

CS6735 Programming Project Report

Ethan Garnier¹ Matthew Tidd² Minh Nguyen²
ethan.garnier78@unb.ca mtidd2@unb.ca mnguyen6@unb.ca

¹Department of Electrical and Computer Engineering, UNB

²Department of Mechanical Engineering, UNB

December 2, 2024

Abstract

In the field of Artificial Intelligence and Machine Learning, it can be very easy for the complexities of learning models to be obscured away behind the black boxes that are machine learning libraries. Despite the ease of use by which these libraries present, they do not always provide a complete understanding of how the learning is taking place. To truly grasp and take advantage of machine learning, one must understand the inner workings of the learning models being applied. This assignment saw students manually implement three machine learning models to truly test their understanding of these models and how they function. These models were: Adaboost with an ID3 weak base learner, an Artificial Neural Network with back-propagation, and Naïve Bayes. In addition to manually implementing these three learning models from scratch, two machine learning problems were solved using pre-existing machine learning libraries. These problems included developing a Deep Fully-Connected Feed-Forward Artificial Neural Network, and a Convolutional Neural Network to be trained and tested on the MNIST dataset of handwritten digits.

1 Introduction

2 Naive Bayes for Car Evaluation

In this section of the report, the team implemented a Naive Bayes (NB) classifier from scratch and investigated the performance of the algorithm. In essence, NB classifiers are a machine learning model that learns the probabilistic distribution in the training data under the naive assumption that the data features are conditionally independent. Therefore, the algorithm is a simple implementation of Bayesian networks. This section starts with a review of the theory and mechanism of NB classifiers, followed by a description of the car evaluation dataset, NB classifier implementation, finally the results and discussion of results.

3 Naive Bayes Classifier Theory

Given a dataset that contains vectors of training instances, each having n feature values $\vec{x} = (x_1, x_2, \dots, x_n)$ and a class value, an NB classifier is trained to classify a test example based on its feature values and the probability distribution in the training set. NB classifier operates on the basis of Bayes theorem, formulated as:

$$p(class|\vec{x}) = \frac{p(class)p(\vec{x}|class)}{p(\vec{x})} \quad (1)$$

The left side of the equation is the probability of an instance belonging to a class given its feature values. Bayes theorem posits that this is dependent on the probability distribution of the class, the probability of instances having the same feature given that they belong to the class

4 Implementation of Naive Bayes Classifier

4.1 Platform

Talk about the training setup (Python + Anaconda + GPU) and libraries used

4.2 Dataset Description

4.3

5 Results and Discussion