(Article Review)

Deconvolutional Networks*

Assignment 4

ENGG 6500 | Intro to Machine Learning

Dr. Graham Taylor

Jesse Knight

2015-12-03

*Article:

Adaptive Deconvolutional Networks for Mid and High Level Feature Learning

Matthew D. Zeiler, Graham W. Taylor and Rob Fergus, ICCV, 2011.

1 Overview & Motivation

1.1 Deconvolutional Networks

Adaptive Deconvolutional Networks for Mid and High Level Feature Learning, 2011 ('the paper') offers an improved training method for *Deconvolutional Networks* (DN). DNs were first proposed in 2010 by the authors of the 2011 paper [with Dilip Krishnan] as a framework under which to learn good hierarchical image features for an arbitrary task – e.g. denoising, object recognition. DNs differ from convolutional neural networks (CNNs) in that the training objective is an unsupervised one: to identify an optimal set of layered filters and feature maps which reconstruct the input image set using alternating layers of pooling and deconvolution. They are also distinct from convolutional RBMs and other image decomposition models in their unidirectional design (top-down) and sparse representation. Sparsity is encouraged in the DN models discussed here using max pooling and the $|\cdot|_1$ norm of latent feature maps (with ISTA).

1.2 Original Training Method

The original (2010) implementation of the DN was trained using a convolutional variant on the greedy layer-wise pre-training method suggested by Hinton *et al.* (2006) for deep neural nets (DNN). Specifically, the objective function for each representation layer is optimized with respect to the previous layer reconstruction. While this method is reasonable for many DNN architectures, sparse encoding for images is arguably more sensitive to errors moving up the net if the only training target is the layer directly below.

2 Model Development

2.1 Layer-Wise Input Target

The 2011 paper addresses this deficiency of the DN layer-layer training using a novel *input image* target for training each layer. This is made possible by a procedure for unpacking the current layer's filters and feature maps (to be trained) down through alternating unpool and convolution operations to give an estimation of the input image (reconstruction process R). The image-specific set of propagated filter activations is represented using a latent set of *switch* variables. While the model is overcomplete in design, this set of switches represents the max-pooling used to encode each image uniquely and nontrivially.

The authors note that the reconstruction pass through the model is simply a linear series of operations, implying the inverse operation R^T can be easily defined. The benefits of this are twofold. First, the 2010 paper described no method for computing the feature maps from the input (i.e. R^T). Second, these linear propagations are computationally efficient and parallelizable, meaning this model can be trained more quickly, or on larger datasets, than many other image decomposition models. Moreover, rapid computation of the feature representation is relevant to potential end goals of robotics-related computer vision tasks.

2.2 Algorithm

Each layer is trained in series, bottom-up, while holding the kernels and switch settings of lower layers fixed. Layers are randomly initialized and trained over the entire dataset for E epochs. Each epoch, the switch variable representation for each image is optimized based on the existing filter bank using T ISTA iterations. At the end of the epoch, the filter bank is updated to optimize all reconstructions simultaneously, holding the switches constant, using conjugate gradients (i.e. the problem is convex).

3 DN for Object Recognition

The authors learned 4 DN layers of 7 x 7 filters for use in an object identification task (an omission of the introductory 2010 paper) to demonstrate competitive utility. The classification network employing the features was a spatial pyramid matching (SPM) model presented by Lazebnik *et al.* (2006), a robust hierarchical improvement on the bag of features framework.

3.1 Dataset

The model was trained on both Caltech datasets (101 and 256). Images were normalized using colour channel averaging, size-matching padding and local contrast normalization (i.e. equal size grayscale images with zero mean, local unit norm). Caltech datasets are popular in computer vision, facilitating algorithmic comparisons; they contain 30 – 800 images each of 101 / 256 object classes, with objects generally centred, in typical orientations. The larger set is more challenging as there are more potential labels per input. In the paper, the DN model is trained using 30 images per class, and tested on the standard test data per set.

3.2 Results

The authors used three DN feature sets for the SPM classifier – layers 1, 4 and a combination of 1+4; all three performed better than the majority of similar SPM-based models on Caltech 101, with accuracy rates between 67.8 ± 1.2 % and 71.0 ± 1.0 %. The paper also presents impressive results for transfer learning between the Caltech 101 and 256 datasets: almost equal errors for test data, irrespective of the training set, including better results over other architectures trained on the native set! This shows that the features learned using this DN model generalize very well to similar images.

Lastly, as with many other convolution-based models, the authors note that the max-pooling operation outperforms average pooling; they also show that activation sparsity in the representation (i.e. one activation per filter in the highest layer) is crucial to good reconstructions and recognition.

4 Limitations & Future Work

4.1 Layer Training Target Comparison

An interesting comparison might have been made between image reconstructions using the 2010 (layer-layer training) versus 2011 (layer-input) training methods. This would provide an intuitive demonstration of any fidelity benefits associated with the new method. However, a good unbiased comparison would be hard to achieve given obligate randomness in learned features.

4.2 Future Work

The authors note that their performance on Caltech-101 is only bested by networks with alternative classifier architectures. This implies that the representation learned by the DN is in fact state-of-the art, and that performance could be maximized on the Caltech-101 data using a combination of DN features and an improved classifying network.

Additionally, while popular, the Caltech databases are not as challenging as other image sets like Image-Net, or tasks like facial recognition. It would be interesting to observe the performance of these deconvolutional network-learned features for these tasks.