능동적인 데이터 선택에 의한 RBF 신경망의 학습

박상욱, 장병탁
서울대학교 컴퓨터공학부
{swpark, btzhang}@scai.snu.ac.kr

Learning RBF Neural Networks by Active Data Selection

Sang-Wook Park and Byoung-Tak Zhang
School of Computer Science and Engeering
Seoul National University

요 약

본 논문에서는 데이터를 능동적으로 선택하고, 그 데이터에 맞추어 RBF 은닉 뉴런을 증가시키는 신경망을 제안한다. 현재의 신경망에 대해서 가장 학습이 어려운 데이터를 선택해서 신경망을 학습하고, 학습한 신경망에 대해서 다시 에러가 가장 큰 데이터를 뽑아서 학습시키는 과정을 반복한다. 5개의 실세계 데이터에 대해 실험을 해보고, Platt이 제안한 RAN과 성능을 비교한다. 점진적으로 임계 데이터를 선택해서 학습을 함으로써, 전체 데이터를 다 사용하지 않고도, 전체 데이터를 다 사용한 경우와 비슷한 성능을 보임을 실험을 통해서 알 수 있다.

1. 서 론

전통적인 RBF 신경망 학습 방식은 입력 데이터의 특징을 이용해서 미리 은닉뉴런의 개수와 그것의 파라미터를 고정시킨후, 은닉뉴런과 출력뉴런 사이의 연결 가중치는 선형 최소제곱 방법을 이용해서 추정한다. 이러한 방법을 이용하면, 처음에 은닉뉴런의 개수를 결정하는 일이 쉽지 않을 뿐만 아니라, 새로운 데이터가 계속해서 들어오는 경우에 새로운 신경망의 크기와 파라미터들을 새로 정해야하기 때문에 순차적 학습에 사용할 수 없다는 단점이 있다.

이러한 약점을 극복하기 위해, RAN(Resource-Allocating Network)[1]에서는 데이타를 순차적으로 입력받고 그것에 기반해서 자동으로 은닉뉴런의 개수와 파라미터를 결정하는 알고리즘을 개발하였다. 이 알고리즘은 은닉뉴런이 없는 상태로 네트워크를 시작하여 순차적으로 들어오는 데이터를 보고 적당한 기준에 의해서 새로운 은닉뉴런을 만들어낸다.

RAN은 가지고 있는 모든 데이타를 순차적으로 한번씩 사용한다. 그러나 데이터는 어느 정도 중복된 정보를 가지게 되므로 모든 데이터를 학습에 사용할 필요는 없다[4, 5]. 본 논문에서는 중요한 데이터만을 능동적으로 선택함으로써 모든 데이터를 학습하는데 다 사용하지 않고도 충분한 성능을 나타내는 ARAN(Active RAN)을 소개한다.

능동적으로 데이터를 선택하는 방법으로는 현재까지 학습을 하지 않았던 데이터를 선택하거나, 에러가 큰 데이터를 선택하거나, 이전 학습시에 이미 나타난 데이터를 선택하는 등의 여러 가지 방법이 있다[6, 7]. ARAN은 현재까지 학습된 신경망에서 학습이 가장 어려운 데이터를 학습 데이터로 선택함으로써 작은 학습 데이터 집합에서도 효율적으로 학습을 할 수 있도록 하고자 한다.

2장에서는 우리가 사용할 알고리즘을 기술하고, 3장에서는 그 학습 알고리즘을 이용해서 UCI 실세계 데이터에 대해 실험한 결과를 보이고, 4장에서는 이 논문의 결론 및 앞으로의 과제에 대해 기술한다.

2. 알고리즘

ARAN 알고리즘은 RAN알고리즘을 바탕으로 능동적인 데이터 선택의 기법을 추가한 것이다. 간단히 그 과정을 살펴보면 표 1과 같다. 알고리즘을 좀 더 자세히 기술하기 위해서 몇가지 표기법을 정의하도록 하자.

