COMS 6998-013
LLM Based GenAI
Instructor: Parijat Dube and Chen Wang

Homework 1

Due: Sept. 30, 2025

Problem 1 - *Bias Variance Tradeoff, Regularization* 35 points

1. **[5 points]** Derive the bias-variance decomposition for a regression problem, i.e., prove that the expected mean squared error of a regression problem can be written as

$$E[MSE] = Bias^2 + Variance + Noise$$

Hint: Let $y(x) = f(x) + \epsilon$ be the true (unknown) relationship and $\hat{y} = g(x)$ be the model predicted value of y. Then MSE over test instance x_i, $i = 1, \ldots, t$, is given by:

$$MSE = \frac{1}{t}\sum_{i=1}^{t}(f(x_i) + \epsilon - g(x_i))^2$$

2. **[4 points]** Consider the case when $y(x) = x + \sin(1.5x) + \mathcal{N}(0, 0.3)$, where $\mathcal{N}(0, 0.3)$ is normal distribution with mean 0 and variance 0.3. Here $f(x) = x + \sin(1.5x)$ and $\epsilon = \mathcal{N}(0, 0.3)$. Create a dataset of size 20 points by randomly generating samples from y. Display the dataset and $f(x)$. Use scatter plot for y and smooth line plot for $f(x)$.

3. **[8 points]** Use weighted sum of polynomials as an estimator function for $f(x)$, in particular, let the form of estimator function be:

$$g_n(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_n x^n$$

Consider three candidate estimators, g_1, g_3, and g_{10}. Estimate the coefficients of each of the three estimators using the sampled dataset and plot $y(x), f(x), g_1(x), g_3(x), g_{10}(x)$. Which estimator is underfitting? Which one is overfitting?

4. **[8 points]** Generate 100 datasets (each of size 50) by randomly sampling from y. Partition each dataset into training and test set (80/20 split). Next fit the estimators of varying complexity, i.e., $g_1, g_2, ...g_{15}$ using the training set for each dataset. Then calculate and display the squared bias, variance, and error on testing set for each of the estimators showing the tradeoff between bias and variance with model complexity. Can you identify the best model?

5. **[10 points]** One way to increase model bias is by using regularization. Lets take the order 10 polynomial and apply $\mathcal{L}_2$ regularization. You can work with any value of regularization rate. You don't need to tune it. Compare the bias, variance, and MSE of the regularized model with the unregularized order 10 polynomial model? Does the regularized model have a higher or lower bias ? What about MSE? Explain.

Note: For part 2 and 3 of this problem limit the range of x range for the 20 points generated to lie between some range, say 0 and 10, to observe overfitting and underfitting. Remember to use the same range for training and testing. Additionally, please note to sort the points (increasing x) before plotting. The graph must contain a scatter plot of the points and line plot of the functions.

For part 4 of this problem there are two different ways to sample x and y when creating 100 datasets.

- Follow the post https://dustinstansbury.github.io/theclevermachine/bias-variance-tradeoff. The idea is to keep the value of x same across all the 100 datasets. The y values will vary since it contains the noise (Normal distribution) component.
- Sample a test set (of size 10) before sampling any training dataset. Then sample training set (of size 40) for each 100 dataset but make sure that none of the 10 test set samples should show in any of the 100 datasets. So all the datasets share this common test set but their train set is different.

  The key is to have a fixed test set even though you have 100 independently sampled training set

Homework 1

Problem 2 - *Efficiency of Synchronous SGD Across Servers* 15 points

Consider a distributed deep learning experiment with data parallelism and synchronous SGD across multiple servers. Each server has 8 P100 GPUs and batch size per GPU is kept fixed at 64. The training dataset has 131072 images. *Scaling efficiency is defined as ratio of per iteration time when training using one 1 server to per iteration time when training using N servers.* The result from this experiment, showing how the per iteration time scales as the number of servers, are shown in Figure 1.

Number of servers	Time per iteration (secs)
1	0.3
2	0.32
4	0.33
8	0.35
16	0.36
32	0.37
64	0.39
128	0.41
256	0.43

Figure 1: Problem 2

Based on this data, answer the following. Show your calculations to plot the data for the plots.

1. **[2+2 points]** Plot per iteration time and per epoch time vs number of servers on the same plot. Use the primary y-axis for per epoch time and the secondary y-axis for the per iteration time with x-axis being the number of servers. What do you observe about the scaling of per iteration time and per epcoh time with the number of servers? Explain the trends in their scaling behavior.

2. **[2 points]** Calculate throughput (images processed/sec) and plot throughput scaling with the number of GPUs. Write your observations.

3. **[2+2 points]** Plot the scaling efficiency vs number of GPUs. In the same chart (using the secondary y-axis) plot speedup vs the number of GPUs. Write your observations.

4. **[5 points]** Suppose that you can provision these servers on a cloud platform for $2/min. You need to train your model for 70 epochs. If you want your training to be completed in 5 hrs (or less) and your budget is $10,000, which is the most cost effective configuration? If there is no configuration in the given cost budget, how much should you relax your budget to find the most cost-effective configuration given your training time constraints. Explain your reasoning.

