Practice 2

Weka (for MLP)
Cross Validation
Recall & Precision

2013. 03.28

Eun-Sol Kim

Make an .arff file

```
@RELATION <dataset name>

@ATTRIBUTE <feature1 name> <feature1 type>
@ATTRIBUTE <feature2 name> <feature2 type>
@ATTRIBUTE <feature3 name> <feature3 type>
@ATTRIBUTE <feature4 name> <feature4 type>
.....
@ATTRIBUTE class <classes name>

@DATA
1,2,3,1, apple
3,2,0,1, book
......
```

Header

Data
(feature values + class)

Data format for Weka (.ARFF)

Header

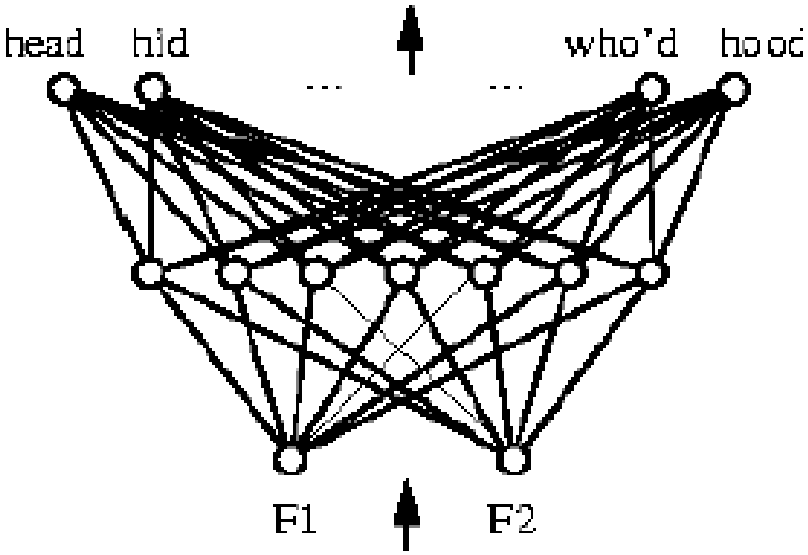
```
@relation heart-disease-simplified

@attribute age numeric
@attribute sex { female, male}
@attribute chest_pain_type { typ_angina, asympt, non_anginal,
    atyp_angina}
@attribute cholesterol numeric
@attribute exercise_induced_angina { no, yes}
@attribute class { present, not_present}

@data
```

Data
(CSV format)

```
63,male,typ_angina,233,no,not_present
67,male,asympt,286,yes,present
67,male,asympt,229,yes,present
38,female,non_anginal,?,no,not_present
```

Note: You can easily generate 'arff' file by adding a header to a usual CSV text file

MLP with WEKA



Training time

GRADIENT-DESCENT($training\_examples, \eta$)

Each training example is a pair of the form $\langle \vec{x}, t \rangle$*, where* $\vec{x}$ *is the vector of input values, and* t *is the target output value.* η *is the learning rate (e.g., .05).*

- Initialize each w_i to some small random value
- Until the termination condition is met, Do
  - Initialize each Δw_i to zero.
  - For each $\langle \vec{x}, t \rangle$ in *training_examples*, Do
    - Input the instance $\vec{x}$ to the unit and compute the output o
    - For each linear unit weight w_i, Do

