Boosting Research with Machine Learning

Franziska Oschmann
Scientific IT Services, ETH
10th of July, 2019

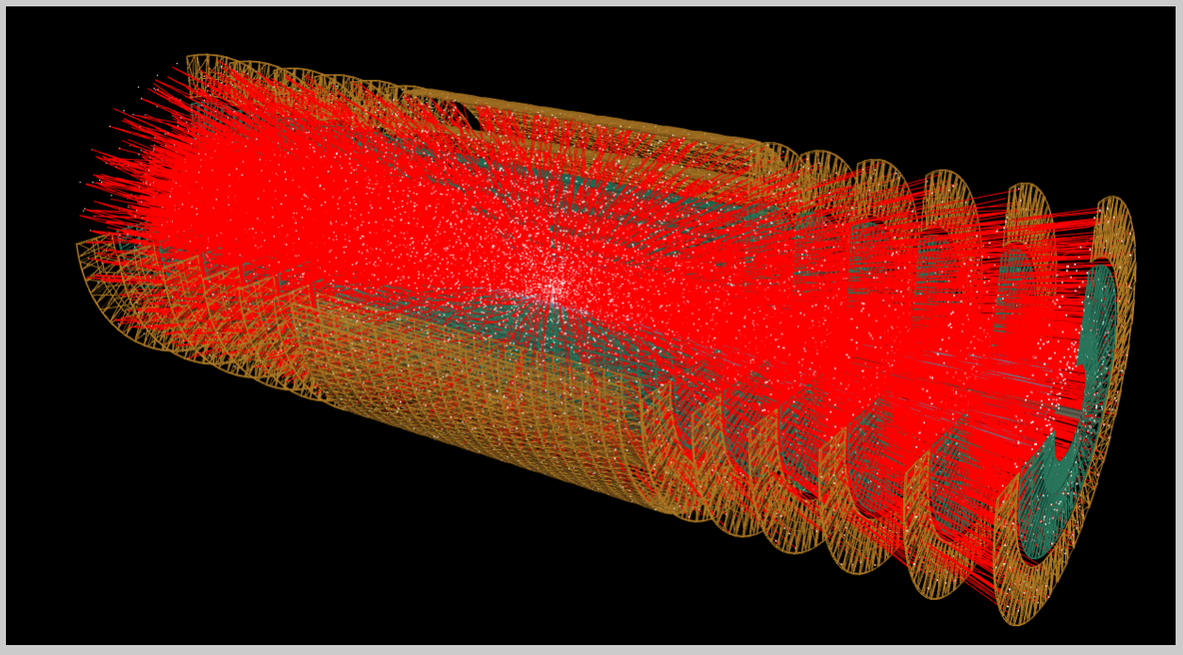
Informatikdienste

Scientific IT Services

Examples for ML in research

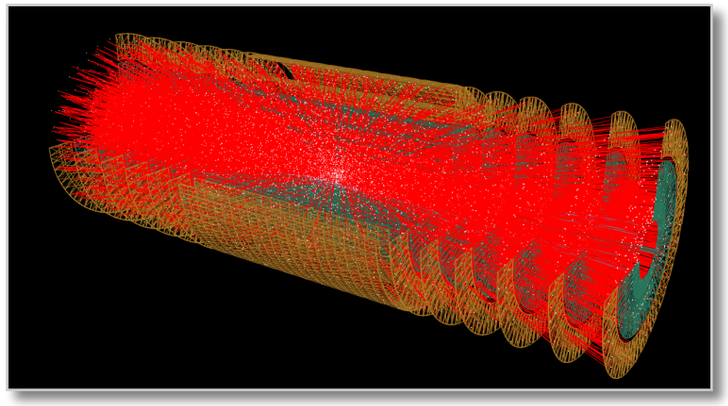
Examples for ML in research

Discovery and characterisation
of new particles

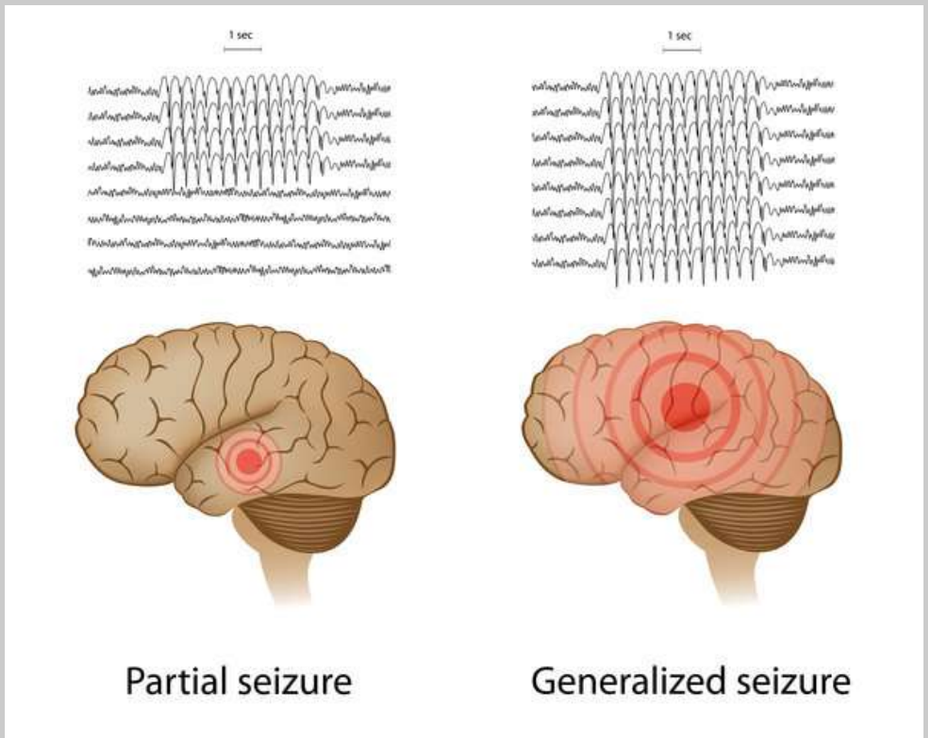


https://home.cern/

Examples for ML in research

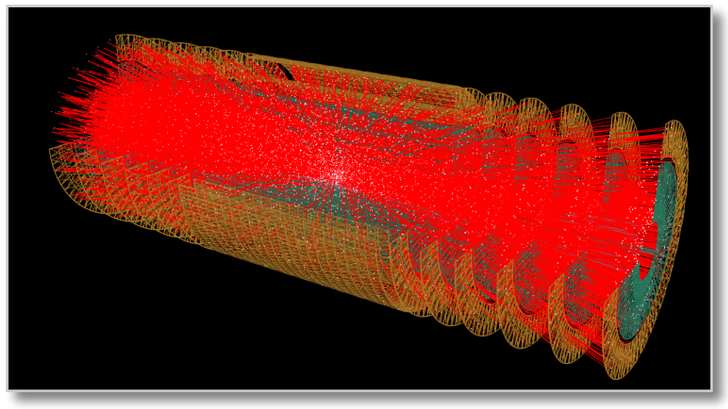


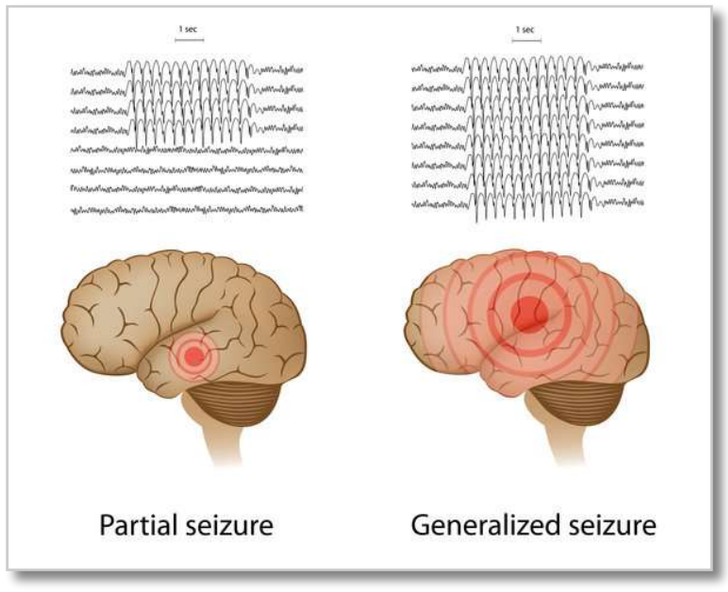
Prediction of
epileptic seizures



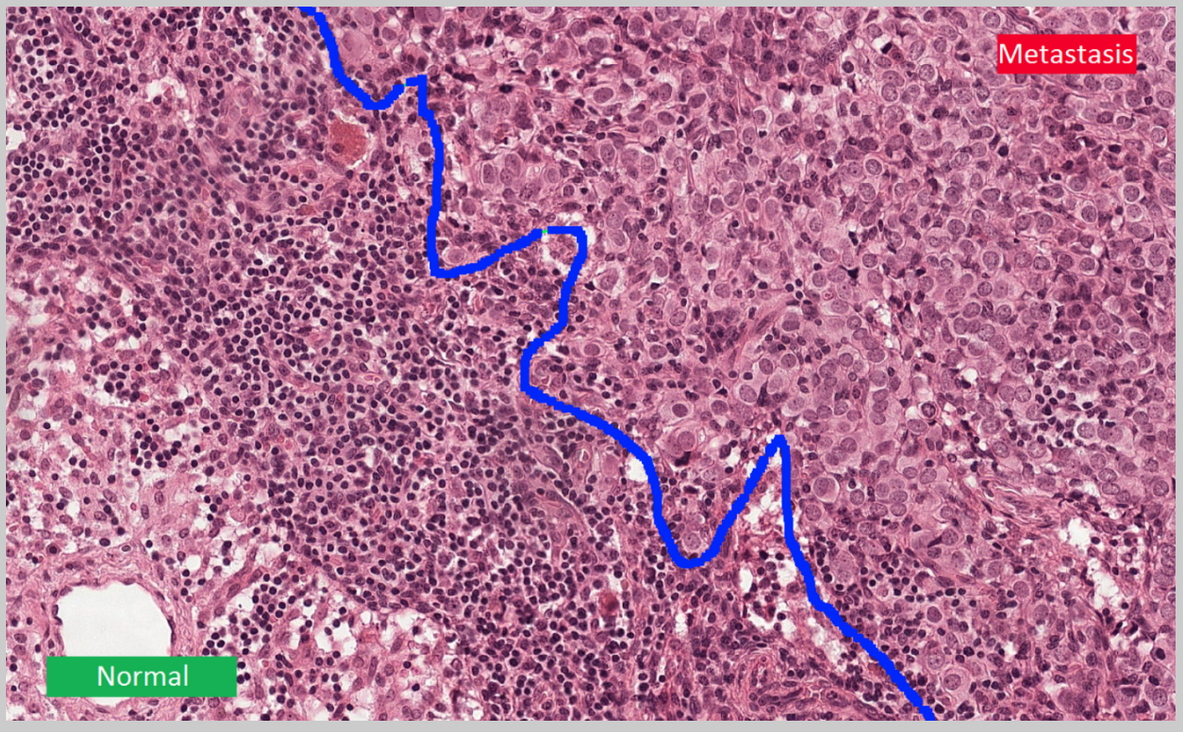
https://medicalxpress.com

Examples for ML in research



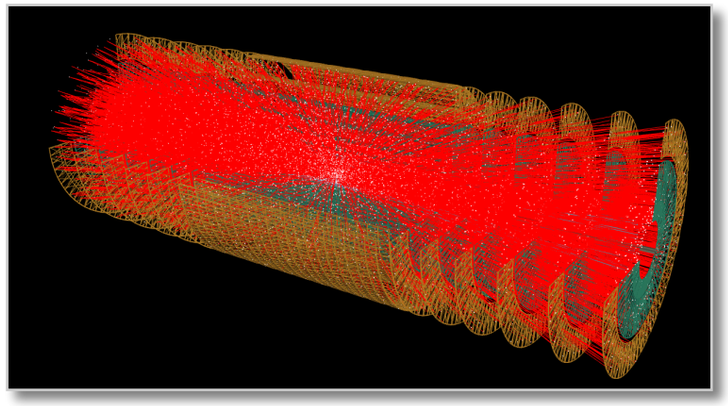


