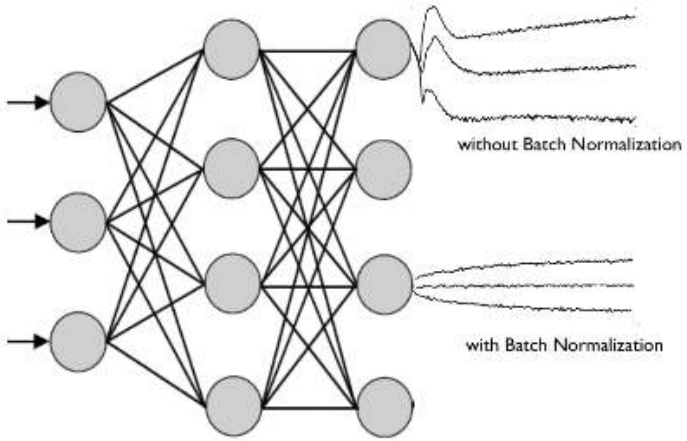
11주차 조별보고서 (Default)	
작성일: 2019년 11월 16일	작성자: 이재은
조 모임 일시: 2019년 11월 16일 9교시	모임장소: 백년관 스터디룸
참석자: 위성조, 이충현, 최진성, 이재은, 김영연	조원: 위성조, 이충현, 최진성, 이재은, 김영연

구 분	내 용
학습 범위와 내용	**5.1절:** 목적 함수로서 평균제곱 오차의 단점을 지적하고, 딥러닝이 많이 활용하는 교차 엔트로피와 로그우도를 소개한다. **5.2절:** 스토캐스틱 경사 하강법의 성능 향상에 효과적인 전처리, 가중치 초기화, 모멘텀, 적응적 학습률, 활성함수, 배치 정규화를 설명한다.
논의 내용	**Q1.** 원 핫 인코딩을 적용할 경우 요소가 늘어나게 된다면 인코딩 되는 라인 또한 끝없이 늘어나게 될 것인데, 이렇게 되면 원 핫 보다는 일반 숫자로 표기하는 것이 더 효율적이지 않은가 생각됩니다. **A1.** 원 핫 인코딩은 요소가 많을 때 사이즈가 급격히 늘어난다는 단점이 있다. 이런 단점을 해결하기 위해서 원 핫 인코딩 방식으로 표현된 벡터를 embedding 시켜 사용한다. embedding 방식이란 요소의 숫자가 늘어나도 차원이 늘리지 않고 고정하는 방식이다. 대표적인 embedding방식으로는 word embedding이 있다. 이는 비슷한 분포를 가진 단어의 주변 단어들은 비슷한 의미를 가진다는 것을 의미한다. 즉 word embedding에서는 원 핫 인

	코딩과 달리 하나의 단어다 미리 정의된 차원에서 연속형의 값을 갖는 벡터로 표현된다. 간단히 말하자면 word를 n차원의 vector로 mapping시키는 것이다. 따라서 embedding을 시킨다면 요소가 늘어나더라도 원 핫 인코딩을 사용하는 것이 효율적이다.
	Q2. 원핫 코드를 부여할 때, 왜 성별에 2비트가 필요한지 모르겠습니다. **A2.** 데이터 전 처리 시, 원핫 코드는 값의 개수만큼 비트를 부여한다. 따라서, 성별은 남과 여 2개이므로 2비트인 것 이다. 아마 강의 자료에서 갑자기 1에서 (1,0)으로 되어있는 부분은 인코딩을 얘기하는 것 같다. 인코딩 할때, 정수 인코딩의 경우 표현하고 싶은 단어의 인덱스 위치에 1을 부여하고 그 외는 0을 부여한다. 일종의 단어의 벡터 표현 방식으로 볼 수 있다. 덧붙여, 이런 방식은 단어의 의미 또는 개념 차이를 전혀 담지 못하는 단점이 있다.
	Q3. 가중치 초기화를 공부하면서 찾아보니 초기 값을 작은 값으로 하지 않는 경우 절대 값이 크고 음수일 경우 dead ReLU가 발생한다고 한다. Dead ReLU에 대해 알고 싶다. **A3.** Dead ReLU란 학습 도중 0보다 낮은 값으로 Gradient 값이 진행되어 활성화 값이 0으로 고정된 상태를 의미한다. 뉴런의 활성화 값이 0으로 바뀔 경우 다음 레이어 층으로 연결되는 가중치를 곱하더라도 결과가 항상 0으로 나타나기에 해당 뉴런을 통한 학습은 불가능해진다. 이는 계산을 줄여주는 효과를 가지기에 연산량을 절감하는데 효과를 가지고 있으나 사용된 뉴런들의 상당한 부분이 Dying ReLU 현상을 일으키게 되면 학습의 효율 또한 급격하게 낮아지기에 보통 이를 보완한 Leaky ReLU를 사용하는 방법을 대안으로 채택한다.

Q4. 배치 정규화와 내부 공변량 변화간의 관계에서 배치 정규화의 stablization은 정말로 공변량 시프트 현상을 누그러뜨리는지 자세히 알고 싶다.

A4. 배치 정규화의 경우, 깊은 레이어의 입력값의 변화(공변량 변화)를 줄이기 위해, 활성함수 입력값, 혹은 활성함수 출력값을 대상으로 평균 0, 분산 1이 되도록 정규화 시키는 것을 의미한다.



