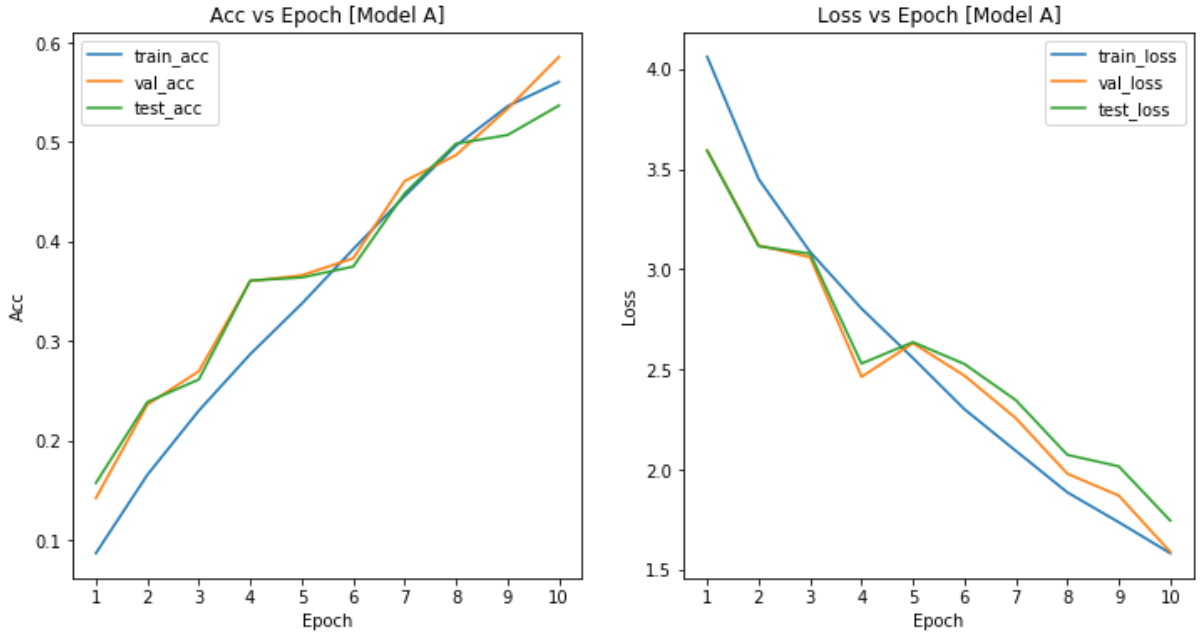
HW4 Finetuning in Neural Networks

Overview:

1. You may view the HW4-finetuning-neural-network jupyter notebook's main() function for the numbers of the accuracy/loss per epoch of the 3 models
2. The jupyter notebook also prints the exact layers (parameters) that are not frozen and trainable.

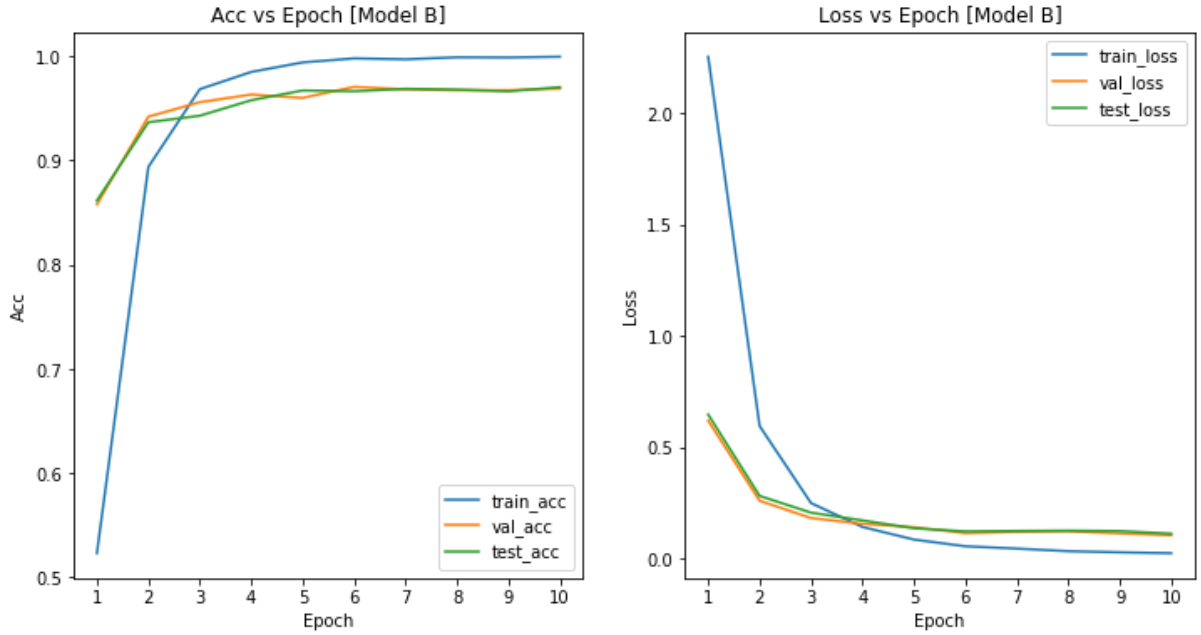
Model A results: once without loading weights and training all layers



