Ein Verfahren zur Beschleunigung eines neuronalen Netzes für die Verwendung im Image Retrieval

Daniel Braun
Heinrich-Heine-Universität
Institut für Informatik
Universitätsstr. 1
D-40225 Düsseldorf, Germany
braun@cs.uni-duesseldorf.de

ABSTRACT

Künstliche neuronale Netze haben sich für die Mustererkennung als geeignetes Mittel erwiesen. Deshalb sollen verschiedene neuronale Netze verwendet werden, um die für ein bestimmtes Objekt wichtigen Merkmale zu identifizieren. Dafür werden die vorhandenen Merkmale als erstes durch ein Art2-a System kategorisiert. Damit die Kategorien verschiedener Objekte sich möglichst wenig überschneiden, muss bei deren Berechnung eine hohe Genauigkeit erzielt werden. Dabei zeigte sich, dass das Art2 System, wie auch die Art2-a Variante, bei steigender Anzahl an Kategorien schnell zu langsam wird, um es im Live-Betrieb verwenden zu können. Deshalb wird in dieser Arbeit eine Optimierung des Systems vorgestellt, welche durch Abschätzung des von dem Art2-a System benutzen Winkels die Anzahl der möglichen Kategorien für einen Eingabevektor stark einschränkt. Des Weiteren wird eine darauf aufbauende Indexierung der Knoten angegeben, die potentiell den Speicherbedarf für die zu überprüfenden Vektoren reduzieren kann. Wie sich in den durchgeführten Tests zeigte, kann die vorgestellte Abschätzung die Bearbeitungszeit für kleine Clusterradien stark reduzieren.

Kategorien

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Neural Network*

Schlüsselwörter

Neuronale Netze, Clustering, Image Retrieval

1. EINLEITUNG

Trainiert man ein Retrieval System mit einem festen Korpus und wendet die berechneten Daten danach unverändert an, so spielt die Berechnungsdauer für einen Klassifikator eine untergeordnete Rolle, da man die Berechnung vor der eigentlichen Anwendung ausführt. Will man allerdings auch während der Nutzung des Systems weiter lernen, so sollten die benötigten Rechnungen möglichst wenig Zeit verbrauchen, da der Nutzer ansonsten entweder auf die Berechnung warten muss oder das Ergebnis, dass ihm ausgegeben wird, berücksichtigt nicht die durch ihn hinzugefügten Daten.

Für ein fortlaufendes Lernen bieten sich künstliche neuronale Netze an, da sie so ausgelegt sind, dass jeder neue Input eine Veränderung des Gedächtnisses des Netzes nach sich ziehen kann. Solche Netze erfreuen sich, bedingt durch die sich in den letzten Jahren häufenden erfolgreichen Anwendungen - zum Beispiel in der Mustererkennung - einer steigenden Beliebtheit in verschiedensten Einsatzgebieten, wie zum Beispiel auch im Image Retrieval.

Der geplante Systemaufbau sieht dabei wie folgt aus: die Merkmalsvektoren eines Bildes werden nacheinander einer Clustereinheit übergeben, welche die Merkmalsvektoren clustert und die Kategorien der in dem Bild vorkommenden Merkmale berechnet. Das Clustering der Clustereinheit passiert dabei fortlaufend. Das bedeutet, dass die einmal berechneten Cluster für alle weiteren Bilder verwendet werden. Danach werden die für das Bild gefundenen Kategorien von Merkmalen an die Analyseeinheit übergeben, in der versucht wird, die für ein bestimmtes Objekt wichtigen Kategorien zu identifizieren. Die dort gefundenen Kategorien werden dann für die Suche dieser Objekte in anderen Bildern verwendet. Das Ziel ist es dabei, die Analyseeinheit so zu gestalten, dass sie nach einem initialen Training weiter lernt und so neue Merkmale eines Objektes identifizieren soll.

Für die Analyseeinheit ist die Verwendung verschiedener neuronaler Netze geplant. Da sie aber für die vorgenommenen Optimierungen irrelevant ist, wird im Folgenden nicht weiter auf sie eingegangen.

Von dem Clusteringverfahren für die Clustereinheit werden dabei die folgenden Punkte gefordert:

- Das Clustering soll nicht überwacht funktionieren. Das bedeutet, dass es keine Zielvorgabe für die Anzahl der Cluster geben soll. Das System soll also auch bei einem bestehenden Clustering für einen neuen Eingabevektor erkennen, ob er einem Cluster zugewiesen werden kann oder ob ein neuer Cluster erstellt werden muss.

- Die Ausdehnung der Cluster soll begrenzt sein. Das soll dazu führen, dass gefundene Merkmalskategorien mit höherer Wahrscheinlichkeit zu bestimmten Objek-

25^{th} GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 28.05.2013 - 31.05.2013, Ilmenau, Germany.

ten gehören und nicht die Vektoren anderer Objekte die Kategorie verschmutzen.

