Deep Compression

October 31, 2017

Procedure

- VGG19 was used in CIFAR10.
- Regularization techniques employed during training were L2, Batch normalization, Dropout, Data Augmentation.
- Bias terms (w_0) were employed in all layers, including the convolutional layers.
- Batch normalization was used after each convolutional layer.
- Model Description: VGG19_BN_drop_10

  Image size: 32x32x3

[64, 64, 'M',	16x16x64
128, 128, 'M',	8x8x128
256, 256, 256, 256, 'M',	4x4x256
512, 512, 512, 512, 'M',	2x2x512
512, 512, 512, 512, 'M',	1x1x512
4906, 4906, 10]	

Procedure

- VGG19 using CIFAR10 generates an overall of 38M (38,969,930) parameters to be trained.
- Pruning employs the standard deviation as a quality parameter.
- 1 iteration is composed by a ”pruning” and ”retraining” stage.
- The retraining has the following configuration:
  iterations: 20
  number of epochs: 20
  initial learning rate: 0.05,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01, 0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.001,0.001
  learning rate schedule: 3,10,16

Experiments

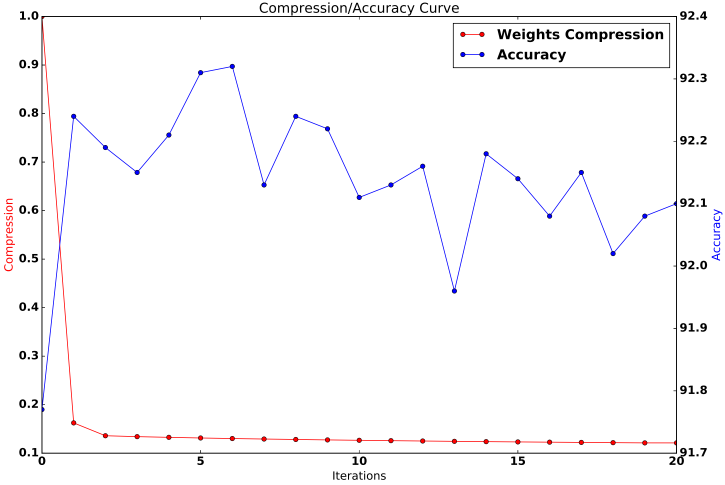


Figure 1: Compression vs Accuracy for a VGG19. zero_weights / all_parameters

Experiments

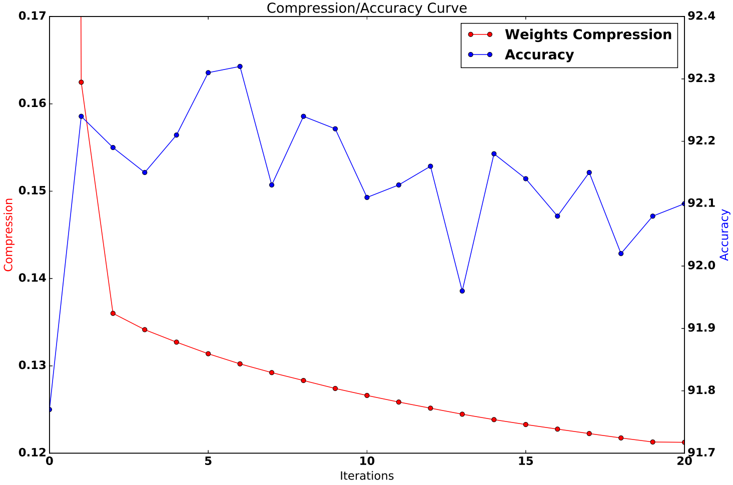


Figure 2: Compression vs Accuracy for a VGG19. zero_weights / all_parameters

Experiments

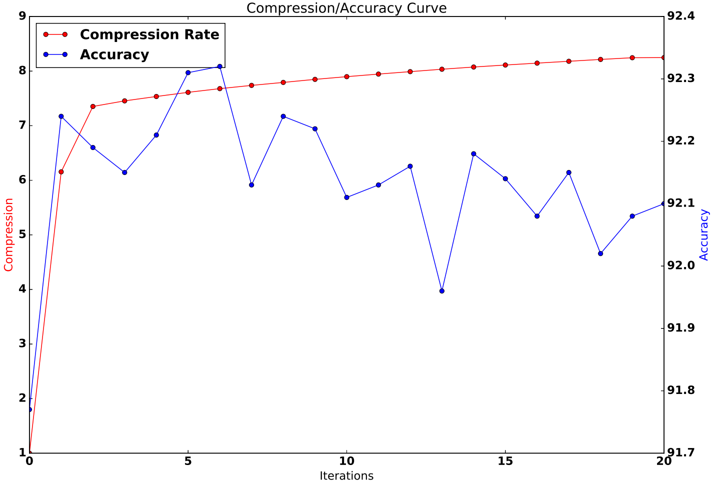
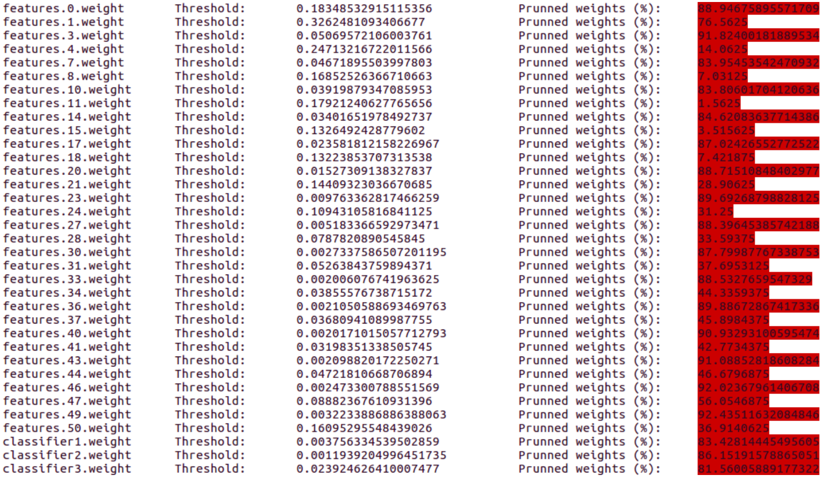


