

CS 6501-001

Deep Learning for Computer Graphics

Luyao Zhou

Lz6nf

Project 2 Deep learning for Image Colorization Brief Report

Part 1 Training on CPU:

Model 1:

Architecture:

This model has 5 SaptialConvolution layers and 5 SpatialFullConvolution layers, with batchNormalization inserted after each convolution. Each SaptialConvolution will half the resolution and each SpatialFullConvolution will double the resolution. The maximum of feature maps is 64.

```
-- create neural network
model = nn.Sequential() -- make a multi-layer perceptron
model:add(nn.SpatialConvolution(1, 2, 5, 5, 2, 2, 2, 2))
model:add(nn.SpatialBatchNormalization(2))
model:add(nn.ReLU(true))
model:add(nn.SpatialConvolution(2, 8, 5, 5, 2, 2, 2, 2))
model:add(nn.SpatialBatchNormalization(8))
model:add(nn.ReLU(true))
model:add(nn.SpatialConvolution(8, 16, 5, 5, 2, 2, 2, 2))
model:add(nn.SpatialBatchNormalization(16))
model:add(nn.ReLU(true))
model:add(nn.SpatialConvolution(16, 32, 5, 5, 2, 2, 2, 2))
model:add(nn.SpatialBatchNormalization(32))
model:add(nn.ReLU(true))
model:add(nn.SpatialConvolution(32, 64, 5, 5, 2, 2, 2, 2))
model:add(nn.SpatialBatchNormalization(64))
model:add(nn.ReLU(true))
model:add(nn.SpatialFullConvolution(64,32,2,2,2,2))
model:add(nn.SpatialBatchNormalization(32))
model:add(nn.ReLU(true))
model:add(nn.SpatialFullConvolution(32,16,2,2,2,2))
model:add(nn.SpatialBatchNormalization(16))
model:add(nn.ReLU(true))
model:add(nn.SpatialFullConvolution(16,8,2,2,2,2))
model:add(nn.SpatialBatchNormalization(8))
model:add(nn.ReLU(true))
model:add(nn.SpatialFullConvolution(8,4,2,2,2,2))
model:add(nn.SpatialBatchNormalization(4))
model:add(nn.ReLU(true))
model:add(nn.SpatialFullConvolution(4,2,2,2,2,2))
model:add(nn.SpatialBatchNormalization(2))
model:add(nn.ReLU(true))
```

Result:

This MSE for channel a* is 2.559097 and The MSE for channel b* is 6.414658

```
luyaaoz@luyaaoz-B85N-PHOENIX-WIFI: ~/Desktop/Graphic_proj2_lz6nf
epoch = 2; test mse = 33.839091
epoch = 3; train mse = 35.618117
epoch = 3; test mse = 33.840890
epoch = 4; train mse = 35.618083
epoch = 4; test mse = 33.798872
epoch = 5; train mse = 35.617670
epoch = 5; test mse = 33.842754
MSE for a*: 34.137632; MSE for b*: 33.547875
MSE = 2538.206543
luyaaoz@luyaaoz-B85N-PHOENIX-WIFI:~/Desktop/Graphic_proj2_lz6nf$ qlua part2.lua
ok
epoch = 1; train mse = 6.344403
epoch = 1; test mse = 5.316696
epoch = 2; train mse = 4.670516
epoch = 2; test mse = 5.226321
epoch = 3; train mse = 4.401128
epoch = 3; test mse = 4.754260
epoch = 4; train mse = 4.194165
epoch = 4; test mse = 4.126492
epoch = 5; train mse = 4.018300
epoch = 5; test mse = 4.486746
MSE for a*: 2.559097; MSE for b*: 6.414658
MSE = 4.486778
```

Model 2:

Architecture:

This model is similar to model 1, it only outputs a greater number of feature maps of each layer. This model has 5 SpatialConvolution layers and 5 SpatialFullConvolution layers, with batchNormalization inserted after each convolution. Each SpatialConvolution will half the resolution and each SpatialFullConvolution will double the resolution. The maximum of feature maps is 512.

```
-- create neural network
model = nn.Sequential() -- make a multi-layer perceptron
model:add(nn.SpatialConvolution(1, 2, 5, 5, 2, 2, 2, 2))
model:add(nn.SpatialBatchNormalization(2))
model:add(nn.ReLU(true))
model:add(nn.SpatialConvolution(2, 64, 5, 5, 2, 2, 2, 2))
model:add(nn.SpatialBatchNormalization(64))
model:add(nn.ReLU(true))
model:add(nn.SpatialConvolution(64, 256, 5, 5, 2, 2, 2, 2))
model:add(nn.SpatialBatchNormalization(256))
model:add(nn.ReLU(true))
model:add(nn.SpatialConvolution(256, 512, 5, 5, 2, 2, 2, 2))
model:add(nn.SpatialBatchNormalization(512))
model:add(nn.ReLU(true))
model:add(nn.SpatialConvolution(512, 512, 5, 5, 2, 2, 2, 2))
model:add(nn.SpatialBatchNormalization(512))
model:add(nn.ReLU(true))
model:add(nn.SpatialFullConvolution(512,512,2,2,2,2))
model:add(nn.SpatialBatchNormalization(512))
model:add(nn.ReLU(true))
model:add(nn.SpatialFullConvolution(512,256,2,2,2,2))
model:add(nn.SpatialBatchNormalization(256))
model:add(nn.ReLU(true))
model:add(nn.SpatialFullConvolution(256,64,2,2,2,2))
model:add(nn.SpatialBatchNormalization(64))
model:add(nn.ReLU(true))
model:add(nn.SpatialFullConvolution(64,16,2,2,2,2))
model:add(nn.SpatialBatchNormalization(16))
model:add(nn.ReLU(true))
model:add(nn.SpatialFullConvolution(16,2,2,2,2,2))
model:add(nn.SpatialBatchNormalization(2))
model:add(nn.ReLU(true))
```

Result:

This MSE for channel a* is 2.113469 and The MSE for channel b* is 5.530940

There is some improvement of in terms of MSE, but I cant tell much difference between the ourput image.

```
luyaocz@luyaocz-B85N-PHOENIX-WIFI: ~/Desktop/Graphic_proj2_lz6nf
epoch = 5; test mse = 4.486746
MSE for a*: 2.559097; MSE for b*: 6.414658
MSE = 4.486778
^C
luyaocz@luyaocz-B85N-PHOENIX-WIFI:~/Desktop/Graphic_proj2_lz6nf$ qlua part2.lua
ok
epoch = 1; train mse = 5.957546
epoch = 1; test mse = 4.963832
epoch = 2; train mse = 3.965080
epoch = 2; test mse = 4.945286
epoch = 3; train mse = 3.435218
epoch = 3; test mse = 4.407832
epoch = 4; train mse = 2.958186
epoch = 4; test mse = 4.070012
epoch = 5; train mse = 2.687208
epoch = 5; test mse = 3.822089
MSE for a*: 2.113469; MSE for b*: 5.530940
MSE = 3.822122
```

Model 3:

Architecture:

This model increase from 10 layer to 16 layers. Instead of greatly increase the feature maps of each convolution layers, I insert more conv/deconv layers after each layers. These added layers will only increase output size and will not change the resolution.

So my resolution flow:

128 – 64 – 32 – 32 – 16 – 16 – 8 – 4 – 8 – 16 – 16 – 32 – 32 – 64 – 64 – 128

feature maps flow:

1 – 2 – 8 – 64 – 64 – 128 – 256 – 512 – 512 – 512 – 256 – 128 – 64 – 32 – 16 – 4 – 2

