THUAI

Sun Yi 2023310664, Fang Jun 2023310770

June 2024

1 Abstract

In this course project, we undertook the CIFAR-10 image classification task, employing various machine learning and deep learning techniques. Specifically, we implemented and evaluated the performance of Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and attention-based networks for which we choose ViT as our model. Our results demonstrate the comparative advantages and limitations of each approach in terms of accuracy and their ability to handle complex image patterns inherent in the CIFAR-10 dataset. Through this project , we gained valuable insights into the practical applications and efficacy of these classification methodologies.

2 Introduction

Image classification is a fundamental task in the field of computer vision, where the goal is to categorize images into predefined classes. With the advent of deep learning, significant progress has been made in improving the accuracy and efficiency of image classification algorithms.

In this project, we aim to explore and compare the performance of three different approaches to image classification: Support Vector Machine (SVM), Convolutional Neural Network (CNN), and Attention Network(ViT as an example). SVM is a traditional machine learning algorithm known for its effectiveness in various classification tasks. On the other hand, CNNs have revolutionized image classification by enabling automatic feature extraction through multiple layers of convolution and pooling operations. Moreover, Attention Networks have further advanced the state-of-the-art by allowing models to focus on important parts of the image, thereby improving classification accuracy.

Through this project, we aim to provide insights into the practical applications of these models in image classification tasks, emphasizing the advancements brought by deep learning and attention mechanisms.

3 Data Preprocessing

Before model training and feature extraction, data preprocessing, such as normalization and data enhancement, are generally required. These preprocessing steps help to improve the generalization ability and training efficiency of the model. We extend the training set by normalizing the CIFAR-10 dataset and applying data enhancement techniques to improve the robustness and accuracy of the model. Since the subsequent models all try different data preprocessing and enhancement methods for methods' comparison and selection, we start with data preprocessing in this session.

In regard to data preprocessing, we have selected these preprocessing methods as below:

1. transforms.RandomCrop: Randomized Cropping of Images
2. transforms.RandomHorizontalFlip:Random Horizontal Flip Image
3. transforms.Normalize:Image Normalization

Moreover, in addition to the basic data preprocessing methods above, we have learned and practiced some data enhancement by passing in args to decide whether to use as followed.

1. RandomCropPaste:Randomly cut and paste images.Chosen by args.rcpaste
2. mixup:Generate new training samples are generated by linearly interpolating a batch of images and labels.Chosen by args.mixup
3. cutmix: Generate new training samples by replacing a portion of one image with the corresponding portion of another image while adjusting the labels.Chosen by args.cutmix
4. autoaugment: select two specific operations and their corresponding probabilities and magnitudes to form a selection pool via SubPolicy, and then call one of the policies randomly to enhance the dataset CIFAR-10. Chosen by args.autoaugment

To be specifically, operations,which is shown below with corresponding explanations ,can be chosen from autoaugment.

1. ShearX: Shear transformation along the X-axis
2. ShearY:Shear transformation along the Y-axis
3. TranslateX: Translation along the X-axis
4. TranslateY: Translation along the Y-axis
5. Rotate:Rotate at a certain angle
6. Color:Adjustment of colors

7. Posterize:Reduces the color depth of the image
8. Solarize: Invert the portion of the image whose pixel value is greater than the threshold value
9. Contrast: Adjusting Contrast
10. Sharpness: Adjusting Sharpness
11. Brightness: Adjusting Brightness
12. AutoContrast: Automatic Contrast Adjustment
13. Equalize: Histogram equalization
14. Invert: Color Reversal

In the following experiments, we will use regular data preprocessing, while deciding whether or not a specific preprocessing method is used by passing in the args parameter.

4 CNN

As a deep learning structure, Convolutional Neural Networks(CNN) perform better on data with lattice-like topology for its basic structure and some other characteristics. In abstract, Convolutional Neural Networks have several basic structures as below, followed with their roles:

1. Convolutional Layer: The convolutional kernel slides over the image and extracts the local features of the image by local receptive field computation.
2. Pooling Layer: The pooling operation reduces the data dimension and computation while keeping the features unchanged.
3. Normalization Layer: Normalize the data in each mini-batch to improve training stability.
4. Fully Connected Layer: fixes the output dimension in the classification task.

