퍼지 및 지능시스템학회 논문지 2006, Vol. 16, No. 6, pp. 736-741

점증적 학습 퍼지 신경망을 이용한 적응 분류 모델

An Adaptive Classification Model Using Incremental Training Fuzzy Neural Networks

이현숙

Rhee, Hyunsook

동양공업전문대학 전산정보학부 부교수

요 약

분류 시스템은 데이터 전처리 모듈, 학습모듈, 의사결정모듈로 구성되어 있으며 지능형시스템의 중요한 구성요소로 활용되어왔다. 특히 학습모듈은 사전정보를 제공하므로 분류를 위한 핵심 역할을 수행하여 왔다. 기존의 학습을 위한 기법은 주로 승자독점방식으로 데이터를 처리하므로 경계가 불명확한 대부분의 실세계 응용에 적합하지 못하다. 또한 학습 알고리즘에 필요한 데이터를 한꺼번에 준비해야 하지만 이는 일반적으로 가능하지 않은 경우가 많다. 이를 위하여 본 논문에서는 점증적 학습 퍼지신경망, FNN-I,를 이용한 적응 분류모델을 설계한다. 이 모델에서는 유용하게 정보를 표현하기 위하여 퍼지이론을 도입하고 계속적으로 모여지는 데이터를 가지고 점증적으로 학습할 수 있는 알고리즘을 제시한다. 제안된 모델을 컴퓨터 바이러스 분류를 위한 실제 데이터에 적용하여 점증적으로 학습할 수 있고 효과적으로 새로운 바이러스 데이터를 분류할 수 있음을 보인다.

Abstract

The design of a classification system generally involves data acquisition module, learning module and decision module, considering their functions and it is often an important component of intelligent systems.. The learning module provides a priori information and it has been playing a key role for the classification. The conventional learning techniques for classification are based on a winner take all fashion which does not reflect the description of real data where boundarues might be fuzzy. Moreover they need all data for the learning of its problem domain. Generally, in many practical applications, it is not possible to prepare them at a time. In this paper, we design an adaptive classification model using incremental training fuzzy neural networks, FNN-I. To have a more useful information, it introduces the representation and membership degree by fuzzy theory. And it provides an incremental learning algorithm for continuously gathered data. We present the experimental results on computer virus data. They show that the proposed system can learn incrementally and classify new viruses effectively.

Key words : Incremental Training(점증적 학습), Fuzzy Neural Networks(퍼지신경망), Classification(분류), Computer Virus(컴퓨터바이러스)

1. 서 론

컴퓨터가 처리하는 정보의 양이 방대해짐에 따라 또한 시스템에 반영되어야 하는 새로운 정보가 계속해서 발생됨에 따라 학습을 통하여 정리된 정보를 추출하고 비슷한 상황에 적절히 대처하는 적응기능은 지능형 시스템이 가져야하는 기본 기능이 되었다. 이러한 학습기능을 부여하기 위한 연구는 기초적인 기계학습의 이론을 토대로 다양한 형태의 신경망구조와 학습 알고리즘과 유전자 알고리즘과 같은 결과를 도출하여 영상 및 음성인식 시스템, 웹이에전트시스템, 자동화시스템 등의 지능형시스템 구축에 활용되어왔다[1,2]. 이러한 시스템의 공통적인 요소로서 분류(classification)는 학습을 통하여 얻을 수 있는 일반적인 결과물로서 데이터 전처리과정에서 의사결정단계까지 활용되어왔다.

신경망(Neural Networks) 분야의 연구는 학습을 통하여 분류해 내는 대표적인 방법으로 단층신경망, 역전파 신경망, 그리고 자기조직화 지도(Self Organizing Map) 등의 기법이 제안되어 널리 활용되어왔다. 특히 자기조직화지도 신경망의 경우 데이터 수가 많을때 스스로 비슷한 속성의 데이터를 비슷한 위치로 모이도록 학습하므로 클러스터분석을 통한 분류에 활용되어왔다. 그러나 이와 같은 방법은 승자독점(winner take all) 형태의 전략을 취하여 특정 뉴런의 가중치를 갱신하면서 학습이 진행되기 때문에 경계가 불명확한 데이터를 많이 가지고 있는 실세계 응용의 경우 임계치 등 파라메터에 의존적인 결과를 가져오게 된다. 이에 퍼지이론을 적용하여 각 데이터의 클래스 소속값을 함께 학습하므로 보다 정확하게 데이터를 표현하고 정확한 분류결과를 얻을 수 있도록 하는 퍼지 클러스터링 기법이 도입되었다[3]. FCM(Fuzzy c-Means) 알고리즘은 대표적인 클러스터 분석 방법으로 제안되어 여러 응용분야에 적용되었으며 아직도 이 방법의 수렴성과 최적화, 일반화에 대한 고찰은 계속되고 있다[4]. 그러나 FCM은 쉽게 국소적 최소값에 수렴하고 신경망의 유연성을 가지지 못하는 단점을 가지고 있다. 이에

