Master of Technology

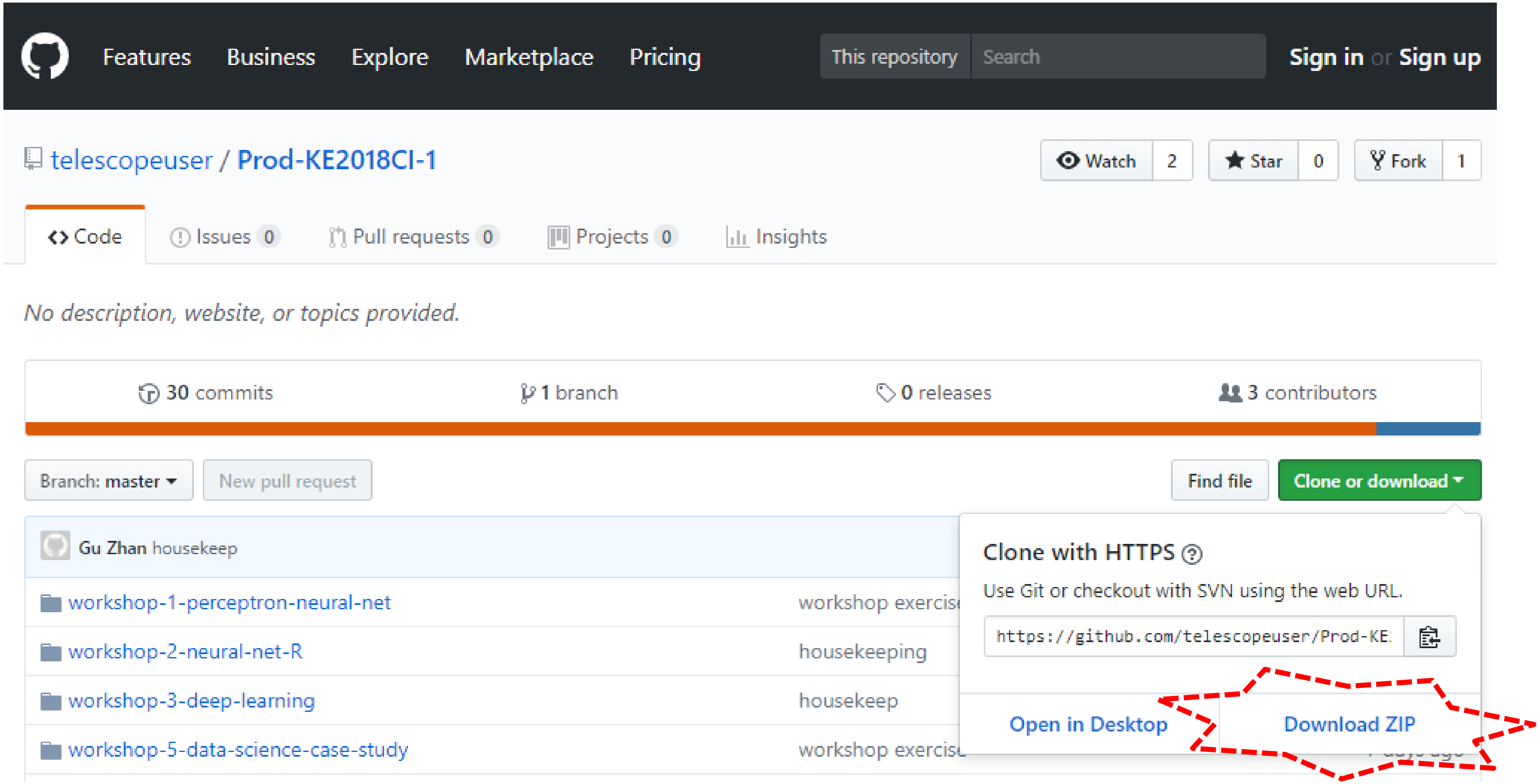
Unit 2/6: Computational Intelligence I

Workshop (1A):
Perceptron Exercise

Institute of Systems Science
National University of Singapore

Download Workshop Materials from Cloud

- **Download Workshop Material**
  - » https://github.com/telescopeuser/Prod-KE2018CI-1



No description, website, or topics provided.