- Das Clustering Verfahren sollte auch bei einer hohen Anzahl an Clustern, die aus der gewünschten hohen Genauigkeit der einzelnen Cluster resultiert, schnell berechnet werden können.

In dieser Arbeit wird ein Adaptive Resonance Theory Netz [5] verwendet, genauer ein Art2 Netz [1], da es die beiden ersten Bedingungen erfüllt. Denn dieses neuronale Netz führt ein nicht überwachtes Clustering aus, wobei es mit jedem Eingabevektor weiter lernt und gegebenenfalls neue Cluster erschafft. Der genaue Aufbau dieses Systems wird in Kapitel 3 genauer dargestellt.

Zur Beschreibung des Bildes dienen SIFT Deskriptoren [9, 10], welche mit 128 Dimensionen einen sehr großen Raum für mögliche Kategorien aufspannen. Dadurch wächst die Knotenanzahl innerhalb des Art2 Netzes rapide an, was zu einer Verlangsamung des Netzes führt. Deshalb wird die Art2-a Variante [2] verwendet, welche das Verhalten des Art2 Systems approximiert. Dieses System hat die Vorteile, dass es zum Einen im Vergleich zu Art2 um mehrere Größenordnungen schneller ist und sich zum Anderen gleichzeitig auch noch größtenteils parallelisieren lässt, wodurch ein weiterer Geschwindigkeitsgewinn erzielt werden kann.

Dennoch zeigt sich, dass durch die hohe Dimension des Vektors, die für die Berechnung der Kategorie benötigten Skalarprodukte, unter Berücksichtigung der hohen Anzahl an Knoten, weiterhin sehr rechenintensiv sind. Dadurch steigt auch bei starker Parallelisierung, sofern die maximale Anzahl paralleler Prozesse begrenzt ist, die Zeit für die Bearbeitung eines neuen Vektors mit fortlaufendem Training kontinuierlich an. Aus diesem Grund wird in Kapitel 4 eine Erweiterung des Systems vorgestellt, die die Menge der Kandidaten möglicher Gewinnerknoten schon vor der teuren Berechnung des Skalarproduktes verkleinert.

Der weitere Aufbau dieser Arbeit sieht dabei wie folgt aus: in dem folgenden Kapitel 2 werden einige ausgewählte Ansätze aus der Literatur genannt, in denen neuronale Netze für das Image Retrieval verwendet werden. Um die Plausibilität der Erweiterung zu verstehen, werden dann in Kapitel 3 die dafür benötigten Mechanismen und Formeln eines Art2-a Systems vorgestellt. Kapitel 4 fokussiert sich danach auf die vorgeschlagene Erweiterung des bekannten Systems. In Kapitel 5 wird die Erweiterung evaluiert, um danach in dem folgenden Kapitel eine Zusammenfassung des Gezeigten sowie einen Ausblick zu geben.

2. VERWANDTE ARBEITEN

In diesem Kapitel werden einige Ansätze aus der Literatur vorgestellt, in denen neuronale Netze für verschiedene Aufgabenstellungen im Image Retrieval verwendet werden. Darunter fallen Themengebiete wie Clustering und Klassifikation von Bildern und ihren Merkmalsvektoren.

Ein bekanntes Beispiel für die Verwendung von neuronalen Netzen im Image Retrieval ist das PicSOM Framework, welches in [8] vorgestellt wird. Dort werden TS-SOMs (Tree Structured Self Orienting Maps) für die Bildsuche verwendet. Ein Bild wird dabei durch einen Merkmalsvektor dargestellt. Diese Vektoren werden dann dem neuronalen Netz präsentiert, welches sie dann der Baumstruktur hinzufügt, so dass im Idealfall am Ende jedes Bild in der Baumstruktur repräsentiert wird. Bei der Suche wird der Baum dann

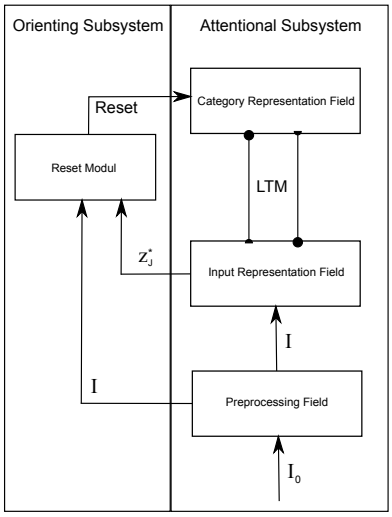

Abbildung 1: Skizze eines Art2-a Systems

durchlaufen und der ähnlichste Knoten als Antwort gewählt. Das Framework verwendet dabei das Feedback des Nutzers, wodurch nach jeder Iteration das Ergebnis verfeinert wird. Das neuronale Netz dient hier somit als Klassifikator.

