CS 599: Foundations of Deep Learning
Assignment #00010

December 17, 2024

Course Policy

Carefully read all the instructions below before you start working on the assignment, and before you make a submission.

- All Problems should be coded in Tensorflow 2 or above
- Please typeset your submissions in LaTeX, give maximum explanation for each subproblems.
- Assignments are due at the end of the day at 11:59 pm on the due date given on the web Portal.
- No single line answers are accepted in the submission.
- Late assignments will suffer 50 percent loss after the first day and all loss after the second.
- All source materials must be cited. The University Academic Code of Conduct will be strictly enforced.
- We will be creating Canvas submission page for this. You have to submit python file[no ipython allowed] given in gitrepo along with your response pdf.

Problem 1: Measuring Catastrophic Forgetting in Multi-layer perceptron (2+2+2+2=8 points)

Deep learning approaches have lead to many breakthrough in various domain, yet they suffer from credit assignment and forgetting problem. Deep learning systems have become more capable over-time, however standard multi-layer perceptron(MLP) and traditional training approaches cannot handle incrementally learning new tasks or categories without catastrophically forgetting previously learned training data. We call these problem catastrophic forgetting in neural networks. Fixing this problem is critical so that agents learn and improve incrementally when deployed in real-life setting. In simple term when you have trained model on Task A, and using same weights for learning a new Task B. Then your model forgets/loss learned information about Task A. This means it catastrophically forgot previous information. In this assignment you will be measuring and analysing catastrophic forgetting in neural networks. How will you do this? Let's get into detail.

- You will be using permuted mnist for this assignment. Script for same is provided in the github repo
- Train a network for more than 50 epochs to reach desirable performance/accuracy.
- Now test on Task A, use the same network and then train on Task B for 20 epochs, now Test on Task A and B. Continue this for N Task(N = 10)
- Total Number of Epochs [50+20+20+20+20+20+20+20+20+20]= 230
- Create Resulting Task Matrix(R) as given in [1] and report ACC and BWT(Formulation given below)

$$ACC = \frac{1}{T}\sum_{i=1}^{T} R_{T,i}, \quad BWT = \frac{1}{T-1}\sum_{i=1}^{T-1}(R_{T,i} - R_{i,i})$$

We will be using standard MLP with various depth in this assignment
You will be creating 3 models of depth 2,3,4 each with 256 hidden units.

Things to Report

**NOTE:- All submissions should use NIPS latex template.
Pdf generated from NIPS template would only be accepted rest all would lead to zero points.**

- Fork repo [`https://github.com/AnkurMali/IST597_Fall2019_TF2.0/Assignment3/`] and modify file forgetting_mlp.py.
- Effect of various loss function on forgetting(NLL,L1,L2,L1+L2)
- Does dropout help? Apply dropout ≤ 0.5. Report your observations.
- Effect of depth on forgetting.
- Report ACC and BWT.
- Does optimizer plays a role in less forgetting. Test with SGD,Adam, RMSProp and report your findings.
- Remember to use tricks from previous assignments and tutorial to get better model.
- Don't forget to use your unique Seed Value.
- Plot your validation results for decrease in model prediction when you have finished training on all given 10 tasks.[Remember:- Validation accuracy is available for n epochs for each task]
- **Bonus Point:-** Report TBWT and CBWT proposed in [2]

References

1. Lopez-Paz, D., et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems* (2017), pp. 6470–6479.
2. Ororbia, A., Mali, A., Kifer, D., and Giles, C. L. Lifelong neural predictive coding: Sparsity yields less forgetting when learning cumulatively. *CoRR abs/1905.10696* (2019).