

❖ در این سوال با توجه به اینکه ورودی یک عکس است و خروجی 5 مختصات، بنابراین روش دسته‌بندی نیست.

✓ تابع Sigmoid برای Activation Function می‌تواند عملکرد خوبی داشته باشد چرا که هر نقطه را مستقل بررسی می‌کند و از طرفی MSE نیز اگر فاصله نقطه انتخاب شده تا Target کم باشد، مقدار کمی این مختصات را جابه‌جا می‌کند و اگر اختلاف زیاد باشد، باز هم درست Adjust خواهد شد. بنابراین با MSE و Sigmoid گزینه مناسبی خواهد بود (همچنین با توجه به مدل‌های مختلف دیده شده در اینترنت این ایده به ذهن رسید). برای تعداد نوروں نیز 5 تا برای لایه آخر در نظر بگیریم که هر کدام نشان دهنده یک Key Point باشد.

✓ کد زده شده برای Activation Function از تابع Sigmoid استفاده کرده و برای Loss Function از MSE که کمی بهبود یافته (برای این مسئله).

❖ نکته اول، تعداد نوروں لایه آخر را برابر 4 می‌گذاریم، با توجه به نوع مسئله که Multi-Class است، از طرفی تعداد نوروں در لایه‌های میانی نیز مناسب است (به تدریج کاهش یافته، چرا که تغییر ناگهانی دقت را کاهش می‌دهد). برای انتخاب Activation و Loss، باید تناسب آن‌ها را در نظر بگیریم. برای Activation تابع Softmax و برای Loss تابع Categoricalcrossentropy را در نظر می‌گیریم که نتیجه زیر را دارد:

```
Epoch 300/300
50/50 [=====] - 0s 5ms/step - loss: 0.4953 - accuracy: 0.7837 - val_loss: 0.5211 - val_accuracy: 0.7750
```

چون 4 کلاس داریم و باید بین خروجی هر نوروں مقایسه کنیم، Activation Function = Softmax مناسب‌تر است و مستقل کار نمی‌کند. برای انتخاب Loss Function نیز MSE نتیجه خوبی برای این سوال به همراه Softmax نداشت (سریع به مقادیر پایین دقت همگرا می‌شود). بنابراین همان Softmax و Categoricalcrossentropy به همراه 4 نوروں در لایه آخر انتخاب می‌شود. البته ترکیب Sigmoid و Categoricalcrossentropy نیز با توجه به فرمول آن دو نتیجه خوبی خواهد داد، خروجی مستقل هر نوروں در فرمول می‌رود و تاثیر مثبتی دارد.

نکته: Optimizer را Adam انتخاب کردم، به علت عملکرد کلی بهتر.

❖ در این سوال 5 حالت مختلف برای Loss Function و Activation Function امتحان شد که نتایج

Accuracy و Loss برای دو حالت Train و Test به شکل زیر است (با استفاده از TensorBoard):
نکته قابل توجه: باید Accuracy را مقایسه کنیم، چرا که مقدار Loss معیار دقیقی برای بررسی عملکرد Model نیست، برای مثال MSE به علت توان 2، مقدار بسیار کمی برای Loss دارد، اما لزوماً عملکرد آن خوب نیست، و برعکس.

- ✓ Sigmoid & Binarycrossentropy:

```
Epoch 20/20  
18/18 [=====] - 0s 8ms/step - loss: 0.0274 - accuracy: 0.8115 - val_loss: 0.0348 - val_accuracy: 0.7520  
_*-*_*-*_*-*_*-*_*-*_*-_  
71/71 [=====] - 0s 1ms/step - loss: 0.0348 - accuracy: 0.7520  
Test Loss: 0.03484814986586571  
Test Accuracy: 0.75200355052948
```

- ✓ Softmax & Categoricalcrossentropy:

```
Epoch 20/20  
18/18 [=====] - 0s 7ms/step - loss: 0.3136 - accuracy: 0.9263 - val_loss: 0.9923 - val_accuracy: 0.7814  
_*_**_*_*_*_*_*_*_*_*_*_*_*_  
71/71 [=====] - 0s 1ms/step - loss: 0.9923 - accuracy: 0.7814  
Test Loss: 0.9923385977745056  
Test Accuracy: 0.7813891172409058
```

- ✓ Sigmoid & MSE:

```
Epoch 20/20
18/18 [=====] - 0s 8ms/step - loss: 0.0097 - accuracy: 0.6030 - val_loss: 0.0104 - val_accuracy: 0.5913
_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*
71/71 [=====] - 0s 1ms/step - loss: 0.0104 - accuracy: 0.5913
Test Loss: 0.010424983687698841
Test Accuracy: 0.5912733674049377
```