[12] benutzt das Fuzzy Art neuronale Netz, um die Merkmalsvektoren zu klassifizieren. Sie schlagen dabei eine zweite Bewertungsphase vor, die dazu dient, das Netz an ein erwartetes Ergebnis anzupassen, das System damit zu überwachen und die Resultate des Netzes zu präzisieren.

In [6] wird ein Radial Basis Funktion Netzwerk (RBF) als Klassifikator verwendet. Eins ihrer Verfahren lässt dabei den Nutzer einige Bilder nach der Nähe zu ihrem Suchziel bewerten, um diese Bewertung dann für das Training ihrer Netzwerke zu verwenden. Danach nutzen sie die so trainierten neuronalen Netze zur Bewertung aller Bilder der Datenbank.

Auch [11] nutzt ein Radial Basis Funktion Netz zur Suche nach Bildern und trainiert dieses mit der vom Nutzer angegebenen Relevanz des Bildes, wobei das neuronale Netz nach jeder Iteration aus Bewertung und Suche weiter trainiert wird.

In [3] wird ein Multiple Instance Netzwerk verwendet. Das bedeutet, dass für jede mögliche Klasse von Bildern ein eigenes neuronales Netz erstellt wird. Danach wird ein Eingabebild jedem dieser Subnetze präsentiert und gegebenenfalls der dazugehörigen Klasse zugeordnet.

3. ART2-A BESCHREIBUNG

In diesem Kapitel werden die benötigten Mechanismen eines Art2-a Systems vorgestellt. Für das Verständnis sind dabei nicht alle Funktionen des Systems nötig, weshalb zum Beispiel auf die nähere Beschreibung der für das Lernen benötigten Formeln und des Preprocessing Fields verzichtet wird. Für weiterführende Informationen über diese beiden Punkte sowie generell über das Art2-a System sei deshalb auf [1, 2] verwiesen.

Wie in Bild 1 zu sehen ist, besteht das System aus zwei Subsystemen: einem Attentional Subsystem, in dem die Bearbeitung und Zuordnung eines an den Eingang angelegten Vektors ausgeführt wird, sowie einem Orienting Subsystem, welches die Ähnlichkeit des Eingabevektors mit der vorher gewählten Gewinnerkategorie berechnet und diese bei zu

geringer Nähe zurücksetzt.

Innerhalb des Category Representation Field F_2 liegen die Knoten die für die einzelnen Vektorkategorien stehen. Dabei wird die Beschreibung der Kategorie in der Long Term Memory (LTM) gespeichert, die das Feld F_2 mit dem Input Representation Field F_1 in beide Richtungen verbindet.

Nach [2] gilt für den Aktivitätswert T von Knoten J in dem Feld F_2:

$$T_J = \begin{cases} \alpha \cdot \sum_{i=1}^{n} I_i, & \text{wenn } J \text{ nicht gebunden ist,} \\ I \cdot z_J^*, & \text{wenn } J \text{ gebunden ist.} \end{cases}$$

I_i steht dabei für den durch das Preprocessing Field F_0 berechneten Input in das Feld F_1 und α ist ein wählbarer Parameter, der klein genug ist, so dass die Aktivität eines ungebundenen Knotens für bestimmte Eingangsvektoren nicht immer größer ist als alle Aktivitätswerte der gebundenen Knoten. Hierbei gilt ein Knoten als gebunden, wenn ihm mindestens ein Vektor zugeordnet wurde.

Da der Aktivitätswert für alle nicht gebundenen Knoten konstant ist und deshalb nur einmal berechnet werden muss, ist dieser Fall für eine Effizienzsteigerung von untergeordnetem Interesse und wird deshalb im Folgenden nicht weiter betrachtet.

Wie in [2] gezeigt wird, sind sowohl I als auch z_J^*, durch die Anwendung der euklidischen Normalisierung, Einheitsvektoren, weshalb folglich

$$\|I\| = \|z_J^*\| = 1 \tag{1}$$

gilt. Deshalb folgt für die Aktivitätswerte der gebunden Kategorieknoten:

$$\begin{aligned} T_J &= I \cdot z_J^* \\ &= \|I\| \cdot \|z_J^*\| \cdot \cos\theta \\ &= \cos\theta \end{aligned} \tag{2}$$

