ARTIFICIAL NEURAL NETWORKS

Meganaa Godhala
Department of Computer and Information Sciences
University of Florida
meganaagodhala@ufl.edu

Abstract: **This project implements different architectures of artificial neural networks across two different datasets: one is the flower species, which has 10 classes, and the other is the bounding box prediction of a car, which predicts the coordinates of the bounding box. The project also discusses the qualitative and quantitative metrics designed and used.**

Dataset-1*:*
A flower species image dataset containing 10 classes The training set contains 1658 RBG images, each of size 300x300x3. Since the kernel was crashing, I had to downscale each image to 75x75x3.

Preprocessing*:*Each image is scaled by 255.0. There is a class imbalance in the data; a few classes have more data than others, which may lead to bias in the model. Hence, I stratified the training set into a training set and a validation set and then trained our model. I also used stratified sampling for training and testing. Whenever we deal with images, it becomes crucial to resize the image. Since the input images are flattened, we have to resize them back to (75, 75, 3) before we input them into the model.

Dataset-2*:*
This dataset consists of training and testing images, few have images of vehicles in them , and few don't have any cars. The training set consists of 559 RBG images of size (380,676,3) each.

Preprocessing*:* Each image is scaled by 255.0. The images with no cars have bounding boxes labeled as (0,0,0,0), and these are excluded during training. I have made custom bounding boxes for testing images that contain images. Eight samples of test data are created. I have split the training set into training and validation sets with 20 percent stratified sampling. Since the input images were flattened, I have to reshape them to their original size, which is (380,676,3), before I train the model.

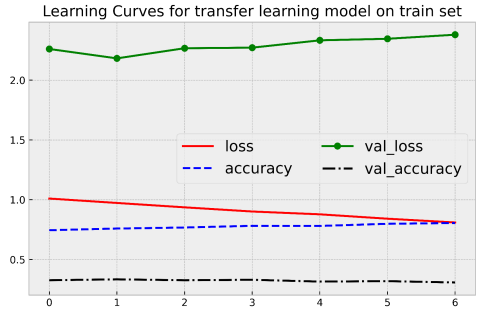
Question-1:
After preprocessing the dataset, our next step would be to come up with an artificial neural network model that would predict the classes with high accuracy. To answer this question, I used three different approaches and evaluated data across three different architectures to find out which would be the most appropriate. The models are referred to as the baseline model, the hyperparameter-tuned model, and the transfer learning model.

Model Architecture:

- The baseline model has 75 neurons in the first hidden layer with 'relu' activation and dropout as 0.2. 50 neurons in the second hidden layer with 'relu' activation and dropout as 0.2. 10 neurons in output with 'softmax' activation since it's a multi-class problem.
- The hyperparameter-tuned model uses a Keras tuner and a search space to find out the best hyperparameters to train the sequential neural network. The hyperparameters tuned include the number of neurons in each hidden layer, the number of hidden layers, the optimizer at each hidden layer, the learning rate, and the dropout rate.
- For the transfer learning model, I have used the Xception model and ImageNet to learn the pre-calculated weights. The performances across three different models in the training set, validation set, and test set are shown below.

	Baseline	**Hyperparameter_tuned**	**Transfer_learning**
train_loss	1.867	1.223	1.0593
train_acc	0.262	0.661	0.75

val_loss	1.987	1.871	2.18
val_acc	0.225	0.302	0.33
test_acc	0.21	0.34	0.39



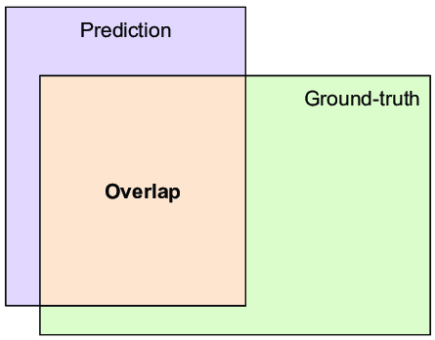
Results: The transfer learning model performed better for this dataset. This is due to using the transfer learning model weights. The hyperparameter-tuned model has better performance than the baseline model but less than the transfer learning model. All the models are overfitting. Since we have already used dropout as a regularizer, we have to use a simpler model or add more data in this case to avoid overfitting.

Question 2:

In answering the first part of the question, where there are no target labels given for the dataset, I have to predict the four points, which are the top left and bottom right coordinates of the bounding box. This becomes a regression task. We have the bounding box coordinates for the training dataset listed in the CSV file. As discussed earlier, there are multiple images in the training set that don't have a car at all. Their labels are 0, and they are not included in the training CSV. Similarly, there are images that have multiple cars in them; hence, each car should have a separate bounding box corresponding to the same image. In this case, I have added them as separate images with a different bounding box for each vehicle. since we don't have a test bounding box CSV file. I have used the software Makesense.AI to create our coordinates for bounding boxes for the test set. I have not included the images with no vehicle since we excluded them during training. For the images with multiple vehicles, I have taken the coordinates of each bounding box separately. I have created a subset of eight such images for testing purposes.

