



More Data Mining with Weka

Class 5 – Lesson 1

簡單神經網路

Simple neural networks

Ian H. Witten

Department of Computer Science University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 5.1: 簡單神經網路

Class 1 探索Weka的介面；處理大數據

Class 2 離散以及文本分類

Class 3 分類規則、關聯規則、聚類

Class 4 選擇屬性以及計算成本

Class 5 神經網路，學習曲線和表現優化

Lesson 5.1 簡單神經網路

Lesson 5.2 Multilayer Perceptrons

Lesson 5.3 Learning curves

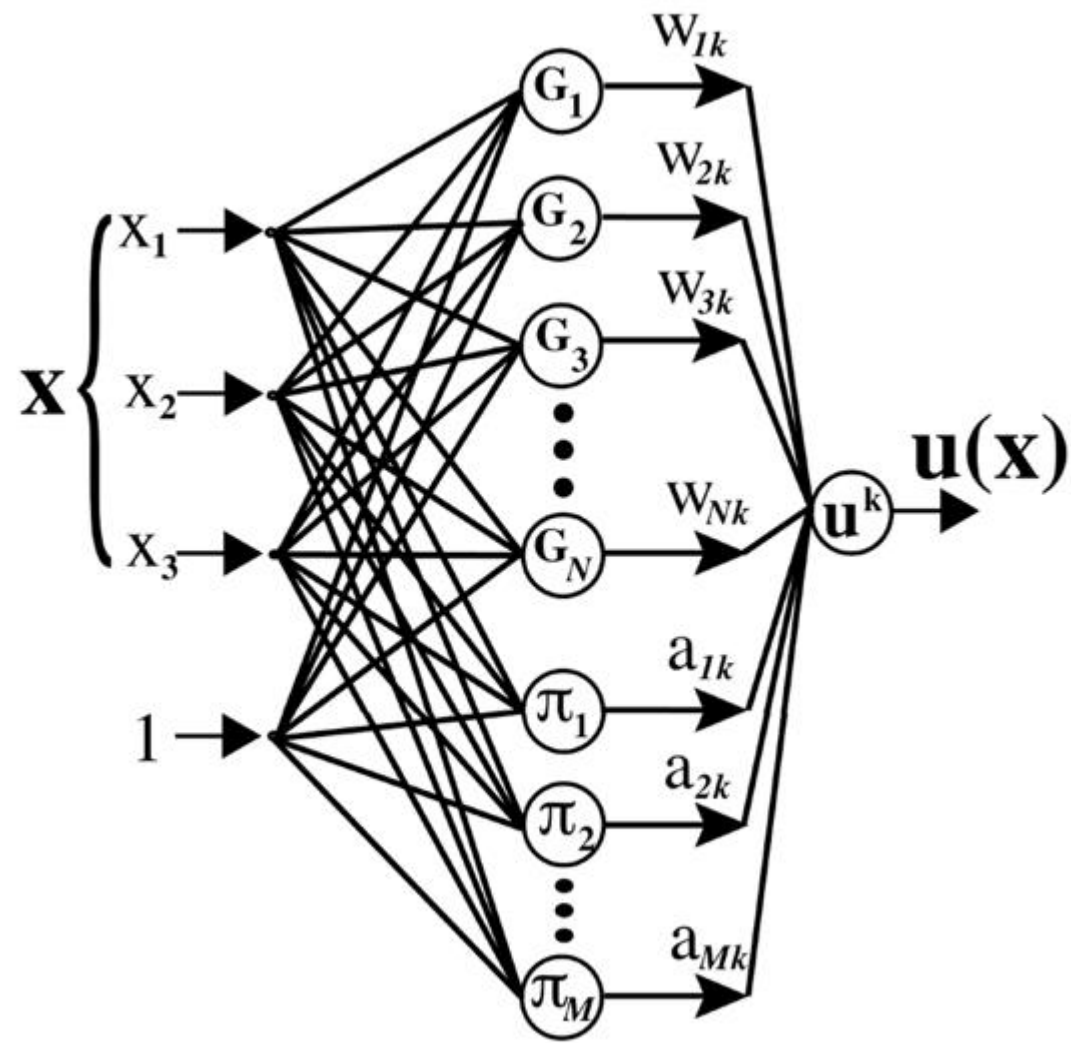
Lesson 5.4 Performance optimization

Lesson 5.5 ARFF and XRFF

Lesson 5.6 Summary

Lesson 5.1: 簡單神經網路

現在，很多人熱衷於神經網路
確切的名字是... intelligence(智慧)!

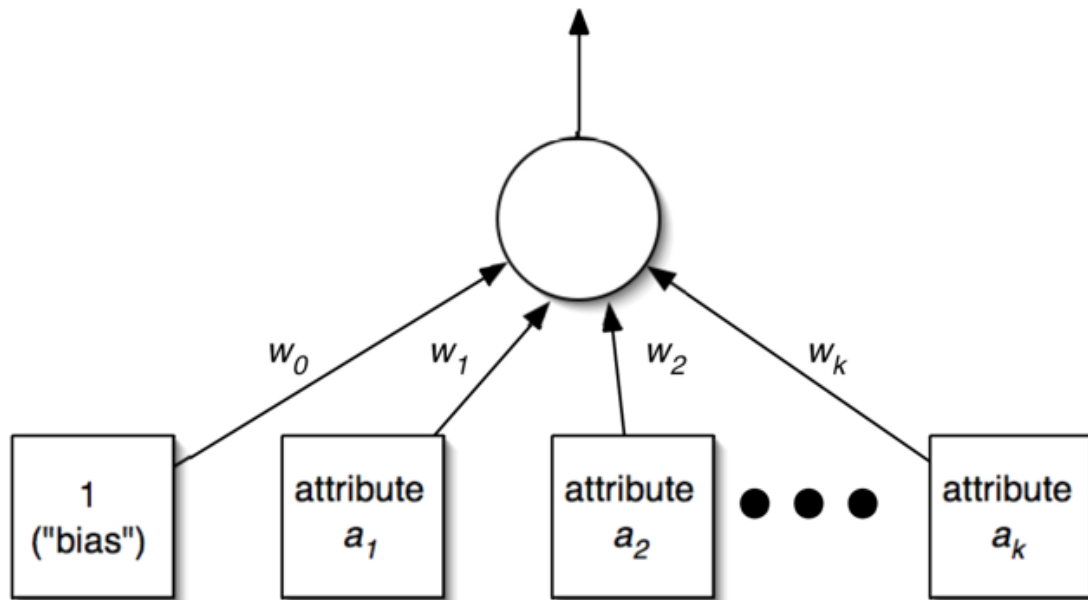


Lesson 5.1: 簡單神經網路

感知器(Perceptron): 最簡單的形式

- ❖ 使用屬性的線性組合為兩類的資料集分類
- ❖ 對於測試實例 \mathbf{a} ,
- ❖ 如果 $x > 0$ 則為類別 1, 如果 $x < 0$ 則為類別 2
 - 最適用於數值屬性

$$x = w_0 + w_1 a_1 + w_2 a_2 + \dots + w_k a_k = \sum_{j=0}^k w_j a_j$$



將所有權值設置為零，直到訓練數據中的所有實例都被正確分類

對於訓練數據中的每個實例 i :

如果 i 分類不正確:

如果 i 屬於第一類，把它加到權值向量上

否則，在權值向量中將它減去

感知器收斂定理

- 如果循環處理訓練數據(可能很多次)，它就會收斂。
- 只要問題是“線性可分(Linearly separable)”

Lesson 5.1: 簡單神經網路

線性決策邊界(Linear decision boundaries)

- ❖ 複習Data Mining with Weka, lesson 4.5所提到的支持向量機(Support Vector Machines)
 - 也受限於線性決策邊界
 - 但是使用“核技巧(Kernel trick)”可以得到更覆複雜的邊界(當時並未解釋)
- ❖ 感知器可以使用同樣的技巧來獲得非線性邊界

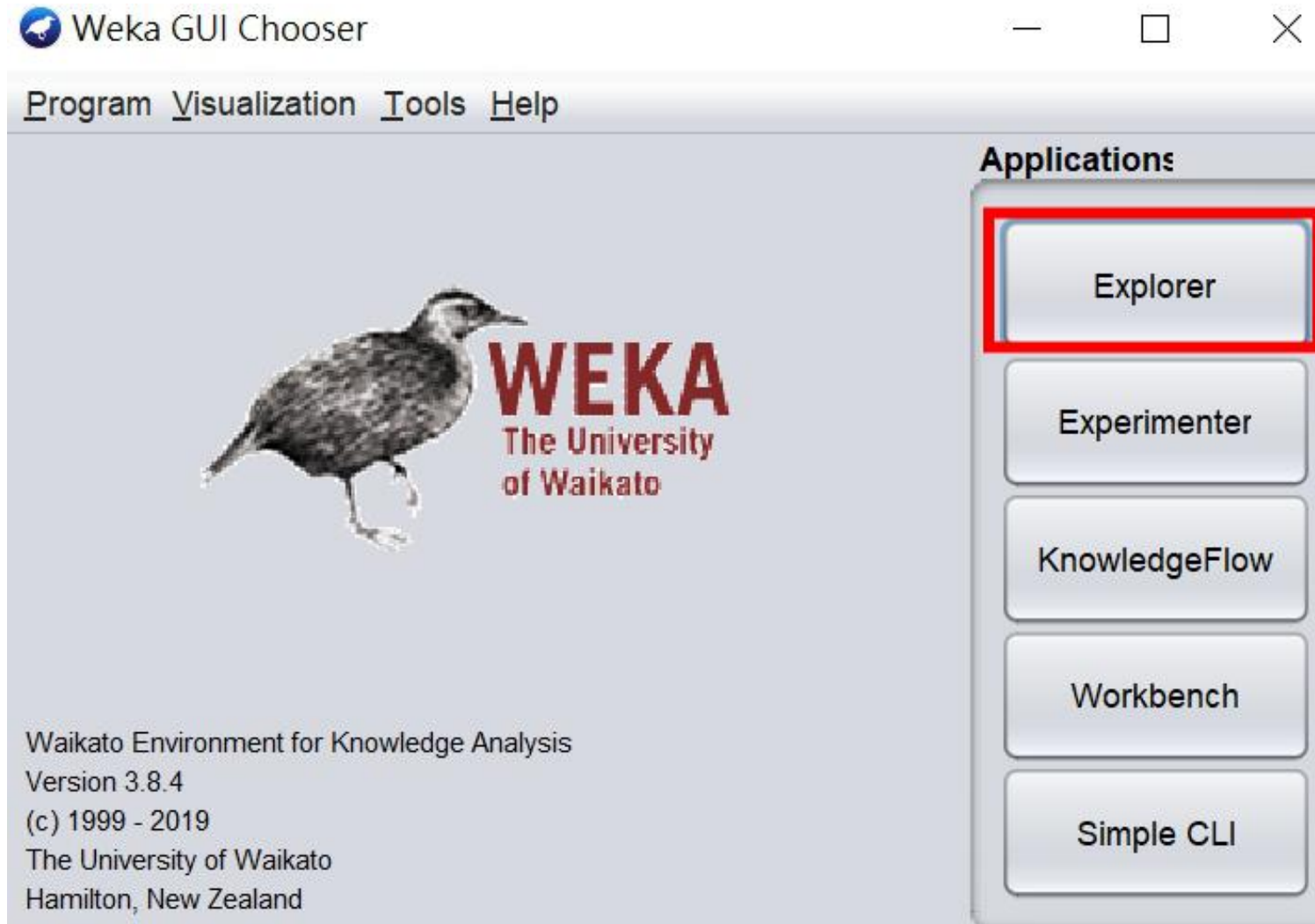
Weka中的Voted perceptron

- ❖ 儲存所有的權重向量，並讓它們對測試實例進行投票
 - 根據它們的“生存(survival)”時間給予它們權重
- ❖ 聲稱具有許多支持向量機的優點
- ❖ ...更快，更簡單，幾乎一樣好

Lesson 5.1: 簡單神經網路

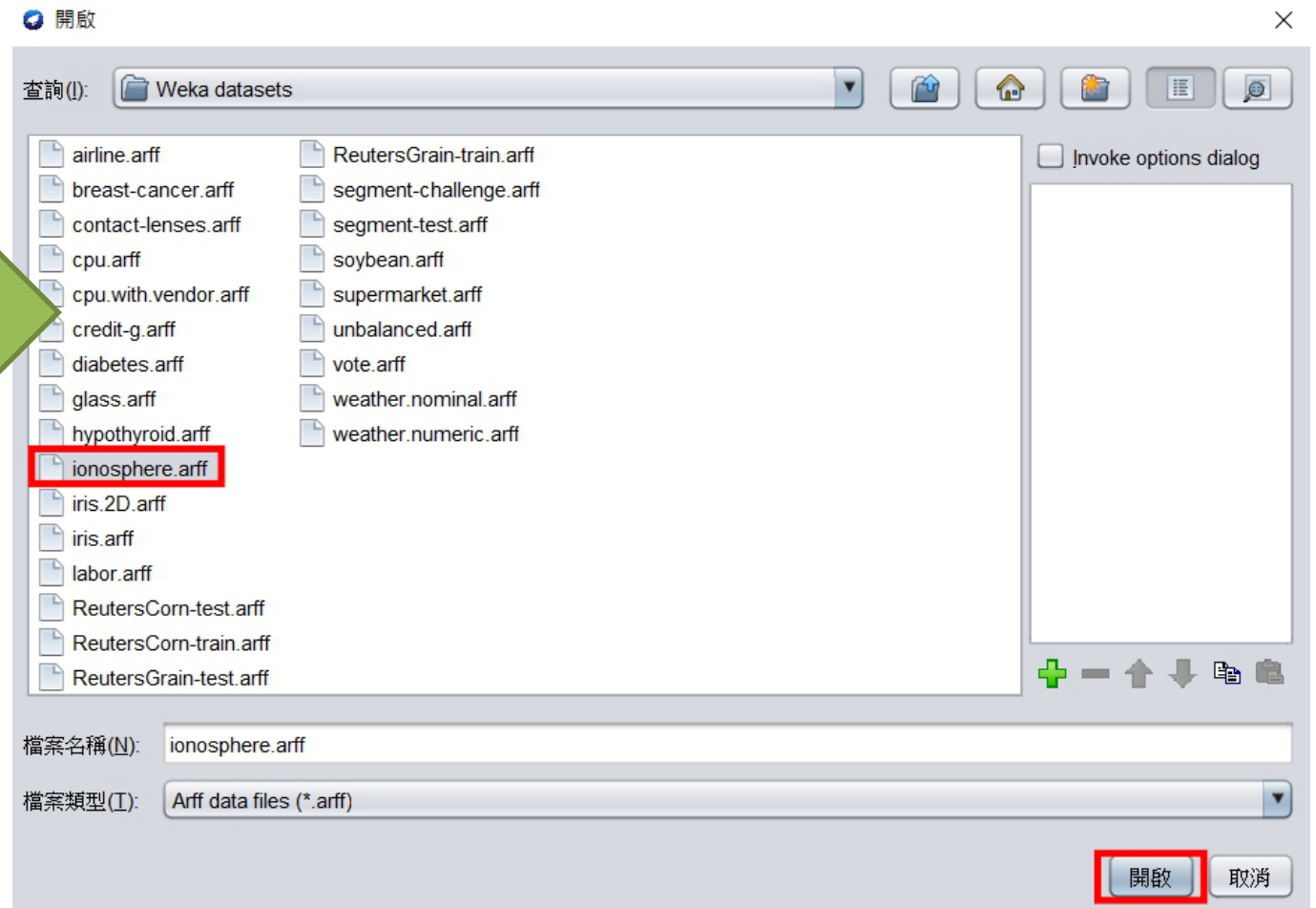
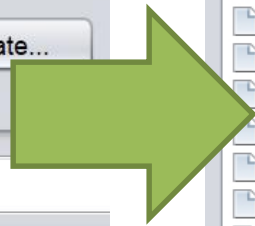
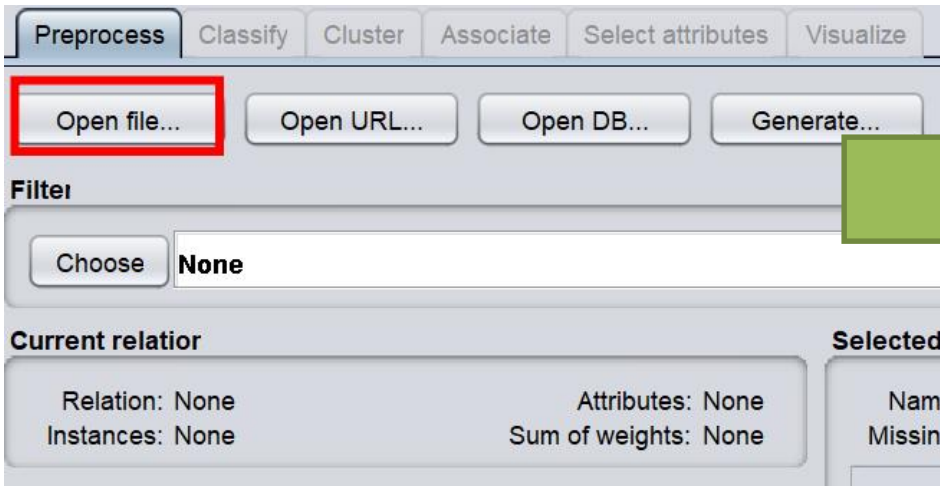
我們試著運行Weka中的Voted perceptron。

