DIPLOMARBEIT

Schreiberadaption für Online-Handschrifterkennung

von

Eduard Hermann

eingereicht am 3. Februar 2004
beim Institut für Logik, Komplexität und
Deduktionssysteme der Universität Karlsruhe

Referent: Prof. Dr.rer.nat. Alexander Waibel
Betreuer: Dr. Hermann Hild

Danksagung

Diese Arbeit entstand in der Firma SMI Cognitive Software GmbH mit Unterstützung des Instituts für Logik, Komplexität und Deduktionssysteme. An dieser Stelle möchte ich dem Leiter des Instituts, Prof. Dr. Alexander Waibel, einen herzlichen Dank aussprechen für seine Betreuung und seine großzügige Erlaubnis, diese Arbeit extern durchführen zu können. Einen besonders herzlichen Dank möchte ich Hermann Hild und Roald Wolff aussprechen, die mich mit viel fachlicher Kompetenz und Geduld bei dieser Arbeit unterstützt haben. Bei SMI hat es mir an nichts gefehlt und ich habe sehr viel dazugelernt. Bedanken möchte ich mich noch bei meinen Freunden, die sehr bereitwillig und geduldig viele handschriftliche Daten gespendet haben - nochmal ein sehr herzliches Dankeschön.

Kurzfassung

Im Rahmen dieser Diplomarbeit werden verschiedene Verfahren zur Schreiberadaption eines Handschrifterkenners untersucht und im Vergleich zu einander bewertet, mit dem Ziel die Erkennungsleistung für den einzelnen Benutzer deutlich zu erhöhen. Für die Durchführung der Experimente wurden schreiberabhängige Daten gesammelt und bearbeitet. Die gesammelten Daten stammen von 17 deutschen Schreibern verschiedenen Geschlechts und Alters und enthalten Einzelzeichen, Wörter und Sätze mit Sonderzeichen sowohl in Druckschrift als auch in Kursivschrift. In dieser Arbeit werden folgende Adaptionsverfahren untersucht: OAM-Adaptionsverfahren eingesetzt auf zwei verschiedenen Ebenen des Handschrifterkenners, Nachtrainieren des MS-TDNN, Kombination der Verfahren, Adaption des Strafterms für die Wortübergänge bei der Suche (lp Parameter).

Das OAM (output adaptation module) ist ein Adaptionsverfahren bei dem die Ausgabe des neuronalen Netzes mit Hilfe von radialen Basisfunktionen (RBF-Funktionen) korrigiert wird. Die wichtigsten Parameter dieses Verfahrens sind das Zentrum und der Radius der RBF-Funktion sowie die Gewichtung des Beitrags zur Korrektur der Ausgabe. Die erste Möglichkeit, die Korrekturen mit OAM vorzunehmen, ergibt sich nach der Berechnung des Pfades im DTW-Modul. Die Fehlerrate bei diesem Versuch für Einzelzeichenerkennung wird dabei von 13.4% auf 7.3% reduziert. Beim Einsatz von OAM auf der Ebene der Zustandsschicht des MS-TDNN wird die Suche unverändert belassen mit dem Vorteil, dass auch die Kursivschrift auf einfache Weise adaptiert werden kann. Dabei kann die Fehlerrate von 13.4% auf 9.5% gesenkt werden. Bei der Adaption des lp-Parameters wurde keine nennenswerte Reduktion der Fehlerrate erzielt. Die Kombination des Nachtrainierens und OAM hat das Nachtrainieren allein nicht übertroffen. Das Nachtrainieren auf schreiberabhängigen Daten hat die Fehlerrate von 13.4% auf 5% gesenkt. Dies ist das beste Ergebnis, das durch die in dieser Arbeit untersuchten Adaptionsverfahren erzielt werden kann.

Inhaltsverzeichnis

1 Einleitung 6
- 1.1 Ziele der Adaption . . . 6
- 1.2 Aufgabenstellung der Arbeit . . . 7

2 Adaptionsverfahren im Überblick 8
- 2.1 Maximum Likelihood (ML) . . . 8
- 2.2 Maximum Likelihood Linear Regression (MLLR) . . . 9
- 2.3 Scaled Likelihood Linear Regression (SLLR) . . . 10
- 2.4 Maximum A Posteriori (MAP) . . . 11
- 2.5 Output Adaptation Module (OAM) . . . 13
  - 2.5.1 Allgemeine Information . . . 13
  - 2.5.2 OAM im Detail . . . 15

3 Handschrifterkenner: Systemüberblick 19

4 Daten 22
- 4.1 Schreiberabhängige Daten . . . 22
- 4.2 Schreiberunabhängige Daten . . . 23
  - 4.2.1 Aufnahme-Hardware . . . 25
  - 4.2.2 Die UKA Datenbasis . . . 26
  - 4.2.3 Die UNIPEN Datenbasis . . . 27
  - 4.2.4 Die IRONOFF Datenbasis . . . 28
  - 4.2.5 Die smiInkE und smiInkF Datenbasen . . . 28
- 4.3 Leistungsmessung des Systems . . . 30
- 4.4 Erkennungsergebnisse des Systems auf allgemeinen Daten . . . 30

5 Adaptionsversuche 32
- 5.1 Überblick . . . 32
- 5.2 Adaption des neuronalen Netzes durch Nachtrainieren . . . 33
- 5.3 OAM: DTW-Ebene . . . 42
- 5.4 OAM: Phonem-Ebene . . . 51
- 5.5 Kombination . . . 55
  - 5.5.1 Nachtrainieren und OAM-Adaption auf der DTW-Ebene 55
- 5.6 Suche . . . 60
  - 5.6.1 Language Penalty (lp) . . . 60
- 5.7 Vergleich der Verfahren . . . 62

6 Zusammenfassung und Ergebnisse 63
- 6.1 Zusammenfassung . . . 63
- 6.2 Die wichtigsten Ergebnisse und Beiträge . . . 63

7 Literaturverzeichnis **66**

1 Einleitung

Im Zeitalter der rasanten Entwicklung der Kommunikationstechnologien gewinnt die automatische Mustererkennung für die Mensch-Maschine-Kommunikation (MMK) sehr stark an Bedeutung. Ziele sind zum einen die benutzerfreundliche Interaktion zwischen Mensch und Maschine und zum anderen die effiziente Verarbeitung durch die Automatisierung der Anwendungen im Multimediabereich. Die gewöhnliche MMK per Tastatureingabe und Bildschirmausgabe soll durch natürlichere Kommunikationsformen ersetzt oder ergänzt werden. Dazu gehören automatische Symbol- und Handschrifterkennung sowohl im On-line- (z.B. elektronische Notizbücher, Bankformulare) als auch im Off-line- (z.B. Postautomatisierung) Bereich.

1.1 Ziele der Adaption

Eine Möglichkeit, die Erkennungsleistung für bestimmte Anwendungen zu steigern geben die Adaptionsverfahren. Das Ziel besteht darin, bestehende Modelle (HMMs oder repräsentative Prototypen einer Klasse, neuronale Netze, etc.) durch Adaption an neue, spezifische Daten anzupassen. Notwendig für eine Adaption sind, wie beim Training, die Merkmale der Daten und die entsprechenden Label. In der Regel handelt es sich dabei um eine Adaption eines allgemeinen (schreiberunabhängigen) Erkennungssystems auf einen bestimmten Schreiber. Prinzipiell sind aber auch Adaptionen auf spezielle Schreibweisen möglich. Durch die Adaption verbessert sich in der Regel die Erkennungsrate für die neuen Daten, die ursprünglichen Basisdaten werden jedoch schlechter erkannt.

Warum werden überhaupt Adaptionsverfahren eingesetzt und kein "normal" trainiertes Erkennungssystem? Dafür sind zwei Gründe ausschlaggebend: Zum einen spielt die benötigte Datenmenge eine große Rolle und zum anderen besteht bei Adaptionsverfahren die Möglichkeit einer unüberwachten Anpassung der Modelle.

Im Allgemeinem ist die Datenmenge, die für eine Adaption benötigt wird, deutlich geringer als die notwendige Trainingsmenge für ein völlig neues System. Bei der Adaption können bereits bestehende Modelle genutzt werden, so dass auch bei wenigen Daten die Parameter robust geschätzt werden können. Ein Nebeneffekt ist die kürzere Trainingsdauer, die sich bedingt durch die kleinere Datenmenge ergibt. Da bei der Adaption auch unüberwachtes Training möglich ist, ergeben sich weitere Vorteile bezüglich der Zeit und den Kosten. Unüberwacht bedeutet, dass die Labels, die für die Adaption benötigt werden, nicht manuell, sondern automatisch mit dem Basissystem

erzeugt werden. Die Standard-Anwendung von Adaptionsverfahren ist die Anpassung eines bestehenden Modells an einen bestimmten Schreiber. Diese Schreiber-Adaption eines schreiberunabhängigen Handschrifterkennungssystems ist in erster Linie in der On-line-Erkennung (z.B. bei PDAs) sinnvoll. Hier bieten Schreiber-Adaptionsverfahren die Möglichkeit, die Erkennungsrate und somit die Benutzerakzeptanz zu steigern, da ein schreiberabhängiges System im Allgemeinem besser als ein schreiberunabhängiges ist. In Fällen, bei denen nur ein Benutzer das Gerät bedient und die Steigerung der Fehlerrate für andere, fremde Personen kein Problem darstellt, ist die Adaption sinnvoll. Hier ist eine überwachte und eine unüberwachte Adaption denkbar. Die Effizienz einer überwachten Adaption ist höher, da die Werte korrekt gelabelt sind. Das bedeutet, dass weniger, aber dafür manuell gelabelte Adaptionsbeispiele vom Benutzer eingegeben werden müssen. Bei einer unüberwachten Adaption kann ein System während der Benutzung ständig weiter adaptieren, ohne dass diese Anpassung für den Schreiber mit Arbeit verbunden ist. Die zur Verfügung stehende Adaptionsmenge ist somit gewissermaßen unendlich, allerdings setzt dieses Verfahren eine bestimmte Basis-Erkennungsrate voraus. Sind die Erkennungsergebnisse des schreiberunabhängigen Systems zu schlecht, driftet das adaptierte System in die "falsche" Richtung und es wird keine Verbesserung erzielt.

1.2 Aufgabenstellung der Arbeit

Im Rahmen dieser Diplomarbeit sollen Techniken zur Schreiberadaption untersucht werden. Das bedeutet, dass ein zunächst schreiberunabhängig ausgelegter Erkenner anhand einer möglichst kleinen Menge von Trainingsbeispielen auf die Erkennung eines bestimmten Schreibers hin optimiert wird. Die dazu notwendige Anpassung der Erkennerparameter kann auf verschiedenen Erkennungsebenen stattfinden: In der Vorverarbeitung, beim Training des Klassifikators (z.B. Adaption des schreiberunabhängigen Neuronalen Netzes), bei der Interpretation der Klassifikationsergebnisse durch output adaptation module (OAM), bei der Parametrisierung der Suche. Dabei sollen Adaptionsverfahren bevorzugt werden, die mit einer kleinen Menge von Adaptionsdaten auskommen und nicht wesentlich mehr Ressourcen bezüglich Speicher und Rechenleistung benötigen als der schreiberunabhängige Erkenner. Für die praktische Anwendung ist weiterhin die Frage interessant, wie die Daten zur Adaption gewonnen werden, z.B. durch die einmalige Eingabe einer expliziten Vorgabe, durch die Rückmeldung von Fehlerkennungen während der initialen Adaptionsphase, oder lebenslänglich durch die inkrementelle Adaption.

2 Adaptionsverfahren im Überblick

Je nach Aufbau des schrifterkennenden Systems und des Adaptionsziels sind auch verschiedene Adaptionstechniken möglich. In der Schrifterkennung lassen sich verschiedene Kategorien für die Adaptionsverfahren definieren:

- Art der Klassifikatoren bzw. der Struktur des Erkennungssystems: Adaption von Prototypen, Hidden Markov Modellen oder neuronalen Netzen
- Anwendungsbereich: Adaption auf einen speziellen Schreiber oder auf Schreibweisen bestimmter festzulegender Gruppen
- Modus: überwachte oder unüberwachte Adaption
- Adaptionstechnik: ML, MLLR, SLLR, MAP, etc.

In den nachfolgenden Kapiteln werden Adaptionsverfahren ML, MLLR, SLLR, MAP kurz beschrieben. Alle diese Verfahren sind nur für die HMM-basierte Erkennungssysteme anwendbar, da sie die HMM-Modelle λ an die Adaptionsmenge anpassen. Lediglich das ML-Verfahren ist prinzipiell auch für Systeme anwendbar, die ein Nachtrainieren erlauben. Eine ausführlichere Beschreibung dieser Verfahren ist in [A.B02] zu finden. In dieser Arbeit wird ein auf MS-TDNN basierendes System für die Adaption verwendet, somit werden andere Adaptionstechniken untersucht.

2.1 Maximum Likelihood (ML)

Die Adaption - oder besser gesagt Nachtrainieren - nach dem Maximum Likelihood Kriterium [S.Y00, uB86] basiert, wie das normale Training von HMMs auf dem EM-Verfahren. Das Ziel besteht darin, die Parameter der HMMs λ, die bereits auf einer Basisdatenmenge (schreiberunabhängige Schriftproben) trainiert werden, so an die neuen (schreiberabhängige Daten) Merkmalsvektoren X anzupassen, dass die Likelihood Wahrscheinlichkeit (das Ähnlichkeitsmaß oder auch die Plausibilität) $P(X|\lambda)$ maximiert wird:

$$\lambda_{ML} = \underset{\lambda}{\operatorname{argmax}} P(X|W) = \underset{\lambda}{\operatorname{argmax}} P(X|W,\lambda) \Rightarrow \begin{cases} \underline{\mu}_{wi} \overset{EM}{\rightarrow} \underline{\mu}_{ML} \\ \sum_{wi} \overset{EM}{\rightarrow} \sum_{ML} \\ \omega_{wi} \overset{EM}{\rightarrow} \omega_{ML} \\ A_{wi} \overset{EM}{\rightarrow} A_{ML} \end{cases} \tag{2.1}$$

Der Baum-Welch Algorithmus wird auch hier zur Lösung eingesetzt. Es werden, wie beim Training, jeweils die HMMs λ des zugehörigen Trainings- bzw. Adaptionswortes W angepaßt. Dieses Adaptionsverfahren kann wahlweise alle oder nur einige der HMM-Parameter beeinflussen, die sich je nach Modellierungstechnik in der Regel aus den Gauß-Funktionen (Mittelwertvektor $\underline{\mu}$, Kovarianzmatrix $\sum$), den Gewichten ω und der Transitionsmatrix A zusammensetzen. Der Index wi in 1 steht hier für das schreiberunabhängige System. Je nach verfügbarer Adaptionsmenge kann die Anzahl der zu adaptierenden Parameter eingeschränkt werden. Häufig werden lediglich die Mittelwerte und/oder die Gaußverteilungen angepasst, wie es auch bei den expliziten Adaptionsverfahren MLLR, SLLR und MAP üblich ist.

2.2 Maximum Likelihood Linear Regression (MLLR)

Bei dem Maximum Likelihood Linear Regression Adaptionsverfahren [uP94] wird das Problem der geringen Adaptionsmenge durch Clustern von Modellen bzw. Gaußverteilungen und einer anschließenden gemeinsamen Transformation mit einer Regressionsmatrix M zur Anpassung gelöst. Das MLLR-Verfahren wurde ursprünglich von Leggetter [uP94] für die Spracherkennung eingeführt. Wie beim ML-Verfahren soll auch diese Methode anhand kontinuierlicher HMMs erläutert werden, das Prinzip kann aber auf die semikontinuierliche Modellierungstechnik übertragen werden. Für diskrete HMM-Strukturen ist dieses Verfahren nicht direkt übertragbar, wie sich aus dem Algorithmus ersehen läßt. In [J.R00] sind ergänzende Adaptionsverfahren auch für diskrete HMM-Strukturen erklärt.

Die Zielfunktion ist wie beim ML-Verfahren, die Maximierung der Likelihood $P(X|\lambda)$ durch Anpassung der HMM-Parameter. Standardmäßig ist damit gemeint, dass zum Training des Modells λ nur die passenden Beispiele W gewählt werden. Lediglich zur Unterscheidung von MLLR- und SLLR-Verfahren, wird explizit von $P(X|W,\lambda)$ gesprochen. Allerdings wird die Anpassung durch eine Regressionsmatrix M durchgeführt, mit der die ursprünglichen HMM-Parameter wie Mittelwertvektor $\underline{\mu}$ und Kovarianzmatrix $\sum$ multipliziert werden:

$$\lambda_{MLLR} = \underset{\lambda}{\operatorname{argmax}}\, P(X|W,\lambda) = \underset{M}{\operatorname{argmax}}\, P(X|W,\lambda,M) \tag{2.2}$$

Die folgenden Betrachtungen beziehen sich auf die Adaption des Mittelwertvektors $\underline{\mu}$. Für die Berechnungen wird ein erweiterter Mittelwertvektor $\underline{\hat{\mu}}$ eingeführt:

$$\underline{\hat{\mu}}_{wi} = \begin{pmatrix} v \\ \underline{\mu}_{wi} \end{pmatrix} \tag{2.3}$$

Der Offset v ist notwendig, um über die Matrixmultiplikation auch eine Verschiebung des Vektors erzielen zu können:

$$\underline{\mu}_{MLLR} = M_{MLLR} \cdot \underline{\hat{\mu}}_{wi} \tag{2.4}$$

Die Matrix M hat die Dimension $(D+1) \times D$. Abhängig von der Clusterung der Gaußdichten wird jeder Mittelwert einzeln oder alle Mittelwerte eines Clusters mit derselben Matrix M multipliziert. Das heißt, für jedes Cluster wird eine Matrix M ermittelt. Sind alle Gaußfunktionen in einem Cluster vereint, spricht man von einer globalen Adaption mit nur einer Matrix M_{global}. Das problem ist die Bestimmung dieser Matrix, die die MLLR-Zielfunktion maximieren soll. Zur Lösung werden zwei Alternativen aufgezeigt: die Standard-Lösung über den Baum-Welch Algorithmus oder eine Variante, die auf einem Gradientenabstieg beruht. Eine ausführliche Beschreibung wird in [uP94] angegeben.

2.3 Scaled Likelihood Linear Regression (SLLR)

Das Scaled Likelihood Linear Regression (SLLR) Adaptionsverfahren ist gewissermaßen eine diskriminative MLLR-Adaption (vgl.z.B. [F.W00, V.V96]). Die Zielfunktion ist die Maximierung der Likelihood $P(X|W,\lambda)$ für die zur korrekten Wortsequenz W gehörenden Modelle λ und gleichzeitiger Minimierung von $P(X|\lambda)$ durch Anpassung der HMM-Parameter:

$$\lambda_{SLLR} = \underset{\lambda}{\operatorname{argmax}} \frac{P(X|W,\lambda)}{P(X|\lambda)} \tag{2.5}$$

mit:

$$P(X|\lambda) \approx \sum_{bestenW} P(X|W,\lambda)P(W) \tag{2.6}$$

Im Unterschied zur MLLR-Adaption muss auch der Term $P(X|\lambda)$ geschätzt werden. Dies könnte, wie in Gl. 2.6 angedeutet, durch die Auswertung von N-Besten-Listen erfolgen. Da die Adaptionsmenge und somit auch die Anzahl dir Verwechslungen zum diskriminativen Training eher gering ist, wird ein frame-basierter Ansatz verfolgt (siehe [F.W00]):

$$P(X|\lambda) \approx \prod_t p(\underline{x}_t) \tag{2.7}$$

mit:

$$p(\underline{x}_t) \approx \sum_q p(q_t)p(\underline{x}_t|q_t) \tag{2.8}$$

Die Zustandsfolge Q wird aus dem Viterbi-Pfad entnommen. $p(\underline{x}_t|q_t)$ kann wiederum durch die Ausgabeverteilungsdichte b, die von M_{SLLR} abhängt, dargestellt werden.

Auch hier wird, wie beim MLLR-Verfahren, die Adaption der Mittelwerte $\underline{\mu}_{wi}$ der Gaußfunktionen durchgeführt, indem diese mit einer Regressionsmatrix M_{SLLR} transformiert werden.

$$\underline{\mu}_{SLLR} = M_{SLLR} \cdot \hat{\underline{\mu}}_{wi} \tag{2.9}$$

Zusammenfassend ergibt sich dann:

$$\lambda_{SLLR} = \underset{\lambda}{\operatorname{argmax}} \prod_t \frac{p(\underline{x}_t|q_t, \lambda)}{\sum_q p(q_t|\lambda) p(\underline{x}_t|q_t, \lambda)} \tag{2.10}$$

Definiert man L_{SLLR} als logarithmische Likelihood, so ergibt sich die mit dem PROP-Verfahren zu lösende Gleichung zu (siehe [F.W99]):

$$\frac{\partial L_{SLLR}(m)}{\partial m_{rs}} = \sum_t \left(\frac{\partial \log p(\underline{x}_t|q_t)}{\partial m_{rs}} - \frac{\partial \log \left(\sum_q p(\underline{x}_t) p(\underline{x}_t|q_t) \right)}{\partial m_{rs}} \right) \tag{2.11}$$

Betrachtet man die Gl. 2.11, wird deutlich, dass dieses Verfahren aufgrund des rechten Terms in der Summe sehr viel rechenzeitaufwendiger ist als das MLLR-Verfahren.

