Techniques for Image Classification on Tiny-ImageNet

Zachary Barnes, Frank Cipollone, Tyler Romero - Stanford University

Motivation

Improvement in image classification is a fundamental goal of computer vision and machine learning.

Data

Tiny-ImageNet Dataset[1]

- ImageNet images cropped and scaled down to 64x64x3
- 100,000 train images
- 10,000 val, and 10,000 test images
- Accuracies tend to be much lower than on ImageNet due to the granularity of the scaled images, and poor cropping

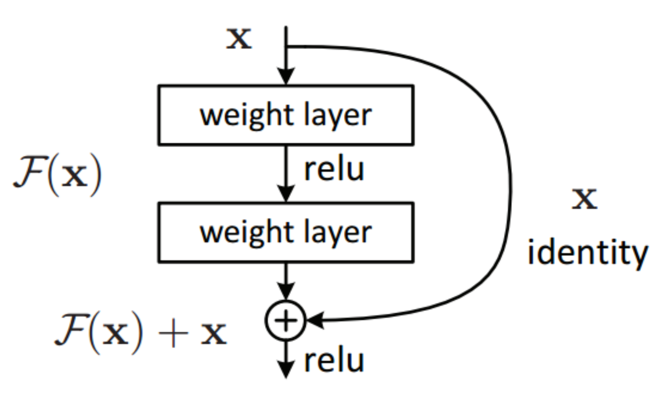
Models

AlexNet[2]

- Uses convolutional layers, max pooling, and fully connected layers to map the input image to a class
- Adjusted for the Tiny-ImageNet dataset by setting the stride of the first conv layer to 1, and by removing a max pooling layer

ResNet[3]

- Uses residual blocks with identity mapping
- Residual connections allow gradients to propagate easily
- ResNet[N] corresponds to a Residual Network with a depth of N



A Residual Block[3]

WideResNet[4]

- Uses an empirically better residual block
- WideResNet[N]-[K] corresponds to a residual network with depth N that has K times more filters than a standard ResNet

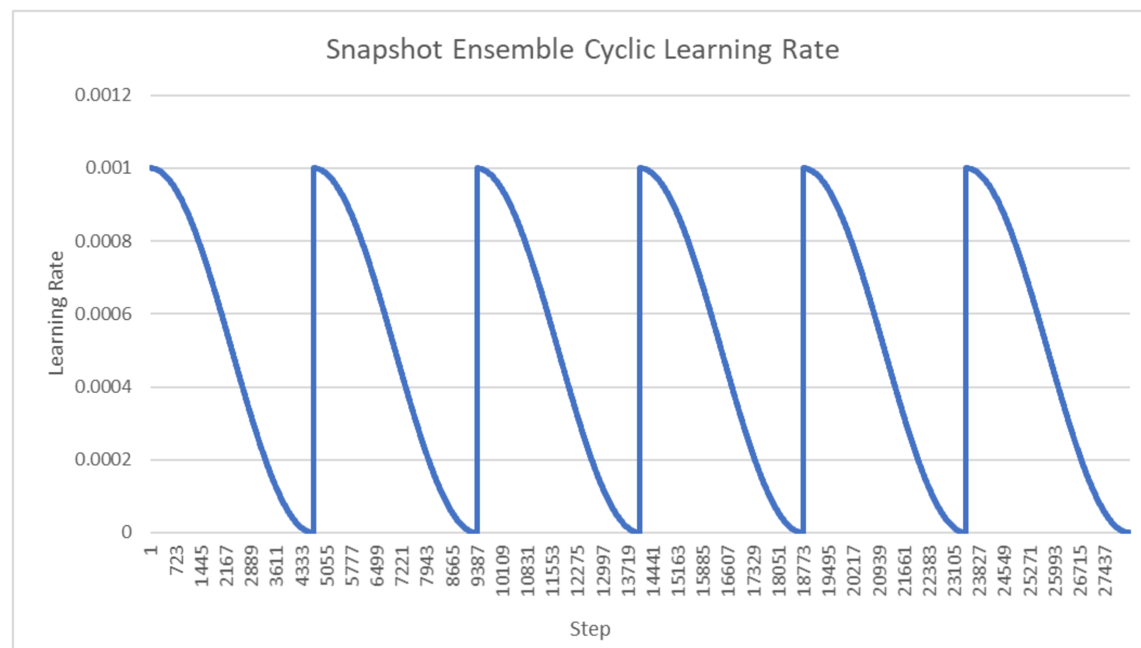
Towards Higher Accuracy

Regularization

- Data Augmentation
  - Randomly crop each image to size 56x56x3
  - With probability of 0.5 we horizontally flip each image
  - Predictions made using 10-Crop averages[2]
- Smaller Models
  - Shallower, thinner models provide implicit regularization
- Weight Decay
  - Standard L2 weight decay has been show to help with both training and validation accuracy for image classification[2]

Snapshot Ensembles using Cyclic Learning Rates[5]

- Use cyclic learning rates in order to train several models with one training pass – allows for ensembles to be created very quickly
- We trained snapshot ensembles using 6 cycles of 12 epochs each over a total of 72 epochs
- The model generated by the first cycle is thrown out because it tends to perform worse than later models
- Weighted averages are used to combine model predictions



Results

Model	Top-5 Val	Top-1 Val	Top-1 Test
AlexNet	0.487	0.254	-
AlexNet + a	0.717	0.487	-
ResNet34 + a	0.734	0.523	-
WideResNet28-10 + a + d	0.773	0.564	-
WideResNet32-4 + a + d	0.778	0.571	-
WideResNet32-4 + a + d + snap	0.803	0.595	-
ResNet18 + a + d	0.795	0.589	-
ResNet18 + a + d + snap	0.814	0.602	0.536

a = data augmentation d = weight decay snap = snapshot ensemble

All values are accuracy rates

Experimentation Details

- Adam Optimizer
- Initial learning rate of 0.001
- Learning rate divided by 10 every 24 epochs
- 72 epochs of train time
- Early stopping to help prevent overfitting

Discussion

Models

- ResNet models outperformed AlexNet models; they also were easier to train
- Deeper Models offered little improvement
  - Postulated that this is due to residual connections allowing layers to be entirely skipped[4]

Regularization

- Significant data augmentation required to prevent heavy overfitting
- Small weight decay coefficients tend to help the model achieve higher validation accuracy
- Our smallest model (ResNet18) performed the best, indicating that our other models overfit due to their size

Snapshot Ensembles

- Snapshot ensembles outperform their corresponding models trained using standard methods and the same training budget
- We found that within three cycles (36 epochs), snapshot modules would have a higher validation accuracy than our traditionally trained models would after 72 epochs

References

1. "Tiny ImageNet Visual Recognition Challenge." Tiny ImageNet Visual Recognition Challenge. N.p., n.d. Web. 05 June 2017.
2. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.
3. He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
4. Zagoruyko, Sergey, and Nikos Komodakis. "Wide residual networks." *arXiv preprint arXiv:1605.07146* (2016).
5. Huang, Gao, et al. "Snapshot ensembles: Train 1, get m for free." *arXiv preprint arXiv:1704.00109* (2017).