ENGN8536 Assignment

Shikhar Mishra
u6203537

Part 1: GoogLeNet (3 marks)

What were the key innovations of GoogLeNet with respect to VGG? : The key innovation that GoogLeNet introduced was the idea of the inception module[11], which utilised sparse connections to enable deeper networks, which are achieved with smaller sized convolutions. There are no fully-connected (FC) layers in GoogLeNet, whereas the VGG[10] model has a densely connected structure with three FC layers at the end of the network, the first of which alone accounts for 100M parameters! Each inception module can extract input feature at different scales (1x1, 3x3, and 5x5) in parallel, followed by concatenation to cluster these features, whereas VGG only use 3x3 filters. Finally, the idea of stacking blocks instead of individual layers has inspired future networks.

One key mechanism reduces the number of parameters in a block. Explain how it does this. GoogLeNet achieves a reduction in total parameters (4 million vs 138 million in VGG) due to the way it implements sparse connections, and dimension reduction before applying the larger filters. A 1x1 convolution filter called 'bottleneck' is used to 'compress' the input channel depth before the 3x3 and 5x5 convolutions, and thereby keeping the same number of spatial locations with less features. This saving allows reduces both the number of parameters and computation time, while allowing for wider and deeper networks.

Explain how this/these mechanisms help improve accuracy. GoogLeNet captures details from three different scales, as described previously. This allows it to detect smaller as well are larger details in an image much better than VGG, thereby increasing generalizability of the model. While there are no FC layers here, global average pooling is able to effectively average out values in the feature map without reducing the accuracy due to compression.

Part 2: ResNet (4 marks)

ResNet created much deeper networks than its predecessors (particularly VGG). Describe the key innovation(s) of ResNet. The ResNet paper [5] utilises their idea of 'skip connection' units, which are effectively shortcuts that jump between layers, taking its inspiration from gates in Recurrent NNs. This enables them to solve the degradation of training accuracy problem. Another inspiring idea is for learning from residuals of the previous layers, rather than relearn for all layers when creating a deeper network. While they weren't the first to propose it [7], there is also an emphasis on using batch normalisation layers after each convolution, a technique which became more popular after this paper. This is to address the vanishing gradient problem (discussed later). Utilising these innovations, the authors managed to train upto a 1202-layer network, which was very deep for contemporary standards, without any difficulties. ResNet won the authors 1st places in ILSVRC and COCO 2015, as well as other benchmarking competitions.

Explain what was the problem that this feature addressed and how ResNet addressed the problem. (Hint: you need to talk about vanishing gradients.) The vanishing gradient problem refers to the attenuation of the signal during backpropagation in very deep networks. This problem is solved mostly by using ReLU as an activation function, since it has a constant gradient, and by using normalised initialisation[3] and Batchnorm layers in the middle[7] of a network. This allows convergence for tens of layers using the SGD. Further, it was empirically found by the authors that stacking extra layers to make a network deeper leads to a higher training error, than compared to a

shallower network, which made training harder. This is called the degradation problem. The authors were able to address this degradation problem of deep networks by making a 'shortcut' or skip path between the input and output layer, so as to enable an identity mapping, such that the extra layers only learn from the residual of the previous. Further, this new path allows proper activation of layers to the deep layers.

Compare and contrast the design of ResNet vs that of VGG. The main object of difference among the two networks is the use of residual blocks in ResNet, while the VGG is densely connected. However, the convolution filters used in ResNet are mostly 3x3, similar to those in VGG blocks. In fact, the modular design of residual blocks, where a combination of filters are used to extract representative and complex features, is inspired from VGG. Another similarity between both networks is the previously mentioned use of global average pooling layers after the classification layer. ResNet also utilises 'bottlenecks' in its deeper variants (50-layer onwards), similar to GoogLeNet.

What is the importance of Batch Normalization in this network? : ResNet uses batch normalisation after every convolution and before the activation function. Batchnorming allows for reparametrising and hence standardising the activations of the prior layer. In the ResNet architecture, this stabilises optimisation using the SGD, allowing much higher learning rates, better weight decay, faster training, and improved generalisation. The authors discuss some of these points in the Implementation and Experiments section of the paper. When used in conjunction with normalised initialisation and ReLU activation unit, it solves the vanishing gradient problem.

Part 3: Squeeze and Excitation Networks (3 marks)

What is the key innovation of Squeeze and Excitation networks? : The key idea of the squeeze and excitation network is the modelling of the depndncies between channels, which allows for an adaptive recalibration of feature responses per channel. This is done in the Squeeze-and-Excitation (SE) block. These SE blocks can be an attachment to the standard Inception or ResNet module. Another key feature is the use of downsampling at later stages of the network.

How does this feature improve performance? : In the Squeeze part of the SE block, using global average pooling, spatial information is 'squeezed' into descriptors per channel. Later in the excitation module, dimensionality reduction is done which reduces total number of parameters. The paper conducts extensive testing of the extra performance achieved by adding SE blocks in various backbone networks. In summary, it is found that they generalise pretty effectively over different nets and datasets.

