

## Assignment 2: Experiment 5

Team 10

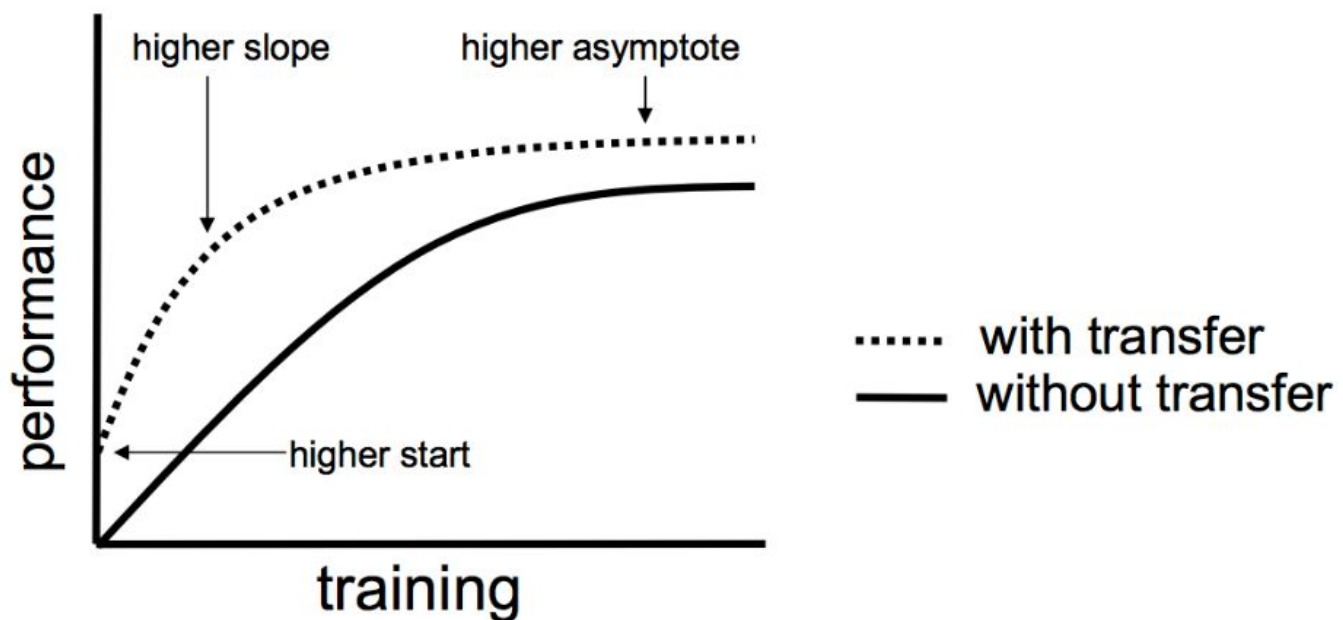
### Experiment 5: Transfer learning

Apply Transfer learning for your team's network to the CIFAR 10 dataset.

Team's network: DenseNet169

[Colab Link](#)

**Introduction:** Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task



#### Steps:

- Import the pre-trained model from keras libraries with input shape as (32,32,3) as the cifar-10 data set has images of that shape.

```
from keras.applications import DenseNet169
##Load model
base_model = DenseNet169(include_top=False, input_shape=(32,32,3), weights='imagenet')
```

- Keep the layer weights same i.e don't train the base model after every iteration.

```
for layer in base_model.layers:
    layer.trainable = False
```

## Assignment 2: Experiment 5

### Team 10

- Build a sequential model adding the imported model at the top, add a new layer at the bottom to classify the categories in the dataset.

```
model = models.Sequential()
model.add(base_model)

##Add new layer

model.add(layers.Flatten())
model.add(layers.Dense(10,activation='softmax'))
```

### Model summary:

Layer (type)	Output Shape	Param #
=====	=====	=====
densenet169 (Model)	(None, 1, 1, 1664)	12642880
flatten_6 (Flatten)	(None, 1664)	0
dense_6 (Dense)	(None, 10)	16650
=====	=====	=====
Total params: 12,659,530		
Trainable params: 16,650		
Non-trainable params: 12,642,880		

### Results:

- The model was compiled with **Adamax** optimizer and learning rate of 0.001
- With the batch size of 500 and 25 epochs, the results were as follows

```
Epoch 20/25
50000/50000 [=====] - 23s 459us/step - loss: 1.6707 - acc: 0.4388 - val_loss: 1.7846 - val_acc: 0.3903
Epoch 21/25
50000/50000 [=====] - 23s 460us/step - loss: 1.6564 - acc: 0.4463 - val_loss: 1.7744 - val_acc: 0.3933
Epoch 22/25
50000/50000 [=====] - 23s 459us/step - loss: 1.6445 - acc: 0.4500 - val_loss: 1.7667 - val_acc: 0.3964
Epoch 23/25
50000/50000 [=====] - 23s 458us/step - loss: 1.6285 - acc: 0.4563 - val_loss: 1.7573 - val_acc: 0.3991
Epoch 24/25
50000/50000 [=====] - 23s 457us/step - loss: 1.6177 - acc: 0.4605 - val_loss: 1.7487 - val_acc: 0.4029
Epoch 25/25
50000/50000 [=====] - 23s 459us/step - loss: 1.6044 - acc: 0.4657 - val_loss: 1.7417 - val_acc: 0.4054
```

### Observation:

- The model was overfitting after 20 epochs

## Assignment 2: Experiment 5

### Team 10

- Training accuracy was 46.5% with a loss of 1.6
- Testing accuracy was 40.5% with a loss of 1.7
- To improve accuracy we need to set the weights of the layers as trainable

```
for layer in base_model.layers:  
    layer.trainable = True
```

- Keeping all other parameters same the results are as follows

Train on 50000 samples, validate on 10000 samples

```
Epoch 1/10  
50000/50000 [=====] - 107s 2ms/step - loss: 1.4922 - acc: 0.4987 - val_loss: 1.0849 - val_acc: 0.6448  
Epoch 2/10  
50000/50000 [=====] - 66s 1ms/step - loss: 0.8134 - acc: 0.7249 - val_loss: 0.8589 - val_acc: 0.7111  
Epoch 3/10  
50000/50000 [=====] - 66s 1ms/step - loss: 0.5861 - acc: 0.8047 - val_loss: 0.7594 - val_acc: 0.7411  
Epoch 4/10  
50000/50000 [=====] - 66s 1ms/step - loss: 0.4323 - acc: 0.8620 - val_loss: 0.7122 - val_acc: 0.7577  
Epoch 5/10  
50000/50000 [=====] - 66s 1ms/step - loss: 0.3112 - acc: 0.9089 - val_loss: 0.7048 - val_acc: 0.7651  
Epoch 6/10  
50000/50000 [=====] - 66s 1ms/step - loss: 0.2148 - acc: 0.9474 - val_loss: 0.6929 - val_acc: 0.7720  
Epoch 7/10  
50000/50000 [=====] - 66s 1ms/step - loss: 0.1428 - acc: 0.9719 - val_loss: 0.7009 - val_acc: 0.7753  
Epoch 8/10  
50000/50000 [=====] - 66s 1ms/step - loss: 0.0933 - acc: 0.9869 - val_loss: 0.7169 - val_acc: 0.7768  
Epoch 9/10  
50000/50000 [=====] - 66s 1ms/step - loss: 0.0603 - acc: 0.9940 - val_loss: 0.7415 - val_acc: 0.7757  
Epoch 10/10  
50000/50000 [=====] - 66s 1ms/step - loss: 0.0423 - acc: 0.9968 - val_loss: 0.7768 - val_acc: 0.7779
```

- The accuracy of the model was high at the initial stage itself, but the model is getting overfitted from the 3rd epoch.