변형된 캐스케이드-상관 학습 알고리즘을 적용한 그룹 고장 데이타의 소프트웨어 신뢰도 예측

이 상 운† · 박 중 양††

요 약

많은 소프트웨어 프로젝트는 시험이나 운영단계에서 고장 시간이나 고장 수 데이타보다 그룹 고장 데이타(여러 고장 간격에서 또는 가변적인 시간 간격에서의 고장들)가 수집된다. 본 논문은 그룹 고장 데이타에 대해 가변적인 미래의 시간에서 누적 고장 수를 예측할 수 있는 신경망 모델을 제시한다. 2개의 변형된 캐스케이드-상관 학습 알고리즘을 제안하였다. 제안된 신경망 모델들은 다른 잘 알려진 신경망 모델과 통계적 소프트웨어 신뢰도 성장 모델과 비교되었다. 실험 결과, 그룹 데이타에 대해 변형된 캐스케이드-상관 학습 알고리즘이 좋은 예측 결과를 나타내었다.

Software Reliability Prediction of Grouped Failure Data Using Variant Models of Cascade-Correlation Learning Algorithm

Sang-Un Lee† · Joong-Yang Park††

ABSTRACT

This Many software projects collect grouped failure data (failures in some failure interval or in variable time interval) rather than individual failure times or failure count data during the testing or operational phase. This paper presents the neural network (NN) modeling for grouped failure data that is able to predict cumulative failures in the variable future time. The two variant models of cascade-correlation learning (CasCor) algorithm are presented. Suggested models are compared with other well-known NN models and statistical software reliability growth models (SRGMs). Experimental results show that the suggested models show better predictability.

키워드 : 캐스케이드-상관 학습 알고리즘(Cascade-correlation learning algorithm), 그룹 고장 데이터(Grouped failure data), 훈련제도(Training regime), 평가 척도(Evaluation metrics)

1. 서 론

소프트웨어 시스템 개발은 사용자가 의도하는 목적에 적합한 충분한 품질 수준에 일치해야 하며, 요구되는 품질 수준에 일치하는지 여부를 평가할 수 있는 척도가 필요하다. 일반적으로 소프트웨어의 품질을 정량적으로 표현하는 속성 중의 하나가 소프트웨어 신뢰도이다[1, 2]. 소프트웨어 신뢰도는 주어진 환경 하에서 주어진 시간동안 소프트웨어가 고장 없이 작동할 확률로 정의된다[1, 3].

소프트웨어 신뢰도를 추정 또는 예측하기 위해, 소프트웨어 시험이나 운영 중에 수집되는 고장 데이타 형태는 고장 수(일정한 시간 간격동안 발생한 고장 수 기록)나 고장 시간(각 고장이 발생한 시점을 기록하는 고장 시간 데이타) 이며, 원래의 고장시간 데이터보다는 고장 시간 데이타를 그룹화한 형태인 그룹 데이타(Grouped data)만 이용하는 경우도 있다[4].

예로, 그룹 데이타는 각 고장시간을 CPU 시간으로 측정한 데이타를 1주일 간격으로 그룹화 하는 경우 그룹 간격은 일정(1주일) 하지만 실제의 그룹화된 CPU 시간은 가변적이 된다. 따라서, 그룹 데이타의 특징은 가변적인 시간 간격을 가지므로 일정한 시간 간격에 적합한 고장 수나 고장시간 데이타에 적합한 모델로 가변적인 미래 시점에서의 신뢰도를 예측하는데는 문제가 발생될 수 있다. 즉, 일정한 간격으로 추출한 표본에 적합한 모델이 가변적인 간격에서 추출한 표본에 적합한 모델이 될 수 없다. 이 문제를 해결하기 위하여 가변시간 간격 데이타에 적합한 모델을 찾기 위해 전향망(FeedForward Networks : FFNs) 신경망에 대해 필수적인 입력을 결정하는 연구를 수행한 바 있다. 그룹 고장 데이타에 대해 Karunanithi et al.[5]을 비롯하여 많은 신경망 모델이 제시되었으나 보다 일반화되고, 성능이 보다 좋은 모델은 제시하지 못하였다.

† 정 회 원 : 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당
†† 정 회 원 : 경상대학교 통계학과 교수
논문접수 : 2000년 8월 31일, 심사완료 : 2001년 6월 8일

본 논문에서는 가변적인 고장 수집 시간 간격을 가진 그룹 데이타에 적합한 신경망 모델을 개발하는 연구를 수행하고자 한다. 제2장에서는 관련 연구 및 문제점을, 제3장에서는 캐스케이드-상관 학습 알고리즘(Cascade-Correlation (CasCor) Learning Algorithm)[6]을 고찰해본다. 제4장에서는 변형된 CasCor 알고리즘을 제안하고, 제5장에서는 변형된 CasCor 알고리즘을 이용하여 모델의 예측 결과를 실험한 후 널리 알려진 신경망과 통계적 모델들과 상호 비교함으로서 제안된 모델의 적합성을 살펴보고자 한다.

