CIFAR 10

Model.compile(loss=’categorical\_crossentropy’,optimizer=’adam’,metrics=[‘accuracy’])

Hist=model.fit(X\_train,Y\_train,batch\_size=128,nb\_epoch=10,verbose=1,validation\_data=(X\_test,Y\_test)

* Changing the epochs from 10 to 15

Epoch 10/20

- loss: 1.2903 - acc: 0.5382 - val\_loss: 1.3977 - val\_acc: 0.5018

Epoch 15/20

- loss: 1.1325 - acc: 0.5944 - val\_loss: 1.3780 - val\_acc: 0.5223

***Here we observe that when we increase the epochs from 10 to 15 more computation happens and because of more computation we get a better accuracy.***

* Changing the epochs from 10 to 20

Epoch 20/20

- loss: 0.9520 - acc: 0.6569 - val\_loss: 1.4917 - val\_acc: 0.5098

***We see further the accuracy increases when epochs goes to 20***

* Changing batch size from 128 to 64

Epoch 10/10

- loss: 1.3053 - acc: 0.5292 - val\_loss: 1.4199 - val\_acc: 0.5028

* Changing batch size from 128 to 256

Epoch 10/10

558us/step - loss: 1.2991 - acc: 0.5373 - val\_loss: 1.4318 - val\_acc: 0.4937

* Changing optimizer from adam to rmsprop

Epoch 10/10

- loss: 1.4248 - acc: 0.4929 - val\_loss: 1.4737 - val\_acc: 0.4861

IRIS

Model.compile(optimizer=’adam’,loss=’categorical\_crossentropy’,metrics=[‘accuracy’])

Model.fit(train\_x,train\_y,verbose=2,batch\_size=5,epochs=200)

* Changing the epochs from 200 to 100

Epoch 100/200

- 0s - loss: 0.0791 - acc: 0.9865

Epoch 200/200

- 0s - loss: 0.0566 - acc: 0.9865

***Here even though the epoch is changed there is no change in the accuracy that we get with both epochs***

* Changing the epochs from 200 to 300

Epoch 300/300

- 0s - loss: 0.0583 - acc: 0.9797

***Here the accuracy decreases***

* Changing batch size from 5 to 10

Epoch 200/200

- 0s - loss: 0.0655 - acc: 0.9730

* Changing optimizer from adam to rmsprop

Epoch 200/200

- 0s - loss: 0.0658 - acc: 0.9797

Final test set loss: 0.036648

Final test set accuracy: 1.000000

***Rmsprop gives a better result than adam***