CSci 343 Fundamentals of Data Science
FINAL CHALLENGE

You may demo your code (instructions below)
as soon as you get it working.
Demo and upload on or before 4:00PM on Wednesday, Dec. 5.

Points Available:
200 XP for a working demonstration
50 XP for readable & understandable code

Objectives:

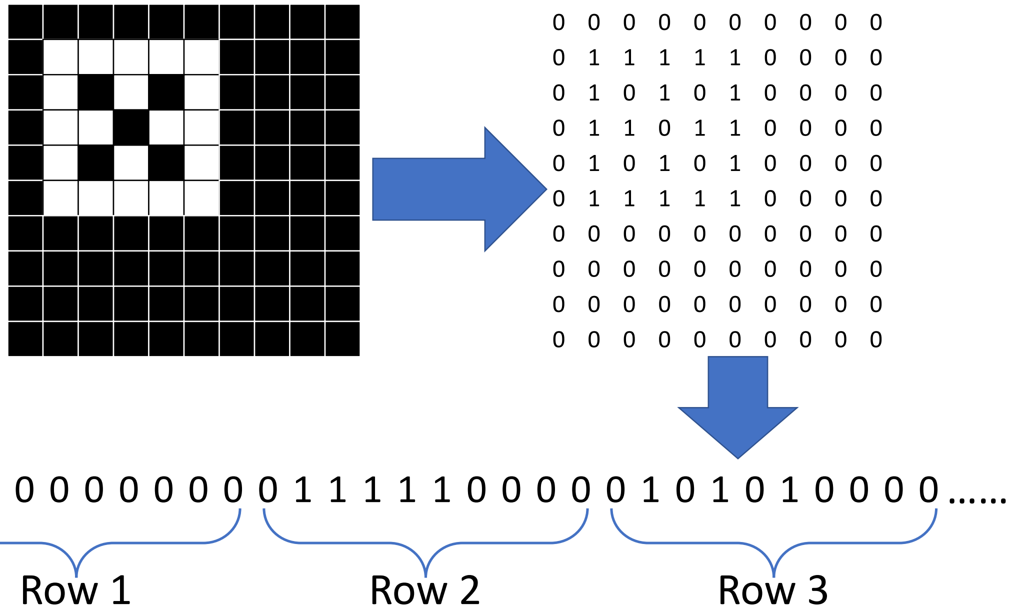
- Learn the basics of Neural Networks
- Have fun!

Assignment:

The semester has been fun! We should end it with something fun too. Let's play a game! I've always enjoyed board games that combine both skill and chance. They usually have something in common. They need a good ol' fashioned random number generator, also known as dice!

You have been hired by the board game company HasBoard, Inc. They are now venturing into the world of robotics and are building a robot that can play board games with you when no one else wants to! However, they need the robot to be able to see dice rolls and know what numbers come up. You've been given the task of writing the neural network that will interpret rolls of the dice. Specifically, you'll be analyzing black and white images of a single die.

The die images are specified using 1's and 0's over a 10 x 10 grid. The die will occupy a 5 x 5 area. The die could be anywhere within the 10 x 10 grid, but it will always be fully visible. The background of the image is black (0s) and the die face is white (1s) with black dots (0s). This is illustrated below. Your neural network must accept a list of a hundred 1's and 0's that represent the image of the die. A sample image and how to encode it for your neural network is show below.



```
0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 0 0 0 0
0 1 0 1 0 1 0 0 0 0
0 1 1 0 1 1 0 0 0 0
0 1 0 1 0 1 0 0 0 0
0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0......

You will need to train your neural network with images of dice at many locations within the 10 x 10 grid. Input to your neural network will be a list of 100 float values (when casting values for this assignment be sure to use `float( )` instead of `np.float32( )`). You must encode your output (and targets) such that the network returns the face value divided by 10 (1 = 0.1, 2 = 0.2, 3 = 0.3, and so on). Your neural network should return a predicted value between 0.1 and 0.6. Values can be fractional beyond the first decimal place and will be rounded up (e.g. 0.475 will round to 0.5).

Your assignment will be tested and scored by uploading your saved neural network file to the testing website (https://john.cs.olemiss.edu/~jones/343NeuralNetwork.html). Your solution will be tested against 1,000 randomly generated dice rolls. Half of the testcases will be of "normal" difficulty. The other half will be of "tricky" difficulty and may include rotated dice. The normal cases will always be oriented up and down. Your functionality points will be awarded based on how well you train your neural network. ***Your grade will only be based on the "normal" difficulty testcases*** (my gift to you, Happy Holidays). The "tricky" testcases are for you to show off your awesome neural network skills. To be awarded any points on this assignment, your network must be correct no less than 20% of the time. Beating this threshold will award you 40XP. For every 10% over this, you will receive an additional 20XP. For example, a correctness score of 81.2% will result in 162.4XP.

When submitting your solution, you may choose to participate in the class leaderboard (shows the top 10 solutions). Participation in the leaderboard is *strictly voluntary*. The top 1st, 2nd, and 3rd leaderboard solutions will receive bonuses of 20XP, 10XP, and 5XP respectively.

You will need the use to neuro.py package available on the data distribution site. An updated version has been posted as of Nov. 14. You are required to use this version. Your neural network may not function during testing if you used the out-of-date version. You will need to save your trained neural network to a file using the `neuro.writeNetworkToFile( filename )` function. You may reference our in-class examples. The primary purpose of this assignment is for you to learn how to train a neural network by generating good testing data.

To get credit for this assignment you must upload both your code AND your training data to Blackboard.

Submission Instructions:

Upload ***all your code and your training data*** to Blackboard as a single ZIP file. Name your ZIP file s*piritAnimal*.zip, where s*piritAnimal* is your class user ID (not your webID or ID number). Be sure to name your main source file "s*piritAnimal*.py". In a comment at the top of the file, include the following information. Spirit Animal User ID, Date the file was last edited, Challenge Number, and cite any sources that you used as a reference for code, data, and content (including title and URL).

Example of Dice Faces