Die SLLR-Adaption wird in der Regel mit einer globalen Regressionsmatrix durchgeführt. Problematisch wird das Verfahren, wenn nicht alle Zeichen in den Adaptionsdaten vorkommen. Die Modelle der nicht vorkommenden Zeichen werden zwar mit anderen ähnlichen Modellen geclustert, können aber nicht von anderen Zeichen diskriminativ abgespalten werden, da keine Daten $p(\underline{x}|q)$ für das richtige Modell λ zur Verfügung stehen.

2.4 Maximum A Posteriori (MAP)

Das Maximum A Posteriori (MAP) Verfahren (siehe [uP73, uCH91, uCH94]) behandelt das Problem der geringen Adaptionsmengen nicht durch Zusammenfassung von Modellen, sondern durch Berücksichtigung der a priori Wahrscheinlichkeit. Diese Adaptionstechnik wird häufig auch als Bayes-Schätzung bezeichnet. Mit Hilfe der MAP-Adaption werden die Mittelwertvektoren und Kovarianzmatrizen der kontinuierlichen HMMs angepasst.

Im Gegensatz zur ML-Adaption wird die a posteriori Wahrscheinlichkeit maximiert. Dazu ergibt sich nach Bayes Formel der folgende Zusammenhang,

wie in Gl. 2.12 dargestellt. Die a priori Wahrscheinlichkeit $P(\lambda)$ der Modelle bzw. Zeichen wird anhand der Basistrainingsdaten bestimmt.

$$\lambda_{MAP} = \underset{\lambda}{\operatorname{argmax}} P(\lambda|X) \approx \underset{\lambda}{\operatorname{argmax}} P(X|\lambda)P(\lambda) \tag{2.12}$$

Wird die a priori Wahrscheinlichkeit $P(\lambda)$ als gleichverteilt angenommen, ergibt sich die in Kap. 2.1 beschriebene ML-Schätzung. Für eine analytische Optimierung der MAP-Zielfunktion werden die a priori Wahrscheinlichkeiten $P(\lambda_k)$ der einzelnen Klassen einer Dirichlet-Verteilung angepasst (*konjugierte Familien*). Somit folgen Gl. 2.13 und 2.14, wie in [uP73, uCH91] gezeigt wird:

$$\underline{\mu}_{MAP_{ij}} = f_L \cdot \underline{\mu}_{wd_{ij}} + (1 - f_L) \cdot \underline{\mu}_{wi_{ij}} \tag{2.13}$$

mit:

$$f_L = \frac{\sum_t l_{ij}(t)}{(\sum_t l_{ij}(t)) + \tau} \tag{2.14}$$

Für diesen Fall kann die Schätzung nach Gl. 2.12 wiederum mit dem EM-Algorithmus erfolgen. Diese Schätzung der neuen Parameter kann iterativ wiederholt werden. $\underline{\mu}_{wi_{ij}}$ beschreibt den Mittelwert der ursprünglichen (schreiberunabhängigen) Daten, $\underline{\mu}_{wd_{ij}}$ beschreibt entsprechend den Mittelwert der neuen beobachteten Datenmenge. Analog gilt die Gleichung 2.15 ebenfalls für das ML-Training:

$$\underline{\mu}_{wd_{ij}} = \frac{\sum_t l_{ij}(t) \cdot \underline{x}_t}{\sum_t l_{ij}(t)} \tag{2.15}$$

Dabei steht $l_{ij}(t)$ für die Aufenthaltswahrscheinlichkeit im Zustand $q_t = s_i$ bei der Gauß-Funktion m_j und kann mit Hilfe des Vorwärts-Rückwärts-Algorithmus (Wahrscheinlichkeiten α_t und β_t) bestimmt werden:

$$l_{ij}(t) = P(q_t = s_i, m_j|X, \lambda) \tag{2.16}$$

Für den Fall einer multivariaten Gaußverteilungsdichte je Zustand s_i ergibt sich $l_{ij} = \gamma_i$. Der Parameter τ in Gl. 2.14 bezieht sich auf die a priori Wahrscheinlichkeitsverteilung und wird empirisch gesetzt. Anhand dieses Parameters lässt sich der Einfluss der Adaptionsdaten regeln. Ist die Aufenthaltswahrscheinlichkeit in einer bestimmten Gaußfunktion sehr klein, so ist der Faktor f_L klein und somit ergibt die MAP-Schätzung einen Wert, der nahe dem schreiberunabhängigen System liegt. Für große Stichproben strebt die MAP-Schätzung gegen die ML-Schätzung.

Dies bedeutet, dass bei der MAP-Adaption im Gegensatz zur MLLR- oder SLLR- Adaption der Mittelwert jeder Gaußverteilung separat geschätzt

werden muss (und kann), abhängig vom ursprünglichen Mittelwert, den eigestellten Gewichten und den Adaptionsdaten. Theoretisch folgt daraus - wie es sich in der Spracherkennung auch bestätigt hat - ein größerer Bedarf an Adaptionsdaten als beispielsweise bei der MLLR-Adaption, bei der Cluster gebildet werden. In [A.B02] beschriebenen Ergebnisse zur Schreiber-Adaption zeigen jedoch, dass die zur erfolgreichen MAP-Adaption benötigte Datenmenge sehr gering ausfallen kann. Der Vorteil dieser Adaptionsmethode liegt gerade in der separaten Schätzung der Parameter. Hier kann, wenn die a priori-Verteilung aussagekräftig ist und die Datenmenge ausreicht, eine sehr spezifische Anpassung je Modell erfolgen. Damit ergibt sich gleichzeitig das Problem der Schätzung von Klassen (bzw. Zeichen), die nur sehr selten oder gar nicht in den Adaptionsdaten vorkommen. Diese haben eine sehr geringe Aufenthaltswahrscheinlichkeit und werden somit nicht (oder nur wenig) angepasst.

Bei der Adaption von semi-kontinuierlichen HMMs wurden nicht die Mittelwerte und/oder Varianzen, sondern die Gewichte adaptiert. Das Codebuch, bestehend aus dem Pool von Gaußverteilungsdichten, wird konstant gehalten und nur die Gewichte w_{ij} in den einzelnen Zuständen werden zwischen dem ursprünglichen und dem nach der ML-Methode geschätzten neuen Gewicht interpoliert.

Eine weitere Adaptionsvariante ist die Kombination aus MLLR und MAP, die die Vorteile beider Verfahren beinhaltet. Mit einer MLLR-Adaption werden zuerst die ermittelten Cluster transformiert (also alle Zeichen, auch die, die in den Daten selbst nicht vorkommen) und anschließend kann eine Art Feinjustierung mittels des MAP-Verfahrens erfolgen.

2.5 Output Adaptation Module (OAM)

2.5.1 Allgemeine Information

Das in diesem Kapitel beschriebene Adaptionsverfahren wird als output adaptation module (OAM) bezeichnet und hat das Ziel, die Ausgabe des neuronalen Netzes zu korrigieren, falls dies erforderlich ist. In der Praxis tritt häufig der Fall auf, dass ein Benutzer leicht untypische Eingaben macht, die zu Fehlerkennung führen. Die Lösung dieses Problems wäre der Bau eines Erkenners, der nur wenige Daten als Trainigsdaten benötigt, um die Adaption während der Benutzung an die untypische Schreibweise des Benutzers durchzuführen.

In dieser Arbeit wird ein Adaptionsverfahren, dass in [CP97] beschrieben ist, in den bestehenden Handschrifterkenner der Firma SMI integriert. Die dabei gewonnenen Ergebnisse ermöglichen den direkten Vergleich mit

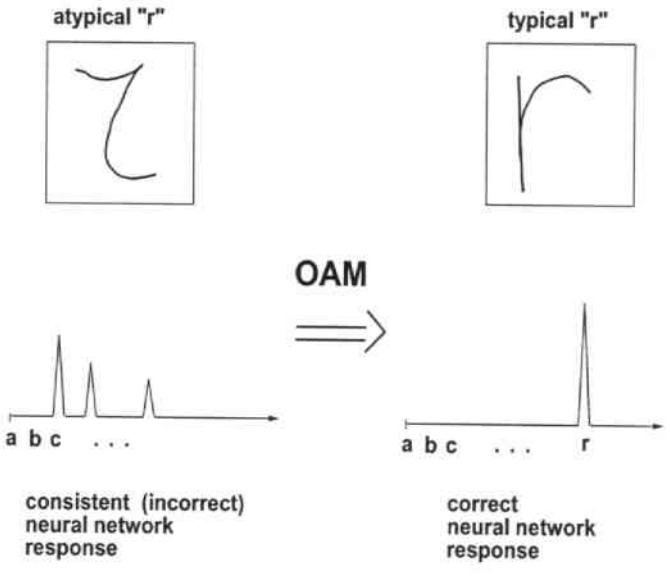

Abbildung 1: Die Funktionsweise des OAM-Verfahrens

den Ergebnissen, die durch "klassisches" Nachtrainieren gewonnen werden. OAM-Verfahren entstand bei der Entwicklung eines auf dem neuronalen Netz basierenden schreiberunabhängigen Handschrifterkenner. Dabei wurde festgestellt, dass die Ausgabe des Netzes konsistent mit der Eingabe ist, auch wenn die Ausgabe inkorrekt ist. Aus dieser Tatsache ist die Idee naheliegend, die untypischen Eingaben an der Ausgabe des neuronalen Netzes zu korrigieren. Zu diesem Zweck wird OAM an die Ausgabe des neuronalen Netzes platziert und lernt automatisch aus den inkorrekten konsistenten Ausgabevektoren korrekte Vektoren zu produzieren (siehe die Abbildung 1). Die Einheiten von OAM sind radiale Basisfunktionen (RBF) [M.P87]. Die Adaption dieser RBF-Einheiten erfolgt unter der Benutzung einer vereinfachten Version des Algorithmus von Platt Resource Allocating Network (RAN) [J.P91, VM93]. Die Anzahl der Einheiten, das RAN alloziert, wächst sublinear mit der Anzahl der präsentierten Lernbeispiele, im Gegensatz zu den anderen Algorithmen, die zu jedem Lernbeispiel eine neue Einheit allozieren. In [CP97] werden folgende Eigenschaften des OAM aufgeführt, die nützlich sind für die Verwendung in einem schreiberadaptiven Handschrifterkenner:

- Die Adaption ist sehr schnell: der Benutzer braucht nur wenige eigene

Lernbeispiele, um die Adaption durchzuführen.

- Ein sehr geringer Geschwindigkeitsverlust beim Einsatz des Systems.
- Eine relativ kleine Menge am zusätzlichen Speicherplatz pro Benutzer ist erforderlich
- OAM ist nicht nur auf neuronales Netz basierte Handschrifterkenner beschränkt
- Die Ausgabe von OAM ist ein Vektor mit Wahrscheinlichkeiten, was natürlich wertvoller ist als eine einfache Entscheidung in Bezug auf die weitere kontextuelle Nachbearbeitung

2.5.2 OAM im Detail

An dieser Stelle wird die detailierte Beschreibung des OAM-Verfahrens gegeben, wobei in dem vorigen Kapitel die Verwendung des OAM im Handschrifterkenner beschrieben ist.

Das OAM-Verfahren wandelt den Ausgabevektor $\vec{V}$ des neuronalen Netzes in die benutzerabhängige Ausgabe $\vec{O}$ durch die Addition eines Adaptionsvektors $\vec{A}$ um:

$$\vec{O} = \vec{V} + \vec{A} \tag{2.17}$$

In Abhängigkeit von dem eingesetzten Algorithmus für das Training des neuronalen Netzwerkes, beide Vektoren, die Netzwerkausgabe $\vec{V}$ sowie benutzeradaptierte Ausgabe $\vec{O}$ können die a posteriori Klassenwahrscheinlichkeiten darstellen, die wichtig für weitere Verarbeitung im Erkenner sind.

Das Ziel von OAM ist die adaptierte Ausgabe O_i näher an die ideale Zielausgabe T_i zu bringen. In den Experimenten von [CP97], wie auch in dieser Arbeit, wird der Wert 0.9 für das richtige Zeichen bzw. Phonem gewählt und 0.1 sonst.

Der Adaptionsvektor $\vec{A}$ wird durch eine Menge der radialen Basisfunktionen berechnet, die als Eingabe den Vektor $\vec{V}$ erhalten:

$$O_i = V_i + A_i = V_i + \sum_j C_{ij} \Phi_j(\vec{V}), \tag{2.18}$$

$$\Phi_j(\vec{V}) = f\left(\frac{d(\vec{V}, \vec{M}_j)}{R_j}\right). \tag{2.19}$$

wobei $\vec{M}_j$ das Zentrum für die j-te radiale Basisfunktion, d eine Distanzfunktion zwischen $\vec{V}$ und $\vec{M}_j$, R_j der Radius der jeweiligen RBF, f eine Glockenfunktion der RBF ist. C_{ij} ist der Gewichtsfaktor, der den Einfluss der j-ten

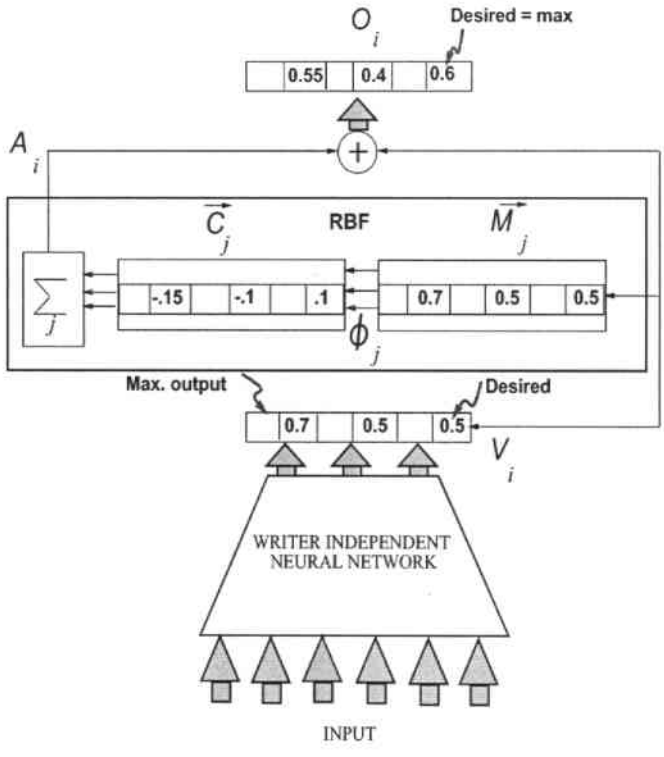

Abbildung 2: Die Struktur von OAM

RBF auf die Ausgabe O_i steuert. Die Vektoren $\vec{M}_j$ werden als *memories* in OAM, und Vektoren $\vec{C}_j$ als *Korrekturvektoren* bezeichnet (siehe Abbildung 2). Die Glockenfunktion f ist eine fallende polynomiale Funktion:

$$f(x) = \begin{cases} (1-x^2)^2, & \text{if } x < 1; \\ 0, & \text{sonst.} \end{cases} \tag{2.20}$$

Die Distanzfunktion d berechnet die euklidische Distanz zwischen den beiden Eingabevektoren, nachdem die Werte dieser beiden Vektoren auf den Interval $[0.1, 0.9]$ beschnitten werden, um das störende Rauschen zu unterdrücken. Das OAM-Verfahren startet mit einer leeren Gewichtsmatrix und hat keine *memories*. Wenn der Benutzer einen Erkennungsfehler korrigiert, wird die Distanz d_{min} von $\vec{V}$ und dem nächsten RBF-Zentrum berechnet. Wenn die Distanz d_{min} größer ist als der Schwellwert δ, dann wird eine neue RBF alloziert mit dem Vektor $\vec{V}$ als RBF-Zentrum. Der entsprechende Korrekturvektor wird gesetzt, um den Fehler mit der Schrittweite a zu korrigieren:

$$C_{ik} = a(T_i - O_i). \tag{2.21}$$

Wenn die Distanz d_{min} kleiner ist als δ, werden keine neue memories alloziert: der Korrekturvektor der zu dem Vektor $\vec{V}$ am nächsten liegenden RBF wird

angepasst, um den Fehler mit der Schrittweite b zu korrigieren.

$$\triangle C_{ik} = b(T_i - O_i)\Phi_k(\vec{V}). \tag{2.22}$$

Für die durchzuführenden Experimente werden die Parameter auf folgende Werte gesetzt: $a = 0.25$, $b = 0.2$. Diese Werte sollen beide kleiner 1 sein, um stabiles Lernen zu gewährleisten. Für δ werden in dieser Arbeit andere Werte als in [CP97] angegeben ermittelt und verwendet. In [CP97] wird $\delta = 0.1$ gesetzt, in dieser Arbeit ist dieser Wert höher und wird in den entsprechenden späteren Kapiteln angegeben. Wie oben schon erwähnt, die Anzahl der RBFs wächst sublinear zu der Anzahl der Lernbeispiele, da die RBFs nur bei den Fehlern alloziert werden, die zuvor noch nicht aufgetreten sind. Für die Fehler, die schon gesehen wurden, passt der Algorithmus den Korrekturvektor unter Verwendung der Updateregel 2.22 an. Der verwendete Adaptionsalgorithmus ist in pseudo-code unten angegeben:

For every character shown to the network {
  If the user indicates an error {
    $\vec{T}$ = target vector of the correct character
    $\vec{Q}$ = target vector of the highest element in $\vec{V}$
    $d_{min} = \min(\min_j d(\vec{V}, \vec{M}_j), d(\vec{V}, \vec{Q}))$
    If $d_{min} > \delta$ { // allocate a new memory
      k = index of the new memory
      $C_{ik} = a(T_i - O_i)$
      $\vec{M}_k = \vec{V}$
      $R_k = d_{min}$
    }
    else if memories exist and $(\min_j d(\vec{V}, \vec{M}_j)) < d(\vec{V}, \vec{Q}))$ {
      $k = \text{argmin}_j\, d(\vec{V}, \vec{M}_j))$
      $C_{ik} = C_{ik} + b(T_i - O_i)\Phi_k(\vec{V})$
    }
  }
}

Bei der Berechnung der nächsten RBF wird ein zusätzlicher Vektor $\vec{Q}$ verwendet, der die gleichen Werte wie der Zielvektor $\vec{T}$ als Koordinatenwerte hat (einmal $0,9$ und $0,1$ sonst), mit dem Unterschied, dass der maximale Wert $(0,9)$ dem höchstem Wert des Ausgabevektors $\vec{V}$ entspricht. Dieser Vektor wird *phantom Vektor* genannt und soll das Hinzufügen neuer RBFs verhindern, wenn die Ausgabe des Netzes eindeutig ist. Er dient dazu die Ausgabe bei korrekt geschriebenen Zeichen unverändert zu lassen.

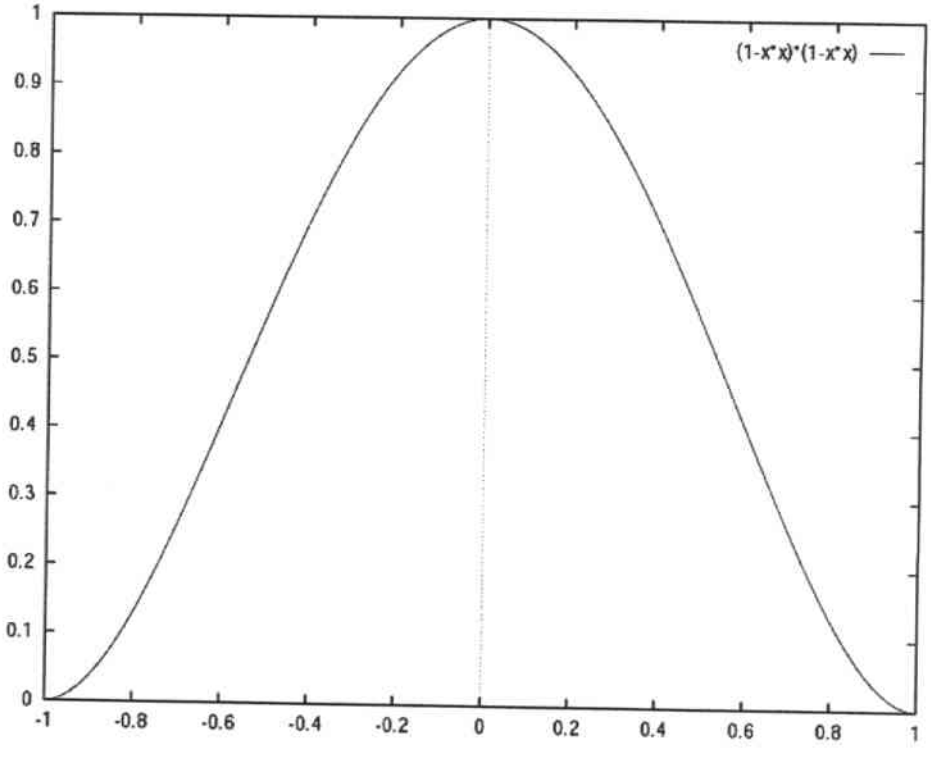

Abbildung 3: Die Glockenfunktion f

3 Handschrifterkenner: Systemüberblick

Dieses Kapitel beschreibt die grundlegenden Prinzipien und gibt einen Überblick über das Erkennungssystem das in dieser Arbeit verwendet wird. Es handelt sich um einen On-line Erkenner für Einzelzeichen und Wörter, der in seiner Architektur dem klassischen hierarchischen Aufbau eines Handschrifterkenners entspricht. Dieser Aufbau ist in vielen Off-line und On-line Erkennungssystemen wiederzufinden (siehe Abbildung 4).

