Introduction to Artificial Intelligence
Project Proposal

Jay Ricco, Hasan Saif, David Van Chu

Due: June 22, 2017

1 Introduction

The motivation for this project comes from trying to understand, implement, and test datasets on a machine learning technique presented in the paper ”Learning Deep Nearest Neighbor Representations Using Differentiable Boundary Trees” (Zoran et. al.), which builds upon work presented in ”The Boundary Forest Algorithm for Online Supervised and Unsupervised Learning”. (Derbinsky, et. al.) Our team aims to present the findings of various data sets and their performance measures on Differential Boundary Trees.

2 Background

The Boundary Forest Algorithm for Online Supervised and Unsupervised Learning[1] presents an algorithm that incrementally builds a decision tree using a distance metric as the discriminator. Starting from the root, one would recurse down the tree looking at each level-set (composed of the parent and it's children), and finding the *closest* node based on the discriminator. If that node does not contain the correct class, a new node must be added as a child to the closest. This brings out an interesting property that, $\forall$ *subtrees* $\in$ Boundary Tree $\mathcal{T}$, all children of the subtree's root represent the crossing of a class boundary w. r. t. the root's feature vector. The original paper reports empirically derived time complexities of $Nlog(N)$ for training, and $log(N)$ for querying.

Learning Deep Nearest Neighbor Representations Using Differentiable Boundary Trees[2] presents an augmentation to the above. Instead of feeding raw pixel values into the tree as features, they adapt the raw values using a multi-step parameterized linear transform (multi-layer neural network), where the parameters are learned via cross entropy loss minimization. However, instead of applying a softmax layer to the output of the neural network, they form the log-normalized class probabilities from the query path through the tree. To regularize (essentially), they stop one transition from reaching the closest node. That probability distribution is applied in the cross entropy loss function as the algorithm's class hypothesis, and the neural network is trained using backpropagation with an `adam` optimizer. This paper gives few test metrics; the authors seem more concerned with minimizing the tree's size while maintaining accuracy, as opposed to overall performance.

[1] https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/viewFile/9848/9953
[2] https://arxiv.org/abs/1702.08833

3 Project Milestones

Below is a tentative schedule which outlines the absolute latest date a milestone can be completed by:

Project Milestone	**Absolute Deadline**
H.S., D.V.C. - Implement Boundary Trees successfully	June 25th
(Meet with Prof. Derbinsky) Solidify testing strategy	June 28th
J.R. - Successfully Implement DBT's H.S., D.V.C - Start testing & data collection	June 30th
J.R. - Deliver DBT implementation guide to group	July 7th
H.S., D.V.C - Finish testing Boundary Trees, start writing reports	July 10th
Project Update Due	July 13th
H.S., D.V.C - Implement DBT's, deliver BT data reports to J.R.	July 14th
J.R. - Start Final Presentation	July 15th
H.S., D.V.C - Start testing DBT's using same methods as BT's.	July 17th
H.S., D.V.C - Finish testing DBT's, start writing data reports	July 25th
H.S., D.V.C - Deliver data reports to J.R.	July 30th
J.R., H.S., D.V.C - Complete deliverable drafts	August 4th
J.R., H.S., D.V.C - Complete revisions and finalizations for delivery	August 7th

4 Work Division

The overall goal driving our work-division model is to maximize learning and experience for every member of the group. With that in mind, the overall work structure per algorithm (i.e. repeated for BT's and DBT's) is as follows:

1. Jay implements algorithm from paper, tests and sanity checks results.
2. Jay writes implementation guide, with the goal of intuitively explaining the algorithm and the reasons why certain conventions hold.
3. David and Hasan implement the algorithm using Jay's guide, and then toy with it to develop a firm understanding.
4. David and Hasan divide testing and data collection amongst themselves - individually write and return reports to Jay[3].
5. Reports and collected data are merged and augmented by the group as a whole to form final deliverables.

[3] Who, during this time - will also be available for consultation should any issues arise; if all else fails - go to Professor Derbinsky.