공학석사학위논문

능동 데이터 선택에 의한 RBF 신경망의 건설적 학습

Learning Constructive RBF Networks by Active Data Selection

2001년 2월

서울대학교 대학원
전기 · 컴퓨터공학부
박 상 욱

능동 데이터 선택에 의한 RBF 신경망의 건설적 학습

Learning Constructive RBF Network by Active Data Selection

지도교수 장 병 탁

이 논문을 공학석사 학위논문으로 제출함

2000년 10월

서울대학교 대학원

전기 · 컴퓨터공학부

박 상 욱

박상욱의 공학석사 학위논문을 인준함

2000년 12월

위 원 장 김 영 택 (인)

부위원장 장 병 탁 (인)

위 원 김 형 주 (인)

초록

본 논문에서는 데이터를 능동적으로 선택하고, 그 데이터에 적합하게 RBF 은닉 뉴런의 갯수를 증가시키는 신경망의 학습 알고리즘을 제안한다. 데이터를 능동적으로 선택하는 것은 주어진 데이터를 그대로 학습에 사용하는 것이 아니라, 학습자가 필요한 상황에 따라 적절하게 데이터를 제어함으로써 학습 성능에 도움이 되도록 하는 방법이다. 이를 능동 학습 (active learning)이라 한다. 또한 학습 과정에서 은닉 뉴런을 늘리는 것은 주어진 문제에 적합한 복잡도를 갖는 신경망을 찾아내는 것으로서 건설적 학습 (constructive learning) 방법이라고 한다. 이러한 데이터를 선택하는 능동 학습과 적합한 MLP 신경망 모델을 찾는 건설적 학습을 자연스럽게 결합하는 학습 알고리즘은 GENIE에서 제안된 바 있으며, 본 논문에서는 이를 RBF 신경망에 적용하였다.

본 논문에서는 현재의 신경망에서 가장 학습이 어려운 데이터를 선택해서 신경망을 학습하고, 학습한 신경망에 대해서 다시 에러가 가장 큰 데이터를 선택해서 학습시키는 과정을 반복하는 ARAN (Active RAN)을 제안하였다. ARAN은 Platt이 제안한 건설적 학습 알고리즘인 RAN에 GENIE에서의 선택적 능동 학습 기법을 결합한 것이다. 5개의 UCI 데이터에 대해 실험을 해보고, RAN과 성능을 비교한다. 이 실험을 통해 점진적으로 임계 데이터를 선택해서 학습함으로써, 전체 데이터를 다 사용하지 않고도, 전체 데이터를 다 사용한 경우와 비슷한 성능을 보이거나 오히려 더 좋은 성능을 낼 수 있음을 알 수 있다. 또한 CoIL Challenge 2000 데이터에 대해 실험을 해봄으로써, 불균형한 데이터 집합에서 능동 학습이 수동 학습보다 더 좋은 일반화 성능을 가질 수 있음을 보이고, ARAN이 실제 문제를 해결하는데 사용될 수 있음을 보인다. 또한 유전자 정보를 이용해서 백혈병의 종류를 예측하는 문제에 대해서도 적용을 해 보았다.

주요어 : ARAN, 능동 학습, 건설적 학습, RBF 신경망, 임계 데이터, 불균형 데이터 집합

학번 : 99419-514

차례

1. 서론 **1**
1.1 연구 배경 1
1.2 연구 내용 2
1.3 논문의 구성 3

2. 관련 연구 **4**
2.1 RBF 신경망 4
2.2 건설적 학습 7
2.3 RAN 9
2.4 능동 학습 12

3. ARAN **15**
3.1 학습 알고리즘 15
3.2 신경망의 구성 18
3.3 다양한 능동 데이터 선택 방법 20

4. 실험 및 결과 **22**
4.1 UCI 데이터 집합 22
4.2 CoIL Challenge 2000 데이터 집합 27
4.3 CAMDA'00 데이터 집합 30

5. 결론 **34**
참고 문헌 35

표차례

표 1 : MLP와 RBF의 비교 4
표 2 : RAN의 학습 알고리즘 11
표 3 : ARAN의 능동 데이터 학습 기법 21
표 4 : 실험에 사용된 5개의 UCI 데이터 집합 22
표 5 : 각 문제에 대한 ARAN의 파라미터 설정 23
표 6 : RAN과 ARAN의 실험 결과 24
표 7 : CoIL Challenge 2000 데이터 집합의 입력 속성 27
표 8 : CoIL Challenge 2000 결과 28
표 9 : CoIL Challenge 2000 데이터 집합 실험 결과 29
표 10 : 정보이론에 기반해서 선택한 10개의 유전자 31
표 11 : 백혈병 종류 예측 실험 결과 32
표 12 : 잘못 분류된 샘플 분석 32
표 13 : 전체 10번의 실험에서 선택 순위가 모두 상위 10위안에 드는 샘플 .. 33

그림차례

그림 1 : RBF **신경망의 구조** 5
그림 2 : Cascor**의 건설적 학습 과정** 9
그림 3 : ARAN**의 순서도** 15
그림 4 : ARAN**의 은닉 뉴런 추가 과정** 19
그림 5 : cancer **문제에 대한** ARAN**의 실험결과** 26
그림 6 : card **문제에 대한** ARAN**의 실험 결과** 26
그림 7 : **두 번째 데이터 집합에 대한 학습 데이터 크기와 성능과의 관계** 33
그림 8 : **세 번째 데이터 집합에 대한 학습 데이터 크기와 성능과의 관계** 33

1. 서론

1.1 연구 배경

전통적인 RBF (Radial Basis Function) 신경망 학습 방식은 입력 데이터의 특징을 이용해서 미리 은닉 뉴런의 개수와 그것의 파라미터를 고정시킨 후, 은닉 뉴런과 출력 뉴런 사이의 연결 가중치는 선형 최소 제곱 방법 (least mean square error, LMS)을 이용해서 추정한다. 이러한 방법을 이용하면, 처음에 문제에 적합한 은닉 뉴런의 개수를 결정하는 일이 쉽지 않을 뿐만 아니라, 새로운 데이터가 계속해서 들어오는 경우에 새로운 신경망의 크기와 파라미터들을 새로 결정해야하기 때문에 순차적 학습 또는 온라인 학습에 사용될 수 없다는 단점이 있다.

이러한 약점을 극복하기 위해, 데이터를 순차적으로 입력받고 그것에 기반해서 자동으로 은닉 뉴런의 개수와 파라미터를 결정하는 알고리즘인 RAN (Resource-Allocating Network)이 개발되었다[Platt91]. 이 알고리즘은 은닉 뉴런이 없는 상태로 신경망을 시작하여 순차적으로 들어오는 데이터를 보고 적당한 기준에 의해서 새로운 은닉 뉴런을 만들어낸다. RBF 신경망에 건설적 학습 방법을 적용하여 신경망의 가중치뿐만 아니라, 신경망의 구조까지 학습시키고자 했다.

한편, RAN은 가지고 있는 모든 데이터를 순차적으로 한번씩 사용한다. 그러나 데이터는 어느 정도 중복이 되어 있으므로 모든 데이터를 학습에 사용할 필요는 없다[Zhang91a, Zhang94a]. 또한 모든 데이터가 학습에서 중요도가 똑같지 않으므로, 중요한 데이터에 초점을 맞추어 학습을 하는 것이 성능에 도움을 줄 수 있다. 본 논문에서는 이러한 동기에서 제한된 학습 데이터 집합으로부터 새로운 학습 데이터를 반복적으로 선택하는 능동 데이터 선택 기법을 RAN에 적용하였다. 따라서 본 논문에서 새로 제안한 알고리즘을 ARAN (Active RAN)이라고 명명하였다[Park00].

1.2 연구 내용

본 논문에서는 능동 데이터 선택 기법과 RBF 신경망의 구조를 필요에 따라 늘리는 건설적 학습 알고리즘과 결합하는 새로운 알고리즘을 제안한다. 이는 앞 절에서 언급한 바와 같이 RBF 신경망 기반의 건설적 학습 알고리즘인 RAN에서 능동 데이터 선택 기법을 추가한 것으로 볼 수도 있고, 또한 능동 학습과 건설적 학습을 하는 GENIE [Zhang93, Zhang94b, Zhang96]의 학습 알고리즘에서 MLP (Multi-layer Perceptron) 대신 RBF 신경망을 이용한 것으로 볼 수도 있다. 이 알고리즘을 통하여 자연스럽게 적절한 RBF 신경망의 구조를 찾아낼 수 있으며, 데이터를 전부 사용하지 않고도 좋은 학습 성능을 낼 수 있음을 보인다.

제안한 알고리즘의 성능을 평가하기 위해 크게 세 가지의 실험을 하였다. 첫째, 기계학습에서 학습 모델의 성능을 평가하기 위해 자주 사용되는 UCI 데이터 집합 중 5개의 데이터 집합을 선택해서 ARAN을 적용하였다. 사용된 데이터 집합으로는 진단 데이터로서 위스콘신 유방암 환자 데이터, 당뇨병 환자 데이터, 심장병 환자 데이터가 있고, 카드사용 허가에 대한 데이터와 말(Horse) 관련 데이터가 있다. 이 데이터는 UCI 기계 학습 데이터베이스 사이트에서 자유롭게 받을 수 있다 (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/). 둘째, 2000년 5월에 열렸던 CoIL (Computational Intelligence and Learning) Challenge 2000에서 사용된 데이터를 가지고 실험을 하였다. 이 데이터 집합은 센티언트 머신 리서치(Sentient Machine Research)라는 네덜란드 데이터 마이닝 회사에 의해 제공된 것으로서 이동식 주택 구매 고객에 관한 정보를 가진다. 이 데이터 집합으로 풀어야 하는 문제는 고객의 제품 구매 정보와 인구 통계학 정보를 이용해서 이동식 주택의 보험 상품 구매 여부를 맞추는 것이다. UCI 데이터 집합을 이용한 실험은 RAN과 비교해서 ARAN의 장점을 보여주었고, CoIL Challenge 2000 데이터 집합을 이용한 실험에서는 ARAN이 실제 상거래 문제를 푸는데 적용될 수 있는 가능성을 제시하였다. 셋째, CAMDA'00 (Critical Assessment of Techniques for Microarray Data Mining) 학회에서 사용한 데이터를 이용해 실험을 하였다. 이 데이터는 각 암

환자에 대한 7129개의 유전자 발현 정보와 함께 2가지 암의 종류중 어떤 병에 걸렸는지가 나와있다. 여기서 우리는 암환자의 유전자 정보를 이용해서 어떤 암에 걸렸는지를 판별해야 한다. ARAN에 이 문제를 적용해 보았더니 만족할 만한 결과가 나왔다.

1.3 논문의 구성

구성은 다음과 같다. 2장에서는 관련 연구로서 ARAN의 기본이 되는 RBF 신경망과 건설적 학습, RAN의 학습 알고리즘과 능동적 학습에 대해서 살펴본다. 3장에서는 본 논문에서 제안한 ARAN의 구성과 학습 방법 및 다양한 능동 학습 기법에 대해서 자세히 살펴본다. 4장에서는 ARAN을 5개의 UCI 데이터 문제를 푸는데 이용해 본 결과를 보여주고, CoIL Challenge 2000 데이터 집합과 CAMDA'00 데이터 집합에 대한 실험 결과도 보여준다. 5장에서는 본 연구 내용을 요약하고, 의의 및 문제점을 설명하며 결론을 짓는다.

