

The Construction of a Boolean Competitive Neural Network Using Ideas from Immunology

Ausarbeitung im Rahmen des Seminars

„Anwendungen und Algorithmen

basierend auf Prinzipien eines künstlichen Immunsystems“

Technische Universität Darmstadt

Fachbereich Informatik

Fachgebiet Sicherheit in der Informationstechnik

Fachgebietsleitung: Prof. Dr. Claudia Eckert

Seminarbetreuer: Thomas Stibor

Carsten Längsfeld

Juni 2005

Inhaltsverzeichnis

ABBILDUNGSVERZEICHNIS	3
1. EINLEITUNG	4
2. VOM GEHIRN ZUM KÜNSTLICHEN NEURONALEN NETZWERK.....	4
3. EINFÜHRUNG IN KÜNSTLICHE NEURONALE NETZE.....	6
<i>3.1 Das Neuron.....</i>	<i>6</i>
<i>3.2 Die Netzwerktopologie</i>	<i>7</i>
<i>3.3 Lernregeln</i>	<i>8</i>
4. GRUNDLAGEN DES IMMUNSYSTEMS.....	9
<i>4.1. Ebenen der Immunabwehr.....</i>	<i>9</i>
<i>4.2 Immunzellen.....</i>	<i>10</i>
<i>4.3 Prinzipielle Arbeitsweise des spezifischen Immunsystems</i>	<i>11</i>
5. KLONALE SELEKTION	12
6. DER SHAPE-SPACE FORMALISMUS.....	14
7. DAS ANTIBODY-NETWORK (ABNET).....	16
<i>7.1 Das Modell des ABNET.....</i>	<i>16</i>
<i>7.2 Netzwerkvergrößerung (network growing)</i>	<i>18</i>
<i>7.3 Netzwerkverkleinerung (network pruning).....</i>	<i>18</i>
<i>7.4 Aktualisierung der Gewichtsvektoren (weights update).....</i>	<i>18</i>
<i>7.5 Konvergenz des Algorithmus.....</i>	<i>19</i>
<i>7.6 Beispielanwendung ANIMALS-Datensatz</i>	<i>20</i>
<i>7.7 Performance und Evaluation.....</i>	<i>22</i>
8. ZUSAMMENFASSUNG	23
9. LITERATUR	24

Abbildungsverzeichnis

ABBILDUNG 1 DAS NEURON	5
ABBILDUNG 2 KÜNSTLICHES NEURON	6
ABBILDUNG 3 BEISPIELE FÜR AKTIVIERUNGSFUNKTIONEN	7
ABBILDUNG 4 ZELLEN UND FLÜSSIGKEITEN DES IMMUNSYSTEMS	11
ABBILDUNG 5 FUNKTIONSWEISE DES ADAPTIVEN IMMUNSYSTEMS...	12
ABBILDUNG 6 DAS PRINZIP DER KLONALEN SELEKTION	13
ABBILDUNG 7 SHAPE-SPACE	14
ABBILDUNG 8 DISTANZFUNKTIONEN	15
ABBILDUNG 9 BEISPIEL HAMMING-DISTANZ MIT $K=2$, $L=8$	15
ABBILDUNG 10 EINFLUSS VON E AUF DAS BINDUNGSVERHALTEN.....	16
ABBILDUNG 11 LERNALGORITHMUS DES ABNET	17
ABBILDUNG 12 BEISPIEL NETZWERKVERGRÖßERUNG	18
ABBILDUNG 13 BSP. GEWICHTSUPDATE MIT MUTATIONSRATE 2	19
ABBILDUNG 14 ANTIGEN COVERAGE.....	19
ABBILDUNG 15 MINIMALE ANZAHL ITERATIONEN F. KONVERGENZ DES ABNET.....	20
ABBILDUNG 16 ANIMALS-DATENSATZ.....	21
ABBILDUNG 17 RESULTATE DES ABNET FÜR D. ANIMALS- DATENSATZ	21

1. Einleitung

In dieser Arbeit wird das, von de Castro, Von Zuben, und de Deus entworfene AntiBody-Network (*ABNET*) vorgestellt. Dabei handelt es sich um ein konkurrierendes, neuronales Netzwerk, dessen Lernalgorithmus Prinzipien des Immunsystems verwendet. Zu Beginn werden dem Leser zunächst die, für die Darstellung des ABNET notwendigen Begriffe Neuronales Netzwerk, Immunsystem, Theorie der Klonalen Selektion und Shape-Space-Formalismus in kurzer Form präsentiert. Im Folgenden wird das Modell und der Lernalgorithmus des ABNET beschrieben und anhand eines Beispiels die Anwendung des Netzwerks demonstriert. Abschließend folgt ein Vergleich mit anderen Netzwerkarchitekturen und eine Evaluierung des ABNET.

2. Vom Gehirn zum Künstlichen Neuronalen Netzwerk

Das Gehirn ist in der Lage komplexe Aufgaben wie z.B. Mustererkennung, Wahrnehmung, Bewegungsplanung und -kontrolle deutlich effizienter, als viele der heute verfügbaren Computersysteme zu lösen. Das Gehirn bewältigt beispielsweise visuelle Wahrnehmungsaufgaben, wie das Erkennen eines bekannten Gesichts in einer unbekannten Umgebung in etwa 100-200 ms, während die Lösung ähnlicher, einfacherer Aufgaben auf modernen Rechner mitunter deutlich länger dauert oder eventuell gar nicht lösbar ist. Ein weiteres Beispiel, diesmal aus dem Tierreich zeigt, dass die Leistungsfähigkeit des Gehirns nicht unbedingt etwas mit seiner Größe zu tun hat. Die Fledermaus verwendet Sonar als ein aktives Echo-Lokalisations-System. Obwohl sie nur ein Gehirn von etwa der Größe einer Pflaume hat, ist sie in der Lage Informationen wie Größe, Entfernung, relative Geschwindigkeit, Flugwinkel und Höhe ihres Ziels mit hoher Genauigkeit in Echtzeit zu bestimmen. Es stellt sich also die Frage wie das Gehirn zu diesen enormen Leistungen fähig ist? Die grundlegende Idee zum Verständnis des Aufbaus des Gehirns wurde 1911 vom Spanier Ramón y Cajal entwickelt, der Nervenzellen (sog. *Neuronen*) als strukturelle Elemente des Gehirns identifizierte. Ein Neuron besteht aus dem Zellkörper (*Soma*), einem oder mehreren Eingängen, die der Reizaufnahme dienen (*Dendriten*) und einem Ausgang (*Axon*), der Informationen zu anderen Neuronen im zentralen Nervensystem (*ZNS*) weiterleitet (siehe Abb. 1). Die eigentliche Verbindungsstelle zwischen dem Ende eines Axons und einem anderen Neuron wird als *Synapse* bezeichnet, die ein weiteres wichtiges funktionelles und strukturelles Element des Gehirns darstellt. In den Synapsen findet eine Gewichtung der Eingangssignale statt, dass heißt die Signale werden entweder geschwächt

(*inhibitorisch*) oder verstärkt (*exzitatorisch*) an das empfangene Neuron weitergeleitet. Übersteigt die Summe der am Neuron ankommenden Signale einen bestimmten Schwellwert, wird ein Aktionspotential ausgelöst und das Neuron aktiviert.

