

딥뉴럴네트워크 기반의 네트워크 침입탐지시스템 설계

A Design of Deep Neural network-based
Network Intrusion Detection System

AM202126802
OH, ji hun



Index

1. 서론(introduction)
2. 관련연구(Related Research)
3. 문제정의(Problem definition)
4. 딥뉴럴네트워크를 이용한 침입탐지시스템
Deep Neural network-based Network Intrusion Detection System
5. 실험환경 및 실험결과(Experimental Environment and Results)
6. 결론(Conclusion)



1. 서론(introduction)

정보통신기술 발달과 더불어 다양하고 복잡한 침입공격들이 발생.

Various and complex intrusion attacks occur along with the development of ICT

이에 따라, 침입공격들로부터 시스템을 안전하게 보호하고 방어하는 침입탐지 시스템의 중요성도 높아짐.

Accordingly, the importance of an intrusion detection system that safely protects and defends the system from intrusion attacks increases.

하지만 기존의 침입탐지시스템은 잘 알려진 파라미터를 기반으로 악의적인 공격자의 행동을 탐지하기 때문에 기존 공격에는 효과적이지만, 새로운 침입에 대하여 상대적으로 취약한 한계점이 있음.

However, existing intrusion detection systems are effective against existing attacks because they detect malicious attacker behavior based on well-known parameters, but have relatively weak limitations against new intrusion.

하지만 이상탐지를 이용한 기법은 정상적인 패턴 행동을 분석하고 이상한 변화가 적정수준을 초과 하였을 때, 이상징후로 침입을 탐지하므로 새로운 패턴 유형에 대해서는 효과적으로 탐지가 가능.

However, the technique using anomaly detection analyzes normal pattern behavior and detects intrusion with abnormal symptoms when abnormal changes exceed the appropriate level, enabling effective detection of new pattern types.



2. 관련연구(Related Research)

기존 침입탐지시스템은 공격 패턴에 대한 매칭을 이용하여 위협을 탐지하고 차단하는 시스템.

The existing intrusion detection system is a system that detects and blocks threats using matching on attack patterns.

룰 기반으로 탐지를 하기 때문에 false detection이 높은 편
detects based on rules, so the false detection is high

It

이러한 침입탐지시스템에 머신러닝 기술을 도입하여 서포트벡터 머신(support vector machine, SVM), 의사결정트리(decision tree, DT), 베이시안 분류(bayesian classification) 등을 도입
By introducing machine learning technology into these intrusion detection systems, support vector machine (SVM), decision tree (DT), Bayesian classification, etc. are introduced

하지만 최근 딥뉴럴네트워크를 이용하여 룰 기반이나 악의적인 공격 패턴을 분석하여 탐지하는 방법과 달리 이상징후에 위협과 관련된 대량의 데이터를 통해서 모델 스스로가 직접적인 관계성을 찾는 알고리즘이 개발됨.

However, unlike the recent method of analyzing and detecting rule-based or malicious attack patterns using deep neural networks, algorithms have been developed to find direct relationships among models through large amounts of data related to hazards in anomalies.



3. 문제정의(Problem definition)

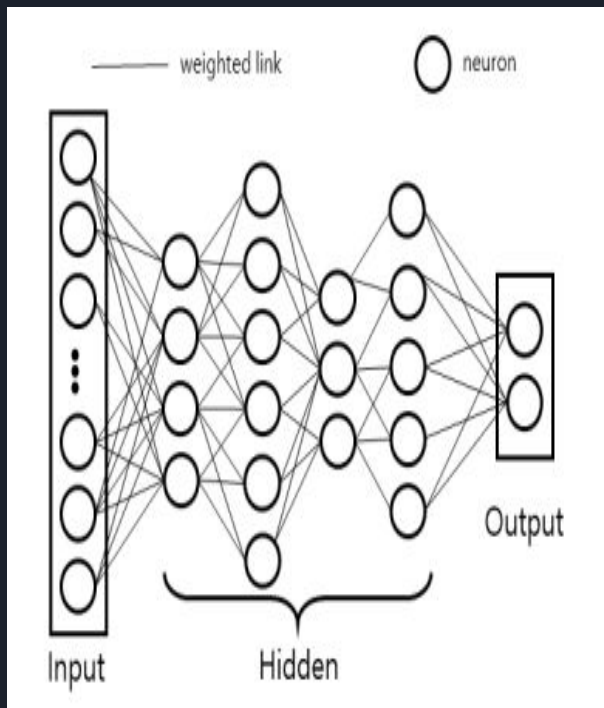
딥러닝 모델 중에 딥뉴럴네트워크를 이용한 방법은 다중 퍼셉트론을 이용하여 구성된 모델. The method of using the deep neural network among deep learning models is a model composed using multiple perceptrons

딥뉴럴네트워크에안에는 각 층 뿐만 아니라 각 층의 노드수, 활성화 함수, drop out, epoch 등에 다양한 파라미터가 존재함. In the deep neural network, various parameters exist not only in each layer, but also in the number of nodes, activation functions, dropout, epoch, etc. of each layer.

