Convolutional Neural Network

UW CSE 473
March/4th 2019
Beibin Li

Review

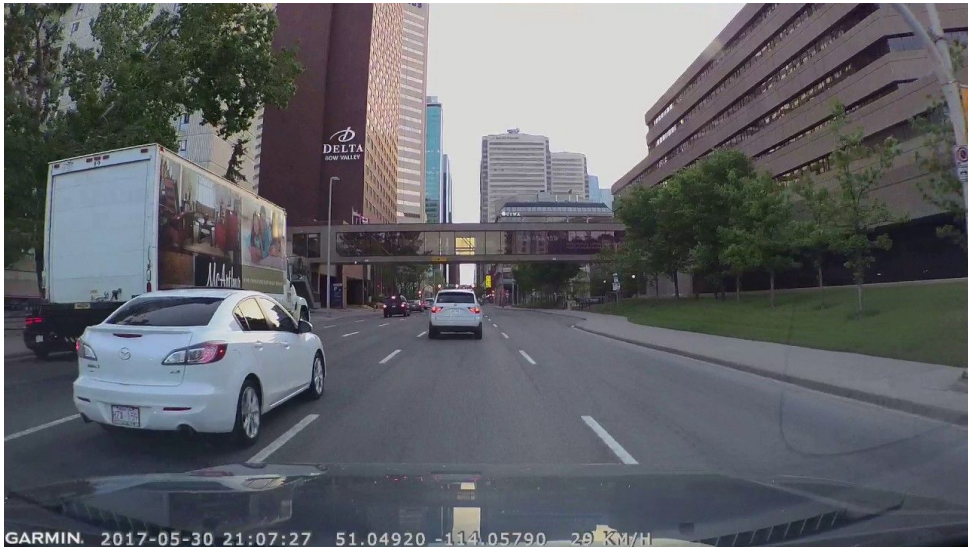
Which Direction

Regression:

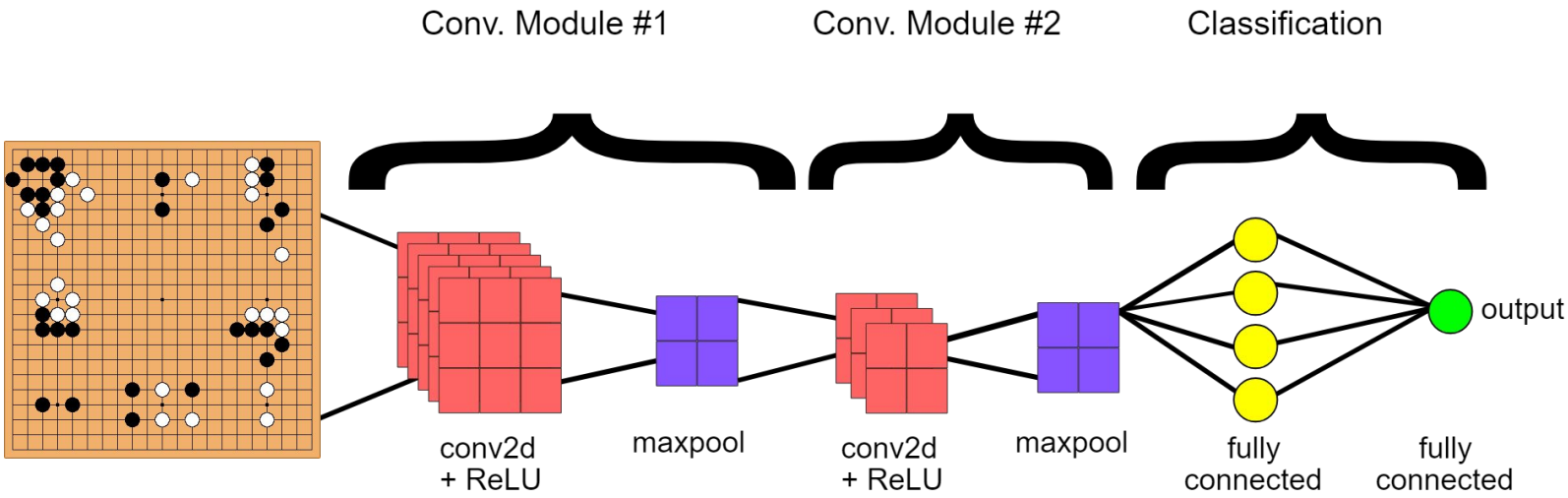
Angle = [-540°, 540°]

Classification:

- Turn left
- Turn right
- Stay Still

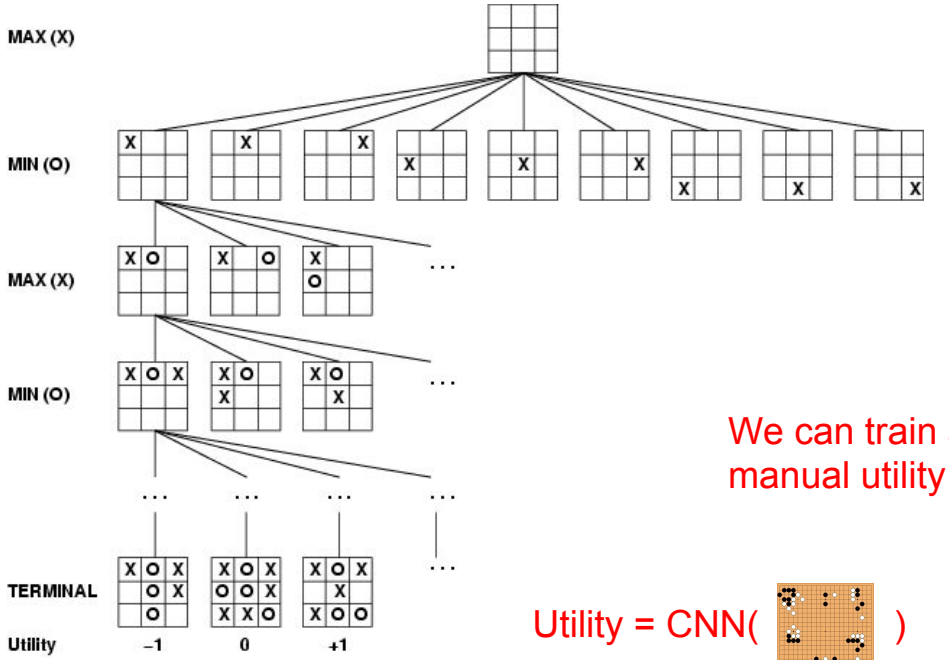


Which Move



https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks

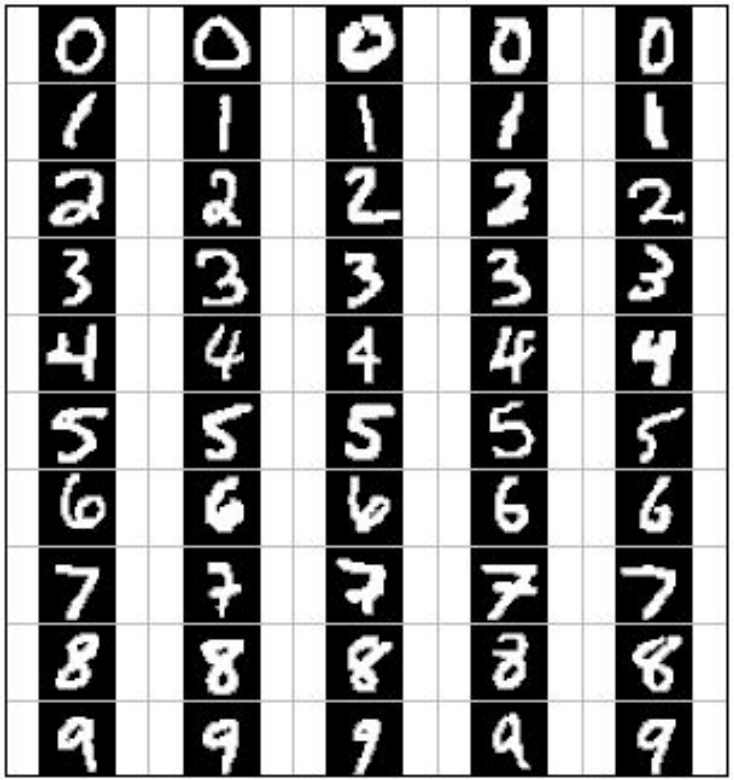
Design Utility Function (HW 3)



We can train a CNN to replace the manual utility function

Utility = CNN( )

Which Digit?

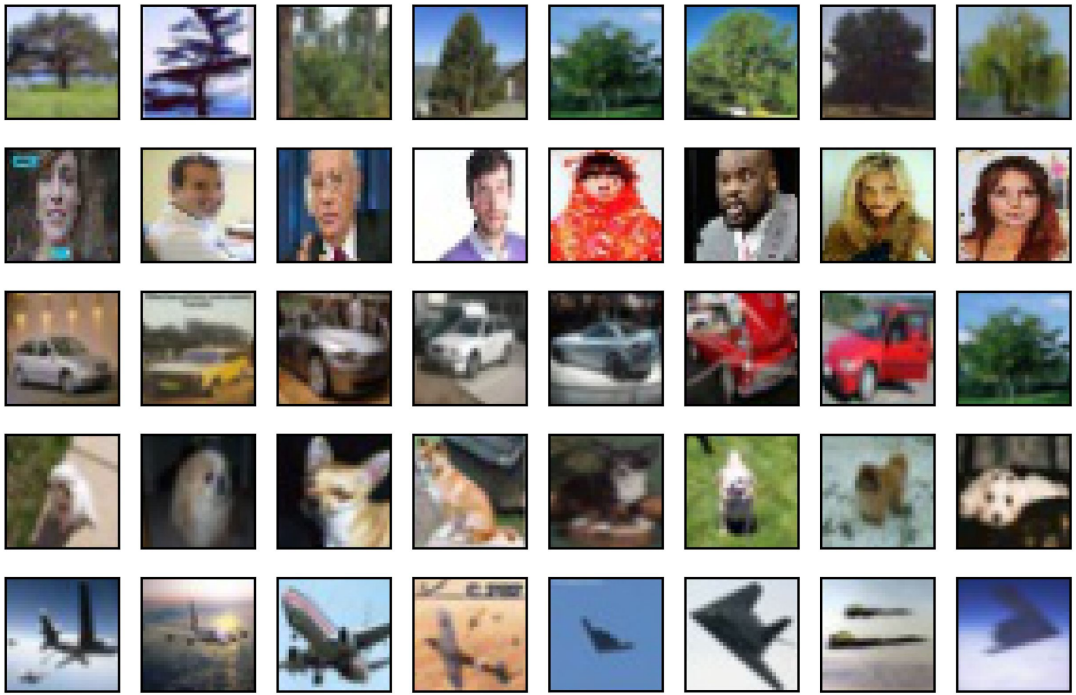


Which Object?

