Project 2 - Convolutional Neural Networks

Christoph Metzner (cmetzner), Anna-Maria Nau (anau)
COSC 525 - Deep Learning
University of Tennessee, Knoxville

February 2020

1 Introduction

The aim of this project is to implement a convolutional neural network (CNN) in an object oriented manner. The library is an extension of project 1 with the addition of the following three classes - one to create the convolutional layer objects ("ConvolutionalLayer"), one to create the max pooling layer objects ("MaxPoolingLayer"), and one to create the flatten layer objects ("FlattenLayer"). In addition, for performance evaluation, our results are compared with those of Tensorflow/Keras.

2 Assumptions and Choices

Assumptions made throughout this project were the selections of the input matrices and weights/biases for each example as well as the learning rate of 0.5.

Assumptions for example 1:

```
input_example1 = [[0.02, 0.21, 0.07, 0.17, 0.78],
                  [0.09, 0.25, 0.78, 0.04, 0.24],
                  [0.97, 0.29, 0.37, 0.27, 0.82],
                  [1.00, 0.29, 0.75, 0.62, 0.56],
                  [0.88, 0.65, 0.09, 0.99, 0.87]]
weights_conv2d= [[[0.12, 0.89, 0.21], [0.04, 0.64, 0.13], [0.91, 0.05, 0.64]]]
bias_conv2D = [1]

weights_dense = [[0.99, 0.93, 0.83, 0.49, 0.59, 0.3, 0.96, 0.72, 0.8]]
bias_dense = [1.5]

output_example1 = 0.5
```

Assumptions for example 2:

```
input_example2 = [[0.02, 0.21, 0.07, 0.17, 0.78],
                  [0.09, 0.25, 0.78, 0.04, 0.24],
                  [0.97, 0.29, 0.37, 0.27, 0.82],
                  [1.00, 0.29, 0.75, 0.62, 0.56],
                  [0.88, 0.65, 0.09, 0.99, 0.87]]
weights_example2_conv1 = [[[0.12, 0.89, 0.21], [0.04, 0.64, 0.13], [0.91, 0.05, 0.64]]]
weights_example2_conv2 = [[[0.49, 0.98, 0.89], [0.46, 0.47, 0.44], [0.26, 0.65, 0.87]]]
output_example2 = 0.5

weights_full = [[0.5]]
bias_full = [1.5]
```

Assumptions for example 3:

```
input_example3 = [[0.61, 0.73, 0.42, 0.97, 0.77, 0.68],
                  [0.80, 0.06, 0.39, 0.11, 0.10, 0.65],
                  [0.95, 0.95, 0.70, 0.57, 0.47, 0.98],
                  [0.32, 0.08, 0.69, 0.02, 0.89, 0.07],
                  [0.58, 0.31, 0.21, 0.03, 0.04, 0.04],
                  [0.93, 0.67, 0.84, 0.7, 0.36, 0.08]]

weights_example3 = [[[0.12, 0.89, 0.21], [0.04, 0.64, 0.13], [0.91, 0.64, 0.65]],
                    [[0.49, 0.98, 0.89], [0.46, 0.47, 0.44], [0.05, 0.26, 0.87]]]

bias_example3 = [0.95, 1.0]

weights_full = [[0.99, 0.93, 0.83, 0.49, 0.59, 0.3, 0.96, 0.72]]
bias_full = [1.5]
output_example3 = 0.5
```

3 Problems and Issues

Nothing to add.

4 Running the Code

The `main` method takes the command line variables: `example1`, `example2`, and `example3` to train different networks. Example command to call the script with one argument:

```
python3 dl_525_proj2.py "example1"
```

5 Examples Weight Comparison

```
Fully Connected Layer:
Weights:
 [[0.98977655]
 [0.9297864 ]
 [0.8297821 ]
 [0.48977062]
 [0.5897749 ]
 [0.29977748]
 [0.9597756 ]
 [0.71976924]
 [0.79977596]]
Bias: [1.49976]

Convolutional Layer:
Weights:
 [[0.11996782 0.88997495 0.20996019]
 [0.03994639 0.6399565  0.12995407]
 [0.9099442  0.04995092 0.6399453 ]]
Bias: [0.99989593]
```

```
Current Layer: <__main__.FullyConnectedLayer object at 0x115198cf8>

Current weights: [0.99 0.93 0.83 0.49 0.59 0.3  0.96 0.72 0.8 ]
--> updated weights:
[[0.98977657]
 [0.92978637]
 [0.82978216]
 [0.48977063]
 [0.58977493]
 [0.29977747]
 [0.95977562]
 [0.71976922]
 [0.79977594]]
Current bias: [1.5]
--> updated bias: [1.49976002]

Current Layer: <__main__.FlattenLayer object at 0x114ea2a20>

Current Layer: <__main__.ConvolutionalLayer object at 0x104ebde48>
Current Feature Map:  1
Current Weights:
 [[0.12, 0.89, 0.21], [0.04, 0.64, 0.13], [0.91, 0.05, 0.64]]
Updated Weights:
 [[0.11996782 0.88997494 0.2099602 ]
 [0.0399464  0.6399565  0.12995408]
 [0.90994418 0.04995092 0.63994533]]
Current Bias:  1
Updated Bias:  0.9998959286141215
```