1. 開啟Weka的Explorer。



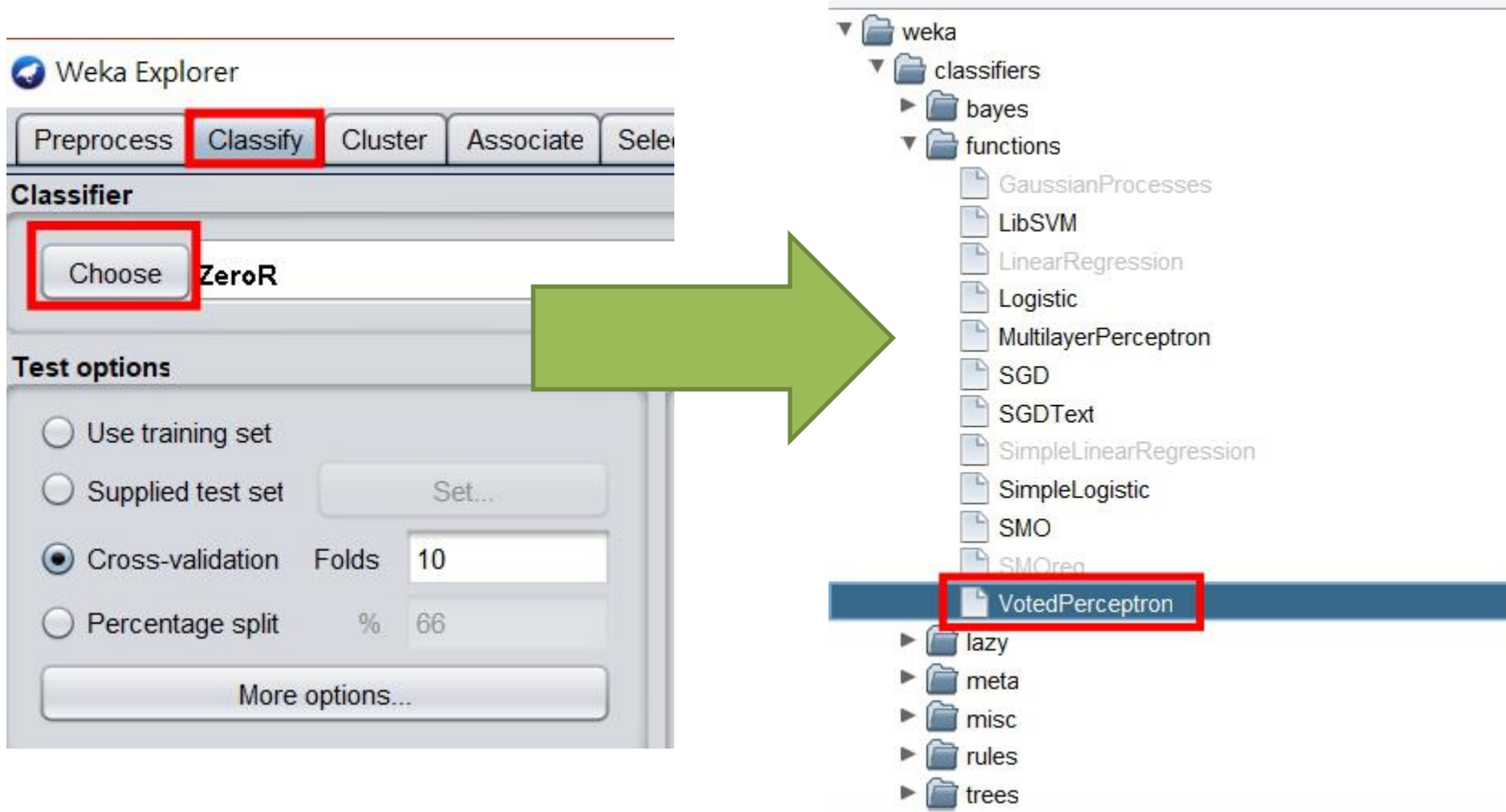
Lesson 5.1: 簡單神經網路

2. 左鍵點擊Open file...開啟右圖視窗，進入自行複製的Weka datasets資料夾，左鍵單擊ionosphere.arff的檔案後，再:left鍵單擊下方「開啟」按鈕以載入此檔案



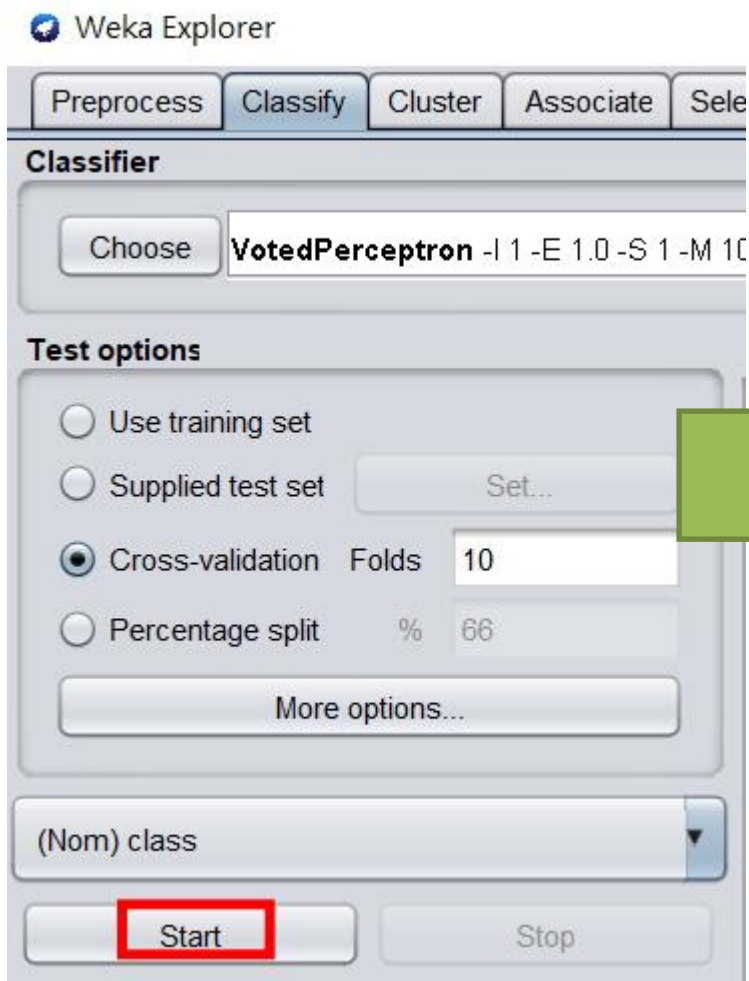
Lesson 5.1: 簡單神經網路

3. 切換到Classify面板點選Choose鈕，在出現的選單中左鍵單擊functions資料夾下的VotedPerceptron分類器



Lesson 5.1: 簡單神經網路

4.左鍵單擊Start按鈕，執行結果如右圖：得到86.0399%準確率。



Classifier output

Time taken to build model: 0.23 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	302	86.0399 %
Incorrectly Classified Instances	49	13.9601 %
Kappa statistic	0.6769	
Mean absolute error	0.1396	
Root mean squared error	0.3728	
Relative absolute error	30.3131 %	
Root relative squared error	77.7099 %	
Total Number of Instances	351	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
	0.667	0.031	0.923	0.667	0.774	0.696	0.837	0.775
	0.969	0.333	0.838	0.969	0.899	0.696	0.840	0.850
Weighted Avg.	0.860	0.225	0.869	0.860	0.854	0.696	0.839	0.823

=== Confusion Matrix ===

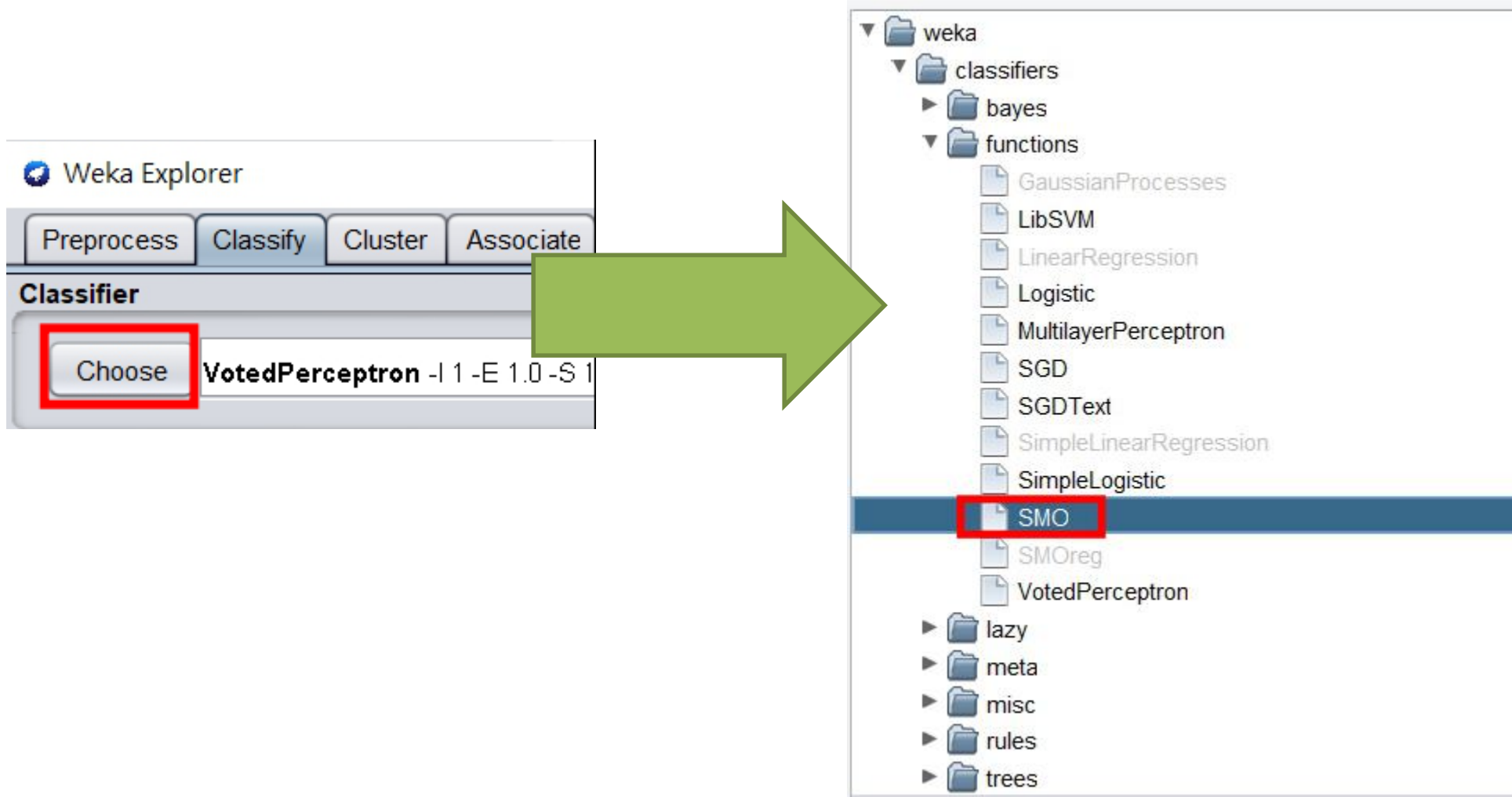
a b <-- classified as

84	42		a = b
7	218		b = g

Lesson 5.1: 簡單神經網路

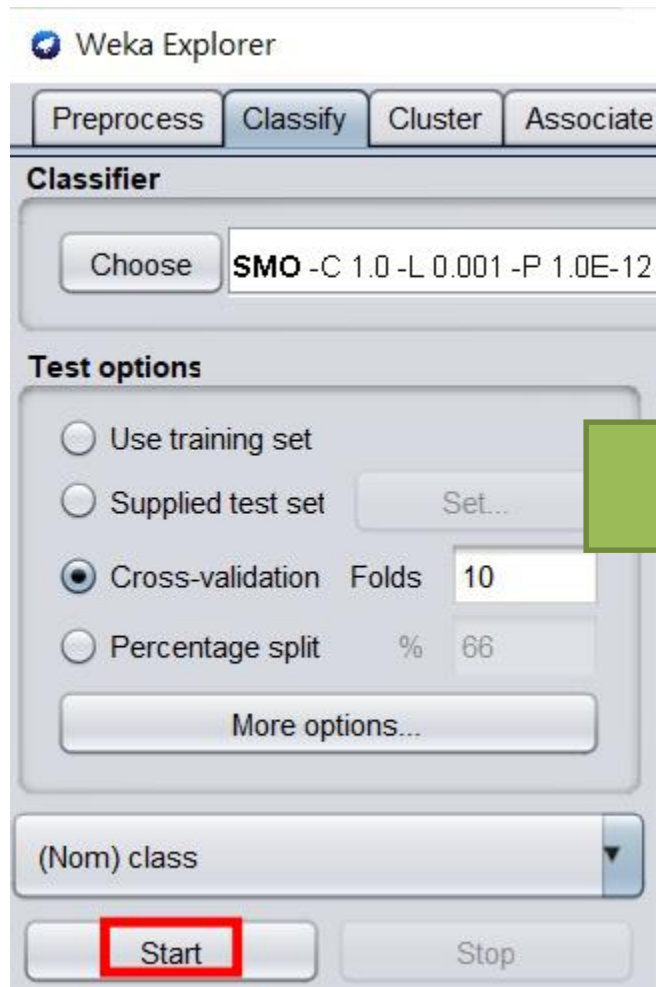
接著我們試著運行SMO分類器。

1. 在Classify面板左鍵單擊Choose按鈕(左圖紅框處)，並在彈出的選單中以左鍵單擊functions資料夾下的SMO分類器。



Lesson 5.1: 簡單神經網路

2. 左鍵單擊Start按鈕，執行結果如右圖：得到88.604%準確率。



Classifier output

Time taken to build model: 0.36 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	311	88.604 %
Incorrectly Classified Instances	40	11.396 %
Kappa statistic	0.7406	
Mean absolute error	0.114	
Root mean squared error	0.3376	
Relative absolute error	24.7463 %	
Root relative squared error	70.3666 %	
Number of Instances	351	

Weighted Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
	0.738	0.031	0.930	0.738	0.823	0.751	0.853	0.780
	0.969	0.262	0.869	0.969	0.916	0.751	0.853	0.861
Weighted Avg.	0.886	0.179	0.891	0.886	0.883	0.751	0.853	0.832

=== Confusion Matrix ===

a b <-- classified as

93	33	a = b
7	218	b = g

Lesson 5.1: 簡單神經網路

VotedPerceptron有多好?

