



Yet Another Pattern Recognizing Network

Projekt im Rahmen von MPG I 3

Benutzerhandbuch

## Inhaltsverzeichnis:

1. Vorwort	S. 3
2. Team	S. 4
3. GUI	S. 5
3.1 Hauptbildschirm	S. 6
3.2 Importieren von Dateien	S. 9
3.3 Erstellen, Laden und Speichern von MLPs	S. 10
3.4 Subsampling	S. 11
3.5 Training	S. 12
3.6 Klassifizierung	S. 14
4. Beispieldurchgang	S. 15

# 1. Vorwort

Das vorliegende Programm ist ein im Rahmen des MPGI 3 Praktikums erstellter Mustererkenner, der mittels maschinellern Lernen ein Neuronales Netzwerk so trainiert, dass dieses unbekannte Daten gleichen Typs wie die bereits bekannten korrekt klassifizieren kann. Angewendet wird er auf Bild- und Sounddateien, die handgeschriebene Ziffern beziehungsweise gesprochene Vokale enthalten.

## 2. Team

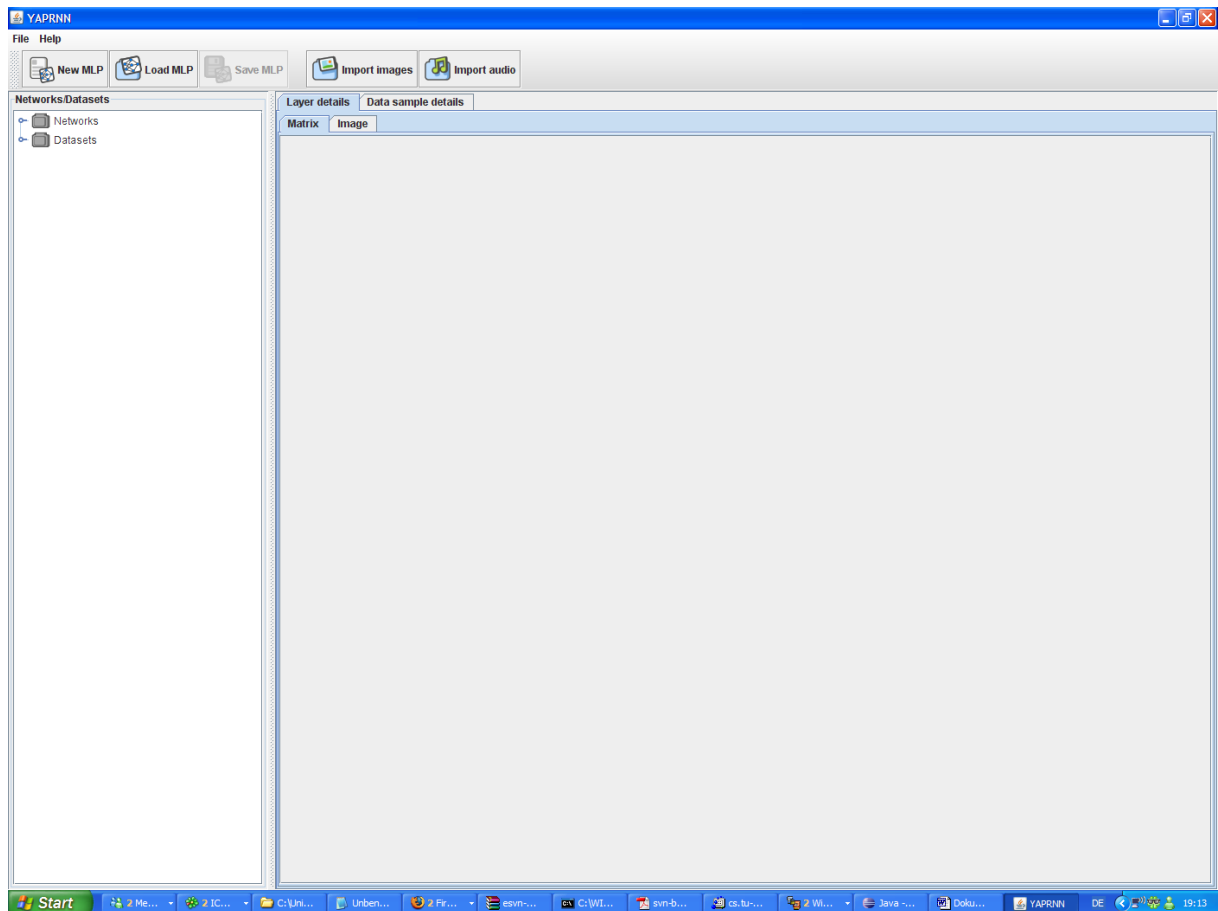
Das Programm wurde von sieben Studenten erarbeitet. Diese waren im Einzelnen:

Leo Bronstein (Datenvorverarbeitung)  
Sascha Sauerstein (Datenvorverarbeitung)  
Felix Rodemund (Multilagenperzeptron)  
Oleg Boroda (Multilagenperzeptron)  
Falko Schwabe (GUI)  
Tim Kwanka (GUI, zwischenzeitlich ausgestiegen)  
Andreas Gruner (Projektleitung und Dokumentation)

### 3. GUI

YAPRNN wird über eine visuelle Benutzeroberfläche gesteuert, deren einzelne Elemente im Folgenden beschrieben werden.

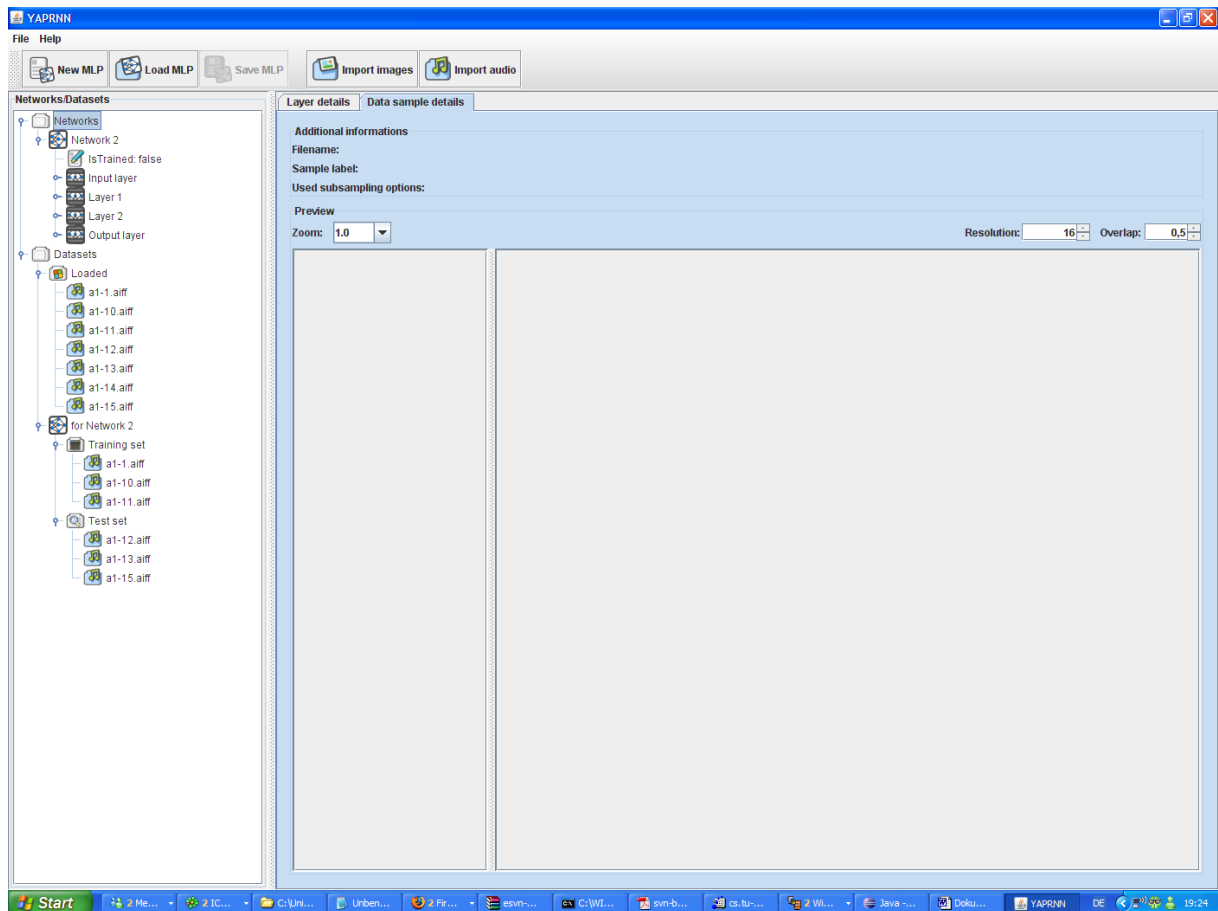
## 3.1 Hauptbildschirm



Auf dem Hauptbildschirm besteht die Möglichkeit, das Multilagenperzeptron und die geladenen Daten zu betrachten und zu verändern. Hierzu gibt es das obere Menü mit den Optionen New MLP, Load MLP, Save MLP sowie Import Images und Import Audio.

