

Temnothorax Nylanderi

Allgemeines:

- leben meist in Haselnüssen im Wald
- benutzen keine Pheromone
- deutlich kleiner als z.B. Waldameisen
- nur 100 bis 200 Ameisen pro Staat



Unterbringung für Untersuchung:

- Jedes Volk in eigener Behausung
- 3 Kammern pro Behausung
 - Wohnkammer (auch zur Eiablage) (abgedunkelt)
 - Trinkwasser - Kammer (Tropfen auf Metallplättchen)
 - Nahrungskammer (Honig in Plastikschiene)
- Boden mit Papier ausgelegt

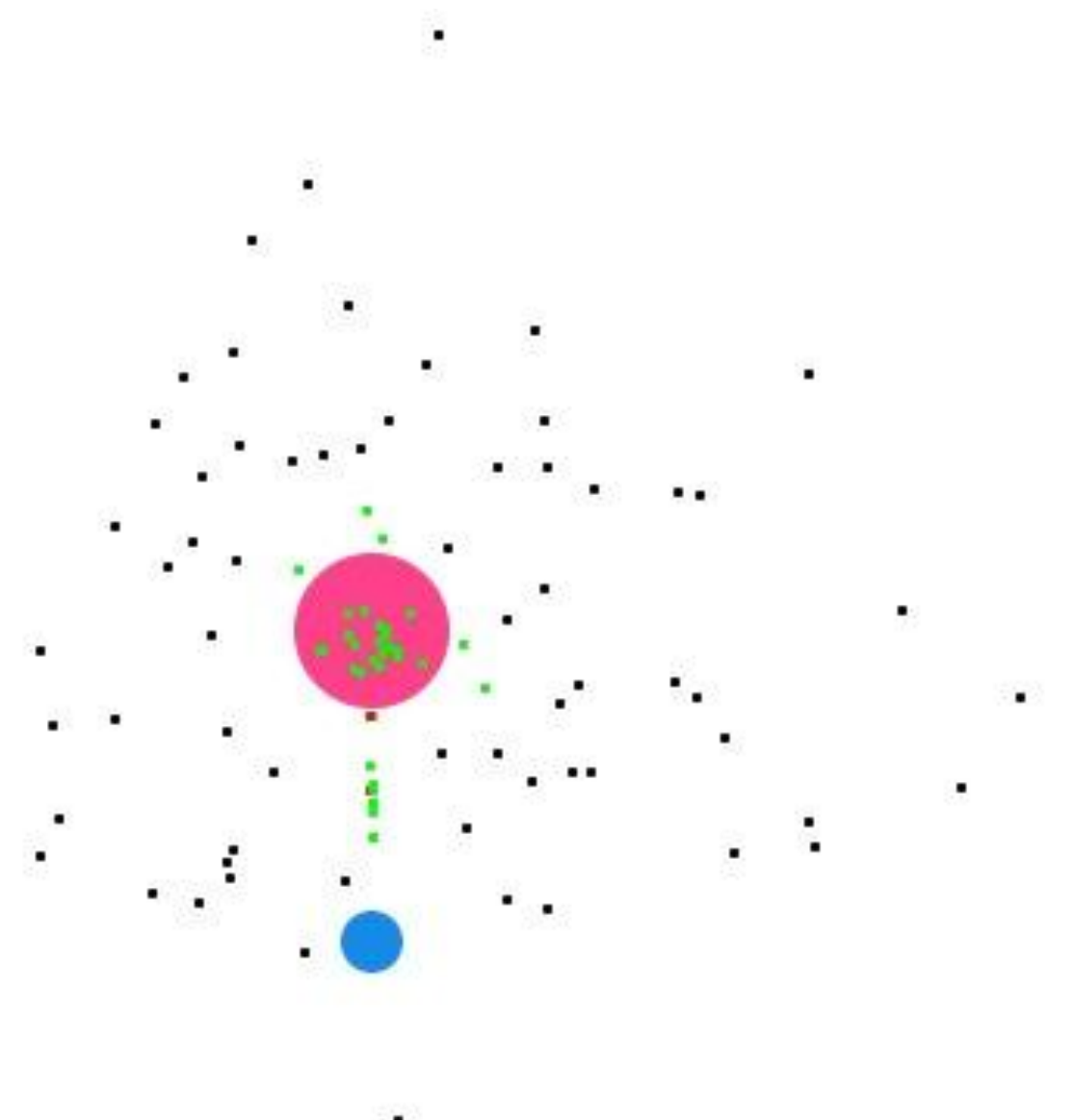


Das Ziel:

Wir wollen ein Neuronales Netz entwickeln, dass Ameisen und andere Gegenstände die sich in deren Behausung befinden erkennen kann und damit Bewegungsprofile erstellt. Ein Convolution Neuronales Network ist dafür am Besten geeignet.

Bisher geschafft:

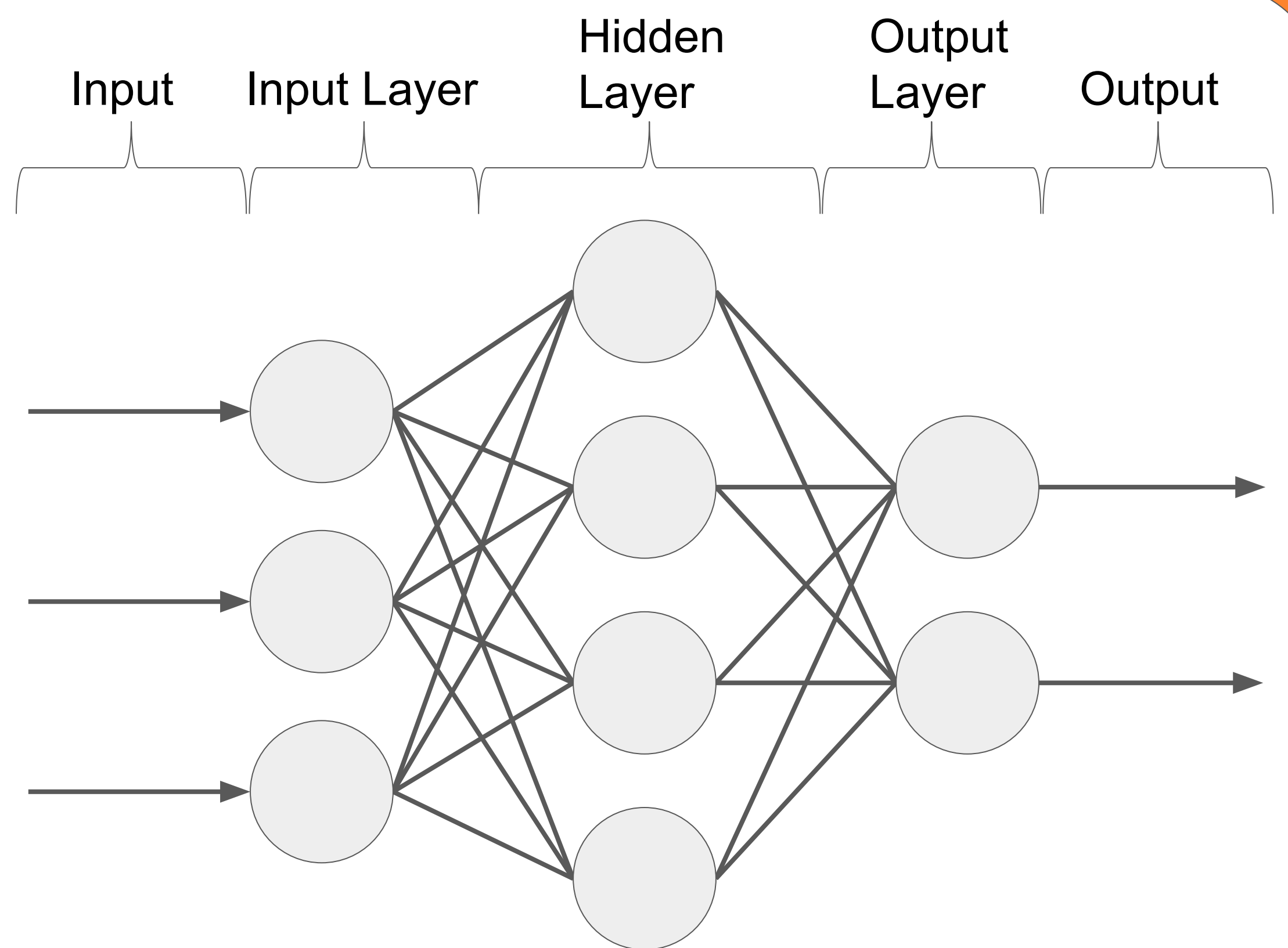
Wir befinden uns noch am Anfang des Projekts, da es sehr aufwendig ist ein NN mit eigenen Daten zu trainieren ohne wie bei vielen anderen Anwendungsfällen auf vorgefertigte Bildbibliotheken (z.B.: MNIST) zurückzugreifen zu können. Wir haben aber bereits ein Programm geschrieben, das die Ameisen anhand unseren Beobachtungen simuliert.



Neuronale Netze

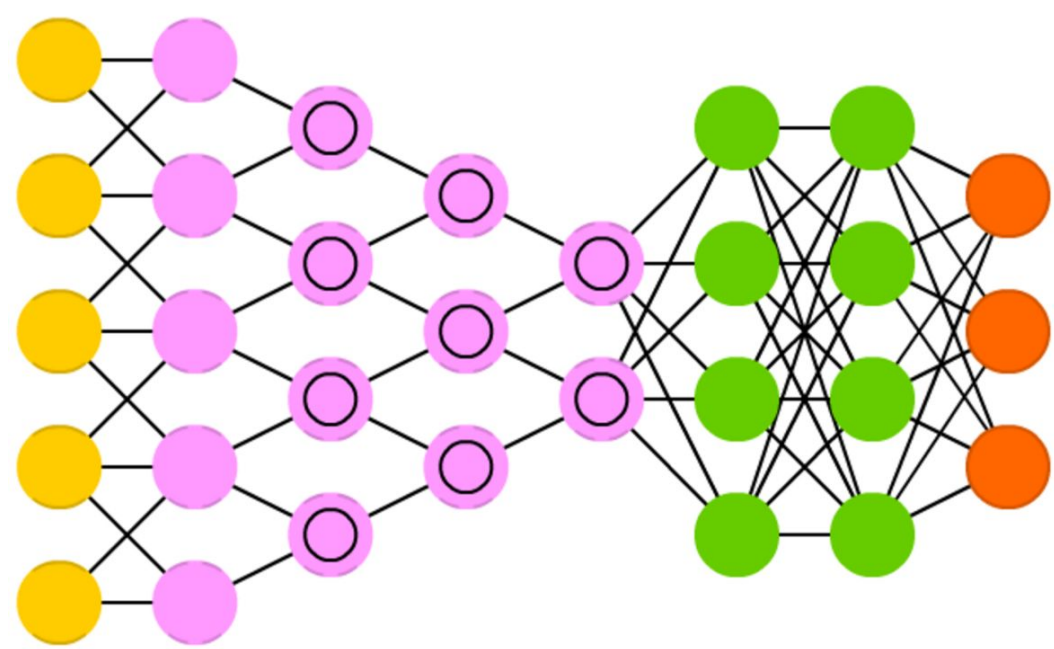
Generelle Struktur:

- mehrere Layer
- Input und Output sind Zahlen
- Verbindungen sind “Gewichte”, die Trainiert werden
- Input = nachbearbeitetes Bild einer oder mehrerer Ameisen
- Output = Positionen der Ameisen

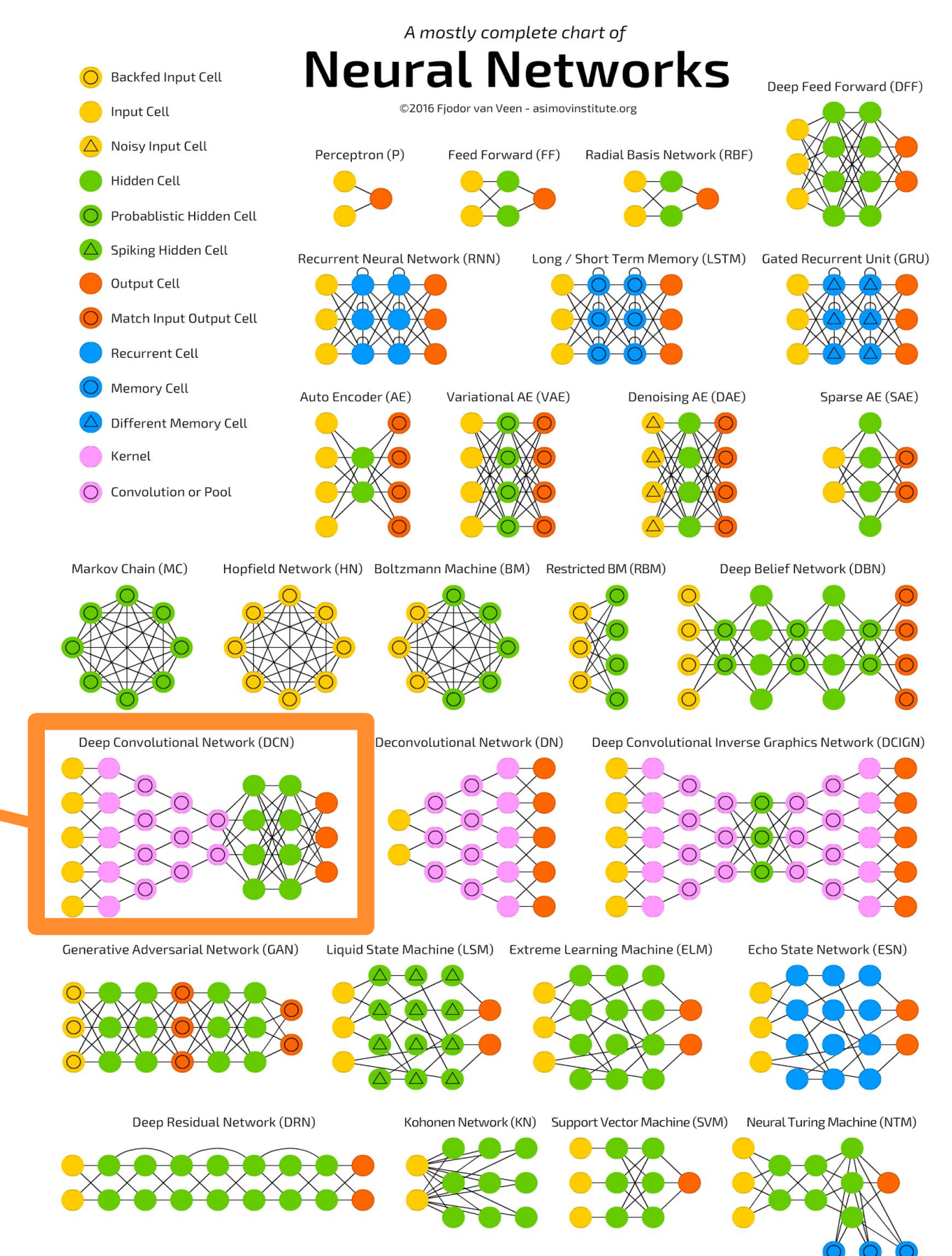


Typen von NNs:

Convolutional Neuronal Network



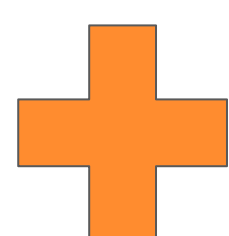
- viele Layer
- eignet sich gut für Objekterkennung
- Layer spezialisieren sich auf Linien, Farbverläufe oder Formen



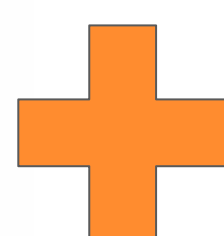
Umsetzung:



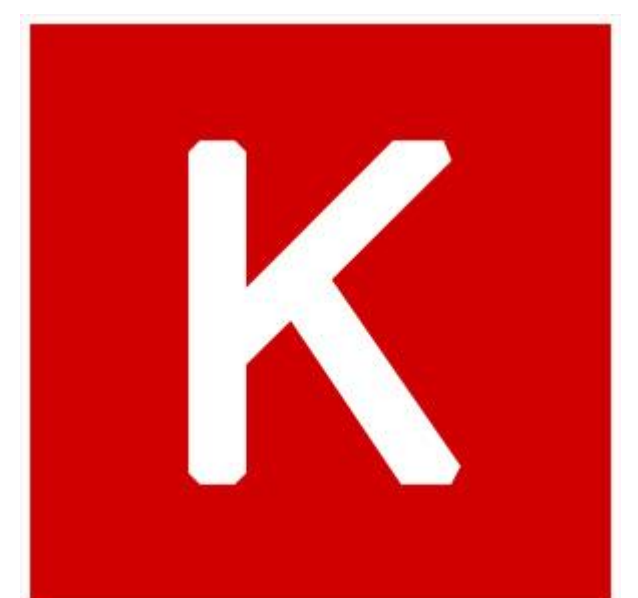
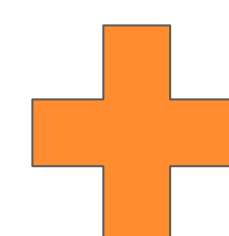
Python