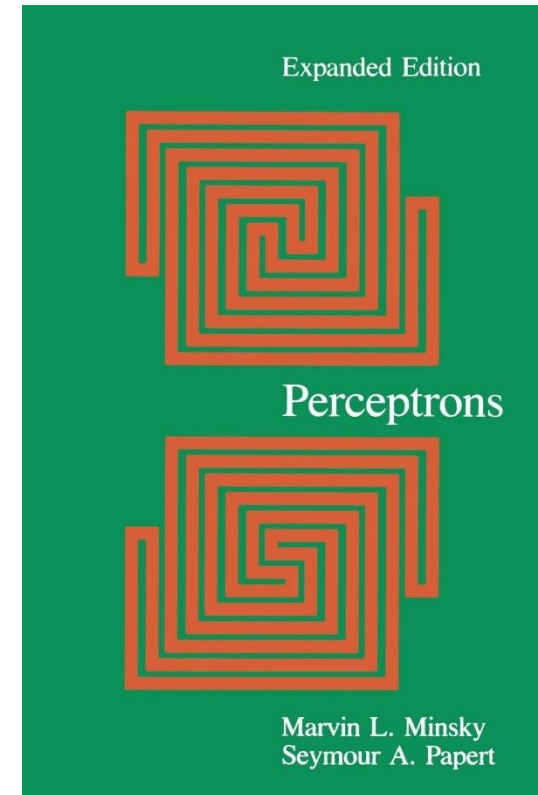
	<i>VotedPerceptron</i>	<i>SMO</i>
電離層資料集 ionosphere.arff	86%	89%
德國人信用資料集 credit-g.arff	70%	75%
乳癌資料集 breast-cancer.arff	71%	70%
糖尿病資料集 diabetes.arff	67%	77%

VotedPerceptron有比較快嗎? ... 有的

Lesson 5.1: 簡單神經網路

感知器的歷史

- ❖ 1957: 基本感知器的演算法
 - 源於大腦的工作原理
 - 是“A perceiving and recognizing automaton”的縮略
 - Rosenblatt 的書: *“Principles of neurodynamics: Perceptrons and the theory of brain mechanisms”*
- ❖ 1970: 感知器突然不再受人關注
 - 因為Minsky和Papert發表的書: *“Perceptrons”*
- ❖ 1986: 被更名為 “連接機制(connectionism)”
 - Rumelhart 和McClelland發表的書: *“Parallel distributed processing”*
 - 有人認為人工神經網路模擬了大腦的機能
- ❖ 多層感知機
 - 非線性決策邊界
 - 反向傳播演算法(Backpropagation algorithm)

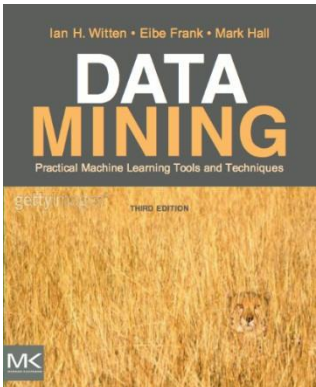


Lesson 5.1: 簡單神經網路

- ❖ 基本的感知器算法:實現線性決策邊界
 - 如同回歸分類(classification-by-regression)
 - 應用於數值屬性
 - 迭代算法(iterative algorithm),取決於訓練實例的順序(order dependent)
- ❖ 作者的MSc碩士論文(1971)介紹了感知器的一個簡單改進!
- ❖ 最近的改進(1999):
 - 使用核技巧得到更複雜的邊界
 - 這種投票感知器策略包含多個權向量和投票

課程文本

- ❖ Section 4.6 *Linear classification using the Perceptron*
- ❖ Section 6.4 *Kernel Perceptron*





More Data Mining with Weka

Class 5 – Lesson 2

多層感知機

Multilayer Perceptrons

Ian H. Witten

Department of Computer Science University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 5.2: 多層感知機

Class 1 探索Weka的介面；處理大數據

Class 2 離散以及文本分類

Class 3 分類規則、關聯規則、聚類

Class 4 選擇屬性以及計算成本

Class 5 神經網路，學習曲線和表現優化

Lesson 5.1 簡單神經網路

Lesson 5.2 多層感知機

Lesson 5.3 Learning curves

Lesson 5.4 Performance optimization

Lesson 5.5 ARFF and XRFF

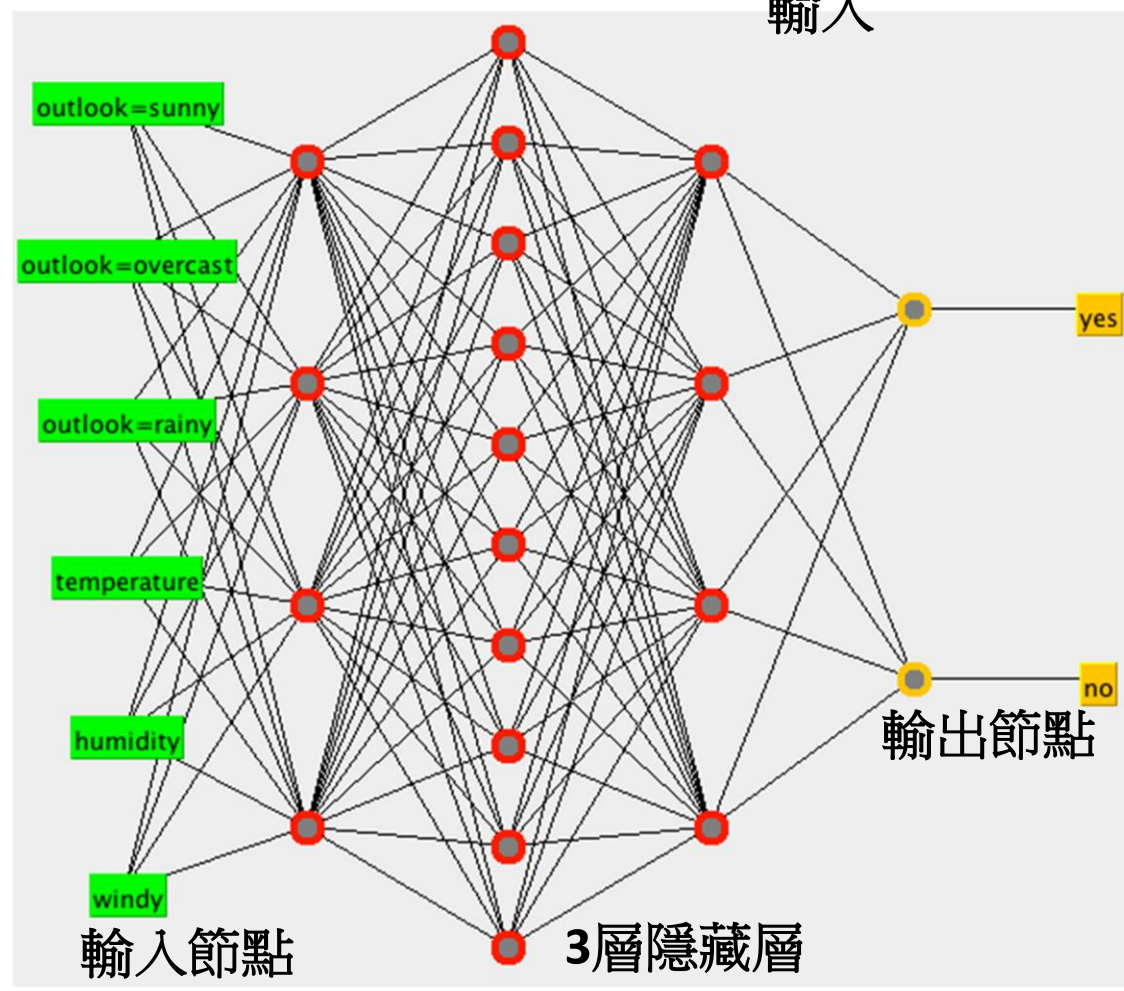
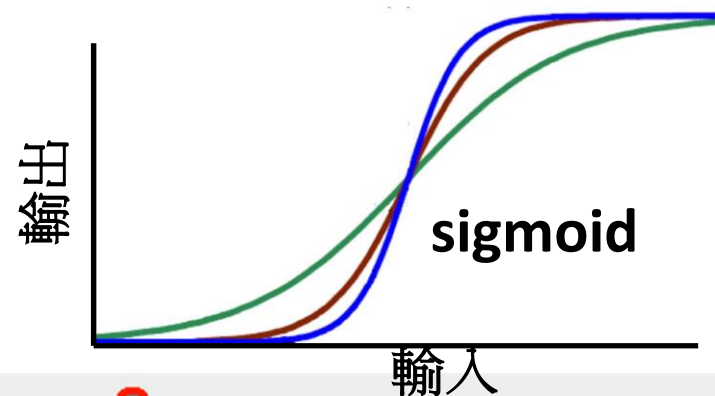
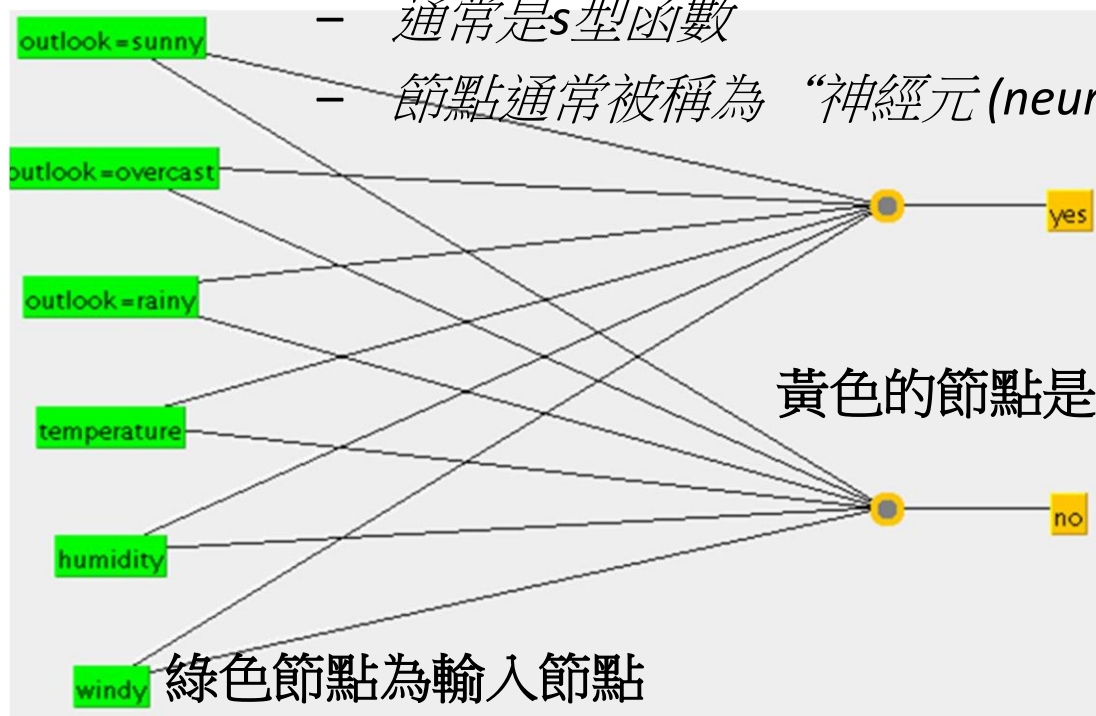
Lesson 5.6 Summary

Lesson 5.2: 多層感知機

感知機的網路

- ❖ 輸入層(Input layer),隱藏層(hidden layer(s)),以及輸出層(output layer)
- ❖ 每個連接點都有權重(一個數字)
- ❖ 每個節點執行其輸入和結果閾值的權重和

- 通常是s型函數
- 節點通常被稱為“神經元(neurons)”



Lesson 5.2: 多層感知機

每個感知器中有多少層，多少節點？

- ❖ 輸入層:每個屬性有一個節點(屬性是數值型或二值型的)
- ❖ 輸出層:每個類別有一個節點(如果類別是數值型就只有1層)
- ❖ 多少的隱藏層? — 大問題#1
- ❖ 0層隱藏層:
 - 標準的感知器演算法
 - 如果資料是線性可分則適用
- ❖ 1層隱藏層:
 - 適用於單一凸區域的決策空間
- ❖ 2層隱藏層:
 - 可以生成任意的決策邊界
- ❖ 隱藏層多大? — 大問題#2
 - 通常選擇在輸入層和輸出層之間
 - 常見啟發式:輸入和輸出層的平均值(Weka的預設值)

Lesson 5.2: 多層感知機

權重多少才好?

