞는것 같음
- ③ 단일 계층 신경망
- ④ 선형 회귀 모델

❖ 딥러닝 신경망에서는 주로 어떤 유형의 활성화 함수를 사용하여 그래디언트 소멸 문제를 완화하는가?

- ① Sigmoid 함수
- ② Step 함수
- ③ ReLU -> 기울기 소실 문제는 ReLU임
- ④ 선형 함수

Quiz

❖ 기울기 소멸 문제가 발생할 때, 가중치 업데이트는 어떤 특성을 보이는가?

① 지나치게 큰 업데이트

② 매우 작은 업데이트 -> 0에 가까운 값이 되어 작은 업데이트 됨

③ 무작위한 업데이트

④ 일정한 크기의 업데이트

❖ 기울기 소멸 문제는 어떤 활성화 함수와 함께 발생할 가능성이 높은가?

① ReLU

② Leaky ReLU

③ Sigmoid -> 0~1로 큰값이나 작은 값일 경우 기울기가 0에 가까워 지는 문제가 있음

④ Softmax

❖ 어떤 활성화 함수가 입력 값의 모든 부분에서 미분 가능한가?

① ReLU

② Leaky ReLU

③ Sigmoid -> 연속의 곡선으로 되어 있어서 모든 부분에서 미분이 가능

④ Step function

❖ 신경망에서 초기 가중치 설정의 중요성에 대한 잘못된 설명은?

① 초기 가중치는 학습의 시작점을 결정한다.

② 너무 큰 초기 가중치는 학습 과정을 불안정하게 만들 수 있다.

③ 모든 가중치를 같은 값으로 초기화하는 것이 바람직하다. -> 바람직하지 않음

④ 초기 가중치는 무작위로 설정되거나 특정 기법을 사용하여 초기화된다.

❖ 신경망에서 배치 크기(Batch Size)의 역할에 대한 잘못된 설명은?

① 한 번에 처리하는 데이터의 수를 결정한다.

② 너무 크면 학습 속도가 느려질 수 있다.

③ 아주 작게 하면 모델의 일반화 성능이 향상된다. -> 배치사이즈는 적절한 사이즈로 해야함

④ 메모리 용량과 계산 속도에 영향을 준다.

❖ 신경망에서 손실 함수(Loss Function)의 목적에 대한 잘못된 설명은?

① 예측값과 실제값 간의 차이를 측정한다.

② 학습 과정에서 최소화되어야 한다.

③ 손실 함수의 선택은 문제 유형에 따라 달라진다.

④ 손실 함수의 값은 학습 과정에 영향을 주지 않는다. -> 중요한 역할을 함

❖ 기울기 소멸 문제가 주로 발생하는 경우는?

① 층이 많은 신경망에서 -> 층이 많을 때 발생

② 학습률이 너무 높을 때

③ 가중치가 무작위로 초기화될 때

④ 활성화 함수로 ReLU를 사용할 때

❖ 기울기 소멸 문제를 일으키는 활성화 함수는?

① ReLU

② 시그모이드 -> 0~1로 큰값이나 작은 값일 경우 기울기가 0에 가까워 지는 문제가 있음

③ Leaky ReLU

④ ELU

❖ 기울기 소멸 문제를 완화하는 방법 중 가장 잘못된 것은?

① 적절한 가중치 초기화 방법 사용

② 활성화 함수로 ReLU 계열 함수 사용

③ 학습률을 점차 증가시키는 방법 -> 오히려 학습률을 감소 시켜야 기울기 소실 문제를 해결

④ 배치 정규화(Batch Normalization)

❖ 가중치 초기화에서 0으로 초기화하는 방법의 문제점은?

① 신경망이 대칭적인 가중치를 갖게 되어 학습이 제대로 이루어지지 않는다.

② 0으로 초기화하면 뉴런의 출력이 항상 0이 되어 학습이 이루어질 수 없다. -> 초기값 0은 학습이 이루어지지 않음

③ 가중치 감소(weight decay) 기법을 적용할 때 문제가 발생할 수 있다.

④ 모든 가중치가 같은 값을 갖기 때문에 기울기 소멸 문제가 발생한다.

❖ Xavier 초기화 방법의 주된 특징은?

① 가중치를 일정 범위 내에서 균등하게 분포시키는 방법이다.

② 초기 가중치를 큰 값으로 설정하여 활성화 함수의 비선형 영역을 활용한다.

③ 각 층의 입력 노드의 개수에 따라 분산의 크기를 조정한다. -> 출력도 포함 되어야 함.