- ✓ Softmax & MSE:

```
Epoch 20/20  
18/18 [=====] - 0s 7ms/step - loss: 0.0042 - accuracy: 0.8701 - val_loss: 0.0075 - val_accuracy: 0.7703  
_**_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_  
71/71 [=====] - 0s 1ms/step - loss: 0.0075 - accuracy: 0.7703  
Test Loss: 0.007509772665798664  
Test Accuracy: 0.7702582478523254
```

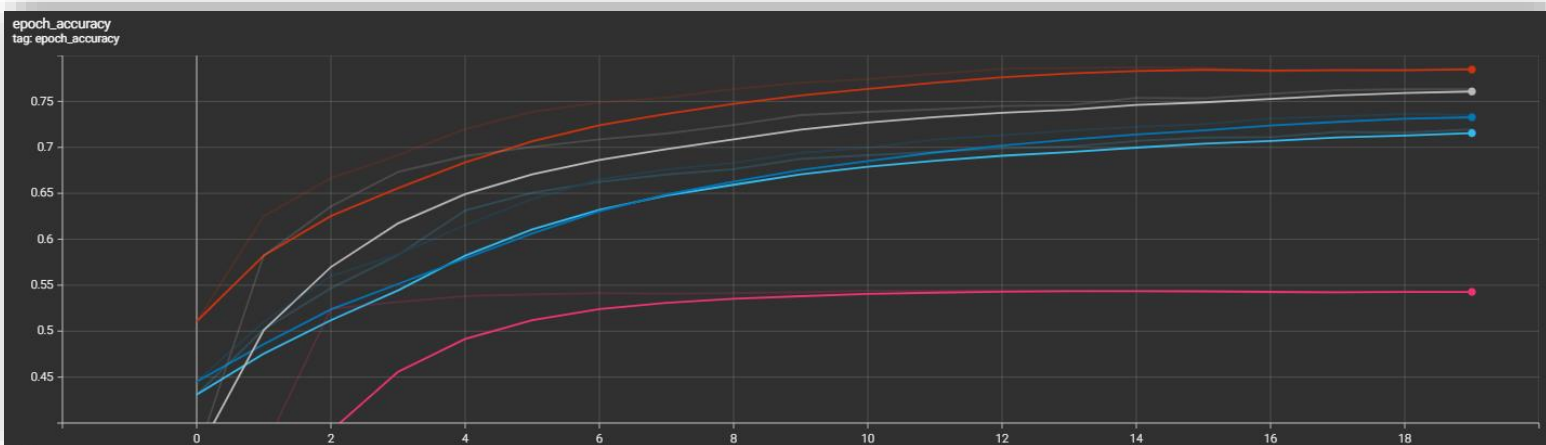
- ✓ Linear & MSE:

```
Epoch 20/20  
18/18 [=====] - 0s 7ms/step - loss: 0.0061 - accuracy: 0.8052 - val_loss: 0.0089 - val_accuracy: 0.7195  
_._*_._*_._*_._*_._*_._*_._*_._*_  
71/71 [=====] - 0s 1ms/step - loss: 0.0089 - accuracy: 0.7195  
Test Loss: 0.00886903703212738  
Test Accuracy: 0.7195013165473938
```

با مقایسه مقادیر بالا متوجه می‌شویم که استفاده از Softmax همراه Categoricalcrossentropy بهترین نتیجه را در Accuracy داشته است.

بدترین عملکرد نیز برای Sigmoid همراه MSE است.

به نمودار Accuracy برای Test Data دقت کنید:



دلیل: تابع Softmax برای مسائلی که چند کلاس دارند بسیار مناسب است، چراکه نتایج حاصل از همه نوروں ها را مقایسه می‌کند و مقیاس دقیق‌تری برای انتخاب دارد (احتمال)، از طرفی هم Sigmoid کاملاً مستقل از هم است، برای همین دقیق نیست. انتخاب Loss Function هم چون چند کلاس داریم Categoricalcrossentropy می‌تواند عملکرد خوبی داشته باشد، اما تابع MSE به علت توان 2 و تغییرات کمی که دارد (برای محاسبه گرادیان) احتمال داشتن Performance پایین بیشتری دارد. برای همین بهترین حالت Softmax, Categoricalcrossentropy و بدترین حالت Sigmoid, MSE است. به ویژه که مقادیر خروجی Sigmoid کوچک است و MSE به توان 2 می‌رساند.

❖ منابع:

- ✓ <https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3>
- ✓ <https://analyticsindiamag.com/tutorial-on-keras-tokenizer-for-text-classification-in-nlp/>
- ✓ https://www.tensorflow.org/tensorboard/get_started