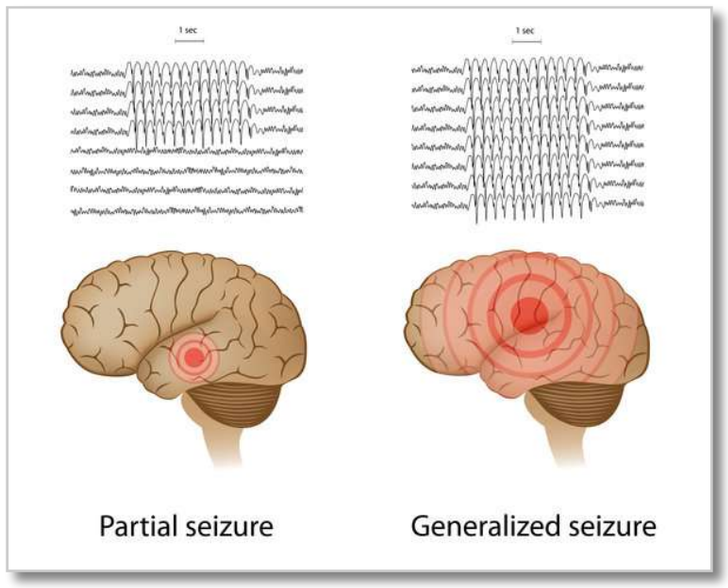
Characterisation
of cancer regions

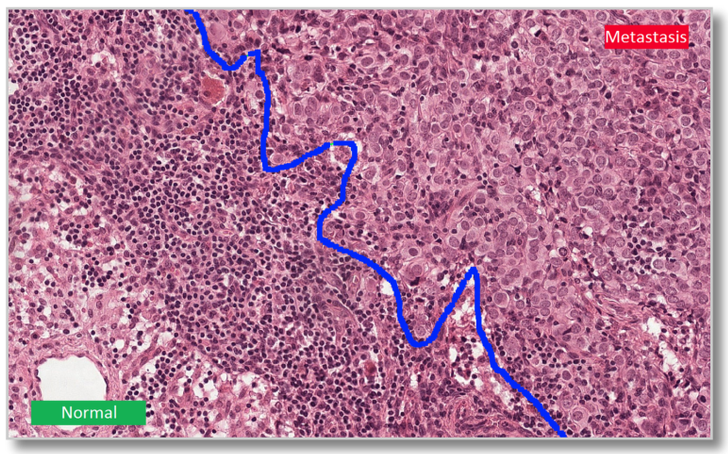


https://camelyon16.grand-challenge.org

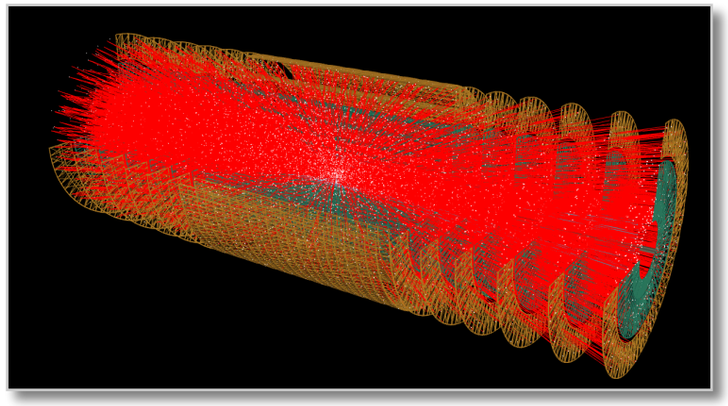
Examples for ML in research

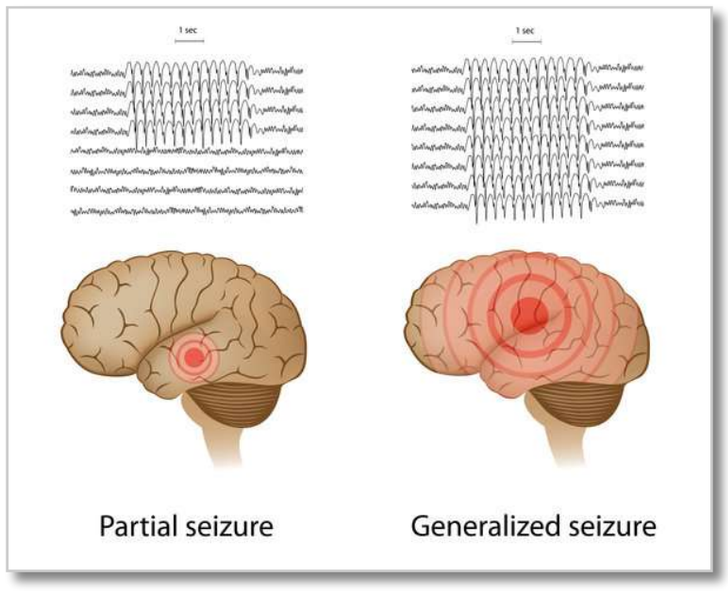






Examples for ML in research

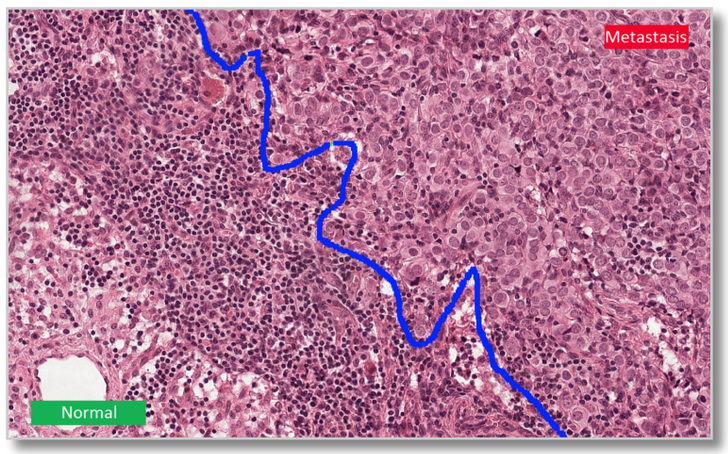




Applications of ML in research:

- Uncover hidden patterns in data
- Automatisation of time-consuming processes

Examples for ML in research



Applications of ML in research:

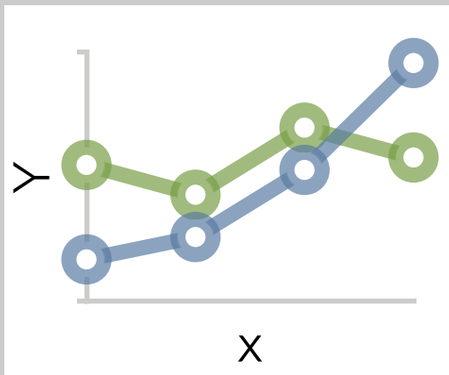
- Uncover hidden patterns in data

- Automatisation of time-consuming processes

How to apply ML in research?