후보집합 (C)은 전체 학습에 사용가능한 데이터 중에서 아직 학습에 사용되지 않은 데이터 집합이라 정의하고, 훈련집합 (T)은 학습에 한번 이상 사용된 데이터의 집합이라 정의하자. 그러면, C의 데이타의 개수와 T의 데이터의 개수를 더하면 전체 학습 데이터 개수 (N)가 된다. 초기에는 모든

1. 은닉뉴런이 없는 임의의 신경망을 구성한다.
2. 학습에 사용되지 않은 데이터중에서 현재의 신경망에 의해서 에러가 가장 큰 데이터를 일정개수 뽑아낸다.
3. 선택된 데이터중에서 가장 에러를 크게 내는 데이터의 에러가 정해진 값보다 크고, 그 데이터와 가장 가까운 은닉뉴런 사이의 거리가 일정수준 이상인 경우, 새로운 은닉뉴런을 기존의 신경망에 추가한다. 그렇지 않은 경우, 최소 평균 제곱에러(LMS)를 이용해서 신경망의 파라미터 값들을 갱신한다.
4. 선택된 데이터를 현재까지 선택되었던 데이터와 함께 LMS 학습을 한다.
5. 만족할만한 성능을 낼 때까지 2~4를 반복한다.

표 1 : ARAN의 학습 알고리즘

학습 데이터 N개가 C에 들어있고, T는 비어있게 된다.

또, 현재의 신경망을 구조 A, 가중치 W를 가진 (W,A) 라 표현하고, (x_m, y_m)을 m번째 데이터라고 하고, $f(x_m ; W,A)$가 (W,A)인 신경망에서 x_m이 입력으로 들어온 경우의 출력이라고 하자. 그러면, m번째 데이터 에러는 아래와 같이 정의될 수 있다.

$$e_m = \frac{1}{\dim(y_m)} \| y_m - f(x_m ; W,A) \|$$

각 단계에서 학습에 사용할 새로운 데이터는 C에 있는 모든 데이터의 에러를 위와 같이 계산한 후, 그 값이 큰 상위 λ개로 선택한다.

선택된 λ개의 데이터중 현재의 신경망에서 가장 큰 에러 $(e^{\lambda}_{\max})$를 보이는 데이터를 $x_{\max}$라 하자. 은닉 뉴런을 새로 생성하기 위해서는 다음 두 조건을 만족해야 한다.

$$\begin{aligned} \| x_{\max} - c_{nearest} \| &> \delta(t) \\ e^{\lambda}_{\max} &> \varepsilon \end{aligned} \qquad (1)$$

여기서 $\delta(t)$와 ε은 임계값으로서, $\delta(t)$는 은닉 뉴런들 사이의 거리를 제어하는 인자이고 ε은 신경망 출력의 원하는 정확도를 의미한다. $\delta(t)$는 아래와 같이 학습이 진행되는 동안 일정수준씩 감소한다.

$$\delta(t) = \max[\delta_{\max} \times \gamma^t, \delta_{\min}]$$

(1)식의 조건을 만족하여 새로운 k번째 은닉 뉴런을 기존의 네트워크에 추가할 때에는 은닉 뉴런에 관한 파라미터들을 아래와 같이 고정한다.

$$\begin{aligned} h_k &= y_{\max} - f(x_{\max} ; W,A) \\ w_k &= \varkappa \| x_{\max} - c_{nearest} \| \\ c_k &= x_{\max} \end{aligned} \qquad (2)$$

여기서 $c_{nearest}$는 기존에 있는 은닉뉴런 중에서 $x_{\max}$와 가장 가까운 은닉뉴런의 중심값을 나타낸다. h_k는 은닉 뉴런과 출력뉴런 사이의 가중치, w_k와 c_k는 각각, 너비와 중심을 나타낸다. 이는 현재의 신경망으로 학습이 잘 안되는 데이터를 중심으로 하는 은닉뉴런을 만듦으로써 그 데이터를 학습가능하게 만든다. $\varkappa$는 너비를 조절하는 파라미터이다.

만약 (1)의 조건을 만족하지 않는 경우에는 선택된 λ개의 데이터를 LMS로 학습을 한다. 그 후에 선택된 λ개의 데이터를 T에 추가하고 T에 있는 데이터를 가지고 다시 LMS학습을 한다. 학습할 데이터가 x_m인 경우 다음과 같이 중심과, 너비와, 가중치를 갱신한다.

$$\begin{aligned} \triangle h_{oh} &= \alpha \varepsilon m_h \\ \triangle w_h &= \alpha \varepsilon h_h m_h \| x_m - c_h \| \\ \triangle center_h(t) &= 2\alpha(x_m - center_h(t-1)) \varepsilon m_h h_h \\ m_h &= \exp\left(-\frac{1}{w_h^2} \| x_m - c_h \|^2 \right) \\ \varepsilon &= (y_{mo} - f_o(x_m ; W,A)) \end{aligned}$$

3. 실험 및 결과

실험은 UCI 데이터 몇가지에 대해 전처리를 한 [3]에 있는 12개의 분류 문제중 5가지의 벤치마크 문제를 대상으로 RAN과 ARAN을 비교한다. 5개의 문제중 Horse 문제만 3 클래스 문제이고, 나머지는 2 클래스 문제이다. 각각의 문제에 대해 사용가능한 전체 데이터중에서 2/3은 학습 데이타로 1/3은 테스트 데이타로 이용하였다. 각 문제

문제	전체 학습 데이터 크기	학습에 사용된 데이터 크기	은닉 뉴런의 개수	학습 성능(%)	일반화 성능(%)
(RAN) **Cancer** (ARAN)	466	466	5.0	97.1	97.0
		72.4	4.7	83.7	97.0
(RAN) **Diabetes** (ARAN)	512	512	10.4	76.4	74.5
		300	8.5	64.1	74.9
(RAN) **Heart** (ARAN)	614	614	5.0	83.1	79.4
		487	6.0	79.0	80.4
(RAN) **Card** (ARAN)	460	460	11.1	81.1	85.1
		290	10.2	83.9	88.3
(RAN) **Horse** (ARAN)	243	243	5.4	59.7	70.8
		112	7.1	63.0	72.1

표 2 : RAN 과 ARAN의 실험 결과

에 대한 파라미터들은 여러번의 실험을 거쳐 문제에 맞게 적절하게 조절하였다. 특별히 ARAN에서의 λ 값은 전체 학습 데이터를 100으로 나눈값을 내림한 값으로 결정을 하였다.

표 2에서는 RAN과 ARAN을 가지고, 각문제에 대해서 10번의 실험을 한 후 평균을 낸 값들을 보여준다. 각 문제별로 윗줄은 RAN의 결과, 아랫줄은 ARAN의 결과이다. RAN의 실험 결과는 알고리즘대로 모든 데이터를 다 사용한 경우, 최종 신경망의 은닉뉴런의 개수와 학습성능, 일반화 성능등을 보여주고 있으며, ARAN의 실험 결과는 데이터를 능동적으로 선택하면서 학습을 한 후에, 어느 정도 좋은 성능을 보이는 지점에서의 은닉뉴런의 개수 및 그때까지 학습한 데이터의 크기를 조사하였다.

표 2의 ARAN의 결과를 보면, 5문제 전부 모든 데이터를 가지고 학습을 하지 않더라도 충분한 성능을 가질 수 있음을 알 수 있다. 특히, Cancer 데이터의 경우에는 전체 학습 데이터의 약 15%정도만을 가지고도 만족할 만한 성능을 보인다. 한편, ARAN의 실험 결과에서 학습 성능이 일반화 성능보다 떨어지는 이유는 학습이 어려운 데이터를 학습 데이터로 사용하고, 그 데이터에 대해서 성능을 측정하기 때문이다. 즉, 비교적 어려운 데이터에 대해서 성능을 측정한 것이기에 일반화 성능보다 꽤 떨어지는 경향을 보인다. 만약, ARAN을 끝까지 실행시켜서 전체 데이터를 전부 사용한 경우에는 학습성능이 일반화 성능보다 더 높아지게 된다.

