데이터마이닝(DataMining)

Chapter 6.1. 신경망

- 신경망(neural networks): 인간의 두뇌구조를 모방한 지도학습법으로서 여러개의 뉴런들을 상호 연결하여 입력값에 대한 최적의 출력값을 예측
- 통계적인 관점에서 입력변수의 선형결합에 비선형 함수를 취하는 사영추적회귀의 일종임
- 예측력이 좋지만 해석이 어려움
- McCulloch과 Pitts (1943): 인간의 뇌 신경노드의 작동 모형을 구축
- Rosenblatt (1958): 단층신경망(single layer perceptron) 알고리즘 개발 1980년대 이전에는 컴퓨터 성능이 낮아서 그리 널리 사용되지 않다가 1980년대에 이르러 다시 각광을 받기 시작
- 다층신경망(multi layer perceptron)과 역전파(back propagation) 알고리즘의 결합으로 신경망 모형의 응용분야가 크게 확장
- 2010년쯤부터 컴퓨터 성능의 향상과 몇 가지 새로운 아이디어 (autoencoder, dropout,...)와 함께 딥러닝이라는 새로운 이름으로 등장. 특히 이미지 분류에서 성공적

신경망

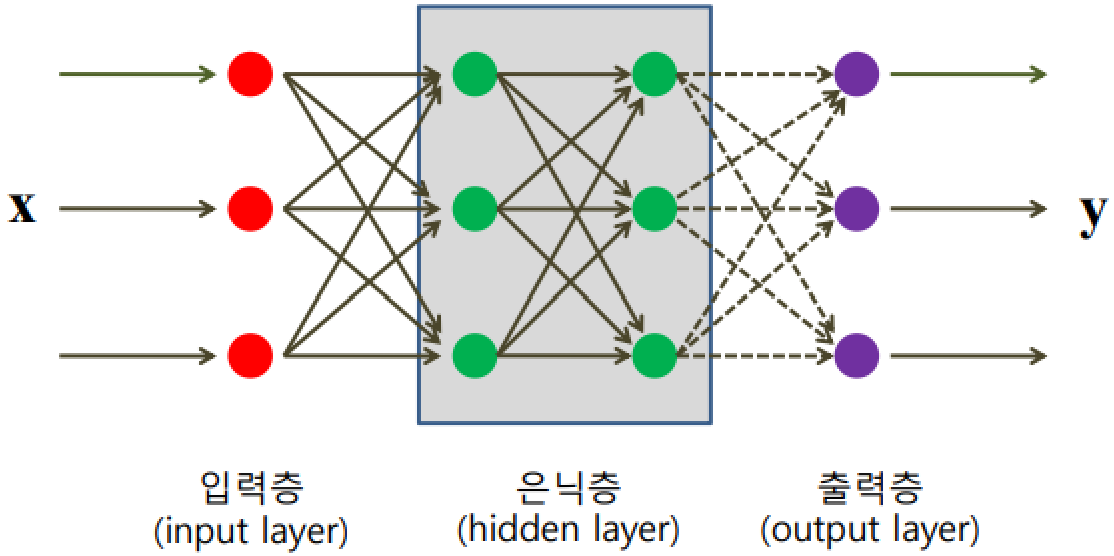
- 입력층(Input Layer) : 각 입력변수에 대응되는 마디들로 구성되어 있으며, 명목형(nominal) 변수에 대해서는 각 수준에 대응하는 입력마디를 가지게 되는데, 이는 통계적 선형모형에서 가변수(dummy variable)를 사용하는 것과 같음

- 은닉층(Hidden Layer) : 여러 개의 은닉마디로 구성되어 있으며 입력층으로부터 전달되는 변수값들의 선형결합(linear combination)을 비선형함수(nonlinear function)로 처리하여 출력층 또는 다른 은닉층에 전달함

- 출력층(Output Layer) 목표변수(target)에 대응하는 마디들을 갖고, 여러 개의 목표변수 또는 세 개 이상의 수준을 가지는 명목형 목표변수가 있을 경우에는 여러 개의 출력마디들이 존재

