Network in Network

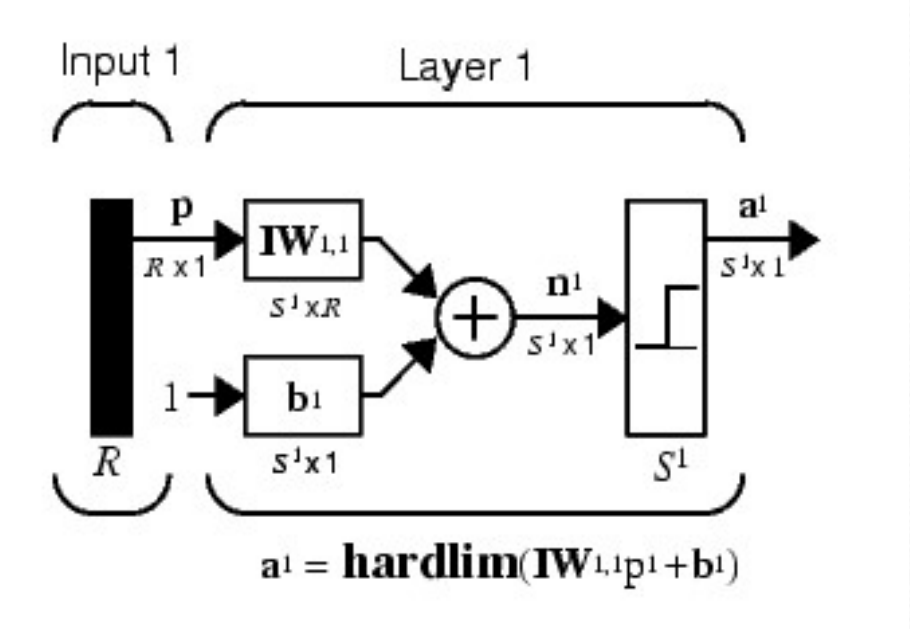
Zachary Jen

Abstract

- Introduction
  - MLP
  - CNN
- Network in Network
  - 1 x 1 convolution kernel
  - Global Average Pooling
- Experiments
- Conclusion

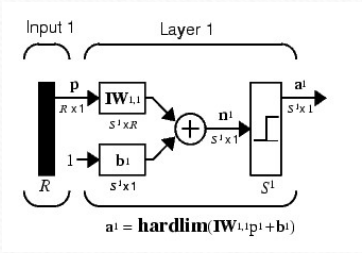
Introduction

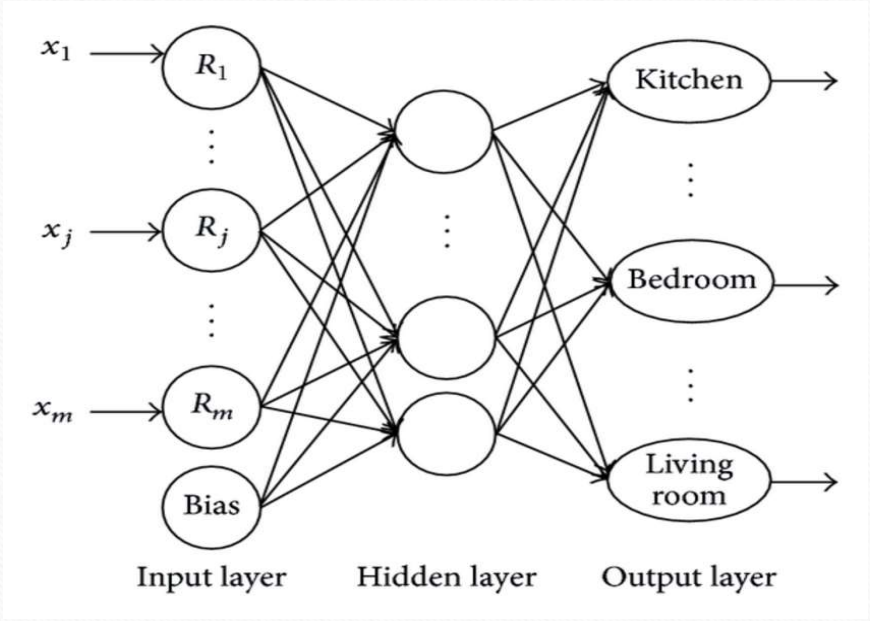
- Perceptron



Introduction (cont.)

