CIFAR-100 Model Report

William Clavier

Created on March 30 | Last edited on March 30

AI usage is mainly documented throughout the code. I mainly used it when things wouldn't be working as expected and I was seeking help. As mentioned in the code, I originally had errors beyond the model, and I didn't realize that. I thought it was the model so I was desperate for some help, and after using the AI-inspired model for part 1 and it still not scoring well, I realized that the issue was elsewhere. After further inspection and testing, I found that the error was in my normalizations and loss calculations. Besides that, the majority of my code was gathered through research or the occasional AI recommendation, which I should have noted where necessary.

Part 1

Description

For the first part, I implemented a simple CNN architecture with three convolutional blocks. Each block consists of a convolutional layer followed by batch normalization, ReLU activation, and max pooling. I finished the network with three fully connected layers to perform the classification.

Hyperparameter Tuning

I used Adam optimizer with a learning rate of 0.001 and a simple step learning rate scheduler that decays by 0.1 every 2 epochs. The batch size I set to 512 to maximize throughput.

Regularization Techniques

I implemented dropout between layers (0.25 for convolutional layers, 0.5 for fully connected layers) and batch normalization after each convolutional layer to prevent overfitting.

Data Augmentation Strategy

Used basic normalization only with mean and std of (0.5, 0.5, 0.5). No additional augmentations were implemented in this initial version. I just wanted to get the pipeline all set up for the later models.

Results Analysis

This simple model provided a baseline but had limited performance due to its relatively shallow architecture and basic augmentation strategy. It was nice to make sure that the rest of the code was working properly before dealing with the larger models. As previously explained the majority of the learnings from working on this model was not the model itself but rather the rest of the pipeline.

Part 2

Description

Built upon Part 1 by implementing a non-pretrained ResNet50 architecture, modified for CIFAR-100's smaller image size. The major changes included using a 3x3 initial convolution and removing the maxpool layer to preserve spatial information. The removal of the maxpool layer was a recommendation by AI when I asked if there were any other recommendations. The techniques used for this model were also similar to my best model, so see Part 3 for a more in depth explanation on the design choices.

Hyperparameter Tuning

I switched to AdamW optimizer with a learning rate of 0.001 and weight decay of 0.01. I used a cosine annealing learning rate scheduler over 30 epochs. I reduced the batch size to 128 due to the large model size.

Regularization Techniques

I utilized ResNet's built-in batch normalization and residual connections, plus AdamW's weight decay for regularization.

Data Augmentation Strategy

I used random cropping with padding and horizontal flips. I changed the basic normalization to CIFAR-100 normalization using dataset statistics found online.

Results Analysis

This model showed significant improvements over the previous simple CNN. This showed the benefits of a well-designed and deeper network. This network also was provided with more appropriate data augmentation for the challenge and, therefore was better prepared for the testing data.

Part 3

Description

The model that achieved the highest score for me is a pre-trained ResNet50 Model that is using the ImageNet v2 weights. The only modifications are to make it match the input and output for the CIFAR-100 dataset. This is the same as the Part 2 model I made. That is crucial because it is trained and made for 224x224 images when the dataset we have is 32x32. I also changed the first convolutional to use a 3x3 kernel because of the massive difference in size between the images.

The removal of the maxpool was recommended by AI when I asked it if there was anything else I should change.

Hyperparameter Tuning

I used the AdamW optimizer with a learning rate of 0.001 and a weight decay of 0.01. The cosine annealing learning rate scheduler helped by gradually reducing the learning rate over the course of the epochs. This helped the model converge to good solutions and was almost a form of regularization. The batch size was 128 because I wanted it to be relatively low because of the size of the model.

Regularization Techniques

I chose to use AdamW in order to have some weight decay because AdamW handles a changing learning rate better than normal Adam. The model benefits from the effects of batch normalization and residual connections that are in the ResNet architecture. The cosine annealing learning rate scheduler also helped by getting the model towards the global minimum instead of falling into local minimums.

