Introduction to Artificial Intelligence
Project Update

Jay Ricco, David Van Chu

Due: July 13, 2017

1 Update

Binary Trees implementation is complete and David has began testing it with MNIST and CIFAR10 datasets. Jay ran into a setback with the Differentiable Boundary Trees portion of the project, and is currently implementing it using Torch rather than TensorFlow. If this is not completed by Tuesday, July 18th, we will pivot to focus on the existing Boundary Trees implementation. Meanwhile, David will continue testing the Boundary Trees:

1. Set a seed for random to make future results reproducable.
2. Set the branching factor to different values.
3. Calculate standard deviation along with averages.
4. Find a way to visualize the datasets during training and testing.

1.1 DBT's Setback

Jay's issue was caused by the underlying computation architecture of TensorFlow. The compute graphs compiled by the library are static, while the algorithm requires dynamically building them. Implementation in PyTorch should be far easier and actually accurate. To date, the logical flow behind the implementation is as follows:

$trainbatch$ = get_mnist_training_batch(BATCH_SIZE)
f_θ = initialize_feedforward_compute_graph()
for $(tx, ty) \in trainbatch$ **do**
| Train Boundary Tree $\mathcal{T}$ using transform f_θ
end
for $(tx, ty) \in \subset trainbatch$ **do**
| Build Network Function Net_θ using 'Torchified' query function.
| Back-propagate error and update θ using Net
end

2 Backup Plan

If we decide to focus on the existing Boundary Trees implementation, we will test it with different distance functions to see how they perform on MNIST and CIFAR10. We will also create synthetic data sets to test speed as dataset size increases, without caring how accurate the boundary tree is.

Average MNIST Accuracy	
Num Examples	Average
1	14.506
10	24.551
100	53.122
1000	71.948
5000	79.371
10000	82.285
20000	84.744
30000	85.893
40000	86.948
50000	87.492
60000	88.201

Figure 1: MNIST Averages.

3 Data Collected

So far, we have collected data using MNIST and CIFAR10 with (unintentionally) a branching factor set to infinity, essentially replicating k-nearest neighbors algorithm. Using all of the examples for each data set, the average accuracy for MNIST was 88.201%, and for CIFAR10, 26.936%. We expect that Differentiable Boundary Trees will be more accurate than Boundary Trees, especially with the CIFAR10 dataset. There may be other distance functions that improve the accuracy rate of CIFAR10 with regular Boundary Trees, and if we are not able to implement DBTs, then we will investigate other distance functions.

Average CIFAR Accuracy	
Num Examples	Accuracy
1	11.559
10	13.904
100	17.468
1000	20.984
5000	22.905
10000	23.758
20000	25.584
30000	26.504
40000	26.858
50000	26.936

Figure 2: CIFAR Averages.