A decorative graphic consisting of a light green circle on the left and a horizontal green bar extending to the right. The title text is centered within this bar. Large white square brackets are positioned on the left and right sides of the bar, framing the title.

Construction of a Boolean Competitive Neural Network

Seminar SS 2005
Künstliche Immunsysteme

Vortrag: Carsten Längsfeld

[Übersicht]

- Vom Gehirn zum Neuronalen Netzwerk
- Einführung Neuronale Netzwerke (ANN)
- Grundlagen Immunsystem
- Klonale Selektion
- Shape-Space Formalismus
- Das AntiBody-NETwork (ABNET)
- Beispielanwendung des ABNET
- Fazit
- Fragen & Diskussion

Vom Gehirn zum ANN

- Gehirn kann bestimmte komplexe Aufgaben (Mustererkennung, Wahrnehmung, Bewegungsplanung und –kontrolle) deutlich effizienter als heutige Computersysteme lösen.

Beispiel: Sonar der Fledermaus

Sonar ist aktives Echo-Lokationssystem
Fledermaus kann bei relativ kleinem Gehirn
Informationen wie Größe, Entfernung, relative
Geschwindigkeit, Flugwinkel und Höhe des Ziels mit hoher
Genauigkeit bestimmen

Wie ist das Gehirn zu dieser Leistung fähig??

Vom Gehirn zum ANN(2)

Idee:

Neuronen als Strukturelemente des Gehirns (1911 Cajal) und Synapsen als Verbindungselemente zwischen Neuronen (Leitungen)

- Menschliches Gehirn besitzt ca. 10 Billionen Neuronen und etwa 60 Trillionen Synapsen
- Langsame Operationsgeschwindigkeit der einzelnen Neuronen, aber => **massive Parallelverarbeitung**
- „Gehirn ist hochkomplexer, nichtlinearer, parallel arbeitender Computer“ (Haykin, 1995)

[Vom Gehirn zum ANN(3)]

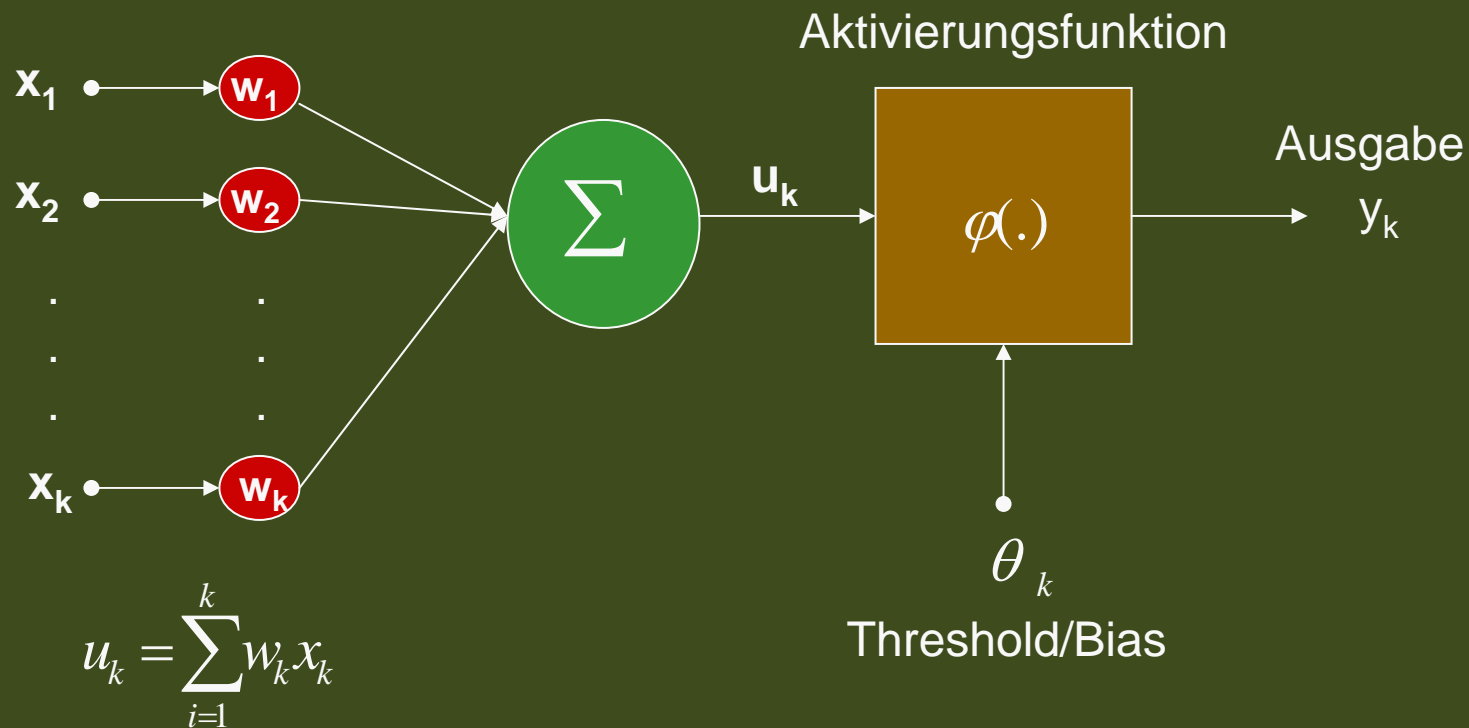
- Synapsen können exzitatorisch (anregend) und inhibitorisch (hemmend) auf ein Neuron wirken
- **Plastizität** des Gehirns ermöglicht Anpassung an Umgebung => erzeugen und modifizieren von Synapsen. Gehirn ist einem **Lernprozess** unterworfen.
- Erfahrung, Wissen wird über **Synapsengewichtung** gespeichert
- Übertragen dieser Prinzipien auf Computersysteme
=> Künstliches Neuronales Netzwerk (ANN)

Einführung ANN

- Grundelemente eines ANN
 - Neuron
 - Aktivierungszustand
 - Propagierungsfunktion (z.B. gewichtete Input-Summe)
 - Aktivierungsfunktion (z.B. Schwellwertfunktion)
 - Threshold/Bias
 - Netzwerktopologie (Feedforward, Feedback)
 - Lernregel

