REPRODUCIBILITY REPORT

Hamish Flynn
ID: 30635462
hef1g18@soton.ac.uk

Richard Cann
ID: 28207696
rmcp1g15@soton.ac.uk

Aaron Tasker
ID: 27637433
at10g15@soton.ac.uk

ABSTRACT

We attempted to reproduce some of the results in George Philipp's paper *Gradients Explode - Deep Networks Are Shallow - ResNets Explained* Philipp et al. (2018). The authors addressed the notion that Adam, batch norm and SeLU nonlinearities solve the exploding gradient problem. They claim that, in general, these techniques do not prevent exploding gradients in MLPs.

1 INTRODUCTION

There is a large body of work that demonstrates the advantages of deeper neural networks Montufar et al. (2014), Mhaskar & Poggio (2016). However, deep feed-forward neural networks are hard to train, in part due to the exploding gradient problem. The authors introduce the gradient scale coefficient (GSC) and propose that it becomes the standard metric for exploding gradients. They use it to show that exploding gradients are present in multilayer perceptrons (MLPs) with a variety of activation functions and normilisation layers. They then try and show why exploding gradients are likely to occur in deep MLPs, and to what extent this limits their effective depth.

2 BACKGROUND

The authors number layers in reverse order, i.e. layer 1 comes after layer 2. The gradient scale coefficient (GSC) for $0 \leq l \leq k \leq L$, for an MLP of depth L is defined as:

$$GSC(k, l, f, \theta, x, y) = \frac{||\mathcal{J}_k^l||_{qm}^2 ||f_k||_2^2}{||f_l||_2^2}$$

where f is the MLP with parameters θ; x and y are the inputs and labels; $\mathcal{J}_k^l$ is the Jacobian of the lth layer with respect to the kth layer; f_k and f_l are the outputs of the lth and kth layers respectively and $||\cdot||_{qm}$ is the quadratic mean norm. The authors say an MLP $f(\theta)$ has exploding gradients if the $GSC(k, l)$ increases exponentially with the number of layers between layer k and layer l. In all the experiments we reproduced, the authors did not use learnable batch and layer normalisation layers. For batch-norm, the mean and standard deviation are the component-wise mean and standard deviation over the current batch. For layer-norm, the mean and standard deviation are component-wise over the features.

3 EXPERIMENTS

3.1 GRADIENTS EXPLODE - DESPITE BOUNDED ACTIVATION FUNCTIONS

The authors use the gradient scale coefficient (GSC) to test 7 different MLPs for presence of exploding gradients: one using ReLU; one with layer normalisation followed by ReLU; one with batch normalisation followed by ReLU; one using tanh; one with layer normalisation followed by tanh; one with batch normalisation followed by tanh and one with SeLU activations. Each network had 50 layers with 100 neurons each followed by an error layer that computes the dot product between the label and the prediction. For ReLU architectures, the weight matrices were initialised with variance $\frac{2}{100}$. Weight matrices for tanh and SeLU architectures were initialised with variance $\frac{1}{100}$. Both the inputs and labels were 100-dimensional vectors where each element was drawn from an independent Gaussian distribution with mean 0 and variance $\frac{1}{100}$. The input vectors were normalised to

have length 10. The authors drew 100 independent datasets of size 10000, however we only drew 10 datasets of size 100 in the interest of speed.

At each layer l the authors computed the pre-activation standard deviation, the pre-activation sign diversity and the expected gradient scale coefficient between layer l and layer 0 (the dot product error layer), which is given by

$$GSC(l,0) = \frac{\mathbb{Q}||\mathcal{J}_l^0||_{qm}\mathbb{Q}||f_l||_2}{\mathbb{Q}||f_0||_2}$$

where $\mathbb{Q}[\cdot]$ is the quadratic expectation. Figure 1 contains the results for this experiment that were reported in the paper.

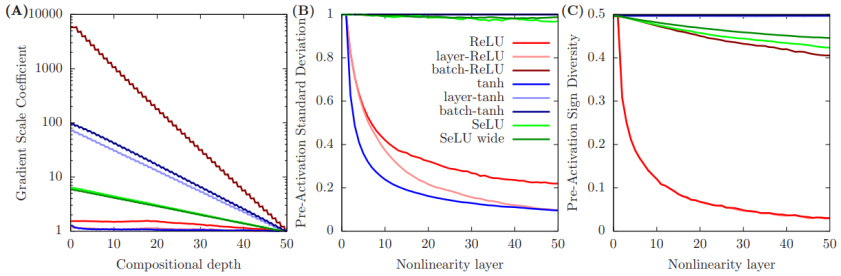

Figure 1: Metrics for the 7 MLPs evaluated on Gaussian noise. Taken from the original paper.

In figure 1(A), the GSC increased roughly exponentially as compositional depth decreased for the Batch-ReLU, Layer-tanh, Batch-tanh and SeLU MLPs. As a result, the authors concluded that these four MLPs had exploding gradients. Frustratingly, the authors don't discuss parts (B) and (C) of Figure 1 until 6 pages later. They claim that diminishing standard deviations and sign diversities are indicative of what they call the collapsing domain problem, which also makes training difficult. They note that all 3 MLPs that did not have exponentially increasing GSCs had diminishing pre-activation standard deviations. Figure 2 shows the results we obtained.