Part 4: What has this lead to? (10 marks)

Source paper: Deep Residual Learning for Image Recognition (ResNet)

Summary, innovations, weaknesses: I have chosen the ResNet paper [5] as my source paper. The authors propose a residual learning framework, such that instead of learning unreferenced functions, the layers learn residual functions with reference its inputs.

We know from theory of machine learning about the Universal Approximation Theorem, that feed-forward neural networks can be thought of as non-parametric 'universal function approximators' [1]. In fact, it has been shown that under certain mathematical conditions, a network with arbitrary depth and neurons can approximate any continuous function [6]. However, in practice, it was hard to train fully connected deep networks, and they were prone to overfitting. This was further complicated due to the vanishing gradient problem, whereby the gradient loses its signal progressively as it is backpropagated. Further, the degradation problem during training saturates the performance improvements from extra added layers in a deeper network.

The authors solve these problems by using 'identity shortcut connections', that skip some layers before joining back into the main network. These connections do not increase either complexity or total parameters, rather they simply conduct identity mapping; the output of the extra layers is then added to this. The key idea here being that extra stacked layers in a deeper model should not degrade performance at minimum, as the extra layers can be set to identity by construction. They also were one of the first to use batch normalisation, which was proposed earlier that year[7]. Further details

about the innovations were described previously. The results and experiments show that ResNet-110 achieved state-of-the-art 6.43% classification error on the CIFAR-10 dataset. Another strength of the framework was its generalisation performance for different tasks such as detection, localisation and segmentation, which is why it is used as the default backbone for many modern networks. ResNet won the authors 1st places in ILSVRC and COCO 2015, as well as other benchmarking competitions.

While ResNet was groundbreaking at the time, its weaknesses were studied and improved upon by future architectures. For example, by reordering the batchnorm and ReLU layers, gradients can be propagated much more easily[4]. Further analysis showed that sometimes, just a couple of residual blocks learned all the key representations, while others weren't learning anything, a phenomenon known as diminishing feature reuse. The ResNet paper focussed on increasing depth using residual blocks, but there was no guarantee that a gradient signal will go through a block. Wide ResNet[13] suggests modifications in the width to increase efficiency, and address the diminishing reuse problem (discussed later).

Follow-up Paper: Aggregated Residual Transformations for Deep Neural Networks (ResNeXt)

For the follow up paper, I am going to talk about ResNext [12] from Facebook Research, which as the name suggests is the next version of ResNet. It was published in the 2017 edition of CVPR. It follows up the original by combining some of the characteristics from GoogLeNet, such as its split-transform-merge paradigm in the Inception module[11].

The key idea of this paper is to have parallel branches within a residual block, so that instead of convolving over the full map, we split the input into smaller channels and apply filters separately to each, before combining the results. However, there are some key differences from Inception, in which each path extracted input feature at different scales (1x1, 3x3, and 5x5) and were depth-concatenated. Here the hyperparameters, that are width and filter size, are the same for each of the paths, and simple addition is done to merge them together. This splitting idea is taken from the group convolutions in the original AlexNet [8].

The methodology of the paper starts of by explaining what splitting, transforming, and aggregating mean in terms of a simple neuron inner product. This analogy is then extended to aggregated transformations which form a basis for their paper. It is here the authors introduce the "next" dimension of ResNext, called cardinality, which is a new hyperparameter that controls the number of independent paths in a ResNext block. The paper then does extensive testing on performance and accuracy to find that it is more effective to increase cardinality than increasing width or depth of a network. Another benefit of this approach is that there is only one hyperparameter to optimise in this framework, compare to the previous networks.

The authors also discuss three equivalent formulations that achieve the same results. For the actual implementation, the authors use the simplified model that utilises group convolutions and a bottleneck design, and runs faster than the theoretically equivalent models. It should be noted that ResNet models are just a special case of the ResNext, for example when cardinality is one, and width of bottleneck is 64, ResNext-50 becomes ResNet-50!

The majority of the paper is focused on statistical analyses of testing results, using the datasets Imagenet-1k/5k and CIFAR-10/100. It was empirically shown that by just increasing cardinality by one helps reduce error significantly, while just increasing the width of the network doesn't help much. This is highlighted by the result that ResNext-101 performs better than ResNet-200 and a Wide ResNet-101. They also conducted a study into the effectiveness of skip connections and found that residual connections are responsible for achieving a 20% better performance. Overall, the experimental analysis of this paper is much more thorough than the original paper, and discusses several issues that the original misses.

The paper also excelled in object detection on COCO dataset, something that was achievement for the original ResNet as well. It is seen that for similarly complex models, ResNext is an improvement over the original. ResNext won second place in the ILSVRC 2016, achieving a 15% improvement in error rate over the original ResNet in one year; this is at the same time as keeping the total parameters roughly on par with original ResNet at about 25 million, for 50 layer variants of each.

Overall the conclusion from this paper is that residual connections with parallel branches are great for training and optimisation. The use of aggregated transformations helps in better and simplified

representations. This model is used as a backbone for many later networks, such as the YOLOv3[9] and v4[2], among other object classification, detection, and segmentation tasks.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016. URL `http://arxiv.org/abs/1603.05027`.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. URL `http://arxiv.org/abs/1804.02767`.

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[12] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] S. Zagoruyko and N. Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. URL `http://arxiv.org/abs/1605.07146`.