

Deep Learning for Object Recognition on a Mobile Robot

Michael Rodriguez, Julian Weisbord, Miles McCall

Written by Team: ASAP BOT

Oregon State University

CS 461 Fall 2017



Abstract

This paper goes in-depth into the requirements that we as a team must accomplish. We will be working to build an image classifier to work with our autonomous robot to help recognize basic objects in different environmental settings.

1. INTRODUCTION

1.1 Purpose/Scope

For our senior design capstone project, we will build an image classifier on top of a PR2 robot running ROS. By leveraging ROS (Robot Operating System) and the existing mobile robot platform, we can devote all of our resources to sequentially training a Convolutional Neural Network (CNN). In this project, we propose a plan of action and several potential solutions to the issues brought about from sequential learning. Additionally, there are multiple environmental variables that must be addressed during training in order to classify everyday objects in a wide variety of settings. To build a robust classifier, we will pursue three different methods to improve on the current system at the Personal Robotics Lab of OSU. These are: new data capture methods, overfitting to data in different environmental contexts using multiple classifiers, and testing different online learning models to achieve the best classification rate.

1.2 Definitions, acronyms, and abbreviations

Machine Learning: This evolved from the study of pattern recognition and computational learning theory in artificial intelligence,[4] machine learning explores the study and construction of algorithms that can learn from and make predictions on data

Overfitting: Trained model only makes predictions for data set that you have trained and not new data.

Activation Function: Introduce non-linear properties to neural networks, we need non-linear outputs to learn more complex functional mappings from data. The main purpose of an activation function is to convert an input signal of a node in an ANN to an output signal, ex:Sigmoid, ReLu, etc.

Backpropagation: Used to calculate the error contribution of each neuron after a batch of data is processed. Calculates gradient of loss function (gradient descent/ synthetic gradients). Backpropagation is a generalization of the Delta Rule.

Bias-Variance Trade-off: Bias occurs when an algorithm has limited flexibility to learn the true signal from a dataset. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting) Variance refers to an algorithms sensitivity to specific sets of training data. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (overfitting). This trade-off applies to all forms of supervised learning.

Forward Propagation: We apply a set of weights to the input data and calculate an output. For the first forward propagation, the set of weights is selected randomly.

Neural Network Training: Forward Prop (start with random weights) sum products of the inputs with their corresponding set of weights to arrive at first values of hidden layer- \sum apply activation function to hidden layers - \sum sum product of hidden layer results with the second set of weights to determine output sum - \sum take activation function at that output sum to get the final output result - \sum Back Propagation [?]

-Output sum margin of error = target -calculated

Hyperparameters: Split your training set into training set and a validation set. Use validation set to tune all hyperparameters. At the end, run a single time on the test set and report performance. Model values set before training on a dataset. Cannot be directly learned from training process. Ex: number of hidden layers in a neural network, learning rate for logistic regression. -Hyperparameter Optimization: Grid Search, Random Search, Manual Search. One of the most popular is Bayesian Optimization.

Recurrent Neural Networks: Connections between units form a directed cycle. They allow forward and backward connections between neurons. Good for unsegmented, connected handwriting recognition or speech recognition

Support Vector Machines (SVM): Supervised machine learning algorithm used for classification and regression, good for small datasets ; 1000 data points.

Online Machine Learning: A method of machine learning in which data becomes available sequentially. The opposite of batch learning, online learning updates its predictor for future data continuously with each new piece of data.

Catastrophic Interference: Catastrophic Interference becomes present when learning is sequential. Sequential training involves the network learning an input-output pattern until the error is reduced below a specific criterion, then training the network on another set of input-output patterns. A backpropagation network will forget information A if it learns input A and then input B. Catastrophic forgetting occurs because when many of the weights, where knowledge is stored, are changed, it is impossible for prior knowledge about past data to be kept intact. During sequential learning, the inputs become mixed with the new input being superimposed over top of the old input. To recognize multiple sets of patterns, the network must find a place in weight space that can represent both the new and the old output. One way to do this is by connecting a hidden unit to only a subset of the input units. [?] [?]

1.3 References

See references at the end of document

1.4 Overview

- Analyze the benefits of at least 4 different lifelong learning techniques added to the pipeline such as: Dual Memory Architecture, Functional Approximation, etc.
- Research different ways to improve the current image capture system in place at the OSU Personal Robotics Lab.
- Classify at least 3 different object classes(i.e: Dog, Mug, table) with a minimum of 80% accuracy.

2. OVERALL DESCRIPTION

2.1 Product perspective

The product we aim to deliver consists of a code base to be ran on a PR2 and/or Fetch robot with the ROS operating system. As a component of this larger existing system, our code base will append to and in some cases, replace current models running on the robots.

2.2 Product functions

Our software package encapsulates a few key functionalities. It will augment the robot's data capturing system with modern image identification algorithms, image tagging, and image recognition software. Next, the robot will take the image from its sensors and query the neural network classifiers that we have built in order to accurately classify the image. Finally, once the image has been successfully classified, the robot will update its classifiers with this image so that it can use it as an example when trying to classify other images. This is known as Sequential Learning or Online Learning. All of these functionalities currently exist within the running system, we simply aim to improve these functions with algorithms that can classify at a higher rate.

2.3 User characteristics

The intended user of our software package primarily consists of individuals already involved in the robotics lab. These users have a high degree of knowledge on the major topics covered within our project, and already have training on how to interact with the robot and integrated systems. Beyond the lab, however, users expand into a much broader range of individuals. Its entirely possible that our models could be deployed on a robot in house settings, labs, or workplace environments. The end goal user will have a large spectrum of knowledge regarding the robot and encompassing software and use the robot in a supporting role to increase their productivity.