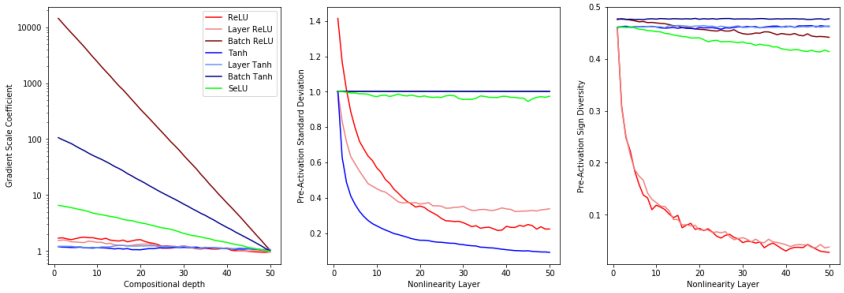

Figure 2: Our reproduction of the Gaussian noise experiment

Our results for the pre-activation standard deviations and sign diversities roughly match those reported in the paper. However, in our reproduction, the GSC for the Layer-tanh MLP did not increase exponentially. This suggests that our Layer-tanh MLP did not have exploding gradients. Its relatively constant pre-activation standard deviations and sign-diversities suggest that it also does not suffer from the collapsing domain problem. This result is very peculiar, since common intuition suggests that a regular 50 layer MLP with tanh and layer normalisation would be difficult to train and therefore would most likely exhibit signs of either exploding gradients or a collapsing domain. Other than this result, our reproduction of Figure 1(A) matches the original closely.

When we reproduced this experiment we found that calculating the GSCs for 100 datasets of 10000 observations was impractical for our setup. It took us roughly 30 minutes to run the experiment with 10 datasets of 100 observations (1000 times fewer total observations than in the paper). The costly part of the experiment was computing the Jacobians of the loss with respect to each layer. Bizarrely, we found that disabling GPU acceleration and running on CPU made this experiment faster.

3.2 EXPLODING GRADIENTS LIMIT DEPTH - EFFECTIVE DEPTH

The authors introduce the concept of effective depth in a network. They state that by the time the gradient has propagated back to the first layer, it will be a sum of terms that are almost all a product of $\frac{L}{2}$ residual Jacobian matrices. If the operator norm of the residual Jacobians is less than p for $p < 1$, then the norm of products of residual Jacobians will decrease exponentially in the number of residual Jacobians they contain. They claim that the information required for sets of λ or more layers to co-adapt during training is contained in terms that are a product of at least λ residual Jacobians. Therefore, they claim that the effective depth of a network is the largest λ such that the sum of all terms containing λ residual Jacobians is non-negligible. The authors try to show that if a network has exploding gradients as shown by GSC, the number of updates required to train that network to have effective depth λ will increase exponentially. Hence, exploding gradients limit effective depth.

To test this idea, the authors trained the four exploding architectures from the previous experiment on the CIFAR10 dataset. Each MLP contained 51 linear layers with 100 neurons each, except for the input and output layers. A softmax was placed just before the error layer. The authors used a KL error layer, which computes the logarithm of the predicted probability of the correct class. They tested several initial learning rates to find the optimal learning rate for each network. The given set of step sizes given for testing were $\{1e5, 3e4, 1e4, 3e3..., 1e{-}4, 3e{-}5, 1e{-}5\}$. For each of these step sizes, each network was trained for 500 epochs where if after 5 epochs, the training error did not decrease the learning rate was multiplied by 0.333. In the end, the step size that achieved the lowest training error was chosen as optimal.

Figure 3(B), (C) and (D) show how classification error, $GSC(L, 0)$ and effective depth changed over 500 epochs of training. Figure 3(E) shows the operator norm of the difference between the trained weight matrices and initial weight matrices at each layer.

