Praktikum Autonome Roboter – Projekt I

Sensordatenfusion mit selbstorganisierenden Karten

PD Dr. R. Der
Universität Leipzig
Institut für Informatik

April 7, 1997

Sensordatenfusion mit selbstorganisierenden Karten

1 Aufgabe: Bestimmung der relativen Stellung aus den IR Sensordaten

Eine wichtige Eigenschaft in der Navigation ist die Bestimmung der Stellung des AR relativ zu einem Hindernis oder einer Wand. Stellung meint dabei sowohl den Abstand als auch den Winkel. Diese zwei Parameter sind aus den Sensordaten zu bestimmen. Die IR Sensordaten sind bei hinreichender Annäherung an das Objekt ausreichend zur Bestimmung der Stellung des AR relativ zu dem Objekt. Die Schwierigkeit besteht in der unbekannten oder zumindest schlecht in eine einfache mathematische Form faßbaren Sensorcharakteristik und in der Aufgabe, aus den acht IR Sensoren und den acht Sensoren für die Umgebungsbeleuchtung (ambient light sensors) die zwei Parameter Abstand und Winkel zur Wand abzuleiten. Mathematisch entspricht das der Abbildung eines 8– bzw. 16-dimensionalen Raumes auf den 2–dimensionalen Raum. Ziel der Aufgabe ist das Erlernen dieser Abbildung während der Gebietsexploration durch den AR. Lassen wir die ambient-light Sensoren zunächst weg, so können die Sensoroutputs als Vektor im $\mathbf{R^8}$ dargestellt werden,

$$\vec{x} = (x_1, \ldots, x_8) \tag{1}$$

Die Aufgabe besteht in der Bestimmung und geeigneten Repräsentation der Funktion

$$\vec{y} = \vec{f}(\vec{x}), \qquad \vec{y} = (R, \alpha) \tag{2}$$

mit $\vec{y} \in \mathbf{R^2}$, die den Sensordaten den Abstand R und den Winkel α einer ausgezeichneten Achse relativ zur Wand zuordnet.

Die Idee ist nun, daß Sensordatenvektoren, die im $\mathbf{R}^8$ benachbart liegen, sich auf benachbarte Stellungen des AR beziehen müssen. Dann ist die Abbildung f also eine topografische, d. h. nachbarschaftserhaltende Abbildung, die mit dem Algorithmus von Kohonen aus den on–line anfallenden Sensordaten gelernt werden kann.

2 Selbstorganisierende Karten – der Kohonenalgorithmus

Unter einer Karte verstehen wir die **topografische** Abbildung eines Inputraumes V auf einen Outputraum $\mathcal{A}$. Solche Abbildungen werden durch spezielle einlagige neuronale Netze realisiert, d. h. wir haben eine Neuronenschicht, wobei jedes Neuron von allen Inputunits direkt gespeist wird. Die Topologie des Neuronenraumes $\mathcal{A}$ ist durch die Vorgabe der Orte $\vec{r}_i$ der Neuronen $i = 1, 2, ..., N$ festgelegt. Diese können die Knoten (Vertices) eines beliebigen Graphen sein, im konkreten Fall sollen die Neuronen auf den Gitterpunkten eines zweidimensionalen Quadratgitters sitzen, d. h. $\vec{r}_i = (n_{i_1}, n_{i_2})\ ,\ n_{i_j} \in \{0, 1, 2, ...\}$.

$$
\begin{array}{cccccccccc}
 & & Neuronen\ (Output) & & & & & & & \\
 & & \odot & \odot & \odot & \odot & \odot & \odot & \odot & \odot \\
 & \odot & \odot & \odot & \odot & \odot & \odot & \odot & \odot & \\
\odot & \odot & \odot & \odot & \odot & \odot & \odot & \odot & & \\
 & & \cdot & & \cdot & & \cdot & & & \qquad Kopplungen \\
 & & & \nwarrow & \uparrow & \nearrow & & & & \\
 & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & \\
 & & & & Inputunits & & & & &
\end{array}
\qquad (3)
$$

Jedes Neuron an einem Gitterplatz $\vec{r}$ ist durch den Vektor seiner synaptischen Gewichte $\vec{w}_{\vec{r}} \in V$ mit den Inputunits verbunden. Da der Vektor $\vec{w}_{\vec{r}}$ in V liegt, ist er gleichzeitig die Abbildung des Gitterplatzes $\vec{r}$ nach V. Die umgekehrte Abbildung $V \to \mathcal{A}$ ergibt sich durch die Vorschrift

$$\vec{x} \mapsto \vec{s} \quad falls\, ||\vec{w}_{\vec{s}} - \vec{x}|| \leq ||\vec{w}_{\vec{r}} - \vec{x}|| \quad \forall \vec{r}$$

oder

$$\vec{s}\,(\vec{x}) = \arg\min_{\vec{r}} ||\vec{w}_{\vec{r}} - \vec{x}|| \qquad (4)$$

