Pulse-Net: Dynamic Compression of Convolutional Neural Networks

D. Browne and S. D. Prestwich
{david.browne,steven.prestwich}@insight-centre.org

Insight Centre for Data Analytics

Objective

To introduce a novel technique that prunes and compresses a Neural Network, thus optimising its structure to make it run efficiently during the inference stage, while maintaining close to state-of-the-art accuracy.

Motivation

In recent years, we have seen the explosion of increasingly deep neural networks which achieve start-of-the-art results in the area of computer vision, among others. The rapid growth of deep convolutional neural networks is due to hardware developments in the form of powerful GPUs, software developments in the form of stochastic gradient descent and new network architectures, and the public availability of large labelled datasets such as the ImageNet Classification tasks.

However, because deep CNNs rely heavily on powerful GPUs and consume a great deal of memory, their practical uses may be limited. AlexNet {1} has about 61 million parameters and needs over 200 MB of storage, and VGG16 has 138 million parameters requiring 500 MB. The more parameters the model has, the more memory it consumes and the more energy is needed during inference. This is particularly important when these networks are deployed on mobile devices, while memory demand is the key resource for usage on the cloud. Inference time can be just as important as accuracy for online image recognition where thousands of images per second may require analysis. We show that by compressing/pruning the network we can greatly reduce the number of parameters, leading to a significant reduction in the number of FLOPs (which is directly associated to inference time).

There are two general approaches to compressing a network: during training {2} or after training {3}. Our proposal, called Pulse-Net, falls in the first group.

Pulse-Net

Repeated step learning rate to determine network convergence

Algorithm 1 Learning Rate Procedure

```
Initialize lr-step = 0
Initialize lr-rate = lr-list[lr-step]
During training of Network; X = Calcul(10-ma)
Repeat until lr-step = length ( lr-list ):
If 10-ma stops decreasing:
    If lr-step < length ( lr-list )
        lr- step = lr- step +1
    Else:
        Break loop due to Network convergence
```

Pulse-Net Network Pruning

Algorithm 2 Pulse-Net

```
Train Network until validiation loss convergence
Calculate validiation acc and store as best acc
Repeat until # Filters/Nodes of max layer removed < β:
    While Diff[validiation acc, Best acc] < λ:
        Remove α min[Filters/Nodes] in all layers of Network
        Fine-Tune Network until validiation loss convergence
        Calculate validiation acc
        If validiation acc > best acc:
            best acc − > validiation acc
        Else:
            α = 0.5(α)
```

Results

CIFAR10

Network	Accuracy (%)	Parameters	MACs (M)	Storage (MB)	Speed (ms)	Energy (mJ)
AlexNet	91.16	5.83 X 107	874	222.5	4.14	0.95
Compressed	2%	95.63%	95.31%	95.63%	50.72%	65.26%
TensorFlow	85.36	1.07 X 106	18.4	4.08	1.69	0.39
Compressed	3.6%	76.25%	72.83%	76.25%	2.95%	13.59%
VGG16	90.87	3.36 X 107	287.2	128.36	6.32	0.84
Compressed	0.9%	87.85%	87.89%	87.85%	47.94%	59.52%

CIFAR100

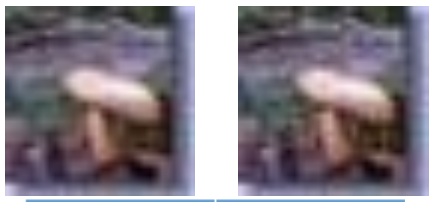
Network	Accuracy (%)	Parameters	MACs (M)	Storage (MB)	Speed (ms)	Energy (mJ)
AlexNet	69.77	5.87 X 107	874.5	223.9	4.16	1.04
Compressed	2.31%	85.95%	85.76%	85.95%	46.15%	67.46%
TensorFlow	58.05	1.09 X 106	18.5	4.14	1.68	0.40
Compressed	4.5%	65.57%	63.03%	65.57%	0.6%	8.79%
VGG16	64.2	3.4 X 107	287.6	129.77	6.29	0.83
Compressed	1.14%	87.6%	87.87%	87.6%	48.33%	57.52%

Tiny-ImageNet

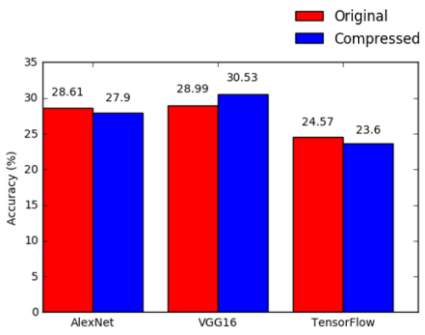
Network	Accuracy (%)	Parameters	MACs (M)	Storage (MB)	Speed (ms)	Energy (mJ)
AlexNet	54.8	7.27 X 107	1266.2	277.47	8.68	1.32
Compressed	3.75%	80.79%	79.20%	80.79%	56.57%	65.15%
TensorFlow	40.45	5.04 X 106	100	19.22	2.1	0.31
Compressed	2.91%	74.25%	71.26%	74.25%	15.24%	12.9%
VGG16	56.05	4.07 X 107	1010.8	155.33	6.67	0.86
Compressed	2%	83.15%	83.81%	83.15%	44.68%	61.63%

Robustness of Pulse-Net

Adversarial Attack



Mushroom 99.82%	Beaver 99.72%
Beaver 0%	Mushroom 0%



Pulse-Net shows it is 75% more robust to adversarial attacks. This is due to the reduced number of parameters that can be attacked.

Conclusions and Future Work

We proposed a novel deep CNN pruning method called Pulse-Net, which compresses a network during training to create a more efficient model for inference. The proposed compression method shows significant improvements in storage, inference timings and energy efficiency, as well as greater robustness under adversarial attack.

In future work we would like to explore different metrics for pruning, as well as removing filters in other types of networks like ResNet. In addition, we believe research into pruning the depth of networks as well as the width, using Pulse-Net, would be an interesting expansion of the work.

References

[1] A Krizhevsky et al. ImageNet classification with deep convolutional neural networks. In NIPS, (2012), pp. 1106–1114
[2] M. Courbariaux et al. Binaryconnect: training deep neural networks with binary weights during propagations. In NIPS, (2015), pp. 3105-–3113.
[3] W. Chen et al. Compressing neural networks with the hashing trick. In ICML, (2015).

A World Leading SFI Research Centre

This project has been funded by Science Foundation Ireland grant SFI/12/RC/2289

Ireland's European Structural and Investment Funds Programmes 2014-2020 Co-funded by the Irish Government and the European Union
EUROPEAN UNION European Regional Development Fund
DCU NUI Galway OÉ Gaillimh UCC University College Cork, Ireland Coláiste na hOllscoile Corcaigh UCD DUBLIN
Science Foundation Ireland sfi For what's next