GR-241 On Training Explainable Neurons

Abstract

Neural networks have become increasingly powerful and commonplace tools for guiding decision-making. However, due to the black-box nature of many of these networks, it is often difficult to understand exactly what guides them to a certain prediction, making them dangerous to use for sensitive decision making, and making it difficult to ensure confidence in their output. For instance, a network which classifies images of dogs and cats may turn out to be flawed with little consequence, but a neural network that diagnoses the presence of diseases should be assured to make sound predictions. By understanding why a network makes the decisions it does, we can help to guarantee that the choices were made in a sensible way. However, part of the reason neural networks are considered a black-box is because it is very difficult computationally to explain how they work. In fact, individual **neurons** are known to be hard to explain already. In our research, we consider whether it is possible to learn an individual neuron that is **explainable** from the start. Unfortunately, our first result tells us that it is **NP-hard** to learn such a neuron. Fortunately, we have found new conditions under which we can learn an explainable neuron in **pseudo-polynomial time**.

Introduction

Neural networks, while powerful, are difficult to explain.

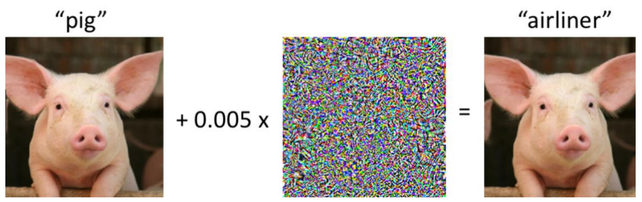
This lends itself to unexpected errors:

Prediction: Wolf
Correctly classified

Prediction: Husky
Correctly classified

Prediction: Husky
Incorrectly classified

Thus, we have need for Explainable Artificial Intelligence (XAI)

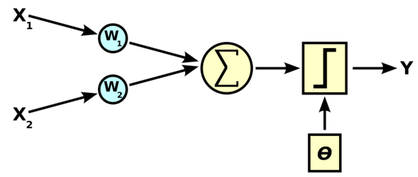


The above is an example of an **adversarial perturbation**. First, a neural network may classify an image of a pig with high confidence. However, one can add an imperceptible amount of (specially selected) noise to trick a neural network into thinking the same image is an “airliner”, also with high confidence.

In classifying images of wolves or pigs, this is a relatively benign error. However, an AI which is used for more sensitive tasks such as guiding a self-driving car cannot afford these kinds of mistakes! An AI should provide **explanations** for the decisions that it makes, in order to instill trust in them.

Results

What is the complexity of training a neuron



A linear classifier with a step activation function

From the AI literature, we know that Boolean functions are explainable. We also know the following.

Theorem: Compiling a neuron to a Boolean function is **NP-hard**.
Theorem: There is a **pseudo-polytime algorithm** for compiling a neuron with integer weights to a Boolean function.

In our research, we present these new results:

Theorem: Training a neuron with integer weights is **NP-hard**.
Theorem: There is a **pseudo-polynomial time** algorithm for training a naive Bayes classifier with integer weights.

A new pseudo-polynomial time approach

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

A **Naive Bayes classifier** is a *generative* classifier whose parameters can be learned in linear time.

A Naive Bayes classifier and a neuron are both linear classifiers, but trained in different ways.

By allowing only a **fixed budget** of weight to be distributed to the parameter weights in this formulation, we can learn an explainable neuron (with integer weights) in **pseudo-polynomial time** with respect to the weight budget.

As we increase the budget available to the model, we increase the complexity and the time required for computation

However, we may also better approximate the **unrestricted** model (without integer weights)

An algorithm to train an explainable neuron

We use **dynamic programming** to train a neuron in pseudo-polynomial time

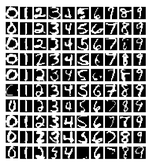
We create a table in which each row adds a parameter to the model to distribute weight to, and each column adds an additional unit weight to be distributed

To fill any given cell, find the cell in the row above which maximizes log-likelihood when distributing the remaining weight to the new parameter

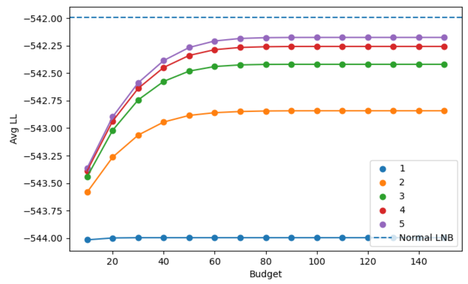
Parameter	Weight					
	0	1	2	3	4	5
b	-277.26	-264.27	-270.64	-287.80	-309.75	-333.62
w_1	-277.26	-264.27	-260.28	-266.65	-282.02	-299.18
w_2	-277.26	-249.27	-236.28	-227.65	-223.66	-225.83
w_3	-277.26	-249.27	-236.28	-227.65	-223.66	-225.83

Experiments

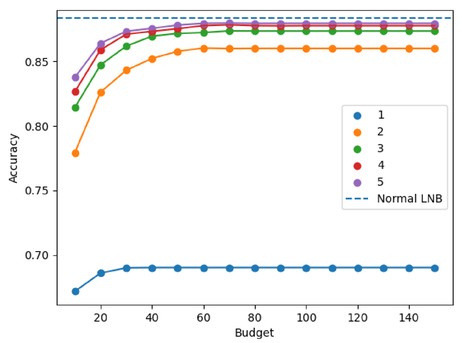
The MNIST hand-written digits dataset was used for experimentation. It consists of 28 x 28 images of digits, binarized to be black and white. All 45 pairs of digits were classified, with the results averaged in the tables below



To control the granularity of the approximation, we introduced the notion of a *grade* g, where weight was distributed in units of 1/g

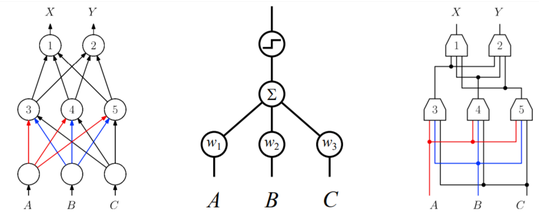


A *g* value of 5 is sufficient to approach the behavior of the unrestricted classifier

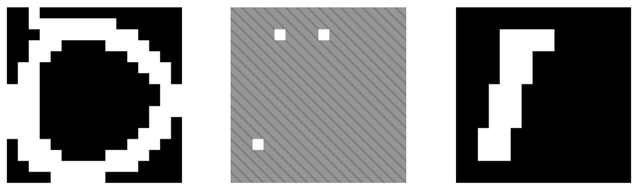


With enough weight budget, the accuracy approaches the unrestricted classifier

Use in XAI



With discrete weights, we may efficiently convert a neuron to a Boolean function. This allows us to make linear time queries into the nature of the neuron behavior



By converting a neuron to its boolean function, we **know** that the first image will be classified as 0 **because** the three white pixels are sufficient to make that decision. Therefore, we also know the third image will be classified as 0

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING

Author: Lance Kennedy - lkenne31@students.kennedaw.edu
Advisor: Arthur Choi - achoi13@kennesaw.edu