Residual Networks on Tiny ImageNet

Lei Sun

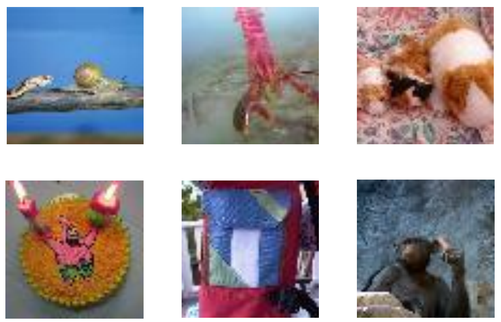
Stanford ENGINEERING Computer Science

Stanford University

Introduction

Deep Residual Networks has been proven to be a very successful model on image classification. A ResNet with 152 layers achieved the best results of 3% error rate, which is even better than human judges. In this project, we trained a state-of-art ResNet with various techniques on tiny ImagetNet. It achieves 45% error rates, ranking the 3rd on the evaluation server (By now)

Examples in Tiny ImageNet



Tiny ImageNet

For this project, we will use the Tiny ImageNet - a subset of ILSVRC. It follows the same principle, though on a much smaller scale:

- 200 different categories
- 500 training images in each (100,000 in total)
- 50 validation images in each (10,000 in total)
- 50 test images (10,000 in total)
- Re-sized to 64x64 pixels (256x256 pixels in standard ImageNet).

Resnet and Model #1

A layer in a traditional neural network learns to calculate a function

$$y = f(x)$$

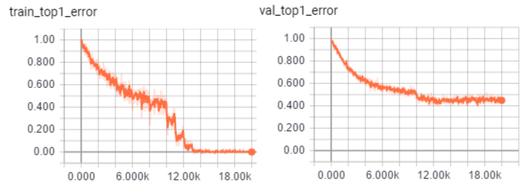
A residual neural network layer approximately calculates

$$y = f(x) + id(x) = f(x) + x.$$

Identity connections enable the layers to learn incremental, or residual, representations. Those shortcuts act like highways and the gradients can easily flowback, resulting in faster training and much more layers.

Result #1

Training Err = 0!
Val Err = 50% (Test Err 48%)
OVERFITTING!

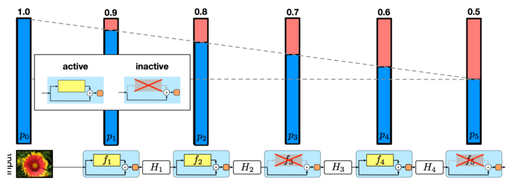


Improved Model #2

So far, the biggest challenge in this project is that we only have 500 training images for each category. During the training process, our model sees the same set of images again and again. To avoid overfitting, the following improvement are made.

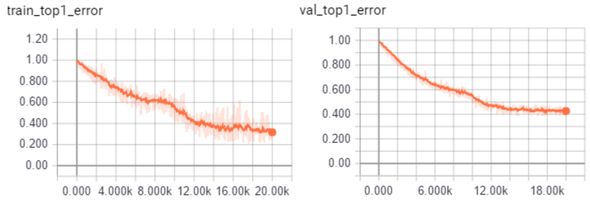
Stochastic Depth

A modified version of classic dropout. Entire layers get dropped randomly instead of nodes within layers.



Improved Results #2

Training Err = ~35%
Val Err = ~40% (Test Err 43%)
Less overfitting
But similar performance

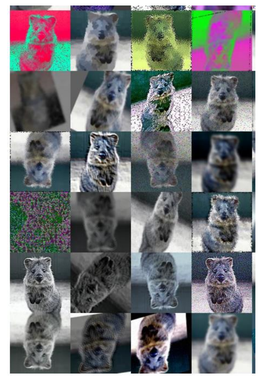


More is coming (Model #3)

Even more Image Augmentation

In order to make the best use of our training data, we augmented them via a number of random transformations

- Invert
- Brightness
- Noise
- Flip
- Rotate
- Scale
- Shift
- Shear



Model #3 is still under training, I expect to have a performance boost on it... Will include it in final report.