Aufgabe: Sie wollen die Oberfläche der Einheitskugel auf eine Dezimale genau durch ein neuronales Netz trainieren. Wie sehen die Trainingsmuster an den 6 Intervallgrenzen aus? Wie gehen Sie bei der Ermittlung der Trainingsgewichte vor?

Lösung: Um den Durchmesser der Kugel auf eine Dezimalstelle genau zu quantisieren, bedarf es in jeder Raumrichtung 21 Werten. Das sind pro Raumrichtung 5 Bit. Wir müssen also jedes Trainingsmuster auf

$$\left(2^5\right)^3 = 2^{15} \equiv 15 \text{ Bit}$$

auslegen und brauchen dafür 3 Neuronen à 5 Bit. Die Bits 1-5 sind der x-Achse vorbehalten, die Bits 6-10 der y-Achse, und die Bits 11-15 reservieren wir für die z-Achse. Damit ist unser gültiger Wertebereich vollständig abgesteckt. Jede Achse und damit jedes Neuron besitzt also folgende Bitmuster für reelle Zahlen zwischen 0 und 2:[1]

0	0	0	0	0	0	0,7	0	0	1	1	1	1,4	0	1	1	1	0
0,1	0	0	0	0	1	0,8	0	1	0	0	0	1,5	0	1	1	1	1
0,2	0	0	0	1	0	0,9	0	1	0	0	1	1,6	1	0	0	0	0
0,3	0	0	0	1	1	1,0	0	1	0	1	0	1,7	1	0	0	0	1
0,4	0	0	1	0	0	1,1	0	1	0	1	1	1,8	1	0	0	1	0
0,5	0	0	1	0	1	1,2	0	1	1	0	0	1,9	1	0	0	1	1
0,6	0	0	1	1	0	1,3	0	1	1	0	1	2,0	1	0	1	0	0

Die 6 Eckpunkte der Kugeloberfläche mit Mittelpunkt $(1,1,1)$ entsprechen damit folgenden Trainingsmustern:[2]

$(1,1,0)$	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0
$(1,1,2)$	1	0	1	0	0	0	1	0	1	0	0	1	0	1	0
$(0,1,1)$	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0
$(2,1,1)$	0	1	0	1	0	0	1	0	1	0	1	0	1	0	0
$(1,0,1)$	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0
$(1,2,1)$	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0

Möchte man die gesamte Oberfläche äquidistant trainieren, so sollte man dazu besser kein sphärisches Koordinatensystem benutzen.[3] Statt dessen wählt man ein Koordinatensystem bestehend aus zwei zueinander senkrechten Ebenenscharen, damit die Einteilung in Längen- und Breitenkreise nicht so unterschiedlich ausfällt wie in einem geographischen Koordinatensystem

[1] Ändert sich die Zahl der Bits oder die gewünschte Genauigkeit, so ändert sich selbstverständlich auch das Bitmuster der reellen Zahlen.

[2] Unser Gehirn kann zwar keine reellen Zahlen interpretieren, dafür entwickelt der Mensch im Laufe seines Lebens ein Gefühl für die Naturgesetze. Er trifft zielgenau über einige Meter in den Abfalleimer.

[3] Das ist auch gar nicht nötig, weil es keine ausgezeichnete Achse gibt, um dies sich die Oberfläche dreht.

und damit es auf jede beliebige Fläche anwendbar ist.[4] Wir schneiden also zuerst den Äquator mit äquidistanten Geraden, die parallel zur y-Achse verlaufen. Sodann drehen wir die Äquatorialebene um die x-Achse und tragen die so erhaltenen Schnittpunkte auf der z-Achse ab. Damit haben wir n Kreisradien erhalten, die wir wiederum konzentrisch zum Mittelpunkt in die x-y-Ebene abbilden.[5] Alle Schnittpunkte, die wir dann noch mit den zur y-Achse parallel verlaufenden und zur x-Achse senkrechten Ebenen erhalten, bilden zusammen mit dem zugehörigen z-Wert die gesuchten Gitterpunkte. Für $x = x_n = (n-1)/10$ haben wir 21 Stützstellen in einem äquidistanten Abstand von

$$x_{n+1} - x_n = \frac{n}{10} - \frac{n-1}{10} = \frac{1}{10},$$

was genau einer Dezimalen entspricht. Die Gleichung der Oberfläche ist gegeben durch

$$z(x,y) = 1 \pm \sqrt{1-(x-1)^2-(y-1)^2}.$$

Zuerst wird diese mit den Ebenen

$$z_n = 1 \pm \sqrt{1-(x_n-1)^2}$$

geschnitten, d.h. für $z(x,y) = z_n$ lautet der so erhaltene Kreis

$$y(x) = 1 \pm \sqrt{(x_n-1)^2-(x-1)^2}.$$

Dieser liefert mit den Ebenen $x = x_m$ zum Schnitt gebracht die noch fehlende y-Koordinate unserer $(x,\, y)$-Paare:

$$y(x_m) = 1 \pm \sqrt{(x_n-1)^2-(x_m-1)^2}.$$

Für $x_m = x_n$ ist $y(x_m) = 1$ und es ergibt sich lediglich ein einzelner Schnittpunkt. In allen anderen Fällen erhalten wir auf der „Nordhalbkugel“ genau zwei Schnittpunkte des Breitenkreises mit der Ebene $x = x_m$ und ebenso viele auf der „Südhalbkugel“.

Die Werte für x_n und y_m in die Oberfläche eingesetzt liefert uns die Schnittpunkte sämtlicher Höhen- und Breitenkreise der Einheitskugel in kartesischen Koordinaten:

$$z(x_n,y_m) = 1 \pm \sqrt{1-(x_n-1)^2-(y_m-1)^2} = 1 \pm \sqrt{1-2(x_n-1)^2+(x_m-1)^2}.$$

Für $m = n,$ d.h. im Meridian, ist

[4] Vor allem auch auf jede Hyperfläche, weil Sphären in höheren Dimensionen ohnehin keinen Anschauungswert mehr besitzen

[5] Hier schwingt die Sphäre doch wieder irgendwie mit.

$$z(x_n, y_n) = 1 \pm \sqrt{1-(x_n-1)^2}.$$

Insgesamt besteht die Oberfläche aus

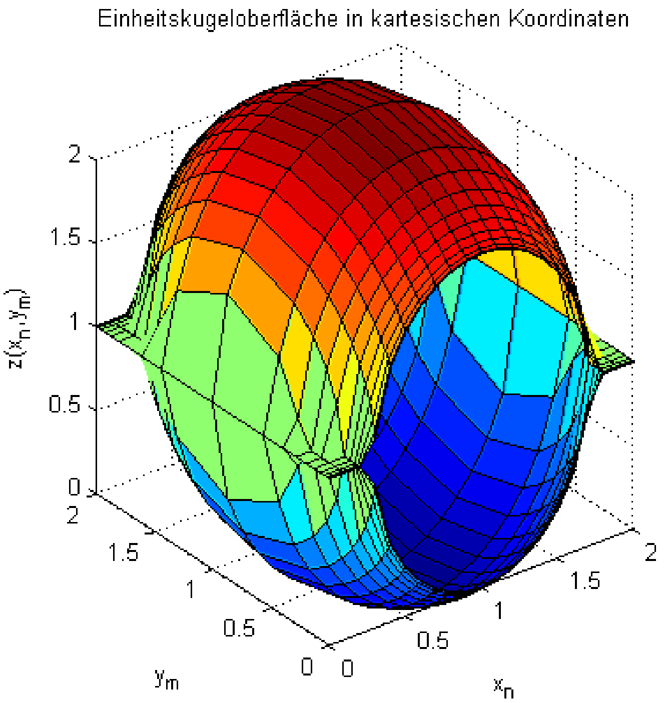
$$\sum_{n=0}^{19}(2n+1)+20 = 2\sum_{n=0}^{19} n + 20 = 19 \cdot 20 + 20 = 400$$

Elementen für die Halbsphäre. Das sind 760 Trainingsmuster $(x_n, y(x_m), z(x_n, y_m))$ für die komplette Sphäre, weil die 40 Äquatorelemente doppelt gezählt wurden. Die Trainingsmuster sind in nachfolgender Abbildung dargestellt. Greifen wir als Beispiel einen beliebigen Wert heraus, etwa $n = 9$ und $m = 17$, so ist

$$(x_9 = 0.8,\ y_{17} = 1.8,\ z(x_9, y_{17}) = 1.57).$$

Wo sich Zahlen ergeben, die nicht auf eine Dezimale genau angegeben werden können, muß gerundet werden. Üblicherweise wird aufgerundet, so daß der Trainingswert z ohne großen Fehler gleich 1,6 gesetzt werden kann. Das Trainingsmuster lautet in diesem Fall

$$(0.8, 1.8, 1.6) \,|\, 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0$$



Wir haben also gezeigt, daß jede beliebige Hyperfläche in neuronale Erkennungsmuster umgewandelt werden kann. Natürlich kann man diese Muster nicht zu Fuß ausrechnen, sondern muß die Rechnung zweckmäßigerweise einem Konversionsprogramm überlassen. Zunächst gilt es die Parameterbereiche abzustecken und dabei die Rechengenauigkeit festzulegen. Schließlich ist immer eine gewisse Zahl reeller Eingangsdaten zu runden, aber am Ende ist der komplette Wertebereich der Hyperfläche abgedeckt. Alle von den Neuronen eingehenden Zahlenwerte werden dann zunächst ebenfalls in ein Bitmuster umgewandelt, quasi in einer Art Vorverarbeitung, ehe das neuronale Netz darauf losgelassen wird. Stimmen alle Parameter konsistent mit einem Punkt der Hyperfläche überein, trifft folglich ein spezielles Trainingsmuster auf diesen Fall zu, muß keine Korrektur vorgenommen werden. Liegt das zu erkennende Muster indes daneben,[6] wird es durch ein Gradientenabstiegsverfahren in wenigen Schritten der Hyperfläche zugeführt. Die Trainingsmuster dienen dazu, die Gewichte des neuronalen Netzes zu bestimmen. Sind diese einmal festgelegt, ist der Lernprozeß im Idealfall abgeschlossen. Es hängt daher ganz von der Güte des klassischen Modells ab, wie gut oder schlecht das neuronale Netzwerk funktionieren wird. Hat die Vorlage nicht alles berücksichtigt, kann die Leistung auch durch ein neuronales Netz nicht übertroffen werden. Letzteres hat primär die Aufgebe, uns den vorgegebenen Weg zu weisen, den wir dazu aber kennen müssen. Ein neuronales Netz kann also immer nur so gut sein, wie es trainiert wurde. Man kann es sich in technischen Abläufen nicht leisten, solange abzuwarten, bis man die Naturgesetze intuitiv[7] erfaßt hat. Ein Mensch braucht dafür eventuell Jahre. Das wiederum ist der Vorteil der künstlichen Intelligenz.

[6] Das können mehr oder weniger starke Abweichungen sein.

[7] Durch try and error

