Assignment 2

CS 375/Psych 249 (Stanford University, Fall 2017)

1 Background

State-of-the-art neural networks learn parameters via heavily supervised methods involving huge numbers of high-level semantic labels, e.g. category labels for thousands examples in each of thousands of categories [1]). Viewed as technical tools for tuning algorithm parameters, such procedures are perfectly acceptable. As real models of biological learning, they are highly unrealistic, because, among other reasons, infants simply do not receive millions of category labels during development. It is clear on basic ethological grounds that a more correct self-, semi- or un-supervised learning target must be found before realistic picture of visual learning in the brain is possible.

Early unsupervised approaches had some significant successes creating rough models of early visual cortex. Most salient was the discovery that a sparse convolutional autoencoder trained on natural images naturally leads to the emergence of Gabor-wavelet like turning curves [2], suggesting that key features of visual processing are the result of properly chosen unsupervised loss functions. However, such methods have not generalized to significantly more powerful results for deeper models. In particular, multi-layer convolutional sparse autoencoders do not produce very powerful representations capable of solving challenging visual tasks (e.g. ImageNet categorization), an have not been shown to predict response patterns of neurons in higher visual cortical areas. And despite recent enthusiasm for more sophisticated kinds of complexity penalties, such as the Kullback-Leibler (KL) divergence-based distribution approach of variational autoencoders (VAEs) [3], existing domain-general self-, semi- and unsupervised approaches do not appear to be sufficiently powerful to learn representations that match those observed in the real visual system.

A more vision-specific line of work exemplified by the work of Alexei Efros and others shows that some very simple image-process-based objective functionss (e.g. grayscale $\rightarrow$ RGB colorization, inpainting, etc) have some power to induce nontrivial representations[4, 5, 6, 7]. However, these methods are still far from producing representations with observed macque or human levels of behavioral performance on categorization tasks. The discovery of deep neural network learning rules that are computationally powerful but psychologically and neurally accurate is a key challenge for computational visual neuroscience.

2 This Assignment

The purpose of this assignment is to become familiar with coding up unsupervised loss functions and training networks with them. As in Assignment 1, there are two basic components to the assignment: network training and network evaluation.

2.1 Network Training

Code up and train the models with the following loss functions and architectures:

- A shallow bottleneck convolutional autoencoder model. Specifically, this involves an architecture of the form:

  $$F(x) = \mathbf{DeConv}[\mathbf{ReLu}[\mathbf{Conv}(x)]]$$

  where the convolution has a significant stride (e.g. 16) and a reasonable number of out-channels (e.g. 64). A reasonable choice for kernel size might be 7 in both the convolution and deconvolution. The autoencoder loss is simply the L2 reconstruction distance between the original image and the predicted output, e.g.

  $$L_2(X) = ||F(X) - X||^2.$$

  What if you use a sigmoid instead of a relu nonlinearity?
- A pooled version of the above, e.g.

  $$F(x) = \mathbf{DeConv}[\mathbf{Pool}[\mathbf{ReLu}[\mathbf{Conv}(x)]]]$$

  where together the pool and conv operators have a significant stride (e.g. 4 each in the pool and convolution).

- Sparse versions of 1 and 2 above, e.g. where the loss function optimized by the network includes terms both for the reconstruction loss and sparsity of activations in the hidden layer. You can experiment about where the sparsity should imposed (e.g. after or before the Relu or pooling layers). This is meant to qualitatively reproduce the results of Olshausen and Field [2, 8], but with a fully learned convolutional autoencoder.
- A deep symmetric convolutional autoencoder. This is an "hourglass shape" architecture in which the encoder consists of multiple repeated layers of pooling and convolution, and the decoder is structurally symmetric to the encoder (e.g. layer i of the encoder has the same shape as layer $n-i$ of the decoder, where n is the total number of layers).[1] You can explore whether to impose a sparseness penalty on the activations of the top encoder output (e.g. the waist of the hourglass network.
- A Variational Autoencoder, as discussed in class.[2]
- The Colorful Image Colorization network [3], where the autoencoder-like network is attempting to infer RGB images from grayscaled images.

Training Sets: You will be given with a dataprovider for the CIFAR-10 dataset [4] in addition to the ImageNet dataprovider you used in assignment 1. To start out with, you should train all the above architectures on CIFAR-10. You should pick two of the architectures, one of which must include the Colorful Image Colorization architecture, to train on ImageNet.

2.2 Network Evaluation

The main bulk of the evaluation step is identical to that in Assignment 1: for each of several time points during training, as well as the final trained point, perform the task generalization, RSA, neural fitting and filter visualization evaluations as in Assignment 1.

The only additional step that you should take here that was unnecessary in Assignment 1 is to add an evaluation of the unsupervised models on ImageNet. Specifically, for each encoding layer of the trained autoencoders above, train a linear classifier from the features at that layer to solve Imagenet[5] To compare this result to a strong baseline, compare to validation performances on the AlexNet you trained in Assignment 1.

2.3 Lab Reports

As with Assignment 1, we expect results to be reported in a Lab Report.

References

[1] Deng, J. *et al.* ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE CVPR* (2009).

[2] Olshausen, B. A. *et al.* Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996).

[3] Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[4] Zhang, R., Isola, P. & Efros, A. A. Colorful image colorization. In *European Conference on Computer Vision*, 649–666 (Springer, 2016).

[5] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T. & Efros, A. A. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2536–2544 (2016).

[6] Pathak, D., Girshick, R., Dollár, P., Darrell, T. & Hariharan, B. Learning features by watching objects move. *arXiv preprint arXiv:1612.06370* (2016).

[7] Noroozi, M. & Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, 69–84 (Springer, 2016).

[8] Olshausen, B. A. & Field, D. J. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research* **37**, 3311–3325 (1997).

[1] You might find `https://github.com/pkmital/tensorflow_tutorials/blob/master/python/09_convolutional_autoencoder.py` helpful.

[2] You might find `https://github.com/pkmital/tensorflow_tutorials/blob/master/python/11_variational_autoencoder.py` or `https://github.com/siavashk/imagenet-autoencoder` helpful.

[3] See `https://arxiv.org/pdf/1603.08511.pdf`.

[4] `https://www.cs.toronto.edu/~kriz/cifar.html`

[5] As described at the top of page 13 in `https://arxiv.org/pdf/1603.08511.pdf`.