워크북

교과목명 : 머신 러닝

차시명: 4차시

◈ 담당교수: **장 필 훈**

⊙ 세부목차

- 퍼셉트론 알고리즘
- 확률적생성모델/판별모델
- 로지스틱회귀
- 신경망(다층 퍼셉트론)

	학습에 앞서

■ 학습개요

퍼셉트론알고리즘과 함께 선형분류를 마무리한다. 퍼셉트론 알고리즘의 기본 성질, 수식을 이해하고 경사하강법에 관해 이해한다. 활성함수인 시그모이드 함수를 함께 이해한다.

뉴럴넷의 기초사항을 익히고, 뉴럴넷이 결국 임의의 함수를 근사할 수 있음을 이해한다. 다층퍼셉트론은 이름에서 오해의 여지가 있으나, 다층으로 이루어진 퍼셉트론이 아니고 로지스틱 회귀모델 여러층이다. 이를 이해한다.

■ 학습목표

1	퍼셉트론에 관해 이해하고 특성을 파악한다.
2	시그모이드 함수, 소프트맥스에 대해 이해한다.
3	피드포워드 네트워크 함수의 성질을 이해한다.

4	뉴럴넷이 임의의 함수를 근사할 수 있음을 개념적 증명을 통해 안다.

■ 주요용어

용어	해설
퍼셉트론	$f(w^T\phi(x))$가 1 또는 -1 의 계단함수로 정의되는 유닛. ϕ는 비선형 변환으로 activation function 이라고도 불린다.
경사하강법	함수의 계산이 해석적으로 매우 어렵거나 불가능할 때, 수치적인 방법으로 근사하는 최적화방법이 쓰이는데 그 중 하나가 경사하강법. 가장 흔하게 쓰이고, 퍼셉트론이나 피드포워드네트워크 모두 이 방법을 사용한다. 식으로 나타내면, $w^{\tau+1} = w^{(\tau)} - \eta\nabla E_p(w)$
로지스틱 시그모이드	시그모이드 함수와 같은 말. $\frac{1}{1+e^{-x}}$
피드포워드 네트워크 함수	뉴럴넷의 기본 유닛. 층 하나는 (모든 element 들의 activation 을 제외하면) 행렬의 곱 하나로 단순하게 표현될 수 있다. 자세한 것은 ppt 참고. 이 함수를 다층으로 쌓으면 모든 공간에서 임의의 함수를 근사할 수 있다.

	학습하기

<퍼셉트론>

퍼셉트론은 다음과 같은 간단한 함수식입니다.

$$y(\vec{x}) = f(\vec{w}^T\phi(\vec{x}))$$

$$\text{where } f(a) = \begin{cases} +1, a \geq 0 \\ -1, a < 0 \end{cases}$$

여기서 ϕ는 비선형변환을 나타냅니다. 위 식은, 특정 데이터포인트가 어떤 클래스에 속하면 1을 출력값으로 주고, 그렇지 않으면 -1을 출력으로 준다는 뜻입니다. 따라서, 모든 패턴에 대해 다음 식을 만족하는 w를 찾는 것이 목표가 됩니다.

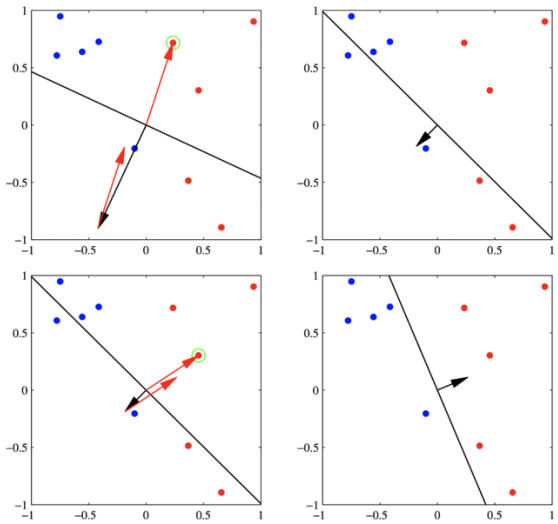
$$\vec{w}^T\phi(\vec{x}_n)t_n > 0$$

w는 경사하강법으로 찾습니다. 2강에 잠깐 등장했었습니다.

$$\vec{w}^{(\tau+1)} = \vec{w}^{(\tau)} - \eta\nabla\mathbb{E}_p(\vec{w}) = \vec{w}^{(\tau)} + \eta\phi_n t_n$$

