Hyperparameter Tuning

Team DGMRW

What are Hyperparameters?

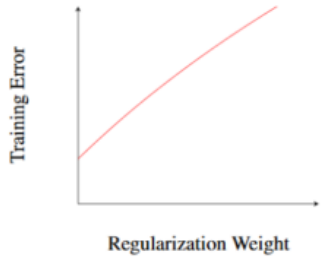
- Hyperparameters are parameters that control how the model learns
- Differ from parameters
  - Parameters are selected through training
  - Hyperparameters are set before training
- Examples: number of layers in neural network, learning rate
- Wide variety of hyperparameters makes it difficult for definite solution to optimizing

Common Hyperparameters

Ridge Regression Regularization Term

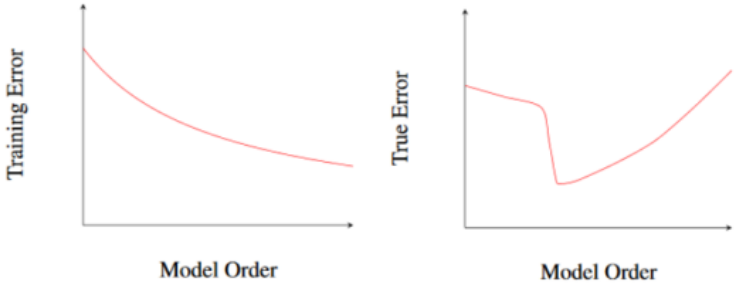
$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$$

- Lambda is a hyperparameter in the Ridge Regression problem
- Controls how much we care about penalizing the size of solution
  - No lambda reduces problem to Least Squares
  - Extremely high lambda means we care a lot about norm of solution
  - Setting lambda to 0 completely eliminates training error
- So far in this class we have set lambda arbitrarily or through guess and check
- Tradeoff between overfitting and underfitting



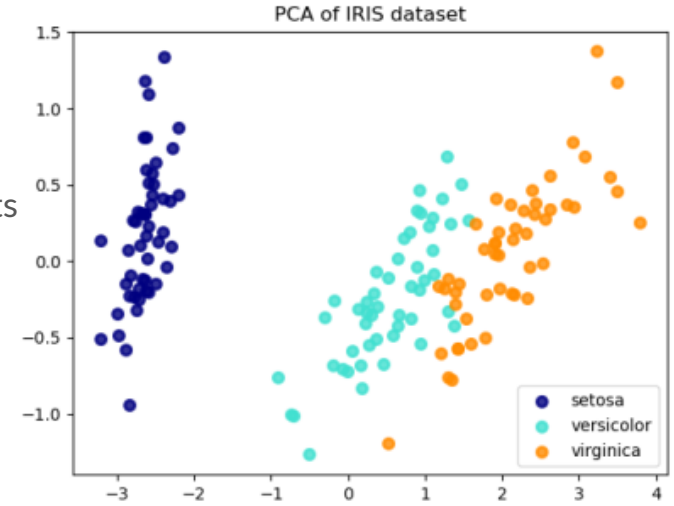
Degree/Type of Approximating Function

- Two parameters when approximating functions
  - Type of approximation (Fourier vs polynomial)
  - Degree
- Training error can be reduced through increasing order
  - Increasing too much leads to overfitting
- Hard to choose type of approximation



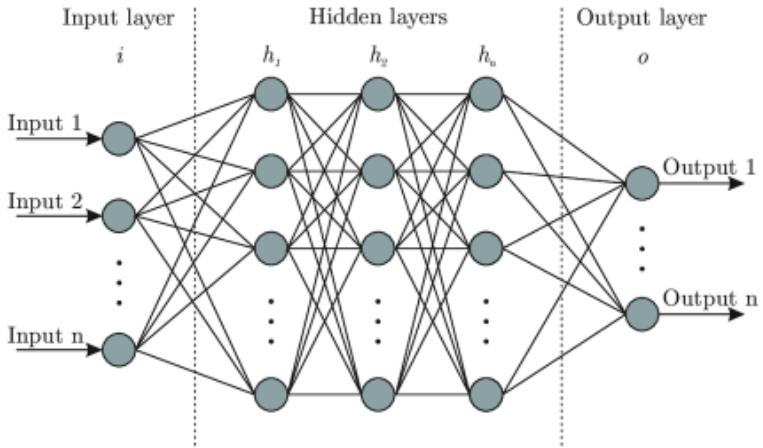
PCA Projection

- PCA is used for dimensionality reduction
- One hyperparameter is dimensionality of input
  - Sometimes low rank approximation of input performs better
- Finding optimal input dimension can produce better results