2. 관련 연구 및 문제점

데이타 쌍들 $(t_i, m_i), i=1,2,\cdots,n$이 소프트웨어 시험 결과 관찰되었다고 가정하자. 여기서 m_i는 i 번째 시험 시간인 t_i까지 발견된 누적 고장 수를 의미하며, 단위 시험시간 $\Delta t_i = t_i - t_{i-1}$로 일정하며, i번째 단위 시험시간에서 발견된 고장 수 $\Delta m_i = m_i - m_{i-1}$로 정의된다. 이들 데이타 쌍들을 고장 수 데이타라 칭한다.

Karunanithi et al.[5, 7]는 신경망을 활용하여 소프트웨어 신뢰도 예측을 모델링하였으며, 소프트웨어 신뢰도 예측에 CasCor 알고리즘[6]을 이용하여 FFNs과 Partial Recurrent Networks(PRNs)인 Jordan과 Elman망이 적용될 수 있음을 보였다. 신경망의 예측력을 평가하기 위해 2가지 훈련 제도(Training regimes)를 고려하였다.

- 훈련제도 (1) : 입력(Input) t_i는 목표(Target) m_i에 관련
- 훈련제도 (2) : 입력 t_{i-1}는 목표 m_i에 관련

Karunanithi et al.[5]이 제안한 훈련제도는 가변 시간간격을 가진 그룹 데이타에 적합하지 않고 단지 일정한 시간간격을 가진 고장 수 데이타에 적합한 훈련제도로 볼 수 있다.

신경망을 함수 근사, 회귀분석 또는 시계열 분야에 이용하는 경우, 주어지는 시간 간격은 일정한 데이타를 활용하는 것이 일반적이며, 또한 소프트웨어를 시험하면서 수집되는 고장 수 데이타는 일반적으로 시간 간격이 일정한 형태를 취한다. 그러나 <표 1>과 같이 데이타 수집 시간 간격 Δt_i이 일정하지 않은 그룹 데이타만 이용 가능할 경우도 발생한다.

가변 시간간격을 가진 그룹 데이타를 기존의 고정된 시간간격에 적합한 신경망 모델을 이용하여 미래의 가변 시간간격 시점에서의 고장 수를 예측할 경우 문제가 발생한다. 즉, 가변 시점에서 표본을 추출한 데이타를 일정한 간격으로 표본을 추출한 데이타에 적합한 모델을 이용시 정확한 추정 및 예측이 불가하다. 예를 들면, Karunanithi et al.[5, 7]이 제안한 예측 훈련제도의 경우 입력은 t_{i-1}인데 목표는 고정된 시간간격인 t_i 시점에서의 누적 고장 수인 m_i이다. 따라서, 그룹 데이타의 가변적 시간간격인 t_i시점까지의 누적 발견 고장 수를 예측하려면 가변 시간간격 $\Delta t_i = t_i - t_{i-1}$ 정보를 알아야 한다. 그러나 기존 모델링에서는 Δt_i 정보가 전혀 들어있지 않음으로 문제가 발생한다. 이 모델이 가능하려면 Δt_i가 일정할 때만 적용 가능하다.

〈표 1〉 그룹 데이타 특성 및 고장 데이타

데이타	참고문헌	LOC	고장 수	Data Size	Application(Time interval)
Data 4	[8]	1,317,000	328	17	PL/I database application software (CPU execution time in Hours)
Data 5	[8]	35,000	279	10	Hardware control program (Time of Observation in Months)
System T1	[4]	21,700	136	21	Real time command control (CPU execution time in Hours)
System T38	[4]	Not known	32	11	Not known (CPU execution time in Hours)

Data4					Data5				
그룹 번호	t_i	Δt_i	Δm_i	m_i	그룹 번호	t_i	Δt_i	Δm_i	m_i
1	2.45	2.45	15	15	1	1.0	1.0	10	10
2	4.9	2.45	29	44	2	1.5	0.5	64	74
3	6.86	1.96	22	66	3	2.0	0.5	18	92
4	7.84	0.98	37	103	4	3.0	1.0	43	135
5	9.52	1.68	2	105	5	4.5	1.5	44	179
6	12.89	3.37	5	110	6	6.0	1.5	13	192
7	17.1	4.21	36	146	7	8.0	2.0	34	226
8	20.47	3.37	29	175	8	11.0	3.0	28	254
9	21.43	0.96	4	179	9	12.0	1.0	15	269
10	23.35	1.92	27	206	10	13.0	1.0	10	279
11	26.23	2.88	17	223					
12	27.67	1.44	32	255					
13	30.93	3.26	21	276					
14	34.77	3.84	22	298					
15	38.61	3.84	6	304					
16	40.91	2.3	7	311					
17	42.67	1.76	17	328					

