

# NOVI – Pong Neural Network Trainer (CH/DE)

Danke, dass du **NOVI** heruntergeladen hast 

Dies ist die **CH/DE-Version** eines Pong-Trainers mit **Neurealem Netzwerk (PyTorch)**.

NOVI kann: - ein neuronales Netz **neu trainieren** - ein bestehendes Netz **weitertrainieren** - ein trainiertes Netz **nur spielen lassen** - **1 oder 2 Spielfelder gleichzeitig anzeigen** - **Live-Graphen** (Trefferquote & Scores) - Modelle und Diagramme **automatisch speichern**

---

## Voraussetzungen

Du brauchst **Python 3.9 oder neuer**.

## Benötigte Libraries

Diese Pakete müssen installiert sein:

- pygame
- torch
- tkinter (meist schon bei Python dabei)
- matplotlib

## Installation (Terminal)

```
pip install pygame torch matplotlib
```

### Hinweis macOS:

Falls tkinter fehlt, installiere Python über [python.org](https://www.python.org), nicht über Homebrew.

---

## Starten von NOVI

Im Projektordner:

```
python Novi.py
```

(Der Dateiname kann bei dir leicht anders sein.)

Danach öffnet sich ein **Start-Menü (Tkinter)**.

---

## Start-Menü erklärt

### 1 Anzahl der Spielfelder

- **1 Spiel** → ein NN gegen einen Bot

- **2 Spiele** → zwei NNs parallel (Vergleich)

## 2 Modus wählen

### Neu lernen

- Startet ein **komplett neues neuronales Netz**
- Training läuft sofort

### Weiter lernen

- Lade eine bestehende .pth-Datei
- NN trainiert **weiter**

### Nur spielen (kein Training)

- NN spielt nur
- **Keine Gewichtsänderung**

---

## Steuerung

 Keine Tastatur nötig – alles läuft automatisch.

Zum Beenden: - Fenster schließen

---

## Spiellogik (kurz erklärt)

- Links: **Neuronales Netz**
- Rechts: **Bot** (absichtlich etwas schlechter)
- Das NN sieht:
  - Ballposition
  - Ballgeschwindigkeit
  - Eigene Paddle-Position

Das NN entscheidet: - **hoch / runter / stehen bleiben**

Beim Training: - Treffer = gut - Vorbeischlagen = schlecht

---

## Graphen (rechts im Fenster)

### Trefferquote

- Wie gut das NN den Ball trifft

### Scores

- NN-Punkte
- Bot-Punkte

## Vergleich (nur bei 2 Feldern)

- Direkter Vergleich von NN1 vs NN2
- 

## Speicherstruktur

Beim Start wird automatisch ein Ordner erstellt:

```
saves/
└── 2026-01-30_14-22-08/
    ├── f1_rate.png
    ├── f1_scores.png
    ├── nn1.pth
    ├── f2_rate.png    (nur bei 2 Feldern)
    ├── f2_scores.png
    ├── nn2.pth
    └── nn_compare.png
```

## Bedeutung

- \*.pth → trainiertes NN-Modell
  - \*\_rate.png → Trefferquote
  - \*\_scores.png → Punktestand
  - nn\_compare.png → Vergleich beider NNs
- 

## Farben anpassen

Die Farben pro Spielfeld kannst du **direkt im Code** ändern:

```
colors_f1 = {
    "nn": (0, 255, 100),
    "bot": (150, 150, 150),
    "x": (255, 255, 255)
}
```

RGB-Werte von 0–255.

---

## Häufige Probleme

✖ **ModuleNotFoundError: pygame**

→ pip install pygame

✖ **Tkinter fehlt (macOS)**

→ Python von **python.org** installieren

## Modell lädt nicht

→ Datei muss eine **.pth** sein

---

## Idee & Zweck

NOVI ist ein **Lern- und Experimentierprojekt** für:

- Neuronale Netze
- Reinforcement-ähnliches Lernen
- Game + KI Kombination
- Visualisierung von Training

Perfekt zum **Verstehen**, nicht für maximale Performance.

---

## Lizenz

Privates Lernprojekt.

Keine Garantie, keine Haftung.

---

Viel Spass beim Trainieren 