In [4]:
```
x=[[0. ,0.],[0. ,1.],[1. ,0.],[1. ,1.]]
y=[0,1,1,0]
from sklearn.neural_network import MLPClassifier
```

In [5]:
```
clf = MLPClassifier(solver='sgd', activation='logistic', alpha=1e-5, random_state
```

In [6]:
```
clf.fit(x, y)
```

Out[6]:
```
MLPClassifier(activation='logistic', alpha=1e-05, batch_size='auto',
       beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08,
       hidden_layer_sizes=(3, 2), learning_rate='constant',
       learning_rate_init=0.001, max_iter=10000, momentum=0.9,
       nesterovs_momentum=True, power_t=0.5, random_state=1, shuffle=True,
       solver='sgd', tol=0.0001, validation_fraction=0.1, verbose=False,
       warm_start=False)
```

In [8]:
```
clf.predict([[2., 2.], [-1., -2.]])
```

Out[8]:
```
array([0, 1])
```

In [9]:
```
clf.predict([[2., 2.], [-1., -2.], [2., 2.], [-1., -2.]])
```

Out[9]:
```
array([0, 1, 0, 1])
```

In [10]:
```
clf.predict([[2., 2.], [-1., -2.], [2., 2.], [-1., -1.]])
```

Out[10]:
```
array([0, 1, 0, 1])
```

In [11]:
```
clf.predict([[2., 2.], [2., 2.], [2., 2.], [-1., -1.]])
```

Out[11]:
```
array([0, 0, 0, 1])
```

In [12]:
```
clf.coefs_[0]
```

Out[12]:
```
array([[-0.18179433,  0.53310908, -1.0951864 ],
       [-0.43306442, -0.77391332, -0.89313476]])
```

In [13]:
```
clf.intercepts_[0]
```

Out[13]:
```
array([-0.68736944, -0.28795345, -0.22617113])
```

In [ ]:
```

```