CMS 165 Foundations of Machine Learning Homework 4

Due date: March 1, 4pm

Question 1.

Consider the expected loss function of the linear regressor $w \in \mathbb{R}^d$:

$$L(w) = \mathbb{E}_{X,Y}(Y - w^\top X)^2$$

where $X \in \mathbb{R}^d$ and $Y \in \mathbb{R}$ are random variables with some joint distribution D. Let $w^* \in \operatorname{argmin}_w L(w)$ be an optimal linear regressor. Let $\hat{w}$ be the empirical best least-squares estimator trained on some training set T picked i.i.d according to D. Prove that

$$\mathbb{E}_T\, L(\hat{w}) = \mathbb{E}_X[VAR(Y|X)] + \mathbb{E}_X(\mathbb{E}\,[Y|X] - (w^*)^\top X)^2 + \mathbb{E}_X((w^*)^\top X - \mathbb{E}_T\,[\hat{w}^\top]X)^2 + \mathbb{E}_{X,T}(\mathbb{E}_T\,[\hat{w}^\top]X - \hat{w}^\top X)^2$$

This roughly means that the expected loss of our empirical estimator can be decomposed as noise variance + approximation error of function class + estimation bias + estimation variance. (Hint: You are free to assume the Bias-Variance Tradeoff without proof. Try rewriting the bias term to get the four terms we are asking for.)

Question 2.

This question is aimed to give hands-on experience on deep neural networks. Please train a neural network that achieves **74% test accuracy in CIFAR-10 dataset** classification task. You can use any library you want. An example Convolutional Neural Network notebook using Pytorch is given in Resources Section of Piazza (`https://piazza.com/class_profile/get_resource/jq1tbyrptsw6ce/jsi4g2e19gx601`). You can modify that code to achieve the sufficient test accuracy. Feel free to use other resources for this question, but please cite the resource or open source code that you use.

In your submission, please include the **scheme of the architecture** you used and the **evolution of training and test accuracies**. You can measure the evolution of test accuracy by saving the network model and/or weights periodically during training and then testing them on the test dataset after you finish training. Please also include **a brief discussion of your observations** while trying to achieve this accuracy. (The ways you tried, what worked, what didn't work...) In your model, try to have the size of each layer easily modifiable.