시스템 관리자 입장에서 어떠한 파라미터가 침입탐지시스템에영향을 주는지 알 필요가 있기에 여러가지 파라미터가 침입탐지시스템에어떠한 영향을 주는지 알아보고자 함. As a system administrator, we need to know which parameters affect the intrusion detection system, so we want to know how the various parameters affect the intrusion detection system.

4. 딥뉴럴네트워크를 이용한 침입탐지시스템

Deep Neural network-based Network Intrusion Detection System



딥뉴럴네트워크에 대한 시스템 구성은 입력층(input layer), 은닉층(hidden layer), 출력층(output layer)으로 구성되어 있음

The system configuration for the DNN consists of an input layer, a hidden layer, and an output layer

입력층에서 보면 각 입력값들은 노드에 매칭이 된다.

When viewed from the input layer, each input value is matched to the node.

은닉층(hidden layer)에서는 각 층의 수는 이 모델의 복잡성을 나타낸다.

In a hidden layer, the number of layers represents the complexity of this model.

출력층의 노드의 수는 예측하는 결과값의 유형들을 나타낸다. 은닉층에 있는 노드와 가중치의 결합을 통해서 출력층의 노드에 영향을 준다

The number of nodes in the output layer represents the types of predicted result values. It affects nodes in the output layer by combining weights with nodes in the hidden layer



4. 딥뉴럴네트워크를 이용한 침입탐지시스템

Deep Neural network-based Network Intrusion Detection System

딥뉴럴네트워크에서구성되는 파라미터와 구성 요소들은

- 활성화 함수(activation function)
- 드랍 아웃(Dropout)
- 배치 사이즈(batch size)
- 반복횟 수(epoch)
- 각 층의 노드수(Number of node)
- 은닉층의 층개수(Number of hidden layers)



4. 딥뉴럴네트워크를 이용한 침입탐지시스템

Deep Neural network-based Network Intrusion Detection System

딥뉴럴네트워크에서구성되는 파라미터와 구성 요소들은

- 활성화 함수(activation function)
step funtion, **sigmoid function**, **ReLU function**
- 드랍 아웃(Dropout)
- 배치 사이즈(batch size)
- 반복횟 수(epoch)
- 각 층의 노드수(Number of node)
- 은닉층의 층개수(Number of hidden layers)



4. 딥뉴럴네트워크를 이용한 침입탐지시스템

Deep Neural network-based Network Intrusion Detection System


딥뉴럴네트워크에서구성되는 파라미터와 구성 요소들은

- 활성화 함수(activation function)
- 드랍 아웃(Dropout)

오버피팅(Over-fitting)을 막기위한 방법으로, 딥뉴럴네트워크가 학습 중일 때, 랜덤하게 특정노드가 학습하는 것을 방해함으로써 학습이 특정 데이터에 치중되는 현상을 막아줌

As a way to prevent overfitting, learning is prevented from focusing on specific data by randomly preventing certain nodes from learning when deep neural networks are learning

- 배치 사이즈(batch size)
- 반복횟 수(epoch)
- 각 층의 노드수(Number of node)
- 은닉층의 층개수(Number of hidden layers)



5. 실험환경 및 실험결과 (Experimental Environment and Results)

텐서플로우(Tensorflow) 머신러닝 라이브러리를 이용

5. 실험환경 및 실험결과 (Experimental Environment and Results)

