

Bernd Glück

Simulation des optimalen Raumlufthzustandes mit neuronalem Netz

In [1] wurde eine neuartige Optimierung für einen Klimaprozeß, der nicht durch eine vorgegebene Regelstrategie beeinflusst ist, vorgestellt. Er dient als Vergleichsprozeß, an dem dann die realen, praxisüblichen Prozeßverläufe gespiegelt werden können. Unter anderem wird dabei der optimale Raumzustandspunkt innerhalb eines zulässigen Behaglichkeitsbereiches bestimmt. Dieses Ergebnis spielt bei der Umsetzung des Prozeßverlaufes eine wichtige Rolle. Es wird über erste Versuche berichtet, wie dazu neuronale Netze eingesetzt werden könnten.

Simulation of the optimal room air condition with neural network

For the optimization of air conditioning processes it can be favourable to use neural networks. However the trained network applies to the used boundary conditions and to the respective air conditioning system only. It is reported on first attempts for the determination of the optimal room air conditioning within the comfort area.

Keywords: room air conditioning, simulation, neural network, optimization algorithm

1 Allgemeine Betrachtung

Zur Bearbeitung der verschiedensten Aufgabenstellungen verwendet man üblicherweise bekannte, spezielle Lösungsansätze. Das deterministische Vorgehen ist uns am vertrautesten. Aus $a + b$ folgt c . Viele Ermittlungen basieren aber auch auf umfangreichen Erfahrungswerten. Dazwischen könnte man das Anwenden der Ähnlichkeitstheorie oder das Arbeiten mit spezifischen Kennzahlen usw. einordnen. Für sehr viele Fälle erscheint das Beschreiben mit Elementen der Fuzzylogik sinnvoll. Aus a und b folgt mit einer Unschärfe c . In der Regel verwendet man – bewußt oder unbewußt – Kombinationen der verschiedenen Verfahren, z. B. eine Zusammenfassung von genauen Zusammenhängen plus extrapolierten Erfahrungswerten. Dies macht man beispielsweise auch bei der Aufstellung von Leistungssummen. Die Daten, die genau bekannt sind, werden genau eingetragen, teilweise unbekannte Größen werden aufgrund der Erfahrung, eventuell mit Hilfe von Kennziffern gewählt. Die neuronalen Netze entsprechen diesen Kombinationen am ehesten. Sie stellen eine Variation von festen Übertragungsfunktionen mit flexiblen Verknüpfungen dar.

2 Grundprinzip des Aufbaus und der Arbeitsweise neuronaler Netze

Die nachfolgende Beschreibung ist in wesentlichen Teilen [2] entnommen. Es sei zum besseren Verständnis erlaubt, diese einfachen Grundlagen wiederzugeben, da diese noch nicht Allgemeingut sein dürften. Es wird ein Neuron mit einem einzigen Input p , dieser ist ein Skalar, betrachtet (Bild 1). Er wird gewichtet mit dem Faktor w , dies ist ein Parameter dieses Neurons. Das Produkt $e = w \cdot p$ stellt dann die Eingangsgröße für die Transferfunktion F dar. Im allgemeinen Fall kann aber zum gewichteten Input $p \cdot w$ noch eine einstellbare Verschiebung b – ebenfalls eine skalare Größe – hinzugefügt werden. Damit ergibt sich die Eingangsgröße e für die Transferfunktion F zu

$$e = p \cdot w + b. \quad (1)$$

Der Output lautet dann

$$a = F(p \cdot w + b). \quad (2)$$

Die Einstellung der Parameter w und b sowie die Wahl der Transferfunktion F dient dazu, den Output des Neuronen-netzwerkes im Training dem gewünschten Ziel anzupassen. Die Transferfunktionen sind in ihrer Form sehr vielfältig,

eine bescheidene Auswahl ist im Bild 2 angegeben.

Meistens stürzen auf ein Neuron multiple Inputs $p_1 \dots p_n$ ein. Aus diesen Informationen wird eine Summengröße gebildet und im Neuron verarbeitet. Die Inputs werden im Ergebnis einer Lernregel nicht gleich bewertet, denn jedem Input ist eine Wichtung zugeordnet. Symbolisch ist dieser Vorgang im Bild 3 dargestellt. Dabei