접수일자 : 2006년 11월 17일
완료일자 : 2006년 11월 30일

신경망의 유연성을 가지면서 FCM의 연구결과를 활용할 수 있는 목적함수기반 퍼지 신경망, FNN-B를 구성하게 되었다[5]. 지금까지 제안된 대부분의 방법들은 FCM이나 FNN-B와 같이 학습하고자하는 데이터가 모두 준비된 후 학습하는 일괄학습(batch training) 방법을 취하고 있다. 이와 같은 방법은 구축하려는 시스템의 목표가 명확하고 데이터가 모두 수집된 경우 효과적으로 활용될 수 있다. 그러나 대부분의 실세계 응용의 경우 한꺼번에 학습을 위한 데이터를 모두 수집하기 어렵고 때로는 새롭게 수집된 데이터에 따라 시스템의 개발목표가 변하기도 한다. 컴퓨터 바이러스의 경우 세대별 분류, 감염위치에 따른 분류, 다양한 형태의 기능에 따른 분류 등 분류 방법이 다양하고 그 정의에 대해서도 논의 중에 있다. 그러므로 어떠한 분류 방법으로도 명확하게 분류할 수 없으며 바이러스 전문가의 지식과 경험을 토대로 설계되고 있다. 즉 바이러스 데이터가 계속 생성되고 새로운 학습 데이터로 활용되어야하는 바이러스 분류모델의 경우 한꺼번에 충분한 학습데이터를 가지고 일반적으로 접근하기 어렵고 점증적으로 학습하는 학습 알고리즘이 필요하다[6,7].

본 논문에서는 경계가 불명확한 대부분의 실세계 데이터에 대하여 퍼지 집합표현과 소속값 처리 방법을 도입하고 계속 수집되고 있는 데이터에 대하여 점증적으로 학습하는 퍼지신경망 FNN-I를 구축하고 이를 이용한 적응 분류 모델을 설계하고자한다. 또한 이를 컴퓨터 바이러스 분류를 위한 실제 데이터에 적용하여 분류 모델이 점증적으로 학습할 수 있음을 확인하고 FNN-B를 적용한 경우와 비교하여 그 타당성을 검토한다. 2장에서는 본 논문의 기초가 되는 관련 연구로서 퍼지클러스터링을 위한 목적기반 퍼지 신경망 FNN-B을 고찰해 보고 그 동안 진행해 온 컴퓨터 바이러스 분류기법에 대하여 기술한다. 3장에서는 논문의 중심 부분으로 제안된 적응 분류 모델과 이에 사용된 점증적 학습 퍼지신경망의 구조와 학습 알고리즘을 기술한다. 4장은 설계된 모델의 타당성을 확인하기 위하여 준비된 컴퓨터 바이러스 데이터를 가지고 실험하고 앞서 소개한 FNN-B와 비교한다. 5장은 결론으로서 제안한 방법을 요약하고 앞으로의 발전 방향을 기술한다.

2. 관련연구

2장에서는 본 논문의 기초가 되는 관련 연구로서 퍼지클러스터링을 위한 목적기반 퍼지 신경망 FNN-B를 고찰해 보고 그동안 진행해 온 컴퓨터 바이러스 분류기법에 대하여 기술한다.

2.2 목적함수 기반 퍼지신경망

FCM 알고리즘 목적함수 J_m을 비교사학습신경망에 결합시켜 (그림 1)와 같은 퍼지신경망, FNN-B 구성하였다[5]. 이렇게 구성된 신경망에서 다음의 알고리즘을 통하여 입력층에 제공된 데이터 X = $\{x_1,...,x_n\}$는 대표정보인 클러스터의 중심점 $(v_1, v_2, \ldots, v_c)$을 학습해 간다. 이러한 학습을 통해 형성된 클러스터 층은 데이터 x_j 의 클러스터 i에 속하는 소속 값, u_{ij} 을 포함하는 정보 사이의 관계를 표현하는 값인 $(\alpha_1,\alpha_2,...,\alpha_c)$를 계산하여 그 결과를 다음 학습에 활용한다. 이러한 학습 알고리즘은 클러스터링의 결과가 만들어내는 오류 값을 요약하는 퍼지 함수를 설정한 후 그 값이 최소가 되도록 학습의 방향을 유도하는 메카니즘에 의해 진행된다. 또한 제안된 방법은 입력과 출력 사이의 관계를 기술하기 어려운 경우도 쉽게 처리하는 비교사 학습신경망의 장점도 함께 가지고 있다.

단계 1 : c, m, ϵ의 값을 설정하고 입력데이터 셋을 준비한다.
c 는 클러스터의 수, m 은 FCM 알고리즘의 weighting exponent 이다.

단계 2 : 초기 가중치 벡터 V= $(v_1, v_2, \ldots, v_c)$와 퍼지 C 분할 U를 0과 1 사이의 난수로 초기화한다. $t = 0$.

