Learning from Data lecture notes

Regularization part 2

* Normal fitting (lin reg)
  + Transform inputs
  + Minimize
* Fitting with regularization
  + Add in optimization constraint, aka
    - Make sure this constraint disallows else you gain nothing
    - Intuitively this constraint requires to stay small
    - Called ‘weight decay’ (bc when applied iteratively, decays the weight a certain amount after each iteration before moving along gradient)
  + Solution to above called
* Solving for
  + Minimize the above, (and we well obtain a solution)
  + decided by validation, inversely related to
* Augmentation error
  + Minimizing
  + So we can conduct vc analysis (with former), and lets you easily optimize (with the latter)
  + is a better estimate of than
* Variations of weight decay
  + - Weight the importance of the weights
    - very strongly penalizes high order terms. Tries to get low order fits
    - is the opposite
    - For Neural Networks, weigh weights based on the layer it is in
    - Tikhonov Regularizer: (matrix selected, very well-studied algorithm)
* Penalizing small weights
  + Impact on is bad. Exponential growth
  + Does this mean don’t use regularizers?
    - It means follow guidelines
    - It means validate value of
  + Practical noise
    - Stochastic noise is ‘high frequency’
    - Deterministic noise is ‘non-smooth’
    - Hence bias regularizer to fit more smooth functions (hence counteract noise more than actual signal)
* Choosing a regularizer
  + Perfect regularizer = restricts in direction of target function
  + Regularizers try to incorporate bias that harms overfitting more than fitting regularizers bias for smoother functions (because noise is not smooth)
    - regularizers bias for ‘average’ solutions (won Netflix competition)
  + If we choose a bad regularizer, can still save us (validation will reveal )
* Neural network regularizers
  + Weight decay
    - If weights are very small, and will only have the hypothesis set of linear regression
    - As weights can increase, nonlinearities are introduced
    - Eventually almost every function can be modeled
  + Weight elimination
    - Why not force some weights to be 0? Reduces VC dimension
    - Clusters weights into 2 groups, big weights and near-0 weights
  + Early stopping
    - Doesn’t mean stop on local minima vs global minima
    - Doesn’t mean sloppy optimizer is okay (behavior can’t be properly analyzed)
* Optimal
  + On case studies/tests, deterministic noise/stochastic noise both react remarkably similarly with respect to values