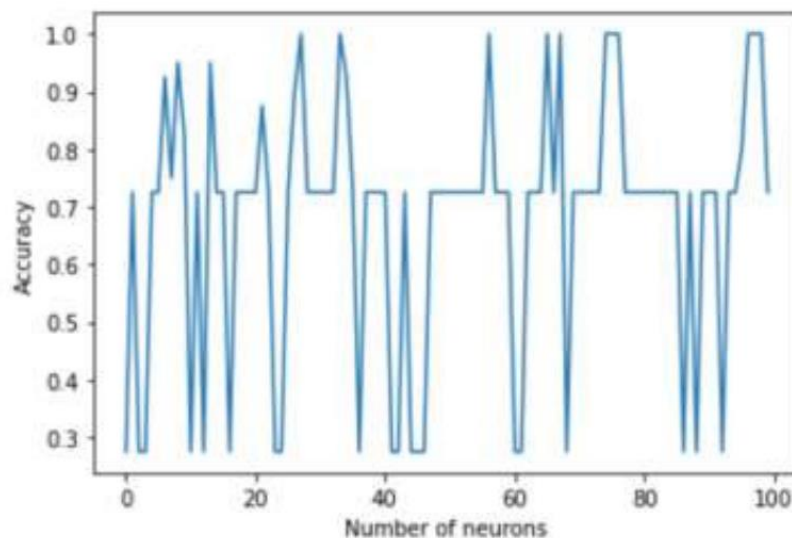


1. Change the number of neurons in the hidden layer and plot the performance for varying number of hidden neurons wrt. the training set for various iterations.

```
1 import pandas as pd
2
3 df = pd.read_csv('../weather.csv').iloc[0:200]
4
5 data = df.drop(['RainTomorrow'], axis = 1)
6 target = df['RainTomorrow']
7
8 target = target.replace('Yes', '1')
9 target = target.replace('No', '0')
10
11 from sklearn.model_selection import train_test_split
12 datasets = train_test_split(data, target,
13                             test_size=0.2)
14
15 train_data, test_data, train_labels, test_labels = datasets
```

```
1 from sklearn.neural_network import MLPClassifier
2 from sklearn.metrics import accuracy_score
3
4 acc = np.zeros(100)
5
6 for i in range(100):
7     mlp = MLPClassifier(hidden_layer_sizes=(i+1,), max_iter=1000)
8     mlp.fit(train_data, train_labels)
9     predictions_test = mlp.predict(test_data)
10    acc[i] = accuracy_score(predictions_test, test_labels)
11
12 plt.xlabel("Number of neurons")
13 plt.ylabel("Accuracy")
14 plt.plot(acc)
```

Out[386]: [<matplotlib.lines.Line2D at 0x7fd4abc10e20>]



2. Change the number of hidden layers for the number of neurons with best performance from 1 and plot the performance for varying number of hidden neurons wrt. the training set for various iterations.

```
1 n = np.argmax(acc)
2 print("Number of neurons = ", n)
3
4 arr = np.zeros(5)
5
6 mlp = MLPClassifier(hidden_layer_sizes=(n, ), max_iter=1000)
7 mlp.fit(train_data, train_labels)
8 predictions_test = mlp.predict(test_data)
9 arr[0] = accuracy_score(predictions_test, test_labels)
10
11 mlp = MLPClassifier(hidden_layer_sizes=(n, n), max_iter=1000)
12 mlp.fit(train_data, train_labels)
13 predictions_test = mlp.predict(test_data)
14 arr[1] = accuracy_score(predictions_test, test_labels)
15
16 mlp = MLPClassifier(hidden_layer_sizes=(n, n, n), max_iter=1000)
17 mlp.fit(train_data, train_labels)
18 predictions_test = mlp.predict(test_data)
19 arr[2] = accuracy_score(predictions_test, test_labels)
20
21 mlp = MLPClassifier(hidden_layer_sizes=(n, n, n, n), max_iter=1000)
22 mlp.fit(train_data, train_labels)
23 predictions_test = mlp.predict(test_data)
24 arr[3] = accuracy_score(predictions_test, test_labels)
25
26 mlp = MLPClassifier(hidden_layer_sizes=(n, n, n, n, n), max_iter=1000)
27 mlp.fit(train_data, train_labels)
28 predictions_test = mlp.predict(test_data)
29 arr[4] = accuracy_score(predictions_test, test_labels)
30
31 plt.xlabel("Number of hidden layers")
32 plt.ylabel("Accuracy")
33 plt.plot(arr)
```

Number of neurons = 27

Out[387]: [<matplotlib.lines.Line2D at 0x7fd4abd5a1f0>]