앞의 식에서 η는 learning rate으로 보통 작은 값을 줍니다. 한 단계(step)에 얼만큼이나 오차를 반영할지 결정하는 하이퍼파라미터입니다. E_p는 패널티의 기댓값을 나타냅니다. 퍼셉트론기준의 경우 올바르게 분류되면 0, 오분류되면 $-w^T\phi(x_n)t_n$을 패널티로 주게 되어있고, 따라서 기댓값은 오분류된 패턴 전체에 대해, $\mathbb{E}_p(\vec{w}) = -\sum_{n\in M}\vec{w}^T\phi_n t_n$ 로 주어집니다. (M은 오분류된 패턴의 집합을 뜻합니다)

아래 그림에 퍼셉트론의 학습과정이 직관적으로 표현되어 있습니다. 비숍책 195쪽입니다.



좌상-우상-좌하-우하 순서입니다. 좌상에서 녹색원으로 된 붉은점이 오분류되었습니다.(검은화살표가 가리키는 방향이 붉은점입니다. 초기설정에 따릅니다. 초기설정은 보통 랜덤하게 합니다) 그래서 오류를 수정하는 벡터를 더하게 됩니다.(위의 경사하강법의 수식참고) 그러면 우상의 그림이 나옵니다. 다시 세번째(좌하)그림에서 녹색동그라미친 붉은점이 오분류이고 같은 방법으로 수정하게 됩니다. 그러면 결국 우하에서 원하는 분류기를 얻게 됩니다. 검은색화살표가 가리키는 방향이 바뀌었음을 보세요. 오분류 데이터포인트가 없기 때문에 추가적인 학습이 진행되지 않습니다. 오류함숫값을 이용했다면 아마 구분선이 조금 더 가운데쯤으로 이동했을 것입니다.

학습과정에서 오분류패턴의 오류함수에 대한 기여도는 점점 감소하게 되어있는데 아래 식에서 확인할 수 있습니다.

$$-\vec{w}^{(\tau+1)}\phi_n t_n = \left(\vec{w}^{(\tau)} + \eta\phi_n t_n\right)^T\phi_n t_n$$

$$= -\vec{w}^{(\tau)^T}\phi_n t_n - (\phi_n t_n)^T\phi_n t_n < -\vec{w}^{(\tau)^T}\phi_n t_n$$

만일 훈련집합이 선형분리 가능하면, 퍼셉트론 알고리즘은 정확한 해를 유한한 단계 안에 구할 수 있음이 알려져 있습니다. 다만 유한하다 해도 너무나 큰 수의 단계가 필요하면 실제적으로 무한이나 다름이 없고, 문제 자체가 선형분리 가능한 것인지 미리 알수 없는 경우가 대다수라, 실질적으로는 일단 알고리즘을 적용해보고 그 결과로 적용가능성을 가늠해봅니다. 어떻게보면 일의 선후가 바뀐것 같지만, 닭이냐 달걀이냐 문제이기 때문에 꼭 그렇다고 볼수도 없지요. 만일 선형분리 불가능한 집합에 퍼셉트론 알고리즘을 적용하면 알고리즘이 수렴하지 않습니다. 그리고 앞서 그림에서 본 것처럼 최적점을 찾아가는 것이 아니라, 오분류된 포인트를 바로잡다가 오분류를 모두 처리하면 멈추기 때문에 입력되는 데이터 순서에 따라 분류함수는 다를 수 있습니다. 다시말해, 답이 하나가 아닐 수 있습니다.

이런 특성들 외에 퍼셉트론은 몇가지 단점을 가지고 있습니다. 첫번째로는 확률적인 출력 값을 내지 않는 다는 것, 두번째로는 다중 class 문제에 대해 일반화 되지 않는다는 것, 세번째로는 고정된 기저함수의 선형결합으로만 이루어져 있다는 것입니다. 여기서 새번째 단점은 지금까지 우리가 살펴본 모든 모델이 공통적으로 지니고 있는 단점입니다. ('고정'이 문제가 아니고 '선형'이 문제입니다.) 다만 우리가 앞에서도 잠시 살펴본바와 같이, 기저함수 자체를 비선형함수를 이용함으로써 기저함수의 선형결합으로도 비선형분리가 가능하게 할 수 있습니다. 다시말해 x공간에서 선형분리 불가능이라도 ϕ의 공간에서는 가능할 수 있다는 이야기입니다.

