Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 104 (2017) 556 – 563

Procedia Computer Science

ICTE 2016, December 2016, Riga, Latvia

Classification Tree Extraction from Trained Artificial Neural Networks

Andrey Bondarenko[a,*], Ludmila Aleksejeva[a], Vilen Jumutc[b], Arkady Borisov[a†]

[a]*Riga Technical University, 1 Kalku Str., Riga, LV-1658, Latvia*
[b]*KU Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium*

Abstract

Recent advances in neural networks design and training provoked the 2nd artificial neural networks (ANN) renaissance. In many cases classification decision made by trained fully connected neural nets is better than that acquired by models like C4.5 or C5.0[1,2]. But in contrast to decision trees, ANN models are "black boxes", i.e., it is impossible to understand how classification decision is made. In many areas it is critical and even obligate to understand how a model performs classification thus rendering ANN usage as obsolete. Recently, some researchers have proposed and described separate steps that would allow extracting knowledge from a trained multi-layered fully connected sigmoidal neural network. This process involves several steps such as trained network training, pruning and knowledge extraction. This paper provides an overview of all the aforementioned steps, as well as describes how a knowledge extraction system can be built. We describe our Neural Network Knowledge eXtraction (NNKX) system and provide experimental results of rule extraction from the trained multi-layered feed-forward sigmoidal artificial neural network in the form of binary classification decision trees. The results obtained suggest that extracted decision trees have good classification accuracy and sizes comparable to C4.5 trees and even overcoming them in some cases. Thus the proposed system can be successfully applied to better understand and validate ANN models. We provide link to source code repository with the implementation of described system.

Peer-review under responsibility of organizing committee of the scientific committee of the international conference; ICTE 2016

Keywords: Classification decision tree; Knowledge extraction; Neural networks

* Corresponding author. Tel.: +371-29713956.
E-mail address: andrejs.bondarenko@gmail.com

Peer-review under responsibility of organizing committee of the scientific committee of the international conference; ICTE 2016
doi:10.1016/j.procs.2017.01.172

1. Introduction

Neural networks have shown themselves as very good classification and regression models. In the late 1990s they were contested by support vector machines (SVM) and Random Forests. In general, the performance of one model over the others was determined by the operator skill set. Although starting from the mid 2000s, with the rise of deep learning (DL) it became clear that deep neural networks are superior in cases with complex data that can be represented by such feature hierarchies. Deep learning allowed acquiring hierarchical features that are built on top of simpler features and they are built on top of even more simple features[3]. Along with deep belief networks that were built using several stacked on top of each other restricted Boltzmann machines, convolutional neural networks (CNN) firstly developed by Yan LeCun[4] gained huge attention after Alex Krizhevsky has won ImageNet completion with his CNN like architecture called AlexNet[5]. All this has led to the 2nd neural networks renaissance.

But neural networks themselves as well as SVMs are black box models, i.e., it is impossible to understand how classification or regression decision is being made. In many mission critical areas it is crucial to have such understanding. For example, it is required by law to provide specific criteria that person should meet before he will be eligible to get loan. In medicine and nuclear fields, domain experts should validate a model before it can be used in production.

This has led to efforts aimed at the extraction of knowledge that is incorporated in trained artificial neural networks. The current paper provides a brief overview of history of such efforts starting with ways of how knowledge can be represented followed by neural networks knowledge extraction in the form of crisp if-then rules or in the form of decision trees.

2. Knowledge extraction overview and related work

Let us first define what it means to extract knowledge from predictive model. Let us take the definition proposed in[6]:

> “Given an opaque predictive model and the data on which it was trained, produce a description of the predictive models hypothesis that is understandable yet closely approximates the predictive models behaviour“

There exist several rule types that can be used to represent knowledge. We will list them as per[6]:

- Propositional If-Then / If-Then-Else rules/Decision tree: These rules contain logical IF part condition which can hold several logical operands like conjunction, disjunction and negation. If part is followed by Then part which denotes class belongingness. In some cases default rule is used to denote default class to which all the points not covered by If statements, belong. If not, then Else part can be utilized to assign class to input data. Classification decision trees can be reduced to If-Then rule sets. Basically, such rules enable making classification decision by making data splits orthogonal to input data dimensions. An example of such rule is *If (a11 < x1 < a12) and (a21 < x2 < a22) Then Class-A Else Class-B*
- M-of-N rules: These rules resemble propositional If-Then rules, but they assume that for class decision to be made for specific point, it should be covered by several M regions (among full N regions set). Essentially they can represent knowledge in a more compact way than If-Then rules
- Oblique rules / Multi-surface method tree: These rules are defining planes that are used to separate space in order to define that a point belongs to specific class. Example: *If (a1*X1 + a2 * X2 + a3 > Z1) & (a4 * X3 + a5 * X6 > Z2) Then Class A*. In case of Multi-surface method tree space is subdivided by set of such planes essentially building a decision tree with splits not parallel to axes
- Equation rules: This rule type is very similar to Oblique rules, but used equations of complex separating hyper planes for example hyper spheres
- Fuzzy rules / Fuzzy decision trees: Fuzzy rules are somewhat similar to propositional If-Then rules, but they are operating in fuzzy set universe. Fuzzy logic is multi-valued. An example of fuzzy rule is: *If (Overcast is Cloudy) &(Temperature is Normal) Then Play Tennis*. Here definitions of *Cloudy* and *Normal* are provided in the fuzzy sets defining input attributes