단계 3 : 다음 식을 이용하여 α_{ij} 를 계산하고

$$\eta_i = \frac{1}{\sum_{j=1}^{n} \alpha_{ij}}$$ 이라고 하자.

$$\alpha_{ij} = \frac{2m}{m-1}\left\{ \sum_{l=1}^{c} (u_{lj})^{m+1} \left(\frac{\| x_j - v_i \|^2}{\| x_j - v_l \|^2} \right)^{\frac{m}{m-1}} \right\}$$

단계 4 : 다음 식을 이용하여 가중치 벡터를 수정한다.

$$\triangle v_i = \eta_i \sum_{j=1}^{n} \alpha_{ij} (x_j - v_{ij}) \quad (1)$$

단계 5 : 다음 식을 이용하여 퍼지 C 분할 U를 계산한다.

$$u_{ij} = \frac{1}{\sum_{s=1}^{c} \left\{ \frac{\| x_j - v_i \|}{\| x_j - v_s \|} \right\}^{2/(m-1)}} \quad (2)$$

단계 6 : $v_{i,t}$ 는 현재 가중치 벡터이고 $v_{i,t+1}$ 는 단계4 에 의하여 수정된 가중치 벡터일때

$diff = \sum_{i=1}^{c} \| v_{i,t+1} - v_{i,t} \|^2 < \epsilon$이면 알고리즘을 끝내고 그렇지 않으면 단계 3으로 가자

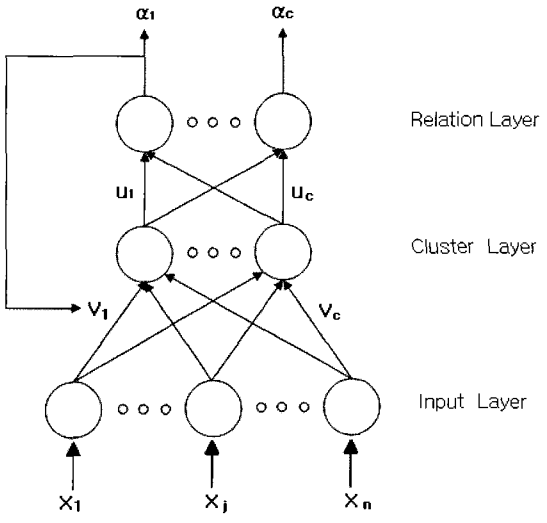

그림 1. 목적함수기반 퍼지 신경망(FNN-B)

2.2 컴퓨터 바이러스 분류기법

1986년 컴퓨터 바이러스가 처음 출현한 이래 매년 수많은 새로운 바이러스가 중요한 정보를 저장하고 있는 컴퓨터를 위협하고 있으므로 이는 점점 현실적인 문제로 대두되고 있다. 이에 미국 주요 대학 컴퓨터공학부 위원회에서는 컴퓨터 바이러스 감염을 정부가 가장 관심을 두고 해결해야 할 문제점으로 지적하기도 했다. 컴퓨터 바이러스를 탐지하는 그동안의 연구는 알려진 바이러스 파일 특성 분석과 단순 매칭에 기반을 두고 있다. 그러므로 알려지지 않은 바이러스를 탐지하지 못하고 그 바이러스에 의하여 시스템이 공격을 받고 난 후 바이러스 유형과 파일상태를 분석한 후에야 탐지 알고리즘에 반영될 수 있다. 그러는 사이 시스템은 손상되고 또 다른 형태의 바이러스가 만들어질 것이다. 이를 위하여 바이러스 전문가의 휴리스틱한 규칙이 적용되기도 했으나 간단하고 유연성이 없어 쉽게 노출되며 실세계에 적용하기 어려운 단점을 가지고 있다. 기존의 바이러스 유형과 특성에 대한 학습을 바탕으로 새로운 바이러스 유형에 적응하기 위한 기계학습, 신경망, 퍼지이론, 데이터마이닝 등의 소프트컴퓨팅 기법들을 적용하기 위한 시도들이 이루어졌다[8, 9]. 데이터 마이닝 분야의 데이터 처리 기법이나 텍스트 마이닝 기법들이 파일의 특징패턴을 찾는데 이용되기도 하였다[10]. 또한 기계학습방법으로 연구되어 WEKA에서 구현된 학습알고리즘 - Instance Based Learner, TFIDF, Nave Bayes, Super Vector machines, Decision Trees C4.5 - 등을 분류에 활용하기도 하였다[11].

