Protokoll

Lernen mit Backpropagation und Momentum-Term

LV Künstliche Neuronale Netze

Matthias Jauernig (03INB), Michael Lahl (03IND)
FB IMN, HTWK Leipzig

13.06.06

Inhaltsverzeichnis

Zusammenfassung

Der Lernprozess von neuronalen Netzen, die ein überwachtes Lernen mit dem Backpropagation-Algorithmus durchführen, kann auf unterschiedliche Arten beschleunigt werden. Eine Variante stellt dabei das Lernen mit Momentum-Term dar, welches im Folgenden empirisch betrachtet werden soll, auch unter Einbeziehung von statischen sowie dynamischen globalen Lernraten.

1 Rahmenbedingungen

1.1 Lern-Aufgabe

Es gilt zu lernen:

- 20 Buchstaben (A,B,...,T),
- Jeder Buchstabe stellt ein Pixelbild der Größe 20x20 dar,
- Die Buchstaben sollen nach dem Lernen richtig klassifiziert werden können.

1.2 Lern-Prozess

Gelernt wird auf folgende Art und Weise:

- Als Lernalgorithmus wird Backpropagation verwendet,
- Gelernt wird eine festgelegte Epochenzahl,
- Eine Epoche entspricht dem einmaligen sequentiellen Trainieren des Netzes mit allen 20 Buchstaben,
- Die Gewichte werden in jedem Lernschritt aktualisiert (kein Batch-Lernen),
- Die initialen Gewichtsmatrizen sind für jeden Lernvorgang dieselben (z.B. durch einmaliges Speichern in einer Datei, die nachfolgend geladen werden kann).

1.3 Netz-Bestimmungen

- 3-schichtiges Feed-Forward-Netz,
- Netztopologie (Eingabe/Verborgen/Ausgabe): 400/20/20,
- Der Input eines E-Neurons N_i stellt den auf den Bereich $[0 \ldots 1]$ skalierten Grauwert des Buchstaben-Pixels $(i/20, i\%20)$ dar,
- Das A-Neuron mit dem größten Output bestimmt die Klasse, zu welcher der eingegebene Buchstabe am stärksten assoziiert werden kann,
- Die Gewichte werden wie folgt aktualisiert:

$$\Delta w_{ij}(t) = sw \cdot y_i \cdot \delta_j + \alpha \cdot \Delta w_{ij}(t-1)$$

sw ... Schrittweite
α ... Momentum-Faktor
$\Delta w_{ij}(t-1)$ ... Gewichtsänderung des letzten Lernschritts

- Als Fehlerfunktion werde verwendet:

$$E = \sum_{i, N_i \in A} (ys_i - y_i)^2$$

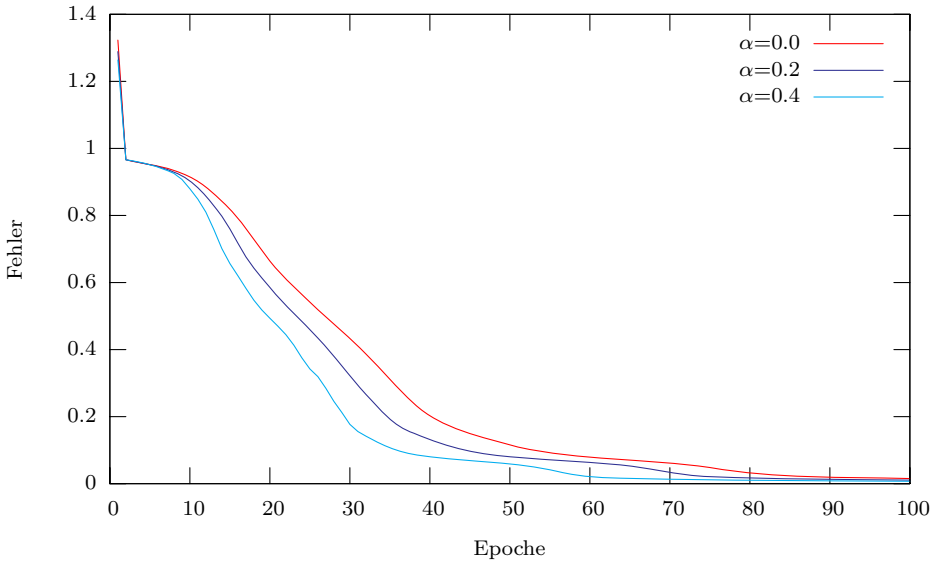
ys_i ... Soll-Ausgabe des Neurons N_i
y_i ... tatsächliche Ausgabe von N_i

