CPSC 8430 Deep Learning

Danish Bhatkar (C15563212) – dbhatka@clemson.edu

Homework 1

Question 1.1 – Deep vs Shallow

Simulate a function

1. Describe the models you use, including the number of parameters (at least two models) and the function you use.

   Model 1:-
   Number of parameters – 2241
   Hidden layers – 2
   Activation layer – ReLU

   Model 2:-
   Number of parameters – 2209
   Hidden layers – 3
   Activation layer – ReLU

   Functions used are : sin(x) and x^3

2. In one chart, plot the training loss of all models.

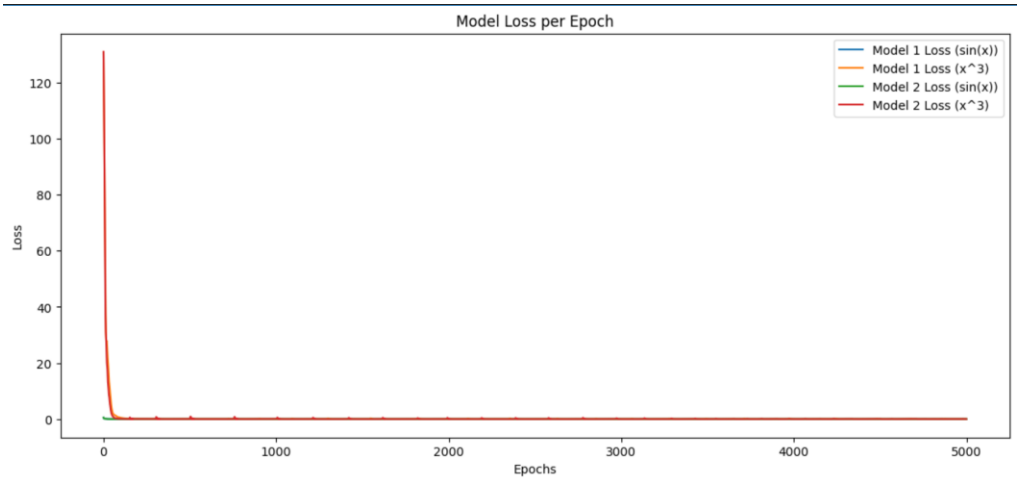


Figure 1 - Training loss of all models

3. In one graph, plot the predicted function curve of all models and the ground-truth function curve.

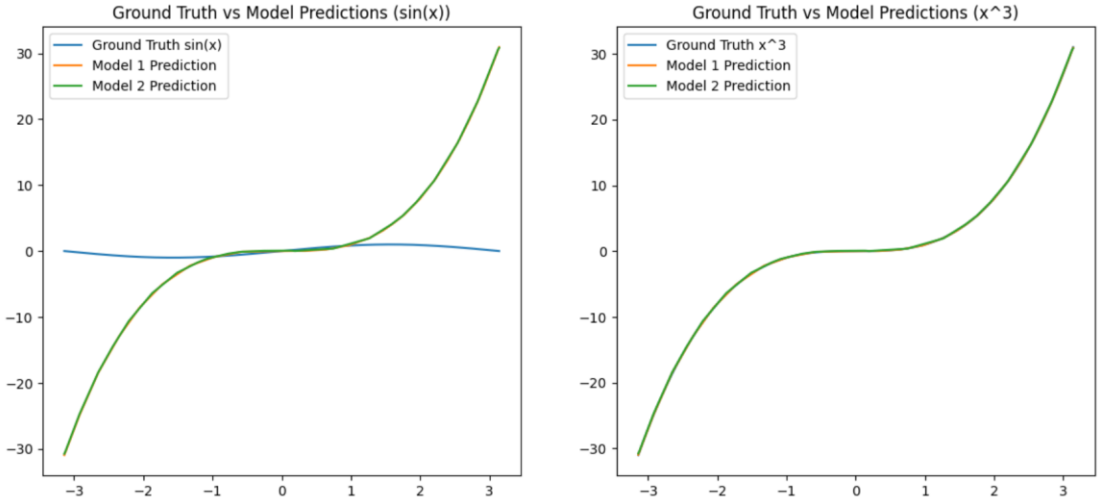


Figure 2 - Ground truth vs Model prediction for both functions

4. Comment on your results.
   Both the models have different number of layers, but the predictions and the losses obtained are indistinguishable to the naked eye.

