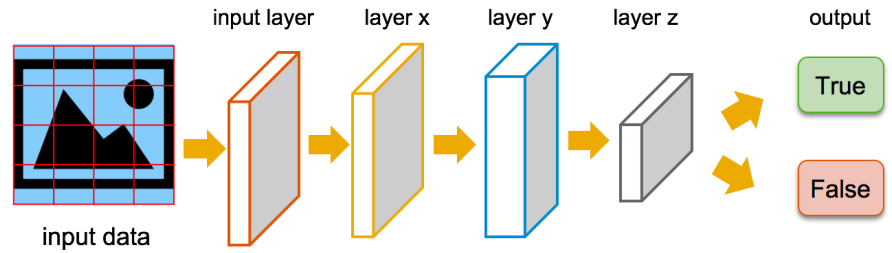
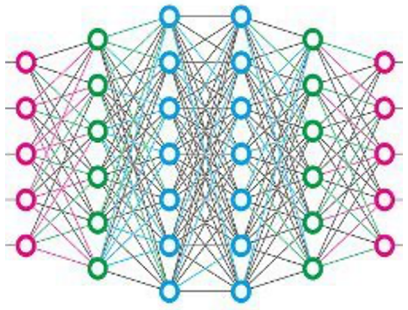
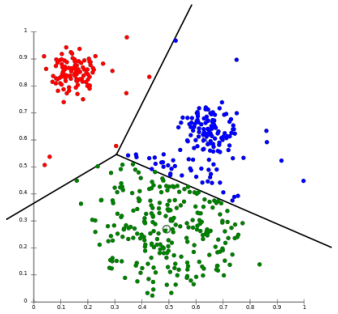
CS 445
Introduction to Machine Learning

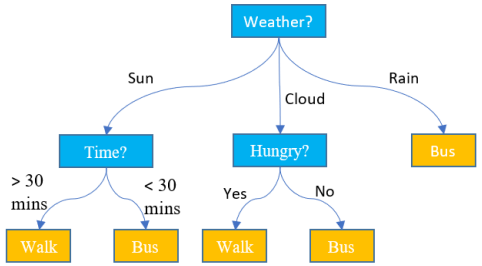
Convolution Neural Networks

Instructor: Dr. Kevin Molloy









Announcements

PA 3

- Due on November 13th at 5:00 pm
- Good goal to complete dense models by the end of the weekend

Review of Last Class

- Softmax and One-Hot Encoding (multiclass classification with ANN)
- Using Convolutions on Images
    - Filters/kernels enable the NN to learn the features that are important in the image for classifying
    - These kernels provide "position invariance" to the features (the pattern can be detected anywhere in the picture)

Learning Objectives

- Reducing the Number of Parameters
- Preventing Overfitting in NNs (and CNNs)
- Data Normalization
- Experimental Design for NNs

Strides and Padding

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

	10	10	10	0	0	0	
	10	10	10	0	0	0	
	10	10	10	0	0	0	
	10	10	10	0	0	0	
	10	10	10	0	0	0	
	10	10	10	0	0	0	

- Corners only appear once. Solution is to pad the image (see left)

