PRE-REPORT: VERY DEEP CONVOLUTION NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

SongSeungWu

1 ABSTRACT

This study investigates the impact of network depth on accuracy in large-scale image recognition settings. The main objective of the research is to evaluate how the performance of the model changes as the depth increases. The researchers constructed a very deep network with 3 x 3 convolution filters and evaluated networks with 16 to 19 weight layers. The results show significant performance improvements over previous techniques.

- Weight Layers: Layers in the neural network responsible for performing linear transformations between the input and output.

2 INTRODUCTION

Various attempts have been made to improve the original ConvNet architecture, including reducing the receptive window size and lowering the stride in the initial convolutional layers. This paper focuses on increasing the depth of the network by fixing other architectural parameters and stacking multiple convolutional layers. By using small 3 x 3 convolution filters, it was possible to increase the depth without a substantial increase in parameters, which resulted in a much more accurate ConvNet architecture.

3 ARCHITECTURE

The network architecture takes 224 x 224 x RGB images as fixed-size input. Preprocessing includes subtracting the mean RGB value computed from the training set. Convolution operations are performed with a stride of 1, and padding is applied to preserve spatial resolution. The filter size is 3 x 3, and Max-pooling is applied with a 2 x 2 window and stride of 2..The network is followed by three fully connected (FC) layers, where the first two FC layers have 4096 channels, and the third FC layer has 1000 channels corresponding to the 1000-way ILSVRC classification. Finally, a softmax layer is used for classification. All hidden layers use ReLU activation functions, and LRN normalization is excluded in most networks.

4 CONFIGURATIONS

The ConvNet configurations proposed in this paper follow a standard design, with variations in depth. The number of channels in the convolution layers starts at 64 and doubles after each Max-pooling operation, reaching 512 channels at the deepest layers. Despite the increase in depth, the number of parameters does not change significantly, thanks to the use of small filters. This approach focuses on how filter size and network depth affect performance.

5 DISCUSSION

The paper uses 3 x 3 receptive fields, which allows for better decision function approximation through multiple layers of ReLU activations. This method also reduces the number of parameters while enhancing generalization capabilities. The inclusion of 1 x 1 convolution layers increases the non-linearity of the decision functions without affecting the receptive field size.

6 TRAINING

The network was trained using mini-batch gradient descent with momentum set to 0.9. The weight decay was set to 0.0005, and dropout of 0.5 was applied to the first two FC layers. The learning rate was initially set to 0.01 and decreased by a factor of 10 when the validation accuracy plateaued. Training was stopped after 74 epochs.

Initialization was crucial for stable training, as improper initialization could lead to gradient instability in deep networks. The weights were initialized using a normal distribution with a mean of 0 and variance of 0.01.

7 TESTING

During testing, images were resized using a test scale Q, which did not have to match the training scale S. Additionally, the fully connected layers were replaced with convolutional layers to create a fully convolutional network, allowing spatial averaging across channels for score calculation. This approach eliminated the need for multiple crops, though multiple-crop evaluation was found to improve performance.

8 CLASSIFICATION EXPERIMENTS

The study used the ImageNet dataset, consisting of 1,000 classes and over 1.3 million training images, 50,000 validation images, and 100,000 test images. Performance was evaluated using Top-1 and Top-5 error rates:

- Top-1 Error Rate: The proportion of images incorrectly classified.
- Top-5 Error Rate: The probability that the true label is not among the top 5 predicted categories.

9 SINGLE SCALE EVALUATION

The performance of individual ConvNet models was evaluated on a single scale, showing that LRN normalization did not significantly improve the accuracy of model A without it. As the depth of the ConvNet increased, classification errors decreased. Networks with small filters and greater depth outperformed shallower networks with larger filters. Additionally, scale jittering during training produced better results than using images of fixed scale.

10 MULTI-SCALE EVALUATION & MULTI-CROP EVALUATION

The effect of scale jittering during testing was evaluated by adjusting the test images to various sizes before averaging the results for prediction. The study concluded that scale jittering during testing led to better performance, particularly for the deeper networks (D and E).

In multi-crop evaluation, performance was compared with dense evaluation, with multi-crop performing slightly better. Combining both methods resulted in optimal performance.

11 CONVNET FUSION

Model fusion was performed by averaging the softmax outputs of multiple models. This technique enhanced performance due to the complementary nature of the models.

12 CONCLUSION

This paper evaluated very deep convolutional networks for large-scale image classification and demonstrated that representation depth is beneficial for classification accuracy. By stacking more layers in a standard ConvNet architecture, the model achieved state-of-the-art performance in the ImageNet competition. The results reaffirm the importance of depth in visual representation learning, making VGGNet (with up to 19 layers) a significant contribution to deep learning research.