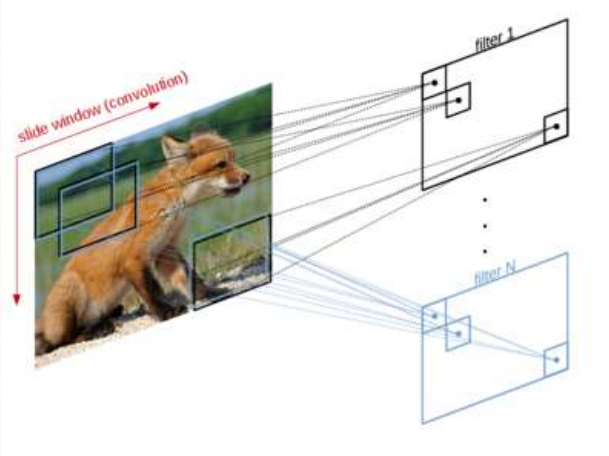
- From Perceptron to MLP

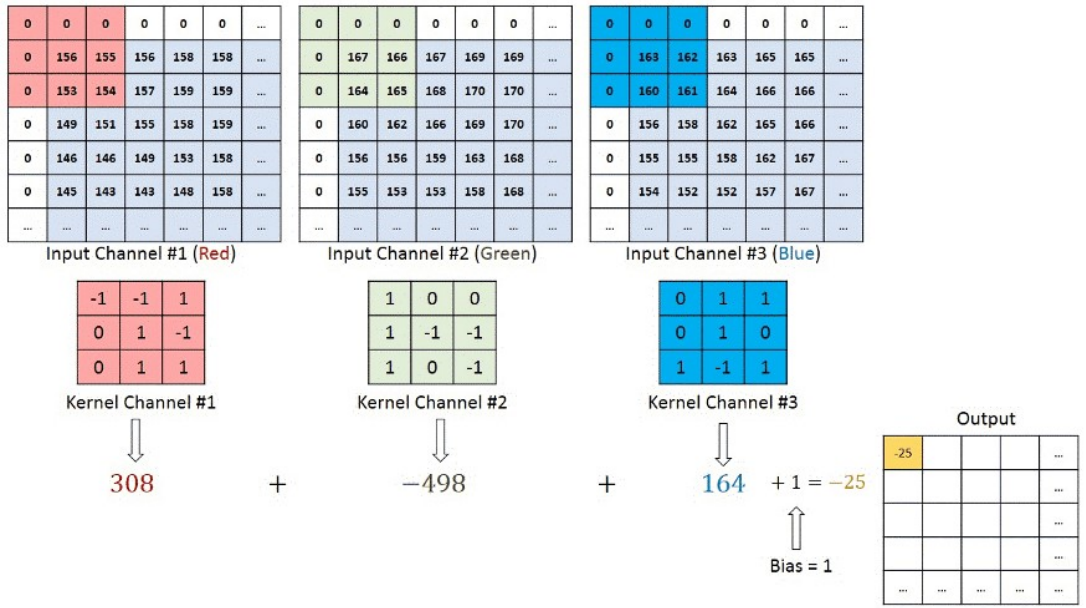




Introduction (cont.)

