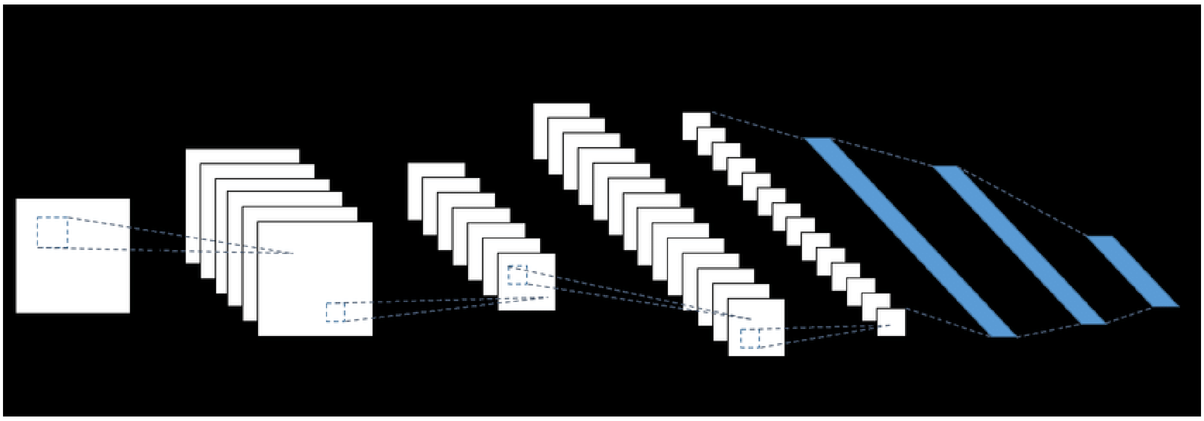


Convolutional Neural Network

HW4

109511276 練鈞揚 |Machine Learning | 5/16/2023

Discuss About the CNN

For this HW4, we are required to use a CNN model for classification. Through this HW, I have learned how to use PyTorch to create a CNN model. However, the most challenging aspect of the problem was choosing the appropriate model for the task. Eventually, I discovered EfficientNetV2-M on the CIFAR-10 Benchmark (Image Classification) | Papers With Code. This model was chosen due to its high test accuracy of approximately 90%, which is superior to ResNet18 and ResNet50.

What is ResNet ?

ResNet (short for "Residual Network") is a deep learning architecture for image classification. It is designed to help overcome the problem of "vanishing gradients", which can occur in very deep neural networks and make them difficult to train effectively. ResNet uses a unique approach called "skip connections" to allow the network to learn more effectively and achieve better accuracy on difficult image classification tasks. The ResNet architecture has become a popular choice for many image recognition tasks and has achieved state-of-the-art results on various benchmark datasets.

What is EfficientNetV2?

EfficientNetV2 is a convolutional neural network architecture that was developed as an improvement over the original EfficientNet. It uses a novel scaling method to increase the depth, width, and resolution of the network, while keeping the number of parameters and computational cost under control. This results in a more efficient and accurate model for image classification tasks. EfficientNetV2 also incorporates other advanced techniques such as squeeze-and-excitation blocks, mobile inverted bottleneck blocks, and Swish activation functions, which further enhance its performance.

Model detail discuss:

When working on this homework, I came to realize that the choice of optimizer and number of epochs are crucial as they greatly influence the model's accuracy. I opted to use Adam as the optimizer for EfficientNetV2-v2-M and resize the training data to 224. In addition, I set the number of epochs to 10 to prevent overfitting.

Optimizer :

1. Stochastic Gradient Descent (SGD): This is a popular optimizer that updates the parameters of the model using the gradient of the loss function with respect to the parameters. It can be used with or without momentum.

2. Adam: This is another popular optimizer that computes adaptive learning rates for each parameter, based on the estimate of the second moment of the gradients. It is well-suited for sparse gradients and noisy data.

3. Adagrad: This optimizer adapts the learning rate for each parameter based on the historical gradients. It works well for sparse gradients and is often used in natural language processing tasks.

4. Adadelta: This optimizer is similar to Adagrad but uses a rolling window of gradient updates to compute the learning rate for each parameter. It is more stable than Adagrad and can handle larger learning rates.

5. RMSprop: This optimizer uses an exponential moving average of the squared gradients to compute the learning rate for each parameter. It is well-suited for recurrent neural networks and can handle non-stationary environments.

