Methodology

1. Overview

Trong phần này mình sẽ giới thiệu các kiến trúc dynamic model dùng trong bài toán Class Incremental Learning. Đầu tiên mình sẽ nói qua về các kiến trúc mình dùng. Tuy nhiên các kiến trúc này được có nguồn gốc từ domain phân loại ảnh trong CIL. Nên mình sẽ trình bày về kiến trúc mình dùng đối với bài toán Network Intrusion Detection Problem.

1. DER Model
2. MEMO Model

DER model

* 1. The paper DER: Dynamic Expandable Representation for Class Incremental Learning introduces expandable representation. At step , the model is composed of a super feature extractor is build by expanding the feature extractor with a newly created extractor . Specifically, given an image , the feature extracted by is obtained by concatenation as follows

Here re reuse the previous and encourage the new extractor to learn only the novel aspect of new classes. The feature u is then fed into the classifier to make predictions as follows

= Softmax())

Then the prediction y = argmaxp\_h\_t(y|x). The classifier is designed to match its new input and output dimension for step t. The parameters of H\_t for the old features are inherited from H\_t-1 to retain old knowledge and its newly added parameters are randomly initialized.

To reduce catastrophic forgetting, we freeze the learned function at step t, as it captures the intrinsic structure of previous data. In detail, the parameters of last step super-feature extractor , and the statistics of Batch Normalization are not updated. Besides, we instantiate Ft with Ft-1 as initialization to reuse previous knowledge for fast adaptation and forward transfer.

MEMO Model: (Memory-Efficient Expandable Model)

The model from Paper A Model or 603 Exemplars: Toward Memory-Efficient Class-Incremental Learning try to solve the problem of expanding memory by answering the question: Given the same memory budget, if we share the generalized block and only extend specialized blocks for new tasks, can we further improve the performance?

Concretely, we redefine the model structure by decomposing the embedding module into specialized and generalized blocks. Specialized blocks correspond to the deep layers in the network, while generalized blocks corresponds to the rest shallow layers. We argue that the features of shallow layers can be shared across different incremental stages, i.e., there is no need to create an extra model.