System T1					System T38				
그룹 번호	t_i	Δt_i	Δm_i	m_i	그룹 번호	t_i	Δt_i	Δm_i	m_i
1	0.0092	0.0092	2	2	1	5	5	1	1
2	0.0192	0.01	0	2	2	15	10	0	1
3	0.0222	0.003	0	2	3	25	10	16	17
4	0.0452	0.023	1	3	4	35	10	1	18
5	0.0862	0.031	1	4	5	45	10	1	19
6	0.0902	0.004	2	6	6	50	5	0	19
7	0.1152	0.025	1	7	7	65	15	1	20
8	0.4172	0.302	9	16	8	75	10	3	23
9	1.3902	0.973	13	29	9	95	20	2	25
10	1.4102	0.02	2	31	10	120	25	7	32
11	1.8602	0.45	11	42	11	125	5	0	32
12	2.1102	0.25	2	44					
13	3.0502	0.94	11	55					
14	4.3902	1.34	14	69					
15	7.7102	3.32	18	87					
16	11.2702	2.56	12	99					
17	13.9302	2.66	12	111					
18	17.7002	3.77	15	126					
19	21.1002	3.4	6	132					
20	23.5002	2.4	3	135					
21	25.3002	1.8	1	136					

이때는 t_i들을 i로 대체하여도 동일한 결과를 얻을 수 있다. 즉, Δt_i가 가변 시간 간격일 경우, 기존 신경망 모델로는 Δt_i 시간 후에 발견되는 고장 수는 예측이 불가하다. 이 문제를 해결하기 위해서는 현재 시점인 t_{i-1}에서 얼마 동안의 기간에 발생하는 고장 수를 예측하는가 하는 가변 시간간

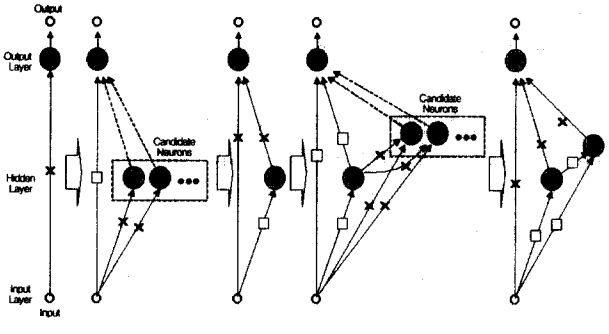
격(Δt_i)이 주어져야 한다. 이러한 그룹 데이타에 적합한 신경망 연구로, 그룹 데이타에 적합한 신경망 모델을 개발하기 위해 다음과 같은 훈련제도를 연구하였다.

- 훈련제도 (3) : 입력 t_i와 Δt_i는 목표 m_i에 관련
- 훈련제도 (4) : 입력 t_{i-1}와 Δt_i는 목표 m_i에 관련

그러나 Karunanithi et al.[5]의 CasCor 알고리즘은 훈련제도 (1)과 (2)를 사용시 0~4개의 은닉 뉴런 수를 사용하여 모델이 단순한데 반해, 이상운 et al.[9]은 훈련제도 (3)과 (4)를 사용한 전향망의 경우 Karunanithi et al.의 연구결과보다 많은 은닉 뉴런 수가 필요 하였다. 또한 모델의 예측 성능도 다른 잘 알려진 통계적 모델 보다 월등히 좋은 결과를 얻지 못하였다. 그러나 그룹 데이타에 적합한 신경망의 입력으로 가변 시간간격 데이타인 Δt_i 정보가 필수적으로 필요함을 보였다.

3. CasCor 알고리즘

CasCor 알고리즘은 2가지 주요 아이디어를 내포하고 있다. 첫째로, 후보 뉴런을 상관(엄밀히 말해 공분산)이 최대가 되도록 훈련시킨다. 둘째로, 후보 뉴런이 은닉층에 추가되면서 은닉층이 캐스케이드 형태를 취한다. 즉, 새로 추가되는 은닉 뉴런은 신경망 입력과 기존에 추가된 은닉 뉴런들 모두에 연결이 된 형태로 새로운 1개의 은닉층을 형성하여 복잡한 캐스케이드의 다층 구조를 형성한다. 1-입력, 1-출력으로 구성된 문제를 풀기 위한 CasCor 알고리즘의 훈련 과정은 (그림 1)에 표현되어 있다.



(그림 1) CasCor 알고리즘(□ : 가중치 고정, × : 가중치 조절)

Step 1. 초기화 : 은닉층이 없이 입력과 출력층으로만 구성되며, SSE(Sum of Squared Error)를 최소화시키도록 입력-출력층 간 연결 가중치와 출력층 뉴런의 바이어스 가중치를 훈련시킨다.

Step 2. 후보 뉴런 훈련 : 입력층 및 기존의 은닉층 뉴런에서 후보 뉴런(일반적으로 8개 사용)으로 연결되고, 후보 뉴런의 출력이 출력층에는 연결되지 않는다("유령 또는 가상" 연결 상태). 이는 후보 뉴런의 작동이 신경망의 출력에 영향을 미치지 않음을 의미하며, 단지 후보 뉴런의 입력 연결 가중치를 훈련시키기 위한 공분산을 계산하기 위해 신경망의 출력 오차를 이용함을 의미한다. 후보 뉴런의 출력과 출력층 뉴런의 오차간의 공분산이 최대가 되도록 후보 뉴런의 입력 연결 가중치를 훈련(입력 가중치 조절)시킨다.

Step 3. 공분산 최대 후보 뉴런을 은닉층에 추가 : 후보 뉴런 중에서 가장 최적의 뉴런(공분산이 최대가 되는 뉴런)을 선택하여 기존의 신경망에 추가하며, 은닉층에 추가된 후보 뉴런은 입력층과, 이전에 추가된 모든 은닉 뉴런과 연결되고, 이 연결의 가중치는 후보 뉴런 훈련에서 결정된 값을 가지며, 고정된다. 추가된 후보 뉴런과 출력층이 연결된다.

Step 4. 출력층 뉴런 가중치 훈련 : 신경망의 오차(SSE)가 최소가 되도록 출력층에 연결된 모든 가중치(입력층, 기존에 추가된 은닉 뉴런들과 새로 추가된 은닉 뉴런으로부터)를 다시 훈련시킨다.

Step 5. 망의 성능 평가 : 훈련 결과 망의 성능을 평가하여 원하는 수준의 결과를 얻으면 훈련을 종료하고, 그렇지 않으면 Step 2로 복귀한다.

4. 변형된 CasCor 알고리즘

본 장에서는 가변적인 시간 간격을 가진 그룹 고장 데이타에 적합한 신경망 모델을 찾고자 한다. 소프트웨어 시험 또는 운영 중에 l 시점까지 획득된 가변 시간간격 고장 수 데이타 (t_i, m_i), $i=1,2,\cdots l$를 활용하여 미래의 가변 시간간격 시점 t_{l+r}에서 고장 수 m_{l+r}를 신경망을 이용하여 예측하는 경우, 최적의 모델 구조를 어떻게 선택할 것인가가 문제로 제기된다. 본 연구에서는 CasCor 알고리즘을 이용하되 Karumanithi et al.[5]와 Fahlman et al.[6]의 CasCor 알고리즘과 같이 은닉층과 출력층 뉴런의 작동함수로 모두 비선형인 시그모이드 함수를 사용하지 않고, 은닉층 뉴런의 작동함수로는 비선형인 시그모이드를, 출력층 뉴런의 작동함수로는 선형 함수를 사용한다. 출력층 뉴런의 작동함수로 선형을 사용하는 이유는 다음과 같다.

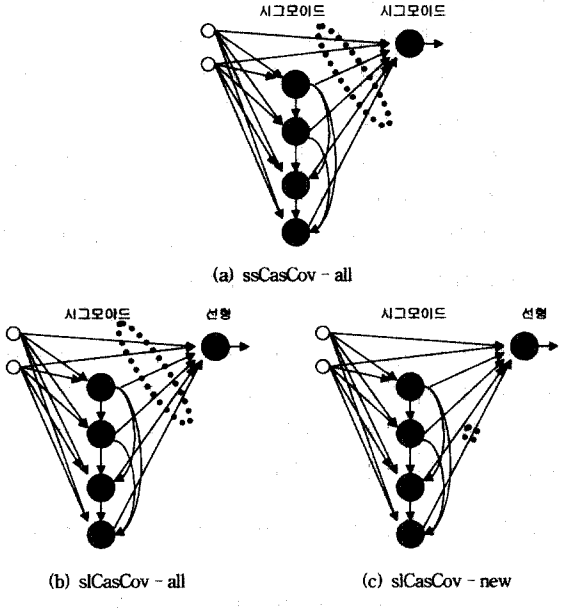
(1) Cybenko[10]는 비선형 함수인 시그모이드 작동함수를 가진 1개의 은닉층으로 주어진 모든 함수를 표현할 수 있다는 보편적 근사 이론(Universal Approximation Theorem)을 증명해 보였으며, Barron[11]은 은닉층에 시그모이드를 출력층에 선형 작동함수를 사용해, 1개의 은닉층을 가진 FFN이 함수근사 능력이 있음을 밝혔다. 따라서, 출력층 뉴런의 작동함수로 선형을 사용하면 주어진 문제를 모두 표현 가능하며, "공분산 최대화" 기준을 이용할 경우 두 변량간에 상관관계를 보다 잘 표현할 수 있다. 따라서, "공분산 최대화" 기준을 이용한 CasCor 알고리즘에서, 패턴분류나 함수근사 문제의 경우 출력층 뉴런의 작동함수로 선형을

사용하는 것이 보다 좋은 모델을 얻을 수 있다.

