UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS MATEMÁTICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

TRABAJO DE FIN DE GRADO

Algoritmos de Aprendizaje Automático aplicados a problemas de Ciberseguridad

Presentado por: Pablo Jiménez Poyatos

Dirigido por: Luis Fernando Llana Diaz

Grado en Matemáticas

Curso académico 2023-24

Agradecimientos

Resumen

Palabras clave:

Abstract

Keywords:

Índice general

Capítulo 1

Introducción

1.1. Motivación y objetivos del trabajo

TensorFlow is a more complex library for distributed numerical computation. It makes it possible to train and run very large neural networks efficiently by distributing the computations across potentially hundreds of multi-GPU (graphics processing unit) servers. TensorFlow (TF) was created at Google and supports many of its large-scale machine learning applications. It was open sourced in November 2015, and version 2.0 was released in September 2019.

Keras is a high-level deep learning API that makes it very simple to train and run neural networks. Keras comes bundled with TensorFlow, and it relies on TensorFlow for all the intensive computations.

1.2. Contexto y antecedentes del trabajo

1.2.1. Movilidad Aérea Urbana

1.2.2. Vertipuertos

1.2.3. Optimización del diseño de un vertipuerto

1.2.4. Optimización heurística

1.3. Estructura de la memoria

1.4. Contribuciones

Capítulo 2

Fundamentos de las redes neuronales

2.1. Revisión teórica

Puedo introducir los tipos de funciones de activavion. Esta bien explicado en el TFG wuolah o en el articulo de KDD cup 199 de DNN network intrusion. Puedo añadir overfitting y underfitting. lo que es aprendizaje supervisado y no supervisao Partes de una neurona y como trabaja(bias, pesos...)

2.2. Arquitecturas relevantes

Mini tabla resumen en Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective y miniresumen de todos los tipos en review Deep Cybersecurity: A Comprehensive Overview from Neural Networkand Deep Learning Perspective y review

2.2.1. Autoencoder

Leer y sacar la información del word autoencoders.

Los autoencoders son una clase de redes neuronales artificiales utilizadas en aprendizaje no supervisado para aprender representaciones eficientes de datos. Su objetivo principal es codificar la entrada en una representación comprimida y significativa, y luego decodificarla de manera que la reconstrucción sea lo más similar posible a la entrada original.

La arquitectura básica de un autoencoder consta de dos partes: el encoder y el decoder. El encoder mapea los datos de entrada a una representación oculta de menor dimensión utilizando funciones principalmente no lineales, mientras que el decoder reconstruye los datos de entrada a partir de esta representación oculta. Durante el entrenamiento, los parámetros del autoencoder se optimizan para minimizar la diferencia entre la entrada y la salida reconstruida, utilizando una función de pérdida que mide esta discrepancia.

Los autoencoders se han utilizado en una amplia variedad de aplicaciones, incluida la reducción de dimensionalidad, la extracción de características, la eliminación de ruido en los datos de entrada y la detección de anomalías. Su versatilidad y capacidad para aprender representaciones útiles de los datos los hacen herramientas poderosas en el campo del aprendizaje automático y la inteligencia artificial.

2.2.2. Deep Belief Networks

Red Neuronal Profunda

Restricted Boltzmann Machine

2.2.3. Red Neuronal Convolucional

2.2.4. Red Neuronal Recurrente

2.3. Paquetes de python

2.3.1. Keras

Capítulo 3

Aplicación en la ciberseguridad

3.1. Clasificación de Malware

3.1.1. Microsoft Malware Classification Challenge

3.1.2. Autoencoder

3.1.3. Red Neuronal Convolucional

3.1.4. Resultados

Poner esto :This experiment is executed on MSI GF75 Thin 9SD laptop which has Intel Core i7-9750H CPU @ 2.60 GHz, 16 GB memory without using GPU

3.2. Detección de intrusiones

3.2.1. KDD Cup 1999

3.2.2. Autoencoder

Para la clasificacion binaria usar autoencoder con el entrenamiento de las imágenes (buenas o malas) y según el error que den, se clasifica. Para la multiclasificación, tenemos dos opciones:

- Usamos autoencoder para comprimir la información de entrada y despues esa informacion la usamos para clasificarla usando una DNN [7]
- Usamos una cadena de autoencoders en el cual la salida de h es la entrada del autoencoder h+1. Utilizo el articulo [4] donde se desarrolla todo el modelo y explicacion

y ademas se hace referencia al artículo [2] porque se basa en él (lo de salida de h es la entrada de h+1). Ver tambien:

- Asymmetric Stacked Autoencoder
- Constrained Nonlinear Control Allocation based on Deep Auto-Encoder Neural Networks.