$$\Delta w_i \leftarrow \Delta w_i + \eta(t - o)x_i$$

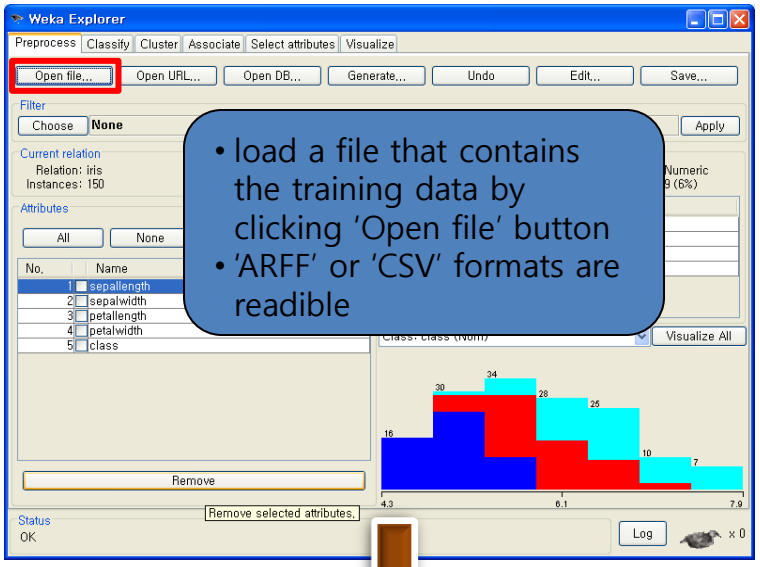
  - For each linear unit weight w_i, Do

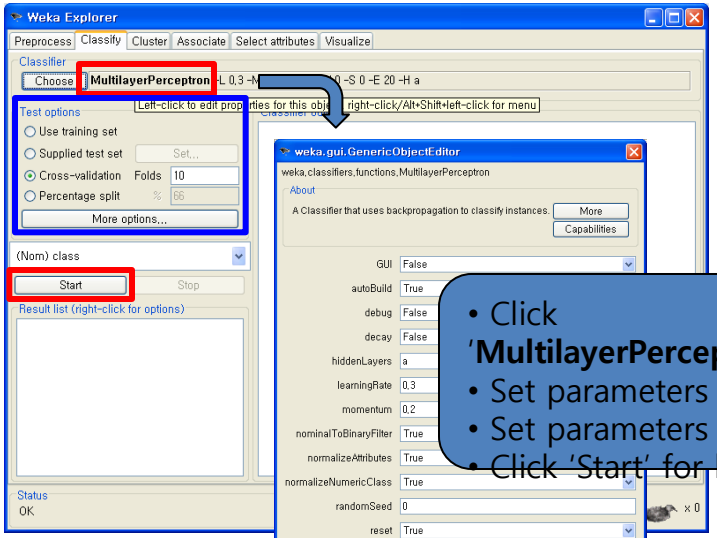
$$w_i \leftarrow w_i + \Delta w_i$$

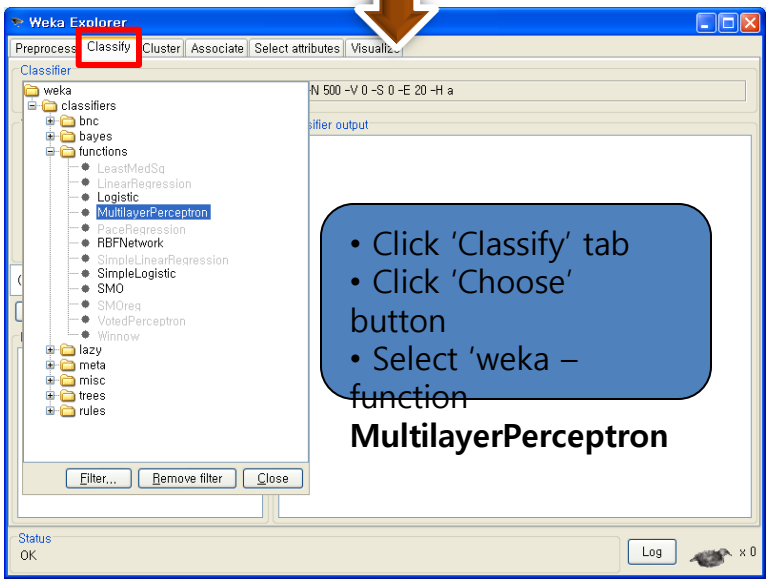
Learning rate

MLP in Weka


Weka GUI Chooser
Program Visualization Tools Help
click
WEKA
The University of Waikato
Explorer
Experimenter
KnowledgeFlow
Simple CLI
Waikato Environment for Knowledge Analysis
Version 3.6.0
(c) 1999 – 2008
The University of Waikato
Hamilton, New Zealand