2 Testfälle

2.1 Statische Lernrate

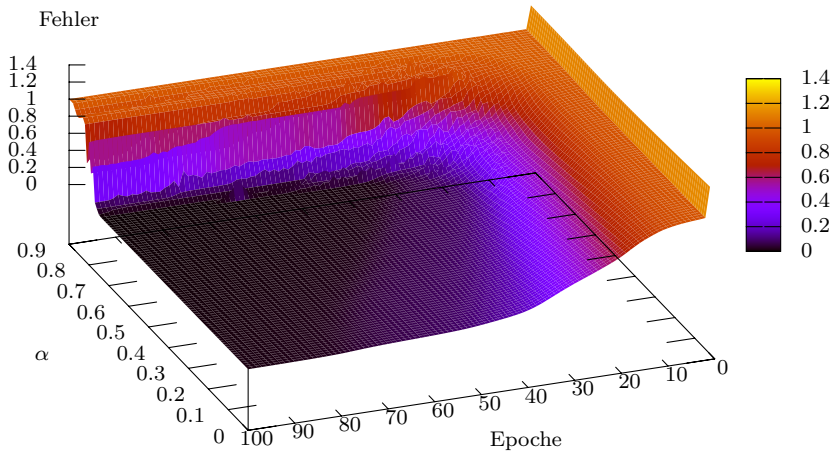
2.1.1 Fall: sw=1

- $\alpha \in 0.0, 0.2, 0.4$



Auswertung: Fehler fällt schneller ab, wenn Momentum-Faktor höher gewählt wird.

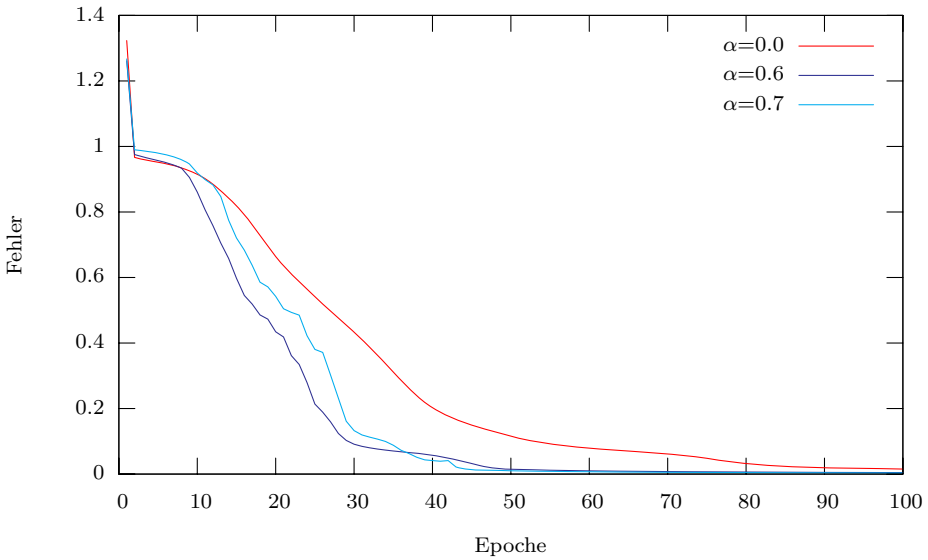
- Frage: Wie weit lässt sich α erhöhen?: $\alpha \in [0.0 \ldots 0.9]$



Auswertung: Fehler fällt mit zunehmendem α schneller ab, konvergiert für Werte größer ca. $\alpha = 0.7$ schlechter bzw. (z.B. für $\alpha = 0.9$) gar nicht mehr gegen ein annehmbares Minimum.