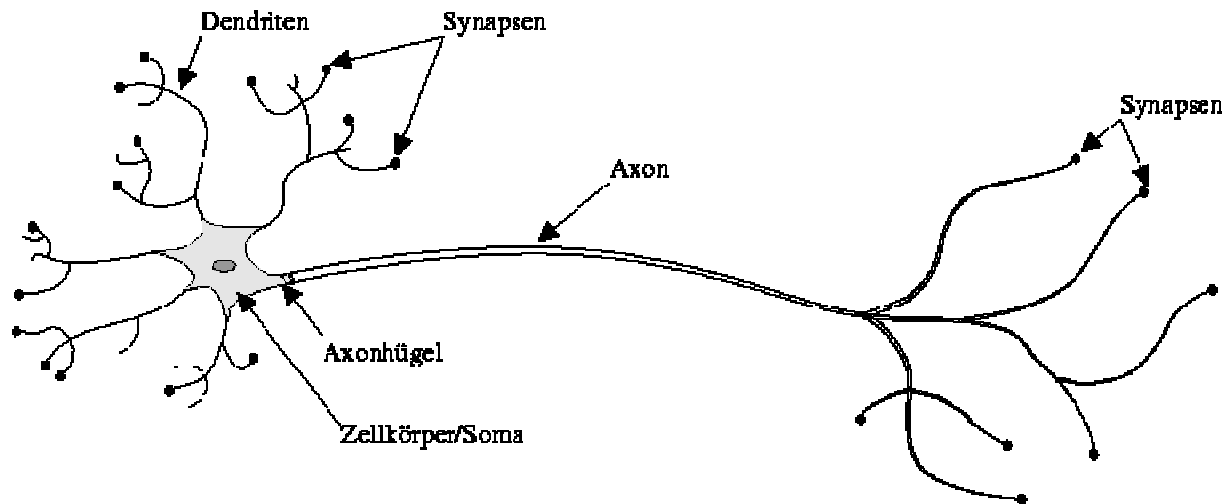


Abbildung 1 Das Neuron

Die Neuronen sind die informationsverarbeitenden Elemente des Gehirns. Obwohl die Verarbeitungsgeschwindigkeit eines einzelnen Neurons deutlich langsamer ist, als beispielsweise die eines digitalen, logischen Gatters, kompensiert das Gehirn dies mit enormer paralleler Verarbeitung durch hochgradige Vernetzung der einzelnen Einheiten. Um sich ein Bild von den Ausmaßen dieser Struktur zu machen: das Gehirn besitzt ca. 10 Billionen Neuronen und etwa 60 Trillionen Synapsen. Demzufolge kann man das Gehirn als einen „hochkomplexen, nichtlinearen, parallel arbeitenden Computer“ bezeichnen (siehe [3]). Das Gehirn ist eine formbare Struktur (*Plastizität*), die in der Lage ist, sich an wechselnde Umgebungen und Situationen anzupassen. Dies geschieht nach heutiger Auffassung über das Erzeugen und die Modifikation von Synapsen. Man geht davon aus, dass all unser Wissen und die Erfahrungen, die wir im Verlauf unseres Lebens machen, über die Verbindungsstärken der Synapsen, die auch als Synapsengewichte bezeichnet werden, gespeichert wird. Wenn man nun versucht, diese Grundprinzipien auf Computersysteme zu übertragen, um die Art, wie das Gehirn bestimmte Aufgaben löst zu modellieren, kommt man zum Begriff der künstlichen, neuronalen Netze.

3. Einführung in Künstliche Neuronale Netze

Der Bereich der Künstlichen Neuronen Netze ist ein sehr komplexes Thema und daher ist die folgende Darstellung nur als Einführung zu verstehen. Für vertiefende Informationen sei auf [4] und insbesondere auf [3] verwiesen.

Ein künstliches neuronales Netzwerk (*KNN*) ist ein System, dass sowohl in Hardware, als auch in Software implementiert sein kann und das versucht die Verarbeitungsweise des Gehirns für ein bestimmtes Problem (wie z.B. Mustererkennung) zu modellieren. Die grundlegenden Elemente eines KNN sind die einzelnen **Neuronen**, die **Netzwerktopologie**, die die Synapsenverbindungen zwischen den Neuronen imitieren soll und die im Netzwerk verwendete **Lernregel**. Eine der wichtigsten Eigenschaften von KNN ist ihre Fähigkeit gelernte Beispiele zu verallgemeinern und darauf basierend neue Sachverhalte zu prognostizieren. In der Regel wird das KNN mit einer Menge von Testdaten trainiert (*Trainingsamples*), um später auf einer unbekannten Datenmenge arbeiten zu können.

3.1 Das Neuron

Jedes Neuron k verfügt über eine Reihe von Eingangssignalen x_i , die über die mit w_i gewichteten Verbindungen am Neuron angelegt werden, sowie einen Ausgabewert y_k . Die **Propagierungsfunktion** gibt an, wie die anliegenden Werte verarbeitet werden (vergleiche Abb. 2). Beispiele für Propagierungsfunktionen sind Summe, Produkt, Minimal oder Maximalwert der gewichteten Eingabe. Oft wird die gewichtete Summe verwendet. Der aktuelle Zustand eines Neurons wird durch die **Aktivierungsfunktion** beschrieben. Sie bestimmt, welche Ausgabe das Neuron, in Abhängigkeit von den anliegenden Daten, liefert. Standardtypen von Aktivierungsfunktionen sind Schwellwert-, (teilweise) lineare und sigmoide Funktionen (s. Abb 3).

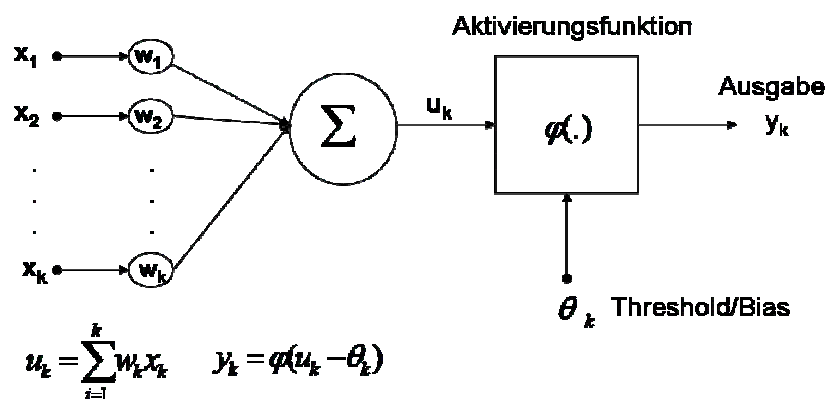


Abbildung 2 Künstliches Neuron

Die **Ausgabefunktion** bildet den aktuellen Zustand des Neurons auf einen gewünschten Wertebereich ab. Meistens wird diese Funktion nicht gesondert aufgeführt, sondern als Bestandteil der Aktivierungsfunktion aufgefasst. Typischerweise wird ein Ausgabeintervall von $[0, 1]$ oder $[-1, 1]$ verwendet. Das Modell des Neuron beinhaltet zusätzlich noch einen Wert, der die Eingabewerte der Aktivierungsfunktion um einen festen Betrag verringert oder erhöht. Im negativen Fall spricht man vom **Threshold**, im positiven Fall vom **Bias** (der Bias ist dabei einfach die Negation des Thresholds).