** Used more than 1 function (bonus)

Train on Actual Tasks

1. Describe the models you use and the task you chose.
   Both the models are CNNs built using PyTorch and are trained for the dataset **CIFAR-10**.
   Model 1:-
   Conv. Layers – 2
   Filter sizes – 12 and 32
   Feature maps size – 8x8
   Number of layers – 5

   Model 2:-
   Conv. Layers – 3
   Filter sizes – 8, 12, 32
   Feature maps size – 4x4
   Number of layers – 6
2. In one chart, plot the training loss of all models.
3. In one chart, plot the training accuracy.

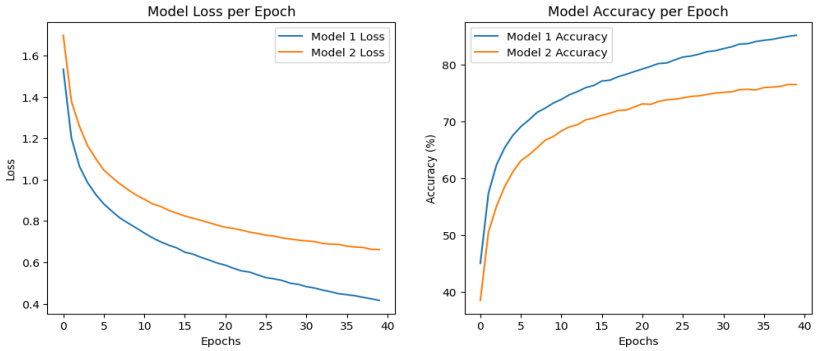


Figure 3 - Training loss and accuracy of both models

4. Comment on your results.
   Although, Model 1 has lesser depth as compared to Model 2, it outperforms Model 2 in terms of accuracy and losses recorded as per the above charts. The differences in filters and feature maps lead to this difference in outputs.

Question 1.2 – Optimization

Visualize the Optimization Process

1. Describe your experiment settings. (The cycle you record the model parameters, optimizer, dimension reduction method, etc)
   The experiment trains a CNN on the CIFAR-10 dataset for 30 epochs using Adam optimizer and CrossEntropyLoss, recording model parameters every 3 epochs. Gradient norms and loss are tracked throughout training, with weights collected and flattened into a matrix.
2. Train the model for 8 times, selecting the parameters of any one layer and whole model and plot them on the figures separately.

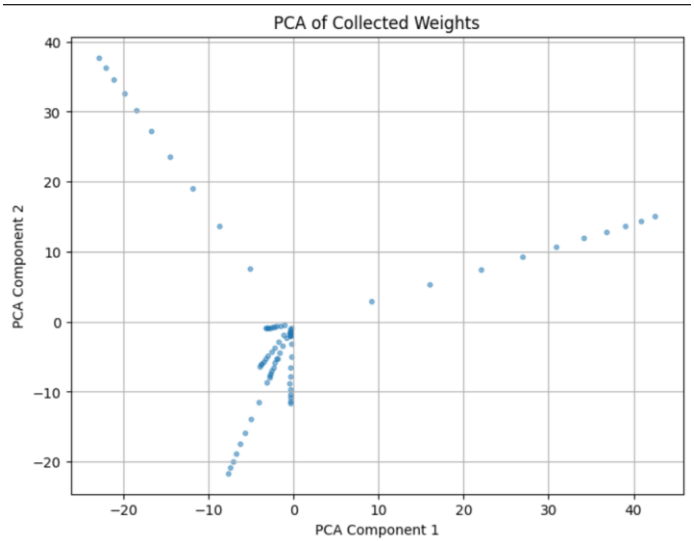


Figure 4 - Convergence of weights in 2D space

3. Comment on your results.
   The 2D scatter plot from PCA shows the evolution of the model's parameters across epochs and runs, revealing distinct trajectories that reflect the variability in training dynamics but convergence toward a similar region in weight space. This suggests consistent learning patterns across different initializations.