(2) CasCor 알고리즘에서 후보 뉴런의 입력 연결 강도를 훈련시키는 것으로 공분산 최대화를 사용하는 경우를 살펴보자. 공분산(또는 상관)은 두 함수간에 선형 관계가 있는가를 보기 위함이며, 두 함수가 모두 비선형인 경우, 상관계수가 크더라도 선형 관계가 크지 않은 결과를 초래할 수 있다. 따라서, 이 문제를 완화시키는 방법으로 1개의 함수(출력층 뉴런의 작동함수)를 선형으로 사용하는 것이 하나의 방법이 될 수 있다.

(3) 이상운[12]은 출력층 뉴런의 작동함수로 선형을 사용하여 정규화 시키지 않은 데이타를 이용하는 경우가 로지스틱 함수를 사용하여 정규화 시킨 데이터 보다 신뢰도 예측에 있어서 보다 좋은 결과를 얻었다.

CasCor 알고리즘을 변형시켜 새로운 모델을 개발하기 위해, Fahlman et al.[6]의 CasCor 알고리즘에서 출력층 뉴런 연결강도를 훈련시키는 방법을 고려해 보자. CasCor 알고리즘은 시그모이드-시그모이드 작동함수를 사용하고, 은닉 뉴런을 추가시킬 때마다 출력층 뉴런에 연결된 모든 연결강도를 다시 훈련시키는 과정을 수행한다. 이 모델을 ssCasCov-all이라 칭하자. ((그림 2) (a))



(그림 2) CasCor 알고리즘의 변형 모델

이 모델의 작동함수와 훈련 형태를 변형시킨 2개 모델을 제안한다. 첫째로, ssCasCov-all 모델과 동일하게 은닉층에 뉴런을 추가할 때마다 출력층 뉴런에 연결된 모든 연결강도를 다시 훈련시키는 방법을 사용하고, 시그모이드-선형 작동함수를 사용하는 모델을 slCasCov-all 이라고 칭하자. (그림 2) (b) 기존에 추가된 은닉 뉴런은 훈련을 통해 결정된 연결강도가 신경망의 출력 오차를 줄이는데 이미 역할을 담당하고 있다. 따라서, 후보 뉴런을 은닉층에 추가한 후 출력층에 연결된 연결 강도를 매번 다시 훈련시킬 경우 원하는 목표를 착아가다가 다시 원점으로 되돌아오는 결과를 토래한다. 그러므로 주어진 함수에 적합한(즉, 오차가 적은) 모델을 찾지 못하는 경우도 발생한다[13]. 이 문제점을 해결하기 위해, 두 번째로 제안하는 모델은 은닉뉴런이 새로 추가될 때, 기존 은닉 뉴런들과 출력층 뉴런간 연결강도를 고정시킨다. 다만, 새로 추가된 은닉 뉴런과 출력층 연결 강도만을 훈련시키고, 시그모이드-선형 작동함수를 사용하는 모델로 이를 slCasCov-new 라고 칭하자. ((그림 2) (c))

제안된 slCasCov-all과 slCasCov-new 모델을 이용하여 그룹 고장 데이타에 대한 다음단계 누적 고장 수를 예측하여 기존의 CasCor 모델인 ssCasCov-all 모델과 성능을 비교해 보고자 한다. 모델의 입-출력으로는 훈련제도 (3)과 (4)를 사용한다. 훈련제도에 따라 현재의 l 시점까지 획득된 데이타로 신경망의 입력과 목표 값으로 설정하고 신경망을 훈련시킨다. 훈련이 완료된 신경망에 대해 미래의 가변 시점인 다음 단계(Next-step)인 $l+1$ 시점에 해당하는 입력 값 $\{(t_l, \Delta t_l)$ 또는 $(t_{l-1}, \Delta t_l)\}$을 신경망에 가하면 훈련된 신경망의 학습 능력에 의해 다음단계의 누적 고장수인 $\hat{m}_{l+1}$이 출력되며, 이 값을 다음 단계의 누적 고장 수 목표값인 m_{l+1}와 비교(상대 오차)하여 훈련된 신경망의 예측 성능을 검증할 수 있다.