Abou-Assaleh 등[12]은 Commom N-Gram(CNG) 방법을 제안하여 알려지지 않은 새로운 파일을 진단하는데 활용하였다. 악성코드와 정상코드로 구성된 데이터베이스로부터 자주 출현하는 n-gram을 시그내처 로서 추출하여 저장한다. 이렇게 추출된 n-gram은 특정파일의 구조를 반영하는 정보를 함축하고 있으며 바이러스 침입자가 쉽게 예측하기 어려운 것으로 알려져 있다. 분석하고자하는 코드에 대하여 이미 저장된 시스내처로부터 k-nearest 알고리즘을 적용하여 정상코드인지 악성코드인지 분류 하게 된다. 이 방법은 파라메타 n과 추출된 시스내처의 수 L에 따라 민감하게 그 성능이 좌우되는 것으로 보고 되었으나 알려지지 않은 새로운 파일을 대상으로 하는 초기연구로서 가치가 있다.

Kolter 등[13]도 비슷한 방법으로 접근하고 있으나 정보공학의 기법을 적용하였다. 특징으로 추출된 n-gram 들이 준비된 각 파일에 존재여부를 지시하는 이진데이터를 모아 평균상호 정보(average mutual information)를 계산하여 그 값이 큰 500개의 데이터를 선택하여 WEKA[11]에서 구현한 학습방법-Instance-based Learner, TFIDF, Naive Bayes, a support vector machines, a decision tree and a booted classifier-에 적용하였다. 준비된 데이터의 분류정확도에 의하여 분류하지 않고 ROC(receiver operating characteristics)에 의하여 평가하였다.

IBM T.J. Watson Research Center에서 바이러스 탐지를 위한 신경망 접근방법이 연구되었고[8] 이와 비슷한 방법이 Anti-Virus 소프트웨어에 부트 섹터 바이러스 탐지를 위하여 활용되어 수백만 대의 PC에 배포되기도 하였다[9]. 이러한 접근방법은 신경망에 제공되는 입력 데이터 표현과 분류를 위해 사용되는 임계치의 결정을 중심으로 연구되었다. 단층 신경망에 trigram feature를 입력으로 적용하여 의미없는 false positive rate를 발생시키기도 했지만 90%이상의 분류율을 얻었다. 이를 일반화하여 512 바이트 스트링을 입력으로 다층구조의 역전파신경망에 학습시킨 결과 0.5의 임계치에 대하여 90-95%의 분류율을 얻었고 IBM 바이러스 전문가의 조언을 얻어 임계치를 0.7로 조절한 경우 false positive rate는 거의 0에 가깝고 false negative rate는 약간 증가한 것으로 보고되었다. 최근 비교사학습신경망인 SOM(Self Organizing Maps)에 의하여 바이러스 패턴을 가시화시키는 연구결과가 발표되었다[14]. 이 연구에서는 SOM안의 뉴런들이 Windows executable files안의 특정패턴의 존재를 알리도록 설계되었으며 비슷한 위치의 뉴런들이 활성화되면 유사한 바이러스가 발견됨을 확인할 수 있었다.

이러한 고찰을 통하여 신경망의 학습능력은 컴퓨터 바이러스 분류에 활용될 수 있는 잠재력을 가지고 있음을 확인하였다. 기존의 연구는 신경망 구성에 있어 필요한 학습데이터를 모두 수집한 후에 이를 구성된 네트워크에 맞는 입력패턴으로 표현하고 학습하게 된다. 신경망 뿐 아니라 기존의 기계학습에도 학습은 한꺼번에 이루어지고있다. 이때 새로운 데이터가 수집되면 처음부터 다시 학습해야하기 때문에 비효율적이고 수집 되는대로 학습하고 또 수집되면 그 위에 학습하는 점증적 학습모델이 필요하다[6,7].

3. 점증적 학습기능을 가지는 적응 분류 모델

지금까지 제안된 분류를 위한 클러스터분석 방법들은 FCM[3,4]이나 FNN-B[5]와 같이 학습하고자하는 데이터가

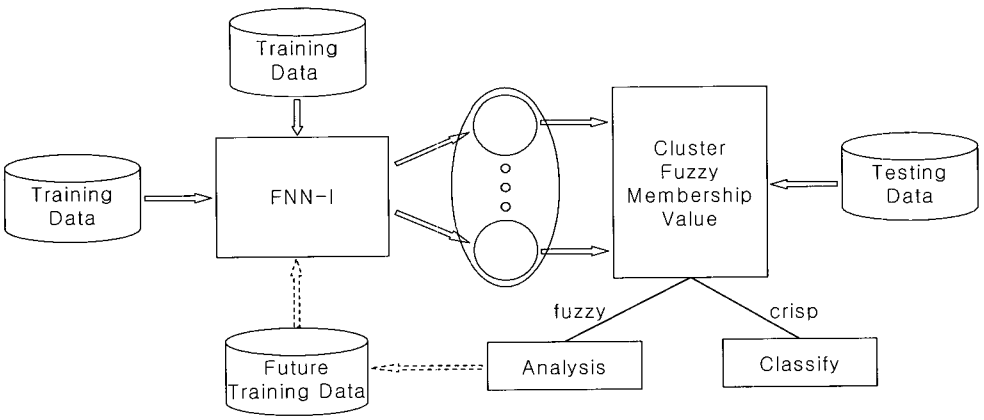

그림 2. 퍼지 신경망(FNN-I)을 이용한 적응 분류 모델

모두 준비된 후 학습하는 일괄학습(batch training) 방법을 취하고 있다. 이와 같은 방법은 구축하려는 시스템의 목표가 명확하고 데이터가 모두 수집된 경우 효과적으로 활용될 수 있다. 그러나 대부분의 실세계 응용의 경우 한꺼번에 학습을 위한 데이터를 모두 수집하기 어렵고 때로는 새롭게 수집된 데이터에 따라 시스템의 개발목표가 변하기도 한다. 본 논문에서는 계속해서 새롭게 발생하는 학습데이터에 대하여 점증적으로 학습하는 퍼지신경망, FNN-I를 구축하고 이를 이용한 (그림 2)와 같은 적응 분류모델을 설계한다.

3.1 점증적 학습 알고리즘

퍼지클러스터링 알고리즘으로 사용되어 온 FCM 알고리즘과 목적함수 기반 퍼지신경망 FNN-B는 일괄처리하는 학습방법에 기초하고 있다. 그러므로 데이터가 한꺼번에 준비되지 않는 현실세계의 대부분의 응용에서 시스템 개발은 계속적으로 지연되고 있다. 이를 위하여 (그림 3)의 퍼지 신경망 구조 FNN-I를 제안하고 지금까지 준비된 데이터 X에 대하여 학습한 후 다시 새롭게 모아진 데이터 Y에 대하여 단계 3의 과정부터 계속하여 학습할 수 있는 알고리즘을 다음과 같이 기술한다.

단계 1 : c, m, ϵ 의 값을 설정하고 입력데이터 셋을 준비한다.
c는 클러스터의 수, m은 FCM 알고리즘의 weighting exponent 이다.

단계 2 : 초기 가중치 벡터 $V=(v_1, v_2, \ldots, v_c)$와 퍼지 C 분할 U를 0과 1 사이의 난수로 초기화한다. $t=1$

단계 3 : 현재 가중치 벡터 $v_{i,t}$를 $v_{i,t-1}$로 저장한다.

단계 4 : 준비된 각 데이터 x_j 에 대하여

(a) 다음 식을 이용하여 α_{ij} 를 계산한다.

$$\alpha_{ij}=\frac{2m}{m-1}\left\{\sum_{i=1}^{c}(u_{ij})^{m+1}\left(\frac{\|x_j-v_i\|^2}{\|x_j-v_i\|^2}\right)^{\frac{m}{m-1}}\right\}$$

(b) 다음 식을 이용하여 가중치 벡터를 수정한다.

이때 $\eta=\frac{1}{n}$ 라고 하자.

$$\triangle v_i=\eta\sum_{j=1}^{n}\alpha_{ij}(x_j-v_{ij}) \qquad (3)$$

(c) 다음 식을 이용하여 퍼지 C 분할 U를 계산한다.

$$u_{ij}=\frac{1}{\sum_{s=1}^{c}\left\{\frac{\|x_j-v_i\|}{\|x_j-v_s\|}\right\}^{2/(m-1)}}$$

단계 5 : $v_{i,t}$ 는 현재 가중치 벡터이고 $v_{i,t+1}$ 는 단계4-(c)에 의하여 수정된 가중치 벡터일때 $diff=\sum_{i=1}^{c}\|v_{i,t+1}-v_{i,t}\|^2<\epsilon$ 이면 알고리즘을 끝내고 그렇지 않으면 단계 3으로 가자

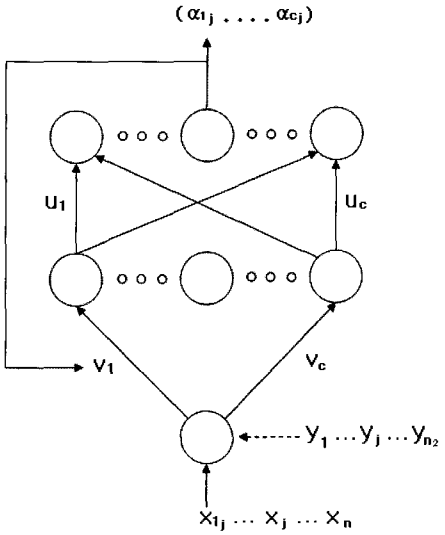

그림 3. 점증적 학습 퍼지신경망 구조(FNN-I)

이렇게 마련된 학습알고리즘을 성능을 확인하기 위하여 널리 알려진 벤치마크 데이터인 iris data를 준비하여 점증적으로 학습시켜 보았다. Anderson의 iris data set은 iris 식물의 형태로부터 추출된 4가지 속성의 값을 가지며 각각 50개의 데이터로 구성된 3 종류의 데이터로 구성되어 있다. 그 중 한 가지 종류는 다른 것들로부터 선형분리 가능하고 나머지 두 종류는 선형분리 가능하지 않은 것으로 알려져 있다. 각 클래스에서 20개의 데이터씩 먼저 60개를 학습하고 그 다음 10개씩 30개를 학습하고 마지막으로 나머지 60개를 학습하는 방식으로 3번에 나누어 점증적으로 학습하였다. 그 결과를 요약하면 다음 <표1>과 같다.

