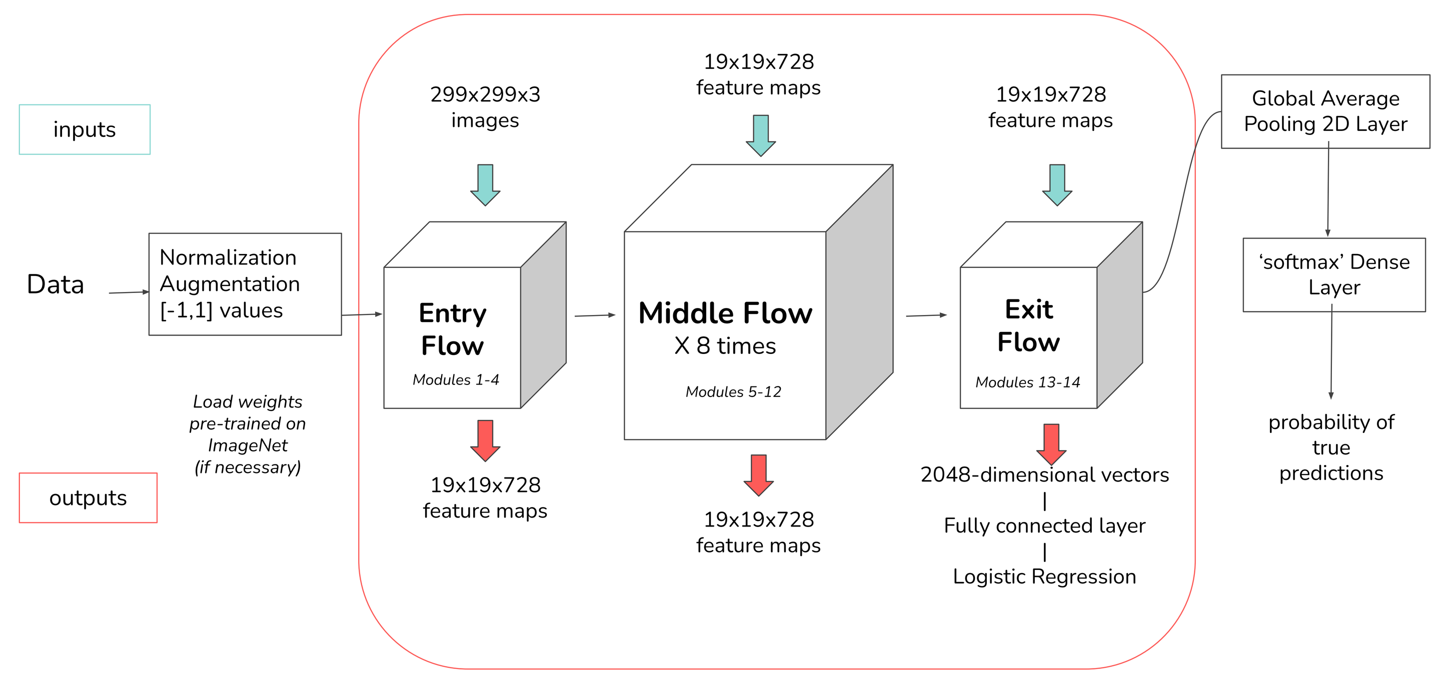
Transfer Learning for Image Classification: Exploring effective strategies

Angela Reyes Cervantes ‘22

Overview

- **Convolutional neural networks**: low-level filters convolutional layers extract are consistent across models for different image recognition tasks[2], making weights at these levels ideal for **transfer learning**.
- We explore transferring weights for different layers. Then, we examine how the complexity of the task the initial classifier was trained to solve affects performance on a task of differing complexity.

