

2019 電子商務技術 期末考

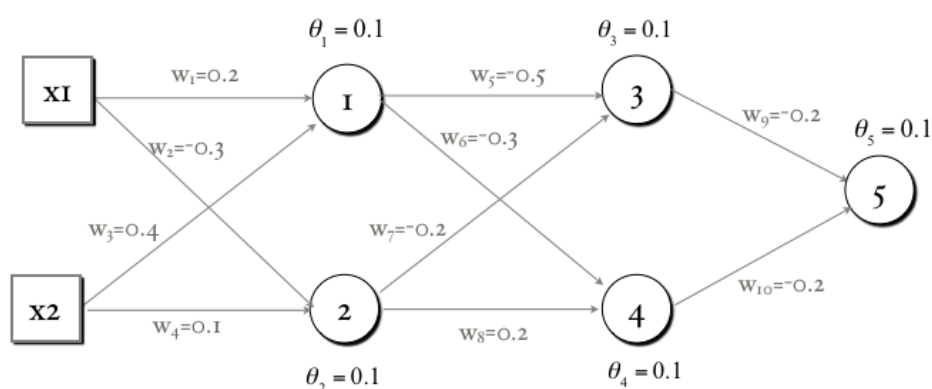
** 考試時間共 3 小時

** 若遇計算，請四捨五入取至小數點第 3 位

- 試說明 SRSWR 與 Reservoir sampling 不同的地方，以及各自的使用時機。(10%)
- 參考下圖 weather 資料回答以下問題：
 - 當” Attribute Evaluator=ChiSquaredAttributeEval and Search Method=Ranker” 時，Weka 會如何找出重要的屬性？(10%)
 - 當” Attribute Evaluator=CfsSubsetEval and Search Method=BestFirst - D 1 - N 5 (forward and after 5 node expansions” 時，Weka 會如何找出重要的屬性？(10%)

| Relation: weather | | | | | |
|-------------------|----------|-------------|----------|---------|---------|
| No. | outlook | temperature | humidity | windy | play |
| | Nominal | Numeric | Numeric | Nominal | Nominal |
| 1 | sunny | 85.0 | 85.0 | FALSE | no |
| 2 | sunny | 80.0 | 90.0 | TRUE | no |
| 3 | overcast | 83.0 | 86.0 | FALSE | yes |
| 4 | rainy | 70.0 | 96.0 | FALSE | yes |
| 5 | rainy | 68.0 | 80.0 | FALSE | yes |
| 6 | rainy | 65.0 | 70.0 | TRUE | no |
| 7 | overcast | 64.0 | 65.0 | TRUE | yes |
| 8 | sunny | 72.0 | 95.0 | FALSE | no |
| 9 | sunny | 69.0 | 70.0 | FALSE | yes |
| 10 | rainy | 75.0 | 80.0 | FALSE | yes |
| 11 | sunny | 75.0 | 70.0 | TRUE | yes |
| 12 | overcast | 72.0 | 90.0 | TRUE | yes |
| 13 | overcast | 81.0 | 75.0 | FALSE | yes |
| 14 | rainy | 71.0 | 91.0 | TRUE | no |

- 下圖為一 FFNN， x_1, x_2 是輸入層屬性，node1-4 為隱藏層節點，node5 是輸出節點，sigmode function 為 node1-5 的 activation function。假設 cost function $c=0.5*(y-y')^2$ ，其中 y 為預測值； y' 為實際值。試輸入訓練資料： $x_1=1, x_2=0, y=1$ ，以更新所有的權重值。(20%)



- 採 J48 為分類方法，詳細說明 Bagging 如何訓練一個分類模型，以及如何利用它來分類。(10%)
- Random Forest 如何訓練一個分類模型？如何做預測？(10%)
- 試依據下述 Keras 程式回答問題：

- (a) 試繪圖說明 CNN 網路的架構，並標示各層的節點數。(10%)
- (b) 就第一個卷積層，說明權重分享的做法。(5%)
- (c) 說明 softmax 如何預測輸出結果。(5%)

```
model = Sequential()
```

```
model.add(Conv2D(32, kernel_size(3,3), activation=' relu' , input_shape=(28,28,1)))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Flatten())
```

```
model.add(Dense(128, activation=' relu' ))
```

```
model.add(Dropout(0.5))
```

```
model.add(Dense(10, activation=' softmax' ))
```

7. 試根據下述 Keras 程式，描繪網路架構，並說明它的使用方式。(10%)

```
model = Sequential()
```

```
model.add(Embedding(input_dim=5000, output_dim=128, input_length=400))
```

```
model.add(LSTM(50))
```

```
model.add(Dropout(0.5))
```

```
model.add(Dense(unit=1, activation=' sigmoid' ))
```