2. 관련 연구

서론에서 언급한 바와 같이 본 논문에서 제안한 ARAN은 RAN 알고리즘에 능동 학습 기법을 추가하여 개선한 것으로 볼 수 있다. RAN 알고리즘은 신경망의 구조를 필요할 때마다 늘려서 원하는 성능을 낼 수 있도록 하는 건설적 학습 알고리즘이다. 이에 이 장에서는 RAN의 기본이 되는 RBF 신경망에 대해서 살펴보고, 건설적 학습 방법에 대해서도 살펴보도록 한다. 또한 RAN과 능동 학습 방법에 대해서도 차례로 살펴본다.

2.1 RBF 신경망

기계학습의 도구로서 신경망을 사용할 때, 가장 많이 연구가 되고, 많이 사용되는 것은 다층 퍼셉트론(MLP)이다. 그러나 전통적인 MLP는 학습 속도가 느리고 학습된 최종 신경망이 표현하는 지식을 이해하기 어렵다는 단점이 있다. 이에 RBF 신경망이 제안되었는데, 이것은 은닉 뉴런들이 확률 분포 함수를 이룰 수 있고, 학습 속도도 빠를 수 있으면서 신경망이 표현하는 지식을 어느 정도 이해할 수 있다는 장점이 있다[Tao93]. 아래 표 1에서 다층 퍼셉트론과 RBF 신경망을 비교 정리하였다.

표 1 : MLP와 RBF의 비교

	MLP	RBF
은닉 층의 수	1 or more	1
활성화 함수	시그모이드	가우시안
활성화 함수 인자	벡터 내적	유클리드 거리
비선형 매핑	전역적	지역적

우선 은닉 층의 수를 보면 RBF는 하나만 가지는데 비해서 MLP는 하나 이상의 은닉 층을 가질 수 있다. 또한 은닉 뉴런의 활성화 함수로 MLP에서는 시

그모이드 함수를 취하는 데 비해, RBF에서는 보통 가우시안 함수를 사용한다. 또한, 은닉 뉴런의 함수 인자로는 MLP에서는 벡터 내적을 사용하는데, RBF에서는 유클리드 거리를 이용한다. RBF와 MLP 모두 보편적인 추정 모델로 사용할 수 있는데, 보통 RBF는 지역적인 비선형 관계를 추정하고, MLP는 전역적인 비선형 관계를 추정한다[Haykin98].

RBF 신경망의 기본 구조는 그림 1과 같다. 그림에서는 $f: R^m \rightarrow R$ 문제를 풀기 위해 k개의 은닉 뉴런을 가지는 RBF 신경망을 보여준다. 설명을 위해 몇 가지 표기법을 정의하기로 한다. 먼저 입력 차원이 m이므로 입력 패턴은 $x = (x_1, x_2, ..., x_m)$라 놓자. 그림 1에서, c_j는 j번째 은닉 뉴런의 중심, w_j는 너비이다. h_0는 출력 유닛의 바이어스이며, h_j는 j번째 은닉 뉴런과 출력 유닛 사이의 가중치이다. 각각의 입력이 은닉 뉴런과 전부 연결되어 있고, 은닉 뉴런에서는 가우시안 함수에 의해 해당 입력 패턴에 대한 출력을 내보낸다. 출력 유닛에서는 은닉 뉴런의 출력 값에 각 은닉 뉴런의 가중치를 곱해서 더한 값을 최종 신경망의 출력으로 내보낸다.

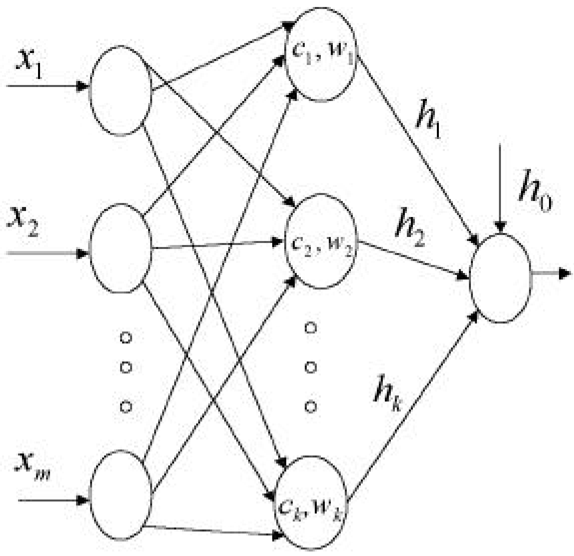

그림 1 : RBF 신경망의 구조

출력 유닛이 하나인 RBF 신경망의 출력 값은 다음과 같이 계산될 수 있다.

$$y = \sum h_j z_j + h_0 \qquad (1)$$

여기서,

$$z_j = \exp\left(- \frac{\| c_j - x \|}{w_j^2}\right) \qquad (2)$$

이다.

RBF 신경망의 각 파라미터들을 학습시키는 방법으로 [Haykin98]에서는 다음 세 가지 해결책을 제시하고 있다. 첫 번째 방법은 임의로 중심을 훈련 데이터 집합에서 일정 개수 선택하고, 너비는 주어진 문제의 입력 차원에 비례해서 고정시킨 다음, 은닉 뉴런과 출력 유닛 사이의 가중치는 pseudoinverse 방법 [Broomhead88]에 의해 찾아낸다. 이 방법은 MLP의 경사 하강법과 달리, 간단한 계산만으로 학습이 되기 때문에 속도가 매우 빠르다. 두 번째 방법은 SOM과 같은 클러스터링 알고리즘을 이용하여 훈련 데이터 집합에서 적절한 클러스터를 찾아낸 후 이를 중심으로 하는 방법이다. 가중치는 최소 평균 제곱 에러 (Least mean square error, LMS) 방법을 이용하여 추정을 한다. 임의로 중심을 선택하는 앞의 방법에 비해서 중심을 보다 적절하게 찾기 때문에 성능의 향상을 꾀할 수 있다. 세 번째 방법은 첫 번째 방법에서처럼 중심과 너비를 임의로 정하지만, 학습은 LMS를 이용하여 가중치와 함께 중심과 너비도 경사 하강법 (Gradient Decent)을 적용하여 변경한다. 이 방법은 MLP의 학습 방법과 유사해서 학습 속도가 느리지만 적절한 가중치를 찾을 수 있게 된다.

이 밖에도 진화 연산을 이용해서 RBF 신경망을 학습하는 방법[Billings95, Burdsall97], EM (Expectation-Maximization) 알고리즘을 이용해 RBF 신경망의 파라미터를 추정하는 방법[Ungar94] 등이 있다.

2.2 건설적 학습

건설적 학습의 기본 아이디어는 작은 신경망으로 시작해서 만족할만한 해답이 나올 때까지 은닉 뉴런의 크기를 증가시키고 가중치를 학습시키는 것이다. 따라서 이 방법을 이용하면, 학습을 시작하기 전에 신경망의 구조를 미리 결정할 필요가 없으며 고정된 신경망을 학습할 때 생기는 문제를 해결할 수 있다. 즉, 신경망을 문제의 복잡도보다 작게 놓고 학습을 한다면, 그 문제를 제대로 학습할 수 없는 반면에, 너무 크게 놓고 학습을 하는 경우에는 오버피팅(Overfitting)으로 인한 일반화 성능의 저하 문제를 가져온다. 건설적 학습 알고리즘은 은닉 뉴런을 유동적으로 증가시킴으로써 문제에 맞는 신경망을 구성할 수 있게 하고, 어떤 문제라도 풀 수 있는 가능성을 제공한다[Baum89].

은닉 뉴런의 수를 학습 도중에 유동적으로 조정하는 방법으로는 건설적 학습 알고리즘과 반대되는 개념으로 가지치기 학습 알고리즘(Pruning algorithm)이 있다. 즉, 필요한 신경망보다 더 큰 구조를 가진 신경망으로 학습을 한 후에, 필요 없는 은닉 뉴런을 제거해 가는 방법이다[Reed93]. 가지치기 알고리즘에 비해서 건설적 학습 알고리즘이 가지는 장점은 다음과 같다. 첫째, 건설적 학습 알고리즘은 초기의 신경망을 은닉 뉴런이 없는 상태로 학습하는 것으로 결정되어 있지만, 가지치기 알고리즘은 초기의 신경망의 크기를 결정하기가 모호하다. 둘째, 건설적 학습 알고리즘은 작은 크기의 신경망에서 적절한 크기의 신경망을 찾아가기 때문에, 큰 크기의 신경망에서 적절한 크기의 신경망을 찾아가는 가지치기 알고리즘에 비해서 학습 시간이 상대적으로 짧아진다[Kwok97].

하지만, 건설적 학습 알고리즘에서도 신경망의 크기를 언제까지 늘릴 것인가를 결정해야 하는 문제점이 있다. 좋은 일반화 성능을 가지기 위해서는 적절한 크기의 신경망을 구성해야하므로, 건설적 학습 알고리즘에서는 은닉 뉴런을 증가시키는 작업의 종료시점을 잘 결정해야 한다. .

대표적인 건설적 학습 알고리즘으로는 MLP기반의 Cascade-Correlation (Cascor) [Fahlman90] 와 RBF 기반의 RAN이 있다. 이 둘의 또 다른 특징은 Cascor가 새로 생성한 은닉 뉴런만을 학습시키는 반면에 RAN은 새로 생성된