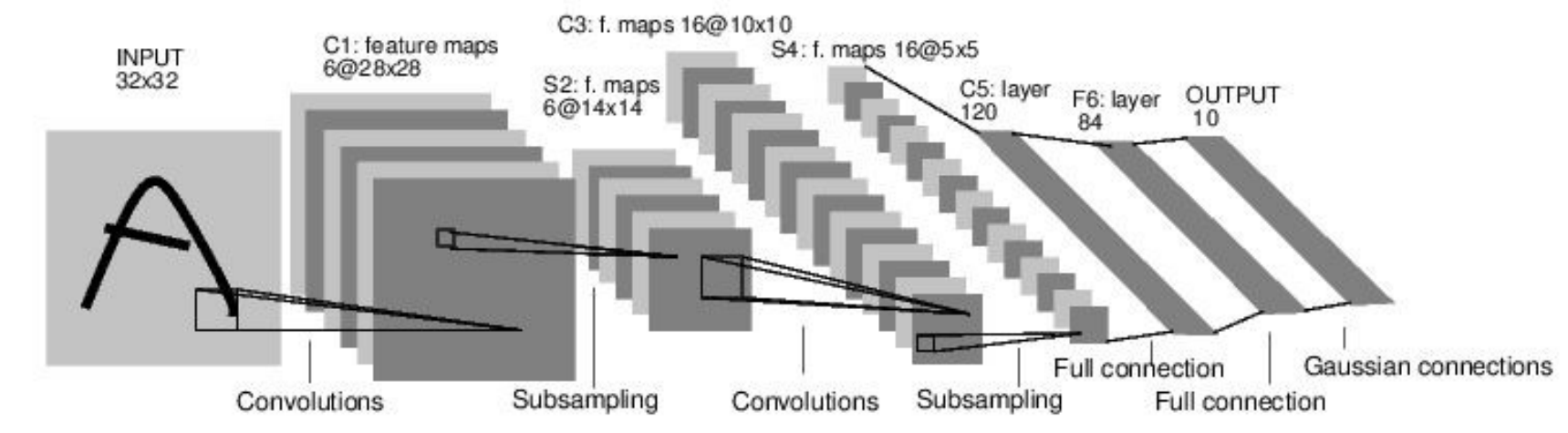
- Convolution





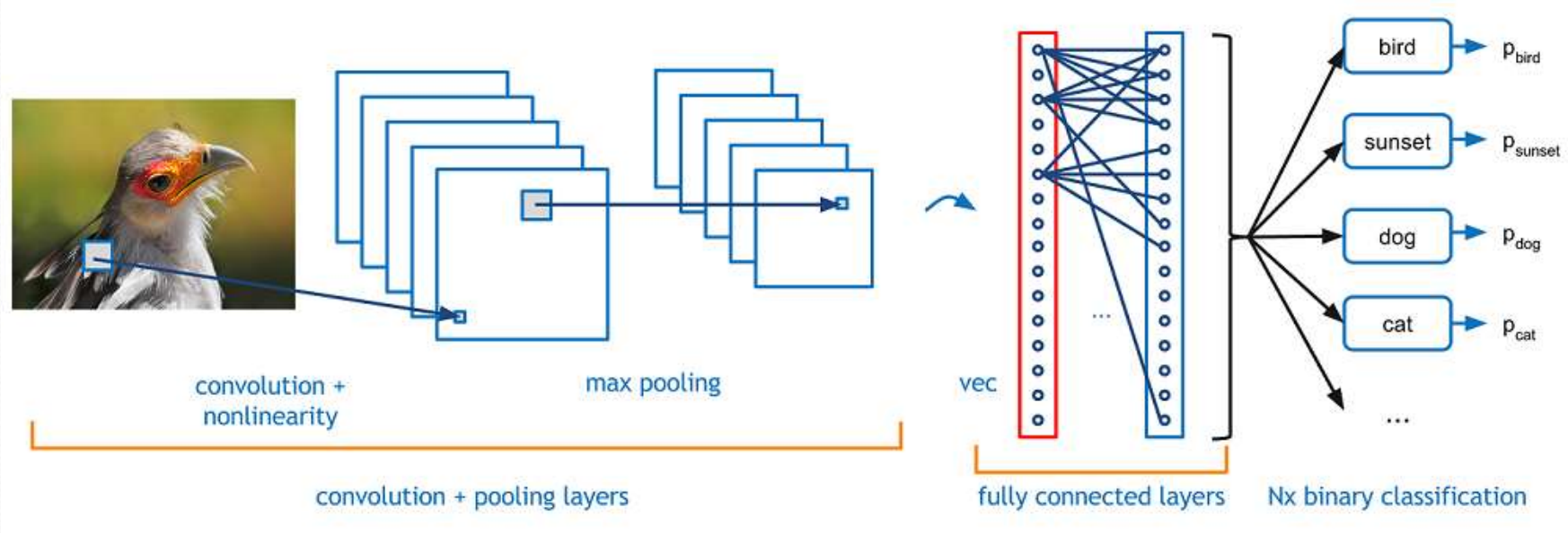
Introduction (cont.)