El algoritmo consiste en entrenar las capas por separado en la que el input del autoencoder es la salida del autoencoder anterior. Lo que de verdad nos interesa es la capa oculta, que tiene una representación comprimida de los datos de entrada y sus pesos. Estos pesos son con los que se inicializa el entrenamiento de la stacked autoencoder acabando en softmax. He usado el url para enterlo `https://amiralavi.com/tied-autoencoders/`. Además en [1] explica bastante bien la diferencia entre capa autoencoder y un autoencoder.

3.2.3. Red Neuronal Convolucional

Para clasificar los datos del dataset KDD 1999 usando las CNN (Convolutional Neural Network) vamos a seguir los siguientes articulos [5, 11, 9, 6]. Prácticamente todo el cuerpo del experimento se encuentra en el artículo [5], pero en el artículo [6] aparece la parte de normalización de los datos y algunos hiperparametros de inicio.

3.2.4. Red Neuronal Profunda

Por otro lado, el método DNN (Deep Neural Network) utiliza una arquitectura muy parecida a una CNN. Podemos ver todo el procesamiento de los datos y el modelo en el artículo [8]. Además, hay buena explicacion del experimento en [10]. Por último, en el articulo [3] están los experimentos con DNN, RNN, RBM que puedo tomar también como referencia porque está muy bien explicado las capas e hiperparametros que utiliza.

3.2.5. Red Neuronal Recurrente

En el articulo [3] están los experimentos con DNN, RNN, RBM que puedo tomar también como referencia porque está muy bien explicado las capas e hiperparametros que utiliza.

3.2.6. Restricted Boltzmann Machine

En el articulo [3] están los experimentos con DNN, RNN, RBM que puedo tomar también como referencia porque está muy bien explicado las capas e hiperparametros que utiliza.

3.2.7. Resultados

Capítulo 4

Conclusiones y Trabajo Futuro

4.1. Conclusiones

4.2. Trabajo futuro

Bibliografía

[1] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944, 2017.

[2] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006.

[3] Wisam Elmasry, Akhan Akbulut, and Abdul Halim Zaim. Empirical study on multiclass classification-based network intrusion detection. *Computational Intelligence*, 35(4):919–954, 2019.

[4] Fahimeh Farahnakian and Jukka Heikkonen. A deep auto-encoder based approach for intrusion detection system. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pages 178–183. IEEE, 2018.

[5] Jiyeon Kim, Jiwon Kim, Hyunjung Kim, Minsun Shim, and Eunjung Choi. Cnn-based network intrusion detection against denial-of-service attacks. *Electronics*, 9(6):916, 2020.

[6] Taejoon Kim, Sang C Suh, Hyunjoo Kim, Jonghyun Kim, and Jinoh Kim. An encoding technique for cnn-based network anomaly detection. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2960–2965. IEEE, 2018.

[7] Ivandro O Lopes, Deqing Zou, Ihsan H Abdulqadder, Francis A Ruambo, Bin Yuan, and Hai Jin. Effective network intrusion detection via representation learning: A denoising autoencoder approach. *Computer Communications*, 194:55–65, 2022.

[8] Mohammed Maithem and Ghadaa A Al-Sultany. Network intrusion detection system using deep neural networks. In *Journal of Physics: Conference Series*, volume 1804, page 012138. IOP Publishing, 2021.

[9] Sinh-Ngoc Nguyen, Van-Quyet Nguyen, Jintae Choi, and Kyungbaek Kim. Design and implementation of intrusion detection system using convolutional neural network for dos detection. In *Proceedings of the 2nd international conference on machine learning and soft computing*, pages 34–38, 2018.

[10] Rahul K Vigneswaran, R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security. In *2018 9th International conference on computing, communication and networking technologies (ICCCNT)*, pages 1–6. IEEE, 2018.

[11] Zhongxue Yang and Adem Karahoca. An anomaly intrusion detection approach using cellular neural networks. In *Computer and Information Sciences–ISCIS 2006: 21th International Symposium, Istanbul, Turkey, November 1-3, 2006. Proceedings 21*, pages 908–917. Springer, 2006.