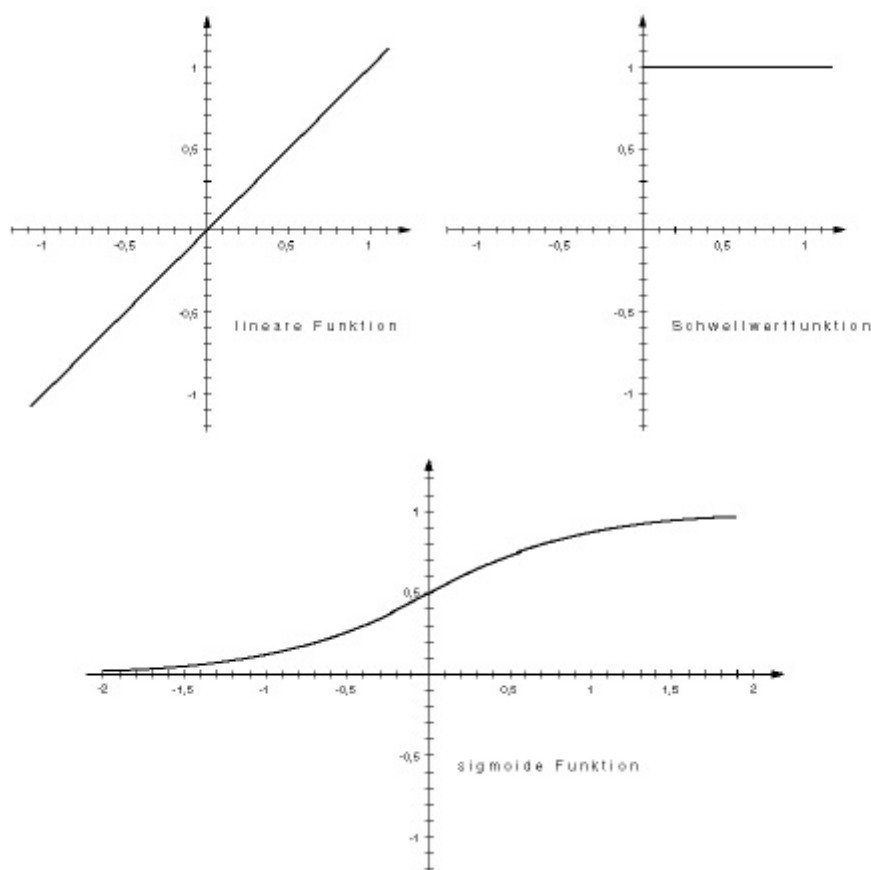


Abbildung 3 Beispiele für Aktivierungsfunktionen

3.2 Die Netzwerktopologie

Wie in Kapitel 1 erläutert, begründet sich die Leistungsfähigkeit des Gehirns Problemstellungen zu lösen, auf der massiven Parallelverarbeitung und der hochgradigen Verdrahtung der einzelnen informationsverarbeitenden Elemente, den Neuronen.

Diese Struktur kann verallgemeinernd als endlicher Graph aufgefasst werden, in dem die Neuronen die Knoten- und die Verbindungen (Synapsen) zwischen ihnen, die Kantenmenge

definieren. Ein neuronales Netz kann in mehrere Neuronenschichten aufgeteilt sein. Die Knotenmenge zerfällt dann in einzelne Partitionen, wobei jedes Neuron zu genau einer Partition (Schicht) gehört.

Man unterscheidet im Allgemeinen zwischen

- Feedforward-Netzen

Die Eingangssignale gehen nur in eine Richtung, von niedrigen zu höheren Neuronenschichten. Es gibt keine Rückkopplung

und

- Feedback-Netzen

Mindestens ein Ausgangssignal wird rückgekoppelt und geht als Eingangssignal in die Berechnungen der Neuronen auf gleicher oder einer niedrigeren Schicht mit ein.

3.3 Lernregeln

Im ersten Kapitel wurde erläutert, dass die Plastizität des Gehirns, die Anpassung an Veränderungen der Umgebung ermöglicht. Dies geschieht durch die Erzeugung und Modifikation von Synapsen, über deren Gewichtung (vermutlich) Erfahrung und Wissen gespeichert werden. Der Lernprozess dem das Gehirn unterworfen ist, wird in Form von Lernregeln auf die künstlichen, neuronalen Netze übertragen. Die Lernregel bestimmt dabei wie die interne Netzrepräsentation der Eingabedaten erzeugt wird und ggf. die Netzwerkstruktur angepasst wird.

Prinzipiell können dabei folgende Änderungen durchgeführt werden:

- Aufbauen und Löschen von Verbindungen
- Aufbauen und Löschen von Neuronen
- Anpassung der Verbindungsgewichte
- Modifikation von Neuronenparametern (Aktivierungsfunktion, Schwellwertänderungen etc.)

Es werden nun drei kurze Beispiele für Lernregeln skizziert.

3.3.1 Delta-Regel

Die von Widrow und Hoff entwickelte Lernregel besagt, dass, wenn es für eine gegebene Eingabe eine Abweichung zwischen der geforderten und der tatsächlichen Ausgabe des Netzwerks gibt, die Verbindungsgewichte angepasst werden, mit dem Ziel den Istwert gegen

den Sollwert konvergieren zu lassen. Die Änderung der Gewichte ist dabei abhängig vom Fehler (also der Differenz zwischen Ist- und Sollwert). Die Delta-Regel ist für einstufige Feedforward-Netze geeignet.

3.3.2 Hebbsche Regel

Die von Donald Hebb Ende der 40er Jahre entwickelte Theorie führt den Lernprozess auf folgendes Prinzip zurück. Sind zwei Neuronen i und j zur gleichen Zeit aktiv und es existiert eine Verbindung zwischen ihnen, dann wird das Gewicht dieser Verbindung in definierter Weise verstärkt.

3.3.3 Konkurrierend (Competitive)

Beim konkurrierenden Lernen kann nur jeweils ein Neuron zu einer Zeit aktiv sein. Die einzelnen Neuronen treten in Wettkampf zueinander, mit dem Ziel das Neuron zu sein, was aktiviert wird. Die Voraussetzung für einen „fairen“ Wettkampf zwischen den Neuronen ist natürlich, dass diese strukturell gleich sind (bis auf zufällig verteilte Synapsengewichte) und jedes Neuron die gleiche Chance hat, den Wettkampf zu gewinnen. In der Regel wird daher auch ein Gewichtungslimit für die Verbindungsgewichte eingeführt. Zudem benötigt man einen Mechanismus der den Wettkampf ermöglicht. Diese Form der Lernregel wird in dem später vorgestellten AntiBody-Network verwendet.

4. Grundlagen des Immunsystems

Die primäre Aufgabe des Immunsystems ist die Erkennung und Bekämpfung von dysfunktionalen, körpereigenen Zellen (*infectious self*) und externen, infektiösen Erregern (*infectious nonself*). Dies können beispielsweise Viren, Bakterien oder Pilze, aber auch entartete körpereigene Krebszellen sein. Die externen Erreger werden auch als *Pathogene* bzw. als *Antigene* bezeichnet. Das Immunsystem arbeitet dabei auf verschiedenen Ebenen, um eventuelle Fremdkörper zu bekämpfen:

4.1. Ebenen der Immunabwehr

- physische Ebene

Auf dieser Ebene treten zum Beispiel die Haut als äußere Barriere oder das Atmungssystem (*respiratorisches System*) als Verteidigungsmechanismen gegen potentielle Angreifer auf, die verhindern sollen, dass ein Erreger überhaupt erst ins Körperinnere gelangt oder ihn möglichst schnell wieder verlässt. Im respiratorischen System fangen sich Eindringlinge z.B. in Nasenhaaren oder Schleim und werden

über Flimmerhärchen aus dem Körper abtransportiert. Auch Niesen und Husten gehören zu den Abwehrmaßnahmen.

- physiologische Ebene

Die im Körper vorherrschenden Bedingungen, wie pH-Wert oder Temperatur stellen eine weitere Hürde für einige Antigene dar. In Speichel, Schweiß oder Tränen befinden sich *Enzyme*, die in der Lage sind Erreger zu zerstören. Ebenso vernichtet die Magensäure einen Großteil der über Nahrungsmittel zugeführten Pathogene.

- unspezifische Immunabwehr

Hierbei handelt es sich um die angeborenen Immunmechanismen, das heißt der Körper ist in der Lage eine beschränkte Anzahl von Antigenen (primär Bakterien) auch bei einer Erstinfektion zu erkennen und zu zerstören. Die Abwehr richtet sich im Gegensatz zur adaptiven Immunabwehr nicht gerichtet gegen einen bestimmten Antigentyp, sondern es wird mehr eine generelle Unterscheidung zwischen Elementen des eigenen Körpers und körperfremden Elementen durchgeführt, wobei alles „Fremde“ soweit möglich direkt bekämpft wird.

- spezifische, adaptive Immunabwehr

Die adaptive Immunabwehr ist im Gegensatz zur unspezifischen Immunabwehr nicht angeboren, sondern wird im Laufe des Lebens erlernt. Wie der Name schon vermuten lässt, bekämpfen die Mechanismen dieses Teilsystems gezielt bestimmte Erregertypen, die auch bei erneuter Infektion wiedererkannt und effektiver bekämpft werden können. Man spricht in diesem Zusammenhang von einer ***Gedächtnisfunktion des Immunsystems***.

