Transfer Learning Experiment in CNNs

Aswathy Gopalakrishnan,Yonge Li

April 7, 2025

1 Introduction

The objective of the assignment is to explore transfer learning in Convolutional Neural Networks (CNNs) through a series of experiments focused on layer replacement. The network architecture and training parameters are kept consistent across the experiments to ensure valid comparisons. The changes in each experiment and their impact on the results will be documented in this comprehensive report.

2 implementation

2.1 Base Model Setup

In this, a CNN model to distinguish between cats and dogs is implemented based on the provided Keras tutorial. This serves as a base model for understanding on how CNN performs on different classification tasks, utilizing a learning rate of 0.0001.

Epoch	Train Accuracy	Val accuracy	Train Loss	Val Loss
1	0.6803	0.4950	0.5995	0.6981
15	0.9376	0.9027	0.1596	0.2528
25	0.9606	0.9424	0.1035	0.1384

Figure 1: Tutorial Model on Cats and Dogs Dataset

The training accuracy reached 96.06% by 25th epoch indicating that the model learned the features well. Validation accuracy is also high at 94.24% indicating that the model generalised well on unseen data. The train loss decreased consistently showing that the model made proper weight updates. Validation loss also decreased showing that the model wasn't overfitting.

2.2 Experiment 1

The base line model is modified to adapt to complex multi-class problems in this experiment. It utilizes the Stanford Dogs dataset directly from Tensorflow datasets, which comprises of 120 different dog breeds. Hence, the output layer is set to classify the 120 different classes(breeds), allowing the output to represent the likelihood of each breed. In addition to this, the loss function is set as sparse categorical cross entropy to make it suitable for multi-class classification. The model is saved to a file.

Epoch	Train Accuracy	Val accuracy	Train Loss	Val Loss
1	0.0286	0.0106	4.6591	4.7883
15	0.4571	0.1681	2.0535	3.9111
25	0.7046	0.3662	1.1486	2.4840
40	0.8570	0.4171	0.6489	2.2685
50	0.8811	0.4435	0.5672	2.1291

Figure 2: Base model trained on the Stanford Dogs dataset

At epoch 50, training accuracy is high as 88.11%. however, validation accuracy is 44.35% suggesting that the model struggled to generalize when it comes to multiclass tasks. Even in terms of loss, validation loss isn't showing a gradual loss as in training loss, indicating that the model could lead to under or overfitting.

Due to high complexity of dataset with its 120 classes, the validation accuracy was initially observed to be approximately 10%. By reducing the model complexity and applying data augmentation, it could be significantly increased to 44%, while creating an observable gap between training and validation metrics in subsequent experiments.

2.3 Experiment 2

In this, the pre-trained model from experiment 1 is utilized. Its output layer is modified to suit binary classification. A dense layer with 1 unit and a sigmoid activation function is added to output the probability of input image as a cat or a dog. The code freezes all layers of the model except the new output layer. This way, the learned features are retained and only allows the output layer to be trained. Also, the loss function is changed to binary cross entropy to suit binary classification.

The results show reasonable increase in train and validation accuracies compared to experiment 2. However, the values are still low compared to baseline model. The loss seems to

Epoch	Train Accuracy	Val accuracy	Train Loss	Val Loss
1	0.5157	0.5651	71.5375	20.1902
15	0.6229	0.7121	7.8344	3.2774
25	0.6496	0.7254	1.7433	1.0008
40	0.6825	0.7057	0.6703	0.5930
50	0.6810	0.7373	0.6469	0.5609

Figure 3: Pretrained Model replace the output layer on Cats&Dogs Dataset

decrease through the epoch. But the initial high loss indicates a misalignment of predictions in the beginning. This shows that freezing all the layers and only training the output layer isn't enough for a good model performance.

2.4 Experiment 3

In contrast to experiment 2, the first two convolutional layers also undergoes unfreezing along with the output layer here. This allows these layers to be trainable which helps the model to adjust better to the classification problem in hand. The first layers capture low level features such as edges, colours, and textures. Unfreezing them allows the model to adjust the feature detection right from the beginning. These are helpful when the data we are working with have shifted dramatically and need a re-evaluation of low-level features. All other frozen layers safeguard the learned weights from the previous training. This way, it benefits from the knowledge gained in the previous training, enabling transfer learning.

Epoch	Train Accuracy	Val accuracy	Train Loss	Val Loss
1	0.5516	0.6328	29.3369	5.4693
15	0.6555	0.7280	0.8524	0.6606
25	0.7003	0.7305	0.6078	0.5656
40	0.7550	0.7603	0.5136	0.5089
50	0.7813	0.7913	0.4653	0.4488

Figure 4: Pretrained Model replace the output and first two conv layers on Cats&Dogs

From the results, we can observe that the performance has significantly improved which shows that unfreezing the initial layers allowed the model to learn more relevant low-level features for the new binary classification task. Both train and validation values follow a similar trend showing that model is generalizing well.

2.5 Experiment 4

The last experiment focuses on unfreezing the last two convolutional layers along with the output layer. The last layers correspond to high level feature abstraction which is important for learning complex class representations. Unfreezing is ideal when we want the model to specialize and refine learned features to suit a new task. This experiment enhances the model to predict the target classes more effectively, while the frozen layers safeguard the previous learned features.

Epoch	Train Accuracy	Val accuracy	Train Loss	Val Loss
1	0.6094	0.6817	14.4871	0.9703
15	0.7707	0.7863	0.4856	0.4623
25	0.8007	0.8052	0.4395	0.4174
40	0.8445	0.8602	0.3558	0.3325
50	0.8616	0.8731	0.3167	0.3046

Figure 5: Pretrained Model replace the output and last two con layers on Cats&Dogs

The experiment shows the best results in the transfer learning experiments so far. When the model could learn and refine the high-level features, the classification accuracy increased. Also, the increase in accuracy from .60 to .86 as epochs increase shows that the model is iteratively learning from the training data, alongside minimizing the loss.

3 Conclusion

The initial training on the Stanford Dogs dataset laid a good foundation. But, in order to adapt the same to solve a new task, we need to do more than just retraining the output layer. By effectively fine tuning the convolutional layers, the model can adapt to new tasks better. The choice of which layers to unfreeze is an important choice and must be done according to the requirements of the task in hand. In short, the experiments show that transfer learning using a pretrained model and fine tuning it layers appropriately can help the model adapt better than training it from scratch.