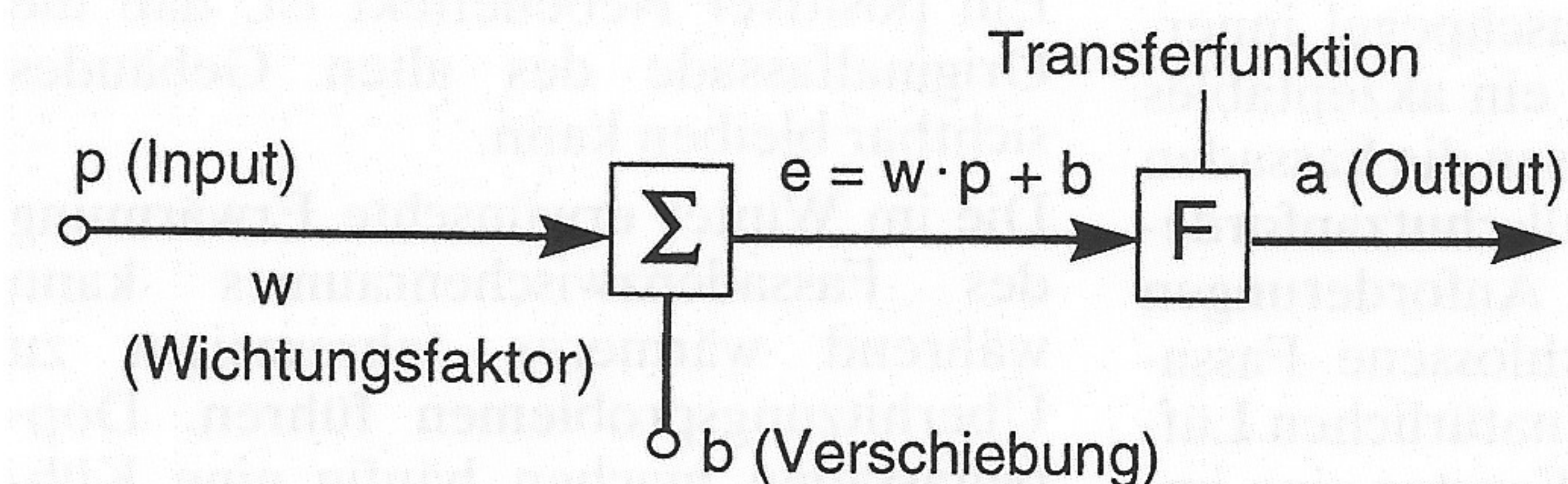


Bild 1 Schematische Darstellung eines Neurons mit der Transferfunktion F

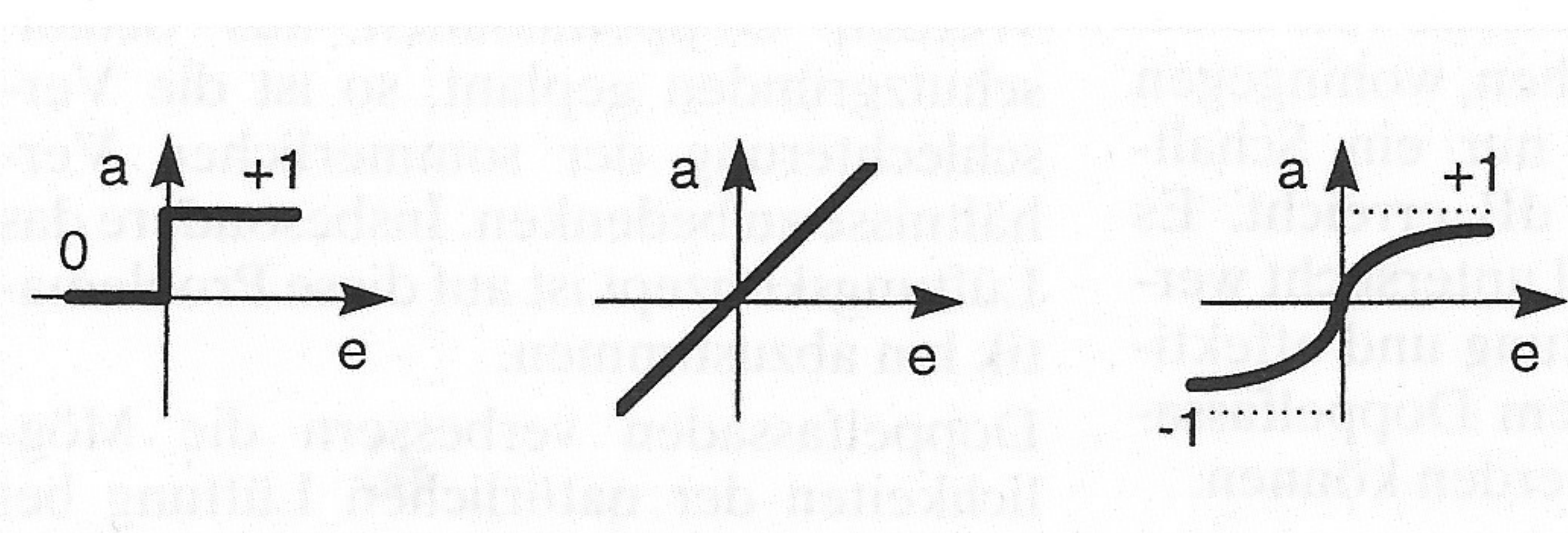


Bild 2 Beispiele für Transferfunktionen

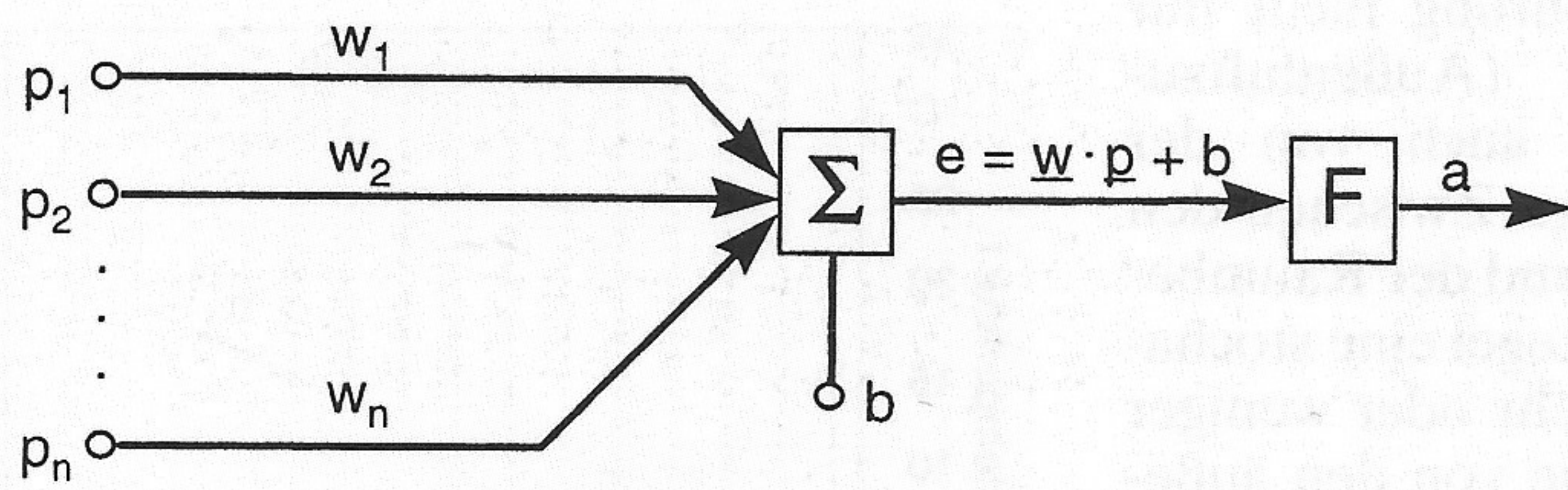


Bild 3 Neuron mit multiplen Inputs

entwickelt sich Gl. (1) weiter zu

$$e = \underline{w} \cdot \underline{p} + b. \quad (3)$$

Die Wichtungen $\underline{w}$ und die Inputs $\underline{p}$ können jetzt als Zeilenvektor vom Typ (1,n) bzw. als Spaltenvektor vom Typ (n,1) aufgefaßt werden, die miteinander als Skalarprodukt (Typ(1,1)) verknüpft sind, so daß daraus eine Zahl e als Eingangsgröße für die Transferfunktion F – die das Neuron symbolisiert – folgt. (Hinweis: Die unterstrichenen Variablen stellen Vektoren dar!)

Für einen sinnvollen Einsatz wird natürlich mit einer ganzen Netzwerkarchitektur von Neuronen gearbeitet. Sie bilden im einfachsten Fall eine Schicht aus k Neuronen (Bild 4).