PRE-REPORT: Deep Residual Learning for Image Recognition

SongSeungWu

1 ABSTRACT

Training deep neural networks is challenging, but residual networks (ResNets) make it easier by optimizing through learning residual functions. This approach allows for the construction of much deeper networks with improved accuracy. In experiments on the ImageNet dataset, a 152-layer ResNet, which is 8× deeper than VGG nets but with lower complexity, achieved a 3.57% error rate and won 1st place in the ILSVRC 2015 classification task.

2 INTRODUCTION

Recent research highlights the critical importance of network depth in improving performance on tasks such as ImageNet classification, where models with 16 to 30 layers have consistently achieved top results. However, the question arises: Is building better networks simply a matter of adding more layers?

Two major challenges hinder deeper network training:

1. Convergence Problem: This problem arises due to vanishing or exploding gradients, which slow or block the training process from the start. This has largely been mitigated by techniques like normalized initialization and intermediate normalization layers, allowing deeper networks to converge during training using SGD with backpropagation.

2. Degradation Problem: As network depth increases, accuracy can first saturate and then degrade unexpectedly, even without overfitting. Adding more layers leads to higher training error, as observed in multiple experiments. This issue highlights that deeper networks are not always as easy to optimize as their shallower counterparts

3 RELATED WORK

1. Residual Representation: In vector quantization, encoding residual vectors is more effective than encoding original vectors. Techniques like good reformulation or preconditioning can simplify the optimization process, leading to more efficient learning and faster convergence.

2. Shortcut Connection: In highway networks, shortcut connections are paired with gating functions that are data-dependent and parameterized. When the gate closes, the layers become non-residual functions. In contrast, residual networks use parameter-free identity shortcuts that always remain open, ensuring that residual learning continues without interruption. This allows for smoother optimization and better performance, especially in deeper networks.

4 RESIDUAL LEARNING

Let $H(x)$ represent the underlying mapping to be learned by a few stacked layers, with x as the input. Rather than directly approximating $H(x)$, residual learning reformulates the problem by focusing on learning the residual function $F(x) = H(x) - x$, which is then expressed as $H(x) = F(x) + x$ This reformulation simplifies learning because if identity mappings are optimal, the solver can easily drive the weights toward zero to approach identity mappings. Although identity mappings are unlikely to be the optimal solution in most cases, this reformulation serves as a useful preconditioning step. It allows the solver to learn small perturbations around an identity mapping, which is often easier than learning a completely new function from scratch. Experiments show that learned residual functions typically have small responses, indicating that identity mappings offer a good starting point for optimization.

5 NETWORK ARCHITECTURES

1. Plain Network: Inspired by VGG nets, this 34-layer model uses 3×3 filters and maintains complexity by doubling the filters when the feature map size is halved. It has 3.6 billion FLOPs, significantly less than VGG-19.

2. Residual Network: Built on the plain network with shortcut connections. Identity shortcuts are used when input and output dimensions match, and for increased dimensions, either zero-padding (no extra parameters) or projection shortcuts (1×1 convolutions) are applied.

6 Implementation

For ImageNet, images are resized and randomly cropped to 224x224 with standard color augmentation applied. Batch Normalization (BN) is used after each convolution and before activation. Training is done using SGD with a mini-batch size of 256, starting with a learning rate of 0.1 and reducing by 10 when the error plateaus. The training runs for 600,000 iterations with 0.0001 weight decay and 0.9 momentum. No dropout is used, and 10-crop testing is performed during evaluation.

7 Experiments

1. Plain Networks: The 34-layer plain network shows higher training error than the 18-layer network, indicating optimization difficulties not related to vanishing gradients. The 34-layer plain network may have a low convergence rate, despite maintaining competitive accuracy.

2. Residual Networks: Residual networks with identity shortcuts exhibit better performance, with the 34-layer ResNet outperforming the 18-layer ResNet by reducing the training error and achieving higher accuracy. Residual learning helps overcome degradation and provides accuracy gains with increased depth.

Three shortcut options were compared:

1. (A): Zero-padding identity shortcuts.

2. (B): Projection shortcuts for dimension increases.

3. (C): Projection shortcuts for all cases.

Projection shortcuts (B and C) offer slight improvements over zero-padding, but the differences are minimal, showing that identity shortcuts are effective for reducing complexity without impacting performance.

50/101/152-layer ResNets were created using a bottleneck design (1x1, 3x3, 1x1 convolutions). These deeper networks do not show degradation and achieve significant accuracy gains, outperforming VGG nets with lower computational complexity.

ResNets overcome optimization difficulties in deep networks, demonstrating accuracy improvements as depth increases. 110-layer and 1202-layer ResNets were tested, with the deeper network achieving a training error of less than 0.1%, though some overfitting occurred on this small dataset.

Using ResNet-101 with Faster R-CNN resulted in a 28% relative improvement on the COCO dataset. ResNets contributed to winning 1st place in several tasks at the ILSVRC & COCO 2015 competitions, including ImageNet detection, localization, and COCO segmentation.