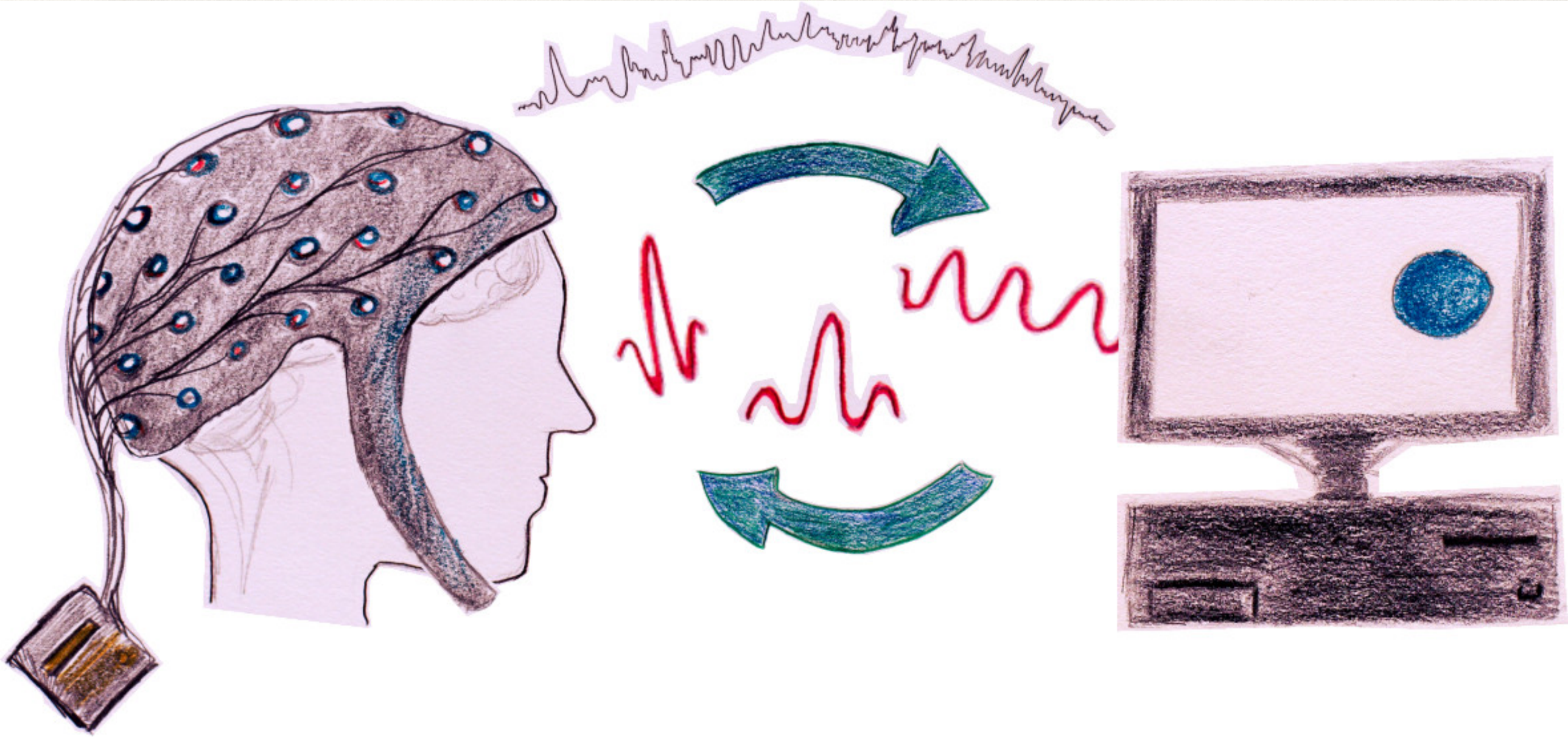


Deep Learning EEG Response Representation for BCI

Paper ID: 12, NNFL Project

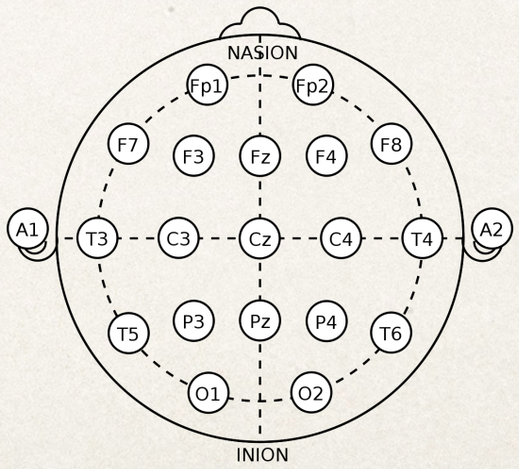
Introduction

What is EEG?

- Acronym for Electroencephalogram
- Measures electric potential across different regions of brain
- Individual neurones - very little electric potential
- Bunch of neurones in harmony - electric potential of measurable quantities

How is EEG performed?

- Electrodes placed on the scalp
- Potential of electrodes measured w.r.t. a pre-defined reference (Cz electrode, in our case)
- Placement of electrodes - a defined standard (the 10-20 system, in our case)
- Non-invasive technique
- High temporal resolution
- Low spatial resolution



The 10-20 System

What is BCI?

- Acronym for Brain Computer Interface