Im Baumenü auf der linken Seite können bereits geladene MLPs und Daten selektiert werden. Im rechten Fenster können selektierte Daten bzw. einzelne Layer des MLPs betrachtet werden.

### 3.1.1 Baumenü

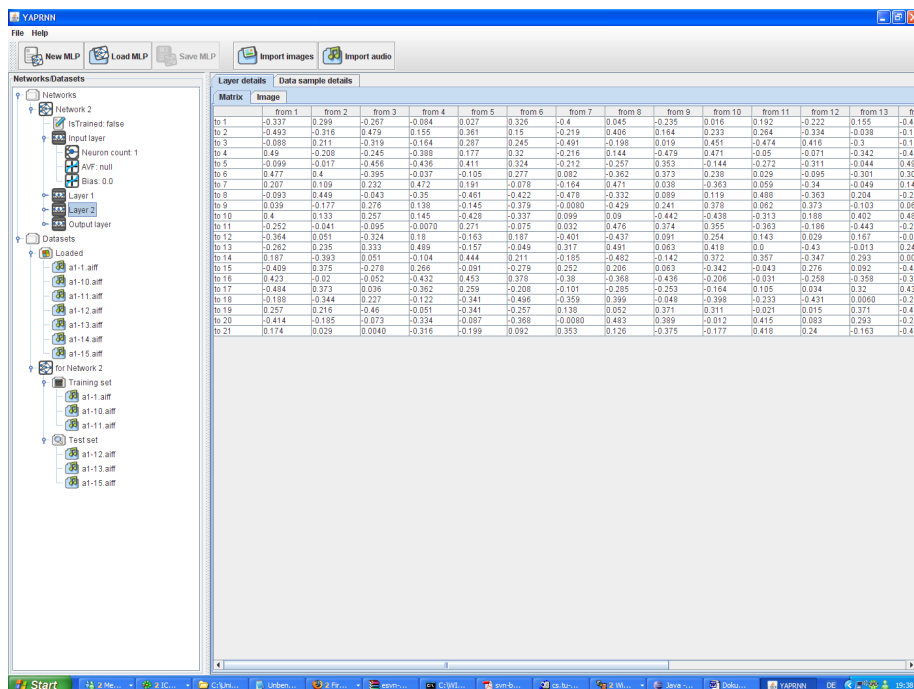
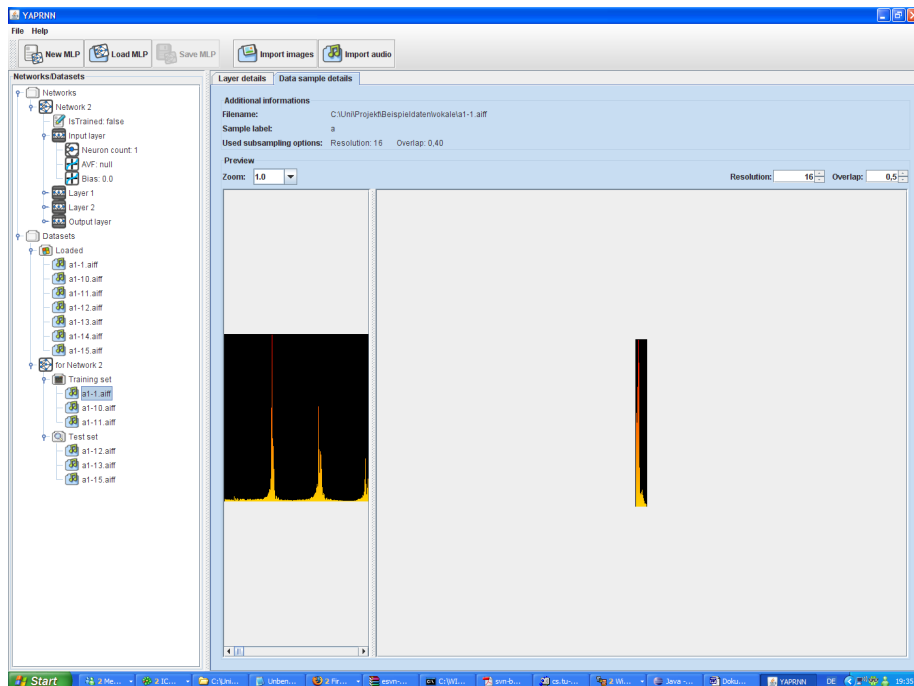


Im Baumenü werden die geladenen Daten und MLPs dargestellt. Per Rechtsklick auf selektierte Einträge kann ein Kontextmenü aufgerufen werden, das je nach Eintrag verschiedene Optionen zur Verfügung stellt:

- Netzwerke können trainiert oder auf ihre Initialwerte zurückgesetzt werden
- In den Unterpunkten der einzelnen Layer eines Netzwerks können durch die Editfunktion die Neuronenzahl, die Aktivierungsfunktion und der Bias gesetzt werden; Diese Funktion ist nicht verfügbar, wenn das Netz bereits mit Daten trainiert wurde, da dies das Netz unbrauchbar machen würde. Ist es gewünscht, die Konfiguration eines bereits erstellten Netzes zu ändern, muss dieses zuerst zurückgesetzt werden.
- Daten, sowohl bereits Netzen zugeordnete als auch freie können gesubsampelt und klassifiziert werden
- die Datasets für ein Netzwerk können mit zufällig gewählten Trainings- und Testdaten bestückt werden

Alle entsprechenden Optionen und Parameter sind dabei relativ selbsterklärend.

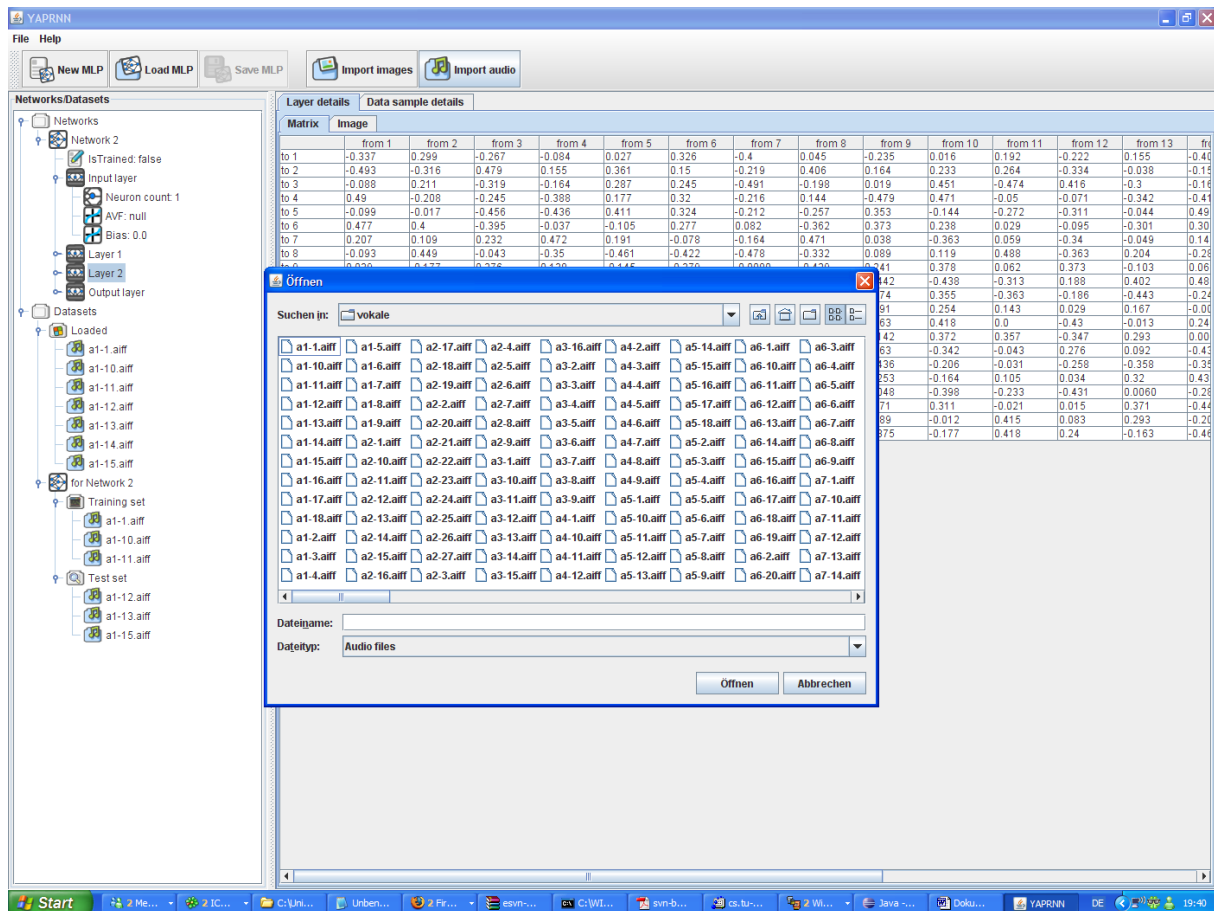
## 3.1.2 Rechtes Fenster



Das rechte Fenster enthält die Repräsentationen von im Baummenü ausgewählten Daten. So ist es hier etwa möglich, sich (über den Reiter Data Sample Details) sich ein ausgewähltes Datum sowohl in Originalgestalt als auch in gesubsampelter Form anzeigen zu lassen oder die Gewichtungen eines Layers entweder als Matrix oder in graphischer Form anzeigen zu lassen.

