Ear Institute

ThAIMed Initiative

Introduction to time-series and AI for healthcare professionals

26th August 2024

Liam Barrett

Overview

1. Introduction to Time Series Data in Healthcare
2. Common Time Series Models
3. Preparing timeseries data for ML (Demo)
4. Introduction to more advanced models
5. Running models on clinical dataset (Demo)
6. Summary
7. Q&A

There are many sources of time-series data in medicine.
E.g., ECG, EEG, continuous glucose monitoring

↓

Here, we will be considering EEG data as an example

↓

The practical sessions follow real EEG data for seizure detection!
(Shah et al., 2018; Khalkhali et al., 2021)

Overview

1. Introduction to Time Series Data in Healthcare
2. Common Time Series Models
3. Preparing timeseries data for ML (Demo)
4. Introduction to more advanced models
5. Running models on clinical dataset (Demo)
6. Summary
7. Q&A

There are many sources of time-series data in medicine.
E.g., ECG, EEG, continuous glucose monitoring

Here, we will be considering EEG data as an example

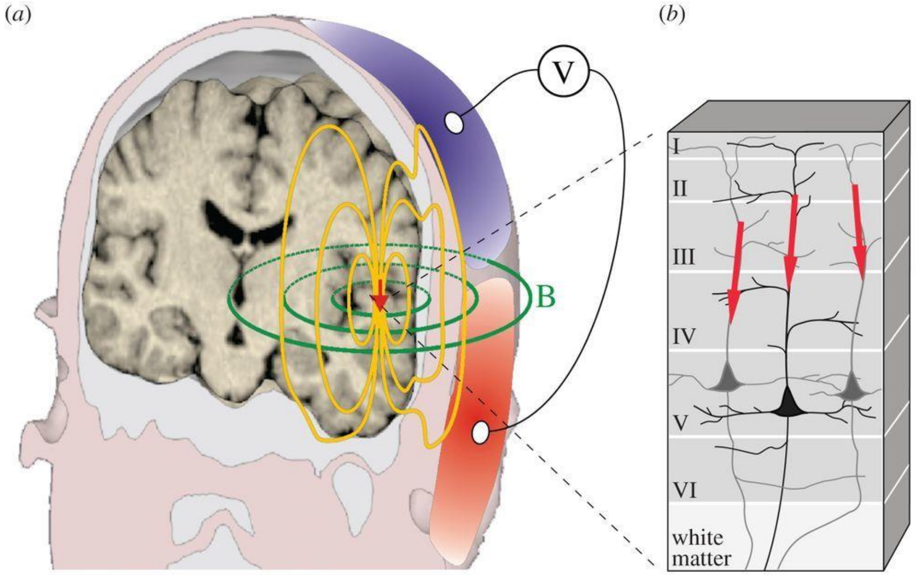
The practical sessions follow a real EEG data for seizure detection!
(Shah et al., 2018; Khalkhali et al., 2021)

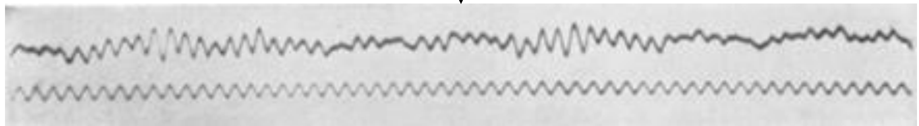
1. Introduction to Time Series Data in Healthcare

- Data structure
- Data Info
- Time-series as a signal
- Time and frequency
- Unique considerations of time-series

1.1 Data structure

- A time-series, as the name suggests, relates to data points of the same phenomenon collected over time
  - In EEG these are repeated measurements of the voltage fluctuations of electrodes placed on the scalp
- A time-series can be described by the measurements being made and the time intervals they are being made





1.1 Data structure

- Import data include but not limited to:
  - Meta Info
  - Time-stamps,
  - Indexes,
  - Raw data,
  - Annotations

Timestamp	Patient ID	Age	Gender	EEG Channel 1	EEG Channel 2	EEG Channel 3	Annotation
2023-01-01 09:00:00	5478	35	F	-23.45	12.78	-5.91	normal
2023-01-01 09:00:01	5478	35	F	-18.62	15.34	-3.05	normal
2023-01-01 09:00:02	5478	35	F	-25.73	11.56	-7.22	normal
2023-01-01 09:00:03	5478	35	F	-30.19	8.87	-9.64	normal
2023-01-01 09:00:04	5478	35	F	-28.83	10.45	-8.79	normal
2023-01-01 10:15:00	3692	52	M	42.19	-26.87	13.64	normal
2023-01-01 10:15:01	3692	52	M	45.76	-28.92	15.38	normal
2023-01-01 10:15:02	3692	52	M	51.38	-32.15	17.93	seizure
2023-01-01 10:15:03	3692	52	M	58.86	-35.57	20.21	seizure
2023-01-01 10:15:04	3692	52	M	53.27	-30.69	18.42	seizure

1.2 Data Info

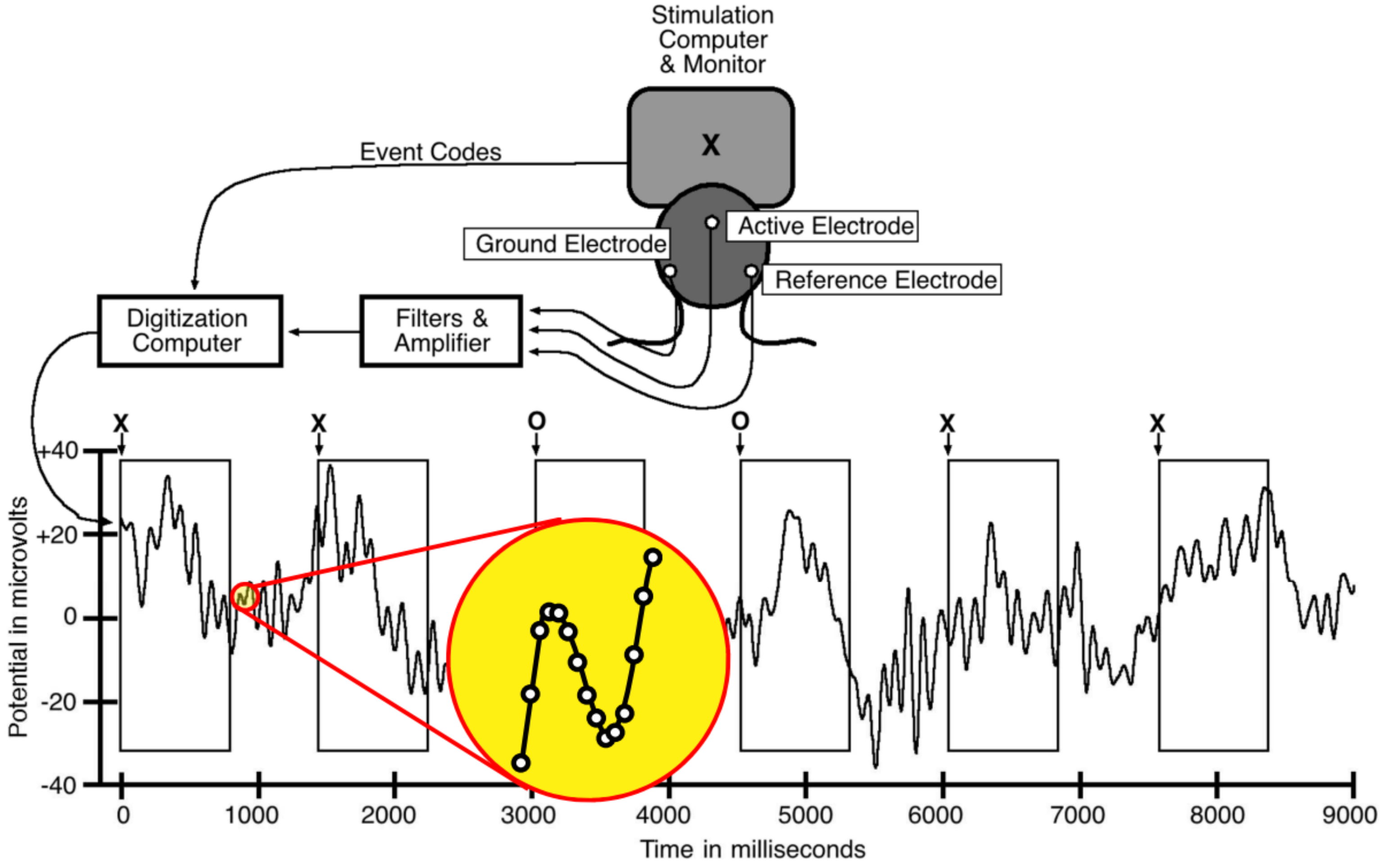
- Important meta data
  - Patient information and ID
    - Patient number
    - Clinical info of interest (Control, Epileptic, etc.)
  - Technical/Collection information
    - Sampling
    - Rate
    - Duration

Timestamp	Patient ID	Age	Gender	EEG Channel 1	EEG Channel 2	EEG Channel 3	Annotation
2023-01-01 09:00:00	5478	35	F	-23.45	12.78	-5.91	normal
2023-01-01 09:00:01	5478	35	F	-18.62	15.34	-3.05	normal
2023-01-01 09:00:02	5478	35	F	-25.73	11.56	-7.22	normal
2023-01-01 09:00:03	5478	35	F	-30.19	8.87	-9.64	normal
2023-01-01 09:00:04	5478	35	F	-28.83	10.45	-8.79	normal
2023-01-01 10:15:00	3692	52	M	42.19	-26.87	13.64	normal
2023-01-01 10:15:01	3692	52	M	45.76	-28.92	15.38	normal
2023-01-01 10:15:02	3692	52	M	51.38	-32.15	17.93	seizure
2023-01-01 10:15:03	3692	52	M	58.86	-35.57	20.21	seizure
2023-01-01 10:15:04	3692	52	M	53.27	-30.69	18.42	seizure

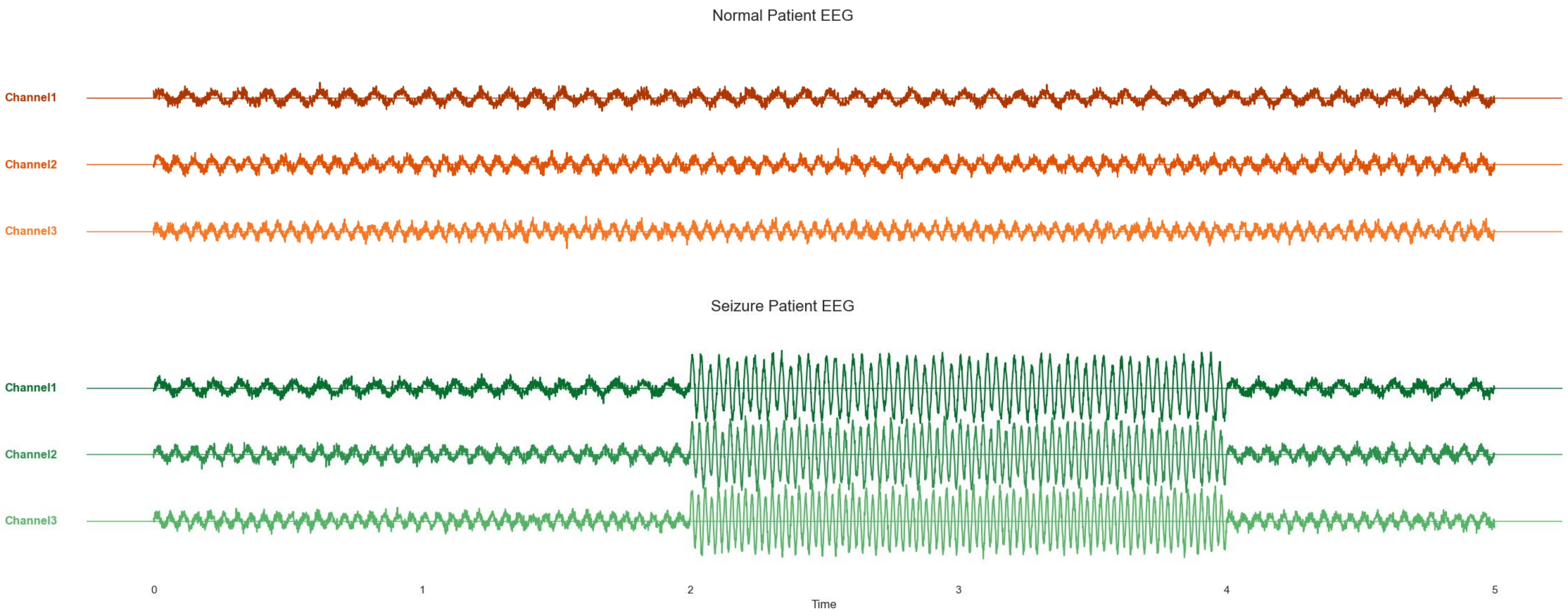
1.3 Time series as a signal

- Data of this type can be considered with respect to changes over time
- This progression can be thought of as a signal generated by a system and recorded by our measurements
  - In EEG these signals are voltages recorded by electrodes and generated by the brain (amongst other systems)
- The generative process of these signals are continuous: the brain is always generating electromagnetic signals
- For computational purposes we discretise this continuous process into individual samples

1.3 Time series as a signal



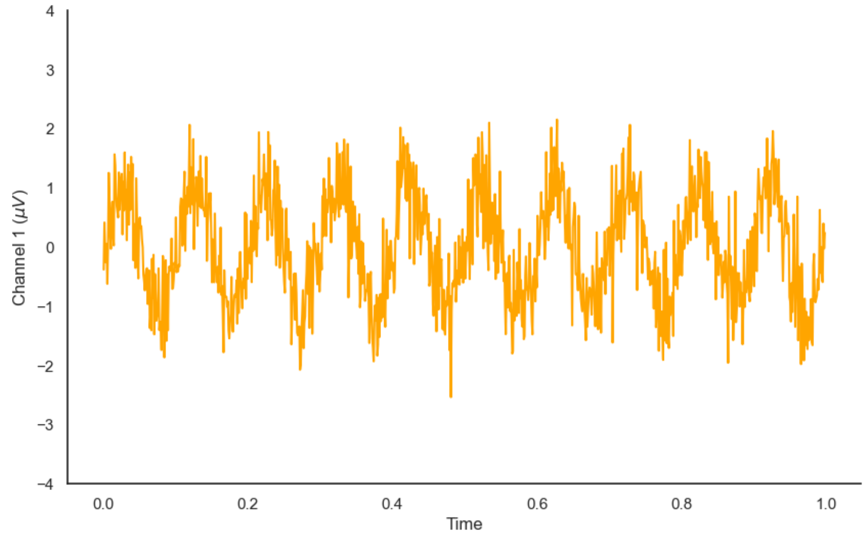
1.3 Time series as a signal



1.4 Time and frequency

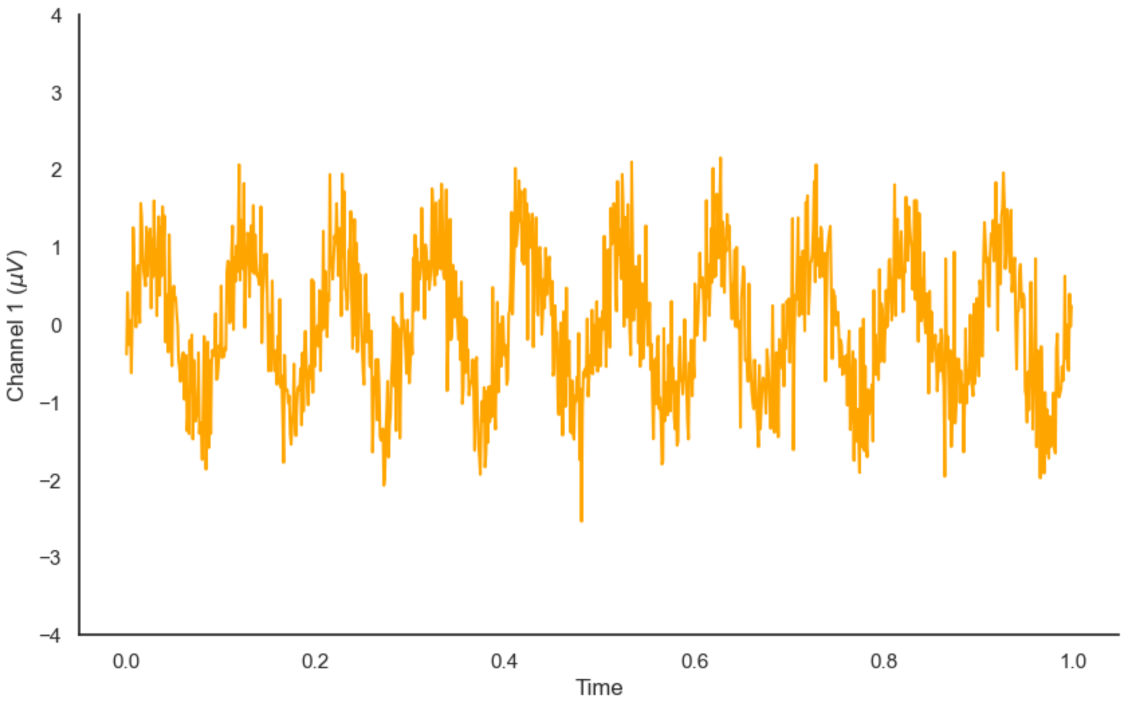
- Thus far we have only considered the data as it progresses over time and changes in level measurements
- However, a signal can also be described by the frequency components that compose the signal
  - This is particularly clear in highly periodic signals

