filter-viz

February 11, 2020

0.1 Visualizing the trained filters

[14]:
```
# some startup!
import numpy as np
import matplotlib
# This is needed to save images
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import torch
```

[32]:
```
# 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('convnet.pt')
```

[38]:
```
# collect all the weights
w = None

w = softmax_model.conv.weight.data.numpy().transpose(0,2,3,1)
print(w.shape)

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)
    fig.add_subplot(9,2,i+1).imshow(wimg.astype('uint8'))
# save fig!
fig.savefig('convnet_filt.png')
print('figure saved')
```

```
(10, 1, 1, 3)
```

```
/Users/ssipani/miniconda3/envs/cs7643/lib/python3.7/site-
packages/ipykernel_launcher.py:11: RuntimeWarning: More than 20 figures have
been opened. Figures created through the pyplot interface
```

```
(`matplotlib.pyplot.figure`) are retained until explicitly closed and may
consume too much memory. (To control this warning, see the rcParam
`figure.max_open_warning`).
  # This is added back by InteractiveShellApp.init_path()


      ␣
↪---------------------------------------------------------------------------

       TypeError                                 Traceback (most recent call␣
↪last)

       <ipython-input-38-0900badacd3d> in <module>
        12 for i in range(10):
        13     wimg = 255.0*(w[i].squeeze() - w_min) / (w_max - w_min)
   ---> 14     fig.add_subplot(9,2,i+1).imshow(wimg.astype('uint8'))
        15 # save fig!
        16 fig.savefig('convnet_filt.png')


       ~/miniconda3/envs/cs7643/lib/python3.7/site-packages/matplotlib/__init__.
↪py in inner(ax, data, *args, **kwargs)
      1597     def inner(ax, *args, data=None, **kwargs):
      1598         if data is None:
   -> 1599             return func(ax, *map(sanitize_sequence, args), **kwargs)
      1600
      1601         bound = new_sig.bind(ax, *args, **kwargs)


       ~/miniconda3/envs/cs7643/lib/python3.7/site-packages/matplotlib/cbook/
↪deprecation.py in wrapper(*args, **kwargs)
       367                 f"%(removal)s.  If any parameter follows {name!r},␣
↪they "
       368                 f"should be pass as keyword, not positionally.")
   --> 369         return func(*args, **kwargs)
       370
       371     return wrapper


       ~/miniconda3/envs/cs7643/lib/python3.7/site-packages/matplotlib/cbook/
↪deprecation.py in wrapper(*args, **kwargs)
       367                 f"%(removal)s.  If any parameter follows {name!r},␣
↪they "
       368                 f"should be pass as keyword, not positionally.")
   --> 369         return func(*args, **kwargs)
       370
       371     return wrapper
```

```
        ~/miniconda3/envs/cs7643/lib/python3.7/site-packages/matplotlib/axes/
↪_axes.py in imshow(self, X, cmap, norm, aspect, interpolation, alpha, vmin,␣
↪vmax, origin, extent, shape, filternorm, filterrad, imlim, resample, url,␣
↪**kwargs)
      5677                                resample=resample, **kwargs)
      5678
   -> 5679         im.set_data(X)
      5680         im.set_alpha(alpha)
      5681         if im.get_clip_path() is None:


        ~/miniconda3/envs/cs7643/lib/python3.7/site-packages/matplotlib/image.py␣
↪in set_data(self, A)
       688                 or self._A.ndim == 3 and self._A.shape[-1] in [3,␣
↪4]):
       689             raise TypeError("Invalid shape {} for image data"
   --> 690                             .format(self._A.shape))
       691
       692         if self._A.ndim == 3:


        TypeError: Invalid shape (3,) for image data
```

[39]:
```
# vis_utils.py has helper code to view multiple filters in single image. Use␣
 ↪this to visuzlize
# neural network adn convnets.
# import vis_utils
from vis_utils import visualize_grid
# saving the weights is now as simple as:
plt.imsave('convnet_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!
```

[ ]:
```

```