Observe gradient norm during training

1. Plot one figure which contain gradient norm to iterations and the loss to iterations.

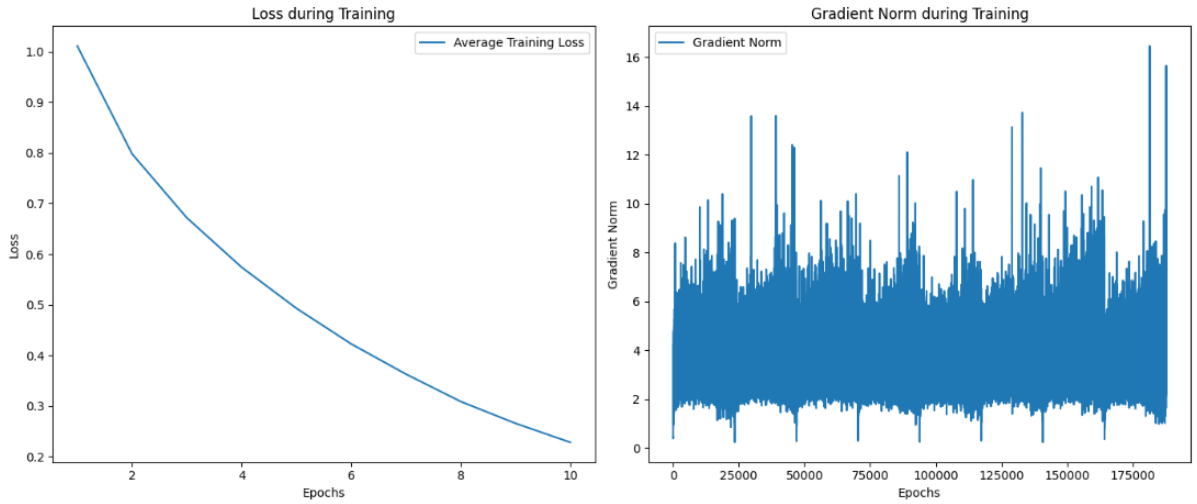


Figure 5 - Loss and Gradient norm

2. Comment on your result.
   The training loss decreases steadily, showing effective learning across all runs, but may suggest room for further optimization if the loss plateaued early.

What happens when gradient is almost zero?

1. State how you get the weight which gradient norm is zero and how you define the minimal ratio.
   To find weights where the gradient norm is zero, we track the L2 norm of the gradients during each training step. If the total gradient norm across all parameters falls below a small threshold, the training loop can terminate early since minimal updates are being made to the model.
   The minimal ratio is defined as the proportion of weights in the model that are close to zero (decided by considering a threshold value).
2. Train the model for 100 times. Plot the figure of minimal ratio to the loss.

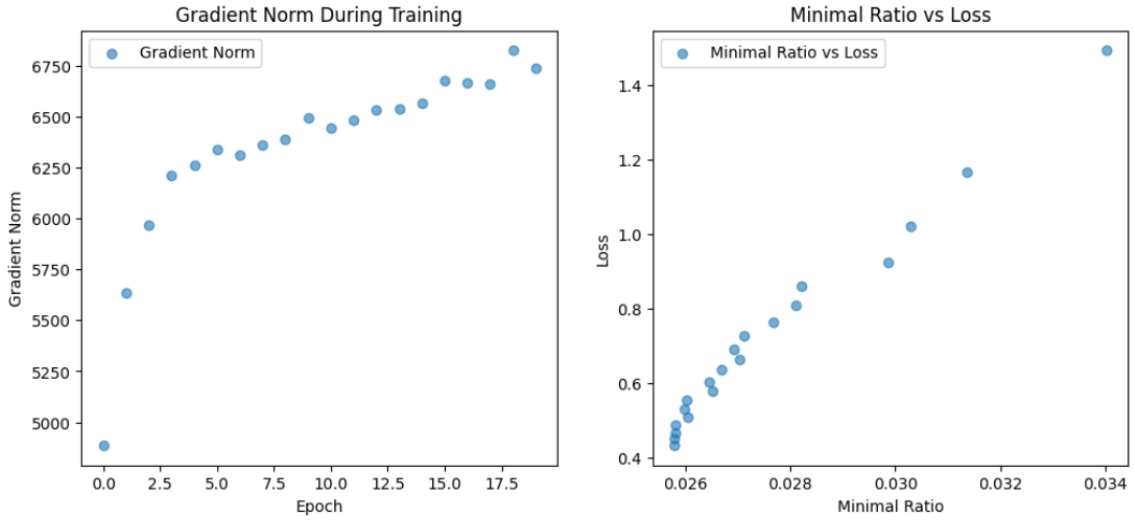


Figure 6 - Gradient norm trajectory and Minimal Ratio to Loss

3. Comment on your results.
   Since the gradient norm increases with training, it suggests that the learning rate might be high or that the models are not converging.
   Also, as the minimal ratio increases with increase in loss, it suggests that the model is losing its learning capacity as more weights shrink towards zero. It could mean over-regularization or underfitting.