- CNN

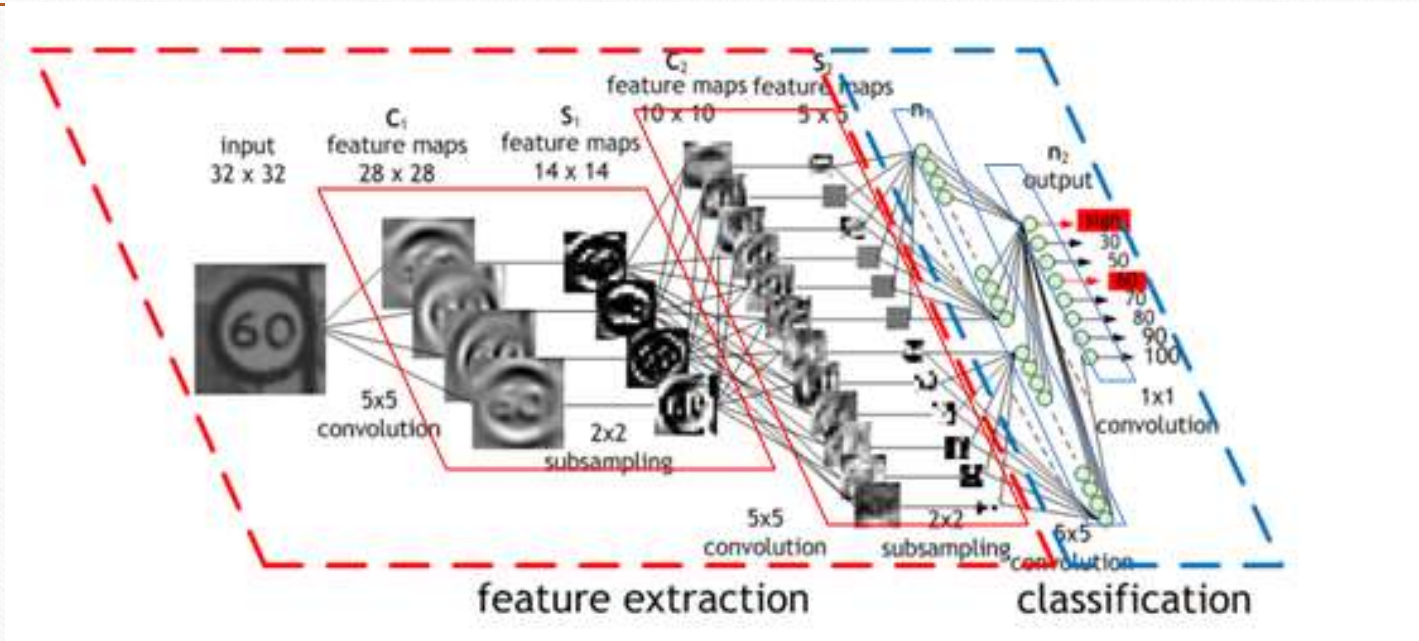


Introduction (cont.)

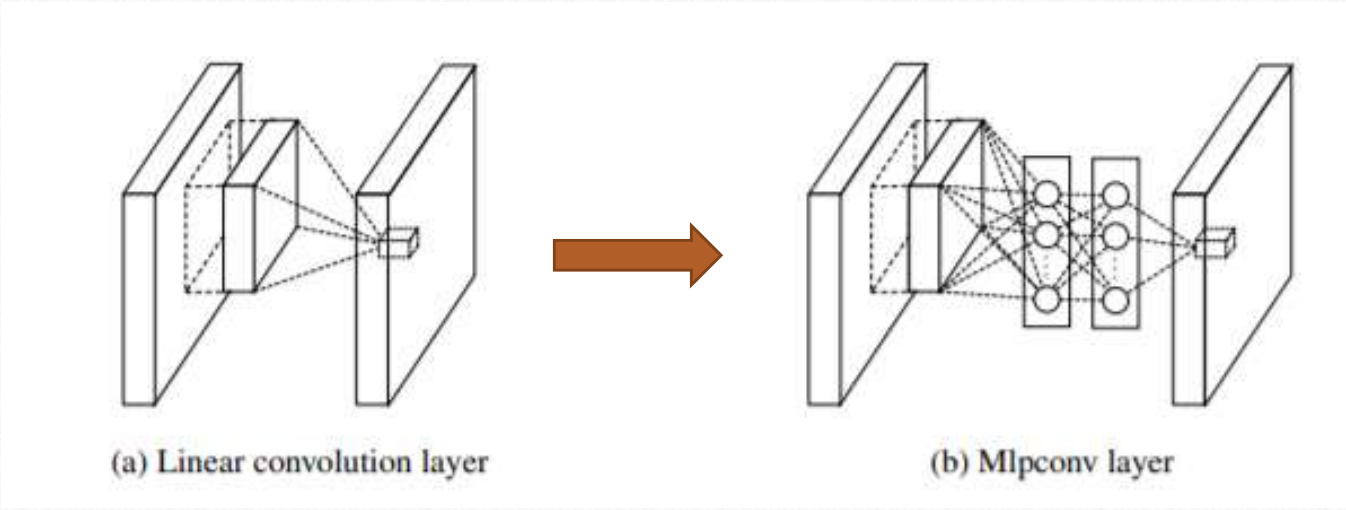
- CNN Model



Introduction (cont.)