- ❖ 得通過訓練數據集來學習權重
- ❖ 使用最陡下降迭代地使誤差最小化
- ❖ 梯度是用“backpropagation”算法確定的
- ❖ 通過將梯度乘以“學習率”，再加上之前的權重變化乘以“動量”來計算權重變化：

$$W_{\text{下一個}} = W + \Delta W$$

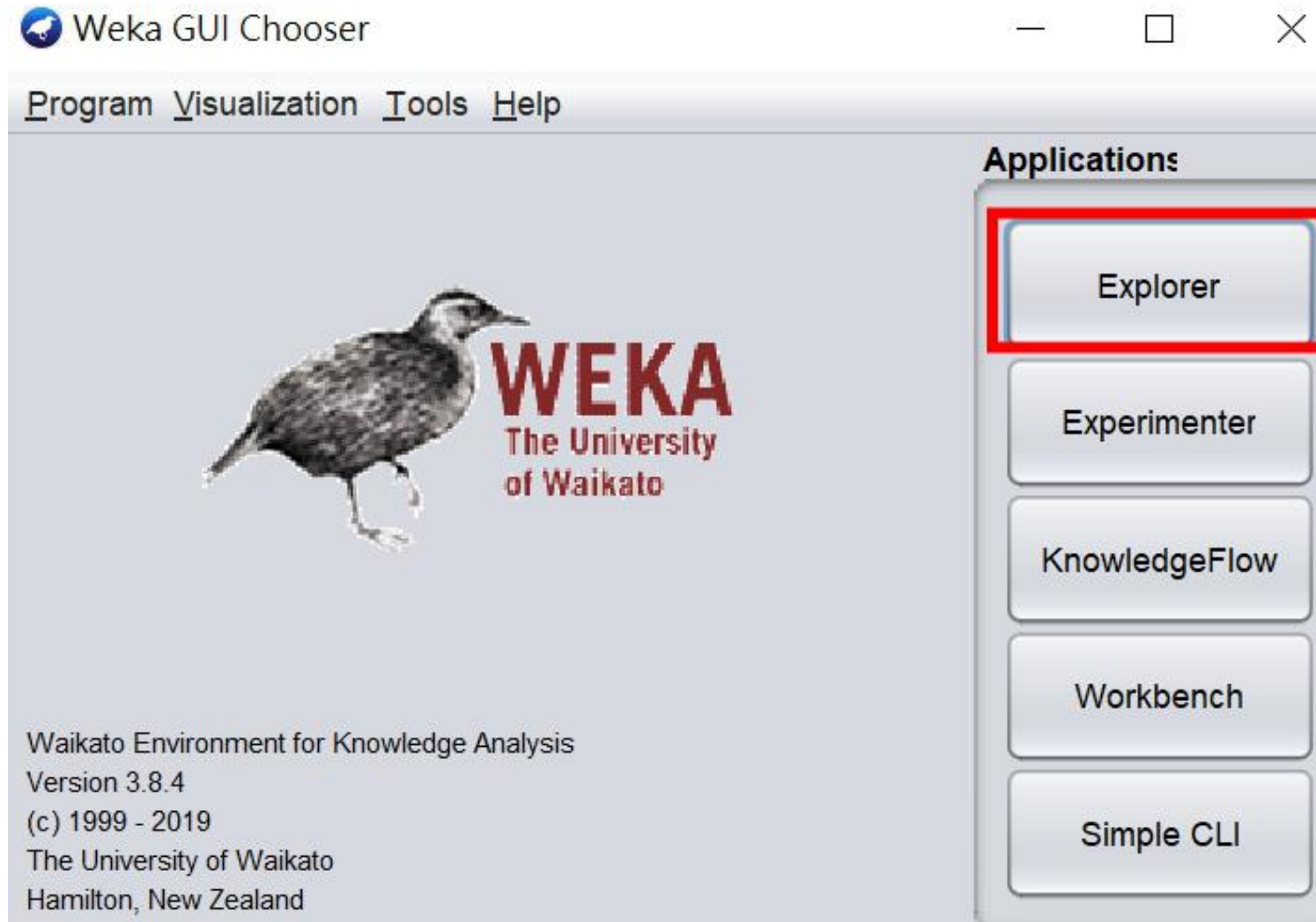
$$\Delta W = -\text{學習率} \times \text{梯度} + \text{動量} \times \Delta W_{\text{前一個}}$$

可以得到很好的結果

- ❖ 經常涉及(很多)實驗
 - 隱藏層的數量和大小
 - 學習速度和動量的價值

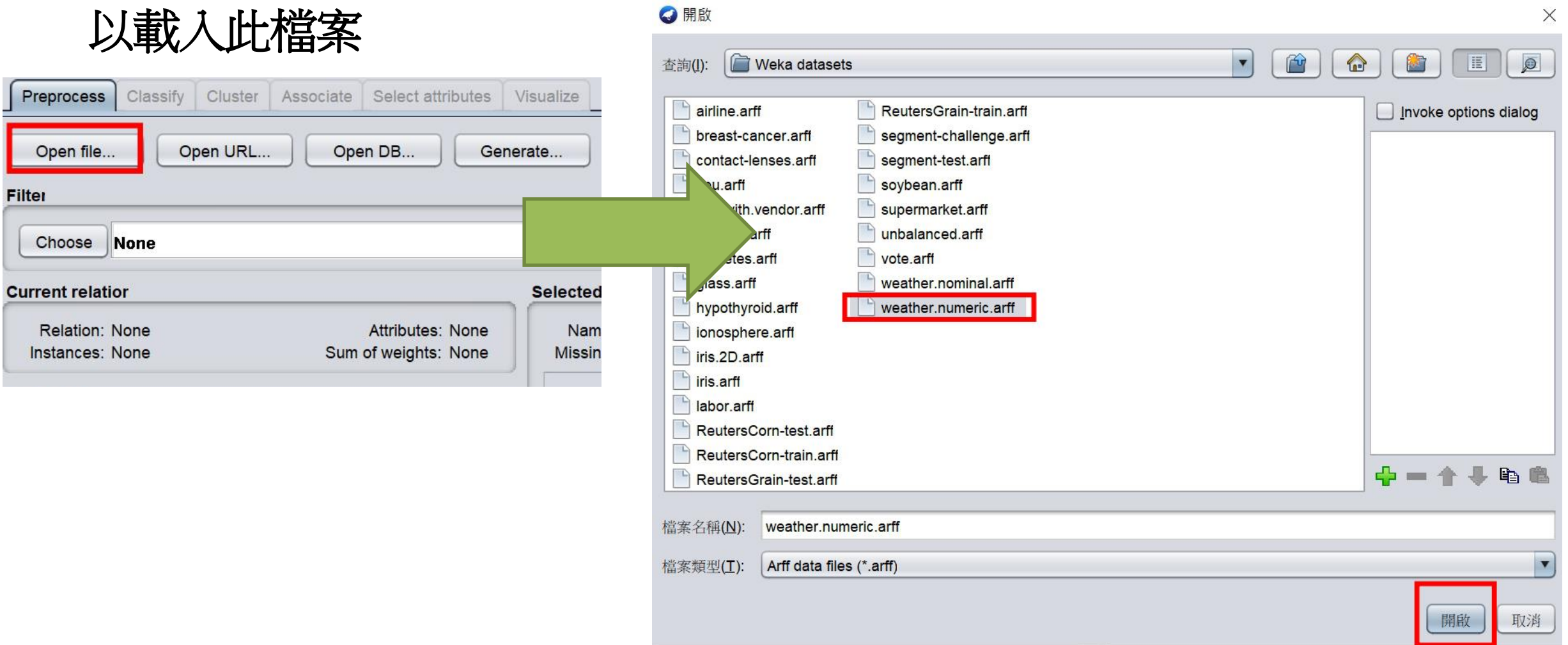
Lesson 5.2: 多層感知機

1. 開啟Weka的Explorer



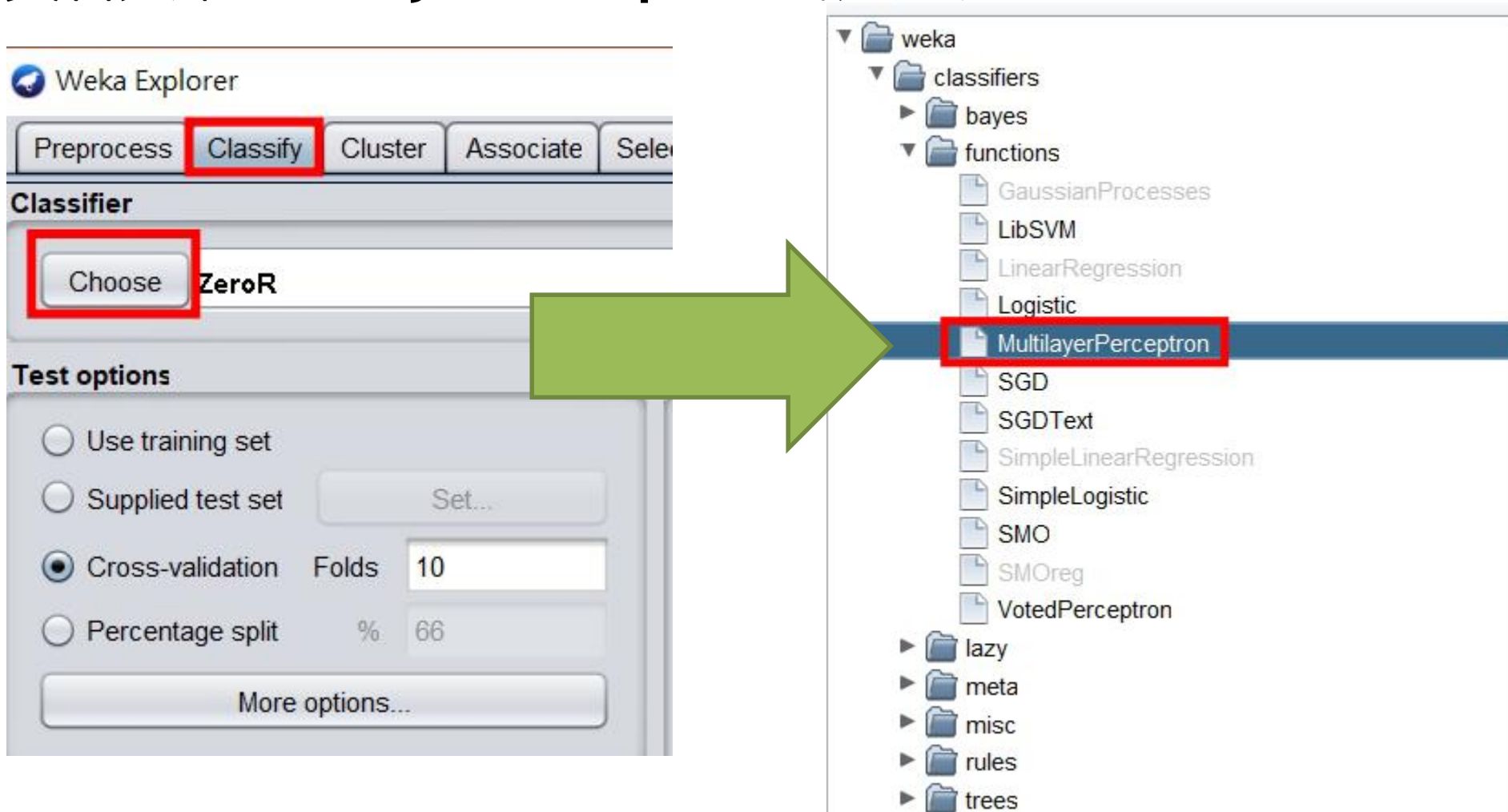
Lesson 5.2: 多層感知機

2. 左鍵點擊Open file...開啟右圖視窗，進入自行複製的Weka datasets資料夾，左鍵單擊weather.numeric.arff的檔案後，再:left鍵單擊下方開啟按鈕以載入此檔案



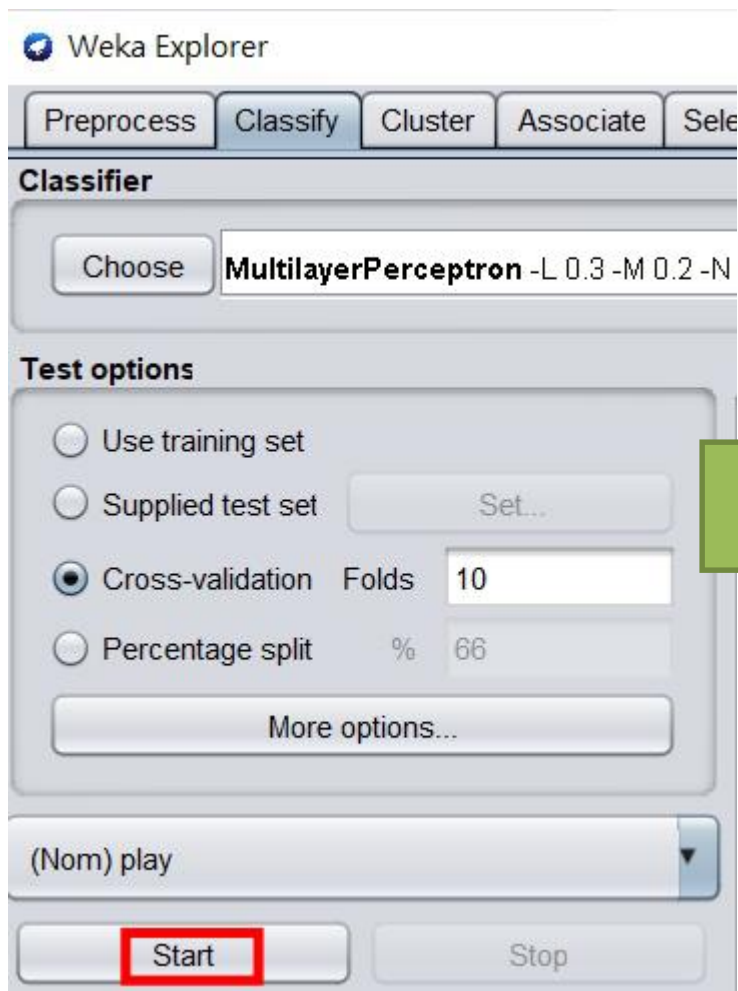
Lesson 5.2: 多層感知機

3. 切換到Classify面板點選Choose鈕，在出現的選單中左鍵單擊functions資料夾下MultilayerPerceptron的分類器



Lesson 5.2: 多層感知機

4.左鍵單擊Start按鈕，執行結果如右圖：得到78.5714%的正確率



Classifier output

Time taken to build model: 0.34 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	11	78.5714 %
Incorrectly Classified Instances	3	21.4286 %
Kappa statistic	0.5116	
Absolute error	0.265	
Squared error	0.4627	
Absolute error	55.6497 %	
Relative squared error	93.7923 %	
Total number of Instances	14	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
	0.889	0.400	0.800	0.889	0.842	0.519	0.733	0.857
	0.600	0.111	0.750	0.600	0.667	0.519	0.733	0.589
Weighted Avg.	0.786	0.297	0.782	0.786	0.779	0.519	0.733	0.761

