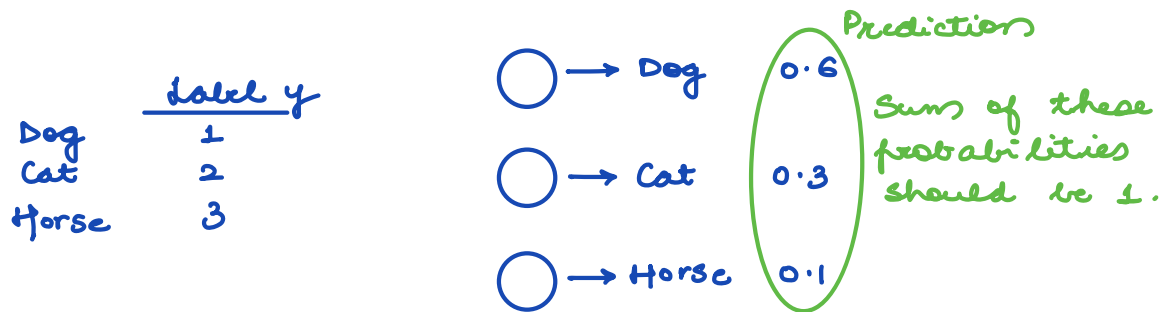


In neural networks, when we feed label to our classifiers you can't feed those labels as strings. That's why we convert those labels into integers.

Suppose you have multi layer perceptron which has 3 output nodes and let us assume each of these nodes predict probabilities for 3 classes.



Let us assume in actual image label was a Dog i.e. 1

One-hot vector for Dog will be: $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ Probability of Dog should be 1 and for all other classes it should be 0.

One-hot vector for Cat will be: $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

One-hot vector for Horse will be: $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

One hot vector = $\begin{bmatrix} \\ \\ \end{bmatrix}$
no of examples \times no. of classes

Given a label matrix y, one hot vector will be:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Examples
↑
4 \times 3 \rightarrow classes

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

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

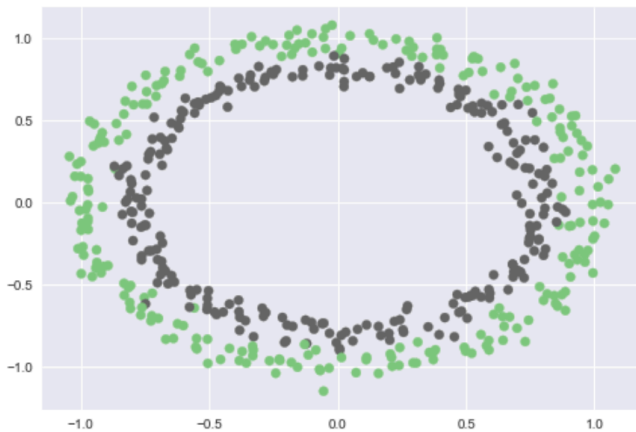
    return y_oht
```

In Tensorflow there is a method `tf.one_hot` but here we will implement from scratch.
 Depth = no. of classes

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

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

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



```
model = NeuralNetwork(input_size=2, layers=[10,4], output_size=2)
```

```
model.forward(X)
```

```
array([[0.07135496, 0.92864504],
       [0.0252705 , 0.9747295 ]])
```

```
print(Y)
print(Y.shape)
```

```
[1 1 0 0 1 1 1 0 0 0 1 0 0 0 1 0 0 1 1 0 0 1 1 1 0 1 0 1 0 0 0 1 0 1 1 0 1
 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 0 0
 0 1 1 1 1 0 1 0 0 1 0 0 1 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 1
 1 1 1 1 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 1 0 1 1 1 1 1 0 0 0 0 1 0 1 0 0
 0 0 0 0 1 0 1 1 0 0 0 1 0 1 1 1 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1
 1 0 0 0 1 0 0 1 1 0 1 0 1 1 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 0 0 1 1 0 1 1 1 1 0 0 1 1 1 1 1
 0 1 1 0 1 0 0 1 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 1 1 0 1 1
 1 0 1 1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 1 0 0 0 0 0 1 0 0 1 1 1 0 0 0 1 1
 1 1 0 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 0 1 0 1
 0 0 1 1 1 0 1 0 1 1 0 0 0 0 1 1 0 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 1 0 1 0 0
 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 1 0 1 1 0 0 0 1 0 1
 0 1 1 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 0 0 0 1 1 0 1 1 1 1 1 1 1 1
 1 0 1 1 0 0 1 0 0 1 1 0 0 1 1 0 1 0 0 0]
(500,)
```

```
y_oht = one_hot(Y, 2)
print(y_oht)
print(y_oht.shape)
```

```
[[0. 1.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [1. 0.]]
(500, 2)
```

```
y_oht = one_hot(Y, 3)
print(y_oht)
print(y_oht.shape)
```

```
[[0. 1. 0.]
 [0. 1. 0.]
 [1. 0. 0.]
 ...
 [0. 1. 0.]
 [1. 0. 0.]
 [1. 0. 0.]]
(500, 3)
```

→ Each entry is a 3 Dimensional Vector

last column is 0 because we have only 2 classes.

Do $\text{one_hot}(y, k)$ if you have k classes in output.