Xception Model: 36 convolution layers structured into 14 modules



Methods

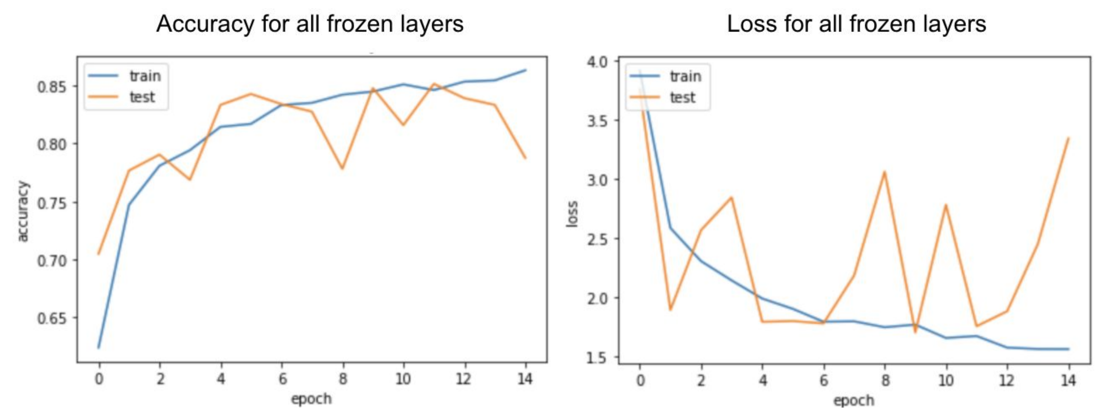
- **Dataset: Natural Images** dataset[4], 6899 examples, 8 classes.
- **Evaluating Performance:** Accuracy and Loss rate over epochs. F1 score for each class of a mode using a validation set.
- **Freezing Sets of Layers:** Four Xception models that use transferred weights trained on ImageNet, where each model has different sets of layers frozen.
- **Transfer Learning**: Train models on different tasks, then transfer weights to other models, and train each model on a task different from the original task.

Weights	Learning Rate	Accuracy	Loss
Random	0.01	137.8%	207.54%
Transferred	0.01	99.71%	1.3%
Random	0.001	88.91%	275.59%
Transferred	0.001	99.95%	0.61%
Random	0.0001	87.6%	37.87%
Transferred	0.0001	99.71%	0.61%

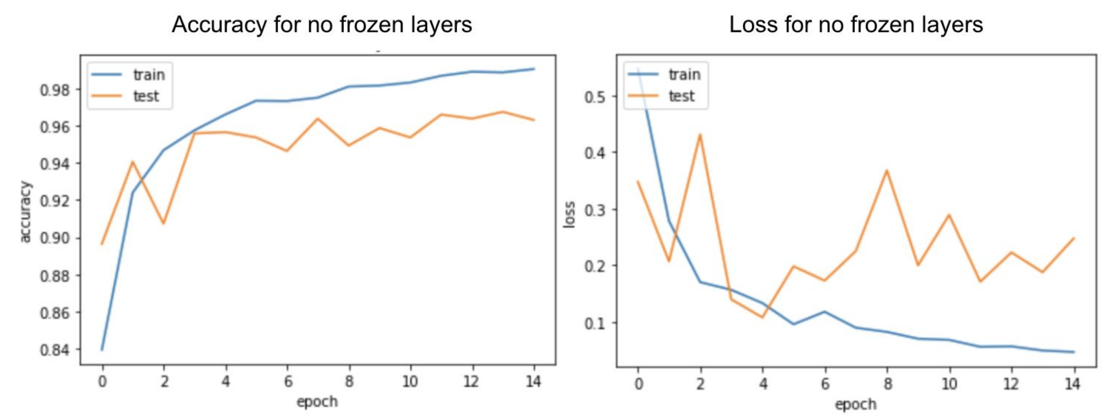
Table 1. Finding optimal learning rate: accuracy and loss of models with random and transferred weights and different learning rates.

