CS6886W - System Engineering for Deep Learning

Assignment 1

Department of Computer Science and Engineering
IIT Madras, Chennai, Tamil Nadu

Due Date: 12, October 11:59, 2025

Instructions:

This assignment focuses on **exploring VGG6 on CIFAR-10** with different configurations and analyzing model performance based on these configurations.

The final best validation accuracy will be considered for evaluation.

Code context: You may start from the provided template code that demonstrates the training and evaluation of the VGG6 model with one configuration:

Template code for VGG6 training and evaluation

You must adapt this to run with **different configurations** and report performance along with the plots requested below.

Allowed libraries: You may use any library of your choice. It is recommended to use PyTorch or TensorFlow. The code must be written in Python, and you are expected to write your own code (no direct copy-paste from external sources).

What to submit: A GitHub repository link. The repository should include a `README.md` file describing exactly how to re-run your code to reproduce your reported results, along with your explanation of how performance varies across different configurations.

You also need to submit a single PDF that contain the detailed answers along with the plots. Also make sure you paste your GitHub link inside this PDF (If we can't access your code, your marks will be deducted).

Question 1. Training Baseline (10 points)

(a) Prepare CIFAR-10 with proper normalization and data augmentation. Specify the transforms used. (5)

(b) Train the model with one chosen configuration (preferably the one that gives the best test accuracy). (3)

(c) Report the final test top-1 accuracy and include loss/accuracy curves. (2)

Question 2. Model Performance on Different Configurations (60 points)

(a) Vary the activation function. Use different activations such as ReLU, Sigmoid, Tanh, SiLU, GELU, etc. Describe how model performance changes when the activation function is varied. (20)

(b) Vary the optimizer. Use different optimizers such as SGD, Nesterov-SGD, Adam, Adagrad, RMSprop, Nadam, etc. Explain how each optimizer affects convergence and how they differ from one another. (30)

(c) Vary the batch size, number of epochs, and learning rate. Explain how the convergence speed and performance vary with these changes. (10)

Question 3. Plots (10 points)

(a) Provide the W&B parallel-coordinate plot that shows which configuration achieves what accuracy (a sample plot is given below).

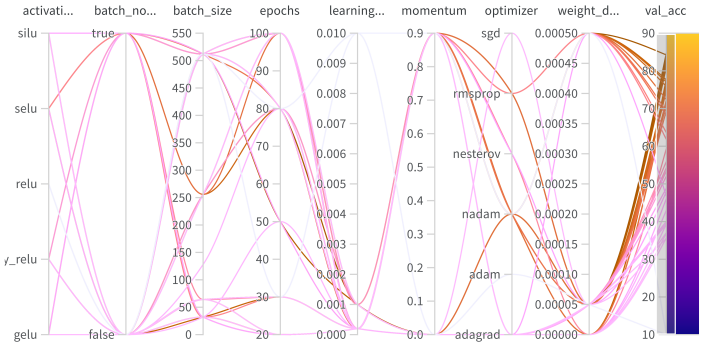

Figure 1: Sample parallel coordinates plot.

(b) Provide the validation accuracy vs. step (scatter) plot.

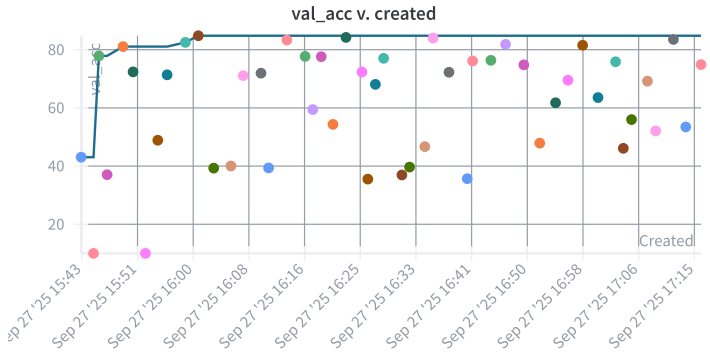

Figure 2: Validation accuracy vs. step plot.

(c) Provide the plots for training loss, training accuracy, validation loss, and validation accuracy respectively. (Sample plots are shown below. Make sure you generate these automatically using W&B.)

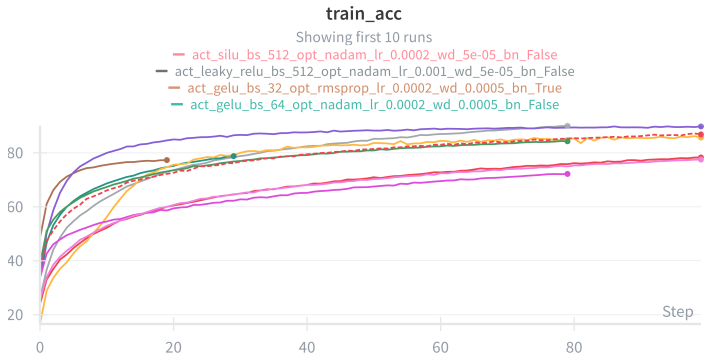

Figure 3: Training accuracy plot.

Figure 4: Training loss plot.

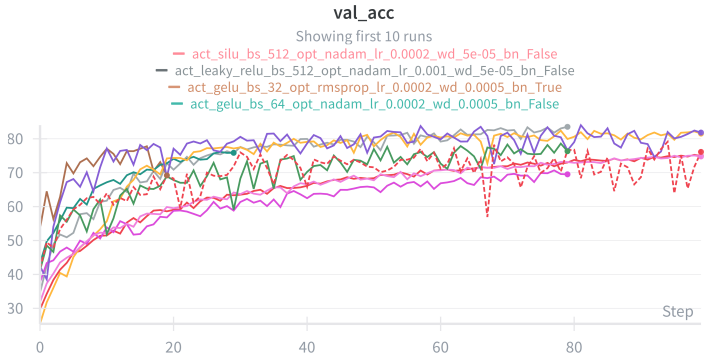

Figure 5: Validation accuracy plot.

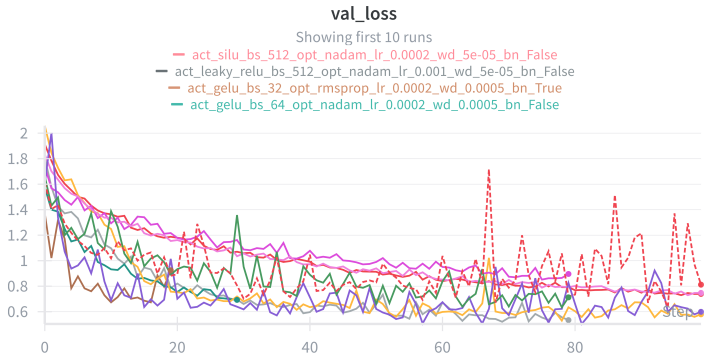

Figure 6: Validation loss plot.

Note: The plots shown here are just examples. You will obtain different values depending on your configurations. Please explore how to generate such plots automatically using W&B sweeps.

Question 4. Final Model Performance (10 points)

Based on the W&B parallel plot, provide the configuration that achieved the best validation accuracy. Do not list multiple configurations. The configuration mentioned here will be verified by re-running your model. Ensure that this configuration indeed reproduces the reported best accuracy.

Question 5. Reproducibility and Repository (10 points)

(a) Provide clean, modular, and well-commented code with clear separation of training, evaluation, and compression modules. (4)

(b) Include a README with exact commands, environment details, and dependency versions. Also include seed configuration. (4)

(c) Upload the trained model to the GitHub repository and provide the GitHub repository link inside the pdf (2)