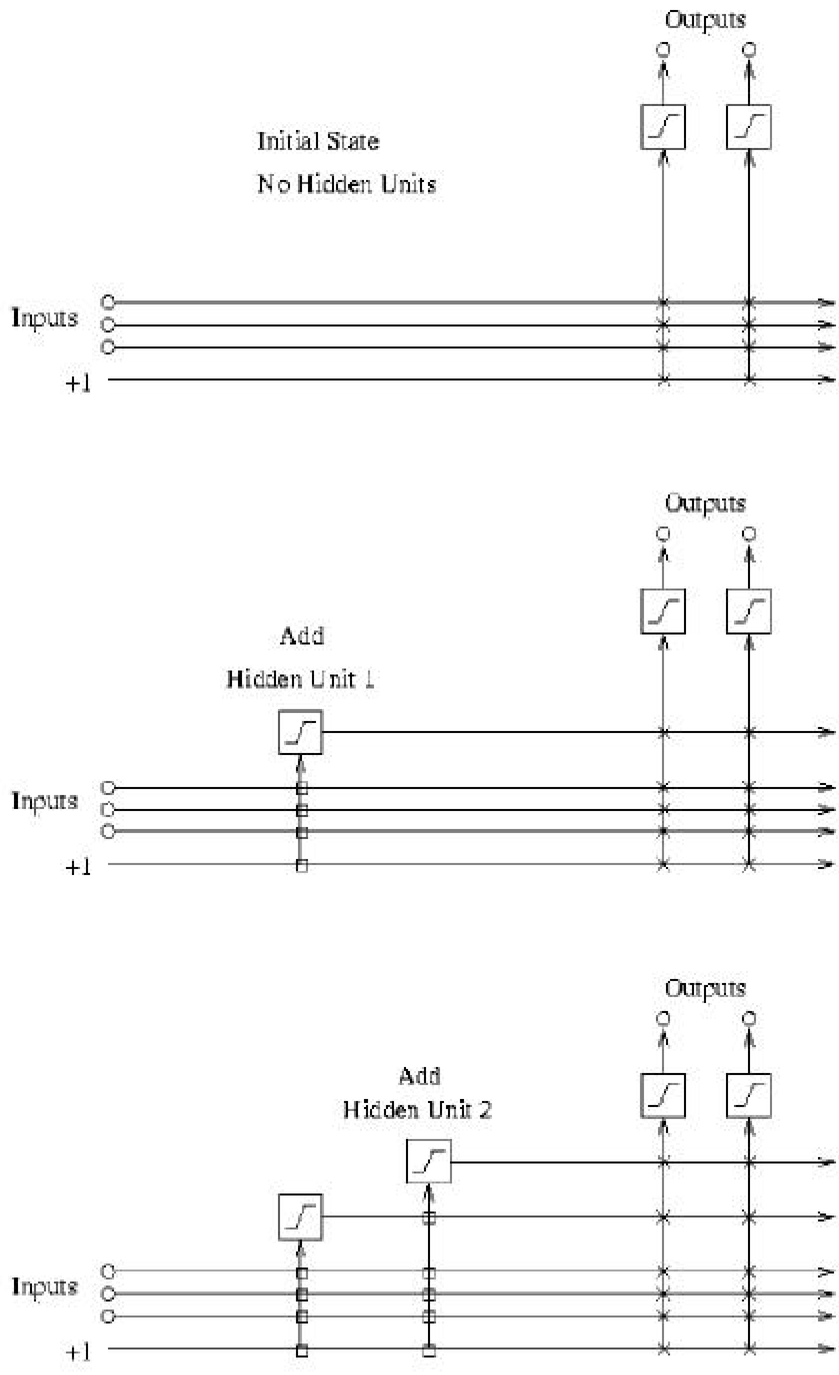

그림 2 : Cascor**의 건설적 학습 과정**

은닉 뉴런 이외에 다른 뉴런까지 같이 학습한다는 점이다. 새로운 은닉 뉴런이 생성될 때마다 전체 네트워크를 다시 학습하는 경우에는 새로운 은닉 뉴런만을 학습하는 경우보다 학습 시간은 오래 걸리지만, 지역 최적화에 빠질 가능성이 더 작다.

그림 2는 Cascor가 은닉 뉴런을 하나씩 추가해 가는 과정을 나타낸 것이다. Cascor는 새로운 은닉 뉴런만 학습을 하기 때문에 학습 속도가 빠른 반면에, 지역 최적화에 빠질 염려가 있다. 그러나 하나의 은닉 뉴런이 신경망의 하나의 층을 이루고 있어서 복잡한 특성을 파악하는데 적합할 수 있다. 간략한 알고리즘은 다음과 같다. 우선 은닉 뉴런이 없는 임의의 신경망을 구성한 후에 그것의 성능을 측정한다. 측정된 성능이 만족할만하지 않은 경우, 새로운 은닉 뉴런을 그 출력 값이 에러와 상관관계가 높도록 만든 후, 기존의 신경망에 추가한다. 다시 학습을 하고, 성능이 만족할만한 경우 멈추고 그렇지 않은 경우 다시 새로운 은닉 뉴런을 추가한다. 이 때, 추가된 은닉 뉴런만을 학습하며, 기존에 만들어진 가중치는 고정시킨다.

한편, RAN은 Cascor와 다르게 새로운 은닉 뉴런이 만들어질 경우에 전체 네트워크를 다시 학습하며, 여러 층을 구성하는 게 아니라 일반적인 RBF 신경망처럼 하나의 은닉 층을 가진다. RAN은 본 논문의 중요한 동기가 되는 알고리즘이므로 다음절에서 자세하게 설명하기로 한다.

이외에도 Dynamic Node Creation (DNC) 알고리즘, Projection Pursuit Regression (PPR), Group Method of Data Handling (GMDH) 등이 존재한다 [Ash89, Friedman81, Farlow84].

2.3 RAN

RAN은 구조상 앞의 그림 1에서와 같이 RBF 신경망을 기반으로 한다. 따라서 RBF 신경망과 똑같이 입력 층으로부터 입력 패턴을 받고, 은닉 층에서 입력 패턴의 프로토타입을 중심으로 저장하고 있어서 현재의 입력 패턴과 중심의 거리에 따라 출력을 내보낸다. 은닉 뉴런의 가중치는 은닉 뉴런이 반응하는 입력 패턴의 범위를 지정한다. 은닉 뉴런은 거리가 가까울수록 높은 값을 내보

내도록 다음과 같은 가우시안 함수를 활성화 함수로 둔다.

$$z_j = \exp\left(- \frac{\| c_j - x \|}{w_j^2}\right) \qquad (3)$$

출력 층에서는 아래와 같이 각 은닉 뉴런의 출력 값의 가중치를 곱해서 더한다.

$$y = \sum h_j z_j + h_0 \qquad (4)$$

이 식들은 앞에서 설명한 RBF 신경망의 활성화 함수(수식 2)와 출력 값(수식 1)과 같다. 반면에 RAN이 기존의 RBF 신경망과 다른 점은 자동으로 은닉 뉴런의 개수를 조절하며, 그것들의 중심과 너비를 에러에 기반 하여 조절한다는 것이다.

다음으로 학습 알고리즘을 살펴보자. 간단한 알고리즘은 표 2에서 보여준다. 처음에는 은닉 뉴런이 없는 임의의 신경망으로 시작을 한다. 입력 패턴이 신경망에 들어오는 경우 아래 두 가지 조건에 따라 새로운 은닉 뉴런을 만들 것인지 결정을 한다.

$$\| x - c_{nearest} \| > \delta(t) \qquad (5)$$

$$\| T_x - O_x \| > \varepsilon \qquad (6)$$

여기서 T_x는 입력이 x인 경우에 목표 값이고, O_x는 x 입력이 들어온 경우에 신경망의 출력 값이다. 첫 번째 조건(수식 5)은 입력 패턴과 가장 가까운 은닉 뉴런 사이의 거리가 일정 수준 이상이 되어야 함을 의미한다. 두 번째 조건(수식 6)은 일정 수준 이상의 에러가 존재할 때에만 새로운 은닉 뉴런을 생성함을 의미한다.

수식 (6)의 ε은 RBF 신경망의 원하는 정확도를 조절한다. 에러가 ε보다 큰 경우에는 즉시 은닉 뉴런을 추가하여 바로 에러를 줄이도록 유도하고, 그렇지 않은 경우에는 경사 하강법(gradient decent rule)을 이용하여 점차로 에러를 줄이도록 한다. 한편, 수식 (5)의 $\delta(t)$는 시간에 따라 감소하는 값으로 δ_{max}로 시작해서 δ_{min}이 될 때까지 다음과 같은 값을 가지게 된다.

$$\delta(t) = \max[\delta_{max} \exp(-t/\tau), \delta_{min}] \qquad (7)$$

수식 (7)의 τ는 감쇄 상수이다. 위와 같은 식을 만들게 된 이유는 처음에는 전체적인 학습을 하고 나중에 좀더 정확한 학습을 할 수 있도록 하기 위해서이다.

표 2 : RAN의 학습 알고리즘

(초기화)
은닉 뉴런이 없는 신경망을 임의로 초기화한다.
 	
(반복)
모든 입출력 쌍에 대해서 다음 과정을 반복한다.
{
주어진 입력 x에 대해서 신경망의 출력 값을 계산한다.
목표 값과 비교하여 에러(E)를 계산한다.
x와 가장 가까운 은닉 뉴런의 중심 사이의 거리(d)를 구한다.
만약 $\|\|E\|\| > \varepsilon$ 이고 $d > \delta$ 이면 다음과 같이 j번째 은닉 뉴런을 생성한다.
$c_j = x$
$w_j = \chi\|\|x - c_{nearest}\|\|$
$h_j = E$
위의 조건을 만족시키지 못한 경우에는 신경망의 구조는 그대로 두고 중심과 너비, 가중치를 수식 (8)과 같이 경사하강법을 이용해 학습한다.
δ값을 갱신한다.
}

수식 (5)과 (6)로 표현되는 두 개의 은닉 뉴런 생성을 위한 조건은 가장 최적의 RBF 신경망을 구성하기 위해서 문제에 맞게 적절하게 조절해야 한다.

은닉 뉴런 생성 조건이 성립하면, 새로운 은닉 뉴런의 중심은 새로운 입력 패턴으로 하고, 너비는 가장 가까운 은닉 뉴런과의 거리에 비례하는 값으로 하고, 가중치는 에러 값으로 놓는다. 조건이 성립하지 않는 경우에는 새로운 은닉 뉴런을 생성하지 않고, 경사 하강법을 사용해서 각각의 파라미터들을 다음과 갱신시킨다[Platt91]. 여기서 α는 학습률이다.

$$h_j(t+1) = h_j(t) + \alpha E\left\{\exp\left(-\frac{\|x - c_j(t)\|^2}{w_j(t)^2}\right)\right\}$$

$$c_j(t+1) = c_j(t) + 2\frac{\alpha}{w_j(t)}(x - c_j(t))E\left\{h_j(t)\exp\left(-\frac{\|x - c_j(t)\|^2}{w_j(t)^2}\right)\right\} \quad (8)$$

$$w_j(t+1) = w_j(t) + \alpha E\left\{h_j(t)\exp\left(-\frac{\|x - c_j(t)\|^2}{w_j(t)^2}\right)\|x - c_j(t)\|\right\}$$

RAN을 개선한 것으로는 CC-RAN (Coarse Coding RAN)[Deco93] 과 M-RAN (Minimal RAN)[Yingwei98] 등이 있다. CC-RAN은 입력 차원을 한 차원씩으로 나누어 저장하고 이를 공유함으로써 더 압축된 형태의 네트워크를 구성할 수 있게 하였고, M-RAN에서는 온라인 학습에서 RAN이 계속해서 증가하는 경향을 막고자 필요 없는 은닉 뉴런을 제거하는 기능을 추가하였다.

2.4 능동 학습

일반적인 신경망 알고리즘에서는 학습 데이터가 외부 환경이나 외부 실험자에 의해 모두 주어진다고 가정한다. 따라서, 신경망의 학습은 데이터에 신경을 쓰지 않고, 오직 신경망의 자유 변수들의 조정에 초점을 맞추게 된다. 반면에 능동 학습에서는 학습자가 자신의 학습 데이터를 스스로 선택하거나 혹은 자신이 학습 데이터에 어떤 영향력을 행사할 수가 있다고 가정한다[Zhang91b, Plutowski93]. 보통 학습의 문제는 훈련 데이터에 기반하여 그 입력과 출력 사

이의 대응 관계를 찾아내는 것으로 생각할 수 있다. 이 때, 능동 학습자는 제한된 데이터 집합으로부터 새로운 입력을 반복적으로 선택하고 그 결과 값을 관찰한 후 새로운 데이터 집합을 자신의 훈련 데이터에 포함시키는 행위를 반복한다.

능동 학습에 있어 가장 중요한 문제는 새로운 데이터를 선택하는 방법이다. 그러한 방법으로는 현재 훈련 데이터가 존재하지 않는 곳에서 새로운 데이터를 선택하는 방법[Whitehead91], 성능이 나쁘거나 신뢰성이 떨어지는 곳에서 데이터를 선택하는 방법[Zhang94], 현재의 모델을 변경시킬 수 있을 만한 데이터를 선택하는 방법[Cohn90], 총괄적인 분산이 최소화될 수 있는 데이터를 선택하는 방법[Cohn96] 등이 있다. 한편 결정적인(deterministic) 알고리즘이 지역 최적화에 빠지는 문제가 있기 때문에 확률적인 능동 학습 방법도 제안되었다[Fukumizu00].