Gu Zhan housekeep

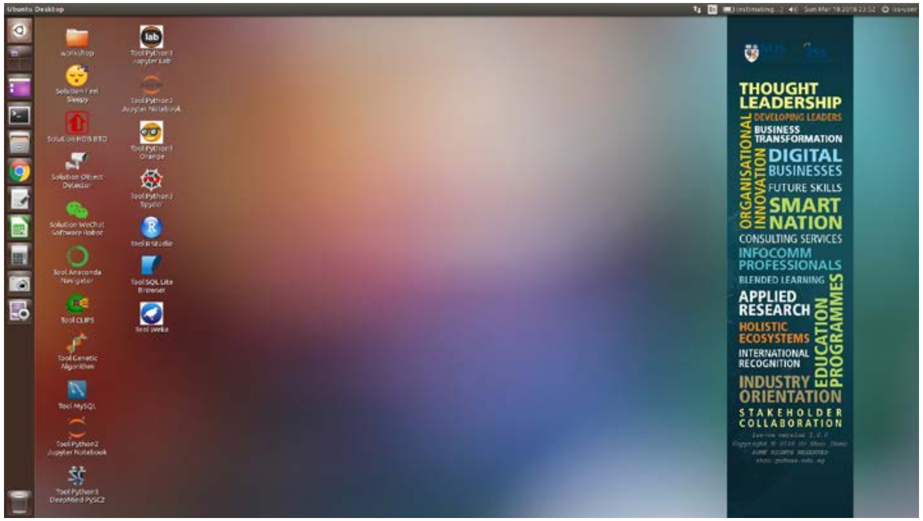
workshop-1-perceptron-neural-net	workshop exercise
workshop-2-neural-net-R	housekeeping
workshop-3-deep-learning	housekeep
workshop-5-data-science-case-study	workshop exercise

Use Git or checkout with SVN using the web URL.

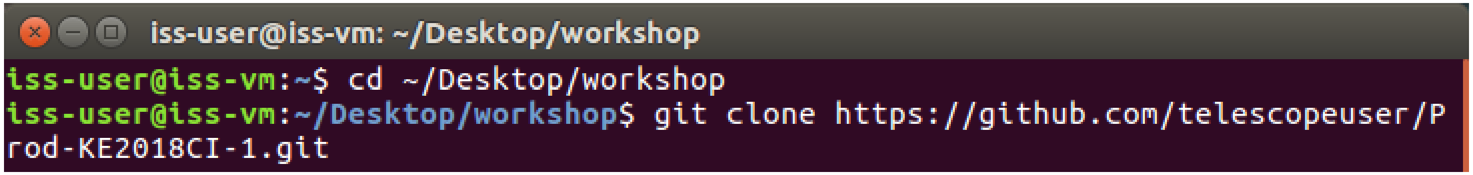
```
https://github.com/telescopeuser/Prod-KE
```

Environment Setup & Working with Git

- **Git Bash Installation** https://git-scm.com/downloads
- **GitHub Tutorial** https://guides.github.com/activities/hello-world/
- **[Optional] Download and install virtual machine iss-vm (25 GB) if necessary**
  - » http://bit.ly/iss-vm



- **[If using iss-vm] Open terminal and run command:**
  - » cd ~/Desktop/workshop
  - » git clone https://github.com/telescopeuser/Prod-KE2018CI-1.git



iss-user@iss-vm: ~/Desktop/workshop

```
iss-user@iss-vm:~$ cd ~/Desktop/workshop
iss-user@iss-vm:~/Desktop/workshop$ git clone https://github.com/telescopeuser/P
rod-KE2018CI-1.git
```

