# Subject

A Multilayer Perceptron for a classification task (neural network): comparison of TANAGRA, SIPINA and WEKA.

When we want to train a neural network, we have to follow these steps:

- Import the dataset;
- Select the discrete target attribute and the continuous input attributes;
- Split the dataset into learning and test set;
- Choose and parameterize the learning algorithm;
- Execute the learning process;
- Evaluate the performance of the model on the test set.

# Dataset

We use the IONOSPHERE.ARFF from UCI IRVINE (ARFF is the WEKA file format). The attributes are standardized. There are 351 examples, 33 continuous descriptors, and a binary class attribute.

# Training a neural network with TANAGRA

### **Dataset importation**

We click on the FILE/NEW menu in order to create a new diagram and import the dataset.



#### Splitting the dataset into learning and test set

In the next step, we have to split the dataset into a learning set, which is used for the computation of the neural network weights, and a test set, which is used for the model performance evaluation.

We add the SAMPLING component; we use 66% of examples for the learning phase.



### Select the class and the predictive attributes

We add the DEFINE STATUS in the diagram, we use the shortcut in the toolbar, we set CLASS as TARGET, and all continuous attributes as INPUT.



### Learning algorithm

We want to add a Multiplayer Perceptron in the diagram. In the first step, we add a learning implementation algorithm (SUPERVISED LEARNING from the META-SPV LEARNING TAB).

💯 TANAGRA 1.4.5 - [Dat	taset (ionosphere.arff)]							
Tile Diagram Componer	nt Window Help						-	Б×
D 📽 🖪   🍇								
	Default title		a08	Continue	-			^
🖃 🏢 Dataset (ionospher	re.arff)		a09	Continue	-			
🖃 💉 Sampling 1			a10	Continue	-			
🖻 🎎 Define stati	us 1		a11	Continue	-			
Supervi:	sed Learning 1		a12	Continue	-			
			a13	Continue				
l î			a14	Continue	-			
			a15	Continue	-			
			a16	Continue	-			
	·		a17	Continue	-			
			a18	Continue	-			~
			<	<u>``</u>				>
			Componen	its	•			
Data visualization	Statistics	Nonpara	metric stat	istics	Instance set	ection	Feature construction	
Feature selection	Regression	<b>F</b> act	orial analys	is	PLS		Clustering	
Spv learning Meta-spv learning Spv learn			ning assessment Sco		Scoring		Association	
🕨 Aggregating 🚺	Arcing [Arc-x4] 🕨 Ba	gging	Þ	Boosting		Supervised	d Learning	

In the second step, we embed in the first one, a learning method algorithm i.e. the MULTILAYER PERCEPTRON from the SPV LEARNING tab.

💯 TANAGRA 1.4.5 - [Dat	aset (ionosphere.arff)]						
🏆 File Diagram Componen	t Window Help						_ 8 ×
D 📽 🖬   👪							
	Default title		a08	Continue	-		^
🖃 🏢 Dataset (ionospher	e.arff)		a09	Continue	-		
🖻 🖍 Sampling 1			a10	Continue	-		
🖃 🐔 Define statu	us 1		a11	Continue	-		
·····( <u>*</u> ] Supervis	ed Learning 1 (Wultilayer pe	rceptron)	a12	Continue	-		
		•	a13	Continue	-		
	•		a14	Continue	-		
		<b>`</b>	a15	Continue	-		
		<b>`</b> 、	a16	Continue	-		~
			<		1111		
		Comp	oments				
Data visualization	Statistics	Nonparametri	c statistics	Instar	nce selection	n Feature constr	ruction
Feature selection	Regression	Factorial a	analysis		PLS	Clusterin	g
Spv learning	Meta-spv learning	Spv learning a	ssessment		Scoring	Associatio	on [
Harse Binary logistic regression	in 🧟 C-RT		🖧 ID3		_		Ē
🚓 C4.5 🔣 c-svc			🔄 K-NN			Multilayer perceptro	i -
🔀 C-PLS	🕍 Linear di	scriminant	analysis	1+2×MUltil omial Legistic	Regression 🔋		
<							>
L							.:

#### Setting the parameters

There are several kinds of parameters. The first ones are the neural architecture parameters (NETWORK tab). We use a hidden layer with two neurons.

ML	P parameters	
ſ	Parameters	
	Network Learning Stopping rule	
	🗹 Use hidden layer 🛛 🗲 🗕 🗕	
	Number of neurons : 2	
	OK Cancel Help	

The next ones are the learning parameters (LEARNING tab). We set the LEARNING RATE to 0.15.

We can define a validation set. This sub-sample enables to compute the error rate on a part of the learning set which is not used for the computation of weights. In this analysis, we do not use a validation set – VALIDATION SET PROPORTION = 0.15.

Last, because we have already standardized descriptors, we set ATTRIBUTE TRANSFORMATION to NONE.

MLP parameters
Parameters Network Learning Stopping rule Learning rate : 0.1500 Validation set proportion : 0 Attribute transformation Onone Centered Standardized Normalized
OK Cancel Help

In the last tab, STOPPING RULE, we set the parameters which enables to stop the learning process: MAX ITERATION is the max number of epochs; ERROR RATE THRESHOLD enables to stop the learning process if the resubstitution error rate is lower than this threshold.

It is possible to stop the learning phase when we note a stagnation of the validation error rate on GAP TEST STAGNATION epochs. But it is not a very efficient option, check only this option if you are confident about the behavior of neural network.

M	.P parameters	
	Parameters	
	Network Learning Stopping rule	L
	Maxiteration : 100	Ļ
	Error rate thresold : 0.01	ŀ
	Test error stagnation	L
	Gap test stagnation : 20	L
		L
	OK Cancel Help	

### **Reading the results**

We select the VIEW menu, the weights are computed and a new window appears.



## **Classifier performances**

Error rate			0.0260						
٧a	lues pre	ediction	Confusion matrix						
∀alue	Recall	1-Precision		good	bad	Sum			
good	1.0000	0.0403	good	143	0	143			
bad	0.9318	0.0000	bad	6	82	88			
			Sum	149	82	231			

In the first part of the window, we can see a summary of the network parameters and the resubstitution confusion matrix (0.026). We know that this estimation of the error rate is often highly optimistic.

In the second part of the window, the weights of the network are displayed. We can copy and paste theses values in a spreadsheet.

The ATTRIBUTE CONTRIBUTION part computes the error rate of the Perceptron when we remove one attribute. TANAGRA compares this error rate with the error rate of the whole model in order to evaluate the importance of each attribute in the prediction.

Excluded attribute	Error rate	Difference	Statistics
none	0.0260	-	
a01	0.1169	0.0909	8.6868-
a03	0.0433	0.0173	1.6546
a04	0.0519	0.0260	2,4819
a05	0.0563	0.0303	2.8956
a06	0.0563	0.0303	2.8956
a07	0.0260	0.0000	0.0000

## Attribute contribution

For instance, we see in this table that the error rate of the whole Perceptron is 0.026. If we remove the "a01" attribute -- i.e. we use the average of the attribute instead of the true values -- the error rate becomes 0.1169. The difference is 0.0909, if we use a statistical comparison between these two proportions; the t-value is 8.6868, it seems highly significant.

In the last part of the window, the ERROR RATE DECREASING shows the error rate during the learning process. We can copy and paste this table in a spreadsheet and create a graphical representation of the error rate progression.



#### Evaluate the network on a test set

We want to compute the test error rate on the 120 remaining examples. We add again the DEFINE STATUS component in the diagram. We set CLASS as TARGET, the prediction of the neural network (PRED\_SPVINSTANCE\_1) as INPUT.

TANAGRA 1.4.5 - [Sam File Diagram Component	npling 1] Window Help		- 0
Dataset (ionospheri Sampling 1 Sampling 1 Supervis Supervis Supervis Supervis Supervis Supervis Supervis	Default title [ e.arff] us 1 ed Learning 1 (Multilay re status 2	Perameters  Attributes:  Target Input Illustrative  pred_SpvInstance_1  a30  a31  a32  a33  a34  base	
Data visualization Feature selection Spv learning Correlation scatterplot	Statistics Regression Meta-spv learning Reference dataset	Image: Description of the selection       Image: Description of the selection       OK       Clear selection	catterplot

Then, we add the TEST component (SPV LEARNING ASSESMENT tab). The error rate must be computed on the test (unselected examples) set.

💯 TANAGRA 1.4.5 - [Sa	mpling 1]		
🏆 File Diagram Componer	nt Window Help		_ 8 ×
🗅 🛩 🖪   🎎			
	Default title		Sampling 1
🖃 🏢 Dataset (ionosphe	re.arff)		Parameters
😑 🧪 Sampling 1			
😑 🚰 Define stat	us 1	Define evaluation set	
😑 🕩 Supervi	sed Learning 1 (Multilayer pe	C6 Parameters	
Contraction of the second seco	ine status 2 Test 1 Parameters Execute View	Used examples Selected Unselected	Cancel Help
Data visualization	Statistics	Nonparametric statistics	Instance selection
Feature construction	Feature selection	Regression	Factorial analysis
PLS	Clustering	Spv learning	Meta-spv learning
Spv learning assessment	Scoring	Association	
Bias-variance decompo	sition 12 Cross validation	<mark>₿?</mark> Train-test	

We click on the VIEW menu; the test error rate is 0.125.

Test 1									
Parameters									
Evaluation set : <b>unselected</b> examples									
Results									
		pr	ed_Spv	Instance_1					
	Error	rate	0.1250						
Va	lues pro	ediction	Confusion matrix						
Value	Recall	1-Precision		good	bad	Sum			
good	0.9634	0.1319	good	79	3	82			
bad	0.6842	0,1034	bad	12	26	38			
			Sum	91	29	120			

#### Modifying the network parameters

We can improve the power of the neural network when we modify the number of neurons in the hidden layer. We set this parameter to 10. A priori, we should obtain a more efficient network.



We click again on the VIEW menu. The resubstitution error rate is 0.0206.

	and Learning 4. (Multilla					
File Diagram Component Wi	odow Help	ver perception)]				
						~
Default titl	e	MLP architecture				<u>^</u>
🖃 🏢 Dataset (ionosphere.arfl	f)	Use hidden layer	yes			_
□- 🖋 Sampling 1		Neurons in the hidden layer	10 🧲			
Define status 1	and a tabulation of	Learning paramete	<b>'5</b>			
Supervised Le	Parameterc	Validation set proportion	0.00			
I Test	Supervised parameters	Learning rate	0.15			
\$** * · · · · ·	Execute	Attribute transformation	none			
	View	Stopping rule				
-		Max iteration	100			
		Error rate thresold	0.0100			
		Verifiy error stagnation	no			
						_
			Res	utts		
		Classifier pe	rforman	ces		
		Error rate		0.0260		
		Values prediction		Confusion matrix		
		Value Recall 1-Precisio	n goo	d bad	Sum	
<	>	<	•			>
		Components				
Data visualization	Statistics	Nonparametric statistics	Instance se	lection	Feature construction	
Feature selection	Regression	Factorial analysis	PLS		Clustering	
Spv learning	Meta-spv learning	Spv learning assessment	Scorin	g	Association	
8     Bias-variance decomposition       8     P       8     P       8     P       9     Cross-validation	79 Test 99 Train-test					

When we click on the VIEW menu of the TEST component, the test error rate is 0.1083.

TANAGRA 1.4.5 - [Tes	t 1]									
💇 File Diagram Componen	t Window Help								-	Ξ×
D 📽 🖬   🎎										
Defa	ault title									~
🖃 🏢 Dataset (ionospher	e.arff)	11 -				Bana	est 1			
🛓 🥢 🖍 Sampling 1		Eva	aluation set	: un	selected e	xamples	meters			
📄 👬 Define statu	ıs 1									
🖹 🕨 🕨 Supervis	ed Learning 1 (Multilayer per					Re	sutts			
🖻 🚮 Defir	ne status 2				р	red_Spy	linstance_1	1		
····· • • • • • • • • • • • • • • • • •	Parameters		Erre	ог га	ate	-		0.1083		
	Execute		Values	prec	diction		Confe	ision matrix	<b>_</b>	
	View	3	Value Reca	ม 1	-Precision		good	bad	Sum	
			good 0.97	56	0.1209	good	- 80	2	82	
			bad 0.710	)5	0.0690	bad	11	27	38	
						Sum	91	29	120	
<	>									
	, <u> </u>			_						
Data visualization	Chablabian	( Nana	Componen	ts -+-+	intin a	Instas		-		
	Statistics	ΝΟΠμ	Deeneed	51dl	ISCIUS	Instance selection				
Peature construction	Clusteries		Cerrelean	on !		Factorial analysis				
PLS	Clustering		Spv learn	Ing		/Weta	-spv learning	8		
Spv learning assessment	Scoring		Associati	on						
Bias-variance decomposition			Cross-validation			? Test			88 <b>1</b>	
<										>
										.::

We have a small test set, the results suffers of a strong variability. This difference is not really significant. We have tried some other algorithms such as Linear Support Vector Machine or Linear Discriminant Analysis. We see in the following screenshot the accuracy on the same test set.





# Training a neural network with SIPINA

#### Importing the dataset

In order to import the dataset, we click on the FILE/OPEN menu.



#### Splitting the dataset

We want to use the 66% of the dataset as learning set. We select the ANALYSIS / SELECT ACTIVE EXAMPLES menu, we select the RANDOM SAMPLING option.

Sipina Research Version			
File Edit Data Statistics Induction method	Analysis View Window He	elp	
🖸 🗎 🐂 🐚	Define class attribute Select active examples		
Attribute selection	Set weight field		
Filter active examples			
Size of sample	i.e 231 of	351 examples	
Sampling strategy			
Merodulassvam Hdl=8 Merge=0.05 Split=0.01 TypeBorferroni= ValueBonferroni=			Eilter method
Examples selecti			All List Random sampling Bule releating
Improved ChAID (List of examples Rando	m sampling Rule filter	J	🗸 OK 🛛 🗶 Annuler

The subsets size appears in a window.

🔉 Sipina Research Version							
File Edit Data Statistics Induc	ion method A	Analysis View	v Window I	Help			
🖰 🖹 📲 h 🕨							
Attribute selection	🗉 📶 🔊 🔊	rning set e	ditor				
		a01	a03	a04	a05	a06	a07
	1	0.35	0.72	-0.24	0.48	-0.21	0.57
	2	0.35	0.72	-0.53	0.63	-1.03	-1.34
	3	0.35	0.72	-0.17	0.77	-0.25	0.91
<u>,</u>	4	0.35	0.72	-1.12	0.77	1.92	0.32
Learning method	5	0.35	0.72	-0.15	0.65	-0.10	0.75
MethodName=Improved ChAID (I sc MethodClassName=TArbreDecision)	<u>6</u>	0.35	-1.25	-0.12	-1.35	-0.51	-1.14
Hdl=8	7	0.35	0.68	-0.35	0.67	-0.71	0.77
Merge=0.05	8	-2.87	-1.29	-0.10	-1.16	-0.25	0.91
Split=U.U1 U TupePonferroni=1	9	0.35	0.64	-0.26	0.77	-0.56	0.91
ValueBonferroni=1	10	0.35	-1.33	-0.28	-1.16	-0.25	-1.12
Sampling=0	✓ 11	0.35	0.72	0.06	0.77	-0.64	0.91
Examples selection	12	0.35	0.72	-1.32	0.77	-2.42	0.91
231 examples selected	13	0.35	0.72	-0.46	0.77	-0.47	0.91
120 examples idle	14	0.35	0.72	-2.07	0.77	0.23	0.61
M	<						
Improved ChAID (Tschuprow Goodnes	ss of Split)						

### Defining the class and the predictive attributes

We click on the ANALYSIS / DEFINE CLASS ATTRIBUTE menu in order to define the role of attributes. We use drag-and-drop in order to define the TARGET and the INPUT attributes.

Didacticiel - Etudes de o	cas								R.F
🔉 Sipina Research Version									
File Edit Data Statistics Induction n	method A	nalysis View	Window He	lp					
<u>۲</u> 🖹 🚔 🖡 🌬		Define class at Select active e	tribute xamples						
Attribute selection	🔊 Le	Set weight field	ł	-04	1.05		lage	6.00	
	1	Attribute sel	ection	804	Taus		1407	Tauo	
	2 3	Class					Variables -		
	<u>4</u> 5	class					a15 a16		<u> </u>
	6	Attributes					a17 a18 a19		
	8	a01 a03			^		a20 a21		
	9 10	a04 a05 a06					a22 a23		
	<u>11</u> 12	a07 a08	H				a24 a25		1
	13	a09 a10					a20 a27 a28		
Learning method	14 15	a11 a12			~		a29 a30		
MethodName=Improved ChAID (I sc MethodClassName=TArbreDecisionI	16 17	C Only di	iscrete				a31 a32		
Merge=0.05	<u>18</u> 19	C Only c	ontinuous				a33 a34		
TypeBonferroni=1 ValueBonferroni=1	20	© Both					Class		
Sampling=0	21							🖌 пк	🗙 Annuler
231 examples selected	23	0.07	0.00	0.47		0.07	4.45		
120 examples idle	24 <	-2.87	-3.30	2.17	-1.16	-0.25	-1.12	-0.23	-2.98 1
Improved ChAID (Tschuprow Goodness of	Split)								

The attributes selection appears on the left part of the window. The type of the attributes is displayed.

	rinication P	-indiy313 vicvv	In the International Internationa International International Internatio	Help				-
🕒 📫 🤷 🕒			******	icip				
×								
ribute selection	AN Lea	rning set ed	nor					
📭 Class attribute ү 🔼		a01	a03	a04	a05	a06	a07	
Class	1	0.35	0.72	-0.24	0.48	-0.21	0.57	
Predictive attributes	2	0.35	0.72	-0.53	0.63	-1.03	-1.34	
aui	3	0.35	0.72	-0.17	0.77	-0.25	0.91	
-04	4	0.35	0.72	-1.12	0.77	1.92	0.32	
2 304	5	0.35	0.72	-0.15	0.65	-0.10	0.75	
2 305	6	0.35	-1.25	-0.12	-1.35	-0.51	-1.14	
- a07	7	0.35	0.68	-0.35	0.67	-0.71	0.77	
ans	8	-2.87	-1.29	-0.10	-1.16	-0.25	0.91	
a09	9	0.35	0.64	-0.26	0.77	-0.56	0.91	
a10	10	0.35	-1.33	-0.28	-1.16	-0.25	-1.12	
a11	11	0.35	0.72	0.06	0.77	-0.64	0.91	
a12	12	0.35	0.72	-1.32	0.77	-2.42	0.91	
- C a13	13	0.35	0.72	-0.46	0.77	-0.47	0.91	
- C a14	14	0.35	0.72	-2.07	0.77	0.23	0.61	
- C a15	15	0.35	0.72	0.06	0.77	-0.19	0.91	
<b>C</b> a16	16	0.35	-0.26	-2.23	0.77	0.33	-1.20	
🔚 🚺 a17 🛛 🖌 💌	17	0.35	0.72	0.04	0.77	-0.27	0.87	
aming method	18	-2.87	-1.29	-0.10	-3.08	-2.42	0.91	
thodName=Improved ChAID (Tsc A)	19	0.35	0.06	-0.03	0.13	-0.14	0.04	
thodClassName=TArbreDecision	20	-2.87	0.72	-2.37	-1.16	-0.25	-1.12	
H8 0.05	21	0.35	0.72	-0.12	0.77	-0.47	0.91	
rge=0.05	22	-2.87	0.72	2.17	-1.16	-0.25	-1.12	
eBonferroni=1	23	0.35	0.64	0.06	0.77	-0.16	0.91	
ueBonferroni=1	24	-2.87	-3.30	2.17	-1.16	-0.25	-1.12	
mpling=U 🕑	25	0.35	0.72	-0.24	0.77	-0.19	0.91	
amples selection	26	0.35	0.72	1.21	0.77	-2.42	0.91	
examples selected	27	0.35	0.72	-0.30	0.77	-0.62	0.65	
) examples ible		10.07	0.00	0.07		0.05	0.00	

### Learning algorithm and parameters settings

The INDUCTION METHOD / STANDARD ALGORITHM menu enables us to choose the learning algorithm. We select the NEURAL NETWORK tab and click on MULTILAYER PERCEPTRON method.

🔉 Sipina Research Ver	sion								
File Edit Data Statistics	Induction	method 🖌	Analysis View	Window H	Help				
<u>B 🗟 😫 h </u>	Standa	rd algorithr	n 🕇						
	Boostin	ig strategie	s						<u> </u>
Attribute selection		INA Lea	rnnig s <mark>et e</mark> o	ditor					
🖃 📭 Class attribute	^		a01	a03	a04	a05	a06	a07	
D class		1	0.35	0.72	-0.24	0.48	-0.21	0.57	
Predictive attributes		2	0.35	0.72	-0.53	0.63	-1.03	-1.34	
	Select an i	induction n	nethod						
a03	Induction Gr	aph Ì Rule Ind	Juction Neural ne	etwork secrimina	ant analysis Dec	sision list Other	1		
a05	Single-Layer	Perceptron	W	N-					_ =
a06	Multi-Layer F Multi-Layer F	Perceptron	st Firor rate contro	)					
- C a07				, ,					
- <b>C</b> a08									
— <mark>C</mark> a09									
<b>C</b> a10									
a11									
a12									_
Learning method	-			Multi-Lay	er Perceptron				-
MethodName=Improved ChA	I						🗸 ОК	🗙 Annuler	
MethodClassName=TArbreDe		16	0.35	-0.26	-2.23	0.77	0.33	-1.20	
Merge=0.05	E	17	0.35	0.72	0.04	0.77	-0.27	0.87	
Split=0.01		18	-2.87	-1.29	-0.10	-3.08	-2.42	0.91	
TypeBonferroni=1		19	0.35	0.06	-0.03	0.13	-0.14	0.04	
Sampling=0	~	20	-2.87	0.72	-2.37	-1.16	-0.25	-1.12	_
Examples selection		21	0.35	0.72	-0.12	0.77	-0.47	0.91	_
231 examples selected		22	-2.87	0.72	2.17	-1.16	-0.25	-1.12	_
120 examples idle		23	0.35	0.64	0.06	0.77	-0.16	0.91	~
		<							>
Improved ChAID (Tschuprow	Goodness of	f Split)							

When we click on the OK button, a new dialog box appears. We can set the architecture of the perceptron and the training parameters.

MLP parameter	
Max. Error 0.01 Max iter	ation   5000 🚖
MLP Structure	Nodes per layer
Hidden layer 1 文	2
Learning rate 0.15	
	🗸 ок

We note that we choose a high MAX ITERATION (5000). That does not matter because we can view the error rate decreasing and interactively stop the learning process in SIPINA.

🔉 Sipina Research Version							×			
File Edit Data Statistics Induction	method Anal	ysis View	Window Hel	p						
🖰 🗃 👺 🐴 🕒										
Attribute selection										
🖃 🗊 Class attribute 🛛 🔼		a01	a03	a04	a05	a06				
📃 🛄 class 📃	1	0.35	0.72	-0.24	0.48	-0.21				
Predictive attributes	2	0.35	0.72	-0.53	0.63	-1.03				
a01	3	0.35	0.72	-0.17	0.77	-0.25				
aU3	4	0.35	0.72	-1.12	0.77	1.92				
au4	5	0.35	0.72	-0.15	0.65	-0.10				
a05	6	0.35	-1.25	-0.12	-1.35	-0.51				
Learning method	7	0.35	0.68	-0.35	0.67	-0.71				
MethodName-Multi-Layer Perceptron	8	-2.87	-1.29	-0.10	-1.16	-0.25				
Hdl=21	9 🛌	0.35	0.64	-0.26	0.77	-0.56				
Max Iter=5000	10	0.35	-1.33	-0.28	-1.16	-0.25				
Max Error=0.01	117	0.35	0.72	0.06	0.77	-0.64				
Layers=1	12	0.35	0.72	-1.32	0.77	-2.42				
Layer_1=2	13	0.35	0.72	-0.46	0.77	-0.47				
	14	0.35	0.72	-2.07	0.77	0.23				
1 1	15	0.35	0.72	0.06	0.77	-0.19				
Examples selection	16	0.35	-0.26	-2.23	0.77	0.33				
231 examples selected	17	0.35	0.72	0.04	0.77	-0.27				
120 examples idle	18	-2.87	-1.29	-0.10	-3.08	-2.42	~			
	<					>				
Multi-Layer Perceptron							//			

### Learning process

We select the ANALYSIS / LEARNING menu. A new window appears, we can follow the error rate progression. A STOP button enables us to stop the processing.



**Error rate evolution** shows the error rate progression, we see that we obtain an error rate of 0.009 at the iteration 624. The confusion matrix is at the right part of the window.

The **STOP PROCESS** button is very important. We can stop the processing when we think that we cannot obtain a significant improvement in the remaining iterations.

In the bottom part of the window, when we select a neuron, the associated weights are displayed.

### Test error rate

In order to apply the prediction model on the test set, we click on the ANALYSIS / TEST menu.

Analysis Now W							
Define class attril	Define class attribute Select active examples						
	inpies						
Set weight held							
Set costs							
Learning							
Stop analysis							
Classification	•						
Test							
LIFT ROC curv	e						
Error measureme	nts 🕨 🕨						
Feature selection	•						
Personnal tests	•						

In the subsequent dialog box, we set the following option.

Apply classifier on							
Apply on							
C Learning set							
Inactive examples of Databases							
🗸 ок	X Annuler						

The confusion matrix appears in a new window. The test error rate is 0,0917.

😹 Confusion ma									
- class									
	good	bad							
good	76	0							
bad	11	33							
Cost : 0.0917									

#### Using a validation set

We can follow the learning process in SIPINA; the utilization of the validation set is more interesting. We can stop the learning process when the validation error rate does not decrease. The learning set is thus split into two parts: the first, says "training set", is used for the computation of the weights of the network; the second, says "validation set", is used for a "honest" evaluation of the error rate.

We close all the windows the WINDOW / CLOSE ALL menu. In order to include the utilization of a validation set in the learning process, we select a new algorithm: INDUCTION METHOD / STANDARD ALGORITHM menu, MULTILAYER PERCEPTRON (TEST ERROR RATE CONTROL) option.



The learning set size is 231. We set 70% of them as a training set (70% of 231 = 161 examples), and the remaining as validation set (231 - 161 = 70 examples).



We click on the ANALYSIS / LEARNING menu in order to execute the learning process. Two curves appear now in the chart. In some cases, the validation error rate may increase when we have overfitting.



The confusion matrix on the test set gives the following results (ANALYSIS / TEST menu).

🔉 Confusion ma	atrix : Test set o	n NEW.FDM 🔳 🗖 🔀
- class		
	dooq	bad
good	76	0
bad	11	33
Cost : 0.0917		

# Training a neural network with WEKA

When we execute WEKA (<u>http://www.cs.waikato.ac.nz/ml/weka/</u>), a dialog bow appears, which allows us to choose the execution mode of the software. We select the **KNOWLEDGE FLOW** mode. We have used the **3.5.1** version in this tutorial.



## Importing the dataset

The ARFF LOADER component enables to import the dataset.



## Splitting the dataset

The TRAINTEST SPLITMAKER (EVALUATION tab) enables to split the dataset into learning and test set.



We connect the ARFF LOADER component to this new component; we use the DATASET connection.

🔹 Weka KnowledgeFlow Environment									
	ataSources D	ataSinks 🛛 Filte	rs Classifiers Clu	sterers Associ	iations Evalu	ation Visualiza	ation	R	
	Evaluation							2	
6	Training	TestSet	CrossValidation FoldWater	TrainTest SolitWakar	Class	ClassValue Dicker	Cl		
:	Seculater	Maker	rolanaker	opriceater	Assigner	FICKEI	Perrorm		
Knowled	lge Flow Layou	rt							
								<u>^</u>	
			_					=	
	4	edata	dataSet					_	
		ARFF		NA COD					
	A	Edit	elete	TrainTest					
		c	onfigure	opiiteakei					
		Conn	ections						
		a in	stabet						
		Actio	ns					~	
<		S	art loading				>		
_ Status —									
Welcom	e to the Weka k	(nowledge Flo	N				Lo	g	

#### Learning algorithm and parameters

In WEKA, the last column of the dataset is the default class attribute; the other columns are the predictive attributes. If we have not this configuration, we must use the CLASS ASSIGNER component.

