

AUTOMATISIERTE KONSTRUKTION VON KÜNSTLICHEN NEURONALEN NETZEN MITHILFE VON BESTÄRKENDEM LERNEN

Masterthesis von Kevin Rojczyk

Hochschule Darmstadt, Studiengang Data Science - 26.09.2019

Kurzfassung

Künstliche neuronale Netze (KNN) besitzen viele Parameter und Freiheitsgrade in der Architektur, was sie zu einer mächtigen Methode des maschinellen Lernens, insbesondere bei der Klassifizierung, macht. Die Entwicklung eines passenden KNNs ist für einen unbekanntem Datensatz, aufgrund der vielen Konstruktionsmöglichkeiten und benötigten Rechenkapazitäten, zeitaufwendig und schwierig. Um dieses Problem zu entschärfen, wurde im Rahmen dieser Masterarbeit mithilfe der Methode bestärkendes Lernen die Software *Network Architecture Construction Environment* (NACEnv) entworfen. Diese soll eine Strategie erlernen, um effizient ein passendes KNN für einen unbekanntem Datensatz zu konstruieren und zu trainieren. Trotz der Beschränkung der Aufgabenstellung auf *Convolutional Neural Networks* ist es nicht gelungen eine allgemeingültige Strategie zu erlernen. Allerdings gelang es mit NACEnv eine sehr simple Strategie in einem vereinfachten Szenario zu erlernen. Dabei konnte mit NACEnv ein passendes KNN für einen Datensatz konstruiert werden, der zuvor ebenfalls beim Training von NACEnv verwendet wurde.

Ziel

In dieser Masterarbeit soll eine Software auf Basis von maschinellem Lernen entwickelt werden, die automatisch eine passende KNN-Architektur für einen beliebigen Datensatz findet. Dazu soll das ML-Verfahren *Reinforcement Learning* (RL) eingesetzt werden, mit dem ein Agent trainiert wird, der eine KNN-Architektur konstruiert und trainiert. Dafür steht dem Agenten kein Vorwissen zur Verfügung. Der Agent muss durch Ausprobieren Strategien und Regeln erlernen, wie eine KNN für einen bestimmten Datensatz aufgebaut sein muss.

Der Agent soll im Resultat möglichst schnell auf iterativerweise, ein passendes KNN für einen gegebenen Datensatz bauen und trainieren.

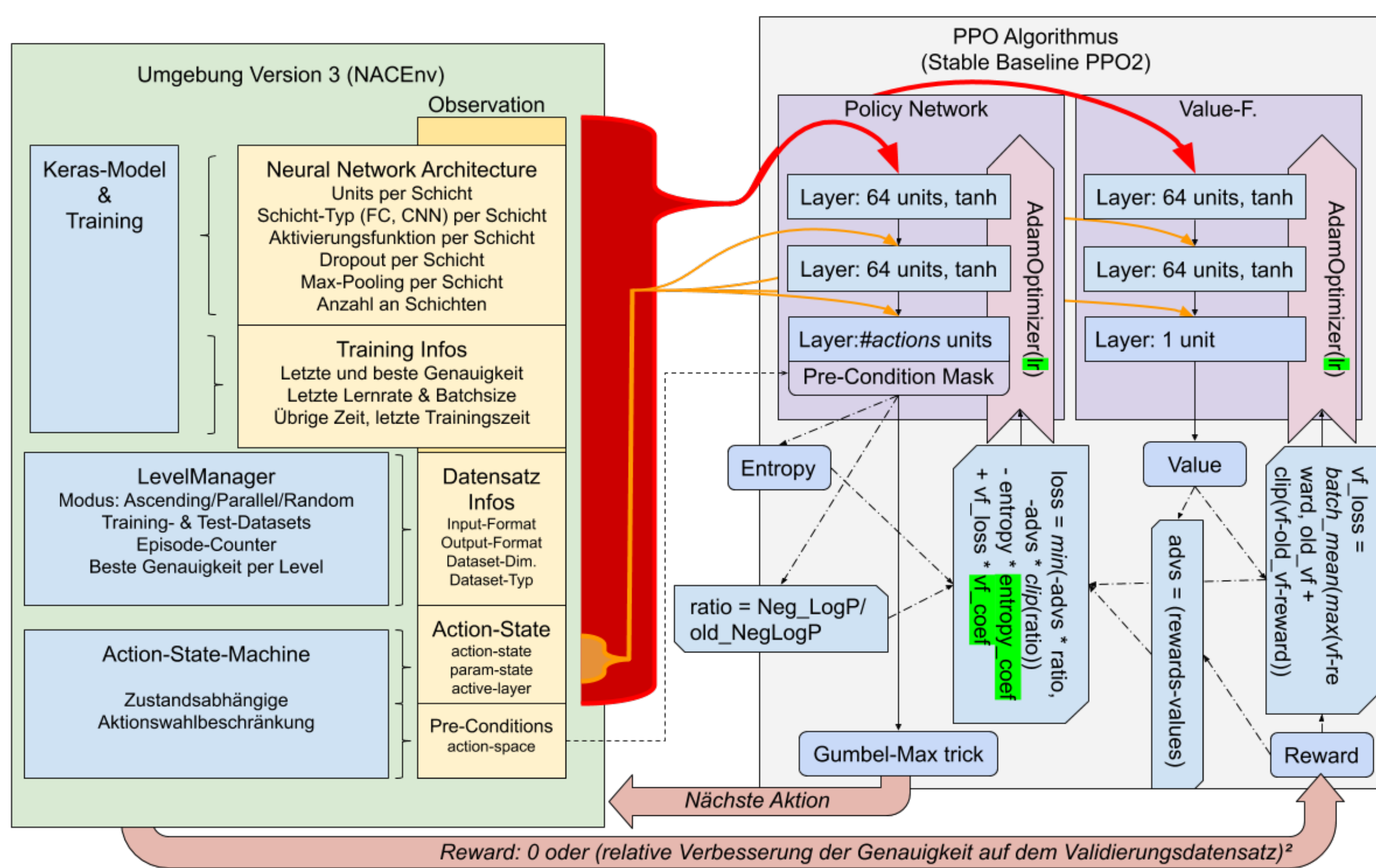
Weil das Trainieren von KNNs sehr ressourcenaufwändig und das Themenspektrum sehr breit ist, wird der Agent im Rahmen dieser Masterarbeit auf einfacheren Datensätze für die Bildklassifikation fokussiert.

