

Motivation



- Implementierung eines Game Controllers, welcher über Gehirnströme, ein Videospiel steuern soll.
- Erlangen von Erkenntnissen über das Konzept "Brain Computer Interface".
- Abgezielt war ein Game Controller mit 3 Freiheitsgraden (Degrees of Freedom).
- Erkenntnisse über ein breites Spektrum von Datenverarbeitung, Datenaufbereitung und der Umsetzung von maschinellen Lernprozessen in Form eines Neuronalen Netzes.
- Ausblick über zukünftige Errungenschaften im Bereich BCI geben.



Programmierung

Programmiersprache: Python Vielzahl an Bibliotheken im Bereich von wissenschaftlichen Arbeiten, wie z.B. Mathematik oder maschinellem Lernen.

Hilfreiche Bibliotheken:

SciPy: Mathematisch, wissenschaftliche Python Bibliothek.

NumPy: Ermöglicht komplexe Funktionen für Array- und Matrixoperationen.

Pywt: (Diskrete-)Wavelet-Transformation

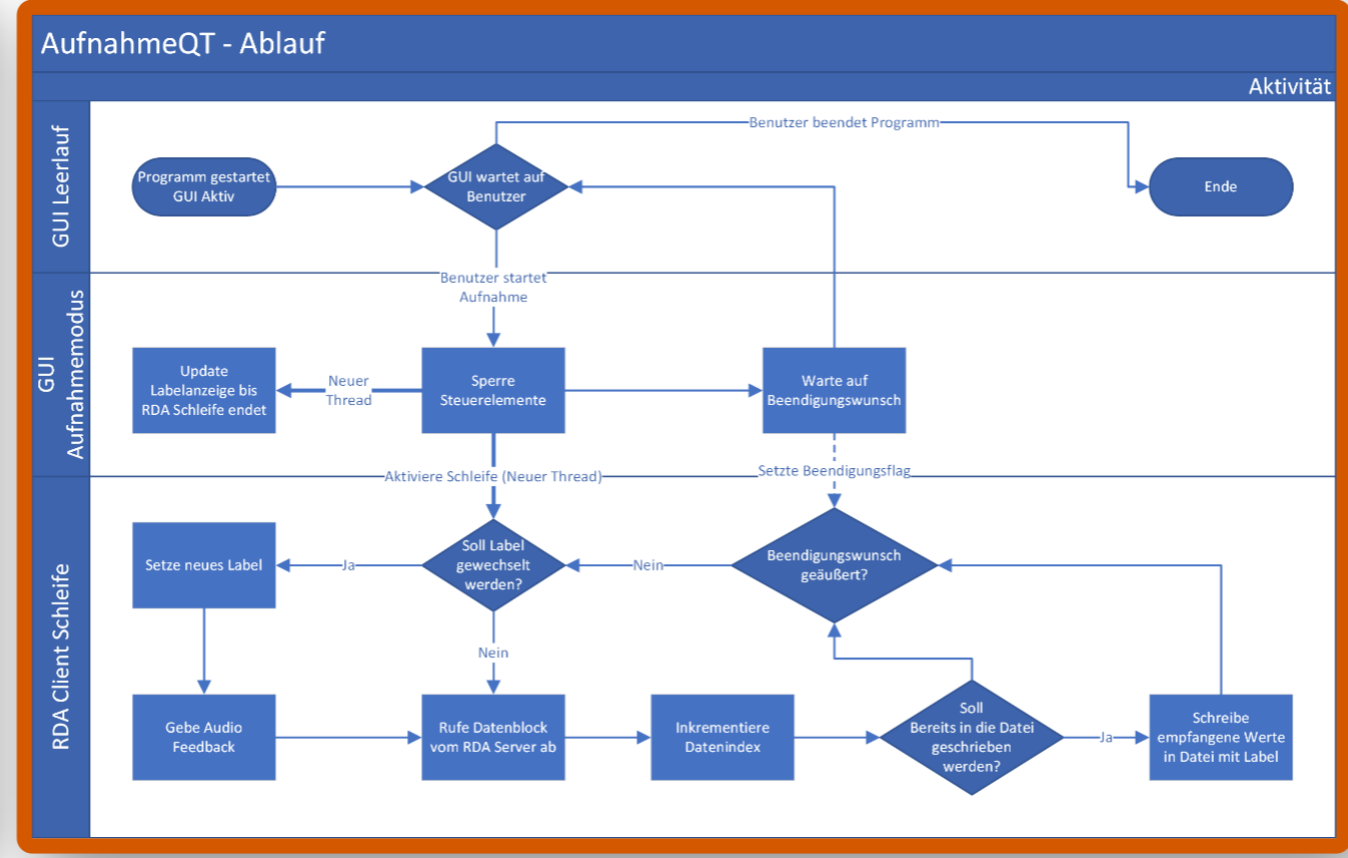
PyTorch: Frameworks für die Implementierung eines Neuronalen Netzes

PyNput: Framework für die Simulation von Tasteneingaben

Qt: Plattformübergreifendes Framework um dynamische grafische Benutzeroberflächen mit Interaktionsmöglichkeiten zu erstellen.

Ablauf

- Datenaufnahme und Verarbeitung mit Aufnahme QT (Bandpassfilter & Wavelet Transformation)
- Auswertung von EEG Daten
- Einrichtung eines Multilayerperzepton Neuronalen Netz
- Trainingvorgänge des Neuronalen Netzes auf einem GPU Cluster
- Liveinterpretation der EEG Daten
 - Liveaufnahme
 - Liveverarbeitung
 - Interpretation der Ergebnisse des Neuronalen Netzes
- Mapping von Ergebnissen der Livedaten auf Spieleingaben (z.B. in Flappybird)



Vorverarbeitung

Bandpass:

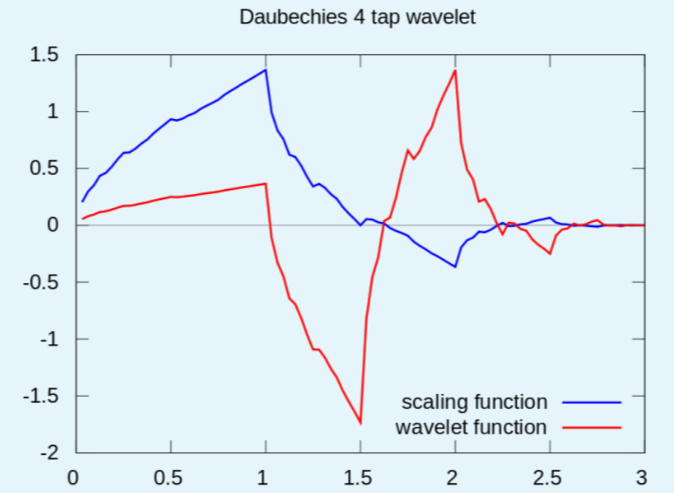
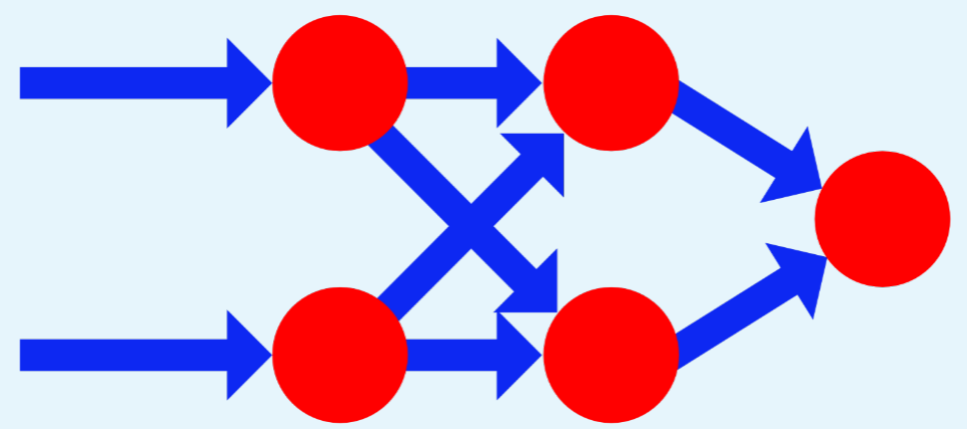
- Butterworth
- 1 - 70Hz, 6. Ordnung.
- Alle Daten unterhalb und oberhalb des definierten Durchlassbereiches werden aus den Rohdaten entfernt.

Wavelets:

- Wellenförmige beschränkte mittelwertfreie Funktionen
- Reduzierung der Inputgröße des Neuronalen Netzes
 - Zuständig für Datenkompression
- Art: Daubechies-4 Wavelet

Neuronales Netz:

- 2 Hidden-Layer
- 120 Neuronen auf Hidden-Layer 1
- 25 Neuronen auf Hidden-Layer 2
- 2 Neuronen auf Output-Layer
- 56 Werte pro Elektrode
 - 13 Werte pro Elektrode werden als Input Vektor benutzt



Quellen

L. M. Alonso-Valerdi and F. Sepulveda. Python in brain-computer interfaces (bci): development of a bci based on motor imagery. In 2011 3rd Computer Science and Electronic Engineering Conference (CEEC), pages 74-79. IEEE, 2011.

P. Jahankhani, V. Kodogiannis, and K. Revett. Eeg signal classification using wavelet feature extraction and neural networks. In IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing (JVA'06), pages 120-124. IEEE, 2006.

T. Kalayci and O. Ozdamar. Wavelet preprocessing for automated neural network detection of eeg spikes. IEEE engineering in medicine and biology magazine, 14(2):160-166, 1995.

Brain Image: <https://images.livemint.com/rf/Image-621x414/LiveMint/Period1/2016/02/17/Photos/brain.jpg>

db4 Wavelet: <https://commons.wikimedia.org/wiki/File:Daubechies4-functions.svg>

Brain Products: <https://www.brainproducts.com/index.php>