능동 학습은 학습에 사용된 데이터를 다음에 다시 선택할 수 있는가에 따라 증가적(incremental) 학습과 선택적(selective) 학습으로 나눌 수가 있다. 즉 학습에 사용할 데이터 집합에서 데이터를 선택하고 학습에 이용한 후 제거하는 학습 방법(incremental)과 제거하지 않는 방법(selective)이 있는 것이다 [Adejumo99]. [Zhang94]는 증가적 학습(incremental learning) 방법을 취하고, [Engelbrecht98]는 선택적 학습(selective learning) 방법을 취한다. ARAN은 알고리즘의 안정성을 위해 증가적 능동 학습 기법 (incremental learning)을 사용한다. 그러나 필요에 따라 선택적 학습 기법을 사용할 수 있게 하였다.

능동 학습 기법은 훈련 데이터 집합의 데이터들이 중복이 많이 되어 있는 경우에 학습 시간을 줄이는 데 사용될 수 있다. [Leisch98]에서는 Cross validation을 수행해서 모델의 에러를 추정할 때 훈련 데이터 집합을 바꾸어가면서 실험을 해야 하기 때문에, 시간이 많이 걸리는 문제를 능동 데이터 선택 기법을 이용하여 해결하였다. 한편, [Liere97]에서는 텍스트 분류 문제를 풀기 위해 위원회(Committee)를 구성하고, 위원회 모델 중 임의로 두 개를 선택해서 두 모델이 다른 출력 값을 내는 데이터만 학습을 하도록 하는 능동 학습 기법을 이용함으로써 성능과 속도 면에서 우수한 결과를 보였다. 또한, 목표

값이 없는 데이터를 학습하는 경우 Committee간에 출력 값이 다른 데이터를 신뢰도가 높은 멤버가 목표 값을 결정한 후 그 데이터를 학습에 이용하는 능동 학습 기법이 이용되기도 하였다[McCallum98]. 중요한 데이터만을 선택함으로써, MLP의 학습 속도와 일반화 성능을 개선하기도 하였다[Zhang94a, Röbel94].

3. ARAN

RAN 알고리즘은 주어진 모든 훈련 데이터를 순차적으로 한번씩 사용한다. 그러나 데이터는 어느 정도 중복된 정보를 가지게 되므로 모든 데이터를 학습에 사용할 필요는 없다[Zhang94a]. ARAN은 중요한 데이터만을 능동적으로 선택함으로써 모든 데이터를 학습하는데 다 사용하지 않고도 충분한 성능을 가질 수 있다. RAN과 달라진 점을 중심으로 ARAN의 학습 알고리즘 및 구성에 대해서 살펴보고, 능동적으로 데이터를 선택하는 다양한 방법을 탐색해보겠다.

3.1 학습 알고리즘

ARAN의 전체적인 학습 알고리즘은 그림 3과 같다. ARAN의 알고리즘을 자세히 기술하기 전에 다음과 같은 몇 가지 용어를 정의하도록 하자. ARAN이 풀려고 하는 문제에서 주어진 전체 데이터의 개수를 N이라 하자. 전체 학습에 사용 가능한 데이터 중에서(N) 아직 학습에 사용되지 않은 데이터 집합을 후

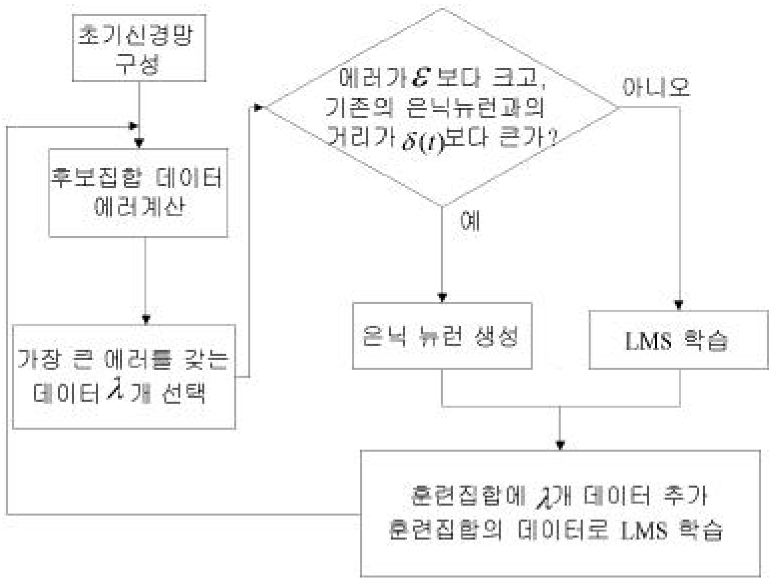

그림 3 : ARAN의 순서도

보 집합이라 하고, 학습에 한번 이상 사용된 데이터의 집합을 훈련 집합이라고 정의한다. 그러면, 각 학습 단계에서 훈련 집합의 데이터 개수와 후보 집합의 데이터 개수의 합은 항상 N이 된다. 초기에는 주어진 모든 학습 데이터 N개가 후보 집합에 들어있고, 훈련 집합은 비어있게 된다. 학습이 진행됨에 따라서 후보 집합의 데이터가 훈련 집합으로 이동하게 된다.

ARAN은 현재까지 학습된 신경망에서 학습이 가장 어려운 데이터를 학습 데이터로 선택함으로써 작은 학습 데이터 집합에서도 효율적으로 학습을 할 수 있도록 하고자 한다. 가장 학습이 어려운 데이터를 임계 데이터라고 한다. 어떤 데이터 (x, y) 가 학습이 얼마나 어려운가 하는 것은 아래와 같은 에러에 의해서 결정이 된다.

$$e = \frac{1}{\dim(y)} \|y - f(x)\| \qquad (9)$$

여기서 $\dim(y)$는 출력의 차원을 나타내고, $f(x)$는 현재의 신경망이 입력 x를 받았을 때의 출력 값을 의미한다.

이제 알고리즘을 살펴보자. 전체적인 흐름은 표 2의 RAN 학습 알고리즘과 유사하다. 그림 3에서처럼 먼저, 은닉 뉴런이 없는 신경망을 임의로 초기화한다. 그 다음, 후보 집합에 있는 데이터 중에서 현재의 신경망에 의해서 에러가 가장 큰 데이터를 일정 개수 뽑는다. 선택된 데이터 중에서 가장 에러를 크게 내는 데이터($x_{\max}(t)$)의 에러($e_{\max}(t)$)가 정해진 값(ε)보다 크고, 그 데이터와 가장 가까운 은닉 뉴런($c_{nearest}$) 사이의 거리가 일정 수준($\delta(t)$) 이상인 경우, 새로운 은닉 뉴런을 기존의 신경망에 추가한다. 그렇지 않은 경우에는 선택된 데이터를 이용해서 최소 평균 제곱 에러(LMS) 방법을 이용해서 신경망의 파라미터들을 변경한다. 선택된 데이터를 훈련 집합에 추가하고, 훈련 집합의 데이터를 가지고 다시 LMS 학습을 한다. 후보집합에 데이터가 존재하지 않을 때까지 앞의 데이터 선택 단계 이후의 과정을 반복한다.

각 단계 t에서 ARAN은 새로운 훈련 데이터로 에러가 가장 큰 λ개의 데이터

를 후보 집합에서 선택한다. 각 단계에서 골라진 λ개의 데이터 중에서 가장 큰 에러를 갖는 데이터를 $x_{\max}(t)$라하고 그 데이터의 에러를 $e_{\max}(t)$라고 하면, 아래의 두 조건이 만족할 때 새로운 은닉 뉴런을 만들어낸다.

$$\|x_{\max} - c_{nearest}\| > \delta(t) \quad (10)$$

$$e_{\max} > \varepsilon \quad (11)$$

이 조건들은 RAN에서의 조건 (5), (6)와 마찬가지로 만드는 RBF 신경망 모델의 복잡도와 정확도를 조절하며, 성능에 중요한 역할을 한다.

한편, 위의 은닉 뉴런 생성을 위한 조건 두 가지를 동시에 만족시키는 경우에는 다음과 같이 새로운 은닉 뉴런을 만들어서 기존의 신경망에 추가한다.

$$c_j = x_{\max}(t) \quad (12)$$

$$w_j = \kappa\|x_{\max}(t) - c_{nearest}\| \quad (13)$$

$$h_j = y_{\max}(t) - f(x_{\max}(t)) \quad (14)$$

즉, 새로운 은닉 뉴런의 중심은 그 단계에서 가장 큰 에러를 내는 데이터로 놓고(수식 12), 너비는 가장 큰 에러를 내는 데이터와 가장 가까운 거리에 있는 은닉 뉴런의 중심 사이와의 거리에 비례한 값(수식 13)으로 놓는다. 또한, 각 은닉 뉴런과 출력 유닛 사이의 가중치는 에러(수식 14)로 놓는다. 이렇게 함으로써, 기존의 신경망이 학습하기 어려웠던 데이터인 $x_{\max}(t)$를 제대로 학습할 수 있도록 도와주고자 한다.

$\delta(t)$는 각 단계를 지날 때마다 RAN에서와 같이 아래처럼 갱신된다.

$$\delta(t) = \max[\delta_{\max}\exp(-t/\tau), \delta_{\min}] \quad (15)$$

은닉 뉴런 생성 조건에 맞지 않는 경우에는 중심과 너비, 가중치 등을 LMS

기법으로 갱신한다. 다음은 데이터 (x, y)를 가지고, j번째 은닉 뉴런을 LMS로 학습할 때의 갱신하는 식이다.

$$
\begin{aligned}
h_j(t+1) &= h_j(t) + \alpha e \exp\left(-\frac{\|x - c_j(t)\|^2}{w_j(t)^2}\right) \\
w_j(t+1) &= w_j(t) + \alpha e h_j(t) \exp\left(-\frac{\|x - c_j(t)\|^2}{w_j(t)^2}\right)\|x - c_j(t)\| \qquad (16) \\
c_j(t+1) &= 2\frac{\alpha}{w_j(t)}(x - c_j(t)) e h_j(t) \exp\left(-\frac{\|x - c_j(t)\|^2}{w_j(t)^2}\right)
\end{aligned}
$$

3.2 신경망의 구성

앞에서 설명한 바와 같이 ARAN 초기의 구조는 은닉 뉴런이 없는 입력과 출력이 바로 연결된 임의의 신경망이다. 그림 4는 학습이 진행되는 동안 증가하는 ARAN의 구조를 보여준다. 그림 4-(a)가 초기화된 ARAN의 모습이고, 그림 4-(b)는 은닉 뉴런이 하나 추가된 경우, 그림 4-(c)는 은닉 뉴런이 두 개 추가된 경우, 그림 4-(d)는 은닉 뉴런이 k개 추가된 경우의 모습이다. 학습이 진행하는 동안 ARAN은 이런 식으로 은닉 뉴런의 수를 늘려가면서 문제에 적합한 신경망의 구조를 찾을 수 있게 된다.

