HyperNEAT

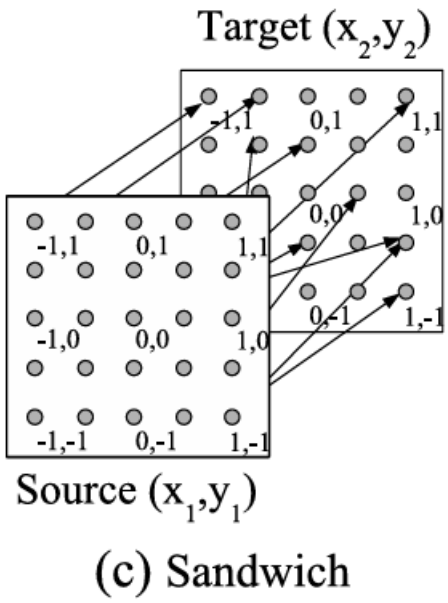
HyperNEAT (Hypercube-based NeuroEvolution of Augmenting Topologies) ist eine Erweiterung des NEAT-Algorithmus, die speziell dafür entwickelt wurde, größere und komplexere neuronale Netze effizient zu erzeugen. Während NEAT sich auf die Evolution der Netzstruktur konzentriert, geht HyperNEAT einen Schritt weiter und ermöglicht es, Muster in der Struktur des Netzes auszunutzen, um effizienter zu lernen.

Wie funktioniert HyperNEAT?

1. **Grundidee**: HyperNEAT basiert auf der Idee, dass neuronale Netze oft wiederkehrende Muster oder Symmetrien in ihren Verbindungen haben. Ein einfaches Beispiel wäre ein neuronales Netz, das Bilddaten verarbeitet, wo benachbarte Pixel oft ähnliche Merkmale aufweisen. HyperNEAT nutzt diese Symmetrien, um Netzwerke zu entwickeln, die solche Muster von vornherein berücksichtigen.
2. **CPPN (Compositional Pattern Producing Network)**: Im Zentrum von HyperNEAT steht das CPPN, ein spezielles neuronales Netz, das als „Generator" für die Gewichte des zu entwickelnden neuronalen Netzes dient. Anstatt jedes Gewicht einzeln zu optimieren, wie es NEAT macht, berechnet das CPPN die Gewichte basierend auf der relativen Position der Neuronen im Netz. Dies ermöglicht die Erzeugung von Netzwerken mit komplexen, aber geordneten Verbindungsstrukturen.

3. **Substrat**: Das Substrat in HyperNEAT bezeichnet den physischen Raum oder die geometrische Anordnung, in dem das neuronale Netz eingebettet ist. Es definiert die Positionen der Neuronen im Netzwerk, auf denen das CPPN operiert. Das Substrat kann als eine Art Raster oder Gitter betrachtet werden, auf dem die Neuronen angeordnet sind. Diese Anordnung ist entscheidend, weil das CPPN die Gewichte zwischen den Neuronen basierend auf deren Positionen im Substrat berechnet. So können räumliche Muster und Symmetrien in den Verbindungen natürlich entstehen, was besonders nützlich für Aufgaben wie Bildverarbeitung oder Robotik ist, bei denen die räumliche Beziehung zwischen den Eingaben eine zentrale Rolle spielt.

   In der Implementierung dieser Anwendung wurde sich für die einfache Sandwich-Architektur entschieden. Das Sandwich-Substrat besteht aus mehreren Schichten von Neuronen, die wie die Schichten eines Sandwichs übereinander angeordnet sind. Jede Schicht kann eine unterschiedliche Funktion im neuronalen Netz haben, beispielsweise als Eingabe-, Verarbeitungs- oder Ausgabeschicht. Diese Anordnung ermöglicht es dem Netzwerk, Informationen in mehreren Verarbeitungsschritten zu verarbeiten, ähnlich wie in einem traditionellen mehrschichtigen neuronalen Netzwerk.