=== Confusion Matrix ===

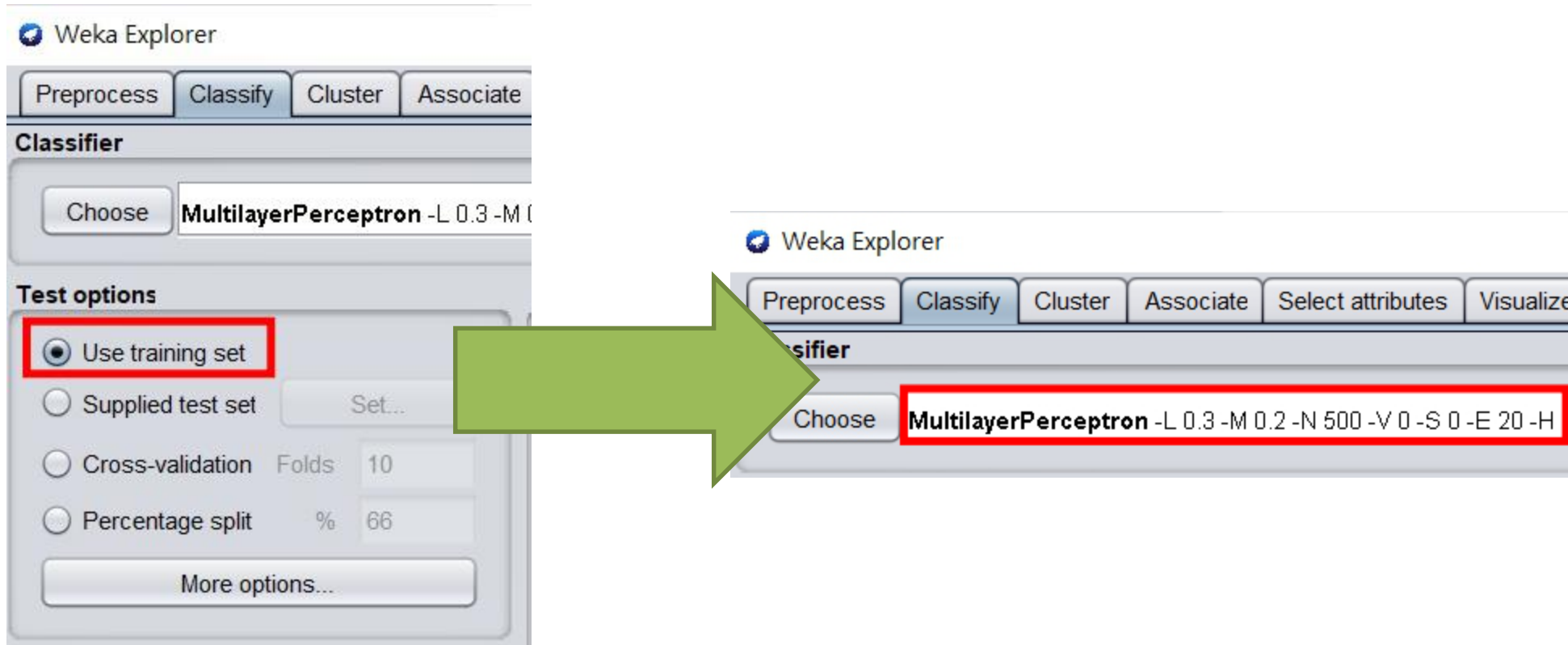
a b <-- classified as

8 1 | a = yes

2 3 | b = no

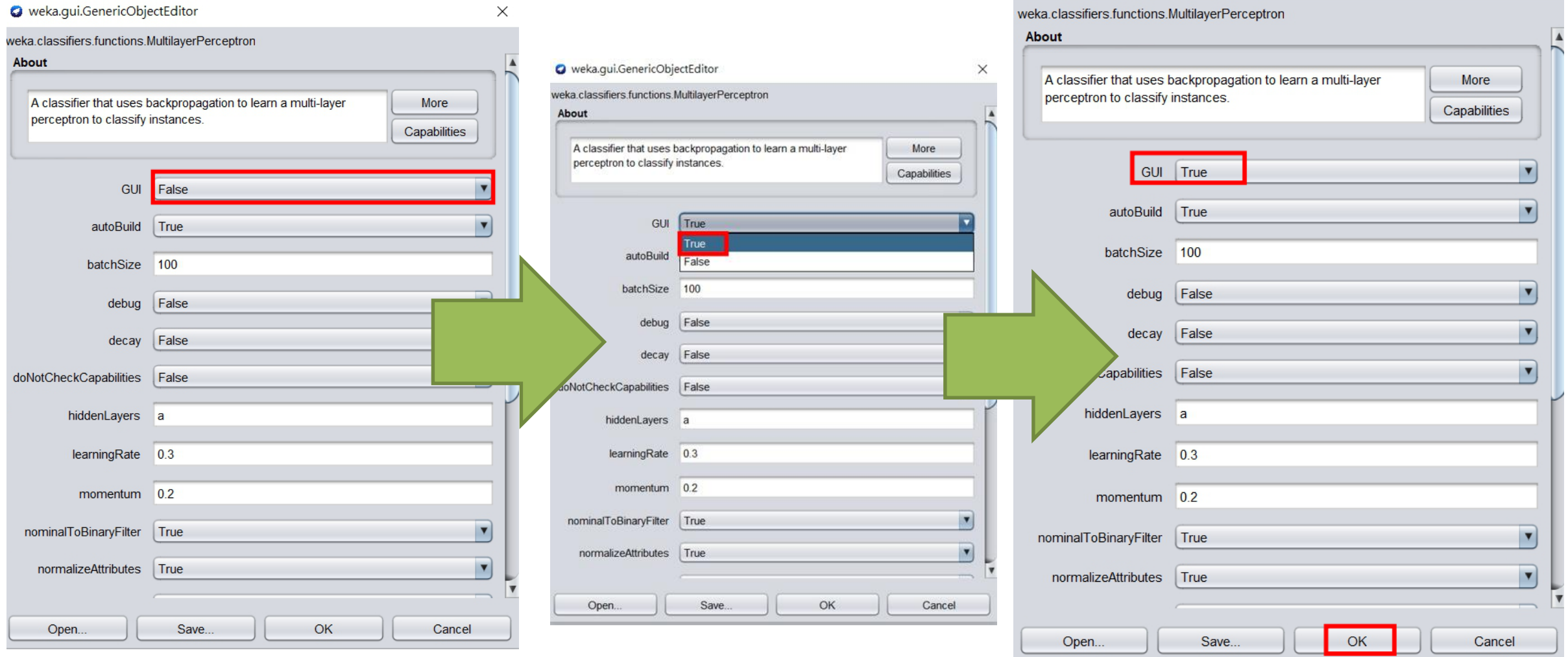
Lesson 5.2: 多層感知機

5. 左鍵單擊Test options中的Use training set前方圓圈，接著左鍵單擊分類器名稱(右圖紅框處)開啟配置視窗。



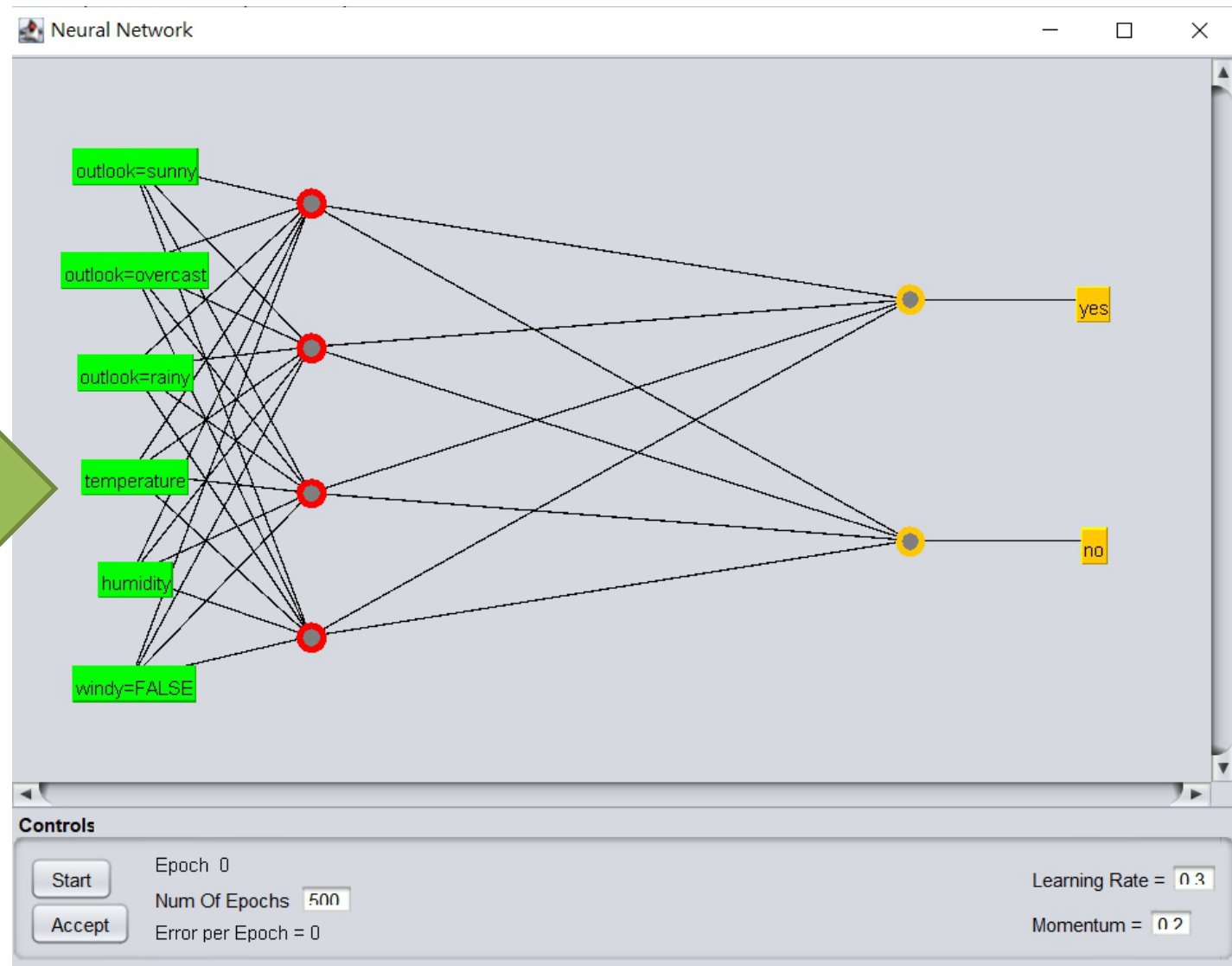
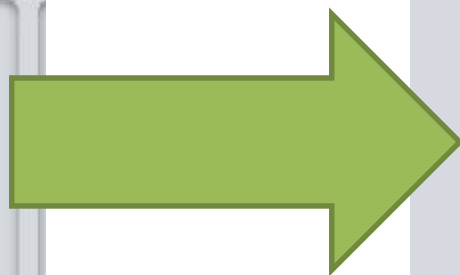
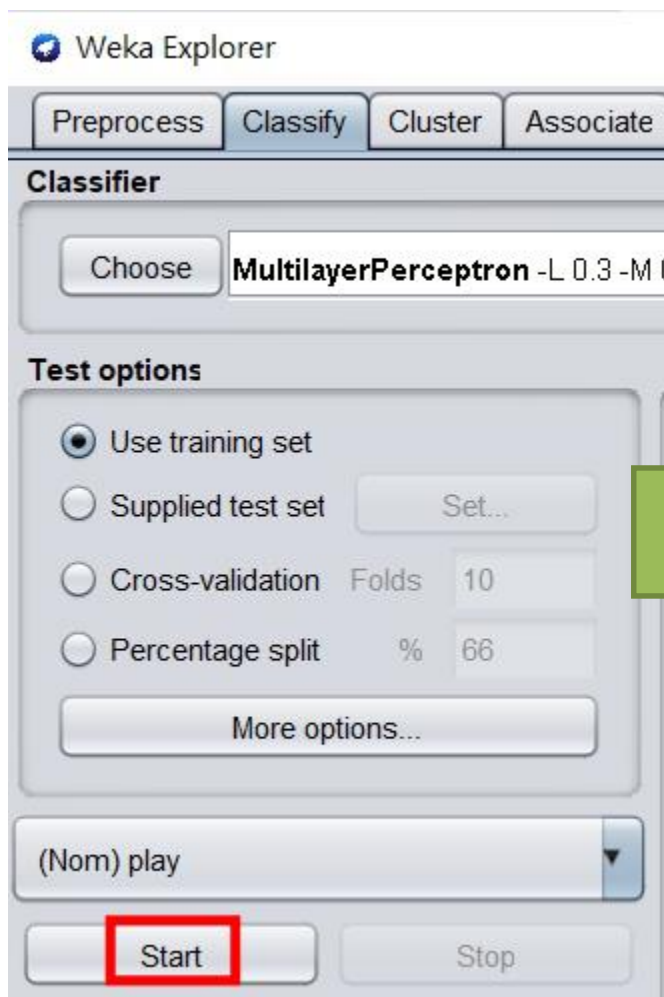
Lesson 5.2: 多層感知機

6. 左鍵單擊參數GUI右方橫條，在選單中左鍵單擊True選項，然後左鍵單擊下方OK按鈕。



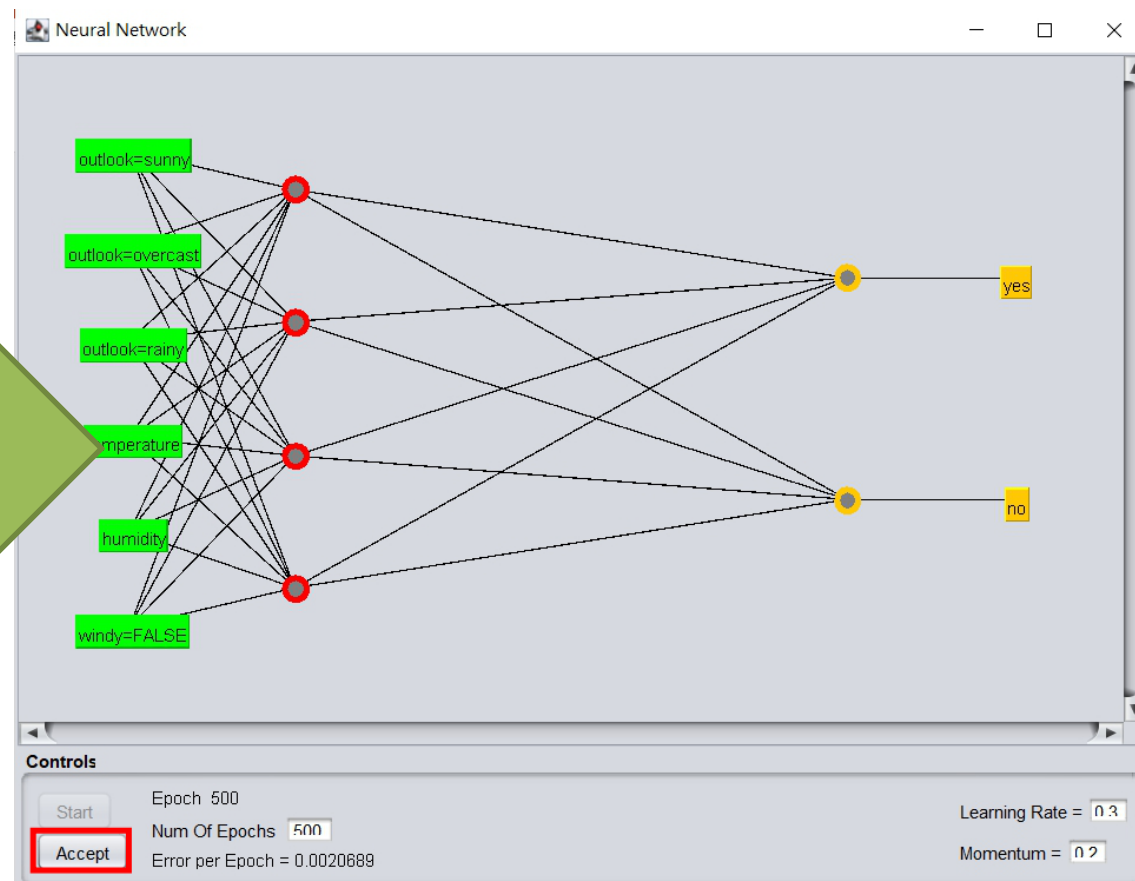
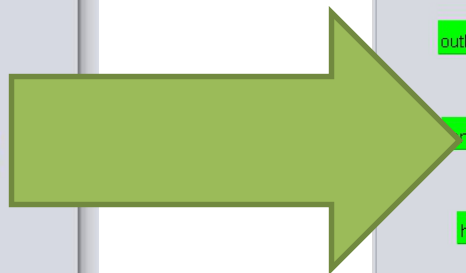
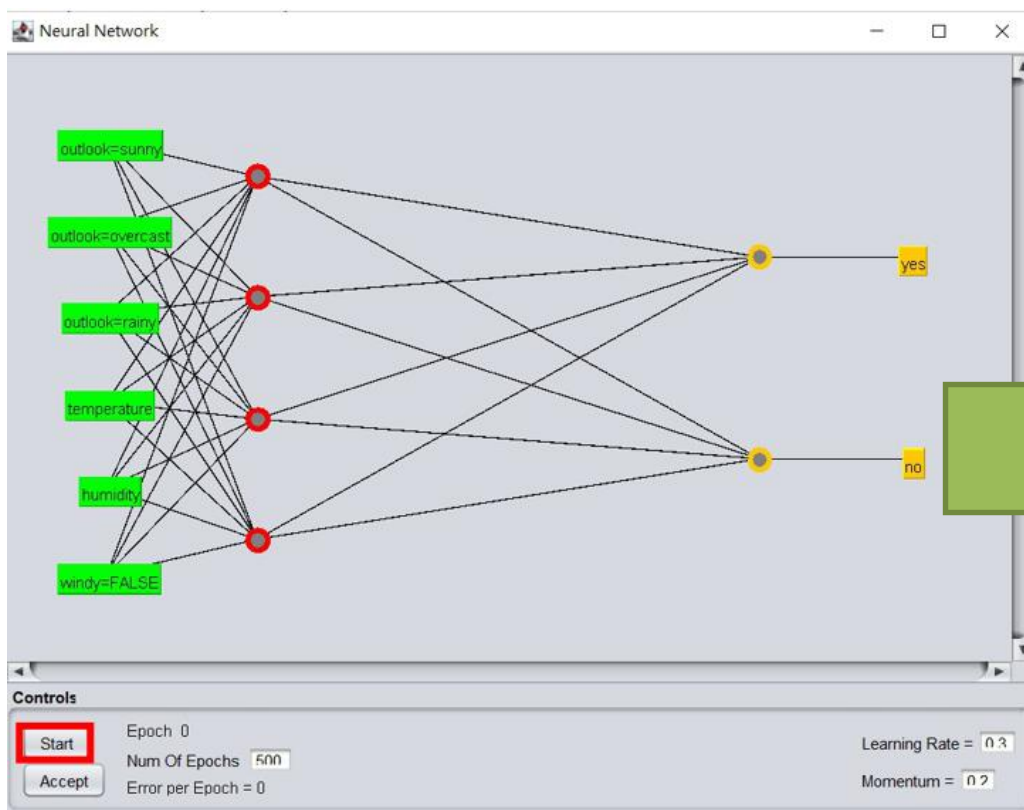
Lesson 5.2: 多層感知機