④ 무작위성을 최대화하여 모든 가중치가 다른 값을 갖도록 한다.

❖ He 초기화 방법에서 사용하는 분산의 계산에 사용되는 요소는?

① 은닉층의 노드 수

② 학습률

③ 각 층의 입력 노드 수 -> 정답

④ 가중치의 개수

Quiz

❖ L1 규제화는 어떤 특징을 가지고 있는가?

- ① 가중치의 제곱을 규제
- ② 가중치의 절댓값을 규제 -> 절댓값 합을 더해주는 방식으로 작동
- ③ 가중치의 합을 규제
- ④ 가중치의 제곱근을 규제

절댓값 합을 더해주는 방식으로 작동

❖ 규제화는 어떤 상황에서 효과적인가요?

- ① 훈련 데이터가 충분할 때
- ② 모델이 너무 단순할 때
- ③ 모델이 너무 복잡할 때 -> 모델이 너무 복잡하면 과적합(overfitting) 문제가 발생
- ④ 훈련 데이터가 아주 적을 때

❖ 규제화를 사용할 때 주의해야 할 점은?

- ① 너무 강한 규제는 과소적합을 일으킬 수 있음 -> 규제를 너무 강하게 사용하면 모델이 데이터를 충분히 학습하지 못해 성능이 저하
- ② 규제는 항상 모델의 성능을 향상시킴
- ③ 규제는 항상 모델의 학습 속도를 늦춤
- ④ 규제는 모든 문제에 적합하다

❖ 배치 정규화는 어떤 값을 이용해 정규화하는가?

- ① 한 배치 내의 데이터들의 중앙값
- ② 전체 데이터의 평균값
- ③ 한 배치 내의 데이터들의 평균값 -> 정답
- ④ 전체 데이터의 중앙값

❖ 배치 정규화에서 발생하는 연산이 아닌 것은?

- ① 스케일(scale) 조정
- ② 이동(shift)
- ③ 회전(rotation) -> 배치 정규화 연산이 아님
- ④ 정규화(normalization)

❖ 배치 정규화의 장점은?

- ① 더 큰 학습률을 사용할 수 있게 해줌 -> 장점임
- ② 가중치 초기화의 영향을 줄여줌 -> 장점임
- ③ 내부 공변량 변화를 완화 -> 장점임
- ④ 모델의 크기를 물리적으로 줄여줌

❖ 배치 정규화(Batch Normalization)의 주된 목적은?

- ① 과적합 방지
- ② 학습 속도 개선 -> 정답
- ③ 모델의 계산 복잡도 감소
- ④ 모델의 크기 축소

❖ 배치 정규화는 어느 위치에 주로 적용되는가?

- ① 활성화 함수 전 -> 정답
- ② 활성화 함수 후
- ③ 손실 함수 전
- ④ 최적화 함수 후

❖ 배치 정규화에서 계산할 필요가 있는 것은?

- ① 평균과 분산 -> 입력 데이터의 평균과 분산을 계산하여 정규화함
- ② 가중치와 편향
- ③ 학습률과 모멘텀
- ④ 손실 값과 그래디언트 값

❖ 경사하강법(Gradient Descent)의 핵심 아이디어는?

- ① 손실 함수의 최대값을 찾는 것
- ② 모든 가중치를 무작위로 조정하는 것
- ③ 손실 함수의 그래디언트를 사용하여 가중치를 업데이트하는 것 -> 정답
- ④ 모든 샘플에 대해 개별적으로 가중치를 업데이트하는 것

❖ Momentum 방법을 사용할 때, 어떤 변수가 이전 그래디언트 값을 사용하는가?

- ① 학습률
- ② 가중치
- ③ 손실 함수
- ④ 속도(velocity) -> 이전 그래디언트 사용하는 값은 속도임

❖ AdaGrad의 특징은?

- ① 과거의 모든 그래디언트의 제곱을 누적한다. -> AdaGrad의 특징임
- ② 속도와 방향을 모두 고려한다.
- ③ 학습률을 일정하게 유지한다.
- ④ 모멘텀만을 사용하여 가중치를 업데이트한다.

Quiz

❖ RMSProp의 핵심 아이디어는?

- ① 과거 그레디언트의 제공의 이동 평균을 유지한다.
- ② 모든 가중치 업데이트는 동일한 크기로 수행된다.
- ③ 학습률이 시간에 따라 증가한다.
- ④ 경사하강법과 동일한 방식으로 작동한다.

-> 1번 정답 인것 같음

❖ Adam는 어떤 최적화 기법의 조합으로 볼 수 있는가?