Weka Explorer
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
• load a file that contains the training data by clicking 'Open file' button
• 'ARFF' or 'CSV' formats are readible


Weka Explorer
MultilayerPerceptron
weka.gui.GenericObjectEditor
Start
• Click 'MultilayerPerceptron'
• Set parameters for MLP
• Set parameters for Test
• Click 'Start' for learning


Weka Explorer
Classify
MultilayerPerceptron
• Click 'Classify' tab
• Click 'Choose' button
• Select 'weka – function
MultilayerPerceptron

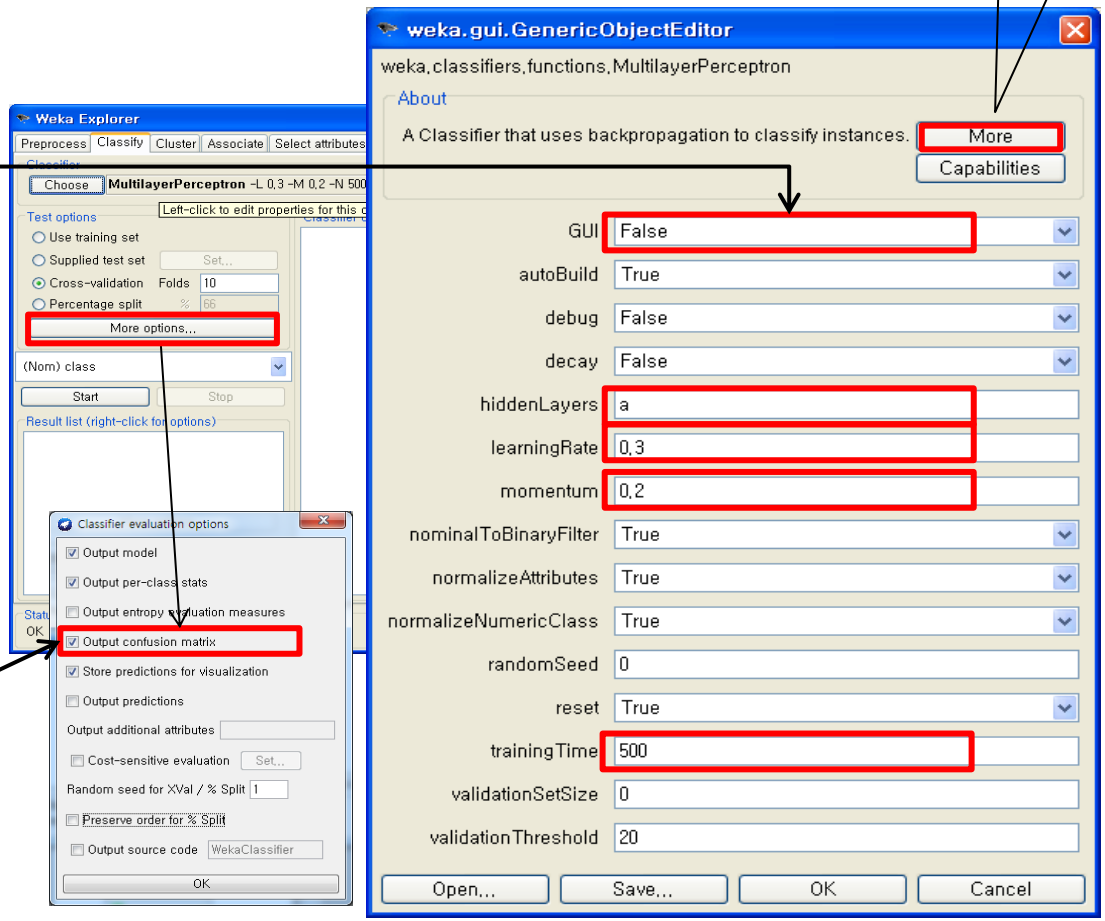
Parameter setting of MLPs

- You can change parameter values
  - Number of hidden layers
  - Number of nodes for each hidden layer
    - (10, 5) : 2 hidden layers, 10 nodes for first hidden layer, 5 nodes for second one
  - Learning rate
  - Momentum
  - Training times