실험 데이터셋. KDD CUP99

- 침입탐지시스템을 평가하기 위하여 KDD CUP 99는 사용되어 왔으며 41개의 feature로 구성
KDD CUP 99 has been used to evaluate intrusion detection systems and consists of 41 features
- 공격으로 시뮬레이터된 유형들은 DoS 공격, U2R 공격, R2L 공격, Probing 공격으로 크게 구분된다. The
types simulated as attacks are largely divided into DoS attacks, U2R attacks, R2L attacks, and Probing attacks.
- DoS은 호스트에 오버헤드가 걸리도록 많은 양의 데이터를 제공하여 정상적인 서비스 제공을 마비시키는 공격 DoS
attacks that paralyze normal service delivery by providing a large amount of data for hosts to be overhead
- U2R 공격은 사용자의 기존 접근을 통해서 root 권한까지 확장하는 공격 방법이다.
U2R attack is an attack method that extends the user's existing access to root authority.
- R2L 공격은 권한 없는 사용자가 외부에서 접근 권한을 얻으려고 패킷을 보내는 공격 방법을 의미한다.
R2L attack refers to an attack method in which an unauthorized user sends packets to obtain access from the outside.
- Probing 공격은 실제 공격을 하기 전에 시스템의 사전 포트 정보 등을 수집하는 패킷 공격방법을 의미 A
probing attack refers to a packet attack method that collects the system's pre-port information, etc. before making an actual attack

5. 실험환경 및 실험결과 (Experimental Environment and Results)

각 층의 제 원	수치값
Fully connected layer	1024
Fully connected layer	768
Fully connected layer	512
Fully connected layer	256
Fully connected layer	128
Fully connected layer	64
Fully connected layer	32
Fully connected layer	1
Fully connected layer	ReLU / Sigmoid function

제 원	종류 및 수치값
Optimizer	Adam
Learning rate	0.01
Dropout	0.01 / 0.05 / 0.1
Batch size	64
Epochs	10~100

각 파라미터에 의하여 딥뉴럴네트워크에 대한 성능을 측정하기 위하여 침입을 탐지하는 DNN의 구성은 오른쪽과 같이 각 파라미터와 구조에 따라서 각 탐지 성능을 측정하였다.

In order to measure the performance of the DNN by each parameter, the configuration of the DNN detecting intrusion measured each detection performance according to each parameter and structure as shown on the right.

딥 뉴럴네트워크의 구조에서 각 은닉층의 개수를 DNN 1개 layer에서부터 7개 layer까지 각각의 성능을 테스트한다.

In the structure of the DNN, the number of each hidden layer is tested from one DNN layer to seven layers.

또한, 활성화 함수를 유형 2가지 를 각각 적용한 성능 값, 드랍아웃, Epoch을 다르게 하여 침입탐지시스템의 성능을 비교하였다.

In addition, the performance of the intrusion detection system was compared by varying the performance values, dropouts, and epochs to which the activation function was applied, respectively.

5. 실험환경 및 실험결과 (Experimental Environment and Results)

○성능측정(Performance measurement)

침입탐지시스템에 대한 성능측정은 다음과 같은 요소를 통해서 측정하였다.

The performance measurement of the intrusion detection system was measured through the following factors.

정확도(Accuracy)는 전체 레코드 중에서 정확히 분류한 레코드로 확률 값을 의미한다.
Accuracy refers to a probability value as a record accurately classified among all records.

정밀도(Precision)는 TP(True positive)와 FP(False positive)합의 TP의 백분율로 정의가 된다.

Precision is defined as a percentage of TP of the sum of true positive (TP) and false positive (FP).

재현율(Recall,)은 TP와 FN(False negative)의 합의 TP의 백분율로 정의 된다.

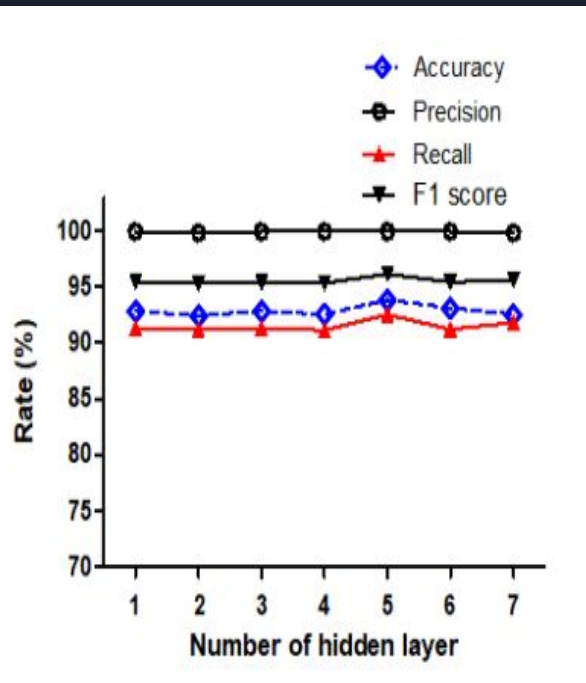
The reproduction rate (Recall) is defined as a percentage of TP that is the sum of TP and false negative (FN).

