1 Introduction

Bla bla bla competition details, why were we interested, outline of how we are going to work

2 Maximizing initial network performance

The first step in this competition is to obtain a network with a very high performance, independent from its size. This can be divided in five different areas:

1. Scalable architecture design
2. Pretraining with other datasets (e.g imagenet)
3. Agressive regularization (e.g dropout)
4. Extreme data augmentation (e.g cutout)
5. Network ensembling

Unfortunately the second one is not allowed in the competition. We will try to avoid option 3 as We will also let option 4 for the end, as it is basically a brute force approach to the problem.

2.1 Scalable architecture design

The main idea in finding networks with very high performance is to use scalable architecture design, which can be searched with a limit amount of parameters and then scaled to achieve better performance. Initially we are going to use Resnets/Wide-Resnet, but the ideal would be to profit from more recent works such as efficient nets.

Decision: WideResNet 110-10

2.2 Extreme data augmentation

Data augmentation has been shown to greatly increase the performance of networks. Two main types of data augmentation have been proposed. First to profit from the fact that we use images and therefore we know a lot of transformations that can be used without altering the class of the input. For this we are going to use recent works such as Cutout and Google AutoAugment. Second

there are methods that use data augmentation as a proxy to better define the classification boundaries, most notably Mixup based methods where a linear combination of two inputs (in any part of the network) leads to a equal linear combination of the outputs.

Decision: Cutout, AutoAugment and Mixup

2.3 Initial network result

Applying all the techniques listed above (WideResNet 110-10 with Cutout, AutoAugment and Mixup) we where able to obtain a network with $X\%$ performance on the CIFAR-100 dataset, with Y flops and Z parameters, which at the end leads to a score of $A + B = C$.

Maybe here it would be cool to have a graph of the score/performance of different "teacher" networks

3 Decreasing network size

The networks trained using the techniques presented in the last section will be (by design) too big to be competitive. The idea now is to reduce the size of the network. We are going to use 4 techniques for decreasing the network size:

1. Student/Teacher networks
2. Replacing KxK convolution by shift + 1x1
3. Binarization
4. Pruning

The idea is to do this step by step, **This will only work if we believe that simple KD will be strong enough to make the student of the student have the same performance.**

3.1 Student/Teacher networks

Now that we have a very strong baseline teacher, we can use it to train smaller networks. We are going to base our efforts here in the LIT technique, which allows us to reduce the amount of layers by trying to imitate the output of the teacher blocks (fitnet technique), while using the teacher outputs to stabilize training. A second step that we would like to add would be to reduce the size of the layers by applying the fitnet technique using to where the blocks will try to reproduce the space distribution of the teacher (Loss based on the distance between similarity matrices of the batches).

The first step allowed us to reduce the WideResNet 110-10 to a WideResNet 28-10. This reduced the performance from X % to $X - \epsilon_1$ % while reducing the flops to $Y - huge$ and parameters to $Z - huge$ which leads to a score of $A_2 + B_2 = C_2 < C$

The second step allowed us to reduce the WideResNet 28-10 to a WideResNet 28-1. This reduced the performance from $X - \epsilon_1\%$ to $X - \epsilon_1 - \epsilon_2\%$ while reducing the flops to $Y - huge - huge$ and parameters to $Z - huge - huge$ which leads to a score of $A_3 + B_3 = C_3 < C_2$

3.2 Replacing KxK convolution by shift + 1x1

Now our scores start to be competitive, but there is still a lot one can do. Recently a lot of works have shown that we can trade the 3x3 convolutions by a shift and a 1x1 convolution. We are going to use the SAL method to do this.

This allows us to reduce the WideResNet 28-1 to 2-SAL-WideResNet 28-1.

3.3 Binarization

BWN

3.4 Pruning

Our performance is still better than the 80% needed to qualify for the competition, therefore we can use the X pruning technique to remove some parameters of our network while keeping the performance high enough.