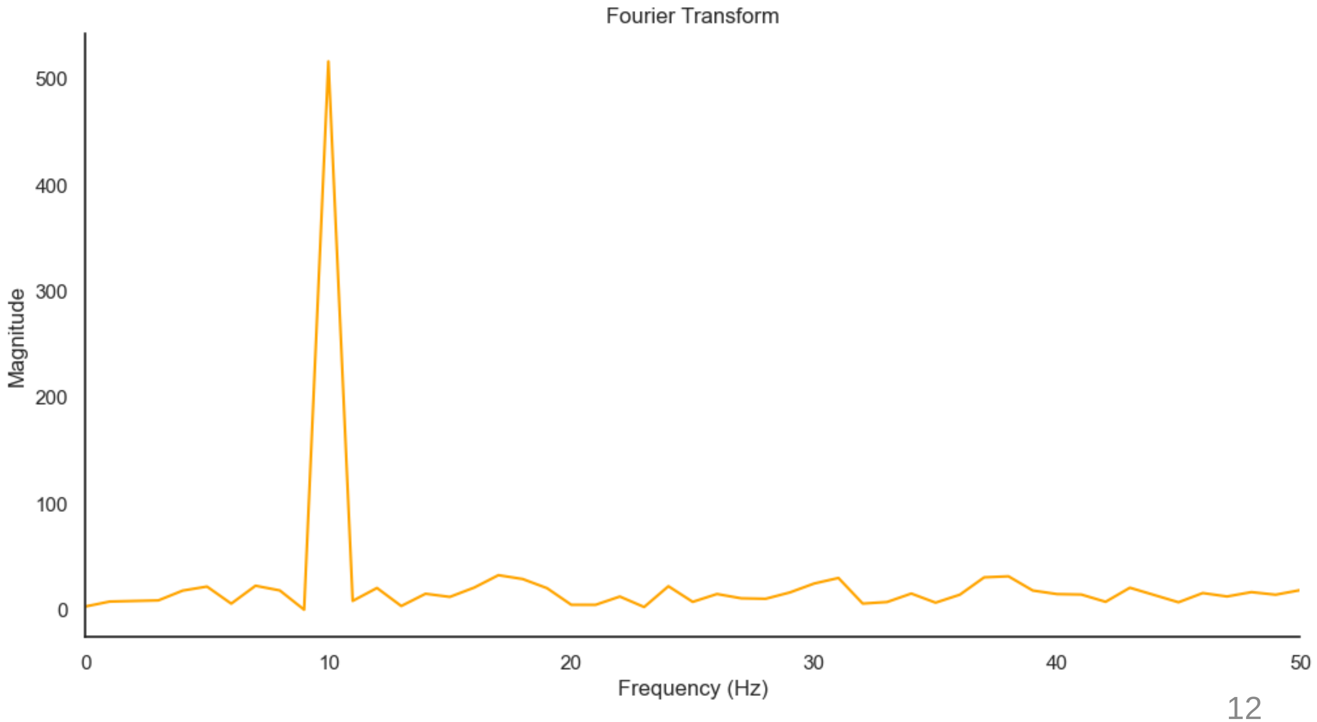
	Time	Channel1
0	0.000	-0.377868
1	0.001	0.418195
2	0.002	-0.009430
3	0.003	-0.256892
4	0.004	0.059624



1.4 Time and frequency

- Thus far we have only considered the data as it progresses over time and changes in level measurements





1.5 Breaking this into regions of interest

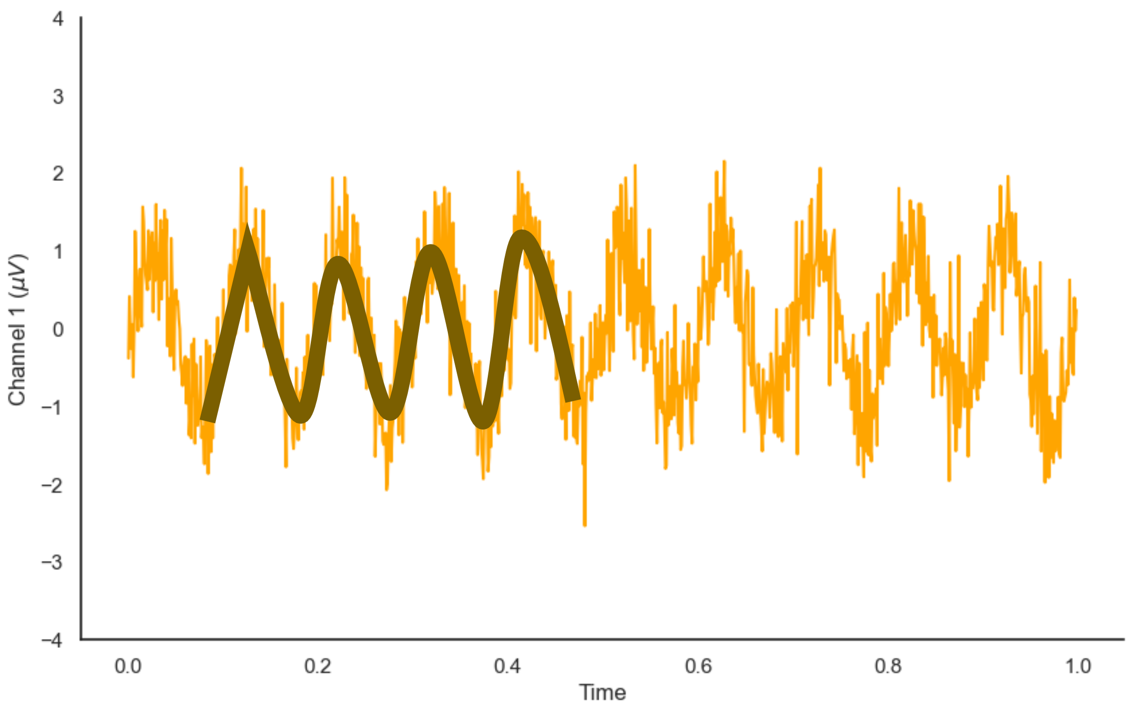
Where to start with a time-series?

- Trends
- Biomarkers

1.5 Breaking this into regions of interest

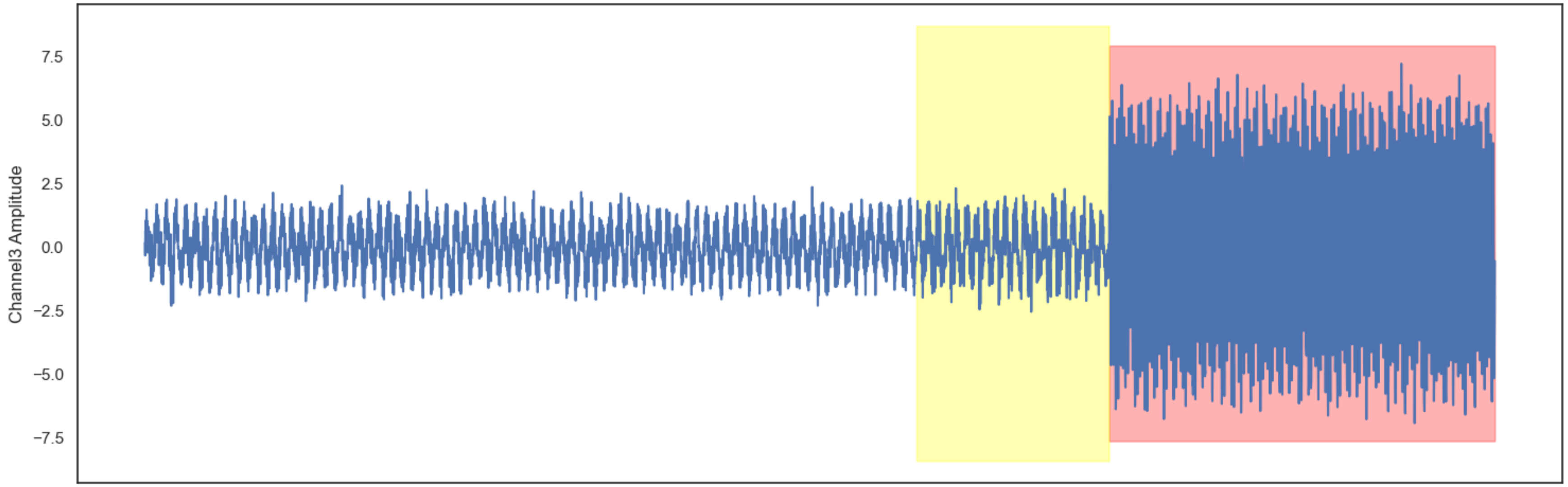
Trends

- Long-term progression or pattern in the data
- Can be:
  - Linear (steady increase or decrease)
  - Non-linear (more complex patterns)



1.5 Breaking this into regions of interest

Biomarkers



1.5 Breaking this into regions of interest

Biomarkers

- Identification tends to be post-hoc
- Can lead to:
  - Spurious associations
  - Misidentification of causal factors

Overview

1. Introduction to Time Series Data in Healthcare
2. Common Time Series Models
3. Preparing timeseries data for ML (Demo)
4. Introduction to more advanced models
5. Running models on clinical dataset (Demo)
6. Summary
7. Q&A

2. Common Time Series Models

- Basic statistical methods like moving averages and exponential smoothing
- Discuss autoregressive (AR) models and their relevance to EEG data analysis
- Consider Deep Learning approaches (Concepts)

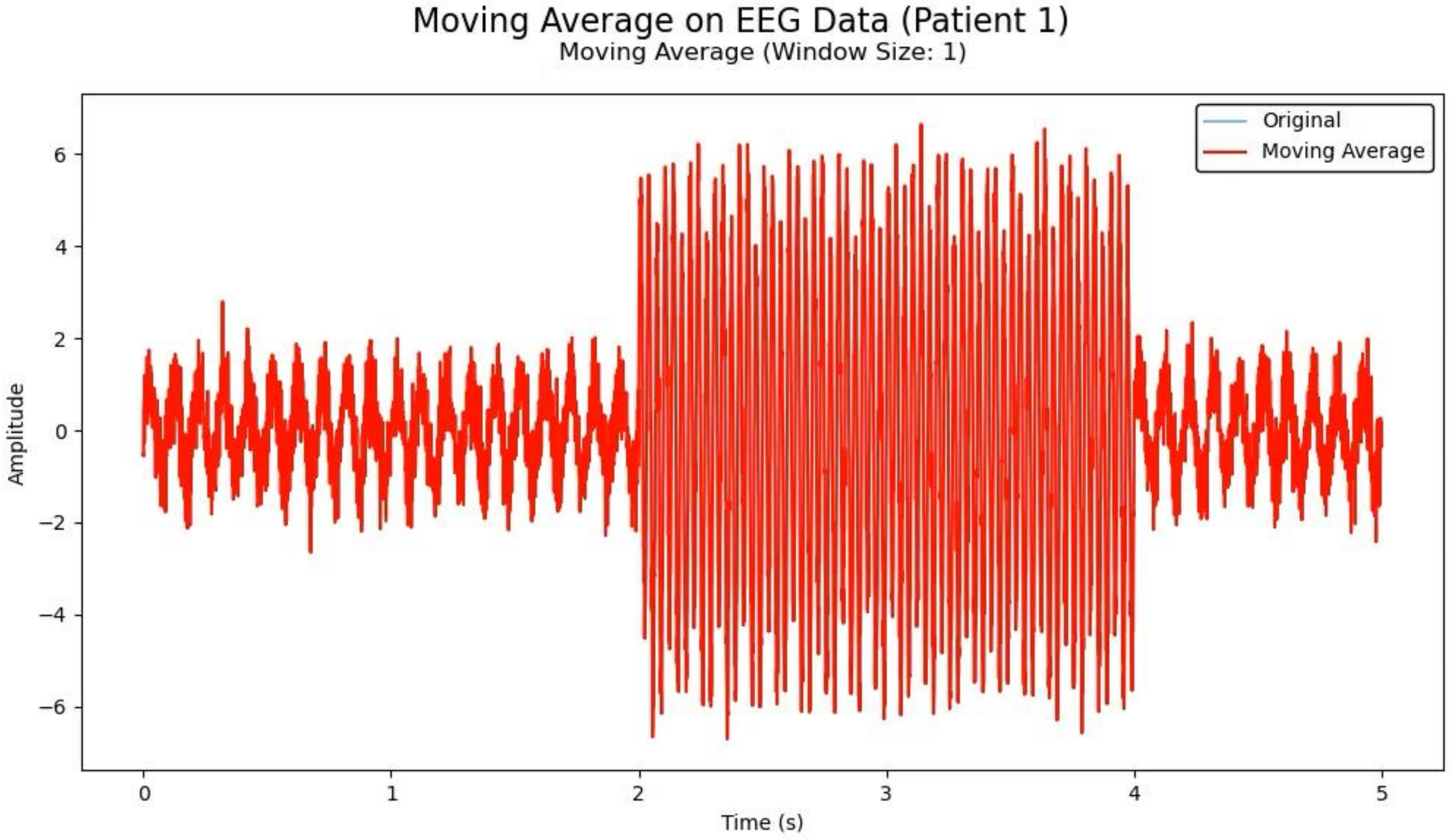
2.1 Basic Statistical Methods

Moving Averages (MA)

- Simple technique for smoothing time series data
- Calculates the average of a fixed number of data points
- Useful for reducing noise in (EEG) signals
- Can be weighted to increase importance of recent data

2.1 Basic Statistical Methods

Moving Averages



2.1 Basic Statistical Methods

Exponential Smoothing (ES)

- More sophisticated smoothing technique
- Assigns exponentially decreasing weights to older observations

Applications:

- Forecasting future values
- Detecting gradual changes

2.1 Basic Statistical Methods

Types of ES

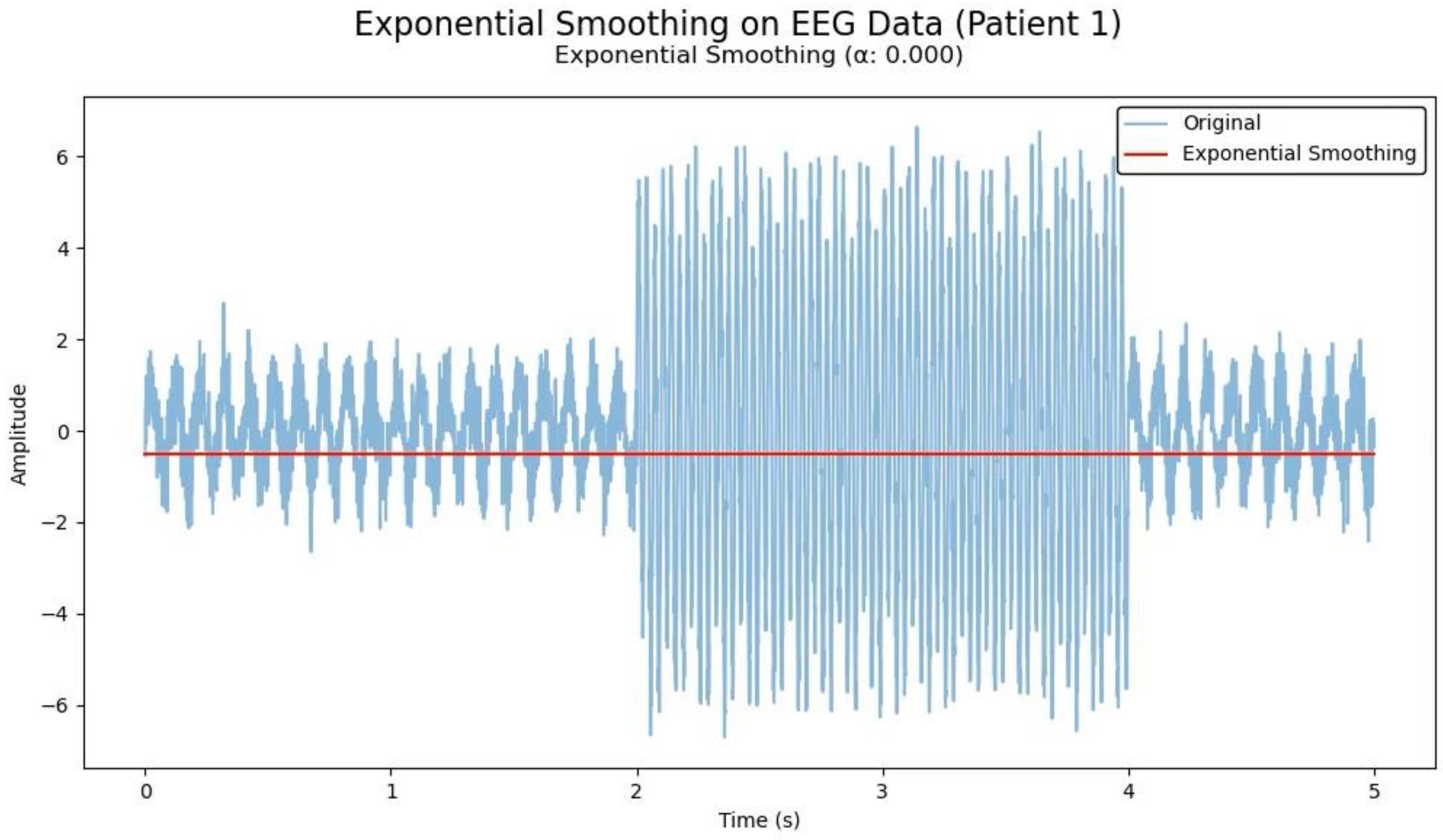
- Simple Exponential Smoothing (SES)
    - For data with no clear trend or seasonality

- Double Exponential Smoothing (Holt's method)
    - For data with a trend

- Triple Exponential Smoothing (Holt-Winters' method)
    - For data with both trend and seasonality

2.1 Basic Statistical Methods

Exponential Smoothing



2.2 Autoregressive models

What are Autoregressive Models?

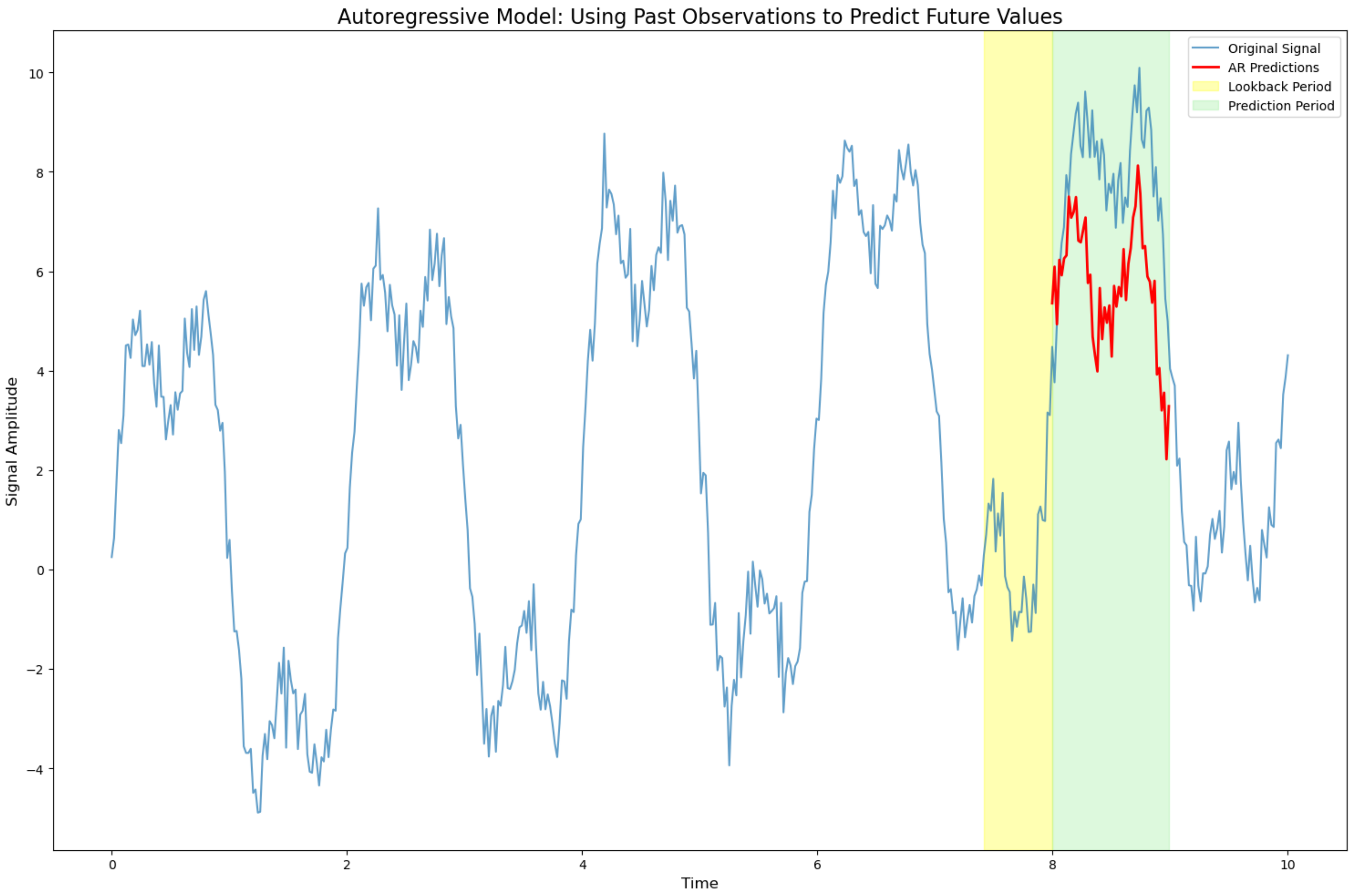
- Time series models that predict future values based on past observations
- "Auto" refers to using the variable's own history for predictions
- "Regressive" indicates the model is based on a regression of past values

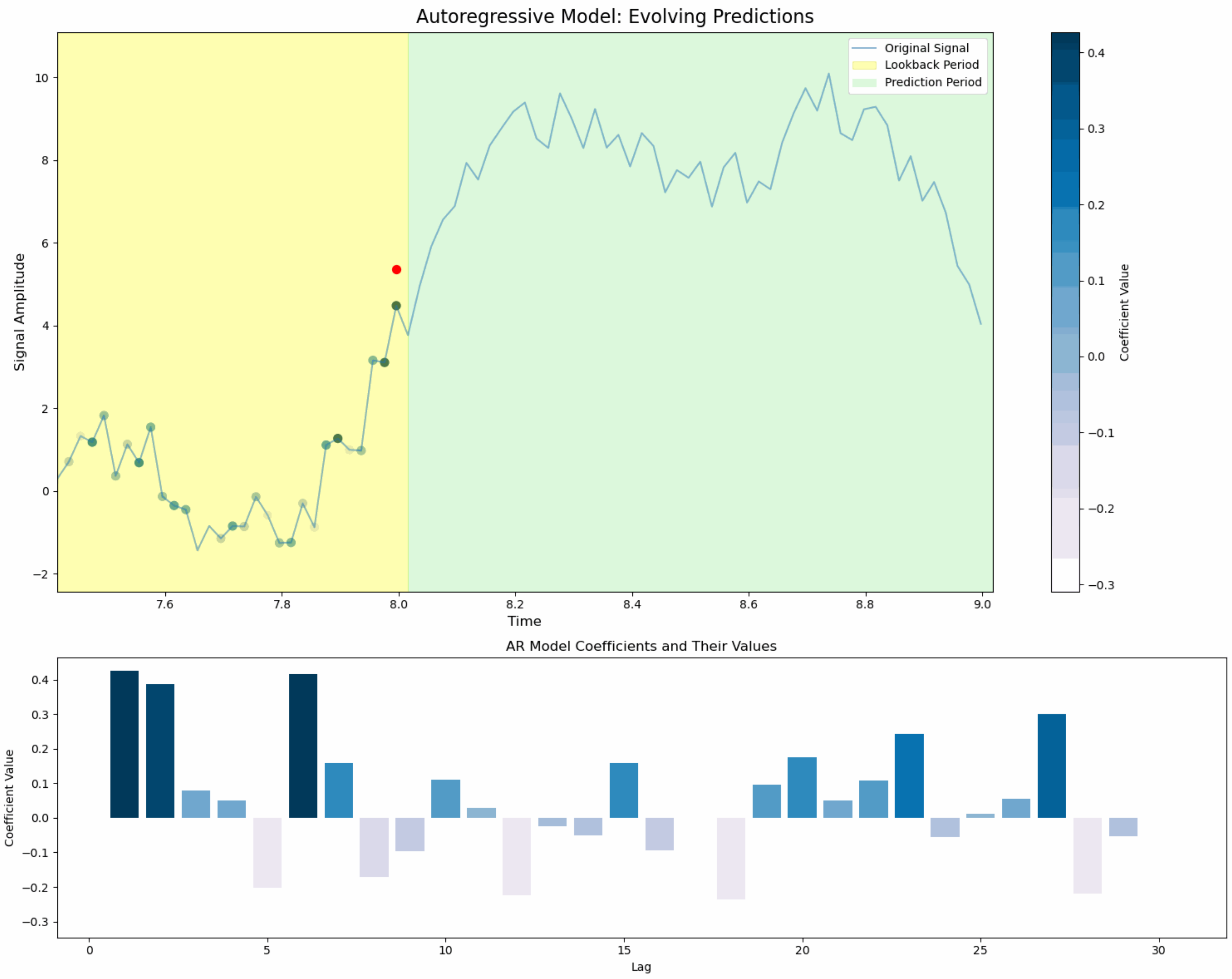
2.2 Autoregressive models

Model specification

- Order
- Lags
- Coefficients
- Stationarity

2.2 Autoregressive models





Autoregressive Model: Evolving Predictions
Original Signal
Lookback Period
Prediction Period
Signal Amplitude
Time
Coefficient Value
AR Model Coefficients and Their Values
Coefficient Value
Lag

2.1 Autoregressive models

Key concepts

1. Pattern Recognition
2. Memory of the Signal
3. Weighted Influence
4. Noise Consideration
5. Customizable Complexity

2.1 Autoregressive models

Limits and Considerations

1. Stationarity Assumption

2. Model Order Selection

3. Non-linear Dynamics

Autoregression is a powerful technique that gets to the main of time-series modelling – but can we do better through Neural Networks?

2.1 Autoregressive models

Conclusions

- Simple, effective and intuitive model of time-series data

- Has been utilised successfully across a range of medical data

    - Forecasting Symptom Complexity
    - Predicting response patterns in rare diseases
    - Evaluating health interventions themselves

- A good model to use as baseline

Overview

1. Introduction to Time Series Data in Healthcare
2. Common Time Series Models
3. Preparing timeseries data for ML (Demo)
4. Introduction to more advanced models
5. Running models on clinical dataset (Demo)
6. Summary
7. Q&A

3. Preparing timeseries data for ML (Demo)

- Import data
- Visualise data
- Perform basic manipulations
- Train AR model

3. Preparing timeseries data for ML (Demo)

Code behind the scenes:

```python
# Import necessary libraries

import numpy as np                                          # for data manipulation
import matplotlib.pyplot as plt                             # for data visualisation
import mne                                                  # for EEG data handling
from statsmodels.tsa.ar_model import AutoReg                # for auto-regressive modelling
import tensorflow as tf                                     # for deep learning
```

Code behind the scenes:

```python
def preprocess_eeg(raw):
    """
    Preprocess the EEG data: filter and remove bad channels.

    Parameters:
    raw (mne.io.Raw): MNE Raw object containing EEG data

    Returns:
    mne.io.Raw: Preprocessed MNE Raw object
    """
    # Apply a bandpass filter
    raw.filter(l_freq=1, h_freq=40)

    # Detect and interpolate bad channels
    raw.interpolate_bads(reset_bads=True)

    return raw
```

Code behind the scenes:

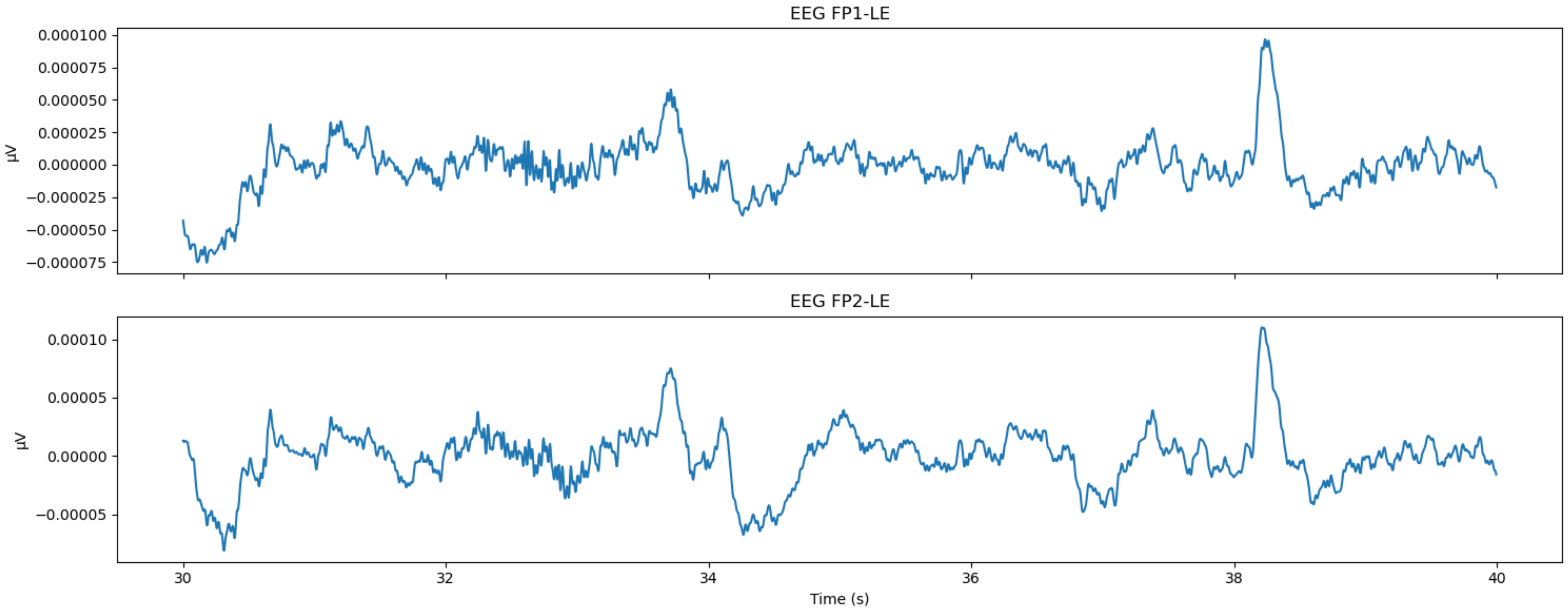
```python
def plot_eeg_channels(raw, n_channels=5, duration=10, start=0):
    """
    Plot the first n_channels of the EEG data.

    Parameters:
    raw (mne.io.Raw): MNE Raw object containing EEG data
    n_channels (int): Number of channels to plot
    duration (float): Duration of the data to plot in seconds
    start (float): Start time for plotting in seconds
    """
    data, times = raw[:n_channels, int(start*raw.info['sfreq']):int((start+duration)*raw.info['sfreq'])]

    fig, axs = plt.subplots(n_channels, 1, figsize=(15, 3*n_channels), sharex=True)
    for i, (ax, ch_name) in enumerate(zip(axs, raw.ch_names[:n_channels])):
        ax.plot(times, data[i])
        ax.set_title(ch_name)
        ax.set_ylabel('µV')

    axs[-1].set_xlabel('Time (s)')
    plt.tight_layout()
    plt.show()
```

```python
# Import example EEG data
file_path = 'path/to/your/eeg_file.edf'
raw = mne.io.read_raw_edf(file_path, preload=True)

# Print information about the imported data
print(raw.info)
```

<Info

ch_names: EEG FP1-LE, EEG FP2-LE, ...
chs: 33 EEG
highpass: 0.0 Hz
lowpass: 125.0 Hz
sfreq: 250.0 Hz
subject_info: 3 items (dict)
>

```python
# Plot the first 5 channels of the preprocessed data
plot_eeg_channels(raw_preprocessed, n_channels=2, duration=10, start=30)
```



3. Preparing timeseries data for ML (Demo)

- Import data
- Visualise data
- Perform basic manipulations
- Train AR model

Break

- Any questions?
- Or Clarifications?

Overview

1. Introduction to Time Series Data in Healthcare
2. Common Time Series Models
3. Preparing timeseries data for ML (practical)
4. Introduction to more advanced models
5. Running models on clinical dataset (practical)
6. Summary
7. Q&A

4. Introduction to more advanced models

- Considered AR models – already complex!
- Simple neural networks
- Neural networks for time-series
- Applications in healthcare

4.1 AR models - recap

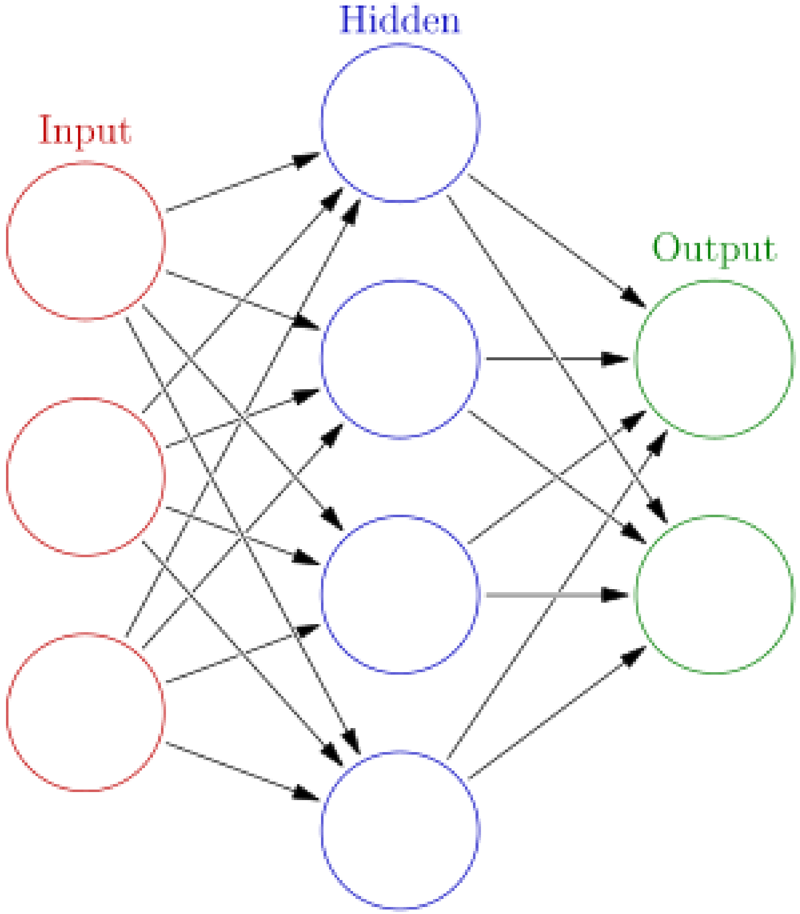
Core principle:

Predict future values based on past observations

- Key features
- Strengths
- Limitations

4.2 Neural Networks

- Learns hierarchical representations through multiple layers of non-linear transformations
- Uses backpropagation to compute gradients of the loss function with respect to weights
- Best uses are:
    - For large, complex datasets
    - When dealing with unstructured data (images, text, audio)
    - Interpretability is less critical

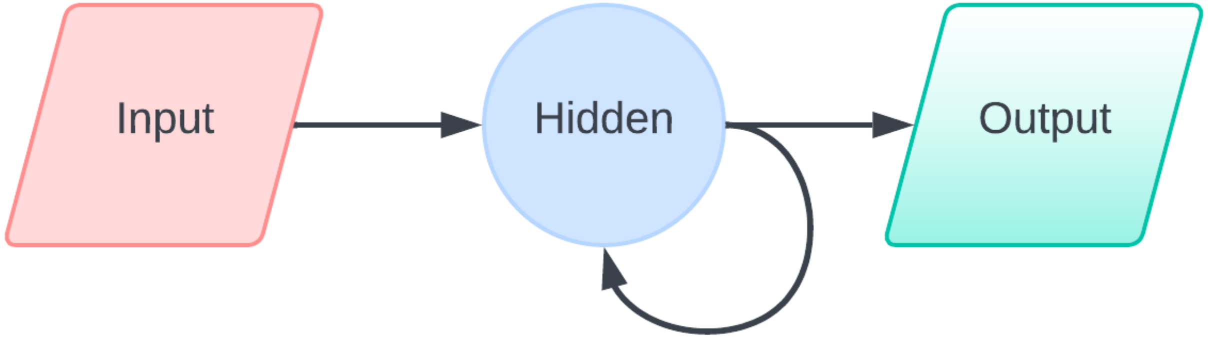


4.3 Recurrent Neural Networks

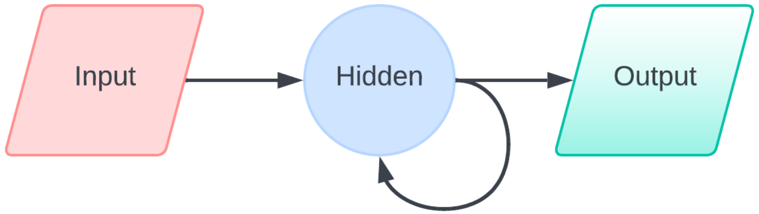
Key feature:

Recurrent connections allow information to persist

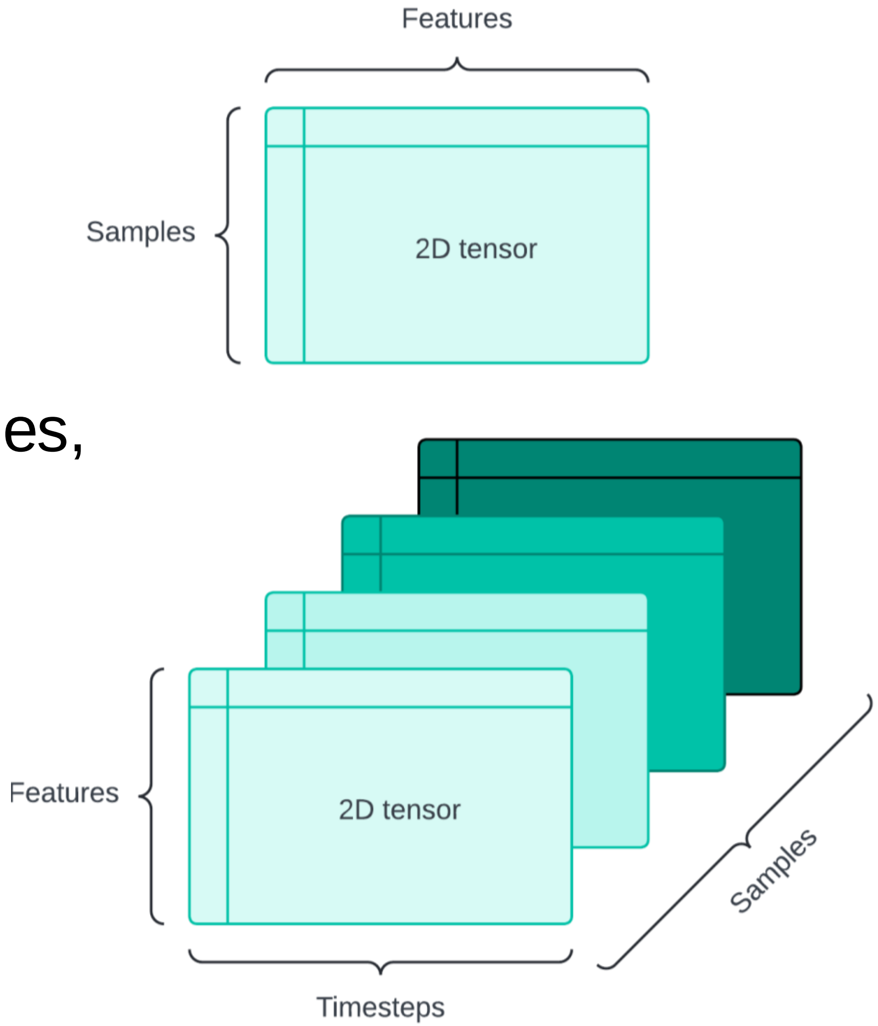
- Designed for sequential data
- Forms an ‘internal memory’
- Limited in long-term dependencies



4.3 Recurrent Neural Networks

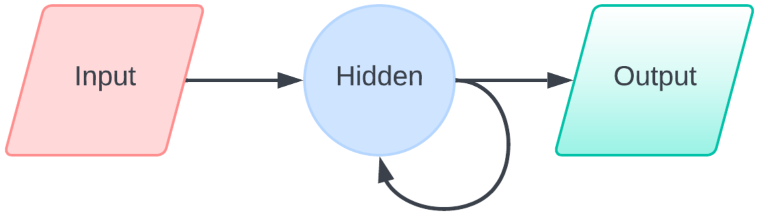


Data flow



- Vector data — 2D tensors of shape (samples, features)
- Timeseries data — 3D tensors of shape (samples, timesteps, features)
  - The time axis is usually the second axis by convention

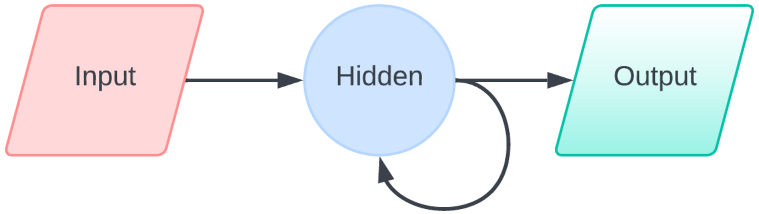
4.3 Recurrent Neural Networks



Data flow

- The time-steps, as in AR, is decided by the researcher and may reflect some priors about the disease/event progression
- The samples may be windows of data within a continuous recording or even separate recordings

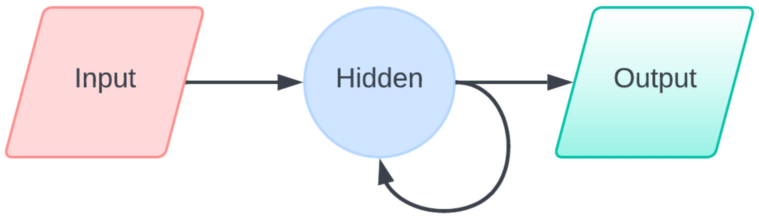
4.3 Recurrent Neural Networks



Applications in time series and healthcare

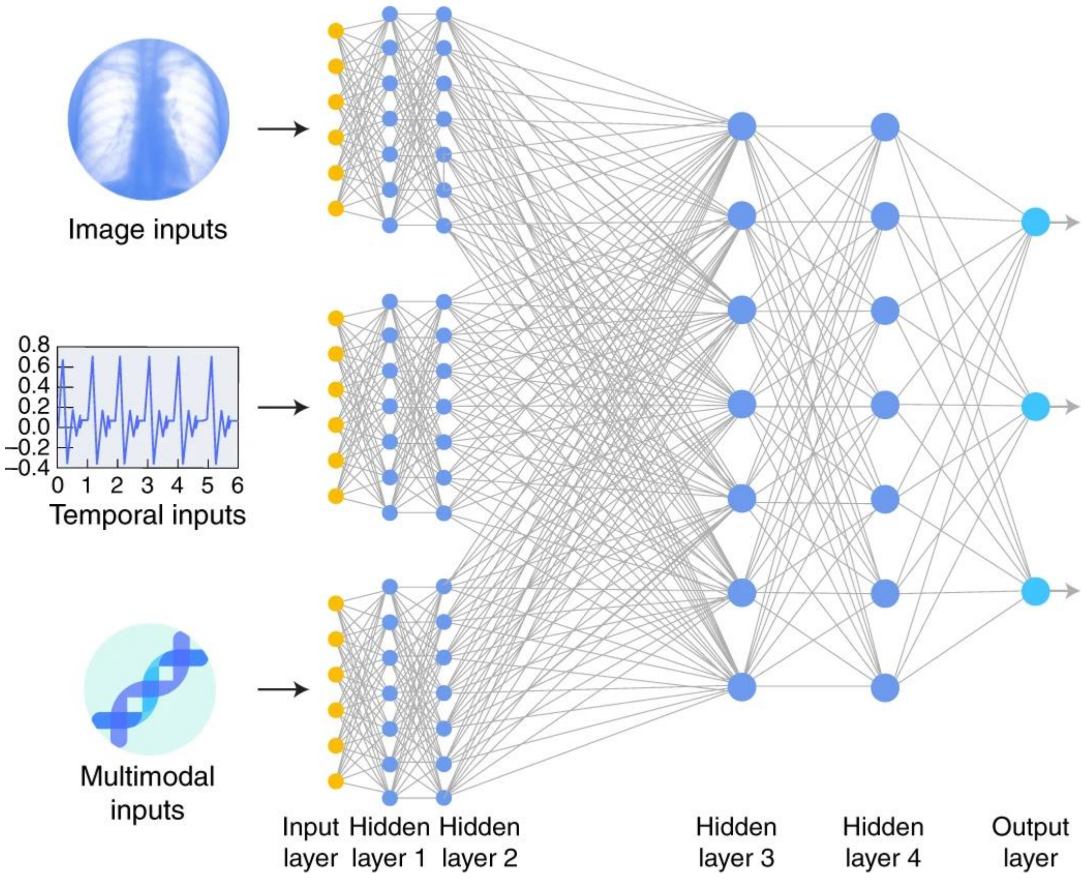
- Outcomes in COVID-19
- Prediction of medical events from EHR's
- Epileptic seizure prediction
- Many more...

4.3 Recurrent Neural Networks

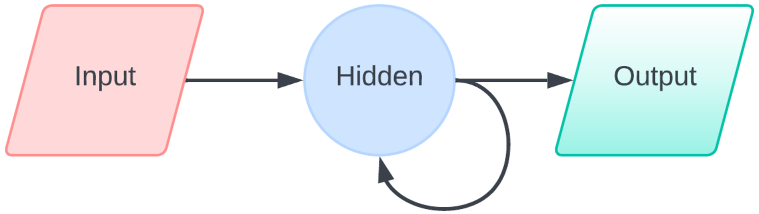


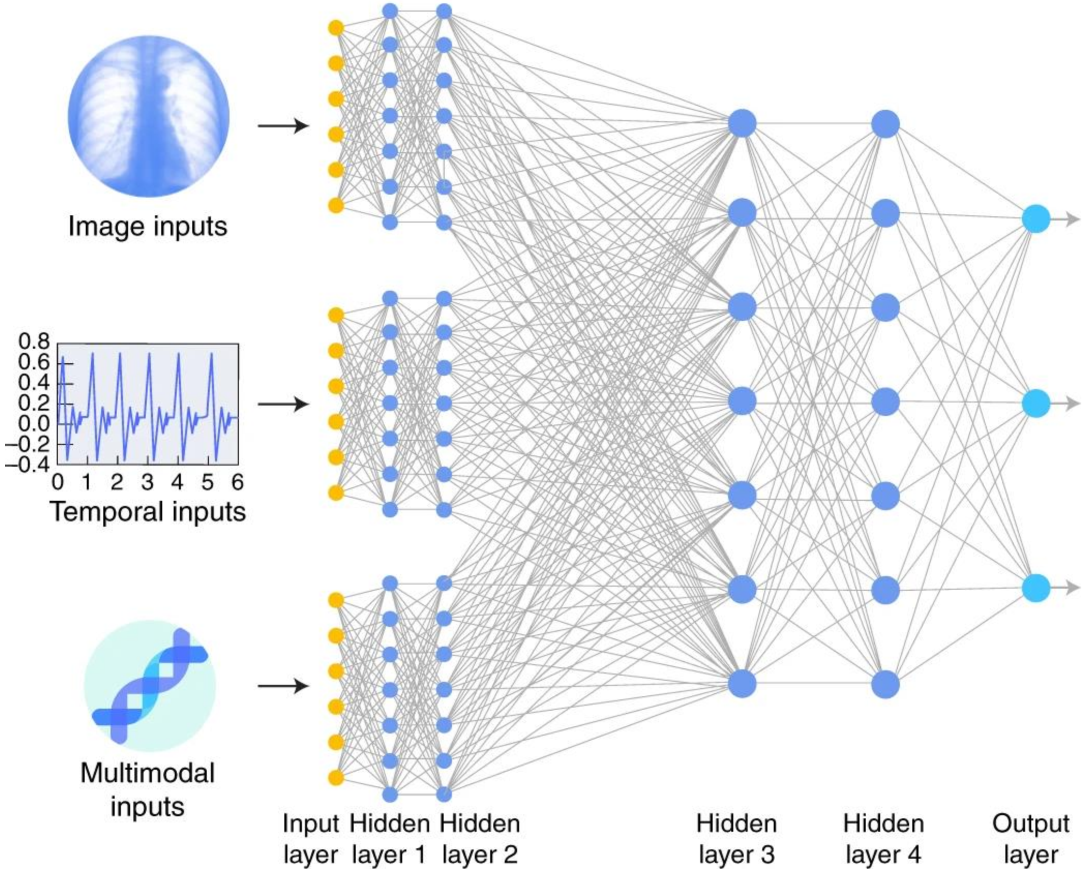
Considerations in healthcare

- Interpretability
- Robustness
- Real-time processing
- Personalization
- Multi-modal integration



4.3 Recurrent Neural Networks





Outputs depend on objective

- Forecasting
- Event detection
- Classification
- Sequence-2-sequence

Overview

1. Introduction to Time Series Data in Healthcare
2. Common Time Series Models
3. Preparing timeseries data for ML (Demo)
4. Introduction to more advanced models
5. Running models on clinical dataset (Demo)
6. Summary
7. Q&A

5. Running models on clinical dataset (Demo)

- Import data, etc.
- Train AR model
- Visualise AR Model
- Train RNN
- Visualise RNN Model

5. Running models on clinical dataset (Demo)

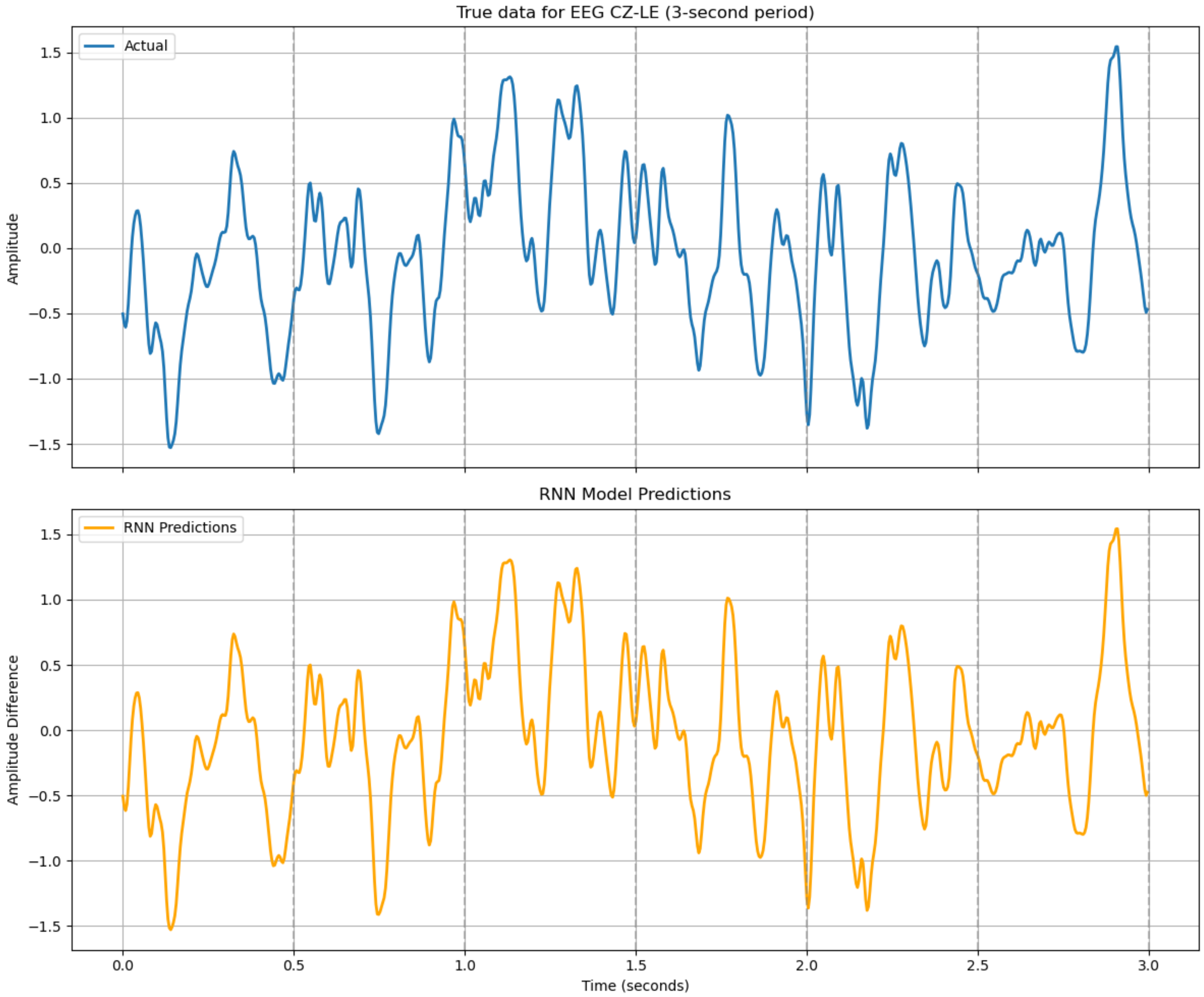
Code behind the scenes:

```python
# Import necessary libraries

import numpy as np                                  # for data manipulation
import matplotlib.pyplot as plt                     # for data visualisation
import mne                                          # for EEG data handling
from statsmodels.tsa.ar_model import AutoReg        # for auto-regressive modelling
import tensorflow as tf                             # for deep learning
```

```python
# Split the data into training and testing sets
train_size = int(len(channel_data) * 0.8)
train_data = channel_data[:train_size]
test_data = channel_data[train_size:]

# Autoregressive Model
def fit_ar_model(data, order=0.5*raw_preprocessed.sfreq):
    model = AutoReg(data, lags=order)
    model_fit = model.fit()
    return model_fit

def forecast_ar(model_fit, steps):
    return model_fit.forecast(steps=steps)

# Fit AR model and make predictions
ar_model = fit_ar_model(train_data)
ar_predictions = forecast_ar(ar_model, steps=len(test_data))
```

```python
# Recurrent Neural Network (RNN) Model
def create_sequences(data, seq_length):
    sequences = []
    targets = []
    for i in range(len(data) - seq_length):
        seq = data[i:i+seq_length]
        target = data[i+seq_length]
        sequences.append(seq)
        targets.append(target)
    return np.array(sequences), np.array(targets)

# Prepare data for RNN
seq_length = 50  # Number of time steps to look back
X_train, y_train = create_sequences(train_data, seq_length)
X_test, y_test = create_sequences(test_data, seq_length)
```

```python
# Build the RNN model
model = tf.keras.Sequential([
    tf.keras.layers.SimpleRNN(50, activation='relu', input_shape=(seq_length, 1)),
    tf.keras.layers.Dense(1)
])

model.compile(optimizer='adam', loss='mse')

# Train the model
history = model.fit(X_train, y_train, epochs=50,
                    batch_size=32, validation_split=0.1, verbose=1)

# Make predictions
rnn_predictions = model.predict(X_test)
```


True data for EEG CZ-LE (3-second period)
Actual
Amplitude
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
RNN Model Predictions
RNN Predictions
Amplitude Difference
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Time (seconds)

5. Running models on clinical dataset (Demo)

- Import data, etc.
- Train AR model
- Visualise AR Model
- Train RNN
- Visualise RNN Model

Overview

1. Introduction to Time Series Data in Healthcare
2. Common Time Series Models
3. Preparing timeseries data for ML (Demo)
4. Introduction to more advanced models
5. Running models on clinical dataset (Demo)
6. Summary
7. Q&A

6. Summary

- Data as a time-series; Time-series as a signal
- Data Preparation
- Basic and complex time-series models
- Broad Applications
- Challenges and Future

Thank you & good luck with using ML/AI with time-series data!

Q&A if time…

References and attributions

Schaffer, A. L., Dobbins, T. A., & Pearson, S. A. (2021). Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions. BMC medical research methodology, 21, 1-12.

Watson, L., Qi, S., DeIure, A., Link, C., Chmielewski, L., Hildebrand, A., ... & Ruether, D. (2021). Using autoregressive integrated moving average (ARIMA) modelling to forecast symptom complexity in an ambulatory oncology clinic: harnessing predictive analytics and patient-reported outcomes. International Journal of Environmental Research and Public Health, 18(16), 8365.

Ihne-Schubert, S. M., Kircher, M., Werner, R. A., Lapa, C., Einsele, H., Geier, A., & Schubert, T. (2023). Vector autoregression: Useful in rare diseases?—Predicting organ response patterns in a rare case of secondary AA amyloidosis. Plos one, 18(8), e0289921.

References and attributions

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., ... & Dean, J. (2018). Scalable and accurate deep learning with electronic health records. NPJ digital medicine, 1(1), 1-10.

Rasmy, L., Nigo, M., Kannadath, B. S., Xie, Z., Mao, B., Patel, K., ... & Zhi, D. (2022). Recurrent neural network models (CovRNN) for predicting outcomes of patients with COVID-19 on admission to hospital: model development and validation using electronic health record data. The Lancet Digital Health, 4(6), e415-e425.

Bongiorni, L., & Balbinot, A. (2020). Evaluation of recurrent neural networks as epileptic seizure predictor. Array, 8, 100038.

References and attributions

Figure of EEG sampling: LUCK, Steven J., An Introduction to the Event-Related Potential Technique, Massachusetts Institute of Technology MIT Press books, 2nd Ed., 2014, ISBN 978-0-262-52585

Figure of Neural Network: Glosser.ca, CC BY-SA 3.0 <https://creativecommons.org/licenses/by-sa/3.0>, via Wikimedia Commons https://commons.wikimedia.org/wiki/File:Colored_neural_network.svg

Figure of Multi-modal Neural Networks: Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., ... & Dean, J. (2018). Scalable and accurate deep learning with electronic health records. NPJ digital medicine, 1(1), 1-10.