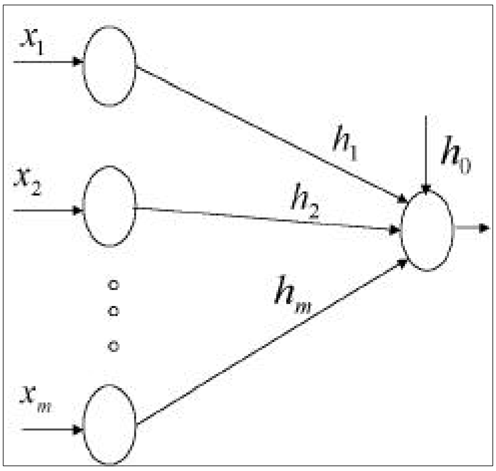

그림 4-(a) : 은닉 뉴런이 없는 경우

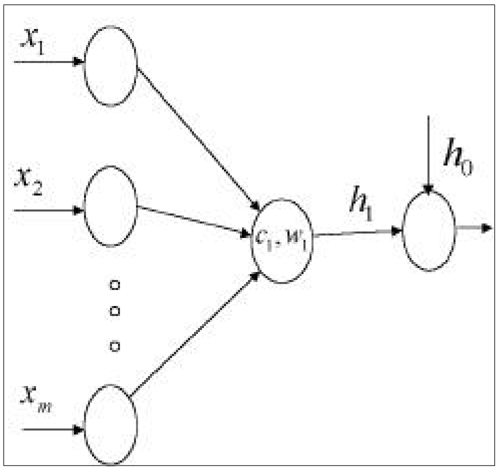

그림 4-(b) : 은닉 뉴런이 하나인 경우

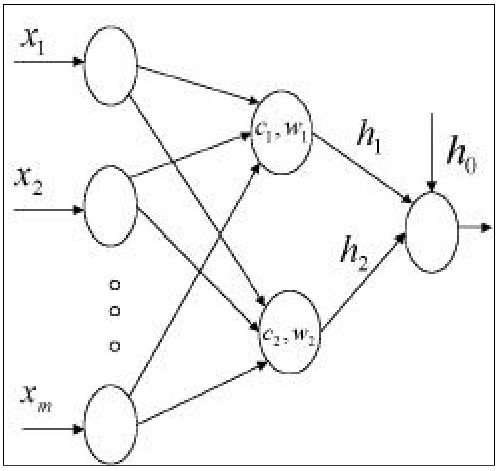

그림 4-(c) : 은닉 뉴런이 두 개인 경우

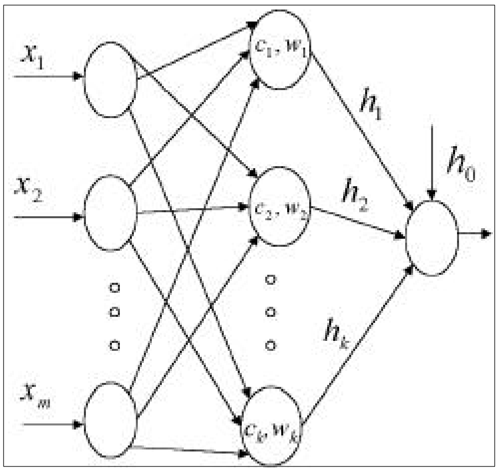

그림 4-(d) : 은닉 뉴런이 k개인 경우

그림 4 : ARAN의 은닉 뉴런 추가 과정

3.3 다양한 능동 데이터 선택 방법

앞 절에서 설명한 ARAN의 능동 데이터 선택 방법은 후보 집합에서 에러가 큰 일정개수의 임계 데이터를 선택하고, 그 데이터는 후보 집합에서 제거되어 훈련 집합에 추가하는 증가적 능동 학습 (incremental learning) 방법이었다. 여기서는 ARAN에 적용해볼 수 있는 몇 가지 다른 능동 데이터 선택 방법들 중에서 선택적 학습 방법을 중심으로 살펴보고자 한다. 기본적으로 선택 학습은 데이터 선택 단계마다 후보 집합에서 선택된 데이터만으로 학습을 시도하고, 다음 단계에서 학습에 사용된 데이터를 포함해서 다시 필요한 데이터를 선택한다.

먼저, 전형적인 선택 학습 (selective learning) 방법으로 매 학습 단계에서 일정 개수의 임계 데이터를 선택하지만 후보 집합에서 제거하지 않는 방법이다. 즉, 데이터를 선택할 때 증가적 학습과 달리, 데이터가 학습에 사용되었는지의 여부와 상관없이 그 순간에 가장 큰 에러를 내는 데이터를 선택한다. 이 방법은 매 학습 순간에서 학습이 잘 되지 않는 데이터를 선택해서 학습함으로써 학습 모델이 그 데이터를 집중적으로 학습할 수 있게 한다.

두 번째 방법은 각 학습 단계에서 선택하는 데이터의 개수를 고정시키지 않고, 무조건 현재의 모델이 잘못된 답을 내보내는 데이터를 학습 데이터로 이용하는 것이다. 이 방법은 학습할 데이터 수를 고정시킴으로써 모델의 학습 능력을 제한한 앞의 방법을 개선시키고자 한 방법이다.

세 번째 방법은 학습 데이터로 선택되는 데이터들이 각 클래스에서 똑같은 수만큼 선택되도록 하는 방법이다. 각 클래스에 속하는 데이터중 에러가 큰 데이터 일정개수를 선택함으로써, 클래스 비율이 불균형한 데이터에서 신경망의 성능이 떨어지지 않도록 하기 위한 방법이다. 또한 이 방법은 모델이 진동하는 현상을 막아주기도 한다.

네 번째 방법은 선택 학습과 증가적 학습을 결합한 방법으로써, 각 단계에서 데이터가 이전에 선택되었는지 여부와 관계없이 데이터를 선택한다. 각 단계에서 선택되는 데이터를 뒤의 학습 단계에서 사용할 수 있도록 훈련 집합에 포함하지만, 후보 집합에서 제거하지 않는 방법이다. 이는 현재 학습이 가장 안

되는 데이터를 선택하는 기능과 함께, 이전에 선택된 데이터도 같이 학습을 함으로써 성능의 안정성을 꾀하는 것이다.

ARAN의 기존의 데이터 선택 기법과 함께 위의 네 가지 방법도 ARAN에 구현이 되어있으며, 풀려고 하는 문제에 적절한 능동 데이터 선택방법을 고르면 된다. 표 3에서는 5가지의 능동 데이터 선택 기법을 정리해 놓았다.

표 3 : ARAN의 능동 데이터 학습 기법

능동 학습 방법	설명
incremental	훈련 데이터로 사용되지 않은 임계 데이터 선택
typical selective	사용여부와 관계없이 매 순간 임계 데이터를 선택
flexible selective	현 모델이 잘못 분류한 데이터를 선택
balanced selective	클래스 별로 일정 개수 임계 데이터 선택
selective incremental	사용여부와 관계없이 선택한 임계 데이터를 누적

4. 실험 및 결과

제안된 알고리즘의 성능을 평가하기 위해 5개의 UCI 데이터 집합과 CoIL Challenge 2000 데이터 집합, CAMDA'00 Conference 데이터 집합에 적용해 보았다.

4.1 UCI 데이터 집합

기계 학습 모델을 평가할 때 가장 많이 사용되는 데이터 집합 중에 하나가 UCI 데이터이다. PROBEN1[Prechelt94]는 이러한 UCI 데이터 중에서 15개의 데이터를 전처리한 후에 신경망에 바로 적용할 수 있는 형식으로 바꾸었다. 그 중 본 논문에서는 5가지의 분류 문제를 선택해서 실험을 하였다. 선택된 데이터에 대한 설명은 표 4에 정리하였다.

표 4 : 실험에 사용된 5개의 UCI 데이터 집합

	Cancer	Diabetes	Heart disease	Card	Horse
전체 데이터 수	699	768	920	690	364
입력 차원	9	8	35	51	58
클래스 수	2	2	2	2	3
클래스당 비율	65.5/34.5	65.1/34.9	33/67	56/44	62/24/14

Cancer 데이터 집합은 위스콘신 병원에서 얻은 유방암 환자들에 관한 데이터로서 환자의 몸 상태를 표현한 9개의 입력 값들로부터 실제로 이 사람이 환자인가 아닌가 하는 것이 목표 값으로 주어진다. Diabetes 데이터 집합은 성별이나 임신 횟수와 같은 개인 정보와 혈압과 같은 의료 검사에 의한 정보를 가지고 그 사람이 당뇨병에 걸려 있는지 아닌지를 판별해야 하는 문제이다. Heart disease 데이터 집합은 앞의 두 데이터 집합과 유사하게 환자의 개인 정보와 의학적 검사에 의한 정보를 가지고 그 환자의 심장병을 진단하는 문제이다. Card 데이터 집합은 각 고객에 대해 카드사용을 허용해야 하는지 하지 않아야 하는지를 판별하는 문제이다. 데이터의 보호를 위해 각 속성에 관한 설명은 되어 있지 않다. Horse 데이터 집합은 산통을 가진 말의 수의학적인 검사를 바탕으로 그 말이 살아 날 것인지, 죽을 것인지, 안락사를 해야 할 것인지를 맞추어야 하는 문제이다. 이 데이터는 전체 데이터의 30%정도가 빠져 있어서 정확한 예측을 하기가 쉽지 않다.

표 5 : 각 문제에 대한 ARAN의 파라미터 설정

파라미터	Cancer	Diabetes	Heart	Card	Horse
각 단계에서 선택한 데이터 개수 (λ)	4	5	6	4	5
에러 기준(ε)	0.2	0.2	0.2	0.3	0.4
$\delta(t)$의 초기값 (δ_{max})	1.5	1.5	6	4	2
$\delta(t)$의 최소값 (δ_{min})	0.2	0.1	0.1	0.2	0.2
각 단계에서 선택한 데이터에 대한 반복 학습 횟수	100	10	100	100	100
훈련 데이터에 대한 반복 학습 횟수	100	100	100	100	100

ARAN에서는 각 데이터 집합의 전체 데이터 중에서 2/3는 학습 데이터로, 1/3은 테스트 데이터로 사용하였다. ARAN의 λ는 대충 전체 학습 데이터를 100으로 나눈 값으로 정하였고, 은닉 뉴런을 새로 만드는 경우에 비례상수로 쓰이는 χ는 1로 했으며, 가중치의 학습률은 0.1, 중심과 너비에 대한 학습률은 0.05로 놓고, 나머지 파라미터들은 각 데이터 집합에 맞게 적절하게 조정을 하

였다. 표 5는 각 문제에 대한 사용된 파라미터 값을 보여준다.

표 6에서는 5개의 UCI 데이터 집합에 대한 RAN과 ARAN의 실험 결과를 보여준다. 표에 나온 값들은 각 문제에 대해서 각 알고리즘으로 10번의 실험을 한 결과의 평균값이다. 각 문제별로 윗줄은 RAN의 결과, 아랫줄은 ARAN의 결과이다. RAN의 실험 결과는 모든 데이터를 순차적으로 다 사용한 경우, 최종 신경망의 은닉 뉴런의 개수와 학습 성능, 일반화 성능 등을 보여주고 있으며, ARAN의 실험 결과는 데이터를 능동적으로 선택하면서 학습을 한 후에, 어느 정도 좋은 성능을 보이는 지점에서의 은닉 뉴런의 개수 및 그때까지 학습한 데이터의 크기를 조사하였다.