그러나 시그모이드 함수의 경우, 평균 0, 분산1을 갖는 입력값의 경우, [-1, 1]의 범위를 가지는데, 이 범위의 sigmoid 함수 값은 0.5를 중심으로 0.2~0.7사이의 거의 직선인 선형 영역이다. 따라서 활성함수의 가장 중요한 역할인 비선형성을 잃어버릴 수 있어서,

	이 점을 보완하기 위해 정규화된 값에 scale 팩터(γ)와 shift 팩터(β)을 도입하고 정규화된 값에 곱하고 더해준다. 두 인자는 오차 역전파 과정에서 학습해 준다. 이를 통해 활성함수로 들어가는 값의 범위를 바꾸어줌에 의해 비선형성을 부여할 수 있다. $$BN(x_i) = \gamma(\frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}) + \beta$$
질문 내용	모두 해결하였습니다.
기타	

<첨부 개인 레포트>

이름	위성조	학번	201402033

구분	내용
학습 범위	5.1 목적함수: 교차 엔트로피와 로그우도 5.2 성능향상을 위한 요령
학습 내용	과학과 공학에서의 최적화 (웨이퍼에 최대 집적하는 반도체공학, 목적지까지의 최단 경로 찾기 등) 목적함수를 정의하고 최적해를 구함. 기계 학습의 최적화는 그것이 고차원에서 이뤄져야 하고, 현장에서 발생하는 새로운 샘플을 잘 예측해야 하는 문제(높은 일반화 능력 요구) 때문에 훨씬 어렵다. (목적함수의 비볼록 성질, 고차원 특징 공간, 데이터의 희소성, 높은 컴퓨팅 파워 요구) 목적함수로서의 MSE(평균제곱오차) $$e = \frac{1}{2}\|\mathbf{y} - \mathbf{o}\|_2^2$$ MSE는 오차가 클수록 e값이 커지므로 목적함수로 적합하나, 오차가 큰 경우에 오히려 기울기가 작아지는 경우가 있음. 활성함수로 사용되는 시그모이드 함수의 특징 때문에, 파라메터의 절대값이 크면 오차가 커도, 그레이디언트가 작게 되는 효과가 생긴다. -> 학습 속도를 떨어뜨리는 부정적 효과

교차 엔트로피 목적함수

$$e = -(y \log_2 o + (1-y) \log_2 (1-o)), \qquad \text{이때, } o = \sigma(z) \text{이고} z = wx + b$$

평균제곱오차 목적함수와 다르게 오류가 더 큰 쪽에 대하여 더 큰 그레이디언트를 도출한다.

Softmax - (자신의 오차/전체 오차의 합)을 구한다고 보면 된다.

$$o_j = \frac{e^{s_j}}{\sum_{i=1,c} e^{s_i}}$$

로그우도 목적함수

$$e = -\log_2 o_y$$

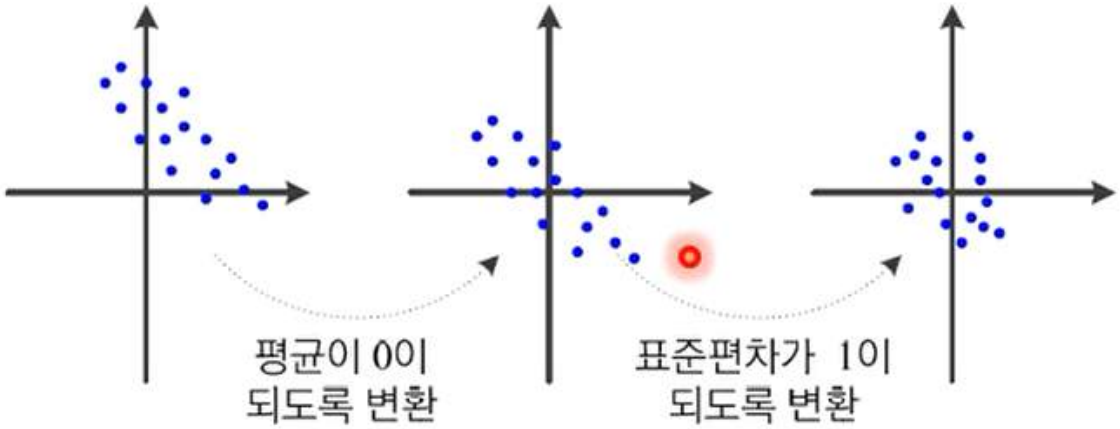
모든 출력 노드값을 사용하는 MSE나 교차 엔트로피와 달리 Oy라는 하나의 노드만 사용,

Oy는 샘플의 레이블에 해당하는 노드의 출력값

Softmax와 로그우도

Softmax는 최댓값이 아닌 값을 억제하여 0에 가깝게 만든다는 의도를 내포하고 있으며, 학습 샘플이 알려주는 부류에 해당하는 노드만 보겠다는 로그우도와 잘 어울려, 같이 사용하는 경우가 많다.

	데이터 전처리 규모(scale) 문제 : ex) 건강에 관련된 데이터(키(m), 몸무게(kg), 혈압)의 경우, 키가 33cm가 차이나더라도 특징값 차이는 0.33에 불과한 반면, 몸무게는 특징값 차이가 크므로, 이런 경우 스케일 차이로, 키에 관련한 가중치가 늦게 학습되는 결과가 나온다. 모든 특징이 양수인 경우의 문제 : 특징이 모두 양수일 때, 어떤 노드에 연결된 가중치가 모두 증가 혹은 감소하는 현상이 발생하여, 학습을 느리게 만든다. 정규화는 규모 문제와 양수 문제를 해결해 줌. $$x_i^{new} = \frac{x_i^{old} - \mu_i}{\sigma_i}$$ 명칭값(norminal)을 원핫(one-hot) 코드로 변환 원핫 코드는 값의 개수만큼 비트를 부여 – 성별은 2비트, 체질은 4비트 부여 대칭적 가중치 문제 같은 가중치를 입력받는 노드는 같은 값으로 갱신되며 이러한 노드들은 같은 작업을 반복 수행하는 효과를 가져오므로, 초기 가중치를 난수로 초기화함으로써 대칭을 파괴.



(가중치를 모두 0으로 초기화 하는 경우 학습 자체가 진행되지 않음)

난수로 가중치를 초기화 할 경우, 가우시언 분포 또는 균일 분포에서 난수를 추출한다.