With these basic structures, convolutional neural networks have better advantages in image problems:

1. The local receptive field captures the local features of the image;
2. Parameter sharing in the convolutional layer reduces the risk of overfitting and improves generalization;
3. Matrix multiplication operation can handle high-dimensional data and synthesize the information of each channel;

4. Appropriately increasing the number of network layers can make the network can learn to obtain more complex feature information and make better classification results.

However, in practice, with the increase of the number of layers, the training error no longer decreases and a degradation problem occurs.In response to this problem, the Resnet network structure is proposed, and its core innovation lies in the design of residual blocks through skip connections to realize the formula $y = F(x, \{\Omega\} + x)$ in logic blocks to alleviate the problem of gradient vanishing and degradation, allowing researchers to deepen the number of network layers by stacking residual blocks to learn more complex feature information and achieve better task results. Meanwhile, the design in Resnet maintains the complexity of network layers by doubling the number of featuremaps when the feature map size is halved.

Back to the Cifar10 classification task in the project, CNN as a classical and very effective method, two attempts are made here. One is to perform basic CNN modules to realize the simplest and commonest CNN structure, and the second one is to mimic the Resnet structure to reproduce it and achieve better classification results on the dataset.

The basic CNN network architecture used in this task is shown as Figure 1.

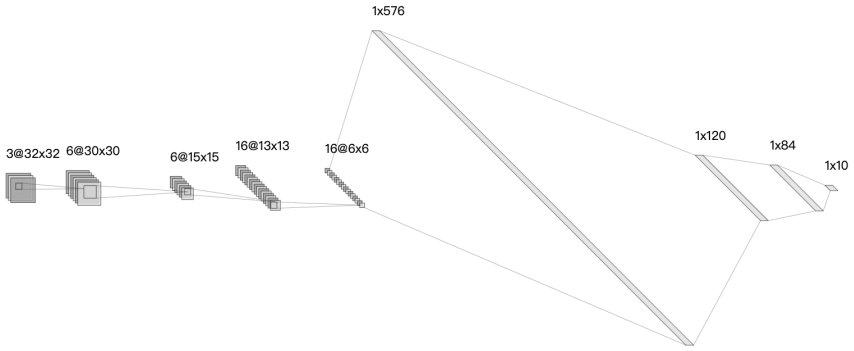

Figure 1: basic CNN structure

To sepcific, after each layer of convolutional operation, batchnormalize, relu activation and dropout are performed, and finally output after flatten. The task was practiced through the underlying CNN network architecture, the parameters were fine-tuned while using different data preprocessing and data enhancement methods, the experimental setup parameters and corresponding results are shown in the table 1 as below.

The loss image and accuracy image are shown below in pic 2 using label-smoothing.

The training parameters of CNN are set to:Epoch=200,Batch Size=128, Optimizer=Adam,Weight Decay=5e-5,LR Scheduler=Cosine, Init LR= 1e-3, Last LR=1e-5, Warmup=5 epochs,Dropout=0.1,Label Smoothing=0.1.

exp name	accuracy	val accuracy
initial	0.708	0.735
label-smoothing	0.739	0.737

Table 1: basic CNN performance

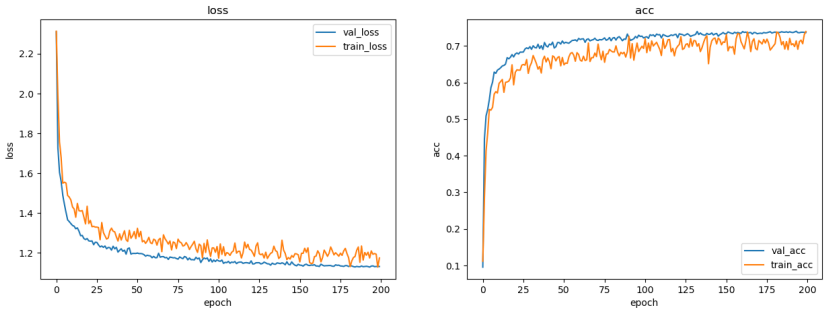

Figure 2: basic cnn loss and acc(ls)