신경망

- 다층신경망 모형의 구조



신경망

- 클래스의 수가 K인 분류 문제
- 출력노드 $k(=1,2,\cdots,K)$: 클래스 k에 속할 확률을 모형화
- 출력변수: 자료가 k 번째 클래스에 속하는 경우 k 번째 좌표는 1이고 나머지 좌표는 0으로 코딩
- 회귀문제는 $K=1$인 경우에 해당 모형
- 모형

$$z_m = \sigma(\alpha_{0m} + \alpha_m^T x), \qquad m = 1, 2, \cdots, M,$$

$$t_k = \beta_{0k} + \beta_k^T z, \qquad k = 1, 2, \cdots, K,$$

$$f_k(x) = g_k(t), \qquad k = 1, 2, \cdots, K.$$

신경망

- $\sigma(\cdot)$: 활성함수(activation function)라 부르며 흔히 시그모이드 (sigmoid) 함수를 사용
  - 단극성 : $\sigma(v) = \frac{1}{1+e^{-v}}$
  - 양극성 : $\sigma(v) = \frac{1-e^{-v}}{1+e^{-v}}$
- 활성함수로 RBF(radial basis function) $\sigma(v) = \exp(-v^2/2)$를 사용하는 경우 RBF 신경망이라 부름
- $g_k(t)$: 출력함수(output function), 출력값 t에 대하여 최종적인 비선형 변환
  - 회귀 : 항등함수(identity function) $g_k(t) = t_k$ 사용
  - 분류 : softmax 함수 $g_k(t) = \frac{e^{t_k}}{\sum_{i=1}^{K} e^{t_i}}$ 사용

신경망

- θ : 모수 $\alpha_{0m}, \alpha_m (m = 1, 2, \cdots, M)$과 β_{0k}, $\beta_k (k = 1, 2, \cdots, K)$ 의 벡터
- 비용함수
  - 회귀 : 오차제곱합 $R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{n} \left(y_{ik} - f_k(x_i)\right)^2$
  - 분류 : 오차제곱합 또는 deviance $R(\theta) = -\sum_{k=1}^{K} \sum_{i=1}^{n} y_{ik} \log f_k(x_i)$
- 예측
  - $G(x) = \arg\max_k f_k(x)$

신경망

- 활성함수: softmax, 비용함수: deviance ⇒ 은닉노드에 대한 선형 로지스틱회귀이며, 모수들은 최대우도법으로 추정
- 일반적으로 $R(\theta)$은 비선형함수이므로 전역 최소값(global minimizer) 을 찾는 것은 거의 불가능
- 대신 벌점항을 이용한 기울기 강하 알고리즘, 알고리즘의 조기 종료(early stopping)등의 간접 벌점화를 결합하여 국소 최소값를 구함

역전파 알고리즘

- 기울기 강하(gradient descent) 알고리즘의 일종
- 오차제곱합을 비용함수로 사용하는 경우 고려

$$R(\theta) = \sum_{i=1}^{n} R_i = \sum_{k=1}^{K} \sum_{i=1}^{n} \left(y_{ik} - f_k(x_i)\right)^2 ,$$

$$z_{mi} = \sigma(\alpha_{0m} + \alpha_m^T x_i), \qquad z_i = (z_{1i}, \cdots , z_{Mi})^T$$

역전파 알고리즘

- 비용함수의 편도함수:

$$\frac{\partial R_i}{\partial \beta_{km}} = -2\big(y_{ik} - f_k(x_i)\big)g_k'\big(\beta_k^T z_i\big)z_m, \quad (1)$$

$$\frac{\partial R_i}{\partial \alpha_{MI}} = -2\sum_{k=1}^{K}\big(y_{ik} - f_k(x_i)\big)g_k'\big(\beta_k^T z_i\big)\beta_{km}\sigma'(\alpha_m^T x_i)x_{iI}\,.$$

