

Universität Hamburg  
Department Informatik  
Knowledge Technology, WTM

# Pong, gespielt von KNNs

Praktikum Paper

Dokumentation der erstellten Applikation

Daniel Speck und Florian Kock

Matr.Nr. 6321317, 6346646

2speck@informatik.uni-hamburg.de, 2kock@informatik.uni-hamburg.de

2. Februar 2015



## Abstrakt

Unsere Aufgabe des letzten Semesters war es, in dem Praktikum Neuronale Netze eine Aufgabe zu lösen. Wir haben uns für das in den 70ern populäre Spiel Pong entschieden. Dieses war dahingehend interessant, da es gleich mehrere Probleme zu lösen gab:

- die zeitverzögerte Bewertung
- das Erkennen der Flugrichtung durch Rekurrenz

Das Spielprinzip von Pong ist simpel und ähnelt dem des Tischtennis: Ein Punkt („Ball“) bewegt sich auf dem Bildschirm hin und her. Jeder der beiden Spieler steuert einen senkrechten Strich („Schläger“), den er mit einem Drehknopf (Paddle) nach oben und unten verschieben kann. Lässt man den „Ball“ am „Schläger“ vorbei, erhält der Gegner einen Punkt. (Wikipedia)

## Inhaltsverzeichnis

<b>1</b>	<b>Vorbedingungen und Anforderungen</b>	<b>2</b>
<b>2</b>	<b>Schnelleinstieg</b>	<b>2</b>
<b>3</b>	<b>Aufbau der Applikation</b>	<b>3</b>
3.1	Die Hauptanwendung - main.py . . . . .	3
3.2	Die Visualisierung - visu.py . . . . .	3
<b>4</b>	<b>Conclusion</b>	<b>5</b>
<b>5</b>	<b>remove this!</b>	<b>5</b>

# 1 Vorbedingungen und Anforderungen

Um die Applikation erfolgreich ausführen zu können sind folgende Anforderungen an die Laufzeitumgebung notwendig:

- Python3.4 

- NumPy 

- The Python Standard Library
  - multiprocessing
  - threading
  - sys
  - random
  - os.path
  - logging
  - json
  - socketserver
  - tkinter
  - socket
  - time
  - datetime
  - copy

## 2 Schnelleinstieg

Der Schnelleinstieg führt über folgende Befehle: (Wenn die Vorbedingungen gegeben sind!)

Starten des Hauptprogramms:

```
> ls
__init__.py      knnframe.py      telegramframe.py
__pycache__      main.py          visu.py
concol.py        recneunet.py
court.py         save
> python3.4 main.py
... starting application ...
```

Starten der Visualisierung zum Hauptprogramm:

```
> ls
__init__.py      knnframe.py      telegramframe.py
__pycache__      main.py          visu.py
concol.py        recneunet.py
court.py         save
> python3.4 visu.py
... starting visualisation ...
```

## 3 Aufbau der Applikation

Unsere Applikation ist in mehrere Dateien und Module aufgeteilt. In der Abbildung 1 auf Seite 4 ist die Grundstruktur dargestellt. Die einzelnen Module sind separiert in die Haupt-Anwendung (`main.py`) 3.1 und die Visualisierung (`visu.py`) 3.2.

### 3.1 Die Haupanwendung - `main.py`

Die Hauptapplikation besteht im Wesentlichen aus einer Hauptschleife, die die Spielzüge (im Quellcode 'Ticks' genannt) kontrolliert. Der relevante Teil befindet sich hierzu in der *main.py* in den Zeilen von ca. 360 bis 460.

Sie liefert in jedem Spielzug vom Spielfeld (siehe 3.1.1) die Ballpositionen an die Spieler (siehe 3.1.2), um anschließend deren Aktion wieder dem Spielfeld zuzuführen.

#### 3.1.1 Die Haupanwendung - `curt.py`

Das Spielfeld implementiert eine Simulation der Spielumgebung. Diese besteht aus zwei Schlägern und einem Ball in einem rechteckigen Rahmen. Bei jedem Aufruf von *tick()* berechnet das Spielfeld den nächsten Zustand des Balles, hierbei wird ebenfalls geprüft, ob er die Bande, den Rahmen, überschritten hat und korrigiert dies. Ein solcher Abprall ist der Physik nachempfunden ('Einfallswinkel = Ausfallswinkel').

Das Ziel des Spieles ist es, den Ball mit dem Schläger zurückzuspielen, gelangt der Ball jedoch über die Linie, bedeutet dies, das der Ball wieder in der Mitte neu initialisiert wird. Hierfür ist die Funktion *\_\_initvectors()* zuständig.

#### 3.1.2 Die Haupanwendung - `knnframe.py`

### 3.2 Die Visualisierung - `visu.py`

Die Visualisierung dient der einfachen Diagnose, des Zustands der Hauptapplikation welche keine eigene Visualisierung bereitstellt. Die Verbindung geschieht über TCP/IP über den Port 6769 und ist aktuell nur vom localhost erreichbar.

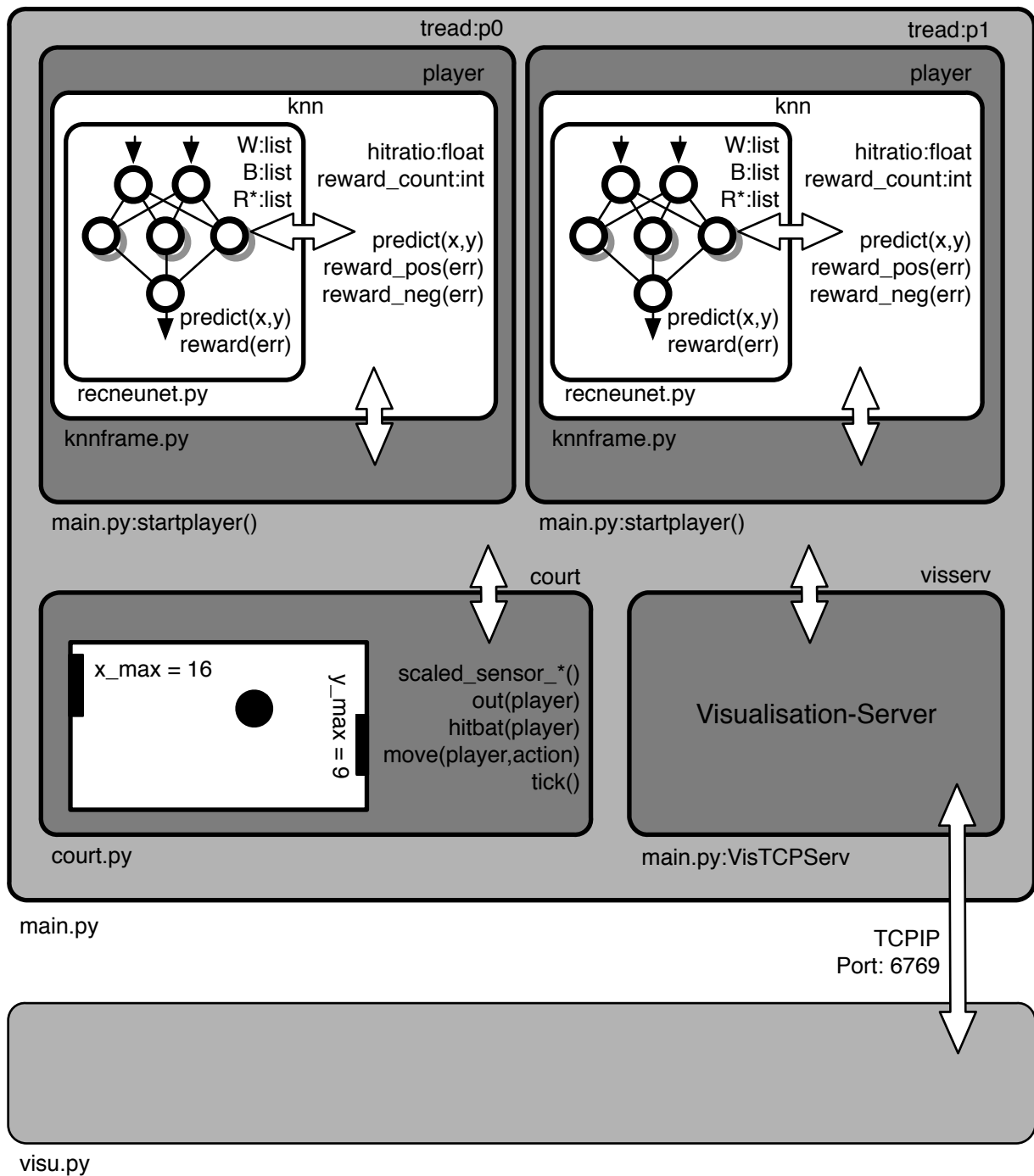


Abbildung 1: Schematischer Aufbau der Pong-Applikation

TODO: Screenshot und erklärung der Funktionen!

## 4 Conclusion

Your text here...

## 5 remove this!

```
nc = NetwCon()
connected = False
while not connected:
    print('Try_to_connect...')
    connected = nc.connect()
    if not connected:
        time.sleep(1)
print('connected, lets_go...')
```