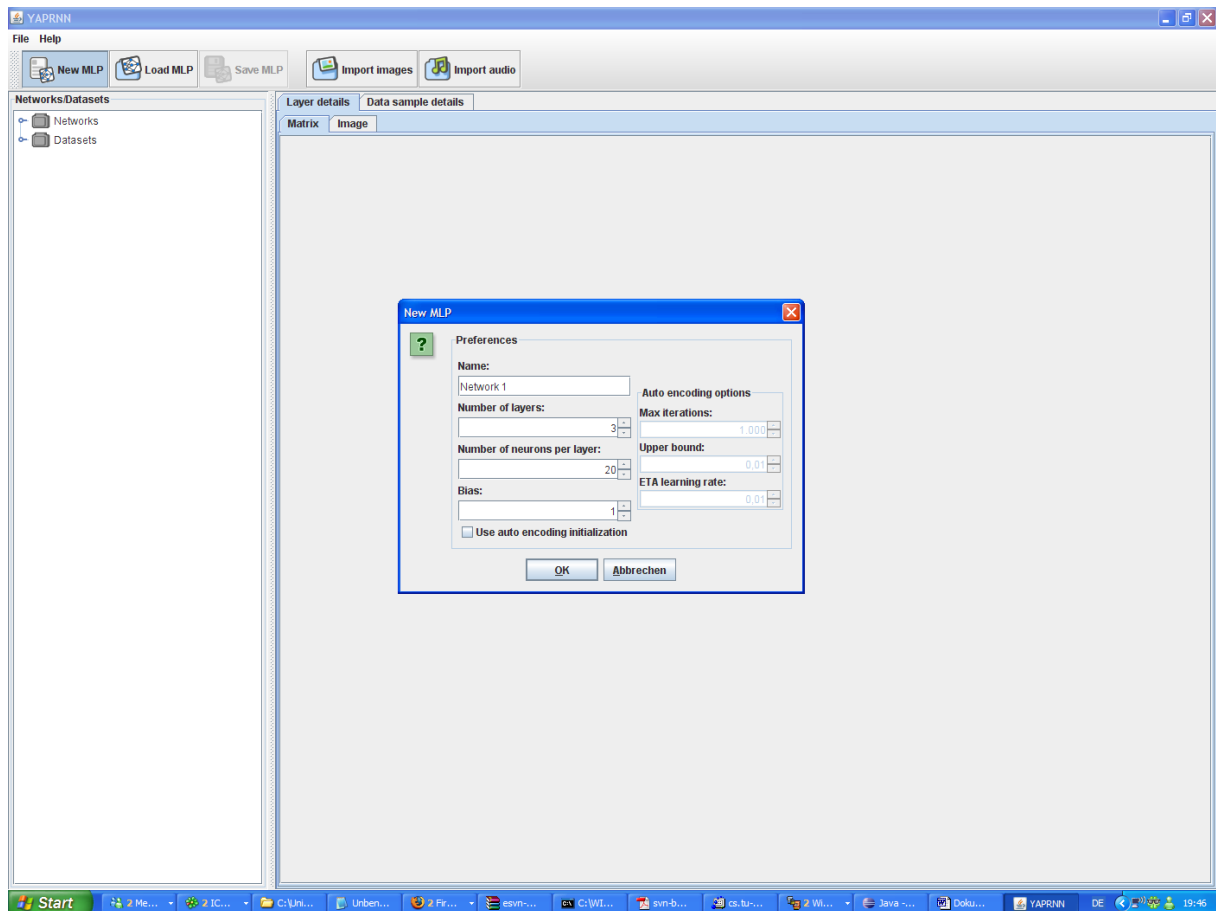


## 3.2 Dateiimport



Über die Buttons Import Images bzw. Import Audio können Audio- und Bilddateien eingelesen werden. Import Audio führt dabei direkt zu einem Dateiauswahldialog, in dem mehrere Sounddateien selektiert werden können, während Import Images zuerst zu einem weiteren Dialog führt, in dem erst eine Labeldatei und dann ein Imagearchiv ausgewählt werden muss.

### 3.3 Erstellen, Laden und Speichern von MLPs



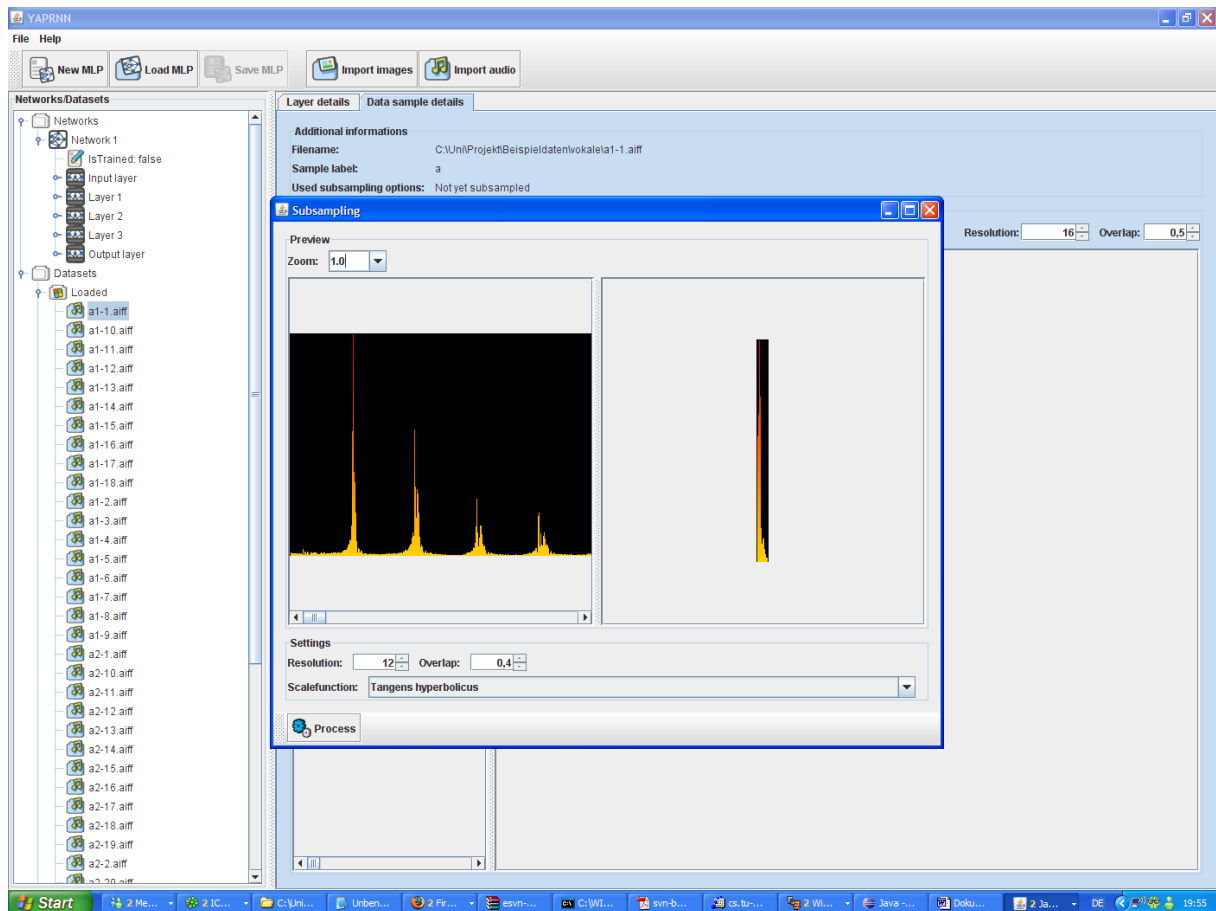
Über die Buttons New MLP, Load MLP und Save MLP können MLPs neu erstellt, geladen und gespeichert werden.