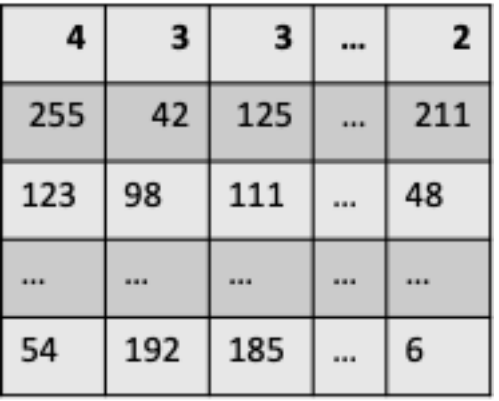
Example of stride = 2

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

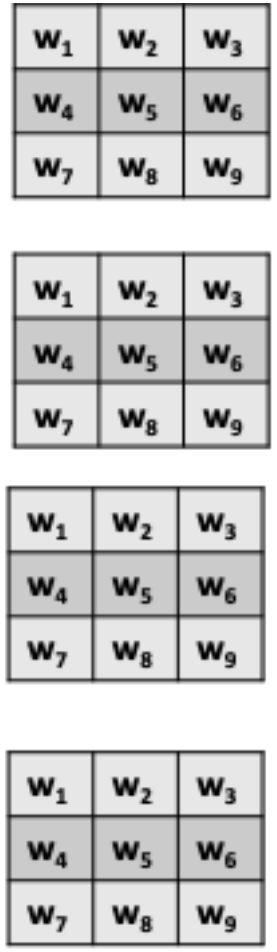
Recall (W – F + 2P/S) + 1

Basic CNN



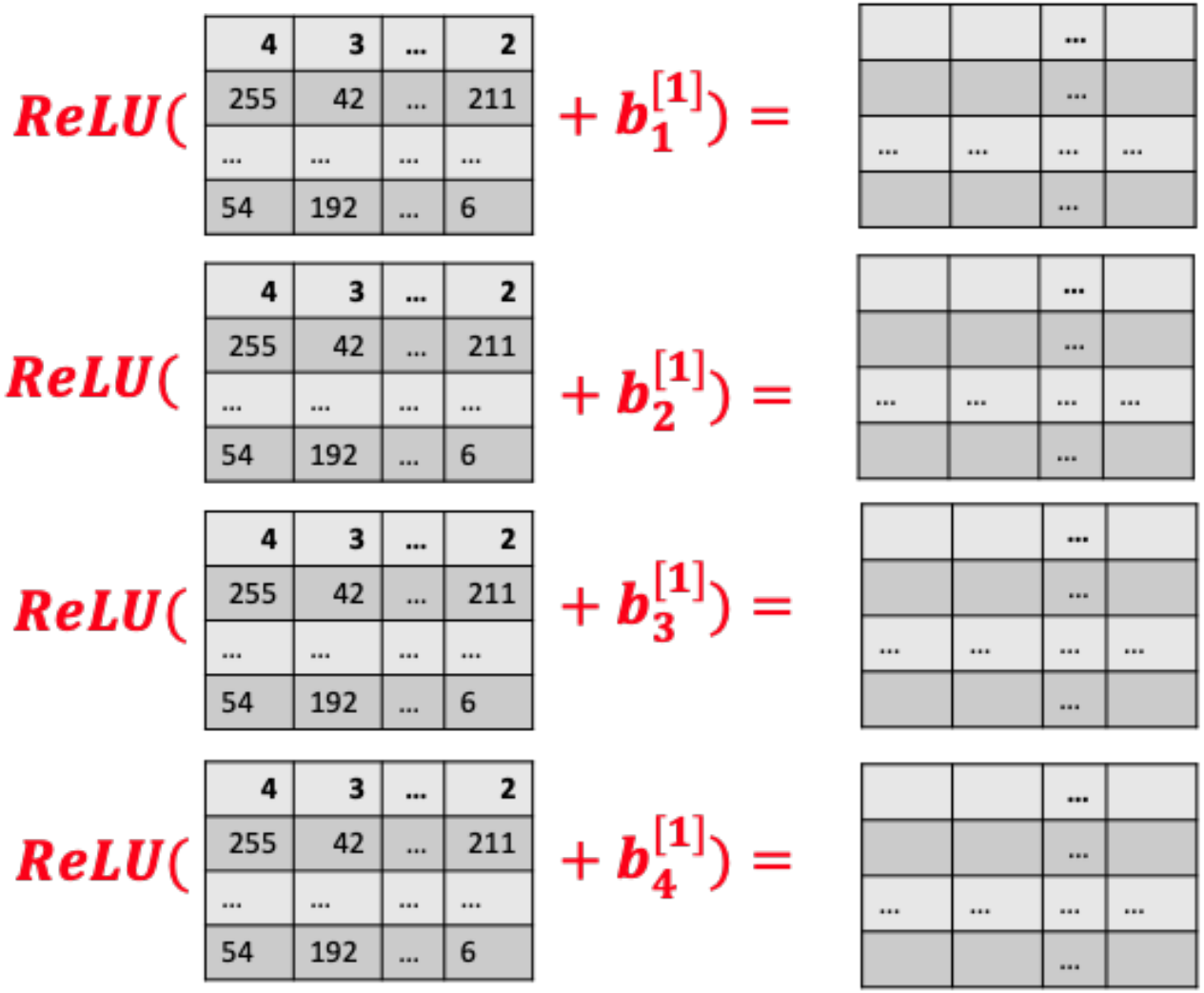
$a^{[0]}$

100 x 100 x 1



4 filters

Shape of W (3,3,4)