<확률적 생성모델>

확률적 모델은 앞서 설명한 것처럼 판별 함수가 class를 바로 결정하는 것이 아니라, 해당 클래스에 속할 확률을 결과값으로 주는 모델을 말합니다. 식으로 나타내면 (이분문제일때) 다음과 같습니다.

$$p(C_1|\vec{x}) = \frac{p(\vec{x}|C_1)p(C_1)}{p(\vec{x}|C_1)p(C_1) + p(\vec{x}|C_2)p(C_2)} = \frac{1}{1+\exp(-a)} = \sigma(a)$$

식 중간에 $a = \ln\frac{p(\vec{x}|C_1)p(C_1)}{p(\vec{x}|C_2)p(C_2)}$ 로 놓고 간단히 정리한 것입니다. 앞서 배운 시그모이드 함수를 얻을 수 있습니다. 시그모이드 함수는 로지스틱 시그모이드라고 불리기도 하며, squashing funciton이라고 불리기도 합니다. 입력으로 실수 전체를 받으면서도 출력을 0~1사이로 제한해주기 때문에 그런 이름이 붙었습니다.

시그모이드 함수는 아래와 같은 성질을 가집니다. (원래 식에 대입해서 계산해보면 쉽게 확인 가능합니다)

$$\sigma(-a) = 1 - \sigma(a)$$

시그모이드 함수를 a에 대해 다시 정리하면 다음을 얻는데,

$$a = \ln\frac{\sigma}{1-\sigma}$$

두 클래스에 대한 확률비의 로그값, 즉 log odds(=logit)을 뜻합니다. 2분문제이므로 두 클래스에 대한 확률값을 합하면 1이 나와야 하고, 따라서 분모에 보이는 1-σ 는 다른 클래스의 확률입니다.

클래스가 다수일 때는 어떨까요, 정리하면 다음과 같습니다.

$$p(C_k|\vec{x}) = \frac{p(\vec{x}|C_k)p(C_k)}{\sum_j p(\vec{x}|C_j)p(C_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

여기서 $a_k = \ln\left(p(\mathbf{x}|C_k)p(C_k)\right)$ 입니다. 위 식에서 오른쪽 항을 보면 소프트맥스 함수라는 것을 쉽게 알아볼 수 있습니다.

다시 2-class문제로 돌아가서, 사후확률을 로지스틱 시그모이드로 적을 수 있으므로 최대가능도방법을 이용해서 모델의 매개변수 구하기가 가능합니다. 항상 그래왔듯이 미분=0으로 두고 계산합니다.

$$\frac{d\sigma}{da} = \sigma(1-\sigma)$$

시그모이드는 출력 y가 확률 그 자체이고, 2클래스 문제이므로 (1-y)가 해당 클래스가 아닐 확률입니다. 따라서 가능도함수를 다음과 같이 적을 수 있습니다.

$$p(\vec{t}|w) = \prod_{n=1}^{N} y_n^{t_n}\{1-y_n\}^{1-t_n}$$

위에서 t_n은 타겟, y_n은 $\sigma(a_n) = \sigma(\mathbf{w}^T\boldsymbol{\phi}_n)$입니다.

이제 여기에 음의 로그를 취하고 미분하면,

$$\nabla\mathbb{E}(w) = \sum_{n=1}^{N}(y_n - t_n)\phi_n$$

위와같이 간단한 모양을 얻습니다.

우리가 선형회귀를 배울때는 여기서 더 정리해서 w에 대해 해를 바로 구했지만, 지금은 식이 시그모이드함수를 포함하고 있으므로 닫힌형태(closed form. 유한개의 수학적 표현을 사용하여 표현이 가능한 형태)의 해를 구할 수 없습니다. 이럴때는 여러가지 방법을 이용해 근사하게 됩니다. 우리가 나중에 배울 지수족분포에서는 위와 같은 형태가 일반적입니다.