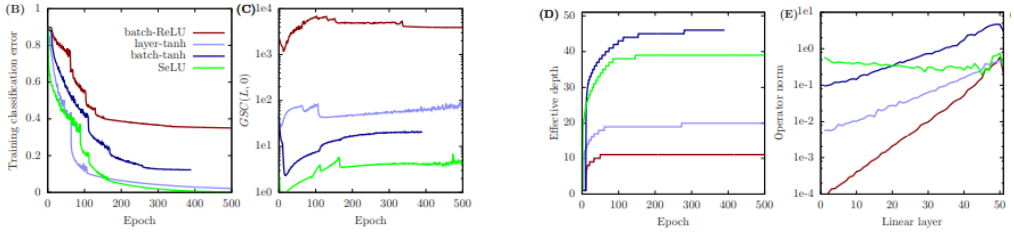

Figure 3: CIFAR10 Paper Results

We ran into difficulties when reproducing this experiment. To get the optimal initial learning rates for each network, we tested each one for 70 epochs instead of 500 as doing the full 500 would have been impractical with our setup. The authors tested initial learning rates as high as $1e5$, which seemed like a waste of time. The high initial learning rates caused the loss to increase and to eventually become a NaN as one would expect. The authors never explained why they used the KL error layer instead of categorical cross entropy. It seems like an inferior choice, since the KL loss only depends on one of the predicted probabilities, whereas categorical cross entropy depends on all 10 predicted probabilities and therefore weight updates computed using cross entropy will adjust all 10 predictions instead of just one each update. Using all 50000 images in the training set to calculate $GSC(L, 0)$ and calculate the effective depth was far too time consuming on our setup and in some cases used up all of our RAM, causing our sessions to crash. We suspect that attempting to store all the information from a backward pass as well as the Jacobians between layers used to compute the GSCs were what caused these memory issues. We settled on using only 100 images to compute $GSC(L, 0)$ for each network and effective depth for only SeLU. Additionally, we only trained each network for 100 epochs instead of 500, however this experiment still took us several hours to run. Figure 4 shows the results we obtained.

Our classification errors over 100 epochs for batch-tanh and SeLU are similar to those obtained in the paper, except swapped around. We had significantly higher error with batch-ReLU. These discrepancies may have happened because we used 70 epochs to find optimal initial step sizes. After around 80 epochs, the GSCs for layer-tanh became 0 and the error became 100%. We are not sure why this happened. Our $GSC(L, 0)$ results do not match those reported in the paper. From their

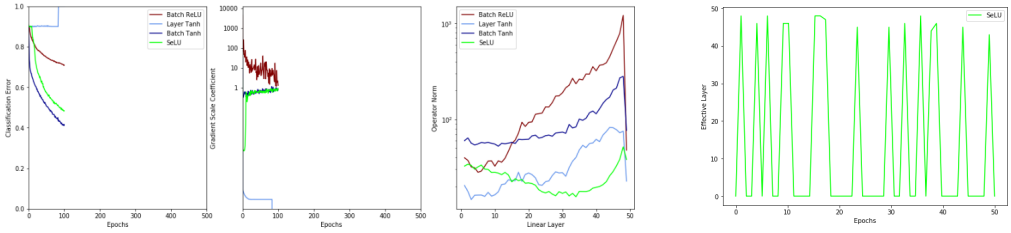

Figure 4: CIFAR10 Reproduced Results

results, the authors conclude that when layers are trained jointly, there is no evidence that the GSC increases or decreases during training. However, our results (apart from layer-tanh) seem to suggest that the GSC tends to converge towards 1 during training. In addition, in architectures where the GSC converged to roughly 1, classification error decreased faster. This may be coincidental, but is an interesting pattern nonetheless. Our effective depth results were different. The values of the residual Jacobian matrices depend on the inputs, and vary more if fewer images are used. This noisiness may have prevented us from observing a consistent increase in the effective depth throughout training, however the effective depth never reached the total amount of layers which supports the idea that layers cannot co-adapt for sufficiently deep networks when exploding gradients are present. The operator norms we attained were not completely the same, but followed the same trends. Each architecture except SeLU showed a downward trend from the error layer, but at much higher values than in the paper. The weights of the later layers changed much more than the earlier layers during training. This suggests that if these exploding networks were to become any deeper, their effective depth would not increase.

4 Conclusion

This work finds that the GSC is a suitable metric for identifying exploding gradients in MLPs. When monitored throughout training, it may offer an insight into how well training is going. Our CIFAR results support the idea that exploding gradients can limit the effective depth of MLPs. All the experiments we reproduced used 50 layer MLPs, which are not used in practice on CIFAR10 or indeed for anything else. As a result, a couple of reviews raised concerns about how valid the conclusions of this paper are. However, we side with the authors on this issue. We're not interested in how practical the networks are. Instead, we're interested in testing whether the GSC can detect exploding gradients in networks that we know have exploding gradients.

The authors did not release the code used to run these experiments so we had to reproduce everything from scratch. We ran our experiments in Google Colab with GPU acceleration. The layout and length of the paper caused difficulties in understanding and reproducing experiments from the paper. Although once we found them, we thought that the explanations of gradient scale coefficients, the collapsing domain problem, the trade off between exploding gradients and collapsing domains, and the problems that these issues can cause were well done. We think that if this paper had been split into two papers: one on GSCs, exploding gradients and collapsing domains and a follow up paper on the residual trick and the effect of skip connections on the GSC, it would have been better received by the reviewers who complained (justifiably) that the paper was too long and might have been better at convincing future research to use the GSC as a metric. The majority of the results are reproducible, although some experiments demanded better hardware than we had access to and some architectures behaved differently than reported. Our code, including code for additional experiments, can be found at `https://github.com/JeremyLinux/ICLR-Reproducability-Challenge`

References

Hrushikesh N Mhaskar and Tomaso Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14 (06):829–848, 2016.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pp. 2924–2932, 2014.

George Philipp, Dawn Song, and Jaime G Carbonell. Gradients explode-deep networks are shallow-resnet explained. 2018.