표 6 : RAN과 ARAN의 실험 결과

문제	전체 학습 데이터 크기	학습에 사용된 데이터 크기	은닉 뉴런의 개수	학습 성능(%)	일반화 성능(%)
(RAN) **Cancer**	466	466	5.0	97.1	97.0
(ARAN)		72.4 (15%)	4.7	83.7	97.0
(RAN) **Diabetes**	512	512	10.4	76.4	74.5
(ARAN)		300 (59%)	8.5	64.1	74.9
(RAN) **Heart**	614	614	5.0	83.1	79.4
(ARAN)		487 (79%)	6.0	79.0	80.4
(RAN) **Card**	460	460	11.1	81.1	85.1
(ARAN)		290 (63%)	10.2	83.9	88.3
(RAN) **Horse**	243	243	5.4	59.7	70.8
(ARAN)		112 (46%)	7.1	63.0	72.1

표 6에서 학습에 사용된 데이터 크기 결과를 보면, ARAN은 데이터의 일부만을 가지고 학습을 하더라도 충분히 좋은 성능을 가질 수 있음을 알 수 있다. 특히 Cancer 데이터의 경우에는 전체 학습 데이터의 약 15% 정도만을 가지고도 만족할 만한 성능을 보인다. 한편, 일반적인 신경망 학습에서는 학습 성능이 일반화 성능보다 더 우수하기 마련인데, ARAN의 결과는 그렇지 않다. 그 이유는 학습이 어려운 데이터를 학습 데이터로 사용하고, 그 데이터에 대해서 성능을 측정하기 때문이며, 만약 ARAN을 끝까지 실행시켜서 전체 데이터를 전부 사용한 경우에는 학습 성능이 일반화 성능보다 높아지게 된다.

그림 5는 Cancer 데이터에 대한 ARAN의 전체 실험 결과로서 학습시 사용된 데이터의 크기와 그 때의 신경망의 일반화 성능 및 은닉 뉴런의 개수의 관계를 나타낸 것이다. 10번의 실험에 대한 평균값을 취했기 때문에 은닉 뉴런의 수가 정수로 늘어나지 않고 있다. 그림을 보면, 처음에는 학습시 사용한 데이터가 증가할수록 일반화 성능 및 은닉 뉴런의 수가 증가하지만, 일정 수준이 지나면 둘다 거의 변화가 없는 모습을 보인다. 이는 Cancer 문제를 푸는데 모든 데이터가 필요하지 않음을 보여주는 한편, 은닉 뉴런의 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. 그림 6의 Card 데이터에 대한 실험 결과에서는 필요한 데이터의 비율이 Cancer 문제보다 높긴 하지만 비슷한 경향을 보인다.

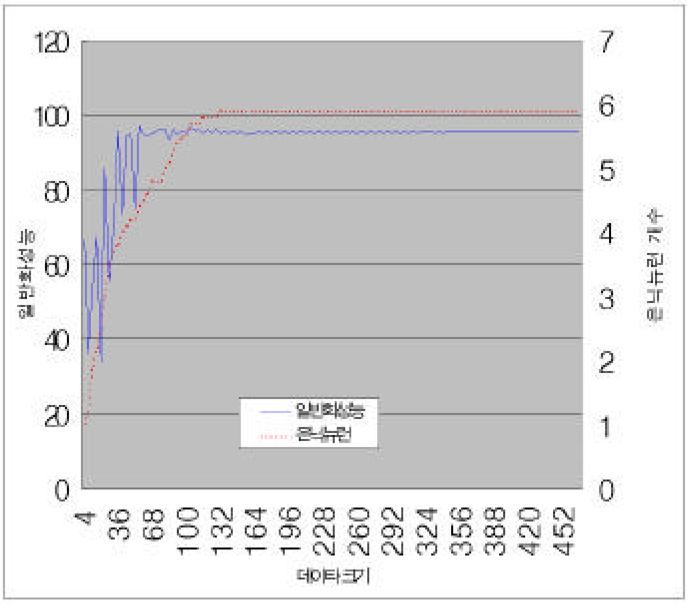

그림 5 : cancer **문제에 대한** ARAN**의 실험결과**

x **축: 학습시 사용된 데이터의 개수**

y **축: 일반화 성능 및 은닉뉴런 수**

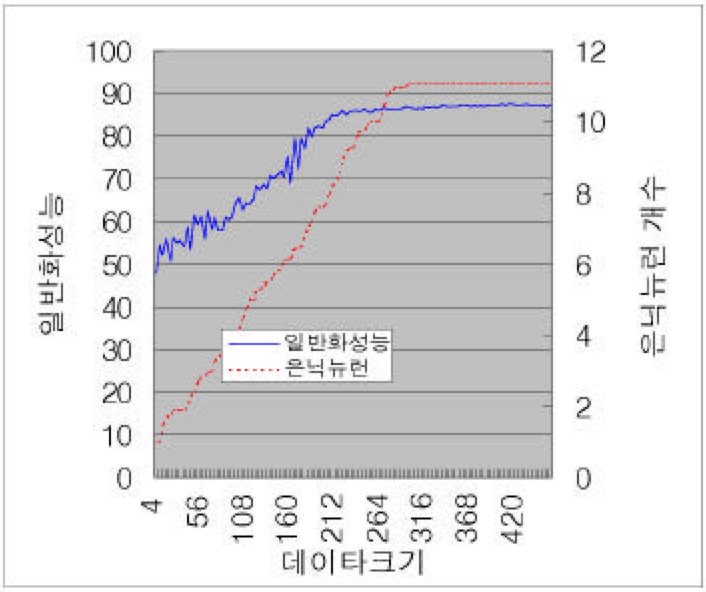

그림 6 : card **문제에 대한** ARAN**의 실험 결과**

4.2 CoIL Challenge 2000 데이터 집합

CoIL Challenge 2000 데이터 집합은 실세계 비즈니스 데이터로서 이동식 주택을 위한 보험의 잠재 고객에 관한 정보로 이루어져 있다. 한 명의 고객에 대해서 전체 86개의 속성이 있고, 그중 43개가 고객의 우편 번호를 통해 알아낼 수 있는 사회 인구 통계 정보이고, 나머지는 이 보험 상품 구매와 관련된 보험 상품의 구매 정보이다. 학습 데이터는 5822명, 테스트 데이터는 4000명으로 이루어져 있다. 실제 대회에서는 참가자들은 5822명에 관한 데이터를 이용하여, 목표 값이 빠진 4000명의 잠재 고객들이 이동 주택 보험에 가입을 할 것인지를 예측해야 했다.

한편 이 데이터는 전체 데이터의 약 6% 정도만 실제 고객으로 그 분포가 매우 불균형적으로 되어있다. 학습 데이터중 348명만 실제 고객이고, 비슷하게 테스트 데이터에서도 238명만이 실제 고객이므로, 고객인지 아닌지를 정확히 판별하는 것은 매우 어려운 문제이다. 이에 실제 대회에서는 4000명의 잠재 고

표 7 : CoIL Challenge 2000 데이터 집합의 입력 속성

사회 인구 통계학 정보	보험 구매 정보
종교 분포 평균 나이 결혼 비율 가족의 크기 교육 수준 직업 분포 사회적 지위 주택 소유 정도 자동차 소유 정도 평균 소득 구매 능력 정도 의료 수준 등	각종 자동차 보험 생명 보험 실업 보험 사고 보험 화재 보험 손해 보험 사회 보장 보험 등

객 중 고객이 될 확률이 높은 800명을 선택해서 이들 중 몇 명이 실제 고객인지를 측정함으로써 우승자를 결정했다. 이는 어떤 회사가 상품 광고 메일을 보내는 경우에, 구매할 확률이 높은 고객에게만 메일을 보내어 비용을 줄이고 효용을 높이는 데 도움이 될 수 있다.

표 7은 CoIL Challenge 2000 데이터의 입력 속성을 정리해 놓았다. 전체 85개의 입력 속성을 크게 사회 인구 통계학 정보와 보험 구매 정보로 나누고, 대표적인 속성을 몇 가지씩 열거해 놓았다. 여기서 주의해야 할 점은 사회 인구 통계학 정보는 구매자 개인의 정보가 아니라, 구매자가 거주하는 곳에서의 항목별 평균 정보이다.

4000명의 잠재 고객 중에서 가장 구매 확률이 높은 800명을 선택하는 경우에 가장 좋은 선택 방법을 적용하면 전체 800명중 실제 고객 238명 전부를 예측할 것이다. 그러나 실제의 대회에서는 121명의 실제 고객을 포함한 참가자가 1등을 하였다. 임의로 선택하는 경우에는 800명의 약 6%인 42명만 찾아낸다. 표 8에 대회 결과가 나와 있다.

ARAN으로 실험을 하기 위해 먼저, 주어진 데이터를 전처리하였다. 주어진 데이터는 각 입력 속성에 따라 차이는 있지만, 0과 몇 개의 자연수로 이루어져 있다. ARAN의 파라미터가 지나치게 커지는 것을 막고, 입력 속성별로 크기가 다르기 때문에 생기는 문제점을 막기 위해 모든 데이터를 0과 1사이의 실수값으로 바꾸었다. 입력 속성을 바꾸기 위해, 단순하게 최소값을 0으로하고 최대값을 1로 놓은 뒤, 나머지 값들은 선형적으로 사상시켰다.

표 8 : CoIL Challenge 2000 결과

방법	결과(전체 800명중 실제 고객수)
최적의 방법	238
임의선택 방법	42
1등	121
2등	115
3등	112

표 9 : CoIL Challenge 2000 데이터 집합 실험 결과

사용된 속성 수	결과(전체 800명중 실제 고객수)	
	능동 학습 방법	수동 학습 방법
85	110	104
4	118	109

실험은 다음과 같이 서로 다른 입력을 가지는 두 가지 방법으로 하였다. 첫째, 전체 85개의 입력 속성을 모두 이용하여 능동 학습을 이용하는 경우와 능동 학습을 이용하지 않는 경우에 대해서 실험을 하였다. 이 실험을 통하여 능동 학습을 이용한 경우에 성능이 좋아짐을 보이고자 했다. 둘째, 주어진 모든 입력 속성을 이용하는 대신에, 가장 중요한 속성으로 알려진 4개의 속성만을 이용해서 앞에서와 같은 방법으로 실험을 하였다. 여기서 사용된 4개의 입력 속성은 자동차 보험, 화재 보험, 트레일러 보험과 교육 수준이다. 이 실험을 통하여 CoIL Challenge 2000 데이터처럼 복잡한 문제에서 데이터 특징 추출이 중요함을 보이고자 했다. 표 9에서는 이러한 실험의 결과를 보여준다. 표 9를 보면 예상한 바와 같이, 85개의 속성을 사용한 경우와 4개의 속성을 사용한 경우 모두 능동 학습 방법을 이용했을 때, 더 좋은 성능을 보인다. 이는 실제 구매자에 대한 정보가 훨씬 적은 불균형한 데이터에서 실제 구매자에 대해 학습의 가중치를 더 줌으로써 좀 더 좋은 성능을 가지는 것으로 보인다.

한편, 85개의 속성을 전부 사용한 경우보다, 오히려 더 적은 4개의 속성을 사용한 경우가 더 좋은 성능을 보인다. 이는 잠재 고객 중에서 실제 고객을 예측하는 문제에서 85개의 속성 전부가 문제 해결에 관련이 있는 것이 아님을 보여준다. 즉, 오히려 더 많은 정보를 가지고 학습하는 경우에는 상관없는 속성들이 노이즈의 역할을 함으로써 좋지 않은 결과를 나타낼 수 있는 것이다. 따라서 CoIL Challenge 2000 데이터와 같이 많은 속성을 가지는 데이터를 학습하는 경우에는 학습 모델을 최적화 시키기 이전에 데이터의 중요한 특징을 파악하는 것이 중요할 수 있다. 대회의 입상자들과 여기서의 실험 결과의 성능

차이는 데이터의 중요한 특징을 파악해 내는 능력의 차이라고 볼 수 있다.