Das ursprüngliche Eingabesignal, die zeitlich geordnete Folge der Datenpunkte $\{(x(t), y(t), p(t))\}_{t\in\{1\ldots T\}}$ mit $p(t) \in \{0, 1\}$ (Stiftzustand), durchläuft während des Erkennungsprozesses mehrere Verarbeitungsstufen, bestehend aus Vorverarbeitung und eigentlichen Erkennung. Die unterschiedlichen bei der Handschrifterkennung auftretenden Probleme werden jeweils durch entsprechenden Entwurf einer oder mehrerer der Verarbeitungsstufen gelöst. Unerwünschte Variabilität des Eingabesignals, verursacht durch die Hardware oder den Schreiber selbst, wird soweit wie möglich in der Vorverarbeitung reduziert. Robustheit gegenüber den verschiedenen Schreibstilen wird durch die Realisierung der Erkennungskomponente mittels eines neuronalen Netzes erreicht. Für die Toleranz gegenüber fehlerhaften Eingaben mit degenerierten oder fehlenden Zeichen sorgt die vokabulargesteuerte Baumsuche mit entsprechendem Wissen über die zugrundelegende Sprache. Nach der Vorver-

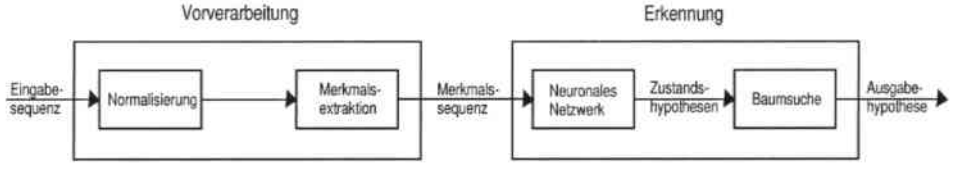

Abbildung 4: Die Verarbeitungsstufen des Systems

arbeitung wird jeweils ein zeitlicher Ausschnitt $x_{t_1}^{t_2}$ der Eingabe betrachtet und für alle Zeichenklassen c_i mittels eines neuronalen Netzes die a posteriori Wahrscheinlichkeit $P(c_i|x_{t_1}^{t_2})$ geschätzt, dass das jeweilige Zeichen bzw. ein Teil davon in diesem Ausschnitt beobachtet wird. Ist die Berechnung der Wahrscheinlichkeit für diesen Ausschnitt beendet, wird der nachfolgende Ausschnitt $x_{t_1}^{t_2}$ entsprechend betrachtet, so dass für jede Ausgabeklasse des neuronalen Netzes eine zeitliche Sequenz von a posteriori Wahrscheinlichkeiten ergibt. Das hier verwendete neuronale Netz ist ein Multi-State Time Delay Neural Network (MS-TDNN, siehe Abbildung 5), ein mehrschichtiges Netz mit lokalen Verbindungen und gemeinsamen Gewichten über die Zeit [SU94, SA95]. MS-TDNNs, ursprünglich für die Spracherkennungsauf-

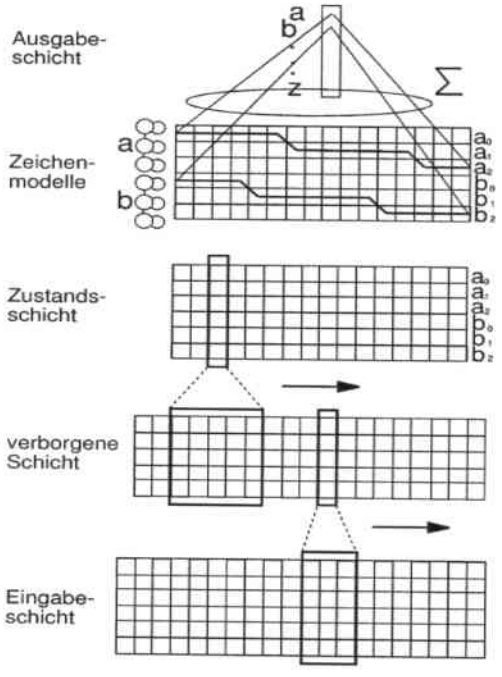

Abbildung 5: Die Architektur des MS-TDNN

gaben entwickelt, haben sich als sehr effektiv bei der Verarbeitung sequentieller Signale erwiesen. Das in dieser Arbeit verwendete MS-TDNN wird mit Backpropagation Algorithmus trainiert. Es besteht aus drei Schichten: der Eingabeschicht, der verborgenen Schicht und der Zustandsschicht. Als Eingabe werden nur lokale Merkmale verwendet. Die Länge des Eingabevektors ist 6 und die Anzahl der Zeitpunkte (Frames) ist variabel. Die Anzahl der verborgenen Einheiten (hidden units) in der verborgenen Schicht ist 130 Neuronen für das erkennende System für Klein- und Großbuchstaben. Die Ausgabelänge für dasselbe erkennende System auf der Phonemebene (Zustandsschicht) beträgt 146. Die Buchstabenmodellierung benutzt das einfache Linear-Modell mit variablen Anzahl der Zustände. Diese Anzahl der Zustände bei der Buchstabenmodelierung hängt von der Komplexität des Buchstaben ab, einmal festgelegt werden sie im Verlauf der gesamten Versuchsreihe nicht mehr geändert. Dabei haben die Buchstaben mit ähnlicher Erscheinung bei Groß- und Kleinschreibung gleiche Modelle, sind aber explizit in der DTW-Ebene berücksichtigt. Das hier beschriebene neuronale Netz wird sowohl für die Adaptionsversuche mit OAM als auch für das Nachtrainieren und lp-Adaption bei der Suche verwendet. Für

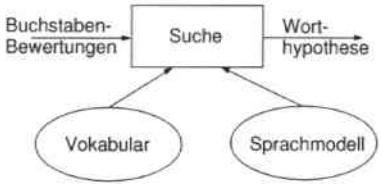

Abbildung 6: Möglichkeiten der Integration syntaktischen Wissens

einige der Adaptionsverfahren werden auch Versuche auf Wortebene durchgeführt, wobei Wörterbücher mit drei verschiedenen Größen benutzt werden. Die Wörterbücher der Größen 10 000, 20 000 und 65 000 Wörter enthalten Wörter der deutschen Sprache. Die Suche mit Vokabularen wird direkt in den Suchprozeß integriert, gegebenenfalls wird ein Sprachmodell auf der Zeichenebene verwendet (siehe die Abbildung 6). In dem Sprachmodell werden ausschließlich nur Buchstabentrigramme verwendet. Die Suche ist so gestaltet, dass die Verwendung von Vokabularen und dem Sprachmodell optional geschaltet werden kann. Die Versuche auf Wortebene werden sowohl für die Druckschrift als auch für kursive Handschrift durchgeführt. Die Versuche mit OAM werden auf zwei Ebenen durchgeführt. Im ersten Teil der Versuche mit OAM adaptiert ein schreiberunabhängiges System auf der DTW-Ebene und als Eingabe von OAM werden die Pfadwerte der unterschiedlichen Zeichen verwendet. Im zweiten Teil wird OAM auf der Phonem-Ebene (Zustandsschicht) eingesetzt. Mit diesem Einsatz soll die Klassifikation innerhalb eines Zeichens bezüglich seiner Zustände verbessert werden.

4 Daten

4.1 Schreiberabhängige Daten

Für die Durchführung der Adaptionsversuche dieser Arbeit wurden schreiberabhängige Daten gesammelt. Das Ziel dieser Sammlung war, zum einen ausreichend Schreiberabhängige Daten mit ausgewogener Buchstabenstatistik und zum anderen Daten mit sprachspezifischer Buchstabenhäufigkeit bereitzustellen. Zusätzlich sollten diese Daten Wörter und Sätze in Druckschrift und kursiven Schreibweise enthalten. Bei den gesammelten Sätzen sollten auch einige festgelegte Satzzeichen und Sonderzeichen enthalten sein. Dazu wurde eine Einteilung des Referenztextes in verschiedene Blöcke vorgenommen. Diese Einteilung sollte später ermöglichen, Versuche mit verschieden Teilmengen der Datensammlung zu definieren. Es wurde auch darauf geachtet, dass Spender unterschiedlichen Alters und Geschlechts an der Datenspende teilnahmen und somit möglichst weitgestreute Daten bezüglich der Schriftart und Schreibqualität ablieferten. Die unten aufgeführte Liste gibt einen Einblick in die Struktur der gesammelten Daten:

I. **Druckschrift (PRINTED)**

(1) Einzelbuchstaben: Druckbuchstaben in Kästchen (boxed fields) gleich für alle Spender. Zwei mal: 0-9, A-Z, a-z.

(2) Einzelne Wörter:

a. mit ausgewogener Statistik (alle Buchstaben mindestens 2 mal)
40 Wörter in boxed fields, gleich für alle Spender
40 Wörter in plain fields, gleich für alle Spender

b. zufällige Worte
40 Wörter in boxed fields, unterschiedlich für alle Spender, davon 10 Wörter nur Großbuchstaben
50 Wörter in plain fields, unterschiedlich für alle Spender, davon 20 in Großbuchstaben

(3) Sätze:

a. kurze Sätze ohne Satzzeichen (ausgewogene Statistik, jeder Buchstabe mindestens ein mal)
44 Sätze in plain fields, gleich für alle Spender, davon 10 Sätze mit Sonderzeichen (.-""?!;:$&%)

b. lange Sätze
10 Sätze mit Satzzeichen und mehrzeiligen Feldern, unterschiedlich für alle Spender

II. **Kursivschrift (CURSIV)** wie in I aber

(1) Einzelbuchstaben: entfällt

(2) Einzelne Wörter: wie printed, boxed fields werden in plain filed umgewandelt, keine Wörter in Großbuchstaben

(3) Sätze: statt 10 langen Sätzen, für jeden Spender unterschiedlich, 15 Sätze benutzen

Diese Datensammlung beinhaltet Daten der 17 Spender, 12 davon sind männlich und der Rest weiblich. Alle Spender waren Rechtshänder im Alter zwischen 21 und 41 Jahren, wobei die meisten junger als 30 Jahre waren. Insgesamt wurden fast 6000 Zeichen pro Schreiber gesammelt und die Dauer der Spende lag zwischen zwei und drei Stunden pro Spender.

Für die Datensammlung wurde das Wacom PL-400 Grafiktablett mit integriertem LCD-Bildschirm benutzt. Die genauere Beschreibung findet man im Kapitel 4.2.1. Dieses Tablett ist ein 13-Zoll beschreibbarer Bildschirm, der an einen gewöhnlichen PC angeschlossen wird. Speziell für die Datensammlung wurde ein Programm entwickelt, das dem Schreiber die notwendige Information und Funktionalität zur Verfügung stellt: die Felder zum Schreiben, den Referenztext, die Hinweise zur Schreibschrift, die Möglichkeiten zur Speicherung und der Verbesserung usw. Der Benutzer kann das Tablett waagerecht auf dem Tisch positionieren und mit einem Spezialstift auf dem Tablett schreiben. Bei der Datensammlung wurde der Schreiber aufgefordert, in dafür vorgesehenen weißmarkierten Flächen zu schreiben, dabei stand der Referenztext, das in dieser Fläche geschrieben werden sollte, unmittelbar über der weißen Fläche. Der Schreiber sollte darauf achten, dass er den Referenztext fehlerfrei niederschreibt in seinem gewohnten Schreibstil. Die auftretenden Fehler konnte man verbessern indem man die ganze weiße Fläche gelöscht hat und den Text erneut geschrieben hat. Für die Korrektur der vom Schreiber übersehenen Fehler konnte man bei der Aufbereitung der gesammelten Daten den Referenztext entsprechend dem geschriebenen Text anpassen.

4.2 Schreiberunabhängige Daten

Datenbasen mit entsprechenden On-line Handschriftdaten bilden eine der wichtigsten Einheiten eines schrifterkennenden Systems. Anhand dieser Daten werden spätere Vergleiche und Bewertungen des Handschrifterkenners

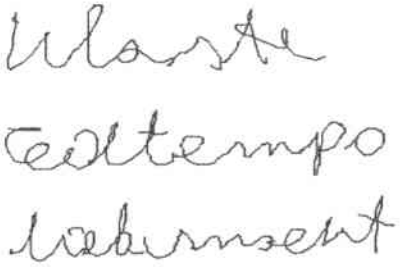

Abbildung 7: Beispiele für die gesammelten Daten, Kursivschrift: klarste, Eiltempo, Liebesnacht

durchgeführt. In den letzten Jahren wurde mehr Aufmerksamkeit der Verfügbarkeit standardisierter Datenbasen mit On-line Daten gewidmet, die es ermöglichen sollten, verschiedene On-line Handschrifterkenner objektiv miteinander vergleichen zu können. In den Versuchen dieser Arbeit werden mitunter auch solche standardisierten Datenbasen verwendet (UNIPEN), um die Basiserkennungsleistung des allgemeinen und des adaptierten Erkenners zu messen. Aufgrund der Verwendung der Datenbasen aus verschiedenen Quellen, die für die Versuche in dieser Arbeit zur Verfügung standen, ist ein direkter Vergleich mit anderen On-line Handschrifterkennern nur bedingt möglich, erlaubt aber dennoch eine fundierte Aussage über die Leistungsfähigkeit des eingesetzten Systems. Wegen der unterschiedlichen Herkunft und eingesetzten Aufnahme-Hardware repräsentieren diese Datenbasen sowohl nationale Besonderheiten als auch unterschiedliche technische Gegebenheiten. An ihnen können also alle wichtigen Eigenschaften des Systems, wie Schreiberunabhängigkeit, Hardware-Unabhängigkeit usw. überprüft werden. Jede der drei ersten Datenbasen (UKA, UNIPEN, IRONOFF) bestehen jeweils aus Einzelzeichen und Einzelworten, die zwei letzten (smiInkE, smiInkF) enthalten lediglich nur Einzelzeichen. In der Tabelle 1 sind nur Daten aus den oben erwähnten Datenbasen aufgeführt, die bei den Adaptionsversuchen dieser Arbeit verwendet werden.

Die Datenbasen sind schreiberdisjunkt in je eine Trainings-, Kreuzvalidierungs- und Testmenge unterteilt, wobei die Unterteilung in Buchstaben und Wörter jeweils getrennt und unabhängig voneinander erfolgte. Die folgenden Abschnitte beschreiben die für die Datensammlung verwendete Aufnahme-Hardware und stellen die fünf Datenbasen und ihre Aufteilung in Trainigs-, Validierungs- und Testmenge im Detail vor.

	UKA	UNIPEN	IRONOFF	smiInkE	smiInkF	$\sum$
A ... Z						
# Schreiber	73	1302	412	87	83	1957
# Zeichen	3446	25500	10666	2719	992	43323
a ... z						
# Schreiber	76	1638	412	81	81	2288
# Zeichen	66086	57055	10497	2729	4798	141165
Wörter:						
# Schreiber	336	1870	-	-	-	2206
# Zeichen	26181	23966	-	-	-	50147

Tabelle 1: Die Datenbasen im Überblick

4.2.1 Aufnahme-Hardware

Die Sammlung der in dieser Arbeit verwendeten Datenbasen wurde mit Hilfe mehrerer unterschiedlicher, kommerziell verfügbarer Digitalisiertabletts durchgeführt, die hier kurz beschrieben werden.

Das **Wacom PL-400** ist ein Grafiktablett mit 13-Zoll Farb-LCD-Bildschirm und erlaubt das "direkte" Zeichnen auf die Bildschirmfläche, was dem natürlichen Umgang mit Zeichenstift und Papier sehr nahe kommen soll. Die Bildschirmauflösung beträgt 1024 x 768 Pixel bei 16 bit Farbtiefe. Ein einstellbarer Fuß erlaubt flache Positionen für Schreiben und Zeichnen und eine nahezu aufrechte Position für das Betrachten des Bildschirminhaltes. Die Abtastrate beträgt bis zu 205 Punkten pro Sekunde, die Auflösung 508 dpi, die Aufnahme ist elektromagnetisch und drucksensetiv. Dieses Tablett wurde für die Erstellung der schreiberabhängigen Datenbasis verwendet.

Das **Wacom SD-421E** ist ein DIN-A4 Grafiktablett mit elektromagnetischer Oberfläche, auf der bei Bedarf ein Blatt Papier fixiert werden kann. Die Eingabe erfolgt mit einem kabellosen Stift entweder direkt auf der glatten Oberfläche des Tabletts, oder auf dem aufgelegten Papier. Die Abtastrate kann von 2 bis 200 Punkten pro Sekunde variiert werden. Dieses Tablett wurde für die UKA Datenbasis verwendet, wobei die Sammlung teils mit aufgelegtem Papier erfolgte teils ohne.

Mit dem Tablett **Wacom Ultrapad A4** mit der Abtastrate von 100 Punkten pro Sekunde und der Auflösung von 300 dpi wurde die Datenbasis IRONOFF erzeugt. Die Sammlung erfolgte ausschließlich mit aufgelegtem Papier.

Das **Wacom PL-300** ist das Vorgängermodell von **Wacom PL-400** mit kleinerem 10.4 inch LCD-Monitor und einer geringeren Auflösung von 800x600 Pixel, drucksensetiv, die Abtastrate beträgt bis zu 205 Punkten

UKA (Einzelzeichen)		
A ... Z	# Schreiber	# Zeichen
Training	53	2349
Validierung	19	165
Test	1	932
Summe	73	3446
a ... z	# Schreiber	# Zeichen
Training	58	55619
Validierung	16	5342
Test	2	5125
Summe	76	66086

Tabelle 2: Die Unterteilung der Einzelzeichen der UKA Datenbasis

pro Sekunde. Mit diesem Grafiktablett wurden die Datenbasen smiInkE und smiInkF gesammelt. Die Sammlung erfolgte direkt auf der Schreiboberfläche.

Die Datenbasis UNIPEN wurde mit verschiedenen Eingabegeräten durchgeführt. Bevorzugt wurde dabei das **Wacom PL 100-V** Grafiktablett mit VGA 640x480 16 Graustufen LCD-Monitor und der Auflösung von 205 Punkten pro Sekunde für das Tablett.

4.2.2 Die UKA Datenbasis

Die UKA Datenbasis wurde gezielt für das Erkennungssystem der Universität Karlsruhe gesammelt. Sie bestand ursprünglich nur aus Einzelzeichen und wurde im Laufe der Zeit um Einzelworte erweitert. Als Schreiber stellten sich Studenten und Mitarbeiter der Universität zur Verfügung, außerdem wurden auch die Sonderveranstaltungen der Universität für Abiturienten zur Datensammlung genutzt.

Jeder Schreiber war aufgefordert eine Datenmenge zu spenden, die entweder jeweils nur Einzelzeichen oder Einzelworte umfasste oder eine Mischung aus beidem. Die Schreiber wurden gebeten so natürlich wie möglich auf dem Grafiktablett zu schreiben und es wurden keine Vorgaben über den Schreibstil gemacht. Die Datenbasis weist folglich eine zufällige Verteilung von Schreibern mit Druck- und Kursivschrift auf.

Für die Durchführung der Experimente in dieser Arbeit werden die Einzelzeichen und Wörter der UKA Datenbasis benutzt. Die Tabellen 2 und 3 geben die Aufteilung der Daten wieder.

UKA (Wörter)		
CURSIVE	# Schreiber	# Wörter
Training	694	13894
Validierung	86	1881
Test	80	1641
Summe	860	17416

Tabelle 3: Die Unterteilung der Einzelworte der UKA Datenbasis

UNIPEN (Einzelzeichen)		
A ... Z	# Schreiber	# Zeichen
Training	1042	20734
Validierung	130	2397
Test	130	2369
Summe	1302	25500
a ... z	# Schreiber	# Zeichen
Training	1310	46440
Validierung	164	5340
Test	164	5275
Summe	1638	57055

Tabelle 4: Die Unterteilung der Einzelzeichen der UNIPEN Datenbasis

4.2.3 Die UNIPEN Datenbasis

Die UNIPEN Datenbasis ist im Rahmen des UNIPEN Projekts [I.G94] auf Initiative des Technical Commitee 11 der International Association for Pattern Recognition (IAPR) entstanden, um dem Mangel der standardisierten Datenbasis und dem Fehlen internationaler Vergleiche entgegenzuwirken. Diesem Projekt haben sich über 40 internationale Teilnehmer aus der Forschung und Industrie angeschlossen. Jeder Teilnehmer musste als Beitrittsbedingung eine festgelegte Mindestmenge (min. 12 000 Zeichen) On-line Handschriftdaten sammeln und für das Projekt frei zur Verfügung stellen. Von National Institute of Standards and Technology (NIST) wird die Datenbasis in 4 disjunkte Mengen unterteilt (Trainingsmenge, Kreuzvalidierungsmenge und 2 Testmengen), um internationale Vergleiche zu ermöglichen. Die Trainings- und Kreuzvalidierungsmenge werden allen Teilnehmern gleichermaßen zur Entwicklung der Erkenner zur Verfügung gestellt. Die beiden Testmengen werden unter Verschluss gehalten, um internationale Vergleiche zu organisieren. Erst durch diesen Vergleich wird eine zuverlässige Aussage über den derzeitigen Leistungsstand heutiger On-line Handschrifterkennungsysteme möglich.

UNIPEN (Wörter)		
PRINTED	# Schreiber	# Wörter
Training	808	5424
Validierung	101	586
Test	101	540
Summe	1010	6550
CURSIVE	# Schreiber	# Wörter
Training	694	13894
Validierung	86	1881
Test	80	1641
Summe	860	17416

Tabelle 5: Die Unterteilung der Einzelworte der UNIPEN Datenbasis

Die bei SMI zur Verfügung stehenden UNIPEN-Daten werden während dieser Arbeit selbständig in Trainings- Kreuzvalidierungs- und Testmengen eingeteilt und für die Adaptionsversuche benutzt. Die Aufteilung der Daten ist in den Tabellen 4 und 5 angegeben.

4.2.4 Die IRONOFF Datenbasis

Die IRESTE On/Off (IRONOFF) Dual Handwriting Database enthält eine große Anzahl an Einzelzeichen, Ziffern und kursiven Wörtern, die von der Hand französischer Schreiber entstanden ist. Diese Datenbasis enthält beide Varianten der Handschriftdaten die On-line und die Off-line. Die Off-line Daten werden aus den On-line Daten durch affine Transformation der On-line Daten in das Koordinatensystem des Scanners gewonnen [CG]. Diese Datenbasis entspricht dem UNIPEN-Format und wird ständig erweitert. Die Aufteilung der Daten, die in dieser Arbeit verwendet werden, ist in den Tabelle 6 angegeben.

4.2.5 Die smiInkE und smiInkF Datenbasen

Die Firma SMI hat On-line Daten für eigene Entwicklungen gesammelt. Als Ergebnis sind die Datenbasen smiInkE und smiInkF entstanden. Die Datenbasis smiInkE enthält Einzelzeichen und Wörter in Druckhandschrift, die Datenbasis smiInkF ist vom Inhalt sehr ähnlich und ist im Zusammenhang mit der Entwicklung eines schrifterkennenden Systems für Formulare entstanden. Die Aufteilung der Daten ist in den Tabellen 7 und 8 angegeben.

IRONOFF (Einzelzeichen)		
A ... Z	# Schreiber	# Zeichen
Training	330	8544
Validierung	41	1061
Test	41	1061
Summe	412	10666
a ... z	# Schreiber	# Zeichen
Training	330	8414
Validierung	41	1037
Test	41	1046
Summe	412	10497

Tabelle 6: Die Unterteilung der Einzelzeichen der IRONOFF Datenbasis

smiInkE (Einzelzeichen)		
A ... Z	# Schreiber	# Zeichen
Training	57	1770
Validierung	10	316
Test	20	633
Summe	87	2719
a ... z	# Schreiber	# Zeichen
Training	51	1715
Validierung	10	338
Test	20	676
Summe	81	2729

Tabelle 7: Die Unterteilung der Einzelzeichen der smiInkE Datenbasis

smiInkF (Einzelzeichen)		
A ... Z	# Schreiber	# Zeichen
Training	53	548
Validierung	10	210
Test	20	234
Summe	83	992
a ... z	# Schreiber	# Zeichen
Training	61	3671
Validierung	10	596
Test	10	531
Summe	81	4798

Tabelle 8: Die Unterteilung der Einzelzeichen der smiInkF Datenbasis

4.3 Leistungsmessung des Systems

Die Vergleiche der Adaptionsverfahren untereinander und mit dem Basiserkennungssystem erfordern die Berechnung der Erkennungsleistung für die zwei unterschiedlichen Aufgaben: Einzelzeichen- und Worterkennung. In dem folgenden Kapitel 5 an vielen Verschiedenen Stellen sind nach diesen zwei Bereichen getrennt Erkennungsergebnisse angegeben. Sofern nicht anderes angegeben, sind diese Ergebnisse für alle Versuche unter den gleichen Bedingungen erzeugt worden. Das Training, die Kreuzvalidierung und das Testen der Leistungsfähigkeit des schreiberunabhängigen Systems und des adaptierten Systems erfolgt über alle fünf verfügbaren Datenbasen hinweg, d.h. die zu einer Erkennungsaufgabe gehörenden Trainings-, Kreuzvalidierungs- und Testmengen der Datenbasen werden zu entsprechenden gemeinsamen Mengen zusammengefasst. Die Erkennungsleistung auf den Testmengen wird als Gesamterkennungsrate angegeben. Für die Messungen der Erkennungsleistung auf der Wortebene wird die Unterteilung nach Schreibstielen Druck- (PRINTED) und Kursiv- (CURSIVE) vorgenommen.

Die Erkennungsrate auf Einzelzeichen und Worten wird direkt als Buchstaben- bzw. Worterkennungsrate angegeben,

$$\text{BE (bzw. WE)} = \frac{N_K}{N_G}$$

wobei N_K die Anzahl der von System korrekt klassifizierten und N_G die Gesamtanzahl der getesteten Zeichen bzw. Wörter angibt. Bei der Worterkennung wird ein Wort als falsch gezählt, wenn ein einziges Zeichen dieses Wortes als falsch erkannt wurde.

Bei der Erkennung auf Wortebene, treten wegen Segmentierungsfehlern zusätzlich zu den Buchstabenverwechslungen noch Einfügungen und Auslassungen als Fehlerquelle auf. Aus diesem Grund wird, wie in der kontinuierlichen Spracherkennung üblich, bei Wörtern anstelle der Buchstabenerkennungsrate die Buchstabenakkuratheit

$$\text{BA} = \frac{N_G - N_E - N_A - N_V}{N_G}$$

als Erkennungsleistung für ein Wort angegeben.

4.4 Erkennungsergebnisse des Systems auf allgemeinen Daten

In den nachfolgenden Kapiteln dieser Arbeit werden an zahlreichen Stellen Ergebnisse von unterschiedlichen Adaptionsversuchen präsentiert. Da die

Gesamt über alle Datenbasen (Einzelzeichen)	
Großbuchstaben	93.6%
Groß- und Kleinbuchstaben	93.3%

Tabelle 9: Die besten Ergebnisse des Systems für die Einzelzeichenerkennung

Einzelwörter UNIPEN PRI	BA	WA
ohne Vokabular	94.3%	82.4%
mit Vokabular 10 000	97.8%	95.4%
mit Vokabular 20 000	97.6%	95.2%
mit Vokabular 65 000	96.7%	93.0%
Einzelwörter UNIPEN CUR	BA	WA
ohne Vokabular	87.3%	-
mit Vokabular 10 000	92.8%	87.6%
mit Vokabular 20 000	92.2%	86.2%
mit Vokabular 65 000	90.0%	83.2%
Einzelwörter UKA CUR	BA	WA
mit Vokabular 10 000	91.4%	91.2%
mit Vokabular 20 000	90.0%	89.6%
mit Vokabular 65 000	87.3%	87.0%

Tabelle 10: Erkennungsleistung des Systems auf Einzelwörtern

On-line Einzelzeichenerkennung für die Adaptionsversuche von großer praktischer Bedeutung ist, wird das System, das bei Einzelzeichenerkennung das beste Ergebnis erzielt als Basissystem genommen und durchweg bei allen Versuchen beibehalten. Die Tabellen 9, 10 fassen die besten Ergebnisse für die wichtigsten Erkennungsaufgaben zusammen. Diese Ergebnisse werden in den nachfolgenden Kapiteln jeweils zum Vergleich herangezogen, um eine Bewertung einzelner Adaptionsversuche zu ermöglichen.

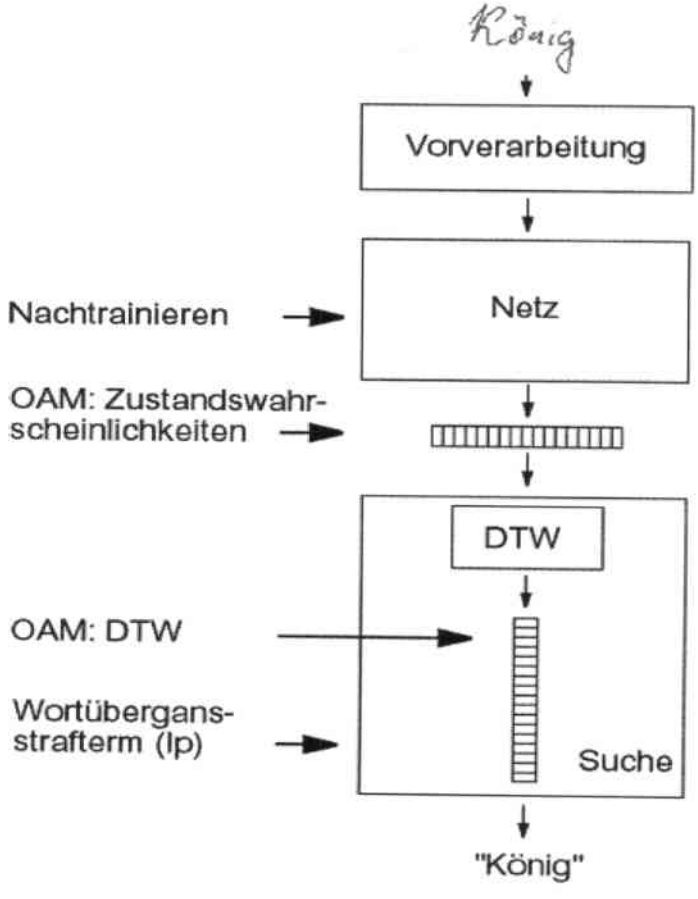

Abbildung 8: verschiedene Ebenen der Integration der Adaptionsverfahren

5 Adaptionsversuche

5.1 Überblick

Die Adaptionsversuche, die in dieser Arbeit durchgeführt werden, unterteilen sich in verschiedene Kategorien, die von dem Einsatz des Adaptionsverfahrens im erkennenden System abhängig sind (siehe Kapitel 3). In den nächsten Unterkapiteln werden Versuche beschrieben, die eine Adaption im neuronalen Netz und bei der Suche vornehmen. Selbstverständlich sind Adaptionsansätze auch bei der Vorverarbeitung denkbar, jedoch in dieser Arbeit werden solche Versuche nicht durchgeführt. Der Vorteil der hier beschriebenen Verfahren ist ihre Unabhängigkeit von der Vorverarbeitung, die bei mehreren Handschrifterkennern unterschiedlich ausfallen kann und somit nur für ein spezi-

elles System verwendbar wäre. Die hier beschriebenen Adaptionsverfahren sind weitgehend universell und bei allen netzbasierten Systemen einsetzbar. Die Adaptionsverfahren auf der Systemebene des neuronalen Netzes werden zwei Adaptionsverfahren untersucht: das OAM-Verfahren und das "einfache" Nachtrainieren (siehe dazu Kapitel 2). Beide Verfahren können als Adaptionsverfahren eingesetzt werden, wobei die Auswahl wird durch die jeweiligen Voraussetzungen für die Adaption eines einzelnen Systems bestimmt. Die Menge der speziellen Daten, die Trainingsdauer, die Leistungsfähigkeit der Rechner auf denen der Erkenner eingesetzt und adaptiert werden soll, die Einbeziehung der Experten – das sind nur einige Beispiele der Faktoren, die berücksichtigt werden müssen bei der Wahl des Adaptionsverfahrens.

Der große Vorteil des Nachtrainierens ist, dass bei der Adaption die herkömmliche Trainingsmethode verwendet werden kann, ohne das zusätzlicher Aufwand bei der Anpassung der Systems erforderlich wäre. Dabei ist aber eine größere Menge an speziellen Daten erforderlich, die für das Adaptionstraining von Experten bearbeitet werden muss (Segmentierung etc.). Das inkrementelle Nachtrainieren ist in diesem Fall denkbar, aber durch die Adaption mit der Auslassung der Repräsentanten einzelner Klassen werden diese Klassen verlernt. Diese Tatsache würde die Effektivität der Adaption und die Akzeptanz bei den Anwendern vermindern. Das OAM-Verfahren eignet sich mehr für Systeme mit inkrementeller Adaption, die auf nicht besonders leistungsfähigen Rechnern (PDAs und ähnliches) eingesetzt werden und mit wenigen Adaptionsbeispielen auskommen müssen. Der Nachteil ist, dass es zusätzlich in das System integriert werden muss und nach bisherigen Ergebnissen dieser Arbeit (siehe folgende Kapitel) das Nachtrainieren nicht übertrifft.

Ein weiterer Adaptionsversuch wird auf der Ebene der Suche durchgeführt. Hier wird der Suchparameter lp (language penalty) jeweils für den einzelnen Schreiber optimal bestimmt, die adaptierten Gewichte geladen und anschließend auf Zeichenebene getestet.

5.2 Adaption des neuronalen Netzes durch Nachtrainieren

Nachtrainieren: Großbuchstaben

Das in diesem Kapitel beschriebene Experiment wird nur auf den Großbuchstaben durchgeführt. Es handelt sich um das Nachtrainieren und das Testen auf speziellen Daten, so wie auch um das anschließende Testen auf den allgemeinen Daten. Durch die Adaption ist eine deutliche Verbesserung festzustellen, wobei der adaptierte Erkenner auf allgemeinen Daten etwas schlechter

PRINTED, Großbuchstaben										
ID	Adaptionsdaten						allgemeine Daten			
	test %		cross %		train %		test %		cross %	
	base	adp	base	adp	base	adp	base	adp	base	adp
1	96.2	96.2	91.1	98.7	97.9	99.3	93.6	92.4	96.3	95.1
2	93.5	94.8	90.2	95.1	96.2	97.7	93.6	93.1	96.3	95.5
3	98.3	98.3	97.5	97.5	98.4	98.4	93.6	93.5	96.3	96.3
4	92.8	97.6	91.7	98.6	96.6	100	93.6	92.5	96.3	95.2
5	100	100	100	100	100	100	93.6	93.7	96.3	96.2
6	100	100	93.3	96.6	95.3	100	93.6	92.3	96.3	94.8
7	93.1	94.8	96.1	97.4	93.1	97.7	93.6	93.3	96.3	95.8
8	97.8	100	77.6	92.5	90.6	100	93.6	92.4	96.3	95.2
9	100	100	97.5	97.5	99.3	100	93.6	93.1	96.3	96.0
10	100	100	100	100	100	100	93.6	93.5	96.3	96.1
11	95.7	97.9	100	100	96.6	99.2	93.6	92.8	96.3	95.6
12	96.6	96.6	97.3	98.7	99.3	100	93.6	93.3	96.3	96.2
13	97.4	100	95.6	100	97.9	100	93.6	93.1	96.3	95.4
14	100	100	99.1	100	99.3	100	93.6	92.6	96.3	95.5
15	98.7	100	98.8	98.8	99.0	100	93.6	94.2	96.3	95.8
16	100	100	100	100	98.4	99.2	93.6	93.6	96.3	96.0
17	100	100	100	100	100	100	93.6	93.5	96.3	96.2
Ø	**97.7**	**98.6**	**95.6**	**98.3**	**97.5**	**99.5**	**93.6**	**93.1**	**96.3**	**95.7**

Tabelle 11: Die BA nach der Adaption mit dem Nachtrainieren auf Großbuchstaben für alle 17 Schreiber

wird (siehe die Tabelle 11). Aus den in den Tabellen dargestellten Daten lässt sich auch eine Bewertung des Schreibstils eines Schreibers im Vergleich untereinander herleiten, wobei bei diesem Versuch die Varianz noch nicht sehr stark ausfällt. Eine stärkere Varianz wird erst bei dem Versuch mit Groß- und Kleinbuchstaben beobachtet. Das Training für jeden Schreiber besteht aus 20 Iterationen, wobei als Trainingsergebnis wird das Ergebnis aus der Iteration mit der besten Erkennungsleistung auf der Kreuzvalidierungsmenge (cross) gewählt. Die Kreuzvalidierungsmenge unterscheidet sich sonst nicht von der Testmenge, die Aufteilung in Mengen wird zufällig vorgenommen.

Die Anzahl der Zeichen in der Trainingsmenge ist ungefähr 140 Zeichen, in der Test- und der Kreuzvalidierungsmenge jeweils etwa 70 Zeichen. In diesem Versuch ist die Anzahl der verborgenen Einheiten (hidden units) in der verborgenen Schicht 100 Neuronen und die Ausgabelänge auf der Phonem-Ebene (Zustandsschicht) beträgt 96. Im Rahmen dieses Adaptionsversuches,

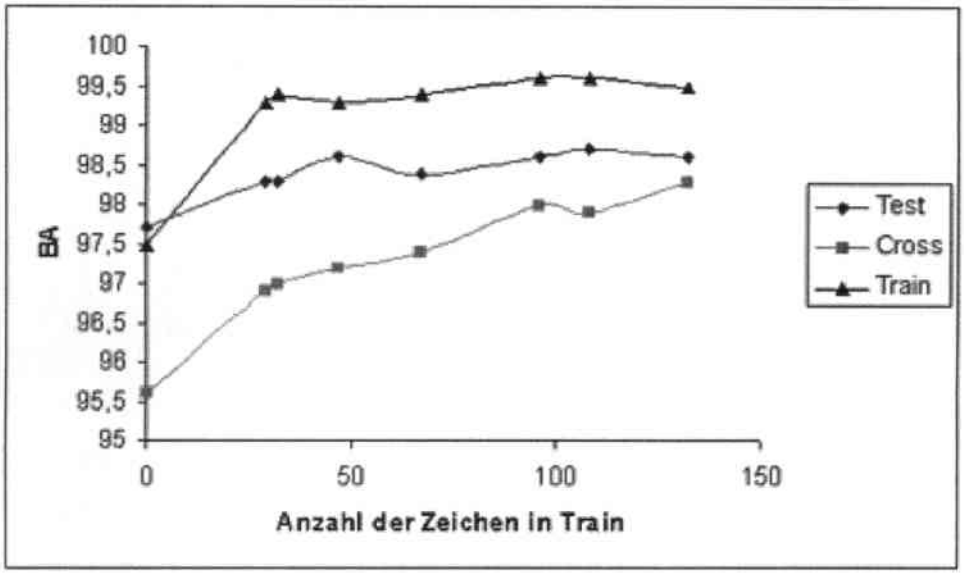

Abbildung 9: Die Adaptionsfähigkeit in Abhängigkeit von der Größe der Trainingsmenge, Nachtrainieren auf Großbuchstaben

wie auch in den meisten anderen Versuchen in dieser Arbeit, wird auch mit kleineren Trainingsmengen trainiert bei der gleichzeitigen Beibehaltung der Größen für die Test- und Kreuzvalidierungsmengen. Dies ermöglicht eine Aussage über die minimale Größe der Trainingsmenge, bei der ein Nachtrainieren noch zu einer Verbesserung der Erkennungsleistung auf den speziellen Daten führt. In diesem Versuch liegt die optimale Anzahl der Trainingsbeispiele etwa bei 100 Einzelzeichen (siehe Abbildung 11). Die Kenntnis der optimalen Menge der Einzelzeichen für die Trainingsmenge erlaubt die Reduktion der Aufwandskosten bei der Beschaffung und der Aufbereitung der speziellen Daten.

Nachtrainieren: Groß- und Kleinbuchstaben

Das Nachtrainieren auf Groß- und Kleinbuchstaben wird mit dem gleichen System wie beim Experiment aus dem vorherigen Kapitel durchgeführt, allerdings wird hier ein Netz mit 130 Neuronen für die verborgene Schicht und der Ausgabe von der Länge 146 verwendet. Dabei haben einige Groß- und Kleinbuchstaben das gleiche Modell z.B. C-c, Z-z, U-u usw. und werden in der Ausgabeschicht einfach geführt. Dieses Vorgehen ist damit begründet, dass die Klein- und Großschreibung dieser Zeichen sehr ähnlich ist und man versucht diese Ähnlichkeit auszunutzen indem man weniger Modelle trainie-

PRINTED, Groß- und Kleinbuchstaben										
ID	Adaptionsdaten						allgemeine Daten			
	test %		cross %		train %		test %		cross %	
	base	adp	base	adp	base	adp	base	adp	base	adp
1	85.2	93.9	86.9	94.6	88.1	98.4	93.3	90.3	92.2	87.1
2	79.9	93.5	75.4	90.3	77.8	96.8	93.3	88.2	92.2	85.9
3	91.6	96.6	88.6	96.0	89.7	98.0	93.3	91.3	92.2	88.6
4	77.7	91.6	82.3	95.3	83.3	98.3	93.3	89.7	92.2	87.4
5	95.1	98.7	95.5	98.6	95.8	99.8	93.3	92.1	92.2	90.6
6	81.5	94.2	80.4	95.0	81.2	97.8	93.3	90.7	92.2	87.6
7	72.3	91.4	74.0	93.0	73.6	97.6	93.3	88.8	92.2	86.1
8	83.9	96.0	87.5	95.5	85.3	98.3	93.3	90.4	92.2	88.7
9	88.2	97.2	89.6	94.6	88.0	99.3	93.3	91.0	92.2	88.4
10	89.7	92.7	94.2	96.5	93.4	97.4	93.3	93.0	92.2	91.1
11	94.7	97.5	94.5	97.4	93.5	97.0	93.3	92.0	92.2	90.4
12	86.2	93.5	86.7	95.8	86.2	97.7	93.3	92.7	92.2	90.7
13	86.3	92.8	88.5	96.2	87.2	96.6	93.3	91.5	92.2	90.7
14	83.9	95.6	90.2	97.1	85.7	98.1	93.3	90.1	92.2	86.8
15	93.1	97.1	90.8	96.9	91.1	98.4	93.3	90.9	92.2	88.8
16	90.8	94.1	94.7	95.5	90.4	96.9	93.3	90.3	92.2	89.0
17	91.3	98.3	93.9	98.6	92.3	99.0	93.3	91.7	92.2	89.5
Ø	**86.6**	**95.0**	**87.9**	**95.7**	**87.2**	**98.0**	**93.3**	**90.9**	**92.2**	**88.6**

Tabelle 12: Die BA nach der Adaption mit dem Nachtrainieren auf Groß- und Kleinbuchstaben für alle 17 Schreiber

ren und voneinander unterscheiden braucht. Die Klein- und Großschreibung wird erst bei der Erkennung auf der Wortebene, wo ein Vergleich der Buchstabengrößen möglich ist, in dem System vorgenommen.