(두 분포 사이의 성능 차이는 거의 없음)

$$r = \frac{1}{\sqrt{n_{in}}}$$

$$r = \frac{\sqrt{6}}{\sqrt{n_{in} + n_{out}}}$$

난수 범위가 매우 중요하며 또는를 이용하여 r을 구하고,

[-r, r] 범위에서 난수를 생성한다.

모멘텀

기계 학습은 훈련집합을 이용하여 그레이디언트를 추정하므로 잡음 가능성이 높음

그레이디언트에 스무딩을 가하여 잡음의 영향을 줄여 수렴 속도를 향상시킨다.

$$\left.\begin{aligned}\mathbf{v} &= \alpha\mathbf{v} - \rho\frac{\partial J}{\partial \boldsymbol{\Theta}} \\ \boldsymbol{\Theta} &= \boldsymbol{\Theta} + \mathbf{v}\end{aligned}\right\}$$

속도 벡터 V는 이전 그레이디언트를 누적한 것에 해당(V 초기값은 0)

a가 0이면 모멘텀이 적용 안된 이전 공식과 같고 a가 클수록 이전 그레이디언트 정보에 큰 가중치를 주는 것과 같다.

네스테로프 모멘텀 – 기존 속도 벡터 V대로 움직였다고 가정한 위치에서의 그레이디언트를 계산하여 그 지점으로 이동.

학습률의 중요성 – 속도와 진자 현상 사이의 문제

적응적 학습률은 매개변수마다 자신의 상황에 따라 학습률을 조정해 사용

선형 활성함수를 쓰는 경우에는 여러 단의 딥 네트워크를 사용할 이유가 없음.(합쳐저서 한 층의 네트워크와 다를 바가 없음)

배치 정규화 – 공변량 시프트 현상을 누그러뜨리기 위해 정규화를 모든 층에 적용하는 기법

$$x_i^{new} = \frac{x_i^{old} - \mu_i}{\sigma_i}$$

훈련집합 전체보다는 미니배치에 적용하는 것이 유리함.

	CNN에서는 노드 단위가 아닌 특징 맵 단위로 적용함. 배치 정규화의 효과 가중치 초기화에 덜 민감함 학습률을 크게 하여 수렴 속도 향상 가능 시그모이드를 활성함수로 사용하는 깊은 신경망도 학습이 이루어짐 드롭아웃이라는 규제 기법을 적용하지 않아도 높은 성능을 보임.
질문 내용	원핫 코드를 부여할 때, 왜 성별에 2비트가 필요한지 모르겠습니다. 정보가 2개이니 1비트로 충분할 것 같은데, 특별한 이유가 있는것인지 알고싶습니다.

학번	201402665	**이름**	이충현
구분	**내용**		
학습 범위	기계학습 5장 딥러닝 최적화 **5.1절** 교차 엔트로피와 로그우드 소개 **5.2절** 스토캐스틱 경사 하강법의 성능 향상에 효과적인 전처리, 가중치 초기화, 활성함수, 배치 정규화		
학습 내용	최적화는 훨씬 복잡하다 훈련집합이 테스트집합의 대리자 역할 수행. MSE, 로그우드와 같은 목적함수가 궁극 목표인 정확률의 대리자 역할을 한다. 어려운 이유는 목적함수의 convex 성질, 고차원 특징 공간, 데이터의 희소성, 긴 시간 소요 **<5.1 절 목적함수: 교차 엔트로피와 로그우드 소개>** ● 그레디언트를 계산 시 더 많은 오류를 범한 상황이 더 낮은 벌점을 받을 수 있다는 오류를 발생할 수 있다. ● 이는 wx+b 가 커지면 그레이언트가 작아지기 때문이다. ● MSE 의 단점은 피드백이 커도 시그모이드 함수의 역할 때문에 앞단으로 전달되는 rate 의 업데이트가 조금씩 밖에 안된다. 교차 앤트로피		

■ 큰 허점

- 식 (5.3)의 그레이디언트가 벌점에 해당

$$e = \frac{1}{2}(y-o)^2 = \frac{1}{2}(y-\sigma(wx+b))^2$$

$$\left.\begin{aligned}\frac{\partial e}{\partial w} &= -(y-o)x\sigma'(wx+b)\\ \frac{\partial e}{\partial b} &= -(y-o)\sigma'(wx+b)\end{aligned}\right\}$$

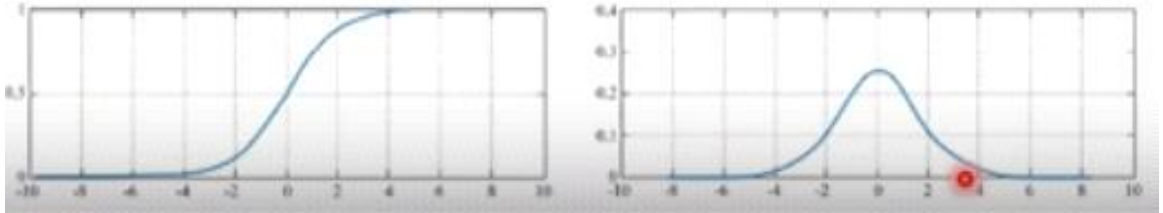


그림 5-2 로지스틱 시그모이드함수와 도함수

● 앤트로피 = 정보량의 기댓값, 교차 앤트로피 = 서로 다른 두 분포가 있을 때 틀릴 수 있는 정보량을 고려한 최적으로 인코딩할 수 있게 해주는 정보량(p-정답 레이블, q-신경망 출력)

$$H(p,q) = -\sum_{x} p(x)\,\log q(x).$$

$$e = -\sum_{i=1,c} (y_i \log_2 o_i + (1-y_i)\log_2(1-o_i))$$

● 교차 엔트로피 목적함수에서 c 개의 출력 노드를 가진 경우로 확장하면

Softmax 활성함수와 로그우드 목적함수

● Softmax는 max를 모방(출력 노드의 중간 계산 결과 최댓값은 더욱 활성화하고 작은 값은 억제). 모두 더하면 1 이 되어 확률 모방 발생.

