TRANSFER LEARNING

Artificial Intelligence and Machine Learning

Ye Gaung Kyaw

N10923543

Subject: We've processed your Assignment extension request FORM-AEX-299422
Date: Tuesday, 1 November 2022 at 02:32:14 Australian Eastern Standard Time
From: no-reply@qut.edu.au
To: Ye Gaung Kyaw

QUT | HiQ

Hi Ye Gaung,

Thank you for your assignment extension request **(FORM-AEX-299422)**.

We have approved your request and the due date for your assignment **Assignment Two - Transfer Learning**, for unit IFN680 has been extended by 48 hours from the original due date. If your unit outline does not specify that your assignment is eligible for an extension, this confirmation email is not valid and unless you submit by the original due date, the late assessment policy will apply.

You are responsible for ensuring that this assignment is eligible for extension before submitting it after the original due date. Check your unit outline for eligibility.

Be aware that a copy of this email is kept on file. You should not alter this email in any way. Email notifications that have been altered or differ in any way from the original may result in an allegation of student misconduct as set out in the Student Code of Conduct.

Need extra support? You can access free, confidential counselling with qualified professionals. We also offer planning and support if you have a disability, injury or health condition that affects your ability to study. If you are struggling to meet your university commitments, academic support is also available.

Have a question? You can contact us by email or phone. We're also available to help you on campus or via online chat. Visit the HiQ website for our contact details and opening hours.

Email us | +61 7 3138 2000

HiQ is your place to go for **student services, support** and **general enquiries:** qut.to/abouthiq

hi, how can we help y

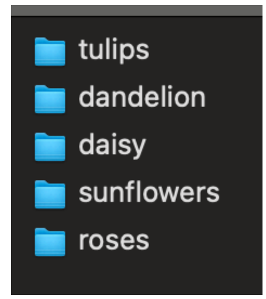
You have received this email because you have submitted an assignment extension request. View our Privacy and Security statement.

Ref ID: 10923543 FORM-AEX-299422

Report

Preparing Dataset

The small flower dataset is provided by the teaching team.



For this assignment, dataset split amount performed is following.

Our dataset has 200 items for each class, altogether 1000 items.

Default batch size “32” is used for each split. (1000 items = 32 batches)

Training	–	75% (768 items or 24 batches)
Validation	–	15% (128 items or 4 batches)
Testing	–	10% (104 items or 4 batches)

Preparing Network

MobileNetV2 is a pretrained network from TensorFlow Keras Module and it can classify nearly 1000 classes. Although it has over 3.5 million params, it is considered to be one of the lightest and smallest pretrained network.

```
=================================================================
Total params: 3,538,984
Trainable params: 3,504,872
Non-trainable params: 34,112
_________________________________________________________________
```

Since we are preparing for a transfer learning, it is important for us to maintain those parameters as old features can be helpful in predicting new classes. For this assignment, the last layer (dense layer with 1000 neurons) from MobileNetV2 is removed. Then, all param values (weights) are made frozen. After those frozen layers, our custom layer (dense layer with 5 neurons) is added to classify our 5 kinds of flowers.

Preprocessing Stages

Dimensions of images from the given dataset are not fixed. Since our network expects to receive fixed width and fixed height to define an input size, we resize all images to 224 x 224 x 3, which is the default input shape of MobileNetV2.

After defining the input shape, pixel values inside each of original image is [0,255] range. MobileNetV2 needs input preprocessing because it expects the input pixel value to be in the range of [-1,1].