For Model A, in general, the test accuracy is similar or slightly less than the validation accuracy. This would mean that the validation and test sets are similar to each other such that they produce similar results. However, since the model is chosen based on the validation accuracy, it would be bias to the validation set. This would explain the slightly lower test accuracy as compared to validation accuracy.

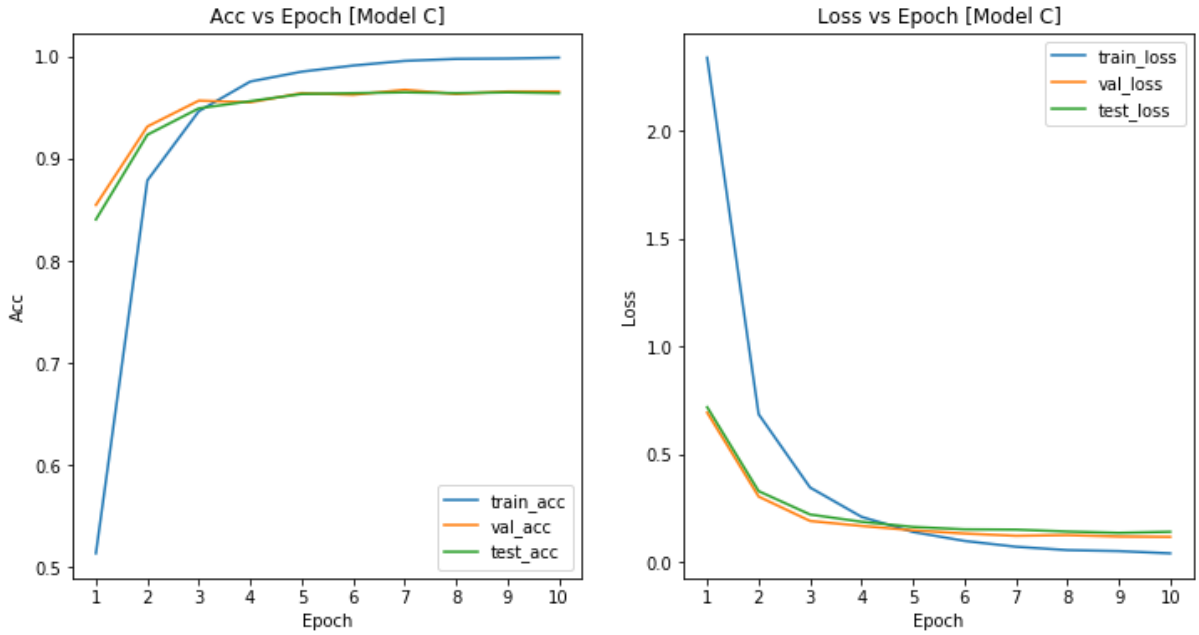
In addition, Model A's accuracy for both the validation and test set increases for every epoch.

Model B results: once with loading model weights before training and training all layers



For Model B, similar to Model A, the test accuracy is similar or slightly less than the validation accuracy due to the same explanation.
However, we notice that the accuracy for Model B is much higher than Model A on both the validation and test set. This is because Model B has loaded pretrained weights of the model (Resnet 18) such that the training serves to finetune the model and not training it from scratch unlike Model A. Hence, the accuracy of Model B is much higher than Model A and also stagnates after epoch 6.

Model C results: once with loading model weights before training and training only the last two trainable layers (note: for quite some problems, the approach B is better than C):



For Model C, similar to Model A, the test accuracy is similar or slightly less than the validation accuracy due to the same explanation.

However, we notice that the accuracy for Model C is much higher than Model A on both the validation and test set. This is similar to Model B which also loaded pretrained weights of the model (Resnet 18) such that the training serves to finetune the model and not training it from scratch unlike Model A. Hence, the accuracy of Model C is much higher than Model A and is similar to Model B. However, training Model C which took about 17 minutes was faster than training Model B which took about 23 minutes due to having lesser trainable parameters because only the last two layers were frozen. In this scenario, using Model C seems like a better option than Model B due to producing similar results in lesser time.