	● 로그우도 목적함수는 모든 출력 노드값을 사용하지 않고 하나의 노드만 사용한다. Softmax 는 최댓값이 아닌 값을 억제하여 0 에 가깝게 만든다는 의도 내포한다. 학습 샘플이 알려주는 부류에 해당하는 노드만 보겠다는 로그우드와 잘 어울림. 따라서 둘이 결합하여 사용하는 경우가 많다. **<5.2 절 성능 향상을 위한 요령>** 1)데이터 전처리 규모 문제 = 모든 특징이 양수일때, 이처럼 뭉치로 증가 또는 감소하면 최저점을 찾아가는 경로가 갈팡질팡하여 느린 수렴 발생 à정규화는 규모 문제와 양수 문제 해결. 명칭값을 원핫 코드로 변환 à 거리 개념이 없음. 값의 개수만큼 비트를 부여를 받는다. 1. 가중치 초기화 처음에 0 이 되면 문제발생. 다 균일한 값을 주면 backprogation 시 두 노드가 같은 일을 하는 중복성 발생한다. 따라서 난수로 가중치를 초기화한다. 가우시언 분포 또는 균일 분포에서 난수를 추출한다. 2. 모멘텀 기계학습은 훈련 집합을 이용하여 그레디언트의 잡음 가능성이 높다. 모멘텀은 그레디언트 스무딩을 가하여 잡음을 줄인다. à 수렴 속도 빨라진다. 모멘텀의 효과는 오버슈팅 현상을 누그러뜨린다. 3. 적응적 학습률 매개변수마다 자신의 상황에 따라 학습률을 조절해 사용한다. 4. 활성함수 Tanh 는 활성값이 커지면 포화 상태가 되고 그레디언트는 0 에 가까워진다. 매개변수 갱신이 매우 느린 요인이다. Relu 함수 살짝 negative 하게 하려면 Leaky Relu 사용한다. 5. 배치 정규화 공변량 시프트 현상을 누그러뜨리기 위해 정규화를 모든 층에 적용하는 기법이다. 최적화를 마친 후 추가적인 후처리 작업이 필요하다. 규제 효과를 제공한다(드롭아웃을 적용하지 않아도 높은 성능을 지닌다).
질문 내용	배치 정규화와 내부 공변량 변화간의 관계에서 배치 정규화의 stablization 은 정말로 공변량 시프트 현상을 누그러뜨리는지 자세히 알고 싶다.

학번	201403474	이름	최진성

구분	내용
학습 범위	기계학습 5장 딥 러닝 최적화 5.1 목적함수 : 교차 엔트로피와 로그 우도 5.2 성능 향상을 위한 요령
학습 내용	기계학습 5장 딥 러닝 최적화 딥 러닝의 최적화 ● 과학과 공확에서 최적화 ● 기계학습에 비해서 딥 러닝의 최적화가 훨씬 간단함 ➔일반화 능력이 뛰어나야 함을 전제로 최적화 ➔훈련집합과 테스트집합 간의 관계(대리자 관계)가 최적화를 어렵게 하고 있음 ➔목적함수의 비블록적인 성질, 고차원 특징 공간, 데이터의 희소성 등 ➔긴 시간이 소요 5.1 목적함수 : 교차 엔트로피와 로그 우도 MSE 목적함수(평균 제곱 오차) -> 오차가 클수록 e의 값이 커지니 벌점으로 표기하기는 알맞음 한계 : 벌점의 학습이 더디기 때문에 벌점의 상대적인 수치가 적절하지 않을 수 있음. Ex : ➔식에 의하면 왼쪽의 e는 0.2815, 오른쪽의 e는 0.4971이므로 오른쪽이 더 많은 벌점을 받아야 하나 Gradient를 계산하면 왼쪽 상황의 Gradient가 더 크다. 교차 엔트로피 목적함수 확률 분포 P는 정답 레이블, Q는 신경망 출력 결과라고 할 때 H(P, Q) = ● 즉, H(P, Q)와 H(Q, P)의 차이를 줄이는 것이 목표 y가 1, o가 0.98일 경우 e = 0.0291로써 예측이 잘 된 경우 하지만 o가 0.0001일 경우 e = 13.2877로써 예측이 안되었다고 판별할 수 있다. 이를 이용하면 MSE 목적함수에서 나타난 문제를 해결할 수 있다.

Softmax 활성함수
Max를 모방하여 만들어진 함수로, 최댓값은 더욱 활성화되고 작은 값은 억제되는 특징을 가진다.
Softmax 활성함수의 식 :
값을 모두 더하면 1이 되어 확률을 모방한다는 특징을 가지고 있다.
Ex : 출력층의 시그모이드 함수 계산 결과 0.8808, 0.7685, 0.9820가 나왔다고 한다면 max를 적용할 경우 0, 0, 1이 출력되지만 softmax는 0.1131, 0.0508, 0.8360의 형식으로 합이 1 이내에서 값을 분류한다.

로그우도 활성함수
출력 노드는 모든 노드를 사용하지 않고 하나만 사용한다.
학습 샘플이 알려주는 부류에 해당하는 노드만 보겠다는 의미
● 최댓값이 아닌 값을 최대한 억제하는 softmax와 같이 사용하기 좋음.

로그우도 활성함수의 식 :
Softmax의 결과로 나온 값을 에적용하였을경우3.1443, 4.2990, 0.2584가 출력되는데, 오류가 높을수록 큰 숫자가, 낮을수록 작은 숫자가 출력된다.

5.2 성능 향상을 위한 요령

경험규칙
경험에 의하여 얻어진 지식, 혹은 법칙을 의미한다.

