Paper Critique - 4

Intro of Data Mining - Fall, 2021

Student

Aissata Diallo (Undergraduate) & Sheng Kai Liao (Graduate)

Date:11/18/2021

Paper:

On the combined effect of class imbalance and concept complexity in deep learning

Critique: (Eddy)

Deep learning, as a sensitive model for attribute noise, is easily harmed by the data of structural concept complexity, class imbalance, class overlap and data scarcity. Looking into how those situations impact a deep learning model with different parameters setting is fascinating and useful as a reference for the deep learning community to deal with those cases in the real-world. In this paper, Japkowicz et. al provide a detailed report about how the different depth of deep learning models performance in those situations above.

The paper mainly addresses three problems: RQ1) How are the deep learning models with different depth settings performance between different numbers and mixing level of class subconcepts, subconcepts' size and training size. RQ2) How are the deep learning models with different depth settings performance in class overlapping data on the class imbalance problem. RQ3) How are the deep learning models with different depth settings performance with the cases combined with RQ1's and RQ2's difficulties. In this paper, they use MLP model and CNN model with two MLP layers as the major testing model. For the dataset, by using the gaussian random function, authors are able to create datasets which have different structural concept complexity, class imbalance level, class overlapping level and data scarcity. In addition, they also use the most popular image datasets as real-world cases, such as MNIST Fashion and CIFAR-10.

For the RQ1, authors test the MLP models from 1 hidden layer to 5 hidden layers with different datasets which are generated with different size, subconcepts and class imbalance lever. According to the observation, while the depth of the MLP network increases, it does mitigate the harmness from structural complexity and the class imbalance problem in easy cases, yet can't help out with more complex and imbalanced domains. The situation is also adopted in bigger size datasets. For the RQ2, authors test the same MLP models with datasets with 10 class overlapping levels. Level one is the densest class overlapping cases and level ten is the lightest. By their observation, the performance of all models are not very different regarding different overlapping level datasets and it turns out that the result is not really meaningful. For the RQ3, they keep the datasets complexity and size as constant but only modify the class balance level and the overlapping lever. According to their observation, they find out the interesting part is that while the depth of the deep learning model increases, it does ease the impact from the class imbalance and overlapping problem but once the depth is 5 (deepest case), the performance goes down a little bit. For the RQ3, authors pick several class images datasets from MNIST Fashion and CIFAR-10 datasets separately and implement random undersampling to create class imbalance. In addition, for embedding analysis , authors trained the CNN models which have 1,3,5 depth of layers with two types of final layers, convolutional block (CNN) and fully connected layer (MLP). In order to evaluate the performance, authors extract the embedding from the trained model and showcase the distribution by using tSNE plotting. Based on their result, while the depth of CNN model increases, it sometimes did help to deal with class

imbalance but most of the time it didn't have significant improvement in both MNIST Fashion and CIFAR-10 datasets. Also, according to the tSNE plotting result, the models trained with class balance dataset have better seperatability than the models trained with class imbalance dataset. Yet, while the model with 5 CNN blocks, it deteriorates class overlapping.

In short, this paper provides a detailed analysis on how the depth of deep learning model increases affect the harmness by the data of structural concept complexity, class imbalance, class overlap and data scarcity. In my opinion, this paper could give a clear vision for deep learning developers to deal with those kinds of problems with real-world applications.

Paper:

A^3 : Activation Anomaly Analysis

Critique: (Eddy)

Anomaly detection is widely used to detect any anomaly data from the datasets which may harm our machine learning system. It has been applied in many AI or machine learning applications, such as face recognition, self driving, industry learning and so on. Following the time, due to the enhancement of hardware, deep learning has been widely implemented for more complex tasks and datasets. Yet, in order to achieve anomaly detection, the most challenging part for deep learning is the class imbalance. Therefore, according to Sperl et. al, the paper proposes a novel anomaly detection method which combines three neural networks in a purely data-driven model which is basically a semi-unsupervised model which is able to train with a lot of data with few labels.

As I just mentioned, their model contains three essential parts, target network, anomaly network and alarm network. The target network basically is trained with normal data which is used as an autoencoder for features extraction. The anomaly network will generate the anomaly samples based on the input form the target network and the result will be fed into the alarm network. The alarm network is trained by the normal or anomalous data by observing the hidden activation from the target function. During the training, the target network will firstly be trained by normal samples, at the same time the normal samples will be fed into the anomaly network in order to generate the anomalous samples. Afterward, for training the alarm network, the input (normal or anomalous) samples will be firstly fed into the target network in order to obtain its hidden activations, then be fed into the alarm network. In addition, to define the anomalous labels, the target network will define the unseen pattern as the anomaly data and label them as anomalous.

