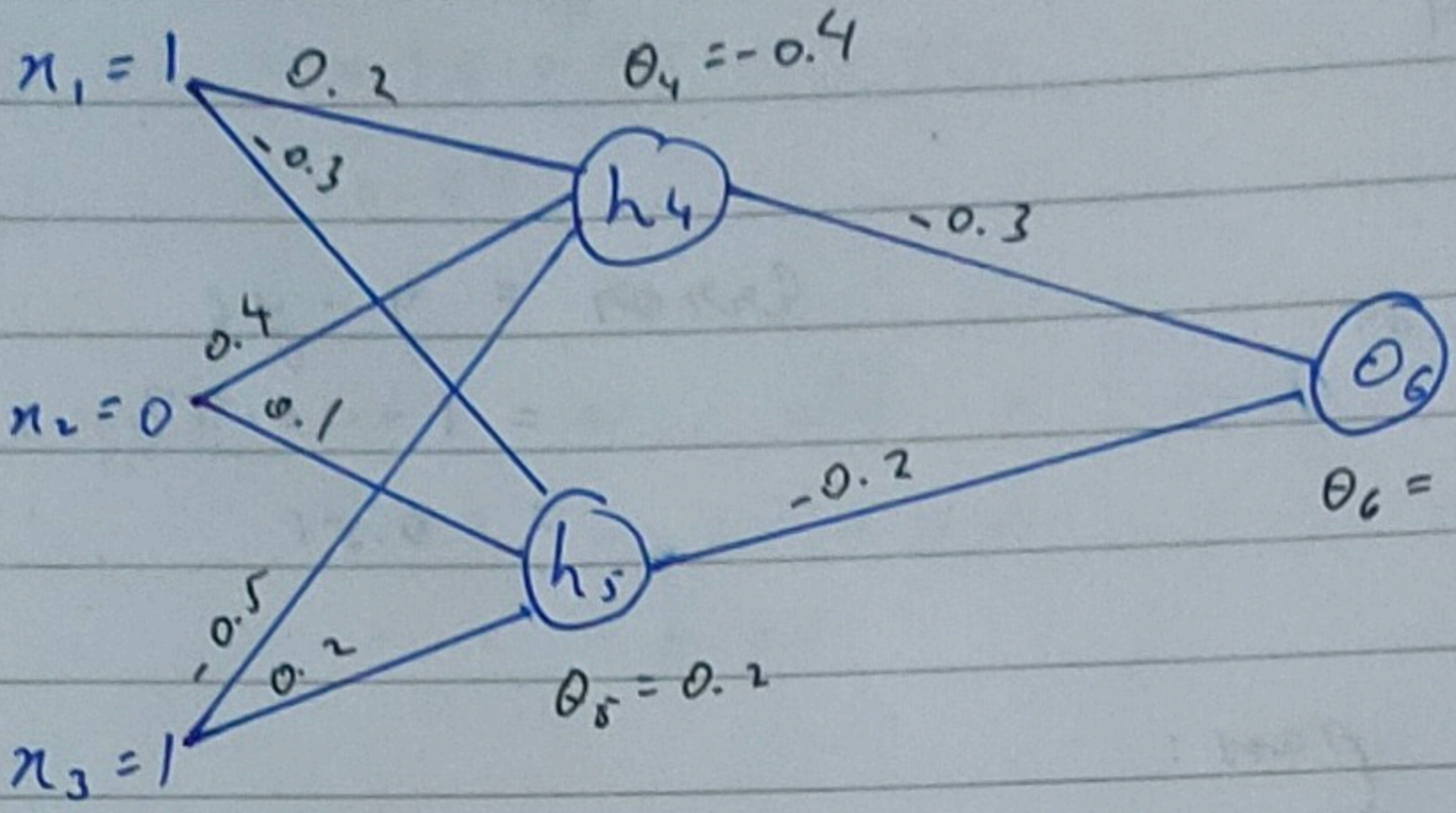
20BRS1094

RISHIK KUMAR



forward pass:

O/P for y_4, y_5, y_6

$$a_j = \sum_j w_{ij} x_i + \theta_j$$

$$y_j = F(a_j) = \frac{1}{1+e^{-a_j}}$$

$$a_4 = w_{14} x_1 + w_{24} x_2 + w_{34} x_3 + \theta_4$$

$$= 0.2 \cdot 1 + 0.4 \cdot 0 + -0.5 \cdot 1 + -0.4$$

$$= -0.3 - 0.4$$

$$= -0.7$$

$$y_4 = \frac{1}{1+e^{+0.7}}$$

$$= 0.332$$

$$a_5 = -0.3 \cdot 1 + 0.1 \cdot 0 + 0.2 \cdot 1 + 0.2$$

$$= 0.1$$

$$y_5 = \frac{1}{1+e^{-0.1}}$$

$$= 0.525$$

$a_6 = -0.3 \cdot 0.332 + -0.2 \times 0.2 + 0.1$

$= -0.04$

$y_6 = \frac{1}{1+e^{0.04}}$

$= 0.49$

Error $= y - y_6$

$= 1 - 0.49$

$= 0.51$

Backward pass:

$\delta_6 = y_6(1-y_6)(y-y_6)$

$= 0.49(0.01)(0.51)$

$= 0.127$

for hidden layer

$\delta_j = y_j(1-y_j)\sum_K \delta_K \omega_{jK}$

$\delta_5 = 0.525(1-0.525)\,0.127 \times -0.2$

$= -0.006$

$\delta_4 = y_4(1-y_4)\sum_K \delta_K \omega_{4K}$

$= 0.332(1-0.332)\,0.127 \times (-0.3)$

$= -0.008$

compute new weight

$\Delta w_{ij} = \alpha \delta_j y_i$

$\Delta w_{56} = 0.9 \times 0.127 \times 0.525$

$= 0.06$

w_{56} new $= w_{56} + \Delta w_{56}$

$= -0.2 + 0.06$

$= -0.14$

$$\Delta w_{46} = \alpha \delta_6 y_4$$
$$= 0.9 \times 0.127 \times 0.332$$
$$= 0.04$$

$$w_{46} \text{ new} = w_{46} \text{ old} + \Delta w_{46}$$
$$= -0.3 + 0.04$$
$$= -0.26$$

$$\Delta w_{14} = \alpha \delta_4 x_1$$
$$= 0.9 \times -0.008 \times 1$$
$$= -0.0072$$

$$w_{14} \text{ new} = w_{14} \text{ old} + \Delta w_{14}$$
$$= 0.2 - 0.0072$$
$$= 0.1928$$

$$\Delta w_{24} = \alpha \delta_4 x_2$$
$$= 0.9 \times -0.008 \times 0$$
$$= 0$$

$$w_{24} \text{ new} = w_{24} \text{ old} + \Delta w_{24}$$
$$= 0.4$$

$$\Delta w_{34} = \alpha \delta_4 x_3$$
$$= 0.9 \times -0.008 \times 1$$
$$= -0.0072$$

$$w_{34} \text{ new} = w_{34} \text{ old} + \Delta w_{34}$$
$$= -0.5 - 0.0072$$
$$= -0.5072$$

$$\Delta w_{1,5} = \alpha \delta_5 x_1$$

$$= 0.9 \times -0.006 \times 1$$

$$= -0.0036$$

$$w_{1,5} \text{ new} = w_{1,5} \text{ old} + \Delta w_{1,5}$$

$$= -0.3 - 0.0036$$

$$= -0.3036$$

$$\Delta w_{2,5} = \alpha \delta_5 x_2$$

$$= 0.9 \times -0.006 \times 0$$

$$= 0$$

no update

$$\Delta w_{3,5} = 0.9 \times -0.006 \times 1$$

$$= -0.0036$$

$$w_{3,5} \text{ new} = 0.2 - 0.0036$$

$$= 0.1964$$

i	j	w_{ij}	δ_j	x_i	α	updated w_{ij}
1	4	0.2	-0.008	1	0.9	0.1928
1	5	-0.3	-0.006	1	0.9	-0.3036
2	4	0.4	-0.008	0	0.9	0.4
2	5	0.1	-0.006	0	0.9	0.1
3	4	-0.5	-0.006	1	0.9	~~0.1964~~ -0.5072
3	5	0.2	-0.006	1	0.9	0.1964
4	6	-0.3	0.127	0.332	0.9	-0.26
5	6	-0.2	0.127	0.525	0.9	-0.14

$$\theta_j \text{ new} = \theta_j \text{ old} + \alpha \delta_j$$

θ_j	old	δ_j	α	θ_j new
θ_6	0.1	0.127	0.9	0.218
θ_5	0.2	-0.006	0.9	0.1946
θ_4	-0.4	-0.008	0.9	-0.4072