데이터의 전처리
데이터는 각 종류별로 데이터의 규모를 같은 선상에서 가늠하기 힘들다. 키와 몸무게를 기준으로 하였을 때 m와 kg를 비교할 경우 kg가 100배 정도 더 큰 규모이기 때문에 첫 번째 특징의 가중치는 100배 정도 느리게 학습된다.
모든 특징이 양수일 경우에는 데이터가 한 번에 뭉쳐져서 증가, 혹은 감소하기 때문에 최저점을 찾아가는 경로가 갈팡질팡하다.
-> 데이터 정규화
데이터 정규화 식 :

데이터 정규화의 데이터 변환 형태

데이터의 명칭을 원 핫 코드로 변환
1.755m, 65.5kg, 122혈압, 남자, 소양인 샘플
-> 1.755, 65.5, 122, 1, 0,[남 여], 0, 0, 1, 0[태양인, 태음인, 소양인, 소음인]

대칭적 가중치
가중치가 아래위로 대칭적으로 같을 경우 데이터가 가중치를 따라 다른 노드에서 같은 일을 하는 중복성이 발생한다.
➔가우시안 분포, 혹은 균일 분포에서 난수를 추출한다. 보통 식은 r = 형태로 결정한 뒤, [-r, r] 사이에서 난수를 결정한다.
● 바이어스는 보통 0으로 초기화 한다.

그 외
● 가중치 벡터가 수직이 되도록 설정(Saxe2014)
● 임의 행로를 활용(Sussillo2014)
● 가중치 초기화와 모멘텀을 동시에 최적화(Sutskever2013)
● 노드의 출력값 분포가 일정하도록 강제화(Mishkin2016)

모멘텀
Gradient의 잡음 현상을 의미하며, 기계 학습은 훈련 집합을 이용하여 Gradient를 추정하므로 잡음이 생길 가능성이 높다.
➔Smoothing을 가하여 잡음 효과를 줄인다(Overshooting 방지 -> 수렴 속도 상향)

가중치 갱신 수식

● v는 속도벡터로, 이전 Gradient를 누적한 것(초기값 : 0)
● 가 0일 경우 모멘텀이 적용되지 않음.
● 가1에 가까울수록 이전 Gradient 정보에 큰 가중치를 준다.
● 보통 0.5, 0.9. 0.99 순으로 진행한다.

네스테로프 모멘텀

현재 v 값으로 다음 이동할 곳인 를예견한후그곳의Gradient를 사용한다. 즉 처음의 값이 아닌 진행 지점을 예측한 다음 그 장소에서 Gradient를 사용하여 모멘텀을 더욱 최소화 하는 효과를 가진다.

적응적 학습률

학습률 p는 너무 크게 주어질 경우 Overshooting 현상으로 인한 진자 현상이 일어날 가능성이 높고 너무 작으면 수렴하는 속도가 매우 느려진다. 이럴 때 Gradient에 학습률 p를 곱하여 모든 매개변수가 같은 크기의 학습률을 사용하게 유도하여 step이 진행됨에 따라 값의 진행을 조절한다.

AdaGrad – Parameter 중 Element의 움직임이 많아질수록 학습률이 낮아지게 하여 element마다 학습률을 다르게 감소시키는 방법.
➔오래된 Gradient와 최근 Gradient가 같은 비중의 역할을 하게 되어 r의 증가로 수렴을 방해할 가능성이 있다.

RMSProp – 기울기를 단순 누적하는 것이 아니라 지수 가중 이동 평균을 사용하여 최신 기울기들이 AdaGrad에 비해 더 크게 반영하도록 하는 것.

Adam – AdaGrad와 RMSProp을 융합한 형태로, 두 기법에서 AdaGrad와 RMSProp의 계수가 초기에 0으로 biased 되는 문제를 해결하기 위하여 고안된 방법

활성함수
활성값 z를 계산하여 나온 결과 r을 적용하는 과정.

초기에는 선형의 활성함수를 사용하였으나 비선형 문제를 해결하기 위해 퍼셉트론에서 계단식 함수를 사용하였다.
그 이후 tanh 형태에서 현재 쓰이고 있는 ReLU 형태로 진화하였다.
- Tanh 함수는 활성값이 커지면 포화상태가 되고 Gradietn가 0에 가까워졌기 때문에 매개변수 갱신이 매우 느렸음.

ReLU(Rectified Linear Unit)

y = ReLU

ReLU의 변형

SoftPlus

- 는보통0.01을 사용함.

LeakyReLU & PReLU

- 를학습으로알아냄

배치 정규화
공변량 시프트 현상
학습이 진행되면서 층 1의 매개변수가 바뀜에 따라 결과값이 따라 바뀜 -> 층 2에 입력되는 데이터의 분포가 변경됨 -> 층 3....으로 갈수록 더 심각해진다
➔학습을 방해하는 요인으로 작용

Gradient Vanishing / Exploding Problem이 발생하지 않도록 하면서 Learning Rate 값을 크게 설정해야 함.
이를 누그러뜨리기 위해 이전의 데이터 정규화를 모든 층에 적용하는 기법.
- 입력 데이터가 아닌 중간 결과 데이터에 적용하는 것이 더 유리하다