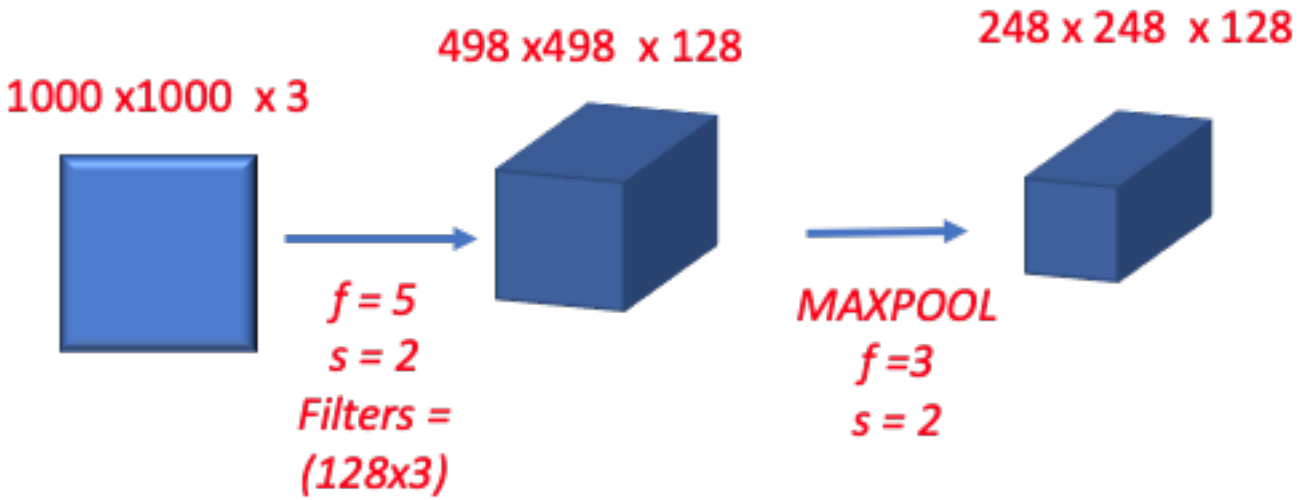
$n - f + 1$

98 x 98 x 4

These are the activations for the next layer $(a^{[1]})$

Demo of convolution at https://cs231n.github.io/assets/conv-demo/index.html

MaxPooling


1000 x1000 x 3
498 x498 x 128
248 x 248 x 128
f = 5
s = 2
Filters =
(128x3)
MAXPOOL
f =3
s = 2

