

Einleitung

Die Architektur eines Deep-Q-Networks (DQN) hat großen Einfluss auf den Lernerfolg eines Reinforcement-Learning-Agents ¹. Üblich ist ein **mehrschichtiges Perzeptron (MLP) mit zwei Hidden Layers** (z. B. je 128 Neuronen) und **ReLU-Aktivierungen** in den Hidden Layers sowie einem **linearen Output-Layer** zur Ausgabe der Q-Werte. Im Folgenden werden klassische und aktuelle Quellen angeführt, die diese Architekturwahl empirisch oder theoretisch begründen und als Argumentationsbasis dienen können.

Klassische Referenzen: DQN-Grundlagen

Bereits **Mnih et al. (2015)** führten DQNs ein und nutzten ein tiefes neuronales Netz, um Q-Funktionen näherungsweise zu lernen ². Ihre Atari-DQN-Architektur besaß mehrere Faltungs- und Pooling-Schichten zur Merkmalsextraktion aus Bilddaten, gefolgt von vollverknüpften Schichten mit **ReLU-Aktivierungen** und einer linearen Ausgabeschicht ³. Insbesondere bestand die finale Hidden-Layer aus 512 **Rectifier (ReLU)**-Neuronen, und der Output-Layer war eine vollständig verknüpfte lineare Schicht mit einer Ausgabe pro Aktion ⁴. Diese Designwahl – **ReLU in den Hidden-Layers und kein Aktivierungsfunktion im Output** – ist bis heute Standard: ReLUs verhindern Sättigungseffekte und erleichtern das Training tiefer Netze, während eine lineare Ausgabeschicht benötigt wird, da Q-Werte prinzipiell unbeschränkt sein können ⁴. Mnih et al. zeigten, dass ein solches deepes Netz die limitierte Ausdruckskraft früherer linearer oder flacher Approximationen deutlich übertrifft und erstmals menschnahes Leistungsniveau auf Atari-Spielen erreichte ².

Auch nach 2015 behielten Verbesserungen des DQN die Kern-Architektur bei. So verwendeten etwa **Double DQN** (van Hasselt *et al.*, 2016) und **Prioritized Replay** (Schaul *et al.*, 2016) dieselbe Netzwerkstruktur wie das Original-DQN, um fair zu vergleichen ⁵. Die **Dueling DQN**-Variante von Wang *et al.* (2016) modifizierte zwar den Netzaufbau, bleibt aber im Grundsatz bei zwei vollverknüpften Hidden Layers: Nach einer ersten gemeinsamen Schicht mit 50 Neuronen verzweigt das Netzwerk in zwei Stränge (Value- und Advantage-Stream), die jeweils als zweilagiges MLP mit 25 Hidden Units umgesetzt sind ⁶. Dies verdeutlicht, dass **zwei Hidden Layers** in unterschiedlichen DQN-Varianten als ausreichende Tiefe angesehen wurden, um komplexe Funktionen zu repräsentieren, während zusätzliche Tiefe selten nötig war. Schon frühe Arbeiten wie **Riedmillers NFQ (2005)** nutzten übrigens ein kleines MLP (zwei Hidden-Layers mit je 5 Neuronen) zur Q-Funktionsapproximation ⁷ – ein Hinweis darauf, dass **mehrlagige Perzeptrons mit zwei verborgenen Schichten** seit langem als leistungsfähiger Funktionsapproximator im Q-Learning gelten.

Begründung der Architekturgröße (2 Hidden Layers, ~128 Neuronen)

Theoretisch ist bekannt, dass bereits ein einzelner Hidden-Layer mit genügend Neuronen jede stetige Funktion approximieren kann (Universal Approximation Theorem). In der Praxis erfordert ein einziger sehr großer Layer aber viel Trainingsdaten und kann schwer zu optimieren sein. **Zwei Hidden Layers** bieten einen Kompromiss: Sie ermöglichen hierarchische Merkmalsbildung, sodass mit insgesamt weniger Neuronen die gleiche Komplexität dargestellt werden kann, und sind empirisch oft leichter zu trainieren als

sehr tiefe Netze. Viele Reinforcement-Learning-Richtlinien empfehlen daher standardmäßig 2 Hidden Layers mit mittlerer Neuronenanzahl. Beispielsweise nutzt die populäre RL-Bibliothek Stable Baselines3 per Voreinstellung **zwei Hidden Layers à 128 Neuronen** für Policy- und Q-Netzwerke ⁸.

Empirische Studien untermauern diese Daumenregel. Eine aktuelle Untersuchung (Zhou *et al.*, Sci. Reports 2025) verglich systematisch verschiedene Hidden-Layer-Größen in einem RL-Szenario. Dabei zeigte sich, dass **mittelgroße Schichten (64–128 Neuronen)** den besten Kompromiss aus Ausdrucksstärke und Effizienz bieten ⁹. Zu kleine Netze (z. B. 32 Neuronen) hatten nicht genug Kapazität (Underfitting), während deutlich größere Netze (256+ Neuronen) zwar anfangs schnell lernten, aber im Verlauf **langsamer konvergierten** und unter **Redundanz/Overfitting** litten ¹⁰. Konkret erreichte ein Netzwerk mit **2×128 Neuronen** den höchsten langfristigen Ertrag, wohingegen ein 2×384-Netz zwar initial gut performte, dann aber stagnierte ¹⁰. Diese Ergebnisse *„stützen empirisch die Wahl angemessen skalierten Netzwerkarchitekturen“*, die sowohl Rechenaufwand als auch Lernerfolg optimieren ¹⁰. Die Autoren begründen, dass 128er-Schichten hinreichend komplexe Policies abbilden können, ohne die Konvergenz durch unnötig viele Parameter zu verlangsamen ¹⁰. Zudem war im gleichen Bericht die Kritiker-Netzwerkkomponente absichtlich größer (128 Neuronen statt 64 im Akteur), um die höherdimensionale Value-Funktion ohne Underfitting darstellen zu können ¹¹ – was zeigt, dass **128 Neuronen pro Layer** als sinnvolle Obergrenze für erhöhte Ausdrucksfähigkeit gelten, bevor Overkill eintritt.