Figure 3: Compression vs Accuracy for a VGG19. all_parameters / non-zero_weights

Experiments

- Final Parameters:
  total parameters: 38969930,
  Total weights: 38939712,
  Zero weights: 34244952,
  Zero weights rate (%): 87.9435163773,
  Zero parameters rate in model (%): 87.8753233583
  Compression rate: x8.24764263 times

Experiments



```
features.0.weight       Threshold:       0.18348532915115356          Prunned weights (%):      88.94675895571709
features.1.weight       Threshold:       0.3262481093406677           Prunned weights (%):      76.5625
features.3.weight       Threshold:       0.05069572106003761          Prunned weights (%):      91.82400181889534
features.4.weight       Threshold:       0.24713216722011566          Prunned weights (%):      14.0625
features.7.weight       Threshold:       0.04671895503997803          Prunned weights (%):      83.95453542470932
features.8.weight       Threshold:       0.16852526366710663          Prunned weights (%):      7.03125
features.10.weight      Threshold:       0.03919879347085953          Prunned weights (%):      83.80601704120636
features.11.weight      Threshold:       0.17921240627765656          Prunned weights (%):      1.5625
features.14.weight      Threshold:       0.03401651978492737          Prunned weights (%):      84.62083637714386
features.15.weight      Threshold:       0.1326492428779602           Prunned weights (%):      3.515625
features.17.weight      Threshold:       0.023581812158226967         Prunned weights (%):      87.02426552772522
features.18.weight      Threshold:       0.13223853707313538          Prunned weights (%):      7.421875
features.20.weight      Threshold:       0.01527309138327837          Prunned weights (%):      88.71510848402977
features.21.weight      Threshold:       0.14409323036670685          Prunned weights (%):      28.90625
features.23.weight      Threshold:       0.009763362817466259         Prunned weights (%):      89.69268798828125
features.24.weight      Threshold:       0.10943105816841125          Prunned weights (%):      31.25
features.27.weight      Threshold:       0.005183366592973471         Prunned weights (%):      88.39645385742188
features.28.weight      Threshold:       0.0787820890545845           Prunned weights (%):      33.59375
features.30.weight      Threshold:       0.0027337586507201195        Prunned weights (%):      87.79987767338753
features.31.weight      Threshold:       0.05263843759894371          Prunned weights (%):      37.6953125
features.33.weight      Threshold:       0.002006076741963625         Prunned weights (%):      88.532765954 7329
features.34.weight      Threshold:       0.03855576738715172          Prunned weights (%):      44.3359375
features.36.weight      Threshold:       0.0021050588693469763        Prunned weights (%):      89.88672867417336
features.37.weight      Threshold:       0.03680941089987755          Prunned weights (%):      45.8984375
features.40.weight      Threshold:       0.0020171015057712793        Prunned weights (%):      90.93293100595474
features.41.weight      Threshold:       0.03198351338505745          Prunned weights (%):      42.7734375
features.43.weight      Threshold:       0.002098820172250271         Prunned weights (%):      91.08852818608284
features.44.weight      Threshold:       0.04721810668706894          Prunned weights (%):      46.6796875
features.46.weight      Threshold:       0.002473300788551569         Prunned weights (%):      92.02367961406708
features.47.weight      Threshold:       0.08882367610931396          Prunned weights (%):      56.0546875
features.49.weight      Threshold:       0.0032233886886388063        Prunned weights (%):      92.43511632084846
features.50.weight      Threshold:       0.16095295548439026          Prunned weights (%):      36.9140625
classifier1.weight      Threshold:       0.003756334539502859         Prunned weights (%):      83.42814445495605
classifier2.weight      Threshold:       0.0011939204996451735        Prunned weights (%):      86.15191578865051
classifier3.weight      Threshold:       0.023924626410007477         Prunned weights (%):      81.56005889177322
```

Figure 4: Compression in each layer. Weights in convolutional layers, Batch normalization and dense layers

Troubles

- may I use mask in weights and gradients? a previous paper employ this approach for weights only, it means the mask was employed only in weights and the paper did not mention anything about gradients.
- Next Steps. Trained Quantization and Weight sharing.