Deep Learning Assignment II - Report

Chen-Yang Yu,

AI Robotics, National Cheng Kung University

Jun 20, 2024

<https://github.com/LittleFish-Coder/deep-learning-assignment-2>

1. **Dataset:**

Mini-ImageNet is a subset of the ImageNet dataset, which contains 50 classes.

* Training Set: 63325
* Validation Set: 450
* Testing Set: 450

**Sample images from the dataset:**

|  |  |  |
| --- | --- | --- |
| n02111277\_1207.JPEG | n02112137\_7.JPEG | n02112137\_112.JPEG |
| 一張含有 哺乳動物, 狗, 戶外, 草 的圖片  自動產生的描述 | 一張含有 哺乳動物, 狗, 狗飼養, 寵物 的圖片  自動產生的描述 | 一張含有 狗, 哺乳動物, 寵物, 狗飼養 的圖片  自動產生的描述 |

**Data Preprocessing**

It's worth noting that each image does not have the same size, so we need to standardize the size of all images. Moreover, some images are in RGB format, while others are in grayscale format.

* Image Size: we resize all images to **256x256** pixels.
* Image Channel: we convert all images to **RGB format (3 channels).**

**Evaluation Performance**

During training, we will evaluate the model using the validation set. Performance metrics such as test accuracy and loss will be discussed in the next 2 sections.

1. **Task 1: Designing a Convolution Module for Variable Input Channels**

We will discuss the 2 models’ performance for dynamic input channels image.

* 1. **Hyperparameters**
* Epoch: 25
* Learning Rate: 0.001
* Batch Size: 64
* Input Size: (3, 256, 256)
* Optimizer: Adam
* Criterion: CrossEntropyLoss
  1. **Simple CNN**
* Model Architecture:

Layer (type) Output Shape Param #

=============================================================

Conv2d-1 [-1, 32, 256, 256] 896

BatchNorm2d-2 [-1, 32, 256, 256] 64

MaxPool2d-3 [-1, 32, 128, 128] 0

Conv2d-4 [-1, 64, 128, 128] 18,496

BatchNorm2d-5 [-1, 64, 128, 128] 128

MaxPool2d-6 [-1, 64, 64, 64] 0

Conv2d-7 [-1, 128, 64, 64] 73,856

BatchNorm2d-8 [-1, 128, 64, 64] 256

MaxPool2d-9 [-1, 128, 32, 32] 0

AdaptiveAvgPool2d-10 [-1, 128, 1, 1] 0

Linear-11 [-1, 256] 33,024

Dropout-12 [-1, 256] 0

Linear-13 [-1, 50] 12,850

=============================================================

Total params: 139,570

Trainable params: 139,570

Non-trainable params: 0

----------------------------------------------------------------

Input size (MB): 0.75

Forward/backward pass size (MB): 63.01

Params size (MB): 0.53

Estimated Total Size (MB): 64.29

----------------------------------------------------------------

* **Validation Accuracy & Loss**

We save the best model based on the validation loss

* + - * Best Validation Accuracy: 0.3379
      * Best Validation Loss: 0.0349

一張含有 圖表, 繪圖, 行, 文字 的圖片

自動產生的描述

* **Test Accuracy & Loss**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **RGB** | **RG** | **GB** | **R** | **G** | **B** |
| **Accuracy** | 0.3556 | 0.3111 | 0.0378 | 0.0222 | 0.0222 | 0.02 |
| **Loss** | 0.0412 | 0.0479 | 0.1202 | 0.2145 | 0.1152 | 0.2187 |

* **Dataset Design**

We will test the model on 6 downstream tasks, each using different combinations of the RGB channels.

Since the simple CNN can only process images with 3 channels, we have redesigned the test dataset to accommodate this limitation. When testing different channel combinations, we will first select the desired channels and then copy these channels to a new image. The channels that are not selected will be padded with zeros. This ensures that the image remains in a 3-channel format.

For example, if the 'RG' combination is selected, we will copy the 'R' and 'G' channels from the raw image. The 'B' channel will be set to zero.

Below are samples for images with each selected channel combination:

|  |  |  |
| --- | --- | --- |
| RGB | RG | GB |
| 一張含有 狗, 哺乳動物, 草, 狗飼養 的圖片  自動產生的描述 | 一張含有 哺乳動物, 狗, 草, 狗飼養 的圖片  自動產生的描述 | 一張含有 狗, 哺乳動物, 草, 狗飼養 的圖片  自動產生的描述 |
| R | G | B |
| 一張含有 哺乳動物, 狗, 紅色, 小狗 的圖片  自動產生的描述 | 一張含有 哺乳動物, 狗, 狗飼養, 寵物 的圖片  自動產生的描述 | 一張含有 礁, 魚缸, 哺乳動物, 狗 的圖片  自動產生的描述 |

* 1. **Dynamic CNN**
     + **Model Architecture**

Layer (type) Output Shape Param #

=============================================================

Conv2d-1 [-1, 16, 254, 254] 160

ReLU-2 [-1, 16, 254, 254] 0

Conv2d-3 [-1, 1, 252, 252] 145

Conv2d-4 [-1, 16, 254, 254] 160

ReLU-5 [-1, 16, 254, 254] 0

Conv2d-6 [-1, 1, 252, 252] 145

PoolChannelAttention-7 [-1, 1, 252, 252] 0

Conv2d-8 [-1, 32, 252, 252] 320

BatchNorm2d-9 [-1, 32, 252, 252] 64

MaxPool2d-10 [-1, 32, 126, 126] 0

Conv2d-11 [-1, 64, 126, 126] 18,496

BatchNorm2d-12 [-1, 64, 126, 126] 128

MaxPool2d-13 [-1, 64, 63, 63] 0

Conv2d-14 [-1, 128, 63, 63] 73,856

BatchNorm2d-15 [-1, 128, 63, 63] 256

MaxPool2d-16 [-1, 128, 31, 31] 0

AdaptiveAvgPool2d-17 [-1, 128, 1, 1] 0

Linear-18 [-1, 256] 33,024

Dropout-19 [-1, 256] 0

Linear-20 [-1, 50] 12,850

CNN-21 [-1, 50] 0

=============================================================

Total params: 139,604

Trainable params: 139,604

Non-trainable params: 0

----------------------------------------------------------------

Input size (MB): 0.75

Forward/backward pass size (MB): 93.98

Params size (MB): 0.53

Estimated Total Size (MB): 95.26

----------------------------------------------------------------

* + - **Validation Accuracy & Loss**

We save the best model based on the validation loss

* + - * Best Validation Accuracy: 0.2889
      * Best Validation Loss: 0.0410

一張含有 文字, 繪圖, 圖表, 行 的圖片

自動產生的描述

* + - **Test Accuracy & Loss**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **RGB** | **RG** | **GB** | **R** | **G** | **B** |
| **Accuracy** | 0.3044 | 0.3111 | 0.2911 | 0.3089 | 0.3089 | 0.2467 |
| **Loss** | 0.0480 | 0.0479 | 0.0483 | 0.0489 | 0.0478 | 0.0526 |

* + - **Dataset Design**

The Dynamic CNN can handle different input channels. Therefore, if 'RG' is selected for testing, we will create a new image containing only the 'RG' channel information from the raw image.

In this way, we can create images with the following dimensions:

* (1, height, width) for 'R', 'G', or 'B'
* (2, height, width) for 'RG' or 'GB'
  + - * (3, height, width) for 'RGB'
  1. **Comparison**

By comparing the results from the testing phase of these two models, the dynamic CNN outperforms the simple one on the 6 downstream tasks.