Einführung ANN(2)

Modell des Neurons



$$y_k = \varphi(u_k - \theta_k)$$

Einführung ANN(3)

■ Beispiele für Lernregeln

○ Delta-Regel

Abweichung zwischen Soll- und Istwert => Anpassen der Gewichte mit dem Ziel, Ist-Ausgabe in Richtung Soll-Ausgabe konvergieren zu lassen

○ Hebb'sche Regel

Zwei Neuronen i und j zur gleichen Zeit aktiv und Verbindung zwischen i und j => Verstärken der Verbindung w_{ij}

○ **Konkurrierend (Competitive)**

Neuronen konkurrieren darum aktiviert zu werden. Es kann nur jeweils ein Neuron zu einer Zeit aktiv sein.
Voraussetzungen: alle Neuronen sind gleich (bis auf zufällig verteilte Gewichte), Gewichtungslimit, Mechanismus der den Wettkampf ermöglicht.

Aufgaben des Immunsystems

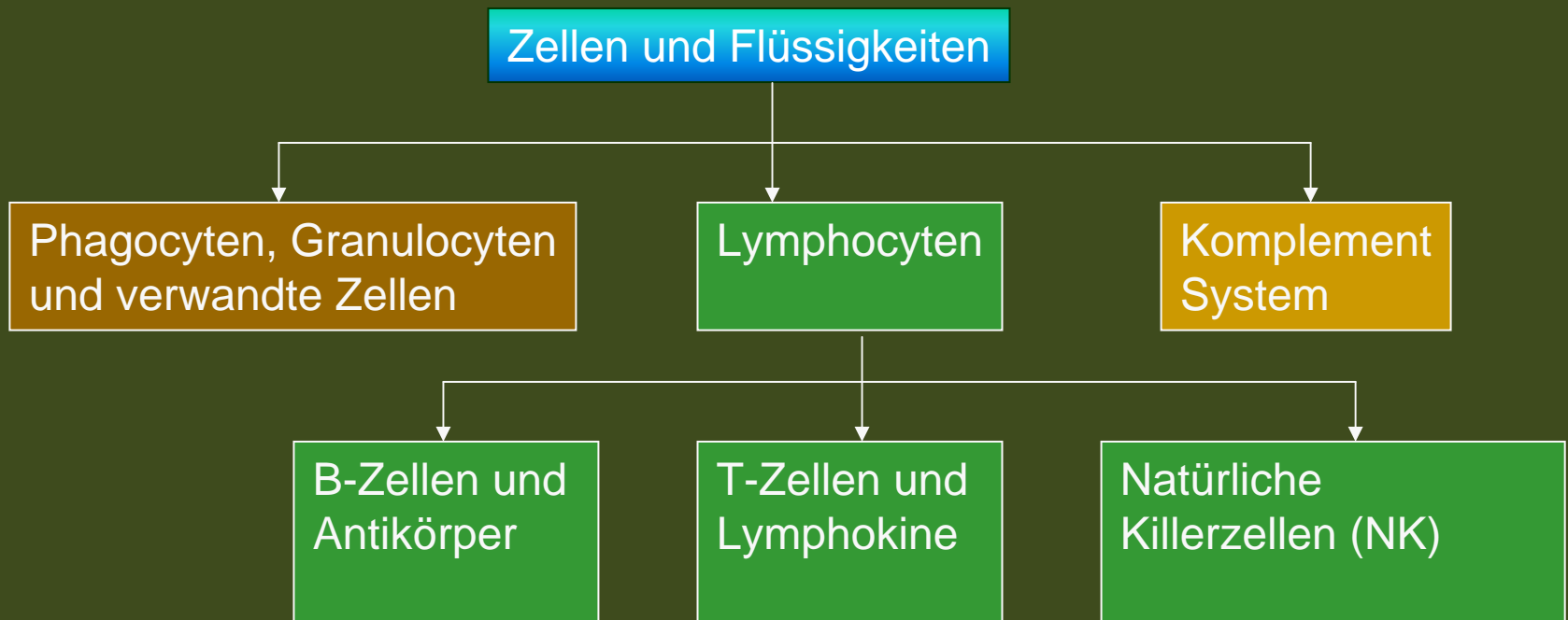
- Erkennen und Bekämpfen von
 - dysfunktionalen, körpereigenen Zellen (infectious self)
 - externen, infektiösen Erregern (infectious nonself)
- IS besitzt Gedächtnisfunktion

Externe Erreger = Antigene (**Ag**) oder Pathogene

Ebenen des Immunsystems

- physisch (z.B. Haut, Atmungssystem)
- physiologisch (Enzyme, pH-Wert, Temperatur des Körpers)
- unspezifische Immunabwehr
 - erste Abwehrmaßnahmen bei Infektion
 - angeborene Mechanismen
- spezifische, adaptive Immunabwehr
 - Mechanismen richten sich gegen einen bestimmten Erreger
 - nicht angeboren, sondern erlernt

[Übersicht Immunzellen]



Funktion Immunzellen

■ B-Zellen

- Produktion/Ausschüttung von Antikörpern (**Ab**)
- jede B-Zelle produziert einen bestimmten Ab
- Proliferation/Differentiation

■ T-Zellen

- kontrollieren die Aktionen anderer Zellen
- Angriff infizierter Zellen
- 3 Klassen (T-Regulator, T-Suppressor, T-Killer)
- arbeiten primär mit starken chem. Botenstoffen
=> Lymphokine
- Proliferation/Differentiation

Funktion Immunzellen(2)

- NK-Zellen

Zerstören von unspezifischen Antigenen (verwenden starke Chemikalien)

- Phagocyten etc.

