

به نام خدا

پارسا پالیزیان-400100865

پروژه هوش مصنوعی و محاسبات زیستی

استاد : دکتر سپیده حاجی پور

در بخش اول ما توابع مورد نیاز برای محاسبه ویژگی ها رو نوشتیم و با محاسبه معیار فیشر یک بعدی ویژگی های محاسبه شده را براساس میزان اهمیت شان مرتب کردیم. 50 ویژگی برتر که در مرحله بعد برای طراحی شبکه `mlp` , `rbf` استفاده کردیم به شرح زیر بود:

```
theta channel 57
theta channel 59
theta channel 50
bin 3 channel 11
medfreq channel 43
gamma channel 43
bin 4 channel 20
bin 6 channel 11
meanfreq channel 23
bin 4 channel 49
maxfreq channel 24
bin 3 channel 9
bin 7 channel 20
maxfreq channel 43
bin 6 channel 20
theta channel 52
maxfreq channel 23
gamma channel 23
bin 7 channel 49
delta channel 43
delta channel 51
bin 2 channel 11
bin 3 channel 49
medfreq channel 23
theta channel 58
medfreq channel 56
bin 4 channel 9
gamma channel 54
medfreq channel 33
medfreq channel 59
medfreq channel 54
bin 7 channel 9
bin 4 channel 11
meanfreq channel 33
mid range beta channel 7
meanfreq channel 59
```

bin 6 channel 9
high range beta channel 43
maxfreq channel 54
maxfreq channel 51
bin 5 channel 25
medfreq channel 47
gamma channel 33
bin 6 channel 18
bin 8 channel 20
var channel 43
bin 7 channel 11
high range beta channel 47
meanfreq channel 56
gamma channel 36

برای طراحی و آموزش شبکه از کتابخانه tensorflow پایتون استفاده کردیم و پارامترهای ما به صورت زیر بود:

MLP:

```
model_mlp_sigmoid = Sequential([
    layers.InputLayer(input_shape=(input_shape,)),
    layers.Dense(units = 32 , activation = "relu"),
    layers.Dense(units = 64 , activation = "relu"),
    layers.Dense(units = 128 , activation = "relu"),
    layers.Dense(units = 64 , activation = "relu"),
    layers.Dense(units = 32 , activation = "relu"),
    layers.Dense(units = 1 , activation = "sigmoid"),
])
model_mlp_softmax = Sequential([
    layers.InputLayer(input_shape=(input_shape,)),
    layers.Dense(units = 32 , activation = "relu"),
    layers.Dense(units = 64 , activation = "relu"),
    layers.Dense(units = 128 , activation = "relu"),
    layers.Dense(units = 64 , activation = "relu"),
    layers.Dense(units = 32 , activation = "relu"),
    layers.Dense(units = 2 , activation = "softmax"),
])
model_mlp_sigmoid.compile(optimizer = "adam" , loss = "binary_crossentropy" , metrics = ["accuracy", "binary_accuracy"])
model_mlp_softmax.compile(optimizer = "adam" , loss = "categorical_crossentropy" , metrics = ["accuracy", "categorical_accuracy"])
```

RBF:

```
model_rbf_sigmoid = keras.Sequential([
    layers.InputLayer(input_shape=(input_shape,)),
    layers.experimental.RandomFourierFeatures(output_dim=550,scale=10.,kernel_initializer='gaussian'),
    layers.Dense(units=1, activation='sigmoid'),
])
model_rbf_softmax = keras.Sequential([
    layers.InputLayer(input_shape=(input_shape,)),
    layers.experimental.RandomFourierFeatures(output_dim=550,scale=10.,kernel_initializer='gaussian'),
    layers.Dense(units=2, activation='softmax'),
])
model_rbf_sigmoid.compile(optimizer='adam',loss='binary_crossentropy',metrics = ["accuracy", "binary_accuracy"])
model_rbf_softmax.compile(optimizer='adam',loss='categorical_crossentropy',metrics = ["accuracy", "categorical_accuracy"])
```

نتایج به شرح زیر بود:

```
##### Average Accuracy #####
-----SIGMOID MLP
0.9813636422157288
-----SOFTMAX MLP
0.9877272725105286
-----SIGMOID RBF (Builtin Class)
0.7081818222999573
-----SOFTMAX RBF (Builtin Class)
0.7140909075737
-----SIGMOID RBF (Custom Class)
0.7245454668998719
```

در مرحله بعد با استفاده از الگوریتم تکاملی pso ویژگی ها را استخراج کرده و شبکه را بر اساس آن طراحی کردیم و بهترین فیچر ها موجب خروجی زیر شدند:

```
##### Average Accuracy #####
-----SIGMOID MLP
0.9436363697052002
```

برای الگوریتم تکاملی از کتابخانه ی pyswarm پایتون استفاده کردیم.

برای سادگی اجرا کد ها را در سه فایل جدا نوشته ایم:

Feature Exraction:

Hosh_Project_feature_extract

MLP-RBF:

Hosh_Project_mlp_rbf

PSO:

Hosh_Project_pso