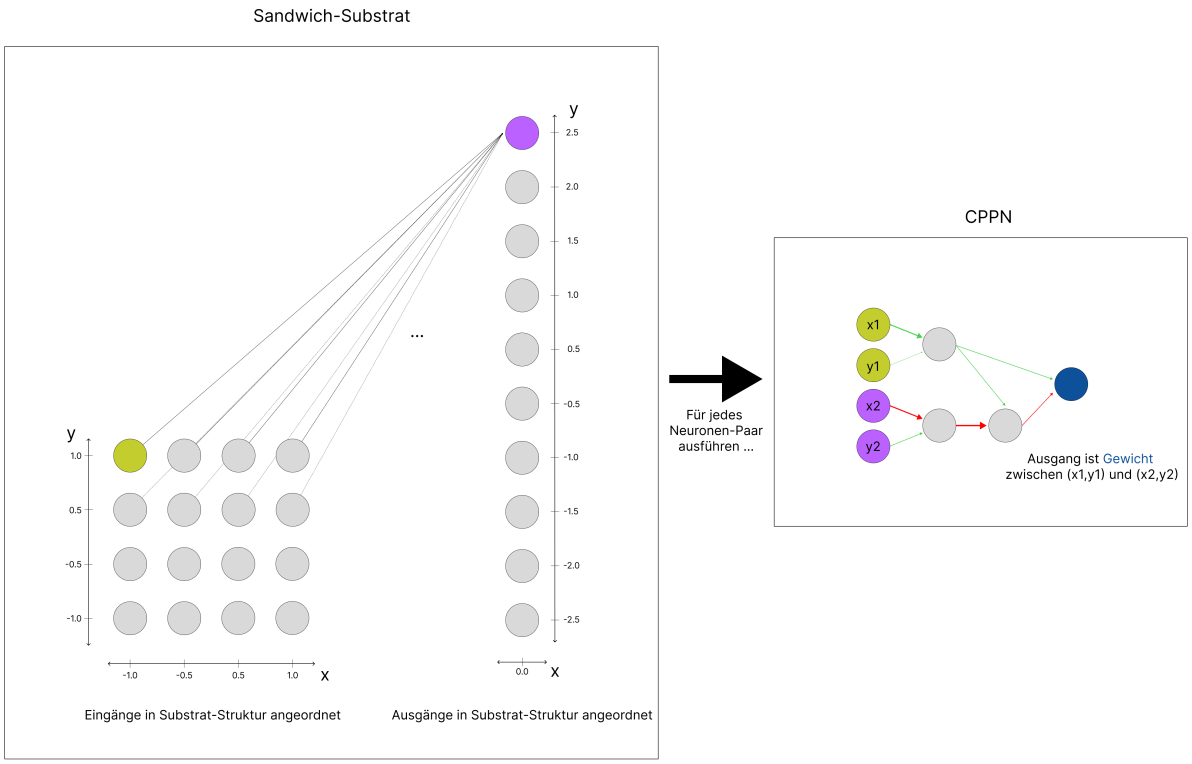


(c) Sandwich

Abbildung einer einfachen Sandwich-Struktur.

Quelle: A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks (2009). Kenneth O. Stanley

4. **Entwicklung des Netzwerks**: HyperNEAT verwendet das CPPN, um die Gewichte für die Verbindungen zwischen den Neuronen zu bestimmen. Das CPPN kann dabei Muster und Symmetrien in den Gewichten erzeugen, die in vielen realen Anwendungen nützlich sind. Zum Beispiel könnte ein CPPN ein symmetrisches Muster erzeugen, das ein neuronales Netz effizienter macht, wenn es mit symmetrischen Daten arbeitet, wie etwa bei der Bilderkennung.



Beispiel für Substart für ein 4×4 Pixel Bild. Alle Neuronen der Eingangsschicht werden mit allen Neuronen der Ausgangsschicht verbunden. Die Koordinaten werden alle nacheinander dem CPPN übergeben, welches die Verbindungsgewichte berechnet.

5. **Rolle von NEAT**: NEAT kommt in HyperNEAT ins Spiel, **indem es das CPPN optimiert**. Es hilft dabei, das beste CPPN zu finden, das die passenden Gewichte für das zu entwickelnde neuronale Netz erzeugt. Durch Evolution verbessert NEAT die Struktur und die Gewichtungen des CPPN, sodass das resultierende neuronale Netz immer besser an die zu lösende Aufgabe angepasst wird.