Homework 06 – Design

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

0 Outline

1 Logistics
2 Reading
3 Theory
4 Practice

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

Complete

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

Complete

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

Complete

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

Complete

3 Theory

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

			Tracking
Start value			1
Level 5 standard	(+0, +2, +0)	repeat 2 times	5
Level 4 – 5 down sample	(+0, +2, *2, +0)		14
Level 4 standard	(+0, +2, +0)	repeat 5 times	24
Level 3 – 4 down sample	(+0, +2, *2, +0)		52
Level 3 standard	(+0, +2, +0)	repeat 3 times	58
Level 2 – 3 down sample	(+0, +2, *2, +0)		120
Level 2 standard	(+0, +2, +0)	repeat 3 times	126

Level 1 – 2 down sample	(*2, +2)	254
Level 0 – 1 down sample	(*2, +6)	514

Answer = 514

Some comments

- If the input image is smaller than 514 pixels on each side then effectively this is saying that the receptive field covers the full image
- The energy aggregation is not uniform over the full receptive field; the shape looks ~ Gaussian like as a consequence of the central limit theorem and convolving filters, so more energy is concentrated in a narrower lobe within the 514 pixel window, maybe 514 / 3 ~ 171 pixels
- When working on vision problems that involve localization and don't include a global average pooling operation ask yourself: how big is the receptive field or receptive field / 3 relative to objects of interest in the image? If I was looking at the image through a window of this size, could I easily identify the underlying object?
- To compute the receptive field for other places in the network, start with a value of 1 and work backwards using a subset of the same + and * operations as above

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 → 32	3 → 16
Level 0 special	32 → 64	16 → 32
Level 0 standard	64	32
Level 1 down sampling	64 → 128	32 → 64
Level 1 standard	128	64
Level 2 down sampling	128 → 256	64 → 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?

- Answer = ?

- 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.

  Answer = ?

- 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.

  Feature map memory reduces by ~ 1/2, filter memory reduces by ~ 1/4 and compute reduces by ~ 1/4; as systems are somewhere in between compute and data movement bound, it's expected that the per epoch training time reduces by 1/4 to 1/2, assuming a larger overhead factor is not in play

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.

  Choice = ?

- 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.

  Complete

- Compute the receptive field size at the feature map before the global average pooling layer. How does this compare to the original image size?

  Receptive field size = <network choice dependent>

- 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)?

  Max feature map size = <network choice dependent; likely towards the beginning>

- 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?

  Max filter memory size = <network choice dependent; likely towards the end>

- 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?

  Most MACs = <network choice dependent; likely towards the beginning>
  Least MACs = <network choice dependent; likely towards the end>

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

  Level = <network choice dependent; likely the level with 14x14 feature maps>

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.

   Complete

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

   Complete

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