Begründung: Lernen mit Momentum verhindert Oszillationen des Gradienten durch Einbeziehen der vorigen Gewichtsänderungen. Ein zu großes α führt jedoch dazu, dass die alte Gewichtsänderung zu stark mit einbezogen wird, wodurch nicht mehr optimal bzw. gar nicht mehr in Richtung des Fehlerminimums gelernt wird.

- Frage: Wie stark ist die Verbesserung gegenüber Lernen ohne Momentum maximal ($\alpha \approx 0.7$)?



Auswertung: Nach 100 Epochen ist der Fehler beim Lernen ohne Momentum mit 0.0152995 3.4-mal so groß wie beim Lernen mit $\alpha = 0.6$ (0.00449479) und sogar 4.6-mal so groß wie bei $\alpha = 0.7$ (0.0033065).

- Folgende Tabelle gibt an, wieviele Epochen lang gelernt werden muss, um für ein bestimmtes α die angegebene Schranke *eps* des durchschnittlichen Fehlers (über alle 20 Muster) zu unterschreiten. Die Prozentzahlen geben die benötigte Epochenzahl in Vergleich zum Fall $\alpha = 0.0$ an:

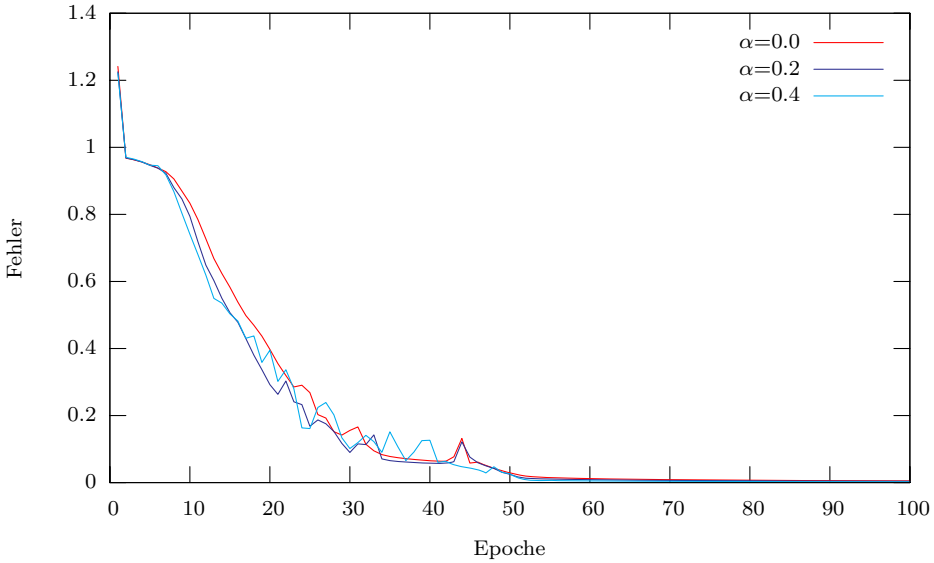
eps	$\alpha = 0.0$	$\alpha = 0.2$		$\alpha = 0.4$		$\alpha = 0.6$		$\alpha = 0.7$	
0.1	53	45	84%	36	67%	30	56%	35	66%
0.05	75	66	88%	53	70%	42	56%	39	52%
0.025	84	73	86%	59	70%	47	55%	43	51%
0.01	127	104	81%	82	64%	60	47%	51	40%
0.005	207	167	80%	130	62%	93	44%	75	36%
0.0025	364	291	79%	227	62%	160	43%	124	34%
0.001	826	658	79%	513	62%	356	43%	274	33%

Tabelle 1: Erreichen einer Fehlerschranke für versch. α

Auswertung: Abhängig von der Fehlerschranke *eps* und α wird hier bis zu 67% der Rechenzeit (Anzahl von benötigten Epochen) eingespart; mit einer Verkleinerung der Fehlerschranke werden i.A. sogar prozentual weniger Epochen benötigt.

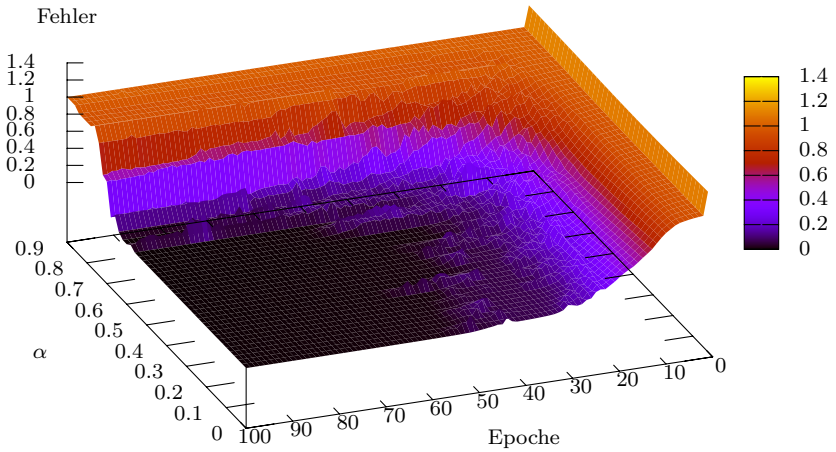
2.1.2 Fall: sw=2

- Versuch: Erhöhe die Lernrate mit dem Ziel noch schneller zu lernen.



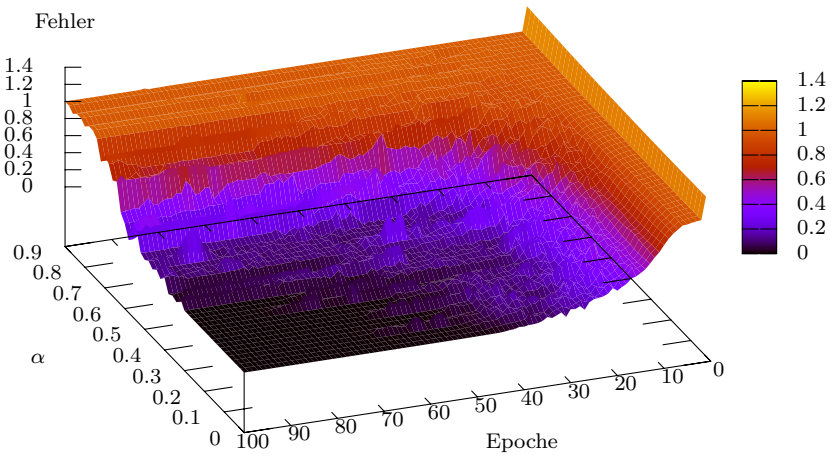
Auswertung: Verwendung eines Momentums bringt keine Vorteile mit sich.
Begründung: Eine höhere Lernrate bedeutet einen größeren Schritt Richtung negativer Gradient, ein höherer Momentum-Faktor einen größeren Schritt in Richtung der alten Gewichtsänderung. Bei zu hohem Momentum wird nicht mehr optimal in Richtung des Fehlerminimums gelernt. Da sw Einfluss auf $\Delta w(t-1)$ hat, genügen schon geringere α um einen optimalen Abstieg zu verhindern.

- Auswirkungen in der 3D-Ansicht:

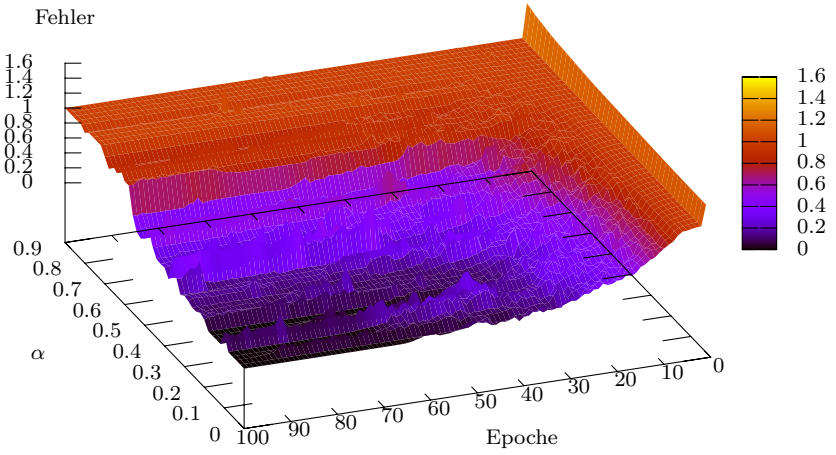


2.1.3 Fall: Größere sw

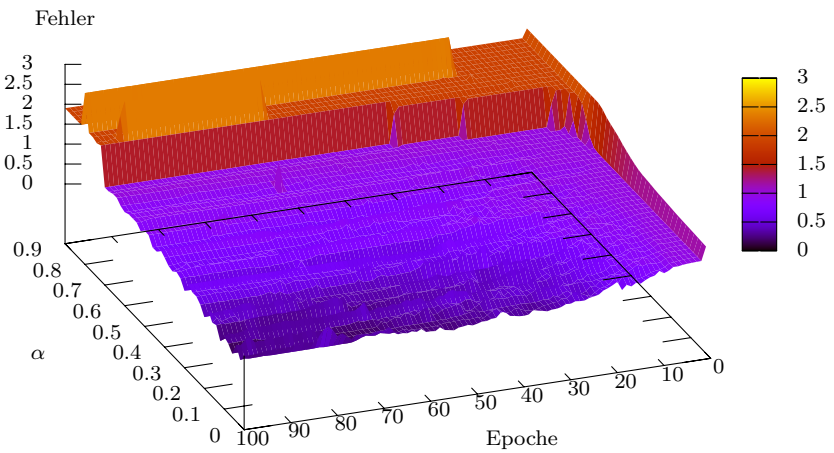
- Für noch höhere sw konvergiert der Fehler nur noch bei kleinen α gegen ein Fehlerminimum, bei höheren Momentum-Faktoren wird ein annehmbares Minimum schlecht oder gar nicht gefunden, da eine zu große Änderung über das Minimum hinaus stattfindet:
  sw=3:



  sw=4:



- Erhöht man sw noch weiter, konvergiert der Fehler nicht einmal für $\alpha = 0.0$ mehr gegen ein akzeptables Minimum, höhere Momentum-Faktoren sind wie zu erwarten undiskutabel:
  sw=6:



2.2 Dynamische Lernrate

Hierbei wird die Lernrate nach jeder Epoche wie folgt aktualisiert:

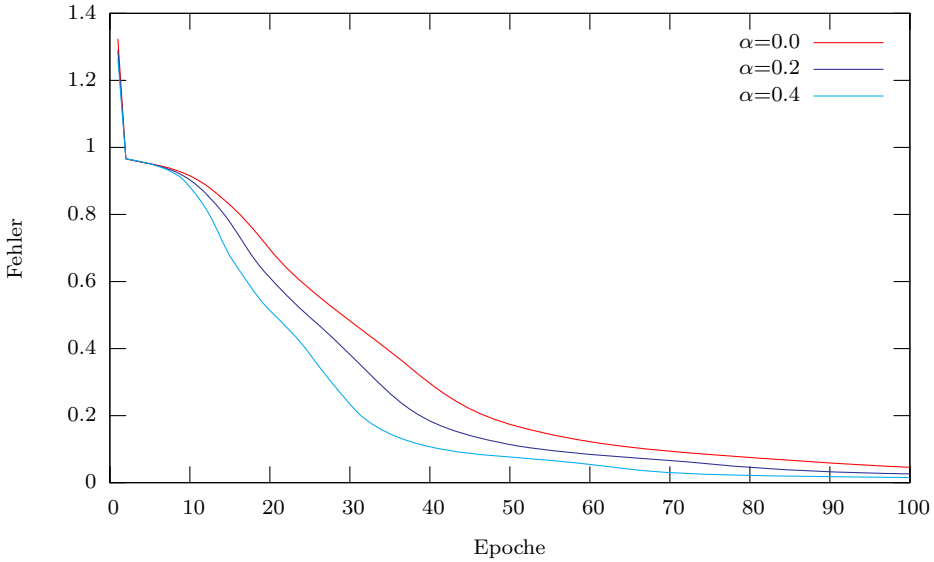
$$sw := \beta \cdot sw$$

β ist hierbei der Aktualisierungsfaktor. Ist $\beta > 1$, so steigt die Lernrate mit jeder Epoche an, was sicher nicht gewünscht ist. Mit $\beta = 1$ ergibt sich der Fall der statischen Lernrate. I.A. möchte man allerdings, dass die Lernrate mit fortschreitendem Lernprozess verkleinert wird, was bei $\beta < 1$ der Fall ist. Je nachdem, wie groß β gewählt wird, konvergiert die Lernrate schneller gegen 0. Dies kann z.B. mit der Halbwertzeit $T_{\frac{1}{2}}$ gemessen werden:

$$T_{\frac{1}{2}} = log_{\beta}\left(\frac{1}{2}\right)$$

2.2.1 Fall: sw=1, β=0.99

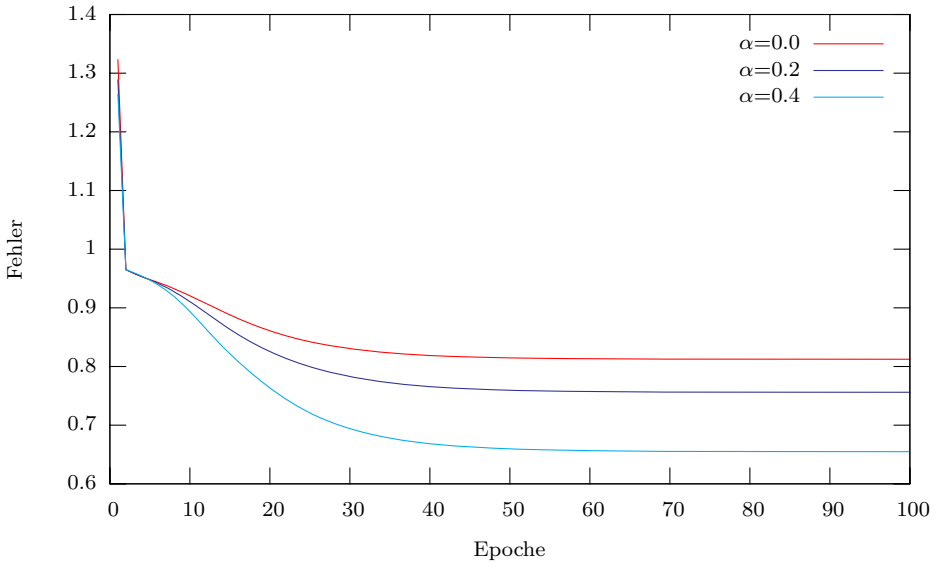
- Vermutung: da nach 69 Epochen sw≈0.5 ist, wird der Fehler mit zunehmendem Lernprozess nicht so schnell gegen 0 konvergieren wie im vergleichbaren Fall 2.1.1.
- Diagramm:



Auswertung: Diagramm bestätigt die Vermutung.