Der Button New MLP führt dabei zu einem Dialog, in dem eine Anfangskonfiguration für das zu erstellende MLP angegeben werden muss. Dabei sind ein Name, eine Anzahl an Layern, eine anfängliche Anzahl an Neuronen pro Layer und ein Bias einzutragen (die Neuronenanzahl und das Bias sind noch über das Baumenü später individuell zu verändern). Sofern es gewünscht wird, kann das Netz zudem per Autoencoder vortrainiert werden. In diesem Fall müssen hierfür weitere drei Parameter eingegeben werden.

Load MLP ruft einen Dateidialog auf, in dem ein gespeichertes MLP ausgewählt werden muss und dann geladen wird.

Save MLP ist nur verfügbar, wenn man ein MLP im Baumenü angewählt hat und speichert dieses unter einem anzugebenden Pfad ab.

## 3.4 Subsampling

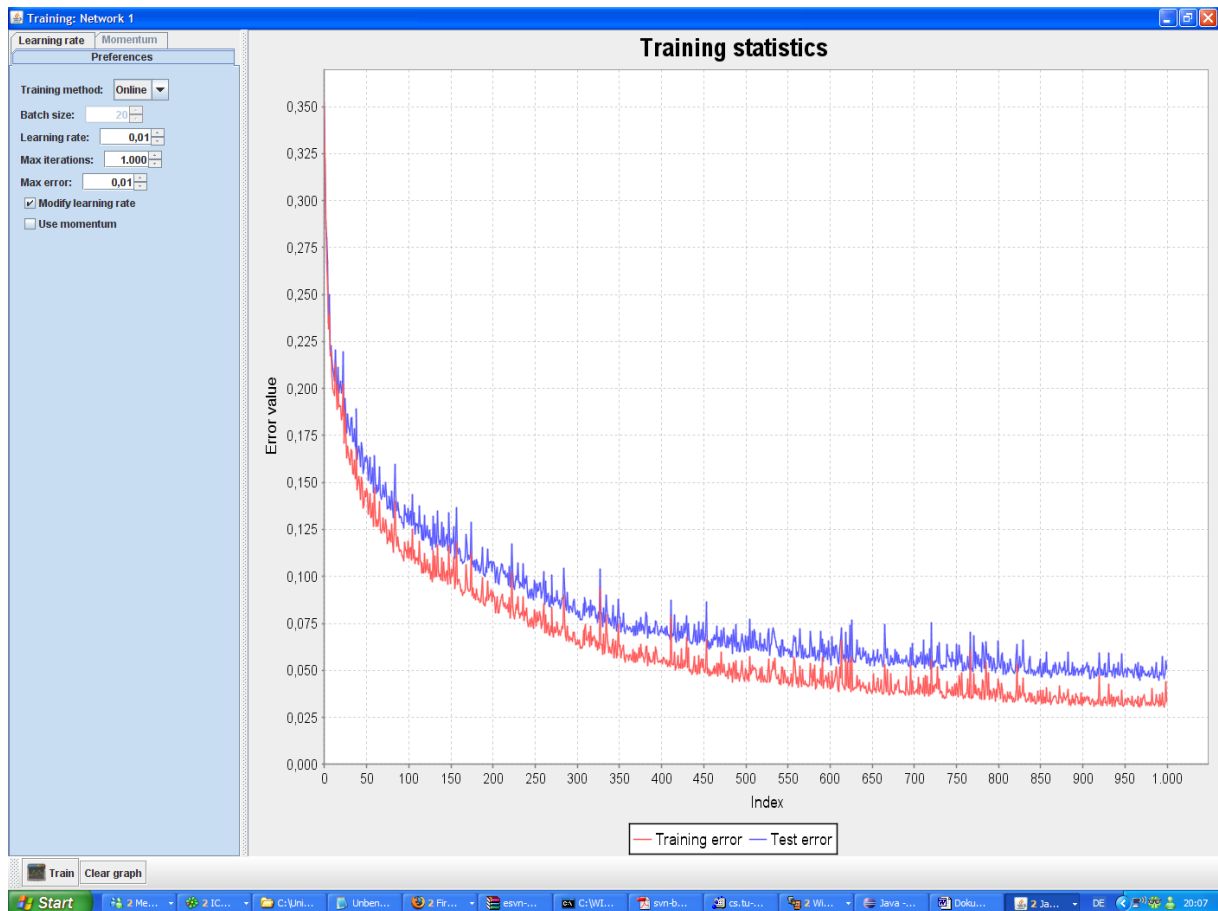


Über das Kontextmenü lässt sich bei geladenen Daten das Subsampling aufrufen, mit dem die Daten auf ein für das MLP bewältigbares Niveau herunterskaliert werden. Das linke Fenster stellt die Daten in der aktuellen Darstellung dar, das rechte bei Verwendung des gewählten Subsamplings.

An Optionen sind wählbar:

- eine Zoomstufe für die Anzeige der Dateien
- eine Zielauflösung
- ein Overlap (Überlappungsfunktion, mit der Datenpunkte aus umstehenden Datenpunkten berechnet werden)
- eine Skalierungsfunktion

## 3.5 Training



Im Trainingsbildschirm (erreichbar über das Baummenü auf Netzwerken) können sowohl die Parameter für das Training des Netzes eingestellt werden als auch das Training durchgeführt und überwacht werden.

Die Optionen hierfür sind:

- Trainingsmethode Online oder Batch
- bei Batchtraining die Batchgröße
- die anfängliche Lernrate
- die größtmögliche Anzahl an Iterationen des Algorithmus, nach der abgebrochen wird
- der Trainingsfehler, bei dessen Unterschreitung das Training abgebrochen wird
- die Verwendung von Lernratenveränderungen und Trägheit, die die jeweiligen Reiter freischaltet

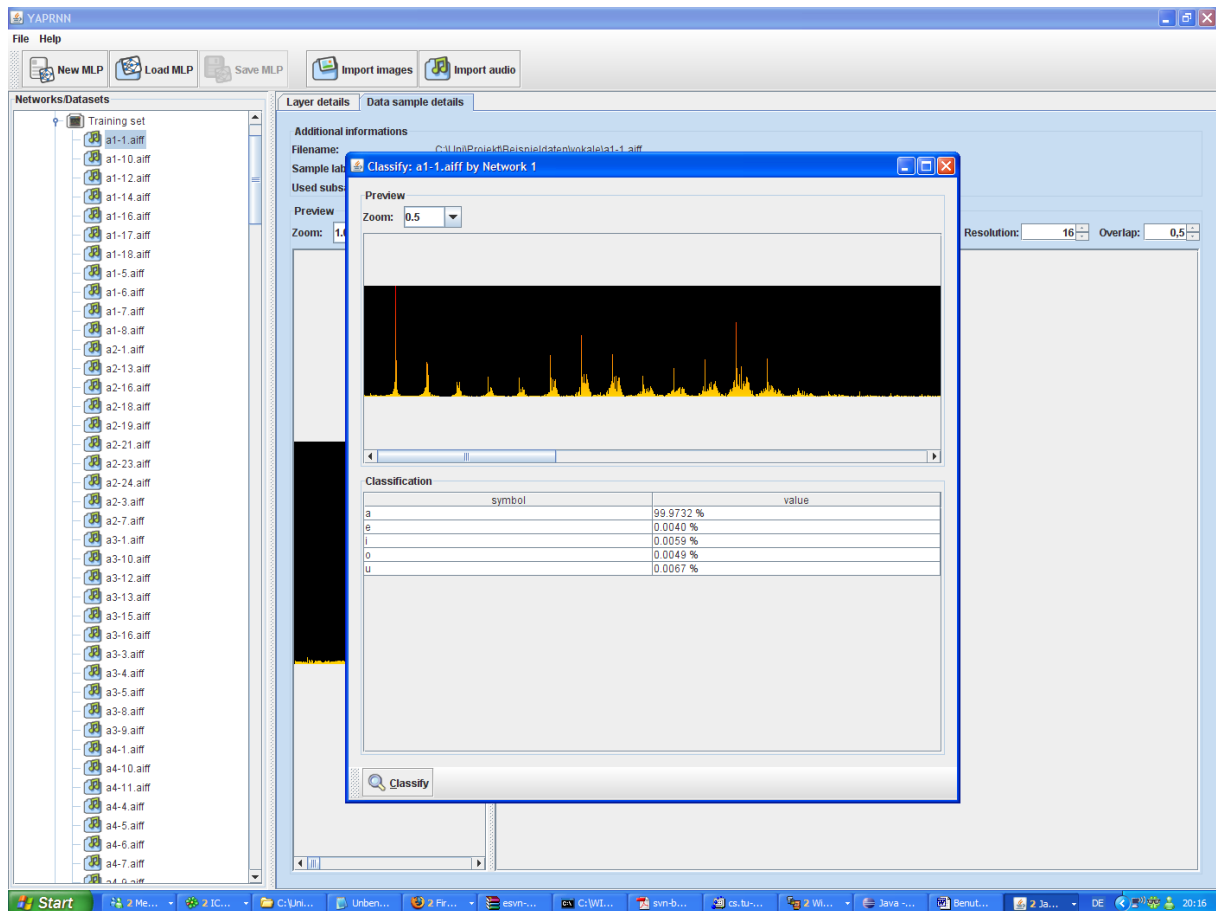
Die Reiter für Lernratenveränderung und Trägheit enthalten als Optionen:

- dynamische oder statische Lernratenveränderung

- dynamische Veränderung erhöht die Lernrate, wenn erfolgreich gelernt wird um den Multiplikator und senkt sie bei Misserfolgen um den Reduction Factor
- statische Lernrate multipliziert die Lernrate nach einer gewählten Anzahl an Iterationen jeweils mit dem gewählten Faktor
- ein Trägheitsfaktor gibt an, inwiefern die früheren Iterationen die jetzige Lernepoche beeinflussen soll

Das Training kann durch den Button links unten gestartet und angehalten werden. Die blaue Kurve zeigt den Testfehler (Fehlerwert für nicht zum Training benutzte Daten) an, die rote den Trainingsfehler (Fehler für zum Training verwendete Daten).

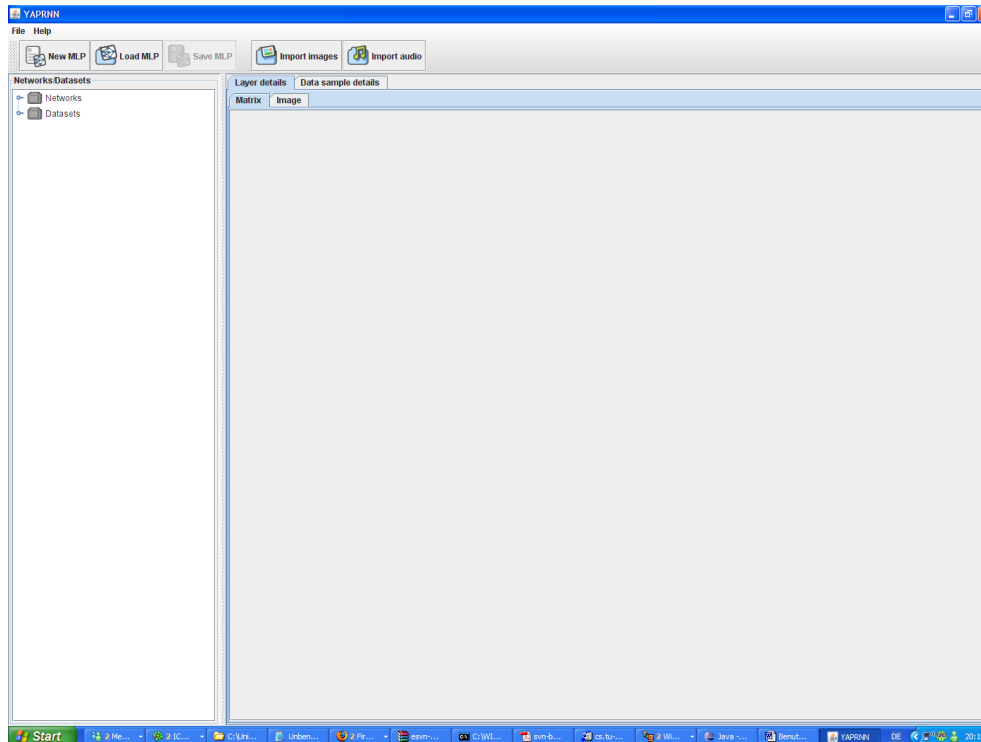
## 3.6 Klassifizierung



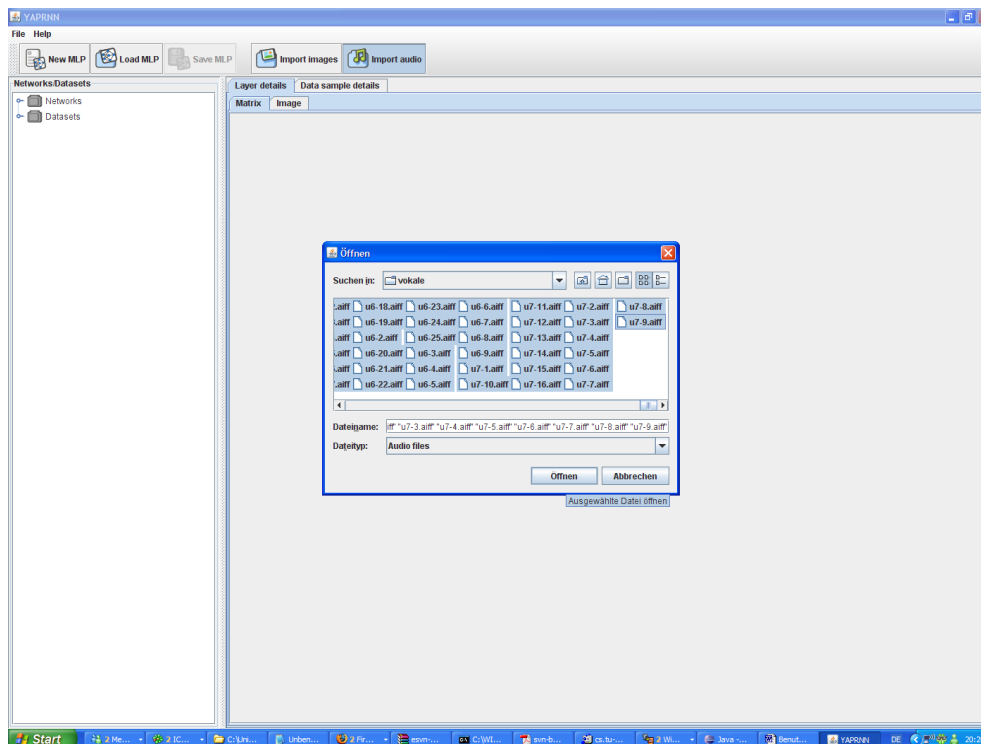
Der Klassifizierungsbildschirm schließlich ermöglicht es, geladene Daten durch das MLP zu propagieren und klassifizieren zu lassen. Mittels des Buttons Classify gibt das MLP eine Schätzung ab, wie das Eingabedatum zu klassifizieren ist.

## 4. Beispieldurchgang

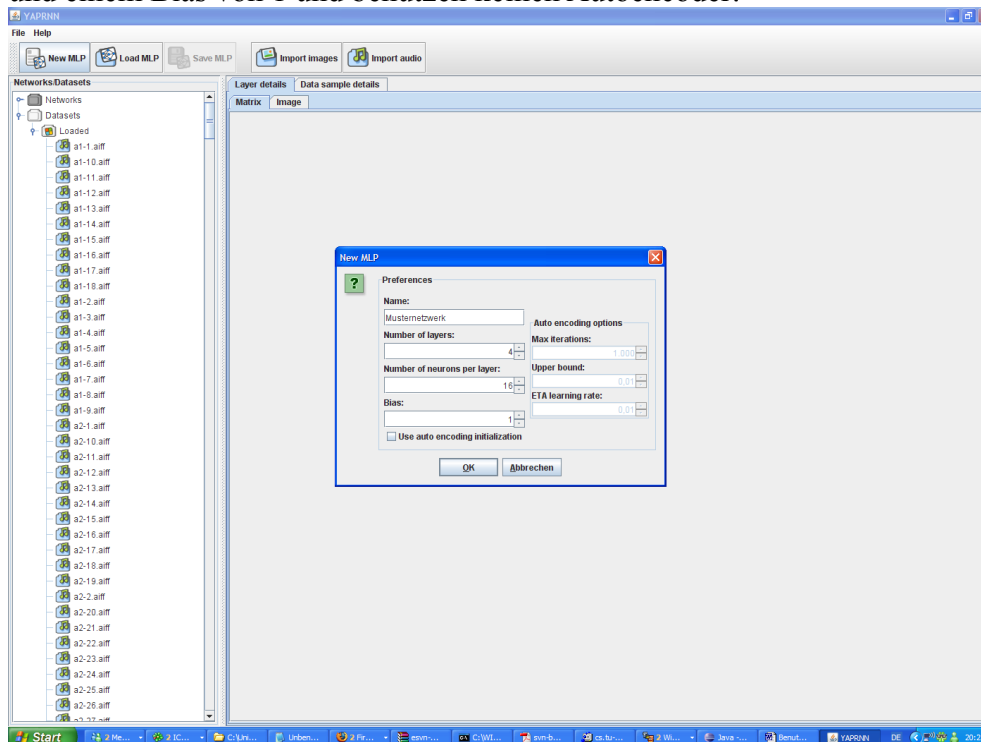
Im Folgenden soll anhand eines beispielhaften Durchgangs nochmals gezeigt werden, wie YAPRNN zu bedienen ist.



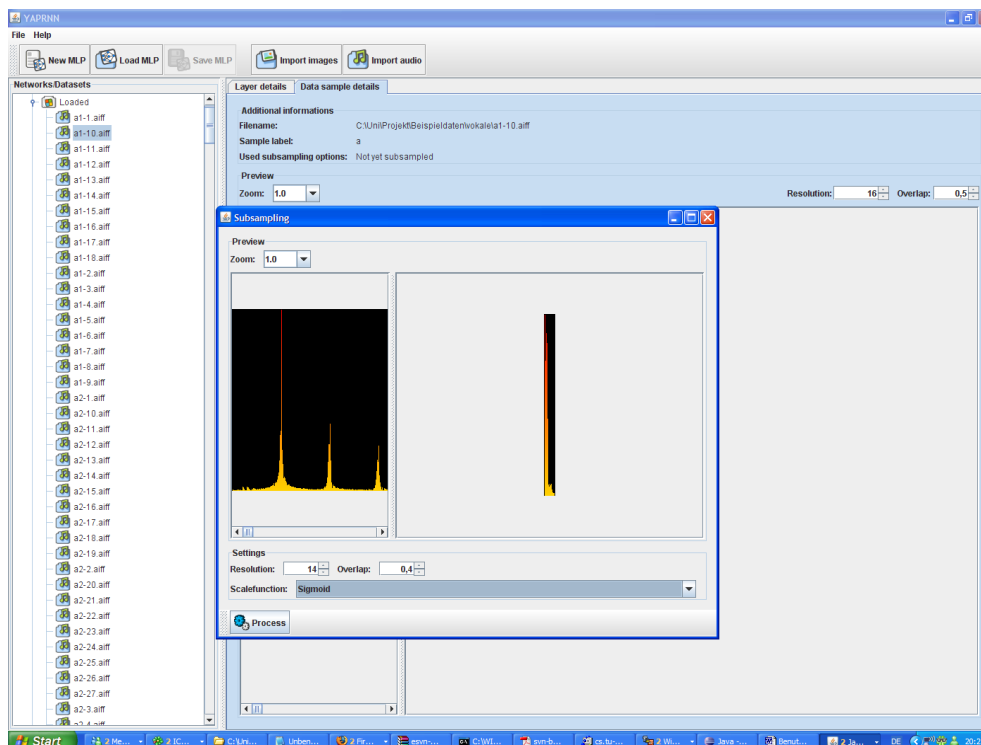
Zuerst importieren wir uns ein paar Audiodaten:



Dann erstellen wir unser MLP, geben ihm als Initialkonfiguration 4 Layer mit je 16 Neuronen und einem Bias von 1 und benutzen keinen Autoencoder:

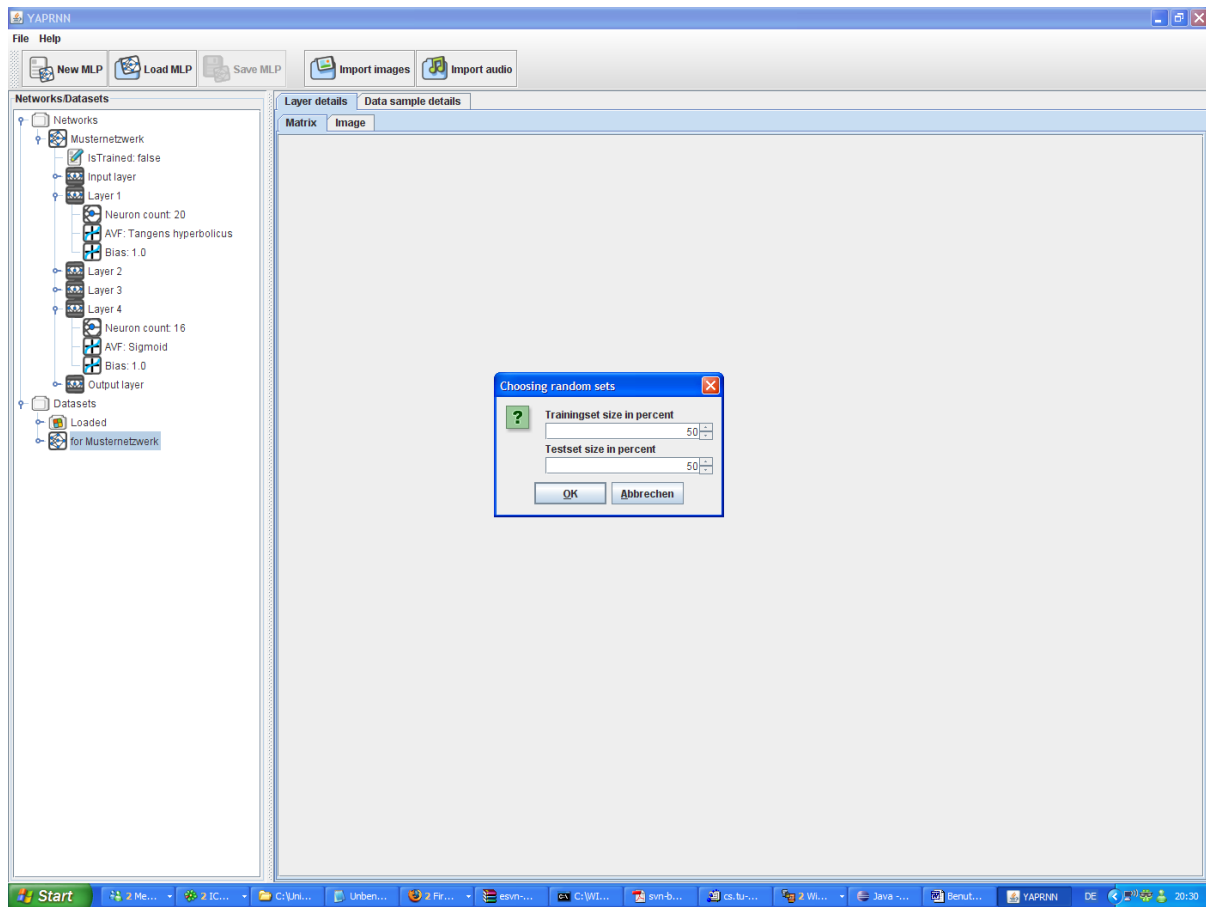


Als nächstes rufen wir durch einen Rechtsklick auf ein beliebiges Datum im Baumenü das Subsamplingfenster auf. Als Parameter wählen wir eine Auflösung von 14, ein Overlap von 0,4 und Sigmoid als Skalierungsfunktion und klicken auf Process.

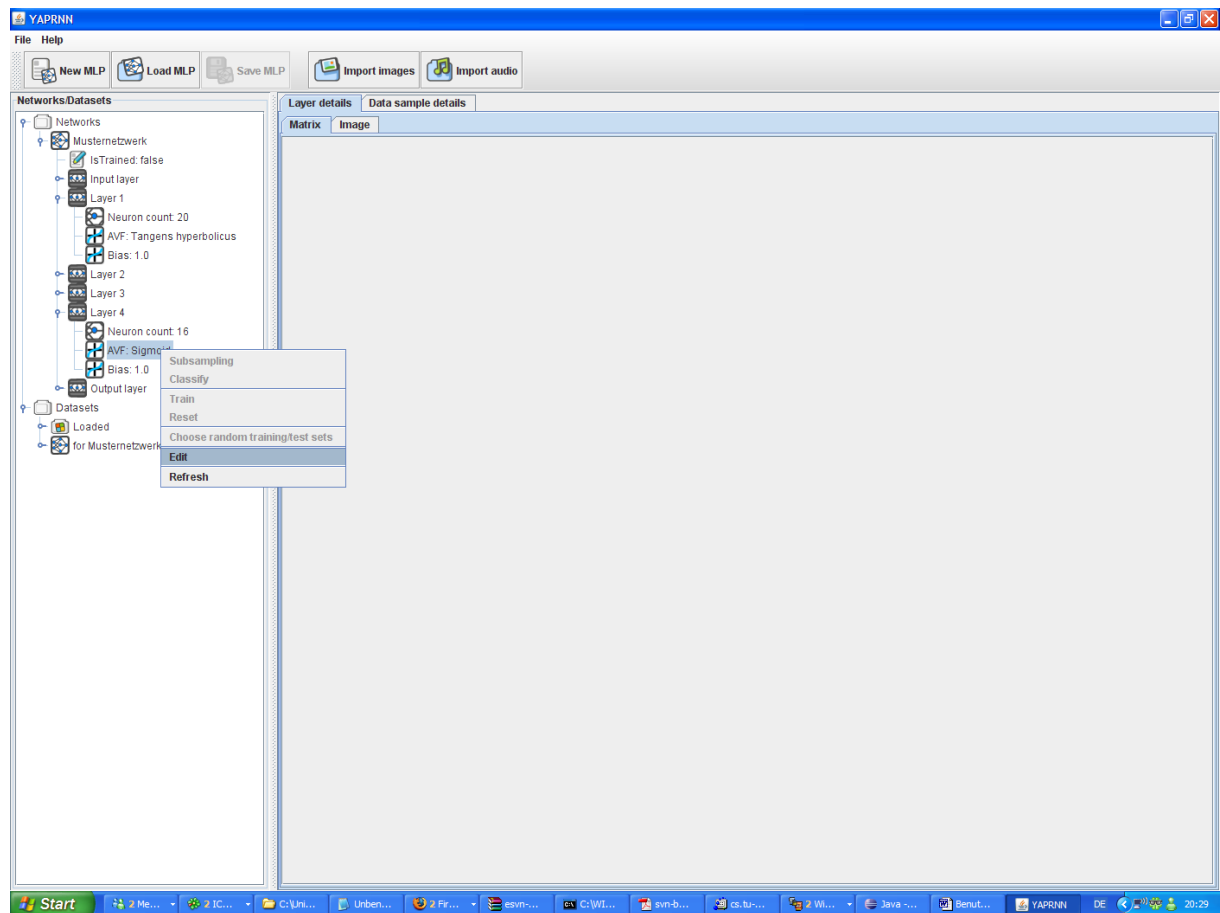




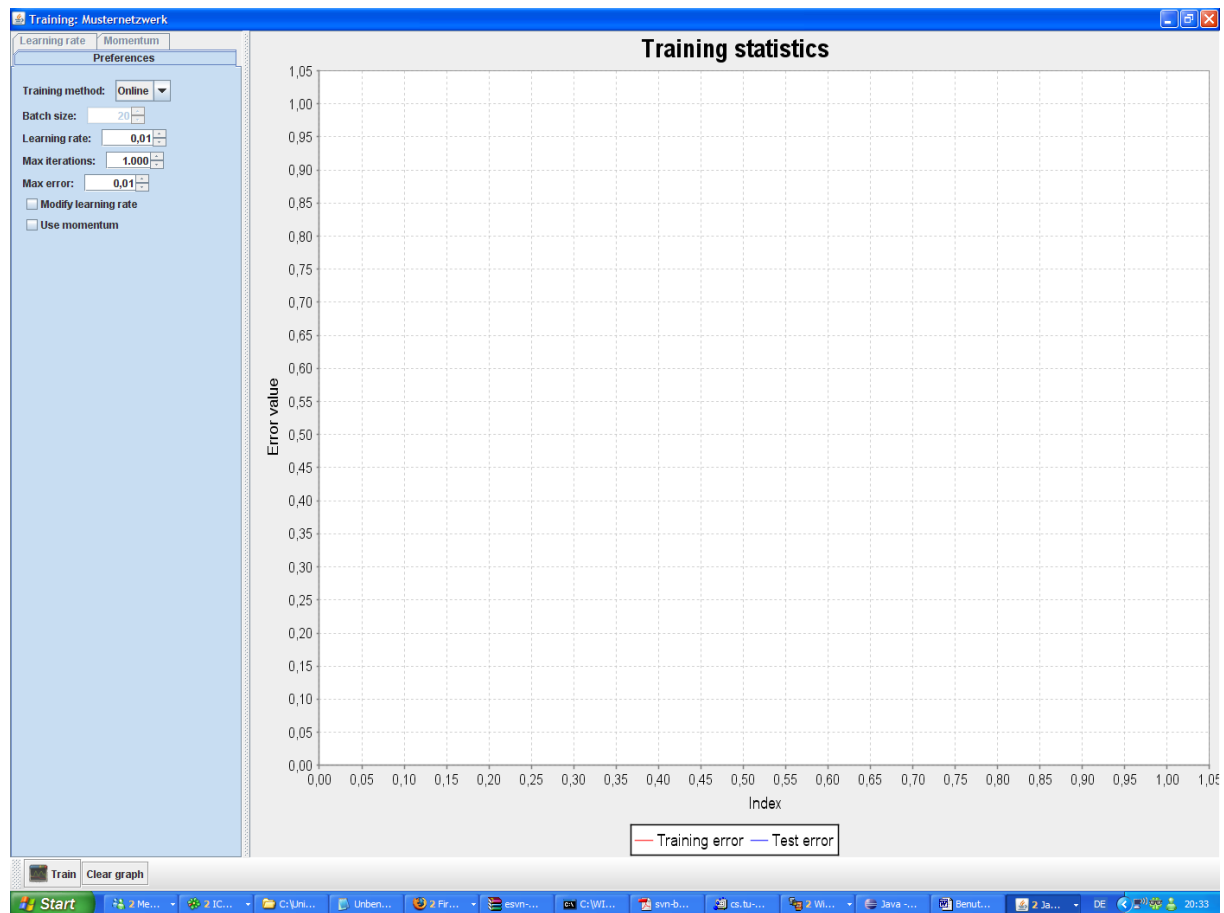
Dann teilen wir unserem Netzwerk Trainings- und Testdaten im Verhältnis 50:50 zu, indem wir im Kontextmenü des Datasets für unser „Musternetzwerk“ Choose Random Training/Test Set wählen.



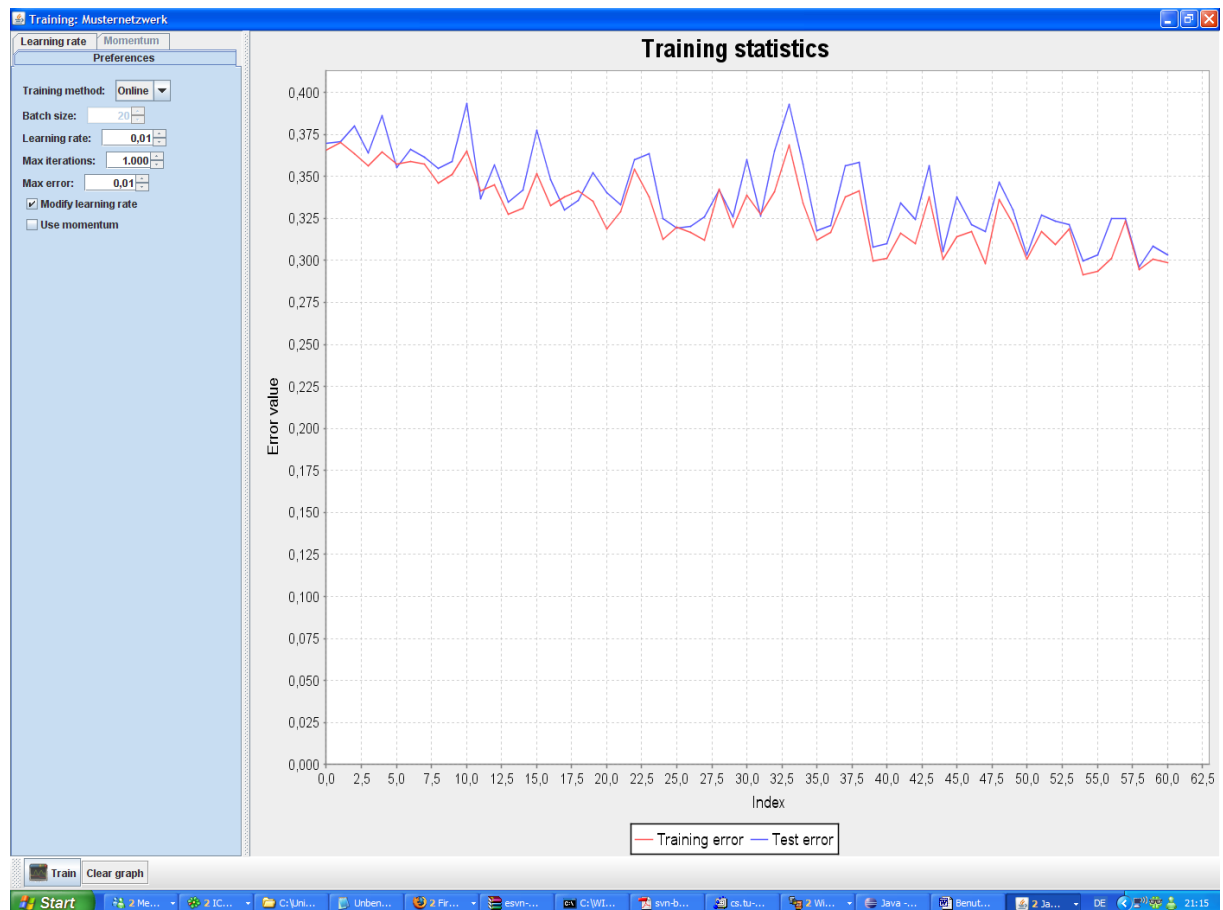
Schließlich ändern wir noch ein paar Werte unseres MLP. Wir stellen beim ersten Layer die Neuronenzahl auf 20 und verwenden beim vierten als Aktivierungsfunktion Sigmoid:



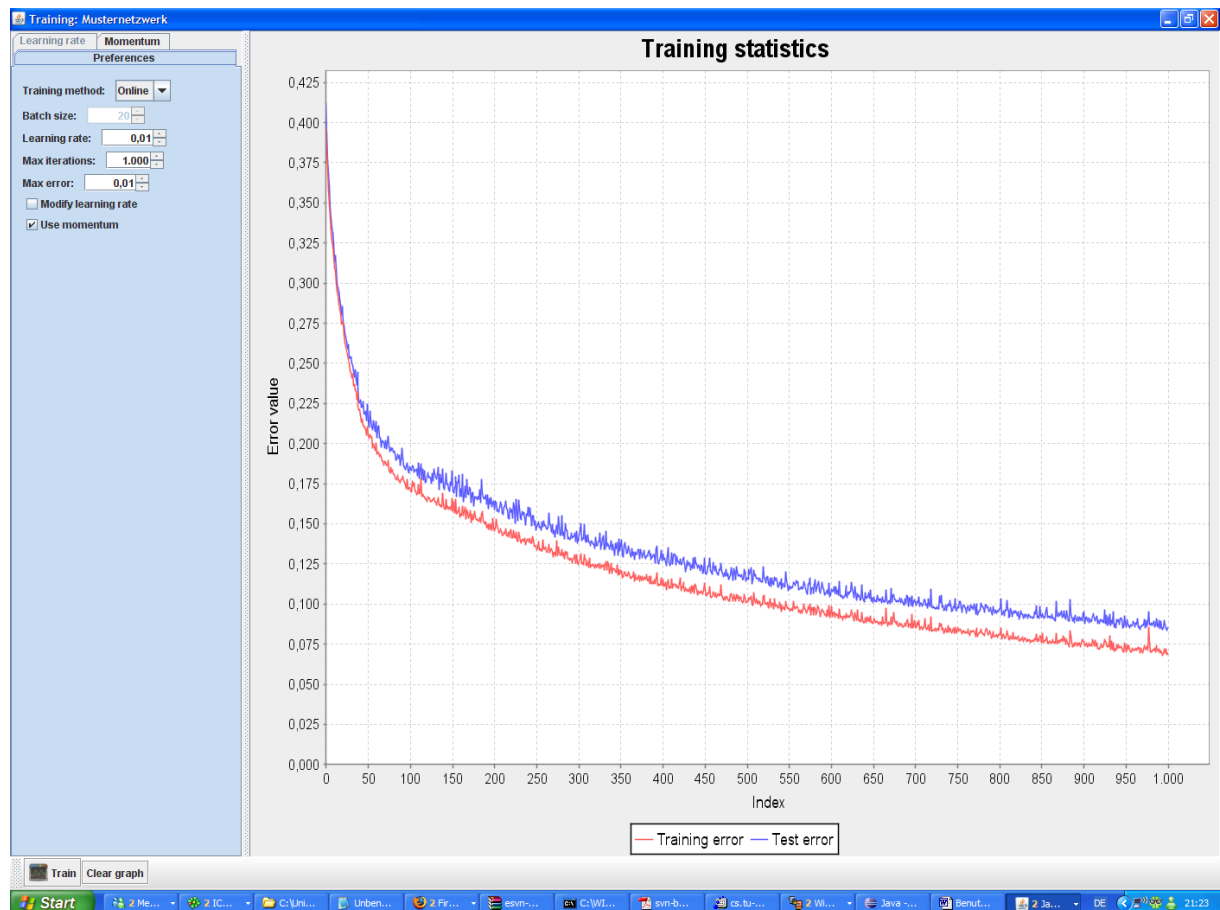
Nun ist alles zum Training bereit, das wir mittels Rechtsklick auf das Dataset des Musternetzwerks initialisieren:



Wir entscheiden uns für eine statisch sinkende Lernrate und ändern sonst nichts an den Parametern, bevor wir auf Train klicken.

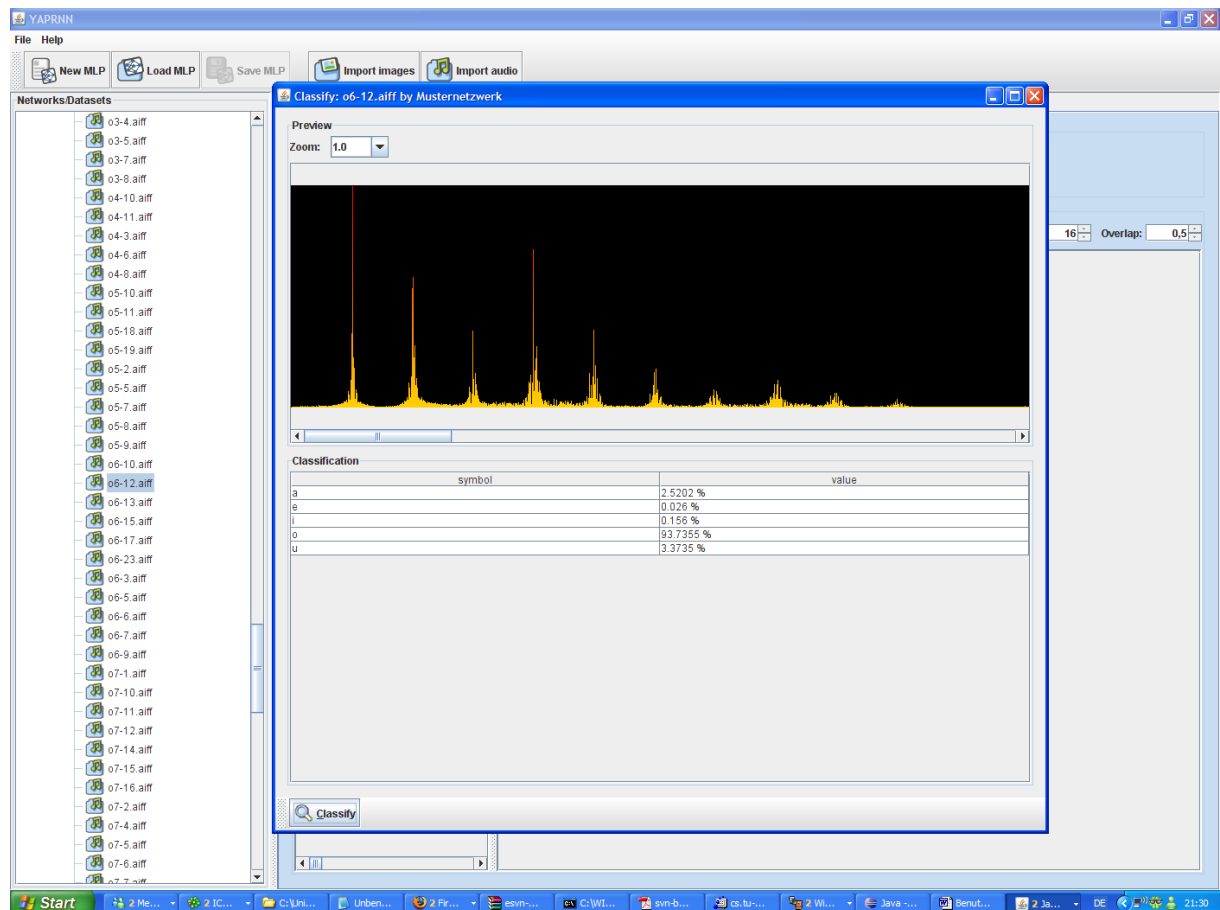


Bei 60 Iterationen brechen wir noch mal ab und lernen von nun an ohne modifizierte Lernrate, dafür aber mit Trägheit weiter.



Bei 1000 Iterationen schließlich endet das Training automatisch. Unseren erstrebten Trainingsfehler haben wir nicht erreicht, sind aber immerhin bis unter die 0.1 gekommen.

Kommen wir also zum Test unseres MLPs. Wir wählen aus dem Baummenü ein Testdatum aus, dass wir über das Kontextmenü klassifizieren lassen und drücken im Klassifizierungsfenster den Button Classify.



Wenn das Training gut verlaufen ist, sollte nun unser MLP mit hoher Wahrscheinlichkeit sagen können, um welchen Buchstaben es sich handelt. Sollte dem nicht so sein, gilt es nun, Parameterexploration zu betreiben und andere Lernmethoden, MLP-Konfigurationen oder Trainingsdaten zu verwenden...