```
//nomedoficheiro16.js

// configurações da rede e treino, num ficheiro nomedoficheiro16.js


   const brain = require('brain.js');

   Const net1 =new brain.NeuralNetwork();

   Const net2 =new brain.NeuralNetwork();

    Const data1 = [

         {input:[0,0], output:[0]},{input:[0,1], output:[1]},
         {input:[1,0], output:[1]},{input:[1,1], output:[0]}
   ];

   Const data2 = [

         {input:[0,0], output:[0]},{input:[0,1], output:[1]},
         {input:[1,0], output:[1]},{input:[1,1], output:[0]}

   ];

  const options = {iteration:10000, errorThresh:0.004};

  conts t1 =net1.trainAsync(data1,options);

  conts t2 =net2.trainAsync(data2,options);


  Promise.all([t1,t2]).then(values => {

        const res1 =values[0];

        const res2 =values[0];

        console.log(’treino 1 - xor\ntaxa de erro:${res1.error}\niteração: ${
res1.iterations}');

        console.log (‘’);

        console.log(’treino 2 - xor\ntaxa de erro:${res2.error}\niteração: ${
res2.iterations}');

        console.log (‘’);


        const xor00 = parseFloat(net1.run([0,0])).toFixed(0);

        const xor01 = parseFloat(net1.run([0,1])).toFixed(0);

        const xor10 = parseFloat(net1.run([1,0])).toFixed(0);

        const xor11 = parseFloat(net1.run([1,1])).toFixed(0);


        const and00 = parseFloat(net2.run([0,0])).toFixed(0);

        const and01 = parseFloat(net2.run([0,1])).toFixed(0);
        const and10 = parseFloat(net2.run([1,0])).toFixed(0);
        const and11 = parseFloat(net2.run([1,1])).toFixed(0);


        console.log(`0 xor 0: ${xor00}’);
        console.log(`0 xor 1: ${xor01}’);
        console.log(`1 xor 0: ${xor10}’);
        console.log(`1 xor 1: ${xor11}’);
        console.log (‘’);
        console.log(`0 and 0: ${xor00}’);
        console.log(`0 and 1: ${xor01}’);
        console.log(`1 and 0: ${xor10}’);
        console.log(`1 and 1: ${xor11}’);
   }).catch(error = {
      console.log(error);
   });

// Guardem e executem
```