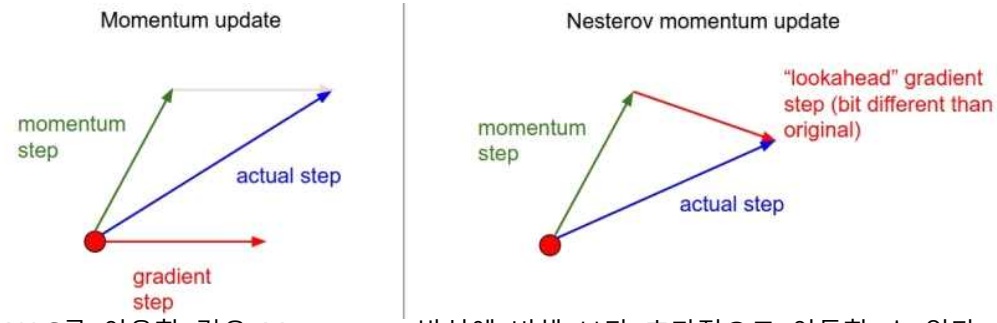
	➔다른 곳에 해도 성능 향상에는 도움이 됨. ● 훈련 집합보다는 미니 배치에 적용하는 것이 훨씬 유리하다. ➔어느 방향으로 업데이트 할 것인지 계산하는 쪽에 정규화를 하는 것이 더 성능에 도움이 된다. 효과 : Traning 결과와 Test 결과 분포가 다를 경우 Traning Data Set에 대한 결과가 Test Data Set의 결과에 맞춰지게 됨. ● CNN에서는 노드 단위가 아니라 특징 맵 단위로 학습, 테스트를 적용 ● 가중치 초기화에 덜 민감함 ● 학습률을 크게 하여 수렴 속도가 향상 ● Sigmoid를 활성함수로 사용하는 깊은 신경망도 학습이 이루어짐 ● Drop Out을 따로 적용하지 않아도 높은 성능이 이루어짐.
질문 내용	원 핫 인코딩을 적용할 경우 요소가 늘어나게 된다면 인코딩 되는 라인 또한 끝 없이 늘어나게 될 것인데, 이렇게 되면 원 핫 보다는 일반 숫자로 표기하는 것이 더 효율적이지 않은가 생각됩니다.

이름	이재은	학번	201502469

구분	내용
학습 범위	5.1절: 목적 함수로서 평균제곱 오차의 단점을 지적하고, 딥러닝이 많이 활용하는 교차 엔트로피와 로그우도를 소개한다. 5.2절: 스토캐스틱 경사 하강법의 성능 향상에 효과적인 전처리, 가중치 초기화, 모멘텀, 적응적 학습률, 활성함수, 배치 정규화를 설명한다.
학습 내용	- 5장은 기계학습의 최적화가 복잡하고 어렵다는 점을 극복하는 여러가지 효과적인 방안에 대해 학습한다. - One hot: In digital circuits and machine learning, one-hot is a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0). - 원핫 코드는 값의 개수만큼 비트를 부여 한다. - Momentum 방식: 말 그대로 Gradient Descent를 통해 이동하는 과정에 일종의 '관성'을 주는 것이다. 현재 Gradient를 통해

이동하는 방향과는 별개로, 과거에 이동했던 방식을 기억하면서 그 방향으로 일정 정도를 추가적으로 이동하는 방식이다.

- Momentum 방식에서는 이동 벡터 vtvt 를 계산할 때 현재 위치에서의 gradient와 momentum step을 독립적으로 계산하고 합친다. 반면, Nesterov momentum update(NAG)에서는 momentum step을 먼저 고려하여, momentum step을 먼저 이동했다고 생각한 후 그 자리에서의 gradient를 구해서 gradient step을 이동한다

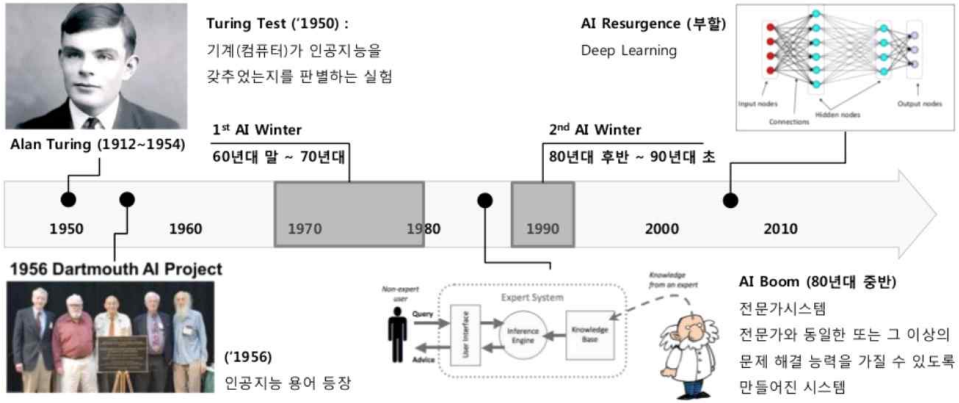


- NAG를 이용할 경우 Momentum 방식에 비해 보다 효과적으로 이동할 수 있다. Momentum 방식의 경우 멈춰야 할 시점에서도 관성에 의해 훨씬 멀리 갈수도 있다는 단점이 존재하는 반면, NAG 방식의 경우 일단 모멘텀으로 이동을 반정도 한 후 어떤 방식으로 이동해야할 지를 결정한다. 따라서 Momentum 방식의 빠른 이동에 대한 이점은 누리면서도, 멈춰야 할 적절한 시점에서 제동을 거는 데에 훨씬 용이하다고 생각할 수 있을 것이다.

- Adagrad(Adaptive Gradient): 변수들을 update할 때 각각의 변수마다 step size를 다르게 설정해서 이동하는 방식이다. 이 알고리즘의 기본적인 아이디어는 '지금까지 많이 변화하지 않은 변수들은 step size를 크게 하고, 지금까지 많이 변화했던 변수들은 step size를 작게 하자' 라는 것이다. 자주 등장하거나 변화를 많이 한 변수들의 경우 optimum에 가까이 있을 확률이 높기 때문에 작은 크기로 이동하면서 세밀한 값을 조정하고, 적게 변화한 변수들은 optimum 값에 도달하기 위해서는 많이 이동해야할 확률이 높기 때문에 먼저 빠르게 loss 값을 줄이는 방향으로 이동한다.

- 인공지능 연대표

- 배치 정규화: 학습 시의 미니배치를 한 단위로 정규화를 하는 것으로 분포의 평균이 0, 분산이 1이 되도록 정규화하는 것이다.

	
질문 내용	

이름	김영연	학번	201500629
구분	내용		
학습 범위	5.1 목적함수: 교차 엔트로피와 로그우도 5.2 성능향상을 위한 요령		
학습 내용	평균제곱 오차(MSE) 목적함수		

$e = \frac{1}{2} \| y - o \|_2^2 e = \frac{1}{2} \| y - o \|_2^2$

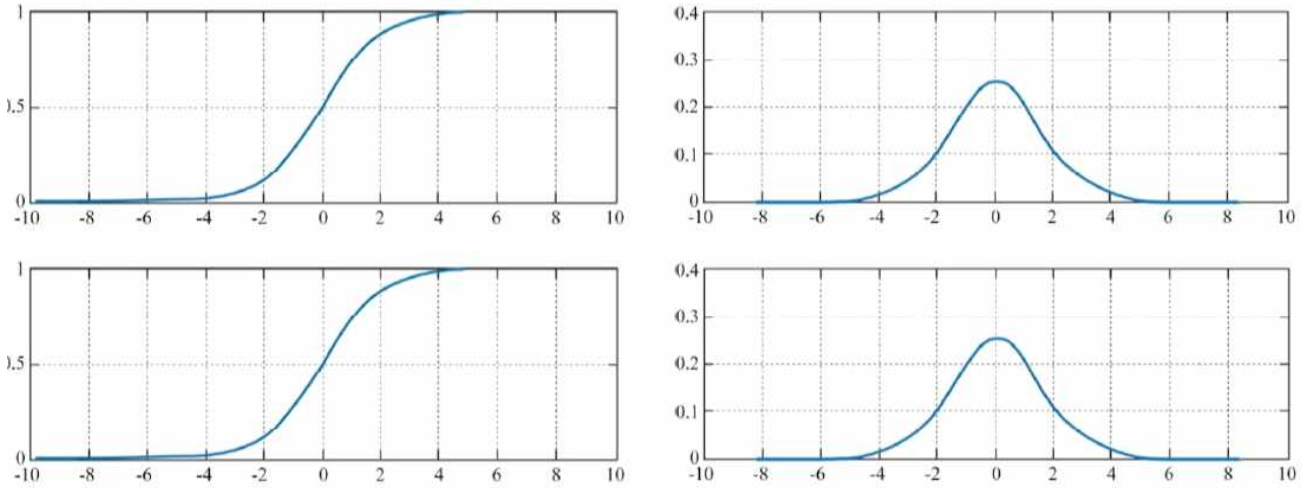
이는 큰 허점이 존재한다.

$e = \frac{1}{2}(y - o)^2 = \frac{1}{2}(y - \sigma(wx + b))^2 e = \frac{1}{2}(y - o)^2 = \frac{1}{2}(y - \sigma(wx + b))^2$

$\frac{\partial e}{\partial w} = -(y - o)x\sigma'(wx + b) \frac{\partial e}{\partial w} = -(y - o)x\sigma'(wx + b)$

$\frac{\partial e}{\partial b} = -(y - o)x\sigma'(wx + b) \frac{\partial e}{\partial b} = -(y - o)x\sigma'(wx + b)$

그레디언트 계산 시 더 많은 오류를 범한 상황이 더 낮은 벌점을 받을 수 있다는 큰 오류를 범할 수 있는데

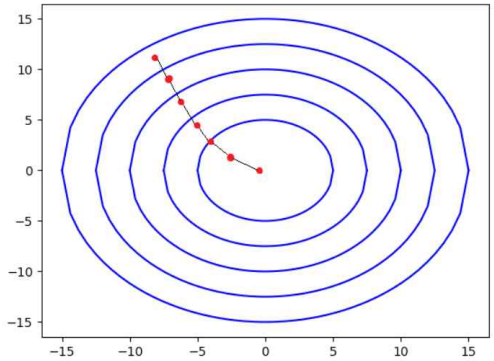
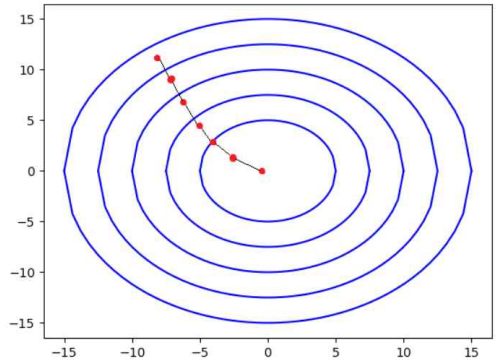


Wx+b가 커지면 그레디언트가 작아지기 때문이다.

6. 교차 엔트로피의 오차

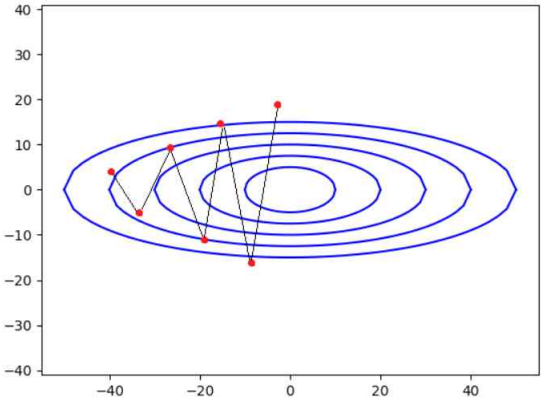
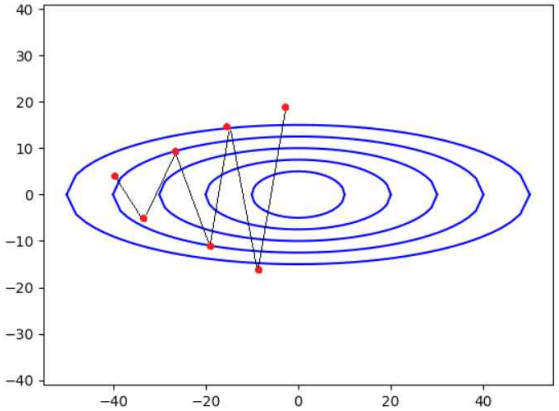
교차 엔트로피의 오차는 정답일 때의 출력이 전체 값을 결정하는 것이다. 모든 값을 평균 내어 제곱한 평균제곱오차와