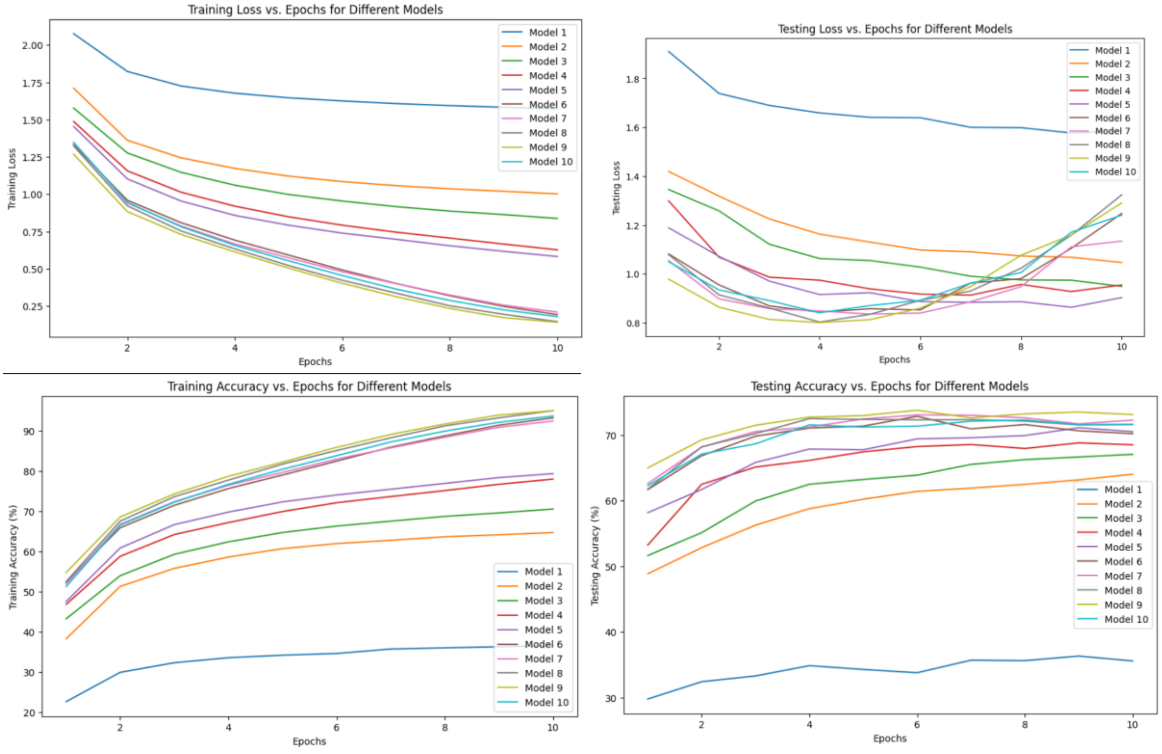
Question 1.3 – Generalization

Can a network fit random labels?

1. Describe your settings of the experiments. (e.g. which task, learning rate, optimizer)/
   Task – Classification of CIFAR-10 dataset, 10 classes of images
   Model – CNN with 2 conv. Layers and 2 fully connected layers
   Optimizer – Adam
   Learning rate – 0.001
   Loss function – Cross-entropy loss