역전파 알고리즘

- 업데이트

$$\beta_{km}^{(r+1)} = \beta_{km}^{(r)} - \gamma_r \sum_{i=1}^{n} \frac{\partial R_i}{\partial \beta_{km}^{(r)}}, \quad (2)$$

$$\alpha_{km}^{(r+1)} = \alpha_{km}^{(r)} - \gamma_r \sum_{i=1}^{n} \frac{\partial R_i}{\partial \alpha_{mI}^{(r)}},$$

여기서 γ_r학습률

역전파 알고리즘

- (1)에서 $\frac{\partial R_i}{\partial \beta_{km}} = \delta_{ki} z_{mi}$와 $\frac{\partial R_i}{\partial \alpha_{mI}} = s_{mi} x_{il}$ 로 놓으면 δ_{ki}와 s_{mi}는 각각 출력층과 은닉층에서의 현재 모형의 오차로서 역전파 등식 만족

$$s_{mi} = \sigma'(\alpha_m^T x_i) \sum_{k=1}^{K} \beta_{km} \delta_{ki} \quad (3)$$

역전파 알고리즘

- (3)을 이용한 (2)의 업데이트
  - 전방 패스(forward pass): 주어진 가중값에 대하여 모형으로부터 예측값 $\hat{f}_k(x_i)$를 계산
  - 후방 패스(backward pass): 오차 δ_{ki}를 계산하고 식 (3)을 이용하여 역전파시켜서 오차 s_{mi}를 계산

모형 구축시 고려사항

- 입력자료의 선택에 매우 민감
    - 범주형: 입력변수의 경우 모든 범주에서 일정 빈도 이상, 출력값의 범주들의 빈도가 차이가 크지 않음
    - 연속형: 변수값들의 범위가 비슷
    - 입력변수의 수가 너무 적거나 많지 않음
- 연속형 입력변수의 변환 또는 범주화
    - 분포가 대략 대칭이 되도록 로그 변환 등을 고려
    - 혹은 범주화
    - (예) 소득: 매우 낮음, 낮음, 중간, 높음, 대단히 높음 등으로 범주화

모형 구축시 고려사항

- 새로운 변수의 생성
  - (예) 고객의 수입, 학력 등 여러 가지 사항을 고려하여 구매지수를 만든 후에 이 지수를 입력변수로 사용하여 특정한 상품의 구매여부를 예측

- 모든 범주형 변수는 같은 범위를 갖도록 가변수화 하는 것이 바람직

모형 구축시 고려사항

- 역전파 알고리즘은 초기값에 따라 그 결과가 많이 달라짐
- 가중치가 0이면 시그모이드 함수는 대략 선형이 되고 따라서 신경망 모형은 근사적으로 선형모형
- 보통 초기치는 0근처에서 랜덤하게 선택되므로 초기의 모형은 선형모형에 가깝고 가중치 값이 증가할수록 비선형모형
- 초기치가 정확히 0이면 반복에 따라 값이 전혀 변하지 않고 너무 큰 값에서 출발하면 좋지 않은 해를 주는 문제점이 있으므로 주의

모형 구축시 고려사항

- 일반적으로 비용함수 $R(\theta)$는 비볼록함수이고 여러개의 국소 최소값들(local minima)을 가짐
- 랜덤하게 선택된 여러개의 초기치에 대하여 신경망을 적합한 후 얻은 해들을 비교
  - 가장 오차가 작은 것을 선택하여 최종예측
  - 예측값의 평균(또는 최빈값)을 구하여 최종예측
- 또다른 방법으로 훈련자료에 대하여 신경망을 기저 학습법으로 사용하는 배깅(bagging)을 적용

모형 구축시 고려사항

- 온라인 학습모드(online learning mode) : 각 관측값을 순차적으로 하나씩 신경망에 투입하여 가중치 추정값을 매번 조정

- 확률적 학습모드(probabilistic learning mode) : 신경망에 투입되는 관측값의 순서가 랜덤

- 배치 학습모드(batch learning mode) : 전체 훈련자료 전체를 동시에 신경망에 투입

모형 구축시 고려사항

- 배치 모드에 대한 온라인 모드의 장점
  - 일반적으로 속도가 더 빠르며 특히 훈련자료에 비슷한 값이 많은 경우에는 그 차이가 더 두드러짐
  - 훈련자료가 비정상성(nonstationarity)과 같은 특이한 성질을 가진 경우에 더 좋음
  - 국소 최소값에서 벗어나기가 더 쉬움
  - 고차원 자료에 대하여 배치 학습모드로 학습하려면 큰 행렬에 대한 연산이 필요