Perceptron Exercise

- **Objectives**
  - » On completion of this workshop, students will have a good understanding on how the perceptron works
- **Exercise 1**
  - » AND/OR (choose one of them) operation
  - » open the data file: Prod-KE2018CI-1/workshop-1-perceptron-neural-net/01-perceptron/PER-Exer.xls
  - » do necessary edit on the data file to provide correct training patterns of AND/OR
  - » try different α:

    0.5, 0.7
  - » try different initial setting of weights:

    (0.5, 0.5, 0.5), (1, 1, 1)
  - » try two different sequence of patterns (decided by you)
  - » summarise the speed of convergence (how many runs of learning on the training patterns) for each trial (with particular α, initial weights, and sequence of patterns) above.
- **Exercise 2**
  - » Try XOR operation:

    with the best set of α and initial weights which you found from the above Exercise. Discuss your findings with classmates .

Perceptron Exercise Sheet

- **The following spreadsheet can be used to program a Perceptron network to learn to solve different logic computation problems, such as logical OR and AND (*the sheet below is for logical AND*)**

- **Using Excel to solve the logical problem.**

Bias Input X0 = +1 Alpha = 0.5

Input					Net Sum	Target	Pred.	Alpha	Weight Values		
X1	X2	1.0*W0	X1*W1	X2*W2	Input	Out	Out	* Error	W0	W1	W2
						Initial weight values -->			1.0	2.5	–0.5
0	0	1.0	0	0	1.0	0	1.0	–0.5	0.5	2.5	–0.5
0	1	0.5	0	–0.5	0	0					
1	0					0					
1	1					1					
0	0					0					
0	1					0					
1	0					0					
1	1					1					

Master of Technology

Unit 2/6: Computational Intelligence I

Workshop (1B):
MLFF/BP & RBF – Weka

Institute of Systems Science
National University of Singapore

Objectives

Objectives

» **On completion of this workshop, students will have a good understanding on the architectures and features of multiple layer perceptron (MLFF) with back propagation (BP) and radial basis function (RBF) networks and be able to**

- ♦ create BP and RBF networks
- ♦ specify parameters, train and test the networks
- ♦ interpret the networks' performance

Workshop (1B)

1. Background information

- We wish to train neural networks to predict the class of a flower in the well-known iris flower classification benchmark problem. We are given a data set of 150 samples (patterns) of Iris flowers each with 4 different feature variables representing petal length, petal width, sepal length and sepal width. Each pattern falls into one of the three classes.

2. What to do

- Construct neural network models. Use the given data file “iris.csv” to train and test the network. Note that the last column represents the ‘class label’ as the output in the given data set.

- Check the NN architecture, the number of training iterations, and network performance.

multiple layer perceptron (MLFF) + back propagation (BP) — Weka

- **Use** http://bit.ly/iss-vm **or Download Weka** http://www.cs.waikato.ac.nz/ml/weka/downloading.html

◆ Launch Weka Explorer from Desktop

◆ Open file… …iris.csv

◆ Check the information about instances, attributes, attribute type, etc.

◆ Click "Classify" tab, Choose the classifier as ***`weka.classifiers.functions.MultiLayerPerceptron`***

