Handwriting BCI Dataset Documentation

Overview

These data are reported in Willett et al. 2021 and consist of ten separate sessions (listed below in "Session Table"). The unprocessed data reside in the "Datasets" folder. Processed versions of the data reside in the "RNNTrainingSteps" folder, which includes time-warped data, HMM-generated data labels, and RNN decoders (see the accompanying code repository https://github.com/fwillett/handwritingBCI). All data files are .mat files that can be loaded with MATLAB or Python (using the scipy function ***scipy.io.loadmat***). The "BigramLM" folder contains Kaldi language model files for running the bigram language model.

Session Table

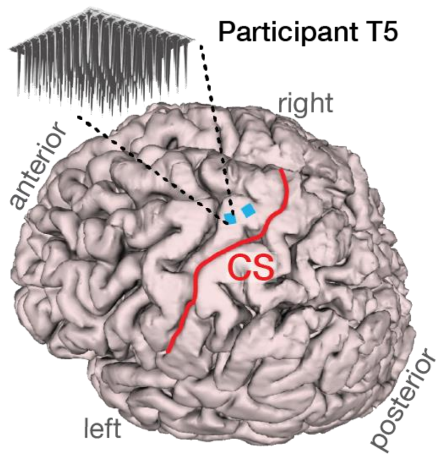
Date (Trial Day)	Description	Data
2019.05.08 (994)	Sentence and character writing pilot day (no real-time decoding)	• Sentence writing collected as training data (102 sentences, no decoding) • Single character writing (27 repetitions per character)
2019.06.03 (1020)	Attempted handwriting of straight lines	• Instructed delay straight-line writing (24 repetitions each for 48 straight-line conditions)
2019.11.25 (1195)	Real-time decoding pilot day	• Sentence writing for training data (50 sentences, no decoding) • Sentence writing for performance evaluation (34 sentences, with real-time decoding) • Single character writing (10 repetitions per character)
2019.12.09 (1209)	Real-time decoding pilot day	
2019.12.11 (1211)	Copy-typing evaluation	
2019.12.18 (1218)	Copy-typing evaluation	
2019.12.20 (1220)	Copy-typing evaluation	
2020.01.06 (1237)	Copy-typing evaluation	
2020.01.08 (1239)	Copy-typing evaluation	
2020.01.13 (1244)	Free-answer evaluation	• Sentence writing with artificial pauses (30 sentences, no decoding) • Phrase writing from memory (100 phrases, no decoding) • Sentence writing for performance evaluation (25 free-answer questions) • Single character writing (10 repetitions per character)
2020.01.15 (1246)	Free-answer evaluation	

Raw Datasets

For sessions containing characters and sentences data (all sessions except 2019.06.03), raw data is contained in two .mat files: ***singleLetters.mat*** (containing trials where participant T5 attempted to write isolated characters) and ***sentences.mat*** (containing trials where T5 attempted to write complete sentences). For session date 2019.06.03, instead of ***singleLetters.mat*** there is a ***straightLines.mat*** file with the same data format, except containing attempted straight-line movements (single, straight pen strokes) as opposed to complete characters.

The neural data contained in these files consists of binned spike counts (10 millisecond bins). Binned spike counts are integer values equal to the number of times the voltage time series on a given electrode crossed a -3.5*RMS threshold during that time bin. The thresholds were kept consistent throughout an entire session (but differ from session to session). Binned threshold crossing counts are a simple way to quantify the amount of spiking neural activity occurring during a time window, and have been shown to perform comparably to spike-sorted single neurons for both brain-computer interfaces (Christie et al., 2014) and for investigating the structure of neural population activity (Trautmann et al., 2019).

The neural recordings are from two 96 channel microelectrode arrays (Utah arrays) implanted in hand knob area of precentral gyrus. The image below shows the array locations (the red line on the brain denotes the central sulcus).