Eigener Ansatz: NACEnv

Verwendet wird für den selbst entwickelten Ansatz der *On-Policy-RL-Algorithmus Proximal Policy Optimization*. Die Arbeitshypothese lautet: das Konstruieren einer guten KNN ist ähnlich wie das Spielen eines Aufbau- oder Strategie-Computerspiels. Ein Einsatzfeld in dem sich PPO schon bewährt hat. Die richtigen Schlussfolgerungen anhand der Observation von der Umgebung zu ziehen, ist dabei die Voraussetzung für ein guten RL-Agenten, weil sich der Zustand einer Umgebung ständig verändert und die Observationen essenzielle Informationen dazu enthalten. In dieser Masterarbeit sind die Observationen der aktuellen Zustand vom KNN sowie Meta-Daten des Trainings-Datensatz. Dieser kann größtenteils nur indirekt untersucht werden, in dem ein KNN anhand des Datensatzes trainiert wird und das Verhalten des KNN analysiert wird. Die Datensätze können, als weitere Analogie, als die *Levels* eines Computerspiels gesehen werden.

Die Umgebung ist somit die Software die alle Funktionalitäten dem Agenten zur Verfügung stellt, um ein KNN zu konstruieren und zu trainieren. Die Observation wiederum repräsentiert ein Teil des inneren Zustands der Umgebung. Ein RL-Agent interagiert mit der Umgebung anhand der Aktionen und bekommt die Observation und ein Reward als Feedback. Weil die Umgebung und der RL-Agent voneinander unabhängig sind, können statt PPO auch andere RL-Algorithmen verwendet werden, das wurde aber aus zeitlichen Gründen nicht ausprobiert.

Die vom RL-Agenten erlernte Policy soll eine Strategie oder ein Regelwerk repräsentieren, in dem Schrittweise für einen unbekanntem Datensatz ein KNN konstruiert wird. Es ahmt auf diese Weise einen Menschen nach, der ebenfalls basierend auf Regeln und bekannten *Best-Practices* iterativ ein KNN konstruiert und trainiert.



Action-State-Machine

Die *Action-State-Machine* (ASM) definiert den Aktionsraum vom Agenten. Der Agent kann immer nur einen numerischen Wert als Aktion wählen. Da zum Beispiel beim Hinzufügen einer neuen Schicht mehrere Parameter gesetzt werden müssen, müssen auch mehrere Aktionen hintereinander ausgeführt werden. Die Bedeutung der gewählten numerischen Aktionen ist abhängig vom Zustand der ASM. Der Vorteil dieses Konzepts ist, dass eine dynamische Beschränkung der wählbaren Aktionen möglich ist und somit nur gültige Aktionen gewählt werden können. Jeder Kreis im unterem Bild repräsentiert ein Zustand der ASM und die Pfeile die Übergänge mit ihren dynamischen Aktionswahlbeschränkungen.