Freezing Layers

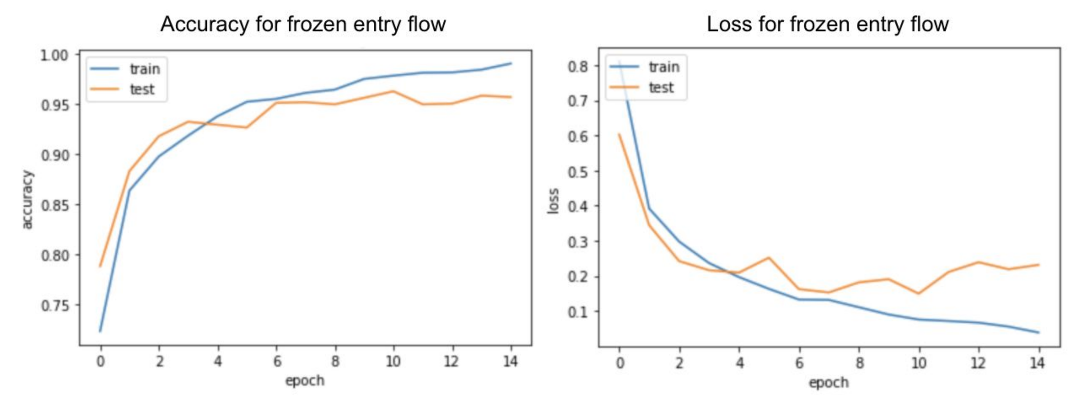
All Frozen



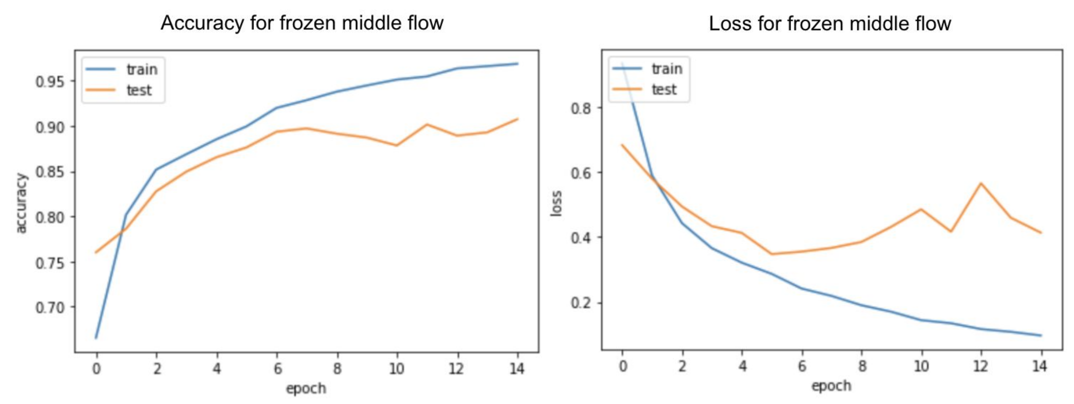
None Frozen



Frozen Entry Flow



Frozen Entry + Middle Flow



Model	Accuracy	Loss
3.1 All Frozen	78.6%	334.1%
3.2 None Frozen	95.7%	26.3%
3.3 Frozen Entry	94.8%	24.79%
3.4 Frozen Entry + Middle	90.8%	42.76%

Table 2. Accuracy and loss of models with different sets of layers frozen.

- Models 3.1 and 3.2 show transferring weights improves training, but re-training is necessary to achieve peak accuracy for Model 3.1.
- Table 2 demonstrates Model 3.3 outperformed Model 3.4, showing that ImageNet weights in early convolution layers are general enough to transfer. Layers in the Middle Flow would have benefited from learning weights specific for classifying the Natural Images set.
- The Xception model with None Frozen layers and imported ImageNet weights performs well in general for the task of image classification of the Natural Images dataset.

Transferring Weights

Original Task	Accuracy	Loss
8 classes	82.1%	69.8%
Person / Other	**98%**	7.8%
Cat / Dog	**64.9%**	63.4%

Table 3. Accuracy and loss of source models.