Der Input $\vec{x}$ wird also immer auf den ihm am nächsten gelegenen synaptischen Vektor (SV) $\vec{w}_{\vec{r}}$ abgebildet. Das entspricht der Voronoi–Parkettierung des Inputraumes. Die Abbildung $\vec{s}\,(\vec{x})$ ist die diskrete Version einer Abbildung der Form 2.

Um eine **topografische** Abbildung $V \to \mathcal{A}$ zu erhalten, müssen benachbarte sensorische Inputs $\vec{x} \in V$ auf die gleichen oder benachbarte Gitterplätze abgebildet werden. Mit dem von Kohonen angegebenen Lernalgorithmus für die synaptischen Kopplungen bilden sich diese Karten in einem

Prozeß der Selbstorganisation heraus, daher der Name selbstorganisierende Karten (englisch self–organizing maps ⇒SOMs)[Kohonen84], [Kohonen95]. Der Lernalgorithmus umfaßt die folgenden Schritte:

1. Initialisierung aller synaptischen Gewichte, z. B. $\vec{w} = \vec{w}_{\vec{r}} = random$ $\forall \vec{r}$. Danach werden bei jeder Präsentation eines Inputs $\vec{x}$ folgende Schritte vollzogen:

2. **Siegerbestimmung**: Bestimmung des *best matching* SV's, d. h. finde Gitterplatz $\vec{s}$ (Neuronenort), dessen SV $\vec{w}_{\vec{s}}$ dem präsentierten Input am nächsten ist, d. h.

$$\vec{s} = \arg\min_{\vec{r}} ||\vec{w}_{\vec{r}} - \vec{x}|| \tag{5}$$

   vgl. 4.

3. **Lernschritt**: Dann wird für **alle** SV $\vec{w}_{\vec{r}}$ ein Lernschritt (Adaptionsschritt) durchgeführt:

$$\Delta \vec{w}_{\vec{r}} = \epsilon h_{\vec{r},\vec{s}} \left(\vec{x} - \vec{w}_{\vec{r}}\right) \tag{6}$$

   wobei die Nachbarschaftsfunktion

$$h_{\vec{r},\vec{s}} = \exp\left(-\frac{d\left(\vec{r},\vec{s}\right)^2}{2\sigma^2}\right) \tag{7}$$

   die Mitlernstärke der Neuronen am Ort $\vec{r}$ bei Siegerneuron $\vec{s}$ und $d\left(\vec{r},\vec{s}\right)$ der (Euklidische) Abstand zwischen dem Siegerneuron und dem betrachteten Neuron im Gitter ist, $d\left(\vec{r},\vec{s}\right) = \sqrt{\sum_{i=1}^{2}\left(r_i - s_i\right)^2}$ für das zweidimensionale Gitter.

"Gängige Parameter": $\epsilon = 0.1$ mit langsamer Abkühlung auf Werte $\epsilon < 0.01$. Die Nachbarschaftsreichweite σ wird im Laufe des Lernens von einem anfänglich hohen Wert (so daß die Nachbarschaft etwa ein Drittel aller Neuronen umfaßt) auf einen Wert $\sigma \cong 1$ "abgekühlt".

3 Anleitung

Das Training der Abbildung kann anfangs off–line erfolgen. Dazu lasse man den AR eine längere Serie von Bewegungen vor der Wand ausführen und trainiere mit den so erzeugten Sensordaten mittels des SNNS ein Kohonennetz. Das hat den Vorteil, daß die Sensordaten auch in ihrer zeitlichen Abfolge vermischt werden können, so daß die gelernte Abbildung unabhängig von der Abfolge der Stellungen des AR ist.

Für des echte on–line Lernen ist eventuell eine Aufmerksamkeitssteuerung zu integrieren.

References

[Kohonen95] T. Kohonen: The self—organizing map. Springer 1995

[Ritter93] H. Ritter, Th. Martinetz, K. Schulten: Neuronale Netze. Addison Wesley, New York, 1993

[Kohonen84] T. Kohonen: Self–Organization and associative memory. Springer Series in Information Science 8, 1984