	달리 정답 레이블의 값과 신경망의 출력만을 고려하기 때문에 전체 값을 고려하게 된다. 신경망의 출력과 정답 레이블의 값이 정해진 경우 발생하는 자연로그 x값에 적용합니다. X 값이 커질수록 오차는 줄어들며, x값이 작아질수록 오차는 커집니다. 레이블에 해단하는 y가 확률변수이고 p는 정답 레이블, Q는 신경망의 출력을 의미한다 교차 엔트로피의 식은 다음과 같다. $H(P,Q) = -\sum_{y\in\{0,1\}} P(y)\log_2 Q(y)$ $H(P,Q) = -\sum_{y\in\{0,1\}} P(y)\log_2 Q(y)$ 교차 엔트로피의 목적함수 $e = -(y\log_2 o + (1-y)\log_2(1-o))$ 이때, $o = \sigma(Z)$ 이고 $z = wx + b$ $e = -(y\log_2 o + (1-y)\log_2(1-o))$ 이때, $o = \sigma(Z)$ 이고 $z = wx + b$ 교차 엔트로피를 이용하면 MSE의 큰 허점을 해결할 수 있다. 7. Softmax 활성함수 3개 이상 분류되어 있는 요소들로부터 하나의 결과 값을 내고자 할 때 사용되는 대표적인 함수이다. Softmax값의 출력으로 0~1 사이의 값을 모두 정규화하여 출력 값들의 총 합은 1이 되는 특징이 있다. $o_j = \frac{e^{sj}}{\sum_{i=1,c} e^{si}}$ $o_j = \frac{e^{sj}}{\sum_{i=1,c} e^{si}}$ 8. 로그우도 $e = -\log_2 O_y e = -\log_2 O_y$ 모든 출력 노드 값을 사용하는 MSE나 교차 엔트로피와 달리 Oy라는 하나의 노드만 사용 9. Softmax와 로그우도 Softmax는 최대 값이 아닌 값을 억제하여 0에 가깝게 만든다는 의도를 내포함 학습 샘플이 알려주는 부류에 해당하는 노드만 보겠다는 로그우도와 잘 어울림 10.데이터 전처리 좀 더 학습이 잘 되도록 입력 데이터를 조정하는 것

다음과 같이 가운데로 갈수록 cost가 작아지는 것이 이상적인 형태이다.

예를 들어 두개의 입력 값의 scale의 크기가 확연히 차이가 나는 경우에 cost 함수를 위와 같이 그리면 찌그러진 모습으로 나타날 것이다.

이러한 현상을 방지하기 위

해 데이터 전처리를 한다.

데이터 전처리는 일반적으로 평균을 0에 맞추는 작업을 우선 거치고, 각 축의 스케일을 맞추는 작업을 한다.

가장 많이 사용하는 방법은 표준화라고 하는 방법이다. $x_i^{new} = \frac{x_i^{old} - \mu_i}{\sigma_i}$ $x_i^{new} = \frac{x_i^{old} - \mu_i}{\sigma_i}$

11.가중치 초기화

12.가중치 초기 값이 0이거나 동일한 경우

가중치의 초기 값을 모두 0으로 초기화하거나 동일한 값으로 초기화 할 경우 모든 뉴런의 동일한 출력 값을 내보내 것이다. 그러면 역전파 단계에서 각 뉴런이 모두 동일한 그래디언트 값을 가지게 된다. 학습이 잘 되려면 각 뉴런이 가중치에 대해 비대칭이어야 하기 때문에 가중치 초기 값을 동일한 값으로 초기화 하면 안된다.

13.작은 난수인 경우

가중치 초기 값은 작은 값으로 초기화 해야 하는데, 활성화 함수가 시그모이드일 경우 만약 가중치 초기 값을 큰 값으로 한다면 0과1로 수렴하기 때문에 그래디언트 소실로 방생하게 된다. 또한 활성화 함수가 Relu일 경우 절대값이 크면 음수일 경우 deadReLU문제가 발생하고 양수일 경우 그래디언트 폭주가 일어난다.

14.모멘텀

기계학습은 훈련 집합을 이용하여 그레디언트를 추정하므로 잡음 가능성이 높다.

모멘텀은 그레디언트 스무딩을 가하여 잡음을 줄이고 이는 수렴 속도를 향상시킨다.

15.적응적 학습률

16.그레이던트에 학습률 ρρ을 곱하면 모든 매개변수가 같은 크기의 학습률을 사용하는 셈이다. 적응적 학습률은 매개변수마다 자신의 상황에 따라 학습률을 조절해 사용한다.

AdaGrad:각 매개변수에 서로 다른 학습률을 적용시킨다. 처음에는 크게 학습하다가 조금씩 작게 시킨다.

Adam: Adam은 모멘텀과 AdaGrad를 합친 것이다.

RMSprop: 기울기를 단순 누적하지 않고 지수 가중 이동 평균을 사용하여 최신 기울기들이 더 크게 반영되도록 한 것

	17.배치 정규화 공변량 시프트 현상: 학습이 진행되면서 층1의 매개 변수가 바뀜에 따라 x1이 바뀜 층2 입장에서 보면 자신에게 입력되는 데이터의 분포가 수시로 바뀌는 셈이다. 이는 층이 깊어질수록 더 심각해지고 학습을 방해하는 요소가 된다. 배치 정규화는 각 층의 출력 값을 정규화 하는 과정을 뜻한다,
질 문 내 용	가중치 초기화를 공부하면서 찾아보니 초기 값을 작은 값으로 하지 않는 경우 절대 값이 크고 음수일 경우 dead ReLU가 발생한다고 한다. Dead ReLU에 대해 알고 싶다.