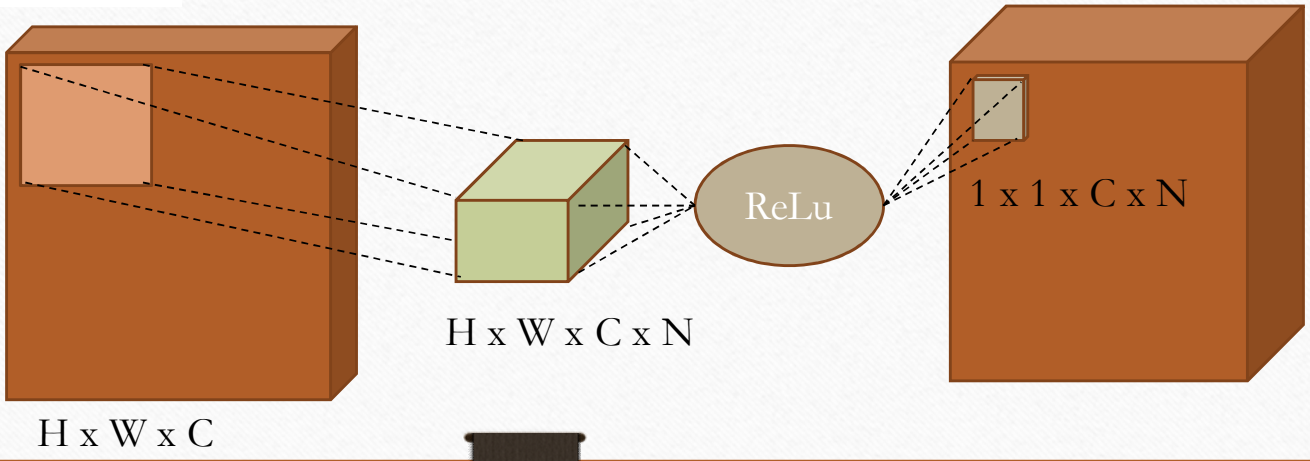
Network in Network



Network in Network (cont.)

- Mathematical function in Convolution Layer

$$f_{i,j,k} = \max(w_k^T x_{i,j}, 0).$$



Network in Network (cont.)

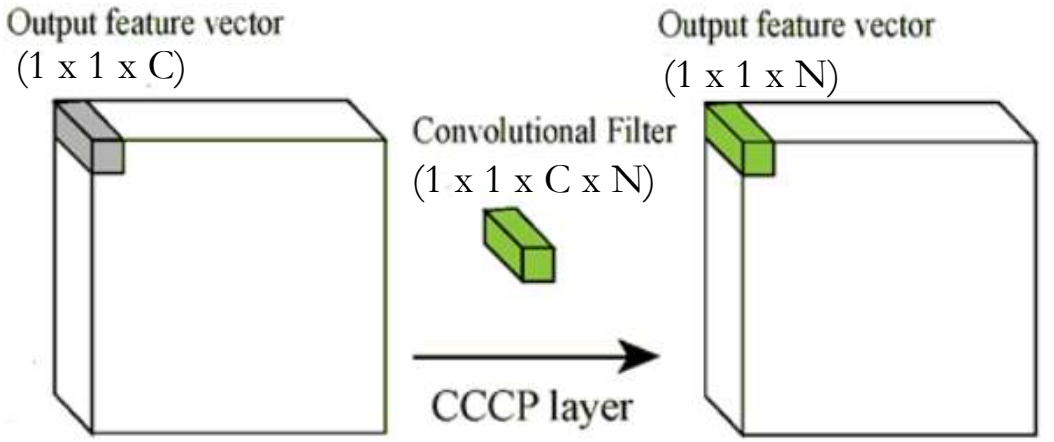
- Motivation for MLP Convolution Layer
  - Individual linear filters can be learned to detect different variations of a same concept
  - Too many filters for a single concept imposes extra burden on the next layer

    -> Needs to consider all combinations of variations from the previous layer
- Expectation
  - It would be beneficial to do a better abstraction on each local patch, before combining them into higher level concepts

Network in Network (cont.)

- MLP Convolution Layer

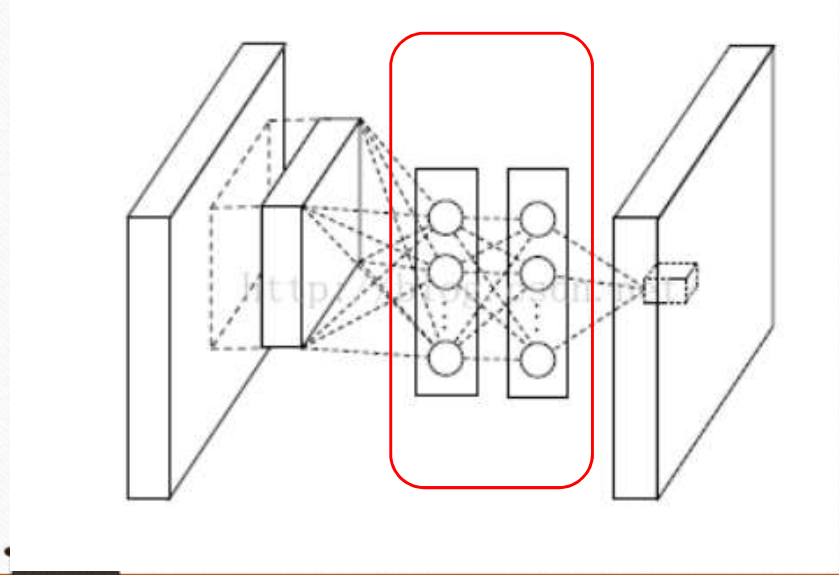
$$
\begin{aligned}
f^1_{i,j,k_1} &= \max(w^{1\,T}_{k_1} x_{i,j} + b_{k_1}, 0). \\
&\vdots \\
f^n_{i,j,k_n} &= \max(w^{n\,T}_{k_n} f^{n-1}_{i,j} + b_{k_n}, 0).
\end{aligned}
$$