- Direction communication between brain and an external device

- Researching, mapping, assisting, augmenting, or repairing human cognitive or sensory motor functions

Objective

- Train a deep neural network to interpret EEG signals, and determine the motor activity being imagined by the person

- Network of choice: Convolutional Neural Network (CNN)

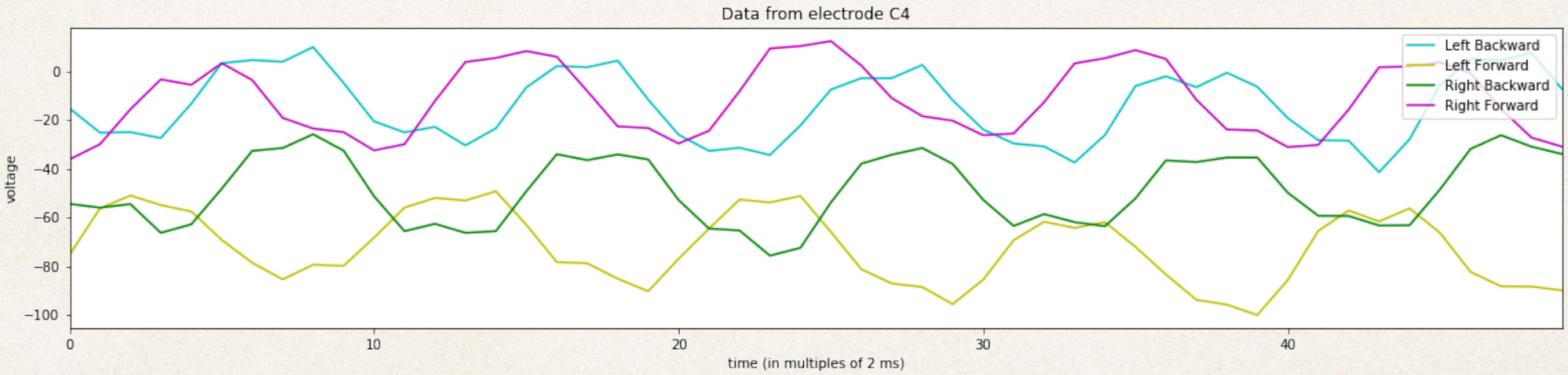
  - EEG signals: image-like data

Dataset

- Recorded EEG signals for 4 imagined motor activities:
  - Left arm forward movement
  - Left arm backward movement
  - Right arm forward movement
  - Right arm backward movement
- Data from 19 (standardised) electrodes: FP_1, FP_2, F_3, F_4, C_3, C_4, P_3, P_4, O_1, O_2, F_7, F_8, T_3, T_4, T_5, T_6, F_z, C_z and P_z
  - Split into 19 channels

Dataset (contd.)

