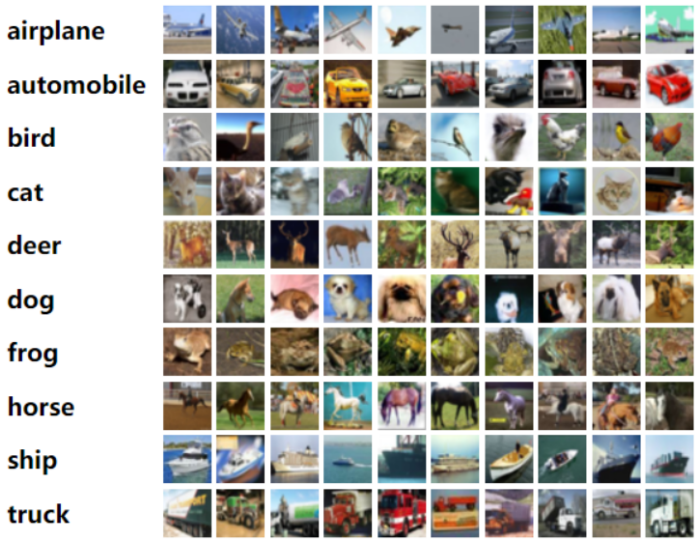
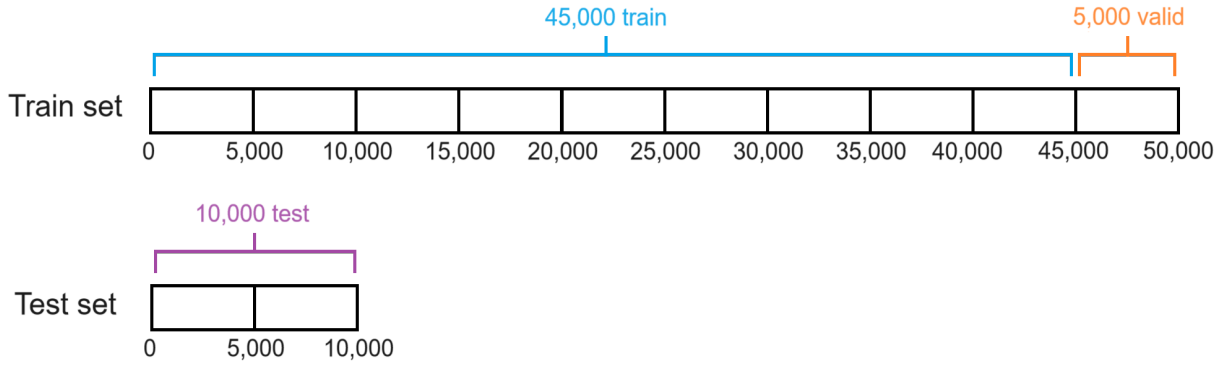
Project 1 CIFAR-10 Classification with Skip Connections
CMSC 598 / 689 Computer Vision with Deep Learning
Fall 2023
Due 10 / 02 / 2023

You must create a convolutional neural network (CNN) to classify CIFAR-10 images into their respective categories. Your neural network architecture must be a ResNet3 like architecture as specified in the description. You must use a custom training loop with minibatch size 100.

