filter-viz

February 10, 2019

0.1 Visualizing the trained filters

```python
In [1]: # some startup!
        import numpy as np
        import matplotlib
        # This is needed to save images
        matplotlib.use('Agg')
        import matplotlib.pyplot as plt
        import torch
```

```python
In [2]: # load the model saved by train.py
        # This will be an instance of models.softmax.Softmax.
        # NOTE: You may need to change this file name.
        softmax_model = torch.load('softmax.pt')
```

```python
In [3]: # collect all the weights
        w,b = [param.data for param in softmax_model.parameters()]
        w = w.view(10, 3, 32, 32) #(N, C, H, W)
        w = torch.transpose(w, 1, 3).numpy() #(N, H, W, C)
        ##############################################################################
        # TODO: Extract the weight matrix (without bias) from softmax_model, convert
        # it to a numpy array with shape (10, 32, 32, 3), and assign this array to w.
        # The first dimension should be for channels, then height, width, and color.
        # This step depends on how you implemented models.softmax.Softmax.
        ##############################################################################

        ##############################################################################
        #                           END OF YOUR CODE                                 #
        ##############################################################################
        # obtain min,max to normalize
        w_min, w_max = np.min(w), np.max(w)
        # classes
        classes = ['plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck
        # init figure
        fig = plt.figure(figsize=(6,6))
        for i in range(10):
            wimg = 255.0*(w[i].squeeze() - w_min) / (w_max - w_min)
            # subplot is (2,5) as ten filters are to be visualized
```

```
        fig.add_subplot(2,5,i+1).imshow(wimg.astype('uint8'))
        # save fig!
        fig.show()
        fig.savefig('softmax_filt.png')
        print('figure saved')
```

figure saved

/Users/alexisdurocher/anaconda3/lib/python3.6/site-packages/matplotlib/figure.py:445: UserWarnin
  % get_backend())

```
In [4]: # vis_utils.py has helper code to view multiple filters in single image. Use this to vis
        # neural network adn convnets.
        # import vis_utils
        from vis_utils import visualize_grid
        # saving the weights is now as simple as:
        plt.imsave('softmax_gridfilt.png',visualize_grid(w, padding=3).astype('uint8'))
        # padding is the space between images. Make sure that w is of shape: (N,H,W,C)
        print('figure saved as a grid!')
```

figure saved as a grid!