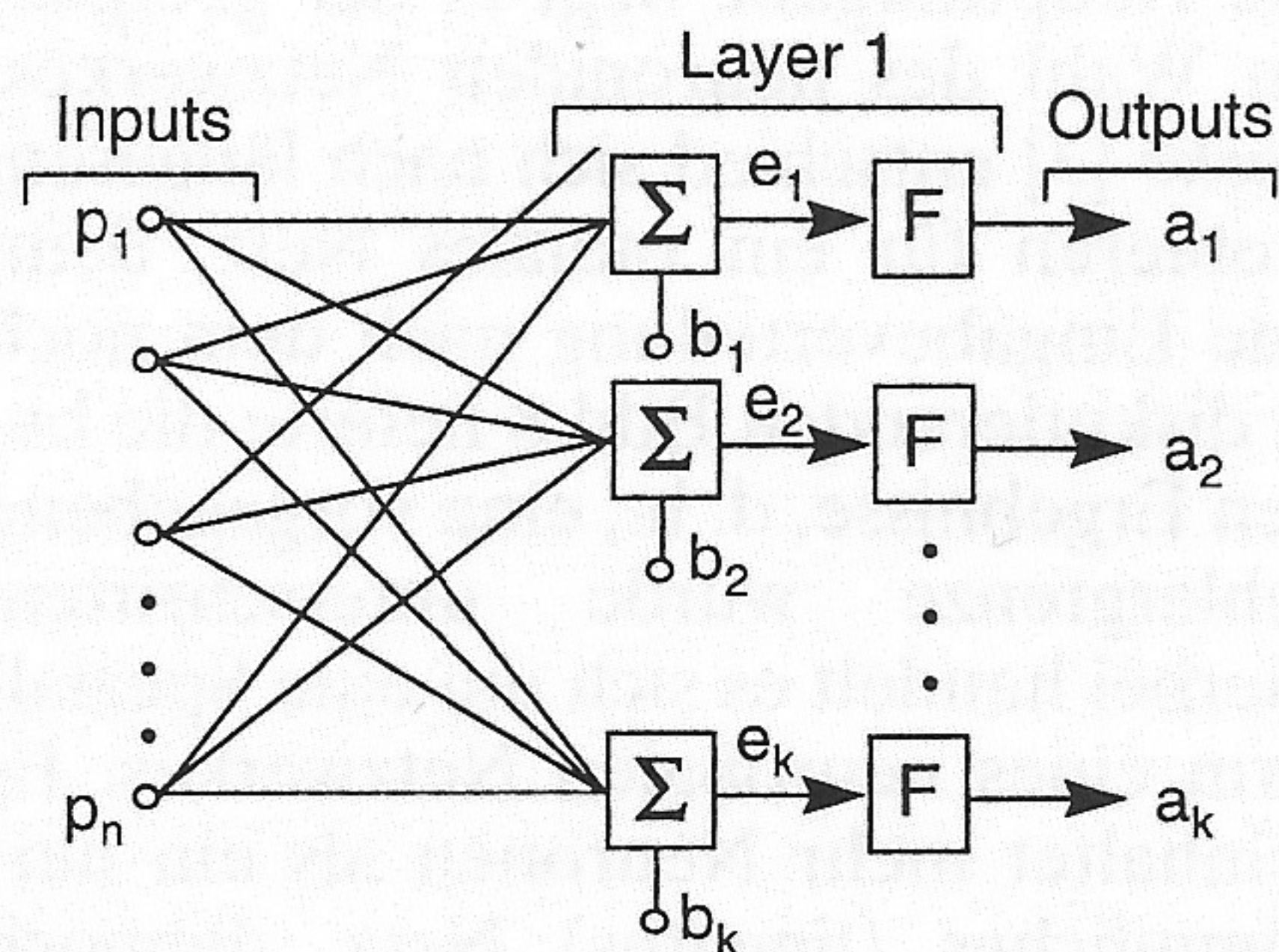


Bild 4 Netzwerk mit einer Neuronenschicht (Layer) aus k Neuronen und n Inputs

Zunächst ist jeder Input p_i – als Teil des Inputvektors $\underline{p}$ vom Typ (n,1) – mit jedem Neuroneneingang über die Wichtungsmatrix $\underline{w}$ – jetzt vom Typ (k,n) – verknüpft. Das Ergebnis $\underline{w} \cdot \underline{p}$ ist nun vom Typ (k,1), das mit den Verschiebungen $b_1 \dots b_k$, d.h. mit dem Spaltenvektor $\underline{b}$ zusammengefaßt werden kann

$$\underline{e} = \underline{w} \cdot \underline{p} + \underline{b}. \quad (4)$$

Die Eingangsgröße $\underline{e}$ stellt nun wiederum einen Spaltenvektor dar, so daß $e_1 \dots e_k$ Eingangswerte entsprechend der Neuronenzahl k pro Schicht verfügbar sind. Schließlich folgen die Ergebnisse (Outputs) $a_1 \dots a_k$, als Ergebnisvektor $\underline{a}$ geschrieben zu

$$\underline{a} = F(\underline{e}) = F(\underline{w} \cdot \underline{p} + \underline{b}). \quad (5)$$

Bei schichtweiser Hintereinanderschaltung der Neuronen spricht man von multiplen Layern (Bild 5). Dadurch entsteht

(Layers) s mit 1, 2, ... und beachtet man, daß jeder Schicht s eine eigene Wichtungsmatrix $\underline{w}_s$ sowie ein eigener Verschiebungsvektor $\underline{b}_s$ zugeordnet sind, so folgen die Stufen:

$$a_1 = F_1(\underline{w}_1 \cdot \underline{p} + \underline{b}_1) \quad (6)$$

$$a_2 = F_2(\underline{w}_2 \cdot \underline{a}_1 + \underline{b}_2) \quad (7)$$

$$= F_2(\underline{w}_2 \cdot F_1(\underline{w}_1 \cdot \underline{p} + \underline{b}_1) + \underline{b}_2)$$

$$a_3 = F_3(\underline{w}_3 \cdot F_2(\underline{w}_2 \cdot F_1(\underline{w}_1 \cdot \underline{p} + \underline{b}_1) + \underline{b}_2) + \underline{b}_3). \quad (8)$$

Die Anzahl der Schichten, ihre integrierten Neuronen, der Typ der Transferfunktionen und die Verbindungen der Schichten untereinander subsumiert man unter dem Begriff Architektur des neuronalen Netzwerkes. Die Verknüpfung der Schichten kann auch bedeutend

eignet. Ein Polynom dritten Grades wäre nicht verwendbar, obwohl es die anspruchsvollste Approximation darstellte. D. h. zuviel Intelligenz kann auch „schaden“. Genauso wie man den richtigen Menschen für eine bestimmte Aufgabe finden muß, ist das richtige neuronale Netz für die jeweilige Aufgabe zu ermitteln, und dies ist mitunter recht aufwendig. Das konfigurierte Netz wird einer Lern- bzw. Trainingsphase unterzogen. In der Realität gehört zu jedem Inputvektor $\underline{p}$ ein Ergebnisvektor $\underline{t}$, auch Zielvektor genannt. Das trainierte Netz liefert aber einen Outputvektor $\underline{a}$, so daß der Fehler

$$\underline{f} = \underline{t} - \underline{a} \quad (9)$$

vorliegt. Zur Bewertung werden intern meistens die Fehlerquadrate herangezogen. Weiterhin gibt es verschiedene Lernregeln, um die Ermittlung der Wichtungs- und Verschiebungsmatrizen $\underline{w}$ bzw. $\underline{b}$ vorzunehmen. Außer dem Input und dem Fehler wird eine Lernrate, die als Änderungs- bzw. Relaxationsfaktor zu verstehen ist, berücksichtigt. Diese beeinflusst das Ergebnis mitunter stark. So kann der Fehler mit Zunahme der Lernrate rasch sinken, mit

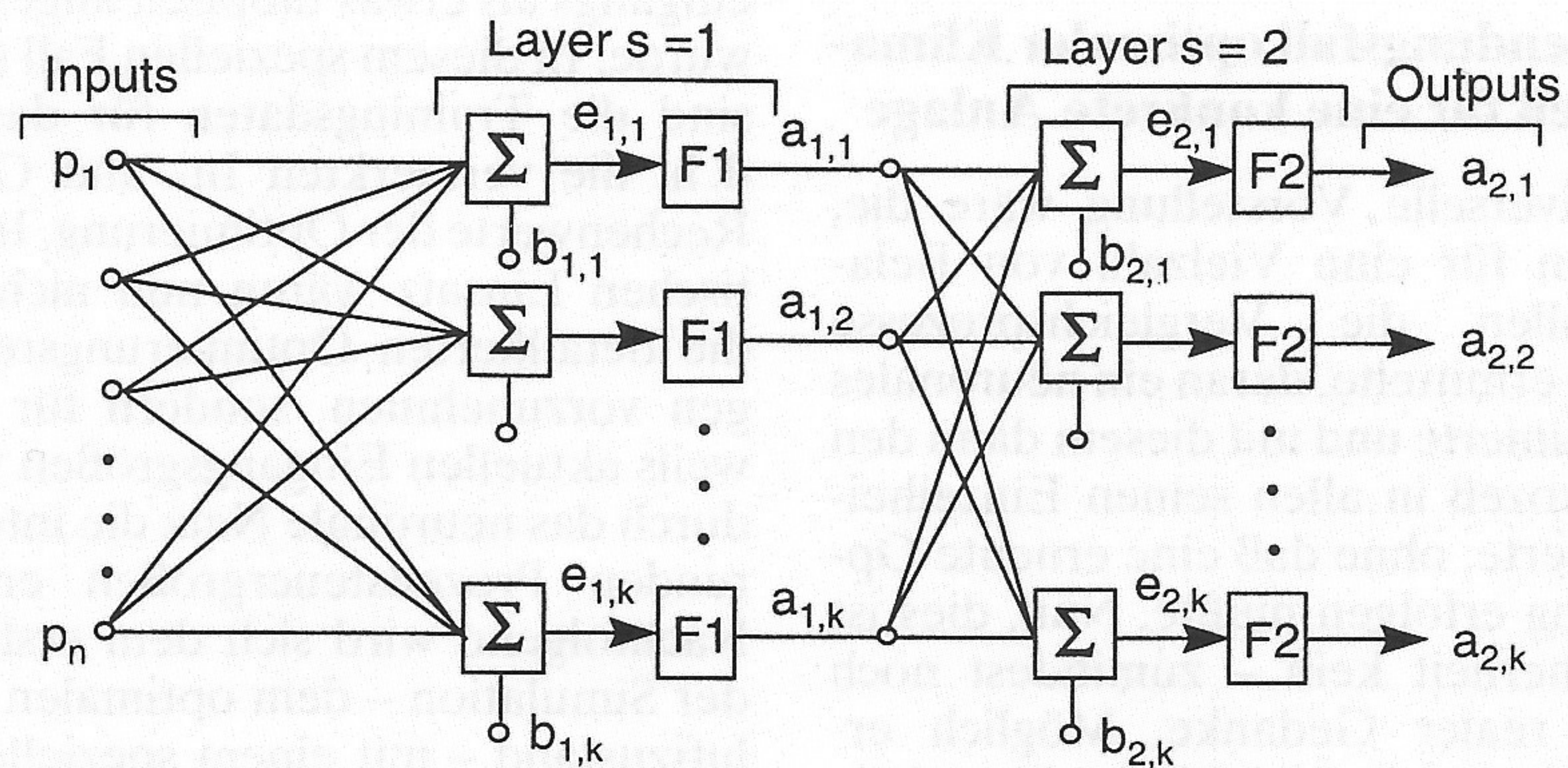


Bild 5 Netzwerk mit zwei Neuronenschichten (Layer 1, Layer 2) und n Inputs sowie deren Verknüpfungen

