

Computational learning assignment 3

Ron Tohar – 209888932

Roi Kant - 212086441

The Notebooks:

We added the notebooks required with the following names:

- **Part1-TwoHidden.ipynb** – our 2 hidden layers extension of the original 1 layer MLP.
- **Part1-Keras.ipynb** – the fully connected ANN implemented in Keras.
- **Part2.ipynb** - the code for part 2 where we applied transfer learning in order to learn the flowers dataset as required.

Part 1:

Accuracy Results:

Accuracy results over the test set:

	Accuracy
One Hidden Layer Implementation from github	94.54
Our Two Hidden Layers Implementation	96.50
Fully Connected ANN Implementation With Keras	91.99

Macro AUC: 0.99527

Firstly, training the model with a train/test split as required (70% - 30%).

Afterwards, applied macro averaging for a one-vs-rest multiclass calculation of the AUC.

Part 2:

In part 2 we were supposed to use to pretrained models to perform transfer learning on the flowers102 dataset.

The models we chose are: **Resnet50** and **MobileNet_v2**.

Preprocessing on the dataset:

On the dataset we preformed preprocessing by:

Normalizing the images we received:

We converted the image data to float and then normalized the image values to 0-1.

Resizing the image size:

we resized the picture to size (224, 224, 3) from (150, 150, 3) to increase the detail of each picture to help with the classification.

Transfer Learning

As we experimented with layers freezing, we got the best results when allowed training for all the layers. By that we performed fine-tuning starting from the original well-trained weights (for image classification) of the pre-trained models we chose.

MobileNet:

the first transfer learning model we used was using MobileNet, the build of the network we used is as follows:

First layer was a pretrained layer brought from the tensor flow hub of ImageNet feature vector for the MobileNet model with fine tuning allowed.

The second layer is a dropout layer with a rate of 0.2

The third layer is a dense layer of the size of the number of classes (102)

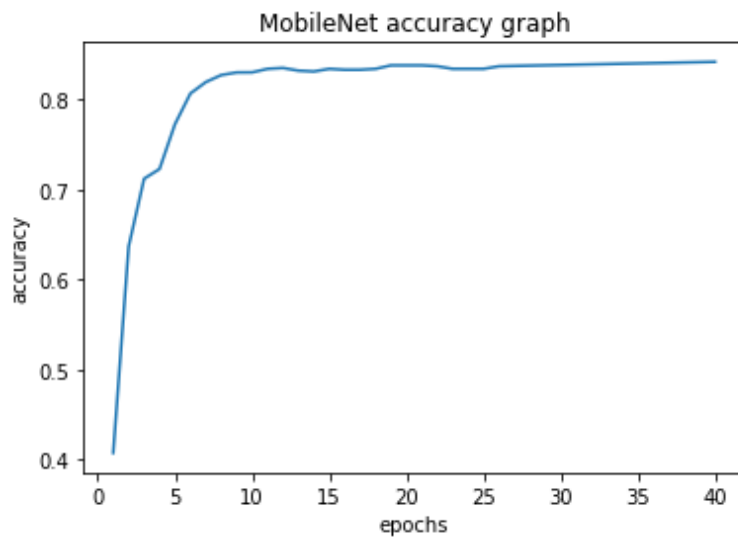
We ran the model with batch size of 16 and 40 epochs.

MobileNet's architecture:

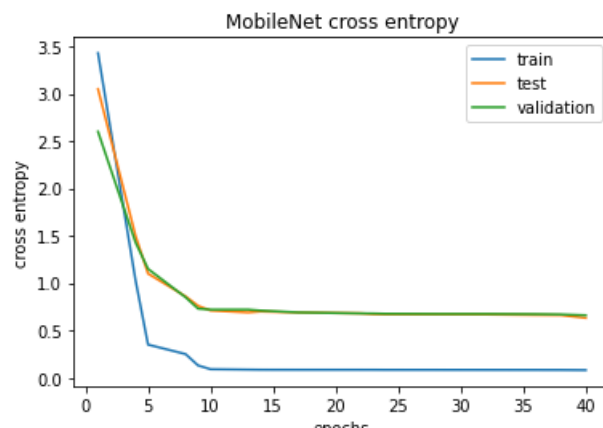
Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32 \text{ dw}$	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64 \text{ dw}$	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5×	Conv dw / s1 $3 \times 3 \times 512 \text{ dw}$	$14 \times 14 \times 512$
	Conv / s1 $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
	Conv dw / s2 $3 \times 3 \times 512 \text{ dw}$	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024 \text{ dw}$	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

The accuracy graph of the run:



The cross entropy graph:



ResNet50:

the second transfer learning model we used was ResNet50, the build of the network we used is as follows:

First layer was a pretrained layer of the ResNet50 model with avg pooling and ImageNet weights. For this layer we allowed fine tuning.

The second layer is a dense layer with 30 neurons and a Relu activation function.

The third layer is a dropout layer with a rate of 0.2

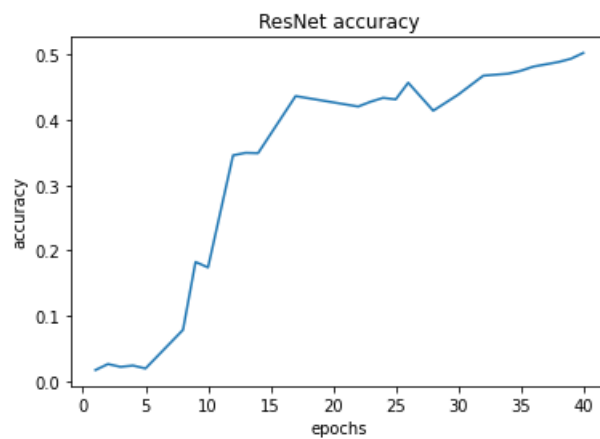
The fourth layer is a dense layer of the size of the number of classes (102)

We ran the model with batch size of 16 and 40 epochs.

The ResNet50 Architecture:

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

The accuracy graph of the run:



The cross entropy graph:

