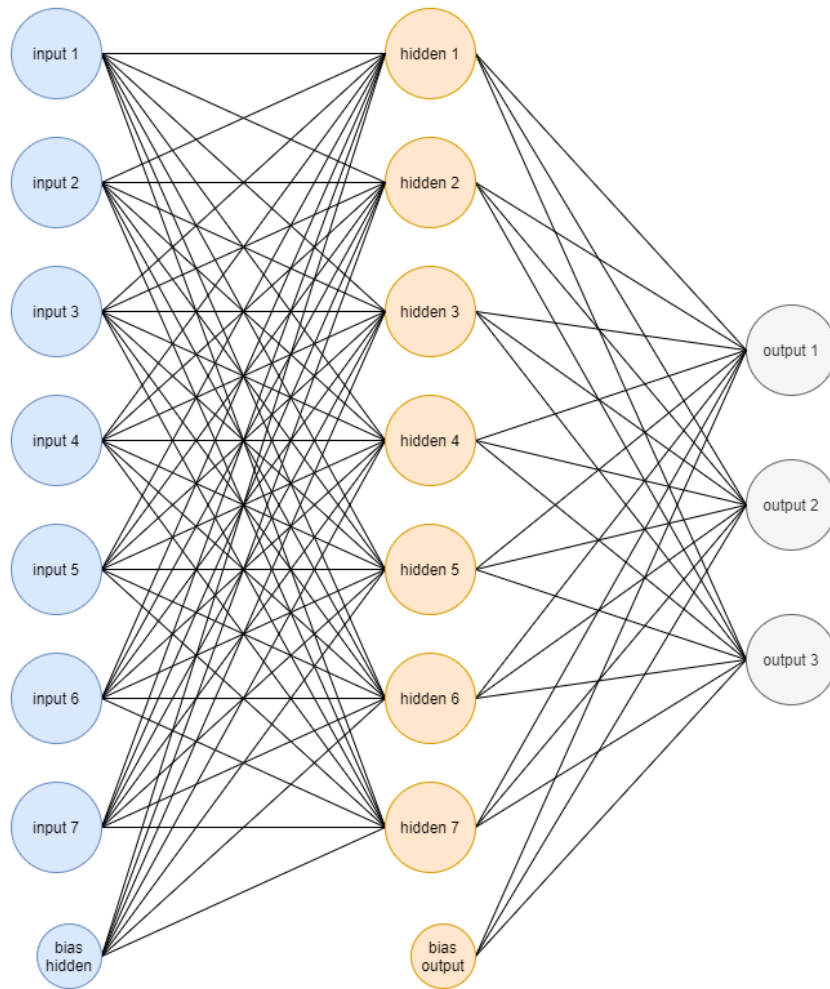


## AIM 5007 Neural Networks and Deep Learning

### Mini-Project 1

#### Instructions

We previously developed individual functions for computing the various pieces to use the following architecture for the seeds data in Bramer:



In this mini-project, you will develop a fully trained model. Using your functions (or adaptations of them as necessary), and using the sigmoid activation function for each layer, train the model and deliver a final set of weights on the seeds data. The data set can be found here (you may need to clean it a bit – if observations have missing values, you may remove them):

<https://archive.ics.uci.edu/ml/datasets/seeds>



You will need to decide on a learning rate (I recommend starting with something like 0.2), but for now use an approach that updates the weights after each pass rather than a batch approach.

Randomly select 150 rows of the data to use for training the model. (Use a seed to create reproducible work.) Hold the remaining observations as your test data set. Once you've trained the model on the 150 observations (using however many epochs you deem appropriate), test it on the holdout data to see how you did.

Your final deliverable should be a python notebook that shows the following (as well as how you got to them):

- The final weight matrices for each layer
- A description of the process (how many epochs, final error rate based on the training data, and so on – just a short paragraph will do)
- Results of your model on the test data (give me the confusion matrix and a few different measures of performance – you may choose which ones to use here)

### Grading Rubric

- |     |  |
|-----|--|
| 0-3 | Reasonable accuracy from the final model   |
| 0-3 | Properly written code that works when I verify it  |
| 0-2 | Organization (easy to follow, well structured, good, concise comments in the code, etc.) |
| 0-2 | Clarity of your description of the process   |