More explanations on the parameters

You can you GUI option to check the structure

If you use ‘Percentage split’ option for test, you can split dataset into training/test data set randomly with ‘Random seed value’

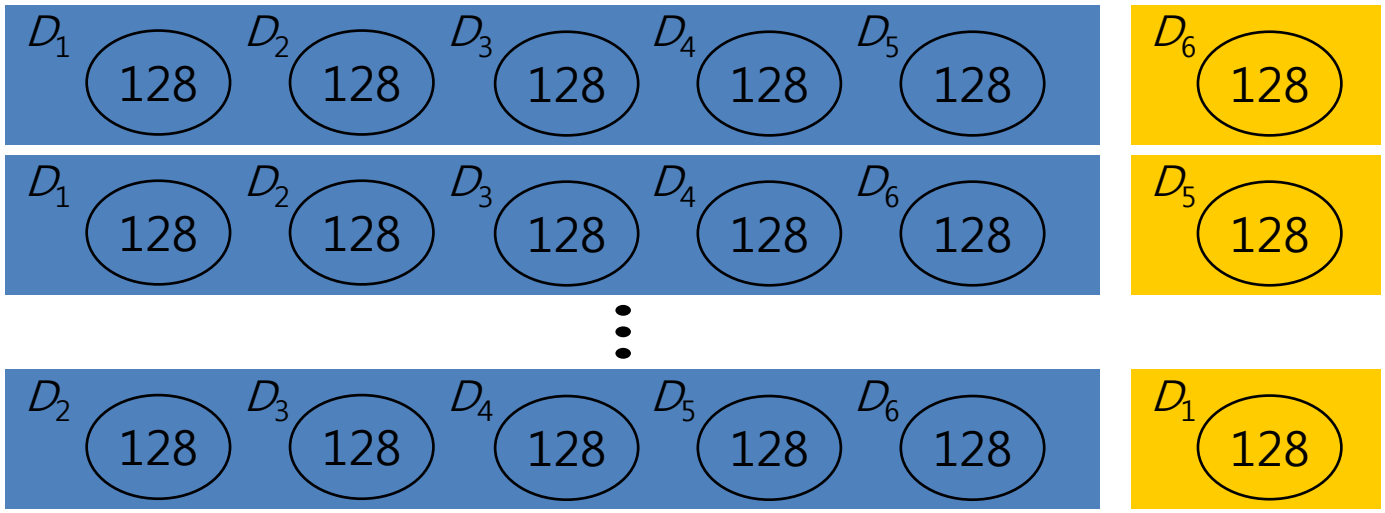


Evaluation Method - Cross Validation

- *K*-fold Cross Validation
  - The data set is randomly divided into *k* subsets.
  - One of the *k* subsets is used as the 'test set' and the other *k*-1 subsets are put together to form a 'training set'.

$$Error = \frac{1}{k}\sum_{i=1}^{k} Error_i$$

6-fold cross validation



Confusion Matrix

Real / Prediction	Positive	Negative	
Positive	**TP**	**FP**	**All with positive Test**
Negative	**FN**	**TN**	**All with Negative Test**
	All with Disease	**All without Disease**	**Everyone**

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

$$F = 2 \times \frac{precision \times recall}{precision + recall}$$

As **recall** ↑ **precision** ↓

conversely:

As **recall** ↓ **precision** ↑