2.2.2 Fall: sw=1, β=0.9

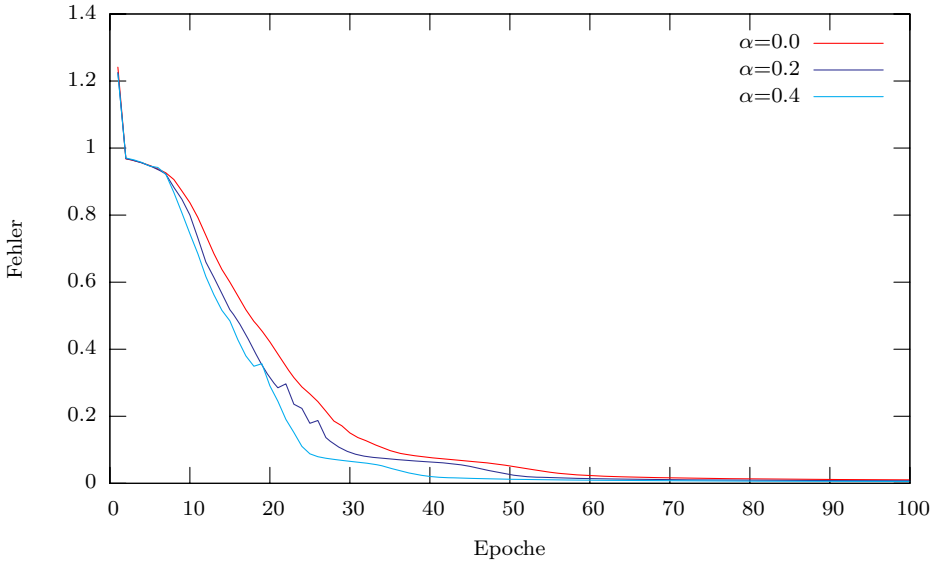
- Vermutung: sw beträgt nach ca. 7 Epochen nur noch 0.5, daher sehr langsames Lernen bzw. mit fortschreitender Lerndauer Stagnieren auf einem Fehler.
- Auch hier bestätigt das Diagramm die Vermutung:



2.2.3 Fall: sw=2, β=0.99

- Nach 69 Epochen beträgt sw nur noch 1.

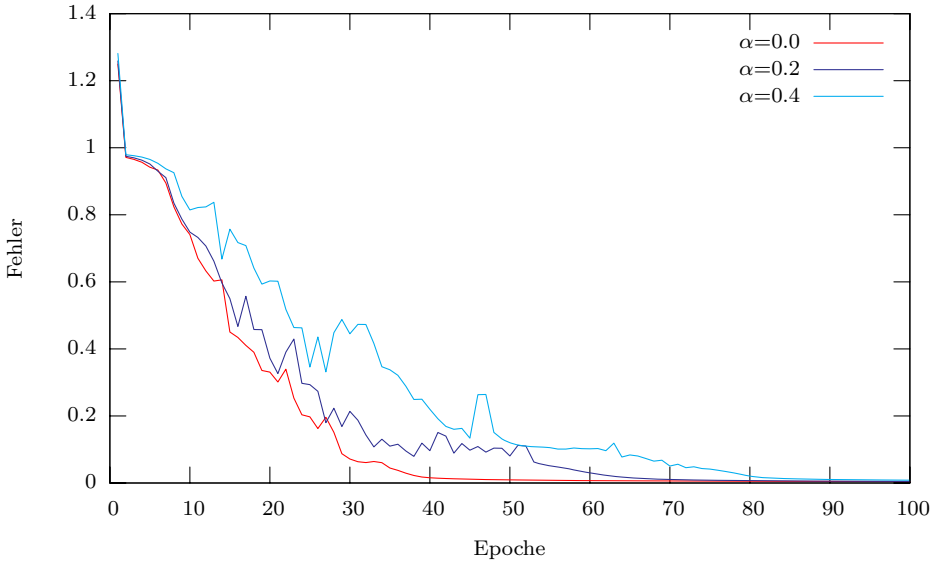
- Diagramm:



Auswertung: vergleichbar mit Fall 2.1.2, nur dass hier der Einsatz der Momentums noch zu einem sichtbaren Erfolg führt.
Begründung: Lernrate wird immer kleiner und somit der positive Effekt des Einsatzes des Momentums wieder sichtbar.

2.2.4 Fall: sw=4, β=0.99

- Vermutung: Konvergenz zu Beginn schlecht, da Startlernrate sehr groß, Einsatz des Momentums dürfte den negativen Effekt noch verstärken.
- Diagramm:



Auswertung: Vermutung größtenteils bestätigt, hohe Startlernrate vor allem bei Einsatz des Momentums hinderlich, gute Konvergenz stellt sich erst mit kleiner werdender Lernrate ein, da dann wieder gezielt auf ein Fehlerminimum hin gelernt werden kann.

2.2.5 Fall: Größere sw

- Für höhere sw stellt sich eine schnelle Fehlerkonvergenz nur bei kleineren β-Werten (und somit geringeren Halbwertzeiten) ein, bei größeren β verbleibt sw zu lange auf einem hohen Wert.
- Man kann die Startlernrate auch sehr hoch wählen, was aber den Effekt hat, dass zu Beginn des Lernprozesses nicht auf ein Fehlerminimum hin gelernt werden kann. Erst mit steigender Epochenzahl (und somit Verkleinerung von sw hin zu einem akzeptablen Wert) stellt sich dann eine Konvergenz ein, deren Geschwindigkeit aber aufgrund des schlechten Lernens zu Beginn nicht zu vergleichen ist mit der bestmöglichen (z.B. im Fall 2.1.1). Die Einbeziehung des Momentum-Terms verschlimmert die Auswirkungen einer zu hohen Startlernrate sogar noch.

3 Auswertung

Bei statischen Lernraten ist zu ersehen, dass der Einsatz eines Momentum-Terms den Lernprozess erheblich beschleunigen kann, vor allem für kleinere sw ($sw \in [1 \ldots 2]]$). Aus Tabelle 1 ist ersichtlich, dass hier je nach Fehlerschranke *eps* teilweise nur ca. 1/3 der Zeit benötigt wird wie beim Lernen ohne Momentum. Bei größeren Lernraten ist der Einsatz eines Momentums mit Vorsicht zu genießen. Da hier die hohen Lernraten mit in den Momentum-Term einfließen, wird nicht optimal in Richtung des Fehlerminimums gelernt, sodass sich vor allem bei größeren α keine Verbesserung bzw. sogar eine Verschlechterung des Lernverhaltens im Vergleich zum Lernen ohne Momentum ergibt. Mit größeren Lernraten steigt dieser Effekt an.

Dynamische Lernraten sind dann sinnvoll, wenn das Fehler-Tal des zu erreichenden Minimums sehr schmal ist. In dem vorliegenden Fall führen statische Lernraten allerdings schon zu einer sehr guten Konvergenz, sodass der Einsatz von dynamischen Lernraten keine Vorteile ergibt.

Verallgemeinernd lässt sich sagen, dass die Faktoren sw (Lernrate) und α (Momentum-Faktor) sehr sorgfältig gewählt werden sollten. Vor allem muss fallweise über ihren Einsatz (sowie den Einsatz einer dynamischen Lernrate) und ihre Größenordnung entschieden werden. Die hier dargestellten Auswertungen sind nur für das zugrunde gelegte Problem gültig und können i.A. nicht auf andere Probleme übertragen werden.