TensorFlow playground Cheat Sheet

Lab 1
In TensorFlow playground, use the bottom left data type (2 groups)

ONE neuron in ONE hidden layer
Learning rate 0.03
Activation Tanh
This should be resolved quickly.

Try to modify the different weights, and the bias of the neuron.

Let's say we have

$X1 \in [-6, +6]$
$X2 \in [-6, +6]$
W1 = -0.69
W2 = -0.67
b = 0.030
Wfinal = -2
Orange point (-4, -3) will result in:

(-4 * -0.69) + (-3 * -0.67) + 0.030 => 4.8 * -2 => -9.6 => tanh(-9.6) = -0.999999990825637
Blue point (3, 4) will result in

(3 * -0.69) + (4 * -0.67) + 0.030 => -4.72 * -2 => 9.44 => tanh(9.44) = 0.999999987365729
-1: orange
+1: blue
Also see spreadsheet neuron.ods, play around with it.

There is this Java Jupyter Notebook illustrating this too.

Lab 2
Use the top-right data (4 groups, in 4 squares)

2 inputs (X1, X2)
Learning Rate 0.03, Activation Tanh
Try with ONE neuron in ONE layer, see why it is not possible to solve this.
Add one neuron (that makes 2), go up to 4
Try it.
Change Activation to Sigmoid, try again
Add One neuron, try again (still with Sigmoid), go up to 1000 epoch at least
Try with Tanh
Try with ReLU <- This one is the best!
Flip the 'Discretize output' box.

Try ***only 1 input***, X_1X_2

Lab 3
Use the 'circular' data (top left)

with 4 neurons, learning rate 0.03, Activation function ReLU, Tanh, or Sigmoid,
compare the output curves, see the different parameters

Lab 4
Use the spiral data (bottom right)

Use the 7 (yes, all of them) input dimensions, 2 hidden layers, 5 and 6 neurons
Activation Function Tanh, ***ReLU***, should be OK around 500 epochs
Do look at the curves when using a Sigmoid
Sigmoid is going from 0 to 1, cannot be suited for the problem here, as we need to go from -1 to +1
Play around, and see how it moves