Network in Network (cont.)

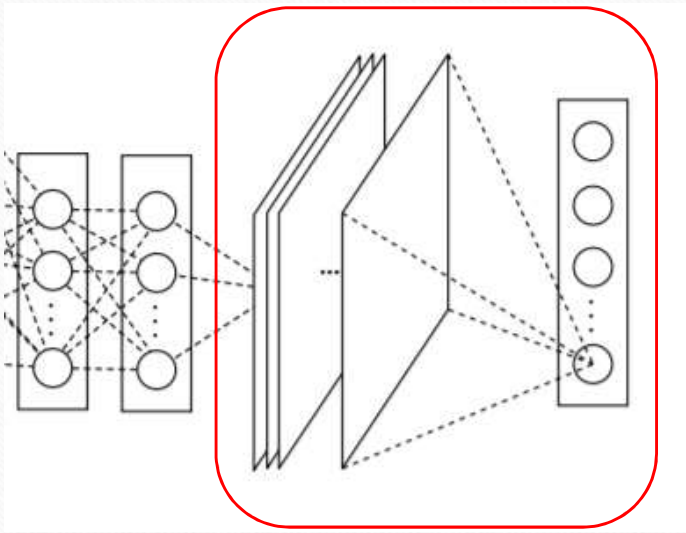
- Cascaded Cross Channel parametric Pooling layer

The cross channel parametric pooling layer is also equivalent to a convolution layer with 1x1 convolution kernel



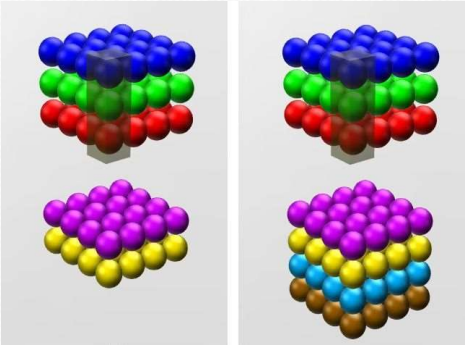
Network in Network (cont.)

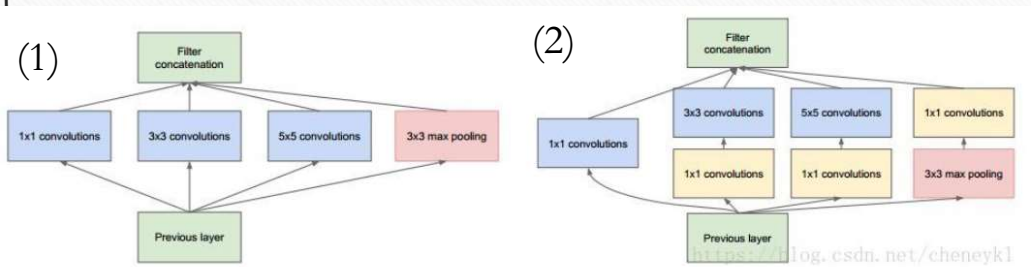
- Global Average Poolling



Network in Network (cont.)

