```
%tensorflow_version 1.x
```

```
TensorFlow 1.x selected.
```

```
import tensorflow as tf
import numpy as np
import pandas as pd
from pylab import rcParams
from pprint import pprint
import matplotlib.pyplot as plt
import warnings
```

```
!pip3 install ann_visualizer
```

```
Collecting ann_visualizer
  Downloading ann_visualizer-2.5.tar.gz (4.7 kB)
Building wheels for collected packages: ann-visualizer
  Building wheel for ann-visualizer (setup.py) ... done
  Created wheel for ann-visualizer: filename=ann_visualizer-2.5-py3-none-any.whl size=4
  Stored in directory: /root/.cache/pip/wheels/1b/fc/58/2ab1c3b30350105929308becddda4fb5
Successfully built ann-visualizer
Installing collected packages: ann-visualizer
Successfully installed ann-visualizer-2.5
```

```
from ann_visualizer.visualize import ann_viz
from graphviz import Source
```

```
from keras.models import Sequential
from keras.layers import Dense
```

```
Using TensorFlow backend.
```

```
training_data = np.array([[0,0],[0,1],[1,0],[1,1]], "float32")
target_data = np.array([[0],[1],[1],[0]], "float32")
```

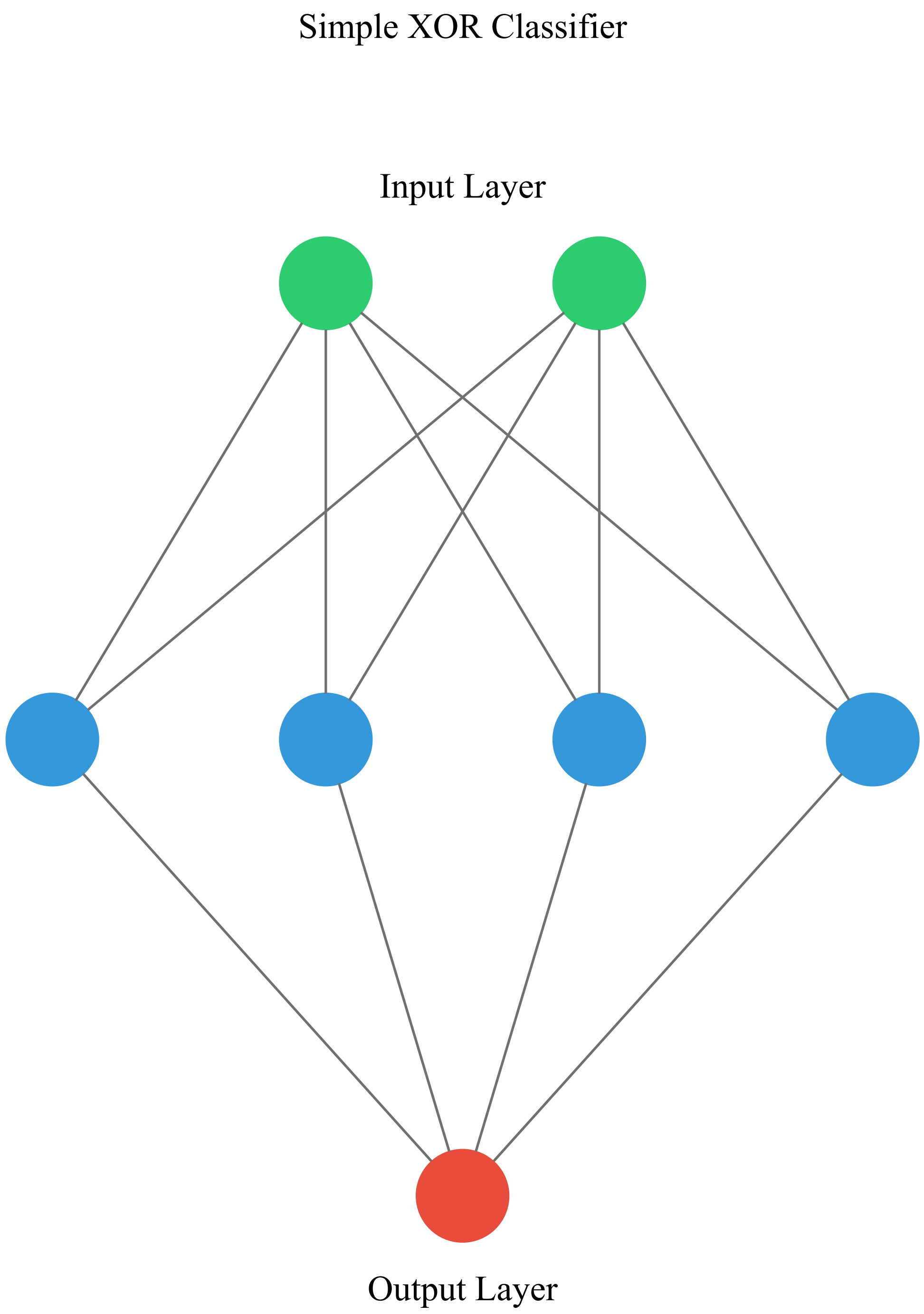
```
model = Sequential()
model.add(Dense(4, input_dim=2, activation='sigmoid'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
```

```
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/tensorflow_core/python/ops/resource
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
```

