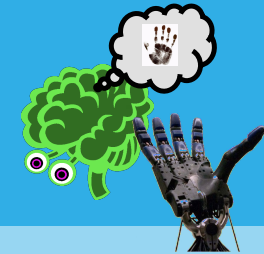
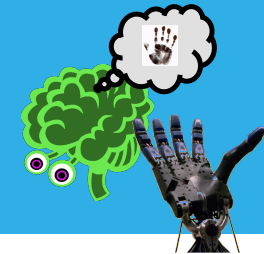


Seminar Brain-Machine Interfaces



BMI
more practical...

EEG Geräte

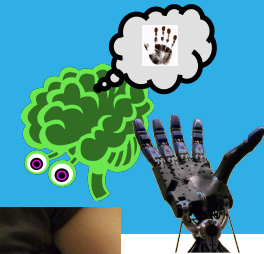


- 16 – 256 Kanäle
- Abschirmung
- interne Filter



- Hauptkriterien für die BMI Entwicklung:
 - Können die Daten in Echtzeit aus dem Gerät gewonnen werden?
 - Gibt es eine API?
 - Wie gut kann der Gerätetreiber in eigenen Code eingebunden werden?
 - Ist das Gerät portabel / mobil?
 - Unter welchem OS kann das Gerät laufen?

EEG Elektroden



- einzelne Elektroden
- Elektroden Kappen
- Elektroden Netze

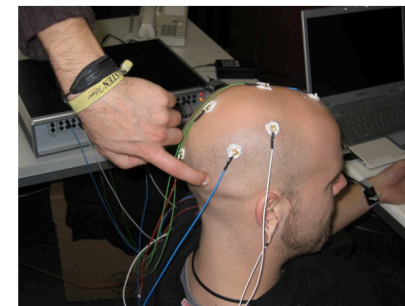
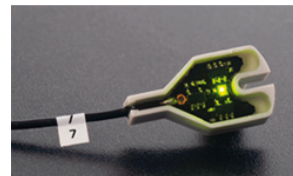
Unterschiedliche Materialien:

- Goldbecher Elektroden
- Ag/AgCl

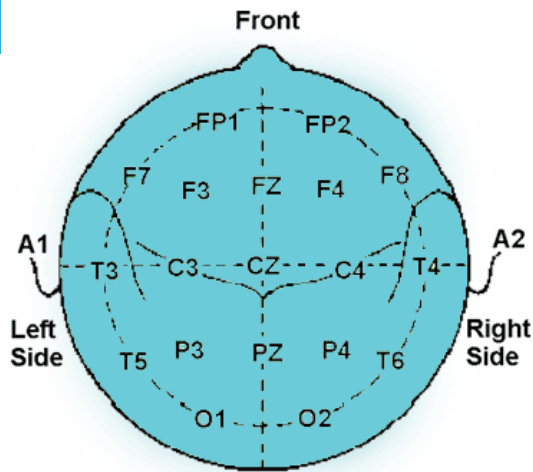
▪ Auch: „trockene“
Elektroden

Seit kurzem:

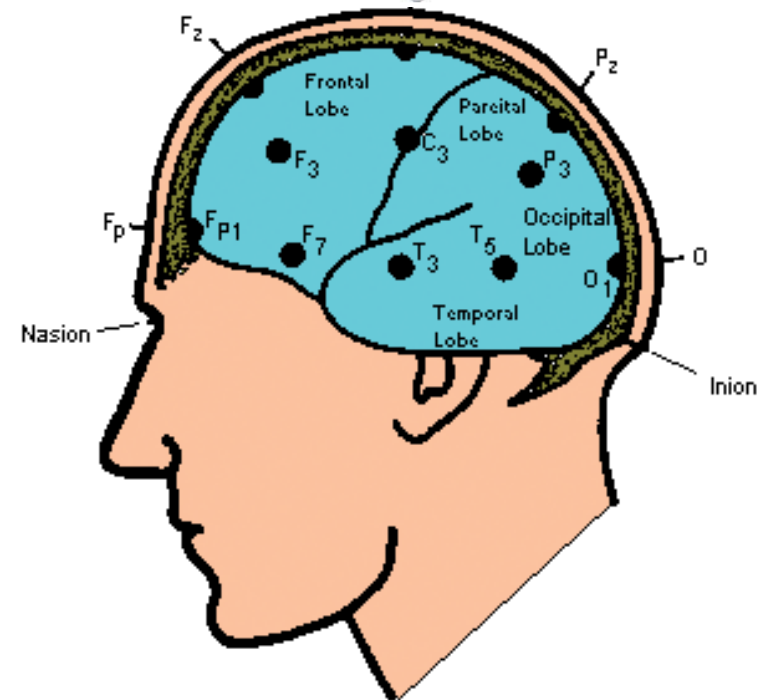
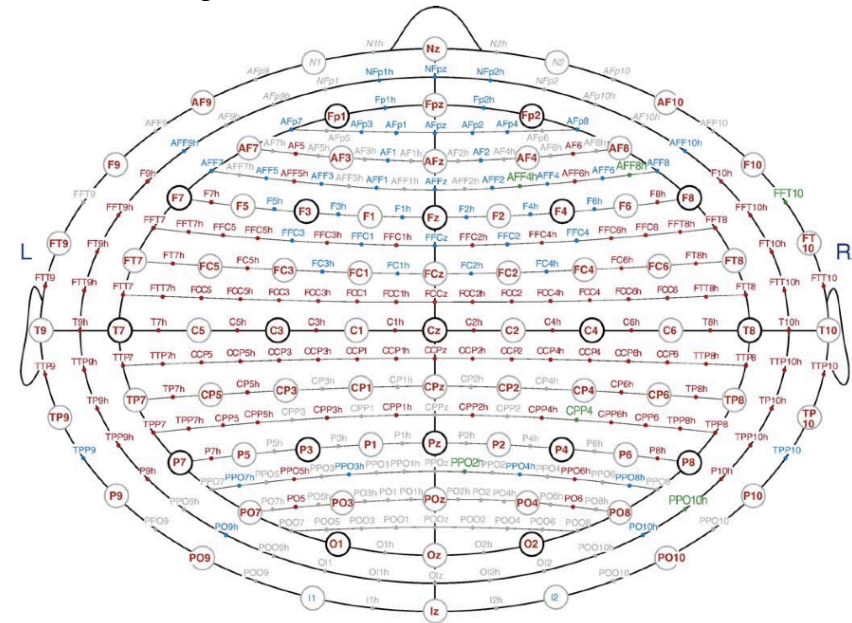
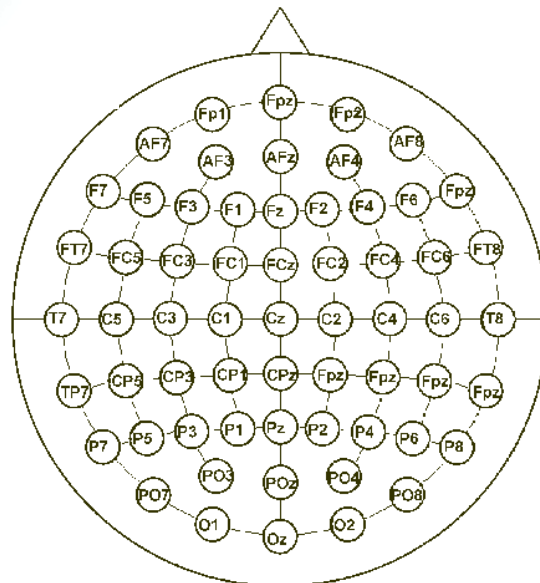
- Aktive Elektroden:
→ Jede Elektrode ist direkt mit
einem eigenen Verstärker
verbunden



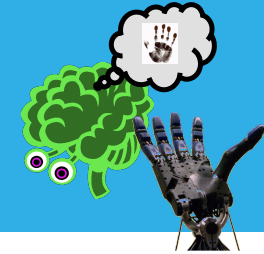
International 10-20 System



Back

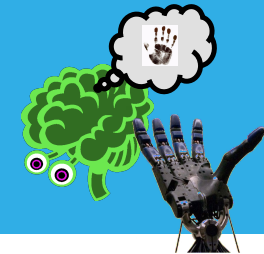


Elektroden Fixierung



- Ausmessen des Kopfes, evt. Bestimmung der Elektrodenpositionen
- Vorhandlung der Haut mittels „abrasiver Paste“
- Anbringen der Elektrode
- Verwendung eines elektrolytischen Gels um Kontakt zwischen Haut und Elektrode herzustellen
- Referenzelektrode (oft am Ohrläppchen)
- Ground Elektrode (oft am anderen Ohrläppchen)

Impedanzen



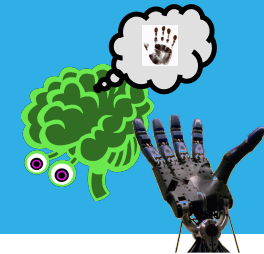
Messung des Leitwiderstandes der Haut

- Leitwiderstand sollte unterhalb eines bestimmten Schwellwerts liegen:
 - $>10 \text{ k}\Omega$: gut
 - $> 5 \text{ k}\Omega$: sehr gut
 - $> 2 \text{ k}\Omega$: perfekt, aber kaum zu erreichen
- Alle Widerstände sollten eine ähnliche Größenordnung haben

Die Impedanzen bestimmen maßgeblich die Güte der Daten!

Es gibt kein Datenverarbeitungsverfahren, das saubere Daten ersetzen kann!

Labor



- Ideal sind vollständig elektro-magnetisch abgeschirmte Räume („Black Box“)

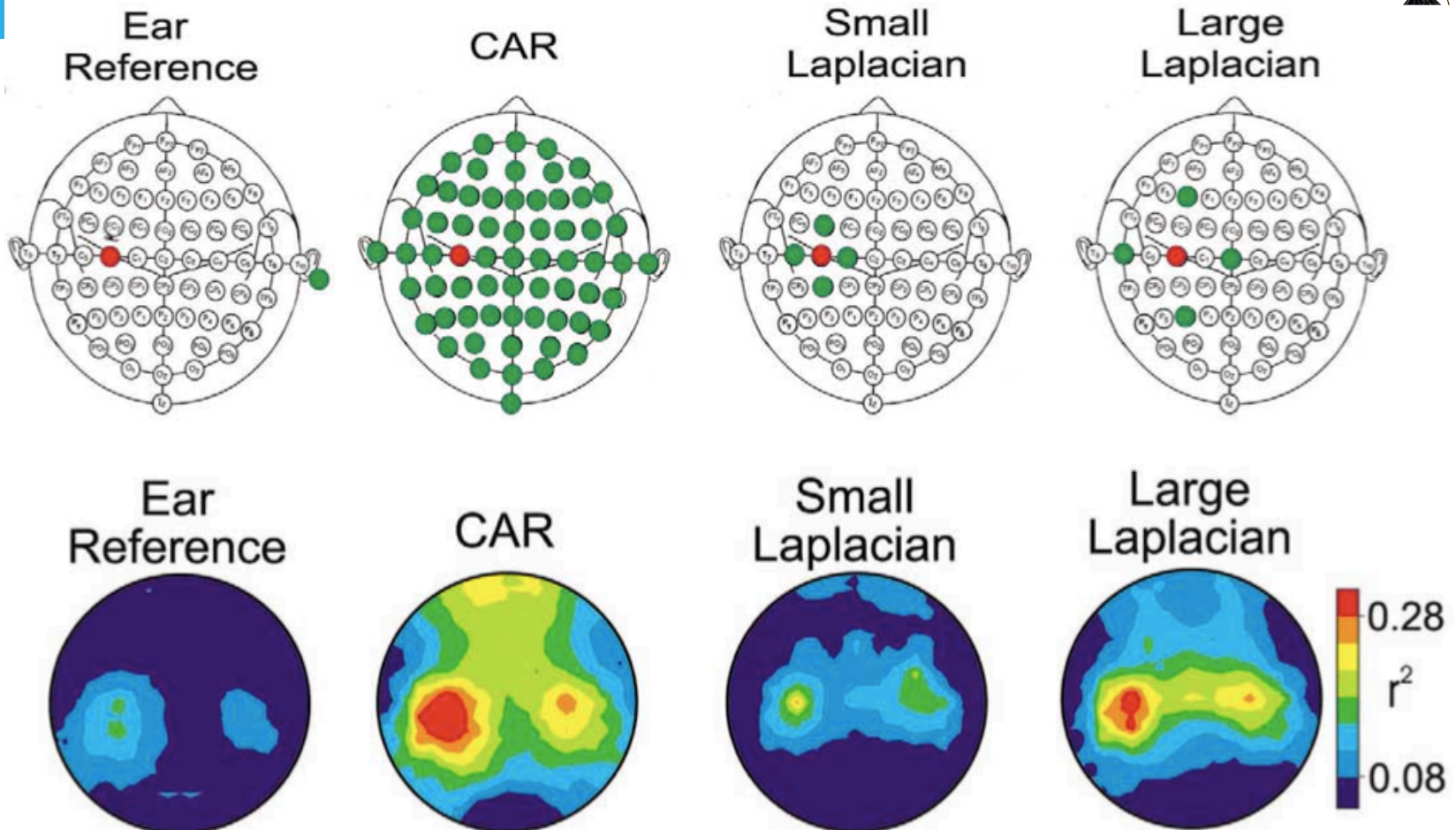
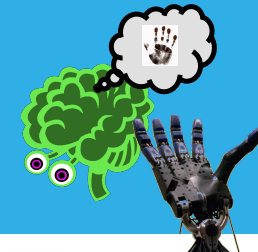
→ Für BMIs nicht praktikabel

- Artefakte durch:
 - Netzbrummen (50 Hz)
 - andere elektrische Geräte (Computer, Monitore, Roboter,...)

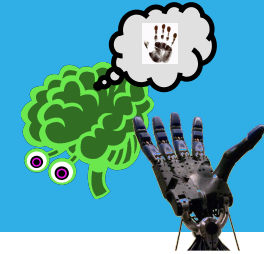
Signale, die in einem BMI verwendet werden sollen, müssen auch bei Aufnahme in einem “normalen” Labor gewonnen werden können!

(Geht z.B. nicht bei gamma Wellen, da interessanter Frequenzbereich auch bei 40 – 60 Hz.)

Referenzierung



Räumliche (spatiale) Filter

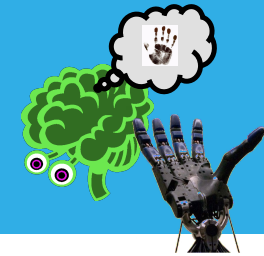


- Räumliche Filter eignen sich nur für Signalkomponenten, die stark lokalisiert sind
 - Motor Imagery

Hier bewirken sie eine **Kontrastverstärkung** an den relevanten Elektrodenpositionen.

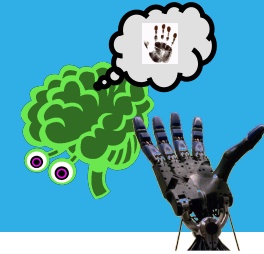
- Sie eignen sich nicht für Signalkomponenten, die über den ganzen Kopf verteilt sind.
 - P300
 - Slow Cortical Potentials

Frequenzfilter: Bandpassfilter / Notchfilter



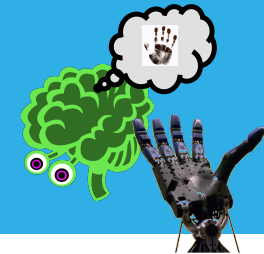
- Hochpassfilter (z.B. 0.5)
 - Entfernt langsamen Drift aus dem Daten, welcher als sehr langsame Schwingung aufgefasst werden kann.
- Tiefpassfilter (z.B. 30 Hz)
 - Die für aktuelle BMIs relevanten Signalanteile liegen im Bereich von unter 30 Hz. Zum Teil werden durch den Tiefpass auch Muskelartefakte entfernt.
- Notchfilter (Bandstoppfilter) (z.B. 48-52 Hz)
 - Entfernen des Netzbrummens (im Prinzip bei entspr. Tiefpass nicht mehr notwendig).

Akquisition von Trainingsdaten



- Alle aktuellen EEG-basierten BMI Systeme bauen auf **überwachte Lernverfahren**.
- Vor der Benutzung eines Systems ist die Akquisition von Daten für das Training des jeweiligen Klassifikators notwendig.

BMI Trainingsmodus



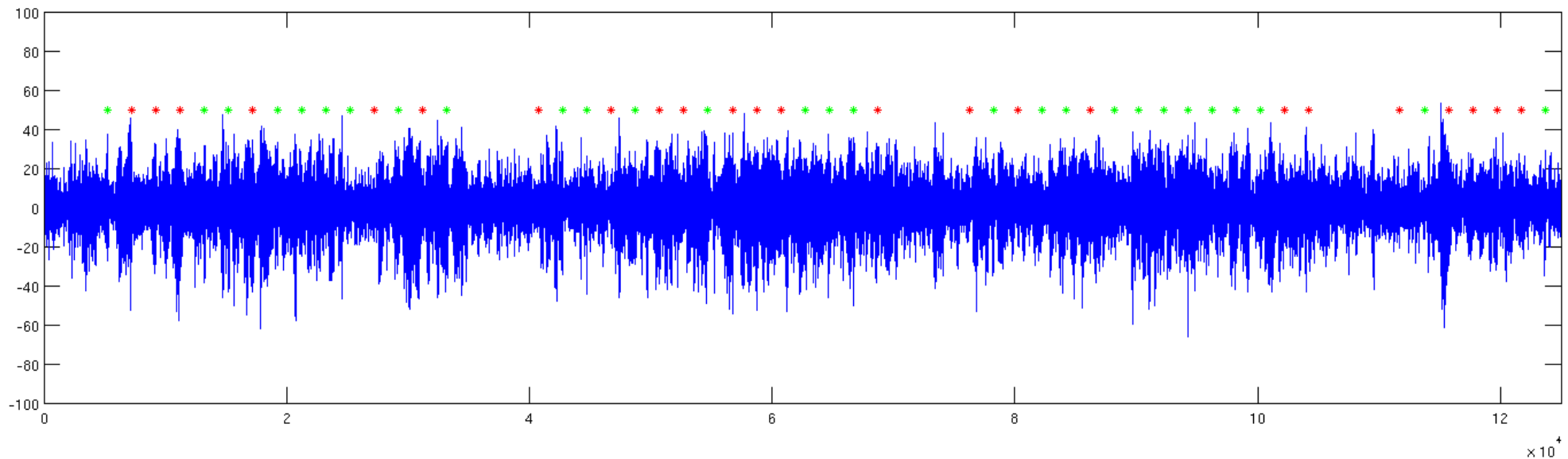
Aufzeichnung von Trainings-Daten

System läuft im Offline Modus

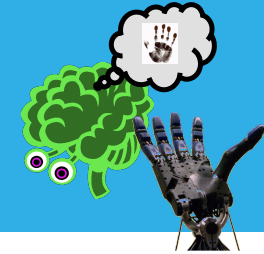
Daten

Label

Segmentierungspunkte



BMI Trainingsmodus



EEG

Aufzeichnungsmodul

System läuft im Offline Modus

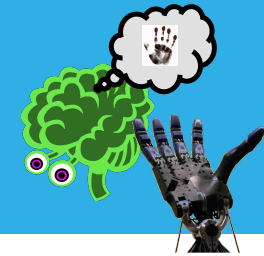
Daten

Label

Segmentierungspunkte

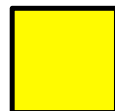
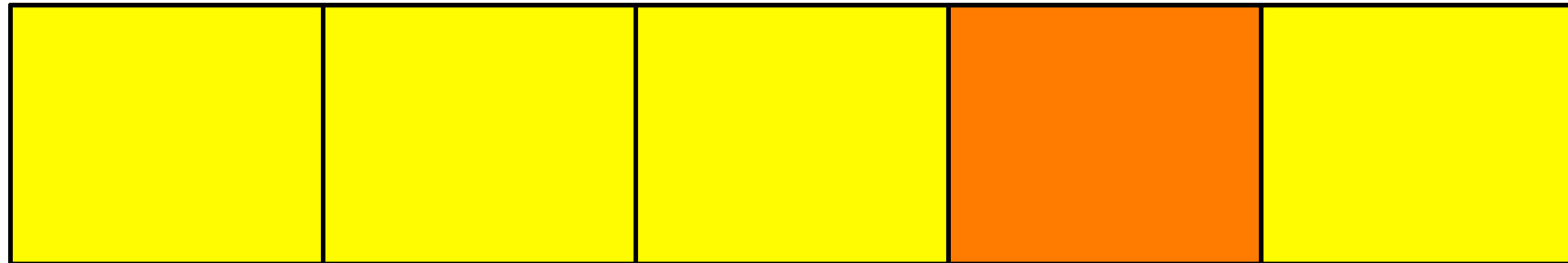
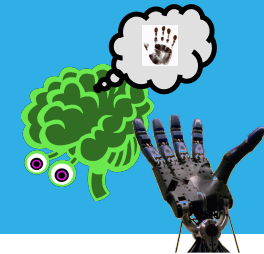
Trainingsmodul
(i.d.R. unabhängig vom online System, d.h. eigene Applikation)

Trainingsmodul



- 1.) Bandpass Filterung - Kanalweise
- 2.) Segmentierung
- 3.) evt. Vorverarbeitung, z.B.
 - Independent Component Analysis (ICA)
 - Principal Component Analysis (PCA)
 - Common Spatial Pattern (CSP)
- 4.) Klassifikator Training einschl. Kreuzvalidierung

Kreuzvalidierung

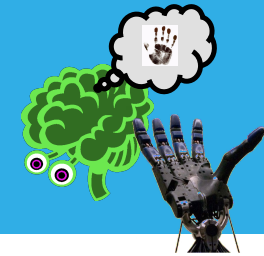


Testset



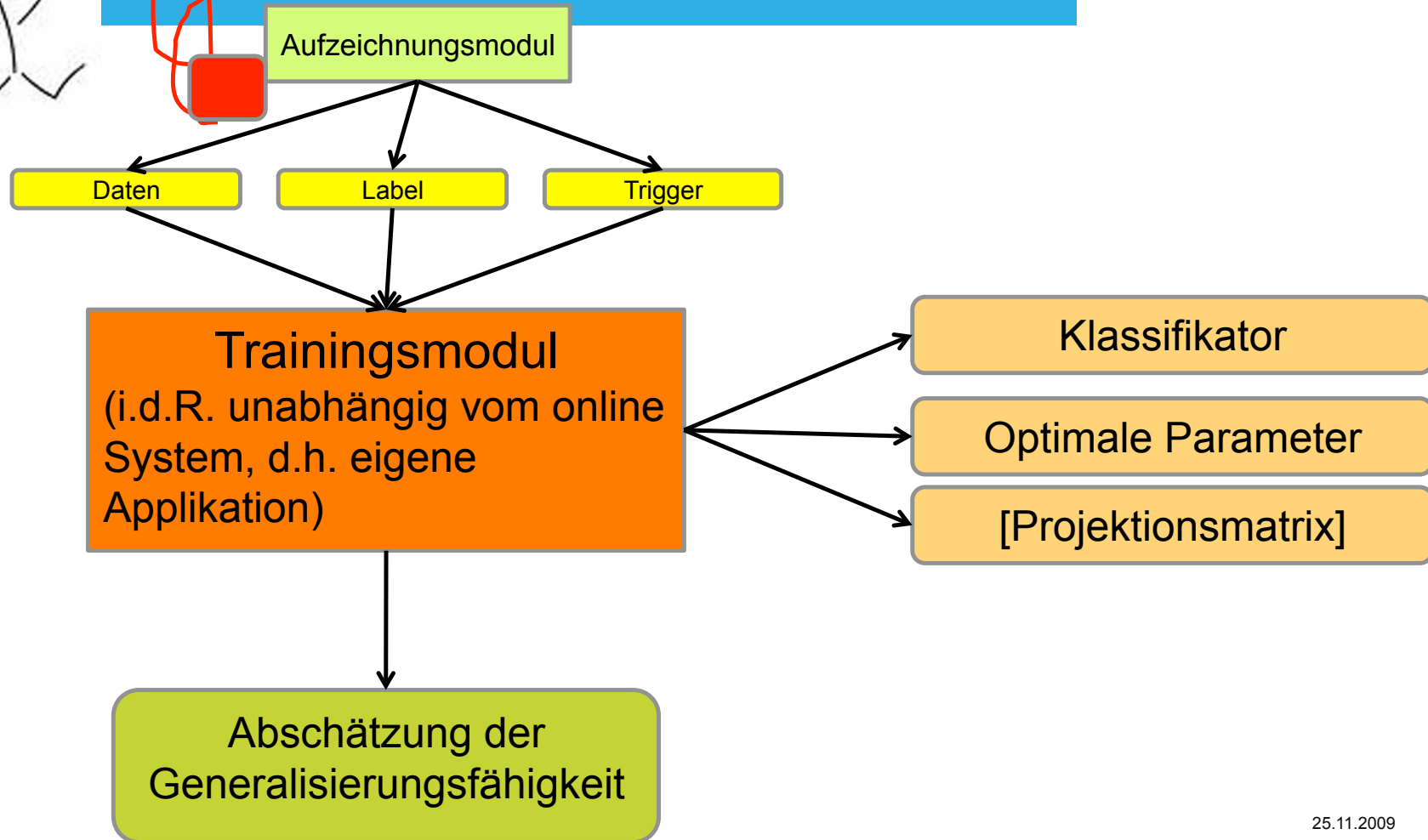
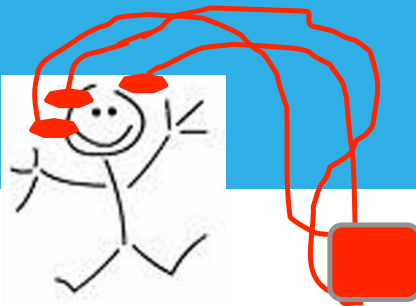
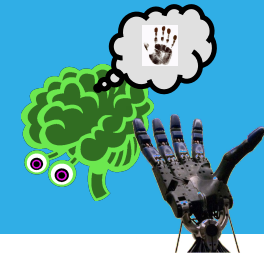
Trainset

Training - Schema

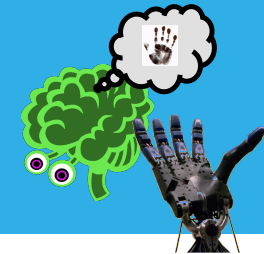


- 1: Init parameters:
 highpass, lowpass, epoch length, #foldings.
- 2: Load data, label, trigger.
- 3: Bandpass data channel wise.
- 4: Segment data according to triggers .
- 5: Create epochs of specified length.
- 6: Sort epochs according to labels.
- [7: Balance data set: Use as many epochs from one class as from the other class.]
- 8: Create sets for cross-validation (balanced for classes).
- 9: **For** i = 1 to #foldings **do**
- 10: select all sets excluding set i and merge to trainset.
- [11: Preprocess trainset, obtain Transformation matrix W].
- [12: Apply W to trainset.
- [13: Apply W to set i (testset) .
- 14: Train classifier Phi with trainset.
- 15: Classify testset using Phi.
- 16: **end for**
- 17: **Calculate** mean classification rate.

BMI Trainingsmodus



Klassifikator – Parameter Suche



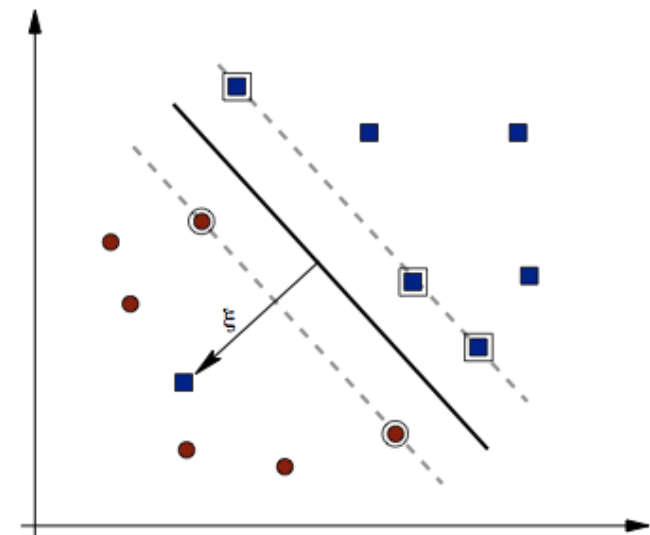
Die Parameter Suche für einen Klassifikator (z.B. für Support Vector Maschinen) findet innerhalb der Kreuzvalidierung statt.

z.B. 2 Parameter bei nicht-linearer SVM mit Gauss-Kernel:

- C (Regularisierungs Parameter)
- γ („Weite“ der Gauss Fkt.)

Verfahren:

- 1.) Einstellen des Parameters
- 2.) Kreuzvalidierung mit Training und Test
- 3.) Bestimmung der Güte des Klassifikators
- 4.) GOTO 1 bis Parameterraum vollständig abgesucht
- 5.) Auswahl der Parameter anhand der besten Klassifikationsleistung



Online System



EEG

Aufnahmemodul

System läuft im Online Modus

Daten:
1-n samples x channels

Segmentierungspunkte

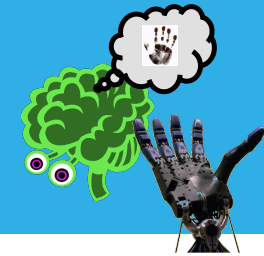
Klassifikator, evt.
Projektionsmatrix

Klassifikationsmodul

Berechnetes Label
(oder kontinuierlicher
Wert wie W.keit)

Kontrollmodul
Steuerung + User Feedback

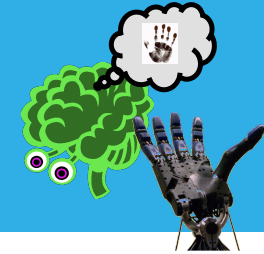
Online System



Algorithm 1 Online classification and feedback generation.

- 1: Initialize $t \leftarrow 0$, $w \leftarrow W$ { W is the window size parameter}
 - 2: Load pre-trained classifier Φ
 - 3: Buffer data until $t=w$
 - 4: **for** $t = w+1$ to T **do**
 - 5: Create vector $\vec{y}_t = (y_t^1, \dots, y_t^n)^T$
 - 6: Create window matrix $\vec{x}_t = (\vec{y}_{t-W+1}, \dots, \vec{y}_t)$
 - 7: Preprocess \vec{x}_t
 - 8: Feature extraction: obtain \hat{x}_t
 - 9: Classify \hat{x}_t
 - 10: **if** $\Phi(\hat{x}_t) > \theta$ **then**
 - 11: Generate feedback
 - 12: **end if**
 - 13: $t \leftarrow t + 1$
 - 14: **end for**
-

Self-paced vs. System-paced



System-paced:

Die on- und offsets, wann das BMI benutzt wird, werden vom System vorgegeben.

→ Klassifiziert wird nur, welche der Klassen aktuell auftritt.



Self-paced:

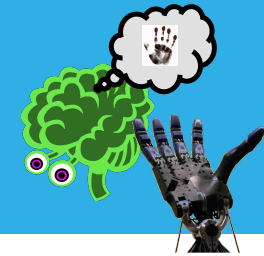
Das System läuft ohne dass eine zeitliche Steuerung vorgegeben wird. Der Benutzer entscheidet, wann er das Interface steuern möchte.

→ Das System muss zusätzlich zur Klassifikation eine Segmentierung erkennen.

“No control state”



Probleme mit Self – paced Systemen



- **Klassifikation:**
In der Regel ist der Klassifikator auf eine bestimmte zeitliche Struktur der Featurevektoren trainiert, bei einem self-paced System ist diese sehr variabel.
- **Evaluation des Systems:**
Keine direkte Kontrolle des aktuellen Zustands möglich.
Evaluation nur über User Feedback oder externe Kontrollmechanismen (wie z.B. Blickrichtung) möglich.
- **Stand der Dinge:**
Erste Ansätze, aber noch keine zufriedenstellenden Lösungen.