In der Tabelle 12 sind die Adaptionsergebnisse für die Adaption auf Groß- und Kleinbuchstaben dargestellt. Im Vergleich zu den Ergebnissen aus der Tabelle 11 fällt die Erkennungsleistung geringer aus für die jeweiligen Schreiber im einzelnen sowie im Durchschnitt (siehe Tabelle 12). Dies ist durch die größere Verwechslungsmöglichkeit und größere Schreibstilvarianz für die gemeinsame Menge der Groß- und Kleinbuchstaben zu erklären. Anhand der Adaptionsergebnisse ist ersichtlich, dass die Schreiber mit der ID 2, 4 und 7 die am schwierigsten zu erkennenden Schreibstile besitzen. Die Schreiber mit den IDs 5, 11 und 15 haben dagegen einen sehr leicht zu erkennenden Schreibstil, so dass schon auf dem schreiberunabhängigen System eine gute Erkennungsleistung gemessen wird. Das Adaptionstraining von 20 Iteratio-

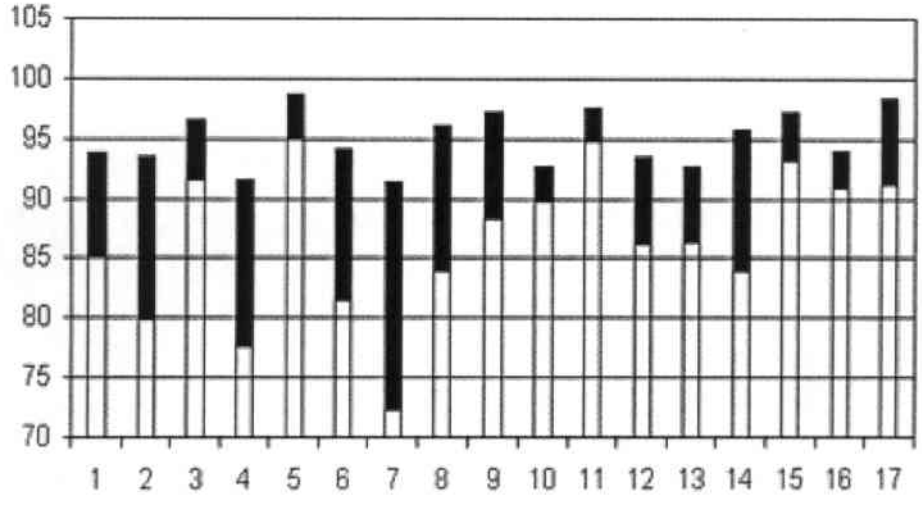

Abbildung 10: Die BA vor und nach der Adaption mit dem Nachtrainieren auf Groß- und Kleinbuchstaben, alle 17 Schreiber

nen braucht etwa 2 min auf der größten Trainingsmenge von 600 Zeichen. Bei der Durchführung der Adaptionsversuche wird ein Rechner mit dem Prozessor Pentium 4 mit der Taktfrequenz 2400 MHz benutzt.

Im allgemeinen ist aus den Ergebnissen der Tabelle 12 bezüglich der Verringerung der Fehlerrate ersichtlich, dass die stärkste Verringerung bei den Schreibern mit schwierigen Schreibstilen zu beobachten ist. Hier ist die Verringerung der Fehlerrate von $27,7\%$ auf $8,6\%$ durch die Adaption erreichbar (vergleiche Schreiber mit der ID 7). Bei den Schreibern mit der niedrigen Fehlerrate auf dem allgemeinem Erkenner fällt diese Verringerung viel schwächer aus (vergleiche Schreiber mit der ID 5). Diese Tatsache ist damit zu erklären, dass die Adaption erst bei einer größerer Menge an eigenartigen Buchstaben eines Schreibstils und somit auch größerer Menge an schreiberspezifischen Daten greift.

Die Abhängigkeit der Erkennungsleistung von der Größe der Trainingsmenge ist in der Abbildung 11 dargestellt. Im Vergleich mit der Abbildung 9 wird hier erwartungsgemäß eine größere Abhängigkeit festgestellt, da es in diesem Versuch mehr Zeichenklassen unterschieden werden müssen. Hier ist es denkbar mit größerer Trainingsmenge noch bessere Erkennungsleistung erzielen zu können, denn ein Overfittingeffekt ist noch nicht erkennbar.

Bei dem "einfachen" Nachtrainieren, bei dem alle Parameter (Gewichte)

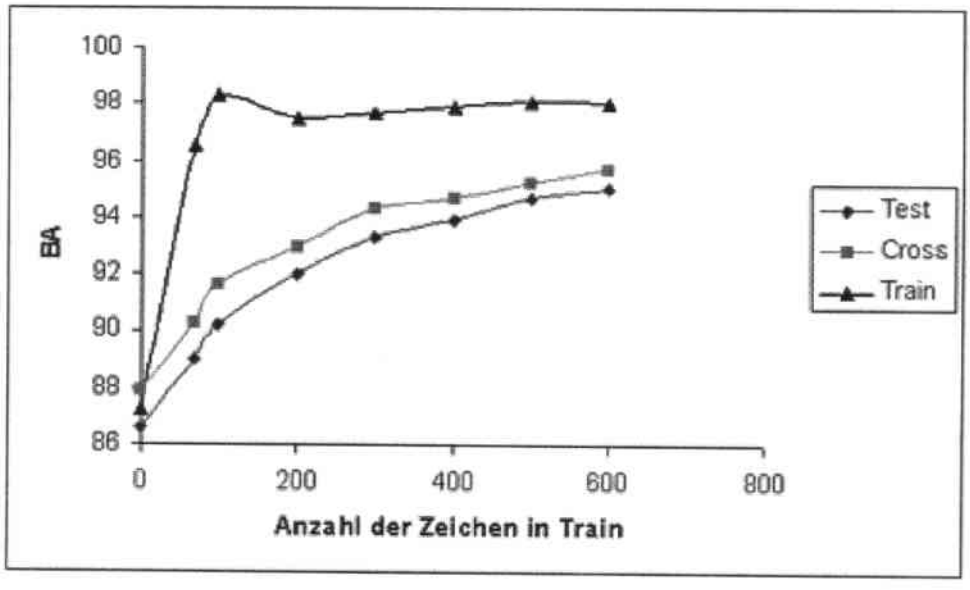

Abbildung 11: Die Adaptionsfähigkeit in Abhängigkeit von der Größe der Trainingsmenge, Nachtrainieren auf Groß- und Kleinbuchstaben

Verfahren	Adaptionsdaten		allgemeine Daten	
	base	adp	base	adp
ohne Einschränkung	86,6	95.0	93,3	90,9
mit Einschränkung	86,6	94.5	93,3	92,0

Tabelle 13: Die BA nach der Adaption mit dem Nachtrainieren auf Groß- und Kleinbuchstaben mit der Parametereinschränkung und ohne

angepasst werden, kann das Netz einige Zeichenklassen wieder verlernen, falls die Adaptionsmenge bezüglich aller zu erkennenden Klassen unvollständig ist. Um diesen Nachteil zu vermeiden, kann man nur einen Teil der Parameter für die Adaption freigeben und die restlichen "einfrieren". Damit wird dem oben beschriebenen Nachteil entgegengewirkt. Die mit so einem eingeschränktem System erzielten Ergebnisse sind in der Tabelle 13 angegeben. Aus dem Vergleich der Ergebnisse eines eingeschränkten und eines nicht eingeschränkten Systems wird ersichtlich, dass die Adaptionsfähigkeit eines eingeschränkten Systems sinkt, wobei die Erkennungsleistung auf allgemeinen Daten verschlechtert sich erwartungsgemäß im geringeren Maße.

PRINTED, freie Suche mit Sprachmodellen auf Buchstabenebene										
Akk	Adaptionsdaten						allgemeine Daten			
	test %		cross %		train %		test %		cross %	
	base	adp	base	adp	base	adp	base	adp	base	adp
BA	90.3	95.1	91.4	94.3	91.0	96.3	94.3	91.6	95.6	93.8
WA	63.3	79.0	65.2	76.9	65.4	84.4	82.4	74.3	86.4	81.2
PRINTED, Vokabulargröße: 10 000 Wörter										
BA	96.7	97.8	97.1	97.8	96.6	98.3	97.8	96.8	99.2	98.8
WA	94.1	96.6	94.8	95.8	94.8	97.5	95.4	92.9	97.8	96.8
PRINTED, Vokabulargröße: 20 000 Wörter										
BA	96.4	97.7	96.8	97.6	96.1	98.3	97.6	96.6	99.1	98.6
WA	93.2	96.2	93.7	95.6	94.0	97.5	95.2	92.3	97.4	96.0
PRINTED, Vokabulargröße: 65 000 Wörter										
BA	95.9	97.6	96.4	97.6	95.9	98.1	96.7	95.8	98.7	98.1
WA	91.7	95.4	92.5	95.2	92.9	97.0	93.0	90.4	96.4	94.6

Tabelle 14: Die Werte für BA und WA vor und nach der Adaption mit dem Nachtrainieren, über alle 17 Schreiber gemittelt

Das Testen auf Wortebene

Bei diesem Experiment wird das auf Zeichenebene adaptierte System auf der Wortebene getestet. Für die Erkennung der Wörter in Druckschrift werden die Wörter in Einzelzeichen unterteilt und die verzögerten Schreibzüge (delayed strokes) extra behandelt, indem man zusammengehörende strokes innerhalb eines Zeichens in richtigen zeitlichen Reihenfolge aufstellt. Die über alle 17 Schreiber gemittelten Adaptionsergebnisse werden in den Tabellen dieses Kapitels angegeben. Die Tabellen enthalten die Werte für die Erkennungsleistung als Buchstaben- und Wortakkuratheit und sind einer festen Vokabulargröße zugeordnet. Erwartungsgemäß steigt die Erkennungsleistung einer Erkennung mit Vokabular gegenüber einer mit freien Suche zunächst stark an, doch mit dem Anstieg der Vokabulargröße steigt auch die Verwechselbarkeit der Wörter und die Erkennungsleistung sinkt. Diese Gesetzmäßigkeit gilt offensichtlich auch für das adaptierte System.

Die hier präsentierten Erkennungsergebnisse auf der Wortebene mit Sprachmodellen (siehe Tabelle 14 oben) werden mit der Unterstützung von Trigrammen erzielt. Damit wird die schwächste Form einer Einschränkung der Suche benutzt. Die direkte Integration der Trigramme in den Suchprozeß erlaubt weiterhin die Erkennung beliebiger Zeichenfolgen. Das Vokabular kommt lediglich bei der Berechnung von Trigrammen, aber nicht mehr bei der Suche

CURSIV, Zeichenebene									
Adaptionsdaten						allgemeine Daten			
test %		cross %		train %		test %		cross %	
base	adp	base	adp	base	adp	base	adp	base	adp
76.3	83.3	68.2	78.1	64.4	85.3	87.3	80.4	82.7	76.6

Tabelle 15: Die Durchschnittswerte der BA vor und nach der Adaption mit dem Nachtrainieren über alle 17 Schreiber

selbst zum Einsatz. Die verwendeten Vokabulare sind direkt in die Suche integriert und schränken bereits während des Suchvorgangs die Anzahl der möglichen Zeichenübergänge ein. Die Einschränkung erfolgt über die Baumsuche auf der Zeichenebene (siehe dazu [Man]).

Bei der Bewertung der Basiserkennungsleistung auf den allgemeinen Daten vor und nach der Adaption werden Vokabulare der englischen Sprache in entsprechender Größe benutzt. Das schreiberunabhängige System ist auch auf englischen Datenbasen trainiert und erst nach der Adaption kann es für die Erkennung in der deutschen Sprache eingesetzt werden. Aufgrund dieser Tatsache ist die Vergleichsmöglichkeit des Basissystems mit den anderen Systemen möglich. Durch die Adaption wird die Fehlerrate für den Schreiber mit der ID 7 (der Schreiber mit der schwierigsten Handschrift) von $14,7\%$ auf $4,4\%$ für BA und von 26.6% auf $8,9\%$ für WA reduziert. Diese Fehlerreduktion ist gleichzeitig die höchste unter allen 17 Schreiber.

In der Tabelle 15 sind Ergebnisse der Tests auf der Wortebene für Kursivschrift angegeben. Die Erkennungsleistung ist hier erwartungsgemäß niedriger. Die Fehlerrate für die WA mit der Vokabulargröße von 20 000 wird von $28,6\%$ auf $12,7\%$, also um mehr als 50% reduziert. In den Abbildungen 12 und 13 ist die Erkennungsleistung des adaptierten Systems auf schreiberabhängigen Daten in Abhängigkeit von der Vokabulargröße wiedergegeben. Die Erkennungsleistung ist auf den Testmengen der jeweiligen 17 Schreiber getestet und gemittelt.

CURSIV, Vokabulargröße: 10 000 Wörter										
Akk	Adaptionsdaten						allgemeine Daten			
	test %		cross %		train %		test %		cross %	
	base	adp	base	adp	base	adp	base	adp	base	adp
BA	83.8	93.2	71.9	85.7	63.5	90.2	92.1	86.3	86.5	87.2
WA	75.5	89.9	61.4	78.6	51.3	84.5	80.8	76.8	74.3	78.9
CURSIV, Vokabulargröße: 20 000 Wörter										
BA	81.2	91.7	68.1	83.1	61.2	88.3	90.0	84.3	89.6	85.1
WA	71.4	87.3	55.1	73.1	47.1	81.1	80.3	72.1	80.7	74.1
CURSIV, Vokabulargröße: 65 000 Wörter										
BA	78.7	89.6	64.3	78.7	57.1	84.8	87.3	80.3	87.0	81.4
WA	66.0	83.0	47.4	66.4	39.1	74.7	74.1	64.0	74.3	66.1

Tabelle 16: Die Werte für BA und WA vor und nach der Adaption mit dem Nachtrainieren, über alle 17 Schreiber gemittelt

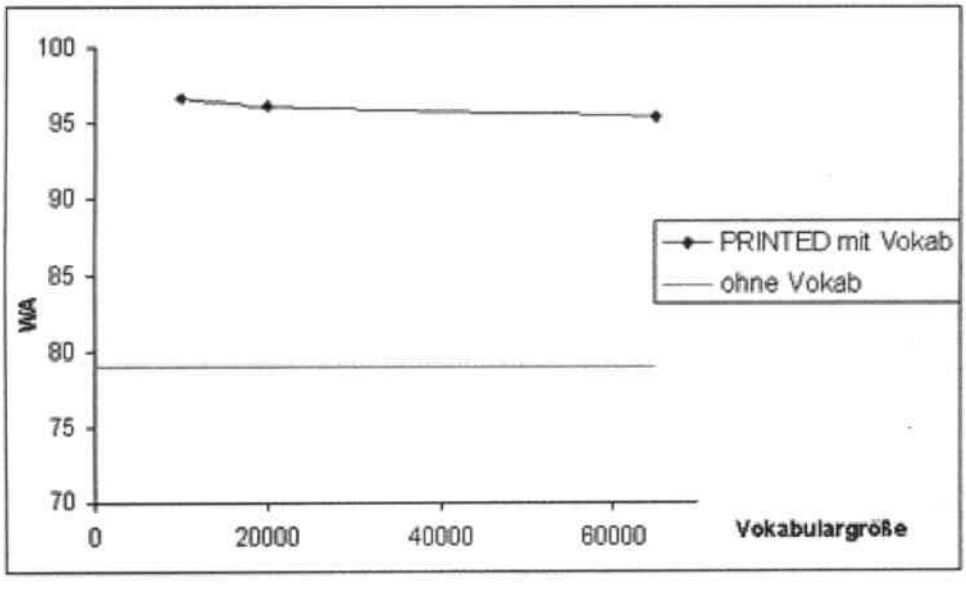

Abbildung 12: Die Erkennungsleistung nach der Adaption mit dem Nachtrainieren in Abhängigkeit von der Größe des Vokabulars

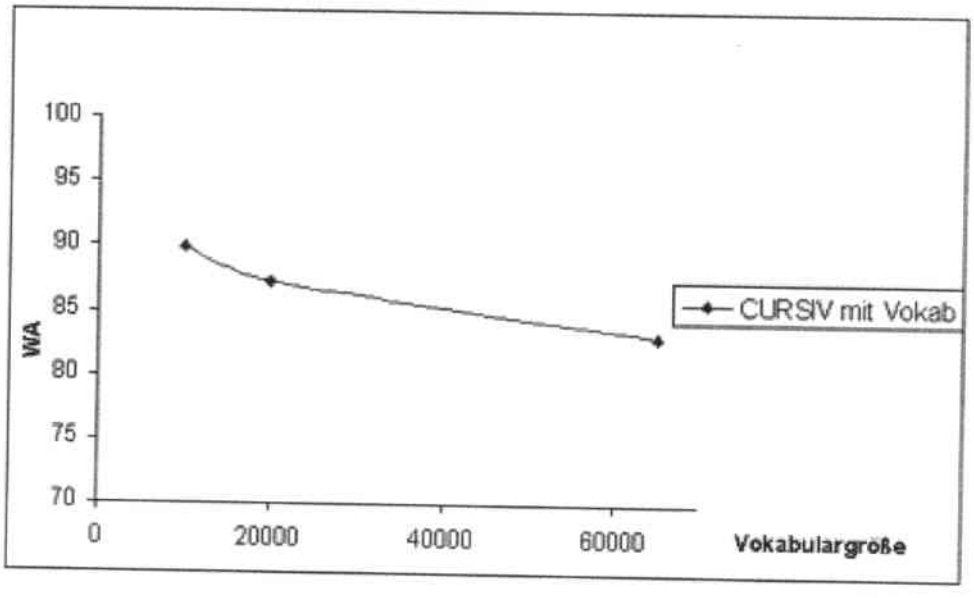

Abbildung 13: Die Erkennungsleistung nach der Adaption mit dem Nachtrainieren in Abhängigkeit von der Größe des Vokabulars

5.3 OAM: DTW-Ebene

Adaption mit OAM auf der DTW-Ebene: Großbuchstaben

Das Adaptionsexperiment, das an dieser Stelle vorgestellt wird, setzt das OAM-Verfahren auf der DTW-Ebene des MS-TDNN ein. Diese Platzierung des OAM entspricht der in [CP97] beschriebenen Verwendung des OAM-Verfahrens mit dem Unterschied, dass dort ein einfaches feed-forward neuronales Netz und kein MS-TDNN verwendet wird.

Die in diesem Versuch vorgestellten Erkennungsergebnisse eines mit OAM-Verfahren adaptierten Systems werden mit der Implementierung der Originalversion des OAM aus [CP97] erzielt (siehe auch Kapitel 2.5). Das OAM bekommt als Eingabe die DTW-Bewertung (score) der Ausgabeschicht des MS-TDNN (Abbildung 5) und in Abhängigkeit davon, ob es eine Fehlerkennung gegeben hat, fügt es ein neues RBF hinzu oder passt den Korrekturvektor an. For der Durchführung dieses Experiments werden einige Parameter des OAM optimiert. Die aus dem Kapitel 2.5 bekannten Parameter δ, a und b werden auf verschiedene Werte gesetzt und die Adaptionsfähigkeit des Systems auf allen 17 Schreibern gemittelt betrachtet. Dabei wird angenommen, dass δ unabhängig und a und b abhängig von einander sind. Der optimale Wert aus dieser Untersuchung für δ liegt bei 0.45, bei diesem Wert wird

PRINTED, Großbuchstaben										
Adaptionsdaten						allgemeine Daten				
test %		cross %		train %		test %		cross %		
base	adp	base	adp	base	adp	base	adp	base	adp	
97.7	98.3	95.6	97.8	97.5	98.9	93.6	92.8	96.3	95.6	

Tabelle 17: Die BA vor und nach der Adaption mit OAM auf der DTW-Ebene, Großbuchstaben, gemittelt über alle 17 Schreiber

die beste Erkennungsleistung auf allen schreiberabhängigen Datenmengen (Groß- und Kleinbuchstaben: test, cross, train) beobachtet, dabei werden die Werte für a und b aus [CP97] übernommen und während der ganzen Parameter-Suche unverändert belassen. Es wird festgestellt, dass bei kleinen Werten (0.1, 0.15) für δ das System die Trainingsdaten "auswendig" lernt und schlechte Generalisierungsleistung aufweist, bei größeren Werten steigt die Generalisierungsleistung und die Anzahl der eingefügten RBFs sinkt.