7. 左鍵單擊**Start**按鈕，執行結果如右圖:得到網路圖。



Lesson 5.2: 多層感知機

8. 左鍵單擊Start按鈕，接著按下Accept按鈕回到Classify面板。



Lesson 5.2: 多層感知機

得到的輸出數據如下。

```
Classifier output

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances      14          100 %
Incorrectly Classified Instances    0           0 %
Kappa statistic                    1
Mean absolute error                 0.036
Root mean squared error             0.0454
Relative absolute error             7.7533 %
Root relative squared error         9.4618 %
Total Number of Instances          14

=== Detailed Accuracy By Class ===

              TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
              1.000    0.000    1.000     1.000    1.000     1.000    1.000    1.000
              1.000    0.000    1.000     1.000    1.000     1.000    1.000    1.000
Weighted Avg.  1.000    0.000    1.000     1.000    1.000     1.000    1.000    1.000

=== Confusion Matrix ===

a b  <-- classified as
9 0 | a = yes
0 5 | b = no
```

Lesson 5.2: 多層感知機

MultilayerPerceptron的表現

- ❖ 在數值的天氣資料上得到79%準確率!
- ❖ (J48, NaiveBayes都是64%, SMO得到57%, Ibk得到79%)
- ❖ 現實問題中，多層感知器性能很好，但是速度慢

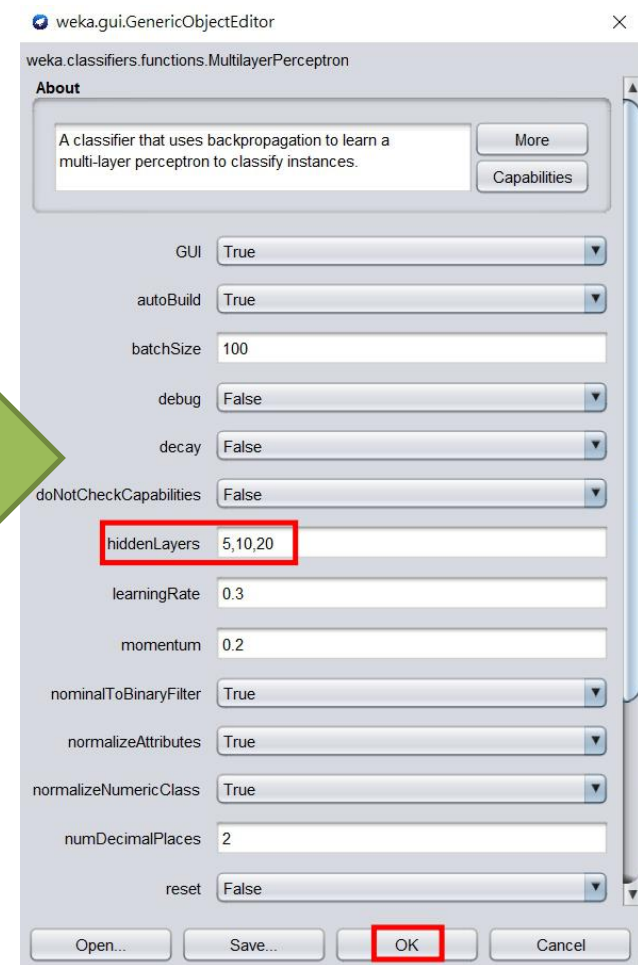
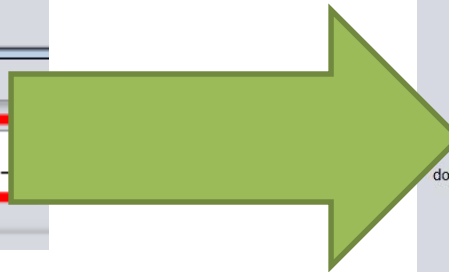
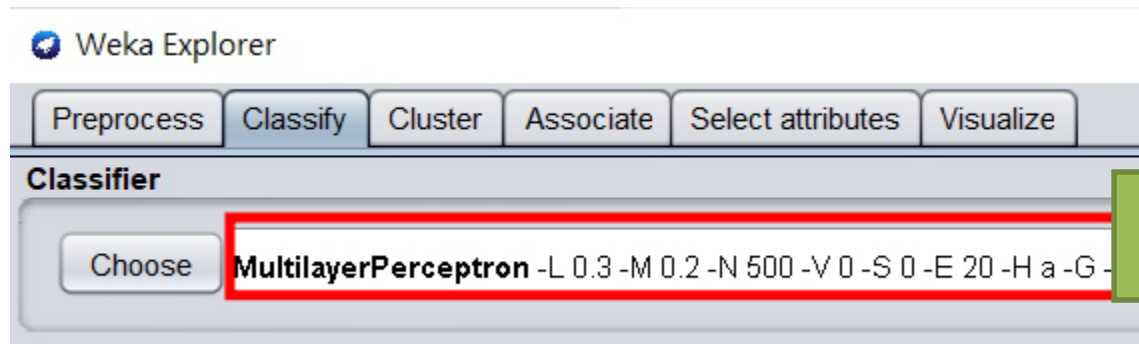
參數

- ❖ hiddenLayer: 將GUI設為`true`並且試試在隱藏層設定5, 10, 20個神經元
- ❖ 學習率, 動量
- ❖ 對數據進行多次傳遞(“epoch”)
- ❖ 持續訓練直到
 - 驗證集中的錯誤持續增加
 - 或超過訓練時間

Lesson 5.2: 多層感知機

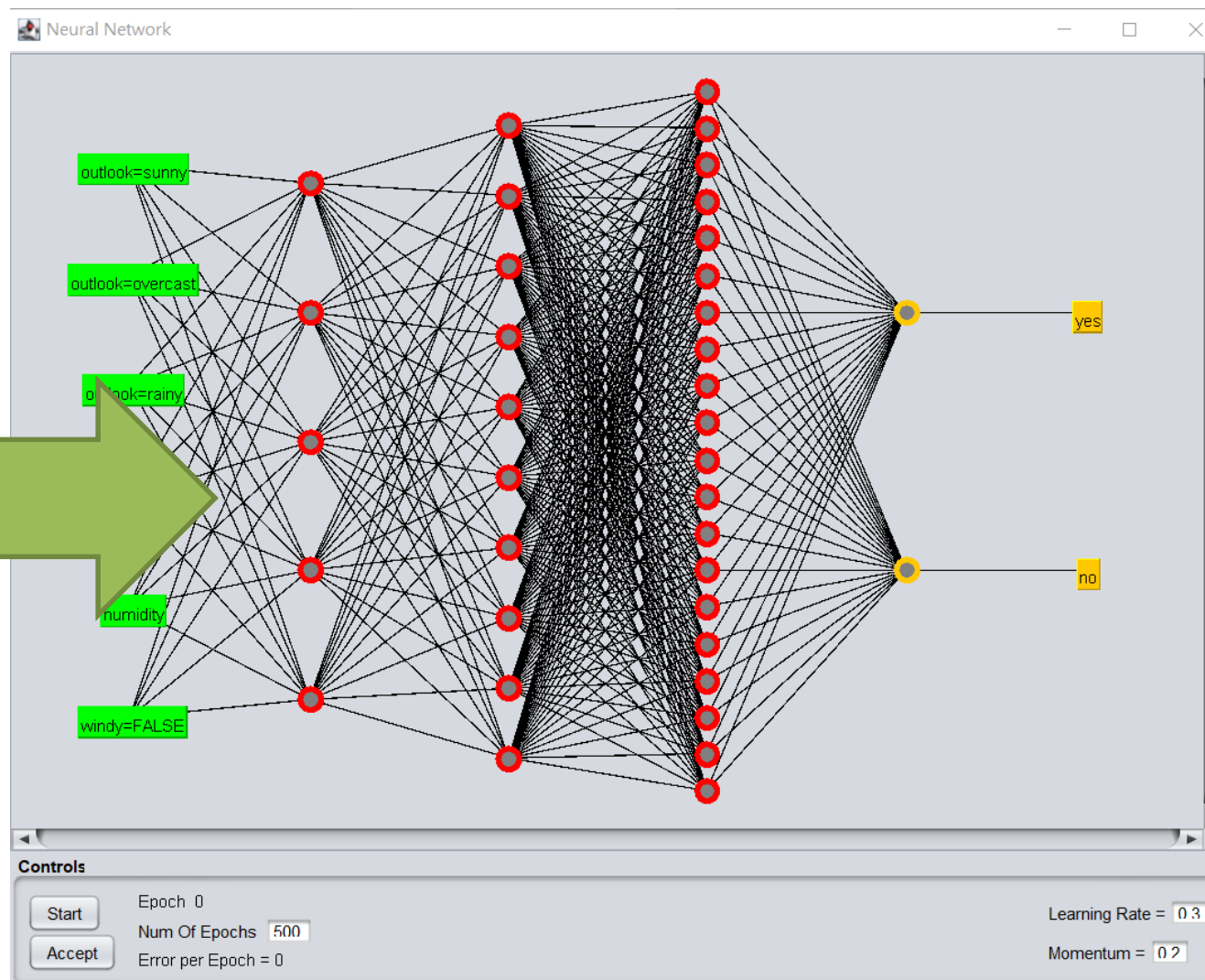
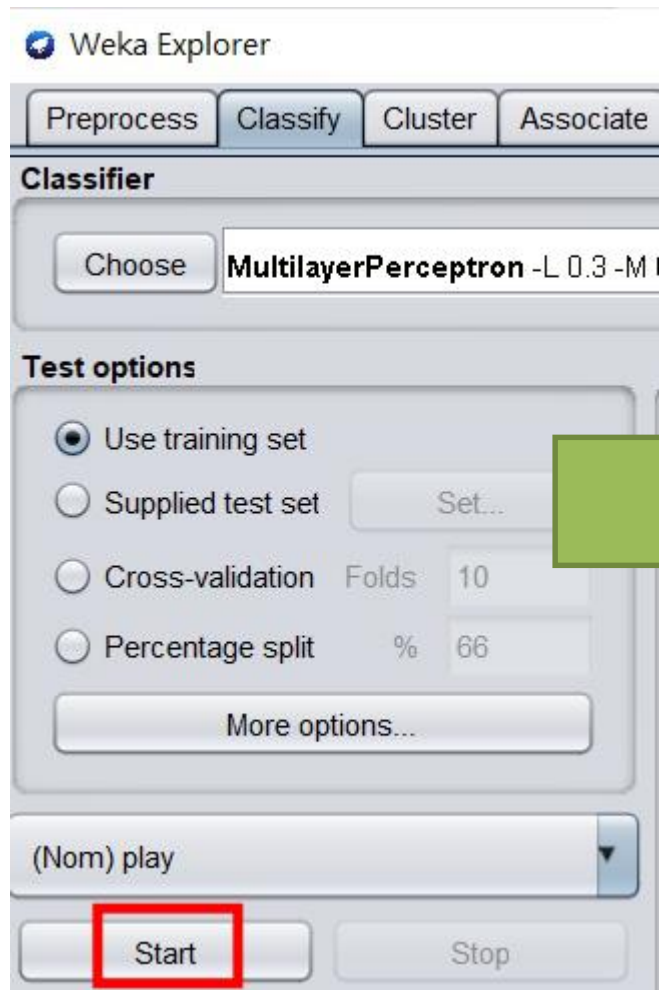
我們試著在3個隱蔽層設定5、10和20個神經元。