A similar experimental design was also conducted for the Resnet architecture. We reproduce the resnet34 network architecture as follows in table 2.

layer name	output size	34-layer
conv1	32*32	3*3, 64, stride 1
conv2_x	32*32	$\begin{pmatrix} 3*3, 64 \\ 3*3, 64 \end{pmatrix} * 3$
conv3_x	16*16	$\begin{pmatrix} 3*3, 128 \\ 3*3, 128 \end{pmatrix} * 4$
conv4_x	8*8	$\begin{pmatrix} 3*3, 256 \\ 3*3, 256 \end{pmatrix} * 6$
conv5_x	4*4	$\begin{pmatrix} 3*3, 512 \\ 3*3, 512 \end{pmatrix} * 3$
avgpool	1*1	

Table 2: Resnet34 structure

The network architecture is realized by stacking residual blocks. Compared with the basic CNN architecture, the network depth is higher, and theoretically more high-dimensional features can be learned to achieve better results.

Specifically, the parameters are specifically chosen while using different data preprocessing and data enhancement methods, and the experimental setup choices and corresponding results are shown in the table 3.

When using data preprocessing as label-smoothing and auto augment, the loss image and accuracy image are shown below in pic 3.

The training parameters of CNN are set to:Epoch=200,Batch Size=128, Op-

exp name	accuracy	val accuracy
initial	1.0	0.943
label-smoothing	1.0	0.942
auto augment	0.975	0.958
ls+aa	0.972	0.959

Table 3: Resnet performance

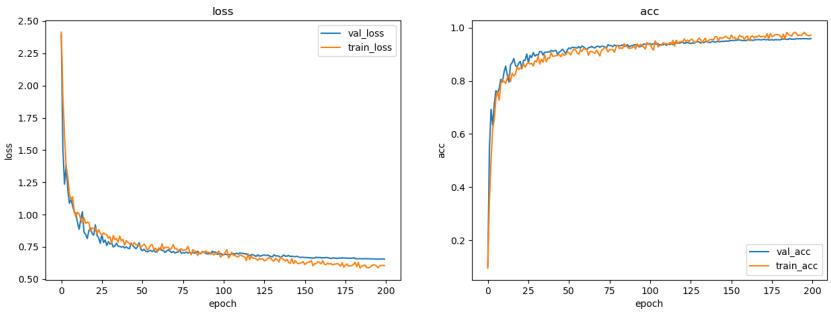

Figure 3: resnet loss and acc(ls+aa)

timizer=Adam,Weight Decay=5e-5,LR Scheduler=Cosine,Init LR=1e-3, Last LR=1e-5, Warmup=5 epochs,Dropout=0.0,Label Smoothing=0.1, autoaugment=True.

5 SVM

For linearly differentiable data, SVM finds an optimal hyperplane such that the support vectors on either side of the hyperplane maximize the distance to the hyperplane. For linearly indivisible data, SVM maps the data to a higher dimensional space by using a kernel function to make it linearly separable in the higher dimensional space. The exact mathematical derivation is not covered in the report.

Since image task classification is obviously linearly indistinguishable data, the experiments are mainly tried with different kernel functions. We conduct several kernel such as gaussian kernel, polynomial kernel, laplacian kernel and sigmoid kernel.

To be specifically, The kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$is used to map the data from the original space to a high-dimensional feature space, making it linearly differentiable in the high-dimensional space.The mathematical expressions for the different kernel functions are as follows:

1. Radial Basis Function Kernel:$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$
2. Sigmoid Kernel:$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma\mathbf{x}_i \cdot \mathbf{x}_j + r)$

The decision function of SVM can be expressed as

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

when using kernel function shown above.
A similar experimental design was conducted for SVM model.

kernel	accuracy
linear	0.228
rbf	0.532
sigmoid	0.387

Table 4: svm performance

It can be found that the SVM model is still inferior to the basic CNN when the appropriate kernel function is selected. In my opinion, the main reason is that the ability of SVM to extract and learn data features is far less than that of resnet, and even if the nonlinear kernel function is used, the effect is still limited when dealing with complex image data.

