Problem 2: Neural Networks (50 points)

Part A: Trial by Big Head (15 points)

You've just been hired by tech-giant Hooli to lead the AI lab. A fellow team member affectionately known as Big Head decides to put your skills to the test with a series of conceptual questions. For each of the following questions, circle the **one** best answer:

A1 (3 points) Your neural net is still inaccurate after having been trained for a fixed number of iterations. You consider adjusting the learning rate. Which of the following adjustments might cause the network to converge to a more accurate solution?

A) Increasing the learning rate.
B) Decreasing the learning rate.
C) Either increasing or decreasing the learning rate.
D) None of these: the change in accuracy is independent of learning rate.

A2 (3 points) One motivation for using the sigmoid instead of the stairstep threshold function in back propagation is:

A) It's computationally inefficient to compute the stairstep gradient.
B) The stairstep function is an odd function.
C) Back propagation is only well-defined for the sigmoid function.
D) The sigmoid function is differentiable everywhere.
E) None of the above.

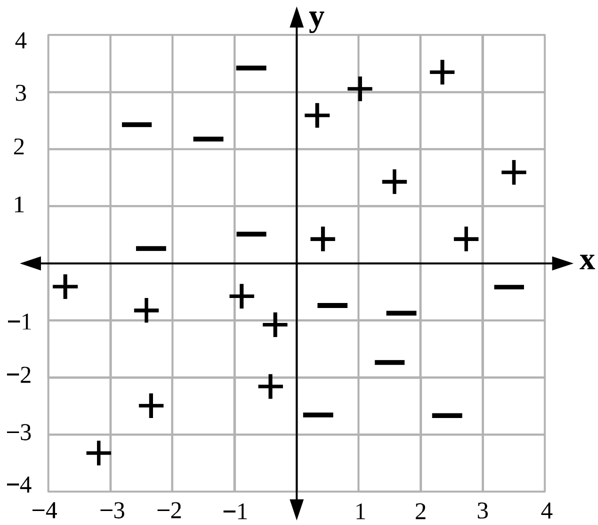
A3 (3 points) Suppose you decide to use **the identity function** as your threshold function:

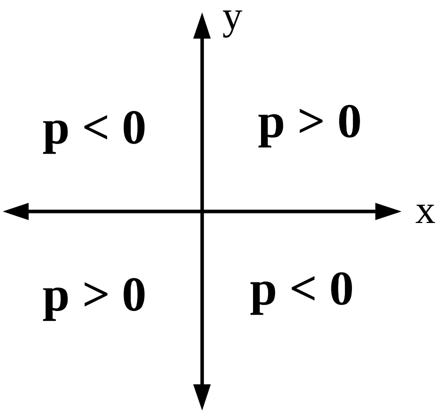
$$f(x) = x$$

In other words, the output of each neuron is just the weighted sum of its inputs. Of the following three neural networks, which one can accurately classify new inputs **that the others cannot**?

A) *Network A*: A neural net with 2 input neurons and 1 output neuron.
B) *Network B*: A neural net with 2 input neurons, 1 hidden-layer neuron, and 1 output neuron.
C) *Network C*: A neural net with 2 input neurons, 3 hidden-layer neurons, and 1 output neuron.
D) Networks A, B, and C can accurately classify the same inputs.
E) Can't tell without more information.

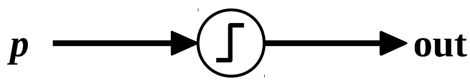
A4 (6 points) You're building a neural net to classify the positive and negative data samples shown below on the left: positive samples will output 1, and negative samples will output 0. Each sample has features *x* and *y*, however, you are only allowed to use the feature $p = x \cdot y$ as input to any neural network. (See below on the right for an illustration of *p*'s value as a function of *x* and *y*.)