<신경망>

앞서 배운 기저함수들의 선형결합은 계산이 가능하고 이해하기도 쉽다는 장점이 있으나 실제로 사용할 때는 데이터의 양이 적거나, 매우 큰 차원의 데이터를 다루어야 해서 한계가 있을 때가 많습니다. 그렇게 큰 스케일의 문제에 적용하려면 기저함수를 늘리거나, 기저함수 자체를 데이터에 adaptive하게 유연하게 만들어야 하는데, 신경망이 대표적으로 그러한 모델입니다. 즉, 매개변수적인 기저함수를 사용합니다.(여담이지만 다층 퍼셉트론은 퍼셉트론 다층이 아니고 로지스틱 회귀의 다층인데 이름이 그렇게 붙여졌습니다.)

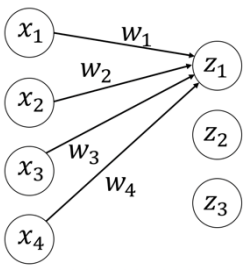
신경망의 경우는 가능도함수가 볼록함수가 아니어서 시작점에 따라 결과가 다르게 수렴할 수 있음에 주의합니다.

피드포워드 네트워크함수는 다음과 같이 나타낼 수 있습니다.

$$y(\vec{x},\vec{w}) = f\left(\sum_{j=1}^{M} w_j\phi_j(x)\right)$$

분류문제라면 f로 비선형 활성화함수를 쓸 것이고, (선형)회귀문제라면 항등함수를 쓸 것입니다. 훈련단계에서 w뿐 아니라 ϕ도 함께 조절되는 것이 피드포워드네트워크의 특징입니다.

다층 퍼셉트론의 가장 단순한 형태인 fcn(fully connected network)의 경우 다음과 같은 모양을 가집니다.



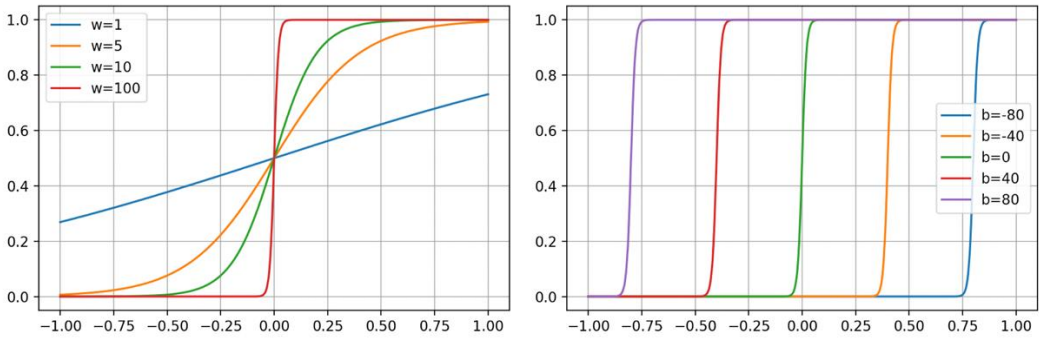
한 층만을 떼어서 본 것이며, z_2, z_3로 가는 모든 선을 생략한 것입니다. 좌측과 같은 한 층은 아래와 같은 행렬곱과 동일한 계산을 하게 됩니다.

$$\begin{pmatrix} w_1 & w_2 & w_3 & w_4 \\ w_5 & w_6 & w_7 & w_8 \\ w_9 & w_{10} & w_{11} & w_{12} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

위와같은 과정을 여러층 반복하게 됩니다. 과정이 반복될수록 차원감소를 위해 점점 유닛수를 줄입니다. 우리가 원하는 단순한 결과(예-분류값)를 얻기 위해 필연적인 과정입니다. PCA등 선형적인 차원감소방법은 나중에 배웁니다.

그러면 퍼셉트론과 다층 퍼셉트론은 어떻게 다를까요. 오히려 둘은 비슷한점을 찾는게 더 이상할정도로 동질적이지 않습니다. 퍼셉트론은 불연속적 비선형 계단함수이고, 다층 퍼셉트론은 연속적 시그모이드 비선형함수이니까요. 시그모이드함수는 비선형함수라도 미분이 가능합니다.

