V1V1_Final_REPORT

Learning Vision Intelligence: Main Report For Classification

1. **Team Name / Date**
    a. V1V1 / 2024/10/31

---

2. **Leader / Members** (with Contribution Statement)
    a. kim dahyun(33%)/ park junwoo(33%), kim geonho(33%)

---

3. **Introduction**

    To select our final base model, we conducted an initial test across candidate models, each trained for 100 epochs under the same data augmentation conditions. Models with significantly low accuracy, like ViT and EfficientNet B2, were removed from the candidate pool.

    In a second test that considered Top-5 and SuperClass accuracy, all models achieved satisfactory performance with over 70% Top-1 accuracy. However, Wide ResNet demonstrated a notable advantage, outperforming the other models by over 10 points in total score. Based on these performance results, our team selected Wide ResNet as our final base model.

    From there, we experimented with various elements based on this model, including Data Augmentation, Optimizer, Learning Rate, DropOut, Learning Scheduler, Loss Function, Widen Factor, Validation Data, Activation Function, and Ensemble methods.

---

4. **Related works**
    a. Data Augmentation
        i. RandomHorizontalFlip
        ii. RandomCrop
        iii. CutMix
    b. Optimizer
        i. SGD_with_momentum

c. Learning Rate
    i. 0.1
d. DropOut
    i. 0.2
e. Learning Scheduler
    i. if(epoch > 80):
       optim_factor = 5
       elif(epoch > 60):
       optim_factor = 2
       elif(epoch > 30):
       optim_factor = 1
f. Loss Function
    i. CrossEntropyLoss
g. Widen Factor
    i. 14 ,16, 18
h. ~~Doubled Data~~
    i. Original + Augmented Data(RandomHorizontalFlip,RandomCrop,CutMix)
i. Validation Data
    i. divide data not **train/val/test** but **train/test**
j. Activation Function
    i. gelu
    ii. relu
    iii. leakyrelu
k. Ensemble
    i. Hard Voting (Top1, SuperClass)
    ii. Soft Voting (Top-5)

---

5. **Method**
    a. Data_Augument and Optimization

During training, we applied SGD with momentum as our primary optimizer. Each model was set up with SGD optimizers initialized with the following parameters: learning rate 0.1, momentum of 0.9, and weight decay of 5e-4. After initializing and setting the models to training mode, we applied CutMix data augmentation and trained each model individually.

- SGD with Momentum: The optimizer updates the weights based on both the current gradient and a fraction of the previous update. This approach helps the model to converge faster and avoid getting stuck in local minima.
- CutMix Data Augmentation: CutMix was applied by randomly replacing regions of an input image with parts of another image, effectively mixing two labels. The model then calculates the weighted loss based on the proportion of each label in the mixed image. This technique aids in improving generalization and robustness by diversifying the training data.

b. Ensemble

For Top-1 accuracy, we utilized hard voting because it involves selecting a single class as the final prediction. Hard voting was advantageous here as it helped improve the certainty and reliability of the top prediction by focusing on the most commonly agreed-upon class across models.

In contrast, for Top-5 accuracy, we applied soft voting. This method allowed each model to contribute to the probability distribution for the top 5 classes, providing a more flexible and robust approach. Soft voting benefited us here by allowing models to consider a broader range of class probabilities, thereby enhancing the likelihood of the correct class being included within the top 5 selections.

c. Fianl_setting

We conducted the final training using Wide-ResNet with widen factors of 14, 16, and 18, optimized with SGD with momentum. For the activation function, we selected Leaky ReLU based on its demonstrated performance boost. The final training was executed on the CIFAR-100 dataset for 95 epochs.

6. **Experimental Results**
    a. optimizer comparison
        i. sgd_with_momentum

            Final General Top-1 Accuracy: 78.27%
            Final General Top-5 Accuracy: 94.17%
            Final Superclass Top-1 Accuracy: 86.49%

        ii. adam

            Final Genral Top-1 Accuracy: 54.31%

            Final Genral Top-5 Accuracy: 83.71%

            Final Superclass Top-1 Accuracy: 67.49%

        iii. adamw

            Final General Top-1 Accuracy: 69.79%
            Final General Top-5 Accuracy: 90.13%
            Final Superclass Top-1 Accuracy: 79.67%

        iv. nadam

            Final General Top-1 Accuracy: 55.27%
            Final General Top-5 Accuracy: 84.98%
            Final Superclass Top-1 Accuracy: 69.13%

    b. learning_rate scheduler
        i. before change

            Final General Top-1 Accuracy: 79.56%
            Final General Top-5 Accuracy: 94.90%
            Final Superclass Top-1 Accuracy: 87.38%

        ii. after change

            Final General Top-1 Accuracy: 80.30%
            Final General Top-5 Accuracy: 95.01%
            Final Superclass Top-1 Accuracy: 87.87%

    c. Ensemble
        a. <Widen_Factor 14, 16, 18>
            - 42 : RTX3090 GPU SERVER

