

```
# -*- coding: utf-8 -*-  
"""Modelcode_ANN.ipynb
```

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1qrWwbUlGyvYIkum440lAFuQJC0J_WaES
"""

```
# importing the Keras Libraries and package  
import keras  
from keras.models import Sequential # For building the Neural  
Network layer by layer  
from keras.layers import Dense # used to add fully connected  
layer in ANN#Simple Neural Network Model Code using Tensorflow  
#Sequential model with one layer, i.e, 1 neuron, input shape =2,  
kernel_initializer=weights, bias_initializer = bias  
model = keras.Sequential([  
    keras.layers.Dense(1, input_shape=(2,), activation='sigmoid',  
kernel_initializer='ones', bias_initializer='zeros')])
```

```
#Multiple Layers  
classifier = Sequential()  
# adding the input layer and first hidden layer  
classifier.add(Dense(output_dim, init = 'uniform', activation =  
'relu', input_dim ))  
# adding the second hidden layer  
classifier.add(Dense(output_dim, init = 'uniform', activation =  
'relu'))  
#Adding the output layer  
classifier.add(Dense(output_dim = 1, init = 'uniform', activation =  
'sigmoid'))
```

```
#"compile" is a method of Tensorflow. "adam" is the optimizer that can  
perform the gradient descent.  
# loss function as binary_crossentropy  
# The optimizer updates the weights during training and reduces the  
loss.  
model.compile(optimizer='adam',  
              loss='binary_crossentropy',  
              metrics=['accuracy'])
```

```
#The neural network has to train on a certain number of epochs to  
improve the accuracy over time.  
#Thus, when you run this code, you can see the accuracy in each epoch.  
model.fit(X_train_scaled, y_train, epochs=5000)
```

```
#evaluating the model  
model.evaluate(X_test_scaled, y_test)
```

```
#prediction of values  
model.predict(X_test_scaled)
```

```
#To get the value of weights and bias from the model
coef, intercept = model.get_weights()
coef, intercept
```