In der nachfolgenden Optimierung der Parameter werden die a und b abhängig voneinander auf verschiedene Werte gesetzt mit der Beibehaltung des schon optimierten Wertes für δ. Als Bewertungskriterien wird die Monotonie der Werte für Erkennungsleistung für verschiedene Iterationen auf allen Datenmengen und die über allen 17 Schreibern gemittelte Adaptionsfähigkeit genommen. Als Ergebnis wird der Wert 0.25 für a und 0.2 für b für optimal befunden. Die gleichen Werte für diese Parameter werden auch in [CP97] benutzt.

In der Tabelle 17 sind die Ergebnisse der OAM-Adaption auf Großbuchstaben dargestellt. Diese Ergebnisse sind mit den Ergebnissen des Nachtrainierens aus der Tabelle 11 direkt vergleichbar, da genau die gleichen Datenmengen benutzt werden. Bei dem Vergleich wird eine geringere, aber doch sehr ähnliche Erkennungsleistung für Großbuchstaben beobachtet, der Unterschied auf der Testmenge ist 0.3%. Bei diesem Experiment wird, wie bei Adaption durch Nachtrainieren, die Trainingsmenge permutiert und es werden wieder 20 Iterationen durchlaufen. Das OAM, das die beste Erkennungsleistung auf der Kreuzvalidierungsmenge nach 20 Iterationen der Adaption erzielt, wird in das System integriert.

Es werden auch Adaptionsversuche mit den kleineren Trainingsmengen durchgeführt und die Ergebnisse (Abbildung 14)sind direkt mit den Ergebnissen des Nachtrainierens aus der Abbildung 9 vergleichbar. Hier wird eine höhere Sensibilität des OAM-Verfahrens zu der Größe der Trainingsmenge als beim Nachtrainieren anhand des unruhigen Verlaufs der Kurven festgestellt.

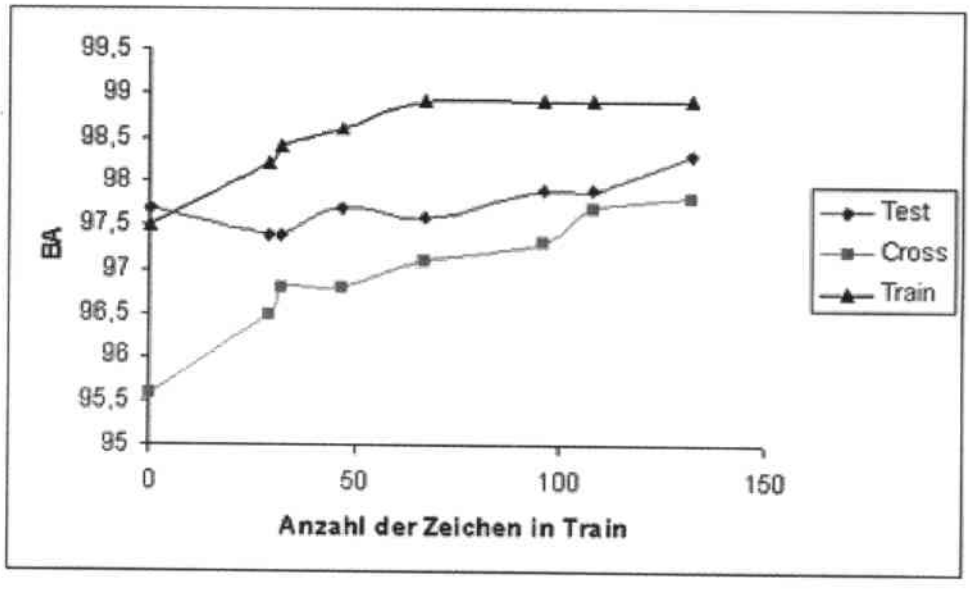

Abbildung 14: Die Adaptionsfähigkeit in Abhängigkeit von der Größe der Trainingsmenge, OAM auf DTW-Ebene, Großbuchstaben

ID	1	2	3	4	5	6	7	8	9
#RBF	7	5	0	3	0	3	0	3	0
ID	10	11	12	13	14	15	16	17	
#RBF	0	0	1	3	2	0	0	0	

Tabelle 18: Die Anzahl der hinzugefügten RBFs pro Schreiber bei OAM-Adaption auf der DTW-Ebene, Großbuchstaben

In der Tabelle 18 ist die Anzahl der bei der OAM-Adaption hinzugefügten RBFs pro Schreiber angegeben. Der maximale Wert ist bei Schreiber mit der ID 1 zu finden und ist relativ klein (vergleiche mit den Werten aus dem OAM-Adaptionsversuch auf Groß- und Kleinbuchstaben). Die Größe der entsprechenden Datei in der die OAM-Gewichtsmatrix und die RBFs gespeichert sind beträgt 12 KB, das nur unwesentlich den Speicherbedarf für das erkennende System erhöhen würde.

Das OAM-Verfahren fügt neue RBFs oder passt den Korrekturvektor an, wenn eine Fehlerkennung festgestellt wird, somit sind die Menge und die Bestandteile der RBFs (Radius und Zentrum der RBF-Funktion) und die Gewichte (C_{ij}) von der Präsentationsreihenfolge der Trainingsbeispiele abhängig. Durch diese Präsentationsreihenfolge, wie auch beim Nachtrainieren, wird die Erkennungsleistung des adaptierten Systems im geringen Maße beeinflusst. Um eine optimale Erkennungsleistung des Systems zu erreichen wird die Trainingsmenge vor jeder Trainingsiteration permutiert. Noch bessere Ergebnisse ließen sich erzielen, wenn man jeden Trainingsvorgang mit 20 Iterationen mehrmals wiederholen würde und anschließend das System mit der besten Erkennungsleistung auf der schreiberabhängigen Kreuzvalidierungsmenge wählen würde, doch diese aufwendige Trainingsmethode wird in dieser Arbeit nicht verwendet.

Adaption mit OAM auf der DTW-Ebene: Groß- und Kleinbuchstaben

Die Ergebnisse der Adaption mit OAM auf Groß- und Kleinbuchstaben sind in der Tabelle 19 angegeben. Diese Ergebnisse sind die wichtigsten Adaptionsergebnisse, denn hier muss das System die minimale Anzahl der Klassen (Groß-und Kleinbuchstaben), wie in den meisten Erkennungsaufgeben gefordert wird, von einander unterscheiden. Die Daten sind direkt mit den Daten aus der Tabelle 12 vergleichbar und erlauben eine Bewertung der beiden Verfahren. Bei der Betrachtung der Tabelle 19 wird man feststellen, dass auch bei der OAM-Adaption der höchste Wert für die Verbesserung bei dem Schreiber mit der ID 7 zu finden ist, d.h. bei dem Schreiber mit dem schwierigsten Schreibstil, jedoch bei diesem Versuch wird die Fehlerrate nur von $27,7\%$ auf $14,7\%$ gesenkt (auf den entsprechenden Daten beim Nachtrainieren wird sie auf $8,6\%$ gesenkt). Bei dem Schreiber mit der ID 5, bei dem vor der Adaption die beste Erkennungsleistung zu beobachten ist, hat sich die Fehlerrate durch Adaption mit OAM auf der DTW-Ebene im gleichen Maße wie bei der Adaption durch Nachtrainieren geändert (von $4,9\%$ auf $3,3\%$).

Die Adaptionsergebnisse gemittelt über alle Schreiber sind in der Tabelle 19 angegeben und aus dem Vergleich mit den entsprechenden Daten aus der

PRINTED, Groß- und Kleinbuchstaben										
ID	Adaptionsdaten						allgemeine Daten			
	test %		cross %		train %		test %		cross %	
	base	adp	base	adp	base	adp	base	adp	base	adp
1	85.2	90.2	86.9	91.7	88.1	98.2	93.3	90.4	92.2	85.3
2	79.9	87.4	75.4	86.4	77.8	95.7	93.3	88.3	92.2	85.6
3	91.6	97.0	88.6	94.5	89.7	96.4	93.3	92.1	92.2	90.6
4	77.7	90.2	82.3	91.8	83.3	97.1	93.3	90.8	92.2	88.8
5	95.1	96.7	95.5	98.3	95.8	98.8	93.3	92.5	92.2	90.3
6	81.5	90.2	80.4	91.9	81.2	95.3	93.3	89.1	92.2	86.0
7	72.3	85.3	74.0	86.8	73.6	95.1	93.3	90.1	92.2	87.4
8	83.9	94.1	87.5	94.9	85.3	98.5	93.3	90.6	92.2	88.6
9	88.2	95.1	89.6	94.0	88.0	98.8	93.3	91.9	92.2	90.7
10	89.7	92.3	94.2	95.8	93.4	97.8	93.3	92.6	92.2	91.1
11	94.7	96.5	94.5	97.4	93.5	96.5	93.3	92.7	92.2	91.3
12	86.2	93.1	86.7	94.7	86.2	97.6	93.3	92.0	92.2	90.5
13	86.3	91.7	88.5	94.1	87.2	94.4	93.3	86.7	92.2	84.3
14	83.9	93.2	90.2	95.6	85.7	96.0	93.3	91.7	92.2	89.4
15	93.1	95.7	90.8	96.6	91.1	98.0	93.3	92.3	92.2	90.4
16	90.8	90.8	94.7	94.7	90.4	90.4	93.3	93.3	92.2	92.2
17	91.3	96.5	93.9	97.6	92.3	99.8	93.3	92.2	92.2	90.1
Ø	**86.6**	**92.7**	**87.9**	**93.9**	**87.2**	**96.7**	**93.3**	**91.1**	**92.2**	**89.0**

Tabelle 19: Die BA vor und nach der Adaption mit OAM auf der DTW-Ebene, Groß- und Kleinbuchstaben, alle 17 Schreiber

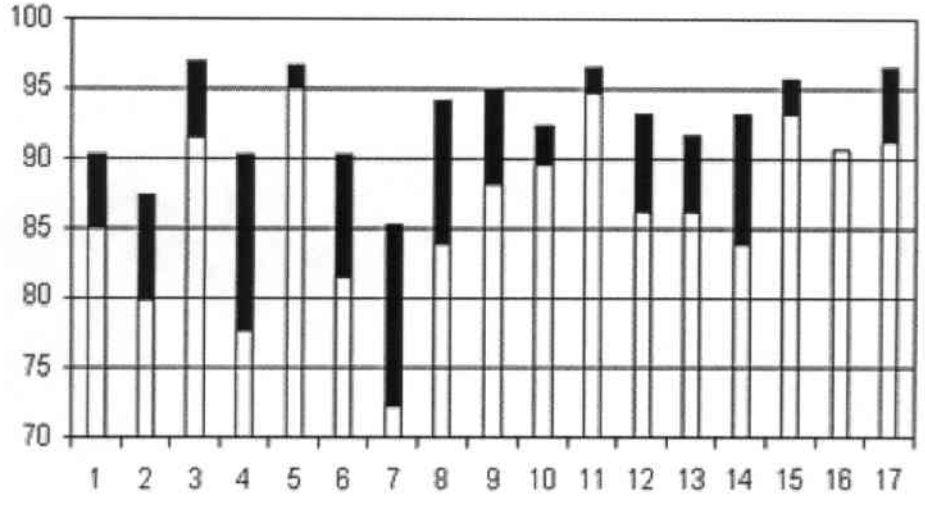

Abbildung 15: Die BA vor und nach der Adaption mit OAM auf der DTW-Ebene, Groß- und Kleinbuchstaben, alle 17 Schreiber

Tabelle 12 ergibt sich ein Unterschied von 2,3% in der Erkennungsleistung auf der Testmenge nach der Adaption, somit reduziert das OAM-Verfahren auf der DTW-Ebene die Fehlerrate im mittel von 13,4% auf 7,3% und das Nachtrainieren von 13,4% auf 5%.

In der Tabelle 20 wird die Anzahl der hinzugefügten RBFs pro Schreiberadaption dargestellt. Der Vergleich mit der entsprechender Tabelle 18 zeigt erwartungsgemäß, das bei der OAM-Adaption auf Groß- und Kleinbuchstaben die Anzahl der RBFs deutlich höher ist als bei der Adaption auf Großbuchstaben. Der Grund dafür ist, dass in diesem Fall wesentlich mehr Fehler auftreten, die ein Hinzufügen der RBF bewirken. Die höchste Anzahl der RBFs wird bei den Schreibern mit den IDs 2 und 7 festgestellt und mit dem schwierigen Schreibstil der Schreiber erklärt. Bei dem Schreiber 16 hat OAM keine RBFs hinzugefügt und somit fand hier keine Adaption statt. Die Erklärung dafür ist, dass der gewählte Wert für δ für den Schreiber mit der ID 16 zu groß gewählt wurde (siehe Kapitel 2.5).

Die Datei in der die OAM-Gewichtsmatrix und die RBFs gespeichert sind beträgt höchstens 42 KB und die Trainingszeit von 20 Iterationen beträgt etwa 2 min. Diese Angaben beziehen sich auf den Versuch mit der größten Trainingsmenge (600 Zeichen).

In der Abbildung 16 sieht man die Adaptionsfähigkeit des OAM-Verfahrens

ID	1	2	3	4	5	6	7	8	9
#RBF	47	76	24	43	13	54	75	35	35
ID	10	11	12	13	14	15	16	17	
#RBF	21	22	25	32	31	27	0	20	

Tabelle 20: Die Anzahl der hinzugefügten RBFs pro Schreiber bei OAM-Adaption auf der DTW-Ebene, Groß- und Kleinbuchstaben

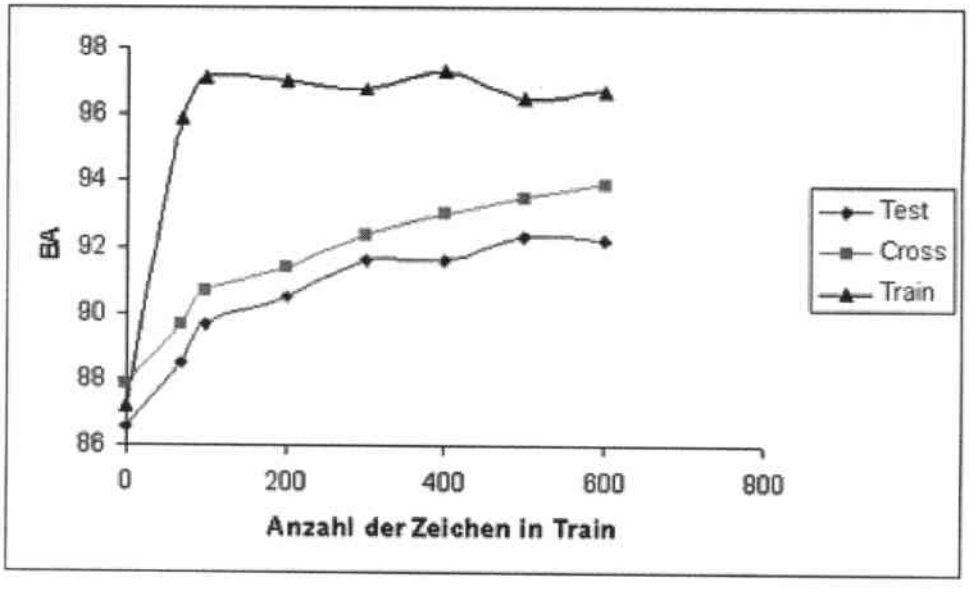

Abbildung 16: Die Adaptionsfähigkeit in Abhängigkeit von der Größe der Trainingsmenge, OAM auf der DTW-Ebene, Groß- und Kleinbuchstaben

auf der DTW-Ebene in Abhängigkeit von der Größe der Trainingsmenge. Im Vergleich zu der Abbildung 11 zeigt die OAM-Adaption ähnliches Verhalten, jedoch die Kurven sind etwas nach unten verschoben und der Kurvenverlauf ist etwas unruhiger.

Das Testen auf Wortebene

Bei diesem Experiment werden Tests auf Wortebene mit unterschiedlichen Vokabulargrößen vorgenommen. In der Tabelle 21 sind Ergebnisse präsentiert, die mit den entsprechenden Ergebnissen des Nachtrainierens aus der Tabelle 14 vergleichbar sind. Im allgemeinen wird festgestellt, dass die Reduktion der Fehlerrate bei diesem Versuch geringer ist. Der Unterschied ist etwa 1% bei den Tests mit Vokabular und 3% bei dem Test ohne Vokabular. Bei

PRINTED, freie Suche mit Sprachmodellen auf Buchstabenebene										
Akk	Adaptionsdaten						allgemeine Daten			
	test %		cross %		train %		test %		cross %	
	base	adp	base	adp	base	adp	base	adp	base	adp
BA	90.3	93.9	91.4	93.3	91.0	94.9	94.3	92.1	95.6	94.1
WA	63.3	75.8	65.2	72.4	65.4	79.7	82.4	76.3	86.4	82.3
PRINTED, Vokabulargröße: 10 000 Wörter										
BA	96.7	97.4	97.2	97.4	96.6	97.9	97.8	97.00	99.2	98.9
WA	94.1	95.8	94.8	95.3	94.8	96.8	95.4	93.6	97.8	97.0
PRINTED, Vokabulargröße: 20 000 Wörter										
BA	96.4	97.1	96.8	96.8	96.1	97.9	97.6	96.7	99.1	98.7
WA	93.2	95.1	93.7	94.2	94.0	96.6	95.2	92.9	97.4	96.3
PRINTED, Vokabulargröße: 65 000 Wörter										
BA	95.9	97.2	96.4	96.7	95.9	97.6	96.7	96.0	98.7	98.3
WA	91.7	94.6	92.5	93.5	92.9	95.9	93.0	91.0	96.4	95.2

Tabelle 21: Die Werte für BA und WA vor und nach der Adaption mit OAM auf DTW-Ebene, über alle 17 Schreiber gemittelt

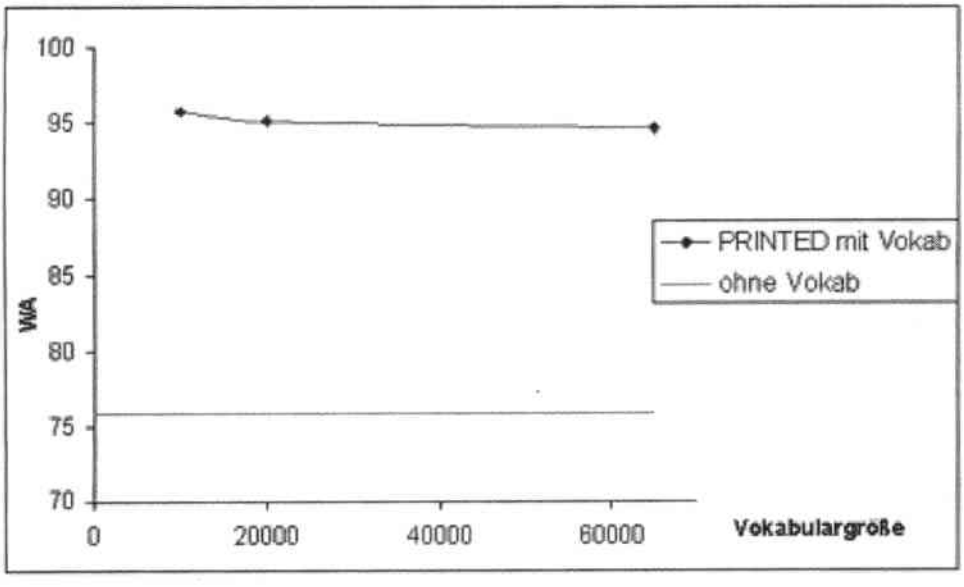

Abbildung 17: Die Erkennungsleistung nach der Adaption in Abhängigkeit von der Größe des Vokabulars

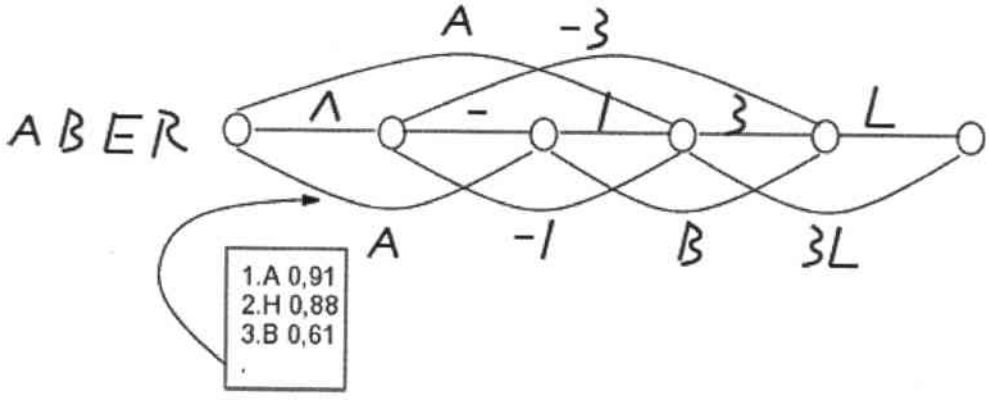

Abbildung 18: Die Suche auf PRINTED mit Graph eingeschränkt

dem gewählten Ausgangssystem lässt sich bei der Einzelzeichenerkennung die Suche auf PRINTED mit Hilfe eines Graphen einschränken (siehe [Wol00]). Dabei wird die gegebene Segmentierung in Schriftzüge (strokes) ausgenutzt. Bei den Berechnungen der N-Besten-Listen, die zu jeder Graphkante erstellt werden (siehe Abbildung 18), werden auch DTW-scores berechnet und an dieser Stelle bietet sich die Integration von OAM an, um diese scores zu korrigieren. Im Falle der Kursivschrifterkennung wird die Suche nicht mit Hilfe des Graphen eingeschränkt, da hier keine Segmentierung vorliegt und somit der kontinuierliche DTW benutzt wird. Bei diesem Ansatz ist an jeder Stelle des Pfades ein Buchstabenübergang möglich, die Berechnung der OAM Korrektur an jeder dieser Stellen wäre zu aufwendig. Eine Möglichkeit die Adaption an die Kursivschrift des Benutzers durchzuführen ist im Kapitel 5.4 beschrieben.

