Assignment 2

Lei Chen
CSCI-GA 2271 Computer Vision

October 25, 2018

The best test accuracy I have achieved is 0.98970 on the public test dataset. However, since it was many days ago when achieving such a result without storing relevant files, I re-trained an exactly same model with test accuracy of 0.98764. The only difference is the random seed.

1 Work

After training over 70 models, my models can be divided into four categories, Shallow ConvNet with Concatenation, Deep ConvNet with BatchNorm, ConvNet with Inception Modules and ResNet.

1.1 Techniques Used

Data Preprocessing. For comparatively shallow models, data is cropped to size of (24,24) after resizing to (32,32) to eliminate noise from corners. Deep models still use data of size(32,32).
Batch Normalization. For Deep ConvNets and ResNets, each convolutional layer is followed by a BatchNorm layer to enhance convergence and work against overfitting. It is also used for concatenation intuitively in Inception module.
Sptial Transformer. It is used in comparatively shallow network to get better resampling of data with basic transforming. Effects in deep networks are waiting for further verification.

1.2 Model details

1.2.1 Shallow ConvNet with Concatenation (SCNC)

The model achieves a test accuracy of without little fine-tuning.
The model takes STN and data of size (24,24). The first convolutional layer uses filters of 3x3, 5x5 and 7x7. The second convolutional layer uses filters of 3x3 and 5x5. Both are followed by concatenation and BatchNorm. Following ConvLayers are an max-pooling layer and two full-connected layer.
The model is stored as *model_scnc.py*. It is a CUDA version and, if anyone would like to run it with cpu, just delete ".cuda()".

1.2.2 Deep ConvNet with BatchNorm (DCNBN)

The model achieves a test accuracy of 0.98970. However, files of acc 0.98764 is provided instead.
The model takes data of size (32,32). It consists of 19 convolutional layers, each followed by a BatchNorm layer. Following ConvLayers are an average layer and two full-connected layer.

1.2.3 ConvNet with Inception Modules (CNIM)

The model achieves a test accuracy of 0.98052 with little fine-tuning.
The model takes STN and data of size (24,24). One modules of Inception are included, consisting of four columns. The module is shown below.

Layer#	Param	Input	Output
Conv1-1	1 x 1 x 16	24 x 24 x 3	24 x 24 x 16
Conv1-2-maxpool	3 x 3	24 x 24 x 3	24 x 24 x 3
Conv1-2	1 x 1 x 16	24 x 24 x 3	24 x 24 x 16
Conv1-3-1	1 x 1 x 8	24 x 24 x 3	24 x 24 x 8
Conv1-3-2	3 x 3 x 16	24 x 24 x 8	24 x 24 x 16
Conv1-4-1	1 x 1 x 4	24 x 24 x 3	24 x 24 x 4
Conv1-4-2	3 x 3 x 8	24 x 24 x 3	24 x 24 x 8
Conv1-4-3	3 x 3 x 16	24 x 24 x 8	24 x 24 16
Concate	-	(24 x 24 x16) x 4	24 x 24 x 64
BN	64	24 x 24 x 64	24 x 24 x 64
Maxpool	2 x 2	24 x 24 x 64	12 x 12 x 64

1.2.4 ResNet

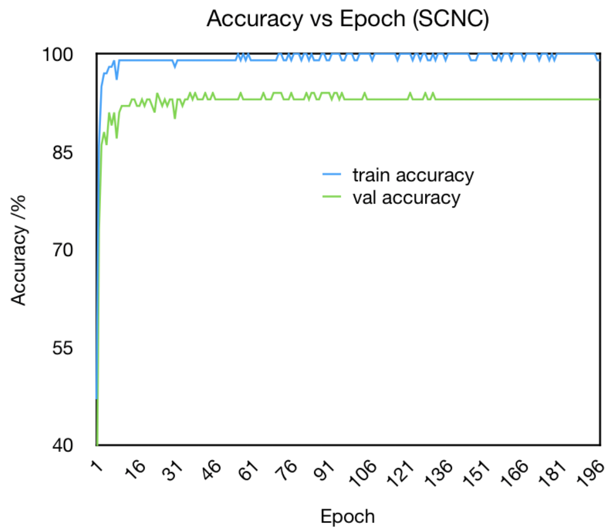
The model achieves a test accuracy of 0.98685. However, due to lack of time to re-train, only *model_res.py* is provided.
The model take data of size (32,32). It consists of 13 bottlenecks, each including kernels of 1x1, 3x3 and 1x1. On top of the network are an average pool layer and a full-connected layer.

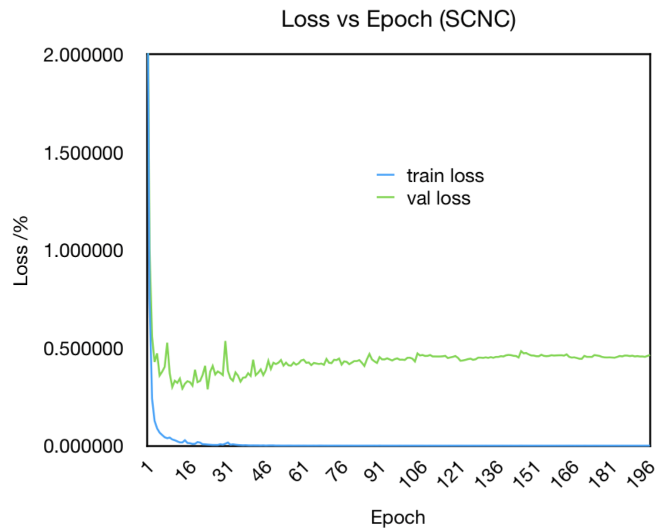
2 Result

Here is plots of accuracy, loss vs epoch of three models, SCNC, DCNBN, CNIM.

2.1 SCNC

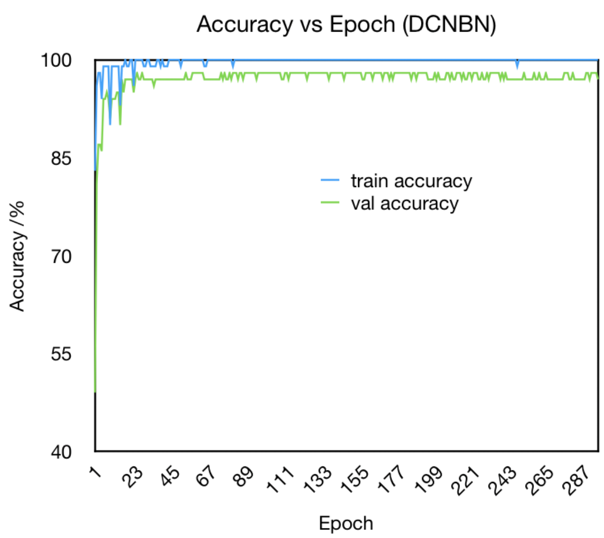
Test accuracy: 0.95977.

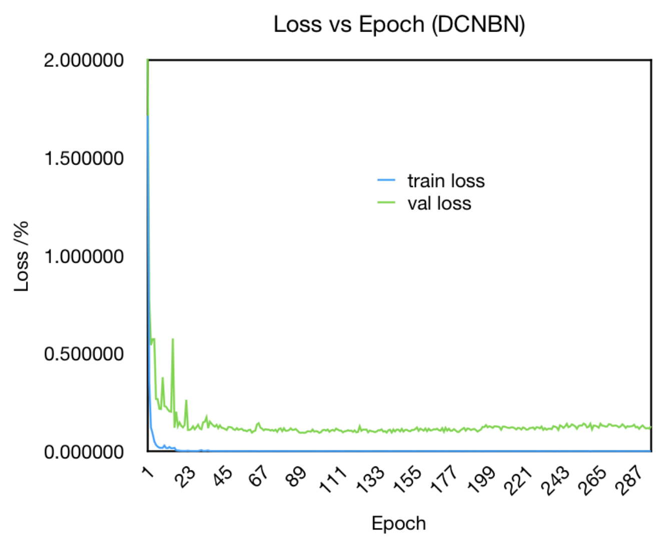




2.2 DCNBN

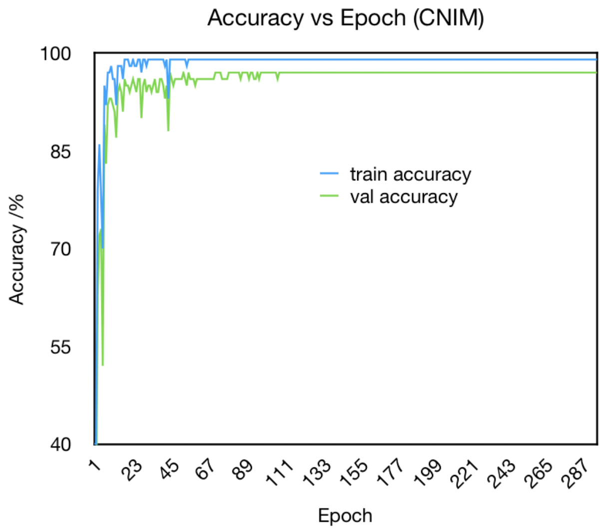
Test accuracy: 0.98764

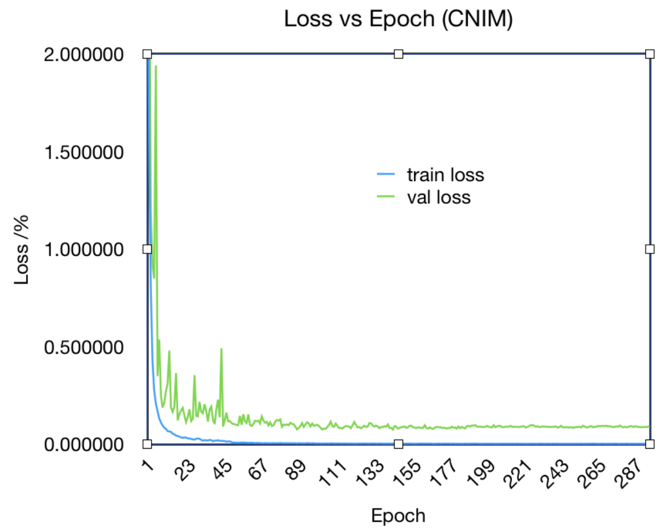




2.3 CNIM

Test accuracy: 0.98052.





3 Related Files

1. Four model files, *data.py* and *main.py* are included in the submission file.

2. Links to trained parameters of the three models on Prince are listed below.
   SCNC:
   /scratch/lc3909/assign2/model_cnn23_200.pth
   DCNBN:
   /scratch/lc3909/assign2/model_cnn18_300.pth
   CNIM:
   /scratch/lc3909/assign2/model_inc8_300.pth

   If you have any question about these files, feel free to contact me: *lc3909@nyu.edu*

4 Reference

1. Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).

2. Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

3. He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

4. Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems. 2015.