◆ Click the classifier chosen to launch the properties window

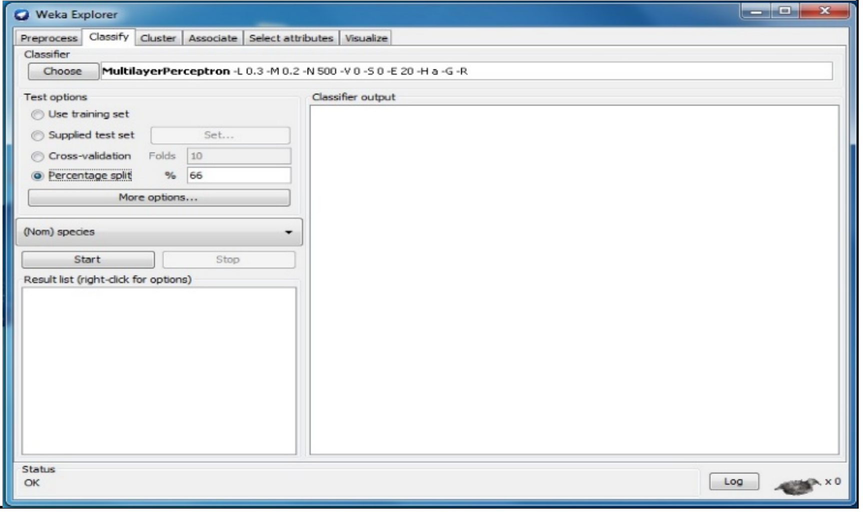
◆ Mouse over each option to understand its definition or click "More" button to get more explanation

◆ Change GUI to "True" -- (to bring up a GUI interface)

◆ Click "OK" to accept the changes

◆ Set "Test Options" as "Percentage split 66%"

◆ Click "Start" to train a BP network



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R

○ Use training set
○ Supplied test set Set...
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) species

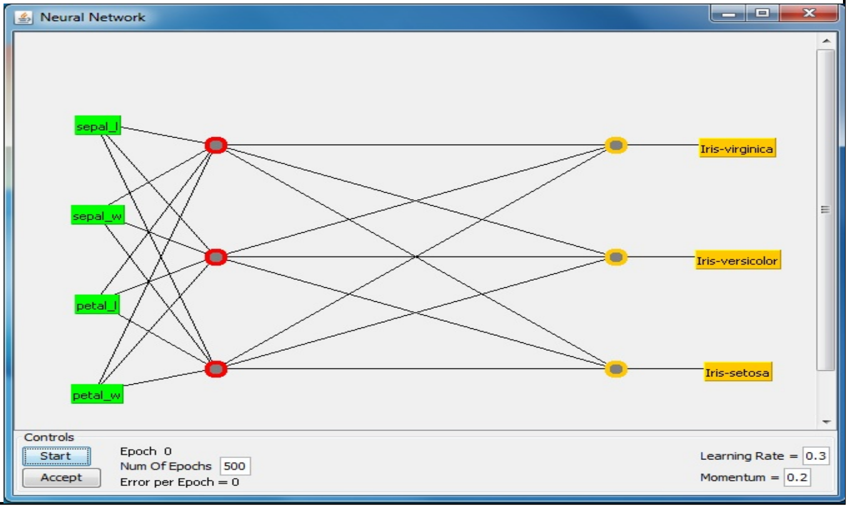
Status: OK

multiple layer perceptron (MLFF) + back propagation (BP) — Weka

- A BP network is auto-built according to the architecture setting
- Modify the network if necessary, such as adding/removing nodes and connections
- Click “Start” to train the NN
- After training is done, click “Accept” to check the model information (nodes, connections and weights) in “classifier output”
- Click “Accept” again to get the testing results in “classifier output”
- Browse the results to find out model accuracy, confusion matrix, etc.

- Experiment with different parameters such as hidden layers, learning rate, epoch, etc.
- Compare the model performance

- Right click on the model item in the Result List and select “save model” to save the model



Epoch 0
Num Of Epochs 500
Error per Epoch = 0
Learning Rate = 0.3
Momentum = 0.2

radial basis function (RBF) network — Weka

- Change the classifier as ***`weka.classifiers.functions.RBFNetwork`***
- Click the classifier chosen to launch the properties window
- Mouse over each option to understand its definition or click “More” button to get more explanation

- Set “Test Options” as “Percentage split 66%”
- Click “Start” to train a RBF network

- Experiment with different ***`numCluster`*** and compare the model performance
- Save the model