

Machine Learning Final Model Competition

Model Report (Test accuracy: 0.8291)

Shanglin Yang
Kaggle Name: Yang

1 Model Illustration

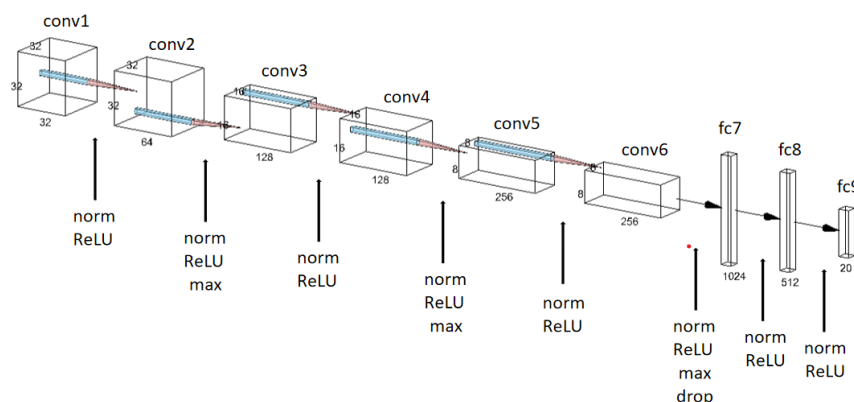


Figure 1: The 9-layer convolution neural network for the classification task

2 Training Methods

2.1 Data Preprocessing

The preprocessing steps involve a series of transformations, primarily using PyTorch's **transforms** module, with a combination of augmentation and normalization techniques.

For training images. I used:

- Random Rotation: 20 degrees
- Random Horizontal Flip
- Padding: 4 pixels, Reflect mode
- Random Crop: 32x32 pixels
- Color Jitter: Brightness 0.2, Contrast 0.2, Saturation 0.2, Hue 0.2
- Conversion to Tensor
- Normalization: Using specific channel mean and standard deviation

For test images. I used:

- Conversion to Tensor
- Normalization: Using specific test mean and standard deviation

2.2 Network Settings

After multiple adjustments on settings, the following is the best setting found so far:

- Loss function: Cross entropy loss
- Optimizer: Adam
- Training batch size: 128 (allows shuffle)
- Learning rate: The learning rate is manually adjusted according to the change in train loss and validation loss. Generally speaking, I begin the training with a learning rate = 0.001 and quickly adjust to 0.0005 after 8 epochs. Each time I adjust the learning rate, I will save the model state for future re-training. In around 120 epochs, I finally adjusted the learning rate to 0.00001 and got the best model weights.

2.3 Resource usage

I used the T4 GPU provided by Google Colab and it took around 2 hours to train 120 epochs.