F1-score는 정밀도(P)와 재현율의 조화평균(R)을 의미한다.

F1-score refers to the harmonic average R of the accuracy P and the reproduction rate.



5. 실험환경 및 실험결과 (Experimental Environment and Results)



은닉층의 층개수에 따른 딥뉴럴네트워크에 대한 성능을 분석한 결과이다. This is the result of analyzing the performance of the deep neural network according to the number of layers of the hidden layer.

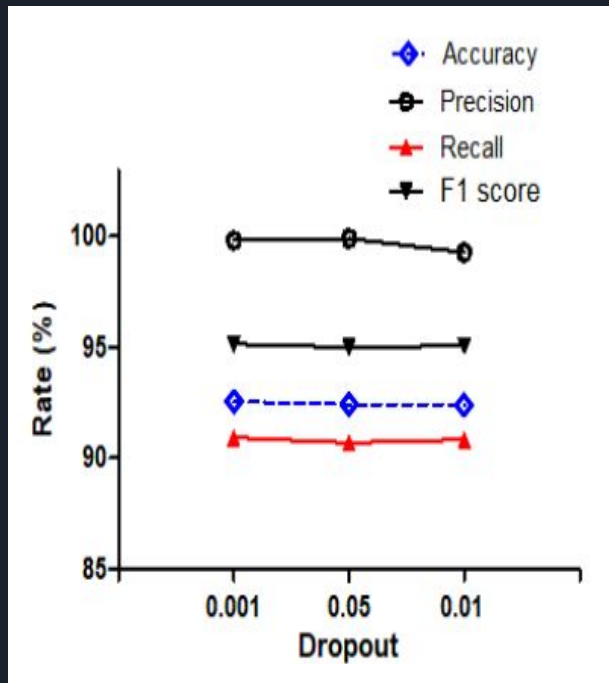
결과를 보면 은닉층의 층개수가 많아지더라도 성능의 개선이 이뤄지지 않은 것을 볼 수 있다.

The results show that even if the number of hidden layers increases, performance has not been improved.

이 실험에서 은닉층의 개수가 5개(5-DNN)일 때 93.1% 정확도의 성능으로 가장 좋은 것을 볼 수 있다

In this experiment, when the number of hidden layers is 5 (5-DNN), the best performance can be seen with 93.1% accuracy

5. 실험환경 및 실험결과 (Experimental Environment and Results)



드랍 아웃(Dropout)에 따른 딥뉴럴네트워크의 성능을 보여준다.
It shows the performance of deep neural network according to dropout

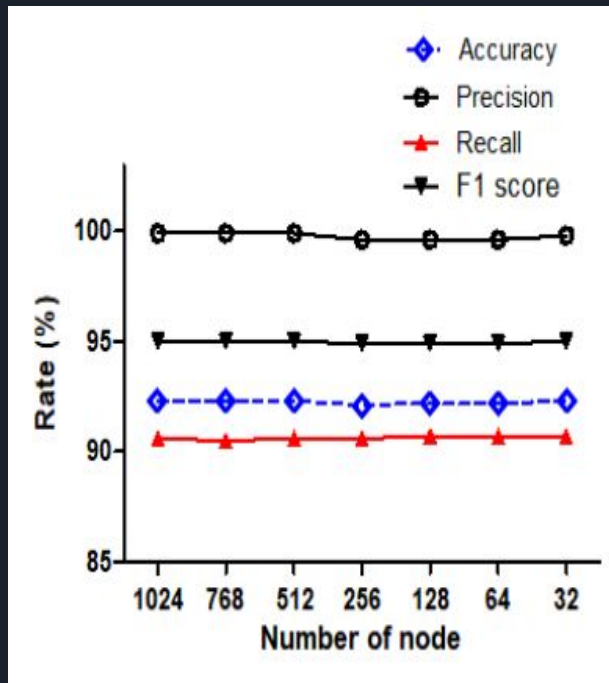
이 때 딥뉴럴네트워크는 은닉층 1개부터 7개(1-DNN~7-DNN)까지의 전체 평균한 값을 보여준다.
In this

case, the deep neural network shows the overall average value from one to seven (1-DNN to 7-DNN).

이 그림에서 보면 드랍아웃의 확률값이 증가할수록 값 차이는 크지 않지만 약간씩 딥뉴럴 네트워크의 성능이 오히려 떨어지는 것을 볼 수 있었다.
In this

figure, it can be seen that as the probability value of dropout increases, the difference in value is not large, but the performance of the deep neural network decreases slightly.