Number of parameters vs Generalization

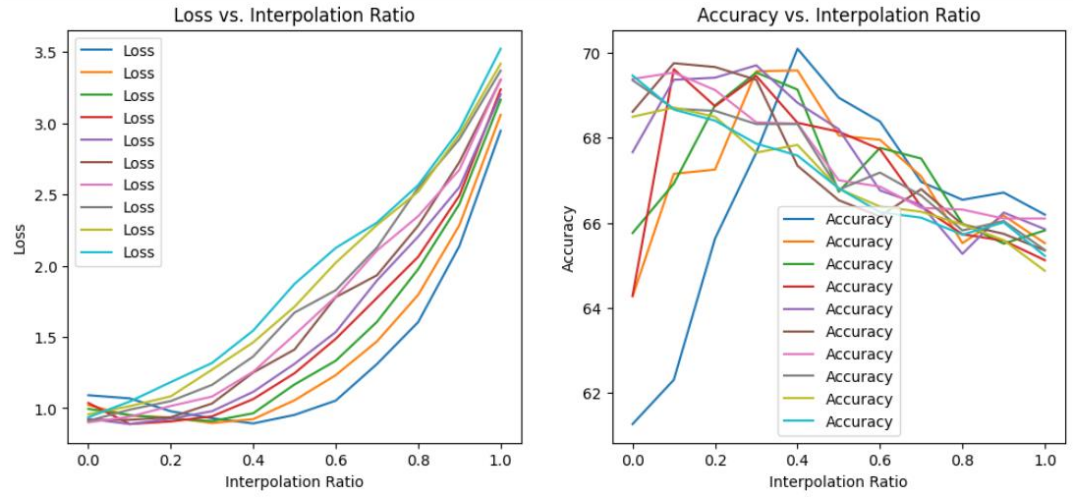
1. Describe your settings of the experiments. (e.g. which task, the 10 or more structures you choose).
   Task – Image classification on CIFAR-10 dataset
   Dataset – CIFAR-10, consisting of 60,000 32x32 color images in 10 classes.
   Model structures – 10 models as follows:-
   a. Model 1: 8 filters in both conv layers, 8 hidden units
   b. Model 2: 8 filters (conv1), 16 filters (conv2), 16 hidden units
   c. Model 3: 16 filters in both conv layers, 32 hidden units
   d. Model 4: 16 filters (conv1), 32 filters (conv2), 64 hidden units
   e. Model 5: 32 filters in both conv layers, 64 hidden units
   f. Model 6: 32 filters (conv1), 64 filters (conv2), 128 hidden units
   g. Model 7: 64 filters in both conv layers, 128 hidden units
   h. Model 8: 64 filters (conv1), 96 filters (conv2), 128 hidden units
   i. Model 9: 96 filters (conv1), 128 filters (conv2), 128 hidden units
   j. Model 10: 128 filters in both conv layers, 256 hidden units
   Optimizer – Adam
   Loss function – Cross entropy loss
2. Plot the figures of both training and testing, loss and accuracy to the number of parameters.

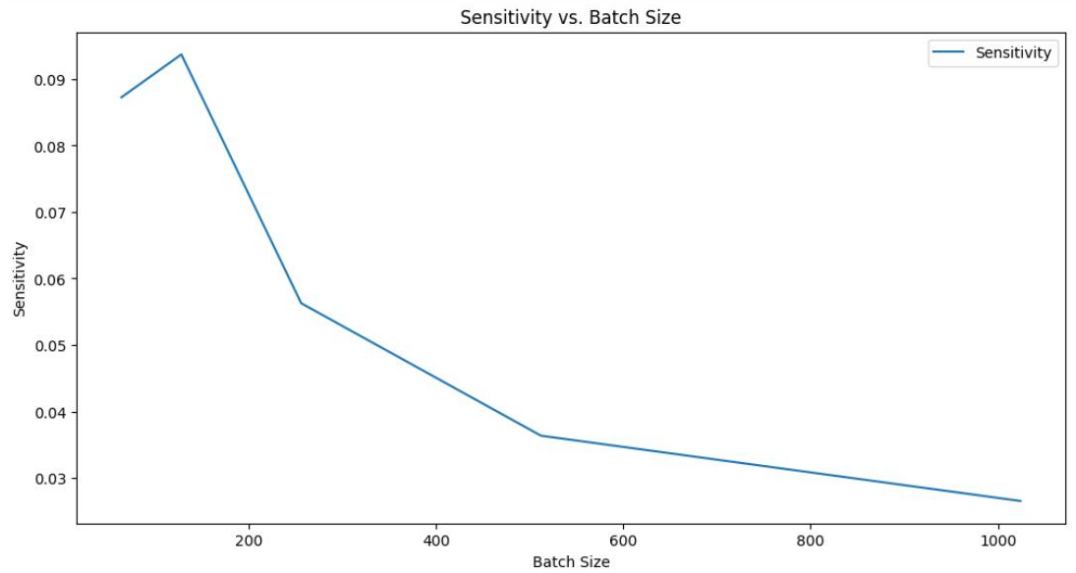


3. Comment on your results.
   The Model #9 outperforms all other trained models in terms of losses and accuracies observed even though it is not the most dense neural network in the list of models.

Flatness vs Generalization

1. Describe the settings of the experiments (e.g. which task, what training approaches).
   Task – Classification of CIFAR-10 dataset using CNN
   Dataset – CIFAR-10
   Model – 2 Conv. Layers (ReLU activation + Max pooling) + 2 fully connected layers
   Training approach – Batch sizes: 64, 128, 256, 512, 1024
   Optimizer – Adam
   Learning rate – 0.001
   Loss function – Cross entropy loss
2. Plot the figures of both training and testing, loss and accuracy, sensitivity to your chosen variable.





3. Comment on your result
   Loss and accuracy curves across interpolation ratios can show model stability. Smooth curves suggest compatibility between parameters, while sharp changes indicate potential sensitivity.
   The sensitivity decreases with increasing batch size.