Die Aktivität eines Knotens entspricht damit dem Winkel zwischen dem Eingangsvektor I und dem LTM-Vektor z_J^*. Damit der Knoten mit dem Index J gewählt wird, muss

$$T_J = \max_j \{T_j\}$$

gelten, sprich der Knoten mit der maximalen Aktivität wird als mögliche Kategorie gewählt. Dabei wird bei Gleichheit mehrerer Werte der zu erst gefundene Knoten genommen. Die maximale Distanz, die das Resetmodul akzeptiert, wird durch den Schwellwert ρ, im Folgenden Vigilance Parameter genannt, bestimmt, mit dem die, für die Art2-a Variante benötigte, Schwelle ρ^* wie folgt berechnet wird:

$$\rho^* = \frac{\rho^2(1+\sigma)^2 - (1+\sigma^2)}{2\sigma}$$

mit

$$\sigma \equiv \frac{cd}{1-d} \tag{3}$$

und c und d als frei wählbare Parameter des Systems, die der Beschränkung

$$\frac{cd}{1-d} \leq 1$$

unterliegen. Damit wird der Knoten J genau dann abgelehnt, wenn

$$T_J < \rho^* \tag{4}$$

gilt. Ist das der Fall, wird ein neuer Knoten aktiviert und somit eine neue Kategorie erstellt. Mit 2 und 4 folgt damit, dass ein Knoten nur dann ausgewählt werden kann, wenn für den Winkel θ zwischen dem Eingabevektor I und dem gespeicherten LTM-Vektor z_J^*

$$\cos\theta \geq \rho^* \tag{5}$$

gilt. Da die einzelnen Rechnungen, die von dem System ausgeführt werden müssen, unabhängig sind, ist dieses System hochgradig parallelisierbar, weshalb alleine durch Ausnutzung dieser Tatsache die Berechnungszeit stark gesenkt werden kann. Mit steigender Knotenanzahl lässt sich das System dennoch weiter optimieren, wie in dem folgenden Kapitel gezeigt werden soll.

Das Art2-a System hat dabei allerdings einen Nachteil, denn bedingt durch die Nutzung des Kosinus des Winkels zwischen zwei Vektoren werden Vektoren, die linear abhängig sind, in dieselbe Kategorie gelegt. Dieses Verhalten ist für die geforderte Genauigkeit bei den Kategorien unerwünscht. Dennoch lässt sich dieses Problem leicht durch die Erhebung weiterer Daten, wie zum Beispiel den Clustermittelpunkt, lösen, weshalb im Folgenden nicht weiter darauf eingegangen wird.

4. VORGENOMMENE OPTIMIERUNG

Dieses Kapitel dient der Beschreibung der verwendeten Abschätzung und ihrer Implementierung in das Art2-a System. Abschließend wird dann noch auf eine weitere Verbesserung, die sich durch diese Implementierung ergibt, eingegangen. Der Aufbau des Abschnitts ist dabei wie folgt: in Unterabschnitt 1 wird das Verfahren zur Abschätzung des Winkels vorgestellt. In dem folgenden Unterabschnitt 2 wird dann gezeigt, wie man diese Abschätzung in das Art2-a System integrieren kann. In dem letzten Unterabschnitt folgt dann eine Vorstellung der Abschätzung als Index für die Knoten.

4.1 Abschätzung des Winkels

In [7] wird eine Methode zur Schätzung der Distanz zwischen einem Anfragevektor und einem Datenvektor beschrieben. Im Folgenden wird beschrieben, wie man Teile dieses Verfahrens nutzen kann, um damit die Menge möglicher Knoten schon vor der Berechnung des Aktivitätswertes T_J zu verringern. Das Ziel ist es, die teure Berechnung des Skalarproduktes zwischen I und z_J^* möglichst selten auszuführen und gleichzeitig möglichst wenig Daten im Speicher vorrätig halten zu müssen. Dafür wird der unbekannte Winkel θ zwischen den beiden Vektoren P und Q durch die bekannten Winkel α und β zwischen beiden Vektoren und einer festen Achse T wie folgt approximiert:

$$\begin{aligned} \cos\theta &\leq \cos\left(|\alpha - \beta|\right) \\ &= \cos\left(\alpha - \beta\right) \\ &= \cos\alpha\cos\beta + \sin\alpha\sin\beta \\ &= \cos\alpha\cos\beta + \sqrt{1-\cos\alpha^2}\sqrt{1-\cos\beta^2} \end{aligned} \tag{6}$$

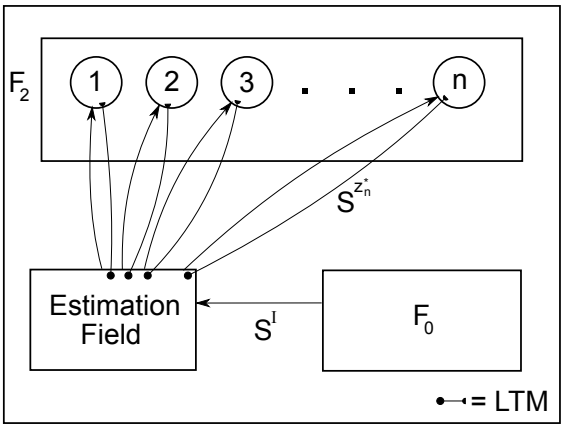

Abbildung 2: Erweiterung des Art2 Systems mit einem neuen Feld für die Abschätzung des Winkels

Als Achse T wird hierbei ein n-dimensionaler mit Einsen gefüllter Vektor verwendet, wodurch für die L2-Norm des Achsenvektors $\|T\| = \sqrt{n}$ folgt. Eingesetzt in die Formel

$$\cos\theta = \frac{\langle P, Q\rangle}{\|P\|\|Q\|}$$

ergibt sich damit, unter Ausnutzung von (1), für das System mit den Vektoren I und z_J^*:

$$cos\alpha = \frac{\sum_{i=1}^{n} I_i}{\sqrt{n}} \text{ und } cos\beta = \frac{\sum_{i=1}^{n} z^*_{J_i}}{\sqrt{n}}$$

Mit S^I und $S^{z_J^*}$ als jeweilige Summe der Vektorwerte reduziert sich, unter Verwendung der Formel (6), die Abschätzung des Kosinus vom Winkel θ auf

$$\begin{aligned}\cos\theta &\leq \frac{S^I * S^{z_J^*}}{n} + \sqrt{(1 - \frac{{S^I}^2}{n})(1 - \frac{{S^{z_J^*}}^2}{n})} \\ &= \frac{S^I * S^{z_J^*} + \sqrt{(n - {S^I}^2)(n - {S^{z_J^*}}^2)}}{n}\end{aligned}$$

Diese Abschätzung ermöglicht es nun, die Menge der Kandidaten möglicher Knoten für einen Eingabevektor I vorzeitig zu reduzieren, indem man ausnutzt, dass der wirkliche Winkel zwischen Eingabevektor und in der LTM gespeichertem Vektor maximal genauso groß ist, wie der mit der gezeigten Formel geschätzte Winkel zwischen beiden Vektoren. Damit ist diese Abschätzung des wirklichen Winkels θ verlustfrei, denn es können keine Knoten mit einem tatsächlich größeren Kosinuswert des Winkels aus der Menge an Kandidaten entfernt werden. Daraus resultiert, dass ein Knoten nur dann weiter betrachtet werden muss, wenn die Bedingung

$$\frac{S^I * S^{z_J^*} + \sqrt{(n - {S^I}^2)(n - {S^{z_J^*}}^2)}}{n} \geq \rho^* \qquad (7)$$

erfüllt wird.

4.2 Erweiterung des Systems

Damit die Bedingung (7) ausgenutzt werden kann, wird das Art2 System um ein weiteres Feld, im Folgenden Estimation Field genannt, erweitert. Dieses Feld soll als Schnittstelle zwischen F_0 und F_2 dienen und die Abschätzung des Winkels zwischen dem Eingabevektor und dem gespeicherten LTM Vektor vornehmen. Dazu wird dem Feld, wie in Abbildung 2 gezeigt wird, von dem Feld F_0 die Summe S^I übergeben. Innerhalb des Feldes gibt es dann für jeden Knoten J im Feld F_2 eine zugehörige Estimation Unit J'. In der Verbindung von jedem Knoten J zu der ihm zugehörigen Estimation Unit J' wird die Summe der Werte des jeweiligen LTM Vektors $S^{z_J^*}$ als LTM gespeichert. Die Estimation Unit berechnet dann die Funktion

$$f(J) = \frac{S^I * S^{z_J^*} + \sqrt{(n - {S^I}^2)(n - {S^{z_J^*}}^2)}}{n}$$

für den ihr zugehörigen Knoten J. Abschließend wird als Aktivierungsfunktion, für die Berechnung der Ausgabe $o_{J'}$ der Estimation Unit J', die folgende Schwellenfunktion verwendet:

$$o_{J'} = \begin{cases} 1, & \text{wenn } f(J) \geq \rho^* \\ 0, & \text{sonst} \end{cases} \qquad (8)$$

Damit ergibt sich für die Aktivitätsberechnung jedes Knotens des F_2 Feldes die angepasste Formel

$$T_J = \begin{cases} \alpha * \sum_i I_i, & \text{wenn } J \text{ nicht gebunden ist,} \\ I * z_J^*, & \text{wenn } J \text{ gebunden ist und } o_{J'} = 1 \text{ gilt,} \\ 0 & \text{wenn } o_{J'} = 0 \text{ gilt.} \end{cases} \qquad (9)$$

mit $o_{J'}$ als Ausgabe des Estimation Field zu Knoten J.