komplizierter sein, als bisher dargestellt wurde, so werden mitunter auch zusätzliche Ergebnissrückführungen eingebaut. Die Netzwerkstruktur ist dem praktischen Problem, das zu lösen ist, anzupassen. Dies ist sehr schwierig und muß in vielen Fällen probiert werden. Ein Exkurs in die Mathematik soll dies an einem einfachen Beispiel illustrieren (Bild 6). Es soll eine Funktion gefunden werden, die drei gegebene Punkte enthält und Zwischenwerte sinnvoll wiedergibt. Eine Gerade erfüllt dies nur in grober Näherung. Ein Polynom zweiten Grades ist vorzüglich ge-

größerer Zahl aber wieder ansteigen bzw. Instabilitäten hervorrufen. Das Training wird solange durchgeführt, bis der Fehler unter eine vorgegebene Fehlerschranke gesunken ist oder die ebenfalls vorgegebene maximale Durchlaufzahl (Epochen) erreicht ist.

Das trainierte Netzwerk ist nun mit Werten zu testen, die nicht im Training verwendet wurden. Waren das Netzwerk

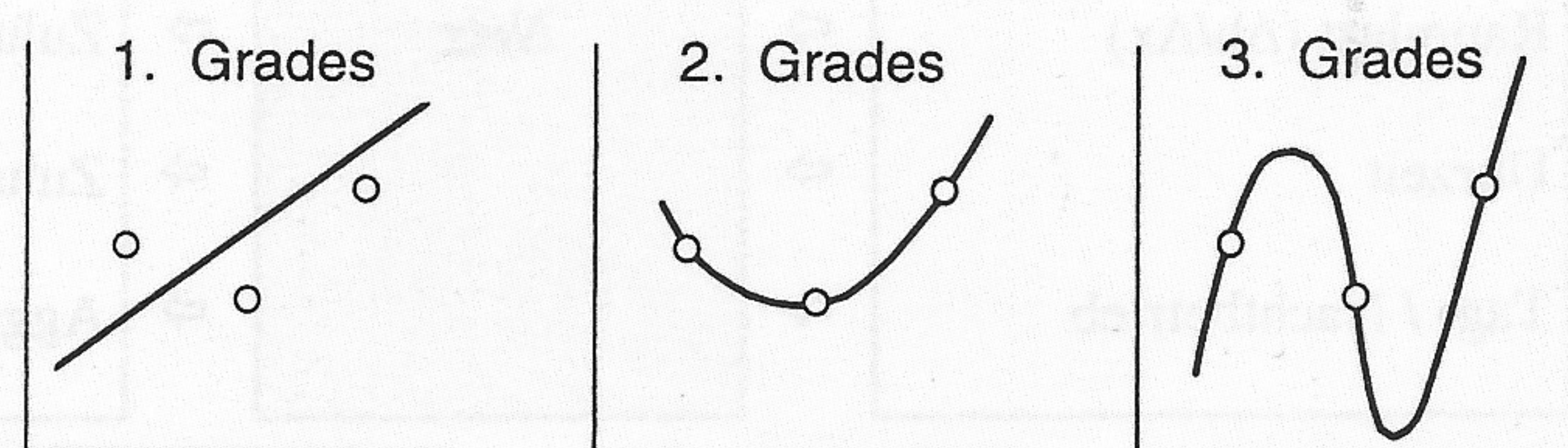


Bild 6 Approximation von drei gegebenen Punkten durch ein Polynom

und die zum Training ausgewählten Datensätze geeignet sowie die Lernrate optimal gewählt, dann werden die nun bereitgestellten Testeingaben auch Ergebnisse produzieren, die nahe an den realen Ergebnissen liegen. Ansonsten ist das Netzwerk ungeeignet, die Trainingsdaten waren nicht repräsentativ oder das Training wurde zu früh abgebrochen.

Das Vorgehen beim Training könnte folgendermaßen erfolgen. Wären 300 Datensätze vorhanden, so würde man z. B. 240 zum Training des Netzes mit dem Ziel der Fehlerminimierung verwenden. Die restlichen 60 Datensätze sind für Proberechnungen mit dem trainierten Netz verfügbar. Die Auswahl der Datensätze zum Netztraining beeinflusst die Leistung des neuronalen Netzes.

Einen Sonderfall bilden die adaptiven Netzwerke. Bei diesen werden die Daten neuer Zeitschritte zur verbesserten Bestimmung der Wichtungs- und Verschiebungsmatrizen $\underline{w}$ bzw. $\underline{b}$ benutzt.

Die Entwicklung der neuronalen Netze – hier wurde nur ein lineares Netzwerk erläutert – birgt noch viel Potential in sich. Die spätere Anwendung benutzt bereits eine kompliziertere Version.

3 Anwendungsfall optimaler Klimaprozeß für eine konkrete Anlage

Die universelle Vorstellung wäre die, daß man für eine Vielzahl von Belastungsfällen die Vergleichsprozesse nach [1] ermittelte, daran ein neuronales Netz trainierte und mit diesem dann den Klimaprozeß in allen seinen Einzelheiten steuerte, ohne daß eine erneute Optimierung erfolgen müßte. Nun, dies ist mit Sicherheit kein – zumindest noch kein – realer Gedanke. Möglich erscheint dagegen, diese Methodik auf die Bestimmung des optimalen Raumzustandes anzuwenden. Man könnte im Ergebnis der zahlreichen Optimierungen beispielsweise jedem Außenluftzustand einen optimalen Raumlufzustand zuordnen. Dies wird im mathematischen Sinne nicht eindeutig möglich sein, da

die optimale Prozeßführung nicht nur vom Ausgangspunkt (Außenluftzustand), sondern z. B. auch von der Raumbelastung abhängt. Zwischen den äußeren Bedingungen und der Raumbelastung besteht in der Regel eine stochastische Bindung mit mehr oder weniger starker Korrelation. Die von den äußeren Bedingungen unabhängige innere Belastung kann jedoch bedeutend stärker die Raumbelastung beeinflussen, die dann keine Berücksichtigung findet. Dies bedeutet, zumindest die Raumbelastung müßte ein weiterer Eingangsparameter sein. Außerdem sollten zeitbezogene Abhängigkeiten Beachtung finden. Bild 7 zeigt eine mögliche Zusammenstellung der Eingangsgrößen. Als Ausgangsgrößen wären die Raumlufthtemperatur und -feuchte bereits ein sehr wertvolles Ergebnis, da man den optimalen Zustandspunkt in einem vorgegebenen Behaglichkeitsbereich – der angesteuert werden müßte – kennen würde. Damit wäre ein erster Schritt – die optimale Prozeßführung zu realisieren – möglich. Hätte man auch die durch *) gekennzeichneten Größen verfügbar, könnte noch eine detailliertere Umsetzung des optimierten Prozesses erfolgen, wie dies eingangs als etwas utopisch angekündigt wurde. In diesem speziellen Fall (Bild 7) sind die Trainingsdaten für das Netz, d. h. die vermerkten In- und Outputs, Rechenwerte der Optimierung. Im praktischen Einsatz wären nun nicht mehr die detaillierten Optimierungsrechnungen vorzunehmen, sondern für die jeweils aktuellen Eingangsgrößen würden durch das neuronale Netz die interessierenden Prozeßsteuergrößen ermittelt. Nachfolgend wird sich dem ersten Teil der Simulation – dem optimalen Raumlufzustand – mit einem speziellen neuronalen Netz zugewandt.