This model has 5 SaptialConvolution layers and 5 SpatialFullConvolution layers that change both feature maps and resolution, with batchNormalization inserted after each convolution. The maximum of feature maps is still 512.

```
73  -- create neural network
74  model = nn.Sequential()  -- make a multi-layer perceptron
75  model:add(nn.SpatialConvolution(1, 2, 5, 5, 2, 2, 2, 2))
76  model:add(nn.SpatialBatchNormalization(2))
77  model:add(nn.ReLU(true))
78  model:add(nn.SpatialConvolution(2, 8, 5, 5, 2, 2, 2, 2))
79  model:add(nn.SpatialBatchNormalization(8))
80  model:add(nn.ReLU(true))
81  model:add(nn.SpatialConvolution(8, 64, 3, 3, 1, 1, 1, 1))
82  model:add(nn.SpatialBatchNormalization(64))
83  model:add(nn.ReLU(true))
84
85  model:add(nn.SpatialConvolution(64, 64, 5, 5, 2, 2, 2, 2))
86  model:add(nn.SpatialBatchNormalization(64))
87  model:add(nn.ReLU(true))
88  model:add(nn.SpatialConvolution(64, 128, 3, 3, 1, 1, 1, 1))
89  model:add(nn.SpatialBatchNormalization(128))
90  model:add(nn.ReLU(true))
91
92  model:add(nn.SpatialConvolution(128, 256, 5, 5, 2, 2, 2, 2))
93  model:add(nn.SpatialBatchNormalization(256))
94  model:add(nn.ReLU(true))
95  model:add(nn.SpatialConvolution(256, 512, 3, 3, 1, 1, 1, 1))
96  model:add(nn.SpatialBatchNormalization(512))
97  model:add(nn.ReLU(true))
98
99  model:add(nn.SpatialConvolution(512, 512, 5, 5, 2, 2, 2, 2))
100 model:add(nn.SpatialBatchNormalization(512))
101 model:add(nn.ReLU(true))
102 model:add(nn.SpatialFullConvolution(512,512,2,2,2,2))
103 model:add(nn.SpatialBatchNormalization(512))
104 model:add(nn.ReLU(true))
105
106 model:add(nn.SpatialFullConvolution(512,256,2,2,2,2))
107 model:add(nn.SpatialBatchNormalization(256))
108 model:add(nn.ReLU(true))
109 model:add(nn.SpatialFullConvolution(256,128,3,3,1,1,1,1))
110 model:add(nn.SpatialBatchNormalization(128))
111 model:add(nn.ReLU(true))
```

```

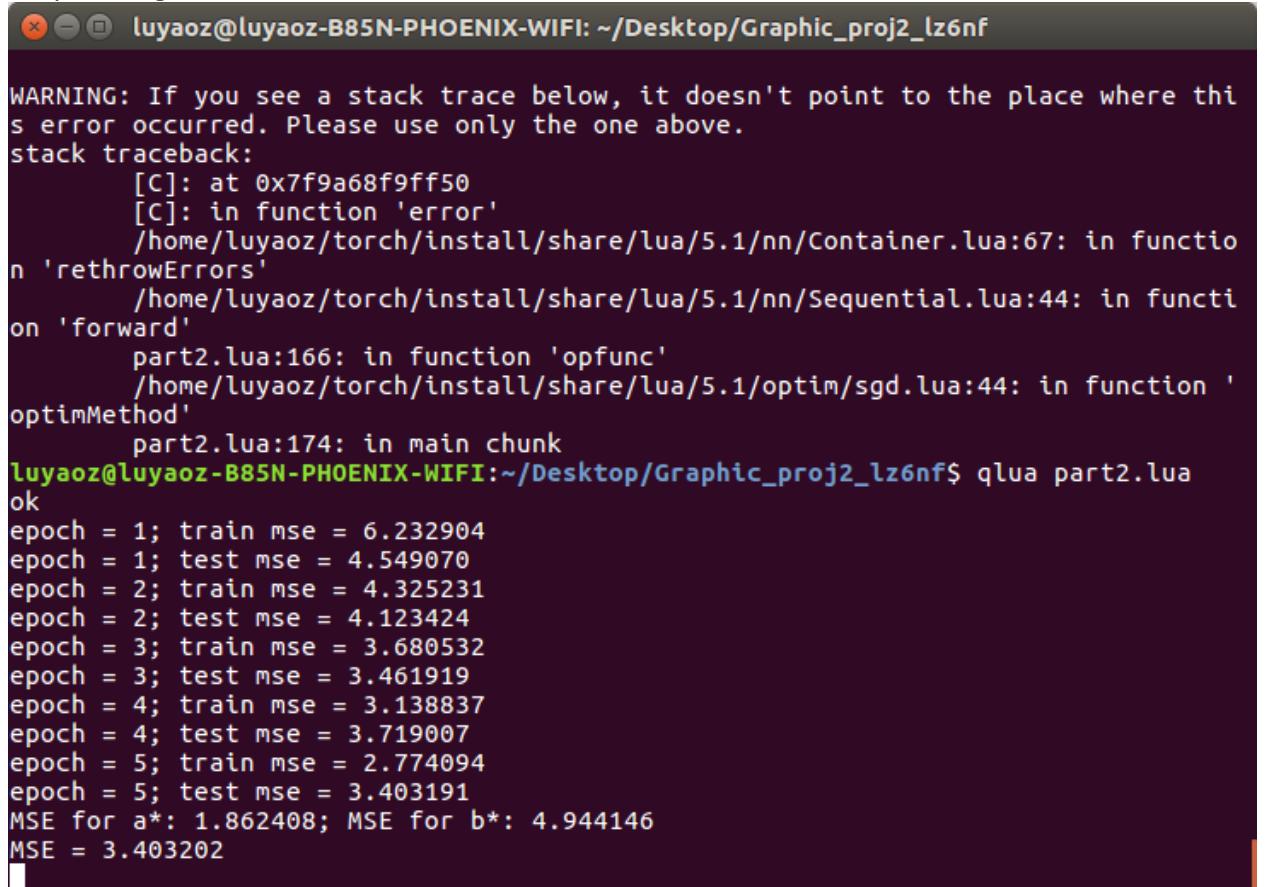
112
113 model:add(nn.SpatialFullConvolution(128,64,2,2,2,2))
114 model:add(nn.SpatialBatchNormalization(64))
115 model:add(nn.ReLU(true))
116 model:add(nn.SpatialFullConvolution(64,32,3,3,1,1,1,1))
117 model:add(nn.SpatialBatchNormalization(32))
118 model:add(nn.ReLU(true))
119
120 model:add(nn.SpatialFullConvolution(32,16,2,2,2,2))
121 model:add(nn.SpatialBatchNormalization(16))
122 model:add(nn.ReLU(true))
123 model:add(nn.SpatialFullConvolution(16,4,3,3,1,1,1,1))
124 model:add(nn.SpatialBatchNormalization(4))
125 model:add(nn.ReLU(true))
126
127 model:add(nn.SpatialFullConvolution(4,2,2,2,2,2))
128 model:add(nn.SpatialBatchNormalization(2))
129 model:add(nn.ReLU(true))