Original Task → New Task	Accuracy	Loss
Person / Other → 8 classes	41.8%	148.9%
Cat / Dog → 8 classes	35.4%	176.9%
8 classes → Person / Other	**97%**	7%
8 classes → Cat / Dog	**77.9%**	49.4%

Table 4. Accuracy and loss of destination models.

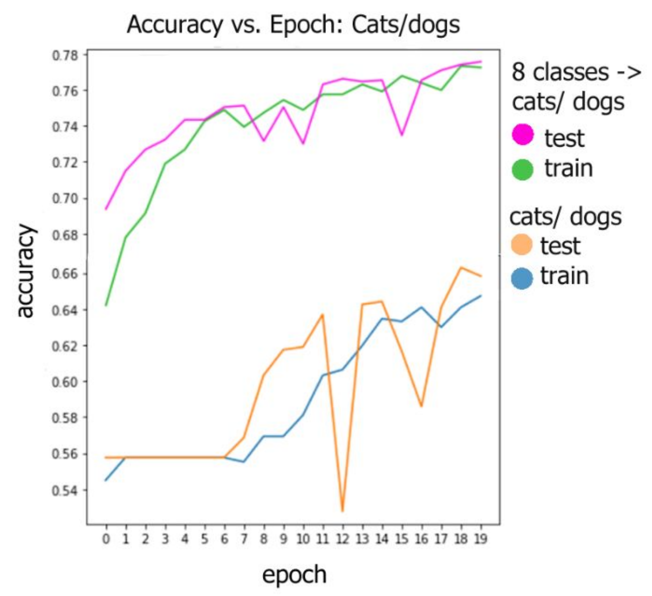


Figure 1. Accuracy of Cats/Dogs model versus 8-classes→Cats/Dogs model on training and testing.

Class	F1-score
airplane	0.11
car	0.29
cat	0.01
dog	0.09
flower	0.21
fruit	0.51
motorbike	0.47
person	0.98

Table 5. F-1 scores for Person/Other→8-classes model.

- Figure 1 shows the 8-classes→Cat/Dog model had an accuracy on epoch 1 that took Cat/Dog model 20 epochs to reach, furthering that transferring weights from a larger, general domain improve training on more specific tasks[5].
- Person/Other→8-classes model’s accuracy was 41%. The F-1 score for *person* in the transferred model was 0.98, higher than the F-1 scores for other classes, as expected, given the original model was trained to identify that class against all others.
- These performances demonstrate transferring weights from a specific task to a more general task is less effective than the reverse.

Conclusions

- Freezing weights is an effective strategy when the weights frozen are part of earlier layers of the model and weights of later sections are allowed to train to be more specific to the classification task.
- Transfer learning is ideal for models trained on simpler tasks if the imported weights were trained on a more complex task. The reverse is less effective.
- Future work would involve replicating these experiments with different model architectures and datasets to see how general these results are.

References

1. Pan, Sinno Jialin and Yang, Qiang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010. doi: 10.1109/ TKDE.2009.191.
2. Yosinski, Jason, Clune, Jeff, Bengio, Yoshua, and Lip- son, Hod. How transferable are features in deep neu- ral networks? CoRR, abs/1411.1792, 2014. URL http://arxiv.org/abs/1411.1792.
3. Chollet, Franc ̧ois. Xception: Deep learning with depthwise separable convolutions. CoRR, abs/1610.02357, 2016. URL http://arxiv.org/abs/1610.02357.
4. Roy, Prasun, Ghosh, Subhankar, Bhattacharya, Saumik, and Pal, Umapada. Effects of degradations on deep neural network architectures. arXiv preprint arXiv:1807.10108, 2018.
5. Donahue, Jeff, Jia, Yangqing, Vinyals, Oriol, Hoffman, Judy, Zhang, Ning, Tzeng, Eric, and Darrell, Trevor. De- caf: A deep convolutional activation feature for generic visual recognition. CoRR, abs/1310.1531, 2013. URL http://arxiv.org/abs/1310.1531.