They experimented their model with the most popular dataset, such as MNIST, EMINST, NSL-KDD, Credit Card Transactions and CSE-CIC-IDS2018. In addition, they compared their model with several baselines which scale to the large dataset, such as Autoencoder Reconstruction Threshold, Isolation Forest, Deep Autoencoder Gaussian Mixture Model and Deviation Networks. For evaluation, authors plotted out the ROC curves for each model and evaluate the model performance by determining if the curve is close to the right top or not. Also, the AUC and the AP score are used as the trade-off between TPR and FPR and between precision and recall.

For the experiment, they tested the models in four scenarios based on three constraints of anomaly detection: scarcity of anomaly samples, variable extent of abnormality and driven by data, not expert knowledge. The testing scenarios are: detection of known anomalies, transferability of unknown anomalies, the generality of the Methods and the outlook of unsupervised anomaly detection. In their observations, their model has best of performance in the first three testing scenarios over all datasets. For the outlook of unsupervised anomaly detection, they implemented the VAE as an anomaly network in this case and compared their

models with other state-of-the-art unsupervised anomaly detection methods. Again, their approach reflects the best performance.

In short, this paper provides a novel deep learning based semi-unsupervised anomaly detection model which is more superior and stable than other unsupervised and semi-supervised approaches. Furthermore, they implemented VAE as the anomaly network which turns their model as an unsupervised learning model and it also achieved superior performance over the other relevant methods. In my opinion, this approach could be implemented in various machine learning domains in order to create a more stable, robust machine learning model.

Paper:

Learning Explainable Representations of Malware Behavior

Critique (Aissata)

Even though there is a ton of information out there on machine learning and its ability to detect malware there are also hindrances that make it difficult to deploy machine learning solutions because those in charge of such tasks such as computer security analysts need access to the network in order to validate, confirm , or even block software that is identified as malware.

Malware can take on different forms of malicious behavior in networks. It achieves this by collecting financial or personal data through encryption of users files for ransom. In return this allows fraudulent activities such as scams through intrusive advertising. In this paper Prasse et al. address the problem of identifying malware in network telemetry logs and provide indication on network vulnerabilities or compromise. They achieve this by developing a neural network that processes the sequence of events and identifies specific threats, such as “malware families” and broad categories of malware. They then use an integrated-gradients method to highlight events that jointly constitute the characteristic behavioral pattern of the threat. Furthermore, the authors compare network architectures based on CNNs, LSTMs, and transformers, and explore the efficacy of unsupervised pre-training experimentally on large-scale telemetry data. Lastly, they demonstrate how the system is able to detect njRAT and other malware based on behavioral patterns.

In the experiment the researchers used a malware taxonomy with three levels, which include the threat ID, malware family, and malware category. All of these levels are beneficial for the threat detection because the threat ID is capable of identifying a particular version of a malware product. For example, the threat ID is able to identify malware that is similar to njRAT malware because they all use the same user agent and URL pattern for communication. The malware family means all versions of the malware product. Third, the malware category has to do with the “monetization scheme” or how harmful in a range of least to most a product of malware is. In order to make this research also effective they collected data and quantitative analysis from a ton of companies for a day in June 2020 as training data and for a day in July 2020 as evaluation data. The entire network traffic of those companies had more than a thousand of data for the researchers to use for the training and evaluation. Then they proceeded to perform malware classification in which they observed that the transformer outperforms CNN and LSTM most of the time. Finally, they used the integrated gradients method for the purpose of indicating compromise in which for the single instance elevation they determined the input of time steps that had to do with the classification and proceeded to add the feature importance values for all the instances classified as a specific malware family.

Overall, after studying the issue of identifying threats based on the sequences of the sets and having an array of specialized detectors Prasse et al. found that the transformer architecture out performs both the CNN and LSTM models at identifying threat IDs, malware families, and malware categories. They also found that unsupervised pre-training improves the results of the transformer more than the supervised learning. My only critique for their research is that the whole research focused on the njRAT malware family which could be a disadvantage for future malware detection considering that malware and attacks are only going to keep advancing and getting stronger.