Architecture of Neural Network

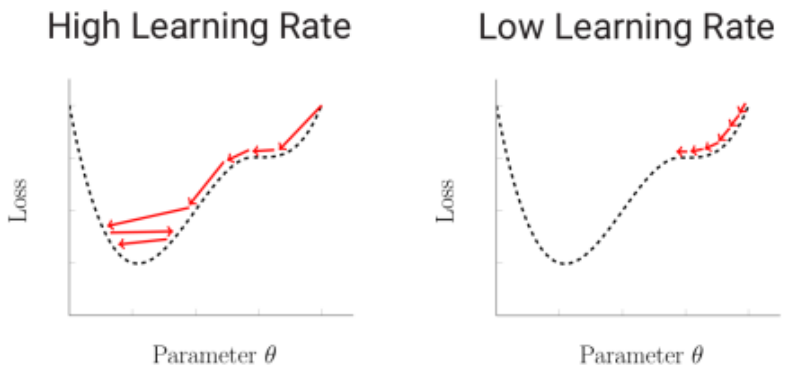
- Multiple hyperparameters in Neural Network
  - Number of hidden layers
  - Number of neurons in each hidden layer
  - Learning Rate
  - Activation Function



Learning Rate/Step Size in Gradient Descent

- Learning Rate controls how much we update in each step
- Setting it too large
  - Overshoot goal
  - Never converge
- Setting it too small
  - Get trapped local minima
  - Convergence very slow

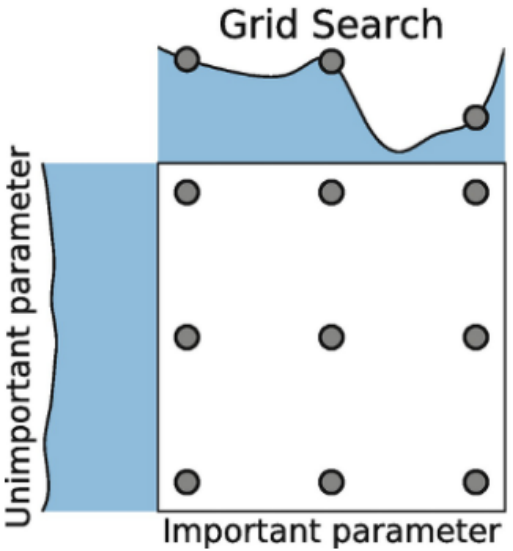
$$x_{k+1} \leftarrow x_k - \alpha \nabla f(x_k)$$



Types of Tuning

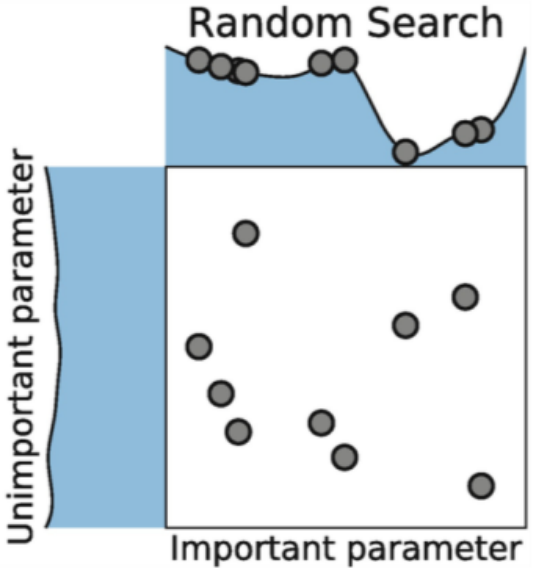
Grid Search

- “Brute Force” method
- Searches through every combination of hyperparameters
- Can be very costly if number of parameters are high
- Performance of a combination is measured through cross validation



Random Search

- Similar to Grid Search but only uses a subset of combinations
- Much more efficient especially with high dimensionality
- Control number of combinations searched



Grid Search vs Random Search

- For hyperparameters of small dimension, both methods are about the same
- Grid Search is inefficient when dealing with large number of hyperparameters
  - Has to exhaust every combination
- Random search is much better for large dimensionality
  - Preset number of combinations
  - May not be as accurate as Grid Search
  - Time vs Accuracy