5. 예측력 실험 및 결과 분석

다른 여러 가지 모델들을 비교하는데 있어, 어떤 의미 있는 척도로서 모델의 예측 정확도를 정량화가 필요하다. 소프트웨어 신뢰도 분야에서 일반적으로 사용되는 방법은 적합도 검증(Goodness-of-fit), 다음 단계 예측력(Next-step predictability)과 가변 미래시점 예측력(Variable-term predictability)이다[5, 14]. 이들 접근법을 사용하여, Malaiya et al.[13]는 2가지 예측 척도로 평균 상대 예측 오차(Average relative prediction error : AE)와 평균 편향(Average bias : AB)을 제안하였다. AE는 모델이 얼마나 잘 예측하는가의 척도이며, AB는 모델의 일반적 편향이다. 본 논문에서는 <표 1>의 4개 데이타에 대한 다음단계 예측력을 평가하기 위해 AE 척도를 사용한다. <표 1>에 제시된 4개의 데이타에 대해, 훈련제도 (3)과 (4)를 slCasCov-all과 slCasCov-new 모델로 시그모이드-선형 작동함수를 사용하여 실험을 수행하였다. 각 훈련제도에 대한 모델의 예측력을 평가하기 위해 다음단계 AE 절대값의 평균인 $\overline{e}_{t,1}$을 사용한다[5].

주어진 데이터로부터 신경망의 적절한 구조와 가중치 값을 정확히 알 수 없기 때문에 대부분의 신경망 훈련방법은 신경망 가중치들의 초기 값을 랜덤하게 설정함으로 인해 신경망을 훈련시킬 때마다 예측력에 차이가 발생한다. 따라서 Karunanithi et al. [5]의 실험방법과 동일하게 50회의 훈련을 통해 평균값을 취한 것으로 예측 오차를 구하고자 한다.

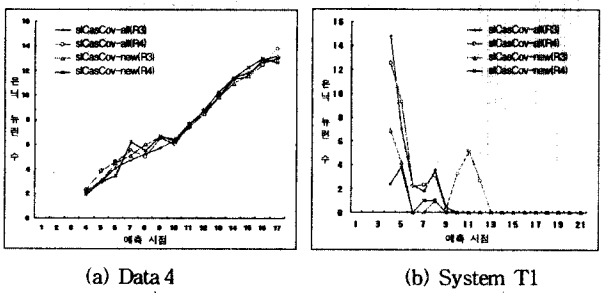
AE는 단일 데이타에 대해 모델들의 예측 정확도를 비교하는데 사용될 수 있으며, 다른 여러 가지 다양한 데이타들에 대해 모델을 비교하기 위해서는 다른 척도가 필요하다. Malaiya et al.[14]는 다양한 데이타들에 적합한 모델을 비교하는 척도로서 정규화된 AE(Normalized AE : NAE)를 제시하였다. 데이타 s에 대해 최대의 AE를 갖는 모델(m)을 m_{max}로, 이 모델의 AE를 $AE_s^{m_{max}}$라 하면 정규화된 상대오차는 $NAE_s^m = AE_s^m / AE_s^{m_{max}}$, $(0.0 < NAE_s^m < 1.0)$가 되며, NAE를 구한 후, 각 모델에 대한 순위 $R_m = \sum_{s=1}^{\#\ of\ data\ set} NAE_s^m$을 구하여 모델의 예측력을 평가할 수 있다. 제안된 4개 모델에 대한 다음단계의 전체 평균 예측오차 $\bar{e}_{t,1}$, NAE 및 모델의 예측력 순위는 <표 2>에 표기하였다.

실험결과, 다양한 환경에서 수집된 4개 데이타에 대해 훈련제도 (4)를 이용한 slCasCov-new 모델의 예측력이 가장 좋은 결과를 나타내었으며, slCasCov-all도 slCasCov-new와 유사한 예측 결과를 나타내었다. 따라서, 출력층 뉴런에 연결된 연결강도 훈련 방법과 주어진 입력을 어떤 것으로 선택하느냐에 따라 모델의 예측력에 영향을 미침을 알 수 있다.

〈표 2〉 모델의 예측력

모델	훈련제도	AE(NAE) Data 4	Data 5	System T1	System T38	R_m	순위
slCasCov-all	(3)	0.4457 (0.1504)	0.0403 (0.5935)	0.2102 (0.2760)	0.0623 (0.4357)	1.4556	2
	(4)	0.2300 (0.7571)	0.0679 (1.0000)	0.1595 (0.2094)	0.0180 (0.1259)	2.0924	3
slCasCov-new	(3)	0.3038 (1.0000)	0.0585 (0.8616)	0.7616 (1.0000)	0.1430 (1.0000)	3.8616	4
	(4)	0.2647 (0.8713)	0.0252 (0.3711)	0.0659 (0.0865)	0.0173 (0.1210)	1.4499	1

주어진 문제에 적합한 신경망의 구조(특히 은닉 뉴런 수)를 결정하는 것이 중요한 문제중 하나이다. Data4와 System T1에 대한 slCasCov-all과 slCasCov-new 모델에 대한 은닉 뉴런 수는 (그림 3)에 제시되어 있다. (그림 3)에서 Data4의 경우, 훈련 데이타의 크기가 클수록 은닉 뉴런 수도 증가하는 경향을 나타냄을 알 수 있다(Data5와 System T38도 Data4와 유사한 경향을 보임). 그러나 특별히 System T1만 감소하는 경향을 보이고 있다.