- ① RMSProp + AdaGrad
- ② RMSProp + Momentum
- ③ Gradient Descent + Momentum
- ④ AdaGrad + Gradient Descent

-> 2번 정답 인것 같음

❖ 미니배치 경사하강법은 어떤 방식으로 데이터를 처리하는가?

- ① 한 번에 하나의 샘플만 처리한다.
- ② 전체 데이터 세트를 한 번에 처리한다.
- ③ 데이터 세트의 작은 부분집합을 한 번에 처리한다.
- ④ 데이터의 순서를 고려하지 않고 무작위로 처리한다.

-> 3번 정답 인것 같음

❖ 어떤 최적화기가 학습률을 동적으로 조정하는 기능을 가지고 있는가?

- ① Gradient Descent
- ② Momentum
- ③ AdaGrad
- ④ Batch Gradient Descent

-> 학습률을 동적으로 조정 기능이 있는 것은 AdaGrad 임

❖ 최적화기(optimizer)의 학습률을 너무 높게 설정하면 어떤 문제가 발생할 수 있는가?

- ① 모델이 과소적합될 수 있다.
- ② 손실 함수의 최소값에 수렴하기 위해 더 많은 시간이 걸릴 수 있다.
- ③ 손실 함수의 최소값 주변에서 진동할 수 있다.
- ④ 학습이 전혀 진행되지 않는다.

-> 3번 정답 인것 같음

❖ saddle point(안장점)에 빠질 경우, 어떤 최적화기가 이를 해결하는 데 가장 도움이 안될까?

- ① Gradient Descent
- ② Momentum
- ③ Batch Gradient Descent
- ④ AdaGrad

-> Momentum,AdaGrad 최적화 방법이 안장점을 벗어나기 더 효과적임

Quiz

❖ 학습 데이터의 특징 Z 가 평균 0, 표준 편차 1로 정규화되어야 할 때, 원본 데이터 값 $[2, 8, -1, 4]$ 를 정규화하고 변환된 값을 계산하세요.

- 평균 계산 : $2+8+(-1)+3/4 = 13/4 = 3.25$

- 표준 편차 :

$$\sqrt{\frac{(2 - 3.25)^2 + (8 - 3.25)^2 + (-1 - 3.25)^2 + (4 - 3.25)^2}{4}}$$

$$\sqrt{\frac{42.75}{4}} = \sqrt{10.6875} \approx 3.27$$

- 정규화

$$z1 = 2-3.25/3.27 = -0.38$$

$$z2 = 8-3.25/3.27 = 1.45$$

$$z3 = -1-3.25/3.27 = -1.30$$

$$z4 = 4-3.25/3.27 = 0.23$$

❖ 학습률이 0.01이고, 현재 가중치가 0.5일 때, 손실 함수의 그래디언트가 0.3으로 계산되었을 때, 다음 단계의 가중치를 계산하시오.

- 가중치

$$0.5-0.01 \times 0.3 = 0.497$$

❖ 데이터 배치가 $[1, 3, 5, 7]$ 로 주어졌을 때, 이 데이터의 평균과 표준편차를 계산하시오.

- 평균 계산 : $1+3+5+7/4 = 16/4 = 4$

- 표준 편차 :

$$1\text{의 경우: } (1 - 4)^2 = (-3)^2 = 9$$

$$3\text{의 경우: } (3 - 4)^2 = (-1)^2 = 1$$

$$5\text{의 경우: } (5 - 4)^2 = 1^2 = 1$$

$$7\text{의 경우: } (7 - 4)^2 = 3^2 = 9$$

$$9+1+1+9/4 = 5$$

$$\text{sqrt}(5) = 2.236$$

Quiz

- ❖ 위에서 계산한 평균과 표준편차를 사용하여 각 데이터 포인트를 정규화하시오.
 - 1에 대한 정규화 : $1-4/2.236 = -1.34164079$
 - 3에 대한 정규화 : $3-4/2.236 = -0.44721360$
 - 5에 대한 정규화 : $5-4/2.236 = 0.44721360$
 - 7에 대한 정규화 : $7-4/2.236 = 1.34164079$

- ❖ 위에서 정규화된 데이터에 학습 가능한 매개변수 $\gamma=2$ 와 $\beta=0.5$ 를 적용하여 출력을 계산하시오.
 - $2 \cdot (-1.34164079) + 0.5 = -2.18328157$
 - $2 \cdot (-0.44721360) + 0.5 = -0.39442719$
 - $2 \cdot 0.44721360 + 0.5 = 1.39442719$
 - $2 \cdot 1.34164079 + 0.5 = 3.18328157$