그러나 앞의 실험 결과는 데이터의 특징 파악 없이도 CoIL Challenge 2000 입선자들의 결과와 많은 차이를 보이지 않는데, 이는 표 9의 실험이 이미 테스트 데이터에 대한 답을 알고 이루어졌기 때문이다. 이미 답을 알고 실험을 진행했기 때문에, 좋은 성능을 내는 쪽으로 학습을 진행시킬 수 있었다. ARAN 모델을 가지고 실제 대회에 참가했더라면 표 9의 결과보다 성능이 더 좋지 않았을 것이다.

4.3 CAMDA '00 Conference 데이터 집합

현대 생물학 기술의 발전은 특정 사람의 수많은 유전자 발현 정보를 한번의 실험을 통하여 얻을 수 있게 되었다. 유전자 발현 정보를 이용해서 어떤 병의 진단이나 치료에 이용할 수 있음에도 불구하고, 정보를 분석하는 표준적인 도구가 없다. 이에, 유전자 정보를 분석하는 방법에 대한 도구를 비교하기 위한 CAMDA'00(Critical Assessment of Techniques for Microarray Data Mining) 학회가 2000년 12월에 열린다. 이 학회에서 사용한 데이터 집합 중 하나인 백혈병의 종류를 구분하는 문제에 ARAN을 적용하였다.

이 데이터 집합은 38개의 샘플로 이루어진 학습용 데이터 집합과, 34개의 샘플로 이루어진 테스트용 데이터 집합으로 나누어진다. 각 샘플은 백혈병의 두 종류인 AML과 ALL로 나뉘는데, 학습 데이터에서는 11개의 샘플이 AML이고, 27개가 ALL이며, 테스트 데이터에서는 14개가 AML, 20개가 ALL이다. 입력 속성은 환자의 유전자 발현 수준이고 전체 7129개의 유전자에 대한 측정치이다. 따라서, 각 백혈병 환자의 7129개의 유전자 발현 정보를 이용해서 이 환자가 AML인지 ALL인지 분류하는 것이 이 데이터 집합의 문제가 되겠다.

ARAN으로 실험을 하기 위해 먼저, 주어진 데이터를 전처리하였다. 주어진 데이터는 각 입력 속성에 따라 차이는 있지만, 음수 수만에서 양수 수만까지의 정수값을 가진다. ARAN의 파라미터가 지나치게 커지는 것을 막고, 입력 속성별로 크기가 다르기 때문에 생기는 문제점을 막기 위해 모든 데이터를 0과 1 사이의 실수 값으로 바꾸었다. 바꾸는 방법으로는 앞에서와 마찬가지로 단순하

게 최소값을 0으로 하고 최대값을 1로 놓은 뒤, 나머지 값들은 선형적으로 사상시켰다.

실험은 다음과 같이 서로 다른 입력을 가지는 세 가지 방법으로 하였다. 첫째, 주어진 7129개의 입력속성을 모두 이용한 방법이다. 둘째, 통계학적으로 백혈병 환자 분류에 도움이 되는 전체 입력 속성중 50개의 유전자만을 이용한 방법이다. 이는 각 유전자별로 AML환자의 평균(μ_1)과 분산(σ_1)을 구하고, ALL환자의 평균(μ_2)과 분산(σ_2)을 구한 뒤 다음 값을 계산한다.

$$\frac{\mu_1 - \mu_2}{\sigma_1 + \sigma_2} \tag{17}$$

이 값을 P-metric이라 하는데 이 값이 양의 방향으로 큰 25개의 유전자와 음의 방향으로 큰 25개의 유전자를 입력 속성으로 선택했다[Golub99]. 셋째, 정보이론에 기반해서 출력과의 정보량이 가장 많은 유전자 10개를 선택해서 실험을 하였다. 찾아진 유전자는 표 10에 나와있으며 굵은 글씨의 유전자는 통계학적으로 선택한 앞의 50개의 유전자와 겹치는 유전자이다. 정보량이 많은 유전자부터 차례대로 나열하였다.

표 10 : 정보이론에 기반해서 선택한 10개의 유전자

Zyxin
ADM Adrenomedullin, Azurocidin gene
DF D component of complement (adipsin)
PTX3 Pentaxin-related gene
rapidly induced by IL-1 beta
CYSTATIN A
NADPH-flavin reductase
Leukotriene C4 synthase (LTC4S) gene
CD36 CD36 antigen (collagen type I receptor, thrombospondin receptor)
Phosphotyrosine independent ligand p62 for the Lck SH2 domain mRNA

표 11 : 백혈병 종류 예측 실험 결과

사용된 유전자의 수	입력 속성 선택 방법	잘못 분류된 샘플 수		은닉 뉴런의 수
		학습 데이터 집합	테스트 데이터 집합	
7129	-	0	4	9
50	통계학 기반	0	1.3	2.4
10	정보 이론 기반	0	2.5	2

세 가지 데이터 집합에 대해서 각각 ARAN을 적용하였다. 표 11에서 그 결과가 나와있다. 모든 유전자 정보를 사용한 첫 번째 실험 결과는 10번의 실험 결과중 가장 좋은 결과를 보여주었고, 나머지 두 개의 실험 결과는 10번 실험의 평균을 기록하였다. 표를 보면 주어진 유전자를 전부 사용할 때보다 적절한 방법에 의해서 골라낸 유전자를 사용한 실험 결과가 더 좋다. 이것으로 모든 유전자가 ARAN이 문제를 푸는데 도움을 주는 것이 아님을 알 수 있다.

표 12는 두 번째 실험과 세 번째 실험에서 잘못 분류된 샘플 정보를 보여준다. 잘못 분류된 샘플들은 모두 AML 샘플인데, 그 이유는 학습 데이터 집합에서 ALL 샘플이 AML 샘플보다 2배 이상 많기 때문에 ARAN이 ALL 샘플 쪽으로 바이어스되기 때문인 것 같다. 한편, 테스트 샘플중 28번째 샘플은 ARAN을 사용한 모든 실험에서 잘못 분류되는데, 아마도 실제로 ALL 샘플인데, AML 샘플로 잘못 측정한 것 같다. 표 12의 잘못 분류된 샘플에서 괄호 안에 있는 숫자는 전체 데이터 중에서의 위치이고 괄호밖에 있는 수는 테스트 데이터 중에서의 위치이다.

표 12 : 잘못 분류된 샘플 분석

두 번째 실험	잘못 분류된 샘플	세 번째 실험	잘못 분류된 샘플
전체 10회중 7회	28 (66)	전체 10회중 5회	5, 28 (42, 66)
전체 10회중 3회	16, 28 (54, 66)	전체 10회중 5회	5, 23, 28 (42, 61, 66)

그림 7과 8은 능동 학습의 효과를 보여준다. 이 그림은 각각 두 번째 실험과 세 번째 실험에서 데이터 크기와 일반화 성능과의 관계를 보여준다. 두 그래프에서 보듯이 처음에 학습 데이터를 선택하면 할수록 성능이 좋지만 일정 수준

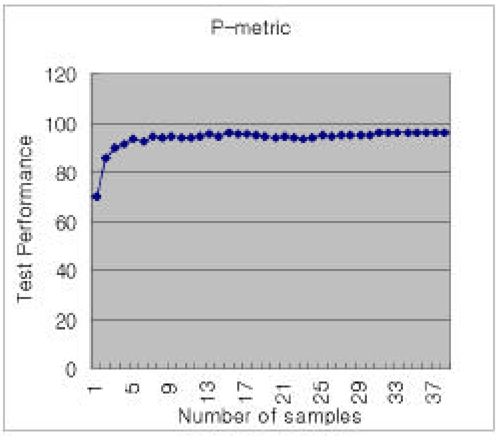

그림 7 : 두 번째 데이터 집합에 대한 학습 데이터 크기와 성능과의 관계

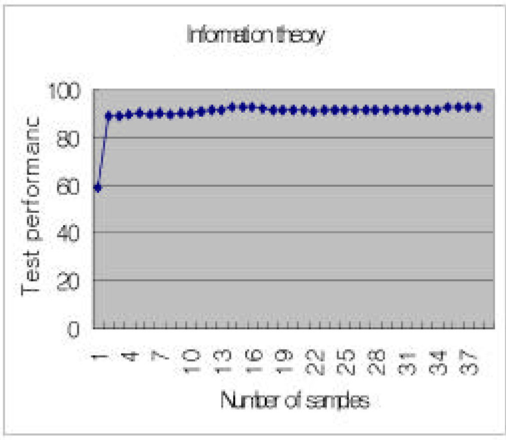

그림 8 : 세 번째 데이터 집합에 대한 학습 데이터 크기와 성능과의 관계

이 지나면 성능은 거의 그대로 유지되고 있음을 알 수가 있다. ARAN학습시 처음에 선택되는 데이터가 중요한 데이터라고 예측할 수 있는데, 표 13에서는 전체 10번의 실험 중에서 선택순위가 상위 10위안에 모두 들어있는 샘플들을 나열하였다. 학습 샘플 중 28번째 샘플과 35번째 샘플은 두 가지의 실험에서 모두 처음 부분에서 뽑히는 샘플로서 다른 샘플들을 대표할 수 있는 중요한 샘플이라고 생각된다.

표 13 : 전체 10번의 실험에서 선택 순위가 모두 상위 10위안에 드는 샘플

두 번째 실험	12, 20, **28**, 31, **35**
세 번째 실험	8, 17, 25, **28**, 32, 34, **35**

5. 결론

본 논문에서는 건설적 학습 알고리즘인 RAN에 능동 데이터 선택 기법을 적용하여 ARAN이라는 새로운 알고리즘을 제안하였다. 능동적인 데이터 선택은 현재의 신경망에서 에러를 가장 크게 내는 데이터를 취하는 방법을 이용하였다. 앞장의 실험 결과에서 보듯이 모든 데이터를 다 사용하지 않고도 데이터를 전부 사용할 때와 비슷하거나 오히려 더 좋은 성능을 보임을 알 수 있다. 이는 학습시간에서의 이점이 있을 뿐만 아니라, 학습에 사용된 데이터들을 데이터 마이닝 관점에서 살펴볼 경우 의미가 있을 수 있다.

한편, ARAN에서는 데이터를 능동적으로 선택해서 학습하는 경우에 합리적인 종료 알고리즘이 없다. 따라서 모든 데이터를 끝까지 사용한 다음에 가장 좋은 성능을 나타내는 모델을 결과로 사용했기 때문에 실제의 문제를 풀려고 하는 경우에 문제가 생긴다. 학습 알고리즘의 안정성과 효용성을 위해서 보다 체계적인 종료 방법이 필요하다. 또한 ARAN은 이론적으로 새로운 데이터가 계속해서 들어오는 온라인 학습에 적용할 수 있다. 그러나 알고리즘 상으로 새로운 데이터가 들어올수록 모델이 복잡해지는 것을 피할 수 없게 되는 단점이 있다. 따라서 학습이 진행되어감에 따라 이전에 만들어진 필요 없는 은닉 뉴런을 제거해야 할 필요성이 있다. 이는 모델이 너무 복잡해지는 것을 막는 역할을 할 뿐만 아니라, 성능에 도움을 주지 못하는 은닉 뉴런을 제거하고, 성능에 도움이 되는 은닉 뉴런을 추가할 가능성을 부여함으로써 더 좋은 성능을 낼 수 있게 될 것이다.

