Dans ce travail j’ai utilisé la bibliothèque Tensorflow,

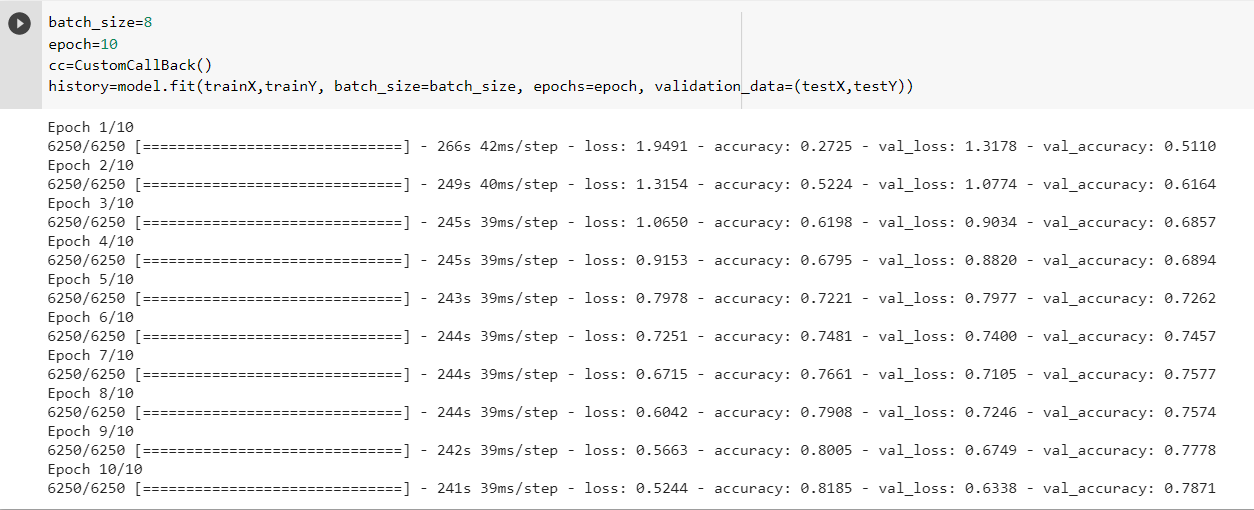
Question 1 :

Après avoir tester plusieurs fois le code en changeant la valeur de l’epoch je n’ai pas trouvé de meilleur résultats par exemple comme mentionnée dans le tableau suivant :

|  |  |  |
| --- | --- | --- |
| Epoch 50 | train loss = 2.2830 | accuracy = 0.102 |
| Epoch 30 | train loss = 2.3040 | accuracy = 0.1 |
| Epoch 15 | train loss = 1.0145 | accuracy = 0.5967 |
| Epoch 6 | train loss = 1.0768 | accuracy = 0.5684 |

Mais après avoir changé les valeurs de batch size et learning rate j’ai eu une amélioration de la précision et une réduction de loss

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epoch | Batch\_size | LR | Train\_Loss | Accuracy |
| 15 | 8 | 0.001 | 0.5824 | 0.6604 |
| 20 | 8 | 0.001 | 0.3099 | 0.6195 |
| 20 | 20 | 0.001 | 0.5546 | 0.8164 |
| 10 | 8 | 0.001 | 0.5244 | 0.8185 |



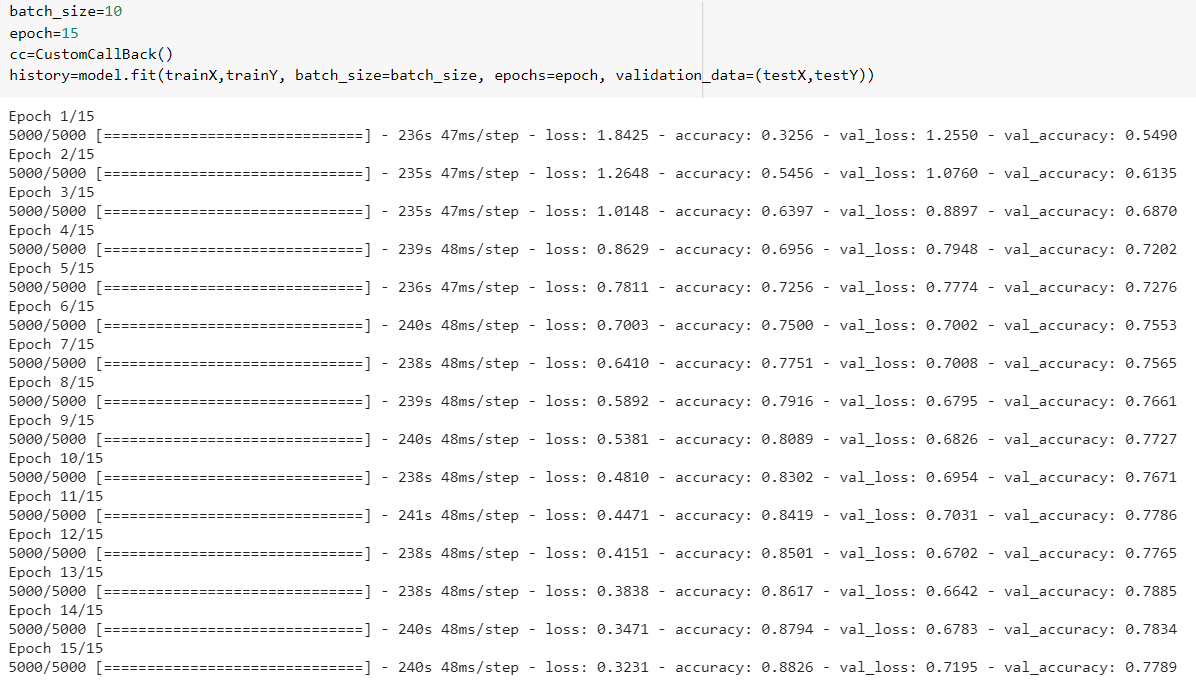
Après avoir essayé de nombreux valeurs de paramètre epoch j’ai trouvé que 10 est le nombre d’epoch optimal.

Question 2 :

Pour éviter le sur apprentissage j’ai utilisé la fonction Dropout qui permet d'étudier les variations de la façon dont l'abandon est appliqué au modèle.

Une variante qui pourrait être intéressante est d'augmenter le taux d'abandon

Elle est efficace car elle oblige les couches profondes du modèle à régulariser plus que les couches plus proches de l'entrée.



On peut voir que le dropout a performé le modèle efficacement en utilisant les mêmes paramètres d’epoch, batch size et learning rate=0.001.

Question 3 :

J’ai modifier le model comme suit :

Convolution=> Convolution =>Pooling =>dropout=> Convolution=> Convolution =>Pooling =>dropout=> flatten=> dense=>dropout=>dense

