Network In Network

1 Citation

Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400 (2013).

https://arxiv.org/pdf/1312.4400.pdf

2 Abstract

Instead of sliding a linear kernel over an image, slide a small feedforward neural network. Now that you have this better local modeling, replace the fully connected layers + softmax with global average pooling.

3 Introduction

Usually, kernels in a CNN are applied as generalized linear model (GLM), but we think using a small feedforward neural network (i.e. a multilayer perceptron) that we call an mlpconv will let us model local behavior more easily. This better local modeling lets us replace the final fully connected layers with global average pooling (requires one channel per class in classification), which is more interpretable and doesn't face overfitting problems like the fully connected layer.

4 Convolutional Neural Networks

Previous work introduced the maxout network, which does max-pooling over affine feature maps (and doesn't use a regular activation function like ReLU). This models a piecewise linear approximator, but assumes that that latent concepts lie in a convex set. We think our Network In Network (NIN) removes this assumption.

5 Network In Network

The mlpconv layer does this:

$$f^1_{i,j,k_1} = \max(w_k^{1T} x_{i,j} + b_{k_1}, 0) \tag{1}$$

$$... \tag{2}$$

$$f^n_{i,j,k_n} = \max(w_k^{nT} f^{n-1}_{i,j} + b_{k_n}, 0) \tag{3}$$

You can view mlpconv as cascaded, cross-channel pooling.

Instead of using fully-connected layers, which easily overfit, we use global average pooling. That is, at the end of the network, we have one channel per class and we average over each channel. Then we feed the averages into a softmax. This acts as a regularizer and also lets us view each channel as a spatial map of category confidence.

6 Experiments

We have three mlpconv layers (each followed by spatial max-poolng which downsamples output by factor of 2) and global average pooling at the end. We regularize with dropout on the first two mlpconv layers and weight decay. We initialize weights as in AlexNet, use 128 batch size, and decay learning rate by factor of 10 each time training accuracy stops improving.

We achieve state of the art on CIFAR-10 (50K/10K train/test images of size 32×32 from 10 classes). Dropout actually gives accuracy boost. We get state of the art on CIFAR-100.

We get state of the art on Street View House Numbers (SVHN) (630K 32×32 images) and MNIST (60K/10K images of size 28×28).

Replacing fully connected layers with global average pooling improves performance (does not overfit). If you use a vanilla CNN with global average pooling, it does worse than if you used fully connected layers with dropout. We think this is because linear Convolution is not as strong as mlpconv.

Near the end of the network, we have one channel per class, and we observe that these can be viewed as confidence maps for the class. There is more activation in the region of the image where the object of that class appears. Maybe we could use this for object detection?

7 Conclusions

The mlpconv layer models local patches well and global average pooling is a good regularizer. Having one channel per class acts as a confidence map.