- Benefits on 1 x 1 kernel
  - Easy to change the dimension of output
  - Low down the counting cost





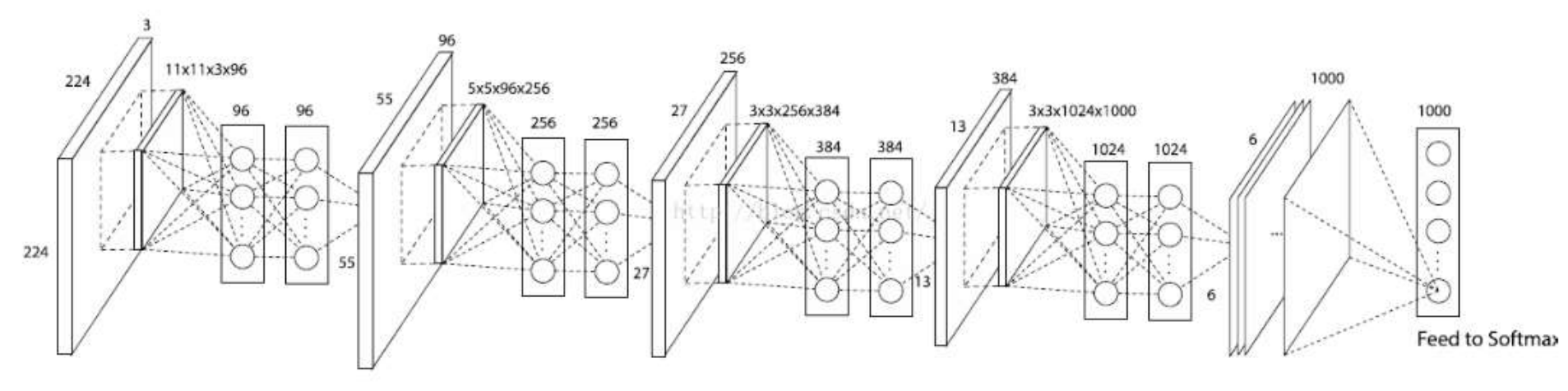
Cost:

(1) 192 x (1x1x64) +192 x (3x3x128) + 192 x (5x5x32) = 387072

(2) 192 x (1x1x64) + (192x1x1x96+ 96 x 3x3x128) + (192x1x1x16+16x5x5x32) = 157184

Network in Network (cont.)

- NIN example

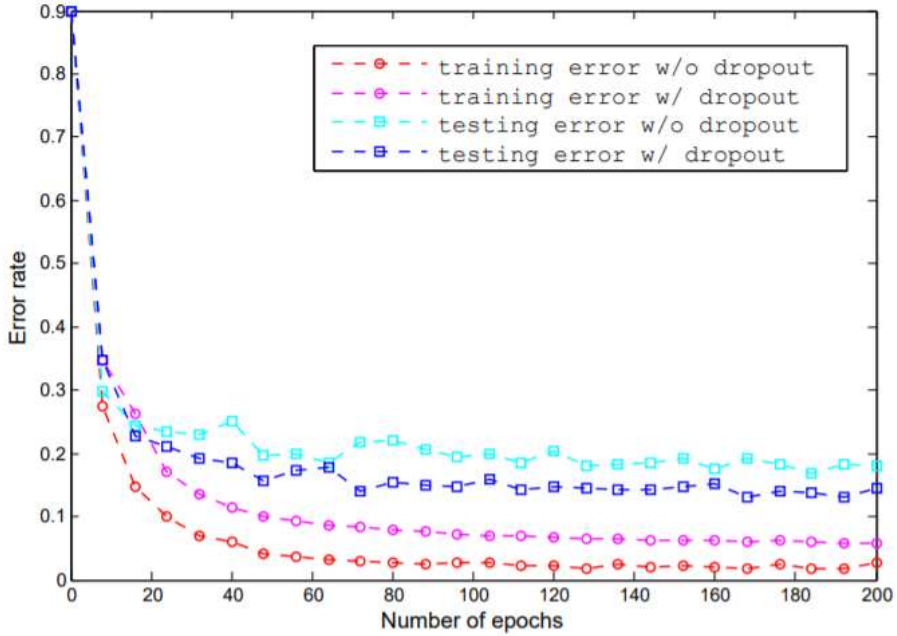


Experiments

- CIFAR-10

Table 1: Test set error rates for CIFAR-10 of various methods.

Method	Test Error
Stochastic Pooling [11]	15.13%
CNN + Spearmint [14]	14.98%
Conv. maxout + Dropout [8]	11.68%
NIN + Dropout	**10.41%**
CNN + Spearmint + Data Augmentation [14]	9.50%
Conv. maxout + Dropout + Data Augmentation [8]	9.38%
DropConnect + 12 networks + Data Augmentation [15]	9.32%
NIN + Dropout + Data Augmentation	**8.81%**



Experiments (cont.)

- CIFAR-100

Table 2: Test set error rates for CIFAR-100 of various methods.

Method	Test Error
Learned Pooling [16]	43.71%
Stochastic Pooling [11]	42.51%
Conv. maxout + Dropout [8]	38.57%
Tree based priors [17]	36.85%
NIN + Dropout	**35.68%**

Experiments (cont.)

