Blatt 08 - Self-Organizing Maps

Abgabe: Bis Mittwoch, 14.Januar 2015, 10:00 Uhr
An: Tobias Rothenberger, <rothenb@informatik.uni-frankfurt.de>

Aufgaben:	Aufgabe 8.1		Aufgabe 8.2					Aufgabe 8.3	
	8.1a	8.1b	8.2a	8.2b	8.2c	8.2d	8.2e	8.3a	8.3b

Aufgabe 8.1 Grundfunktionen

In dieser Aufgabe wollen wir uns erst einmal einige grundlegende Eigenschaften von Kohonenkarten ansehen.

a. Trainieren Sie ein 1-dimensionales Kohonennetz, bestehend aus 256 Neuronen, mit zufälligen Punkten aus einem Quadrat (zwei-dim., uniform verteilten Zufallswerten aus einem zwei-dim. Intervall). Stellen Sie anschließend die gelernten Positionen der Gewichtsvektoren zusammen mit den Nachbarschaftsbeziehungen (den Verbindungen zwischen den Neuronen) für verschiedene Iterationen (inklusive der letzten Iteration) graphisch dar.
   Wurde das Netz korrekt trainiert, dann sollten Sie in der letzten Iteration eine Art Kurve erkennen, die sich durch den Eingaberaum (das 2-dimensionale Quadrat) schlängelt. Können Sie an dieser Kurve schon einige allgemeine Eigenschaften erkennen, die das Kohonennetz in Bezug auf benachbarte Punkte im Eingaberaum hat?

b. Lesen Sie als Nächstes die Punkte aus dem Datensatz der Datei `distribution.dt` ein. *Hinweis*: Die Verteilung

dieser Punkte ist nicht uniform so wie in Teil1, sondern weist in bestimmten Bereichen eine höhere Dichte auf. Trainieren Sie nun ein Kohonennetz, bestehend aus einem Gitter aus 24x24 Neuronen, mit den Punkten dieses Datensatzes und stellen Sie anschließend wieder die gelernten Positionen der Gewichtsvektoren zusammen mit den Nachbarschaftsbeziehungen (Verbundlinien) für verschiedene Iterationen (einschließlich der letzten Iteration) graphisch dar.
Über welche Eigenschaft scheint das Kohonennetz noch zu verfügen?

-

nach oben

Aufgabe 8.2 Veranschaulichung höherdimensionaler Daten

Kohonennetze sind im Gegensatz zu Neuronengas zwar schlechter darin, die Topologie (Nachbarschaftsbeziehungen) eines bestimmten Raumes zu lernen, dafür sind sie aber hervorragend dafür geeignet, *topologieapproximierende Abbildungen* von einem Eingaberaum in einen *festen* Ausgaberaum zu lernen. Der Ausgaberaum, der durch die feste Nachbarschaft der Ausgabeneuronen gegeben ist, ist dabei recht niedrigdimensional, meist werden 1, 2 oder 3 Raumdimensionen verwendet, und besteht meist aus einer Linie, einem Rechteck oder einem Kubus, während der Eingaberaum sehr viel mehr Dimensionen besitzen kann. Dabei werden die Nachbarschaftsverhältnisse des Eingaberaums weitestgehend erhalten, so daß Punkte, die im Eingaberaum benachbart waren, oft auch benachbarte Neuronen auf dem Rechteck (bzw.der Linie, dem Kubus,...) aktivieren.

Diese Eigenschaft kann man ausnutzen, um höherdimensionale Eingaberäume zu "kartographieren". Was damit genau gemeint ist, werden wir uns nun in diesem Aufgabenteil ansehen.

Wir betrachten uns den Diabetes-Datensatz (`diabetes1.dt`) aus der NNProblems-Datenbank. Ziel dieses Datensatzes ist die Diagnose von Diabetes bei Pima-Indianern. Der Datensatz besteht aus 8 Eingaben, die zu jeder Person persönliche und medizinische Daten wie Alter, Blutdruck, *body mass index* und Glukosekonzentration im Blutplasma, etc. enthalten, sowie 2 Ausgaben (1-aus-2-Kodierung), die angeben ob die betreffende Person Diabetes-Positiv oder -Negativ ist. Insgesamt sind Daten über 768 Personen im Alter zwischen 21 und 81 Jahren in diesem Datensatz enthalten.

Eine interessante Frage lautet nun, welche Struktur diese Daten besitzen. Sind die Datenpunkte für Diabetes-positive und -negative Personen eher zufällig im Raum verteilt? Oder gibt es Zusammenhänge, die sich etwa für ein Klassifikationssystem ausnutzen ließen? Sich mal eben 8-dimensionale Zahlenkolonnen anzusehen, bringt allerdings nicht sehr viel. Etwas Einsicht über die Struktur kann aber ein Kohonennetz liefern.

a. Lassen Sie sich den Diabetes-Datensatz (`diabetes1.dt`) einmal über die Konsole ausgeben. Was können Sie über die Verteilung der diabetes-positiven und -negativen Datensätze im 8-dimensionalen Eingaberaum folgern, wenn Sie sich die ausgegebenen Daten genauer ansehen?

b. Lesen Sie den Diabetes-Datensatz ein und trainieren Sie ein Kohonennetz bestehend aus 40x40 Neuronen darauf, eine topologieerhaltende Abbildung vom 8-dimensionalen Raum der Eingabevektoren des Diabetes-Datensatzes in ein 2-dimensionales Quadrat zu lernen.