Compiling and Training

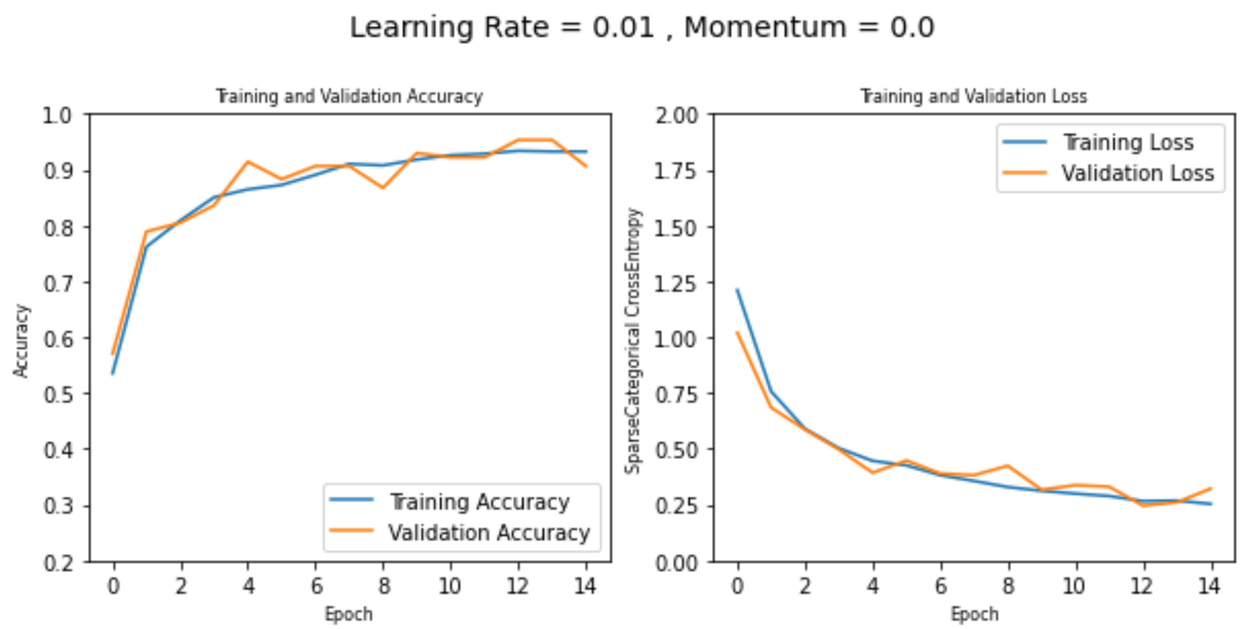
Once the model is ready, it was compiled using Stochastic Gradient Descent (SGD) optimizer with following parameters.

learning rate	-	0.01
momentum	-	0
nesterov	-	False

After defining an optimizer, the model was trained and iterated 15 times by setting Epoch = 15. It used both training dataset and validation dataset.

Exploring history of learning rate 0.01

When trained using learning rate 0.01, the model reaches above 90% accuracy after 10 iterations. And the loss keeps decreasing which we can assume that our model is performing well on training and validation dataset.



Let’s evaluate with our testing set.

Test Dataset Loss: 0.2656 - Test Dataset Accuracy: 0.9423

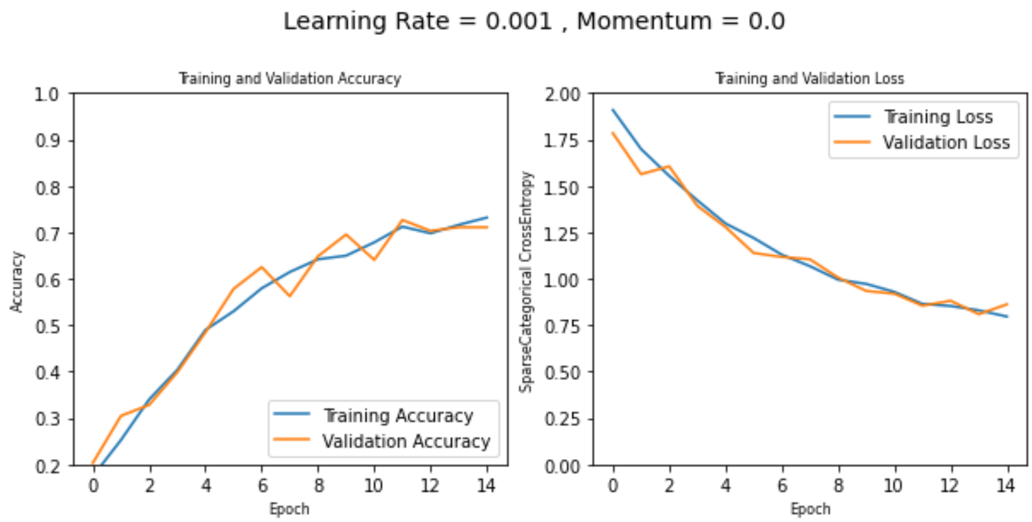
By using learning rate 0.01, our model can perform transfer learning quite correctly, however before a conclusion, experimenting with various learning rates should be done.