Homework 5

Image Classification:

- Tree
- Face
- Car
- Dog
- Plane



Homework 5

- How many trainable parameters are in the model?

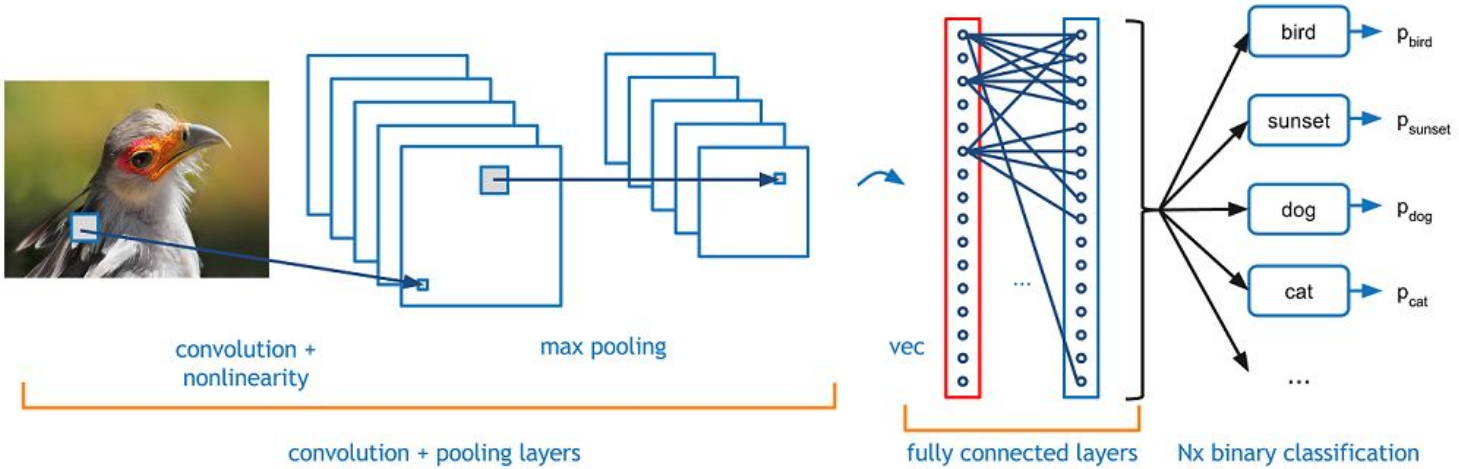
- What is the best training accuracy?

- What is the best validation accuracy? Is it better than the ones in previous questions?

- According to the training and validation accuracies, does the model overfit your training data?

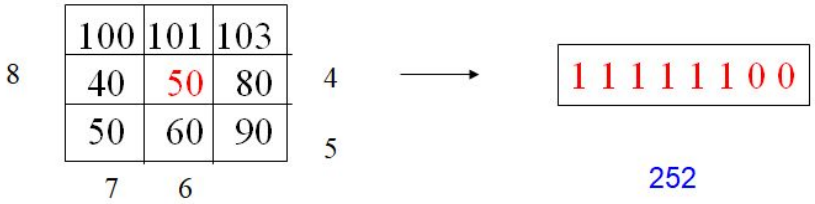
Note: It is ok if your accuracy is not the same as your friend’s.

CNN

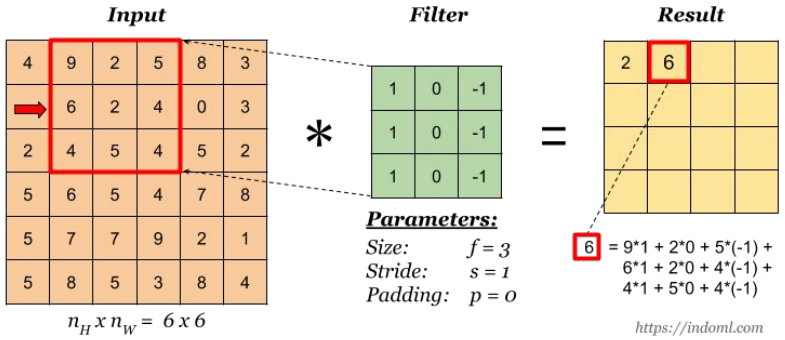

bird
sunset
dog
cat
...
p_{bird}
p_{sunset}
p_{dog}
p_{cat}
convolution + nonlinearity
max pooling
vec
convolution + pooling layers
fully connected layers
Nx binary classification