Auch andere Arbeiten stützen die Wahl von **zwei Hidden Layers** im **128er-Bereich**. Ein **Nature Communications Physics (2021)**-Paper über quantenmechanische Steuerung probierte z. B. schrittweise größere Netzwerke aus und fand heraus, dass bereits *„ein relativ kleines Netzwerk aus zwei Hidden Layers mit je 128 Neuronen ausreicht, um die Aufgabe erfolgreich zu bewältigen“* ¹². Größere Netze brachten dort keinen signifikanten Vorteil mehr. Ähnlich nutzt **Tian *et al.* (ECCV 2020)** in ihrem Off-Policy-RL-Ansatz ein **vollvernetztes Q-Netz mit zwei Hidden Layers à 128 Neuronen und ReLU** in allen Hidden Layers ¹³ – eine Architektur, die sich als stabil und effektiv erwies. Ebenso setzen Goyal *et al.* (ICLR 2019) in ihren Backtracking-Modellen auf zwei Hidden Layers mit 128 Neuronen ¹⁴. Die fortgesetzte Verwendung dieser Größenordnung in aktuellen Forschungsarbeiten – oft ohne umfangreiche Hyperparameter-Suche bei der Layergröße – deutet darauf hin, dass sich **128×128 als bewährter Standard** etabliert hat.

Fazit

Zusammenfassend sprechen sowohl **klassische Erfahrungen** mit DQNs als auch **neuere Studien** dafür, dass ein **MLP mit zwei Hidden Layers** im Bereich von **hundert bis wenigen hundert Neuronen** pro Schicht eine vernünftige Architektur für Deep-Q-Networks darstellt. Diese Struktur bietet **hinreichende Repräsentationskraft** für komplexe Q-Funktionen, bleibt aber **trainierbar und datenökonomisch**. Die Verwendung von **ReLU-Aktivierungen** in den verborgenen Schichten ist dabei de facto Standard und wurde schon im Original-DQN eingeführt ⁴, da ReLUs tiefe Netze erst praktikabel machen. Ein **linearer Output-Layer** ist unerlässlich, um nicht die Skala der Q-Werte einzuschränken ⁴. Zahlreiche Arbeiten – von Mnih *et al.* (2015) ⁴ über Wang *et al.* (2016) ⁶ bis zu aktuellen Publikationen (2019–2024) ¹² ⁹ – untermauern die **Effektivität und Suffizienz eines 2×128-ReLU-MLP** als DQN-Architektur. Insbesondere wird immer wieder betont, dass diese mittlere Netzwerkgröße **genug Kapazität** für die meisten Aufgaben bietet, ohne die **Trainingsstabilität** zu gefährden oder unnötigen Overhead zu erzeugen ⁹ ¹⁰. Solche fundierten Referenzen liefern eine nachvollziehbare Begründung, warum ein MLP mit zwei Hidden Layers (etwa 128×128, ReLU) und linearem Ausgang in wissenschaftlichen Projektberichten als gut begründete Architekturwahl für DQNs präsentiert werden kann.

Quellen: Mnih *et al.* (2015) ⁴ ; van Hasselt *et al.* (2016); Wang *et al.* (2016) ⁶ ; Schaul *et al.* (2016); Goyal *et al.* (2019) ¹⁴ ; Tian *et al.* (2020) ¹³ ; Maronese *et al.* (2021) ¹² ; Zhou *et al.* (2025) ⁹ ¹⁰ .

¹ ¹² Quantum compiling by deep reinforcement learning | Communications Physics

https://www.nature.com/articles/s42005-021-00684-3?error=cookies_not_supported&code=4960b244-97b0-4816-a69b-71de003c5bef

² Deep Q-Network - an overview | ScienceDirect Topics

<https://www.sciencedirect.com/topics/computer-science/deep-q-network>

³ ⁴ Human-level control through deep reinforcement learning - nature14236.pdf

<https://web.stanford.edu/class/psych209/Readings/MnihEtAlHassibis15NatureControlDeepRL.pdf>

⁵ Constrained Deep Q-Learning Gradually Approaching Ordinary Q ...

<https://pmc.ncbi.nlm.nih.gov/articles/PMC6914867/>

⁶ [PDF] Dueling Network Architectures for Deep Reinforcement Learning

<https://proceedings.mlr.press/v48/wangf16.pdf>

⁷ endtoend.ai

<https://www.endtoend.ai/assets/blog/paper-unraveled/nfq/original.pdf>

⁸ Policy Networks — Stable Baselines3 2.7.1a0 documentation

https://stable-baselines3.readthedocs.io/en/master/guide/custom_policy.html

⁹ ¹⁰ ¹¹ The analysis of deep reinforcement learning for dynamic graphical games under artificial intelligence | Scientific Reports

https://www.nature.com/articles/s41598-025-05192-w?error=cookies_not_supported&code=501204de-1da5-42dc-bb00-94833d513fea

¹³ ecva.net

https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123520171-sup.pdf

¹⁴ openreview.net

<https://openreview.net/pdf/3c2f3116d77fa2ce0af93b56a95feb16e9b9d9a3.pdf>