Homework 06 – Design

Arthur J. Redfern
arthur.redfern@utdallas.edu
Feb 25, 2019

---

0 Outline

1 Logistics

Assigned:	Mon Feb 25, 2019
Due:	Mon Mar 06, 2019
Format:	PDF uploaded to eLearning
	Python file that can be run in Google Colab

2 Reading

1. Read the class slides

Design
https://github.com/arthurredfern/UT-Dallas-CS-6301-CNNs/blob/master/Lectures/xNNs_06_Design.pdf

2. Read the ResNet arc of papers (a suggestion: set aside an hour of time and read all 3 of these together in 1 sitting)

Deep residual learning for image recognition
https://arxiv.org/abs/1512.03385
Identity mappings in deep residual networks
https://arxiv.org/abs/1603.05027

Aggregated residual transformations for deep neural networks
https://arxiv.org/abs/1611.05431

3. Read the following 2 neural architecture search papers (a suggestion: set aside an hour of time and read both of these together in 1 sitting)

Learning transferable architectures for scalable image recognition
https://arxiv.org/abs/1707.07012
MnasNet: platform-aware neural architecture search for mobile
https://arxiv.org/abs/1807.11626

4. Read and fully understand all lines of code for the following 2 examples

MNIST
https://github.com/arthurredfern/UT-Dallas-CS-6301-CNNs/blob/master/Code/xNNs_Code_01_Vision_Class_MNIST.py
CIFAR-10
https://github.com/arthurredfern/UT-Dallas-CS-6301-CNNs/blob/master/Code/xNNs_Code_02_Vision_Class_CIFAR.py

3 Theory

5. Using pencil and paper, compute the receptive field size at the input to the global average pooling layer for ResNet 50.

4 Practice

6. Create and train a "1/2 wide" version of the ResNet 4-6-3 model in the example CIFAR-10 code. 1/2 wide means that the number of feature maps is reduced by 1/2, the following is a comparison of the main path current width and 1/2 wide width that you'll create:

Block	Current width	1/2 wide
Tail	$3 \rightarrow 32$	$3 \rightarrow 16$
Level 0 special	$32 \rightarrow 64$	$16 \rightarrow 32$
Level 0 standard	64	32
Level 1 down sampling	$64 \rightarrow 128$	$32 \rightarrow 64$
Level 1 standard	128	64
Level 2 down sampling	$128 \rightarrow 256$	$64 \rightarrow 128$
Level 2 standard	256	128
Level 2 special	256	128

Note that the residual path width (not listed here) will also reduce by 1/2. You may have to modify the training hyper parameters.

After training the network, answer the following:

- How does the accuracy of the 1/2 wide version compare to the original version on this problem with this training method?
- Approximately how long (wall clock) does an epoch of training take for the original version and how long does an epoch of training take for the 1/2 wide version? A suggestion: compare the time that epoch number ~ 3 takes to eliminate startup effects.
- Mathematically approximate (in your head, no pencil and paper or computer calculation needed) how the following items change if the network width is reduced by 1/2: feature map memory, filter memory and compute.

7. Similar to the ResNet 4-6-3 example for CIFAR-10, use pencil and paper to plan out your own version of a popular network for CIFAR-10 by doing the following:

- Choose 1 of the following networks: MobileNet V1, MobileNet V2, ResNeXt, Inception V4, NASNet, MnasNet or AmoebaNet.
- Draw out the network structure and each of the basic building blocks.
    - These networks were originally designed for 3 x 224 x 224 images in ImageNet, so you'll likely replace portions of the network until ~ after the 3rd level of down sampling with a simple tail.
    - Skipping these initial levels will also make the network less "wide" than the original ImageNet optimized version.
    - The final feature map before global average pooling should be ~ N x 8 x 8 where N is > the number of classes (maybe >>).
    - Pay careful attention to any places where multiple paths add together and make sure that the ranges of both paths is compatible.
    - Take inspiration from the ResNet example.
- Compute the receptive field size at the feature map before the global average pooling layer. How does this compare to the original image size?
- Compute the feature map size and feature map memory required for each of the linear transforms. What is the maximum feature map size (this will set the optimal on device memory size)?
- Compute the filter coefficient size and filter coefficient memory required for a complete block in each of the levels. Which level has the largest filter memory in a block? Which level has the smallest filter memory in a block?
- Compute the MACs required for a complete block in each of the levels. Which level has the largest number of MACs in a block? Which level has the smallest number of MACs in a block?

- From the perspective of increasing receptive field size, minimizing filter memory and minimizing MACs, which level is best to repeat blocks within?

8. In software, implement the pencil and paper designed network from above. Ideally, create the network using a generator such that blocks can be repeated different numbers of times to build larger or smaller versions of the network.

9. Train the network and report the accuracy. You may have to modify the training hyper parameters.

10. [Optional] The following is a laundry list of additional items to consider trying

- Modify the network to add squeeze and excite style feature map re weighting
- Repeat blocks at different levels different numbers of times and record the accuracy of the trained network; create an optimal frontier of accuracy vs MACs and filter memory
- Add better variable naming and scoping for TensorBoard
- Go 1 level down in the TensorFlow API and do everything with the nn operators