6 ViT

Vision Transformer(ViT) is an image classification model based on the Transformer. In a nutshell, ViT classifies images by dividing them into fixed-size patches, then spreading and mapping these patches to fixed-length vectors, which are passed as input sequences to the Transformer model. The core structure of the model is shown in the original paper as pic 4 shows below.

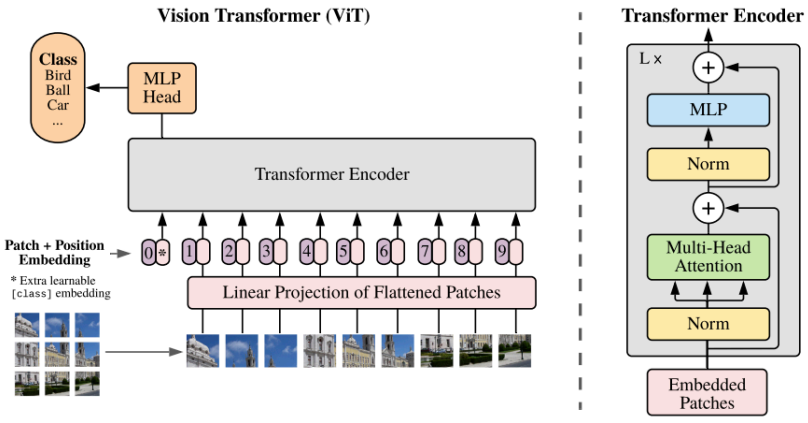

Figure 4: Vit structure in the original paper

As shown in the original paper, The model divides an image into fixed-size patches, linearly embeds each patch, and adds positional embeddings. The resulting sequence of vectors is then fed into a standard Transformer encoder. For classification, we follow the standard approach of adding an additional learnable "classification token" to the sequence.

The core point of ViT is the application of the attention mechanism to the image domain. Specifically, firstly, ViT uses a multi-head self-attention mechanism to compute the similarity between different positions in the input sequence, and aggregates the information from different positions by means of a weighted sum. Each head computes the attention independently and then joins the results together to perform a linear transformation to capture the interrelationships of different features. Second, unlike Convolutional Neural Networks (CNNs), ViT's attention mechanism can directly capture long-range dependencies of an image on a global scale, which means that ViT is able to focus on important regions in an image and make classification decisions based on these regions.

We design the basic structure of model with the following parts:

1. head:The number of heads in a multi-head self-attention mechanism indicates the number of attention heads.

2. layer:Transformer The number of encoder layers, indicating the depth of the model.

3. hidden:The hidden layer dimension of each attention header, indicating the size of the output vector of the attention mechanism.

4. mlp:The dimension of the hidden layer of a multilayer perceptron in a feedforward neural network, denoting the size of the intermediate layer of the FFN.

head	layer	hidden	mlp
12	7	384	1536

Table 5: Details of Vision Transformer model variants

After an account of the basic structure of the model and the model in the experiment, we focus on some tricks related to specific experiments to improve the effect and performance of the model through rational design.

Data augmentation is an important step to improve the performance of ViT. First, since ViT has high model complexity and is prone to overfitting on training data, data enhancement improves its generalization ability by generating diverse training samples so that the model can see more different image variants during training. Second, data augmentation can simulate different shooting conditions, viewpoints, and image distortions to increase the diversity of the dataset. This helps the model to be more robust in processing real-world images. Further, in some cases, training data may be limited. Data augmentation compensates

for the lack of data by transforming the existing data to generate more training samples, which of course does not exist in CIFAR-10. We conduct data enhancing with args parameters mentioned in session 3.

Besides, in our experiments, we choose the warmup algorithm combined with the cosine annealing algorithm to design the learning rate. Through the warmup phase, the learning rate is gradually increased from a smaller value, and the model can gradually adjust the parameters during the stable learning process, thus reducing the initial training instability. The cosine annealing learning rate strategy can gradually reduce the learning rate in the later stages of training, so that the model parameters can be adjusted more finely when they are close to the global optimal solution, reducing the risk of jumping out of the optimal solution, thus improving the final performance of the model. The learning rate changes during training are shown in Fig 5

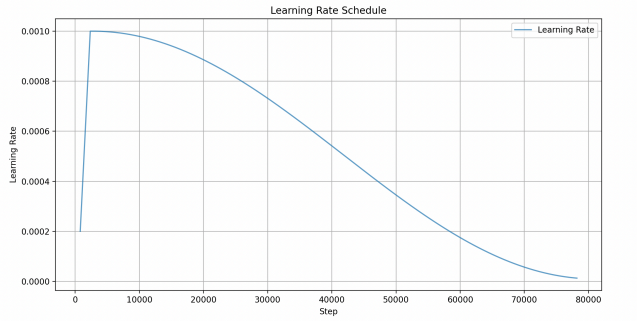

Figure 5: change of learning rate in ViT experiments

Specifically, the parameters are specifically chosen while using different data preprocessing and data enhancement methods, and the experimental setup choices and corresponding results are shown in the table 6.

exp name	accuracy	val accuracy
initial	1.0	0.854
auto augment	0.944	0.899
ls+aa	0.937	0.898

Table 6: ViT performance

The loss image and accuracy image without any data preprocessing are shown below in pic 6.

When using data preprocessing as auto augment, the loss image and accuracy image are shown below in pic 7.

When using data preprocessing as label-smoothing and auto augment, the loss image and accuracy image are shown below in pic 8.

As shown from the figures above, the overfitting of the model training is significantly alleviated by adding the auto augment operation.

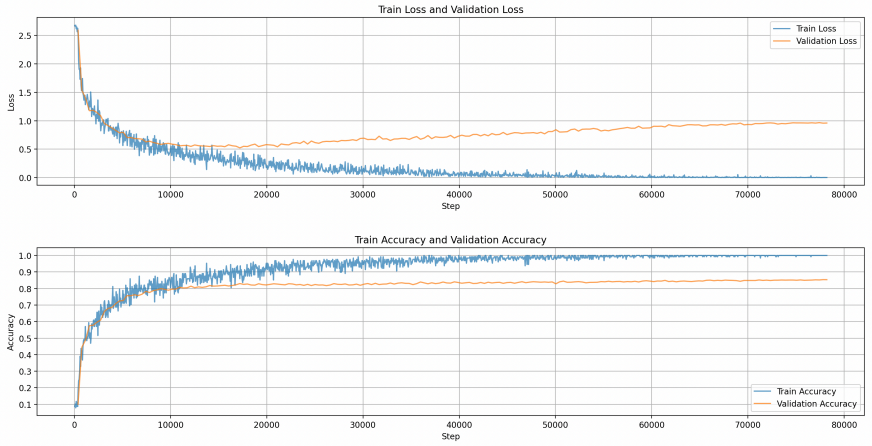

Figure 6: ViT loss and acc

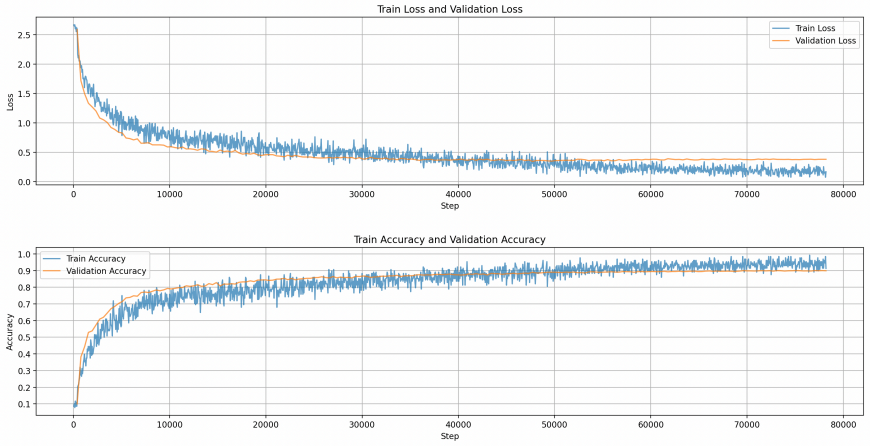

Figure 7: ViT loss and acc(aa)

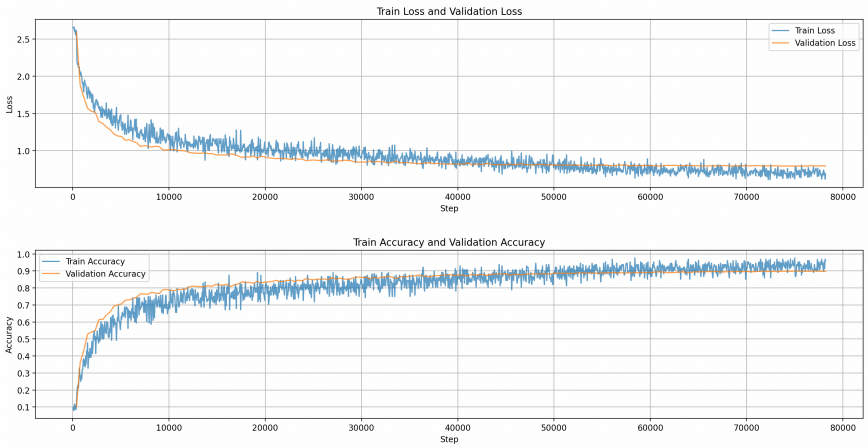

Figure 8: ViT loss and acc(aa+ls)

The training parameters of ViT are set to:Epoch=200,Batch Size=128, Optimizer=Adam,Weight Decay=5e-5,LR Scheduler=warmup+Cosine,Init LR=1e-3, Last LR=1e-5,Warmup=5 epochs,Dropout=0.0,AutoAugment=True,Label Smoothing=0.1,Heads=12,Layers=7,Hidden=384,MLP Hidden=1536.

7 Demo

The Demo part randomly selects three images in the CIFAR-10 dataset, and demonstrates the comparison between the predicted and standard results by importing the parameters stored in the above experiment in the corresponding model.

The model can be selected by passing the parameter below to demo.py :

1. –model:[SVM, basic_cnn, resnet, Vit]
2. –weights: model name + '_c10' + data preprocess methods+'.pth'

Figure 9: basic cnn demo

In this way, a simple demo program can be obtained, and by choosing the model parameters with the best results in the experiment for demonstration, the corresponding results are obtained as follows in fig 9 10 11.

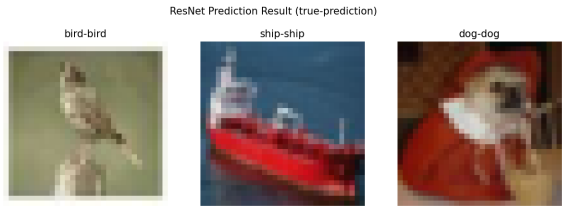

Figure 10: Resnet demo

Figure 11: ViT demo

8 Conclusion

Through this task, image classification on CIFAR-10 dataset was practiced based on SVM, CNN, and attention network (ViT).

In terms of results, SVM is less effective when choosing linear kernels and generally effective when choosing appropriate nonlinear kernels; basic CNN has similar classification effects to nonlinear kernel SVM due to its own simpler structure, but still performs much better; ResNet deepens the network structure by stacking appropriate residual blocks, thus learning more high-dimensional features, and has the best results on the overall classification task; ViT has the best results due to its model has a large number of parameters and does not have a convolutional layer to extract local features, so it needs to use appropriate data preprocessing and data enhancement methods, and also achieves better classification accuracy after choosing the methods designed in the experiments.

In regard to the reason why ViT does not perform as well as ResNet on the CIFAR-10 dataset, we think it's mainly due to the smaller dataset size, the difference in model architectures, the need for data enhancement, and the complexity of the training strategy and hyper-parameter tuning.The convolutional

architecture of ResNet is more suitable for small-scale and small-size images, while ViT is more suitable for large-scale datasets.