

امیرمهدی ابوطالبی

امیررحیمی

علی شیخ عطاری

نکته : فایل ها به دلیل بهم ریخته بودن باعث شد تا یک دیتا ست دیگری را تهیه کنم از همون داده ها که با خود پروژه بار

گذاری می شود به نام DataSet

1 – همان طور که در صورت سوال خواسته شده فایل ها به صورت خواسته شد ابتدا داده ها را به صورت 80 به 20 جدا کردیم با استفاده از کتاب خانه splitfolders که نصب و ایمپورت در برنامه قابل انجام است

2 – با توجه به استفاده شدن از کتاب خانه sklearn و الگوریتم mlp پس این برنامه به خودی خود با forward و backward propagation استفاده می کند به صورتی که در ابتدا y بار را پیدا کرده و سپس به عقب بر میگردد با مشتق گرفتن

3 – accuracy این مدل با 2 لایه که یک لایه هیدن و یک لایه خروجی است که لایه هیدن دارای 100 نود است 0.90 است ولی با بالا بردن یا پایین آوردن نود های لایه یا اضافه کردن یا کم کردن لایه های هیدن این عدد کم و زیاد ممکن است بشود

4 – همان گونه که در کد مشاهده می شود به ترتیب برای هر کدام از تعداد نود های خاسته شده

```
with 2 element in hidden layer accuracy_score is : 0.496
with 4 element in hidden layer accuracy_score is : 0.496
with 10 element in hidden layer accuracy_score is : 0.792
with 50 element in hidden layer accuracy_score is : 0.848
```

ما این مقادیر را داریم که نشان می دهد با افزایش نود ها مدل بهتر شده اما توجه شود که با افزایش بیش از اندازه ممکنه به overfitting برخوریم دلیل این که با افزایش نود دقت بالا میره به این دلیل است که هر کدام از نود ها کار محاسبات رو انجام می دهند و توی این بخش مانند cnn ما ابتدا edge detect می کنیم پس هرچه محاسبات ما بیشتر بشه دقت بالا تر می رود

5 – همان جور که در کد در بخش آخر آورده شده است با افزایش لایه ها به مراتب بهتر می شود accuracy

3 Layer (2 hidden ,1 output)

```
In [494]: Newmlp = MLPClassifier(hidden_layer_sizes = (100,100,),max_iter=500, activation='relu',learning_rate="adaptive",solver="adam")
Newmlp.fit(train_x_flatten,train_y)
Newpred = Newmlp.predict(test_x_flatten)
print(accuracy_score(test_y, Newpred))
0.912
```

4 Layer (3 hidden ,1 output)

```
In [496]: Newmlp = MLPClassifier(hidden_layer_sizes = (128,64,32,),max_iter=500, activation='relu',learning_rate="adaptive",solver="adam")
Newmlp.fit(train_x_flatten,train_y)
Newpred = Newmlp.predict(test_x_flatten)
print(accuracy_score(test_y, Newpred))
0.92
```

5 Layer (4 hidden ,1 output)

```
In [497]: Newmlp = MLPClassifier(hidden_layer_sizes = (256,128,64,32,),max_iter=500, activation='relu',learning_rate="adaptive",solver="adam")
Newmlp.fit(train_x_flatten,train_y)
Newpred = Newmlp.predict(test_x_flatten)
print(accuracy_score(test_y, Newpred))
0.944
```