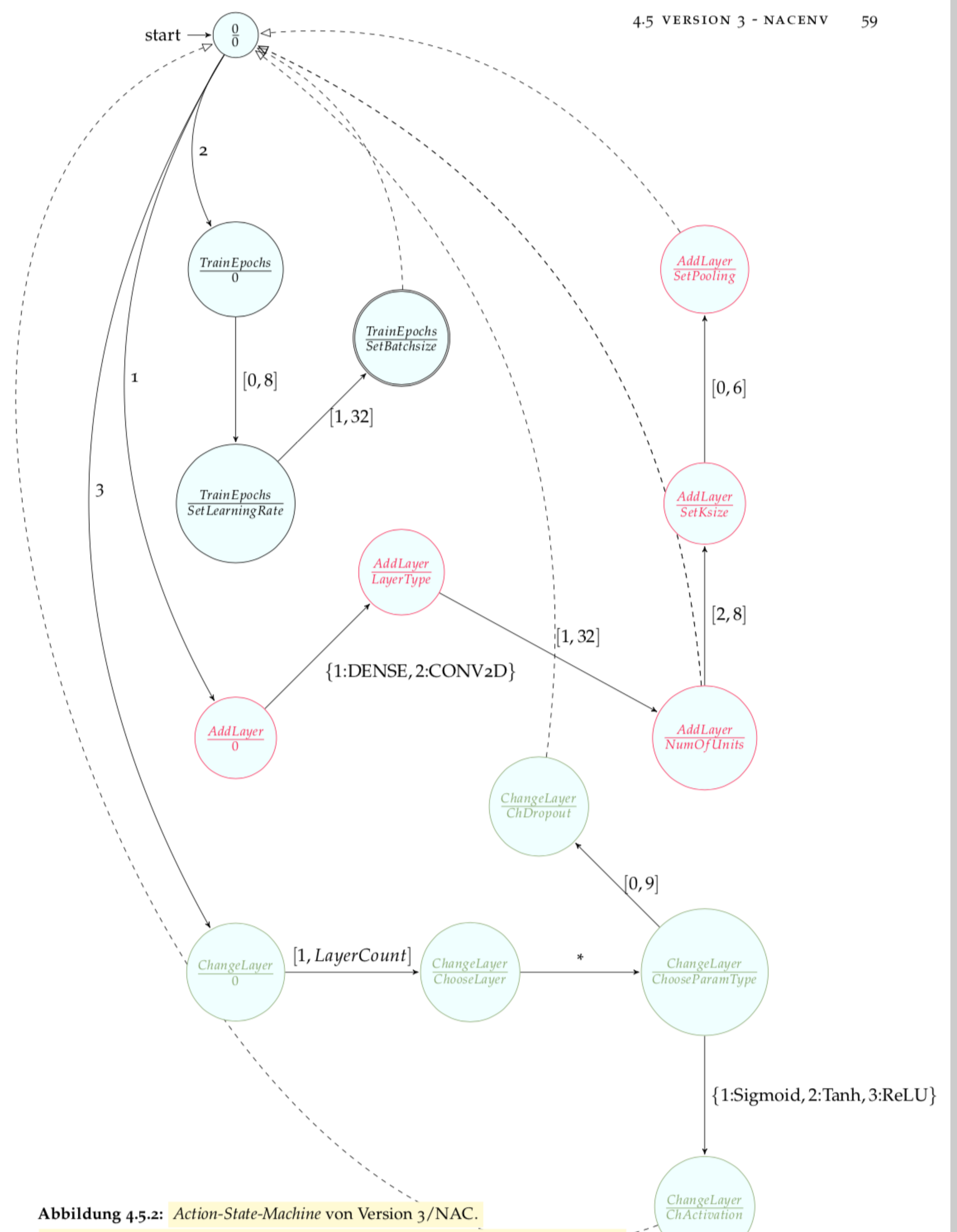


Abbildung 4.5.2: Action-State-Machine von Version 3/NAC. Die Anordnung der Zustände ist größtenteils identisch wie bei der Version 2. Siehe dazu Seite 43 zum Vergleich.

Resultate

Während der Bearbeitung der Masterarbeit ist aufgefallen, dass die Aufgabenstellung noch schwerer ist als ursprünglich angenommen. Das Ziel, ein Agenten zu trainieren, der ein KNN für einen unbekanntem Datensatz konstruiert und trainiert, konnte nicht erreicht werden. Es wurde zwar nicht erwartet, dass am Ende der Masterarbeit ein Agent zur Verfügung steht, der ein perfektes KNN konstruiert, aber dass der Agent zumindest versuchen würde für jeden Datensatz eine eigene passende KNN-Architektur zu konstruieren.

Mit der schlussendlich entwickelten Umgebung NACEnv wurde zumindest gezeigt, dass der Agent für einen bekannten Datensatz ein KNN zu konstruieren (Evaluation 6 in Unterkapitel 4.5.2) erlernen konnte. Dafür musste die Umgebung insofern stark vereinfacht werden, dass der Agent nur auf einem einzigen kleinen Datensatz experimentieren konnte. In Evaluation 6 konnte ebenfalls beobachtet werden, wie der Agent selbständig eine neue Strategie erlernen konnte. Der Agent fixiert beim Trainieren des KNNs die Lernrate und erhöht nach einigen Epochen die Batchsize. Diese Strategie wurde erst kürzlich in der Publikation von Smith et al. "Don't Decay the Learning Rate, Increase the Batch Size" (2018) als eine effiziente Variante zum Trainieren eines KNNs vorgestellt. Dass der Agent diese ungewöhnliche Strategie erlernen konnte, unterstreicht das Potenzial vom Ansatz, welcher in dieser Masterarbeit vorgestellt wurde. Es kann automatisiert nach neuen, noch unbekanntem Strategien und Regeln zum Konstruieren und zum Trainieren von KNNs gesucht werden, die manuell nur sehr aufwendig zu entdecken sind.