```

Result:

This MSE for channel a* is 1.862408 and The MSE for channel b* is 4.944146

There is some improvement of in terms of MSE, but I still cant tell much difference between the output image.



luyaaoz@luyaaoz-B85N-PHOENIX-WIFI: ~/Desktop/Graphic_proj2_lz6nf

```

WARNING: If you see a stack trace below, it doesn't point to the place where this error occurred. Please use only the one above.
stack traceback:
[C]: at 0x7f9a68f9ff50
[C]: in function 'error'
  /home/luyaaoz/torch/install/share/lua/5.1/nn/Container.lua:67: in function 'rethrowErrors'
  /home/luyaaoz/torch/install/share/lua/5.1/nn/Sequential.lua:44: in function 'forward'
    part2.lua:166: in function 'opfunc'
    /home/luyaaoz/torch/install/share/lua/5.1/optim/sgd.lua:44: in function 'optimMethod'
    part2.lua:174: in main chunk
luyaaoz@luyaaoz-B85N-PHOENIX-WIFI:~/Desktop/Graphic_proj2_lz6nf$ qlua part2.lua
ok
epoch = 1; train mse = 6.232904
epoch = 1; test mse = 4.549070
epoch = 2; train mse = 4.325231
epoch = 2; test mse = 4.123424
epoch = 3; train mse = 3.680532
epoch = 3; test mse = 3.461919
epoch = 4; train mse = 3.138837
epoch = 4; test mse = 3.719007
epoch = 5; train mse = 2.774094
epoch = 5; test mse = 3.403191
MSE for a*: 1.862408; MSE for b*: 4.944146
MSE = 3.403202

```

Model 1 output:



Model 2 output:



Model 3 output:



Model 3, colorize example on train images