4.2 Immunzellen

Das Immunsystem besteht aus einer Vielzahl unterschiedlicher Zellen (s. Abb. 4). Die für die unspezifische Immunantwort relevantesten Zellen sind die ***Phagozyten*** und das ***Komplementsystem***. Die Phagozyten sind die Zellfresser des Immunsystems. Ihre Aufgabe besteht in der Aufnahme und Verdauung von Mikroorganismen und Antigenen. Zudem besitzen einige Phagozyten (z.B. *Makrophagen*) die Fähigkeit der Antigenpräsentation (*APC*). Sie spielen zu Beginn der Immunreaktion eine wichtige Rolle. Das Komplementsystem ist ein Komplex aus Plasma Proteinen, der die Zellwand eines Eindringlings beschädigen kann, um die Zelle zu zerstören oder das Antigen für die Zerstörung durch Phagozyten markieren kann. Werden körpereigene Strukturen vom Komplementsystem umlagert, schütten diese, um den eigenen Organismus zu schützen ein Protein aus, welches die Komplementmoleküle

deaktiviert. Die wichtigsten Zellen des adaptiven Immunsystems sind die **Lymphozyten**, die sich in **B- und T-Zellen** unterteilen. Diese Zellen besitzen an Ihrer Oberfläche hochspezifische Rezeptoren für bestimmte Antigene. Die Hauptaufgabe der B-Zellen ist die Produktion und Ausschüttung von Antikörpern (**Ab**), die den Hauptteil der spezifischen Immunreaktion ausmachen. Jede B-Zelle produziert einen bestimmten Antikörper und differenziert bei Aktivierung in Plasmazellen (die die eigentlichen Antikörperproduzenten sind). Einige B-Zellen werden zu Gedächtniszellen, um bei erneuter Infektion durch ein Antigen mit dem passenden Antikörper schneller und effektiver zu reagieren.

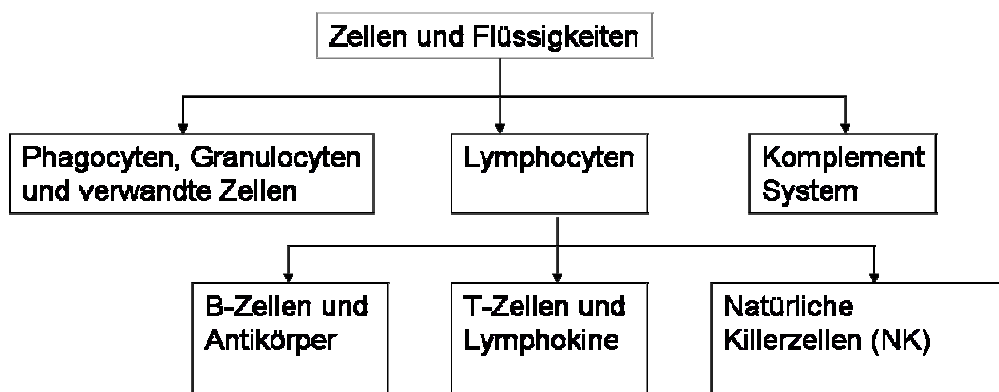


Abbildung 4 Zellen und Flüssigkeiten des Immunsystems

T-Zellen kontrollieren die Aktionen anderer Immunzellen und sind in der Lage infizierte Zellen anzugreifen. Sie arbeiten primär mit starken, chemischen Botenstoffen, den sog. *Lymphokinen*. Man unterscheidet drei Klassen von T-Zellen. Die **T-Helfer-Zelle** (oder *T-Regulator-Zelle*) aktiviert anderer Immunzellen, wie z.B. B- und andere T-Zellen. Der entsprechende Antagonist ist die **T-Suppressor-Zelle**, die die Aktionen anderer Immunzellen unterdrückt und eine wichtige Rolle beim Beenden der Immunreaktion spielt. **Die T-Killer-Zelle** hat die Aufgabe spezifische Antigene oder infizierte Zellen zu vernichten.

4.3 Prinzipielle Arbeitsweise des spezifischen Immunsystems

Die Arbeitsweise des adaptiven Immunsystems wird hier nur vereinfacht dargestellt. Für detaillierte Informationen wird auf [5] verwiesen (siehe auch Abb. 5).

1. Antigenpräsentierende Zellen (APC, z.B. Makrophagen) durchqueren den Körper, verspeisen Antigene die sie finden und fragmentieren sie. Aus den Fragmenten entstehen sog. **MHC- Moleküle** (*major histocompatibility complex*), die an der Oberfläche der APC präsentiert werden.

2. Eine T-Helfer-Zelle mit passendem Rezeptor für das präsentierte MHC-Molekül kann nun an der Makrophage andocken und wird dadurch aktiviert.
3. Die aktivierte Zelle teilt sich und schüttet Chemikalien (Lymphokine) aus, die wiederum B-Zellen anlocken.
4. Eine B-Zelle mit passendem Rezeptor reagiert auf die ausgeschütteten Chemikalien und kann direkt an das Antigen andocken (in diesem Fall ohne MHC-Molekül), fragmentiert es und präsentiert es ebenfalls an der Oberfläche. Die aktivierte T-Helferzelle dockt an das präsentierte Antigen der B-Zelle an und aktiviert diese.
5. Die aktivierte B-Zelle teilt sich in Plasmazellen, die die Antikörper (Kopien des Rezeptors) ausschütten.
6. Die Antikörper können nun die Antigene neutralisieren und für die Phagozyten markieren.

Dieser Ablauf ist in folgender Grafik schematisch dargestellt.

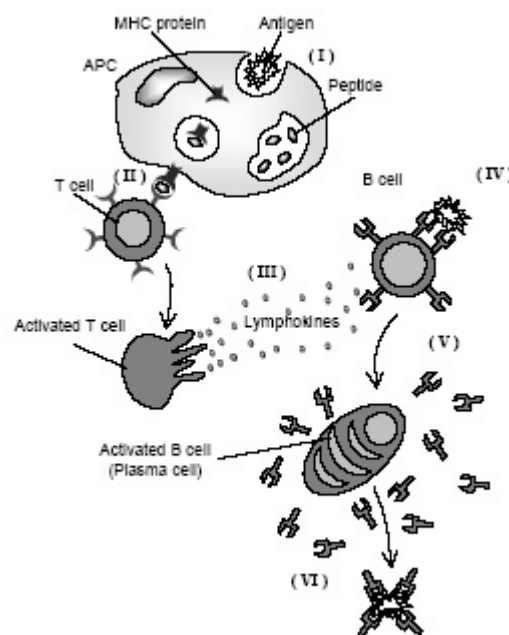


Abbildung 5 Funktionsweise des adaptiven Immunsystems

5. Klonale Selektion

Ein weiterer, für die Darstellung des ABNET wichtiger Begriff, ist die Theorie der Klonalen Selektion. Diese Theorie versucht die grundlegenden Eigenschaften der Immunantwort auf einen Antigen-Stimulus zu beschreiben (vgl. Abb. 6).

Die wichtigsten Merkmale dieser Theorie, die in das Design des ABNET mit aufgenommen wurden sind:

- Proliferation/Differentiation bei Kontakt mit Antigenen
Wird der Körper einem Antigen-Stimulus ausgesetzt, werden Antikörper mit hoher Affinität für das betreffende Antigen selektiert und kloniert. Dieser Vorgang wird als **Proliferation** bezeichnet. Die Klone **differenzieren** dann entweder in Gedächtniszellen oder in Plasmazellen.
- Klone sind Kopien der Eltern
Die neu generierten Zellen sind Kopien der Eltern, die einem Mutationsmechanismus unterworfen sind, den man als *somatic hypermutation* bezeichnet. Diese natürliche Anpassung der Zelle dient dazu, die Affinität mit dem gegebenen Antigen zu erhöhen, damit bei einem erneuten Befall des Körpers durch dieses Antigen, der Erreger effektiver bekämpft werden kann (**Affinity Maturation**).
- Die Elimination von nutzlosen Zellen, die nicht auf Antigene reagieren oder sogar selbstreagierend sind.

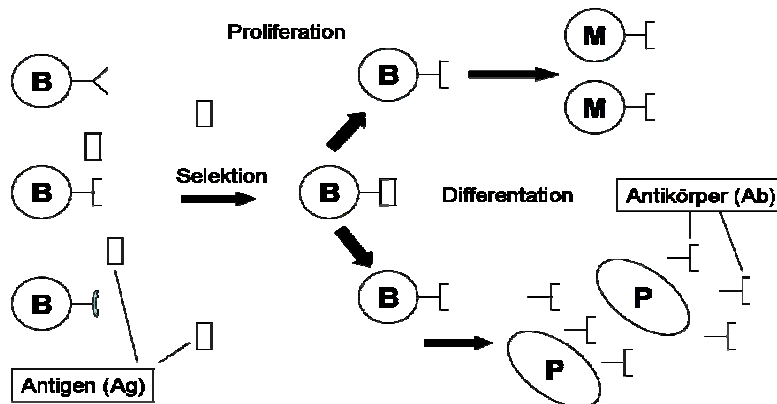


Abbildung 6 Das Prinzip der Klonalen Selektion

Da nur Zellen mit hoher Stimulation selektiert und kopiert werden, können Parallelen zu der natürlichen Selektion gezogen werden (*survival of the fittest*), da nur die „besten“ Zellen in die nächste Generation übernommen werden. Hohe Stimulation bedeutet hier, dass der Antikörper eine hohe Affinität zum betreffenden Antigen hat und einer ausreichenden Konzentration dieser Antigene ausgesetzt ist.

6. Der Shape-Space Formalismus

Um die Interaktionen zwischen den Antikörpern und den Antigenen quantitativ beschreiben zu können, wurde 1979 von Perelson & Oster das Konzept des **Shape-Space** eingeführt. Die Grundidee des Shape-Space ist, dass die Affinität zwischen Rezeptor- und Self- oder Nonsel-Molekül durch verschiedene Interaktionen in Abhängigkeit von elektrostatischen, chemischen und geometrischen Eigenschaften beschrieben werden kann (z.B. elektrostatische Ladung, Wasserstoffbindungen, van der Waals-Interaktionen etc.). Da sich die Moleküle bei einer Bindung über einen großen Bereich annähern müssen, geht man in diesem Modell davon aus, dass sie in größeren Bereichen zueinander **komplementär** sein müssen.

Die genannten Eigenschaften werden als allgemeine Form eines Moleküls oder **Generalized Shape** bezeichnet. Die allgemeine Form eines Antikörpers (oder eines Antigens) kann nun durch L reellwertige Parameter beschrieben und als Punkt in einem L -dimensionalen Raum, dem Shape-Space, dargestellt werden. Die Anzahl der Punkte im Shape-Space ist durch die Größe des Antikörper-Repertoires bestimmt. Die L Parameter haben endliche Wertebereiche und man geht davon aus, dass alle Punkte in einem endlichen Volumen V des Shape-Space liegen (vgl. Abb. 7). Da die Affinität zwischen den Molekülen über ihre Komplementarität beschrieben wird, bindet ein Antikörper ein Antigen genau dann, wenn das Komplement des Antigens in einem Bereich V_ϵ , der **Recognition Region**, um den Antikörper liegt. Der Parameter ϵ , der als **Affinity Threshold** bezeichnet wird, gibt hierbei an, wie weit das Komplement des Antigens vom perfekten Match zum Antikörper entfernt sein darf, damit es trotzdem noch zu einer Bindung kommt.

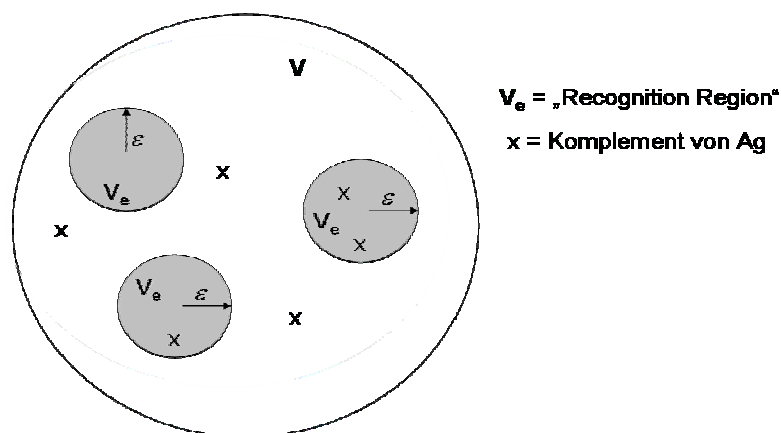


Abbildung 7 Shape-Space

Werden ein Antigen $ag = \langle ag_1, ag_2, \dots, ag_L \rangle$ und ein Antikörper $ab = \langle ab_1, ab_2, \dots, ab_L \rangle$ als L-dimensionale Vektoren im Shape-Space betrachtet, kann man ihre Affinität über ihre Distanz abschätzen. Hierfür kann ein beliebiges Distanzmaß, wie z.B. die *Euklidische Distanz* oder die *Manhattan-Distanz* (s. Abb. 8) verwendet werden. Je nachdem welches Distanzmaß man verwendet, wird der zugrunde liegende Shape-Space als *Euklidischer*- oder *Manhattan-Shape-Space* bezeichnet. In der Regel wird allerdings die euklidische Distanz verwendet.

$$\text{Euklidische-Distanz: } \sqrt{\sum_{i=1}^L (ab_i - ag_i)^2}$$

$$\text{Manhattan-Distanz: } \sum_{i=1}^L |ab_i - ag_i|$$

Abbildung 8 Distanzfunktionen

Eine alternative Darstellung zum Euklidischen Shape-Space ist der *Hamming-Shape-Space*. Im Hamming-Shape-Space werden Antikörper und Antigene nicht mehr durch reelle Werte beschrieben, sondern als Sequenzen der Länge L mit Symbolen aus einem Alphabet der Größe k dargestellt. Ein einfaches Beispiel, dass auch im restlichen Teil dieser Arbeit verwendet wird, ist der binäre Hamming-Shape-Space, in dem das Alphabet nur aus den Symbolen 0 und 1 besteht und die Moleküle als Bitstrings beschrieben werden. Abbildung 9 zeigt ein Beispiel für k=2 und L=8. Die Affinität (Distanz) zwischen den Molekülen kann dann relativ einfach mit der XOR-Funktion bestimmt werden, in dem die Anzahl komplementärer Stellen aufsummiert wird.

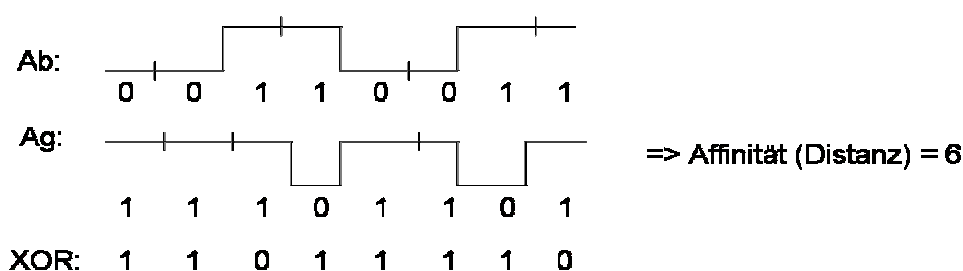


Abbildung 9 Beispiel Hamming-Distanz mit k=2, L=8

Das Bindungsverhalten von Antigen und Antikörper wird durch den bereits erwähnten Affinity Threshold ϵ beeinflusst. Hier gibt ϵ an, um wie viele Bits die Strings voneinander abweichen dürfen, so dass es trotzdem zu einer Bindung kommt. Ist $\epsilon=0$ müssen die Strings in allen Bits komplementär sein. Im allgemeinen Fall kommt es, bei einer Stringlänge von L, zu einer Bindung zwischen einem Antigen ag und einem Antikörper ab , wenn die Distanz zwischen beiden größer oder gleich $L-\epsilon$ ist (vgl. Abb. 10).

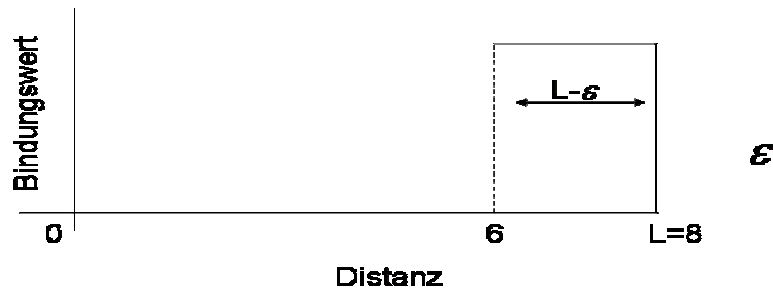


Abbildung 10 Einfluss von ε auf das Bindungsverhalten

7. Das AntiBody-NETwork (ABNET)

Das ABNET ist ein binäres, konkurrierendes neuronales Netzwerk, dessen Lernalgorithmus auf den Prinzipien der klonalen Selektion und der Affinity Maturation basiert. Um ein gegebenes Problem zu lösen, werden die dem Netzwerk präsentierten Input-Patterns als Antigen-Population aufgefasst, mit dem Ziel, ein minimales Antikörper-Repertoire zu generieren, das alle präsentierten Antigene erkennt.

Die Haupteigenschaften des ABNET sind:

- dynamische Anpassung der Netzwerkarchitektur (network growing/pruning)
- binäre Gewichte und Darstellung von Antigenen/Antikörpern im binären Hamming-Shape-Space
- konkurrierendes Netzwerk, mit unüberwachtem Lernprozess, basierend auf einem direkten Mutationsmechanismus

Die einzelnen Merkmale werden im Folgenden detailliert beschrieben.

7.1 Das Modell des ABNET

Das ABNET verwendet den binären Hamming-Shape-Space, das heißt Antigene und Antikörper werden als binäre Strings gleicher Länge repräsentiert, daher auch die Bezeichnung *binäres* oder *boolean* Netzwerk. Sollte das zu lösende Problem noch nicht in binärer Form zur Verfügung stehen, müssen die Eingabedaten gegebenenfalls zuerst konvertiert werden. Ein Antikörper AB_k wird im ABNET durch einen Gewichtsvektor w_k repräsentiert, der die Netzwerkeingaben mit einer einzelnen Ausgabeeinheit k (einem Neuron) verbindet. Alle Neuronen sind lineare Verarbeitungselemente, das heißt, sie berechnen die gewichtete Summe ihrer Eingaben (als Linear-Kombination von Eingabe- und Gewichtsvektoren, siehe Abschnitt 3). Die initiale Architektur des Netzwerks besteht in der

Regel aus einem einzigen Neuron, dessen Gewichtsvektor das zunächst einzige Element des Antikörper-Repertoires darstellt. Der Lernprozess des ABNET wird durch eine Reihe von Parametern bestimmt. τ_j gibt die Konzentration (Anzahl) an Antigenen an, für die ein Antikörper j zuständig ist bzw. die er erkennt. v_a ist ein Bezeichner für den Antikörper, der die höchste Affinität zum Antigen AG_a besitzt. Hat also beispielsweise der Antikörper AB_7 die höchste Affinität zum Antigen AG_9 , dann ist $v_9=7$. Der Parameter α gibt die Mutationsrate, in diesem Fall die Anzahl zu mutierender Bits an (siehe Abschnitt Gewichtsanzpassung). β beschreibt die Anzahl an Iterationsschritten, bis geprüft wird, ob die Netzwerkstruktur geändert werden muss und der Parameter ε gibt, wie zuvor beschrieben, den Affinity Threshold und somit die erlaubte Abweichung vom perfekten Match zwischen Antikörper und Antigen an. Da sowohl Antikörper, als auch Antigene in binärer Form angegeben werden, wird die Hammingdistanz als Distanzmaß verwendet. Der Antikörper k mit der höchsten Affinität zu einem gegebenen Antigen ag ist demnach der, mit der größten Hammingdistanz zu ag .

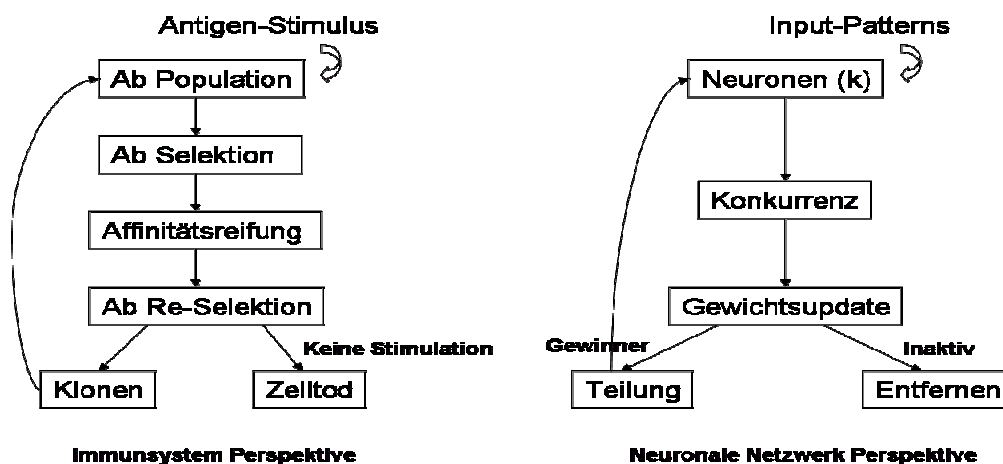


Abbildung 11 Lernalgorithmus des ABNET

Der konkurrierende Lernalgorithmus des ABNET kann wie folgt beschrieben werden:

1. Dem Netzwerk werden alle Input-Patterns Ag_i präsentiert
2. Für jedes Ag_i wird der Antikörper Ab_k mit der höchsten Affinität bestimmt
3. Das Konzentrationslevel τ_k für Ab_k wird erhöht und das Label $v_i=k$ gesetzt
4. Dann wird der Gewichtsvektor w_k aktualisiert
5. Nach β Schritten wird in Abhängigkeit von τ und ε überprüft, ob die Netzwerkstruktur angepasst werden muss

Das Ziel des Algorithmus ist es ein minimales Antikörper-Netzwerk, mit maximaler Abdeckung des Antigenrepertoires zu generieren (s. Abb. 11).

7.2 Netzwerkvergrößerung (network growing)

Die Netzwerkvergrößerung simuliert die Selektion und Reproduktion der am meisten stimulierten Zelle, basierend auf dem Prinzip der klonalen Selektion. Im ABNET ist das gerade der Antikörper, mit dem höchsten Konzentrationslevel, der also die meisten Antigene erkennt. Bei der Vervielfältigung wird immer nur genau ein Klon erzeugt. Alle Zellen Ab_j , für die $\tau_j > 1$ gilt, sind potentielle Kandidaten für die Klonierung. Gibt es keine solche Zelle, bleibt das Netzwerk unverändert, ansonsten wählt man die Zelle AB_s , mit der höchsten Antigenkonzentration. Ist nun das Konzentrationsniveau τ_s größer als ε wird die Zelle vervielfältigt, ansonsten bleibt das Netzwerk unverändert. Die Gewichte der neuen Zelle sind das exakte Komplement des Antigens mit der niedrigsten Affinität zu Ab_s (s. Abb. 12). Dies soll bewirken, dass das erzeugte Netzwerk zu allen Antigenen maximale Affinität hat.

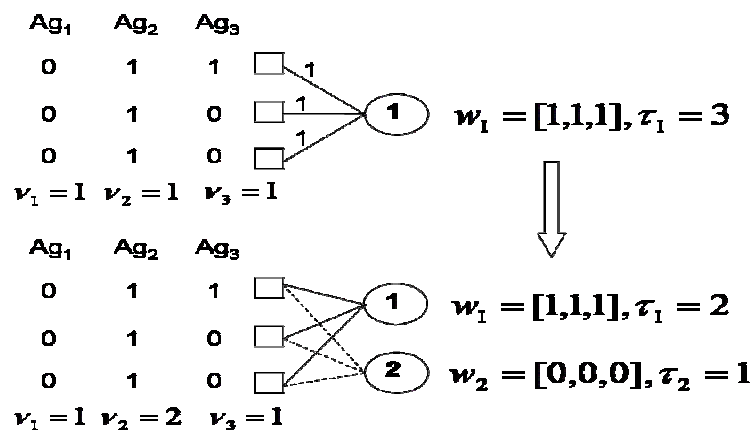


Abbildung 12 Beispiel Netzwerkvergrößerung

7.3 Netzwerkverkleinerung (network pruning)

Die Netzwerkverkleinerung simuliert den Prozess des Zelltods (*Apoptosis*). Wird nach β Schritten ein Antikörper von keinem Antigen stimuliert und ist somit sein Konzentrationslevel gleich null, dann wird diese Zelle aus dem Netzwerk entfernt. Um die Möglichkeit der Redefinition des Verbindungsschemas nach einer Netzwerkverkleinerung zu ermöglichen, wird die Lernrate, in diesem Fall die Mutationsrate α , auf ihren initialen Wert zurückgesetzt.

7.4 Aktualisierung der Gewichtsvektoren (weights update)

Die Aktualisierung der Gewichte versucht das Konzept der Affinitätsreifung (Affinity Maturation), also den natürlichen Mutationsvorgang des Immunsystems, umzusetzen.

Gedächtniszellen, die erneut einem bestimmten Erreger ausgesetzt werden, haben im Durchschnitt eine höhere Affinität, als Zellen, die zum ersten Mal auf dieses Antigen treffen. Im ABNET wird die Mutation durch die Hypermutationsrate $\alpha \in \mathbb{Z}_0^+$ bestimmt. α gibt die Anzahl der zu ändernden Bits im Antibody-String an. Kandidaten für Mutation sind Stellen, an denen Antikörper und Antigen nicht komplementär sind (s. Abb. 13). Nach einer bestimmten Anzahl an Iterationen x wird die Mutationsrate um eins erniedrigt, solange bis α null wird und keine Änderungen mehr stattfinden. Zellen mit hoher oder maximaler Affinität werden nicht mutiert.

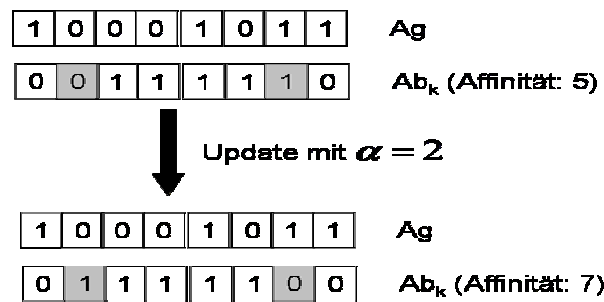


Abbildung 13 Bsp. Gewichtsupdate mit Mutationsrate 2

In der Beschreibung des ABNET (s. [1]) bleibt allerdings unklar, wie der Updateprozess für einen Antikörper, der für mehr als ein Antigen zuständig ist, abläuft. Würde der Mutationsvorgang für jedes dieser Antigene durchgeführt werden, könnte dies, für sehr unterschiedliche Antigene, zu kontraproduktiven Ergebnissen führen.

7.5 Konvergenz des Algorithmus

Der Lernalgorithmus des ABNET prüft alle β Iterationen, ob Netzwerkanpassungen durchgeführt werden müssen, dass heißt, ob eine neue Einheit hinzugefügt oder entfernt werden muss. Die Entwickler des ABNET haben durch empirische Analyse festgestellt, dass dieser Parameter die Lerngeschwindigkeit des Netzwerks maßgeblich beeinflusst und es stellt sich die Frage, wann der Algorithmus terminiert? Die Anzahl verschiedener Bitstrings im Hamming-Shape-Space ist bei einer Stringlänge von l gerade 2^l . Die Antigen-Abdeckung oder **Antigen-Coverage** C eines einzelnen Antikörpers bestimmt wie viele Antigene ein einzelner Antikörper erkennt. Sie hängt vom Affinity Threshold ε ab (vgl. Abb. 14).

$$C = \sum_{i=0}^{\varepsilon} \binom{l}{i} = \sum_{i=0}^{\varepsilon} \frac{l!}{i!(l-i)!}$$

Abbildung 14 Antigen Coverage

Da eine Abdeckung aller Antigene erreicht werden soll, ist die maximale Anzahl an eingefügten Antikörpern N_{\max} während des Iterationsprozesses gerade $2^I/C$. Da die Abdeckung C vom Affinity Threshold ε abhängt, hängt auch N_{\max} von ε ab. Das Ziel des ABNET ist die Erzeugung eines minimalen Antikörper-Netzwerks. Ist N_{\max} also größer als die Anzahl zu erkennender Antigene M , so kann N_{\max} gleich M gesetzt werden, da es nicht nötig ist mehr als einen Antikörper für die Erkennung eines bestimmten Antigens zu generieren. Da nur alle β Iterationen ein Antikörper dem Netzwerk hinzugefügt (oder entfernt) wird, ist die minimale Anzahl an Iterationen für die Konvergenz des Algorithmus gegeben durch Abbildung 15. β wird in der Klammer um eins erhöht, um dem Netzwerk die Möglichkeit zu geben, noch eine Gewichts Anpassung nach dem Einfügen der letzten Zelle durchzuführen.

$$N_{\max} = \frac{2^I}{C} \quad , \quad N_{it} = (\beta + 1)N_{\max}$$

Abbildung 15 Minimale Anzahl Iterationen f. Konvergenz des ABNET

7.6 Beispielanwendung ANIMALS-Datensatz

In diesem Kapitel wird an einem einfachen Beispiel die Anwendung des ABNET illustriert. Als Trainingsdaten wird der von Ritter und Kohonen 1989 vorgeschlagene **ANIMALS-Datensatz** verwendet, der einer Menge von Tieren bestimmte Eigenschaften zuweist (s. Abb. 16). Der Wertebereich der Eigenschaften ist binär, d.h. entweder ein Tier besitzt dieses Attribut oder nicht. Da die Daten bereits im binären Format vorliegen, muss also keine Konvertierung mehr durchgeführt werden. Das ABNET wird, wie vorher beschrieben, mit genau einem Antikörper initialisiert. Die Iterationsrate β wird auf drei gesetzt, ebenso die Mutationsrate α . Die für verschiedene Parameter ε generierten Netzwerke sind in Abbildung 17 dargestellt. Wie sich zeigt, kann man ε als Steuerparameter auffassen, der die Bildung von zusammenhängenden Blöcken (**Cluster**) im Netzwerk beeinflusst. Im ersten Fall, für $\varepsilon = 0$, bildet sich ein Netzwerk hochspezifischer Antikörper heraus. Mit anderen Worten, bis auf zwei Ausnahmen (die sich durch identische Bitstrings für Pferd/Zebra bzw. Eule/Adler begründen lassen), ist für jedes Antigen (Tierart) genau eine Zelle (Neuron) zuständig. Der obere Bereich der Ausgabeeinheiten lässt sich den Säugetieren zuordnen, während die Neuronen im unteren Bereich offensichtlich für Vögel zuständig sind.