1. 左鍵單擊分類器名稱(左圖紅框處)開啟配置視窗(右圖)。將參數 hiddenLayers 設定為5,10,20(逗號後不需空格)，然後左鍵單擊下方OK按鈕。



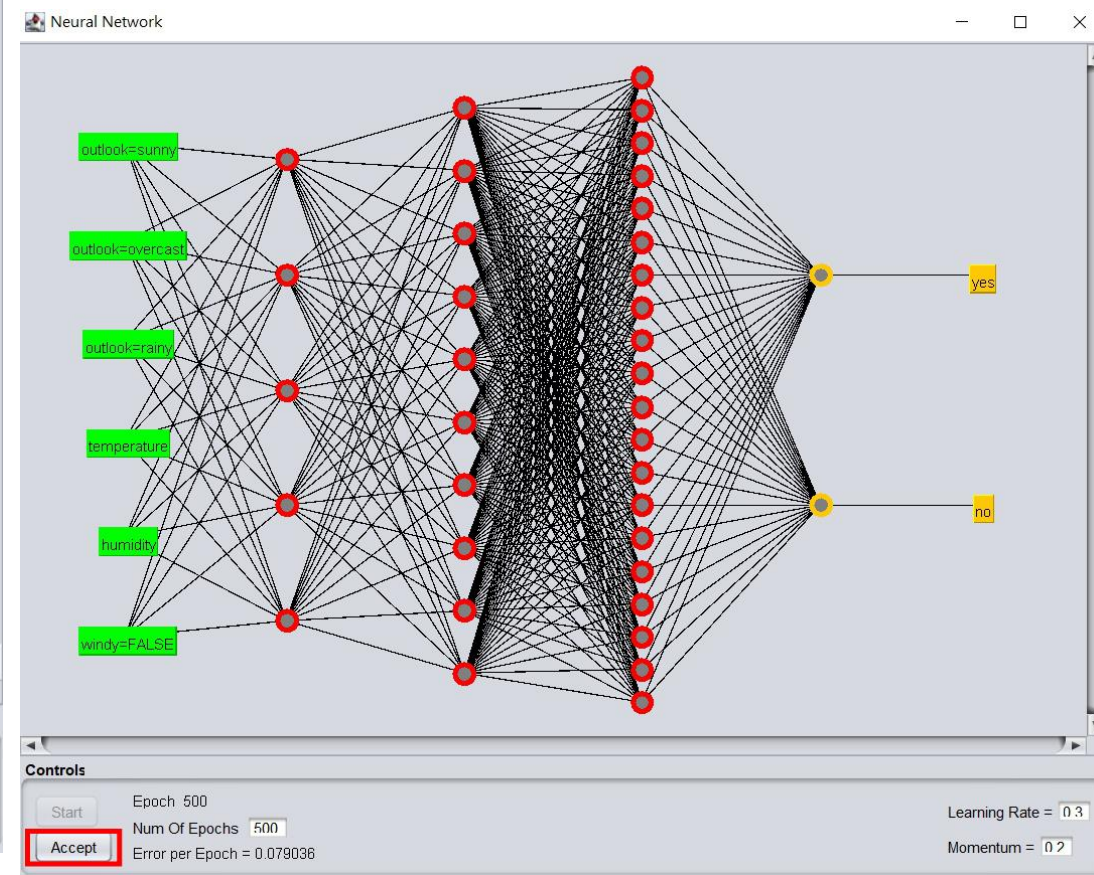
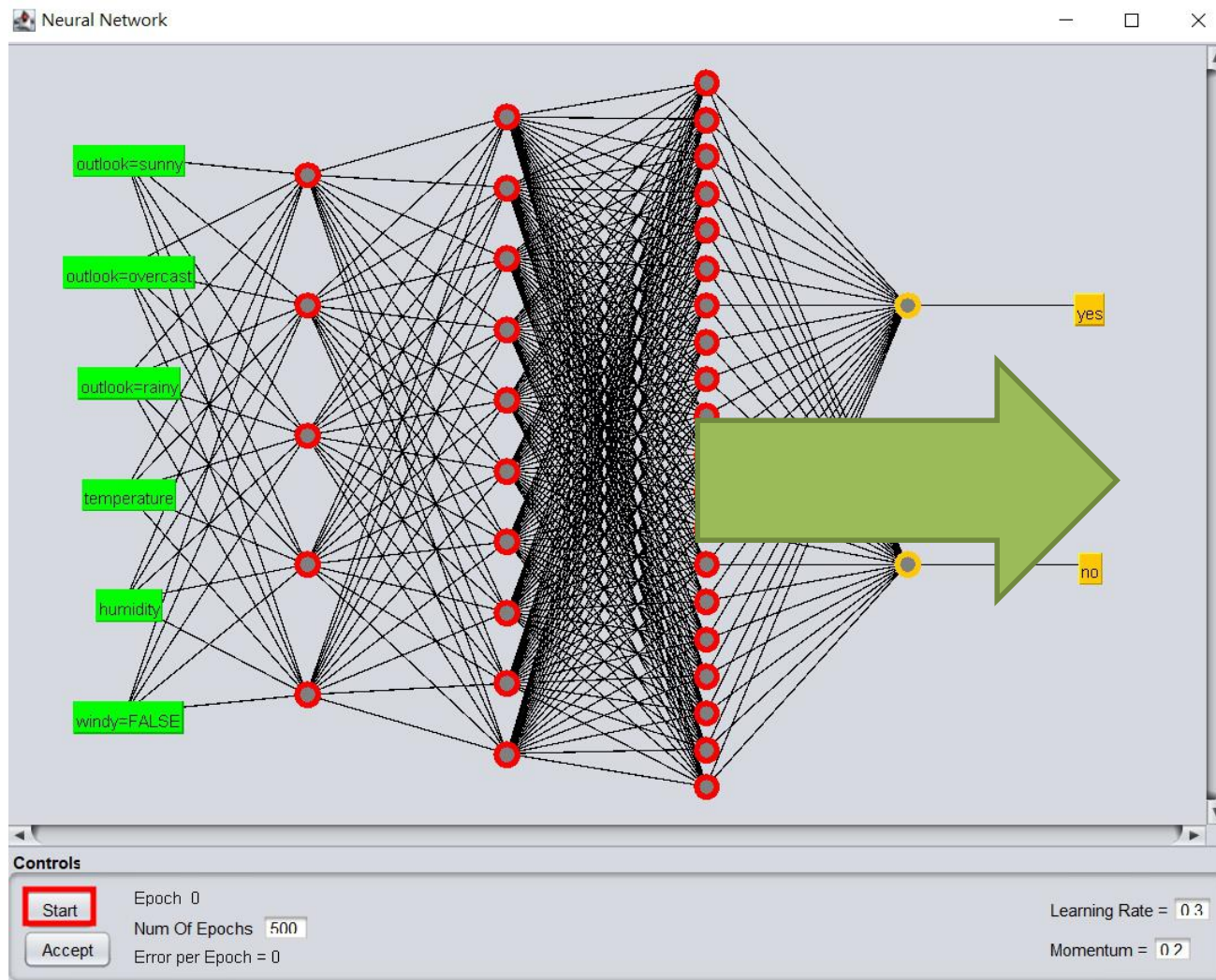
Lesson 5.2: 多層感知機

2. 左鍵單擊Start按鈕，執行結果如右圖。



Lesson 5.2: 多層感知機

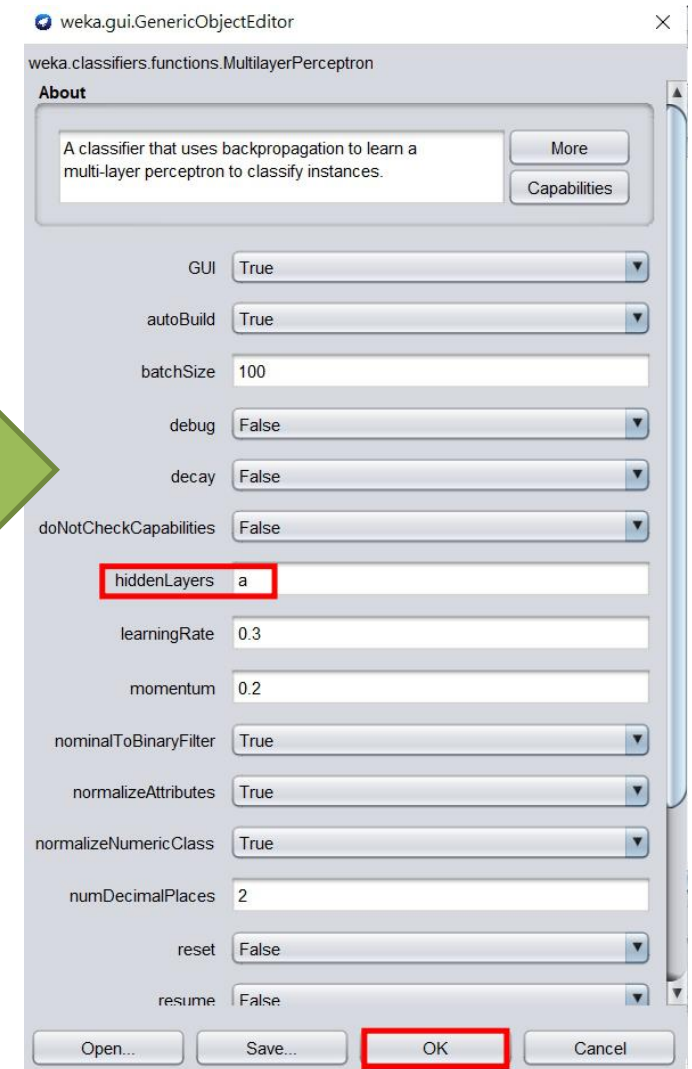
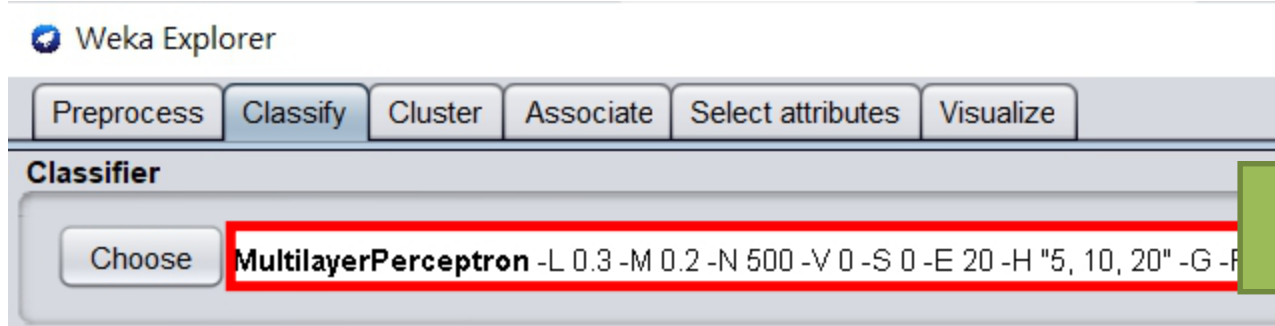
3. 左鍵單擊Start按鈕，接著按下Accept按鈕回到Classify面板。



Lesson 5.2: 多層感知機

接著我們嘗試將參數hiddenLayers設定為a(Weka預設值)。

1.左鍵單擊分類器名稱(左圖紅框處)開啟配置視窗(右圖)。將參數hiddenLayers設定為a，然後左鍵單擊下方OK按鈕。



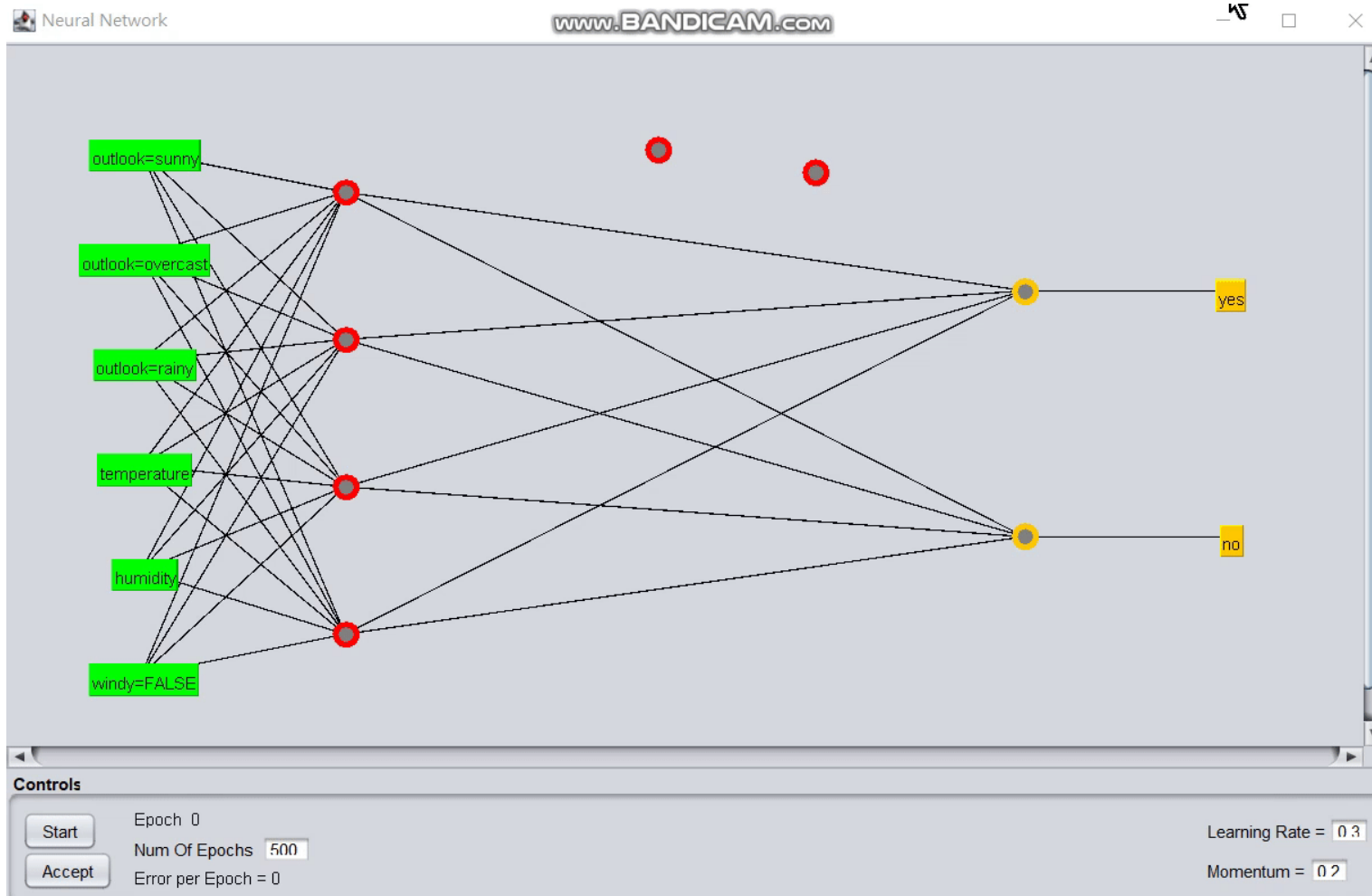
Lesson 5.2: 多層感知機

2. 左鍵單擊**Start**按鈕，執行結果如右圖：我們可以在此網路圖上進行操作。

The image displays the Weka Explorer interface for configuring and running a Multilayer Perceptron (MLP) classifier. On the left, the 'Classifier' tab is selected, showing 'MultilayerPerceptron -L 0.3 -M 0'. The 'Test options' section includes radio buttons for 'Use training set' (selected), 'Supplied test set', 'Cross-validation' (with 10 folds), and 'Percentage split' (with 66%). A large green arrow points from the 'Start' button in the Weka Explorer to the 'Neural Network' window on the right. The 'Neural Network' window shows a diagram of the MLP architecture with five input nodes (outlook=sunny, outlook=overcast, outlook=rainy, temperature, humidity, windy=FALSE), four hidden nodes, and two output nodes (yes, no). The 'Controls' section at the bottom of the 'Neural Network' window includes buttons for 'Start' and 'Accept', and displays 'Epoch 0', 'Num Of Epochs 500', 'Error per Epoch = 0', 'Learning Rate = 0.3', and 'Momentum = 0.2'.

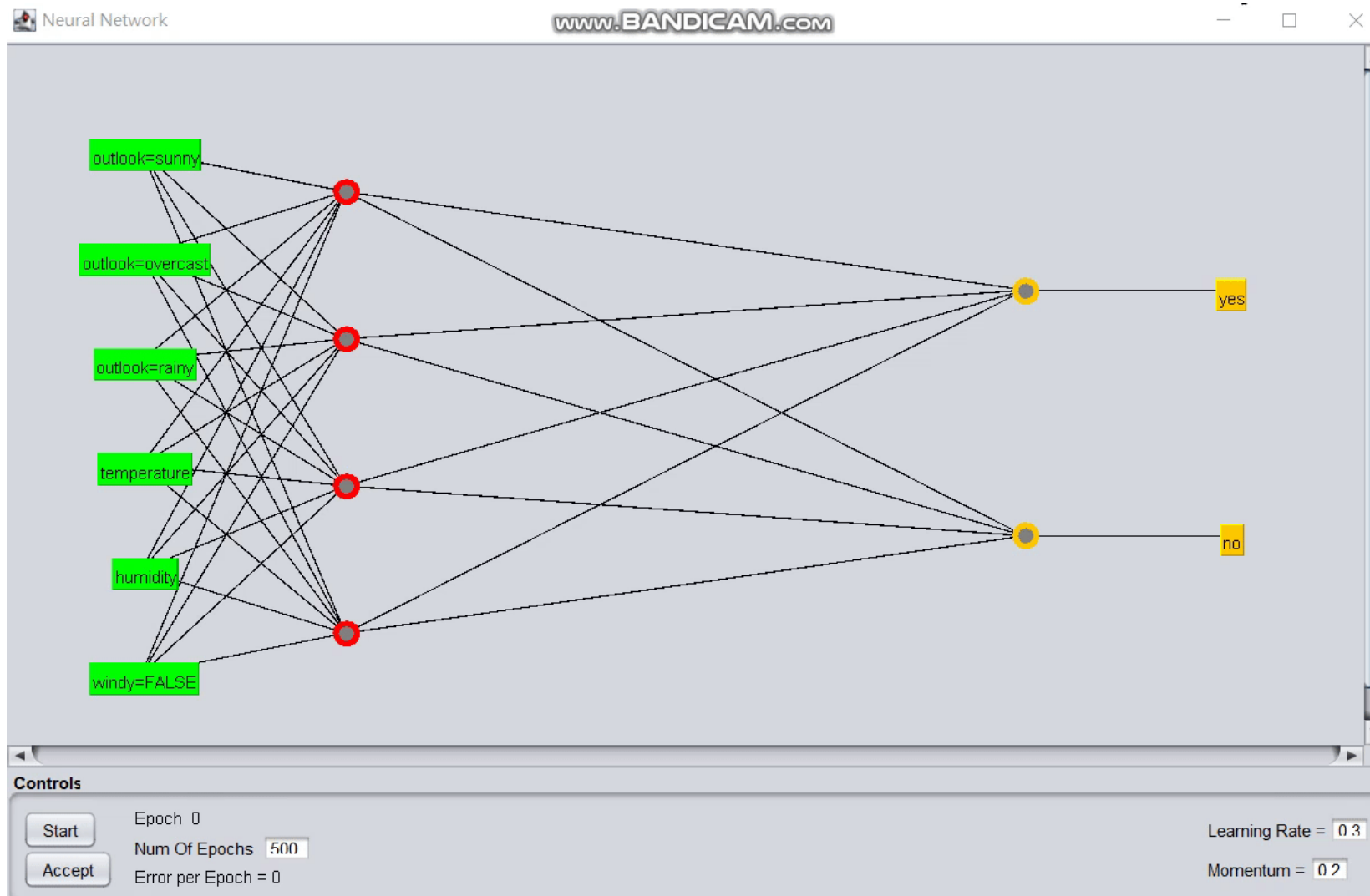
Lesson 5.2: 多層感知機

選取節點：當節點中心為黃色代表被選取，在位選取任何節點的情況下，左鍵單擊節點可進行選取，右鍵單擊空白處可取消選取當前節點。



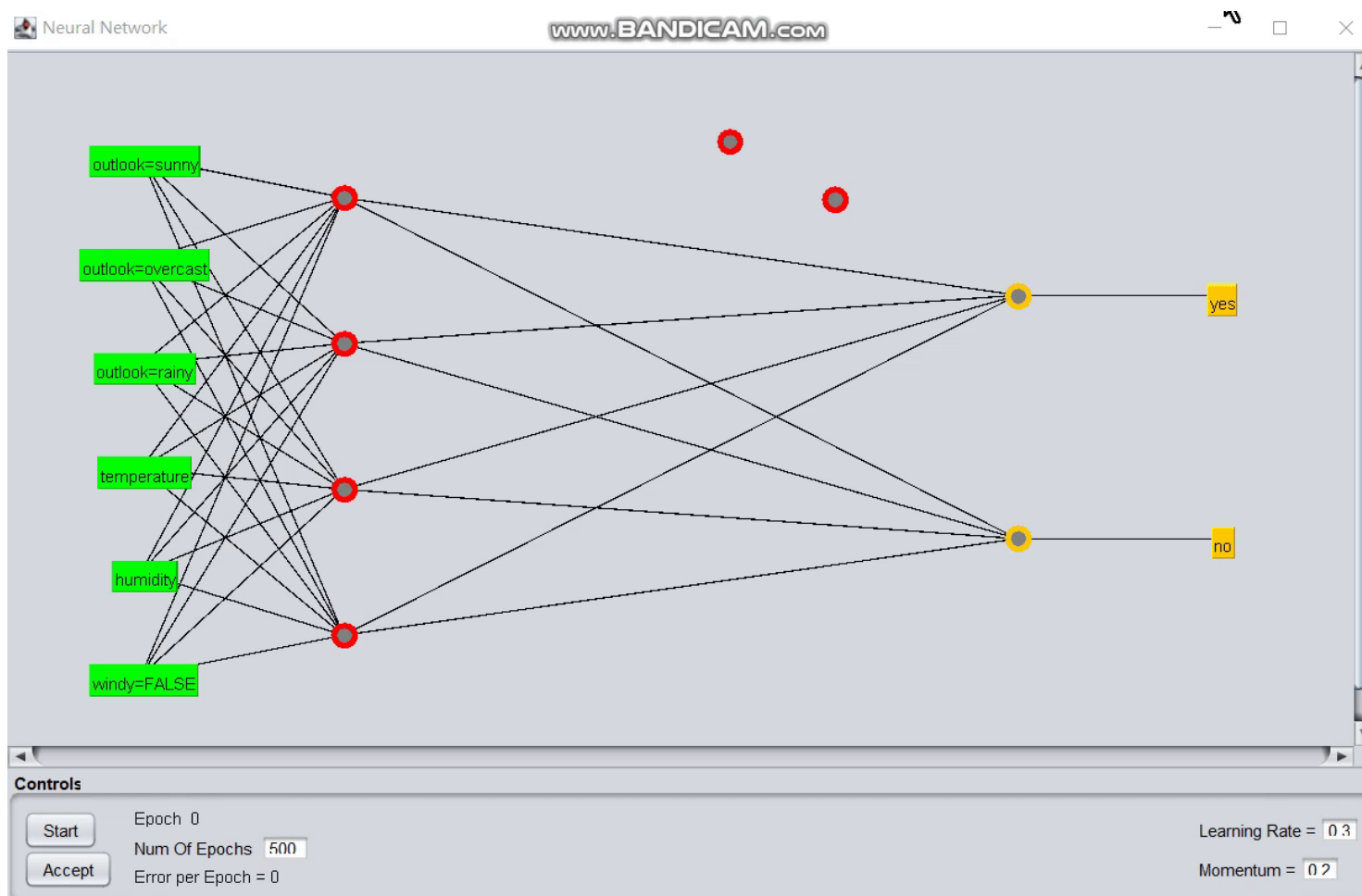
Lesson 5.2: 多層感知機

新增節點：左鍵單擊空白處可新增節點。



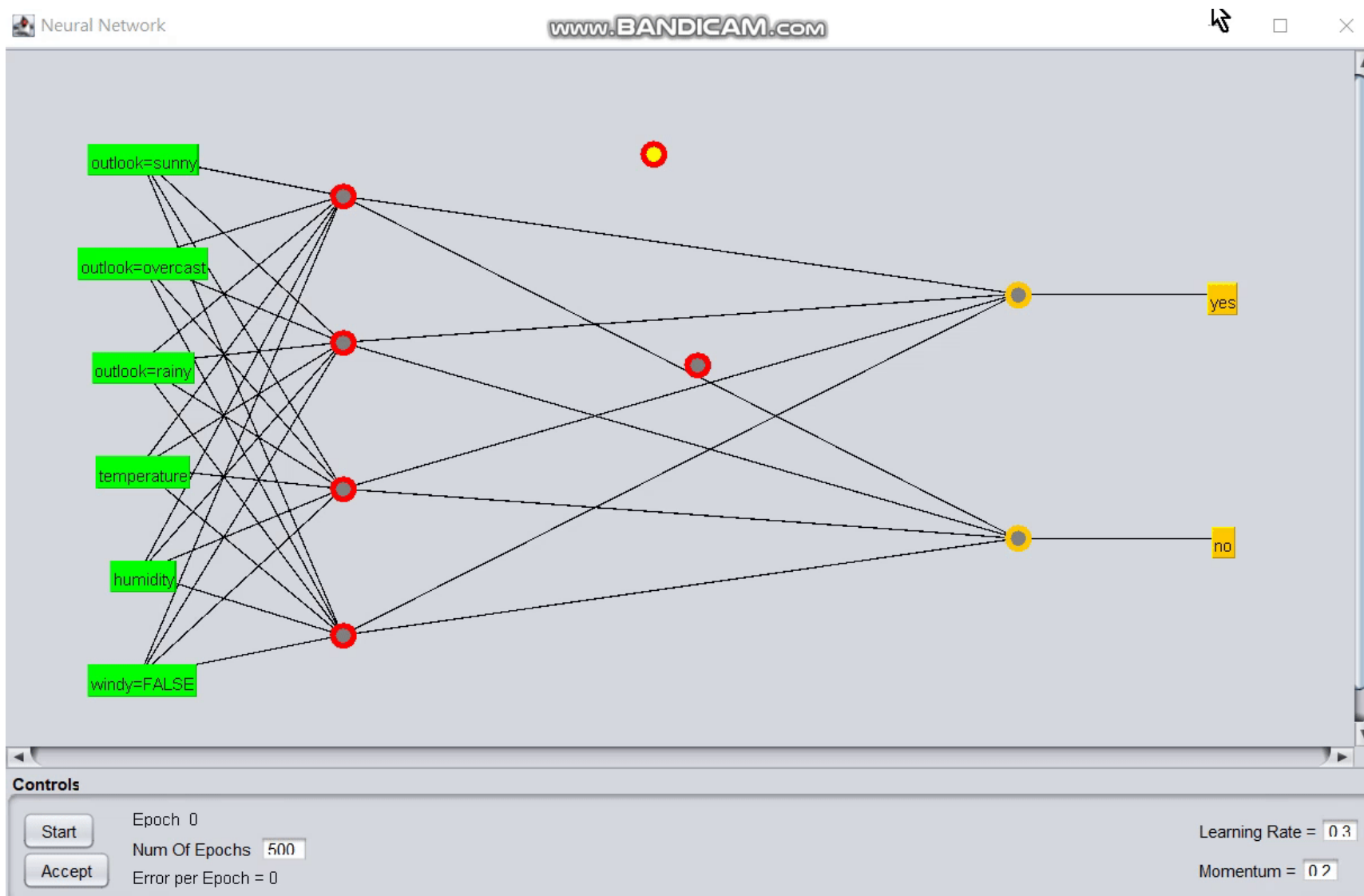
Lesson 5.2: 多層感知機

刪除節點：在未選取任何節點的狀況下，以右鍵單擊節點可進行刪除動作。



Lesson 5.2: 多層感知機

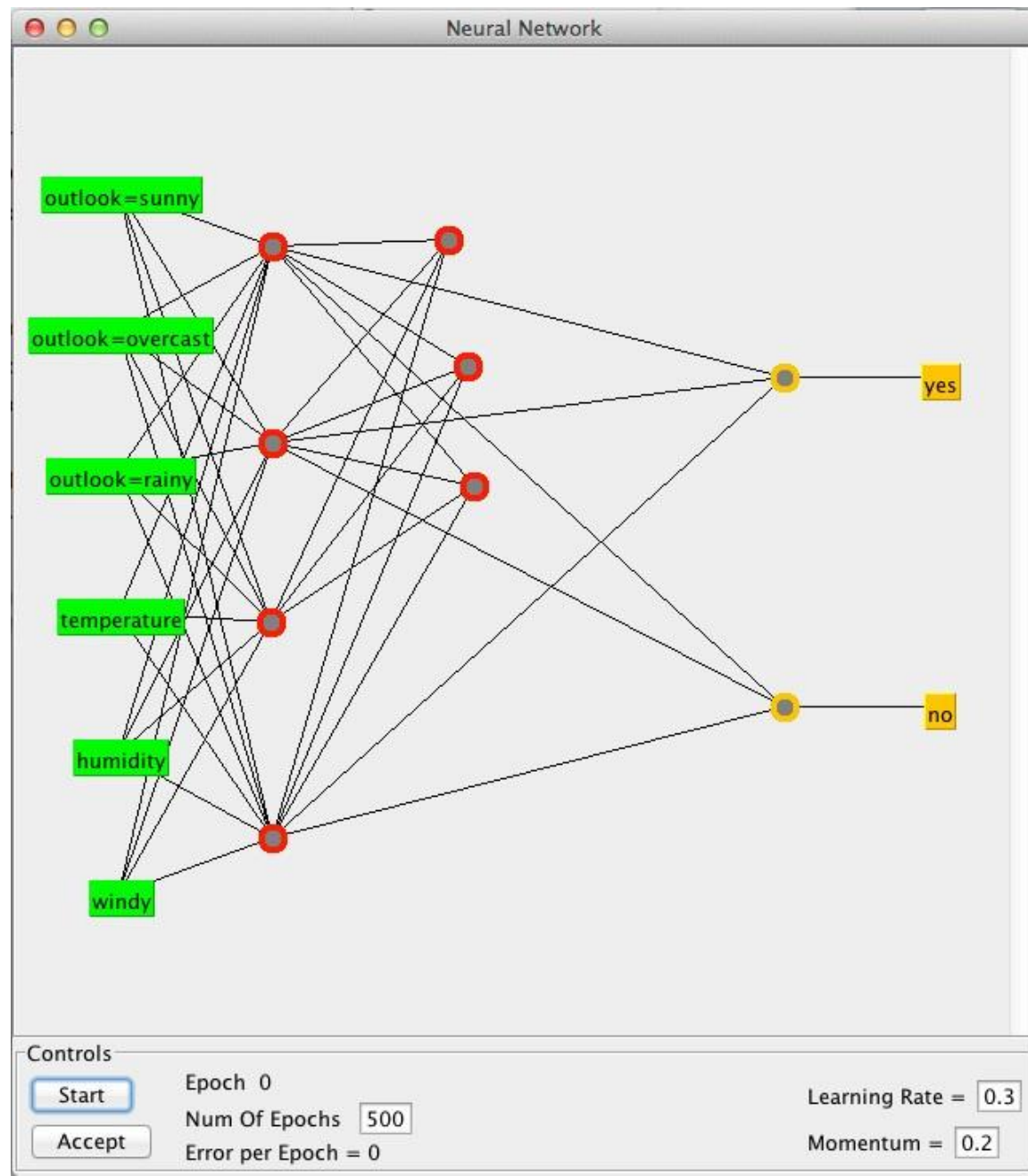
連接節點：在選取節點的狀況下，左鍵單擊其他節點會進行連接動作。



Lesson 5.2: 多層感知機

創建自己的網路架構!

- ❖ 選取節點
 - 點選以選取
 - 右擊空白處可以取消選取
- ❖ 創建/刪除節點
 - 點選空白處來創建節點
 - 右擊(在未選取任何節點的狀況下)來進行刪除
- ❖ 創建/刪除連接線
 - 選取一節點，點擊其他節點以進行連接
 - 右擊可刪除連接線
- ❖ 也可以在此處設定參數



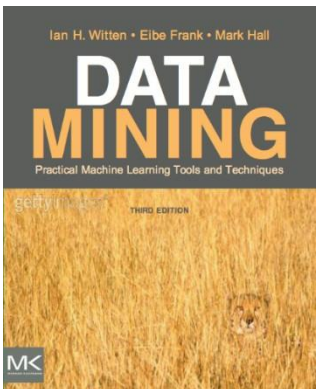
Lesson 5.2: 多層感知機

效果好嗎?

- ❖ 使用了6個資料集進行實驗
 - *Iris, breast-cancer, credit-g, diabetes, glass, ionosphere*
- ❖ 9個演算法
 - *MultilayerPerceptron, ZeroR, OneR, J48, NaiveBayes, IBk, SMO, AdaBoostM1, VotedPerceptron*
- ❖ MultilayerPerceptron在兩個資料集獲勝
- ❖ 其他的贏家:
 - *SMO*在2個資料集中獲勝
 - *J48*在1個資料集中獲勝
 - *IBk*在1個資料集中獲勝
- ❖ 多層感知器雖然不錯，但事實上它比其他方法慢10到2000倍

Lesson 5.2: 多層感知機

- ❖ 多層感知機實現任意的決策邊界
 - 給定兩個(或更多)足夠大的隱藏層
 - 並且接受了適當的訓練
 - ❖ 通過反向傳播訓練
 - 基於梯度下降的迭代算法
 - ❖ 在實踐中??
 - 相當好的性能，但非常慢
 - 未能留給人深刻印象
 - 在更複雜的數據集上可能比較能令人印象深刻
- 課程文本
- ❖ Section 4.6 *Linear classification using the Perceptron*
 - ❖ Section 6.4 *Kernel Perceptron*





More Data Mining with Weka

Department of Computer Science
University of Waikato
New Zealand



Creative Commons Attribution 3.0 Unported License



creativecommons.org/licenses/by/3.0/

weka.waikato.ac.nz