(a) Data 4 (b) System T1

(그림 3) 은닉 뉴런 수(Data4와 System T1)

제안된 모델이 일반적으로 많이 사용되는 통계적 모델들과 신경망 모델보다 좋은 예측 결과를 나타내는지를 살펴보고자 한다. 본 비교에는 Data4와 Data5에 대한 Karunanithi et al.[5]의 신경망과 통계적 모델들의 예측 결과와 제안 모델인 slCasCov-all과 slCasCov-new에 대한 NAE를 사용하였으며, <표 3>에 기술되어 있다.

〈표 3〉 다양한 모델의 예측력 비교

모델			Data 4 AE	Data 4 NAE	Data 5 AE	Data 5 NAE	R_m	순위
신경망	예측필터		6.27	0.6670	8.48	0.3630	1.0300	11
	제안된 CasCor 알고리즘 (시그모이드-선형)	slCasCov-all (R3)	0.45	0.0479	0.04	0.0017	0.0496	4
		slCasCov-all (R4)	0.23	0.0245	0.07	0.0030	0.0275	1
		slCasCov-new (R3)	0.30	0.0319	0.08	0.0034	0.0353	3
		slCasCov-new (R4)	0.26	0.0277	0.03	0.0013	0.0290	2
	Original CasCor 알고리즘[5] (시그모이드-시그모이드)	FFN(R1)	5.28	0.5617	10.00	0.4281	0.9898	9
		FFN(R2)	4.64	0.4936	6.95	0.2975	0.7911	5
		Jordan(R1)	8.84	0.9404	5.09	0.2179	1.1583	14
		Jordan(R2)	6.11	0.6500	8.67	0.3711	1.0211	10
	BP 알고리즘[9] (시그모이드-선형)	FFN(R3)	5.91	0.6287	7.77	0.3326	0.9613	7
		FFN(R4)	5.41	0.5755	11.97	0.5124	1.0879	12
통계적 모델[5]		Logarithmic	5.93	0.6309	6.42	0.2748	0.9057	7
		Inverse Polynomial	7.95	0.8457	9.71	0.4157	1.2614	15
		Exponential	6.01	0.6394	6.15	0.2633	0.9027	6
		Power	9.40	1.0000	23.36	1.0000	2.0000	16
		Delayed S-shape	6.25	0.6649	10.90	0.4666	1.1315	13

일반적으로 많이 사용되고 있는 신경망 모델, 통계적 모델들과 제안된 slCasCov-all과 slCasCov-new 모델을 비교한 결과 제안된 모델이 소프트웨어 신뢰도 예측에서 보다 좋은 성능을 나타내었다. 즉, slCasCov-all의 Rank R_m (0.0275)이 14위인 신경망 모델인 Jordan(R1)의 R_m (1.1583)에 비해 42.15배 이상의 성능 향상을 보였다. 또한, 통계적 모델인 Power 모델의 R_m (2.0000)에 비해 72.73배의 성능이 향상되었다.

결론적으로, slCasCov-all이나 slCasCov-new 모델 모두 모델 상호간에는 유사한 예측력을 가지므로 신뢰도 예측모델로 모두 적용할 수 있다. 또한, 다른 신경망이나 통계적 SRGMs 모델 보다 좋은 예측 결과를 나타내는 이유는 다음과 같다.

(1) 시그모이드-시그모이드 작동함수를 사용한 ssCasCov-all 모델에 비해 slCasCov-all 모델은 출력층에 선형을 사용하여 시그모이드-선형 형태를 취함으로써, 후보뉴런을 훈련시킬 때 적용하는 "공분산 최대화"의 문제점인 상관계수가 클 경우에도 두 함수간에 선형관계가 작을 수 있는 문제점을 보완할 수 있다.

(2) 제안된 slCasCov-new는 CasCor 알고리즘의 문제점

인 후보뉴런을 추가한 후 기존에 훈련이 종료되어 이미 각각의 역할을 수행하고 있는 출력층 연결 강도들을 다시 훈련시킴으로써 함수에 적합한 모델을 찾지 못하는 문제점을 제거하기 위해, 훈련을 종료하여 기존의 출력층에 연결된 연결 강도는 고정시키고, 새로 추가되는 후보뉴런과 관련된 연결강도만을 훈련시킴으로써, 원하는 함수에 적합한 모델을 보다 쉽게 찾을 수 있다.

