Lecture 2 Image Classification

The difficulty of image classification

semantic gap

- Viewpoint variation
- Illumination
- Deformation
- Occlusion
- Background Clutter
- Intraclass variation

Model building

KNN

- drawback
  - Very slow at test time
  - Distance metrics on pixels are not informative
    - the pixel distance don't correspond very well to perceptual similarity
      - different image have the same pixel distance
    - L1 && L2 is not a good measure of similarity between images
  - Curse of dimensionality
    - 要想很好地建模，希望样本点能密集地填充整个区域，这意味着随着维度的增加，需要的数据量是几何增加的
- DIstance measurement
  - L1
    - L1 distance depends on your choice of coordinates system
  - L2

Hyperparameters

- These are hyperparameters: choices about the algorithm that we set rather than learn
  - Training 的过程不能学习到hyperparameter的值
- 如何设置Hyperparameter

Idea #1: Choose hyperparameters that work best on the data

BAD: K = 1 always works perfectly on training data

Your Dataset

Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data

BAD: No idea how algorithm will perform on new data

train	test

Idea #3: Split data into **train**, **val**, and **test**; choose hyperparameters on val and evaluate on test

Better!

比较不同的hyperparameter，获得最优的hyperparameter

用于比较不同模型的效果（不同算法）

按照设置好的hyperparameter，获得最优的参数

train	validation	test

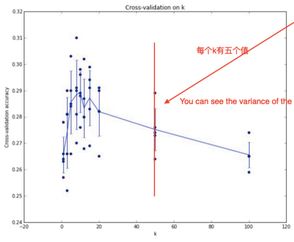
Idea #4: **Cross-Validation**: Split data into **folds**, try each fold as validation and average the results

fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test

Useful for small datasets, but not used too frequently in deep learning

- **How to understand the result of k-folds learning?**

Setting Hyperparameters



Example of 5-fold cross-validation for the value of **k.**

Each point: single outcome.

The line goes through the mean, bars indicated standard deviation

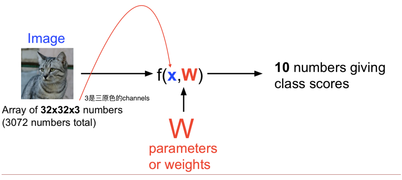
(Seems that k ~= 7 works best for this data)

Linear Classification

The relationship between Neural Network and Linear classifiers：

Parametric approach

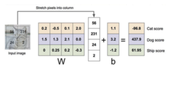
- main idea：提供x和参数，能过得一个评分



- f(x, W)是一个general form
  - The whole story in deep learning is coming up with the right Structure for this function F
- One instance
  - Linear Classifier：An instance of F，simply multiple them
  - f(x, W) = Wx + b
    - bias that does not interact with the training data and instead just gives us some sort of data independent Preferences for some classes over anohter
  - Pointview of this f

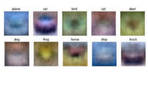
Algebraic Viewpoint

f(x,W) = Wx



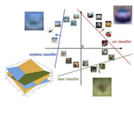
Visual Viewpoint

One template per class



Geometric Viewpoint

Hyperplanes cutting up space



- Visual Viewpoint
  - template matching
    - inner product gives the similarity between the template and the image
    - This gives us the intuitive understanding of what the linear classification doing
    - We can actually take the rows of that weight matrix and unravel them back into images and actually visualize those templates as images
    - The Problem is that the linear classification only learn one template for each class. So if there is some variations appear，it's trying to average out all those different variations

Model building finish

How to come up with a W?