		Dove	Hen	Duck	Goose	Owl	Hawk	Eagle	Fox	Dog	Wolf	Cat	Tiger	Lion	Horse	Zebra	Cow
Is	Small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	Medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	Big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Has	Two legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	Four legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	Hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	Hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	Mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	Feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Likes to	Hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	Run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	Fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	Swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Abbildung 16 ANIMALS-Datensatz

Wählt man $\varepsilon = 3$, werden verschiedene zusammengehörige oder artverwandte Tiergruppen, einer Zelle zugeordnet. Die Säugetiere werden wiederum den oberen Zellen zugeordnet, während im unteren Bereich die verschiedenen Vogelarten unterschieden werden. Setzt man schließlich $\varepsilon = 6$ entsteht ein Netzwerk, dass nur noch eine Unterscheidung zwischen Säugetieren und Vögeln treffen kann. Je größer man also den Affinity Threshold wählt, desto größer ist die Generalisierung der erzeugten Antikörper und die Zusammenfassung strukturähnlicher Gruppen. Ein kleines ε hingegen erzeugt ein sehr antigenspezifisches Netzwerk. Dies hat allerdings den Nachteil, dass für eine große Eingabemenge sehr viele Ausgabeeinheiten erzeugt werden, da für jedes präsentierte, von den anderen verschiedene Antigen, genau ein Antikörper generiert wird. Man muss hier also je nach Problemstellung zwischen Generalisierung und Komplexität der Netzwerkarchitektur abwägen.

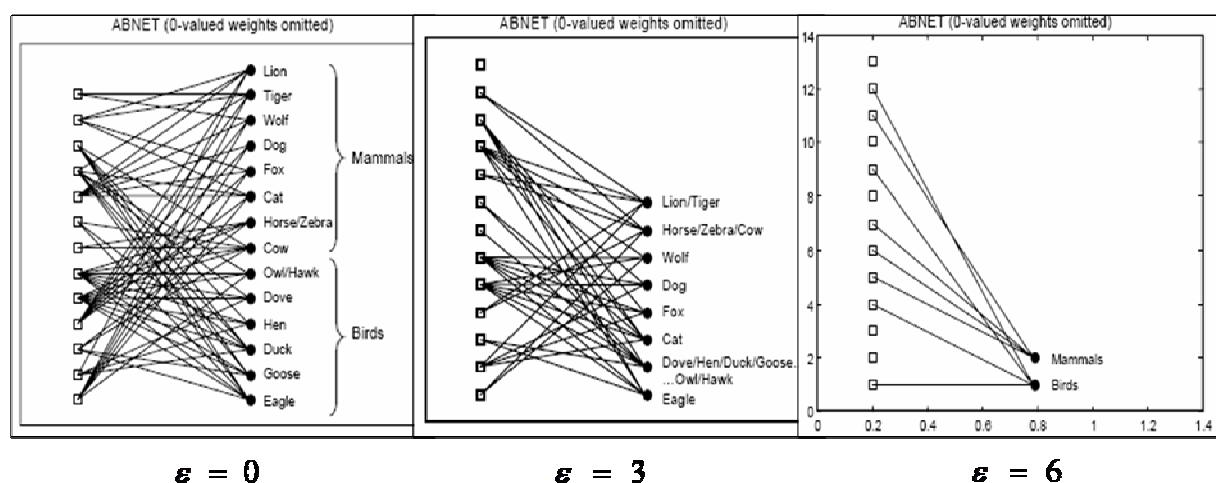


Abbildung 17 Resultate des ABNET für d. ANIMALS-Datensatz

7.7 Performance und Evaluation

De Castro, Von Zuben und de Deus haben in ihrer Arbeit in drei Evaluierungstest die Leistungsfähigkeit des ABNET mit anderen KNN-Architekturen, wie Self-Organizing-Maps (SOM), Hopfield-Netzwerk oder MultiLayer-Perceptron (MLP), verglichen (eine Beschreibung dieser Netzwerktypen findet sich in [3]). Im Fokus stand dabei, bestimmte Fähigkeiten der Netzwerke anhand unterschiedlicher Problemstellungen zu testen. Die verwendeten Problemstellungen waren das *Major Logic Decision*-Problem (MLD), das *Fitting Contact Lenses*-Problem (LENSES) und eine Aufgabe aus dem Bereich der *Binary Character Recognition*. Bewertungskriterien waren die Komplexität der erzeugten Architekturen (Anzahl der Verbindungen) und die Qualität der Pattern-Klassifikation. Eine genaue Beschreibung der ausgewählten Probleme und der Durchführung der Tests findet sich in [1] und wird an dieser Stelle nicht explizit wiederholt.

Die Vorteile des ABNET gegenüber den anderen Netzwerkarchitekturen, die von de Castro, Von Zuben und de Deus aufgeführt werden, lassen sich wie folgt zusammenfassen:

- Flexible Struktur

Einige der in den Tests verwendeten Netzwerke wurden speziell für die Lösung eines bestimmten Problems konstruiert und sind nicht direkt auf andere Problemstellungen anwendbar. Im Gegensatz dazu besitzt das ABNET eine sehr flexible Struktur, die direkt auf verschiedene Problemtypen anwendbar ist (sofern die Eingabedaten in binäre Form gebracht werden können).

- Komplexität vs. Genauigkeit

Das ABNET verfügt über eine kontrollierbare Plastizität, da es über den Affinity Threshold ϵ parametrisierbar ist. Je nach zu lösender Aufgabe kann man damit ggf. zwischen Komplexität des Netzwerks und Qualität der Erkennungsrate abwägen.

- Hardwareimplementierung

Da der Lernalgorithmus und die Gewichte des ABNET binärer Natur sind, eignet es sich insbesondere sehr gut für eine Hardwareimplementierung und unterliegt im Gegensatz zu anderen Architekturen keinen Quantisierungseffekten.

8. Zusammenfassung

Das ABNET stellt einen interessanten Ansatz dar, Grundprinzipien und Mechanismen der Immunologie, auf den Bereich der künstlichen neuronalen Netze zu übertragen. Basierend auf den Prinzipien der klonalen Selektion und der Affinitätsmaturation wurde gezeigt, wie diese Ideen erfolgreich für die Konstruktion eines binären, konkurrierenden Netzwerks angewendet werden können, welches ein minimales Antikörper-Repertoire zur Erkennung einer gegebenen Antigenmenge erzeugt. Anhand eines einfachen Beispiels, unter Verwendung des ANIMALS-Datensatzes, wurde die prinzipielle Anwendung des ABNET illustriert und in, von den Entwicklern des ABNET durchgeführten Evaluierungstests wurde die Leistungsfähigkeit des ABNET anhand bestimmter Problemstellungen, im Vergleich zu anderen, bereits etablierten Netzwerkarchitekturen demonstriert. Inwieweit diese Möglichkeiten für die Entwicklung von neuartigen Lernalgorithmen und Architekturen tatsächlich in der Realität Anwendung finden, bzw. inwieweit das ABNET tatsächlich in Real-World Applikationen eingesetzt wird, bleibt abzuwarten.

9. Literatur

- [1] Leandro Nunes de Castro, Fernando J. Von Zuben, Getúlio A. de Deus Jr. *The Construction of a Boolean Competitive Neural Network Using Ideas from Immunology*. 2003
- [2] Leandro Nunes de Castro, Fernando J. Von Zuben. *Artificial Immune Systems: Part I – Basic Theory and Applications*. 1999
- [3] Simon Haykin. *Neural Networks - A Comprehensive foundation*. MacMillan College Publishing, 1994
- [4] Andreas Scherer. *Neuronale Netze. Grundlagen und Anwendungen*. Vieweg, 1997
- [5] Charles A. Janeway, Paul Travers, Mark Walport. *Immunologie*, Spektrum Akademischer Verlag, 2003
- [6] Stefan Silbernagl, Agamemnon Despopoulos. *Taschenatlas der Physiologie*, Thieme, 2003