Contents

0 Preamble

0.1 Changing Output Dimensions

Since the selected dataset, CIFAR-10, classifies images into 10 distinct classes, each of the models therefore needed to have their output dimensions reconfigured. For details as to where and how this was accomplished for each model, view the notebook implementing each model, in particular. Each notebook will be referenced in the appropriate section.

0.2 Source Code Adaptations

Furthermore, the `train` and `evaluate` methods utilised in each of the following models are adapted from the `train_ch6` [source] and `evaluate_accuracy_gpu` [source] methods from Dive into Deep Learning, listed in References.

1 Training AlexNet from Scratch and Fine-Tuning the Model on CIFAR-10

1.1 Training from Scratch

The notebook is given by `alexnet_from_scratch.ipynb`. The file path is given by `Assignment/Source/src/section_1/alexnet_from_scratch/alexnet_from_scratch.ipynb`.

1.1.1 Hyperparameters

For this implementation of AlexNet, `learning_rate`, `num_epochs`, and `batch_size` are defined as `0.05`, `10`, and `256`, respectively.

1.1.2 Results

Epoch	Training Loss	Training Accuracy	Testing Accuracy
1	2.302	0.102	0.101
2	2.301	0.115	0.119
3	2.298	0.116	0.106
4	2.256	0.127	0.196
5	2.132	0.232	0.257
6	2.015	0.265	0.269
7	1.952	0.282	0.298

8	1.876	0.312	0.307
9	1.791	0.347	0.385
10	1.704	0.379	0.399

Table 1.1 Training AlexNet from Scratch

The raw output is given in `alexnet_from_scratch.txt.` The file path is given by `Assignment/Source/src/section_1/alexnet_from_scratch/alexnet_from_scratch.txt`.

1.2 Fine-Tuning

The notebook is given by `alexnet_fine_tuned.ipynb`. The file path is given by `Assignment/Source/src/section_1/alexnet_fine_tuned/alexnet_fine_tuned.ipynb`.

1.2.1 Hyperparameters

For this implementation of AlexNet, `learning_rate`, `num_epochs`, and `batch_size` are defined as `0.05`, `10`, and `256`, respectively.

1.2.2 Results

Epoch	**Training Loss**	**Training Accuracy**	**Testing Accuracy**
1	0.762	0.734	0.822
2	0.470	0.836	0.835
3	0.404	0.859	0.873
4	0.353	0.877	0.860
5	0.320	0.887	0.862
6	0.300	0.895	0.859
7	0.277	0.902	0.899
8	0.256	0.910	0.891
9	0.238	0.917	0.885
10	0.223	0.922	0.897

Table 1.2. Fine-Tuning AlexNet

The raw output is given in `alexnet_fine_tuned.txt.` The file path is given by `Assignment/Source/src/section_1/alexnet_fine_tuned/alexnet_fine_tuned.txt`.

1.3 Comparison

Ultimately, the most performant model was the fine-tuned AlexNet. It classified unseen images with 225% greater accuracy than the AlexNet model trained from scratch.

2 Fine-Tuning ResNet-18 and VGG16 on CIFAR-10

2.1 ResNet-18

The notebook is given by `resnet18.ipynb`. The file path is given by `Assignment/Source/src/section_2/resnet18/resnet18.ipynb`.

2.1.1 Hyperparameters

For this implementation of ResNet-18, `learning_rate`, `num_epochs`, and `batch_size` are defined as `0.05`, `10`, and `256`, respectively.

2.1.2 Results

Epoch	**Training Loss**	**Training Accuracy**	**Testing Accuracy**
1	0.420	0.866	0.918
2	0.140	0.956	0.927
3	0.070	0.980	0.943
4	0.032	0.994	0.946
5	0.015	0.999	0.941
6	0.008	1.000	0.947
7	0.005	1.000	0.947
8	0.003	1.000	0.948
9	0.002	1.000	0.949
10	0.002	1.000	0.950

Table 2.1. Fine-Tuning ResNet-18

The raw output is given in `resnet18.txt.` The file path is given by `Assignment/Source/src/section_2/resnet18/resnet18.txt`.

2.2 VGG16

The notebook is given by `vgg16.ipynb`. The file path is given by `Assignment/Source/src/section_2/vgg16/vgg16.ipynb`.

2.2.1 Hyperparameters

For this implementation of ResNet-18, `learning_rate`, `num_epochs`, and `batch_size` are defined as `0.05`, `10`, and `256`, respectively.

