Assignment 1 Part A

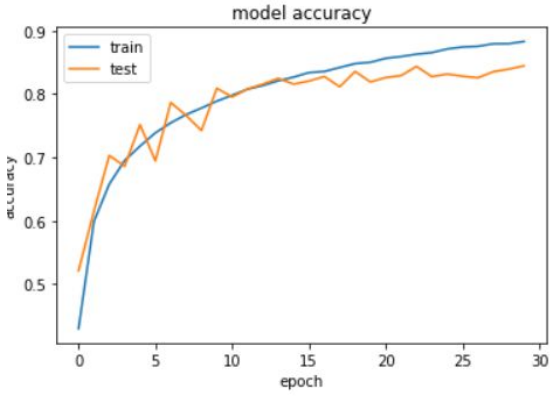
Discuss and use different types of layers that you could use for the problem (See Dense, Dropout, Activation, Flatten)

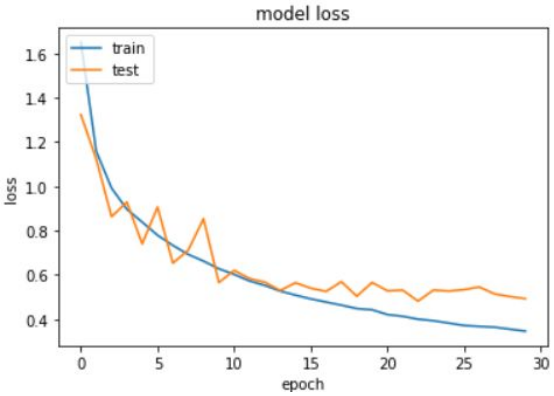
- We choose the best performance model as follow:

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 32, 32)	896
activation_1 (Activation)	(None, 32, 32, 32)	0
batch_normalization_1 (Batch	(None, 32, 32, 32)	128
conv2d_2 (Conv2D)	(None, 30, 30, 32)	9248
activation_2 (Activation)	(None, 30, 30, 32)	0
batch_normalization_2 (Batch	(None, 30, 30, 32)	128
max_pooling2d_1 (MaxPooling2	(None, 15, 15, 32)	0
dropout_1 (Dropout)	(None, 15, 15, 32)	0
conv2d_3 (Conv2D)	(None, 15, 15, 64)	18496
activation_3 (Activation)	(None, 15, 15, 64)	0
batch_normalization_3 (Batch	(None, 15, 15, 64)	256
conv2d_4 (Conv2D)	(None, 13, 13, 64)	36928
activation_4 (Activation)	(None, 13, 13, 64)	0
batch_normalization_4 (Batch	(None, 13, 13, 64)	256
max_pooling2d_2 (MaxPooling2	(None, 6, 6, 64)	0
dropout_2 (Dropout)	(None, 6, 6, 64)	0
conv2d_5 (Conv2D)	(None, 6, 6, 128)	73856
activation_5 (Activation)	(None, 6, 6, 128)	0
batch_normalization_5 (Batch	(None, 6, 6, 128)	512
conv2d_6 (Conv2D)	(None, 4, 4, 128)	147584
activation_6 (Activation)	(None, 4, 4, 128)	0
batch_normalization_6 (Batch	(None, 4, 4, 128)	512
max_pooling2d_3 (MaxPooling2	(None, 2, 2, 128)	0
dropout_3 (Dropout)	(None, 2, 2, 128)	0
flatten_1 (Flatten)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262656
activation_7 (Activation)	(None, 512)	0
dropout_4 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130
activation_8 (Activation)	(None, 10)	0

Total params: 556,586
Trainable params: 555,690
Non-trainable params: 896

Test loss: 0.492332640553
Test accuracy: 0.8443





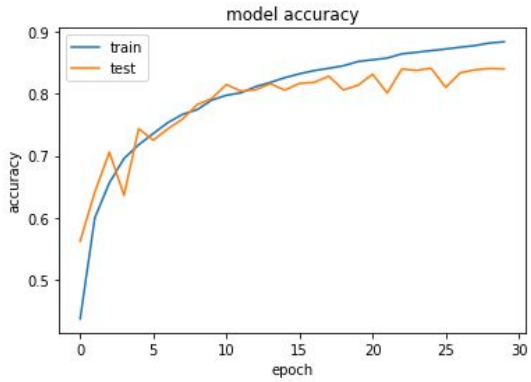
- Changing Flatten() to Reshape()
  Flatten() function flattens the input and does not affect the batch size. It can be replaced by Reshape() function with certain input shape.
  model.add(Flatten())

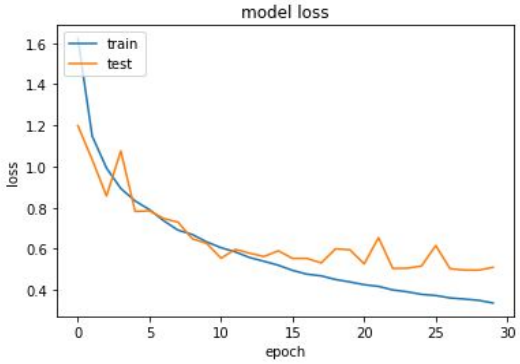
flatten_1 (Flatten)	(None, 512)	0

model.add(Reshape((512,)))

reshape_1 (Reshape)	(None, 512)	0

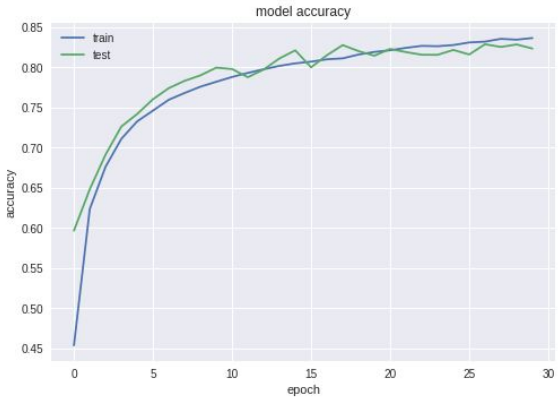
Test loss: 0.510427865696
Test accuracy: 0.8399

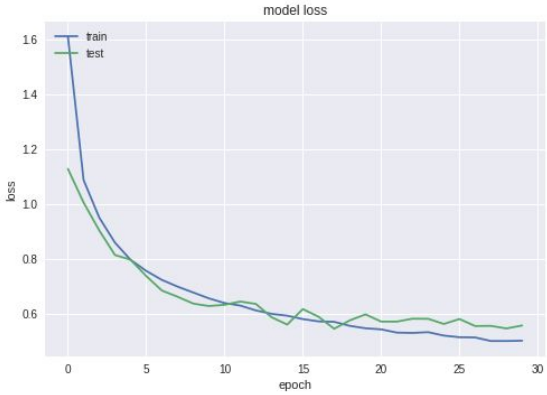




Comparing with the best model above, the accuracy drops and the loss increases which means that Reshape() function is not as good as Flatten() function.

- Changing Dense() to GlobalAveragePooling2D()
  Replace Flatten layer and Dense layer with GlobalAveragePooling2D. It can bring less parameters which is helpful to restrict overfitting.





- Changing Dropout() to SpatialDropout2D()
  It slightly reduced overfitting, even after a batch regularization. A correlation between pixels in early layers can be inferred. Because it is dropping neurons more ‘efficiently’, the performance may not be increased as good as the plain dropout layer until a few more epochs later.