Convolution: from fixed to learnable

LBP in HW 4



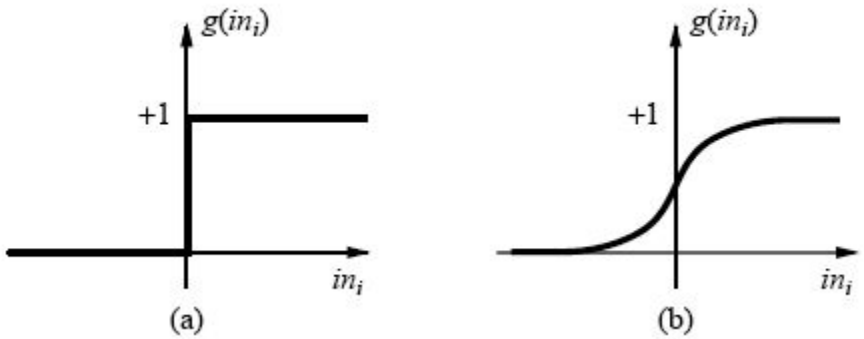
Convolution in CNN and HW5



Expert-Designed Convolution: SIFT, HoG, LBP, …

Learn Flexible Parameters

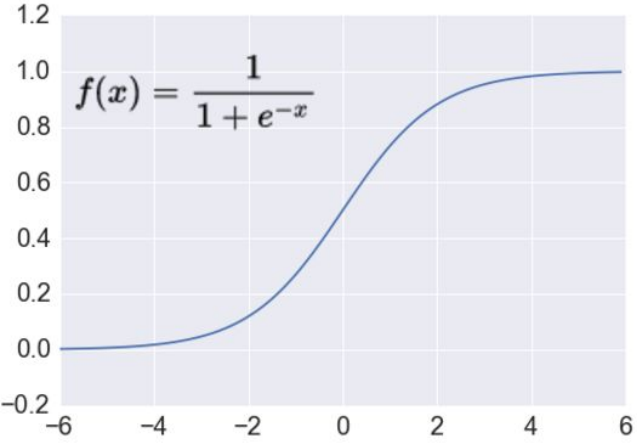
FC Activation: From Step Function to Sigmoid



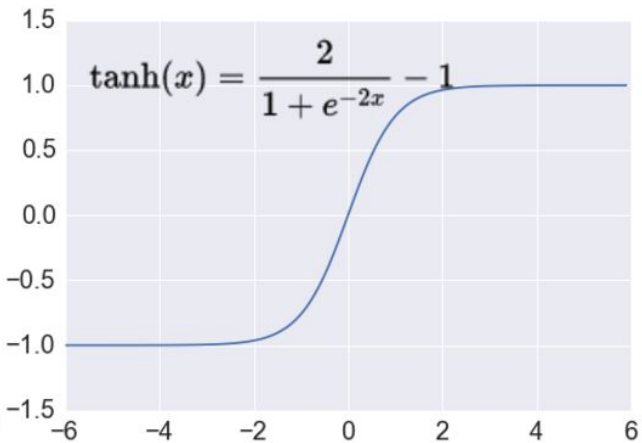
Note: In homework 5, because there is only one FC layer, we do not need nn.Sigmoid()

CNN Activation: From Sigmoid to ReLU

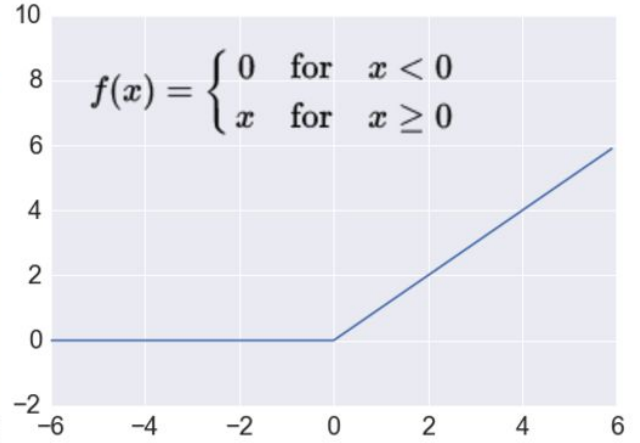
Sigmoid



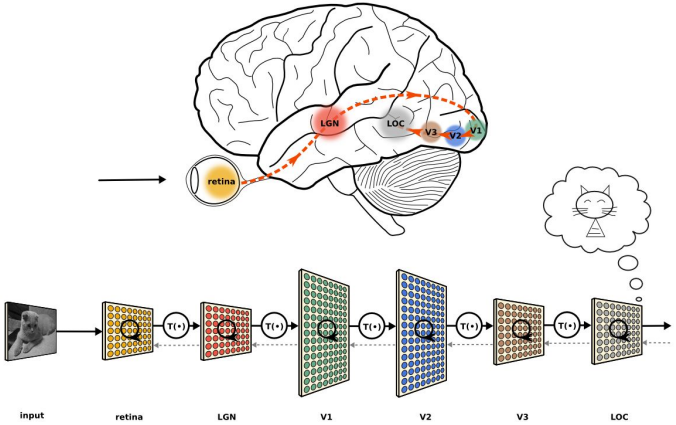
TanH

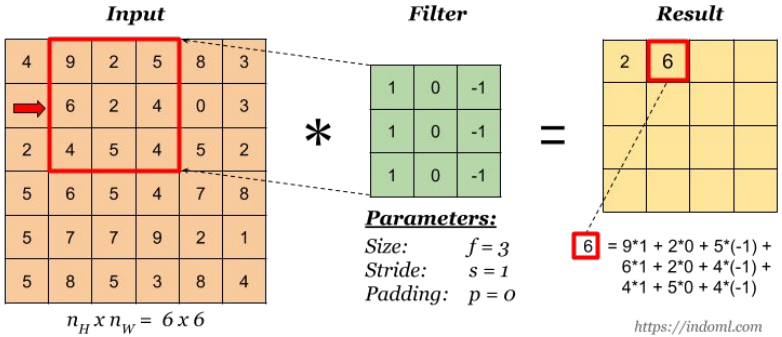


ReLU



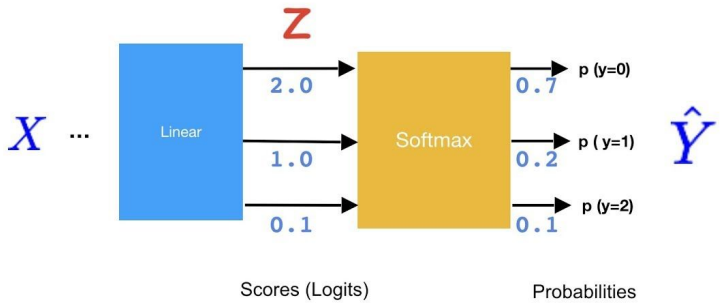
Representation



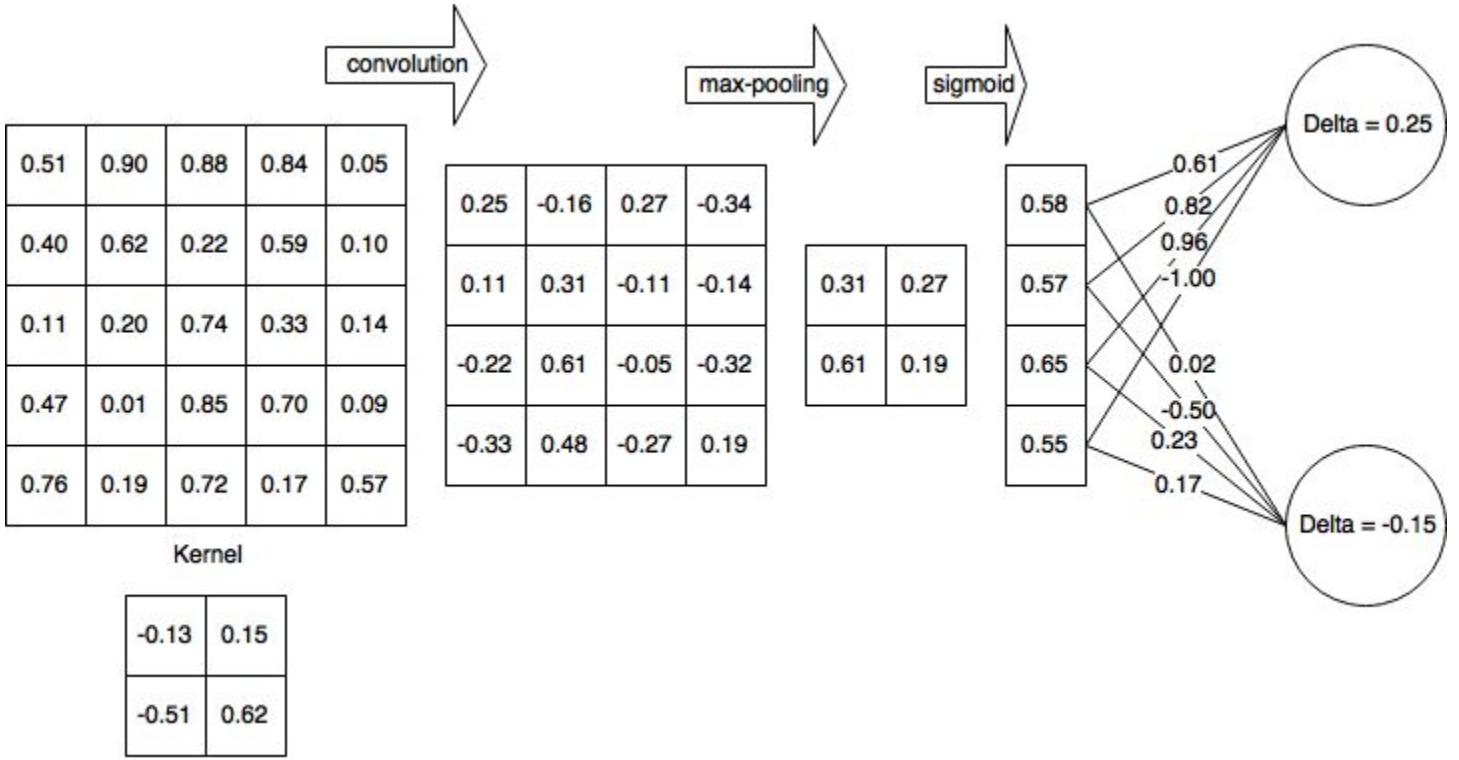


Meet Softmax

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$



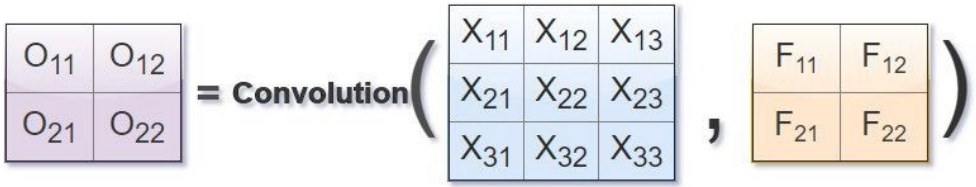
Inference



https://datascience.stackexchange.com/questions/27506/back-propagation-in-cnn

Note: In homework 5, you need to reshape the tensor between Convolutional and Fully-Connected Layers

Learning



$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

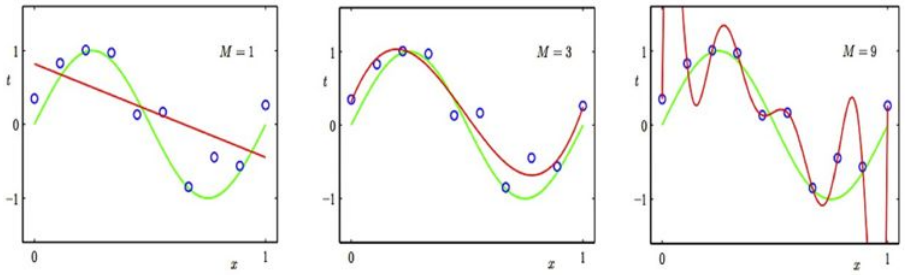
$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

- Details:
- https://www.slideshare.net/EdwinEfranJimnezLepe/example-feedforward-backpropagation
- https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Underfit v.s. Overfit

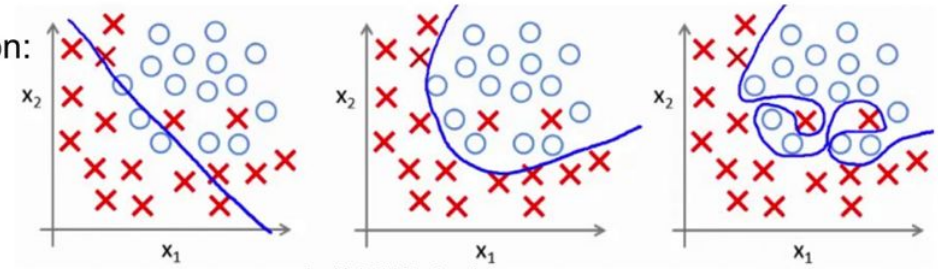
Regression:



predictor too inflexible:
cannot capture pattern

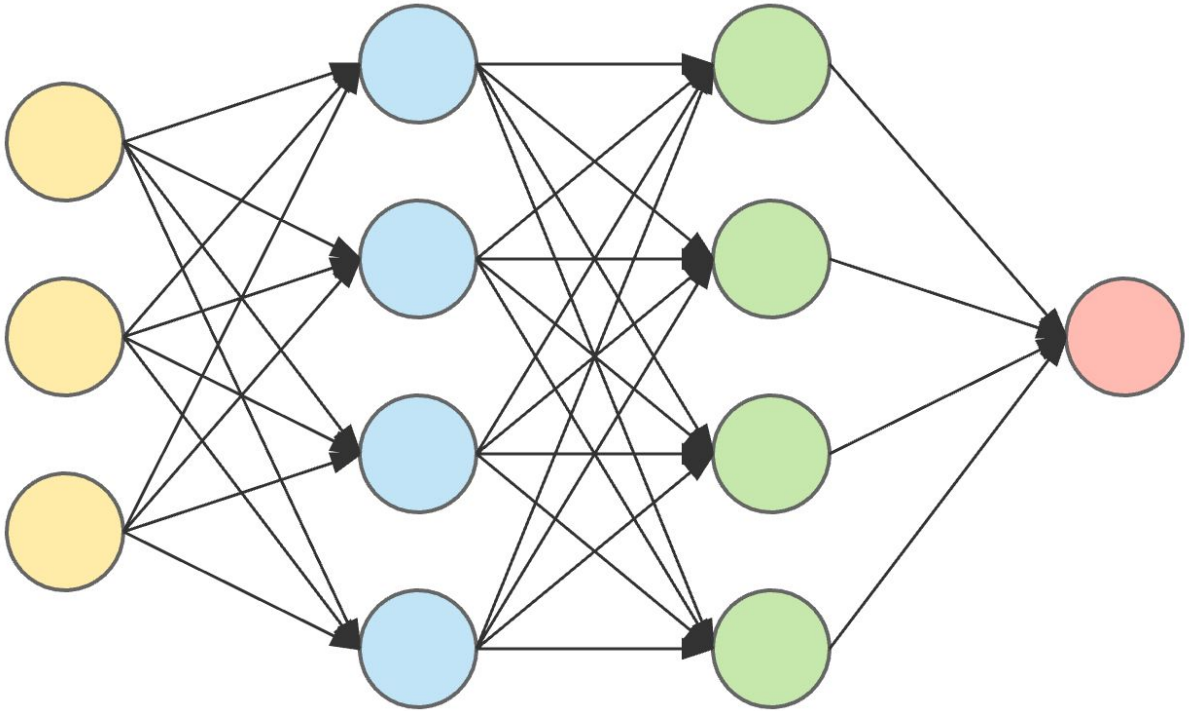
predictor too flexible:
fits noise in the data

Classification:



Homework 5

Neural Network (Q1)



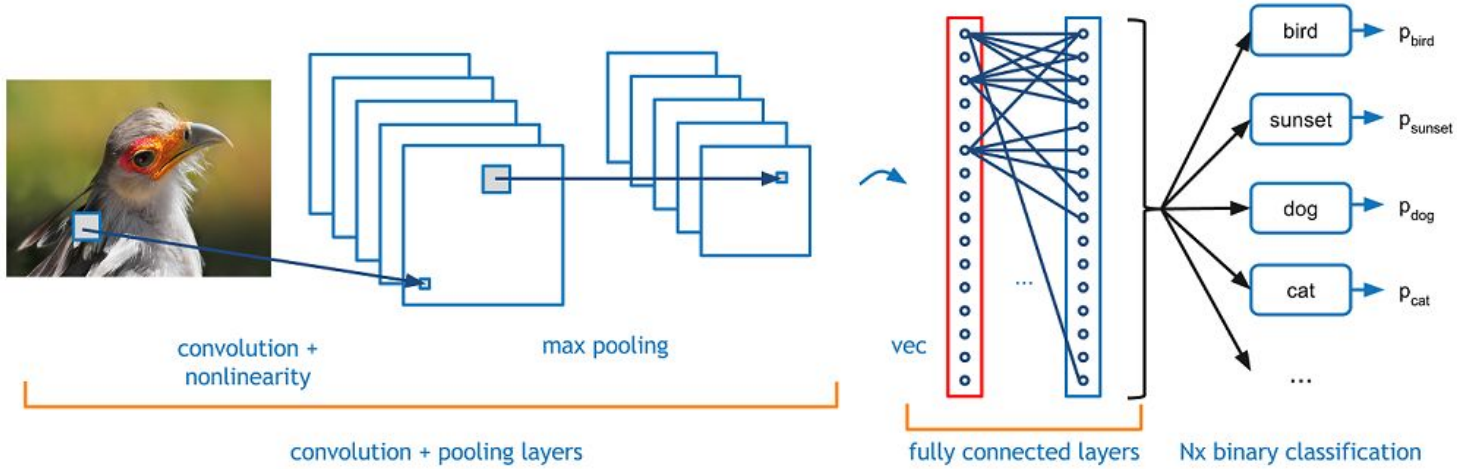
Convolutional Neural Network (Q2)

Conv

Pool

FC

Cross Entropy



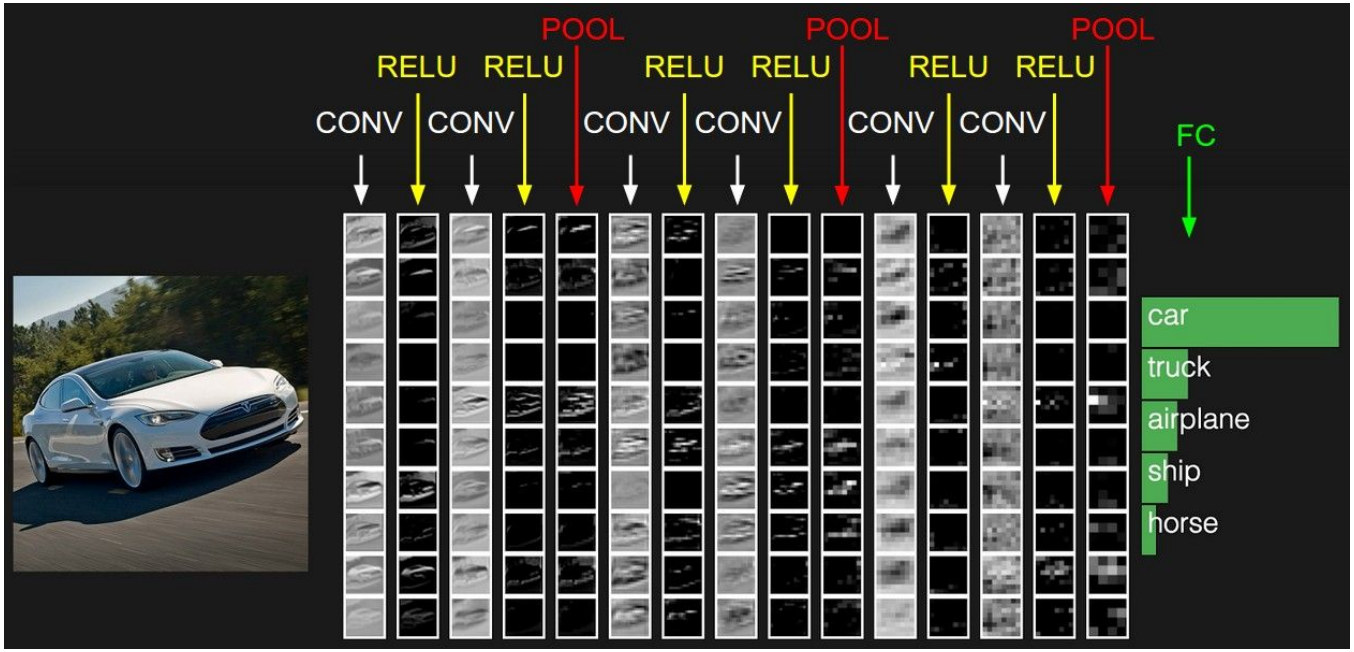
Yellow or Blue?

Color Normalization (Q3)

That is NOT Enough

We Have to Go ***DEEPER***

Deep Convolutional Neural Network (Q4)



Make the Design More Flexible

Input:

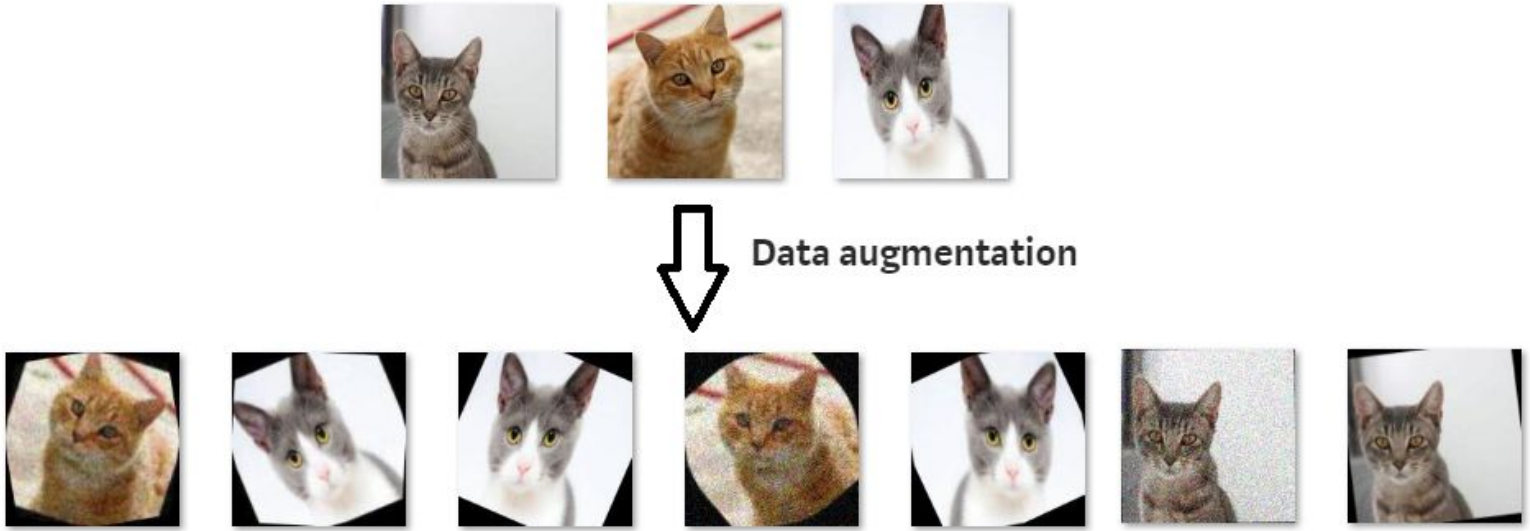
[8, 16, 32, "pool"]

Layer	Output Size	Output Channels
Input	30 x 30	3
Conv	28 x 28	8
ReLU	28 x 28	8
Conv	26 x 26	16
ReLU	26 x 26	16
Conv	24 x 24	32
ReLU	24 x 24	32
Max Pool	12 x 12	32
Linear	5	

Exercise: Input [8, ‘pool’, 16, "pool"]

Data Augmentation (Q5)

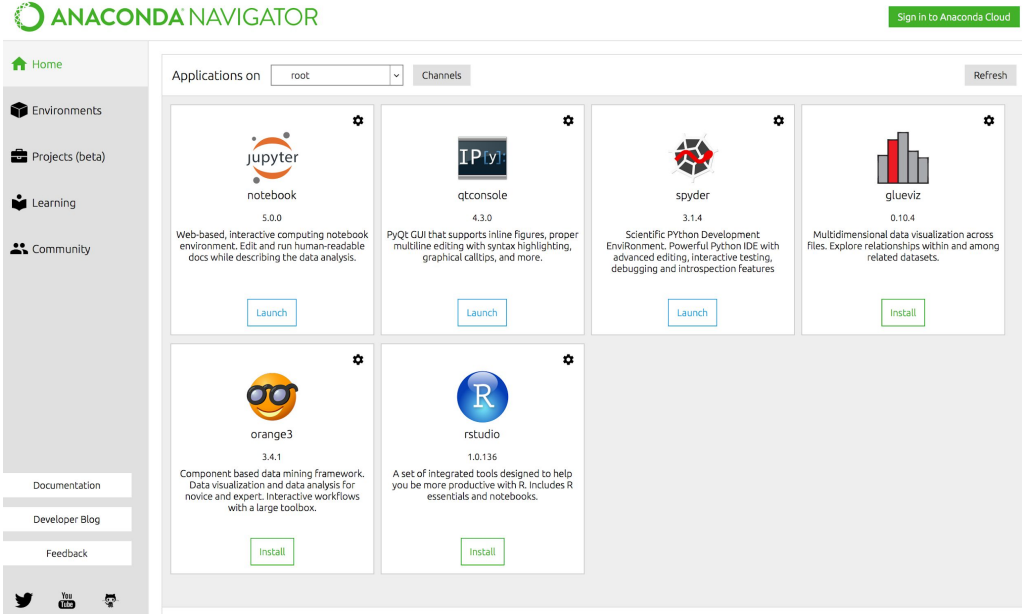
Random Affine Transformation



Starter Code

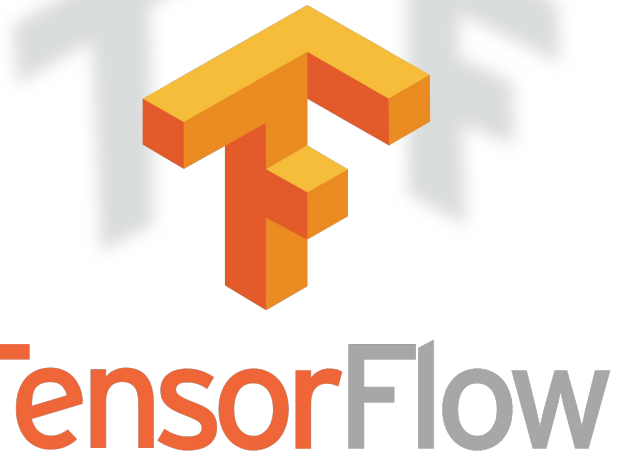
Python 3 and Anaconda

- Python: Created at 1991
- Python 1.0: released at 1994
- Lambda, Map, Reduce, etc
- Exception Handling
- Python 2.0: at 2000
- Garbage collection
- Unicode Support
- Python 3.0: at 2008
- Performance & Design Improvements
- Syntax is different, and not backwards compatible



- (optional if you are already familiar with Python)
- Package management
- Environment control
- Nice IDE (Spyder) and debugger
- Jupyter Notebook
- and more…

Deep Learning Libraries



PyTorch

- Dynamic Graph
- Easy and Flexible
- Popular
- Well maintained
- Compatible with Torch and Caffe
- more...