CS231n- Lecture 2

February 11, 2017

1 Image Classification

```
Take image and classify into one label
  Semantic Gap: 300 x 100 x 3 being the number of RGB number and size
  Problems:
    Work with millions of numbers is hard.
    Intrinsics of Cameras are different
    Illumination affects images
    Deformation affects images
    Occlusion ( can't see full objects like cat)
    Background color
    Intra Class Variation
  Rule based approaches are super hard because it is not scalable.
  Data has tremendously increased => better training
  First Classifier:
    Nearest Neighbor Classifier
    CIFAR-10
    10 labels
    50k training images, 32x32
    10k test images
    How can we compare Images: Distance metric
      Manahattan Distance(L1)
      Return label with argmin(Md)
    Linearly slower as the training set is bigger.
    CNNs work much faster after training is done(const)
    We can use L2(Euclidean Dist)
    This distance is a hyperparameter.
    If we generalise it to the k-Nearest Neighbors we'll get k nearest neighbors
    and then vote
    As k increases it tends to smooth out.
    Choice of k is a hyperparameter too
    Nearest neighbor gives 100% accuracy on training set when using
    Euclidean norm or Manhattan Norm
    K-nearest neighbor doesn't give a 100% accuracy on training set
```

Hyperparameter testing can be done with trial and errors
Split training and test data to avoid such bs.
We separate Training data sometimes to 5 folds and then train on 4 folds
and test on the other one(say 5th fold to tune hypparams) This fold 5 is cal
called Validation set Cross Validation is iterating accross these folds as
validation sets and then average for each.
We find k=7 to be best as it is peeking a plot of acc vs k
k-NN is never used: Terrible performance at testing time
  Distance metrics on level of whole images can be unintuitive

Linear Classification:
NNs can see, hear, language, control(robots)
Building blocks involved.
Image Captioning:
  CNN(to see) -> RNN (to model sequences(in this case words))
KNN is nonparametric
in Lin Classification:f(x,W) x-image, W is parameters to give 10 numbers
indicating class scores
Image [32x32x3] we stretch it into a long vector

```
Suppose f(x,W)= W          x        + b
          10x1  10x3072    3072x1    10x1
```

We resize images in this approach to 32x32 although not optimal
Score is a weighted sum of all the pixel vals
if you reshape rows of W to image we see a weirdish template image of each
label
Mostly only captures colors which means it is not good at all
GreyScale will work terribly in linear classifiers too
The score is the output we get from this fn