6. 결론 및 향후 과제

본 논문에서는 가변 시간간격을 가진 그룹 고장 데이타에 대한 소프트웨어 신뢰도 예측을 하기 위한 일반화된 신경망 모델을 연구하였다. CasCor 알고리즘을 변형시킨 slCasCov-all과 slCasCov-new 모델을 제시하고, 그룹 데이타에 적합한 가변시간 간격 정보를 입력으로 하는 훈련제도 (3)과 (4)를 이용하였다. 본 연구에는 4개의 데이타가 사용되었으며, 모델의 예측력을 평가하기 위해 다음 단계 고장 수에 대한 AE를 계산하고, NAE와 순위를 계산하여 다양한 모델의 결과와 비교함으로서 제안된 모델의 적합성을 평가하였다. 실험결과 제안된 slCasCov-all과 slCasCov-new 모델 모두 잘 알려진 신경망과 통계적 모델들 보다 좋은 성능 향상을 나타내었다.

본 연구결과를 기반으로 하여, 제안된 slCasCov-all과 slCasCov-new 모델을 일정한 시간 간격을 가지는 고장 수나 고장시간 데이타에도 적용하여 소프트웨어 신뢰도 예측 분야에 일반적으로 적용 가능한 모델로 적용이 가능할 것이다. 따라서, 차후에는 이 분야에 대한 연구를 수행할 예정이다.

참 고 문 헌

[1] M. R. Lyu, "*Handbook of Software Reliability Engineering*," IEEE Computer Society Press, 1996.

[2] F. Popentiu and D. N. Boros, "Software Reliability Growth Supermodels," Microelectron. Reliab. Vol.36, No.4, pp.485-491, 1996.

[3] A. L. Goel, "Software Reliability Models Assumptions, Limitation, and Applicability," IEEE Trans. on Software Eng. Vol.SE-11, No.12, pp.1411-1423, 1985.

[4] J. D. Musa, A. Iannino, and K. Okumoto, "*Software Reliability : Measurement, Prediction, Application*," McGraw-Hill, 1987.

[5] N. Karumanithi, D. Whitley, and Y. K. Malaiya, "Prediction of Software Reliability Using Connectionist Models," IEEE Trans. on Software Eng., Vol.18, No.7, pp.563-574, July. 1992.

[6] S. E. Fahlman and C. Lebiere, "The Cascade-Correlation Learning Architecture," Advances in Neural Information Processing Systems II, pp.525-532, 1990.

[7] N. Karunanithi, D. Whitley and Y. K. Malaiya, "Using Neural Networks in Reliability Prediction," IEEE Software., pp. 53-59, 1992.

[8] M. Ohba, "Software Reliability Analysis Models," IBM Journal of Research and Development, Vol.21, No.4, pp.428-443, 1984.

[9] 이상운, 박영목, 박수진, 박재홍, "그룹 고장데이타의 소프트웨어 신뢰성 예측에 관한 신경망 모델", 한국정보처리학회 논문지, 제7권 제12호, pp.3821-3828, 2000.

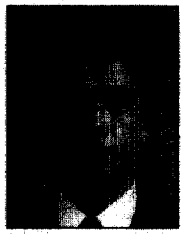
[10] G. Cybenko, "Approximation by Super-positions of A Sigmoidal Function," Mathematics of Control, Signals and Systems, Vol.2, pp.303-314, 1989.

[11] A. R. Barron, "Neural Net Approximation," In Proceedings of the Seventh Yale Workshop on Adaptive and Learning Systems. New Haven, CT. Yale University, pp.69-72. 1992.

[12] 이상운, "비정규화 데이타를 이용한 신경망 소프트웨어 신뢰성 예측", 한국정보처리학회논문지, 제7권 제5호, pp.1419-1426, 2000.

[13] T-Y. Kwok and D-Y. Yeung, "Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems," IEEE Trans. on Neural Networks, Vol.8, No.3, pp.630-645, 1997.

[14] Y. K. Malaiya, N. Karunanithi, and P. Verma, "Predictability Measures for Software Reliability Models," IEEE Trans. on Reliability, Vol.41, No.4, pp.539-546, 1992.



이 상 운

e-mail : sangun_lee@hanmail.net

1983년~1987년 한국항공대학교 항공전자공학과(학사)

1995년~1997년 경상대학교 컴퓨터과학과(석사)

1998년~2001년 경상대학교 컴퓨터과학과(박사)

1992년~현재 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당

관심분야 : 소프트웨어 공학(소프트웨어 시험 및 품질보증, 소프트웨어 신뢰성), 소프트웨어 메트릭스, 신경망, 뉴로-퍼지

박 중 양

e-mail : parkjy@nongae.gsnu.ac.kr

1982년 연세대학교 상경대학 응용통계학과(학사)

1984년 한국과학기술원 산업공학과 응용통계전공(석사)

1994년 한국과학기술원 산업공학과 응용통계전공(박사)

1984년~1989년 경상대학교 전산통계학과 교수

1989년~현재 경상대학교 통계학과 교수

관심분야 : 소프트웨어 신뢰성, 선형 통계 모형, 신경망