5. 실험환경 및 실험결과 (Experimental Environment and Results)



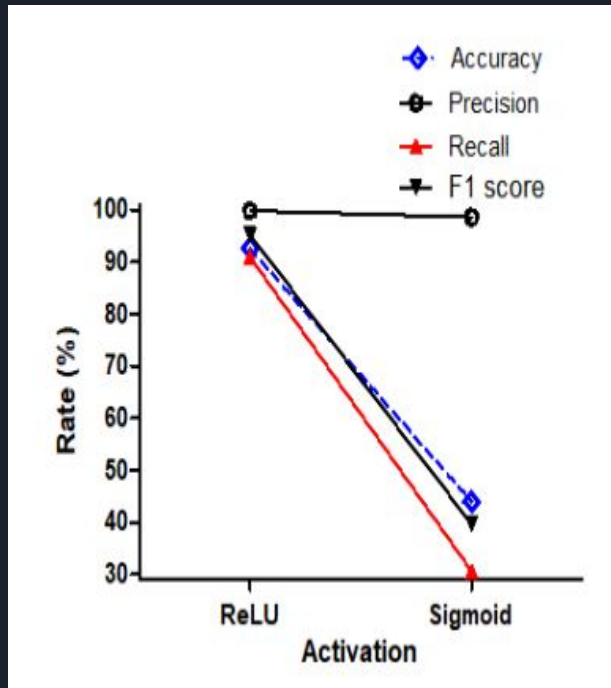
은닉층 개수가 1개인 딥뉴럴네트워크(1-DNN)에 노드수에 따른 성능 분석한 결과이다.

This is the result of performance analysis according to the number of nodes in a deep neural network (1-DNN) with one hidden layer.

결과를 보면 노드수가 1024, 768, 512 일 때, 다른 노드보다 약 1~2% 정확도가 높은 것을 볼 수 있었지만 노드 수의 상관없이 거의 성능이 비슷한 것을 볼 수 있었다.

The results showed that when the number of nodes was 1024, 768, 512, it was about 1-2% more accurate than other nodes, but the performance was almost similar regardless of the number of nodes.

5. 실험환경 및 실험결과 (Experimental Environment and Results)



활성화 함수에 따른 DNN의 성능을 분석한 결과이다.

This is the result of analyzing the performance of the deep neural network according to the activation function.

이 때, 딥뉴럴네트워크는 은닉층 1개부터 7개(1-DNN~7-DNN)까지 전체 평균값을 보여준다.

In this case, the deep neural network shows an overall average value from one to seven hidden layers (1-DNN to 7-DNN).

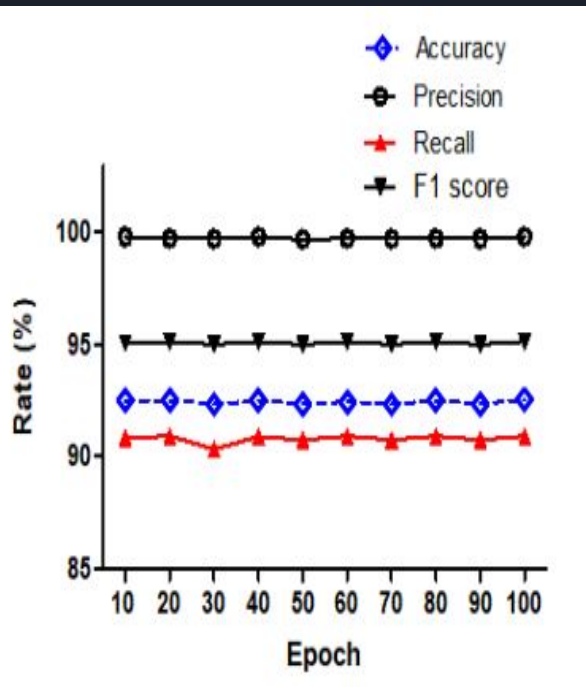
결과를 보면 ReLU 함수를 사용할 때 성능이 더 좋은 것을 볼 수 있었다.

The results showed that the performance was better when using the ReLU function.

이는 gradient vanish 현상으로 인해서 sigmoid 함수가 취약점이 있지만 ReLU는 특정 임계치 이상부터는 1차원 함수로 증가하기 때문에 gradient vanish 현상을 막아 노드가 잘 학습하는 것을 볼 수 있었다.

