Malware Identification Using Deep Learning

Problem Statement

Deep learning (also known as **deep structured learning** or **hierarchical learning**) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised. Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain. Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics and drug design where they have produced results comparable to and in some cases superior to human experts. In hierarchical Feature Learning, we extract will multiple layers of non-linear features and pass them to a classifier that combines all the features to make predictions. We are interested in stacking such very deep hierarchies of non-linear features because we cannot learn complex features from a few layers. It can be shown mathematically that for images the best features for a single layer are edges and blobs because they contain the most information that we can extract from a single non-linear transformation. To generate features that contain more information we cannot operate on the inputs directly, but we need to transform our first features (edges and blobs) again to get more complex features that contain more information to distinguish between classes.

Background

Most modern deep learning models are based on an artificial neural network, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in Deep Belief Networks and Deep Boltzmann Machines.

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level *on its own*. (Of course, this does not completely obviate the need for hand-tuning; for example, varying numbers of layers and layer sizes can provide different degrees of abstraction.

The "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial *credit assignment path* (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited. No universally

agreed upon threshold of depth divides shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth > 2. CAP of depth 2 has been shown to be a universal approximator in the sense that it can emulate any function. Beyond that more layers do not add to the function approximator ability of the network. The extra layers help in learning features.

Deep learning architectures are often constructed with a greedy layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out which features improve performance.

Methodology

Step1: take some data

Step2: train a model on that data

Step3: use the trained model to make predictions on new data.

The process of training a model can be seen as a learning process where the model is exposed to new, unfamiliar data step by step. At each step, the model makes predictions and gets feedback about how accurate its generated predictions were. This feedback, which is provided in terms of an error according to some measure (for example distance from the correct solution), is used to correct the errors made in prediction.

<u>Anomaly detection in network activities</u>

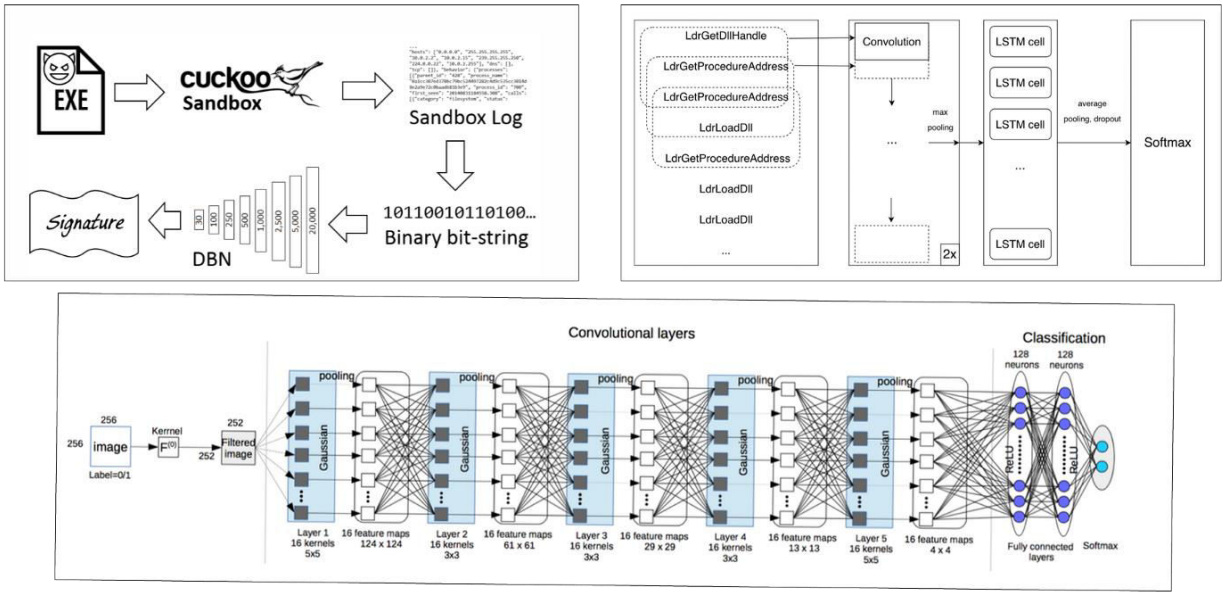

Fig 2: Architecture of Malware Detection using deep learning

Experimental Design

Dataset: All types of malwares.

Evaluation Measures: Measures such as accuracy and Mean Average Precision (MAP) will be computed by comparing the two different bounding boxes and ground truth boxes from the datasets.

Software and Hardware Requirements: Python based Computer Vision and Deep Learning libraries will be exploited for the development and experimentation of the project. Tools such as Anaconda Python, and libraries such as OpenCV, Tensorflow, and Keras will be utilized for this process. Training will be conducted on NVIDIA GPUs for training the end-to-end version of CNN based object detection model.