Qualitative metrics: We know that there can be a slight margin of error in predicting the bounding box for the dataset. Hence, there is a need for additional qualitative metrics to define how well our model can predict the bounding box. The metric that will help us understand the overlap better is the intersection over the union. It can be calculated as the ratio of the intersection of true and predicted and the union of true and predicted. This determines how much overlap can be seen between the true and predicted values. The union penalizes the predicted if it's larger than the ground truth values. Here, ground truth represents the bounding box of actual test images that were manually created by us using the Makesense.AI software.

IOU = Area of Intersection/Area of Union



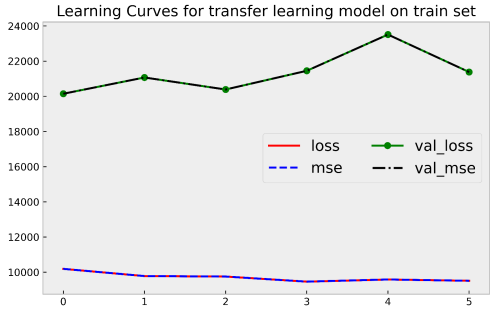
There may be other metrics that can be used as well. But this seemed to be more intuitive for this dataset. The intersection over the Union ratio is also known as the Jacquard index. For the images that don't contain any cars, they are labeled as (0,0,0,0), which means the intersection would be zero, which also means that our index has the value zero for the images that don't have cars in them. Even if there is a car in the image, but the model predicted it so poorly that there is no overlap between the true and test images, the index is zero. For all other cases, the ratio comes to zero. I have calculated the average index over all the images as our qualitative metric. Since there are many images where there is no overlap and no cars in them, this eventually reduces the average of the index. The functions to calculate the IOU index are listed in the training notebook along with the helper functions.

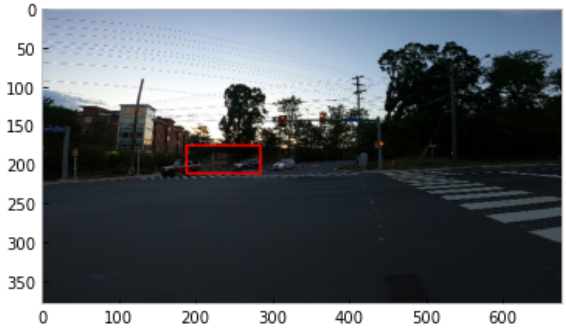
Question 3:
To predict bounding boxes for the car dataset, I have used a transfer learning model with hyperparameter tuning.

Model Architecture:

- The transfer learning base model used is Xception, and weights come from "Imagenet." Xception uses (300, 300, 3) as the required image input size.
- I have set the base model's trainable parameter to true in this case since our dataset is quite small, and I would like to tweak the weights specific to our car dataset for better performance.
- The input for the model will be images of sizes (380,676,3) with a batch size of 2. This is the best batch size which gave a higher performance. Global avg pooling was used on the inputs, and the images were then flattened, while the output is going to be a 4-layer output with a linear activation function, where each layer represents the coordinate point of a predicted bounding box.
- Hyperparameter tuning was done based on the learning rate to determine the best parameters. The best optimizer was Nadam, and the metrics used were "mse." The best learning rate was 0.0001.
- The loss and mse for train, test, and validation sets are discussed below. The bounding box for a sample predicted test image is also attached for reference.

	Model_3
train_mse	8764.45
train_IOU	0.207
val_mse	17613.3
val_IOU	0.155
test_mse	13811
test_IOU	0.0516





Results: Since the task is a regression task, I used mean squared error as the metric for this task. The training, validation, and test metrics, as well as the IOU index, are shown in the table above. The model is underfitted by the shape of the learning curve, and we need more data to converge. The mean squared error and loss for the training and validation sets are quite high. Adding more data is likely to improve the model's performance and result in better weight training. The IOU index gives around 20% overlap in the training set, 15.5% in the validation set, and 5% in the test set. So the model's performance is quite low, both in terms of quantitative and qualitative metrics.

Conclusion:
The learning outcomes of this project include understanding the implementation of artificial neural networks with hyperparameter tuning and different transfer learning approaches using TensorFlow. It helped me understand the importance of qualitative metrics and design my own metric to find the percentage of overlap. Performance in the car dataset is quite low, and we need more data for better training. The species had better performance when used with transfer learning, but this resulted in

overfitting.The learning curves are not convergent with a high mean squared error, loss, and accuracy.

Acknowledgements:
This project was accomplished under the guidance of Prof. Catia Silva; a special thanks to her for clarifying my queries end-to-end throughout the project.

References:

- Class notes, lecture notes
- https://github.com/UF-Applied-ML-Systems-F22/Lectures/blob/main/Lecture%2025/Lecture%2025-in-class%20edits.ipynb
- https://github.com/UF-Applied-ML-Systems-F22/Lectures/blob/main/Lecture%2024/Lecture%2024-in-class%20edits.ipynb
- https://www.tensorflow.org/api_docs