Problem Statement

In this project, we are expected to accurately classify the images with the noisy version of "CIFAR-10" dataset. To achieve the balance between the complexity of the model and the prediction performance, we have to design a sophisticated model with high classification efficiency as well as an approach to address the label noise issue with high generalization ability.

Data Augmentation

We developed five types of data augmentation: flip, rotate, distort, lighting, color on 10,000 clean data. The generated 50,000 augmented data and 10,000 clean data are the dataset the model II bases on. And the total 100,000 images with 50,000 original data and 50,000 augmented data are the total processed data which were stored in npy files.

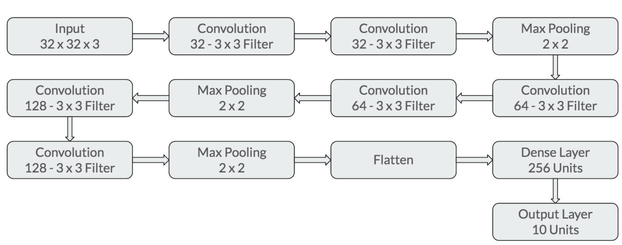
Baseline Model: Logistic Regression

In this model, we use simple multinomial logistic regression to classify the images. Image data is transformed into RGB histograms. This model is trained on 10 thousand clean labeled and 40 thousand noisy labeled data. The test accuracy of this model is around 24% which is only a little better than random label assignment.

Model I: CNN Implementation

The entire dataset is 100K. In general, training is 75K, validation is 25K (random split).

In Model one, we use a convolution neural network for image classification. Here we are using six convolutional layers with batch optimization. To reduce the dimensions of the feature maps, we are using Max Pooling. In the last layer, we are using Softmax as a squashing function to interpret out as a probability. To help with the overall loss and improve the accuracy, we used the Nadam optimizer. This model is trained on 10 thousand clean labeled and 40 thousand noisy labeled data. The validation accuracy of this model is 60.51%



Model details:

- 6 Convolution Layers
- 3 Max Pooling
- Total parameters: 816,938

Training scheme:

- 75000 images
- 256 batch size
- 20 epochs

CPU (type: Apple M1 8 cores) training time for model 1: ~34 min (1993.2634029388428 s)
GPU (type: Tesla K80) training time for model I: ~9 min (507.0870819091797 s)
CPU(Intel(R) Xeon(R) CPU @ 2.20GHz) training time for model I: ~ 225 min (13527.60789)

Model II: CNN + Semi-Supervised Learning

In Model two, we used the definition of model one to train the model with only clean labels(60k) and then predict labels for noisy data(40K). We retrained the model one again with newly cleaned image labels. This process significantly improved our accuracy. The validation accuracy of model two is 82.66%.

Training scheme (clean_label_model):

- 60000 clean images
- 256 batch size
- 20 epochs

CPU (type: Apple M1 8 cores) training time for clean model: ~26 min (1551.1493549346924 s)
CPU (type: Apple M1 8 cores) training time for label model: ~10 min (621.3688530921936 s)
CPU(Intel(R) Xeon(R) CPU @ 2.20GHz) training time for model I with clean label: ~ 167 min (9987.13235 s)
GPU (type: Tesla K80) training time for clean_model: ~4 min (204.50233840942383 s)

Training scheme (model II):

- 75000 images
- 256 batch size
- 20 epochs

CPU (type: Apple M1 8 cores) training time for model II: ~34 min (2063.428079843521 s)
CPU(Intel(R) Xeon(R) CPU @ 2.20GHz) training time for model II: ~ 225 min (13527.60789)
GPU (type: Tesla K80) training time for model II: ~9 min (510.91714239120483 s)

Conclusion

Model	Validation Accuracy	Total Runtime (i9, Apple M1, Tesla K80)	Predict 25K Runtime (i9, Apple M1, Tesla K80)
Baseline	20%	~10 s	~2 s
Model I	60.51%	(~3.75 hr, ~35 min, ~9 min)	(~15 min, ~3 min, ~3s)
Model II	82.66%	(~8 hr, ~1 hr, ~13 min)	(~15 min, ~3 min, ~3s)

We created two predictive models for image classification. For Model I, we designed a Convolutional Neural Network (CNN) model based on VGG-Net, with 60% validation accuracy which improved significantly compared with the baseline logistic regression model (20% accuracy). For Model II, we designed a weakly supervised learning model. In the first part, we used Model I to train clean and augmented data, mapping the noisy labels to clean labels. In the second part, we used combined data for predicting labels to train our CNN model in Model I, which yielded 83% test accuracy.