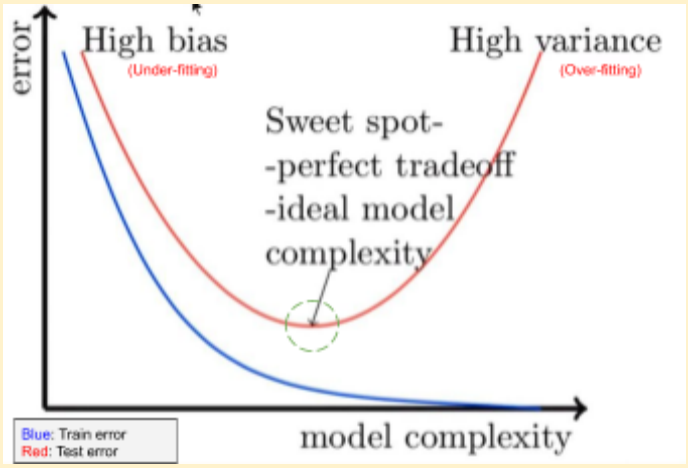
PadhAI: Regularization

One Fourth Labs

Overfitting in deep neural networks

Why do we care about bias variance trade-off in the context of Deep Learning

1. Consider the same image from the previous section



2. Deep Neural Networks are highly complex models (many parameters and many non-linearities)
3. Easy to overfit (drive training error to 0)
4. The aim is to maintain the model complexity near the sweet-spot and not have it get too complex.
5. How do we deal with this in practice in Deep Neural Networks? Let's look at some of the recommended practices
6. Divide data into train, test and validation/development splits
   a. Good rations would be (60:20:20) or (70:20:10) in the order train:validation:test
   b. Never handle the test data except during the final evaluation. All other evaluation must be done with the training set first then the validation set.
   c. Training data is used to minimise the loss/error
   d. Validation data is used to check if the model has become too complex or not.
   e. We must aim to get a good score during evaluation of the validation set
7. Start with some network configuration (say, 2 hidden layers, 50-100 neurons each)
8. Make sure that you are using the:
   a. Right activation function (tanh(RNN), ReLU(CNN), leaky ReLU(CNN))
   b. Right initialisation method (Xavier, He)
   c. Right optimization method (say Adam)
9. Monitoring training and validation error similar to the figure in point number 1.

Training Error	Validation Error	Cause	Solution
High	High	High Bias	• Increase model complexity • Train for more epochs
Low	High	High Variance	• Add more training data (**dataset augmentation**) • Use **regularization** • Use **early stopping** (train less)
Low	Low	Perfect trade-off	• You are done!