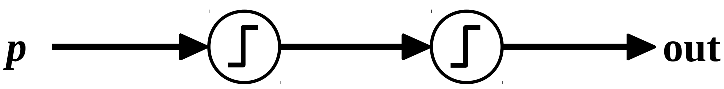
For each of the following network architectures, could a neural network of that shape correctly classify the data? (Assume all neurons use the stairstep threshold function.) In each row, circle either **YES** or **NO**:

Network 1:



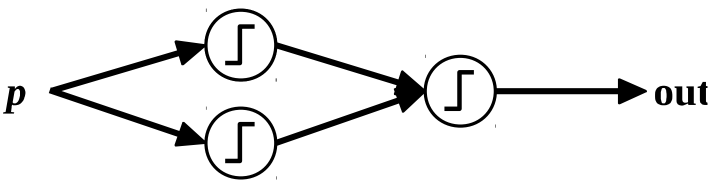
YES **NO**

Network 2:



YES **NO**

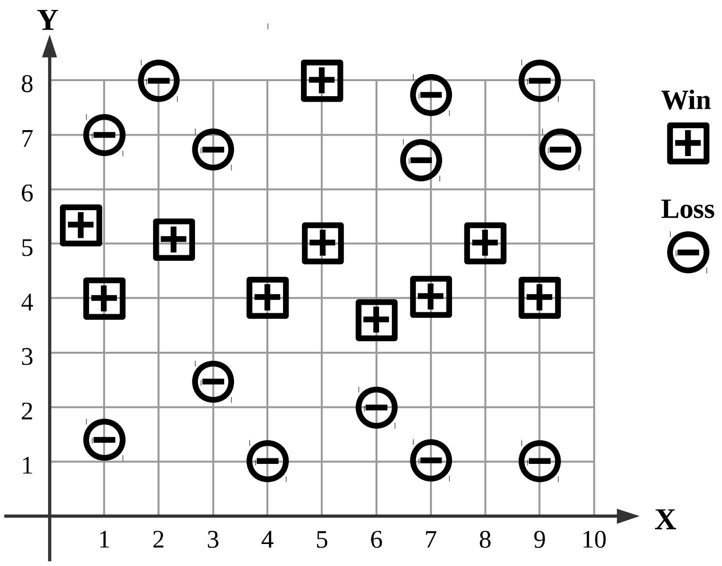
Network 3:



YES **NO**

Part B: Let's Go! (21 points)

Impressed with your knowledge, Big Head places you on a top-secret team tasked with building a Go-playing bot called BetaGo. Your job is to build a neural net that can be used to classify winning (⊞) and losing (⊖) board positions, as shown on the right.