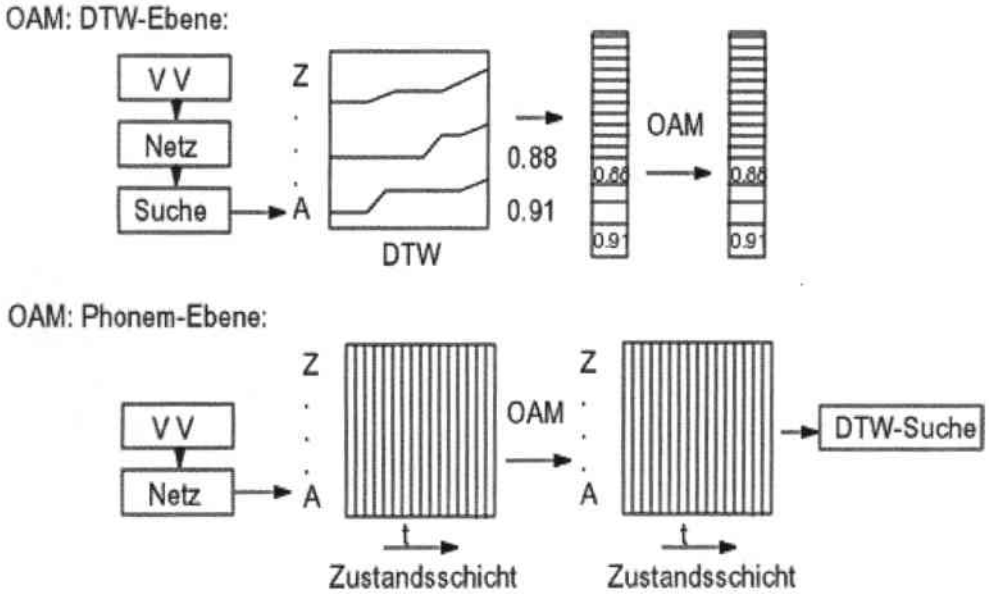

Abbildung 19: Die Funktionsweise der OAM Einsätze: OAM auf der DTW-Ebene und OAM auf der Phonem-Ebene

5.4 OAM: Phonem-Ebene

In diesem Kapitel wird die Anwendung des OAM-Adaptionsverfahrens auf der Phonem-Ebene (der Zustandsschicht im MS-TDNN) beschrieben. Die Benutzung der Zustandswahrscheinlichkeiten als Eingabe für OAM erlaubt, die später in der Hierarchie des erkennenden Systems angesiedelte Suche unverändert zu lassen. Diese frühe Platzierung von OAM-Adaption gib die Möglichkeit auch Kursivschrift zu adaptieren. Die Berechnung des Zielvektors T (siehe Kapitel 2.5) erfolgt für jeden Zeitpunkt (frame) über die gesamte Zeitdauer eines Zeichens (siehe Abbildung 19). Der maximale Wert des Vektors (0,9) entspricht dem Zustand (Phonem) innerhalb des Zeichenmodells. Die Verwendung von OAM auf der Phonem-Ebene ist rechenintensiver als auf der DTW-Ebene, da OAM-Adaption für jeden Frame durchgeführt werden muss und die Anzahl der Zustände ist ebenfalls größer.

Adaption mit OAM auf der Phonem-Ebene: Großbuchstaben

Für diesen Experiment wird der Parameter erneut δ optimiert und auf den Wert 1.7 gesetzt (siehe Kapitel 2.5 und 5.3). Die Werte für die restlichen Parameter a und b werden aus [CP97] übernommen.

Die Ergebnisse aus der Tabelle 22 sind direkt mit den entsprechenden Er-

PRINTED, Großbuchstaben									
Adaptionsdaten						allgemeine Daten			
test %		cross %		train %		test %		cross %	
base	adp	base	adp	base	adp	base	adp	base	adp
97.7	98.1	95.6	97.3	97.2	98.6	93.6	93.1	96.3	95.8

Tabelle 22: Die BA vor und nach der Adaption mit OAM auf der Phonem-Ebene, Großbuchstaben, gemittelt über alle Schreiber

ID	1	2	3	4	5	6	7	8	9
#RBF	105	149	0	94	0	88	0	165	0
ID	10	11	12	13	14	15	16	17	
#RBF	0	0	23	0	0	0	0	0	

Tabelle 23: Die Anzahl der hinzugefügten RBFs pro Schreiber bei OAM-Adaption auf der Phonem-Ebene, Großbuchstaben

gebnissen anderer Experimente dieser Arbeit vergleichbar (siehe dazu die Tabellen 11 und 17). Nach dem Vergleich wird festgestellt, dass die mit diesem Experiment erzielte Erkennungsleistung für Großbuchstaben nur unwesentlich geringer ist als die aus dem Kapitel 5.3, jedoch bei diesem Experiment ist das Hinzufügen mehrerer RBFs pro Fehlerkennung möglich, was auch aus den Statistiken der Tabellen 23 und 25 ersichtlich ist. Die Tabellen 23 und 18 weisen eine Korrelation in der Anzahl der hinzugefügten RBFs sowie bei den Schreiber-IDs, für die RBFs hinzugefügt werden. Der Verlauf der Kurven in den Abbildungen 20 und 14 ist sehr ähnlich und damit wird eine gleichstarke Abhängigkeit der Adaption von der Größe der Trainingsmenge bei beiden Versuchen festgestellt.

Adaption mit OAM auf der Phonem-Ebene: Groß- und Kleinbuchstaben

Bei den Ergebnissen dieses Versuchs auf der gemeinsamen Menge für Groß- und Kleinbuchstaben wird ein deutlicher Unterschied in der Erkennungsleistung zu dem Experiment aus dem Kapitel 5.3 festgestellt. Hier wird die Fehlerrate von 13.4% auf 9.5% auf der Testmenge verbessert, wobei mit dem OAM-Verfahren auf der DTW-Ebene eine Reduktion der Fehlerrate von 13.4% auf 7.3% erreicht wird. Als Erklärung wird die Vermutung aufgestellt, dass durch die Zustandswahrscheinlichkeiten, die hier als Eingabe für OAM benutzt werden, die Generalisierung der Adaption mehr eingeschränkt wird, da auf Modellzustände eines Zeichens hin adaptiert wird und nicht auf

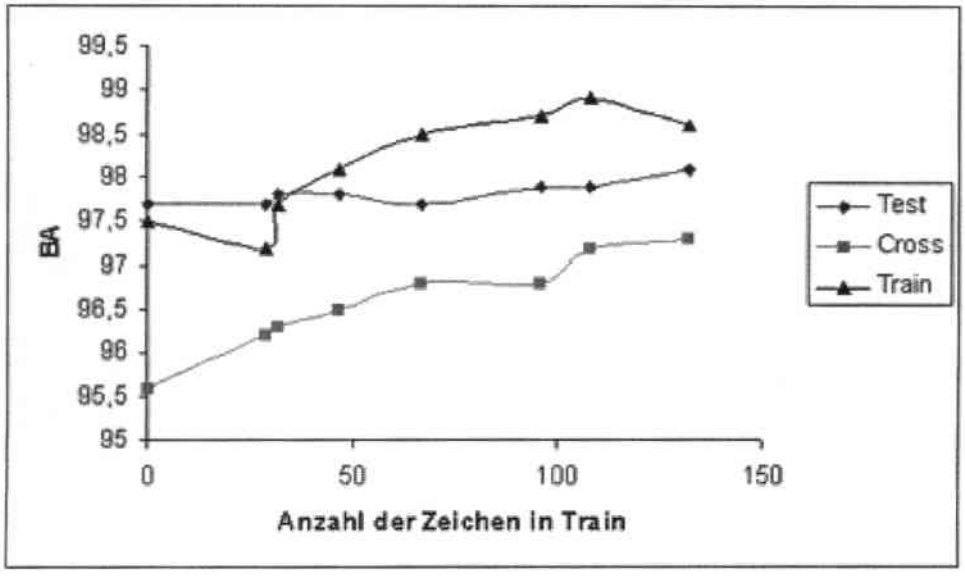

Abbildung 20: Die Adaptionsfähigkeit in Abhängigkeit von der Größe der Trainingsmenge, OAM auf Phonem-Ebene, Großbuchstaben

PRINTED, Groß- und Kleinbuchstaben									
Adaptionsdaten						allgemeine Daten			
test %		cross %		train %		test %		cross %	
base	adp	base	adp	base	adp	base	adp	base	adp
86.6	90.5	87.9	92.2	87.2	97.7	93.3	92.1	92.2	90.6

Tabelle 24: Die BA vor und nach der Adaption mit OAM auf der Phonem-Ebene auf Groß- und Kleinbuchstaben, gemittelt über alle Schreiber

den Zeichen selbst. Aus dem Vergleich der Abbildungen 22 und 16 wird die geringere Generalisierung dieses Experimentes deutlich, hier stellt man eine höhere Erkennungsleistung für die Trainingsmenge fest und eine geringere für die Test- und Kreuzvalidierungsmenge, somit lernt das System in diesem Versuch die präsentierten Daten eher "auswendig". Die Anzahl der hinzugefügten RBFs pro Schreiber ist in der Tabelle 25 deutlich höher als in der Tabelle 20, diese Tatsache wird auf die oben schon erwähnte höhere Häufigkeit der OAM-Berechnung zurückgeführt. Wie schon bei den Großbuchstaben festgestellt, wird auch hier eine Korrelation der Werte der beiden entsprechenden Tabellen beobachtet.

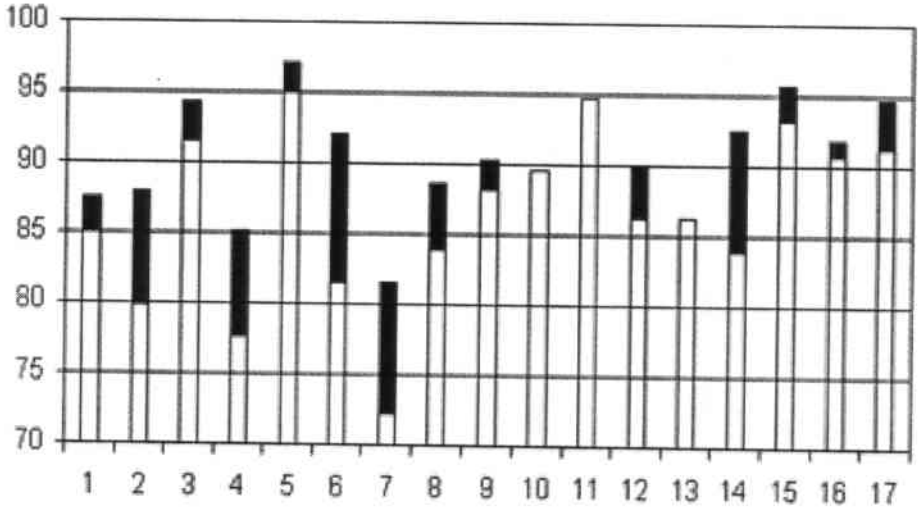

Abbildung 21: Die BA vor und nach der Adaption mit OAM auf der Phonem-Ebene auf Groß- und Kleinbuchstaben, alle 17 Schreiber

ID	1	2	3	4	5	6	7	8	9
#RBF	673	834	517	547	139	766	1150	689	636
ID	10	11	12	13	14	15	16	17	
#RBF	0	413	500	467	561	509	382	339	

Tabelle 25: Die Anzahl der hinzugefügten RBFs pro Schreiber bei OAM-Adaption auf der Phonem-Ebene auf Groß- und Kleinbuchstaben

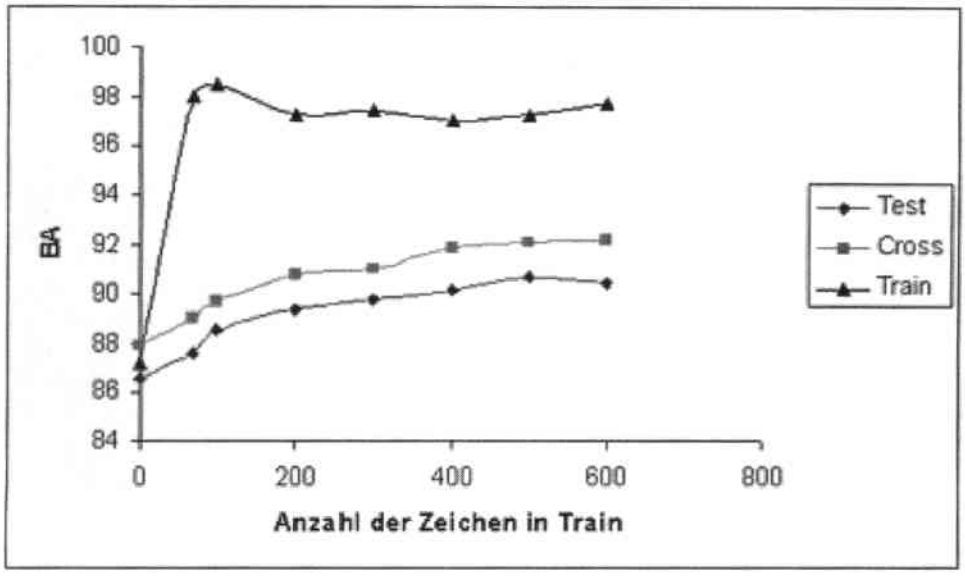

Abbildung 22: Die Adaptionsfähigkeit in Abhängigkeit von der Größe der Trainingsmenge, OAM Phonem-Ebene auf Groß- und Kleinbuchstaben

5.5 Kombination

Die Idee der Kombination der unterschiedlichen Adaptionsverfahren hat das Ziel die Vorteile dieser Verfahren zusammen zu bringen und eine höhere Erkennungsleistung des adaptierten Systems zu erzielen. Das häufige Problem dabei ist, dass auch die Nachteile der kombinierten Verfahren übernommen werden und die Erkennungsleistung nicht immer besser wird.

5.5.1 Nachtrainieren und OAM-Adaption auf der DTW-Ebene

In diesem Kapitel wird eine Kombination der in dieser Arbeit untersuchten Adaptionsverfahren vorgestellt. Das Nachtrainieren wird mit der OAM-Adaption kombiniert, indem man zuerst die adaptierten Gewichte in das System lädt und anschließend die OAM-Adaption auf der DTW-Ebene durchführt. Die Adaption wird jeweils nur auf speziellen Daten eines Schreibers durchgeführt.

Großbuchstaben

In der Tabelle 26 sind die Adaptionsergebnisse der Kombination der oben beschriebenen Adaptionsverfahren auf Großbuchstaben angegeben. Der di-

Großbuchstaben, PRINTED										
Adaptionsdaten						allgemeine Daten			
test %		cross %		train %		test %		cross %	
base	adp	base	adp	base	adp	base	adp	base	adp
97.6	98.4	95.6	97.9	97.4	100.0	94.6	93.7	97.1	96.1

Tabelle 26: Die BA vor und nach der OAM-Adaption auf DTW-Ebene und dem Nachtrainieren auf Großbuchstaben, gemittelt über alle Schreiber

rekte Vergleich zu den Tabellen 11 und 17 zeigt, dass die erzielten Ergebnisse dieses Versuches die Ergebnisse des OAM-Verfahrens allein übertreffen, doch die Ergebnisse erzielt nur mit dem Nachtrainieren sind besser. Der Tabelle 27 kann man die Anzahl der hinzugefügten RBFs entnehmen und mit der entsprechender Tabelle 18 vergleichen. Der Vergleich zeigt, dass in diesem Versuch weniger RBFs hinzugefügt werden, was auf die geringere Fehlerkennung des schon mit dem Nachtrainieren adaptierten Systems zurückgeführt wird. So werden dem OAM-Verfahren weniger Fehler präsentiert und es tritt seltener der Fall auf bei dem ein RBF hinzugefügt werden muss. Bei dem

ID	1	2	3	4	5	6	7	8	9
#RBF	1	3	2	0	0	0	7	0	2
ID	10	11	12	13	14	15	16	17	
#RBF	0	3	0	0	1	0	1	0	

Tabelle 27: Die Anzahl der hinzugefügten RBFs pro Schreiber bei kombinierter Adaption auf Großbuchstaben

Vergleich der Abbildung 23 mit den Abbildungen 9 und 14 wird festgestellt das die Adaptionsfähigkeit in ihrem Kurvenverlauf dem aus der Abbildung 9 (Nachtrainieren) sehr ähnlich ist, mit dem Unterschied, dass hier die Trainingsmenge weiter nach oben verschoben ist und der Wert 100% der Erkennungsleistung mehrmals erreicht wird. Diese Unterschiede werden auf den Einfluss von OAM-Verfahren zurückgeführt.

Groß- und Kleinbuchstaben

Bei der Analyse der Ergebnisse auf der gemeinsamen Menge von Groß- und Kleinbuchstaben wird festgestellt, dass die gewünschte Verbesserung der Erkennungsleistung gegenüber den beiden Adaptionsverfahren gleichzeitig nicht erreicht wird. Der direkte Vergleich der entsprechenden Tabellen zeigt eine ähnliche Situation wie schon bei den Ergebnissen auf Großbuchstaben beobachtet. Die Ergebnisse erzielt durch die Kombination beider Verfahren sind

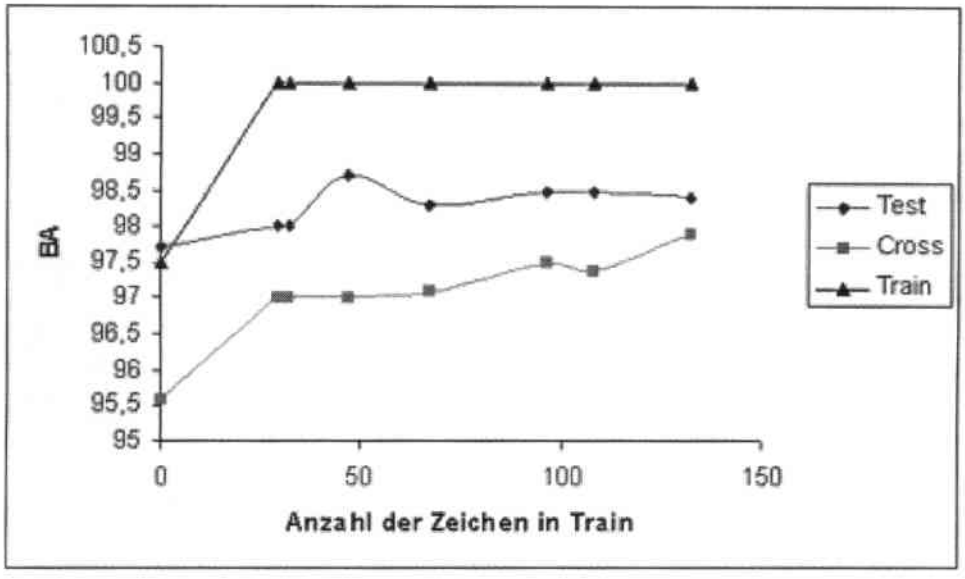

Abbildung 23: Die Adaptionsfähigkeit in Abhängigkeit von der Größe der Trainingsmenge, Kombination, Großbuchstaben

besser als die Ergebnisse die durch OAM-Adaption allein erreicht werden und schlechter als die Ergebnisse die durch Nachtrainieren erzielt werden. Auch die Anzahl der hinzugefügten RBFs ist wieder niedriger als bei der OAM-Adaption allein aus dem gleichen Grund, wie oben beschrieben. Die Kurven aus der Abbildung 25 sind in ihrem Verlauf ähnlich den Kurven aus der Abbildung 11, jedoch sind etwas mehr nach unten verschoben. Hier würde man mit einer größeren Menge an Trainingsbeispielen bessere Adaption erzielen, doch dieses muss mit deutlicher Vergrößerung der Trainingsmenge erkauft werden, da die Steigung der Kurven abnimmt.