본 논문에서는 증가적 능동 선택에 대해서만 다루었는데, 앞으로 3장에서 제안한 다양한 능동 데이터 선택 방법에 대한 비교를 통하여, 특정 문제를 풀 때, 어떤 방법이 유리한지 살펴봐야 할 것이다. 그리고 학습에 사용한 데이터들을 분석함으로써 그것들이 전체 데이터 집합을 어떻게 대표하는 것인지를 살펴볼 필요도 있다. 은닉 뉴런의 중심을 살펴보는 것도 같은 맥락에서 의미가 있을 것이다.

참고 문헌

[Adejumo99] A Adejumo, AP Engelbrecht, A Comparative Study of Neural Network Active Learning Algorithms, *Proceedings of the International Conference on Artificial Intelligence*, pp. 32-35, 1999.

[Ash89] T. Ash, Dynamic Node Creation in Backpropagation Networks, *Connectionist Science*, vol. 1, no. 4, pp. 365-375, 1989.

[Baum89] E.B. Baum, A Proposal for More Powerful Learning Algorithms, *Neural Computation*, vol. 1, no. 2, pp. 201-207, 1989.

[Billings95] Steve A. Billings and Guang L. Zheng, Radial Basis Function Network Configuration Using Genetic Algorithms, *Neural Networks*, vol. 8, no. 6, pp. 877-890, 1995.

[Broomhead88] D.S. Broomhead and D. Lowe, Multivariable Functional Interpolation and Adaptive Networks, *Complex Systems*, vol. 2, pp. 321-335.

[Burdsall97] Ben Burdsall and Christophe Giraud-Carrier, GA-RBF: A Self-Optimising RBF Network, *In Proceedings of the Third International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA '97)*, pp. 348-351. 1997.

[Cohn90] D. Cohn, L. Atlas and R. Ladner, Training Connectionist Networks with Queries and Selective Sampling, *Advances in Neural Information Processing Systems 2*, 1990.

[Cohn94] D. Cohn, L. Atlas and R. Ladner, Improving Generalization with Active Learning, *Machine Learning*, vol. 15, no. 2, pp. 201-221, 1994.

[Cohn96] D. Cohn, Z. Ghahramani and M.I. Jordan, Active Learning with Statistical Models, *Journal of Artificial Intelligence Research 4,* pp. 129-145, 1996.

[Deco93] Gustavo Deco and Jurgen Ebmeyer, Coarse Coding Resource-Allocating Network, *Neural Computation*, vol. 5, no. 1, pp. 105-114, 1993.

[Engelbrecht98] AP Engelbrecht, I Cloete, Selective Learning using Sensitivity Analysis, *International Joint Conference of Neural Networks (IJCNN '98)*, pp.1150-1155, 1998.

[Fahlman90] Scott E. Fahlman and Christian Lebiere, The Cascade-Correlation Learning Architecture, *Advances in Neural Information Processing Systems (NIPS) 2*, pp.524-532, 1990.

[Farlow84] S.J. Farlow, Self-Organizing Methods in Modeling: GMDH Type Algorithms, *Statistics: Textbooks and Monographs*, vol. 54, 1984.

[Friedman81] J.H. Friedman and W. Stuetzle, Projection Pursuit Regression, Journal of American Statistics Association, vol. 76, no. 376, pp. 817-823, 1981.

[Fukumizu96] Kenji Fukumizu, Active Learning in Multilayer Perceptrons, *Advances in Neural Information Processing Systmes (NIPS) 8*, 1996.

[Fukumizu00] Kenji Fukumizu, Statistical Active Learning in Multilayer Perceptrons, *IEEE Transactions on Neural Networks*, vol. 11, no. 1, 2000.

[Golub99] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander, *Science*, Oct 15, pp. 531-537, 1999.

[Haykin99] Simon Haykin, *Neural Networks: A Comprehensive Foundation 2nd Edition*, Prentice-Hall, 1998.

[Kwok97] Tin-Yau Kwok and Dit-Yan Yeung, Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems, *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 630-644, 1997.

[Leisch98] F. Leisch, L.C. Jain and K. Hornik, Cross-Validation with Active Pattern Selection for Neural-Network Classifiers, *IEEE Transactions on Neural Networks*, vol. 9, no. 1, pp. 35-41, 1998.

[Liere97] Ray Liere and Prasad Tadepalli, Active Learning with Committees in Text Categorization, *AAAI-97*, pp. 583-590, 1997.

[McCallum98] Andrew Kachites McCallum and Kamal Nigam, Employing EM and Pool-Based Active Learning for Text Classification, *In Proceedings of the 15th Conference on Machine Learning*, pp. 350-358, 1998.

[Park00] Park, S.-W. and Zhang, B.-T, Learning Constructive RBF Networks by Active Data Selection, *7th International Conference on Neural Information Processing (ICONIP-2000)*, vol. 2, pp. 1411-1415, 2000.

[Platt91] John Platt, A Resource-Allocating Network for Function Interpolation, *Neural Computation*, vol. 3, no. 2, pp. 213-225, 1991.

[Plutowski93] M. Plutowski and H. White, Selecting Concise Training Sets from Clean Data, IEEE Transactions on Neural Networks, vol. 4, pp. 305-318, 1993.

[Prechelt94] Lutz Prechelt, PROBEN1-A Set of Neural Network Benchmark Problems and Benchmarking Rules, *Fakultät für Informatik, Univ. Karlsruhe, Germany, Tech. Rep. 21/94,* 1994.

[Reed93] R. Reed, Pruning algorithm - A survey, *IEEE Transaction on Neural Networks*, vol. 4, pp. 740-747, 1993.

[Röbel94] A Röbel, Dynamic Pattern Selection: Effectively Training Backpropagation Neural Networks, International Conference on Artificial Neural Networks, vol. 1, pp. 643-646, 1994.

[Tao93] K. Mike Tao, A Closer Look at the Radial Basis Function (RBF) Networks, *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, vol 1, pp. 401-405, 1993.

[Ungar94] Lyle H. Ungar and Ricahrd D. De Veaux, EMRBF: A Statistical Basis for Using Radial Basis Functions for Process Control, *American Control Conference Proceedings*, 1994.

[Yingwei98] L. Yingwei, N. Sundarara and P. Saratchandran, 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.

[Zhang91a] Byoug-Tak Zhang and Gerd Veenker, Focused Incremental Learning for Improved Generalization with Training Sets, *Proceedings of the International Conference on Artificial Neural Networks (ICANN '91)*, pp. 227-232, 1991.

[Zhang91b] Byoung-Tak Zhang and Gerd Veenker, Neural Netoworks that Teach Themselves through Genetic Discovery of Novel Examples, *Proceedings of the 1991 IEEE International Joint Conference on Neural Networks (IJCNN '91)*, vol. 1, pp. 690-695, 1991.

[Zhang93] Byoung-Tak Zhang, Self-development Learning: Constructing Optimal Size Neural Networks via Incremental Data Selection, Zhang, B.-T., Arbeitspapiere der GMD, No 768, German National Research Center for Computer Science (GMD), St. Augustin/Bonn, July 1993.

[Zhang94a] Byoung-Tak Zhang, Accelerated Learning by Active Example Selection, *International Journal of Neural Systems*, vol 5, pp. 67-75, 1994.

[Zhang94b] Byoung-Tak Zhang, An Incremental Learning Algorithm That Optimizes Network Size and Sample Size in One Trial, Byoung-Tak Zhang, *Proceedings of IEEE International Conference on Neural Networks (ICNN '94)*, vol. 1, pp. 215-220, 1994.

[Zhang96] Byoung-Tak Zhang, GENIE: A Learning Intelligent System Engine Based on Neural Adaptation and Genetic Search, ***한국퍼지 및 지능 시스템 학회 추계 학술발표 논문집***, vol. 5, no. 2, pp. 27-34, 1996.

Abstract

In this paper we propose a learning algorithm for RBF(Radial Basis Function) networks that selects data actively and increases the number of hidden units to fit the data. By selecting data actively, a learner does not simply use given all examples for training. Instead a learner controls the data properly to get better performance by active learning. Since constructive learning algorithms can find proper architecture of the neural networks by increasing the number of hidden units in learning process, so we apply a new learning algorithm to combine both active learning and constructive learning naturally to RBF networks.

In this paper we propose an ARAN (Active RAN) that iteratively selects data where the current RBF network performs the worst, then train the network on these data. ARAN is a learning algorithm which active data selection scheme is added to RAN (Resource-allocating network) by Platt. Experiments have been performed on five UCI data set and their performance was compared with RAN. In these experiments, ARAN has enough performance by learning without using all data. And Experiments have been performed on CoIL Challenge 2000 data set. The results of these experiments show that active learning method has better generalization performance than passive learning method in imbalanced data set like CoIL Challenge 2000 data. ARAN can be also used in solving real business problems. We also used ARAN for analysis of gene expression data set.

Keywords : ARAN, Active learning, Constructive Learning, RBF Networks, Critical Data, Imbalanced Data Set

감사의 글

대학원에 자리를 잡고 공부를 시작한지 어느새 2년이 지났습니다. 그동안 여러 면에서 부족한 저를 학문적으로나 인간적으로 성숙하게 만들어 주신 모든 분들에게 감사의 말씀을 드립니다. 그분들 덕분에 부끄럽지만 하나의 학위 논문을 완성을 할 수 있었던 것 같습니다.

먼저, 2년동안 아무 것도 모르는 저를 격려해주시고 지도해주신 장병탁 교수님께 감사드립니다. 여러 가지 일들로 바쁘신 와중에서도 교수님의 학문에 대한 열정은 게으른 저에게 좀 더 연구를 열심히 할 수 있도록 의지를 주셨습니다. 앞으로 교수님께서 하시는 연구가 더 큰 빛을 낼 수 있기를 바라며, 또 그렇게 되리라고 믿습니다.

한편, 2년동안의 연구실 생활을 나름대로 재미있고, 많은 것을 얻을 수 있도록 도와준 모든 선후배님들, 동기들에게 특히 감사를 드립니다. 새로운 연구실 생활에 빨리 적응할 수 있도록 도와준 여러 선배님들과 함께 공부하면서 부족한 저의 지적 능력을 채워준 동기들에게 진심으로 감사합니다. 단조롭고 지루할 수도 있었던 연구실 생활에서 많은 선후배님들과 동기들은 저에게 기쁨과 용기를 주셨습니다. 앞으로 그분들이 하시는 모든 일들이 좋은 결실을 맺기를 간절히 바랍니다. 제가 연구실을 떠나더라도 잊지 않고 기쁜 일이나 슬픈 일이 생기면 꼭 연락해주길 바랍니다.

또한, 그 동안 스트레스가 있을 때마다 같이 술마시면서 고민을 나누어주었던 여러 친구들한테도 깊은 감사를 느낍니다. 그들은 이 세상을 같이 살아가는 동반자로서 제 삶의 안목을 넓히는데 많은 도움이 되었습니다.

끝으로, 가장 가까이 있으면서도 많은 시간을 같이 지내지 못한 가족들에게 미안함과 함께 감사의 말씀을 드립니다. 2000년 힘든 한 해를 보내신 부모님께 장남으로서 큰 위로를 드리지 못해 죄송스럽습니다. 그 동안 이만큼 키워주시고 격려를 해주셔서 감사합니다. 2001년부터는 우리 가족에게 기분 좋은 일만 생기기를 바라고, 새로운 시작을 하려는 동생 상준이의 앞날에도 축복이 있길 바랍니다.

2001년 1월 연구실에서