

# README for convolutional machine learning model 2

---

## 1. Vorbereitung der Datei

---

- Run `classifyData_2.py` (a bissi modifiziert, damit es `.npy` Datei ausgibt.)
- Dann kriegst du `LabeledOriginalMatrix.npy`. (shape 13211,4,25)
- Run [rotatingData.py](#), um mehrere Datei durch Rotation zu erzeugen. (die Anzahl der Variationen der Rotation ist hier einstellbar. Aber normale Einstellung ist schon vielgenug, schätze ich. (X Richtung 10, Y 10, Z 20) (Zeile 100-102, die letzte Zahl))
- Dann hast du `DataForTraining.npy`

## 2. Training und Speichern des Model

---

- Öffne [MLconvolution2.py](#). Checke ob `data=np.load` (Zeile 39) korrekt eingetragen ist. Wir verwenden hier `DataForTraining.npy`.
- Run mal [MLconvolution2.py](#), um zu schauen, wie lange ein Epoch dauert.
- Stop das Run des Programms und stelle die Anzahl des Epochs in der Zeile 122 ein (in `model_1.fit`: das ist der Befehl zum Training.). Dabei muss die maximale Dauer der Rechnung berücksichtigt werden. also \$wie\ lange\ du\ schräfst>maximale\ Dauer = Dauer\ eines\ Epochs \times Anzahl\ des\ Epochs\$
- Ändere den Namen des Save file (Zeile 123, `model_1.save`). Stelle die Zahlen nach X, Y und Z, je nachdem wie du in [rotatingData.py](#) eingestellt hast.
- Run das Programm

Gute Nacht!

## 3. (falls du selber ausprobieren möchtest) Visualisierung

---

- öffne [LoadAndVisualization2.py](#)
- Checke ob die zu ladende Model und Datei correct sind (Zeile 13 und 17)
- Run das Programm.
- Falls es die Reagierung des 3D Bild zu langsam ist, kannst du skip in Zeile 79 vermehren, um die gesamte Anzahl der Punkten zu verringern.

## 4 Wenn du Lust hast...

---

Wenn du etwas mit dem Model rumspielen möchtest, kannst du mehr hidden layer hinzufügen oder andere activation functions verwenden. Sie sind im Block “Model erzeugen (Zeile 88)”