Building High-level Features Using Large Scale Unsupervised Learning

1 Citation

Le, Quoc V. "Building high-level features using large scale unsupervised learning." Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

`https://arxiv.org/pdf/1112.6209.pdf`

2 Abstract

We train an 9-layer autoencoder on 10M unlabeled images and find it learns to recognize human faces, body parts, and cat faces. We then take this pretrained model and train on ImageNet to get state of the art performance.

3 Introduction

Can unsupervised learning yield neurons that are tuned to specific classes (e.g. human faces)? Restricted Boltzmann Machines (RBM), Autoencoders, and sparse coding are popular unsupervised learning approaches, and people don't really train deep nets on large unlabeled datasets because of the compute cost (our company has a lot of compute resources though).

4 Training Set Construction

We sample 10M random frames from YouTube videos (each video contributes at most one frame). There are about 100k faces (according to an OpenCV face detector).

5 Algorithm

Sparse coding can learn low level features like Gabor filters. RBMs learn higher level features, but require some labeled data for fine-tuning.

Our model is a deep autoencoder. We use local receptive fields, which means each layer connects to a small part of the layer before (this prevents the system from having too many parameters). We use L2 pooling (this outputs square root of sum of squares of input, and thus enables learning invariances), and local contrast normalization. We repeat local filtering, local pooling, and local contrast normalization three times to get a 9 layer model. We have 1B weights. We fix the parameters of the second sublayer and train the first (encoding) and third (decoding) sublayers with the following (where we pick λ to trade reconstruction for sparsity):

$$\text{minimize}_{W_1,W_2} \sum_{i=1}^{m} ||W_2 W_1^T x^{(i)}||_2^2 + \lambda \sum_{j=1}^{k} \sqrt{\epsilon + H_j (W_1^T x^{(i)})^2} \tag{1}$$

We give each machine a piece of the local weights and train in parallel with DistBelief using asynchronous gradient descent.

6 Experiments on Faces

We use the Labeled Faces in the Wild dataset and ImageNet (for distractor images). We then treat each neuron as a decision stump and pick a threshold of its activation for classifying faces. The best neuron recognizes faces with accuracy of 81.7% (guessing all negative gives 64.8% and a random filter and single-layer network gives at most 74%).

Once we have some candidate neurons, we test if they actually detect faces by optimizing (with gradient descent with line search):

$$x^* = \text{argmin}_x f(x; W, H) \text{ subject to } ||x||_2 = 1 \tag{2}$$

The neuron does indeed learn faces. We took sample images and did scaling, translation, and out-of-plane rotation (this requires a 3D model of a face) and found the neuron is invariant to all of these things.

If we remove all the faces (as detected by OpenCV) from the dataset, the best neuron is much worse (just as good as random filter).

7 Cat and Human Body Detectors

We get a dataset of cats and human bodies (these are common YouTube), and some distractor images from ImageNet. We find there are neurons for both of these things.

8 Object Recognition with ImageNet

We added one vs. all logistic classifiers at the end of our network and fine-tune the entire network. We set state of the art on ImageNet. If you don't do the unsupervised pretraining, our model does much worse.

9 Conclusion

By training a deep autoencoder on a 10M image unlabeled dataset and then fine-tuning on ImageNet, we set state of the art.