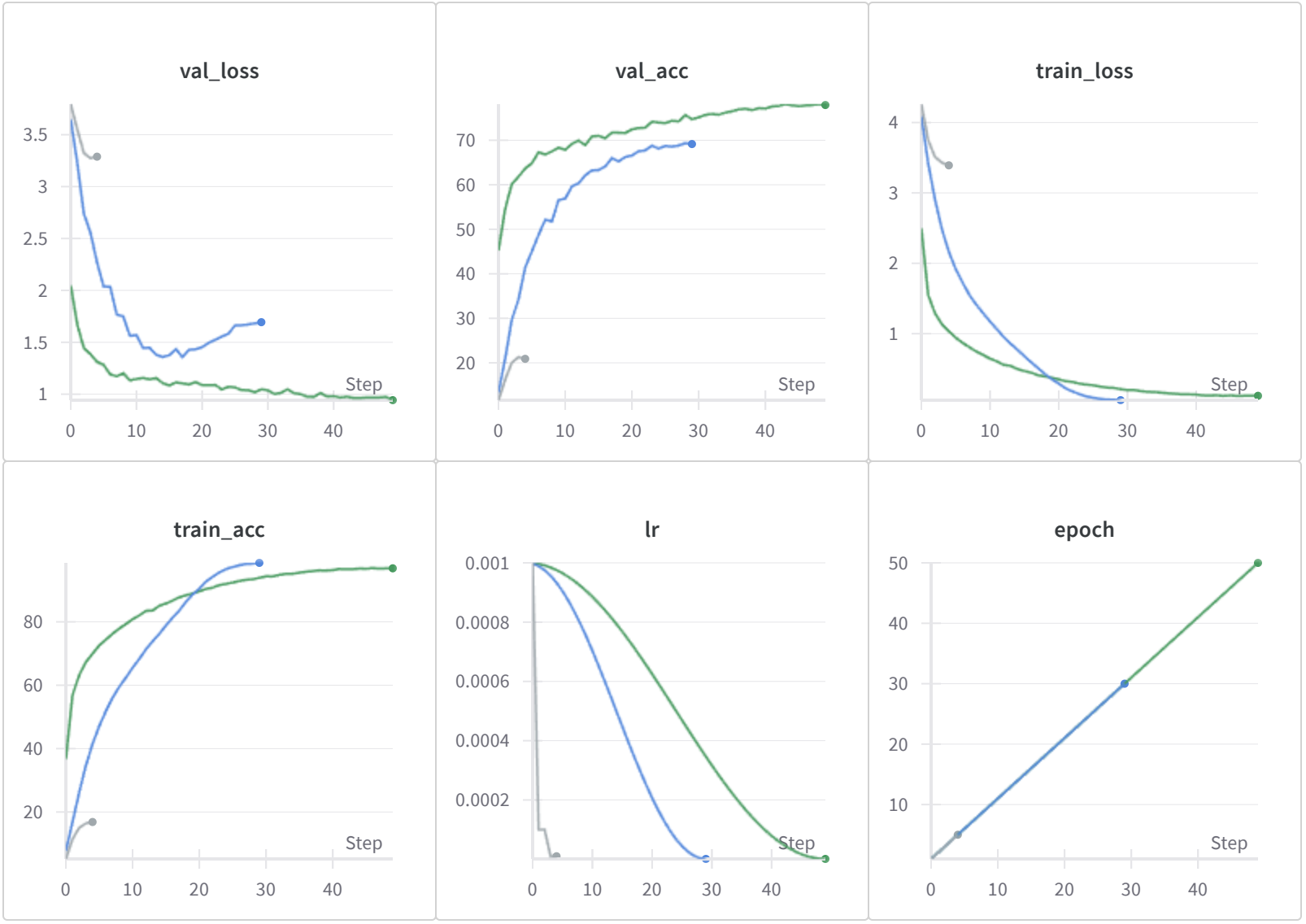
Data Augmentation Strategy

I used a couple of methods for this. I used random cropping with padding and horizontal flips as the base augmentations. I wanted to be able to cover more bases by using these, and these were commonly described when I did research. I think the main advantage came when I implemented AutoAugment with the CIFAR10 policies. This definitely helped a lot because it handled some of the more complex operations that the other augmentations didn't cover. I also chose to normalize my images using the actual CIFAR-100 statistics that I found online. The combination of these augmentations helped my model be prepared to handle the variance in images while still retaining the features for classification.

Results Analysis

I didn't have too much time to be able to experiment with other models out there. I figured that ResNet50 was a good starting point, and it blew my expectations out of the water when I submitted it for scoring. This model did well because it carried its knowledge base over to the slightly different CIFAR-100 dataset that has its own challenges. The regularization and augmentations that I made helped prevent overfitting despite being a deep network.

Tracking Summaries



Created with ❤️ on Weights & Biases.

https://wandb.ai/wclavier-boston-university/sp25-ds542-challenge/reports/CIFAR-100-Model-Report--VmlldzoxMjA1NTAzMw