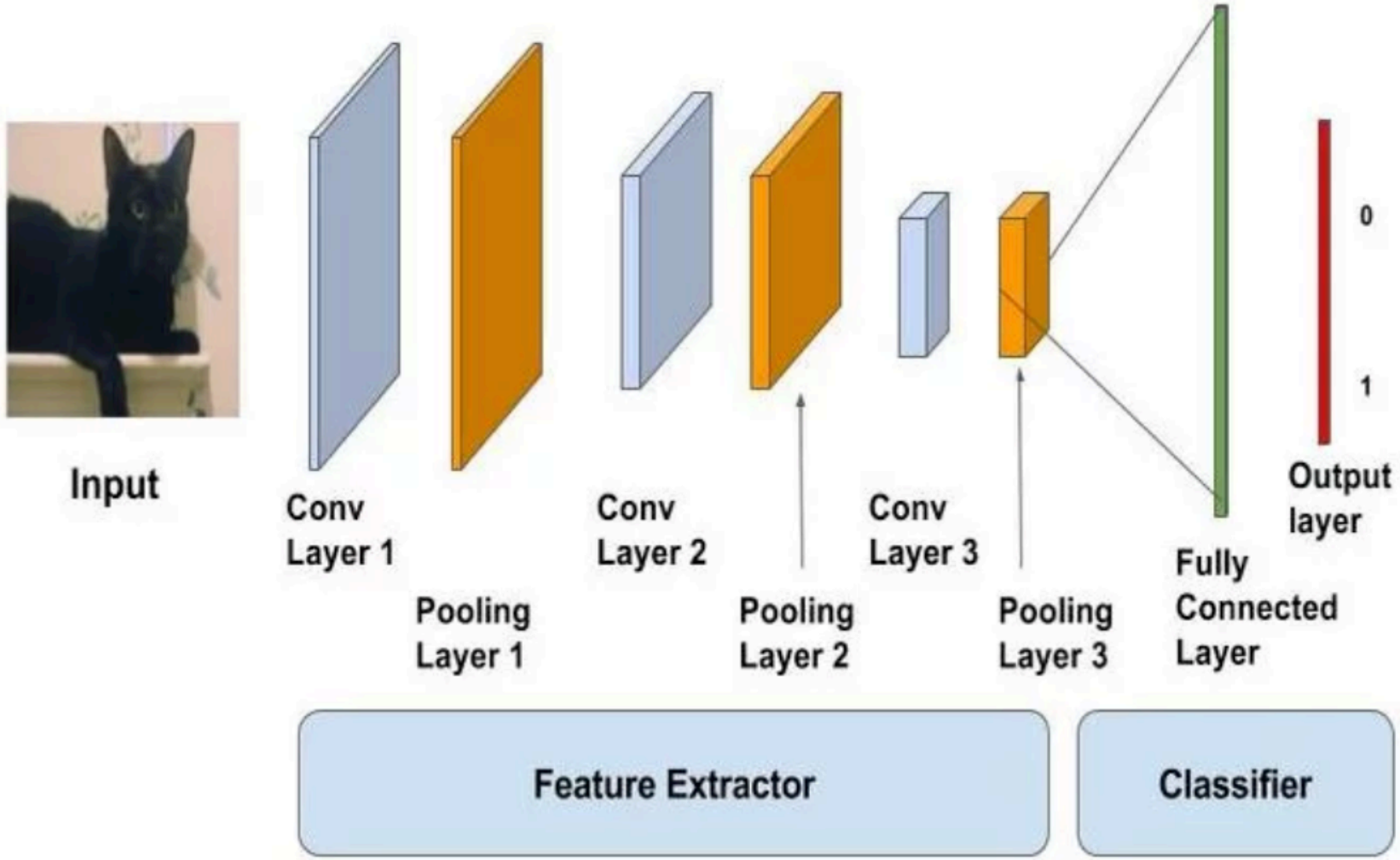
Pooling

Goal: Reduce number of parameters in the model

480 x 640 = 307,200

84 x 111 = 9,324

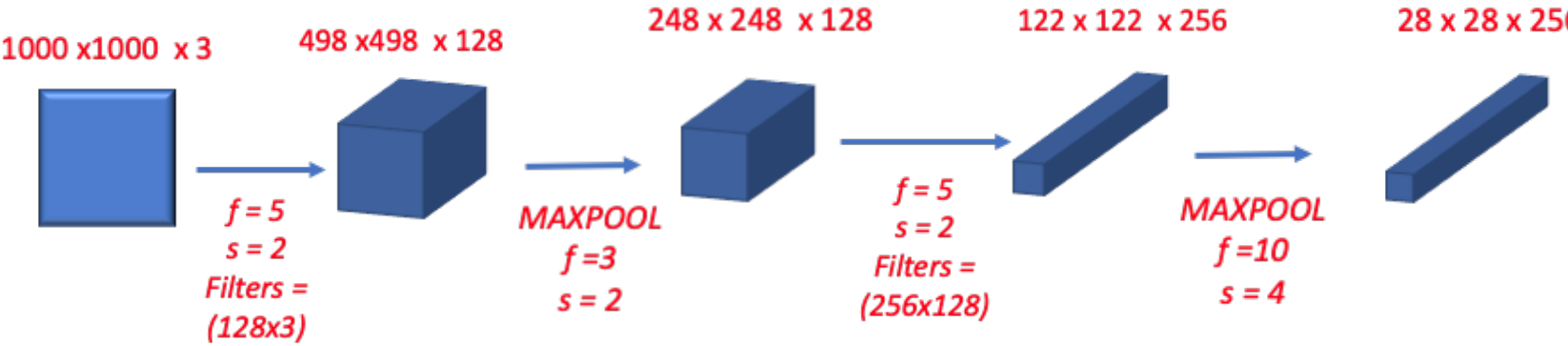
Think of it as down sampling an image.

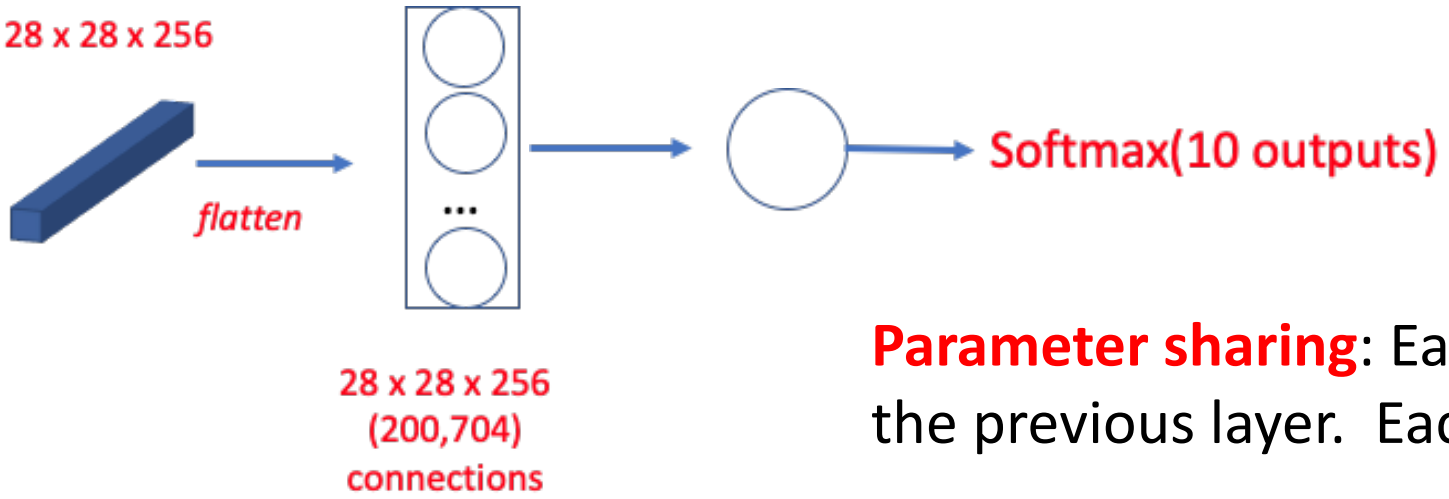


Trend: Each layer shrinks the dimensions of the data/image but the number of filters increase

Image from towardsdatascience.com

Why Convolutions Work





Parameter sharing: Each filter is used by all the data from the previous layer. Each act as a feature detector

Translation Invariance: Maxpooling helps find the features wherever they reside in the input data.

Normalization

Do we need to normalize data for NNs?