Stellen Sie anschließend den 2-dimensionalen Ausgaberaum (das 40x40-Gitter der Neuronen) graphisch dar und gehen Sie alle Eingabevektoren des Diabetes-Datensatzes durch, wobei Sie jeweils das dadurch aktivierte Neuron geeignet nach dem Typ (Diabetes-Positiv oder -Negativ) einfärben. Wählen Sie für Neuronen, die sowohl bei Diabetes-positiven, als auch bei Diabetes-negativen Personen aktiv waren, eine gesonderte Farbe (dadurch wird kenntlich gemacht, daß das betreffende Neuron nicht sehr aussagekräftig ist). Anhand der Farbgebung sollte auch klar werden, welche Neuronen *nicht* aktiviert wurden.

c. Da wir nur sehr wenige (768) Datenpunkte haben, werden in der Graphik aus Teil a) in der Regel nur ca. die Hälfte aller Neuronen aktiviert werden. Denken Sie sich daher nun eine einfache Methode aus, wie Sie mit Hilfe der bestehenden Graphik aus Teil a) auch den nichtaktivierten Neuronen eine sinnvolle Klassenzugehörigkeit zuordnen können.
   Stellen Sie das Ergebnis dann wieder wie in Teil a) graphisch dar. Haben Sie alles richtig gemacht, dann erhalten Sie eine Graphik, auf der die sogenannten Entscheidungsbebiete für die beiden Klassen "diabetes-positiv" und "diabetes-negativ" zu erkennen sind.

d. Was fällt Ihnen auf, wenn Sie sich die Verteilung der

Neuronenaktivität der Graphik aus Teil a) bzw. die Entscheidungsgebiete aus Teil b) ansehen? Lassen sich Rückschlüsse darauf ziehen, wie die Datenpunkte im ursprünglichen 8-dimensionalen Eingaberaum verteilt sind?

e. Ein besonders schönes Beispiel für die Kartographierung eines höherdimensionalen Eingaberaumes gibt auch der Iris-Datensatz (`iris1.dt`) aus dem ersten Aufgabenblatt ab. Im Gegensatz zum Diabetes-Datensatz weist dieser eine extrem simple Struktur auf, die auf einer Kohonenkarte sehr gut sichtbar ist. Der Iris-Datensatz besteht aus 4 Eingaben, die die Höhe und Breite des Kelch- bzw. Kronblattes angeben und besitzt 3 Ausgaben (1-aus-3-Kodierung), die den Iristyp bestimmen.

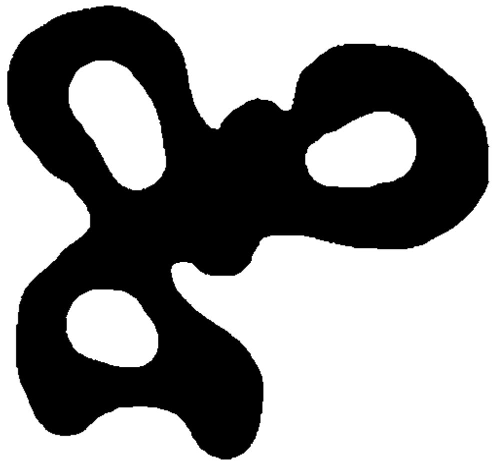
   Fertigen Sie wie in Teil a) eine Graphik an, die den 4-dimensionalen Eingaberaum kartiert. Was schließen Sie aus der Verteilung der einzelnen Datenpunkte? Wo liegen die Unterschiede zum Diabetes-Datensatz?

nach oben

Aufgabe 8.3 Neuronengas im Eingaberaum

Im Gegensatz zum Kohonennetz wird Neuronengas zu Beginn keine feste Nachbarschaft vorgegeben. Das Netz ist dadurch in der Lage, selbständig Nachbarschaftsbeziehungen im Eingaberaum zu erkennen und sich an die vorhandene "Form" dieses Raumes ziemlich genau anzupassen, ungeachtet wie verschlungen oder komplex dieser Raum ist. Ein kleines Beispiel dazu wollen wir uns einmal in diesem Aufgabenteil ansehen.

a. Lesen Sie das Bild aus der Datei `form.png` ein, siehe Abbildung.



Die schwarze Menge stellt den Eingaberaum dar, aus dem wir mit einer zufälligen Verteilung jeweils nacheinander einzelne Punkte ziehen und einem Neuronengas-Netzwerk präsentieren wollen.

Verwenden Sie den Algorithmus von Martinetz und Schulten, um das Netz darauf zu trainieren, bei Eingabe dieser Punkte den abgebildeten Eingaberaum möglichst gut zu lernen. Verwenden Sie dazu 256 Neuronen. Stellen Sie anschließend die gelernten Positionen der Gewichtsvektoren und die gelernten Nachbarschaftsbeziehungen (die Verbindungen zwischen den Neuronen) für verschiedene Iterationen, inklusive der letzten Iteration, graphisch dar.

b. Das Neuronengas-Netzwerk (wie auch in etwas schwächerer Form das Kohonennetz) hat die interessante

Eigenschaft, selbständig eine kompakte Repräsentation des Eingaberaums zu lernen. Die Gewichtsvektoren des Neuronengas-Netzwerkes stellen eine endliche Menge typischer Vektoren des Eingaberaums dar. Man spricht dabei auch von sogenannter Vektorquantisierung, mit denen es möglich ist, den originalen Eingaberaum wieder weitestgehend zu rekonstruieren.

Zeigen Sie, dass die Rekonstruktion möglich ist, indem Sie versuchen, aus den Gewichten und den Nachbarschaftsbeziehungen der Neuronen den originalen Eingaberaum wiederherzustellen. Stellen Sie das Ergebnis anschließend graphisch dar und vergleichen Sie es mit dem Original aus der Datei `form.png`.

Haben Sie eine Idee, welche Vorteile eine solche kompakte Repräsentation des Eingaberaumes durch Neuronengas bringen kann?

nach oben