Top-1 Accuracy = 84.83%
Top-5 Accuracy = 97.20%
Superclass Accuracy = 91.43%

Total_score: 273.46

- 27： A100 Colab

  Top-1 Accuracy = 84.65%
  Top-5 Accuracy = 97.05%
  Superclass Accuracy = 91.60%

  Total_score: 273.3

- 14：A100 Colab

  Top-1 Accuracy = 84.69%
  Top-5 Accuracy = 96.96%
  Superclass Accuracy = 91.45%

  Total_score: 273.1

b. <Widen_Factor 14, 14, 14>

   Top-1 Accuracy: 84.37%

   Top-5 Accuracy: 97.01%

   Superclass Accuracy: 91.33%

c. <Widen_Factor 14,16,16>

   Top-1 Accuracy: 84.44%

   Top-5 Accuracy: 97.34%

   Superclass Accuracy: 91.51%

d. Data

   a. Data x1
      Top-1 Accuracy: 83.52%
      Top-5 Accuracy: 96.06%
      Superclass Accuracy: 90.77%

   b. Data x2
      Top-1 Accuracy: 83.59%
      Top-5 Accuracy: 96.27%
      Superclass Accuracy: 90.24%

c. Base
   Top-1 Accuracy: 80.30%
   Top-5 Accuracy:95.01%
   Superclass Accuracy: 87.87%

d. Base +CutMix
   Final General Top-1 Accuracy: 82.26%
   Final General Top-5 Accuracy: 96.00%
   Final Superclass Top-1 Accuracy: 89.87%

e. Loss,Label_smoothing

   a. CrossEntropyLoss

      Top-1 Accuracy: 78.02%
      Top-5 Accuracy: 95.17%
      Superclass Accuracy: 86.66%

   b. Label_smoothing

      Top-1 Accuracy: 68.64%
      Top-5 Accuracy: 90.45%
      Superclass Accuracy: 79.07%

   c. Focal_Loss

      Top-1 Accuracy: 76.87%
      Top-5 Accuracy: 95.02%
      Superclass Accuracy: 87.48%

f. Epoch

   a. 100 Epoch
      Top-1 Accuracy: 78.02%
      Top-5 Accuracy: 95.17%
      Superclass Accuracy: 86.66%

   b. 200 Epoch
      Top-1 Accuracy = 77.94%
      Top-5 Accuracy = 94.45%
      Top-1 Accuracy = 85.62%

g. Batch_Size

   a. 128 Batch Size
      Top-1 Accuracy: 78.02%

Top-5 Accuracy: 95.17%
Superclass Accuracy: 86.66%

b. 64 Batch Size
Top-1 Accuracy = 78.05%
Top-5 Accuracy = 94.76%
Top-1 Accuracy = 86.62%

h. train,val

a. train
Top-1 Accuracy: 83.52%
Top-5 Accuracy: 96.06%
Superclass Accuracy: 90.77%

b. train+validation
Top-1 Accuracy: 81.32%
Top-5 Accuracy: 95.37%
Superclass Accuracy: 88.91%

i. Drop_out

a. dropout- 0

Top-1 Accuracy: 77.70%
Top-5 Accuracy: 94.18%
Superclass Accuracy: 86.12%

b. dropout- 0.2

Top-1 Accuracy: 83.52%
Top-5 Accuracy: 96.06%
Superclass Accuracy: 90.77%

c. dropout- 0.3
Top-1 Accuracy: 78.02%
Top-5 Accuracy: 95.17%
Superclass Accuracy: 86.66%

d. dropout 0.4
Top-1 Accuracy: 75.15%
Top-5 Accuracy: 93.29%
Superclass Accuracy: 84.31%

j. Learning_rate

a. Data x1
   Top-1 Accuracy: 78.02%
   Top-5 Accuracy: 95.17%
   Superclass Accuracy: 86.66%