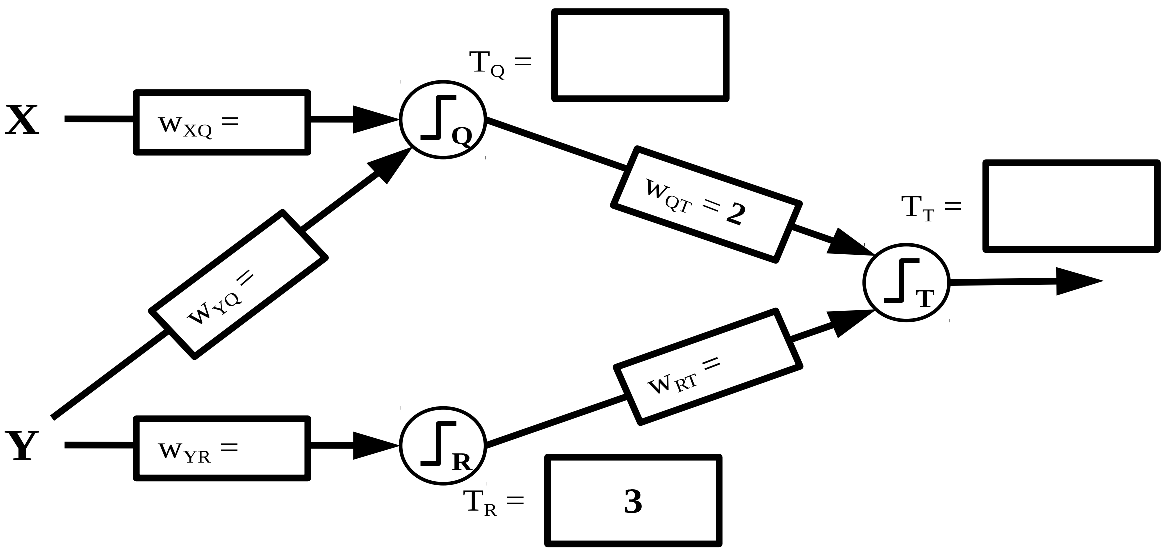


Due to budget cuts, you are limited to using a small net with only three neurons (each of which uses a stairstep threshold function).

Big Head tells you that the small network won't be able to classify *all* of the samples correctly: in fact, it will misclassify *one* sample.

In the neural net skeleton below, we have already filled in weight w_{QT} and threshold T_R for you. Fill in the remaining **four** weights and **two** thresholds with **<u>INTEGER values</u>** such that the net will correctly classify all but one sample. *(We have provided space on the next page to show your work for partial credit.)*

Note: A Win outputs 1; a Loss outputs 0.



For partial credit for part B, you can show your work here:

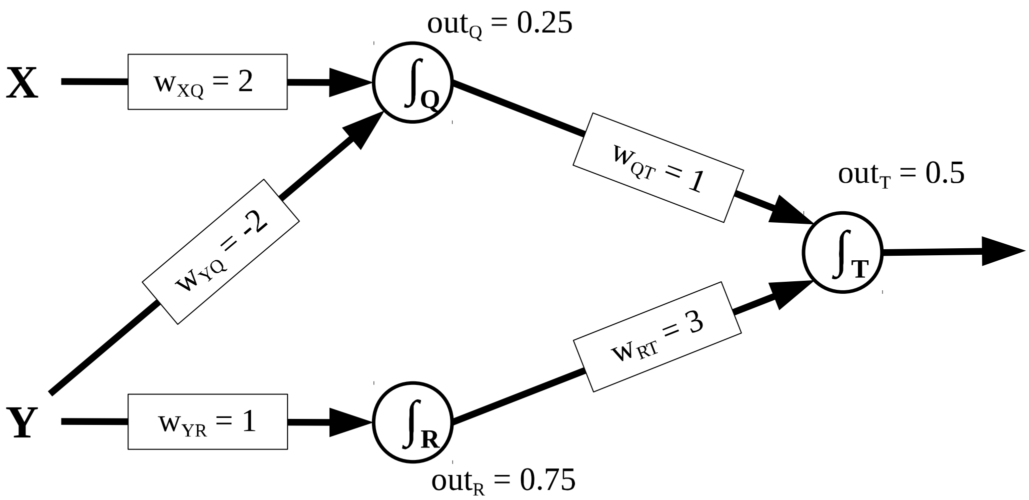
Part C: Back-handed Back-prop (14 points)

Downtrodden startup founder Richard Hendrix approaches and tells Big Head that he doesn't need to manually pick all the weights: there's an automatic way to train the weights of the network using back propagation! Big Head asks for a demonstration of back propagation.

For your convenience, an equation sheet for neural nets is provided on a tear-off sheet at the end of the quiz.

C1 (8 points) The following neural network uses the **sigmoid threshold function**. The weights have been initialized randomly, and you have just performed forward propagation to determine the current outputs of each of the network's neurons. Calculate the values of δ_T and δ_Q, assuming the following:

X = 3, Y = 7, Learning Rate = 1, Desired Output = 0



Space is provided on the next page to show your work.

$\delta_T =$

$\delta_Q =$

For partial credit for part C1, you can show your work here:

C2 (6 points) Calculate the weight update for W_{YR}^{new}, the weight between input Y and neuron R. **You can leave your answer in terms of δ_Q, δ_R, and δ_T.**

$W_{YR}^{new} =$

For partial credit for part C2, you can show your work here: