

Accelerating Fuzzy Clustering

Christian Borgelt

Intelligent Data Analysis and Graphical Models Research Unit
European Center for Soft Computing
c/ Gonzalo Gutierrez Quiros s/n, 33600 Mieres, Spain

`christian.borgelt@softcomputing.es`
`http://www.borgelt.net/`

Overview

- **Brief Review of Neural Network Training**
 - Standard Error Backpropagation and Momentum Term
 - (Super) Self-adaptive Error Backpropagation
 - Resilient Error Backpropagation
 - Quickpropagation
- **Brief Review of Fuzzy Clustering**
 - Basic Idea and Objective Function
 - Alternating Optimization
 - Fuzzy C-Means and Gustafson–Kessel Algorithm
- **Transfer of NN Techniques to Fuzzy Clustering**
- **Comparing Clustering Results**
- **Experimental Results**
- **Summary**

Review: Neural Network Training

General approach: **gradient descent on the error function.**

- The error is a function of the network weights.
- Approach minimum by small weight changes opposite to the gradient.

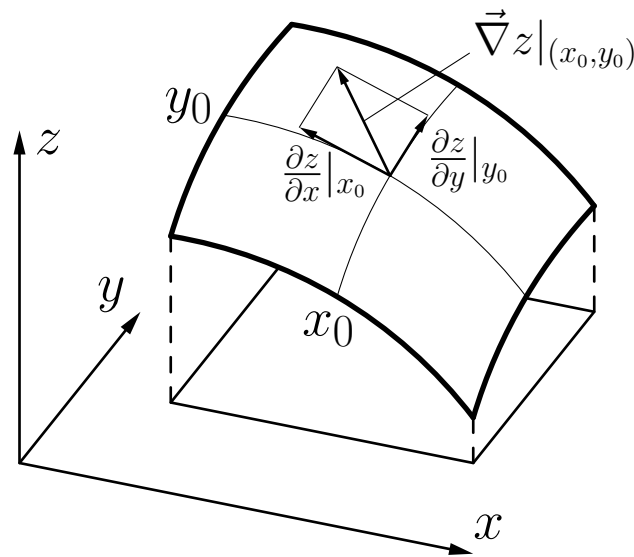


Illustration of the gradient of a real-valued function $z = f(x, y)$ at a point (x_0, y_0) .

It is $\vec{\nabla} z|_{(x_0, y_0)} = \left(\frac{\partial z}{\partial x}|_{x_0}, \frac{\partial z}{\partial y}|_{y_0} \right)$.

Neural Network Gradient Descent: Variants

Weight update rule:

$$w(t + 1) = w(t) + \Delta w(t)$$

Standard backpropagation:

$$\Delta w(t) = -\eta \nabla_w e(t)$$

Manhattan training:

$$\Delta w(t) = -\eta \operatorname{sgn}(\nabla_w e(t))$$

Momentum term:

$$\Delta w(t) = -\eta \nabla_w e(t) + \beta \Delta w(t - 1)$$

Neural Network Gradient Descent: Variants

(Super) Self-adaptive error backpropagation:

$$\eta_w(t) = \begin{cases} \gamma^- \cdot \eta_w(t-1), & \text{if } \nabla_w e(t) \cdot \nabla_w e(t-1) < 0, \\ \gamma^+ \cdot \eta_w(t-1), & \text{if } \nabla_w e(t) \cdot \nabla_w e(t-1) > 0 \\ & \wedge \nabla_w e(t-1) \cdot \nabla_w e(t-2) \geq 0, \\ \eta_w(t-1), & \text{otherwise.} \end{cases}$$

Resilient error backpropagation:

$$\Delta w(t) = \begin{cases} \gamma^- \cdot \Delta w(t-1), & \text{if } \nabla_w e(t) \cdot \nabla_w e(t-1) < 0, \\ \gamma^+ \cdot \Delta w(t-1), & \text{if } \nabla_w e(t) \cdot \nabla_w e(t-1) > 0 \\ & \wedge \nabla_w e(t-1) \cdot \nabla_w e(t-2) \geq 0, \\ \Delta w(t-1), & \text{otherwise.} \end{cases}$$

Typical values: $\gamma^- \in [0.5, 0.7]$ and $\gamma^+ \in [1.05, 1.2]$.

Neural Network Gradient Descent: Variants

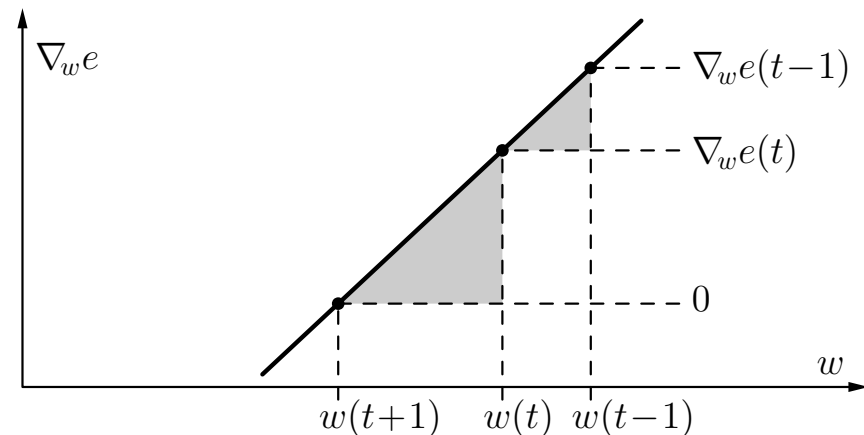
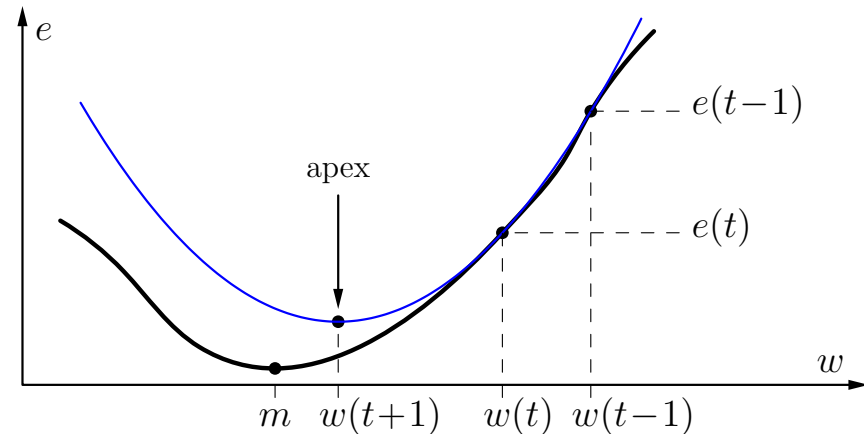
Quickpropagation

The error function is locally approximated by a parabola.

The weight update “jumps” to the apex of the parabola.

The weight update rule can be derived from the triangles:

$$\Delta w(t) = \frac{\nabla_w e(t)}{\nabla_w e(t-1) - \nabla_w e(t)} \cdot \Delta w(t-1).$$



Review: Standard Fuzzy Clustering

- Allow degrees of membership of a datum to different clusters.
(Classical c -means clustering assigns data crisply.)

- **Objective Function:** (to be minimized)

$$J(\mathbf{X}, \mathbf{C}, \mathbf{U}) = \sum_{i=1}^c \sum_{j=1}^n h(u_{ij}) d^2(\mathbf{c}_i, \vec{x}_j)$$

- $\mathbf{U} = [u_{ij}]$ is the $c \times n$ fuzzy partition matrix,
 $u_{ij} \in [0, 1]$ is the membership degree of the data point \vec{x}_j to the i -th cluster.
- $\mathbf{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_c\}$ is the set of cluster prototypes.
- Usually $h(u_{ij}) = u_{ij}^\alpha$ is chosen, where α is the so-called “fuzzifier”
(the higher α , the softer the cluster boundaries).

- Constraints:

$$\forall i \in \{1, \dots, c\} : \sum_{j=1}^n u_{ij} > 0 \quad \text{and} \quad \forall j \in \{1, \dots, n\} : \sum_{i=1}^c u_{ij} = 1$$

Review: Alternating Optimization

- **Problem:** The objective function J cannot be minimized directly.
- Therefore: **Alternating Optimization**
 - Optimize membership degrees for fixed cluster parameters.
 - Optimize cluster parameters for fixed membership degrees.
(Update formulae are derived by differentiating the objective function J)
 - Iterate until convergence (checked, e.g., by change of cluster center).
- **Update Rules:** (for Euclidean distance and only centers, i.e. $\mathbf{c}_i = (\mu_i)$)

$$\forall i; 1 \leq i \leq c : \forall j; 1 \leq j \leq n : \quad u_{ij} = \frac{d_{ij}^{\frac{2}{1-\alpha}}}{\sum_{k=1}^c d_{kj}^{\frac{2}{1-\alpha}}}$$
$$\forall i; 1 \leq i \leq c : \quad \vec{\mu}_i = \frac{\sum_{j=1}^n u_{ij}^\alpha \vec{x}_j}{\sum_{j=1}^n u_{ij}^\alpha}$$

Review: Gustafson–Kessel Fuzzy Clustering

- Introduce a covariance matrix to describe the cluster shape.
- **Objective Function:** (to be minimized)

$$J(\mathbf{X}, \mathbf{C}, \mathbf{U}) = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^{\alpha} (\vec{x}_j - \vec{\mu}_i)^{\top} \Sigma^{-1} (\vec{x}_j - \vec{\mu}_i)$$

- **Update Rule** for the covariance matrix:

$$\Sigma = \mathbf{S} |\mathbf{S}|^{-\frac{1}{m}} \quad \text{where} \quad \mathbf{S} = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^{\alpha} (\vec{x}_j - \vec{\mu}_i) (\vec{x}_j - \vec{\mu}_i)^{\top}$$

- **Axes-parallel version** of Gustafson–Kessel Fuzzy Clustering:
Restrict the covariance matrix to a diagonal matrix.

$$\Sigma = \text{diag}(\sigma_1^2, \dots, \sigma_m^2).$$

Transfer to Fuzzy Clustering

- Compute one update step of fuzzy clustering, i.e., compute new membership degrees and new centers.
- Compute change of centers (cluster parameters), i.e., difference of coordinates to preceding step.
- **Consider this difference as a gradient and apply the improvements from neural network training.**
- Note: Standard backpropagation also yields a modification: Introduction of a learning rate $\eta \geq 1$. (This approach is generally known as over-relaxation.)
- **Expectation:** This transfer of neural network methods leads to a **speed-up of fuzzy clustering**.

Transfer to Fuzzy Clustering: Variants

General parameter update rule:

$$\theta(t + 1) = \theta(t) + \Delta\theta(t)$$

Update step expansion:

$$\Delta\theta(t) = \eta\delta\theta(t) \quad \text{with } \eta \in [1.05, 2],$$

where $\delta\theta(t)$ is the change of the cluster parameter θ as it is computed with the standard update rule in step t .

Momentum term:

$$\Delta\theta(t) = \delta\theta(t) + \beta\Delta\theta(t - 1) \quad \text{with } \beta \in [0, 1).$$

$\Delta\theta(t)$ is clamped to $[\delta\theta(t), \eta_{\max}\delta\theta(t)]$ with $\eta_{\max} = 1.8$ for robustness.

Transfer to Fuzzy Clustering: Variants

Self-adaptive step expansion:

$$\eta_{\theta}(t) = \begin{cases} \gamma^{-} \cdot \eta_{\theta}(t-1), & \text{if } \delta\theta(t) \cdot \delta\theta(t-1) < 0, \\ \gamma^{+} \cdot \eta_{\theta}(t-1), & \text{if } \delta\theta(t) \cdot \delta\theta(t-1) > 0, \\ \eta_{\theta}(t-1), & \text{otherwise.} \end{cases}$$

Resilient update:

$$\Delta\theta(t) = \begin{cases} \gamma^{-} \cdot \Delta\theta(t-1), & \text{if } \delta\theta(t) \cdot \delta\theta(t-1) < 0, \\ \gamma^{+} \cdot \Delta\theta(t-1), & \text{if } \delta\theta(t) \cdot \delta\theta(t-1) > 0, \\ \Delta\theta(t-1), & \text{otherwise.} \end{cases}$$

Quickpropagation analog:

$$\Delta\theta(t) = \frac{\delta\theta(t)}{\delta\theta(t-1) - \delta\theta(t)} \cdot \Delta\theta(t-1).$$

In my experiments I used $\gamma^{-} = 0.7$ and $\gamma^{+} = 1.2$ and clamping.

Updating Covariance Matrices

- Center coordinates can be updated independently and arbitrarily.
- (Co)variances, however, have a bounded range of values and depend on each other (e.g. $s_{xy}^2 \leq s_x^2 s_y^2$).
- (Co)variances are updated before normalization to determinant 1.
- **Variances:** (axes-parallel Gustafson–Kessel clustering)
 - Are treated independently of each other.
 - Check for a positive value (a variance must be > 0), otherwise do standard update step.
- **Covariances:** (general Gustafson–Kessel clustering)
 - Check for positive definite matrix with Cholesky decomposition.
 - If the updated matrix is not positive definite, do a standard update setp for the matrix as a whole.

Convergence Evaluation

- General idea: Use **relative cluster evaluation measures**.
- Simplest approach:

$$Q_{\text{diff}}(\mathbf{U}^{(1)}, \mathbf{U}^{(2)}) = \min_{\pi \in \Pi(c)} \frac{1}{cn} \sum_{i=1}^c \sum_{j=1}^n \left(u_{ij}^{(1)} - u_{\pi(i)j}^{(2)} \right)^2.$$

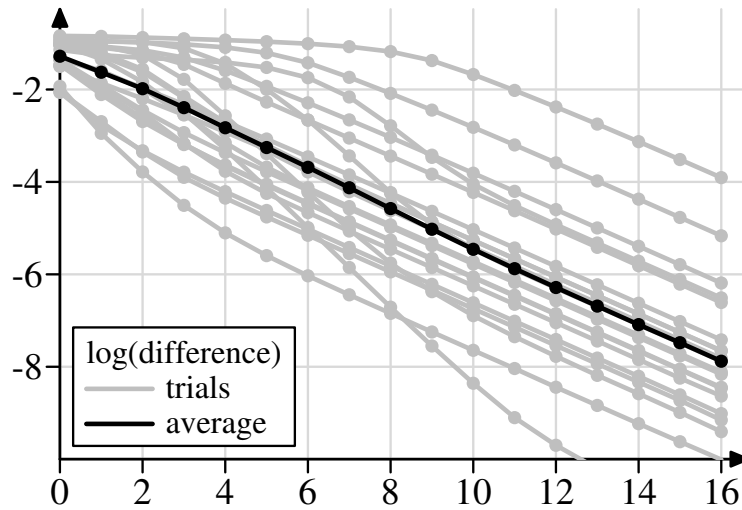
$\mathbf{U}^{(k)} = (u_{ij}^{(k)})_{1 \leq i \leq c, 1 \leq j \leq n}$ for $k = 1, 2$ are the two partition matrices to compare, n is the number of data points, c the number of clusters, and $\Pi(c)$ is the set of all permutations of the numbers 1 to c .

- Other possibilities:
 - (cross-classification) accuracy
 - Rand statistic / Rand index
 - Fowlkes–Mallows index
 - F_1 -measure
 - Jaccard coefficient / Jaccard index
 - Hubert index / Hubert-Arabie index

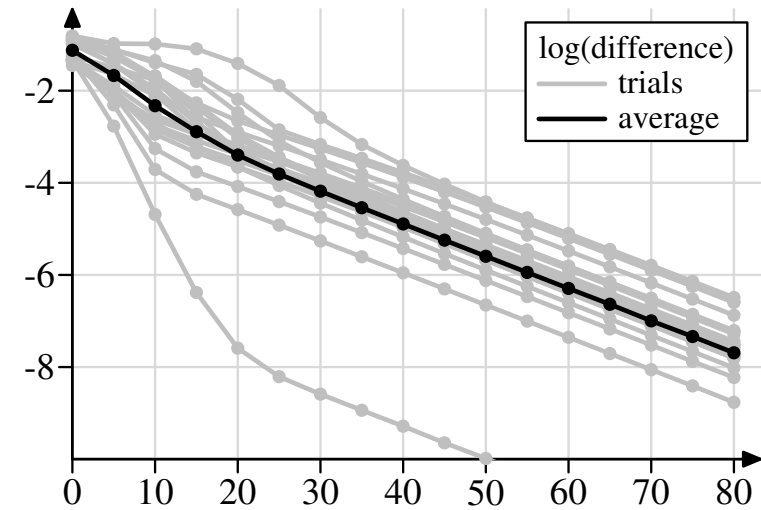
The last four measures are based on evaluating *coincidence matrices*.

Experimental Results: Clustering Trials

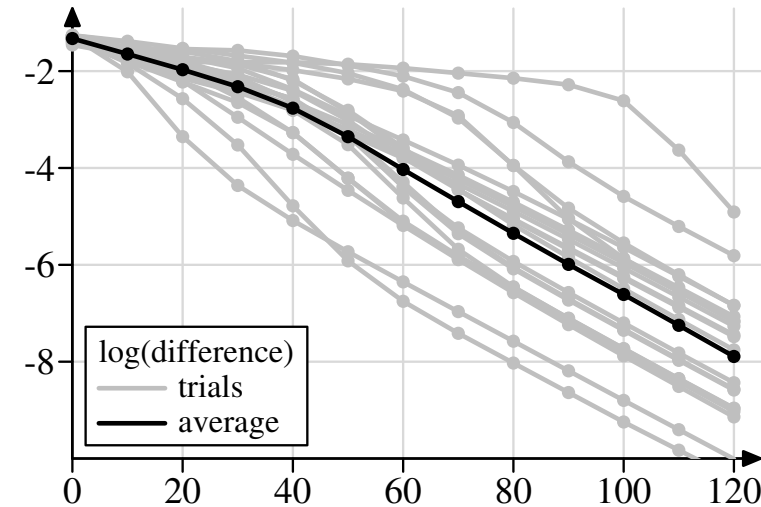
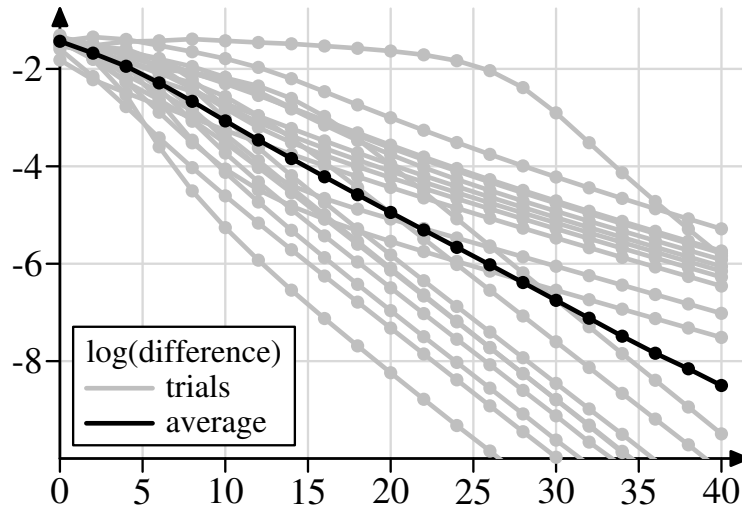
fuzzy *c*-means



general Gustafson–Kessel

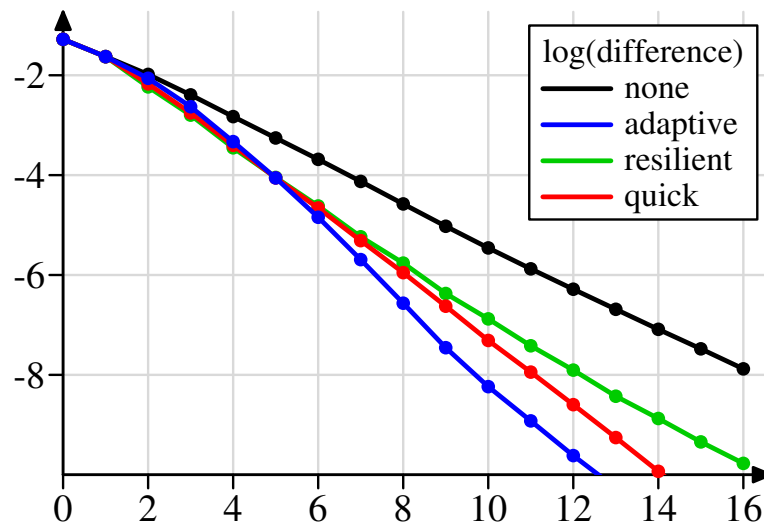
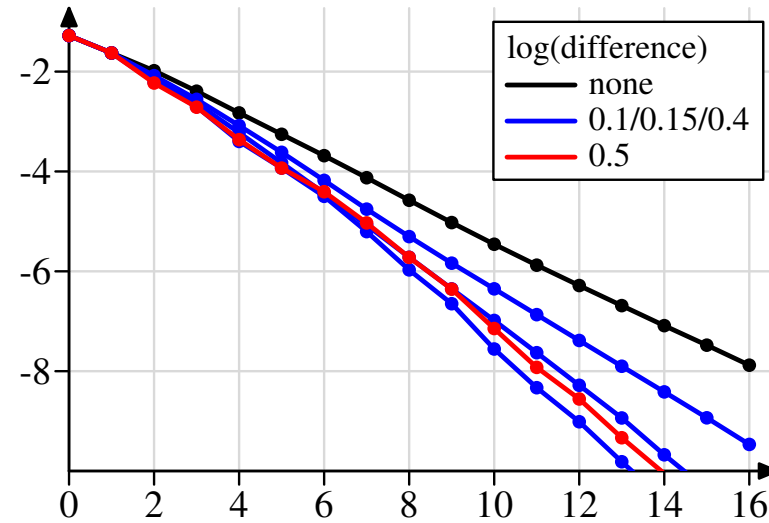
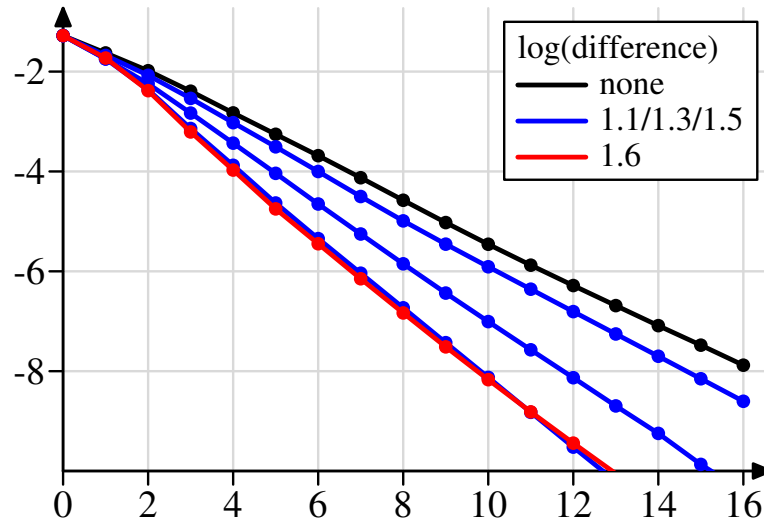


iris
3 clusters



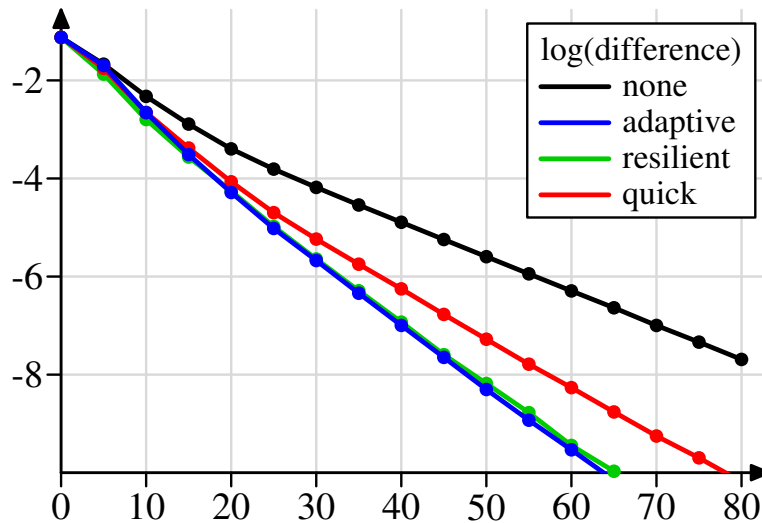
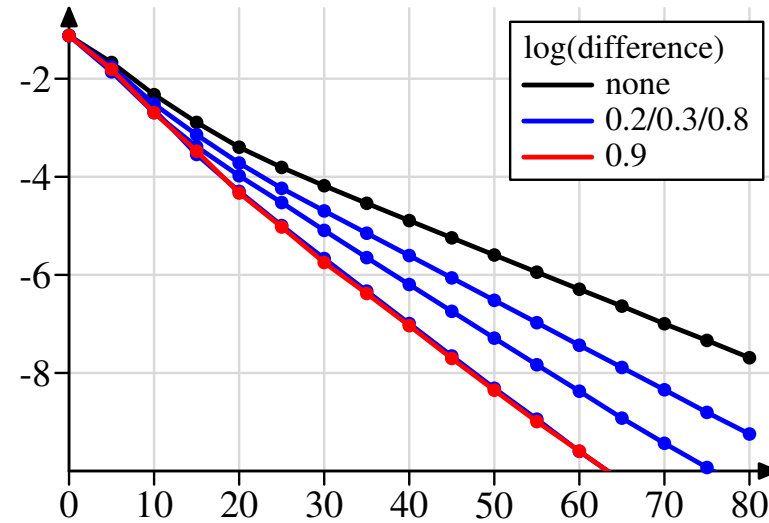
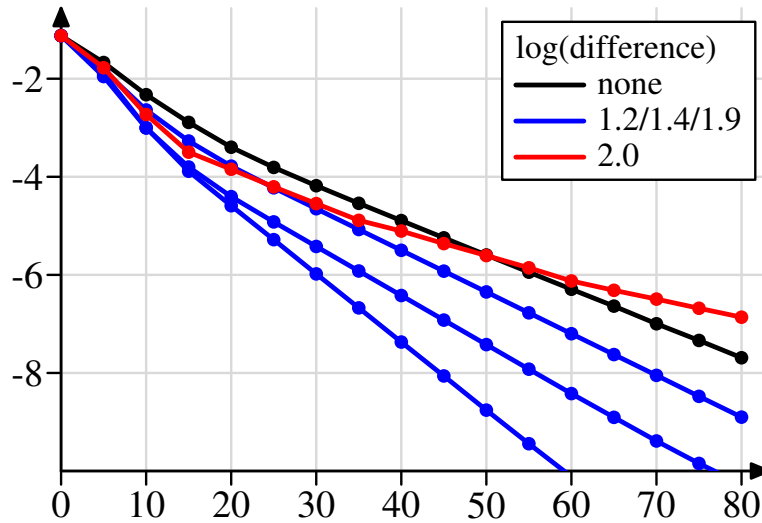
wsel
6 clusters

Experimental Results: iris, 3 clusters, FCM



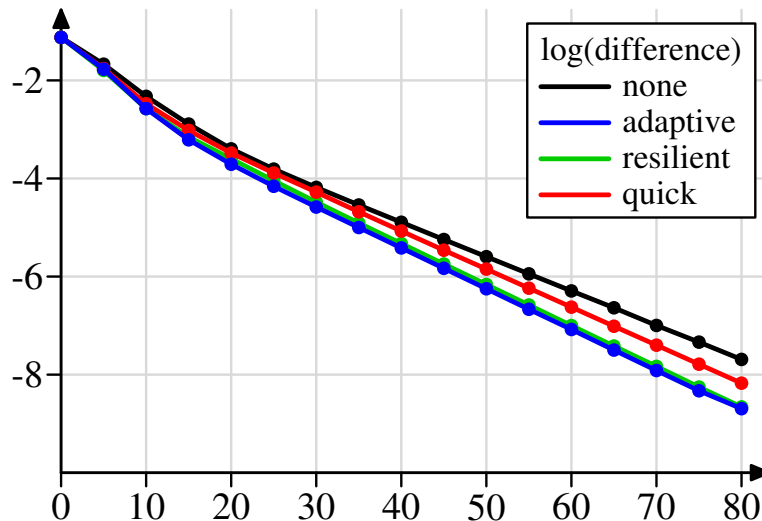
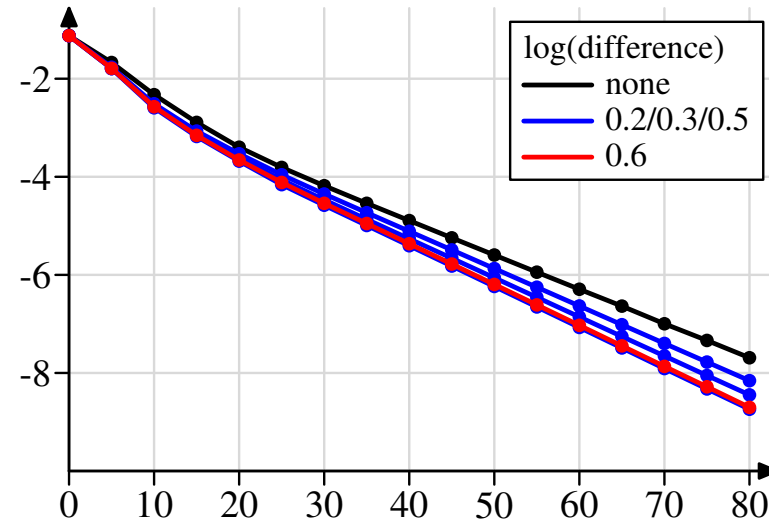
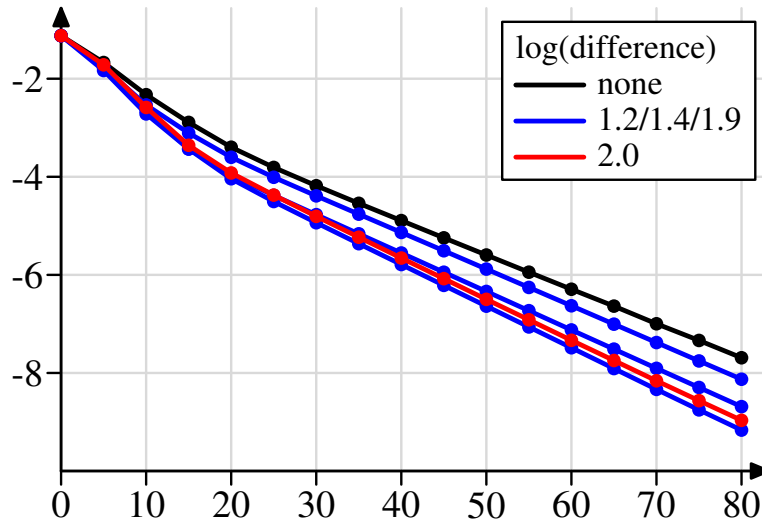
Clustering the iris data with the fuzzy c -means algorithm and update modifications;
 top left: step expansion,
 top right: momentum term,
 bottom left: other methods.

Experimental Results: iris, 3 clusters, GK



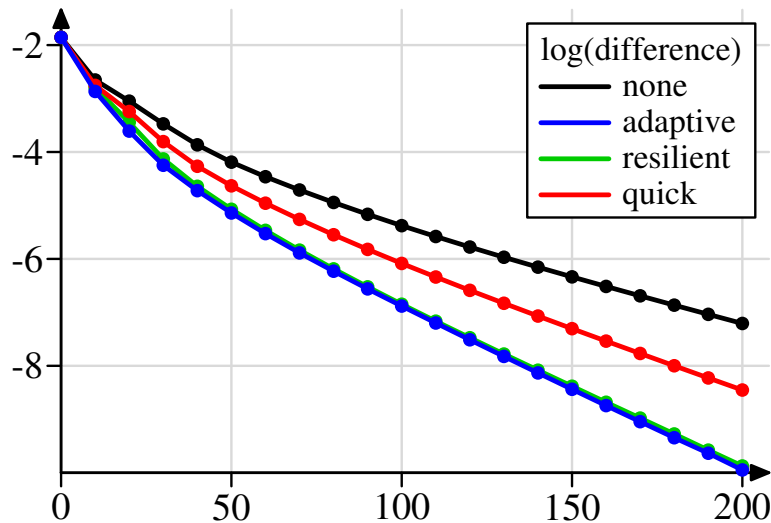
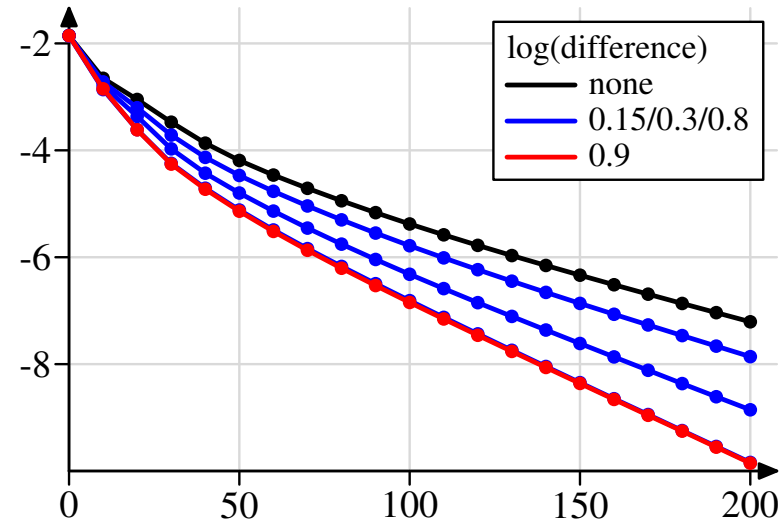
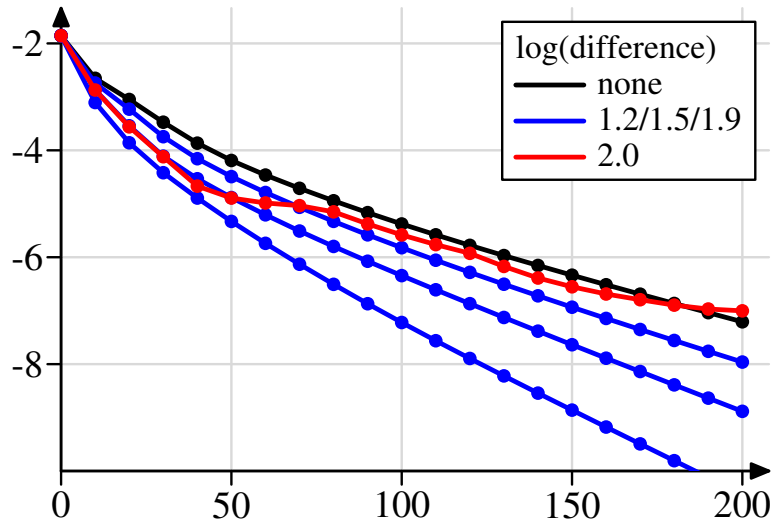
Clustering the iris data with the Gustafson–Kessel algorithm and update modifications for all parameters;
top left: step expansion,
top right: momentum term,
bottom left: other methods.

Experimental Results: iris, 3 clusters, GK



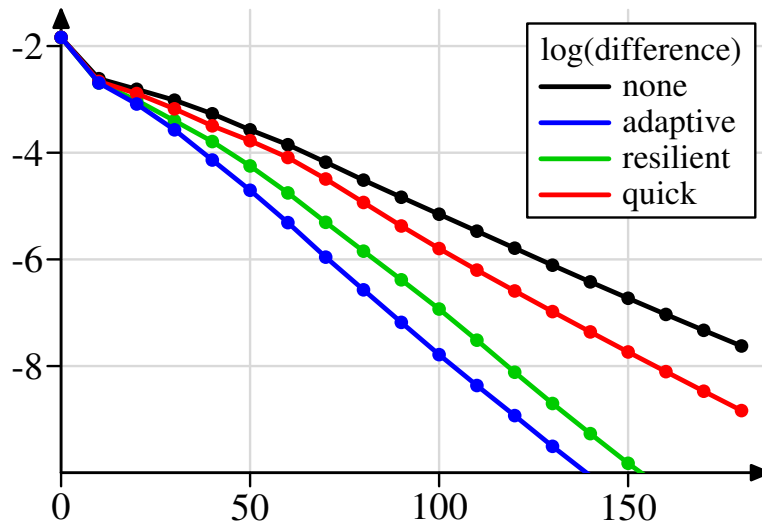
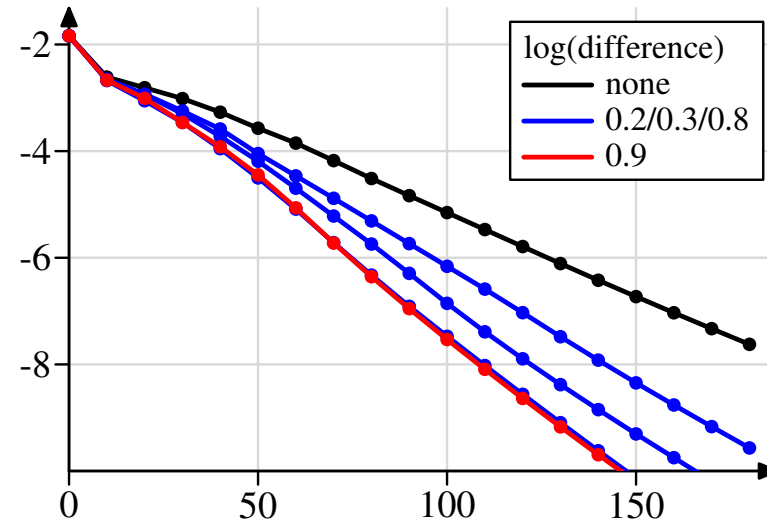
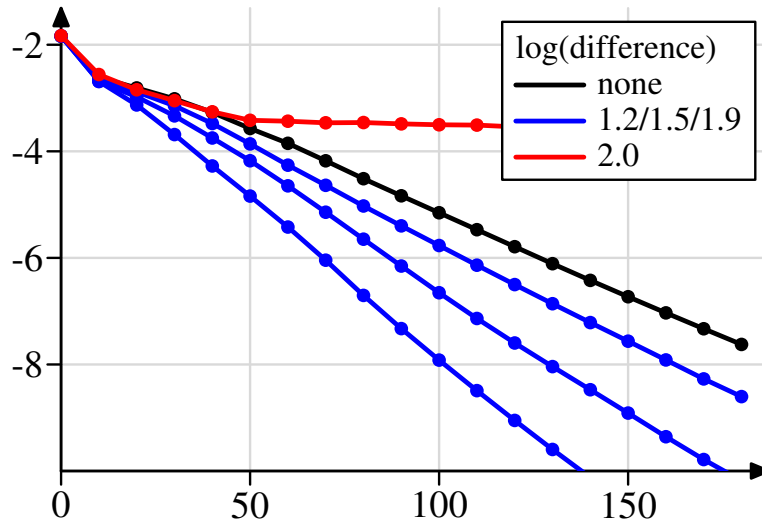
Clustering the iris data with the Gustafson–Kessel algorithm and update modifications applied only for cluster centers; top left: step expansion, top right: momentum term, bottom left: other methods.

Experimental Results: wine, 6 clusters, ap. GK



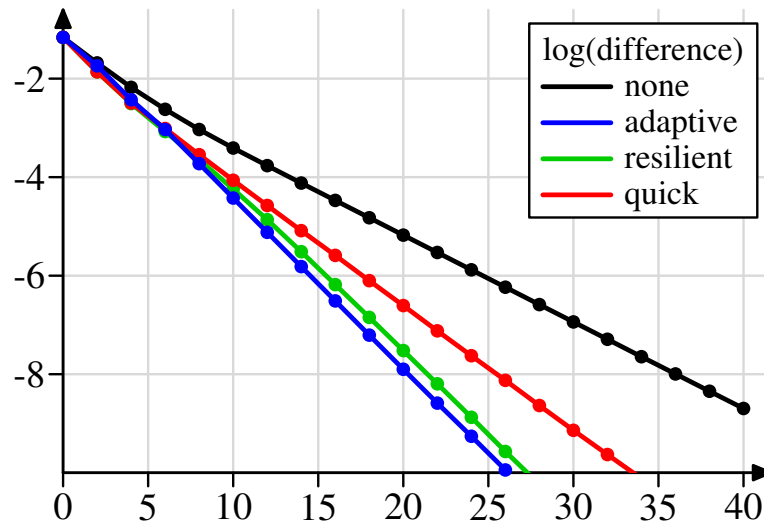
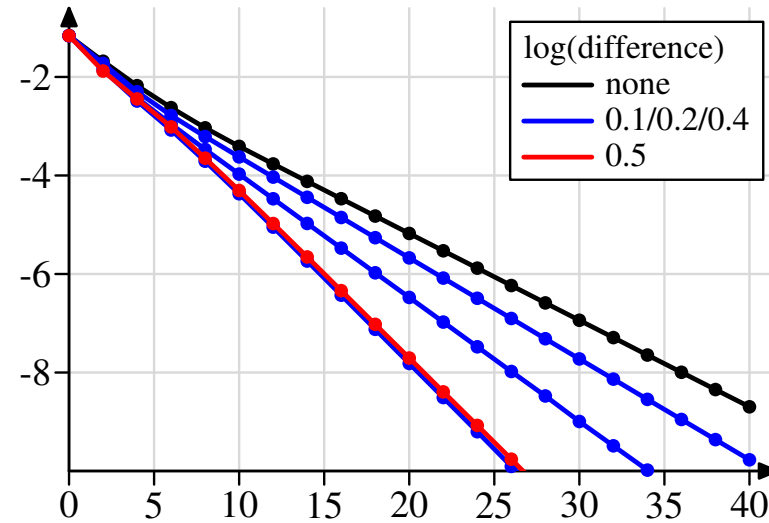
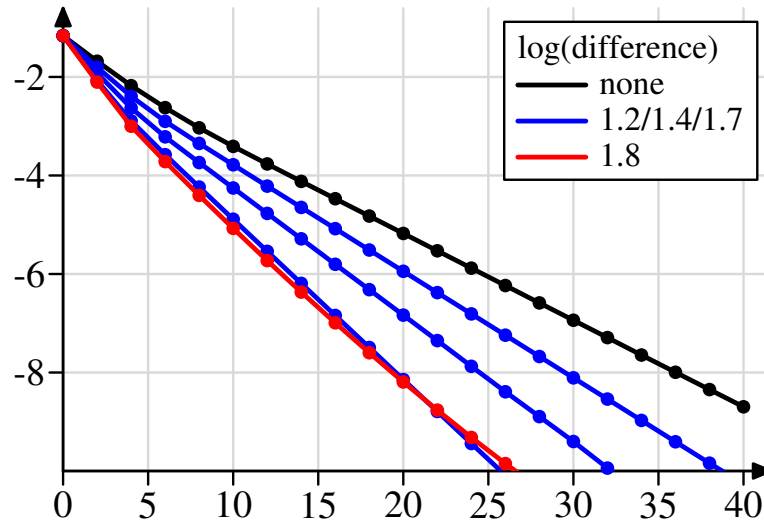
Clustering the wine data with the axes-parallel Gustafson–Kessel algorithm and 6 clusters; top left: step expansion, top right: momentum term, bottom left: other methods.

Experimental Results: wine, 6 clusters, GK



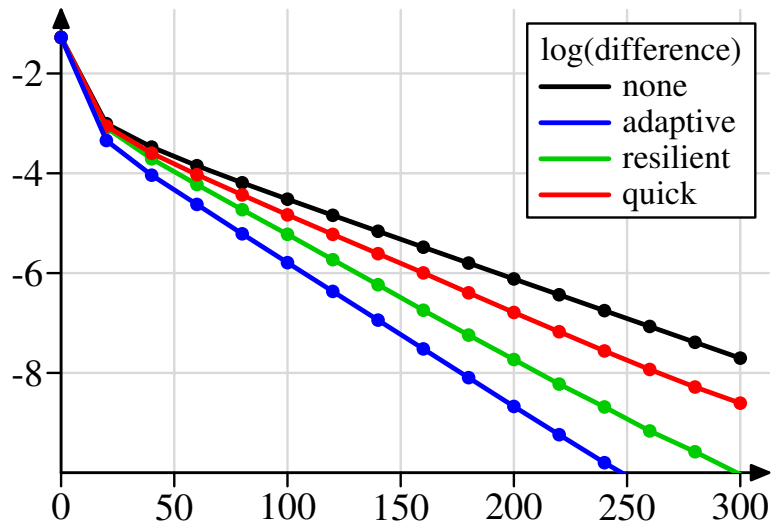
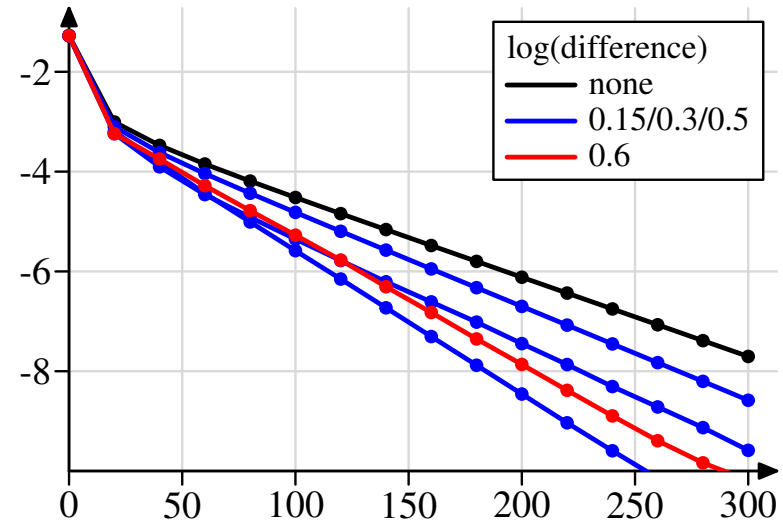
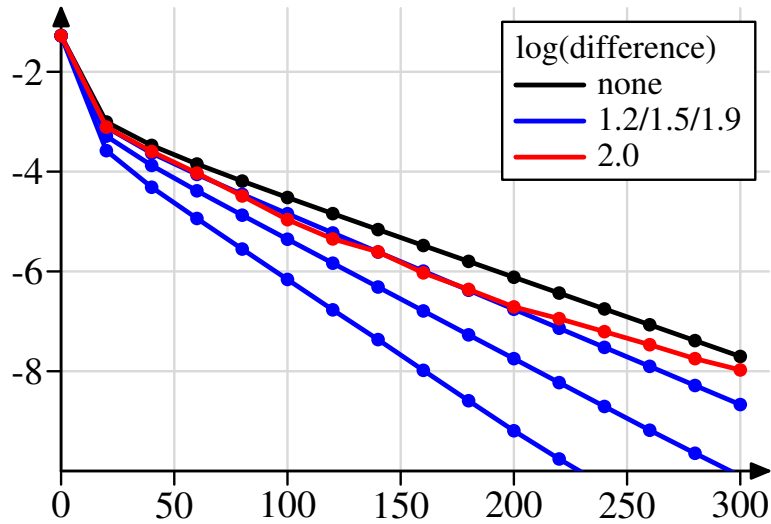
Clustering the wine data with the general Gustafson–Kessel algorithm and 6 clusters; top left: step expansion, top right: momentum term, bottom left: other methods.

Experimental Results: abalone, 3 clusters, FCM



Clustering the abalone data with the fuzzy c -means algorithm and 3 clusters;
top left: step expansion,
top right: momentum term,
bottom left: other methods.

Experimental Results: abalone, 3 clusters, GK



Clustering the abalone data with the general Gustafson–Kessel algorithm and 3 clusters; top left: step expansion, top right: momentum term, bottom left: other methods.

Summary

- Fuzzy clustering as well as neural network training are iterative processes.
- Executing alternating optimization can be seen as providing a gradient step.
- Thus all variants of neural network gradient descent become applicable.
- Some of these variants lead to a considerable speed-up.
- A transfer to estimating a mixture of Gaussian is also possible.

An implementation of these techniques can be retrieved free of charge at

<http://www.borgelt.net/cluster.html>

The full set of diagrams for the experiments is available at:

<http://www.borgelt.net/papers/nndv1.pdf> (color)

http://www.borgelt.net/papers/nndv1_g.pdf (greyscale)