Although the sigmoid function is vulnerable due to gradient vanish phenomenon, ReLU increases as a one-dimensional function from above a certain threshold, so it can be seen that nodes learn well by preventing gradient vanish phenomenon.

5. 실험환경 및 실험결과 (Experimental Environment and Results)



Epoch 수에 따른 딥뉴럴네트워크의 성능을 분석한 결과이다.

Epoch It analyzed the performance of the network a result that the number.

이 때, 딥뉴럴네트워크는 은닉층 1개부터 7개(1-DNN~7-DNN)까지 전 체 평균값을 보여준다. Epoch 수가 10 이상일 때, 거의 성능이 비슷하게 유지되는 것을 볼 수 있었다.

In this case, the deep neural network shows an overall average value from one to seven hidden layers (1-DNN to 7-DNN). When the number of epochs was 10 or more, it could be seen that the performance remained almost similar.

5. 실험환경 및 실험결과 (Experimental Environment and Results)

제 원	Accuracy	Precision	Recall	F1 score
1-DNN	92.80%	99.90%	91.30%	95.40%
2-DNN	92.40%	99.80%	91.20%	95.30%
3-DNN	92.80%	99.90%	91.30%	95.40%
4-DNN	92.50%	99.90%	91.10%	95.30%
5-DNN	93.80%	99.90%	92.50%	96.10%
6-DNN	93.10%	99.90%	91.20%	95.40%
7-DNN	92.50%	99.80%	91.80%	95.60%
LR	84.90%	98.90%	82.10%	89.70%
NB	92.90%	98.90%	92.40%	95.50%
KNN	92.90%	99.80%	91.50%	95.50%
DT	92.90%	99.90%	91.30%	95.40%
Adaboost	92.50%	99.60%	91.40%	95.30%
RF	92.60%	99.90%	91.10%	95.30%

다른 머신러닝 방법과의 성능비교

Performance Comparison with Other Machine Learning Methods

다른 머신러닝기법도 약 92.5~93.8%사이의 성능을 보여줌 Other machine learning techniques also showed performance between 92.5 and 93.8%.

딥뉴럴네트워크에서는5-DNN 일 때 정확도 93.8%와 F1 score가 96.1%으로 가장 성능이 좋은 것을 볼 수 있었다. In the deep neural network, the accuracy of 93.8% and F1 score of 5DNN were 96.1%, indicating the best performance.

LR = Lear Regression

NB = Naive baysian

KNN = K-Nearest Neighbors

DT = Decision Tree

Adaboost = Adaptive Boosting

RF = Random Forest



6. 결론

침입탐지를 위한 DNN을 구성할 때, DNN에 대한 파라미터에 따른 성능 변화에 대해서 분석할 필요성이 있다. 왜냐하면 잘못된 파라미터 설정으로 시스템의 성능에 영향을 미칠 수 있기 때문이다.

When configuring a deep neural network for intrusion detection, it is necessary to analyze the performance change according to the parameters for the deep neural network. This is because incorrect parameter settings can affect the performance of the system.

DNN에 있는 파라미터를 조정하여 성능을 분석한 것을 보면, 상대적으로 은닉층의 개수, 각 층의 노드의 수, Dropout에 따른 성능이 거의 유사한 것을 볼 수 있었다. 또한 Epoch 수도 10이상 일 경우에는 비슷한 성능으로 10 epoch이면 거의 최적화된 파라미터로 설정된 것을 볼 수 있었다

When analyzing the performance by adjusting the parameters in the deep neural network, it can be seen that the number of hidden layers, the number of nodes in each layer, and the performance according to Dropout are relatively similar. In addition, when the number of epochs is 10 or more, it can be seen that with similar performance, 10 epochs are almost optimized parameters

반면에 활성화함수 선정은 중요한 것을 볼 수 있었다. LeRU를 사용하는 대신 Sigmoid function을 사용하게 되면 딥뉴럴네트워크의 성능이 떨어지는 것을 볼 수 있었다. gradient vanish 현상을 줄이는 활성화 함수의 선정이 중요한 것을 볼 수 있었다.

On the other hand, it can be seen that the selection of the activation function is important. It could be seen that the performance of the DNN was poor if the Sigmoid function was used instead of the LeRU. It was found that it was important to select an activation function that reduces the gradient vanish phenomenon.



7. 참조

딥뉴럴네트워크기반의 네트워크 침입시스템 설계(한국차세대컴퓨팅학회논문지) - 투고일 2020.02.01

A Design of Deep Neural network-based Network Intrusion Detection System

(The Journal of KINGComputing)