Medical Imaging Computing Assignment 1

Suneet Tipirneni

March 20, 2023

1 Code

Code for the assignment can be found at `https://github.com/suneettipirneni/med-proj-1`

2 Task 1

2.1 Training resnet-18 from scratch

2.1.1 Implementation

The model I used for this assignment was the low-fidelity resnet-18 model. For input transforms each image needed to be resized to 224 as this is what the resnet model expects as an input dimension.

For the number of epochs I chose 30 as it seem to give the best balance between both computational runtime and overall accuracy. For a batch size a size of 5 was chosen as this was a similiar batch size used by [1] for their model training, mine is increased by one as it helps improve how fast the model trains. The results are shown in figure 1.

2.1.2 Graphs

2.1.3 Accurracies

The overall accuracy of this model was 92%, the outline of the different percentages for each class is shown below in table 1

Normal	87%
Pneumonia	97.4%

Table 1: Individual Accurracies for model 1.1

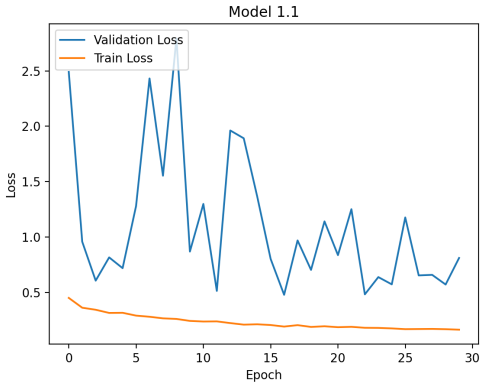

Figure 1: Results from pretraining the model fully

The calculation of these individual accuracies uses the code from [2].

2.2 Finetuning a resnet-18

2.2.1 Implementation

For the second task a resnet-18 model was once again used, however the model was trained with it's preexisting weights that were previously learned on the ImageNet dataset.

As with the previous model, a batch size of 5 and and the number of epochs were set to 30. These values were kept the same in order to make sure a fair comparison could be formed between both models.

This model mainly differs in the architecture described in section 2. A dropout layer with a rate of 0.5 is employed before the final fully-connected layer to prevent over-fitting. In addition to architecture

changes, data augmentation is performed to increase the fidelity of the test dataset. Firstly, a random horizontal transformation is applied to input images, horizontal transforms were only used as opposed to vertical transformsas the body would be misrepresented by a vertical flip. Secondly, a random crop to the network input size of 224 is employed this allows the inputs to crop to a variety of different areas and helps the model to not over-train onspecific formats and shapes of images.

2.2.2 Graphs

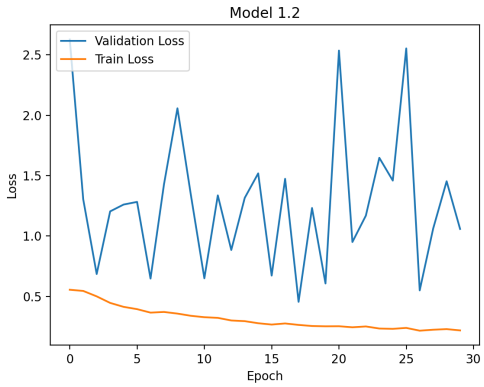

Figure 2: Results from finetuning the model using pretrained resnet-18 weights

2.2.3 Accurracies

The overall test accuracy for this model was 89%. This was slightly lower than the previous model, this may be due to lack of training time. The per-class accuracies for this model are shown in table 2.

Normal	76.5%
Pneumonia	95.4%

Table 2: Individual Accurracies for model 1.2

Compared to the first model these are slightly lower scores. I once again believe this may be be caused by the lack of training time for the model. Perhaps increasing the epoch count would yield better results.

References

[1] Chilamkurthy, S. Transfer learning for computer vision tutorial. `https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html`.

[2] PyTorch. Training a classifier. `https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html`.