

# Aufgaben Übung Künstliche Neuronale Netze Computational Intelligence

Ralf Mikut, Markus Reischl, Wilfried Jakob

Karlsruher Institut für Technologie, Institut für Angewandte Informatik

E-Mail: [ralf.mikut@kit.edu](mailto:ralf.mikut@kit.edu)

## Installation im Rechnerpool:

1. Mit RZ-Login einloggen

<https://sourceforge.net/projects/scixminer/files/scixminer2017a.zip/download>

herunterladen.

2. scixminer2017a.zip auf u:\ (und nicht auf u:\scixminer2017a) entpacken, die Inhalte sollten dann in u:\scixminer (nicht in u:\scixminer2017a\scixminer!) stehen  
Wenn es auf u:\ zu eng wird, beide Programme auf c:\temp speichern!
3. Datei uebung\_ci.zip aus ILIAS-System (Ordner: Übungen) herunterladen.
4. uebung\_ci.zip auf u:\ entpacken, die Inhalte sollten dann in u:\uebung\_ci (und nicht in u:\uebung\_ci\uebung\_ci) stehen
5. Matlab starten (im I-Pool Rechtsklick auf das Windows-Symbol, „Ausführen“, „matlab“).  
ACHTUNG: Wenn Matlab nicht gefunden wird, liegt das meistens an der falschen Auswahl beim Start des PC, es muss mit Windows 10 und nicht mit BW-Lehrpool gestartet werden

6. in Matlab ausführen (bei Speichern auf u:\):

```
>> cd u:\uebung_ci
```

```
>> scixminer
```

oder bei Entpacken in c:\temp

```
>> cd c:\temp
```

```
>> scixminer_ctemp
```

7. Bei Erfolg gezippte Dateien uebung\_ci.zip und scixminer2017a.zip löschen.

## Installation auf eigenem Laptop:

1. MATLAB muss installiert sein, u.U. gemäß KIT-Installationshinweisen gemäß Anleitung installieren:

[http://www.scc.kit.edu/downloads/sca/Registrierung\\_bei\\_MathWorks.pdf](http://www.scc.kit.edu/downloads/sca/Registrierung_bei_MathWorks.pdf)

Das funktioniert aber nur mit einer \*.kit.edu-Adresse.

Key Studenten (\*student.kit.edu)

75081-47343-68795-80444-65108

2. SciXMiner-Installation wahlweise über Executable für Windows (die kümmert sich auch um alle Pfadeinstellungen)

<https://sourceforge.net/projects/scixminer/files/scixminer2017a.zip/download>

oder analog zu Schritten 2-3 von "Installation im Rechnerpool" herunterladen (ACHTUNG! hier müssen dann später die SciXMiner-Pfade angepasst werden, siehe Schritt 4 im Folgenden)

3. Schritten 4-7 von "Installation im Rechnerpool" ausführen natürlich mit anderen Pfaden

4. Nur bei Download der zip-Datei in Schritt 2:

>> edit scixminer

und dann Pfade an Installationspfad scixminer anpassen

## Hinweise:

- Die Kompatibilität zu MAC-OS und Linux ist nicht gesichert.
- SciXMiner ist auch zu älteren MATLAB-Versionen kompatibel, getestet bis Version 2015b.
- U.U. Probleme mit chinesisch- und evtl. auch englischsprachigem Windows. Funktionieren sollte Umstellen der Windows-Sprache (Hinweis einer chinesischen Studierenden):

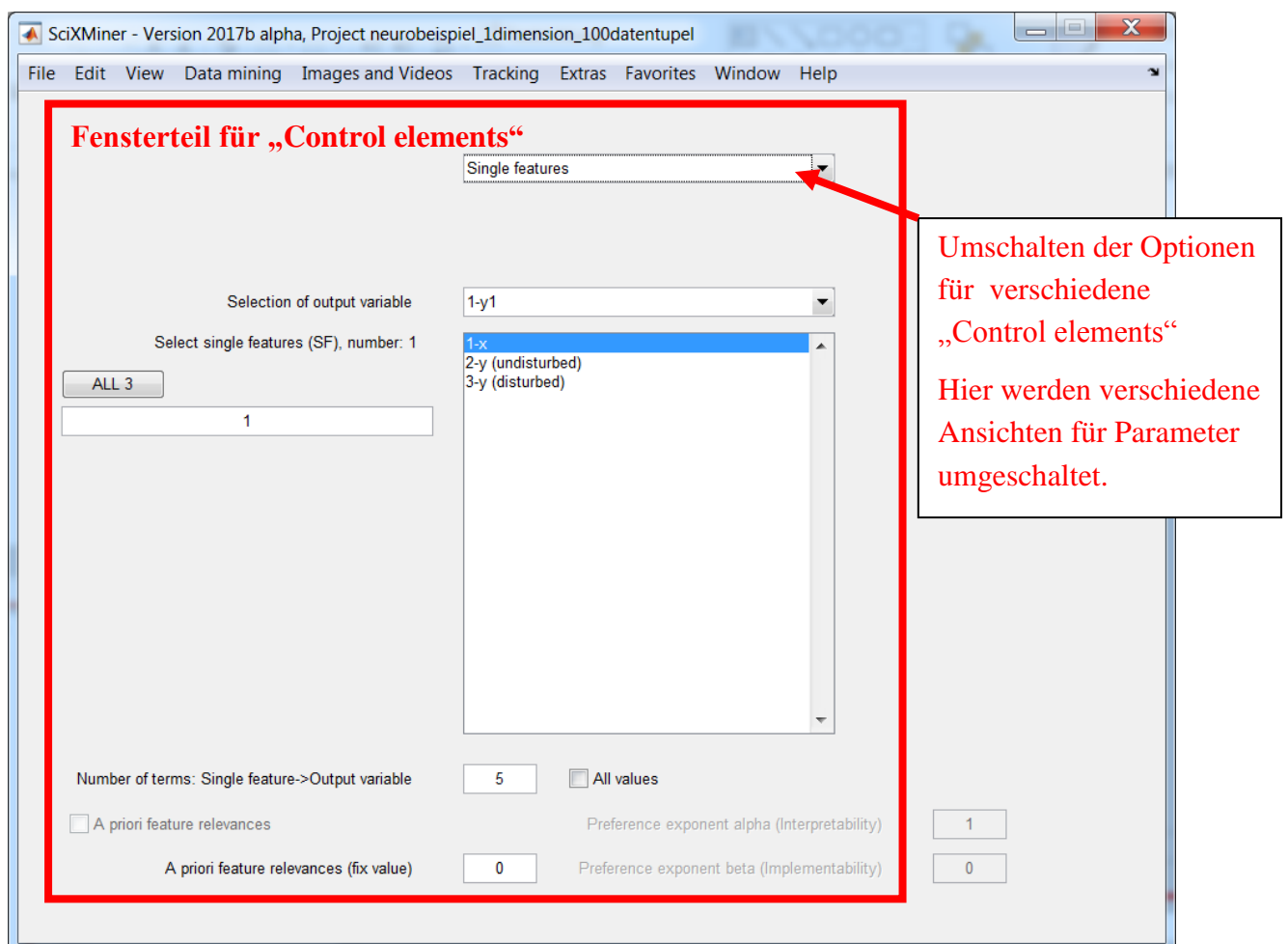
[Start>---<Control Panel>---<Clock, Language and Region>---<Region>---<Formats>-----<Format> bei Menü des Formats wählen wir einfach <Deutsch(de)>.

Dann Neustart von Matlab.

## Zielstellungen Übung:

- Kennenlernen der Bedienung von SciXMiner
- Entwurf eines einfachen eindimensionalen Regressionsmodells für Polynome und Künstliche Neuronale Netze
- Entwurf eines einfachen zweidimensionalen Regressionsmodells für Polynome und Künstliche Neuronale Netze
- Zusatzaufgaben bei Interesse

## Bedienung SciXMiner



- Start mit *scixminer*
- Menü (*Menu*) und Bedienelemente (*Control elements*)  
(Beispiel Bild oben: Auswahl von Einzelmerkmalen:  
*Control elements – Single features*)

- Bedienelemente oft über verschiedene Ebenen geschachtelt
- z.T. ähnliche Namen für Menüs (auszuführende Befehle) und Bedienelemente (zugehörige Parameter), deshalb stehen die jeweiligen Bezeichner immer dabei also  
*Menu - ...*  
*Control elements - ...*
- *Control elements – Project overview*  
gibt immer Kurzzusammenfassung über das Projekt
- Anzeige des SciXMiner-Handbuchs durch  
*Menu - Help – Show SciXMiner documentation (PDF)*
- Tooltips für die einzelnen Bedienelemente (Mauszeiger auf das gewünschte Menüelement bringen und einen Moment warten) und Menüelemente (letztere unten bei *Project overview*)
- Fehlbedienungen oft über CTRL+C (z.B. extrem lange Rechenzeiten) abbrechbar
- Wenn mal zu viele Fenster geöffnet sind, können sie am Einfachsten durch  
*Menu – Window – Close figures*  
wieder geschlossen werden.
- Möglichkeiten zur Automatisierung von Verarbeitungsketten durch Makro- und Batch-Files

Regressionen in SciXMiner:

Notwendige Schritte zur Bildung eines Regressionsmodells:

- Auswahl der Datentupel, die zur Modellbildung verwendet werden:  
*Menu – Edit – Select - ...*
- Auswahl der Merkmale, welche die unabhängigen Variablen im Modell darstellen:  
*Control elements - Single features - ...*  
oder  
*Control elements – Time series - ...*
- Grundlegende Einstellung des Regressionsverfahrens:  
*Control elements - Data mining: Regression*  
(erneute Merkmalsauswahl, Normierung, Regressionsverfahren, Zielgröße / abhängige Variable der Regression)
- Feintuning der Modellbildung unter  
*Control elements – Data mining: Special methods*

- Entwurf des Regressionsmodelle:  
*Menu – Data mining – Regression – Design*  
Beachte: Ausschlaggebend für das verwendete Verfahren ist die entsprechende Einstellung in  
*Control elements - Data mining: Regression*
- Anwendung eines bereits entworfenen Regressionsmodells auf andere Datentupel:  
*Menu – Data mining – Regression – Apply*
- Wurde ein Modell angewandt, kann das Ergebnis visualisiert werden  
*Menu – View – Regression - ...*

Führen Sie die folgenden Aufgaben durch und protokollieren Sie die Ergebnisse in ein Word-Dokument durch Reinkopieren ausgewählter Bilder und Ergebnisse aus Textdateien!

## Aufgaben:

### Eindimensionale Modellbildung in SciXMiner:

- Datei laden:

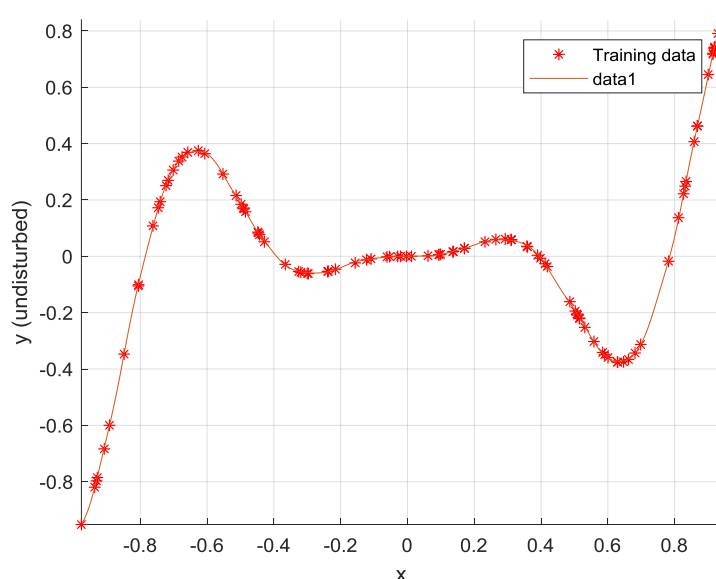
*Menu - File - Load project:*

u:\uebung\_ci\neurobeispiel\_1dimension\_200datentupel.prjz

Funktion:  $y = \sin 8x * x^2$

Lerndatensatz mit 200 Datentupeln (die ersten 200)

Testdaten ohne Extrapolation (x zwischen -1 und 1) und  
mit Extrapolation (x zwischen -2 und 2)



- Modell anlernen (zunächst Polynom)

Mit dem folgenden Befehl das Makro zum Anlernen ausführen:

*Menu - Extras - Play macro:*

*anlernen\_1D\_ungestoert.makrog*

(ab der 2. Ausführung auch unter *Menu - Favorites* angezeigt)

Das Makro wählt die ungestörte Ausgangsgröße für die Regression aus und fasst die Funktionen

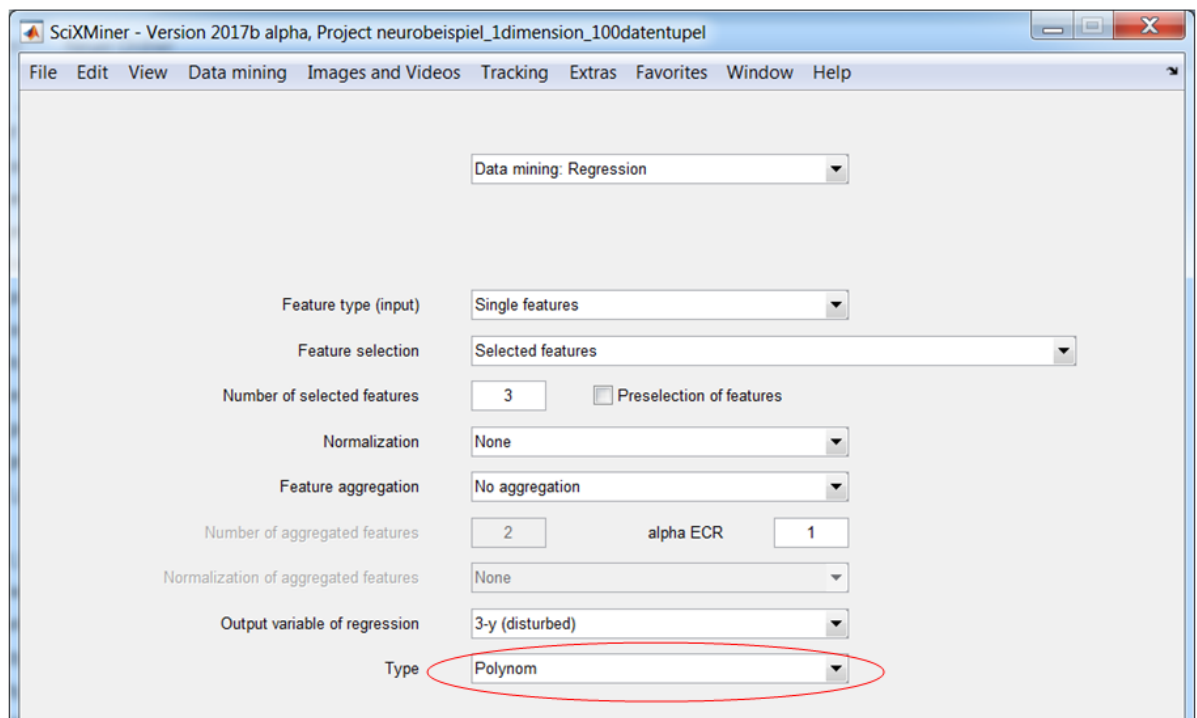
- Auswählen der Lerndaten
- Entwurf des Regressionsmodells
- Anwendung des Regressionsmodells
- Visualisierung der Eingangsgröße(n), der Ausgangsgröße und der geschätzten Regressionsfunktion (2D oder 3D)

zusammen.

**ACHTUNG!** Nach der Veränderung von Parametern müssen Makros immer neu ausgeführt werden, um einen Effekt zu erzielen!

## Fragen:

1. Kann ein Polynom 1., 2. oder 3. Ordnung das Problem lösen?



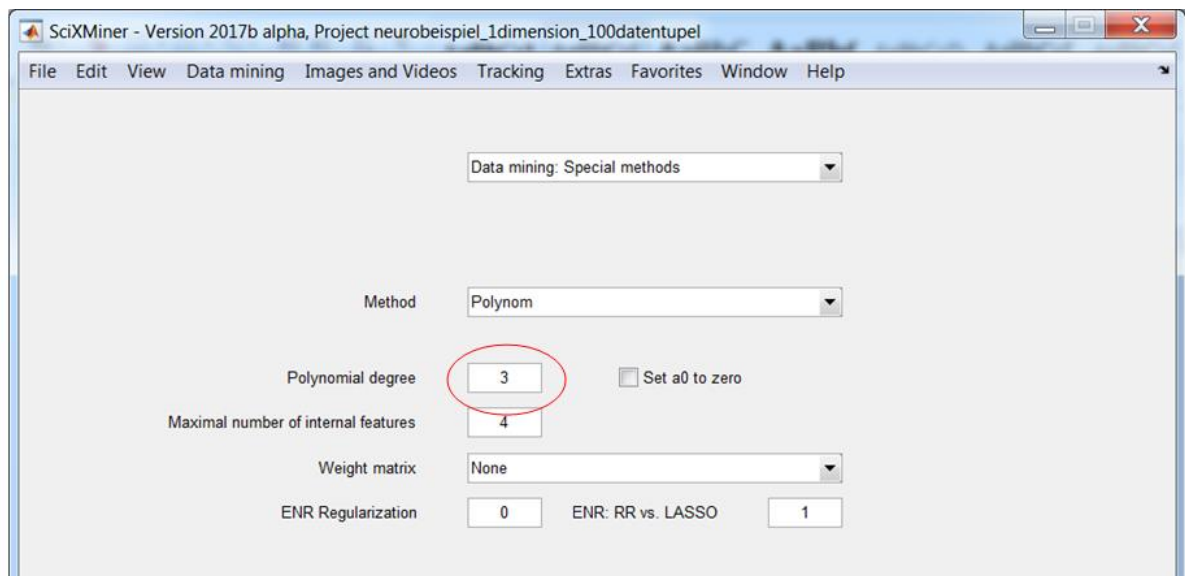
Dazu in

*Control elements – Data mining: Special methods – Method*

den Wert

*Polynom*

auswählen und die folgenden Einstellungen wählen:



Danach wieder das Makro anlernen\_1D\_ungestoert.makrog ausführen.

Die Ergebnisse stehen dann im MATLAB-Kommandofenster:  
Beispielausgabe:

*Fitness of regression:*

*Mean absolute value: 0.0254274*

*Fitness improvement for the mean absolute error to the trivial estimation (0-1):  
0.959164*

*Correlation coefficient between true value and estimation: 0.999543*

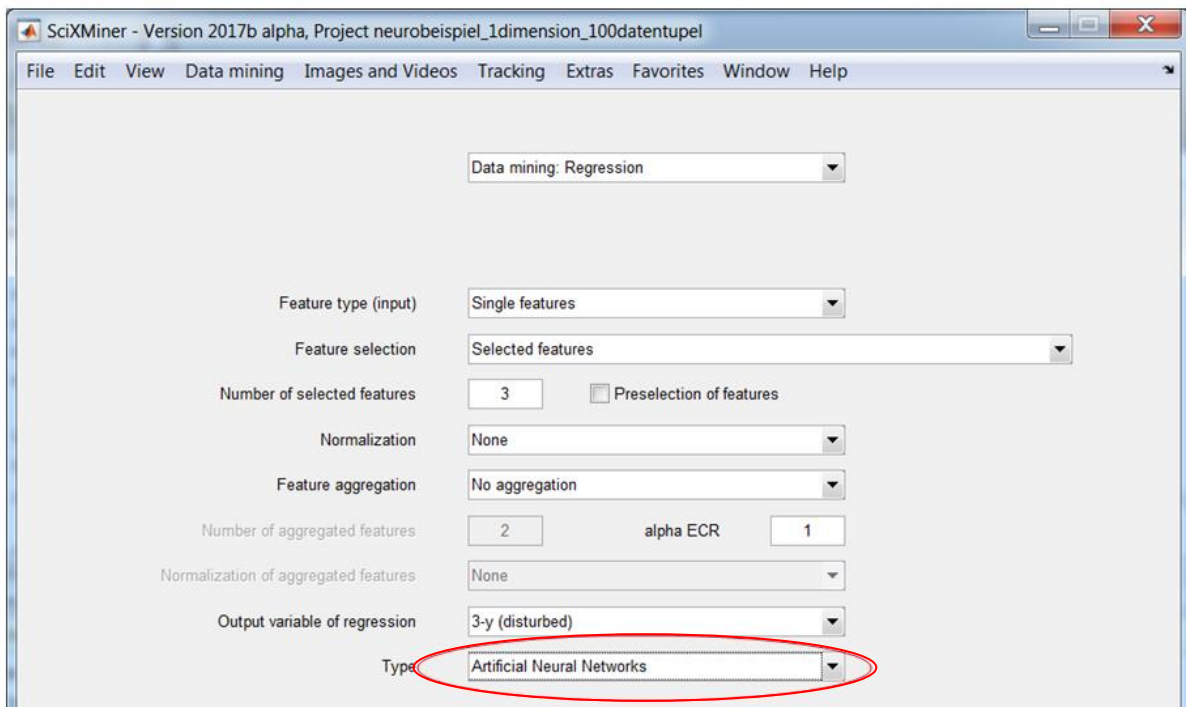
*Coefficient of determination  $R^2$  between true value and estimation: 0.999086*

Wichtig ist z.B. der Mittlere absoluter Fehler (*mean absolute value*) als Maß für die Abweichung zwischen wirklicher Ausgangsgröße und Ausgangsgröße des Modells (idealer Wert 0).

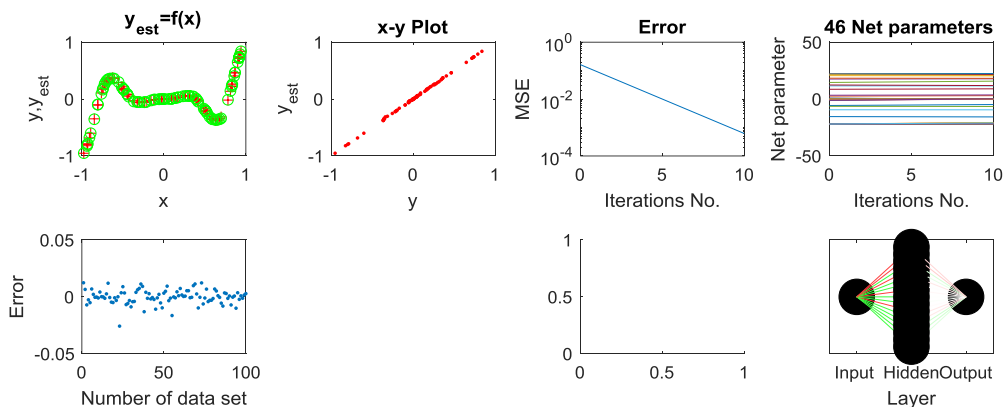
Also bitte vergleichen, welches Polynom den geringsten Fehler schafft und auch die Ergebnisbilder dazu anschauen.

2. Kann ein Neuronales Netz das Problem lösen?  
ACHTUNG! Dazu muss das Bedienelement für  
*Data Mining: Regression - Type*  
von *Polynom* (siehe Bild) auf Neuronale Netze (*Artificial Neural Networks*)  
umgeschaltet werden, sonst werden immer noch Polynome gelernt.





Anzeige Zwischenergebnisse:



Danach vergleichen, ob das Neuronale Netz einen geringeren Fehler (mean absolute value) schafft und auch die Ergebnisbilder dazu anschauen.

3. Was sind geeignete Einstellungen für
  - Anzahl Neuronen pro Schicht (*Number of neurons per layer*),
  - Anzahl Lernepochen (*Number of learning epochs*),
  - Fehlerziel (*Error goal*),
  - MLP-Lernalgorithmus (*MLP-Lernalgorithmus*) und
  - Normierung (*Normalization*, siehe Bild bei Frage 2)?

Was ändert sich in der Visualisierung und beim Fehler? Was ist erfolgreich?

Anmerkung: Bei der Wahl von

*Artificial Neural Network: Type*

kann sowohl 'MLP' (mit mehr Einstellmöglichkeiten) oder die neuere Variante 'Feedforwardnet (MLP)' (das ist nichts anderes als ein speziell

vorkonfiguriertes MLP-Netz) verwendet werden. Die genannten Einstellmöglichkeiten beeinflussen beide Varianten.

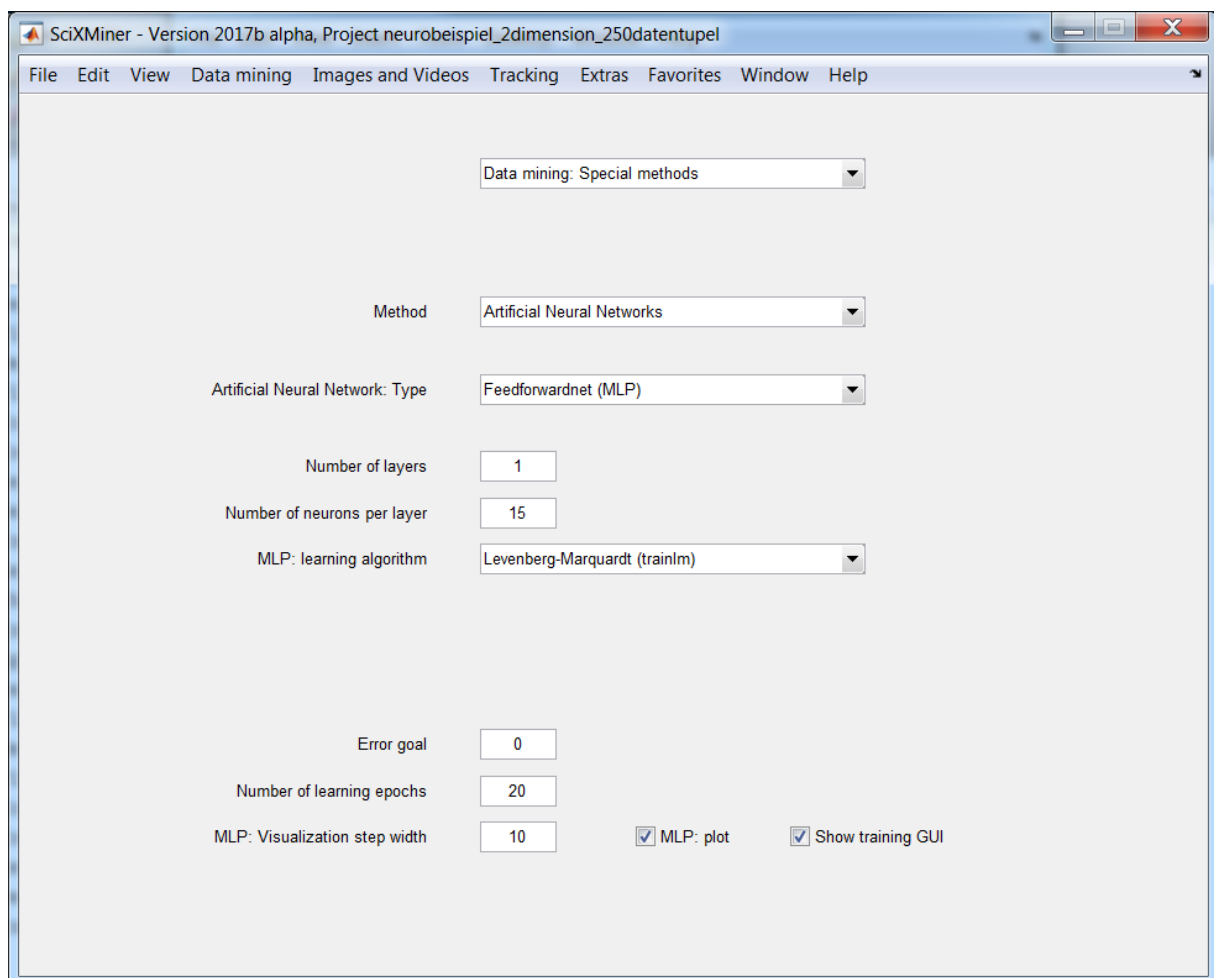
Zur Wahl der Einstellungen muss

*Control elements – Data mining: Special methods – Method*

auf

*Artificial Neural Networks*

gestellt werden (außer für die Normierung, die ist bei *Control elements - Data mining: Regression*).



4. Was passiert bei den besten Einstellungen beim Anlernen mit gestörten Eingangsdaten?  
Interpretieren Sie die Ergebnisse!  
*Menu - Extras - Play macro:*  
*anlernen\_1D\_gestoert.makrog*

Das Makro wählt die gestörte Ausgangsgröße für die Regression aus, ansonsten passiert das gleiche wie bei den ungestörten Ausgangsgrößen.

5. Was passiert bei den besten Einstellungen bei der Anwendung auf (unbekannte) Testdaten? Interpretieren Sie die Ergebnisse!  
*Menu - Extras - Play macro:*  
*anwenden\_testdaten.makrog*  
(ab der 2. Ausführung auch unter *Menu - Favorites* angezeigt)  
Das Makro fasst die Funktionen
  - Auswählen der Testdaten
  - Anwendung des Regressionsmodells
  - Visualisierung der Eingangsgröße(n), der Ausgangsgröße und der geschätzten Regressionsfunktion (2D oder 3D)zusammen.
6. Was passiert bei den besten Einstellungen bei der Anwendung auf (unbekannte) Testdaten mit Extrapolation? Interpretieren Sie die Ergebnisse!  
*Menu - Extras - Play macro:*  
*anwenden\_testdaten\_1D\_extrapolation.makrog*

## **Zweidimensionale Modellbildung in SciXMiner:**

Führen Sie die Fragen für 1. und 2. des eindimensionalen Beispiels analog für die zweidimensionale Funktion durch

*Menu - File - Load project:*

u:\uebung\_c\neurobeispiel\_2dimension\_250datentupel.prjz

Funktion:  $y = \exp(-10*((x_1-0.25))) * x_2$

Lerndatensatz mit 250 Datentupeln (die ersten 250)

Testdaten ohne Extrapolation

Anlernen: *anlernen\_2D.makrog*

Anwenden: *anwenden\_testdaten.makrog*

Interpretieren Sie die Ergebnisse!

## **Interessante Zusatzfragen (bei ausreichend Zeit):**

- eindimensionales Beispiel: Was passiert bei einer zu kleinen Anzahl von Lerndaten (z.B. 10...20)? Datentupel 1- N manuell auswählen unter *Menu – Edit – Select – Data points using numbers* und dann Modell neu entwerfen
- zweidimensionales Beispiel:  
Fragen 3 und 5 für das eindimensionale Beispiel
- Zeitreihenprognose für Vorhersage Energieverbrauch  
(Projekt: u:\scixminer\prj\building.prjz),  
TIPPS: Dokumentation lesen und Betreuer fragen....