2.2.2 Results

Epoch	**Training Loss**	**Training Accuracy**	**Testing Accuracy**
1	1.234	0.566	0.862
2	0.336	0.885	0.881
3	0.196	0.933	0.906
4	0.137	0.953	0.900
5	0.108	0.965	0.907
6	0.056	0.981	0.926
7	0.047	0.984	0.916
8	0.063	0.980	0.918
9	0.025	0.992	0.935
10	0.021	0.994	0.937

Table 2.2. Fine-Tuning VGG16

The raw output is given in `vgg16.txt.` The file path is given by `Assignment/Source/src/section_2/vgg16/vgg16.txt`.

2.3 Comparison

Overall, the most performant model was the fine-tuned ResNet-18. It classified unseen images with 1.4% and 5.9% greater accuracy than VGG16 and AlexNet, respectively.

3 Training Softmax Regression from Scratch on ResNet-18 Features and CIFAR-10

3.1 ResNet-18 Features at Last Convolutional Layer

The notebook is given by `softmax_resnet18_last.ipynb`. The file path is given by `Assignment/Source/src/section_3/softmax_resnet18_last/softmax_resnet18_last.ipynb`.

3.1.1 Hyperparameters

For this model, `learning_rate`, `num_epochs`, and `batch_size` are defined as `0.05`, `10`, and `256`, respectively.

3.1.2 Results

Epoch	Training Loss	Training Accuracy	Testing Accuracy
1	1.744	0.762	0.909
2	1.543	0.934	0.931
3	1.513	0.958	0.937
4	1.497	0.972	0.940
5	1.487	0.980	0.942
6	1.481	0.985	0.944
7	1.476	0.989	0.944
8	1.473	0.991	0.946
9	1.471	0.992	0.949
10	1.470	0.993	0.945

Table 3.1. Training Softmax Regression Model on ResNet-18 Features at the Last Convolutional Layer

The raw output is given in `softmax_resnet18_last.txt.` The file path is given by `Assignment/Source/src/section_3/softmax_resnet18_last/softmax_resnet18_last.txt`.

3.2 ResNet-18 Features at Selected Intermediate Layer

The notebook is given by `softmax_resnet18_intermediate.ipynb`. The file path is given by `Assignment/Source/src/section_3/softmax_resnet18_intermediate/softmax_resnet18_intermediate.ipynb`.

3.2.1 Hyperparameters

For this model, `learning_rate`, `num_epochs`, and `batch_size` are defined as `0.05`, `10`, and `256`, respectively.

3.2.2 Results

Epoch	**Training Loss**	**Training Accuracy**	**Testing Accuracy**
1	2.361	0.100	0.100
2	2.361	0.100	0.100
3	2.361	0.100	0.100
4	2.361	0.100	0.100
5	2.361	0.100	0.100
6	2.361	0.100	0.100
7	2.361	0.100	0.100
8	2.361	0.100	0.100
9	2.361	0.100	0.100
10	2.361	0.100	0.100

Table 3.2. Training Softmax Regression Model on ResNet-18 Features at a Selected Convolutional Layer

The raw output is given in `softmax_resnet18_intermediate.txt.` The file path is given by `Assignment/Source/src/section_3/softmax_resnet18_intermediate/softmax_resnet18_intermediate.txt`.

3.3 Original Input

The notebook is given by `softmax_resnet18_original.ipynb`. The file path is given by `Assignment/Source/src/section_3/softmax_resnet18_original/softmax_resnet18_original.ipynb`.

3.3.1 Hyperparameters

For this model, `learning_rate`, `num_epochs`, and `batch_size` are defined as `0.1`, `10`, and `256`, respectively.

3.3.2 Results

Epoch	Training Loss	Training Accuracy	Testing Accuracy
1	1.744	0.762	0.909
2	1.543	0.934	0.931
3	1.513	0.958	0.937
4	1.497	0.972	0.940
5	1.487	0.980	0.942
6	1.481	0.985	0.944
7	1.476	0.989	0.944
8	1.473	0.991	0.946
9	1.471	0.992	0.949
10	1.470	0.993	0.945

Table 3.3. Training Softmax Regression Model on the Original Input

The raw output is given in `softmax_original.txt.` The file path is given by `Assignment/Source/src/section_3/softmax_original/softmax_original.txt.`

3.4 Comparison

Overall, the most performant model was the Softmax Regression Model trained on the ResNet-18 features at the last convolutional layer. It classified unseen images with 945% greater accuracy than both the Softmax Regression model trained on the ResNet-18 features at an intermediate layer and the Softmax Regression model trained on the original input.

References

Zhang, Aston, Lipton, Zachary C., Li, Mu, and Smola, Alexander J. *Dive into Deep Learning*. https://d2l.ai/.