Figure 1: Example 1 final weights/bias of Keras (left) and our CNN (right).

```
Fully Connected Layer:
Weights:
 [[0.46003813]]
Bias: [1.4600126]

2nd Convolutional Layer:
Weights:
 [[0.48998755 0.9799878  0.88998765]
 [0.45998743 0.4699875  0.43998757]
 [0.25998753 0.6499874  0.86998755]]
Bias: [1.9999872]

1st Convolutional Layer:
Weights:
 [[0.11999948 0.88999957 0.20999917]
 [0.03999903 0.6399991  0.12999927]
 [0.90999913 0.04999909 0.63999885]]
Bias: [1.9999981]
```

```
Current Layer: <__main__.FullyConnectedLayer object at 0x11fc72be0>

Current weights: [0.5]
--> updated weights:
[[0.46003813]]
Current bias: [1.5]
--> updated bias: [1.46001253]

Current Layer: <__main__.FlattenLayer object at 0x11fc72ba8>

Current Layer: <__main__.ConvolutionalLayer object at 0x11fc35ba8>
Current Feature Map:  1
Current Weights:
 [[0.49, 0.98, 0.89], [0.46, 0.47, 0.44], [0.26, 0.65, 0.87]]
Updated Weights:
 [[0.48998755 0.97998776 0.88998767]
 [0.45998742 0.46998751 0.43998757]
 [0.25998753 0.64998739 0.86998753]]
Current Bias:  2
Updated Bias:  1.999987209031821

Current Layer: <__main__.ConvolutionalLayer object at 0x10f996e80>
Current Feature Map:  1
Current Weights:
 [[0.12, 0.89, 0.21], [0.04, 0.64, 0.13], [0.91, 0.05, 0.64]]
Updated Weights:
 [[0.1199995  0.88999959 0.20999921]
 [0.03999906 0.63999916 0.12999929]
 [0.90999915 0.04999912 0.6399989 ]]
Current Bias:  2
Updated Bias:  1.9999981182242998
```

Figure 2: Example 2 final weights/bias of Keras (left) and our CNN (right).

```
Fully Connected Layer:
Weights:
 [[0.9896061 ]
 [0.92960566]
 [0.8296091 ]
 [0.4896032 ]
 [0.58961123]
 [0.29960617]
 [0.9596223 ]
 [0.71961445]]
Bias: [1.4995973]

Convolutional Layer:
Weights (feature map 1):
 [[0.11996951 0.8899806  0.20996983]
 [0.03998502 0.6399962  0.1299894 ]
 [0.9099624  0.6399651  0.6499711 ]]
Bias (feature map 1): 0.94995165

Weights (feature map 2):
 [[0.4899835  0.97998434 0.88998157]
 [0.4599926  0.46998858 0.4399926 ]
 [0.04998935 0.25999022 0.86999094]]
Bias (feature map 2): 0.99997526
```

```
Current Layer: <__main__.FullyConnectedLayer object at 0x1198d4ef0>

Current weights: [0.99 0.93 0.83 0.49 0.59 0.3  0.96 0.72]
--> updated weights:
[[0.98960607]
 [0.92960565]
 [0.8296091 ]
 [0.48960316]
 [0.58961124]
 [0.29960615]
 [0.9596223 ]
 [0.71961442]]
Current bias: [1.5]
--> updated bias: [1.49959734]

Current Layer: <__main__.FlattenLayer object at 0x1198d4748>

Current Layer: <__main__.MaxPoolingLayer object at 0x1198d4cc0>

Current Layer: <__main__.ConvolutionalLayer object at 0x10952ae80>
Current Feature Map:  1
Current Weights:
 [[0.12, 0.89, 0.21], [0.04, 0.64, 0.13], [0.91, 0.64, 0.65]]
Updated Weights:
 [[0.11996951 0.88998062 0.20996984]
 [0.03998502 0.63999621 0.12998941]
 [0.90996238 0.63996513 0.64997116]]
Current Bias:  0.95
Updated Bias:  0.9499516746847478

Current Feature Map:  2
Current Weights:
 [[0.49, 0.98, 0.89], [0.46, 0.47, 0.44], [0.05, 0.26, 0.87]]
Updated Weights:
 [[0.48998349 0.97998432 0.88998156]
 [0.45999257 0.46998858 0.43999261]
 [0.04998935 0.25999021 0.86999093]]
Current Bias:  1.0
Updated Bias:  0.9999752591458744
```

Figure 3: Example 3 final weights/bias of Keras (left) and our CNN (left).