How to apply ML in research?

Data



Preprocessing

scipy

pandas

pandas

Model

scikit-learn

keras

Prediction

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \dots \end{pmatrix}$$

How to apply ML in research?

Data

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from my_helper import data, preprocess

## Load data
X = data.data
y = data.target
```

Preprocessing

```
## Preprocessing of data
X_proc = preprocess(X)

## Split into training and validation set
X_train, X_val, y_train, y_val = train_test_split(
    X_stand, y, test_size=0.33)
```

Model

```
## Model
lr = LogisticRegression()
lr.fit(X_train, y_train)
```

Prediction

```
y_pred = lr.predict(X_val)
print(accuracy_score(y_val, y_pred))
```

Use case 1: EEG signal detection

Use case 1: Experimental setup

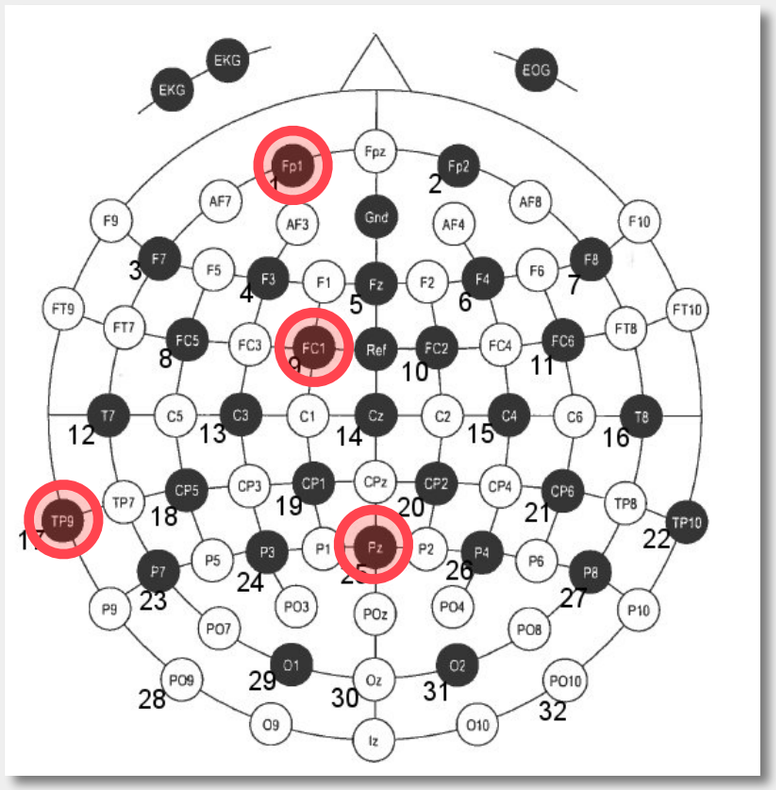
Experimental setup

Hand movement

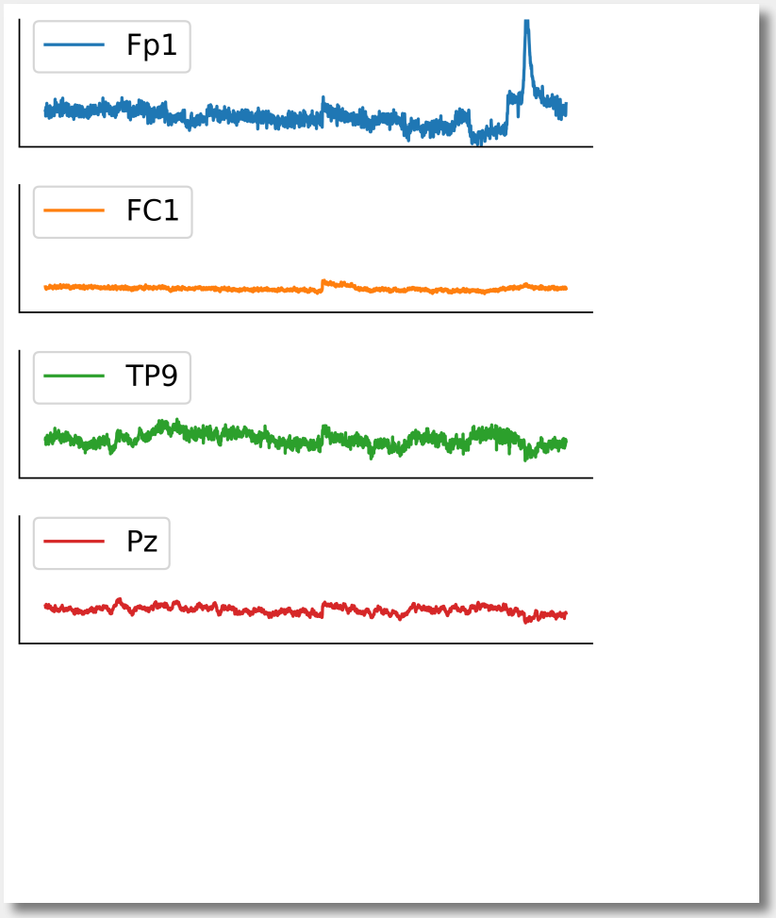
Luciw et al., Nature, 2014

Use case 1: Preprocessing

Recording

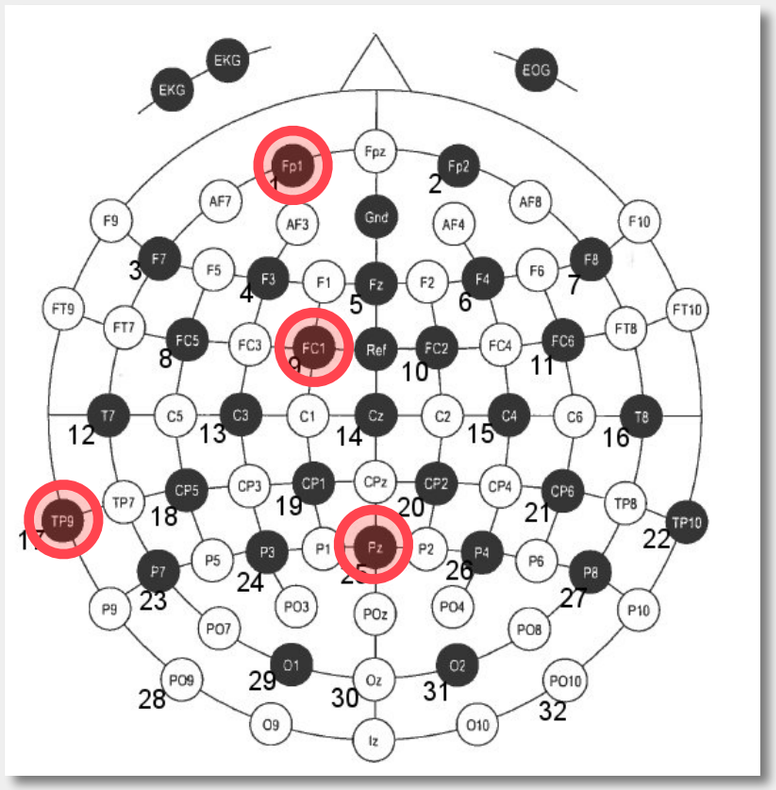


Recording

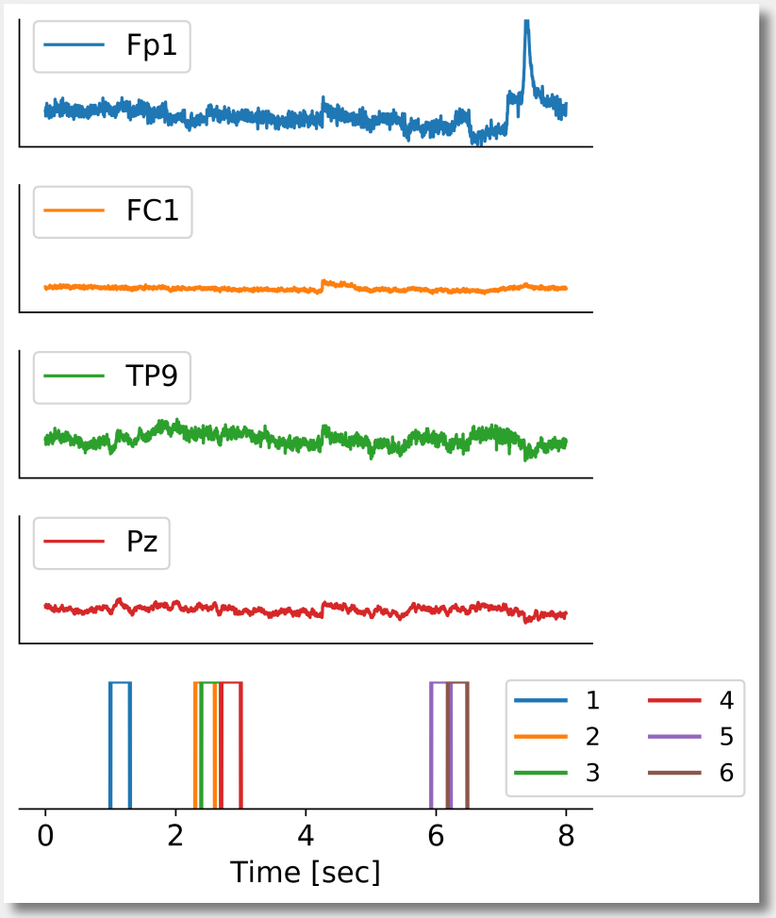


Use case 1: Preprocessing

Recording

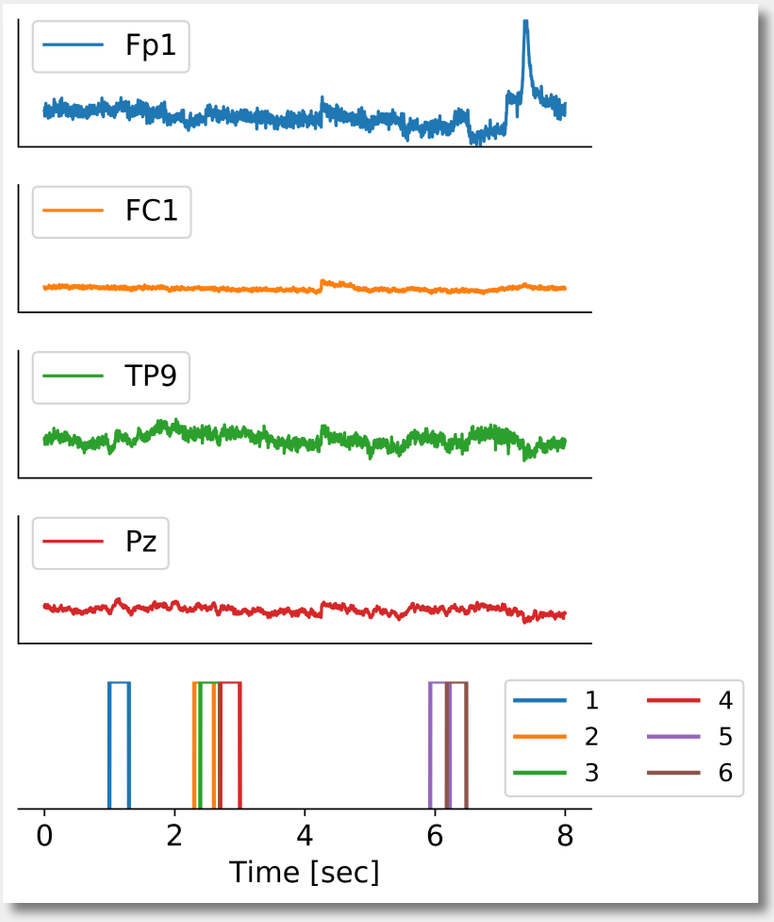


Recording

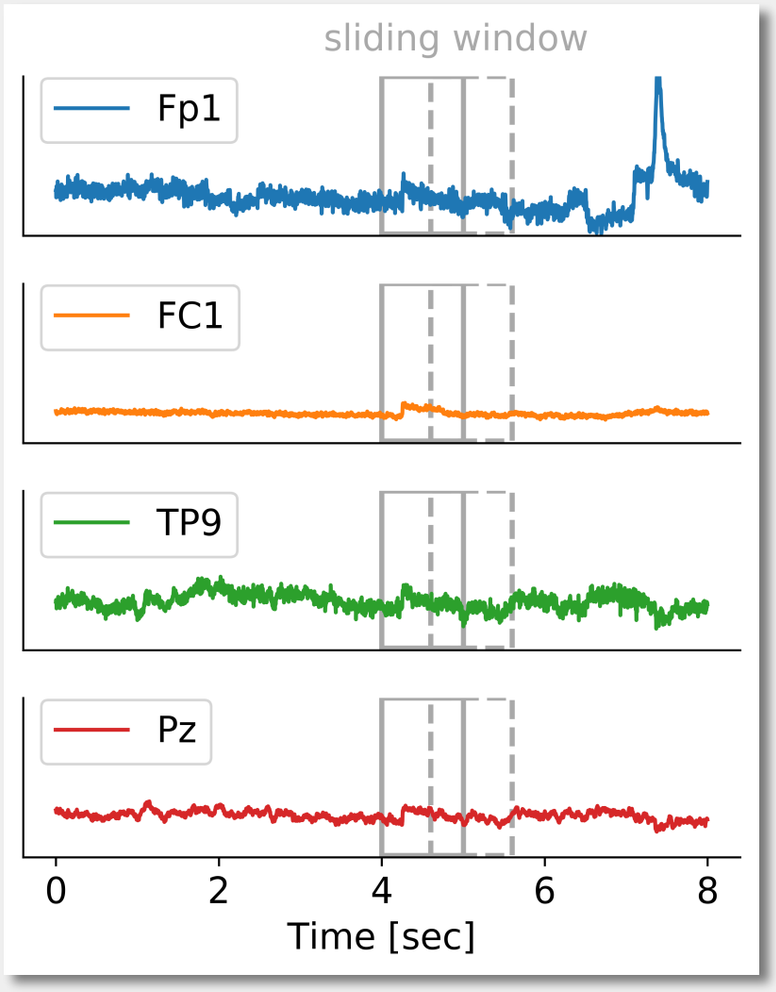


Use case 1: Preprocessing

Recording

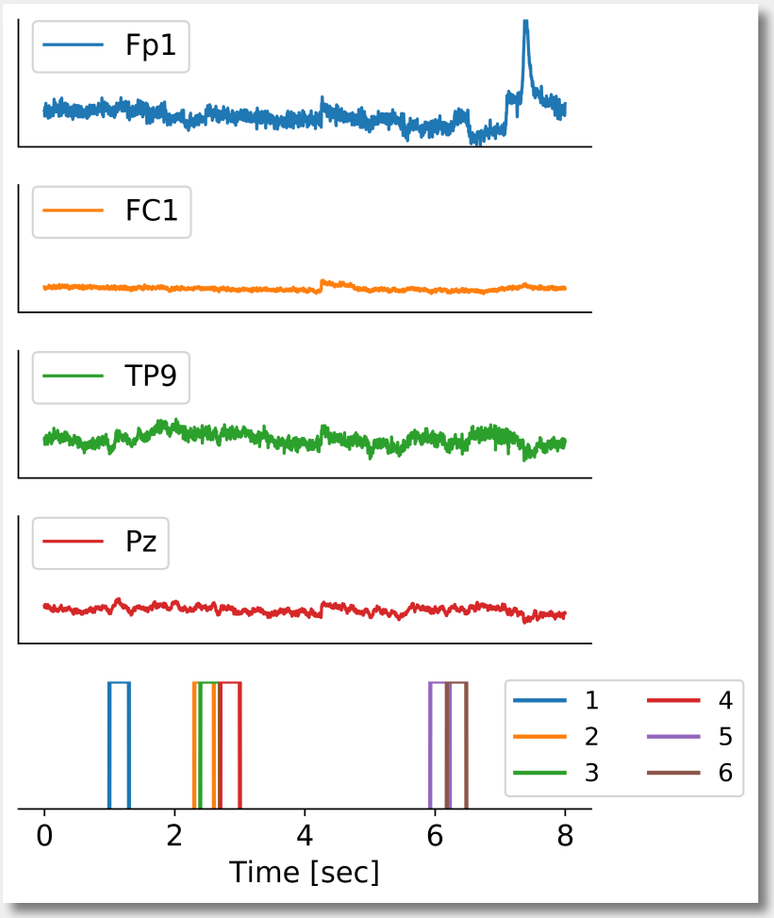


Sliding window

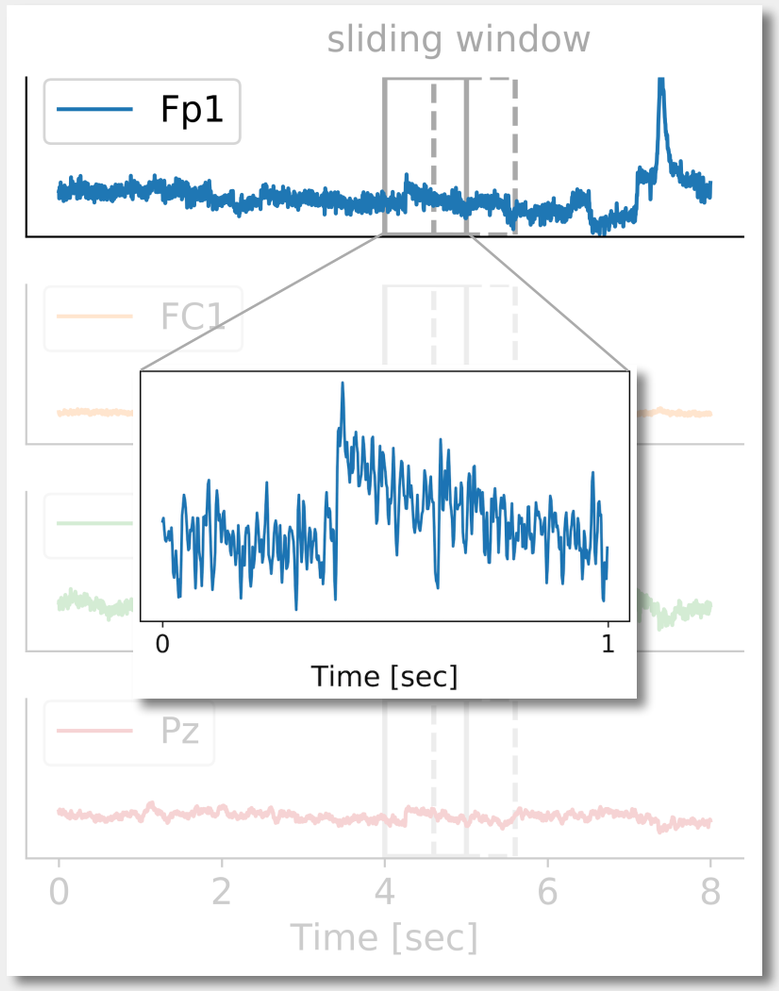


Use case 1: Preprocessing

Recording

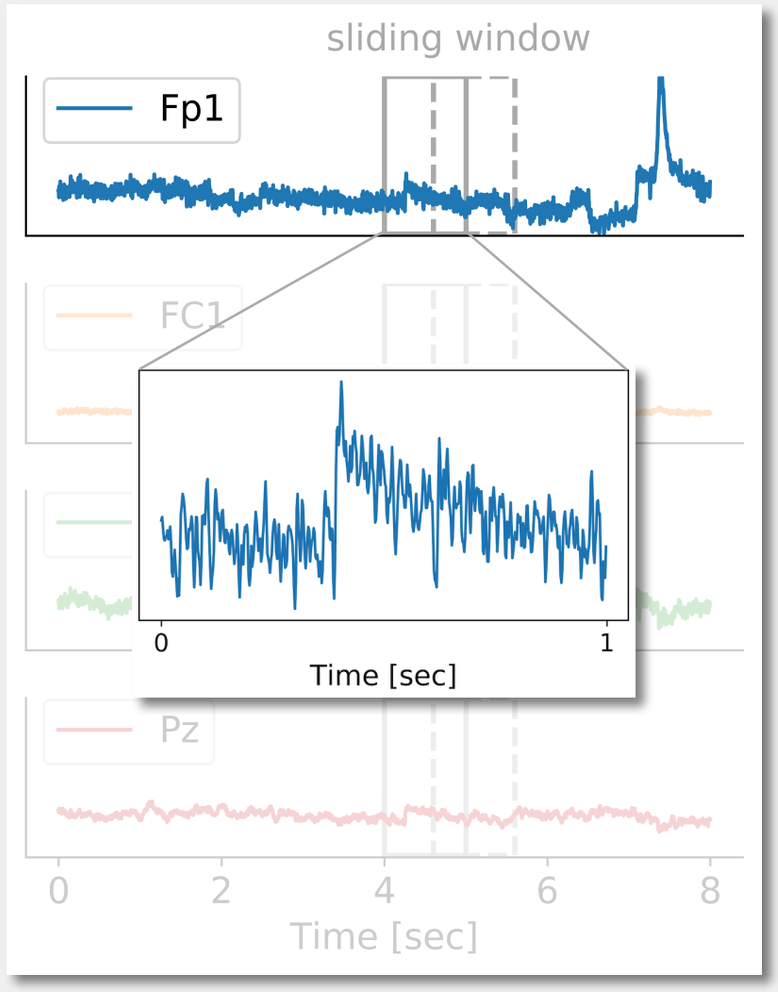


Sliding window

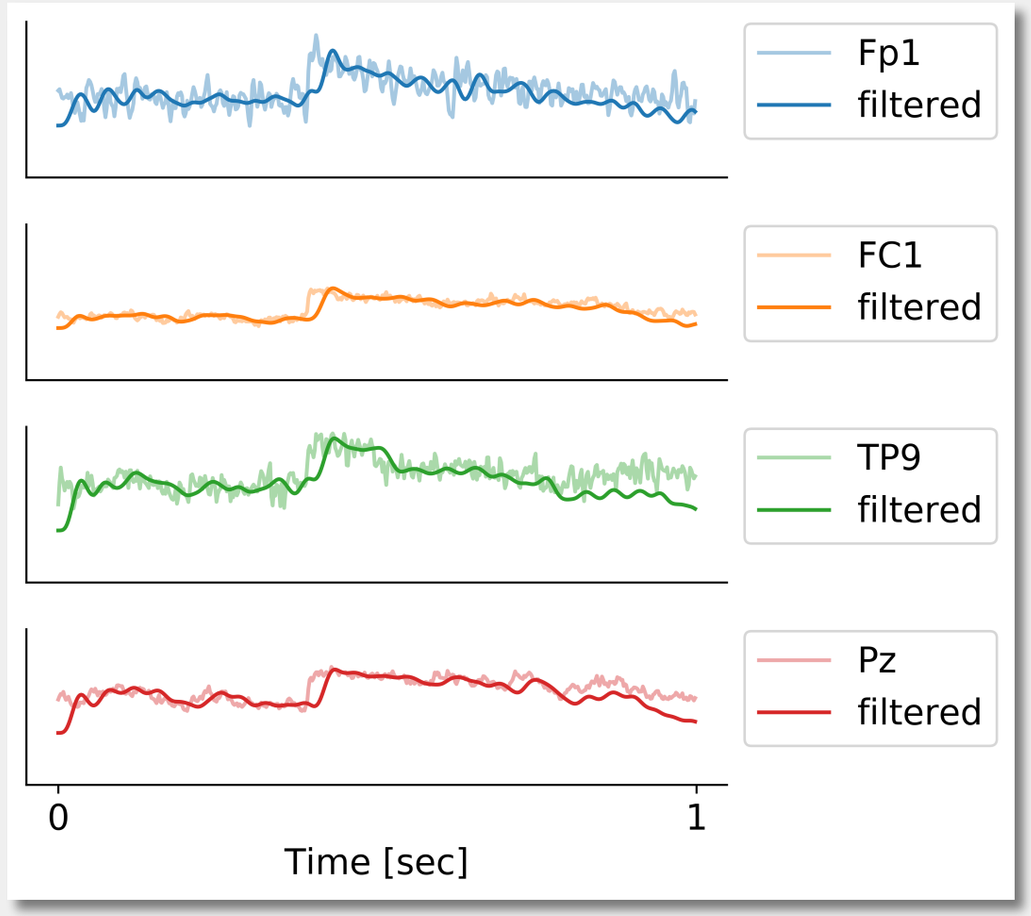


Use case 1: Preprocessing

Sliding window

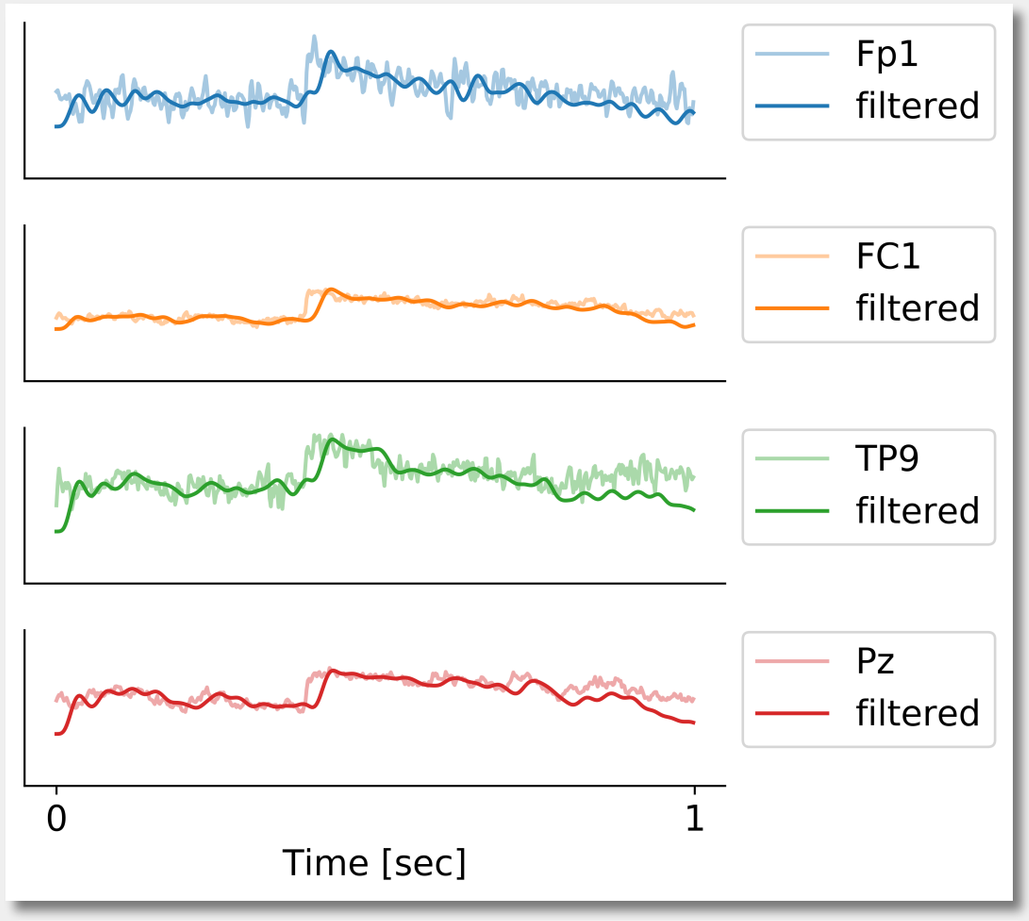


Low-pass filter

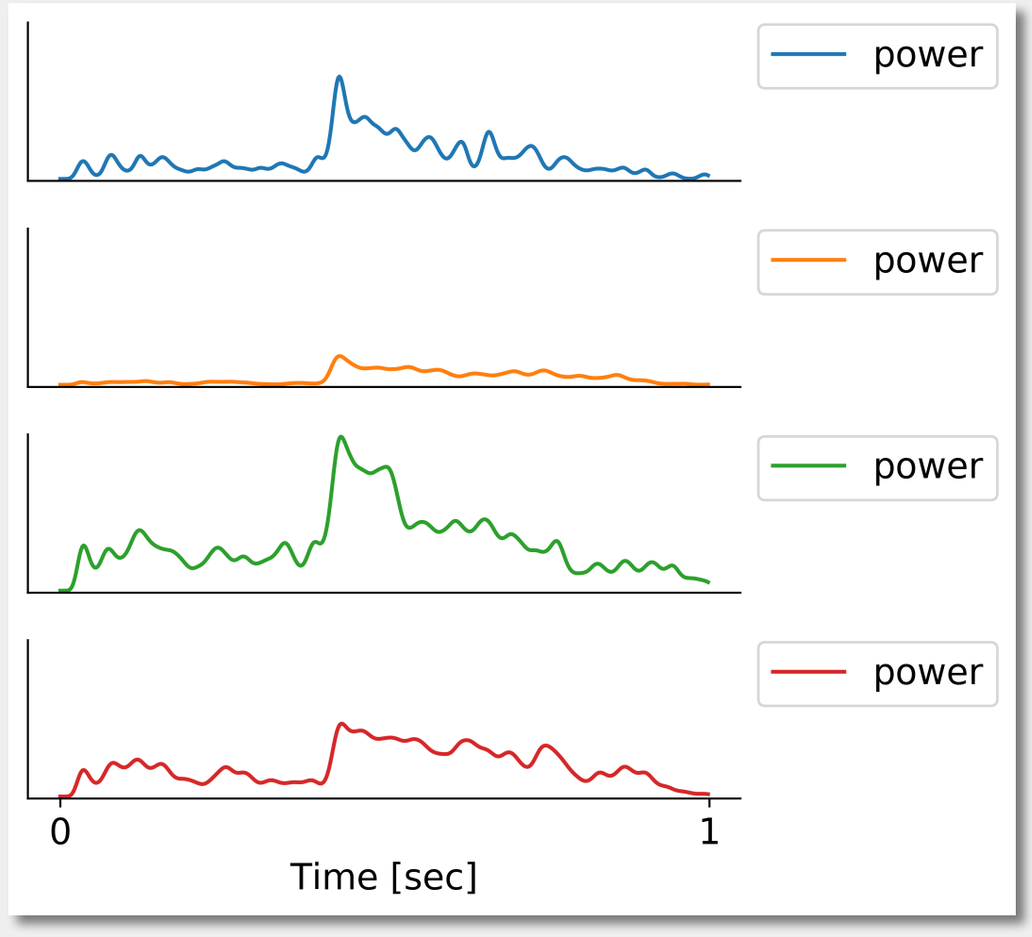


Use case 1: Preprocessing

Low-pass filter

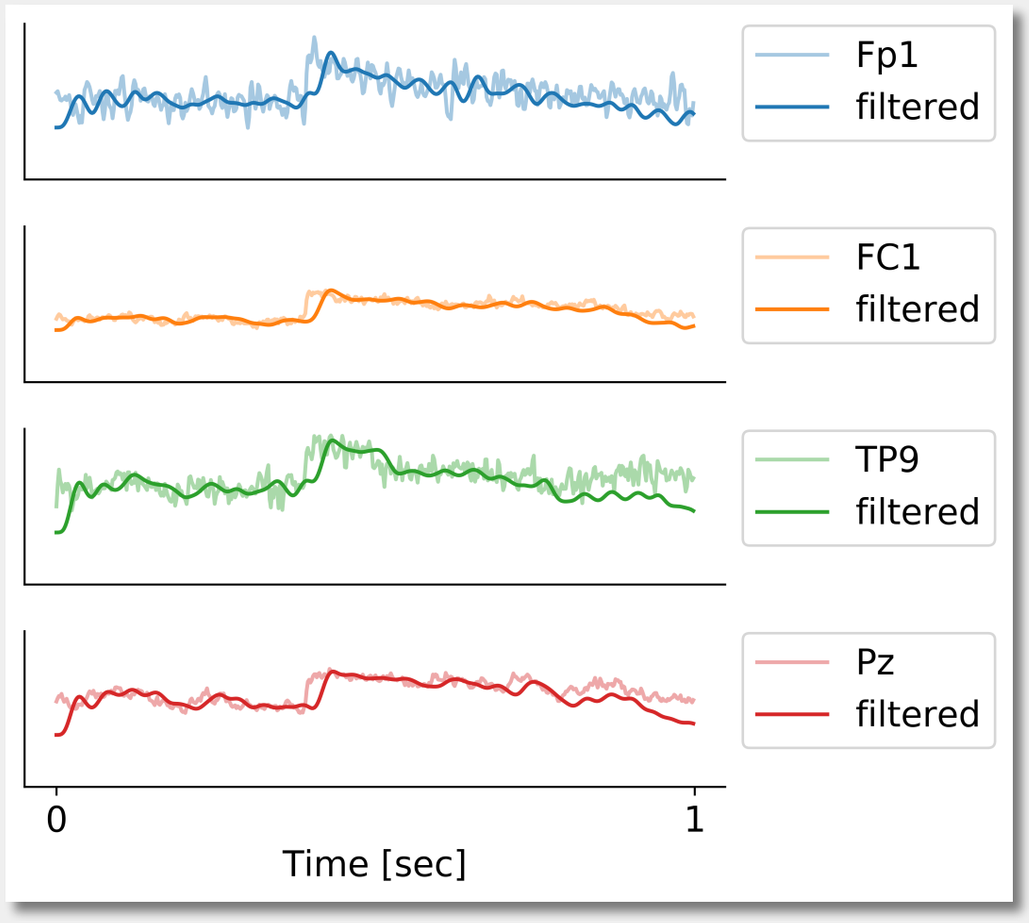


Power

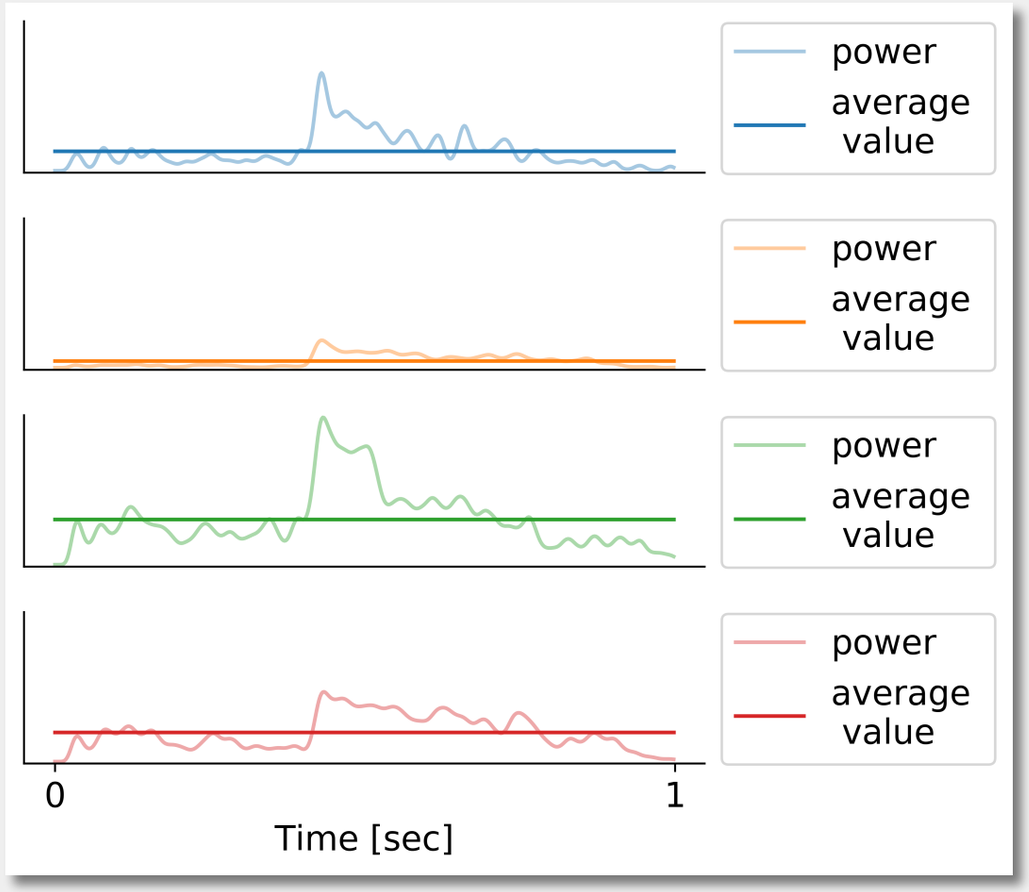


Use case 1: Preprocessing

Low-pass filter



Average Power



Use case 1: Model

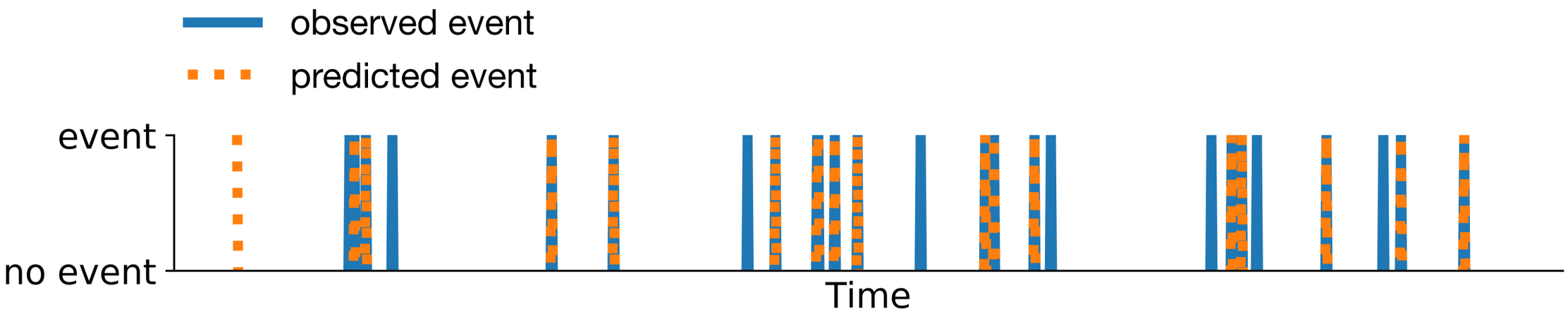
Model

```
lda = LDA()
rf = RandomForestClassifier(class_weight = 'balanced')
lr = LogisticRegression(class_weight = 'balanced')

eclf = VotingClassifier(estimators=[('lda', lda),
                                    ('rf', rf),
                                    ('lr', lr)],
                        voting = 'soft', weights=[1,1,1])
```

Prediction

```
eclf.fit(X_train, y_train)
y_pred = eclf.predict(X_test)
```

Use case 1: Prediction

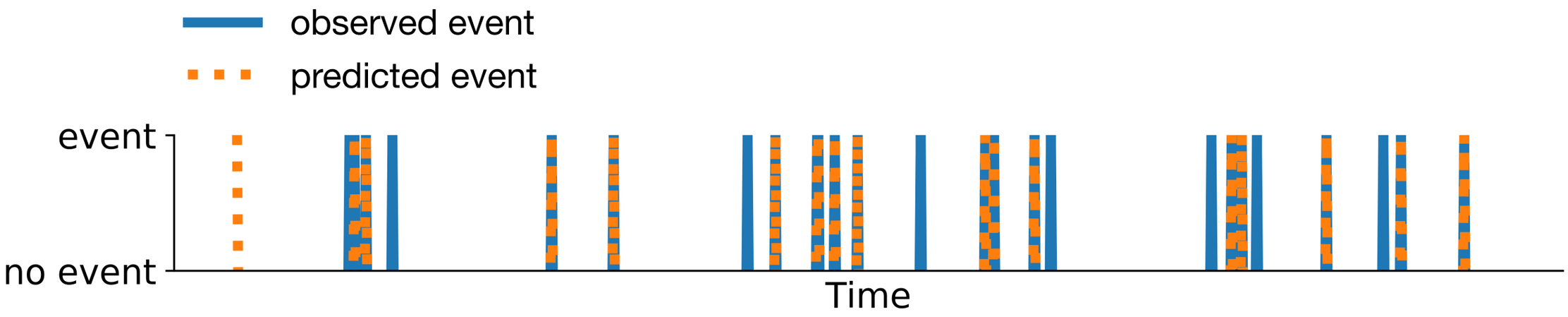


confusion matrix

	Predicted: No	Predicted: Yes
Actual: No	456263	113
Actual: Yes	3833	9016

- 70% of the events were correctly predicted

Use case 1: Prediction



confusion matrix

	Predicted: No	Predicted: Yes
Actual: No	456263	113
Actual: Yes	3833	9016

- 70% of the events were correctly predicted
- hardly any false alarm

Use case 1: Summary

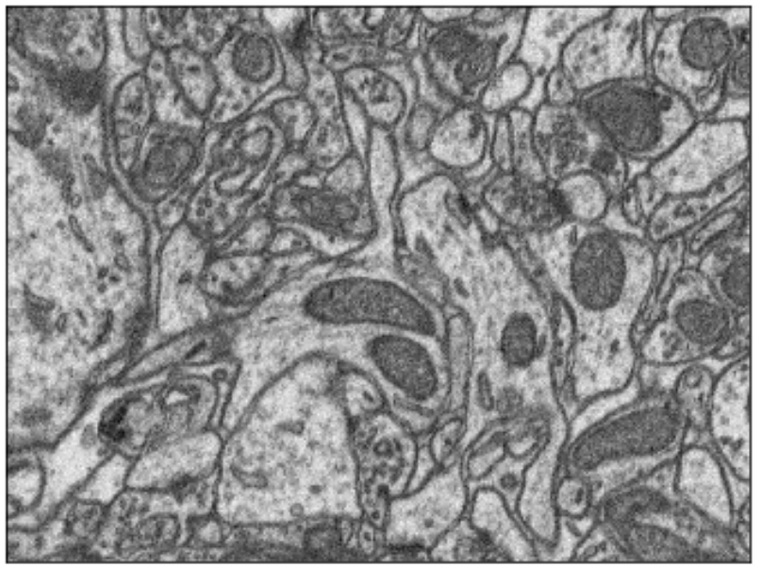
Classic ML model provides:

- a reasonably good prediction
- deeper insight into data due to interpretable models
- computational low costs (training: ~30m on single CPU)

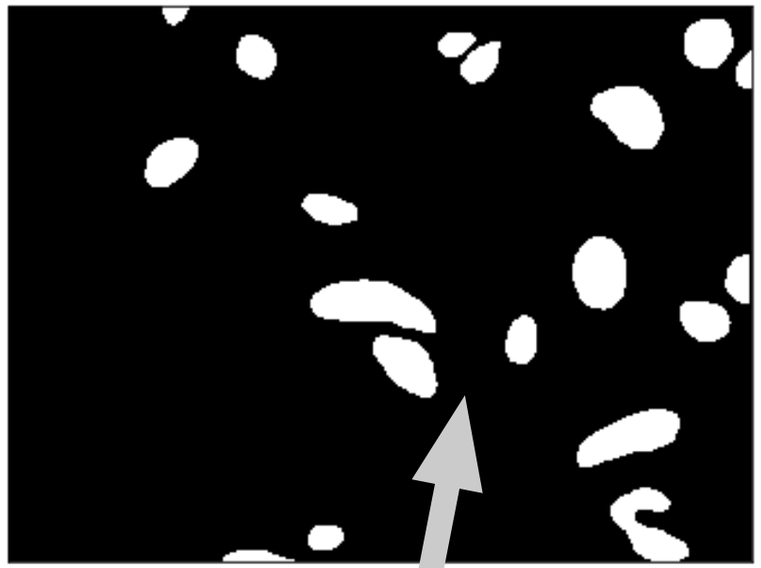
Use case 2: Segmentation

Use case 2: Data

Raw image

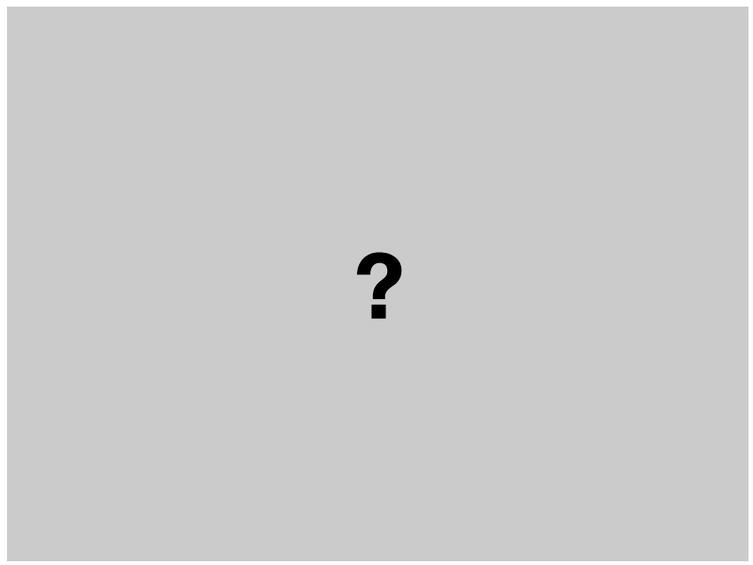


Segmentation



done by hand

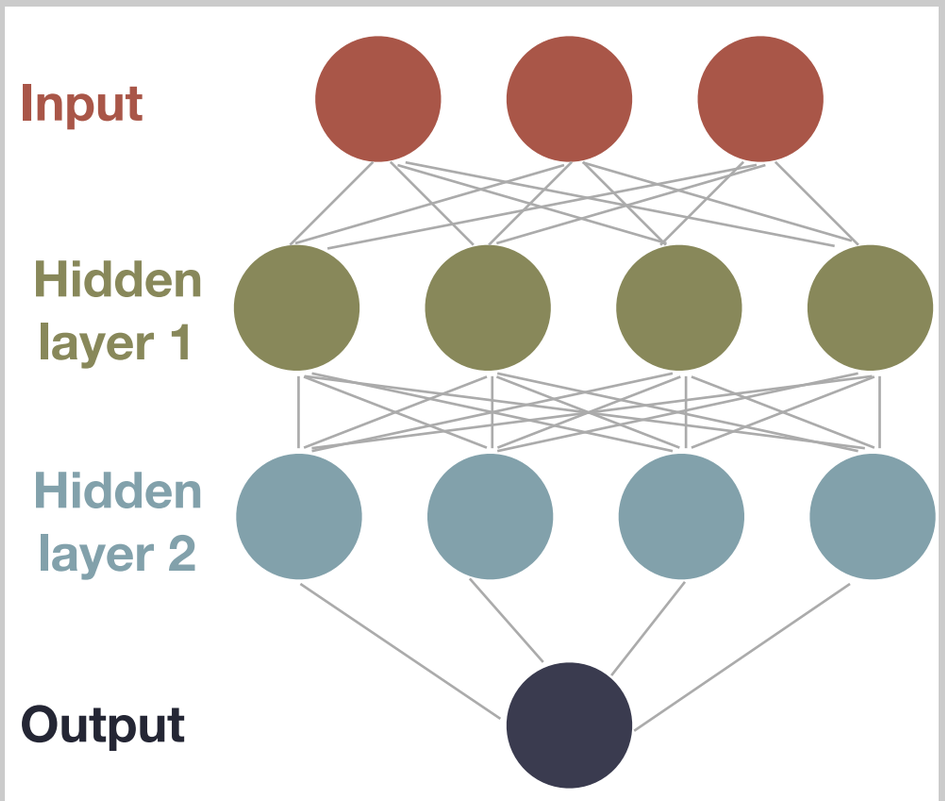
Automatic detection



Data acquired by: Graham Knott and Marco Cantoni at EPFL

Use case 2: Model

Neural Network

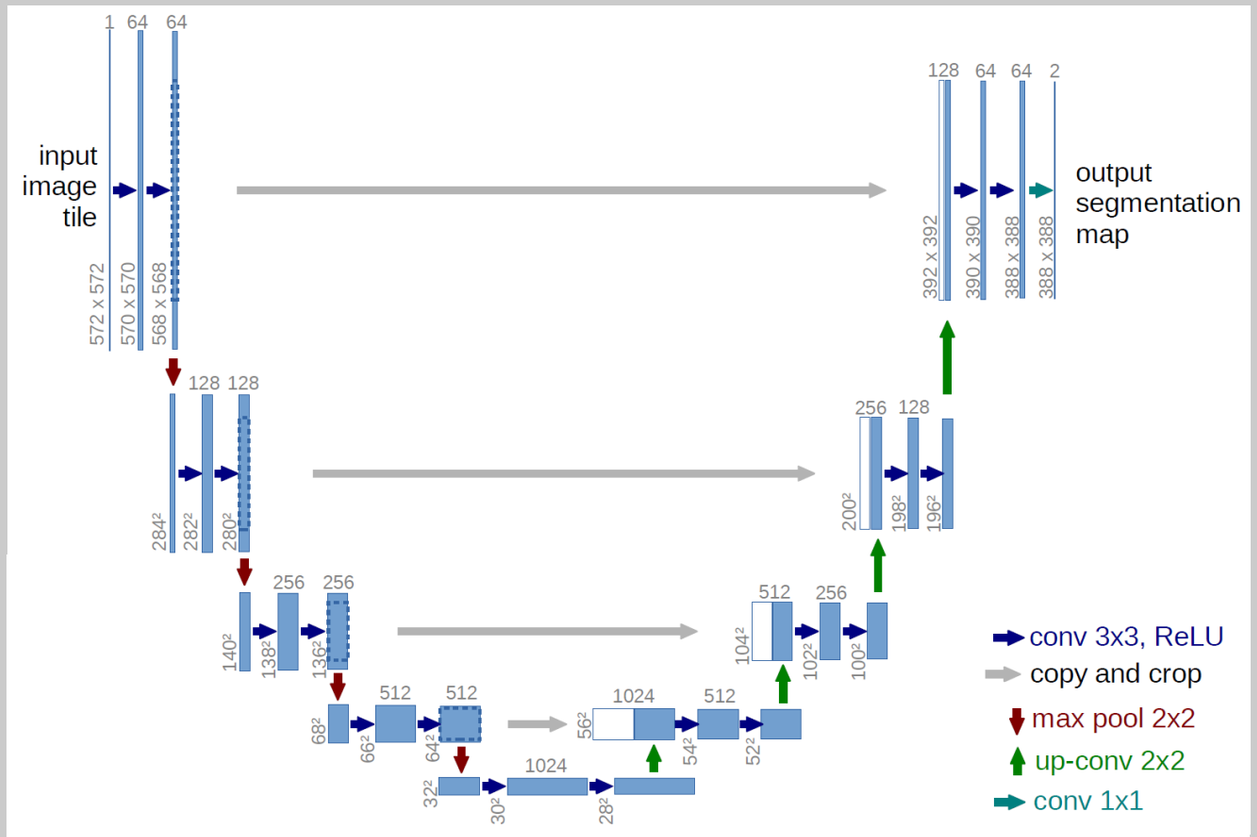


Implementation

```
from keras.models import Model
from keras.layers import Input, Dense

inp = Input(shape=(3,))


hidden_1 = Dense(4)(inp)


hidden_2 = Dense(4)(hidden_1)


outp = Dense(1)(hidden_2)

model = Model(inputs=inp, outputs=outp)
```

Use case 2: Model

U-Net



- **Downstream branch**: ‘what’-information
- **Upstream branch**: ‘where’-information

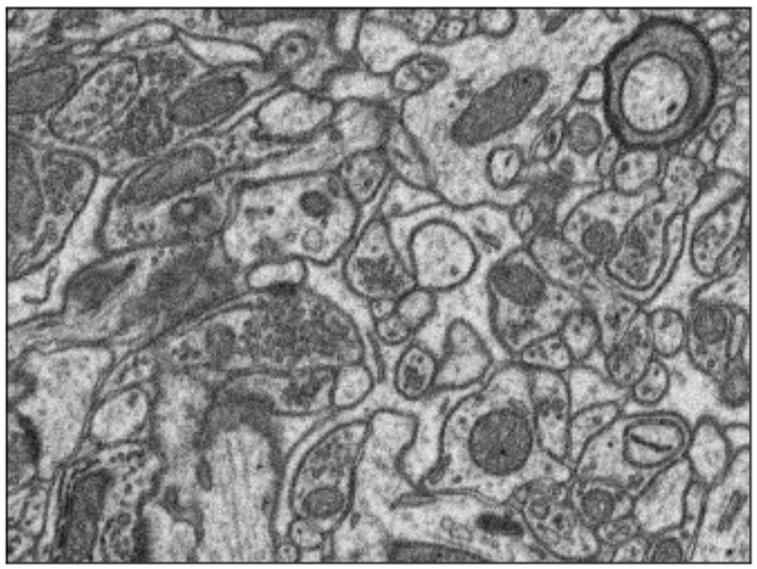
Implementation

```
from my_models import unet

model = unet()

model.fit(X_train, y_train)

results = model.predict(X_test)
```

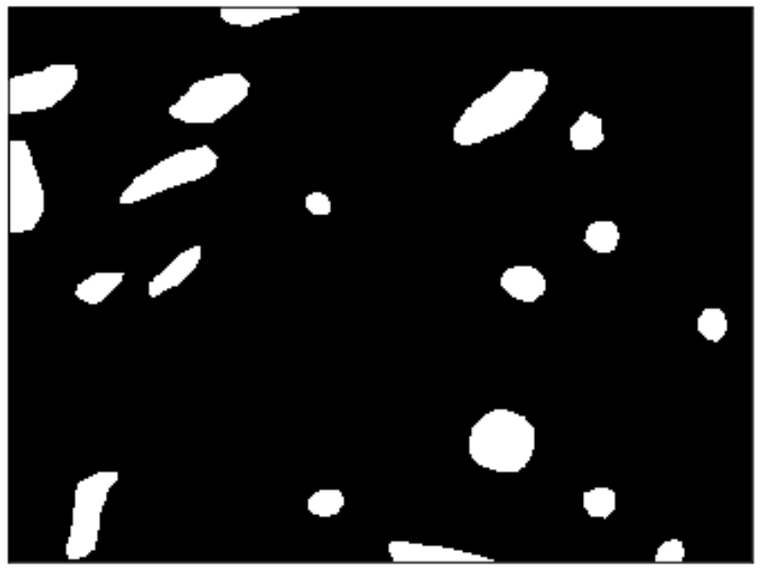
Ronneberger et al, MICCAI 2015

Use case 2: Prediction

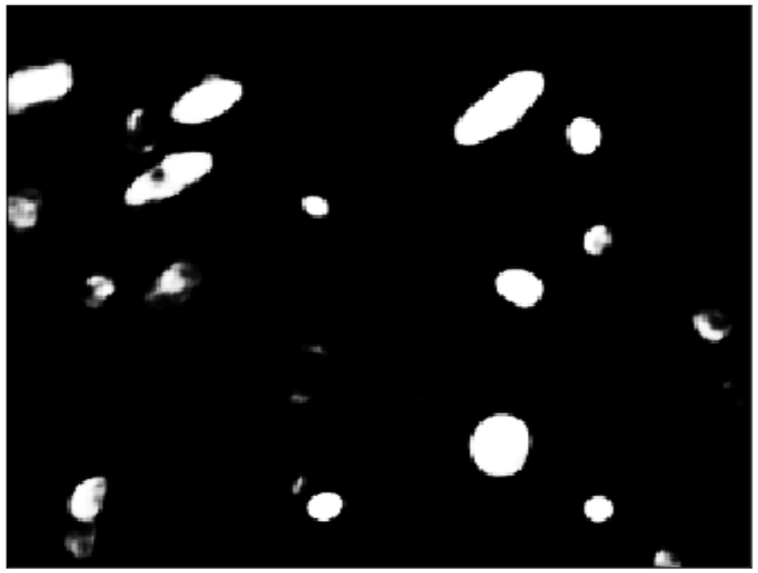
Raw image



Ground truth



Prediction



Use case 2: Summary

Deep learning model provides:

- automatisation of time-consuming process
- recognition of patterns in complex dataset
- no interpretability of model
- computationally heavy solution

  (Training: ~2h runtime on single GPU/~2d on single CPU)

Summary

Machine Learning in research:

- uncover hidden patterns in data
- interpretable models allow further insight
- automatisation of time-consuming processes

Thank you for your attention!