- SVHN (Street View Hourse Numbers)

Table 3: Test set error rates for SVHN of various methods.

Method	Test Error
Stochastic Pooling [11]	2.80%
Rectifier + Dropout [18]	2.78%
Rectifier + Dropout + Synthetic Translation [18]	2.68%
Conv. maxout + Dropout [8]	2.47%
NIN + Dropout	2.35%
Multi-digit Number Recognition [19]	2.16%
DropConnect [15]	**1.94%**

Experiments (cont.)

- MNIST

Table 4: Test set error rates for MNIST of various methods.

Method	Test Error
2-Layer CNN + 2-Layer NN [11]	0.53%
Stochastic Pooling [11]	0.47%
NIN + Dropout	0.47%
Conv. maxout + Dropout [8]	**0.45%**

Experiments (cont.)

- Global Average Pooling as a Regularizer

Table 5: Global average pooling compared to fully connected layer.

Method	Testing Error
mlpconv + Fully Connected	11.59%
mlpconv + Fully Connected + Dropout	10.88%
mlpconv + Global Average Pooling	10.41%

Experiments (cont.)

- Visualization of NIN

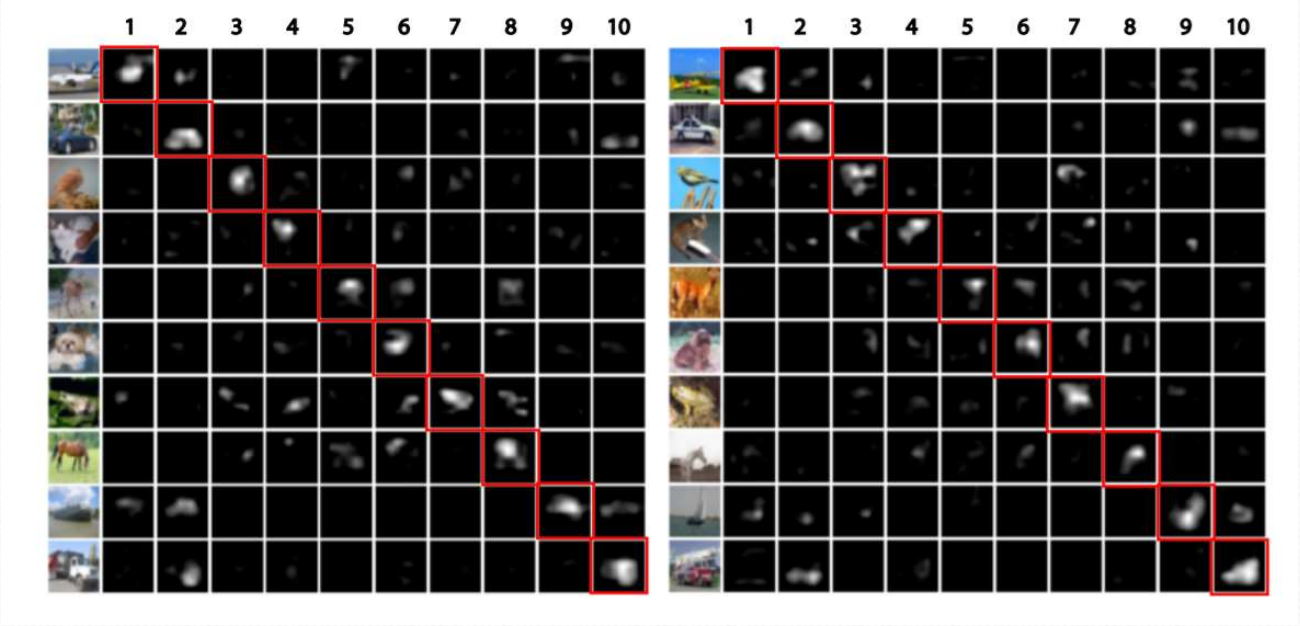


Figure 4: Visualization of the feature maps from the last mlpconv layer. Only top 10% activations in the feature maps are shown. The categories corresponding to the feature maps are: 1. airplane, 2. automobile, 3. bird, 4. cat, 5. deer, 6. dog, 7. frog, 8. horse, 9. ship, 10. truck. Feature maps corresponding to the ground truth of the input images are highlighted. The left panel and right panel are just different examplars.

Conclusions

- New structure : NIN
  - Use multilayer perceptrons to convolve the input and a global average pooling layer as a replacement for the fully connected layers in conventional CNN
- Function contributes
  - MLPconv layers model
    1) make the local patches better
  - Global average pooling
    1) acts as a structural regularizer
    2) prevents overfitting globally

Q & A


Thanks for your
Attention
meme-arsenal.ru