

Simon Ziegler
Dr. sc. hum.

Lokalisierung neuronaler Strukturen in MRT Datensätzen

Promotionsfach: Neurologie
Doktormutter: Prof. Dr. Sabine Heiland

Die zentrale Idee dieser Arbeit besteht in der Verwendung von künstlichen neuronalen Netzen (KNN), um Gehirnstrukturen in MRT Datensätzen zu lokalisieren. KNN sind bewährte Konzepte um Muster zu klassifizieren. Das wesentliche Merkmal von KNN ist, dass sie vorgegebene Muster erlernen können und diese wiedererkennen. Das Erlernen erfolgt anhand klassifizierter Trainingsmuster. Der Lernprozess führt zu einer Generalisierungsfähigkeit, wodurch das KNN in der Lage ist, nicht klassifizierte Muster sinnvoll entsprechenden Klassen zuzuordnen.

Diese Art der Mustererkennung wird hier verwendet, um eine bestimmte, durch den Benutzer festgelegte Gehirnstruktur in MRT-Datensätzen zu lokalisieren. Dabei geht man schrittweise vor: Ein Experte definiert in einer Anzahl an MRT-Datensätzen die Position der gewünschten Struktur. Dadurch erhält man eine durch den Experten klassifizierte Trainingsmenge. KNN erlernen nun das Muster der Gehirnstruktur anhand von Voxeldaten. Die Voxeldaten werden aus einem Ausschnitt, der die Struktur enthält, sowie deren näheren Umgebung gewonnen, z.B. in Form eines Quaders. Das Lernen der KNN läuft automatisch ab, ohne weitere Interaktion mit dem Benutzer. Nach dem erfolgreichen Training ist es nun möglich, die entsprechende Struktur in einem neuen Datensatz zu lokalisieren. Hierbei definieren die KNN für verschiedene Positionen im Gehirn die Ähnlichkeit der Voxeldaten mit der trainierten Struktur. Die Position, für deren Voxeldaten das KNN die maximale Ähnlichkeit ausgibt, wird als geeignete Stelle interpretiert. Für die Lokalisierung wurden dazu zwei verschiedene Varianten entwickelt.

1. Die erste Variante ist eine Suche mit zwei KNN, 2-KNN-Suche genannt. Für diese Art der Suche werden beide KNN mit unterschiedlich großen Ausschnitten trainiert. Der größere Ausschnitt definiert die grobe Position der Gehirnstruktur, der kleinere Ausschnitt beschreibt die genaue Position. Beide Ausschnitte werden von einem Experten platziert. Sind beide KNN trainiert, findet zunächst die Suche mit dem größeren Ausschnitt im gesamten Datensatz statt. Ist diese Suche beendet, findet innerhalb dieses größeren Ausschnittes eine Suche mit dem kleineren Ausschnitt statt. Das Suchergebnis ist dann die genaue Position der Struktur. Die Suche des KNN mit dem größeren Ausschnitt benötigt sehr viel Rechenzeit, da sehr viele Positionen überprüft werden müssen. Um die Anzahl der Positionen, die überprüft werden zu reduzieren wurde ein genetischer Algorithmus verwendet. Der genetische Algorithmus vermindert die Rechenzeit deutlich, da er weniger geeignete Positionen nicht berücksichtigt.

2. In der zweiten Variante wird die Suche mit einem KNN durchgeführt, 1-KNN-Suche genannt. Für diesen Fall werden zusätzliche Ortsinformationen der Struktur verwendet.

Dazu werden die Datensätze zunächst so rotiert und verschoben, so dass messungsbedingte Rotations- und Translationsvarianzen kompensiert werden. Dies findet automatisch statt. Nun definiert der Experte in diesen normierten Datensätzen jeweils einen Ausschnitt, der die relevante Gehirnstruktur einschließt. Das liefert die Trainingsdaten für das 1-KNN. Aus den Positionen der Ausschnitte in den Trainingsdaten wird nun eine Wahrscheinlichkeitsverteilung berechnet, die die räumliche Verteilung der Gehirnstruktur in den normierten Datensätzen beschreibt. Das KNN lernt nun die Ausschnitte in den Trainingsdaten. Nach erfolgreichem Training des KNN findet die automatische Suche statt. Zunächst wird der zu durchsuchende Datensatz wie beim Training normiert. Die Wahrscheinlichkeitsverteilung beschränkt nun den Suchraum für das KNN. Das bedeutet, es werden nur solche Positionen überprüft, für die die Wahrscheinlichkeit des Auftretens der entsprechenden Gehirnstruktur hinreichend groß ist. Ergebnis der Suche mit dem KNN ist nun die genaue Position der Gehirnstruktur. Die Suche des KNN im eingeschränkten Suchraum benötigt wenig Rechenzeit, deshalb ist hier keine weitere Optimierung notwendig.