<표1> Iris data 학습 결과

	학습한데이터 수	반복횟수	정확도
FNN-B	150	13	91.2
FNN-I	60/30/150	12/10/18	90.4

<표1>로부터 세 번에 나누어 점증적으로 학습한 FNN-I가 150개의 학습데이터를 모두 가지고 일괄 학습한 FNN-B와 유사한 결과를 얻었음을 확인할 수 있다. 일괄 학습의 경우 모든 데이터를 가지고 학습하므로 성능이 더 좋은 것은 당연하다. 평균적으로 하나의 데이터에 대한 학습 반복횟수는 FNN-I의 경우 더 적은 것을 알 수 있다.

3.2 컴퓨터 바이러스 분류 모델

제안된 FNN-I를 이용하여 학습한 클러스터는 (그림 2)의 분류모델에 활용되어 다음과 같은 과정으로 분류하게된다. 우선 주어진 테스트 데이터에 대하여 각 클러스터의 소속값을 계산한다. 퍼지 소속 값은 FCM 알고리즘 구축과정에서 얻은 결과를 가지고 식(2)을 이용하여 계산한다. 주어진 데이터 x_j 에 대하여 각 클러스터 i 에 대하여 u_{ij} (i=1,2, . . , c)를 구하면 x_j 가 각 클러스터 정보에 일치하는 정도를 알 수 있다.

다음 단계로서 u_{ij} (i=1,2, . . , c)를 통하여 주어진 데이터

x_j 가 이미 학습된 사전지식을 가지고 분류할 수 있는 척도를 측정하게 된다. 이를 주어진 데이터의 결정상태라고 정의하고 (정의1)에 의하여 퍼지 상태(fuzzy status)와 분명한 상태(crisp status)로 분류 하게 된다.

[정의 1] 다음 식(4)의 조건을 만족하면 주어진 데이터 x_j는 이미 획득된 지식에 대하여 퍼지 상태(fuzzy status)인 것으로, 아니면 분명한 상태(crisp status)로 정의된다.

$$\frac{u_{ij}-u_{sj}}{u_{ij}} \le \frac{1}{c}, where\, u_{ij}=\max\{u_{ij}\}, u_{sj}=\max\{u_{ij}\}, s \neq l \quad (4)$$

(정의 1)은 주어진 데이터의 가장 큰 소속 값과 두 번째로 큰 소속 값의 차이가 충분히 크면 데이터를 판정하기에 필요한 지식을 이미 가지고 있는 상태(crisp, known)임을 말해주며 그렇지 않으면 주어진 데이터는 현재 학습한 정보로는 판단하기 어려운 상태(fuzzy status)임을 말해 주고 있다. crisp 상태인 경우 바이러스를 분류해 주고 fuzzy 상태인 경우 데이터를 모아 전문가에 의해 분석한 후 다음 학습 단계의 학습데이터로 활용되기도 한다.

4. 실험 및 고찰

4.1 데이터 준비

FNN-I와 같은 신경망 구조에 문제를 적용하기 위하여 가장 중요한 과정은 입력데이터의 표현이다. 컴퓨터바이러스 분류를 위하여 학습할 파일을 준비하고 여기서 입력에 사용될 특징패턴을 추출하여 입력데이터를 준비하게 된다. 이를 위하여 VX heaven[15]로부터 450개의 바이러스 파일과 윈도우시스템 실행 파일로부터 450개의 정상파일을 수집하여 다음과 같은 전 처리 과정을 통하여 제안된 모델의 입력 데이터를 준비하였다. 우선 수집한 이진실행파일(binary executables)을 IDA Pro[16]를 사용하여 역어셈블 한다. 이 과정에서 대부분의 실행파일은 완전하게 역어셈블 되지 않기 때문에 경우에 따라 휴리스틱을 사용하여 추출하기도 한다. 이제 역어셈블 된 코드를 블록으로 나누고 각 블록에 대하여 블록이름과 그 안에 있는 명령어의 연산 코드(instruction operation code)로 구성된 중간 파일을 만든다. 이 중간 파일로부터 각 명령어의 출현빈도수를 구하고 이를 바탕으로 특정 연산코드를 추출하게 된다. 이러한 처리과정은 악성코드의 탐지는 정상코드와 구별되는 특징패턴으로부터 알 수 있고 그런 정보는 파일을 구성하는 명령어로부터 얻어질 수 있다는 자연스러운 아이디어에서 출발하고 있다. 특히 명령어는 연산코드와 피연산자 부분으로 되어 있는데 연산코드만으로도 파일의 내용을 모두 표현할 수 있고 특징패턴을 추출할 수 있다. 이때 피연산자 부분은 고려하지 않으므로 중간파일의 크기를 상당히 줄일 수 있고 입력데이터의 준비를 단순화 시킬 수 있다. 본 실험에서는 명령어 열(instruction sequence)의 출현빈도수와 바이러스 파일에 자주 나오는 명령어 열은 정상파일에는 잘 나오지 않는다는 휴리스틱을 이용하여 26개의 명령어 열을 특징패턴으로 선정하였다[10]. 이와 같은 특징패턴 추출과정은 중요한 연구과제이고 시스템의 성능을 좌우하는 중요한 요소이나 본 논문에서는 그 개념을 도입하여 데이터 준비과정으로 활용하였다. 이제 마련된 특징패턴을 가지고 중간파일을 분석하여 각 특징패턴의 파일 안에서의 정규화 된(normalized) 출현횟수를 구한다. 이렇게 만들어낸 900×26의 데이터를 300×26의 세 그룹으로 나누어 점증적으로 학습하였다.