2.4 Constraints

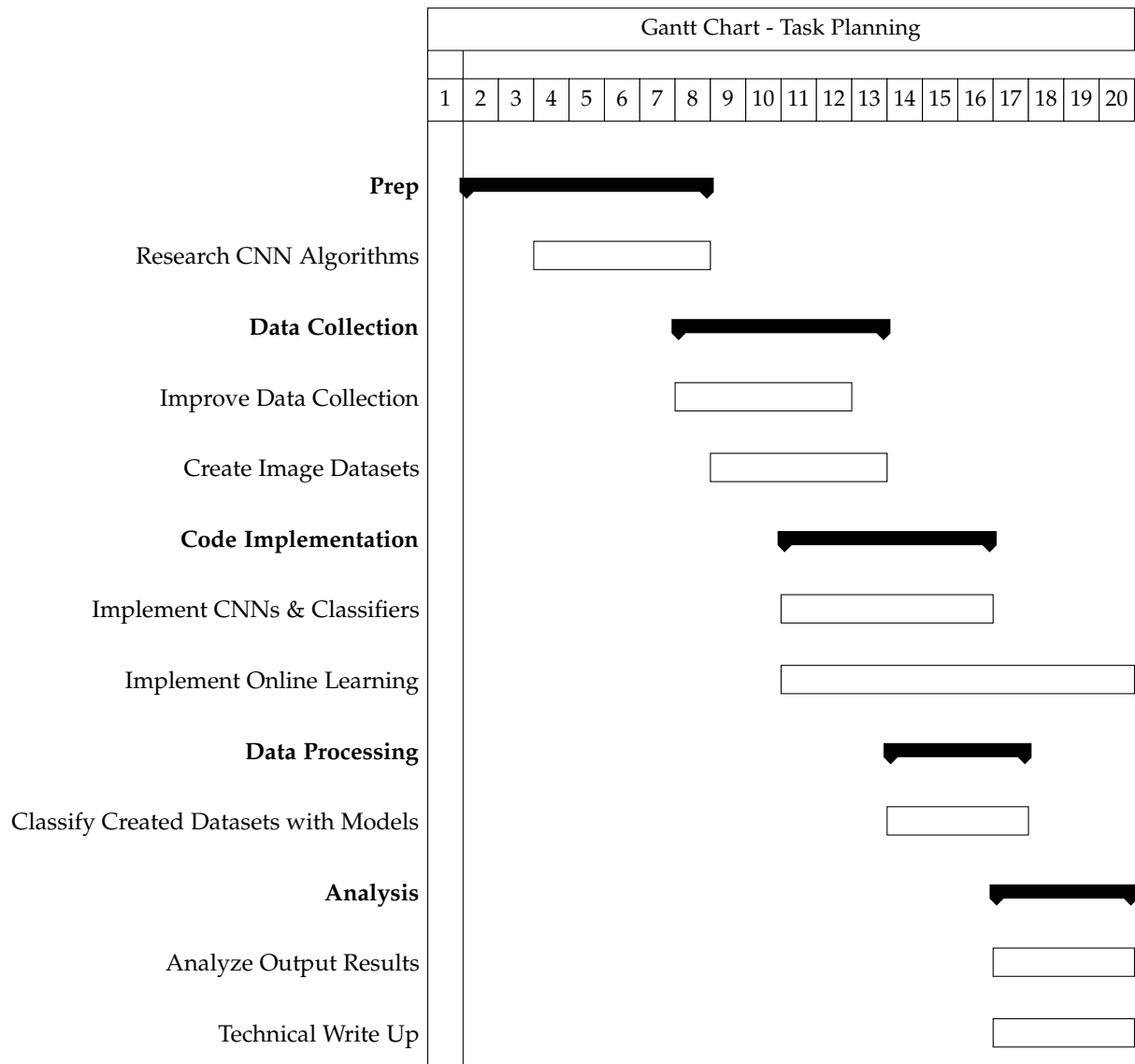
Our project is largely framed by the technologies provided to us. We are given, the Fetch and the PR2 mobile robots, servers with NVIDIA GPU's, and lab access. We are very much constrained by the quality of the camera/sensors and the overall movement capabilities. In regards to training, testing, and designing our machine learning algorithms, we will be utilizing the robotics lab's server environments. Our local/personal machines will likely lack the performance needed to effectively train our models.

2.5 Assumptions and dependencies

Our overarching project depends entirely on the described and agreed upon robot hardware and operating system. The robots have different functionality and features which will affect the resulting software we create. The operating system, however, should remain more consistent across different platforms. Luckily we can assume the ROS platform handles all communications with the underlying hardware and robot accessories, so API's will make our code more portable and modular.

3 SPECIFIC REQUIREMENTS

- Research different ways to improve the current image capture system in place at the OSU Personal Robotics Lab.
- Analyze the benefits of at least 3 different lifelong/online learning techniques added to the pipeline such as: Dual Memory Architecture, Functional Approximation, etc. Use the most accurate (remembers the most information) Online Learning technique in the final classifiers.
- Classify at least 3 different object classes with a minimum average of 80% accuracy. Evaluate the success of the different algorithms by how much the average rate of classification increases.



4 INDEX

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, acronyms, and abbreviations
 - 1.4 References
 - 1.5 Overview
2. Overall description
 - 2.1 Product perspective
 - 2.2 Product functions
 - 2.3 User characteristics
 - 2.4 Constraints
 - 2.5 Assumptions and dependencies

3. Specific requirements