The supervised learning methods are in the CLASSIFIERS tab. We add the MULTILAYER PERCEPTRON component in the diagram. We click on the CONFIGURE menu in order to set the right parameters.



We set two neurons in the hidden layer (HIDDENLAYERS); the learning rate is 0.15 (LEARNING RATE); we do not use attribute transformation (NORMALIZE ATTRIBUTES = FALSE); the max number of iteration is set to 100 (TRAINING TIME); and we do not use a validation set (VALIDATION SET SIZE = 0).

We connect twice the TRAINTEST SPLITMAKER to this new component; we use the "training set" (1) and the "test set" (2) connections.



In order to visualize the weights of the network, we add the TEXT VIEWER component from the VISUALIZATION tab. We use the TEXT connection.



To launch the learning process, we click on the START LOADING menu of the ARFF LOADER component (1). And to display the results, we click on the SHOW RESULTS menu of TEXT VIEWER (2).



The weights of the Multilayer Perceptron appear in a new window.

👉 Text Viewer					
Result list	_ Text				
13:54:38 - Model: MultilayerPe	=== Classifier model ===	^			
	Scheme: MultilayerPerceptron				
	Relation: std_ionosphere.arff				
	Sigmoid Node O	_			
	Inputs Weights				
	Threshold -4.244848248550268				
	Node 2 4.168023899769535				
	Node 3 4.474384831290525				
	Sigmoid Node 1				
	Inputs Weights				
	Threshold 4.244855092664962				
	Node 2 -4.149867185140021				
	Node 3 -4.491785451471164				
	Sigmoid Node 2				
	Inputs Weights				
	Threshold 0.270235840429924				
	Attrib a01 2.7367610444517725				
	Attrib a03 -0.7144783417552728	~			
		>			

#### Test error rate

We must add two new components in the diagram in order to apply the network on the test set and visualize the confusion matrix. First, we add the CLASSIFIER PERFORMANCE EVALUATOR (EVALUATION tab) in the diagram and we use the BATCH CLASSIFIER connection.



Second, we add a new TEXT VIEWER component to visualize the results (TEXT connection).



We must execute again the learning process (START LOADING of the ARFF LOADER component). We click on the SHOW RESULTS menu of TEXT in order to display the results.

🖢 Text Viewer							
Result list	Text						
14:07:42 - MultilayerPerceptro	=== Evaluation re	sult ===					^
	Scheme: Multilaye Relation: std_ion	rPerceptron osphere.arff					
	Correctly Classified Instances Incorrectly Classified Instances			98 22	81.6667 %		
	Kappa statistic 0.6162						
	Mean absolute error 0.2005			0.2005			
	Root mean squared error 0.395			0.3938 40 6467 %			
	Relative absolute error 40.0407			40.0407 % 79.7035 %			
	Total Number of Instances 120			120	. 🔳		
	=== Detailed Accuracy By Class ===						
	TP Rate FP Rate	Precision	Recall	F-Measure	ROC Area	Class	
	0.955 0.358	0.771	0.955	0.853	0.792	good	
	0.642 0.045	0.919	0.642	0.756	0.792	bad	
	=== Confusion Mat	rix ===					
	a b < clas	sified as					
	643 a=goo	a 丿					
	1934   b = bad						
						) -	<u> </u>
						J .	7

There are 120 examples in the test set. The test error rate is 18.33%.

# Conclusion

We note in this tutorial that the logic of the training and the evaluation of a neural network is the same one, whatever the software used.

The implementation of a perceptron is finally rather simple. The interpretation of the results, in particular the comprehension of the weights of the network, is definitely more complicated.