Problem 3 - *Sync and Async SGD* 10 points

Consider a case with P learners where $P = 3$ in distributed training. Let the mini-batch processing times (in milliseconds) of six successive mini-batches at the three learners be given:

$$\begin{aligned} learner-1 &: 1.5, 0.9, 2.5, 1.2, 1.8, 0.9 \\ learner-2 &: 3, 2.5, 1.7, 3.0, 0.7, 0.8 \\ learner-3 &: 2.5, 1.5, 0.7, 0.9, 2.0, 2.2 \end{aligned}$$

Homework 1

Calculate the time to have three updates of the model parameters at the parameter server under following three algorithms:

1. **[2 points]** Sync (fully synchronous)
2. **[2 points]** 2-sync
3. **[2 points]** 2-batch sync
4. **[2 points]** Async
5. **[2 points]** 2-batch async

Problem 4- *Staleness in Async SGD* 5 points

In a Parameter-Server (PS) based Asynchronous SGD training system, there are two learners. Assume a learner sends gradients to the PS, PS updates weights and a learner pulls the weights from the PS in zero amount of time (i.e. after learner sends gradients to the PS, it can receive updated weights from PS immediately). Let us assume that learner 1 runs at about 2.5x speed of learner 2. Learner 1 calculates gradients $g[L_1, 1]$ at second 1, $g[L_1, 2]$ at second 2, $g[L_1, 3]$ at second 3, $g[L_1, 4]$ at second 4. Learner 2 calculates gradients $g[L_2, 1]$ at second 2.5, $g[L_2, 2]$ at second 5. Updates to weights are instant once a gradient is available. Calculate the staleness (number of weight updates between reading and updating weights) of $g[L_1, 1], g[L_1, 2], g[L_1, 3]g[L_1, 4], g[L_2, 1], g[L_2, 2]$. ($g[L_i, j]$ means i-th learner's j-th calculated gradients).

Problem 5 - *Training a simple chatbot using a seq-to-seq model* 25 points

We will train a simple chatbot using movie scripts from the Cornell Movie Dialogs Corpus based on the **PyTorch Chatbot Tutorial**. This tutorial allows you to train recurrent sequence-to-sequence model. You will learn the following concepts:

- Handle loading and pre-processing of **the Cornell Movie-Dialogs Corpus dataset**
- Implement a sequence-to-sequence model with **Luong attention mechanism(s)**
- Jointly train encoder and decoder models using mini-batches
- Implement greedy-search decoding module
- Interact with the trained chatbot

We will use the code in the tutorial as the starting code for the assignment:

1. **[5 points]** Make a copy of the notebook of the tutorial, follow the instructions to train and evaluate the chatbot model in your local Google Colab environment
2. Learn how to use Weights and Biases (W&B) to run a hyperparameter sweep and instrument the notebook to use the Weights and Biases integration to help you run some hyperparameters sweeps in the next steps. Watch the video tutorial provided in the references section.
3. **[5 points]** Create a sweep configuration using the using the **W&B Random Search** strategy for the following hyperparameters:
    - Learning rate: [0.0001, 0.00025, 0.0005, 0.001]

- Optimizer: [adam, sgd]
- Clip: [0, 25, 50, 100]
- teacher_forcing_ratio: [0, 0.5, 1.0]
- decoder_learning_ratio: [1.0, 3.0, 5.0, 10.0]

4. **[5 points]** Run your hyperparameter sweeps using the GPU-enabled Colab and observe the results in the W&B console.
5. **[10 points]** Extract the values of the hyperparameters that give the best results (Minimum loss of the trained model). Explain which hyperparameters affect the model convergence. Use the feature importance of W&B to help guide your analysis.

References:

- The Cornell Movie Dialogs Corpus
- Hyperparameter sweeps with Weights and Biases Framework video tutorial
- Sample Google Colab project that accompanies the video above
- Weights and Biases Website

Problem 6 - *Paper Reading and Analysis* 10 points

Select a research paper from the provided list that aligns with your research interests and academic focus. After a thorough and critical reading of your chosen paper, address the following components:

1. **[3 points] Main Contributions Analysis**
   Provide a concise summary of the key contributions of the article, including:
   - Primary research findings
   - Novel innovations or methodologies
   - Significant conclusions
2. **[3 points] Personal Learning Reflection**
   Identify and explain one significant concept, technique, or insight that:
   - Was previously unknown to you
   - Enhanced your understanding of the field
   - Demonstrates particular value or interest
3. **[4 points] Brainstorming Proposal**
   Develop one of the following:
   - An extension idea of the paper's research
   - An alternative approach to the problem
   - A novel application in a different domain

Submission Instructions: Include your complete response under "Problem 6" in your homework document. Begin with a clear citation of your chosen paper. Keep your answers for each part under 100 words, and focus on being clear and direct. Long, wordy answers will lose points.