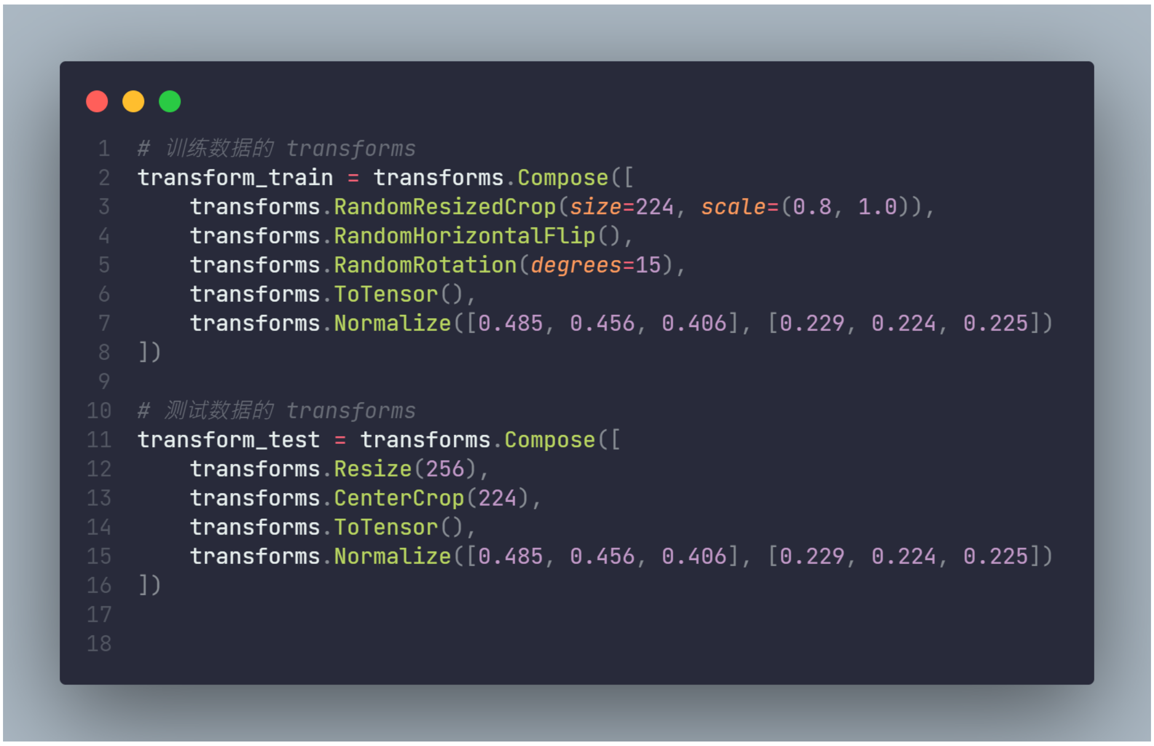
Result:

Model:	Train Time:	Accuracy:
ResNet18	Around 30 mins	78.620
ResNet50	Around 45 mins	77.625
EfficientNetV2-v2-M	Around 49 mins	91.468
EfficientNetV2-v2-L	Around 90 mins	91.786

Best Model :

EfficientNetV2-v2-M, less time , high accuracy

Training dataset setting:



```python
# 训练数据的 transforms
transform_train = transforms.Compose([
    transforms.RandomResizedCrop(size=224, scale=(0.8, 1.0)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(degrees=15),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])

# 测试数据的 transforms
transform_test = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])


```

Refence:

- CIFAR-10 Benchmark (Image Classification) | Papers With Code
- What is Resnet or Residual Network | How Resnet Helps? (mygreatlearning.com)
- [2104.00298] EfficientNetV2: Smaller Models and Faster Training (arxiv.org)
- torch.optim — PyTorch 2.0 documentation
- (160) Image Classification using CNN from Scratch in Pytorch- Part 1 Training - YouTube
- Training a Classifier — PyTorch Tutorials 2.0.1+cu117 documentation
- 通过 Pytorch 实现 ResNet18 - 知乎 (zhihu.com)
- Pytorch 实战 2：ResNet-18 实现 Cifar-10 图像分类（测试集分类准确率 95.170%）_cifar10 准确率 pytorch_杭州豆角焖面的博客-CSDN 博客
- aiml_tutotials/santa_pytorch_pretrained_model.ipynb at master · shounak8/aiml_tutotials · GitHub