신경망은 universal approximator입니다. 원하는 수준으로 모든 데이터에 적합시킬 수 있다는 말입니다. 이것은 쉽게 보일 수 있는데, 일단 우리가 원하는 모든 함수는 결국 step function들의 수많은 합으로 충분히 근사할 수 있다는 점, 그리고 신경망으로 step function을 근사해낼 수 있다는 점만 알면 됩니다. 아래는 $\sigma(wx+b)$를 다양한 w, b값으로 그려본 것입니다.



w와 b를 적절히 조절하고 이러한 함수를 원하는만큼 여러개 조합하면 임의의 위치에서 원하는만큼의 정확도로 step function을 근사해낼 수 있음을 알 수 있습니다. 이것은 2차원에서만 보인 것이지만, 똑같은 방식으로 다차원에서도 가능하고, 따라서 우리가 원하는 임의의 모든 분포를 다층퍼셉트론이 근사해낼 수 있음을 보였습니다. 이론적으로도 이렇지만 실제로도 훌륭하게 동작해서, 신경망은 최근 비전분야와 음성인식 분야에서 압도적인 성능을 보여주면서 순식간에 주류로 자리잡았습니다.

연습문제

1. (OX문제) 퍼셉트론 알고리즘은 오류함수가 조각별 선형이다

   O 맞는 설명이다(오분류 영역에서만 오류가 정의된다)

2. (OX문제) 훈련집합이 선형분리 가능하면, 퍼셉트론 알고리즘은 원하는 오차 내에서 정확한 해를 유한한 단계로 구해낼 수 있다.

   O 그렇다. (수업에 직접 증명하지는 않았고, 옳다고 알려져 있다)

3. (OX문제) 선형분리가 불가능할 경우에도 일반적으로 퍼셉트론 알고리즘을 수렴시킬 수 있는 방법이 있다.

   X 그런 방법은 존재하지 않는다.

4. (OX문제) 다중클래스문제에도 일반적으로 퍼셉트론 알고리즘을 적용할 수 있다.

X 아니다.

5. (OX문제) 클래스별 조건부밀도가 가우시안이라고 가정하고, 모든 클래스가 같은 공분산행렬을 공유한다면 결정경계는 무조건 선형이다.

O 그렇다.

6. (OX문제) 사후확률을 로지스틱 시그모이드형태로 적을 수 있다면, 최대가능도방법을 사용해서 모델의 매개변수를 구할 수 있다.

O 그렇다. 어떤 식이든 사후확률이 해석적이면 최대가능도 방법을 사용해볼 수 있다.

7. (OX문제) 피드포워드 네트워크 함수의 경우 기저함수가 매개변수adaptive한 형태이다.

O 그렇다. 퍼셉트론의 경우 고정된 기저함수를 사용하기 때문에 큰 데이터에서 사용하기 적당하지 않다.

8. (OX문제) 뉴럴넷으로 일반적인 차원에서 모든 형태의 함수를 원하는 만큼 정확하게 근사할 수 있다.

O 그렇다. 충분한 hidden unit들이 주어지면 가능함을 보일 수 있다.

정리하기

1. 퍼셉트론에서 하나의 unit은 비선형변환함수를 통과한 입력벡터의 출력이 특정값에서 step function을 이룬다. $y(x) = f(w^T \phi(x))$
2. 퍼셉트론의 기댓값은 오분류된 패턴 전체에 대해서만 정의되므로 모든 공간에서 미분은 불가능하다.
3. 퍼셉트론기준의 계산은 경사하강법을 이용한다.
4. 오분류된 패턴의 오류함수에 대한 기여도는 점점 감소한다.
5. 훈련집합이 선형분리 가능하면 퍼셉트론 학습 알고리즘은 정확한 해를 유한한 단계로 구할 수 있다.
6. 선형분리 불가능하면 수렴하지 않는다.
7. 퍼셉트론알고리즘은 확률적 출력값을 내지 않고 다중클래스에 대해 일반화되지 않는다.
8. 확률적 생성모델의 경우 logistic sigmoid를 출력함수로 이용한다.
9. 로지스틱 회귀라고 불리지만 분류모델에 속한다.
10. 로지스틱 회귀는 수치적 방법으로 근사하여 계산한다.
11. 데이터의 양이 너무 크면 기저함수 자체를 매개변수 적응가능하게 만든다. 신경망이 그 대표적인 예.
12. 피드포워드 네트워크함수는 모든 공간에 대해 정의되므로 퍼셉트론과 다르다.
13. 피드포워드 네트워크함수는 차원감소역할을 하지만, 선형은 아니다.
14. 뉴럴넷은 universal approximator다.

참고하기

Bishop, C. M. "Bishop-Pattern Recognition and Machine Learning-Springer 2006." Antimicrob. Agents Chemother (2014): 03728-14.

다음 차시 예고

- 신경망 네트워크의 훈련(오차역전파)
- 신경망의 정규화