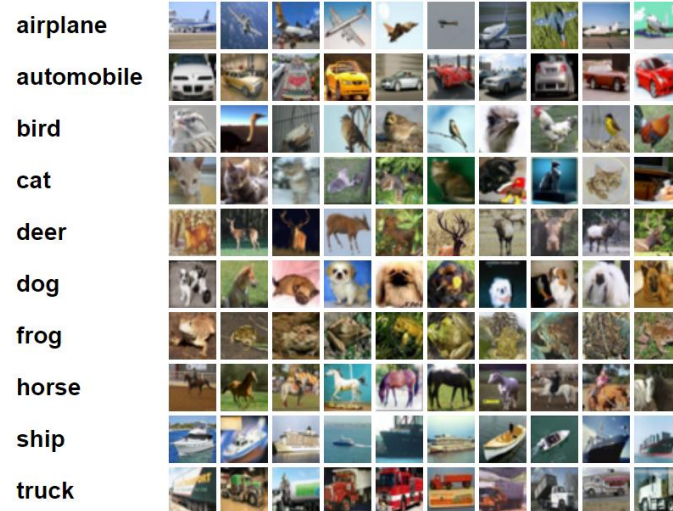# #HW2 CNN on Cifar-10

<u>Requirements</u>:

- Implement "**Network In Network**" (NIN) [1] convolutional architecture

<u>Environment</u>:

- Cifar-10 dataset The CIFAR-10 dataset consists of 60000 32 × 32 color images (RGB) in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

- Download: https://www.cs.toronto.edu/~kriz/cifar.html
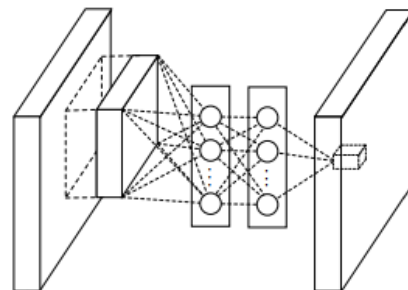

<u>Lab Description</u>:

- Network In Network (NIN)

    The overall structure of the NIN is the stacking of multiple mlpconv layers. It is called "Network In Network" (NIN) as we have micro networks (MLP), which are composing elements of the overall deep network, within mlpconv layers.
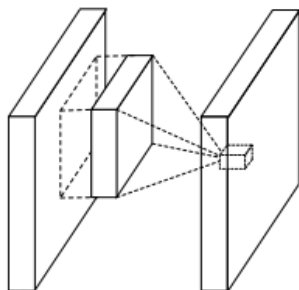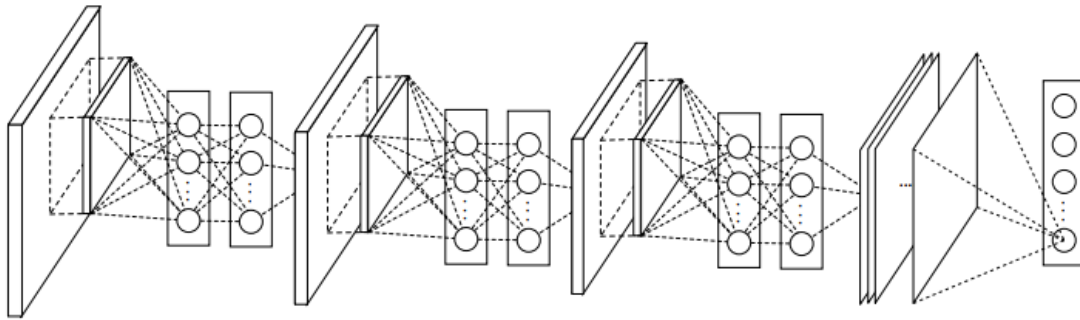

(a) Linear convolution layer   (b) Mlpconv layer

    Figure illustrates the overall structure of Network In Network , you can
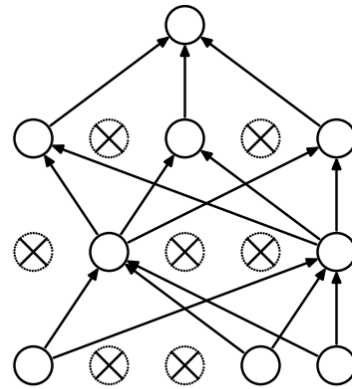
reference our suggestion below.

Layer	Detail	Act fun
Conv 1	filter size = 5x5, # of filter =192, pad = 2, stride = 1	Relu
Mlp 1	filter size = 1x1, # of filter =160, pad = 0, stride = 1	Relu
Mlp 2	filter size = 1x1, # of filter =96, pad = 0, stride = 1	Relu
Pool 1	3x3 max pooling, stride = 2	
	Dropout 0.5	
Conv 2	filter size = 5x5, # of filter =192, pad = 2, stride = 1	Relu
Mlp 2-1	filter size = 1x1, # of filter =192, pad = 0, stride = 1	Relu
Mlp 2-2	filter size = 1x1, # of filter =192, pad = 0, stride = 1	Relu
Pool 2	3x3 max pooling, stride = 2	
	Dropout 0.5	
Conv 3	filter size = 3x3, # of filter =192, pad = 1, stride = 1	Relu
Mlp 3-1	filter size = 1x1, # of filter =192, pad = 0, stride = 1	Relu
Mlp 3-2	filter size = 1x1, # of filter =10, pad = 0, stride = 1	Relu
Pool 3	8x8 average pooling, stride =1	
	softmax	

● Dropout

Dropout prevents overfitting and provides a way of approximately combining exponentially many different neural network architectures efficiently. The term "dropout" refers to dropping out units (hidden and visible) in a neural network. The choice of which units to drop is **random**.
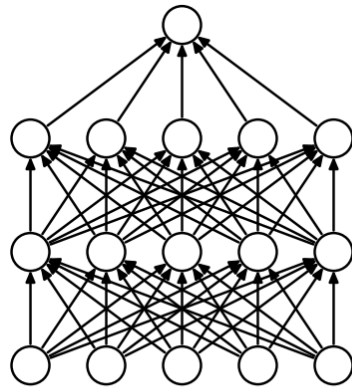
(a) Standard Neural Net   (b) After applying dropout.

- Data augmentation

  Translation, flipping, noise, Brightness…

References:

[1] Lin, M., Chen, Q., & Yan, S. (2013). Network in network. arXiv preprint arXiv:1312.4400.

[2] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.

[3] https://www.tensorflow.org/tutorials/deep_cnn/

[4] Department of Computer Science, National Chiao-Tung University, IOC5184(5259) Deep Learning and Practice Lab3.