

assignment 1 report - deep learning

enwin anson and antoine dufour

experiment 1

For this classification of Cat and Dog images, inspired by the tutorial of keras, using a CNN model. We got :

Number of Epoch : 30
Train Loss: 0.2869 | Acc: 87.70%
Val Loss: 0.3206 | Acc: 85.73%
Best Validation Accuracy: 86.57%

We can see that the accuracy is rather good.

For the classification on Stanford Dogs dataset, we got :

Number of Epoch : 30
Train Loss: 2.2665 | Acc: 42.84%
Val Acc: 13.80%
Best Validation Accuracy: 13.80%

We can see that the accuracy and the loss are not as good as the first part of the experiment, and rather medium.

This poor performance of the model can be explained by the fact that the model architecture, which is the same as the first part experiment, is designed for 2 class (cat and dog), explaining why it's struggling when there are in this experiment 120 classes. Also, the number of epoch might not be enough to get a good classification on this amount of different classes.

We can also say that with a better normalisation, better data scaling, and better data augmentations, we might see better result.

experiment 2

We can see that when we load the saved model and replace only the output layer of the model, using a Binary Cross-Entropy Loss, adding padding in the convolutional layers, with pre trained layers, we got better results :

Number of Epoch : 50
Train Loss: 0.1067 | Acc: 96.21%
Validation Loss: 0.5644 | Acc: 83.01%

Training complete!
Best Validation Accuracy: 83.75%

We got better result because now the model can map to a single probability score (of dog versus cat), and not to choose between 120 classes, hence a much simpler task.

experiment 3

We can see that when we load the model and replace the output layer and the two first convolutional layers we good lesser good results :

with :

Feature shape: torch.Size([1, 128, 20, 20])

Classifier input: 51200

Using device: cuda

we got :

Number of Epoch : 50

Train Loss: 0.6035 | Acc: 67.08%

Validation Loss: 0.5978 | Acc: 67.78%

Best Validation Accuracy: 68.16%

This is due to the fact that experiment 3 have the first two convolutional layers reinitialised with random initialisation, while experiment 2 have all convolutional layers on pre-trained weights. We know that using pre-trained weights is useful to get better result, while reinitialised layers are erasing the previously learning dog-specific patterns.

We can see a poor adaptation on the cat/dog task also because of frozen layers, who prevent adaptation to the new features of the cat/dog dataset.

experiment 4 :

With these architectures :

```
Trainable parameters: 92,865
Total parameters: 26,308,673
Model architecture:
Experiment4Model(
  (features): Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
    (7): ReLU()
    (8): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (classifier): Sequential(
    (0): Flatten(start_dim=1, end_dim=-1)
    (1): Dropout(p=0.5, inplace=False)
    (2): Linear(in_features=51200, out_features=512, bias=True)
    (3): ReLU()
    (4): Linear(in_features=512, out_features=1, bias=True)
    (5): Sigmoid()
  )
)
```

Three convolutions, first convolution is a pre trained layer, then second and third are for being trained. The classifier contains a linear hidden layer, then a ReLu activation, then a final linear followed by a sigmoid activation, producing the probability.

We got this result :

Number of Epoch : 50

Train Loss: 0.3804 | Acc: 83.16%

Validation Loss: 0.3796 | Acc: 82.77%

Best Validation Accuracy: 83.85%

We can conclude that the fined tuning of the pre-trained model with modified layer was successful, getting a good validation loss and accuracy.