4.2 실험 및 고찰

900×26 입력데이터는 3종류의 바이러스 파일과 정상파일로부터 준비하였으므로 클러스터의 수 c를 4로 하고 학습을 시작한다. 준비한 입력데이터로부터 나눈 세 그룹의 300×26의 입력데이터를 DataA, DataB, DataC라고 하자. 각각 학습한 후 제안된 FNN-I가 점증적으로 학습하고 있음을 알아보기 위하여 테스트 데이터를 준비한다. 1차 학습의 경우 학습결과를 테스트하기 위하여 학습에 참여한 DataA를 테스트 데이터로 하여 93.5%의 분류정확도를 얻었고 학습에 참여하지 않은 300×26의 테스트 데이터 TestA를 분류한 결과 78.2%의 분류정확도를 얻었다. 2차 학습의 경우 학습에 참여한 DataB를 테스트 데이터로 하여 92.7%의 분류정확도를 얻었으며 학습에 참여하지 않은 300×26의 테스트 데이터 TestB를 분류한 결과 81.4%의 분류정확도를 얻었다. 3차 학습의 경우 학습에 참여한 DataC를 테스트데이터로 하여 93.1%의 분류정확도를 얻었으며 학습에 참여하지 않은 300×26의 테스트 데이터 TestC를 분류한 결과 82.3%의 분류정확도를 얻었다. 이를 요약한 <표 2>로 부터 3차의 경우 모두 점증적으로 학습이 진행되면서 TestA, TestB, TestC와 같은 학습에 참여하지 않은 새로운 유형의 데이터를 더욱 잘 분류하는 것을 알 수 있다. 이를 기존의 FNN-B와 비교하기 위하여 전체 학습 데이터로부터 추출한 300×26의 테스트 데이터 DataD 와 전체 학습에 참여하지 않은 비슷한 유형의 바이러스 파일과 정상파일로부터 준비한 300×26의 테스트 데이터 TestD를 가지고 분류한 결과를 요약하면 <표 3>와 같다. 이때 학습에 사용한 DataD를 가지고 실험한 경우, FNN-B와 FNN-I의 분류율은 각각 92.5%, 90.4%로서 FNN-B가 나은 결과를 얻었으나, 학습에 사용하지 않은 새로운 데이터 TestD의 경우는 각각 80.7%, 81.9%의 분류율을 보여주므로 논문에서 제안한 FNN-I가 더 나은 결과를 얻은 것을 확인할 수 있다.

이와 같은 실험 결과는 같은 유형의 데이터에 대하여 100번 실험한 평균치이고 퍼지 상태로 판별된 경우는 분류에 실패한 것으로 하였고 이 경우 (그림 2)의 제안된 적응 분류 모델에서 보여주는 바와 같이 전문가의 도움을 받아 처리하고 나중에 추가 학습을 위한 데이터로 활용될 수 있도록 하였다. 또한 이 경우는 지금 형성된 클러스터로는 판단하기 어려운 데이터이고 파라메터에 따라 다른 결과를 나타내는 것을 확인하였다. 모든 데이터를 한꺼번에 학습한 FNN-B가 이미 학습한 데이터에 대하여 더 나은 결과를 얻었지만 학습하지 않은 새로운 데이터에 대하여 논문에서 제안한 FNN-I가 더 나은 결과를 얻은 것을 확인할 수 있다.