Experiment with 3 different orders of magnitude

To test, three values of learning rate are chosen. After we compare them, we will be able to conclude and choose the best learning rate among them.

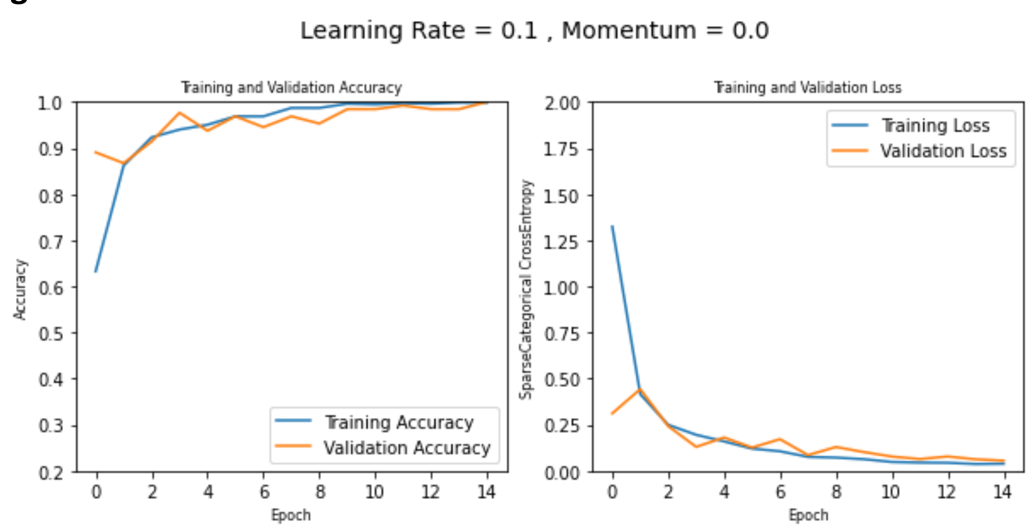
Three values of learning rates 0.001, 0.1 and 1.0 were chosen. Compared to the original value 0.01, I have nominated values both larger and smaller than it.

Learning Rate 0.001



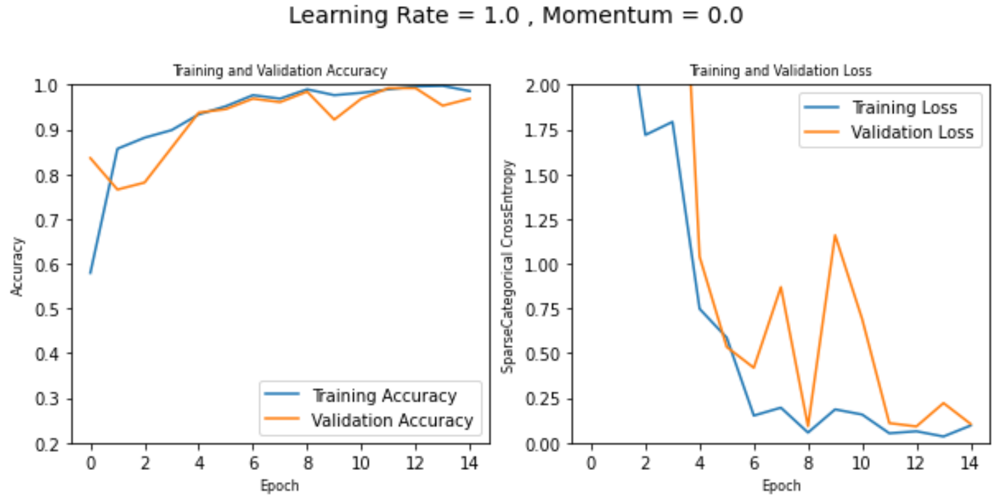
For learning rate of 0.001, it seems to be working quite well, but even at 15th epoch, the model accuracy is still at around 70%. Since learning rate is too less, now it takes a lot of time to reach the convergence.

Learning Rate 0.1



For learning rate of 0.1, both training accuracy and validation accuracy reached above 95% at 9th epoch. Also, both error values reduce as the training goes on. It is the best learning rate that we have encountered till now.

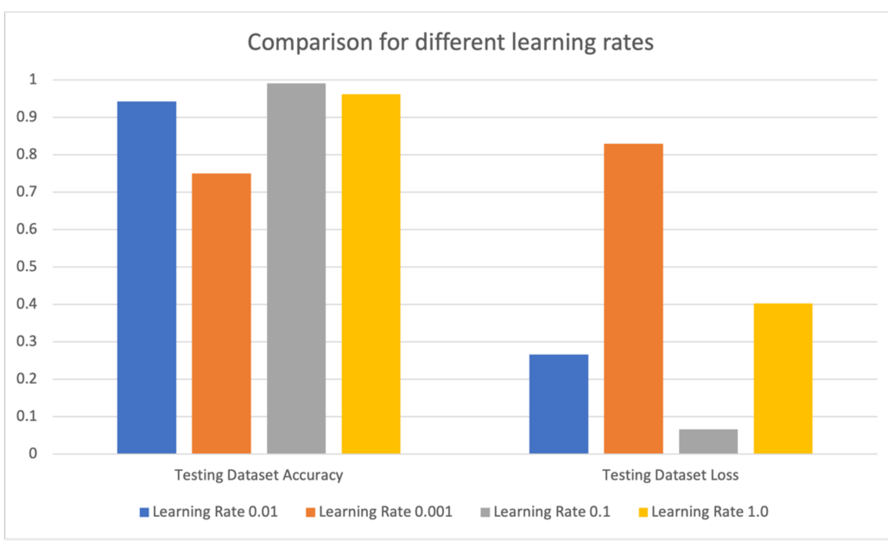
Learning Rate 1.0



For learning rate of 1.0, we can see that there are a lot of oscillations in the graph. Since the learning rate is large now, sometimes we can see that errors does not decrease in training process.

Conclusion for learning rate

After fitting models with different learning rate values, it is the time to evaluate them using testing dataset that we have created. Here, all models are trained using the same optimizer SGD, the same momentum value in the same epoch.

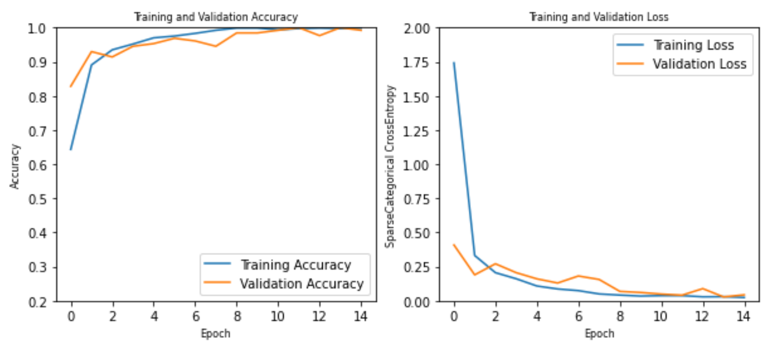


Fitting using Learning rate 0.001 seems promising but it takes too much time to reach convergence. Learning rate 1.0 can reach above 90% of accuracies easily, however the errors do not increase because of the usage of large learning rate. Learning rate 0.1 gives us the best training accuracy and the lowest error in the shortest amount of time for current dataset. We will pick 0.1 as the best learning rate.