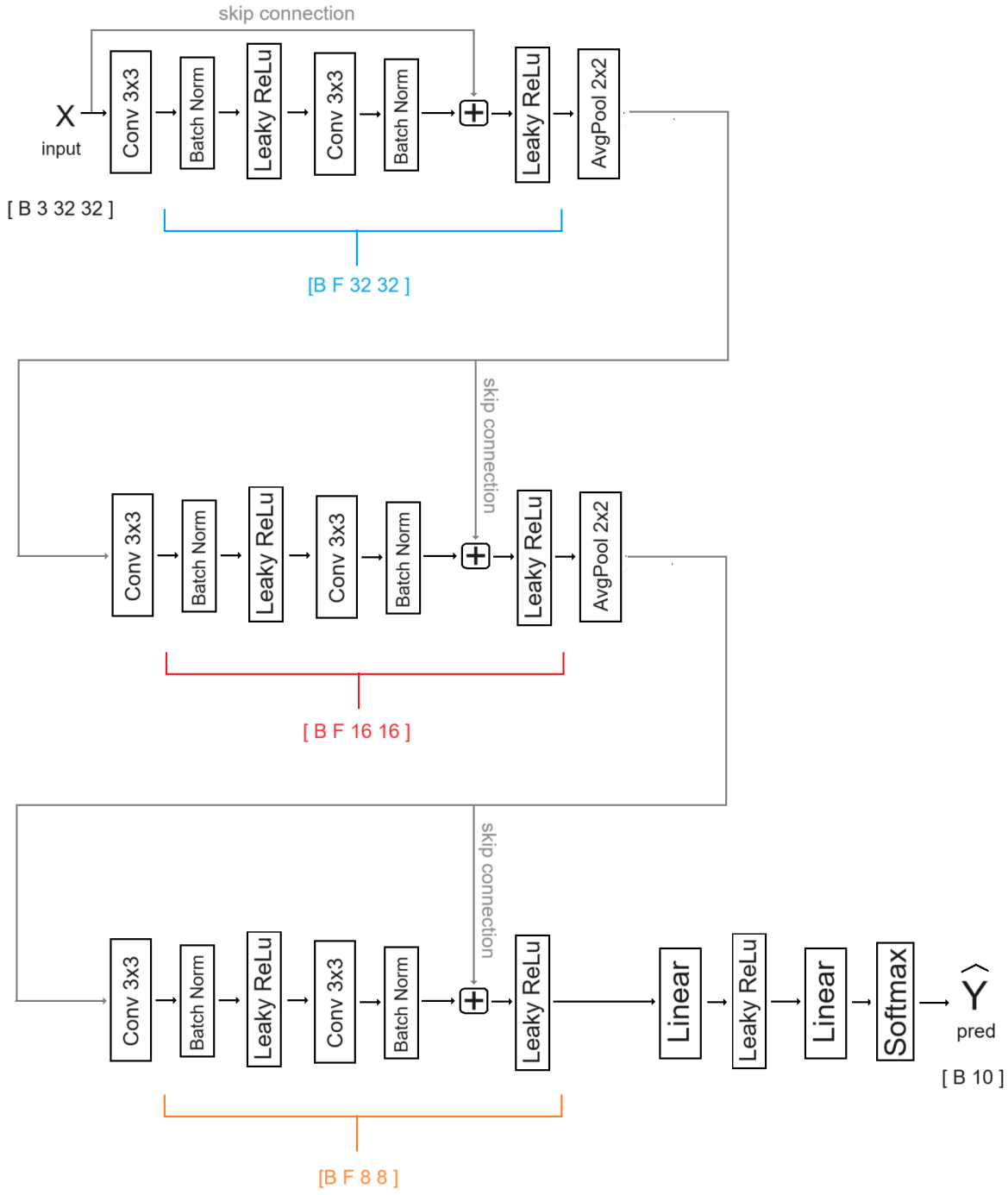


A ResNet architecture has skip connections between each pair of layers. We must implement a specific ResNet-3 like architecture. You must create this architecture from scratch using the PyTorch functional model. The details of this architecture are shown in the below diagram.

You must separate the training data into train, validation and test splits, the CIFAR-10 data has 50000 training images, and 10000 test images. You must split the first 45000 “training” images as “train” images, and you must extract the next 5000 “training” images as “validation” images. Leave the separate 10000 images as “test” images.



Your program must create a directory called “output” and you must save the first 5 characters of each test, test, and validation split as follows. Your output must nearly-identically match the following,



You must create a custom training loop, and train the model

You cannot use Data Loader

(extra credit) you must calculate accuracy and loss plots by hand.

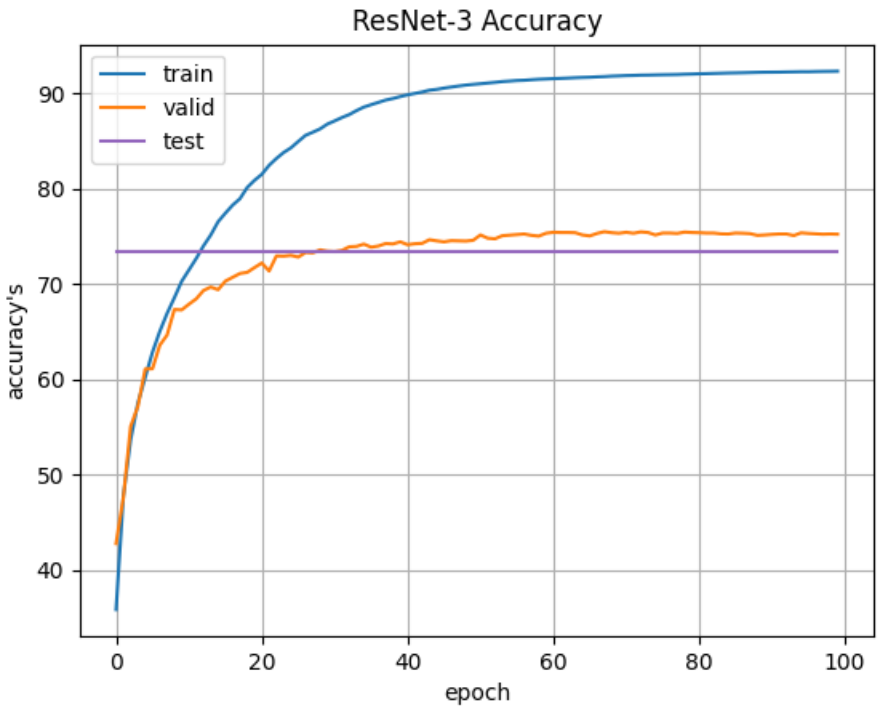
must be of the following form

```
0:      for epoch in range(num_epochs):
1:          for batch in range(num_batch):
2               # perform gradient update
```