Insgesamt wird die Kombination der beiden Adaptionsverfahren nicht als

Groß- und Kleinbuchstaben, PRINTED									
Adaptionsdaten						allgemeine Daten			
test %		cross %		train %		test %		cross %	
base	adp	base	adp	base	adp	base	adp	base	adp
86.6	94.2	87.9	94.1	87.2	98.7	93.3	87.9	92.2	84.4

Tabelle 28: Die BA vor und nach der OAM-Adaption auf DTW-Ebene und dem Nachtrainieren auf Groß- und Kleinbuchstaben, gemittelt über alle Schreiber

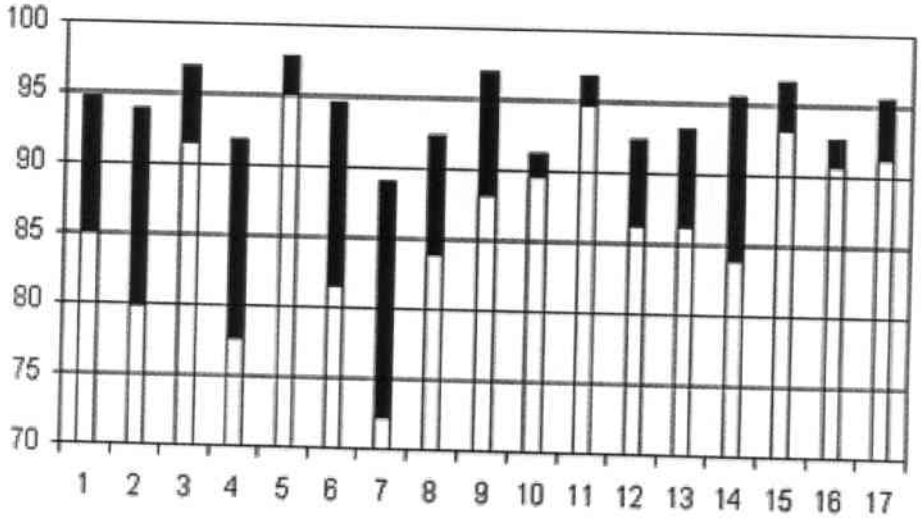

Abbildung 24: Die BA vor und nach der OAM-Adaption auf DTW-Ebene und dem Nachtrainieren auf Groß- und Kleinbuchstaben, alle 17 Schreiber

ID	1	2	3	4	5	6	7	8	9
#RBF	16	39	13	19	14	16	49	19	6
ID	10	11	12	13	14	15	16	17	
#RBF	22	29	26	29	11	12	27	10	

Tabelle 29: Die Anzahl der hinzugefügten RBFs pro Schreiber bei kombinierter Adaption auf Groß- und Kleinbuchstaben

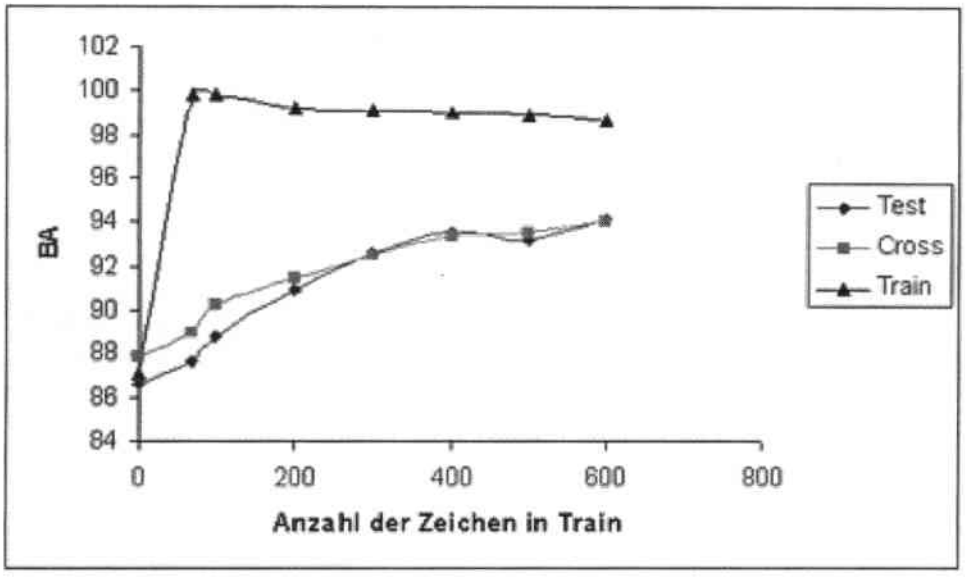

Abbildung 25: Die Adaptionsfähigkeit in Abhängigkeit von der Größe der Trainingsmenge, Kombination, Groß- und Kleinbuchstaben

erfolgversprechend gewertet, da die Ergebnisse schlechter sind als die Ergebnisse, die nur durch das Nachtrainieren erreicht werden. Dabei hat man die Nachteile der beiden Verfahren zu berücksichtigen, die bei der Kombination gemeinsam auftreten.

PRINTED, freie Suche mit Sprachmodellen, optimaler lp Parameter										
Akk	Adaptionsdaten						allgemeine Daten			
	test %		cross %		train %		test %		cross %	
	org	adp	org	adp	org	adp	org	adp	org	adp
BA	95.1	95.4	94.3	94.6	96.3	96.5	91.6	90.7	93.8	93.1
WA	79.0	80.00	76.9	77.6	84.4	84.9	74.4	72.4	81.2	80.1

Tabelle 30: Die Werte für BA und WA vor und nach der Adaption: Wahl des optimalen lp für die Suche und anschließendes Nachtrainieren, über alle 17 Schreiber gemittelt

5.6 Suche

5.6.1 Language Penalty (lp)

Aus der Statistik über die Adaption mit dem Nachtrainieren auf Wortebene ohne Vokabular wird festgestellt, dass das Verhältnis insertions/deletions für verschiedene Schreiber unterschiedlich ist. Bei nicht ausgeglichenem Verhältnis bietet sich die Adaption des lp-Parameters an. lp steht für language penalty und ist ein Strafterm für den Wortübergang bei der Berechnung des Suchpfades und beeinflusst das Verhältnis insertions/delitions. Bei diesem Adaptionsversuch wird das lp-Parameter mit binärer Suche für jeden Schreiber optimal bestimmt, indem man versucht einen Wert für lp zu finden so, dass der Quotient für insertions/delitions möglichst nah bei 1 ist. Mit dem Anheben von lp steigt die Anzahl der insertions und sinkt die Anzahl für delitions und umgekehrt. Bei dem Versuch wird der lp-Wert gesetzt und anschließend ein Nachtrainieren auf schreiberabhängigen Daten durchgeführt. Hat sich der Wert für das Verhältnis insertions/delitions in Richtung 1 bewegt oder in umgekehrter Richtung, so wird der neue kleinere Wertebereich für den lp in der binären Suche definiert. Mit dieser Vorgehensweise wird ein lp-Wert für jeden Schreiber bestimmt und anschließend die Ergebnisse für die Erkennung auf der Wortebene berechnet (siehe Tabelle 30). Im Vergleich zu den Daten aus der Tabelle 14 (oder siehe die Spalte org in der Tabelle 30) ist eine minimale Verbesserung der Erkennungsleistung feststellbar. Diese sehr kleine Verbesserung ist der Grund, weshalb keine weiteren Versuche in dieser Arbeit in diese Richtung unternommen werden. In der Tabelle 31 sind die optimalen lp-Werte für die Schreiber angegeben. Diese Werte werden für die Berechnung der Ergebnisse der Tabelle 30 benutzt. Der Standardwert für lp für die schreiberunabhängigen Daten ist $-1,2$. Es kann festgestellt werden, dass dieser Wert für einige Schreiber (2, 3, 10, 11, 13, 16, 17) als optimaler Wert gilt. Diese Tatsache kann damit erklärt werden, dass diese Schreiber

ID	1	2	3	4	5	6	7	8	9
lp	-2.9	-1.2	-1.2	-2.2	-1.9	0.05	-1.9	-2.9	-2.4
ID	10	11	12	13	14	15	16	17	
lp	-1.2	-1.2	-1.9	-1.2	-1.9	-1.4	-1.2	-1.2	

Tabelle 31: Optimale schreiberabhängige Werte für lp

einen verbreiteten, gewöhnlichen Schreibstil besitzen.

5.7 Vergleich der Verfahren

In diesem Kapitel werden die Adaptionsverfahren nach ihren Eigenschaften für den praktischen Einsatz beurteilt und die Daten aus den Tabellen 32 und 33 erläutert. Die Angaben für die Adaptionsgeschwindigkeit werden auf Adaptionsdaten des Schreibers mit der am schwersten zu erkennenden Schrift gewonnen (Schreiber mit der ID 7). Dazu werden zwei unterschiedlich große Trainingsmengen benutzt: 600 und 69 Einzelzeichen. Die Menge mit 69 Einzelzeichen enthält alle deutschen Groß- und Kleinbuchstaben mindestens ein mal. Für alle Verfahren wird die maximale Anzahl der Trainingiterationen auf 20 gesetzt. Das Nachtrainieren wird mit dem Backpropagation-Algorithmus durchgeführt. Hier werden relativ gute Zeiten erreicht, da das benutzte Basiserkennungssystem ein relativ kleines MS-TDNN besitzt und die Implementierung auf Geschwindigkeit optimiert ist. Die Adaptionsgeschwindigkeit der beiden Einsätze des OAM-Verfahrens ist von der Größe der Trainingsmenge stark abhängig. An den gemessenen Zeiten wird der Unterschied in der Komplexität deutlich. Bei OAM-Verfahren auf der DTW-Ebene werden im Vergleich zu OAM-Verfahren auf der Phonem-Ebene deutlich weniger RBFs hinzugefügt, die bei der Berechnung der Ausgabenkorrektur immer mitberücksichtigt werden müssen. Auch die Dimension der RBF-Zentren (memories) ist bei dem OAM-Verfahren auf der DTW-Ebene kleiner, was zusätzlich den Geschwindigkeitsunterschied vergrößert. Bei der großen Adaptionsmenge unterliegen beide OAM-Einsätze dem Nachtrainieren, da in diesem Fall eine große Menge an RBFs alloziert wird (bei OAM-Verfahren auf der Phonem-Ebene einige Hundert). Bei der geringeren Adaptionsmenge (69 Einzelzeichen) wird eine höhere Adaptionsgeschwindigkeit der beiden Einsätze des OAM-Verfahrens als beim Nachtrainieren gemessen. Bei dem OAM auf der Phonem-Ebene muss erwähnt werden, dass das Verfahren die kleine Trainingsmenge schnell (in 5 Iterationen) "auswendig" lernen kann und keine weiteren Iterationen benötigt. Aufgrund dieser Eigenschaften eignen sich die OAM-Einsätze eher für kleine Adaptionsmengen eines Benutzers, wenn seine Schrift nur einige wenige konsistente Besonderheiten aufweist. Bei der Berechnung der Adaptionszeit für die Kombination aus dem Nachtrainieren und OAM auf der DTW-Ebene werden die Adaptionszeiten für das Nachtrainieren und OAM summiert.

In der Tabelle 33 werden die Verfahren bezüglich der aus praktischer Sicht wichtigen Kriterien beurteilt. Solche Kriterien sind: zusätzlicher Speicher, zusätzliche Funktionalität und ist das Verfahren inkrementell oder nicht.

Adaptionsgeschwindigkeit				
Verfahren	600 Zeichen	# Iter	69 Zeichen	# Iter
Nachtrainieren	1 min 44 s	20	0 min 30 s	20
OAM: DTW-Ebene	2 min 18 s	20	0 min 17 s	20
OAM: Phonem-Ebene	35 min 19 s	20	0 min 10 s	5
Nachtr+OAM	3 min 33 s	20	0 min 32 s	20

Tabelle 32: Adaptionsgeschwindigkeit für die Trainingsmenge mit 600 und 69 Zeichen. Schreiber mit der ID 7, schwierigste Schrift.

Vergleichseigenschaften			
Verfahren	zuszl. Speicher	zuszl. Funktionalität	inkrementell
Nachtrainieren	nein	nein	nein
OAM: DTW-Ebene	mind. 50 KB	ja	ja
OAM: Phonem-Ebene	mind. 1 MB	ja	ja
Nachtr+OAM	mind. 50 KB	ja	nein
Natrain + lp-Adaption	nein	ja	nein

Tabelle 33: Vergleich der Verfahren aus praktischer Sicht

6 Zusammenfassung und Ergebnisse

6.1 Zusammenfassung

In dieser Arbeit werden verschiedene Adaptionsverfahren für On-line Handschrifterkenner vorgestellt, die eine Adaption an einen bestimmten Benutzer vornehmen. Für die Durchführung der Experimente wurden viele schreiberabhängige Daten gesammelt und bearbeitet. Diese Datensammlung kann auch für weiterführende Experimente der Adaption genutzt werden. Als Ausgangssystem wird ein auf dem MS-TDNN basierender Erkenner, der auf den allgemeinen Daten trainiert ist, benutzt. Die hier untersuchten Verfahren werden an verschiedenen Ebenen des hierarchisch aufgebauten Basissystems eingesetzt und wertvolle Ergebnisse gewonnen. Durch den direkten Vergleich der Ergebnisse ist eine effektive Bewertung der untersuchten Adaptionstechniken möglich.

6.2 Die wichtigsten Ergebnisse und Beiträge

Die Schreiberadaption durch **Nachtrainieren** bietet eine einfache und gleichzeitig eine sehr effektive Möglichkeit zur Adaption. Bei diesem Adaptionsverfahren wird eine altbewährte Technik des Trainings eines neuronalen Netzes mit Backpropagation-Algorithmus eingesetzt. Hier wird das Ausgangssystem

zusätzlich auf Adaptionsdaten des Benutzers trainiert. Das benutzeradaptierte System reduziert die Fehlerrate bis auf 2/3 auf Einzelzeichen (CURSIV, Groß- und Kleinbuchstaben, kein Vokabular) und auf der Wortebene bis auf 1/3 für die Wortakkuratheit (PRINTED, Groß- und Kleinbuchstaben mit Vokabular von 20 000 Wörtern). Diese Adaptionsmethode stellt keine neue Adaptionstechnik dar und weist auch einige Nachteile auf. Der wichtigste davon ist: beim Nachtrainieren müssen die Trainingsdaten alle auf einmal in einer vollständigen Menge bzg. der zu erkennenden Klassen zusammengefasst und bearbeitet vorliegen, da sonst bei der iterativen Präsentation der Adaptionsdaten das Netz die Klassen, zu denen keine Adaptionsbeispiele vorliegen, wieder verlernt. Aus diesem Grund wird nach neuen Adaptionsverfahren gesucht, die diesen Nachteil nicht aufweisen und eine iterative Adaption ermöglichen. In dieser Arbeit wird die Adaption durch Nachtrainieren, unter anderem, auch aus dem Grund untersucht, um die für den Vergleich mit den anderen Adaptionsverfahren wichtige Ergebnisse zu erhalten.

Im Gegensatz zum Nachtrainieren bietet das **OAM-Adaptionsverfahren** die Möglichkeit der iterativen Adaption an. In dieser Arbeit werden zwei unterschiedliche Einsatzmöglichkeiten des OAM untersucht: OAM auf der DTW-Ebene und OAM auf der Phonem-Ebene (Zustandsschicht des MS-TDNN). Die dabei erzielten Ergebnisse werden für den Vergleich untereinander und zum Nachtrainieren bereitgestellt. Das OAM wird aus der in der Arbeit von Platt und Matic (siehe dazu [CP97]) angegebenen Beschreibung implementiert und in das Ausgangssystem integriert. Dabei stellt die Integration des OAM auf der Phonem-Ebene eine Erweiterung der in [CP97] beschriebenen Verwendung für OAM dar. Der große Vorteil der Verwendung von OAM auf der Phonem-Ebene ist die Möglichkeit die nachgestellte Suche des Ausgangssystems unverändert zu lassen, was die Möglichkeit bietet dieses System auch an die Kursivschrift des Benutzers anpassen zu können. Bei allen aufgeführten Vorteilen hat das OAM-Adaptionsverfahren auch einen wichtigen Nachteil: dieses Verfahren übertrifft das Nachtrainieren in der Erkennungsleistung des adaptierten Systems nicht. Für Systeme bei denen die iterative Adaption jedoch unbedingt notwendig ist, stellt das OAM eine Lösung dar.

Im Rahmen dieser Arbeit wird auch die **Kombination** der beiden Adaptionsverfahren OAM und Nachtrainieren untersucht, jedoch sind die Ergebnisse nicht besser geworden als die, die durch das Nachtrainieren allein erzielt werden.

Bei der Suche wird ein weiteres Adaptionsverfahren untersucht, bei dem ein **Strafterm** für Wortübergänge (**lp**) für die Adaptionsdaten des Benutzers explizit bestimmt wird. Dieses Adaptionsverfahren hat eine sehr geringe Verbesserung der Erkennungsleistung bewirkt.

Verfahren	BA %
Basiserkenner	86,6
Nachtrainieren	95,0
OAM: DTW-Ebene	92,7
OAM: Phonem-Ebene	90,5
Nachtr+OAM	94,2

Tabelle 34: Die Buchstabenakkuratheit, Groß- und Kleinbuchstaben, gemittelt über alle 17 Schreiber

	PRINTED				CURSIV		
Verfahren	Vokabular				Vokabular		
	10000	20000	65000	frei	10000	20000	65000
Basiserkenner	94,1	93,2	91,7	63,3	75,5	71,4	66,0
Nachtrainieren	96,6	96,2	95,4	79,0	89,9	87,3	83,0
OAM: DTW-Ebene	95,8	95,1	94,6	75,8	-	-	-
Natrain + lp-Adaption	-	-	-	80	-	-	-

Tabelle 35: Die Wortakkuratheit, Groß- und Kleinbuchstaben, gemittelt über alle 17 Schreiber

Die in dieser Arbeit untersuchten Adaptionsverfahren geben teilweise sehr interessante Ergebnisse. Mit dem OAM-Verfahren werden zwei unterschiedliche Möglichkeiten gegeben die Adaption eines Handschrifterkenners an einen bestimmten Benutzer zu bewerkstelligen.

7 Literaturverzeichnis

Literatur

[A.B02] A.Brakensiek. *Modellierungstechniken und Adaptionsverfahren für die On- und Off-Line Schrifterkennung.* TH München, 2002.

[CG] S.Knerr-P.Binter C.Viard-Gaudin, P.M.Lallican. *The IRESTE On/Off (IRONOFF) Dual Handwriting Database.*

[CP97] John C.Platt and Nada P.Matic. *A Constructive RBF Network for Writer Adaptation.* Januar 1997.

[F.W99] F.Wallhoff. *Überwachte und unüberwachte Sprecheradaption mit verschiedenen Trainingskriterien für die automatische Spracherkennung.* Gerhard-Mercator-University Duisburg, 1999.

[F.W00] G.Rigoll F.Wallhoff, D.Willett. *Frame Diskriminative and Confidence-Driven Adaption for LVCSR.* IEEE, 2000.

[I.G94] R.Plamondon M.Liberman-S.Stanet I.Guyon, L.Schomaker. *UNIPEN Project of On-line Data Exchange and Recognizer Benchmarks.* In Proceedings of the International Conference on Pattern Recognition, October 1994.

[J.P91] J.Platt. *A resource-allocating network for function interpolation.* Neural Computation., 1991.

[J.R00] J.Rottland. *Ein hybrider Ansatz zur automatischen Spracherkennung und Sprecheradaption für große Wortschätze.* Gerhard-Mercator-Universität Duisburg, 2000.

[Man] Stefan Manke. *On-line Erkennung kursiver Handschrift bei großen Vokabularen.* Universität Karlsruhe, Dissertation.

[M.P87] M.Powel. *Radial basis functions for multivariate interpolation: A review.* Oxford. Clarendon Press., Januar 1987.

[SA95] M.Finke S.Manke and A.Waibel. *A Writer Independent, Large Vocabulary On-Line Cursive Handwriting Recognition System.* Proceedings of the International Conference on Document Analysis and Recognition, August 1995.

[SU94] S.Manke and U.Bodenhausen. *A Connectionist Recognizer for On-Line Cursive Handwriting Recognition.* IEEE, April 1994.

[S.Y00] J.Odell D.Ollason und P.Woodland S.Young, J.Jansen. *The HTK Book.* University of Cambridge, 2000.

[uB86] L.Rabiner und B.Jang. *"An Introduction to Hidden Markov Models.".* IEEE ASSP Magazine, 1986.

[uCH91] J.-L.Gauvain und C.-H.Lee. *Bayesian Lerning of Gaussian Mixture Densities for Hidden Markov Modells.* Pacific Grove, 1991.

[uCH94] J.-L.Gauvain und C.-H.Lee. *Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observation of Markov Chains.* IEEE, 1994.

[uP73] R.Duda und P.Hard. *Pattern Classifikation and Scene Analysis.* John Wiley and Sons, 1973.

[uP94] C.Leggetter und P.Woodland. *Speaker Adaption of Continuos Density HMMs using Multivariate Linear Regression.* Int. Conference on Spoken Language Processing ICSLP, 1994.

[VM93] V.Kadirkamanathan and M.Niranjan. *A function estimation approach to sequential learning with neural networks.* Neural Computation., Januar 1993.

[V.V96] P.Woodland S.Young V.Valtchev, J.Odell. *Lattice-Based Diskriminative Training for Large Vocabulary Speech Recognition Systems.* IEEE, 1996.

[Wol00] Roald Wolff. *Erkennung von handgeschriebenen mathemetischen Formeln.* Karlsruhe, 2000.

Ehrenwörtliche Erklärung

Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, 3. Februar 2004

Eduard Hermann