

In [7]:

```

1 import numpy
2 import pandas
3 from keras.models import Sequential
4 from keras.layers import Dense
5 from keras.layers import Dropout
6 from keras.wrappers.scikit_learn import KerasClassifier
7 from keras.constraints import maxnorm
8 from keras.optimizers import SGD
9 from sklearn.model_selection import cross_val_score
10 from sklearn.preprocessing import LabelEncoder
11 from sklearn.model_selection import StratifiedKFold
12 from sklearn.preprocessing import StandardScaler
13 from sklearn.pipeline import Pipeline

```

In [8]:

```

1 # وارد کردن کتابخانه
2 import pandas as pd
3 import numpy as np
4
5 # وارد کردن دیتاست دیابت
6 dataset = pd.read_excel("C:\\Users\\ShahinN\\Desktop\\pima-indians-diabetes.
7 dataset.head()

```

Out[8]:

	0	1	2	3	4	5	6	7	8
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [9]:

```

1 # بخش بندی دیتاست
2 X = dataset.values[:,0:8]
3 Y = dataset.values[:,8]

```

In [10]:

```

1 # ساخت مدل
2 model = Sequential()
3 model.add(Dense(12, input_dim=8, kernel_initializer= 'uniform' , activation=
4 model.add(Dense(8, kernel_initializer= 'uniform' , activation= 'relu' ))
5 model.add(Dense(1, kernel_initializer= 'uniform' , activation= 'sigmoid' ))

```

In [11]:

```

1 sgd = SGD(lr=0.01, momentum=0.8, decay=0.0, nesterov=False)
2 model.compile(loss= 'binary_crossentropy' , optimizer=sgd, metrics=[ 'accura

```

In [12]: 1 model.fit(X, Y, epochs=150, batch\_size=10)

```
Epoch 1/150
768/768 [=====] - 2s 2ms/step - loss: 0.6689 - accuracy: 0.6497
Epoch 2/150
768/768 [=====] - 0s 616us/step - loss: 0.6543 - accuracy: 0.6510
Epoch 3/150
768/768 [=====] - 0s 537us/step - loss: 0.6431 - accuracy: 0.6458
Epoch 4/150
768/768 [=====] - 0s 570us/step - loss: 0.6358 - accuracy: 0.6602
Epoch 5/150
768/768 [=====] - 0s 642us/step - loss: 0.6292 - accuracy: 0.6589
Epoch 6/150
768/768 [=====] - 0s 575us/step - loss: 0.6255 - accuracy: 0.6628
Epoch 7/150
768/768 [=====] - 0s 644us/step - loss: 0.6217 - accuracy: 0.6617
```

In [13]: 1 # ارزیابی مدل  
2 scores = model.evaluate(X, Y)  
3 print("%s: %.2f%%" % (model.metrics\_names[1], scores[1]\*100))

```
768/768 [=====] - 1s 1ms/step
accuracy: 65.10%
```

روی لایه آشکار drop out استفاده از

In [4]:

```

1  # وارد کردن کتابخانه ها
2
3  import numpy
4  import pandas
5  from keras.models import Sequential
6  from keras.layers import Dense
7  from keras.wrappers.scikit_learn import KerasClassifier
8  from sklearn.model_selection import cross_val_score
9  from sklearn.preprocessing import LabelEncoder
10 from sklearn.model_selection import StratifiedKFold
11 from sklearn.preprocessing import StandardScaler
12 from sklearn.pipeline import Pipeline
13
14
15
16 # بار گذاری دیتاست
17
18 dataset = pandas.read_csv("C:\\Users\\ShahinN\\Desktop\\sonar.txt", header=N
19
20 # بخش بندی دیتا
21
22 X = dataset.values[:,0:60].astype(float)
23 Y = dataset.values[:,60]
24
25 # کمی کردن مقادیر ویژگی هدف
26
27 encoder = LabelEncoder()
28 encoder.fit(Y)
29 encoded_Y = encoder.transform(Y)
30
31 # تعریف تابع
32 def create_baseline():
33     # create model
34     model = Sequential()
35     model.add(Dropout(0.2, input_shape=(60,)))
36     model.add(Dense(60, input_dim=60, kernel_initializer='normal', activa
37     model.add(Dense(30, kernel_initializer='normal', activation='relu',
38     model.add(Dense(1, kernel_initializer='normal', activation='sigmoid'
39
40     # Compile model
41     sgd = SGD(lr=0.1, momentum=0.9, decay=0.0, nesterov=False)
42     model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['acc
43     return model
44
45 # ارزیابی مدل با دیتای استاندارد شده
46
47 estimators = []
48 estimators.append(('standardize', StandardScaler()))
49 estimators.append(('mlp', KerasClassifier(build_fn=create_baseline, epochs
50                 batch_size=16, verbose=0)))
51 pipeline = Pipeline(estimators)
52
53
54 kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
55 results = cross_val_score(pipeline, X, encoded_Y, cv=kfold)
56 print("Standardized: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*1

```

Standardized: 87.99% (4.46%)

روی لایه پنهان **drop out** استفاده از

In [5]:

```

1  # وارد کردن کتابخانه ها
2
3  import numpy
4  import pandas
5  from keras.models import Sequential
6  from keras.layers import Dense
7  from keras.wrappers.scikit_learn import KerasClassifier
8  from sklearn.model_selection import cross_val_score
9  from sklearn.preprocessing import LabelEncoder
10 from sklearn.model_selection import StratifiedKFold
11 from sklearn.preprocessing import StandardScaler
12 from sklearn.pipeline import Pipeline
13
14
15
16 # بار گذاری دیتاست
17
18 dataset = pandas.read_csv("C:\\Users\\ShahinN\\Desktop\\sonar.txt", header=N
19
20 # بخش بندی دیتا
21
22 X = dataset.values[:,0:60].astype(float)
23 Y = dataset.values[:,60]
24
25 # کمی کردن مقادیر ویژگی هدف
26
27 encoder = LabelEncoder()
28 encoder.fit(Y)
29 encoded_Y = encoder.transform(Y)
30
31 # تعریف تابع
32 def create_baseline():
33     # create model
34     model = Sequential()
35     model.add(Dense(60, input_dim=60, kernel_initializer='normal', activa
36     model.add(Dropout(0.2))
37     model.add(Dense(30, kernel_initializer='normal', activation='relu',
38     model.add(Dropout(0.2))
39     model.add(Dense(1, kernel_initializer='normal', activation='sigmoid'
40
41     # Compile model
42     sgd = SGD(lr=0.1, momentum=0.9, decay=0.0, nesterov=False)
43     model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['acc
44     return model
45
46 # ارزیابی مدل با دیتای استاندارد شده
47
48 estimators = []
49 estimators.append(('standardize', StandardScaler()))
50 estimators.append(('mlp', KerasClassifier(build_fn=create_baseline, epochs
51     batch_size=16, verbose=0)))
52 pipeline = Pipeline(estimators)
53
54
55 kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
56 results = cross_val_score(pipeline, X, encoded_Y, cv=kfold)

```

```
57 print("Standardized: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
```

Standardized: 82.80% (9.68%)

از مقادیر دراپ اوت در بازه 0.2 تا 0.5 در لایه پنهان استفاده کنید

یک شبکه عصبی بزرگتر طراحی کنید

از دراپ اوت در لایه ورودی یا آشکار استفاده کنید

نرخ یادگیری و مومنتوم را افزایش دهید. نرخ یادگیری را به 0.1 و مومنتوم را بین 0.9 تا 0.99 قرار دهید

محدودیت ماکزیمم نرم وزن ها را 4 یا 5 قرار دهید

In [ ]:

1