1. **Task 2: Designing a Two-Layer Network for Image Classification**
2. **Hyperparameters**
   * Epoch: 50
   * Learning Rate: 0.001
   * Batch Size: 64
   * Input Size: (3, 256, 256)
   * Optimizer: Adam
   * Criterion: CrossEntropyLoss
3. **ResNet34**

* **Model Architecture**

----------------------------------------------------------------

Total params: 21,310,322

Trainable params: 21,310,322

Non-trainable params: 0

----------------------------------------------------------------

Input size (MB): 0.75

Forward/backward pass size (MB): 125.75

Params size (MB): 81.29

Estimated Total Size (MB): 207.80

----------------------------------------------------------------

* **Validation Accuracy & Loss:**
  + - * Best Validation Accuracy: 0.5622
      * Best Validation Loss: 0.0212一張含有 文字, 圖表, 行, 繪圖 的圖片

        自動產生的描述
* **Test Accuracy & Loss**
  + - * Test Accuracy: 0.5978
      * Test Loss: 0.0221

1. **Attention CNN**
   * **Model Architecture**

Layer (type) Output Shape Param #

=============================================================

Conv2d-1 [-1, 32, 256, 256] 896

BatchNorm2d-2 [-1, 32, 256, 256] 64

ReLU-3 [-1, 32, 256, 256] 0

Conv2d-4 [-1, 32, 256, 256] 128

BatchNorm2d-5 [-1, 32, 256, 256] 64

ReLU-6 [-1, 32, 256, 256] 0

Block-7 [-1, 32, 256, 256] 0

MaxPool2d-8 [-1, 32, 128, 128] 0

Conv2d-9 [-1, 64, 128, 128] 18,496

BatchNorm2d-10 [-1, 64, 128, 128] 128

ReLU-11 [-1, 64, 128, 128] 0

Conv2d-12 [-1, 64, 128, 128] 2,112

BatchNorm2d-13 [-1, 64, 128, 128] 128

ReLU-14 [-1, 64, 128, 128] 0

Block-15 [-1, 64, 128, 128] 0

MaxPool2d-16 [-1, 64, 64, 64] 0

Conv2d-17 [-1, 128, 64, 64] 73,856

BatchNorm2d-18 [-1, 128, 64, 64] 256

ReLU-19 [-1, 128, 64, 64] 0

Conv2d-20 [-1, 128, 64, 64] 8,320

BatchNorm2d-21 [-1, 128, 64, 64] 256

ReLU-22 [-1, 128, 64, 64] 0

Block-23 [-1, 128, 64, 64] 0

MaxPool2d-24 [-1, 128, 32, 32] 0

Conv2d-25 [-1, 256, 32, 32] 295,168

BatchNorm2d-26 [-1, 256, 32, 32] 512

ReLU-27 [-1, 256, 32, 32] 0

Conv2d-28 [-1, 256, 32, 32] 33,024

BatchNorm2d-29 [-1, 256, 32, 32] 512

ReLU-30 [-1, 256, 32, 32] 0

Block-31 [-1, 256, 32, 32] 0

MaxPool2d-32 [-1, 256, 16, 16] 0

Conv2d-33 [-1, 1, 16, 16] 98

Sigmoid-34 [-1, 1, 16, 16] 0

SpatialAttention-35 [-1, 1, 16, 16] 0

Dropout-36 [-1, 256, 16, 16] 0

AdaptiveAvgPool2d-37 [-1, 256, 1, 1] 0

Linear-38 [-1, 50] 12,850

=============================================================

Total params: 446,868

Trainable params: 446,868

Non-trainable params: 0

----------------------------------------------------------------

Input size (MB): 0.75

Forward/backward pass size (MB): 218.01

Params size (MB): 1.70

Estimated Total Size (MB): 220.46

* + **Validation Accuracy & Loss**
    - * Best Validation Accuracy: 0.5222
      * Best Validation Loss: 0.0235

一張含有 文字, 繪圖, 圖表, 行 的圖片

自動產生的描述

* + **Test Accuracy & Loss**
    - * Test Accuracy: 0.5911
      * Test Loss: 0.0239

1. **Comparison**

By adding a Spatial Attention module to a simple CNN with some residual blocks, we are able to rival the ResNet34 model.

|  |  |  |
| --- | --- | --- |
|  | ResNet34 | Attention CNN |
| Test Accuracy | 0.5978 | 0.5911 |
| Test Loss | 0.0221 | 0.0239 |
| Total params | 21,310,322 | 446,868 |