모형 구축시 고려사항

- 학습률은 보통 상수값을 사용

- 온라인 학습모드에서는 처음에는 큰 값으로 정하고 반복이 진행되어 해에 가까울수록 학습률이 0으로 수렴하도록 줄임

- (예) $\gamma_r = \frac{1}{r}$처럼 $\gamma \rightarrow 0,\ \sum_r \gamma_r = \infty, \sum_r \gamma_r^2 < \infty$를 만족하면 적절한 조건하에서 해로 수렴

은닉층과 은닉노드의 수

- 모형 선택: 은닉층의 수와 은닉노드의 수 결정
- 은닉층의 수 : 은닉층이 하나인 신경망은 범용 근사자(universal approximator) 이므로 많은 경우 은닉층은 하나로 하고 은닉 노드수를 적절히 선택
- 은닉노드의 수 : 교차확인오차를 사용하여 결정하는 것보다는 적절히 큰 값으로 놓고 가중치 감소(weight decay)라는 모수에 대한 벌점화를 적용

과대적합

- 많은 모수를 추정해야 하므로 과대적합 문제가 빈번히 발생
- 과대적합을 피하기 위한 방법
  - 조기종료 : 검증오차가 증가하기 시작하면 반복을 중지하는 방법으로 최종모형을 선형 모형으로 축소시킴
  - 가중치 감소 : 벌점화된 목적함수 $R(\theta) + \lambda J(\theta),\ J(\theta) = \sum_{k,m} \beta_{km}^2 + \sum_{m,l} \alpha_{ml}^2\ ,\ \lambda \geq 0$: 교차검증법으로 추정

과대적합

- 가중치 제거(weight elimination) 벌점항

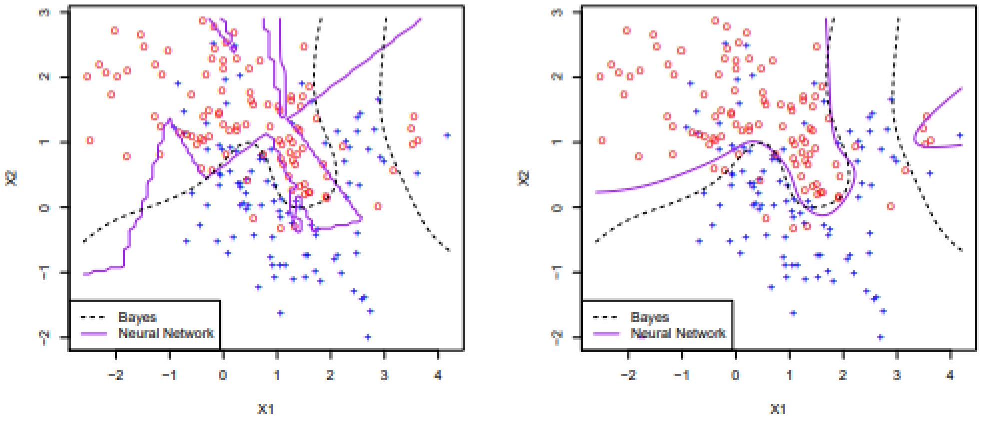
$$J(\theta) = \sum_{k,m} \frac{1 + \beta_{km}^2}{\beta_{km}^2} + \sum_{m,I} \frac{1 + \alpha_{mI}^2}{\alpha_{mI}^2}$$

가중치 감소에 비하여 작은 계수값들을 더욱 줄여줌

- DNN(deep neural network)에서 dropout(랜덤하게 노드들을 제거)은 일종의 벌점화로 볼 수 있음

과대적합

- 혼합 자료에 대한 가중치 감소의 효과



(a) 가중치 감소 없음($\lambda = 0$)　(b) 가중치 감소($\lambda = 0.02$)