4 Anwendungsfall optimaler Raumzustand für eine konkrete Anlage

Für eine bestimmte Klimaanlage wurden mit dem Optimierungsalgorithmus

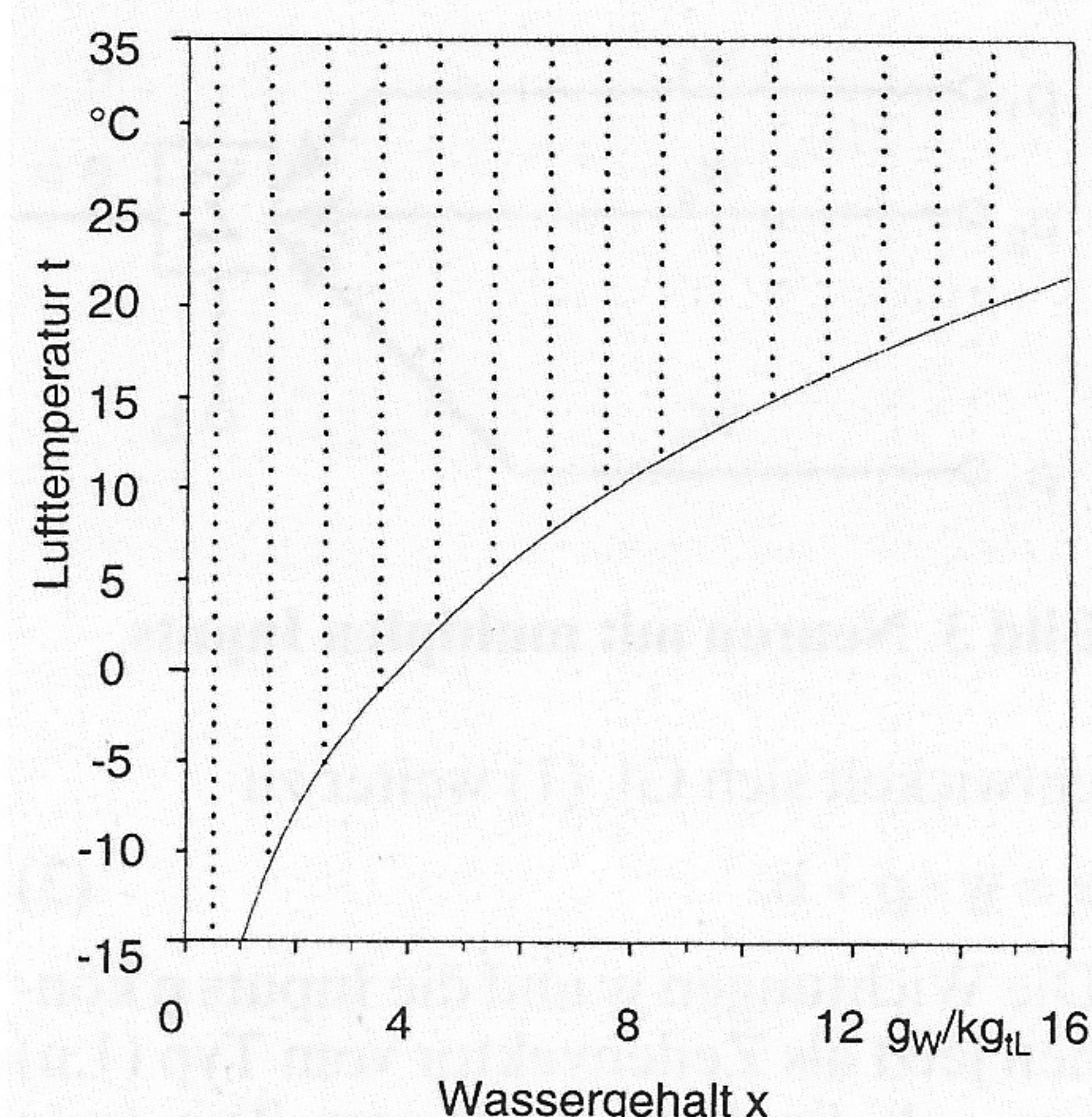


Bild 8 Verteilung der Außenluftzustände zum Training eines neuronalen Netzes (1. Modell: gleichmäßige Verteilung im theoretisch möglichen Gebiet)

nach [1] Beispielrechnungen durchgeführt. Von Interesse soll die Zuordnung der optimalen Raumparameter (Temperatur t_R und Feuchte x_R) zu den Außenluftparametern sein, wobei die teilweise schwache Korrelation zu der definierten Raumbelastung (Δh , Δx) als Störgröße wirkt.

Die Hauptaufgabe liegt in der geeigneten Wahl des neuronalen Netzwerkes. Starke [3] entschied sich nach längerem Probieren für ein radiales Netz, denn eine Eingabeverteilung nach dem noch zu diskutierenden Bild 8 lieferte die besten Ergebnisse, d. h., eine vorgegebene Fehlergrenze wurde unterschritten. Hierbei handelt es sich um eine Spezialform eines neuronalen Netzwerkes. Es beinhaltet mehr Neuronen als ein herkömmliches (lineares) Netz, dennoch kann die Konfiguration des Netzes sehr schnell erfolgen. Das Netzwerk selbst besteht aus einer ersten Schicht – einem sogenannten radialen Basislayer – und einer folgenden linearen Schicht. Kernstück des Basislayers ist die Transferfunktion, die im Verlauf der Gaußverteilung ähnelt. Sie hat bei der Eingabe Null den Wert Eins, bei von Null abweichenden Eingaben fällt ihr Wert nach beiden Seiten symmetrisch ab. Die Verarbeitung der Matrizen $\underline{w}$ und $\underline{b}$ mit dem Input ist jedoch stark verändert gegenüber den linearen Netzwerken, wobei $\underline{b}$ die Sensitivität der inneren Neuronenschicht sehr beeinflusst. Beim Training des radialen Netzes werden zur Anpassung (Minimierung der Fehlerquadrate) Neuronen bis zu einer maximalen vorgegebenen Anzahl hinzugefügt.

Weiterhin gab man sich vor, stets 80 % der jeweils vorhandenen optimalen Werte zum Training des Netzwerkes zu nutzen und die restlichen 20 % der Werte für den abschließenden Test zu verwenden.

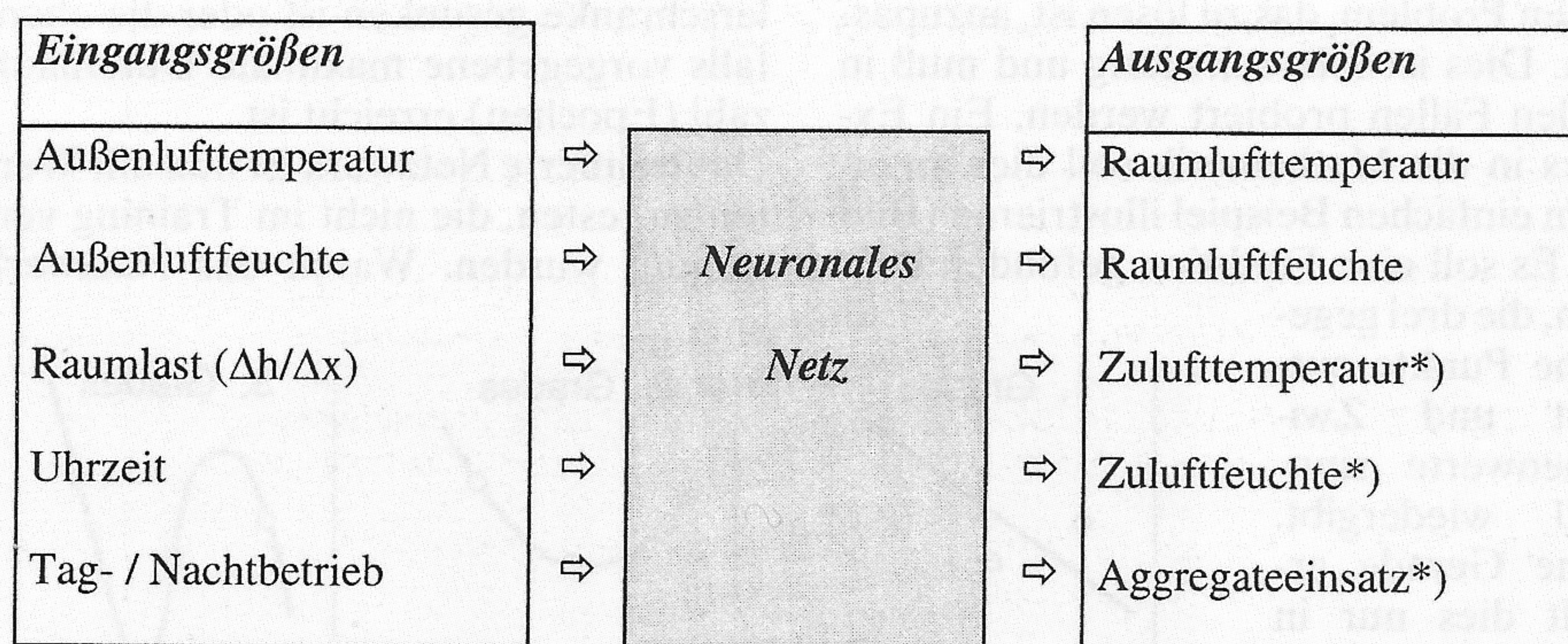


Bild 7 Beispiel für spezielle Ein- und Ausgangsgrößen zum Training eines neuronalen Netzes. Die Outputs entstammen hierbei rechnerisch ermittelten, optimalen Vergleichsprozessen

Die Auswahl der Außenluftzustände, für die die optimalen Prozeßdaten mit dem Optimierungsprogramm bestimmt werden, ist nach unterschiedlichen Gesichtspunkten möglich.

Im *ersten Modell* ging man davon aus, daß rein theoretisch alle Außenluftzustände zwischen $t_a = -15 \dots 35^\circ\text{C}$ und $x_a = 0 \dots 16 \text{ g}_\text{W}/\text{kg}_\text{tL}$ vorkommen können. Deshalb wurde eine homogene Verteilung über diesen Bereich mit insgesamt 430 Punkten – wie im Bild 8 dargestellt – gewählt. (Ein Fußballbegeisterter würde von idealer Raumdeckung sprechen.) Die zum Training verwendeten Werte sind aus der gegebenen Datenfülle ebenfalls gleich verteilt ausgewählt. Die Proberechnungen mit den restlichen 20 % der Werte (86 Stück) führten zu guten Ergebnissen, teilweise aber auch zu großen Ausreißern. Bild 9 zeigt die Abweichungen der optimalen Raumtemperaturen (Kreuze), die im Bereich von -3 K bis $+9 \text{ K}$ liegen. Bei der Raumluftfeuchte ergeben sich ähnlich große Abweichungen von $-1,3 \text{ g}_\text{W}/\text{kg}_\text{tL}$ bis $+2,2 \text{ g}_\text{W}/\text{kg}_\text{tL}$.

Im *zweiten Modell* wollte man realistischer sein und verwendete eine Verteilung der Außenluftzustände nach dem

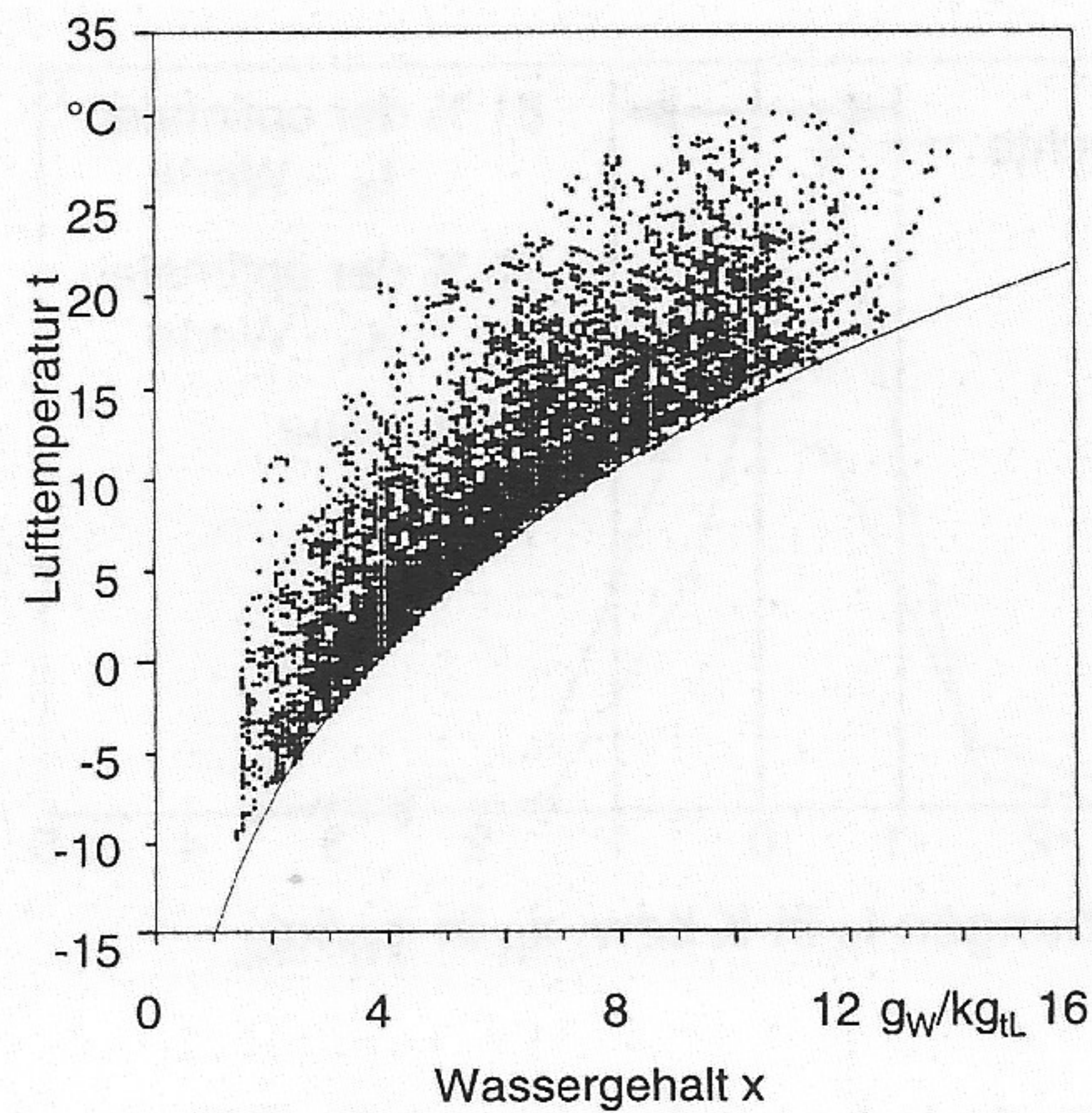


Bild 10 Verteilung der Außenluftzustände zum Training eines neuronalen Netzes (2. Modell: alle Daten des Testreferenzjahres TRY 03)

Testreferenzjahr 03. Damit standen 8760 Stundenwerte zur Verfügung, so wie sie im langjährigen mittleren Wettergeschehen auch zu erwarten sind. Sie sind im Bild 10 eingetragen. (Der Fußballsachverständige würde in diesem Fall von einer idealen Manndeckung sprechen.) Obwohl man annahm, ein besseres Modell zu haben, ergab sich überhaupt keine Lösung. Die Datenfülle und die enge Punktbelegung sind mögliche Ursachen. Damit wurde deutlich, daß man auf jeden Fall die Ausgangsdatensätze qualifiziert auswählen muß. Literaturhinweise zur Lösung des vorliegenden Problems wurden nicht gefunden.

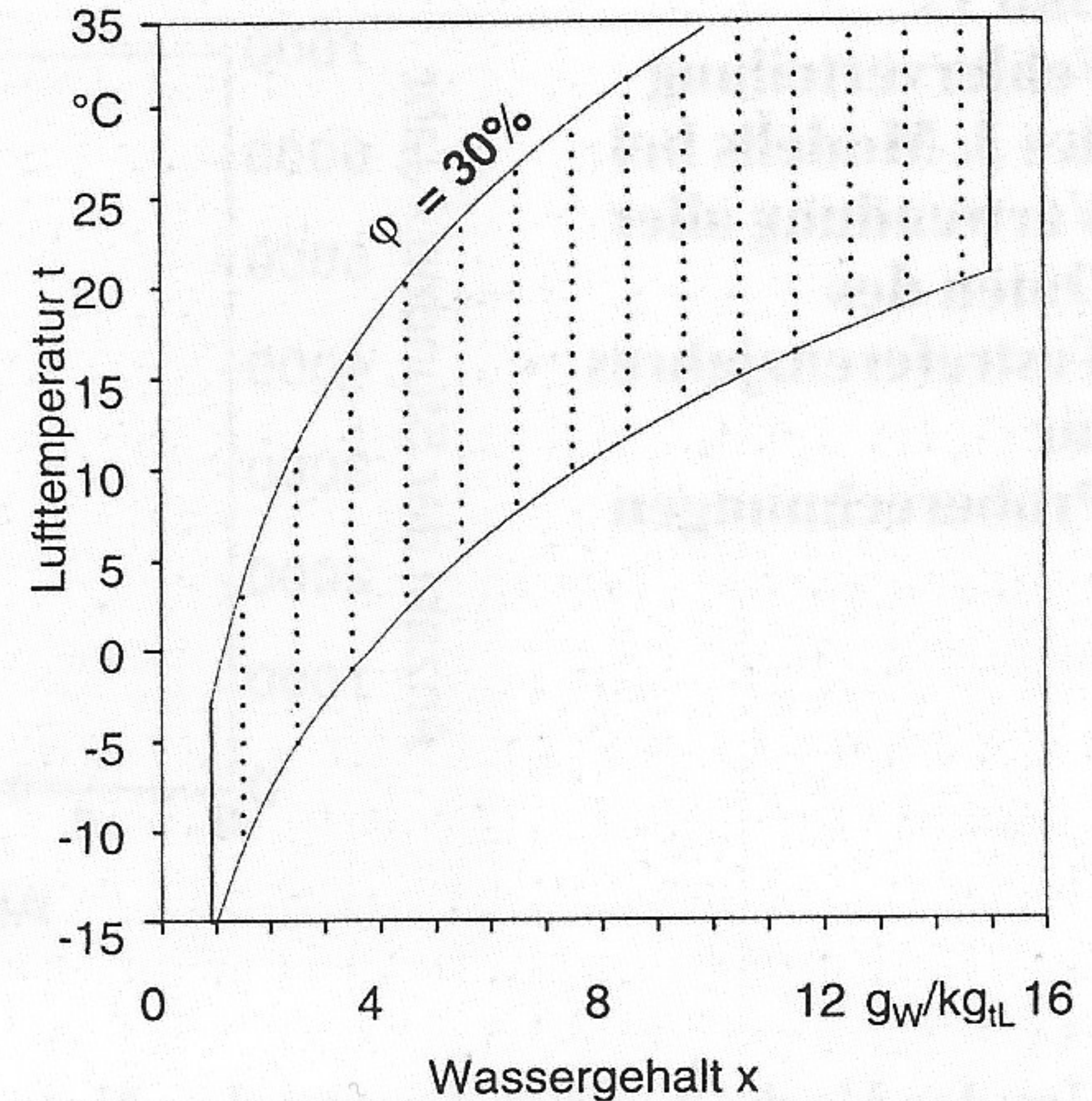


Bild 11 Verteilung der Außenluftzustände zum Training eines neuronalen Netzes (3. Modell: gleichmäßige Verteilung im praktisch vorkommenden Gebiet)

Als *drittes Modell* wurde eine rein pragmatische Lösung kreiert. Ausgehend von der Verteilung nach dem Testreferenzjahr ist in diesem Gebiet eine gleichmäßige Verteilung im realen Bereich der Außenluftzustände angenommen worden. Bild 11 zeigt 253 Punkte. (In der Sprache der Fußballanhänger könnte man dies eine eingeschränkte Raumdeckung nennen. Denn wenn man weiß, daß der Gegner aus Erfahrung stets einen Seitenflügel vernachlässigt, dann muß dort auch der Raum nicht besetzt werden.) Das Netz wurde nun mit 80 % der fixierten Außenluftzustände trainiert. Es folgte ein erfreuliches Ergebnis. Die Fehlerzusammenstellung der optimalen Raumtemperaturen schwankt nur noch im Bereich von -3 K bis $+2 \text{ K}$. Bild 9 zeigt die Gesamtverteilung aller 51 Proberechnungen (Kreise) deutlich. Bei der Raumluftfeuchte sind die Abweichungen von $-1 \text{ g}_\text{W}/\text{kg}_\text{tL}$ bis $+1,2 \text{ g}_\text{W}/\text{kg}_\text{tL}$ auch deutlich kleiner als im ersten Modellfall.