4.3 Verwendung als Index

Durch die gezeigte Kosinusschätzung werden unnötige Skalarprodukte vermieden und somit das System beschleunigt. Allerdings kann es bei weiterhin wachsender Anzahl der Knoten, zum Beispiel weil der Arbeitsspeicher nicht ausreicht, nötig werden, nicht mehr alle LTM Vektoren im Speicher zu halten, sondern nur ein Set möglicher Kandidaten zu laden und diese dann gezielt zu analysieren. In dem folgenden Abschnitt wird gezeigt, wie die Abschätzung sinnvoll als Index für die Knoten verwendet werden kann.

Für die Indexierung wird als Indexstruktur ein B^+-Baum mit der Summe der Werte jedes LTM Vektors $S^{z_J^*}$ und der ID J des Knotens als zusammengesetzten Schlüssel verwendet. Für die Sortierreihenfolge gilt: zuerst wird nach $S^{z_J^*}$ sortiert und dann nach J. Dadurch bleibt der B^+-Baum für partielle Bereichsanfragen nach dem Wert der Summe optimiert. Damit das funktioniert muss allerdings die Suche so angepasst werden, dass sie bei einer partiellen Bereichsanfrage für die ID den kleinstmöglichen Wert einsetzt und dann bei der Ankunft in einem Blatt der Sortierung bis zum ersten Vorkommen, auch über Blattgrenzen hinweg, der gesuchten Summe folgt.

Dieser Index wird nun verwendet, um die Menge der Kandidaten einzuschränken, ohne, wie in der vorher vorgestellten Optimierung durch die Estimation Unit, alle Knoten durchlaufen zu müssen. Anschaulich bedeutet das, dass das Art2-a System nur noch die der Menge an Kandidaten

für den Eingabevektor I angehörenden Knoten sehen soll und somit nur in diesen den Gewinnerknoten suchen muss. Für diesen Zweck müssen mögliche Wertebereiche der gespeicherten $S^{z_J^*}$ für einen beliebigen Eingabevektor festgelegt werden. Dies geschieht wieder mit Hilfe der Bedingung (7):

$$\frac{S^I \cdot S^{z_J^*} + \sqrt{(n - {S^I}^2)(n - {S^{z_J^*}}^2)}}{n} \geq \rho$$

$$\sqrt{(n - {S^I}^2)(n - {S^{z_J^*}}^2)} \geq \rho \cdot n - S^I \cdot S^{z_J^*}$$

Für $\rho \cdot n - S^I \cdot S^{z_J^*} < 0$ ist diese Ungleichung offensichtlich immer erfüllt, da die Quadratwurzel auf der linken Seite immer positiv ist. Damit ergibt sich die erste Bedingung:

$$S^{z_J^*} > \frac{\rho \cdot n}{S^I} \tag{10}$$

Nun wird im Folgenden noch der Fall $\rho \cdot n \geq S^I \cdot S^{z_J^*}$ weiter betrachtet:

$$\sqrt{(n - {S^I}^2)(n - {S^{z_J^*}}^2)} \geq \rho \cdot n - S^I \cdot S^{z_J^*}$$

$$n \cdot (1 - \rho^2) - {S^I}^2 \geq {S^{z_J^*}}^2 - 2\rho S^I S^{z_J^*}$$

$$(n - {S^I}^2)(1 - \rho^2) \geq (S^{z_J^*} - \rho \cdot S^I)^2$$

Damit ergibt sich:

$$\sqrt{(n - {S^I}^2)(1 - \rho^2)} \geq S^{z_J^*} - \rho \cdot S^I \geq -\sqrt{(n - {S^I}^2)(1 - \rho^2)} \tag{11}$$

Mit den Bedingungen (10) und (11) können nun die partiellen Bereichsanfragen an den Index für einen beliebigen Eingabevektor I wie folgt formuliert werden:

$$r_1 = [\rho S^I - \sqrt{(n - {S^I}^2)(1 - \rho^2)}, \rho S^I + \sqrt{(n - {S^I}^2)(1 - \rho^2)}]$$

$$r_2 = [\frac{\rho \cdot n}{S^I}, \infty]$$

Da für diese Bereichsanfragen die Bedingung (7) genutzt wird und somit alle geschätzten Winkel größer als ρ^* sind, hat bei der Verwendung des Indexes das Estimation Field keinen Effekt mehr.

5. EVALUATION

In diesem Kapitel wird die gezeigte Abschätzung evaluiert. Der vorgeschlagene Index wird dabei aber noch nicht berücksichtigt.

5.1 Versuchsaufbau

Für die Evaluierung des gezeigten Ansatzes wurde ein Computer mit einem Intel Core 2 Duo E8400 3 GHz als Prozesser und 4 GB RAM benutzt.

Als Datensatz wurden Bilder von Flugzeugen aus dem Caltech 101 Datensatz [4] verwendet. Diese Bilder zeigen verschiedene Flugzeuge auf dem Boden beziehungsweise in der Luft. Für den Geschwindigkeitstest wurden 20 Bilder aus dem Pool ausgewählt und nacheinander dem neuronalen Netz präsentiert. Im Schnitt produzieren die benutzten Bilder dabei 4871 SIFT Feature Vektoren pro Bild.

Bedingt dadurch, dass die Ansätze verlustfrei sind, wird nur die Rechenzeit der verschiedenen Verfahren gegenüber gestellt, denn es sind keine Einbußen in der Güte des Ergebnisses zu erwarten. Außerdem wird die mögliche Parallelisierung nicht weiter betrachtet, da bei einer begrenzten Anzahl von parallelen Prozessen die Anzahl der Knoten pro Prozess mit jedem weiteren Bild steigt und den Prozess so verlangsamt. Als mögliche Werte für den Schwellwert ρ wurden die zwei, in der Literatur öfter genannten, Werte 0.95 und 0.98 sowie der Wert 0.999 verwendet. Für die restlichen benötigten Parameter aus Formel (3) und (9) gilt: $c = 0.1$, $d = 0.9$ und $\alpha = 0$

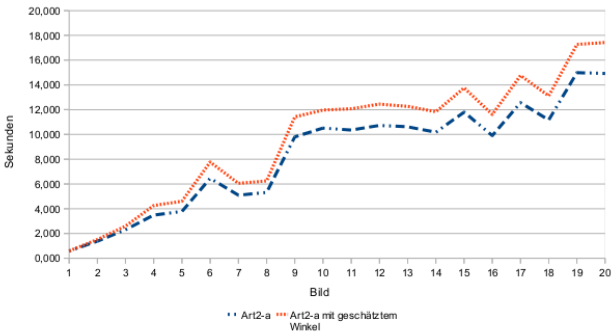

Abbildung 3: Zeitmessung für $\rho = 0.95$

5.2 Ergebnisse

Für die kleineren Vigilance Werte von 0.95 und 0.98 zeigt sich, wie in den Abbildungen 3 und 4 zu sehen ist, dass die Abschätzung hier kaum einen Vorteil bringt. Sie ist sogar langsamer als das originale System. Dies liegt daran, dass die Abschätzung durch Verwendung nur eines Wertes, nämlich der Summe, viel zu ungenau ist, um bei diesem Vigilance Wert genug Knoten herauszufiltern, da fast alle Knoten über der Grenze liegen. Da deshalb kaum Zeit gewonnen wird, wird das System durch den betriebenen Mehraufwand langsamer. Mit steigendem Vigilance Parameter nimmt auch der Nutzen des Verfahrens zu, da die Anzahl der entfernten Knoten signifikant zunimmt. Dies sieht man deutlich in Abbildung 5, in der die benötigte Rechenzeit für einen Wert von 0.999 dargestellt ist. In diesem Fall filtert die gezeigte Abschätzung sehr viele Knoten heraus, weshalb der Zeitgewinn den Zeitverlust durch den größeren Aufwand weit übersteigt. Da aber möglichst genaue Kategorien erwünscht sind, ist ein hoher Vigilance Parameter die richtige Wahl. Deshalb kann das gezeigte Verfahren für das angestrebte System adaptiert werden.

6. RESÜMEE UND AUSBLICK

In dieser Arbeit wurde eine Optimierung des Art2-a Systems vorgestellt, die durch Abschätzung des Winkels zwischen Eingabevektor und gespeichertem Vektor die Menge an zu überprüfenden Kandidaten für hohe Vigilance Werte stark reduzieren kann. Des Weiteren wurde ein Ansatz zur Indexierung der Knoten basierend auf der für die Abschätzung nötigen Summe vorgestellt. Auch wenn eine abschließende Analyse des gezeigten noch offen ist, so scheint dieser Ansatz dennoch erfolgversprechend für die erwünschten hohen Vigilance Werte.

Aufbauend auf dem gezeigten wird unsere weitere Forschung die folgenden Punkte beinhalten:

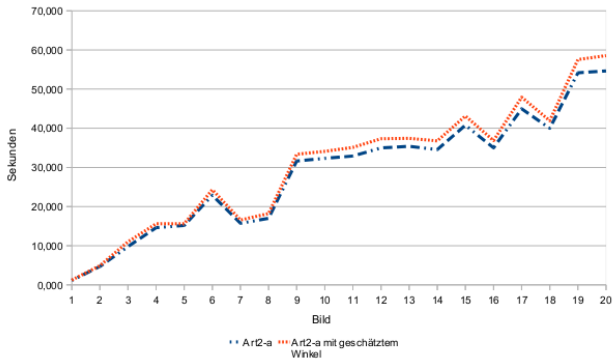

Abbildung 4: Zeitmessung für $\rho = 0.98$

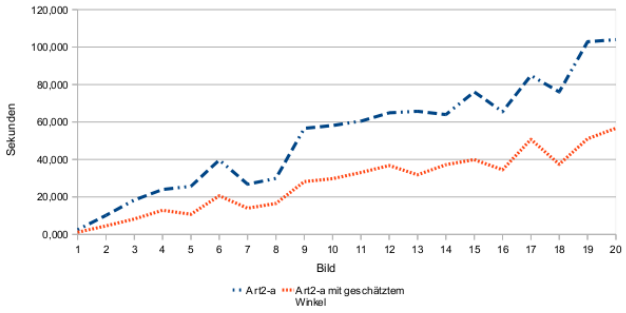

Abbildung 5: Zeitmessung für $\rho = 0.999$

- Es wird geprüft, ob die Abschätzung durch die Hinzunahme weiterer Daten verbessert werden kann und somit eine weitere Beschleunigung erzielt wird. Dafür kann man, um das Problem der zu geringen Präzision der Abschätzung bei kleinerem Vigilance Parameter zu umgehen, die Vektoren teilen und die Abschätzung wie in [7] aus den Teilsegmenten der Vektoren zusammensetzen. Dafür bräuchte man aber auch die Summe der Quadrate, da die Teilsegmente der Vektoren keine Einheitsvektoren mehr sind. Deshalb wird es sich noch zeigen, ob der Gewinn an Präzision durch eine Aufteilung den größeren Aufwand durch Berechnung und Speicherung weiterer Werte rechtfertigt. Des Weiteren soll damit überprüft werden, ob die Abschätzung auch für kleinere Vigilance Werte verwendet werden kann.

- Es wird überprüft, wie groß die Auswirkungen der vorgestellten Verfahren bei einer parallelen Berechnung des Gewinnerknotens sind. Des Weiteren wird das Verfahren auf größeren Datenmengen getestet, um zu überprüfen, ob eine weitere Beschleunigung nötig ist, damit man das Verfahren im Live Betrieb verwenden kann.

- Die Verwendung der Abschätzung zum Indexieren soll getestet und mit anderen Indexierungsverfahren verglichen werden, um ihren Nutzen besser bewerten zu können. Aber vor allem ihre Auswirkungen auf das Art2-a System im parallelisierten Betrieb sind noch offen und werden überprüft.

- Danach werden wir die Analyseeinheit konzipieren. Dafür wird als erstes überprüft, welche Daten man für ein fortlaufendes Lernen braucht, um einem Objekt keine falschen neuen Kategorien zuzuweisen oder richtige Kategorien zu entfernen. Danach soll ein geeignetes neuronales Netz aufgebaut werden, um damit die Zuordnung der Kategorien zu den Objekten durchführen zu können. Das Netz muss dann an die vorher erhobenen Daten angepasst werden, um die Präzision des Netzes zu erhöhen. Abschließend wird das Verfahren dann gegen andere populäre Verfahren getestet.

7. REFERENZEN

[1] G. A. Carpenter and S. Grossberg. Art 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26(23):4919–4930, 1987.

[2] G. A. Carpenter, S. Grossberg, and D. B. Rosen. Art 2-a: an adaptive resonance algorithm for rapid category learning and recognition. In *Neural Networks*, volume 4, pages 493–504, 1991.

[3] S.-C. Chuang, Y.-Y. Xu, H. C. Fu, and H.-C. Huang. A multiple-instance neural networks based image content retrieval system. In *Proceedings of the First International Conference on Innovative Computing, Information and Control*, volume 2, pages 412–415, 2006.

[4] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories, 2004. CVPR 2004, Workshop on Generative-Model Based Vision.

[5] S. Grossberg. Adaptive pattern classification and universal recording: II. Feedback, expectation, olfaction, illusions. *Biological Cybernetics*, 23:187–202, 1976.

[6] B. Jyothi and D. Shanker. Neural network approach for image retrieval based on preference elicitation. *International Journal on Computer Science and Engineering*, 2(4):934–941, 2010.

[7] Y. Kim, C.-W. Chung, S.-L. Lee, and D.-H. Kim. Distance approximation techniques to reduce the dimensionality for multimedia databases. *Knowledge and Information Systems*, 2010.

[8] L. Koskela, , J. T. Laaksonen, J. M. Koskela, and E. Oja. Picsom a framework for content-based image database retrieval using self-organizing maps. In *In 11th Scandinavian Conference on Image Analysis*, pages 151–156, 1999.

[9] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, 1999.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[11] K. N. S., Čabarkapa Slobodan K., Z. G. J., and R. B. D. Implementation of neural network in cbir systems with relevance feedback. *Journal of Automatic Control*, 16:41–45, 2006.

[12] H.-J. Wang and C.-Y. Chang. Semantic real-world image classification for image retrieval with fuzzy-art neural network. *Neural Computing and Applications*, 21(8):2137–2151, 2012.