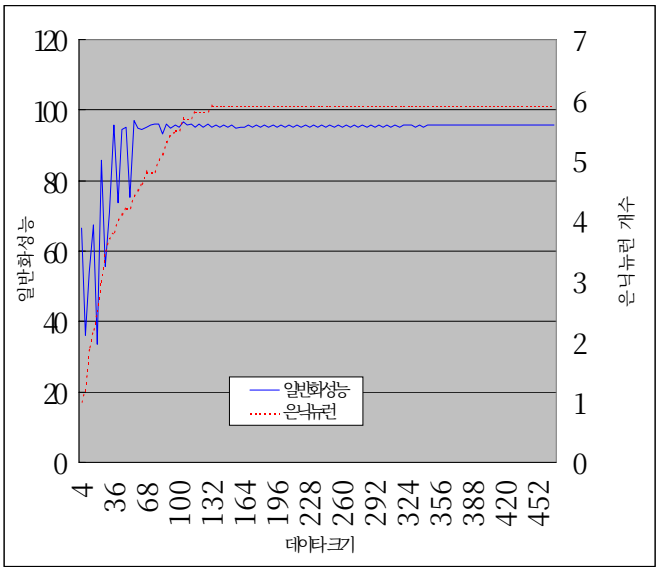

그림 1 : cancer 문제에 대한 ARAN의 실험결과
x 축: 학습시 사용된 데이터의 개수
y 축: 일반화 성능 및 은닉뉴런 수

그림 1은 Cancer 데이터에 대한 ARAN의 전체 실험 결과로서 학습시 사용된 데이터의 크기와 그 때의 신경망의 일반화 성능 및 은닉뉴런의 개수의 관계를 나타낸 것이다. 10번의 실험에 대한 평균값을 취했기 때문에 은닉뉴런의 수가 정수로 늘어나지 않고 있다. 그림을 보면 처음에는 학습시 사용한 데이터가 증가할수록 일반화 성능 및 은닉뉴런의 수가 증가하지만, 일정수준이 지나면 둘다 거의 변화가 없는 모습을 보인다. 이는 Cancer 문제를 푸는데 모든 데이터가 필요하지 않음을 보여주는 한편, 은닉뉴런의 개수도 적절하게 찾아짐을 알 수 있다. 따라서 ARAN의 파라미터를 알맞게 조절하면 최소한의 데이터를 가지고 학습을 할 수도 있을 뿐만 아니라, 문제에 적당한 신경망의 크기도 구할 수 있게 될 것이다.

4. 결론

본 논문에서는 데이터를 능동적으로 선택하면서 네트워크의 크기를 필요에 따라서 증가시키는 RBF 신경망을 제안하였다. 능동적인 데이터 선택은 현재의 신경망에서 에러를 가장 크게 내는 데이터를 취하는 방법을 사용하였다. 그림 1에서 나타나듯이 모든 데이터를 다 사용하지 않고도 다 사용할 때와 비슷하거나 오히려 더 좋은 성능을 보임을 알 수 있다.

본 논문에서는 은닉뉴런을 생성하면서, 신경망의 크기를 늘려가는 알고리즘을 제안하였는데, 새로운 데이터가 계속해서 들어오는 경우, 이전에 만들어진 은닉뉴런이 필요가 없는 경우도 생길 것이다. 그러한 경우 [2]에서 제안된 은닉뉴런 제거 방법등을 참고하여, 신경망이 지나치게 커지는 현상을 방지하면서, 더 좋은 성능을 낼 수 있도록 하는 것이 향후 연구되어야할 과제이다.

감사의 글

본 연구는 과학기술부 뇌연구개발사업(BR-2-1-G-06)에 의하여 일부 지원되었음.

참고 문헌

[1] John Platt, A Resource-Allocating Network for Function Interpolation, *Neural Computation Vol 3 No. 2,* pp. 213-225, 1991.

[2] Lu Yingwei et al, Performance Evaluation of a Sequential Minimal Radial Basis Function (RBF) Neural Network Learning Algorithm, *IEEE Transactions on Neural Networks Vol 9 No.2,* pp. 308-318, 1998.

[3] Lutz Prechelt, PROBEN1-A set fo neural network benchmark problems and benchmarking rules, *Fakultät für Informatik, Univ. Karlsruhe, Germany, Tech. Rep. 21/94,* 1994.

[4] B.T. Zhang, Learning by Incremental Selection of Critical Examples, *Arbeitspapiere der GMD, No 735, German National Research Center for Computer Science (GMD), St. Augustin/Bonn,* 1993.

[5] B.T. Zhang, Accelerated Learning by Active Example Selection, *International Journal of Neural Systems 5(1),* pp. 67-75, 1994.

[6] David A. Cohn et al, Active Learning with Statistical Models, *Journal of Artificial Intelligence Research 4,* pp. 129-145, 1996.

[7] M. Plutowski et al, Selecting concise training sets from clearn data, *IEEE Transactions on Neural Networks, 4,* pp 305-318, 1993.