Landon Buell

Error Compensation Attacks

Data Analysis Report

19 July 2020

**Error Compensation Attacks - Data Analysis Report**

Convolutional Neural Network with “Approximation Layer” and “Compensation Layer” Objects after Input Layer Object. All sample used are images with shape 32 x 32 x 3 matrices of bytes. A “Mask vector” is hard coded within the program, which is used to determine the indexes of pixels with will be subject to the approximation or compensation.

Approximation Layer Object:

Takes a batch of samples shaped (batch\_size,32,32,3) and tests for a “beta” condition, based on the internal clock. Given the FLOP frequency, the beta condition is set to a microsecond accuracy. If the systems internal clock measures the microsecond counter as an even number when the layer is called, then the ask is used to apply the approximation to the designated pixels. If the microsecond counter is an odd number, no approximation is applied, and the batch is passed through the layer untouched. The approximation method approximates all designated bits to *0*.

Compensation Layer Object:

Takes a batch of samples shaped (batch\_size,32,32,3) and always attempts to compensate for the previous’ layer’s approximation. It does not use a clock to check for beta. Sections of data preserved in the images interior area “copy and pasted” to the exterior as the attempt to *compensate* for lost or approximated data.

**Landon’s Hypothesis**

The results of the previous related experiment showed that applying the approximation layer to the classifier models often reduced the neural networks ability to converge on a set of parameters to minimize the loss function and thus the precision and recall score were also compromised. In this experiment, the approximation boundary is far less defined since the layer only enacts when the “beta” condition is met. Because of this, the image that is fed into each of the convolution layers is either the full image, or an approximation of the image.

Thus, I predict in the convolutional network performance will be impacted even more, due to the inconsistencies of the input data. This will cause a generally high loss score and low precision and recall scores across the model subject to approximations. I predict the magnitude of the deviation from the baseline model to be directly proportional to the depth of the approximation border.

**Analysis of Results**

Loss Values for 1 and 2 layer groups

In all tested models with approximation layers, the average loss function value for one and two hidden layer groups showed much higher values, meaning that selectively choosing when to approximate data, blurs the approximation boundary, but provides the model with a no-homogenous set for training/testing data (i.e. an approximated batch is different that non-approximated batch.) Additionally, the deviate from the baseline is not directly proportional to the depth of approximated pixels, but rather with increasing depth shows increasingly more *chaotic* behavior.

In all tested models with approximation layers followed by compensation layers showed drastic improvements in the reduction of the loss function, almost comparable to the baseline. I would attribute this to the homogeneity of the compensation data. – All batched are compensated regardless, which means all the training and evaluation data is consistent.

Against the original goal of this experiment, the compensation layer has effectively compensated for the approximation errors made the in the approximation layer and allowed for the neural network to appropriately minimize the loss function.

Precision Score Values and Recall Score Values for 1 and 2 layer group

Precision and recall scores for all models with approximation layers showed significant performance compromises when compared to the baseline. Rather than these deviations being proportional to the depth od approximated pixels, the greater the pixel depth, the more chaotoic the socre seem – the more they vary.

Precision and recall scores for all models with approximation and compensation layers showed no major deviations from the baseline. Since the loss function was able to remain consistent with the baseline, so to do the metric score stay comparable to the baseline.

Again, against the original goal of this experiment, the compensation layer has effectively compensated for the approximation errors made the in the approximation layer and allowed for the neural network to maintain a reasonable performance by these two metrics.

Execution Time for 1 and 2 layer Groups

The execution time across models with approximation and compensation showed no major deviations from the baseline and remained roughly consistent throughout. The execution time seems to be a function of the number network parameters, not necessarily the depth of width explicitly.