Overfitting

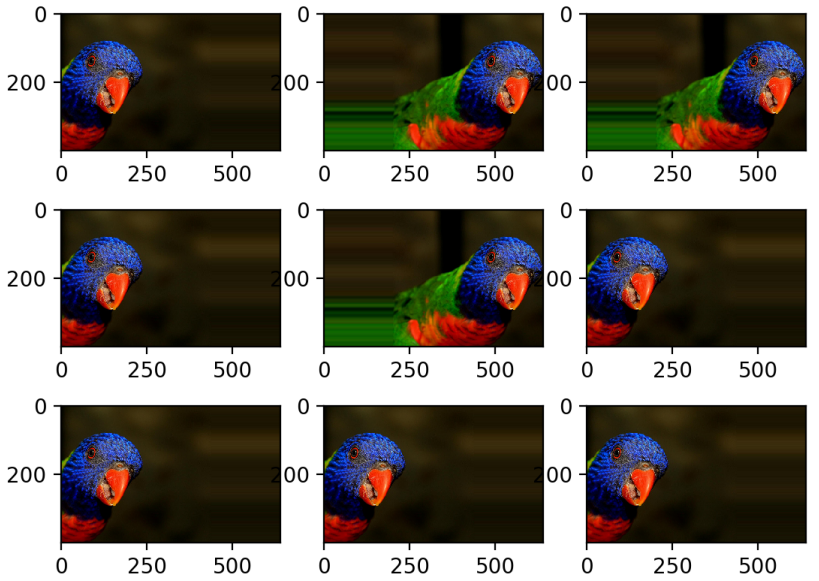
Ideas on Preventing overfitting?

What about Ensemble learning?

Dropout layers

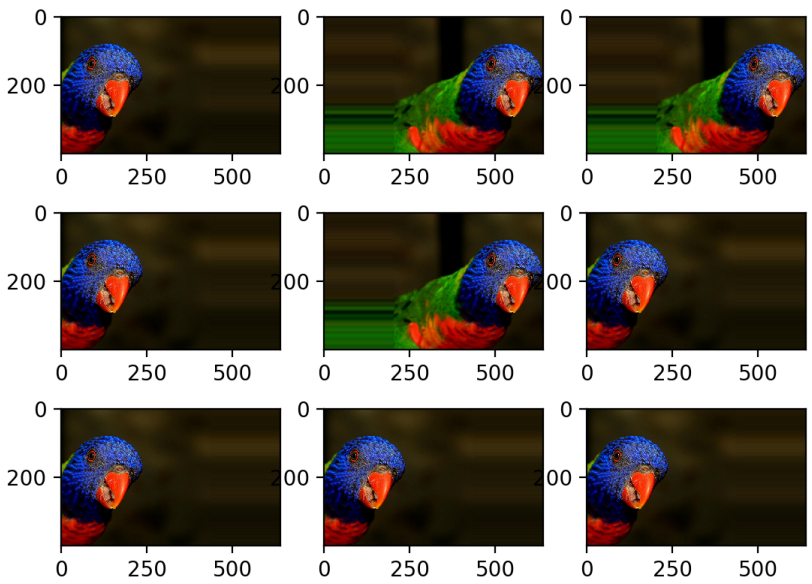
Dropout layers (Mode.add(Dropout(0.2))

Image Augmentation to Prevent Overfitting



Images from Machinelearningmastery.com

Image Augmentation to Prevent Overfitting



```
from keras.preprocessing.image import
ImageDataGenerator
train_gen = ImageDataGenerator(rotation_range=8,
    width_shift_range=0.08, shear_range=0.3,
    height_shift_range=0.08,zoom_range=0.08 )

training_set= train_gen.flow(X_train, y_train,
    batch_size=64)

classifier.fit_generator(training_set,
    steps_per_epoch=60000//64, epochs=5)
```

Images from Machinelearningmastery.com

Hyperparameters

What are the hyper-parameters for CNNs?

- Number of epochs?
- Dropouts
- Maxpooling and network size and shape