- Recording frequency - 500 Hz
- 50 sample points - 1 training instance

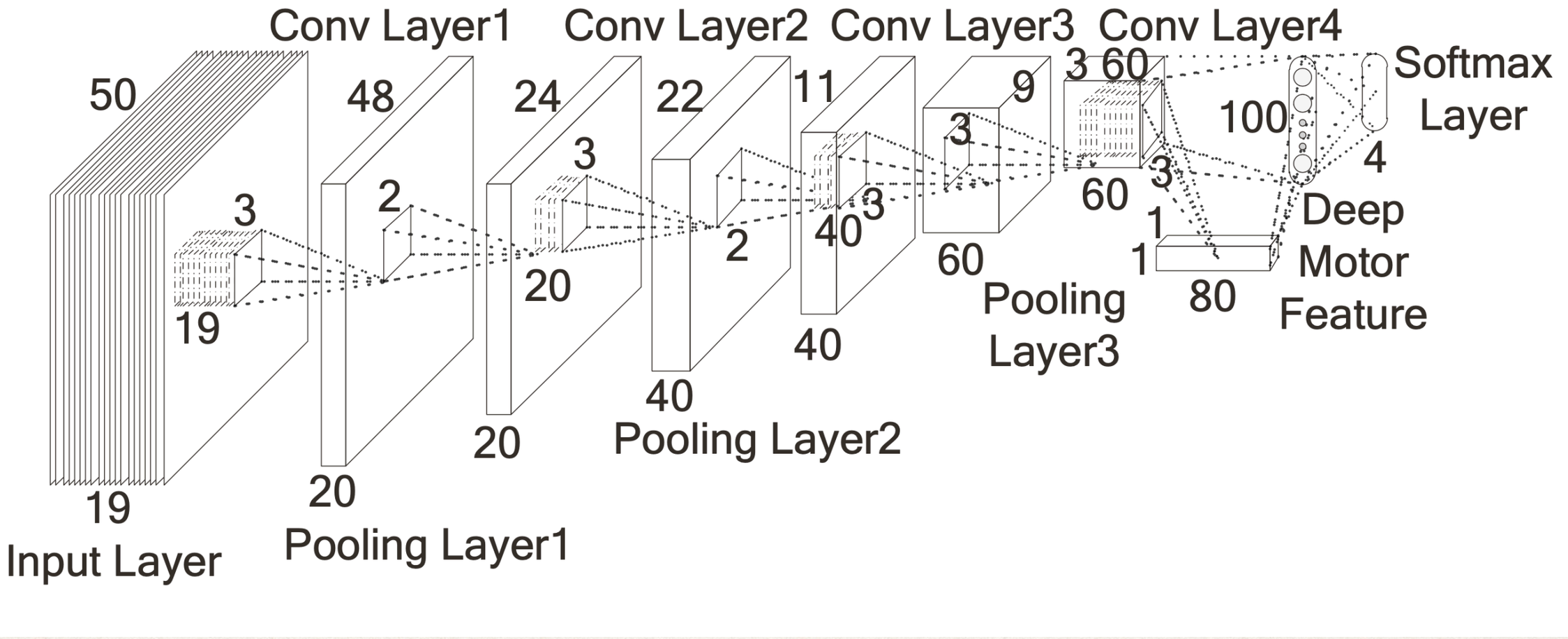


Sample of Data - Channel C_4

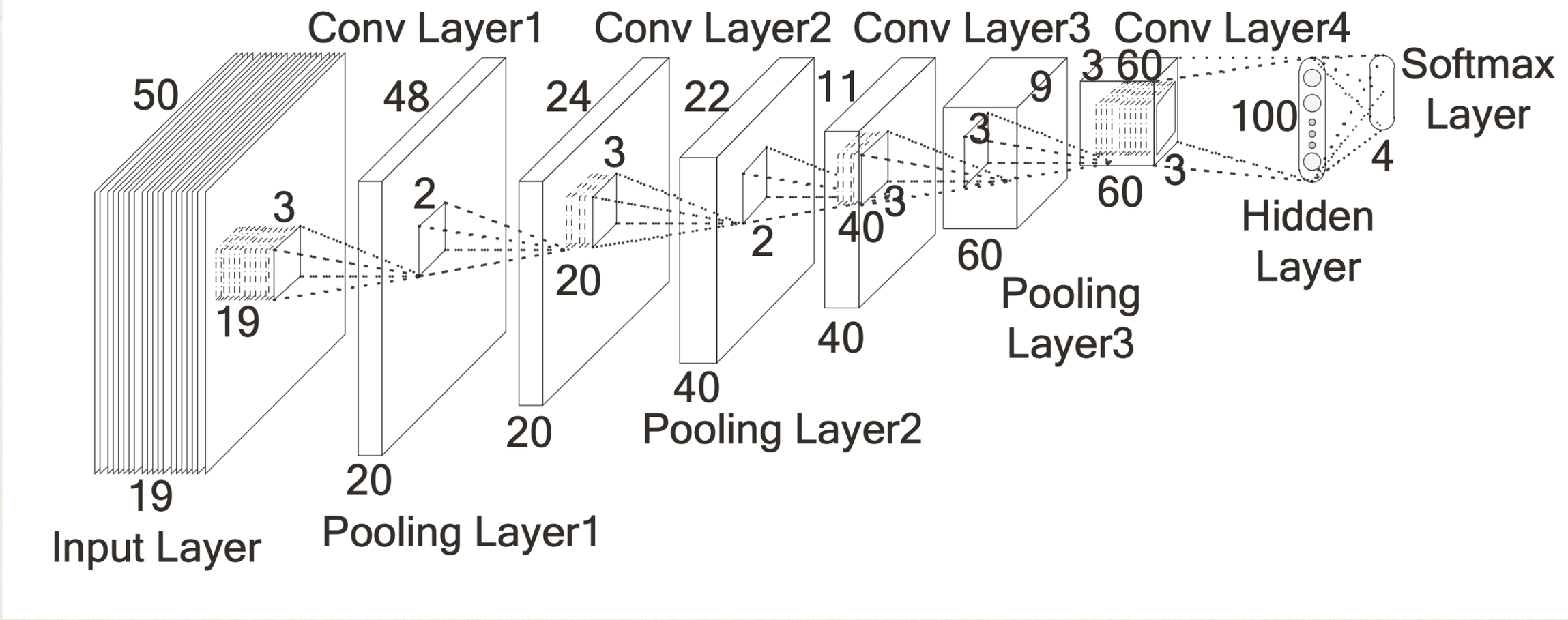
Preprocessing Steps

- Original data - 7000 sample points for each motor activity
  - Split into 140 instances of 50 sample points each
    - 19 channels per input
- Standardised to z-scores
- One-hot encoded targets
- Shuffled & split into training set (), validation set (), test set()
  - Stratified split

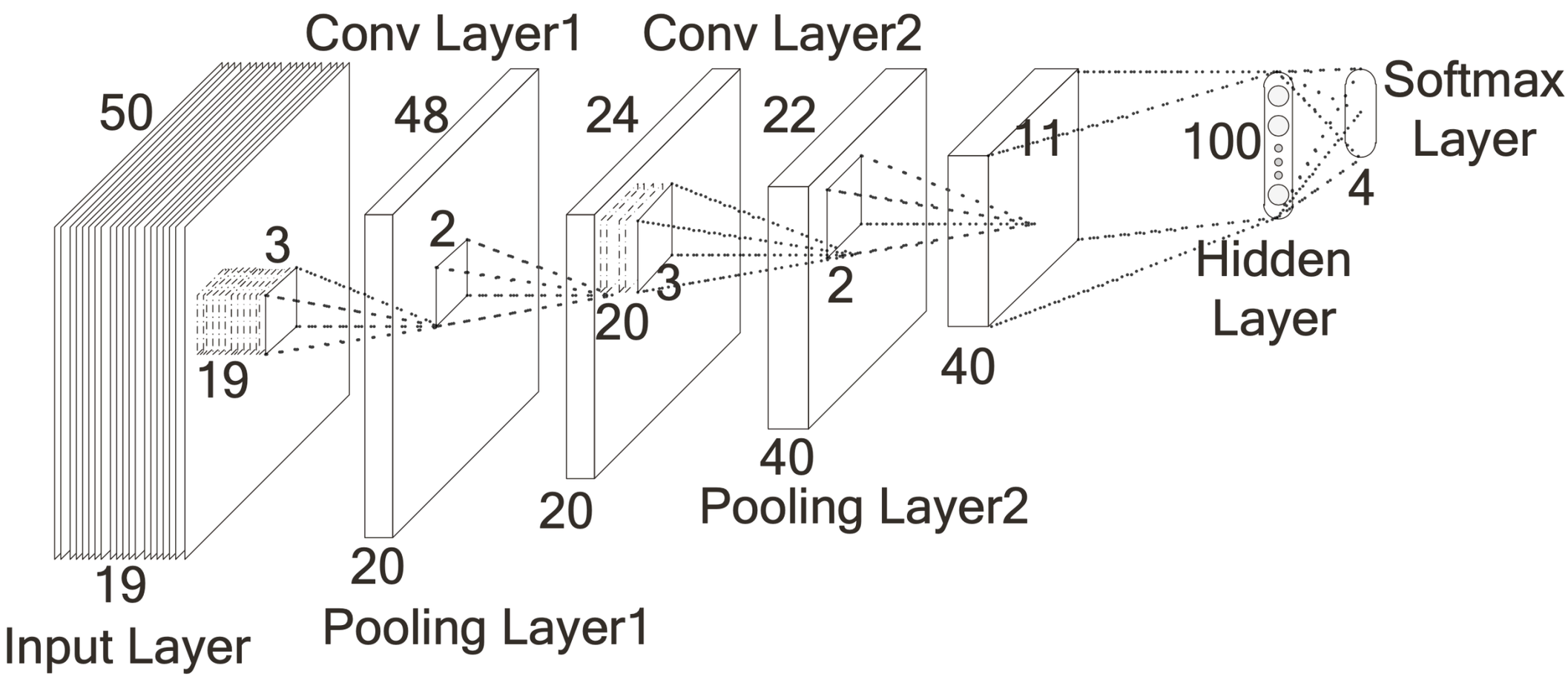
Network Architecture: Multi-Scale CNN



Network Architecture: Single-Scale CNN



Network Architecture: Shallow CNN



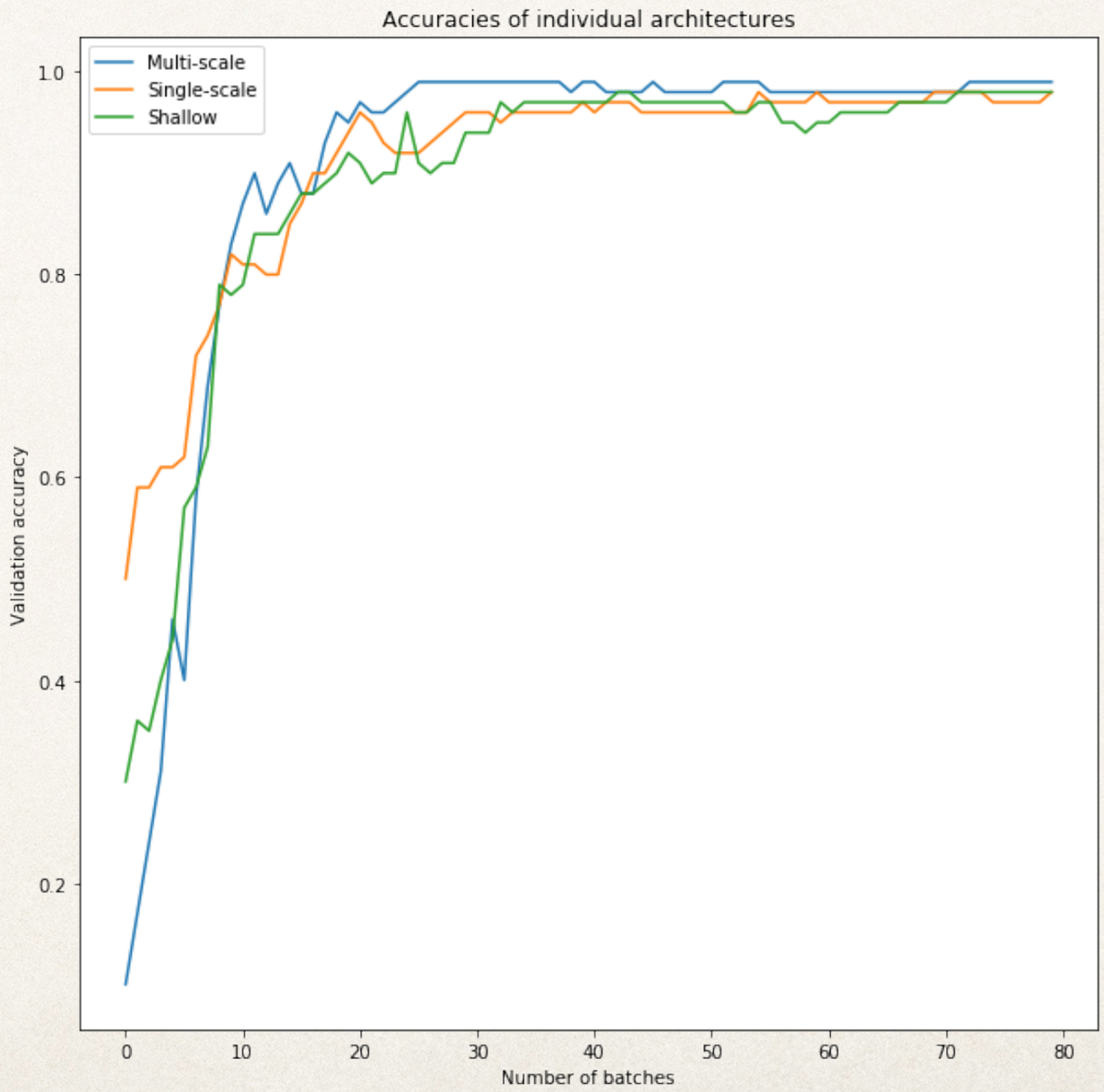
Training Methodology

- 5 training instances - 1 mini batch
- Optimiser - SGD
- Initialisers used (in select layers, according to paper) -
  - Biases - zero
  - Weights - he_uniform initialiser

Results

- Accuracy obtained - between 0.99 and 1
- Number of epochs taken - 2

Accuracy vs Time



Comparison of Architectures

	Loss	Validation Accuracy
Multi-scale CNN	0.146	0.996
Single-scale CNN	0.149	0.981
Shallow CNN	0.102	0.988

Libraries Used

- Keras -
  - Building & training the network
- Pandas -
  - Data storage and processing
- Numpy -
  - Mathematical operations
- Sci-kit learn -
  - Scaling to z-scores
  - One hot encoding target
  - Splitting data into test, validation and training sets
- Matplotlib -
  - Plotting graphs

Thank You

- Paper implemented as part of the course ‘Neural Networks and Fuzzy Logic’ [BITS F312], BITS Pilani, Pilani Campus, First Semester 2019-2020

- Paper ID: 12

- Group members:

  - Jivat Neet - 2017A7PS0050P

  - Laksh Singla - 2017A7PS0082P

  - Shubham Saxena - 2017A7PS0302P