deep-neural-network/ | MLP_Tutorial | New Tab

localhost:8888/notebooks/deep-neural-network/MLP_Tutorial.ipynb

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

```
        print("A2 ",a2.shape)

        print("W3 ",W3.shape)
        print("Y_ ",y_.shape)
```


In [ ]:

```
def loss(y_oht,p):
    l = -np.mean(y_oht*np.log(p))
    return l

def one_hot(y,depth):
```

In [28]:

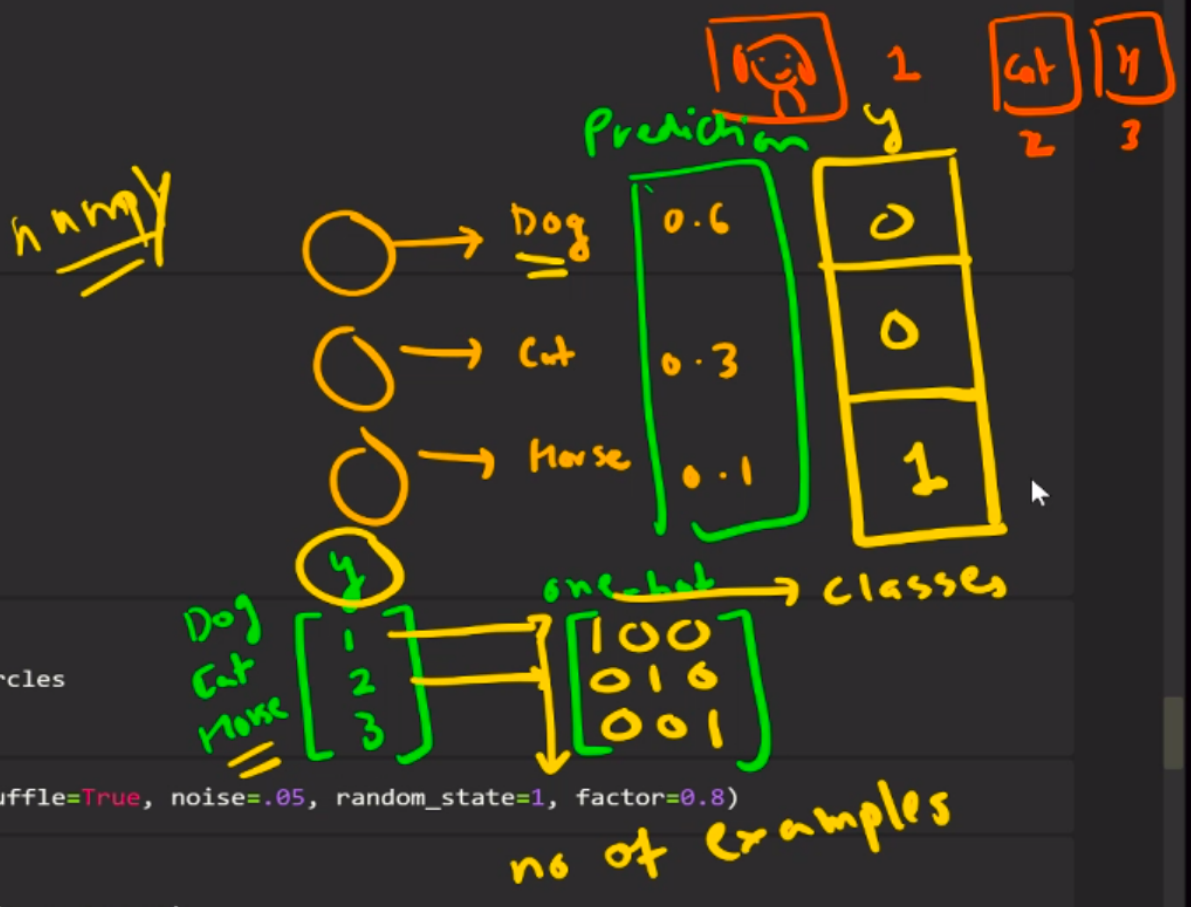
```
## Generate Dataset
from sklearn.datasets import make_circles
import  matplotlib.pyplot as plt
```

In [29]:

```
X,Y = make_circles(n_samples=500, shuffle=True, noise=.05, random_state=1, factor=0.8)
```

In [30]:

```
plt.style.use("seaborn")
plt.scatter(X[:,0],X[:,1],c=Y,cmap=plt.cm.Accent)
```



deep-neural-network/ | MLP_Tutorial | New Tab

localhost:8888/notebooks/deep-neural-network/MLP_Tutorial.ipynb

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

```
        print("Y_ ",y_.shape)
```

In [27]:

```
def loss(y_oht,p):
    l = -np.mean(y_oht*np.log(p))
    return l

def one_hot(y,depth):

    m = y.shape[0]
    y_oht = np.zeros((m,depth))
    y_oht[np.arange(m),y] = 1
```

y=2 ⇒ $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

rohit58405840@gmail.com

In [28]:

```
## Generate Dataset
from sklearn.datasets import make_circles
import  matplotlib.pyplot as plt
```

In [29]:

```
X,Y = make_circles(n_samples=500, shuffle=True, noise=.05, random_state=1, factor=0.8)
```

In [30]:

```
plt.style.use("seaborn")
```