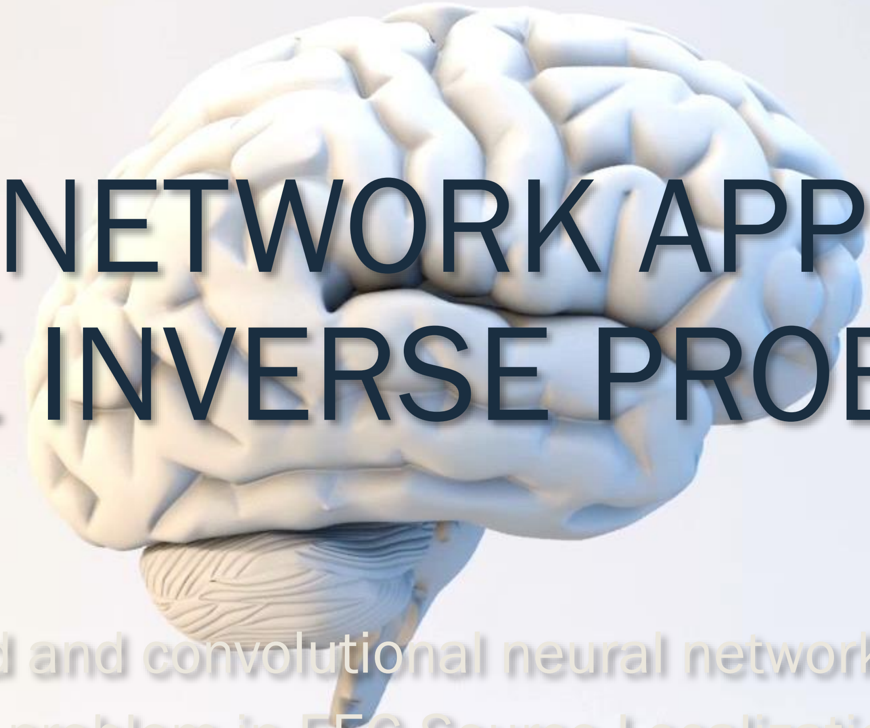
NEURAL NETWORK APPROACH TO THE INVERSE PROBLEM



Using feed forward and convolutional neural network to solve the inverse problem in EEG Source Localization

Electroencephalogram

- Cheap and efficient
- High temporal resolution
- Fixed positions
- Information loss and low spatial resolution

Source Localization: Forward Model

- Simulation of data
- Projection matrix
- Noise
- Volume conduction

$$\Phi = KJ + n$$

Source Localization: Inverse Problem

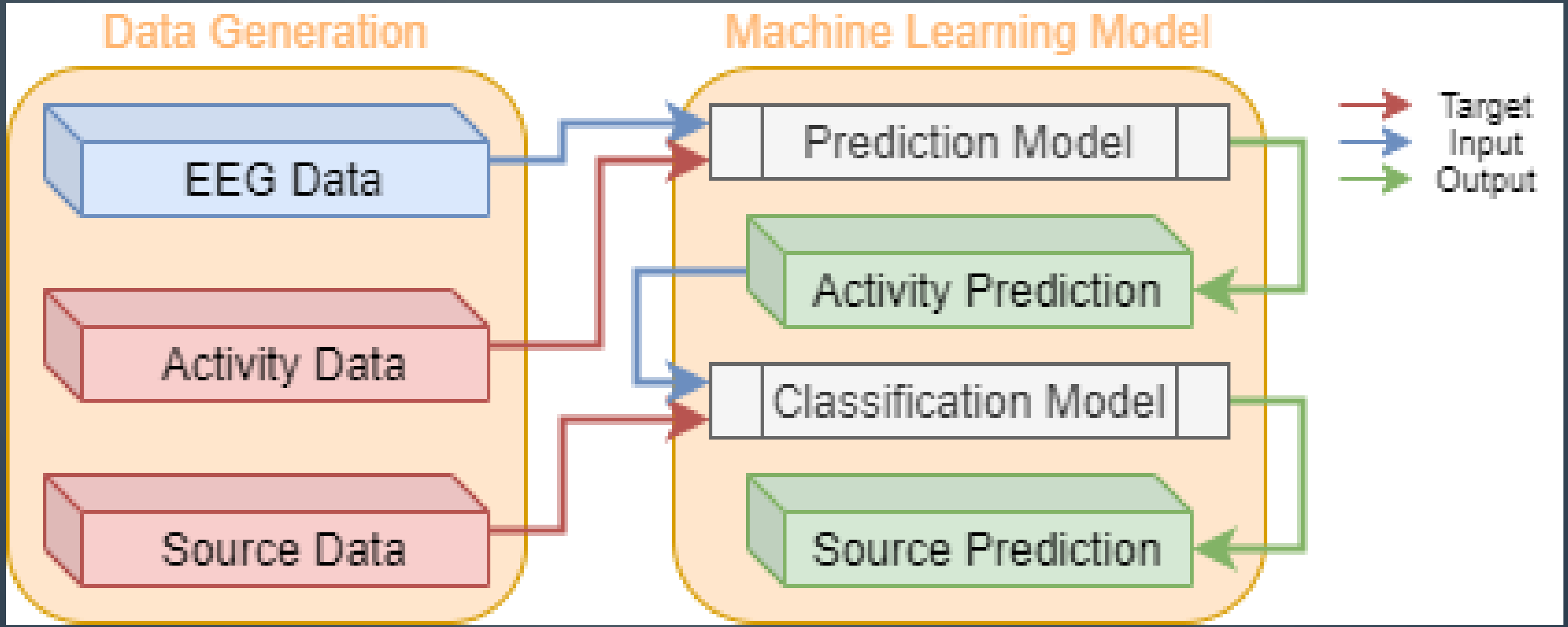
Ill posed problem

- Noise reduction
- Inverse projection matrix
- Spread of neural activity
- 3D to 2D

Mostly mathematical

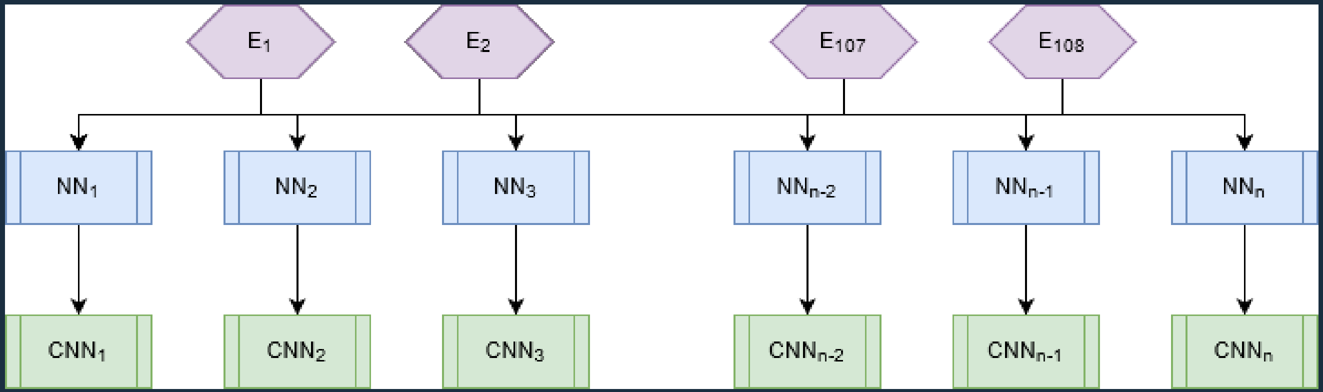
Methodology

- Forward Model ⟶ Data Generation
- Inverse Model ⟶ Machine Learning Model



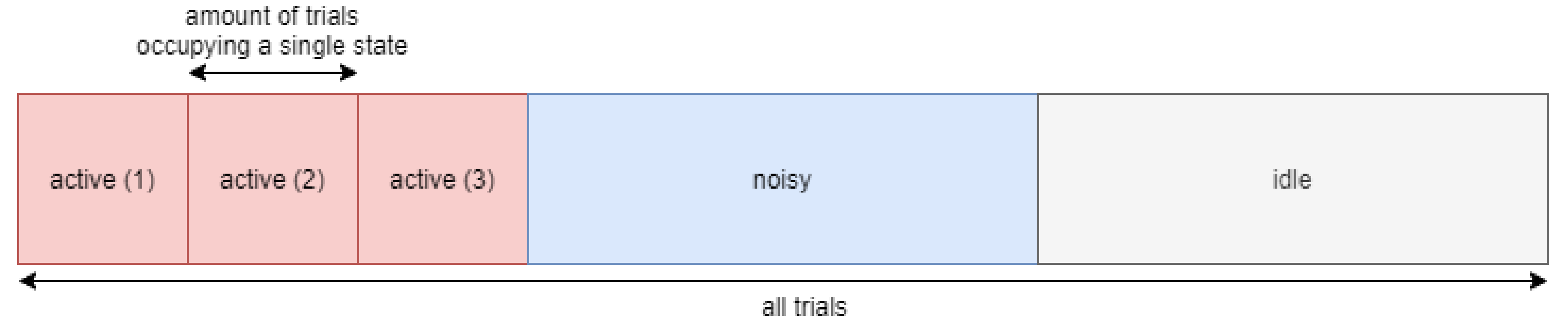
Machine Learning Model

- There 108 electrodes present
- Each dipole has a corresponding Standard Neural Network and Convolutional Neural Network



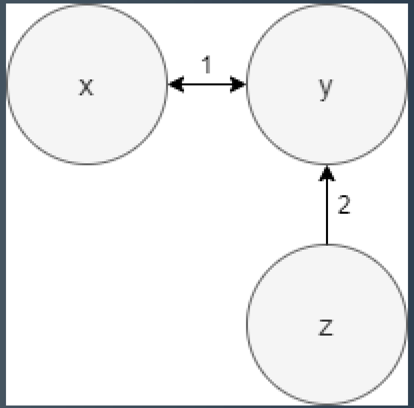
Data Generation: State Determination

- Each dipole is in one state during the whole trial
- The states of the dipoles differ in their neural activity
- There are 3 active (source) dipoles
- There are n/2 noisy dipoles



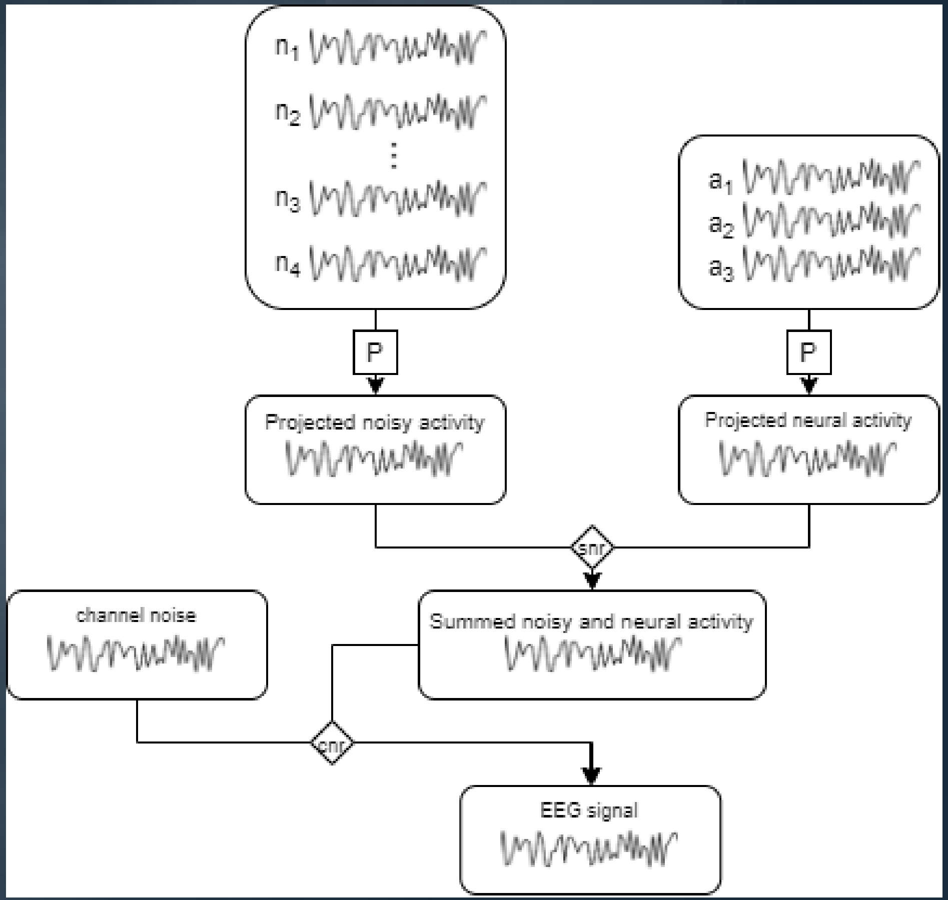
Data Generation: Neural Activity Simulation

- Active: AR(1) process:
  - *The connectivity of each source dipole differs*
- Noisy: pink noise characterized by 1/f shaped power
- Idle: no recorded dipole magnitude



Data Generation: Projection

- Noise ratio's: Signal to noise & Channel Noise
- Projection matrix from Huang et al. (2016)



Time Series Prediction

- Data is shuffled removing time dependency:

  Trials x Timesteps x electrodes → Randomized Timesteps x electrodes

- The data input is of size 108 (electrodes)
- The data output is of size 1 (dipole magnitude)
- Hyper parameters: Batch sizes, Adam optimizer, MSE, learning rate, nodes

Time Series Classification

- The input data is the dipole magnitude during one trial
  - The order of the time steps is important
- The output is the state of the dipole
  - 1 if active (source)
  - 0 if noisy or idle
- Hyper Parameters: Convolutional Nodes, Kernel Size, Stride, Dense Nodes, Batch size, Adam Optimizer, Log loss, learning rate

Validation

- Hyper parameters: Max validation fails, Validation frequency, Validation

Algorithm 1: Testing validation criteria

```
if lowest previous loss - loss ≥ min_val then
    copy Model;
    set validation counter to zero;
else
    increase validation counter by one;
    if validation counter ≥ max_val_amount then
        Stop training process
    end
end
```

Performance Measures

- Prediction: Mean Squared Error
- Classification: True Positive, True Negative, Dipole Accuracy
- Dipole accuracy: choosing the correct three source dipoles

Benchmark Models

- eLoreta Model(Prediction):
  - *Iteratively minimizes weighted minimum norm equations*
  - $\hat{J} = T\Phi$
  - *Resolution matrix* $(R = KT)$
  - *Spread function*
- Variance Model (Classification)

Gridsearch

- Bayesian Tuning
- 628 billion options
- Trials: 1000
- Timesteps: 100
- Dipoles: 1000

Hyper Parameters	Time Series Model	Classification Model
Epochs	50, 100, 200	50, 100, 200
Batch Size	50, 100, 200	10, 20, 30
Learning rate	2e-4, ... , 9e-4	2e-4, ... , 9e-4
Minimum Validation Increment	1e-8, 5e-8, 1e-7	1e-8, 5e-8, 1e-7
Validation Frequency	5, 6, 7, 8, 9, 10	1, 2, 3, 4, 5
Maximum Validation Fails	50, 100, 200	50, 100, 200
Amount of Nodes (Prediction Model)	20, ... , 200	-
Amount of Nodes (Classification Model):		
Convolutional Layer	-	10, 12, ... , 30
Kernel Size (Convolution layer)	-	2, 4, ... , 10
Stride (Convolution Layer)	-	1, 2, 3
Amount of Nodes (Classification Model):		
Dense Layer	-	5, 10, ..., 40

Table 4.2: Hyper parameter values Machine Learning Model

Evaluation: Prediction

- Research Questions:
  - *What is the difference between prediction of dipoles near the scalp and at the center*
  - *Is there a difference when active sources are close to each other*
  - *Are there significant differences between the outputs of the different neural networks*
  - *Are the optimal hyper parameters similar for all neural networks*
- Benchmark:
  - *How does the model compare in terms of MSE performance*

Evaluation: Classification

- Research Questions:
  - *How confident is the model*
  - *How do noisy and idle time series differ*
- Benchmark models:
  - *Use the Variance model on both the prediction output of the eLORETA and the Machine Learning model*
  - *Use the Machine Learning Classification model on the output of the eLORETA*
  - *How does the model compare in terms of TN, TP and dipole accuracy*

Future Work & Discussion

Improving the prediction model by implementing an RNN

Taking the distance to a source dipole into account

Training using a data set where each dipole occupies the active state

Using concepts of previous mathematical models

QUESTIONS