Jupyter



Tensor Flow

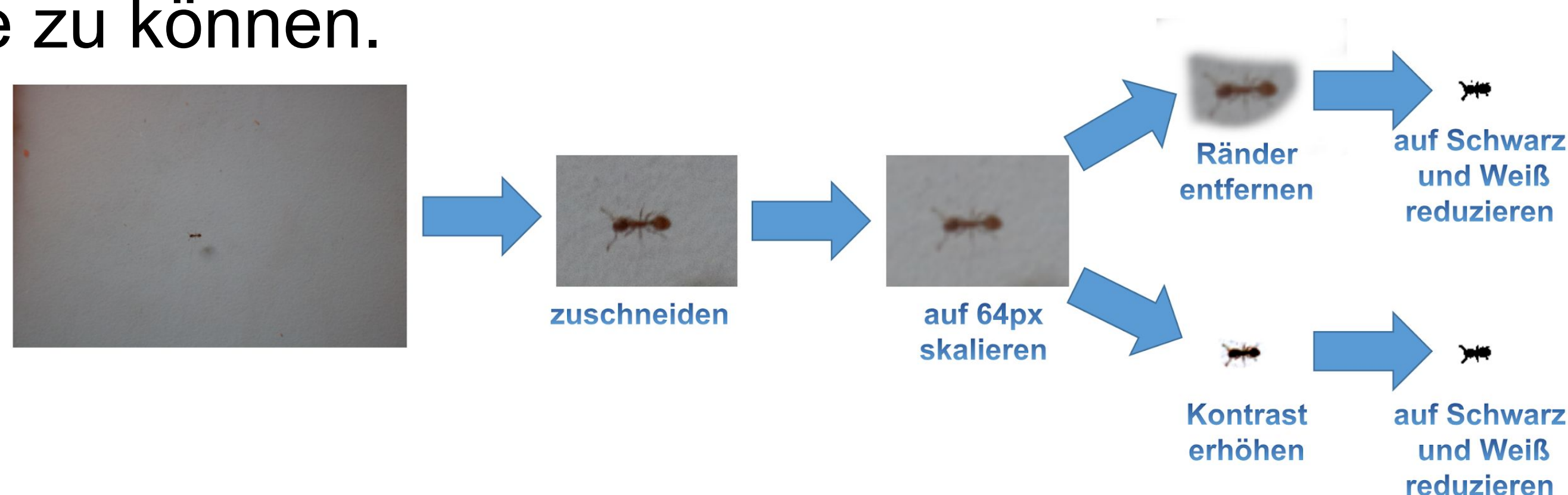


Keras

Vorbereitungen für unser NN

Bildbearbeitung:

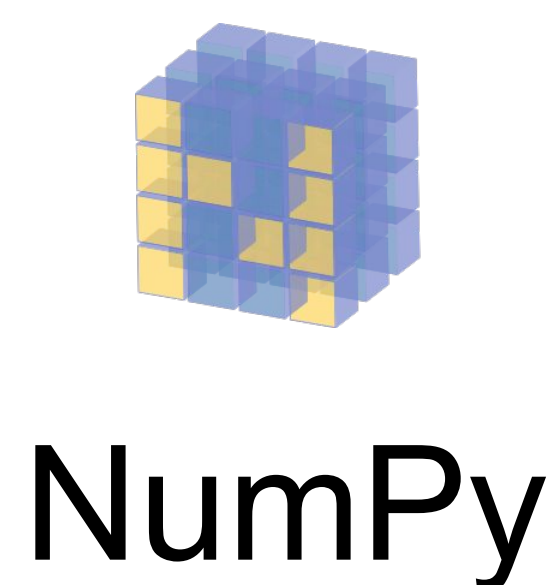
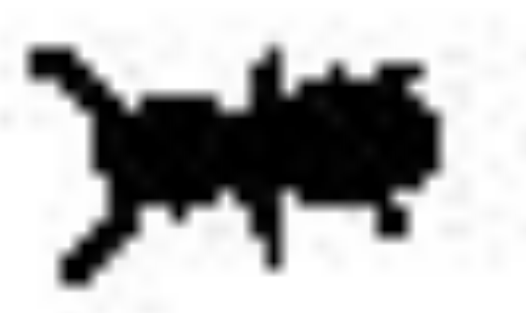
Um ein Neuronales Netz zu trainieren braucht man viele Beispielbilder von den Objekten die es erkennen soll. Bis jetzt haben wir 13 Gigabyte an Rohdaten die allerdings per Hand zugeschnitten und aufbereitet werden müssen. Wir werden auch versuchen dies zu Automatisieren und untersuchen wie das die Ergebnisse verändert. Die Aufbereitung ist nötig, da das neuronale Netz sich sonst unwichtige Dinge wie z.B. den Hintergrund einprägen könnte was sich negativ auf die Ergebnisse auswirken würde. Wir haben auch Fotos der Wasser und Futter - Schüsseln aufgenommen um auch diese erkennen zu können.



In Pixelarray konvertieren:

Damit das NN die Bilder verwerten kann müssen diese in Pixelarrays umgewandelt werden.

$[0,0,0]$ = Schwarz, $[255,255,255]$ = Weiß



[255 255 255] → 1 Pixel
[255 255 255] → Rot
[255 255 255] → Grün
[255 255 255] → Blau
[255 255 255]
[255 255 255]
[255 255 255]
[0 0 0]
[0 0 0]
[255 255 255]
[255 255 255]
...

Codebeispiele (Keras):

```
//erstelle neues Modell  
model = Sequential()
```

```
//füge einen neuen Convolutional Layer mit 32 Neuronen hinzu  
model.add(Conv2D(32))
```

```
//Kompiliere d. Modell = Bereite auf Datenaufnahme u Training vor  
model.compile()
```

```
//Trainiere d. Modell (x_train und y_train sind die Trainingsdaten)  
model.fit(x_train, y_train ...)
```