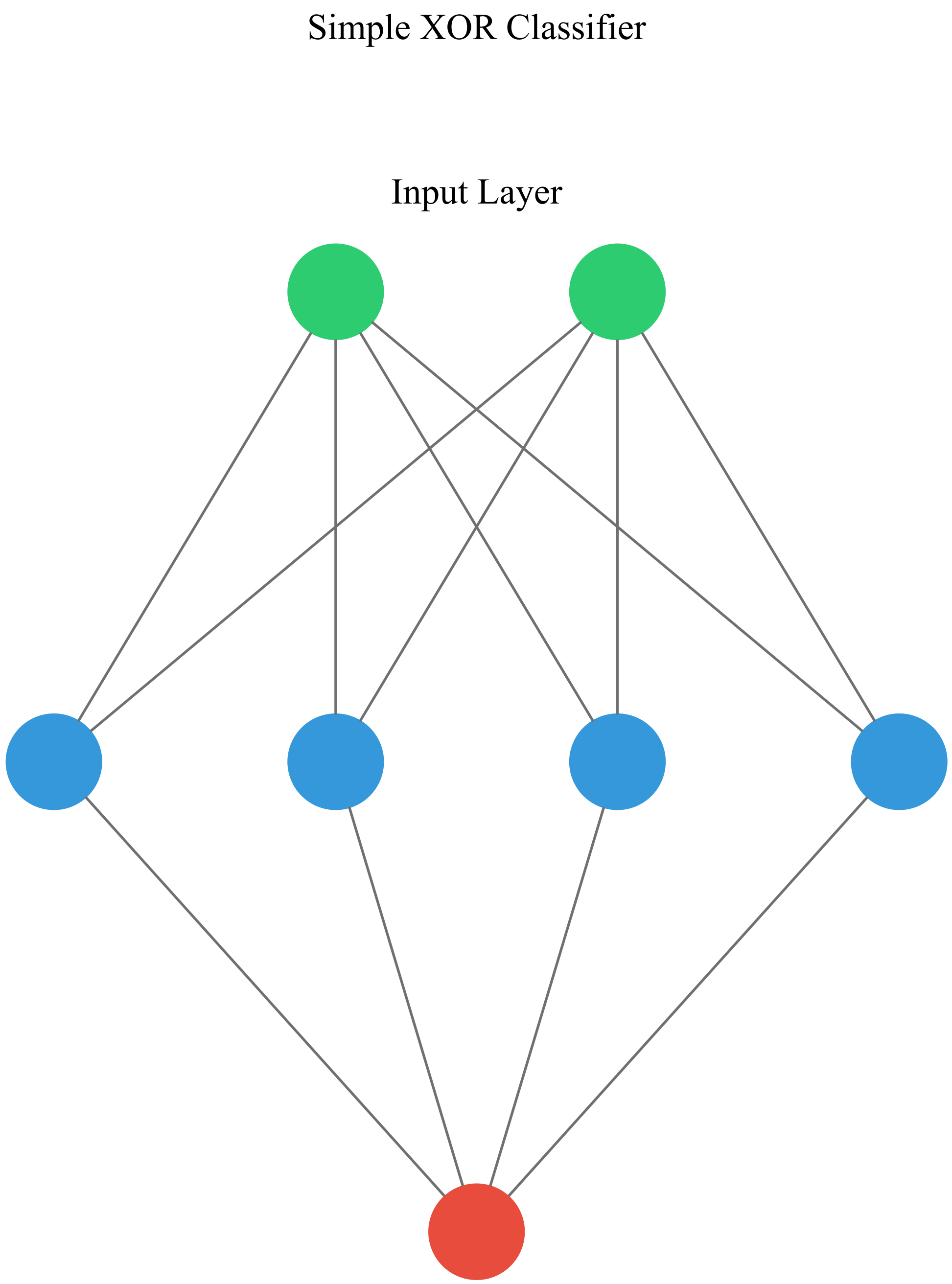
```
ann_viz(model, title="Simple XOR Classifier")

graph_source = Source.from_file('network.gv')
```

```
graph_source
```



```
graph_source
```



```
print(graph_source.source)
```

```
    digraph g {
            graph [nodesep=1 ranksep=2 splines=false]
            edge [arrowhead=none color="#707070"]
            subgraph cluster_input {
                    node [color="#2ecc71" fontcolor="#2ecc71" shape=circle style=filled]
                    color=white
                    1
                    label="Simple XOR Classifier



    Input Layer"
                    rank=same
```

```
                2
                label="Simple XOR Classifier



Input Layer"
                rank=same
        }
        1 -> 3
        2 -> 3
        1 -> 4
        2 -> 4
        1 -> 5
        2 -> 5
        1 -> 6
        2 -> 6
        subgraph cluster_1 {
                color=white
                rank=same
                label="" labeljust=right labelloc=b
                3 [color="#3498db" fontcolor="#3498db" shape=circle style=filled]
                4 [color="#3498db" fontcolor="#3498db" shape=circle style=filled]
                5 [color="#3498db" fontcolor="#3498db" shape=circle style=filled]
                6 [color="#3498db" fontcolor="#3498db" shape=circle style=filled]
        }
        3 -> 7
        4 -> 7
        5 -> 7
        6 -> 7
        subgraph cluster_output {
                node [color="#2ecc71" fontcolor="#2ecc71" shape=circle style=filled]
                color=white
                rank=same
                labeljust=1
                7 [color="#e74c3c" fontcolor="#e74c3c" shape=circle style=filled]
                label="Output Layer" labelloc=bottom
        }
        arrowShape=none
}
```

```
model.get_weights()
```

```
[array([[-0.6296656 , -0.72231627, -0.91836095,  0.14923406],
        [ 0.12357736,  0.37921667,  0.13376975, -0.9686239 ]],
       dtype=float32),
 array([0., 0., 0., 0.], dtype=float32),
 array([[ 0.41817117],
        [ 1.0707359 ],
        [ 0.40867305],
        [-0.89745086]], dtype=float32),
 array([0.], dtype=float32)]
```

● ✕