- „Zellfresser“ => Aufnahme und Verdauung von Mikroorganismen und Antigenen
- einige Phagocyten (z.B. Makrophagen) haben die Fähigkeit der Antigenpräsentation (**APC**)

Funktion Immunzellen(3)

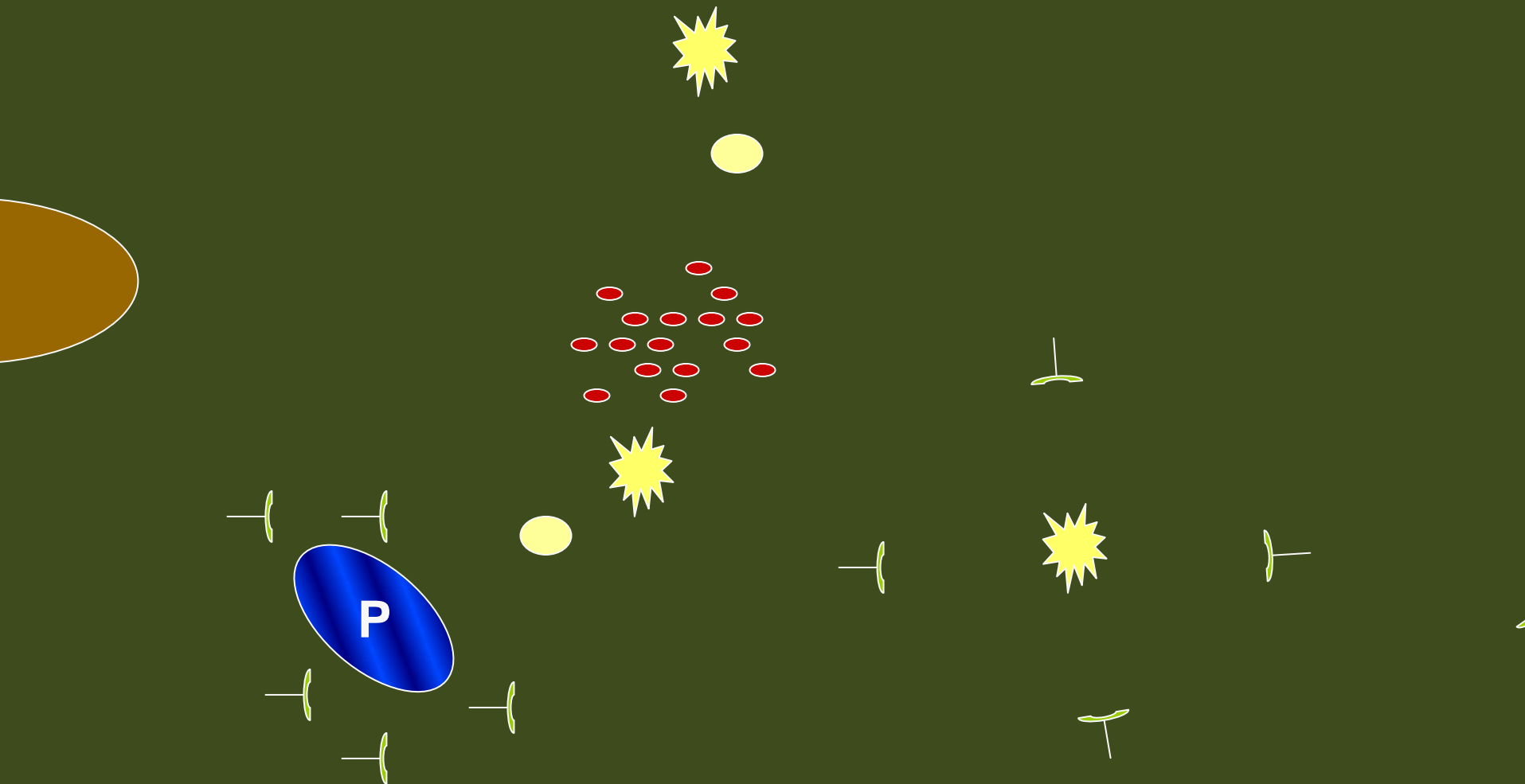
- Komplement

- Komplex aus Plasma Proteinen
- kann Zellwand eines Eindringlings beschädigen, um die Zelle zu zerstören

oder

- Antigen für die Zerstörung durch Phagocyten markieren.

[Arbeitsweise des IS]