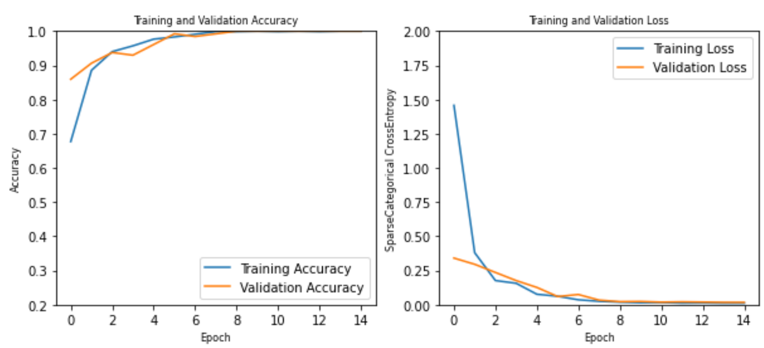
Non-zero momentum with the best learning rate 0.1

Next task is to experiment different non-zero momentum values with the best learning rate. For this task, I have nominated three different momentum values 0.3 , 0.6 and 0.9.

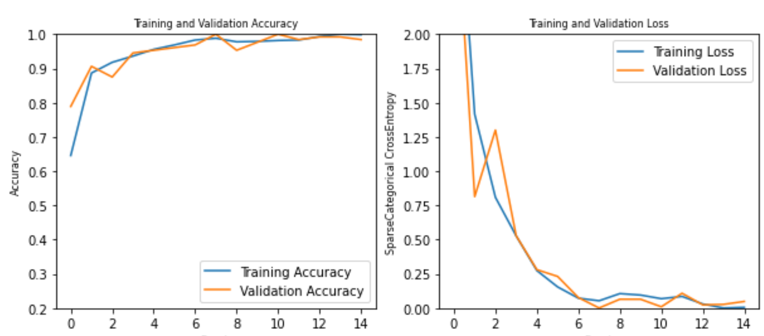
Learning Rate = 0.1 , Momentum = 0.3



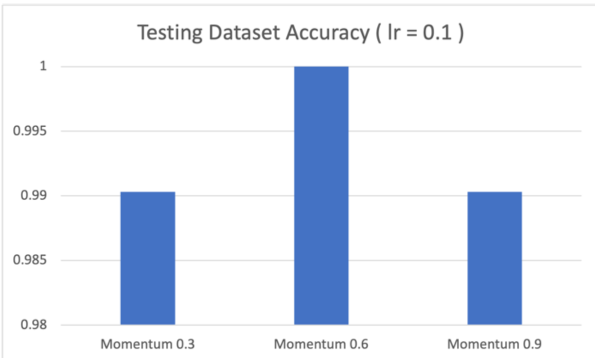
Learning Rate = 0.1 , Momentum = 0.6

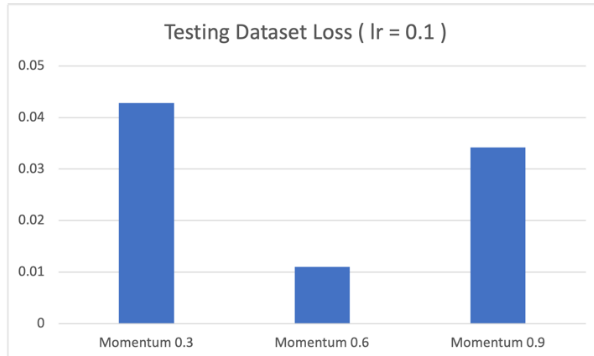


Learning Rate = 0.1 , Momentum = 0.9



Although overall accuracy and loss do not change too much, momentum 0.6 accelerates and improves the accuracy quite faster than compared to other two. Moreover, model compiled with momentum 0.6 is more stable than other models and converge to nearly 100% accuracy.





Also evaluating with testing dataset, momentum 0.6 has the highest accuracy and the lowest error. To conclude, momentum values can help the model to reach global minima and choosing the best momentum value can overcome oscillations in training process and reach to global minima faster.

Transfer Learning (Accelerated Learning)

An accelerated version of transfer learning was performed with different non-zero momentums again. In this case, our new network has only one layer with 5 neurons as the output layer. Compared to normal transfer learning, overfittings can be seen in all values. As the name suggests, the training duration accelerated hugely, but in compensate to the overall performance.

The training error has reached a plateau when using momentum 0.3 and 0.6 within 4th epoch, suggesting that it has reached local or global minima in a short time, although accuracies cannot be improved anymore. On the other hand, large momentum value can oscillate the error value as in momentum 0.9 graph.