<표 2> 세 번의 학습후 FNN-I의 분류율

1차 학습 후	93.5%(DataA)	78.2%(TestA)
2차 학습 후	92.7%(DataB)	81.4%(TestB)
3차 학습 후	93.1%(DataC)	82.3%(TestC)

<표 3> FNN-B와의 분류율 비교

	FNN-B	FNN-I
DataD	92.5%	90.4%
TestD	80.7%	81.9%

5. 결 론

본 논문에서는 경계가 불명확한 대부분의 실세계 데이터에 대하여 퍼지 집합표현과 소속 값 처리 방법을 도입하고 계속 수집되고 있는 데이터에 대하여 점증적으로 학습하는 퍼지 신경망 FNN-I를 구축하고 이를 이용한 적응 분류 모델을 설계하였다. 또한 이를 컴퓨터 바이러스 분류를 위한 실제 데이터에 적용하여 분류 모델이 점증적으로 학습할 수 있음을 확인하고 FNN-B를 적용한 경우와 비교하여 그 타당성을 검토하였다. 또한 현재 학습한 데이터로는 판단하기 어려운 테스트 데이터인 경우 전문가에 의해 처리되고 수집되어 다음 학습단계에 반영될 수 있도록 하여 시간에 따라 변하는 데이터에 적응할 수 있도록 하였다. 실험을 통하여 모든 데이터를 한꺼번에 학습한 FNN-B가 이미 학습한 데이터에 대하여 더 나은 결과를 얻었지만 학습하지 않은 새로운 데이터에 대하여 논문에서 제안한 FNN-I가 더 나은 결과를 얻은 것을 확인하였다.

본 연구는 점증적 학습을 통하여 적응하는 분류시스템의 기본적인 시도로서 이와 같은 시스템의 성능은 준비된 샘플 데이터와 준비과정에서 사용한 파라메터 등에 따라 다르다. 그러므로 다양한 실험 데이터에 대한 체계적인 분석과정을 통하여 수정 보안 되어야 한다. 또한 분류에 실패했던 퍼지 상태의 데이터를 모아 학습할 때 클러스터를 재조정할 수 있는 메커니즘이 마련하여 보다 체계적으로 적응할 수 있는 방법을 계속적으로 연구해야할 것이다.

Reference

[1] Gupta, M. M., Jin, L., and Homma, N., Static and Dynamic Neural Networks : From Fundamentals to Advanced Theory, Wiley-IEEE Press, April 2004.

[2] Yi-Ta Wu, Yoo Jung An, James Geller and Yih-Tyng Wu, "A Data Mining Based Genetic Algorithm", Proc. og the 4th IEEE workshop on SEUS-WCCIA, 2006.

[3] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum press, New York, 1981.

[4] Jian Yu and Miin-Shen Yang, "Optimality Test for Generalized FCM and Its Application to Parameter Selection", IEEE Transactions on Fuzzy Systems, Vol. 13, No. 1, Feb. 2005.

[5] 이현숙, "퍼지 성능 측정자를 이용한 적응 데이터 마이닝 모델", 정보처리학회 논문지, 제13-B권 5호, 2006.

[6] Constantinos Constantinopoulos and Aristidis Likas, "An Incremental Training Method for the Probabilistic RBF Network", IEEE Trans. on Neural Networks, Vol. 17, No. 4, July 2006.

[7] Vicente O. Baez-Monroy and Simon O'Keefe, "Modelling Incremental Learning With The Batch SOM Training Method", Proc. of 5th International Conference on HIS, 2005.

[8] William Arnold, Gerald Tesauro, "Automatically generated Win32 heuristic virus detection", Virus Bulletin conference, September, 2000.

[9] G. Tesauro, J. Kephart, and G. Sorkin, "Neural networks for computer virus recognition", IEEE Expert, 11:5-6, August 1996.

[10] Jianyong Dai, Joohan Lee and Morgan C. Wang, "Detecting Unknown Computer Virus Using Data Mining Techniques", Business Intelligent Symposium, poster presentation, April, 2006.

[11] I. Witten and E. Frank, "Data mining: Practical machine learning tools and techniques with java implementations", Morgan Kaufmann, San francisco, CA, 2000.

[12] Abou-Assaleh, Nick Cercone, Vlado Keselj, and Ray Sweidan, "Detection of New Malicious Code Using N-grams Signatures, Proceedings of the Second Annual Conference on Privacy, Security and Trust (PST'04), pp. 193-196, 2004.

[13] Kolter, J.Z., and Maloof, M. A., "Learning to detect malicious executables in the wild", In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 470-478. New York, NY, 2004.

[14] InSeon Yoo, "Visualizing Windows Executable Viruses Using Self-Organizing Maps", Proceedings of the 2004 ACM workshop on Visualization and Data Mining for Computer Security, 2004.

[15] VX Heaven : http://vx.netlux.org

[16] http://www.datarescue.com

저 자 소 개

이현숙(Rhee, Hyunsook)

1989년 : 서강대학교 전자계산학과(학사)
1991년 : 포항공과대학교 컴퓨터공학과(석사)
1997년 : 서강대학교 컴퓨터학과(박사)
1991년~1997년 : 한국전자통신연구소(ETRI) 연구원
1997년~현재 : 동양공업전문대학 전산정보학부 부교수

email : hsrhee@dongyang.ac.kr