b. Data x2
   Top-1 Accuracy: 77.46%
   Top-5 Accuracy: 94.88%
   Top-1 Accuracy: 85.96%

c. Learning Rate 0.1
   Top-1 Accuracy: 78.02%
   Top-5 Accuracy: 95.17%
   Superclass Accuracy: 86.66%

d. Learning Rate 0.05
   Top-1 Accuracy = 78.09%
   Top-5 Accuracy = 94.20%
   Superclass Accuracy = 86.55%

k. Activation_Function(widen_factor:4)

a. ReLU
   Top-1 Accuracy = 79.12
   Top-5 Accuracy = 95.25%
   Superclass Accuracy = 87.66%

b. Leaky ReLU(widen factor 4)
   Top-1 Accuracy = 79.62%
   Top-5 Accuracy = 95.19%
   Superclass Accuracy = 87.94%

c. GELU(widen factor 4)
   Top-1 Accuracy = 79.82%
   Top-5 Accuracy = 94.79%
   Superclass Accuracy = 87.47%

- **Report Best Scores (with 3 different random seeds)**
  - 42： RTX3090 GPU SERVER
    - Top-1 Accuracy = 84.83%
      Top-5 Accuracy = 97.09%
      Superclass Accuracy = 91.43%

        - Total_score: 273.35
    - 27 : RTX3090 GPU SERVER
        - Top-1 Accuracy = 85.08%
          Top-5 Accuracy = 97.10%
          Superclass Accuracy = 91.41%
        - Total_score: 273.59
    - 14 : A100 Colab
        - Top-1 Accuracy = 84.69%
          Top-5 Accuracy = 96.96%
          Superclass Accuracy = 91.45%
        - Total_score: 273.1

---

7. **Implementation Details** (include project schedule you did, self-evaluation, used tool/library, technical documents such as a manual for random users)
    - project schedule

#2	**0911~0917**	**•Write the proposal. • Understand the project requirements. • Analyze the data. • Research the models and techniques to use.**
#3	**0918~0924**	**•Compare the training and testing results for each model and technique.**
#4	**0925~1001**	**•Select the model and techniques to use. • Design the neural network structure.**
#5	**1002~1008**	**•Training and testing the model. • Normalize, improve, debug the model.**
#6	**1009~1015**	**•Use some regularization or data augmentation. • Write the technical documentation. change activation function, dropout, learning rate scheduler**

#7	**1016~1022**	**•Prepare the final presentation. ensemble wideresnet 14, 16, 18 test our model with differenet seeds** **•** **Complete the final report.**
#8	**1022~1031**	**•Give a final presentation.** **•** **Submit the final report.**

- self-evaluation

  In the early stages of the project, we set an initial goal of achieving a total score of 270. Although we have met this target, we aimed to reach a Top-1 accuracy of 85 during the course of the project. Unfortunately, we were unable to achieve this goal.

  But we significantly enhance the model’s performance from 259 to 273 by ensemble, adjusting the activation function and other techniques. We tried to squeeze out model’s performance until the last moment.

  Little bit disappointed that we couldn't achieve an accuracy above 85% in terms of Top-1.

  With the current level of performance improvement, if we had explored a better architecture like PyramidNet and used a regularization technique suitable for the ResNet model structure, such as ShakeDrop, we believe we could have achieved even better performance, potentially surpassing the top team’s score.

  We believe that by surpassing at least one team that utilized PyramidNet, which is known for its top performance in this challenge, we have effectively improved the existing base model, Wide_ResNet, and achieved strong results. So we feel proud of this accomplishment.

- used tool/library
  - Python 3.10

    - torchvison 0.19.1+cu118
    - torch 2.4.1+cu118

- technical documents
    - Transforming and augmenting images — Torchvision 0.20 documentation
    - torch.nn — PyTorch 2.5 documentation

8. **Conclusion**

   Based on these results, we ensemble models with widen factors of 14, 16, and 18 using both soft voting and hard voting, achieving the following:

   - **Top-1 Accuracy**: 85.08%
   - **Top-5 Accuracy**: 97.10%
   - **Superclass Accuracy**: 91.43%
   - **Total_score: 273.59**

9. **Reference**

   a. bmsookim/wide-resnet.pytorch: Best CIFAR-10, CIFAR-100 results with wide-residual networks using PyTorch
   b. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features