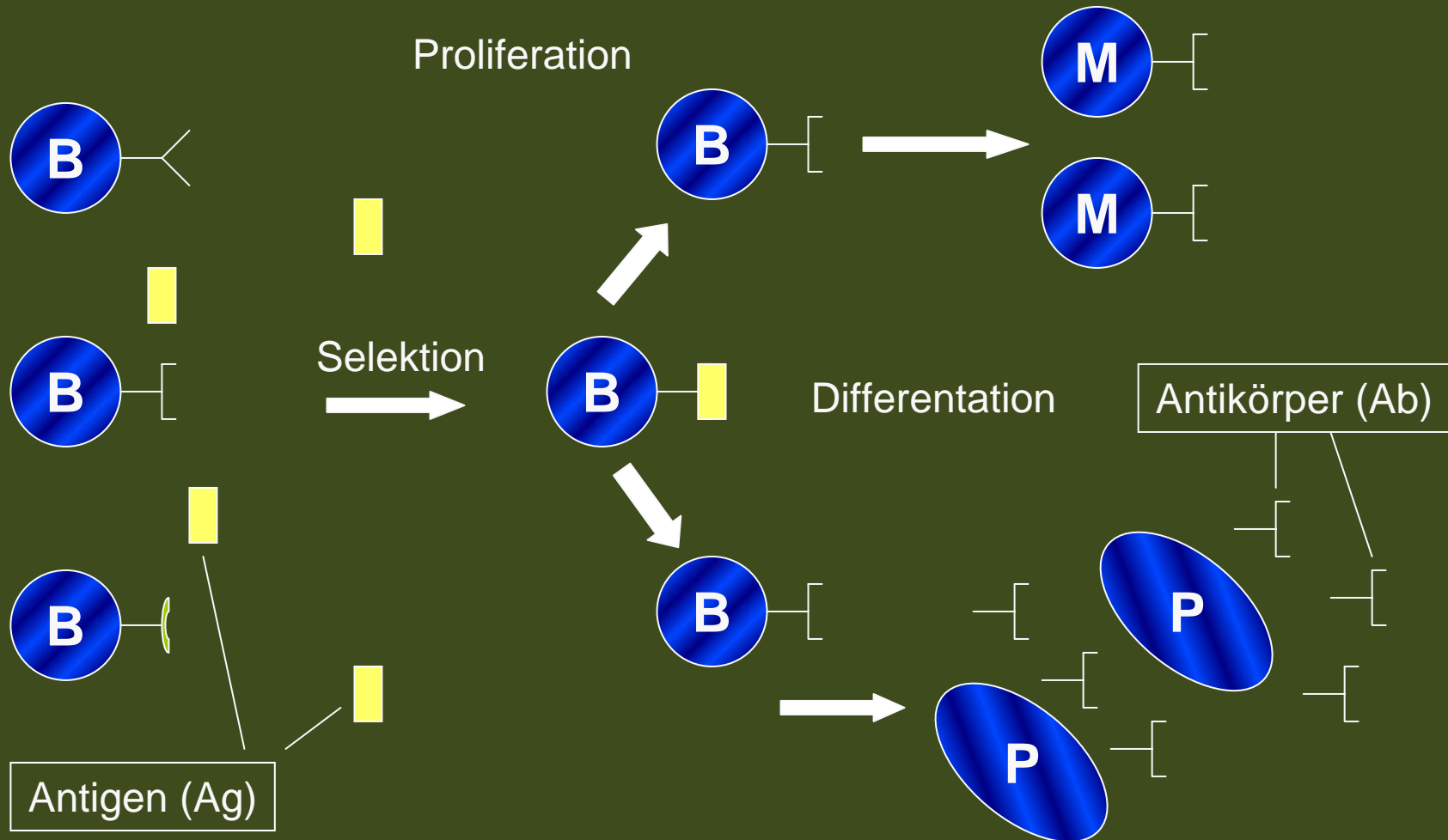
Klonale Selektion

Versucht die grundlegenden Eigenschaften der Immunantwort auf Ag-Stimulus zu beschreiben

- Proliferation/Differentiation bei Kontakt mit Ag
- Klone sind Kopien der Eltern
(aber Mutationsmechanismus unterworfen
=> somatic hypermutation, cross-reactive response, affinity maturation)
- Elimination von nutzlosen oder selbstreagierenden Zellen

Nur Zellen mit hoher Ag-Affinität werden geklont
=> „Survival of the fittest“

[Klonale Selektion(2)]



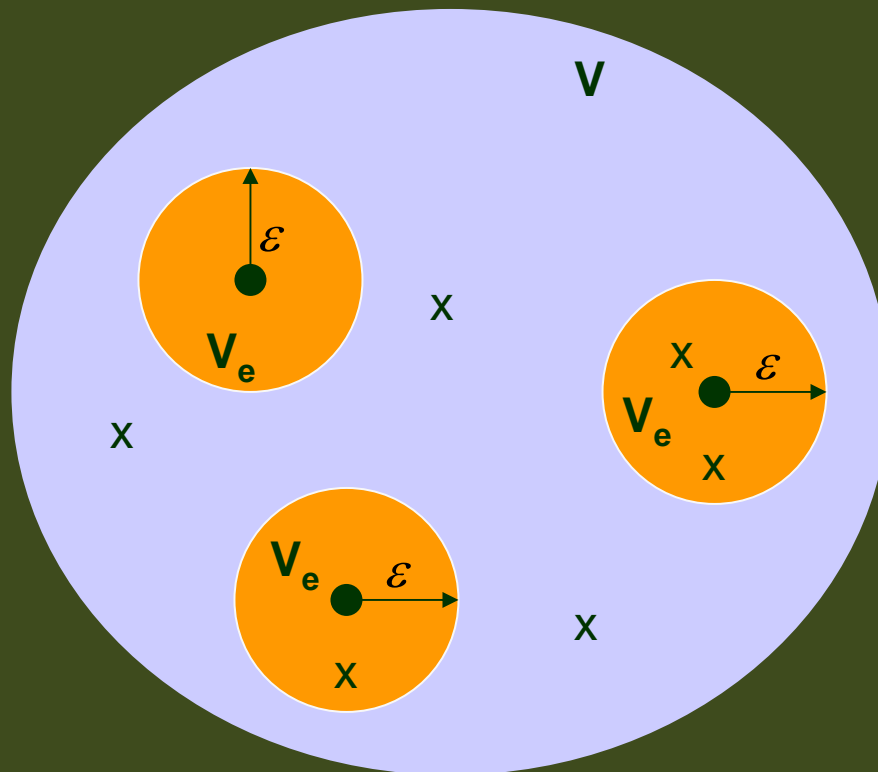
[Shape-Space]

- Affinität zwischen Ab und Ag hängt u. a. von elektro-statischen, chemischen und geometrischen Eigenschaften ab
- Bindung zwischen Ag und Ab
=> große komplementäre Regionen
- Problem: Quantitative Beschreibung der Affinität zwischen Ab und Ag
=> Shape-Space

[Shape-Space(2)]

- Generalized Shape
 - Beschreibung von Ab/Ag durch L Parameter (z.B. Länge, Breite, Ladung etc.)
 - Darstellung des Shapes durch Punkt im L-dimensionalen Raum (Shape-Space)
 - Größe des Ab-Repertoire = Anzahl der Punkte im Shape-Space
 - Werte sind endlich \Rightarrow Punkte liegen in endlichem Bereich V

[Shape-Space(3)]



V_e = „Recognition Region“

x = Komplement von A_g

[Shape-Space(4)]

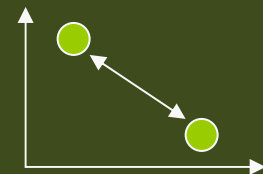
- Representation von Ab-Ag

$ab = \langle ab_1, ab_2, \dots, ab_L \rangle,$

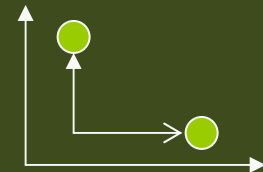
$ag = \langle ag_1, ag_2, \dots, ag_L \rangle$

- Messen der Affinität Ab-Ag über Distanzfkt.

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

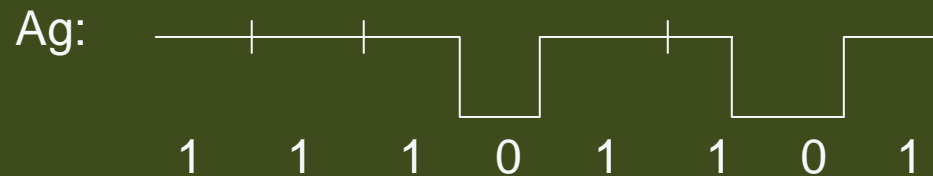
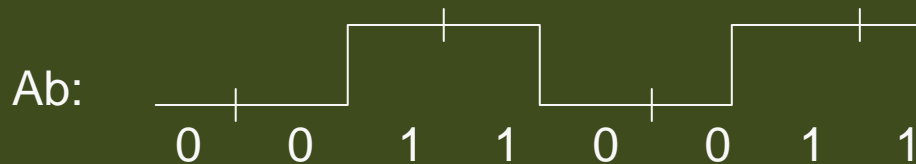


Manhattan-Distanz: $\sum_{i=1}^L |ab_i - ag_i|$



Hamming Shape-Space

- Alternative zu euklid. Shape-Space
- Darstellung durch Sequenz der Länge L mit Symbolen aus Alphabet der Größe k
- Hier binäre Darstellung $k=2$, $L=8$

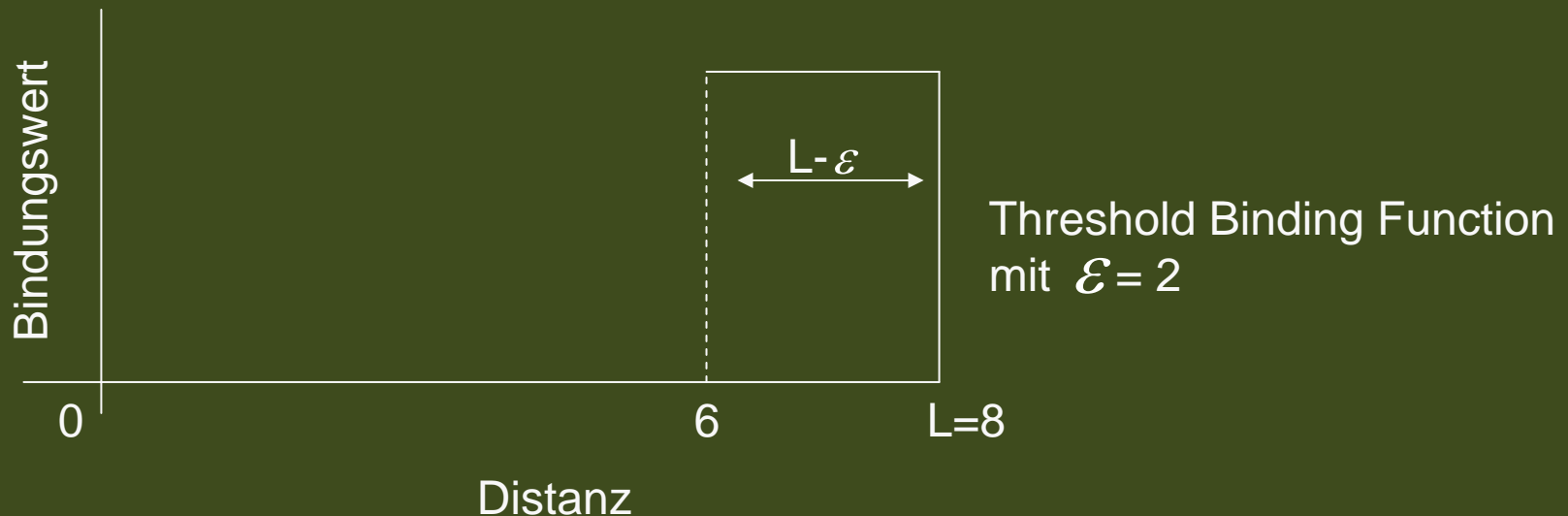


=> Affinität (Distanz) = 6

XOR: 1 1 0 1 1 1 1 0

Hamming Shape-Space(2)

- Affinity Threshold ε beeinflusst Bindungsverhalten
- $\varepsilon = 0$, Ab benötigt perfektes Match



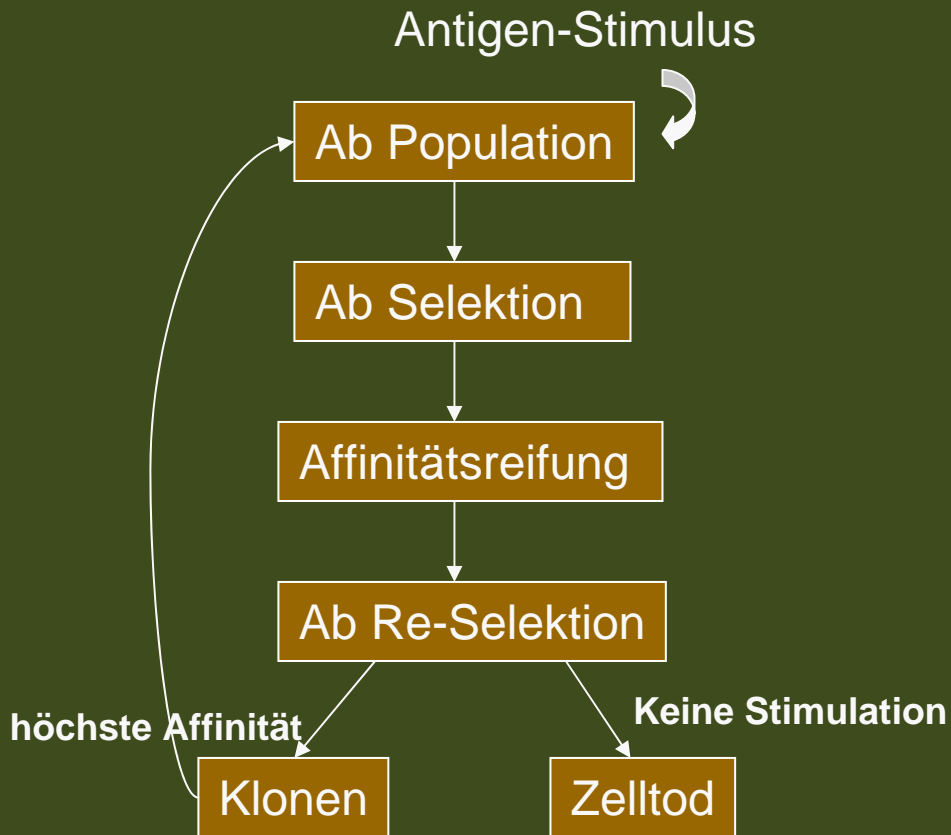
[ABNET]

- Alternativer ANN Lernalgorithmus
- Input-Patterns sind Antigene (das zu lösende Problem, in binärer Form)
- Neuronen werden als Zellen bezeichnet (werden durch Ab repräsentiert)
- Eigenschaften
 - Veränderliche Netzwerkstruktur (growing, pruning) => Klonale Selektion, Apoptosis
 - Binäre Gewichtsvektoren (Hamming Shape-S.)
 - Konkurrierendes Netzwerk, „unsupervised Learning“ (basierend auf Mutationsmechanismus)

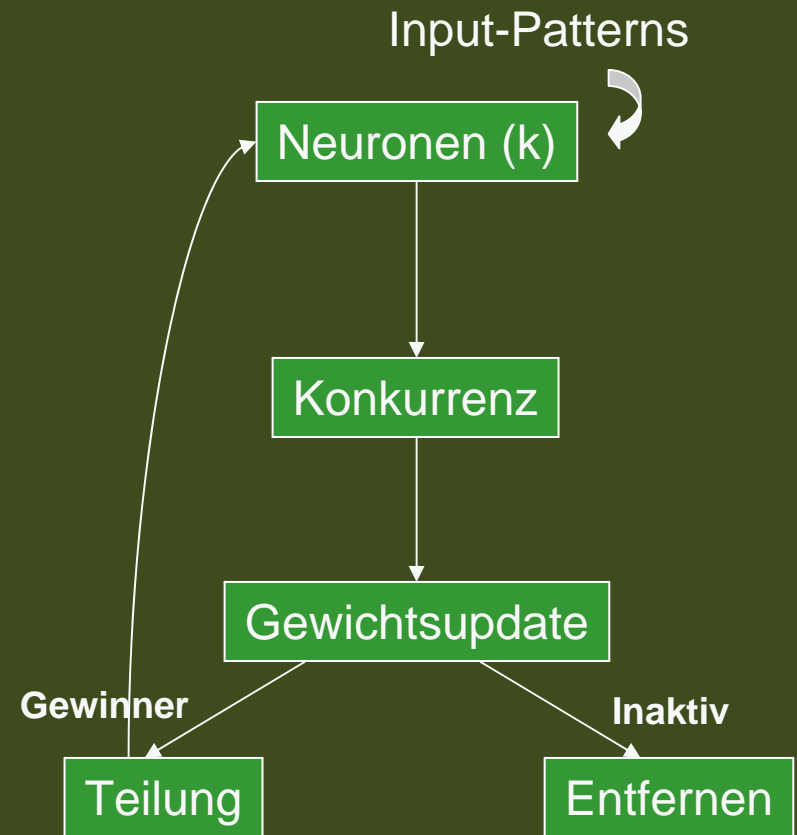
ABNET: Modell

- Antikörper k (Ab_k) wird durch binären Gewichtsvektor w_k repräsentiert (Input \rightarrow Output-Einheit k)
- τ_j = Antigen-Konzentrationslevel für Ab_j
- ν_a = Label für Ab mit höchster Affinität zu Ag_a
Beispiel: Ab_7 hat höchste Affinität zu $Ag_9 \Rightarrow \nu_9 = 7$
- α = Mutationsrate, Anzahl mutierender Bits
- β = Anzahl Iterationen bis geprüft wird, ob Netzwerkstruktur verändert werden muss
- ε = Affinity Threshold

ABNET: Algorithmus



Immunsystem Perspektive



Neuronale Netzwerk Perspektive

ABNET: Algorithmus(2)

Initialzustand: Ein Ab mit zufälligem w aus dem Ab-Repertoire

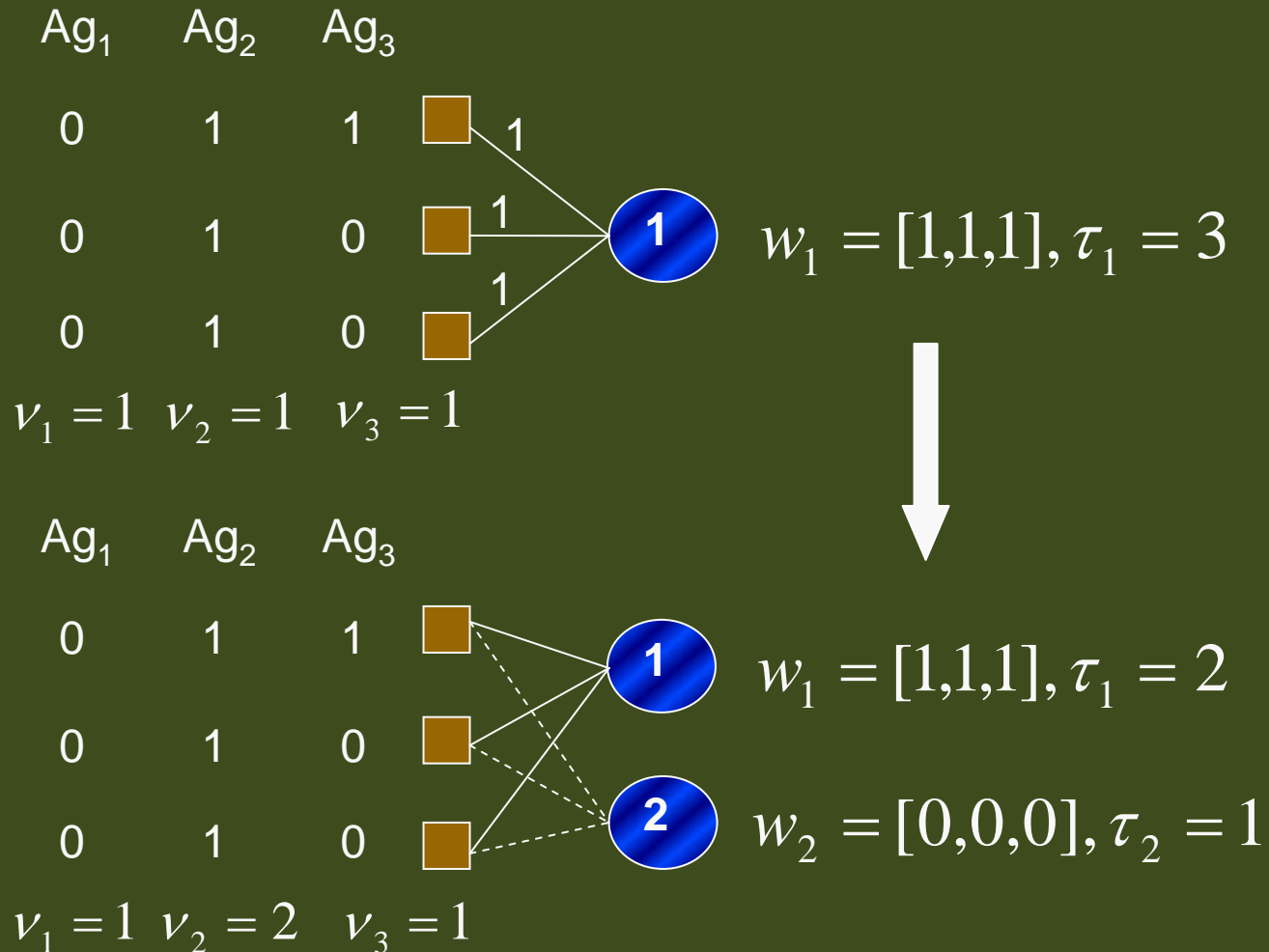
1. Präsentation aller Input-Patterns Ag_i
2. Für jedes $Ag_i \Rightarrow$ bestimme Ab_k mit höchster Affinität
$$k = \arg \max_k \|Ag_i - Ab_k\|$$
3. Erhöhen von τ_k und setze $\nu_i = k$
4. Update von w_k
5. Nach β Schritten \Rightarrow Network growing/pruning in Abhängigkeit von τ und ε

Ziel: Minimales Ab-Netzwerk mit maximaler Abdeckung des Antigen-Repertoires

ABNET: Network Growing

- Zelle mit höchster Stimulation wird geklont
- Zellen mit $\tau_j > 1$ sind potentielle Kandidaten
- Gibt es keine Zelle $j \Rightarrow$ Netzwerk bleibt unverändert
- Sonst wähle $s = \arg \max_{j \in O} Ab_j, O = \{j \mid \tau_j > 1\}$
- Wenn $\tau_s > \varepsilon$, dann wird die Zelle geklont, sonst bleibt das Netzwerk unverändert
- Gewichte der neuen Zelle sind exaktes Komplement des Antigens mit niedrigster Affinität zu Ab_s

ABNET: Network Growing(2)



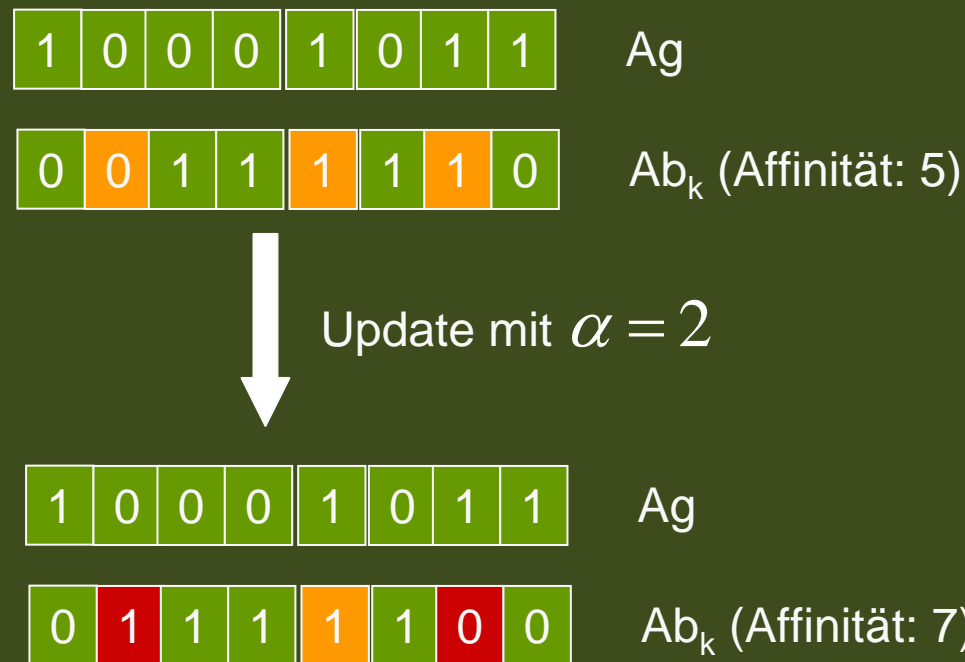
ABNET: Network Pruning

- Simuliert den Zelltod (Apoptosis)
- Gilt für Zelle p nach β Iterationen $\tau_p = 0$
=> p wird aus dem Netzwerk entfernt
- Lernrate α wird auf Initialwert zurückgesetzt
=> Möglichkeit der Redefinition des Verbindungsschemas

ABNET: Gewichtsupdate

- Simuliert den Mutationsvorgang des IS (Affinity maturation)
- Hypermutationsrate $\alpha \in \mathbb{Z}_0^+$ bestimmt Anzahl der zu ändernden Bits im Ab-String
=> nichtkomplemente Stellen sind Kandidaten für Änderung
- Nach x Iterationen:
 $\alpha = \alpha - 1$ bis $\alpha = 0$ (kein Update)
- Zellen mit hoher oder maximaler Affinität werden nicht mutiert

[ABNET: Gewichtsupdate(2)]



[ABNET: Konvergenz]

- Prüfung auf Netzwerkveränderungen alle β Iterationen
- β beeinflusst maßgeblich die Lerngeschwindigkeit des Netzwerks (ermittelt durch empirische Analyse)

Frage: Wann konvergiert der Algorithmus?