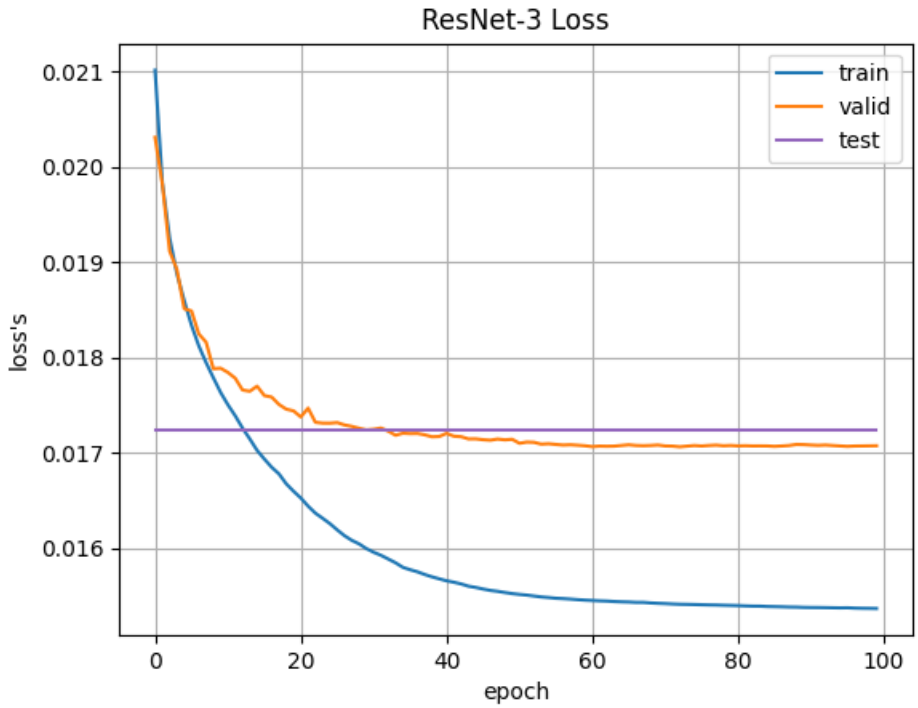
You must use cross-entropy loss

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\cdot\log(\hat{y}_{i,c})$$

You must plot your your training, validation, and test accuracy and save them to the folder “output”



output/accuracy.png



output/loss.png

Requirements **Deductions**

Submission must be a zip file over Blackboard with the following contents

proj1.zip:
- proj1.py
- outdir
- outdir/accuracy.png
- outdir/loss.png
- outdir/test_00000.png
- outdir/test_00001.png
- . . .
- outdir/test_00049.png

Code must be written in Python using Pytorch -100 pts

Code must be a single python file named proj1.py -100 pts

Code must use only the following dependencies -100 pts
- torch
- torchvision
- matplotlib
- cv2
- <<any python standard library such as os, sys, time etc.>>

Code must not use DataLoader	-100 pts
Python file must run straight through to completion without user input	-50 pts
Matplotlib must use the 'Agg' backend, and not display to the screen (all figures must be directly saved to output folder) matplotlib.use('Agg')	-10 pts
Architecture must be ResNet-3 nearly identical as described in above architecture diagram	-75 pts
First 50 test images must be written to output directory	-30 pts
Test image titles must display the following information test <imgno> label <true_cat> pred <pred_cat> accuracy <percent> Note: true_cat and pred_cat must be the 'names' of the categories	-20 pts
Accuracy (with train validation and test) and Loss (with train and validation) must be saved to accuracy.png and loss.png	-30 pts
Training loop must implement the cross-entropy loss	-30 pts
Accuracy and loss plots must be implemented by hand using matplotlib	-15 pts
All loss plots must divide by number of images in the epoch (such that losses are comparable on plot)	-10 pts
Accuracies must be displayed in percentiles (out of 100)	-10 pts
Script must attempt to create folder "output", and must not crash if output folder already exists, or if output folder does not exist.	-30 pts
Architecture must learn and achieve at least 65% test accuracy	-50 pts

Extra Credit

Students enrolled in 689 must implement extra credit, and will receive 5 pt deduction otherwise

Implement cross-entropy loss by hand using basic operations	EX 5 pts