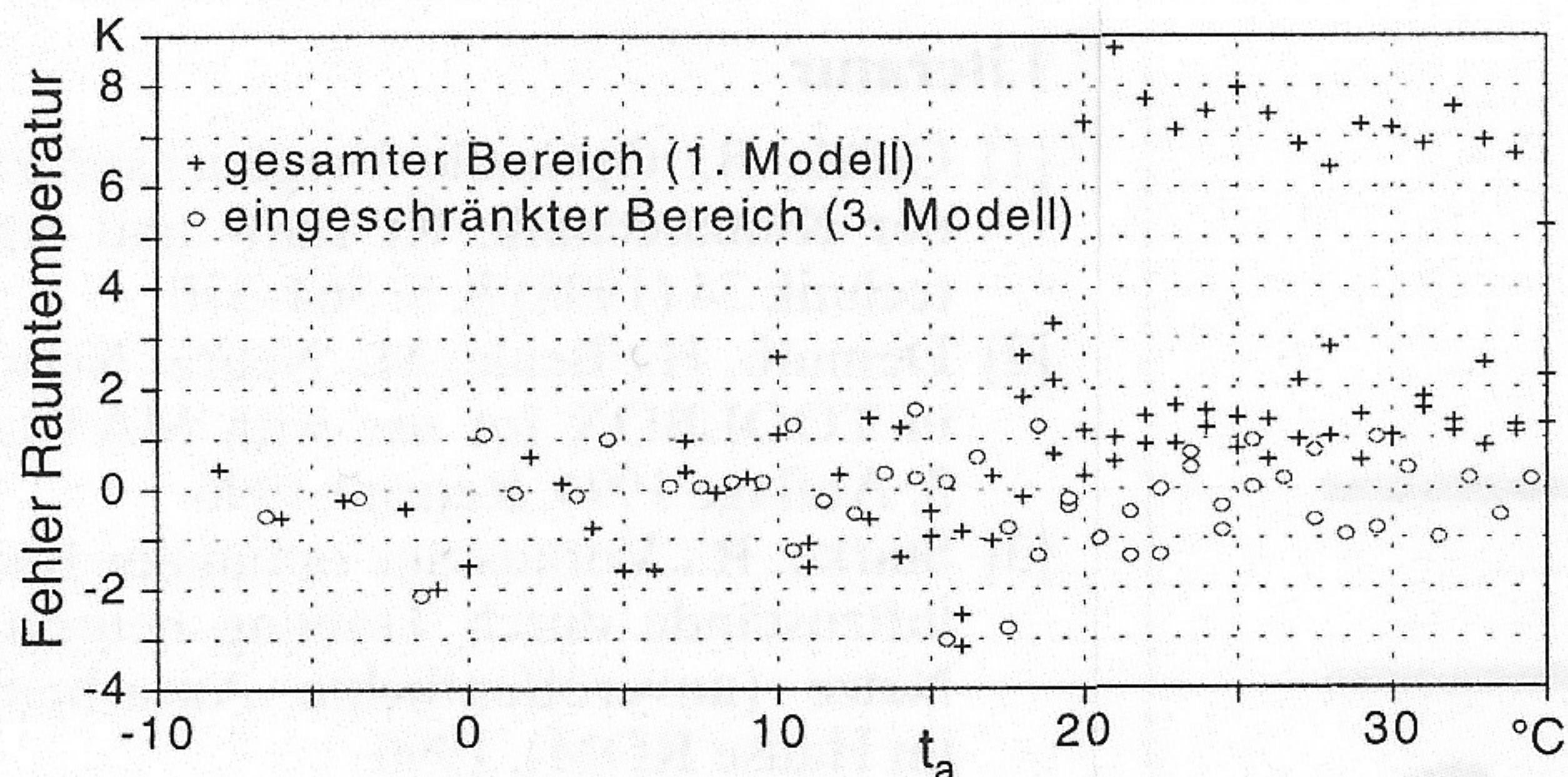
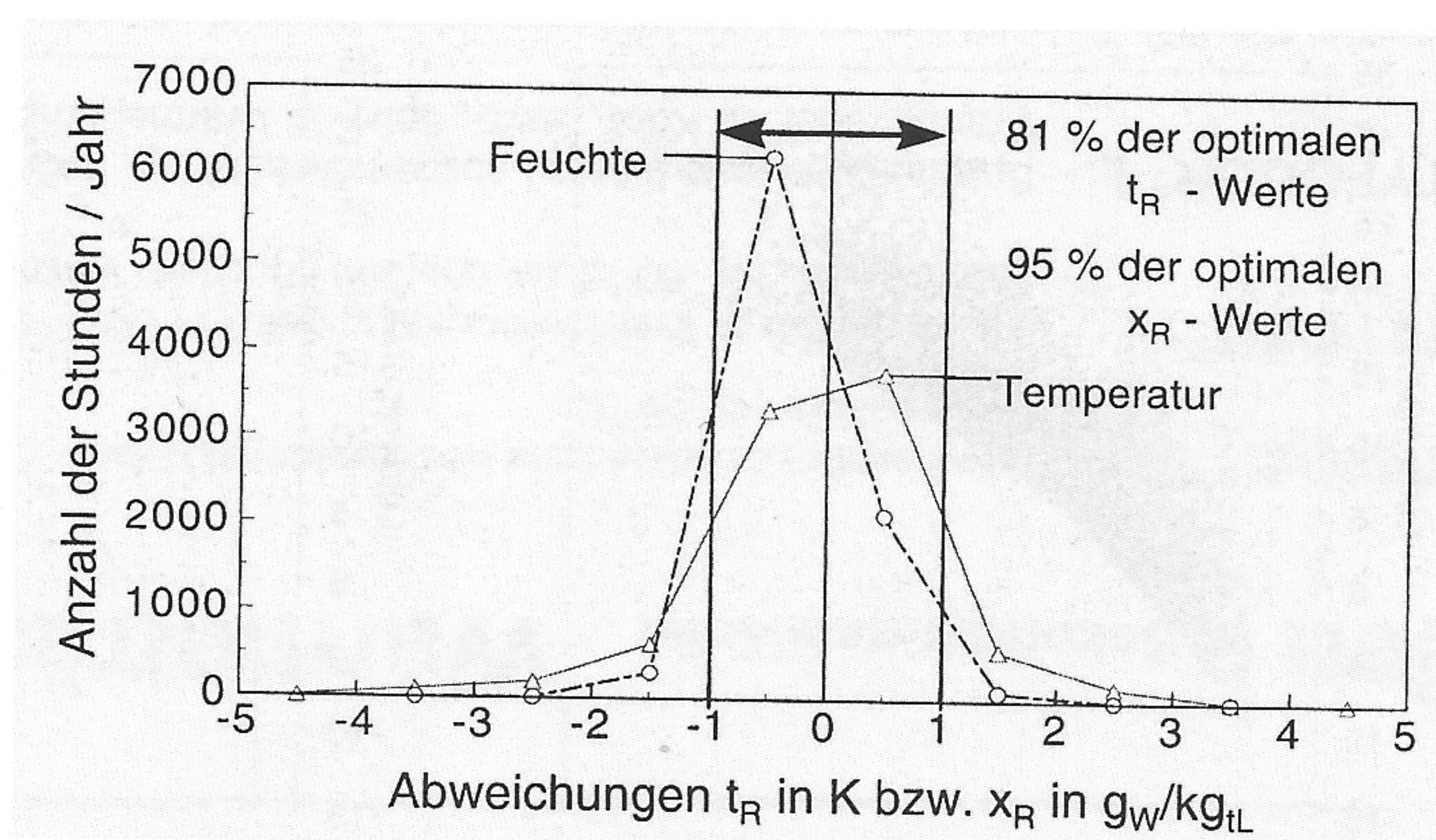


Bild 9 Fehlerverteilung. Verteilung der Temperaturabweichungen von den optimalen Werten bei Einsatz des 1. und 3. Modells bei Verwendung der Probedaten (20 % der bekannten Daten)

Bild 12
Fehlerverteilung
des 3. Modells bei
Verwendung aller
Daten des
Testreferenzjahres
für
Proberechnungen



Um das Verhalten des neuronalen Netzwerkes nach dem dritten Modell umfassend beurteilen zu können, wurden alle Daten des Testreferenzjahres – also 8760 Wertepaare – für Proberechnungen verwendet. Die Verteilung der Abweichungen von den optimalen t_R - und x_R -Werten mit den zugehörigen Stundenzahlen sind Bild 12 zu entnehmen. Wie sich zeigt, liegen 81 % der Raumtemperaturen innerhalb eines Fehlerbereiches von ± 1 K und 95 % der Raumfeuchten innerhalb eines Bereiches von ± 1 g_W/kg_{tL}. Dies ist ein doch recht gutes Ergebnis. Grundsätzlich ist der Einbau eines Si-

cherheitsbausteines zweckmäßig, um überraschenden Fehlern vorzubeugen. Man würde beispielsweise das Ergebnis prüfen, ob es im zulässigen Behaglichkeitsfeld liegt. D. h., wenn es sich um einen „Ausreißer“ handelt, dann sollte der Vorgängerzustand beibehalten und damit ein stabiler Anlagenbetrieb gesichert werden.

5 Fazit

Der Einsatz von neuronalen Netzen kann im Falle der optimalen Prozeßführung vorteilhaft sein, da nicht der

eigentliche Optimierungsalgorithmus eingesetzt werden muß. Dies bringt Zeitersparnis und Sicherheit gegenüber nie in allen Punkten ausgetesteten Algorithmen. Allerdings gilt das trainierte Netzwerk nur für die jeweilige Klimaanlage mit den verwendeten Randbedingungen, z. B. dem Lastspektrum. Zu den Randbedingungen gehören bei einer Optimierung mit der Zielfunktion „minimale Kosten“ natürlich auch die Preisrelationen der Energieträger. Das Netz ist somit bei einem Umbau der Anlage, bei Energiepreisänderungen usw. neu zu trainieren. Dies könnte dann auch eine Aufgabe im Rahmen des Anlagenservices sein.

Es sei abschließend aber auch noch einmal deutlich gemacht, daß im vorliegenden Fall lediglich die optimalen Raumzustände Einsatzgegenstand des neuronalen Netzwerkes waren. Man kann sich leicht vorstellen, wie sich die Probleme potenzieren würden, wenn man zusätzlich direkte Aussagen zur Steuerung der Einzelaggregate in der Anlage verlangte.

Momentan wird der Einsatz noch nicht empfohlen, dennoch sollte dem Verfahren weiterhin Beachtung geschenkt werden.

Literatur

- [1] Glück, B.: Optimale Vergleichsprozesse der Klimatechnik, Ki Luft- und Kältetechnik 34 (1998) 8, S. 364–370
- [2] Demuth, H.; Beale, M.: Neural Network in TOOLBOX for use with MATLAB. 3. Auflage 1994, Reprint 1996
- [3] Starke, R.: Voraussage optimaler Raumluftzustände durch Training neuronaler Netze (unveröffentlichte Ausarbeitung im Hause ROM), 1996

Schlüsselwörter

Raumluftzustand
Simulation
neuronales Netz
Optimierungsalgorithmus