ABNET: Konvergenz(2)

- Anzahl verschiedener Bitstrings im Hamming Shape-Space ist 2^l , l = Stringlänge
- Coverage C eines einzelnen Antikörpers ist gegeben durch

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

- Maximale Anzahl eingefügter Zellen

$$N_{\max} = \frac{2^l}{C}$$

Wenn $N_{\max} > M$ zu erkennende Antigene: $N_{\max} = M$

[ABNET: Konvergenz(3)]

Antwort: $N_{it} = (\beta + 1)N_{\max}$

$\beta + 1$ wegen Gewichts Anpassung nach
letzter eingefügter Zelle

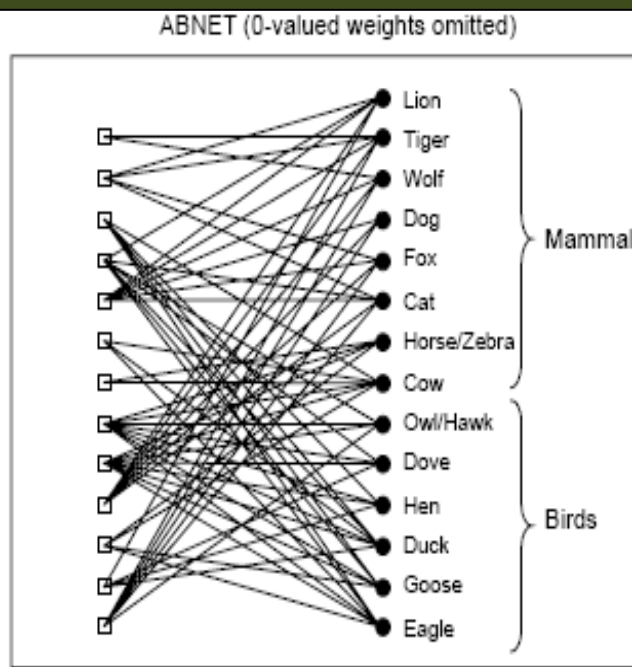
ABNET: Beispiel ANIMALS

■ ANIMALS Dataset

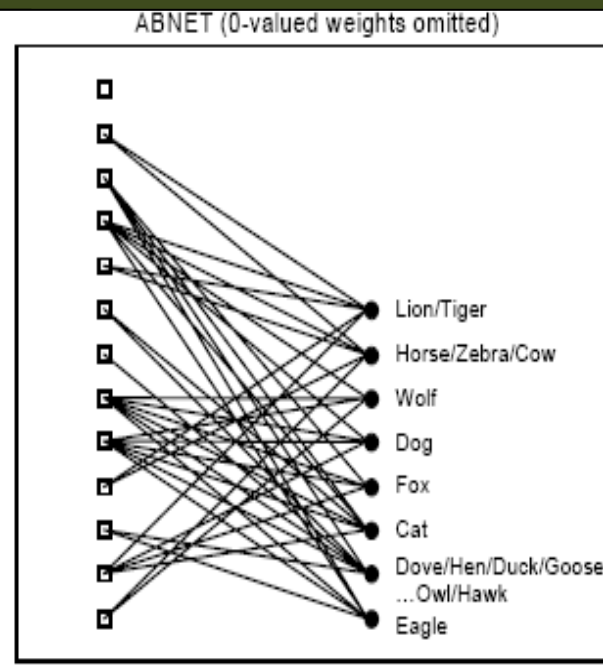
		Dove	Hen	Duck	Goose	Owl	Hawk	Eagle	Fox	Dog	Wolf	Cat	Tiger	Lion	Horse	Zebra	Cow
Is	Small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	Medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	Big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Has	Two legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	Four legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	Hair	0	0	0	0	0	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

[ABNET: Beispiel ANIMALS(2)]

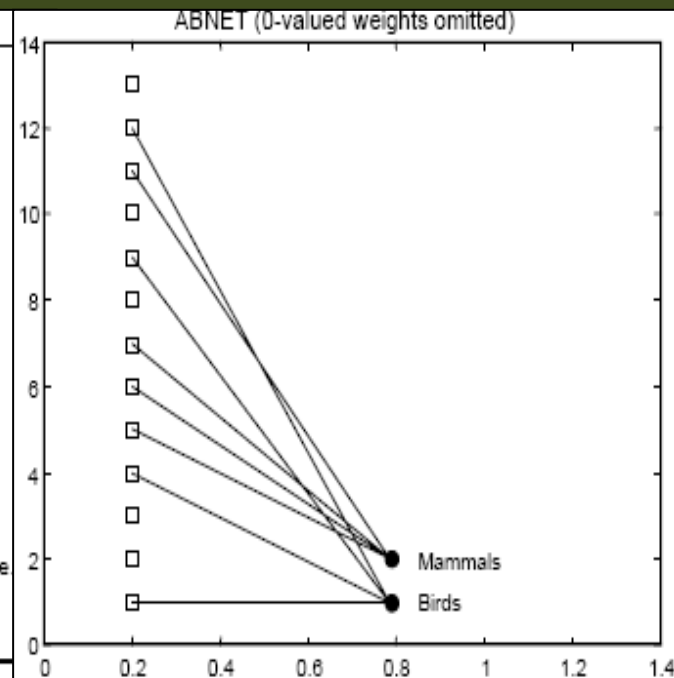
- Initialisierung: $\#Ab=1$; $\alpha, \beta = 3$



$\varepsilon = 0$



$\varepsilon = 3$



$\varepsilon = 6$

ABNET: Performance

- Vergleich mit anderen ANN-Architekturen (MLP, Hopfield, SOM) in „Real-World-Beispielen“, z.B. Binary Character Recognition
- Bewertungskriterien: Architekturkomplexität (Anzahl Verbindungen) und Qualität der Klassifikation

Resultat: ABNET erzielt sehr gute Ergebnisse

- Vorteile
 - Flexible Architektur
 - Komplexität vs. Genauigkeit (parametrisierbar)
 - gut geeignet für Hardwareimplementierung
 - 100% Genauigkeit bei Problemen wie XOR, Addition, Negation lösen

Fazit

- Prinzipien und Mechanismen des IS als neuartiges Paradigma für die Entwicklung von Lernalgorithmen und ANN-Architekturen
- Erfolgreiches Anwenden dieser Ideen für die Konstruktion eines binären, konkurrierenden ANN am Beispiel des ABNET

[The End

]

Gibt es Fragen?

42