20BRS1094 Rishik Kumar

Importing Libraries

In [22]:
```
import numpy as np
import pandas as pd
from sklearn.neural_network import MLPClassifier
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

Importing data

In [4]:
```
data=pd.read_csv('HR_comma_sep.csv')
```

Visualizing data

In [5]:
```
data.head()
```

Out[5]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident
0	0.38	0.53	2	157	3	0
1	0.80	0.86	5	262	6	0
2	0.11	0.88	7	272	4	0
3	0.72	0.87	5	223	5	0
4	0.37	0.52	2	159	3	0

In [7]:
```
data.tail()
```

Out[7]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accid
14994	0.40	0.57	2	151	3	
14995	0.37	0.48	2	160	3	
14996	0.37	0.53	2	143	3	
14997	0.11	0.96	6	280	4	
14998	0.37	0.52	2	158	3	

In [8]:
```
data.size
```

Out[8]:
```
149990
```

In [9]:
```
data.shape
```

Out[9]:
```
(14999, 10)
```

In [10]:
```
data.ndim
```

Out[10]:
```
2
```

PreProcessing

In [13]:
```
label_encode = preprocessing.LabelEncoder()
data['salary']=label_encode.fit_transform(data['salary'])
data['sales']=label_encode.fit_transform(data['sales'])
```

In [15]:
```
data.head()
```

Out[15]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident
0	0.38	0.53	2	157	3	0
1	0.80	0.86	5	262	6	0
2	0.11	0.88	7	272	4	0
3	0.72	0.87	5	223	5	0
4	0.37	0.52	2	159	3	0

In [16]:
```
x=data[['satisfaction_level', 'last_evaluation', 'number_project', 'average_montly_hours', 'time_
y=data['left']
```

In [18]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```

Model

In [19]:
```
clf = MLPClassifier(hidden_layer_sizes=(2,),random_state=1,verbose=True,learning_rate_init=0.9)
```

In [20]:
```
clf.fit(x_train,y_train)
```

```
Iteration 1, loss = 0.58423237
Iteration 2, loss = 0.55316835
Iteration 3, loss = 0.55527560
Iteration 4, loss = 0.55256458
Iteration 5, loss = 0.55143350
Iteration 6, loss = 0.55178761
Iteration 7, loss = 0.55703662
Iteration 8, loss = 0.55289345
Iteration 9, loss = 0.55424176
Iteration 10, loss = 0.55471256
Iteration 11, loss = 0.55316791
Iteration 12, loss = 0.55158679
Iteration 13, loss = 0.56315907
Iteration 14, loss = 0.55408114
Iteration 15, loss = 0.55226062
Iteration 16, loss = 0.55492089
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
```

Out[20]:

▾ MLPClassifier

```
MLPClassifier(hidden_layer_sizes=(2,), learning_rate_init=0.9, random_state=1,
              verbose=True)
```

Testing

In [21]:
```
ypred=clf.predict(x_test)
```

In [23]:
```
accuracy_score(y_test,ypred)
```

Out[23]:
```
0.7663333333333333
```

In [24]:
```
#Changing random state
clf = MLPClassifier(hidden_layer_sizes=(2,),random_state=5,verbose=True,learning_rate_init=0.9)
```

In [25]:
```
clf.fit(x_train,y_train)
```

```
Iteration 1, loss = 1.88619726
Iteration 2, loss = 0.55200770
Iteration 3, loss = 0.55201958
Iteration 4, loss = 0.55237469
Iteration 5, loss = 0.55392407
Iteration 6, loss = 0.55163944
Iteration 7, loss = 0.55246462
Iteration 8, loss = 0.55376748
Iteration 9, loss = 0.55204566
Iteration 10, loss = 0.55375577
Iteration 11, loss = 0.55434810
Iteration 12, loss = 0.55228695
Iteration 13, loss = 0.55223622
Iteration 14, loss = 0.55365294
Iteration 15, loss = 0.55310372
Iteration 16, loss = 0.55694861
Iteration 17, loss = 0.56117006
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
```

Out[25]:
```
MLPClassifier(hidden_layer_sizes=(2,), learning_rate_init=0.9, random_state=5,
              verbose=True)
```

In [26]:
```
ypred=clf.predict(x_test)
accuracy_score(y_test,ypred)
```

Out[26]:
```
0.7663333333333333
```

In [ ]: