



GINNet et DynNet

Plate-forme d'aide à la décision

<http://ginnet.gforge.inria.fr>

Marie TONNELIER Laurent BOUGRAIN

INRIA Lorraine - Equipe CORTEX

Juin 2007

Sommaire

1 Objectifs

- C'est quoi ?
- À quoi ca sert ?
- L'extraction de connaissances
- Tâches
- Modèles

2 Description des outils

- GINNet ou DynNet ?
- Interface
- Documentation
- Avantages et inconvénients
- Pour aller plus loin

3 Démon

- GINNet
- DynNet
- Interfaçage avec d'autres applications

Sommaire

1 Objectifs

- C'est quoi ?
- À quoi ça sert ?
- L'extraction de connaissances
- Tâches
- Modèles

2 Description des outils

3 Démonstrations

DynNet et GINNet : C'est quoi ?

- **Librairie** (DynNet) + **Interface graphique** (GINNet)
- Logiciels libres (licence CeCILL)
- **100% Java**
- Projet créé par Laurent Bougrain en 2002
- <http://ginnet.gforge.inria.fr>

À quoi ca sert ?

Plate forme d'aide à la décision :

Enseignement et découverte

- Didactique
- Facile à utiliser
- Déjà utilisée comme support de cours et sujet de stages

Recherche et développement

- Extraction d'informations en quelques clics
- Comparaisons de différentes méthodes d'analyse
- Ajout/test de nouvelles méthodes

Extraction de connaissances

Définition

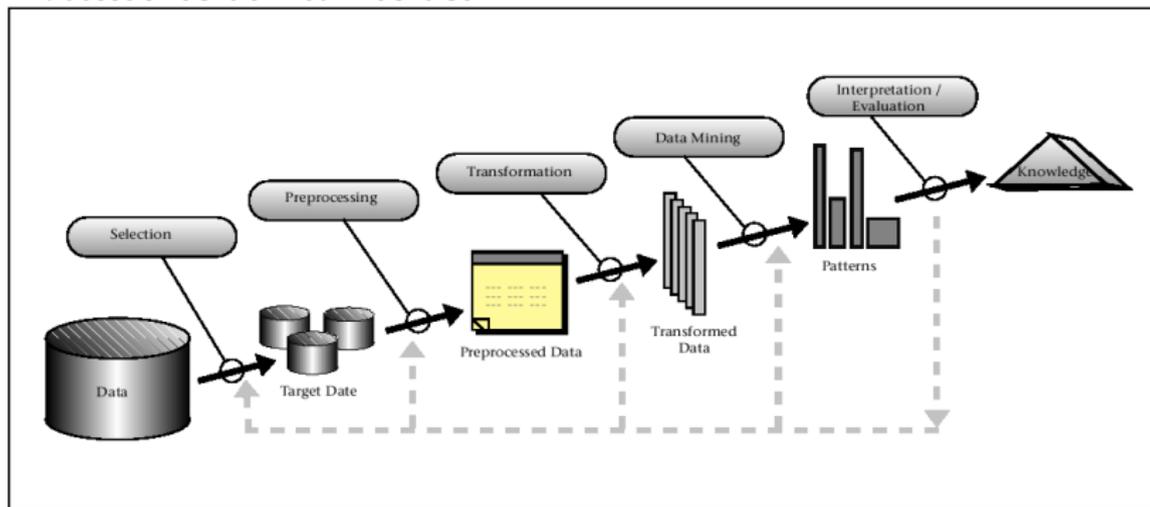
- Théorie et outils computationnels permettant d'**assister les humains dans l'extraction d'informations** utiles (connaissances) à **partir de bases de données volumineuses**, de manière objective, automatique et pertinente.
- **Prédiction, classification, description**, etc.

Applications

- **Applications diverses** : science, médecine, industrie, marketing, etc.
- **Exemples** :
 - Prédiction de la présence d'or dans le sous-sol en fonction de données géographiques et géologiques
 - Détection de crises d'épilepsie ou étiquetage des phases du sommeil en fonction de signaux EEG

Processus d'extraction de connaissances

Processus **itératif** et **interactif** :



An Overview of the Steps That Compose the KDD Process.

Détails du processus(1/6)

1. Sélection

Sélection des variables et des exemples à étudier

Dans GINNet

- Importation depuis fichiers textes et base de données MySQL
- Recherche, édition, fusion, sauvegarde
- Statistiques descriptives, histogramme, etc.

Détails du processus(2/6)

2. Prétraitements

- Traitement des valeurs manquantes
- Suppression du bruit, des doublons
- Ré-équilibrage des classes
- etc.

Dans GINNet

- Recherche des valeurs manquantes, aberrantes, redondantes
- Equiprobabilité
- Analyse en Composantes Principales
- Recodage

Détails du processus(3/6)

3. Transformation

Trouver la bonne représentation de données par rapport à la tâche à effectuer

- Réduction des dimensions
- Normalisation, standardisation
- Recodage

Dans GINNet

- Normalisation, standardisation
- Exportation en corpus
- Distribution des exemples en apprentissage et test
- Permutation
- Filtres de sélection de variables (Backward et Forward)

Détails du processus(4/6)

4. Data-mining

Tâche :

- Classification
- Régression
- Catégorisation (= clustering)

Choix d'un algorithme

Dans GINNet

- Réseau de neurones (Perceptron multi-couches, Cartes auto-organisatrices de Kohonen, Neural Gas et Growing Neural Gas, K-Means)
- Arbre de décision (C4.5, CART, ID3)
- Méta-apprentissage (Arcing, Bagging, Boosting)

Détails du processus(5/6)

5. Analyse et interprétation

- Performance / tests
- Analyse : matrices de confusion, intervalles de confiance

Dans GINNet

- Visualisation des modèles et performances en cours d'apprentissage
- Résultats en HTML (avec matrices de confusion, intervalles de confiance, etc.)
- Tests par exemple

Détails du processus(6/6)

6. Utilisation externe

- Application des modèles entraînés

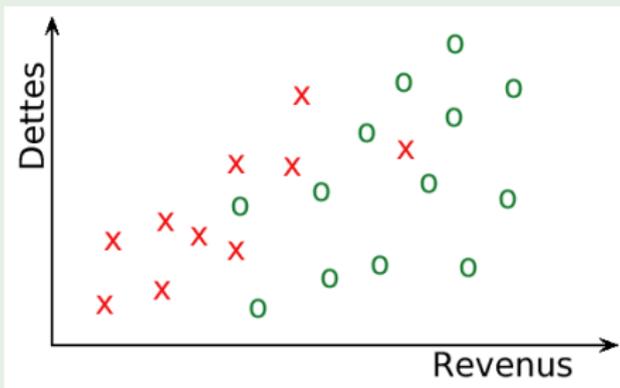
Dans GINNet

- Exportation en jar externe
- Sauvegardes en tous genres

Tâches (1/4)

Exemple illustratif : Les prêts

- Un point = une personne qui a reçu un prêt d'une banque
- 2 dimensions et 23 exemples :
 - En abscisse : Revenu de la personne
 - En ordonné : Montant de ses dettes
- 2 classes :
 - x : Personnes qui n'ont pas pu rembourser leur prêt
 - o : Personnes remboursant leur prêt correctement

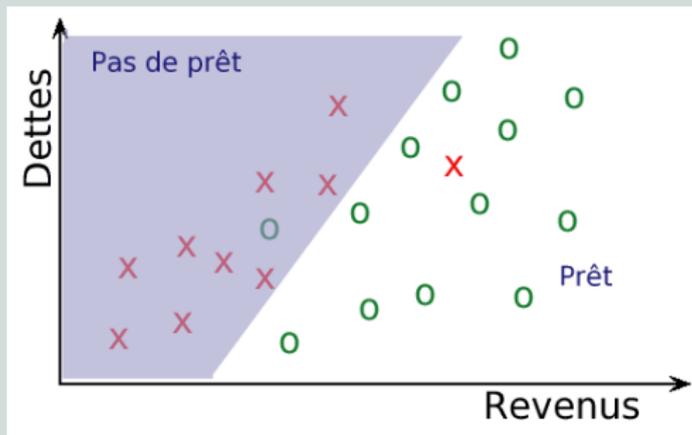


Tâches (2/4)

Classification

Apprendre une fonction qui associe les données à des classes prédéfinies

Exemple des prêts



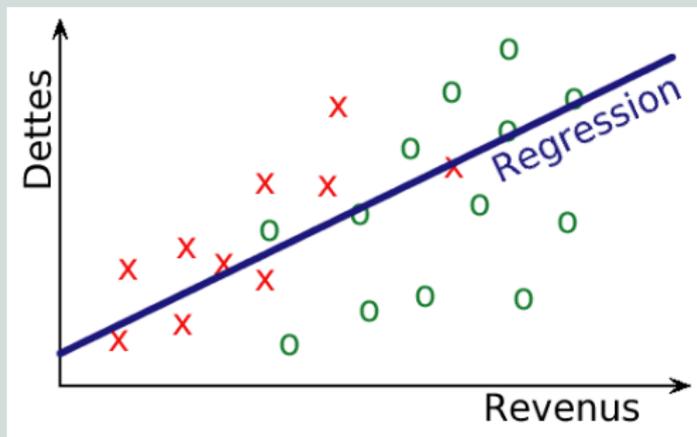
Autre exemple : Classifier les fleurs par espèce en fonction des dimensions de leur pétales et sépales

Tâches (3/4)

Regression

Apprendre une fonction qui associe les données à une variable réelle à prédire

Exemple des prêts



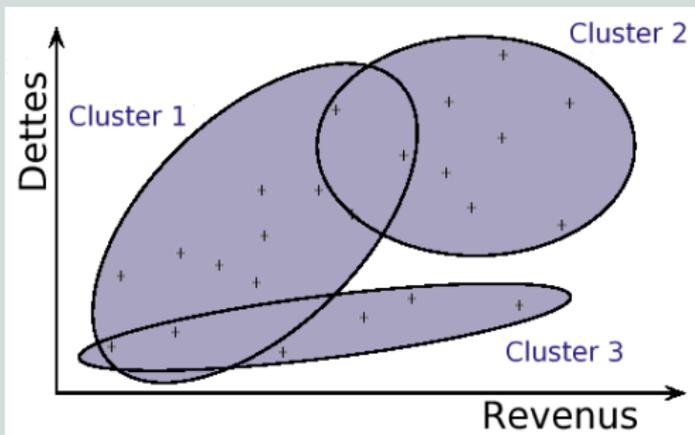
Autre exemple : Prédire la quantité de biomasse dans une forêt

Tâches (4/4)

Catégorisation

Essayer de ranger les données par catégories / clusters

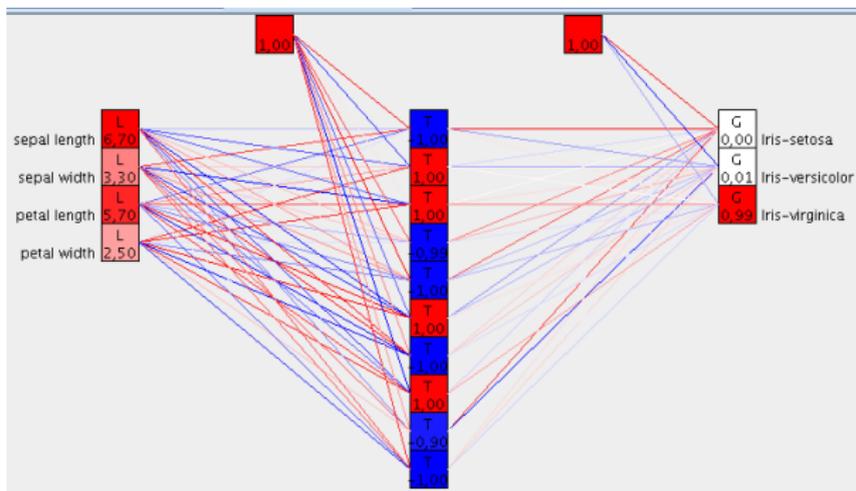
Exemple des prêts



Autre Exemple : Catégoriser les consommateurs d'un produit

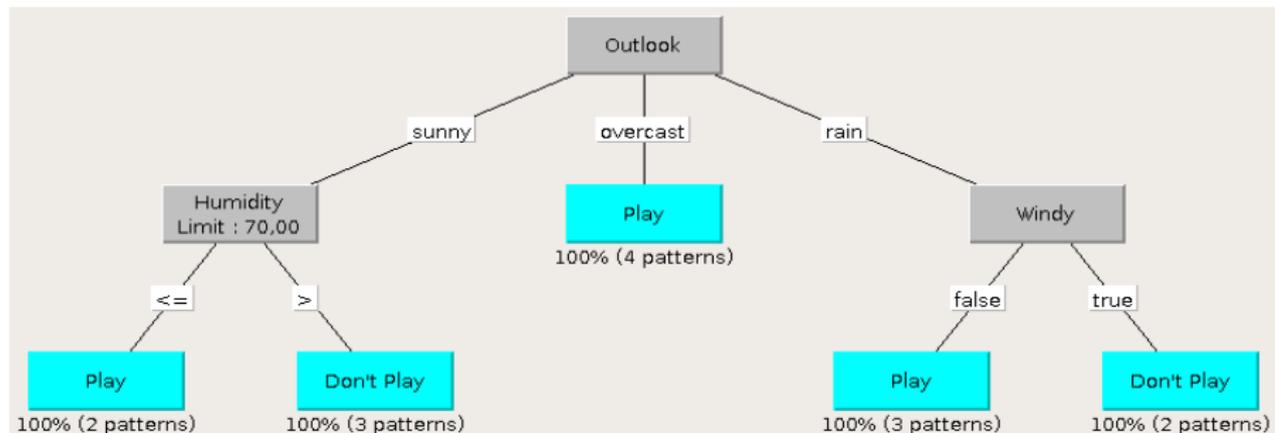
Modèle 1/3 : Réseaux de neurones

- **Modèle de calcul très schématiquement inspiré du fonctionnement de neurones biologiques**
- Apprentissage par expérience
- Capacité de classification et de généralisation
- Boîte noire qui répond par une probabilité de certitude
- Tâches :
 - **Classification et régression** : Réseaux supervisés (Perceptron)
 - **Catégorisation** : Réseaux compétitifs (Kohonen, Neural Gas, etc.)



Modèle 2/3 : Arbres de décision

- Répartit une population d'individus en groupes homogènes, selon un ensemble de variables discriminantes et en fonction d'un objectif fixé et connu
- Intérêt : Modéliser simplement, graphiquement et rapidement
- Tâches :
 - **Classification** : C4.5, CART, ID3
 - **Régression** : CART



Modèle 3/3 : Méta-apprentissage (Agrégation de modèles)

- **Modèle combinant des classifieurs faibles, qu'il évalue pour créer un classifieur "fort"**
- Compense l'instabilité des perceptrons et optimise les performances
- Tâche : **Classification** : Arcing, Bagging, Boosting

SAMPLER								
Sub-learners:	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8
Learn errors:	33 %	19 %	29 %	23 %	35 %	23 %	29 %	47 %
Test errors:	33 %	7 %	33 %	0 %	67 %	13 %	47 %	60 %
Answers:	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa	Iris-versicolor	Iris-setosa	Iris-setosa	Iris-versicolor
Iris-setosa:	33 %	29 %	15 %	14 %	6 %	10 %	7 %	7 %
Iris-versicolor:	33 %	41 %	29 %	48 %	29 %	42 %	29 %	50 %
Iris-virginica:	33 %	30 %	56 %	38 %	65 %	48 %	64 %	43 %
COMBINER								
Marks:	0,69	1,51	0,82	1,32	0,46	1,14	0,93	0,29
RESULT								
Errors:	Learn: 8% Validation: 7% Test: 0%							
Votes:	Pattern #41 Iris-versicolor: 0.7557938 Iris-setosa: 6.4091434							
Result (Status):	Iris-setosa (Good)							

Sommaire

1 Objectifs

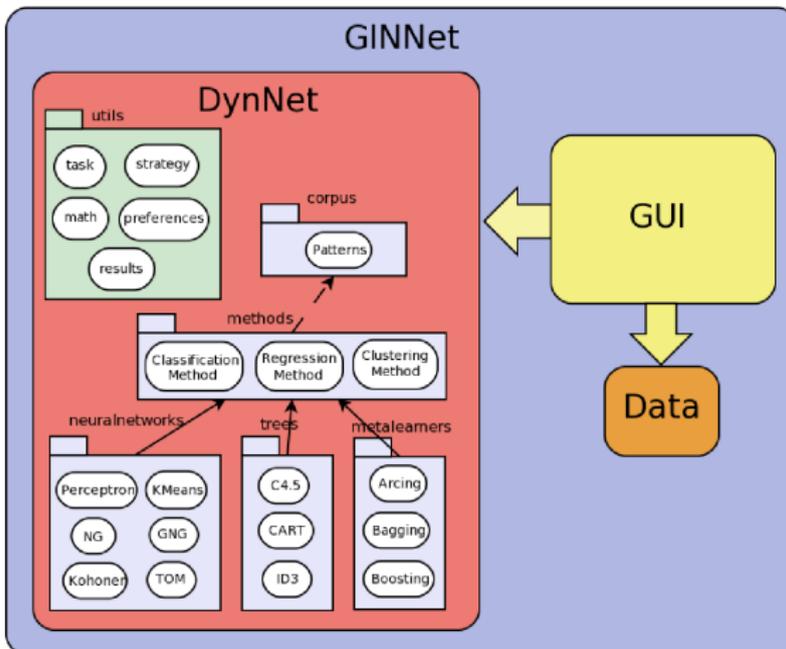
2 Description des outils

- GINNet ou DynNet ?
- Interface
- Documentation
- Avantages et inconvénients
- Pour aller plus loin

3 Démonos

GINNet ou DynNet ?

- **GINNet** (**G**raphical **I**nterface for **N**eural **N**etworks)
= une plate-forme complète d'extraction de données
- Utilisant la librairie **DynNet** (**D**ynamic **N**etworks)



Data Corpus Network Tree Meta-learner Result Help Screenshot Preferences Quit

Classification Corpus Perceptron Network Result

+
-
Hide

sepal length: 7
sepal width: 3
petal length: 7
petal width: 5

Iris-setosa
Iris-versicolor
Iris-virginica

Initialization

Init Random Min: Max: Reload best network

Saving

Save as external Jar

Learning

Learn
 Test
 Validation

Stopping function: Time max epoch:

Learning rate function: Constant

Backpropagation: on-line Momentum:

Test

Test All Save test results

Error: Learn: 0,15 Validation: 0,13 Test: 0,20

Cycle: 62 Clear

Documentations

Documentation en anglais :

Utilisateur

- Documentation intégrée
- Tutoriels
- Vidéo de démo (en français)
- Cette présentation

Développeur

- JavaDoc
- Charte
- Guide

Points forts et points faibles

Points forts

- Générique
- Processus complet
- Portable
- Libre
- Facilement accessible (en ligne) et facile à manier

Points faibles

- Ne supporte pas les grosses bases de données
- La librairie est complexe à appréhender, ce qui peut rendre difficile l'ajout de nouveaux modèles
- Bugs
- Fonctionnalités manquantes

Pour aller plus loin

<https://gforge.inria.fr/projects/ginnet/>

INRIA Le projet entier Rechercher Recherche avancée

Se déconnecter
Mon compte

Quick Jump To...

Accueil Ma page Arbre des projets Demande d'aide GINNet and DynNet

En bref Administration Forums Suivi Documents Annonces Sources Fichiers

GINNet is a graphical platform of data-mining, developed in Java and bases on DynNet library, allowing easy manipulation of neural networks, decision trees, etc.

- Development Status: [5 - Production/Stable](#)
- Environment: [Console \(Text Based\)](#), [Win32 \(MS Windows\)](#), [X11 Applications](#)
- Intended Audience: [Developers](#), [End Users/Desktop](#), [Other Audience](#)
- License: [CeCILL](#)
- Natural Language: [English](#)
- Operating System: [OS Independent](#)
- Programming Language: [Java](#)
- Topic: [Artificial Intelligence](#)

Enregistré le : 21/09/2005 09:13
Taux d'activité : 96.24%
Voir les [statistiques d'activité du projet](#).
View list of [RSS feeds](#) available for this project

Equipe-Projet

Administrateurs :
[Laurent Bougrain](#)
[Marie Tonnelier](#)

Développeurs :
[Nizar Kerkeni](#)

[\[Voir les membres\]](#)
[\[Demander à rejoindre le projet\]](#)

Derniers fichiers publiés

Paquet	Version	Date	Remarques / Surveillance	Téléchargement
GINNet	2.1.1	December 1, 2006		Téléchargement
DynNet	2.1.1	December 1, 2006		Téléchargement

[\[Voir tous les fichiers du projet\]](#)

Sommaire

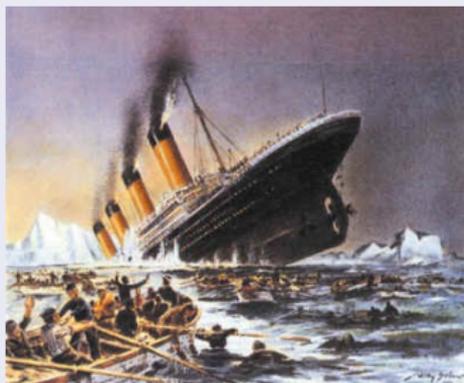
- 1 Objectifs
- 2 Description des outils
- 3 Démos**
 - GINNet
 - DynNet
 - Interfaçage avec d'autres applications

Démo de GINNet (1/2)

Exemple 1 : Le Titanic

Données pour 755 passagers : 442 survivants et 313 morts

- **Name** : tous différents → inutile
- **Class** : "1st", "2nd" ou "3rd"
- **Age** : de 0.17 à 71 (30 en moyenne)
- **Type** : "male", "female" ou "child"
- **Survived** : 1 ou 0



Enquêtons sur leur mort...

<http://ginnet.gforge.inria.fr/examples>

Démo de GINNet (2/2)

Exemple 2 : Reconnaissance de chiffres manuscrits

GINNet (Graphical Interface for Neural Networks & co) version 2.3

Classification Corpus Perceptron Network

Initialization: Int: Random Min: -1 Max: 1 Reload best network Save as external jar Save network

Learning: Stop Stopping function: min error 0.2000000 Learning rate function: Decreasing max time 100 Final learning rate: 0.0050000

Test: All Save test results Save test errors

Error: Learn: 0.00 Validation: 0.44 Test: 0.36

Cycle: 10 Clear

Démo de DynNet (1/2)

Exemple 1 : Création d'un arbre de décision

```
import java.io.File;
import fr.loria.cortex.ginnet.dynnet.corpus.ClassificationCorpus;
import fr.loria.cortex.ginnet.dynnet.methods.entries.TreeEntries;
import fr.loria.cortex.ginnet.dynnet.trees.TreeNode;
import fr.loria.cortex.ginnet.dynnet.trees.models.C45;
[... ]

// Create a C4.5 decision tree
ClassificationCorpus corpus =
    new ClassificationCorpus("examples/corpus/discrimination/golf.crp");
TreeEntries entries = new TreeEntries(corpus.getPatterns());
C45 c45 = new C45(entries, false);
c45.stoppingPercent = 100;

// Learn
TreeNode root = c45.build(4);
root.showTree();

// Test
c45.test();
float error = c45.getTestError();
System.out.println("\nTest error : "+ error);

// Save
c45.saveAsExternalJar(new File("myC45.jar"), "The C4.5 decision tree I created");
```

Démo de DynNet (2/3)

Exemple 2 : Création d'un Perceptron

[...]

```
// Load forecast corpus from file
AbstractFileCorpus corpus
    = new ClassificationCorpus("examples/corpus/discrimination/iris.crp");

// Create network topology : nbAttribute x 10 x nbTarget
int nbAttribute = corpus.getNbAttribute();
int nbTarget = ((TargetedPatterns)corpus).getNbTarget();
int [] sizes = { nbAttribute, 10, nbTarget };

// Create transfert functions for each layer : Linear x TanH x Linear
AbstractTransferFunction [] functions = { new LinearTransferFunction(),
    new TanHTransferFunction(), new LinearTransferFunction() };

// Create perceptron
Perceptron perceptron = new Perceptron(
    sizes, // layer sizes
    functions, // layer transfert functions
    0.005f, // lambda
    0.0f, // momentum
    false, // bias
    new ConstantLearningRateFunction(), // learning rate function
    new TimeStoppingFunction(), // stopping function
    new SimpleNumericEntries(corpus.getPatterns()), // network patterns
    false); // regression
```

Démo de DynNet (3/3)

Suite

```
// Set error function
AbstractErrorFunction function = new SEErrorFunction();
perceptron.errorFunction = function;

// Set stopping function to stop after 200 cycle
((TimeStoppingFunction)perceptron.getStoppingFunction()).setMaxEpoch(200);

// Separate learn and test patterns
corpus.separateLearnAndTestAndValidation(0.3f, 0.1f);

// Initialize network
perceptron.initializeConnections();

// Perform learn/test/validation cycles
perceptron.learn();
System.out.println("Learn error : " + perceptron.getLearnError());
perceptron.test();
System.out.println("Test error : " + perceptron.getTestError());

// Save as jar
perceptron.saveAsExternalJar(new File("myPerceptron.jar"), "No comment");
```

Exportation en jar (1/4)

Pourquoi exporter en jar ?

- Modèle exporté = une version réduite du modèle de GINNet :
 - Incapable d'apprendre
 - Capable de donner la réponse associée à un exemple
- Plus rapide
- Utilisable par une application, indépendamment de DynNet ou GINNet (en utilisant la mini librairie *ModelFromDynNet*)

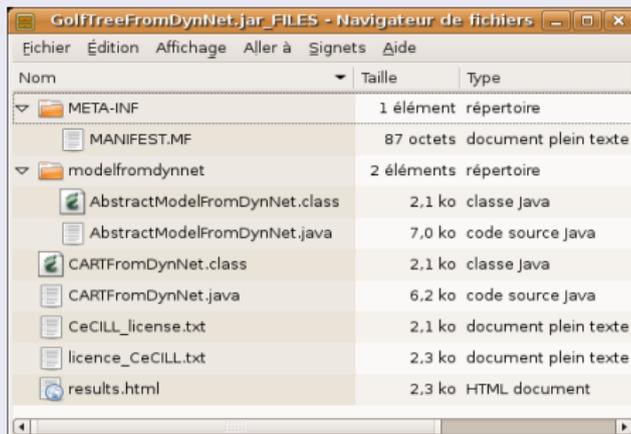
Comment exporter en jar ?

- **Dans GINNet** : bouton " *Save as external jar* "
- **Dans DynNet** : méthode `public void saveAsExternalJar(File jarFile, String comment)` de l'interface `fr.loria.cortex.ginnet.dynnet.methods.DataMiningMethod`, implantée par tous les modèles

Exportation en jar (2/4)

Que contient ce jar ?

- Les sources Java et les classes compilées
- Un fichier "manifest" rendant le jar exécutable
- Un fichier HTML décrivant les résultats obtenus pour le modèle
- La licence CeCILL



On peut également générer la JavaDoc

Exportation en jar (3/4)

Comment l'exécuter ?

- 1) Lancer la librairie ModelFromDynNet avec en argument les entrées du modèle :

```
java -jar lib/modelfromdynnet_v1.0.jar 1 2 3 4
```

- 2) Sélectionner le fichier jar à charger via l'interface graphique
- 3) Lire la sortie standard :

```
Model name : CART
```

```
Inputs :
```

```
1
```

```
2
```

```
3
```

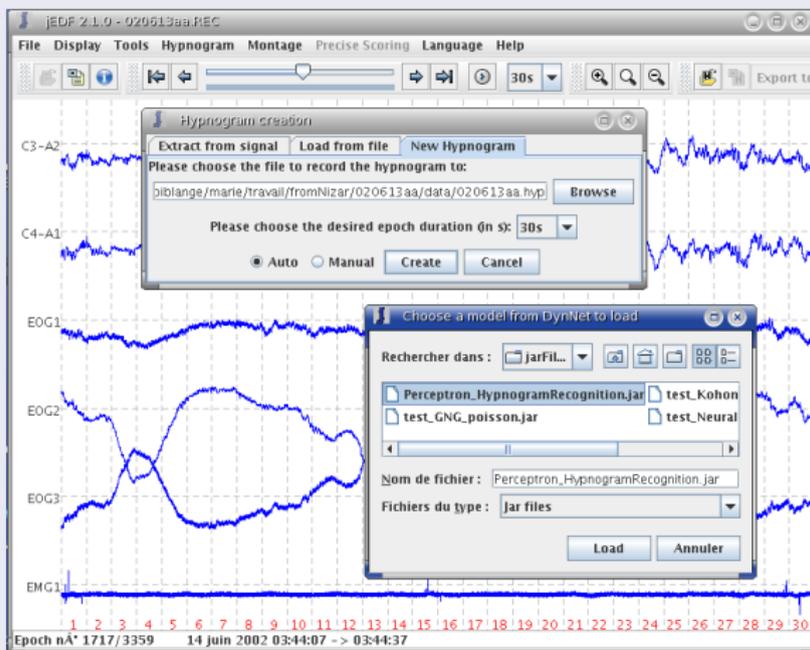
```
4
```

```
Output : Play
```

Exportation en jar (4/4)

Ou l'incorporer dans une autre application

Exemple dans jEDF : Génération automatique d'un hypnogramme à partir d'un signal EEG



Des questions ?

Références :

- "*From Data Mining to Knowledge Discovery in Databases*", Usama Fayyad, Gregory Piatetsky-Shapiro, and Pahravic Smyth
- jEDF : <http://www.loria.fr/~kerkeni/jEDF.html>

*Cette présentation a été réalisée avec LaTeX et la classe Beamer
(<http://latex-beamer.sourceforge.net/>)*

Récapitulatif des tâches

Classification

Apprendre une fonction qui associe les données à des classes prédéfinies

Exemple : Classer les fleurs par espèce

Regression

Apprendre une fonction qui associe les données à une variable réelle à prédire

Exemple : Prédire la quantité de biomasse dans une forêt

Catégorisation

Essayer de ranger les données par catégories / clusters

Exemple : Catégoriser les consommateurs d'un produit