Single Letters

<u>Overview</u>

The ***singleLetters.mat*** files contain neural data recorded while participant T5 attempted to write single letters and symbols according to text cues displayed on a computer monitor in an instructed delay paradigm (no real-time handwriting decoding occurred during these blocks). Each trial first began with a delay period or variable length, during which a single character appeared on the screen above a red square. During the delay period, T5 waited and prepared to write that character. Then, the red square in the center of the screen turned green (go cue), instructing T5 to immediately begin trying to write the character. After the 1 second go period, the next trial's delay period began immediately.

Variables

neuralActivityCube_{x}*:** These are 3-dimensional neural data tensors, each containing the neural activity recorded while T5 attempted to write the character *{x}*. These tensors have dimensionality K x T x N, where K is the number of trials, T is the number of time steps (201), and N is the number of electrodes (192). Each element is the binned threshold crossing count observed for a single trial, time step and electrode. The data have been aligned such that time step 51 corresponds to the go cue time. These tensors can be used to quickly analyze the data, including averaging over multiple trials to denoise the neural activity. They were created from ***neuralActivityTimeSeries and are included for convenience. Trials were placed into each cube in the order in which they occurred during the session.

neuralActivityTimeSeries: A time series of binned spike counts for all 192 electrodes and all 10 millisecond time steps of data. The first 96 channels correspond to array 1 (the more lateral array) and the last 96 channels correspond to array 2 (the more medial array). Note that there are discontinuous breaks in the time series whenever the block number changes (neural activity recorded in between blocks was not saved).

clockTimeSeries: A time series of the system clock time, in seconds, for each time 10 millisecond time step of data. The clock time resets to 0 at the beginning of each new block.

blockNumsTimeSeries: A time series of the current block number, for each 10 millisecond time step. This can be used to determine the block during which any time step of data was recorded.

goCueOnsetTimeBin: Each element i indicates the time step on which the go cue was given for trial i.

delayCueOnsetTimeBin: Each element i indicates the time step on which the delay period began for trial i.

characterCues: Each element i is a string representation of the character that T5 was cued to attempt to write for trial i (the strings match the names of the ***neuralActivityCubes***). Note that t5.2019.05.08 is a unique session in that it contains one additional cue not tested on any other session: 'doNothing'. During this condition, T5 was instructed to sit idly and perform no motor action.

blockList: A list of the blocks included in this dataset. The block numbers do not necessarily start at 1 and may skip numbers.

blockStartDates: An array of date strings that describe the start time of each block (up to 1 second resolution). Can be used to understand how much time passes between blocks.

meansPerBlock: Each row i of this matrix represents the mean spike count observed on each electrode during the ith block. This can be easily computed from ***neuralActivityTimeSeries*** and ***blockNumsTimeSeries***, and is included only for convenience (as some of our post-processing steps use this).

stdAcrossAllData: This vector represents the standard deviation of the spike count observed on each electrode. This can be easily computed from ***neuralActivityTimeSeries*** and ***blockNumsTimeSeries***, after subtracting the mean within each block. It is included only for convenience (as some of our post-processing steps use this).

arrayGeometryMap: A matrix where each entry (i,j) contains the electrode number that is at location (i,j) on the physical microelectrode array. Increasing values of i (rows) move towards the central sulcus (along the anterior-posterior axis) and increasing values of j (columns) move medially along the precentral gyrus (medial-lateral axis). Note that the electrodes are spaced 0.4 mm apart on the microelectrode array.

Sentences

Overview

The ***sentences.mat*** files contain neural data recorded while participant T5 attempted to write complete sentences according to text cues displayed on a computer monitor in an instructed delay paradigm. For some of the trials, no real-time handwriting decoding was performed; these data were used to train the RNN decoder during the session. In other trials, the RNN decoder was operating in real-time as T5 attempted to write the sentence; these data were used to evaluate the real-time decoder performance.

Each trial first began with a delay period lasting 5 seconds, during which a sentence (or question) appeared on the screen above a red square. During the delay period, T5 waited and prepared to write that sentence (or answer the question). Then, the red square in the center of the screen turned green (go cue), instructing T5 to immediately begin trying to write the sentence. When T5 determined that he was finished writing, he turned his head to the right. Our system automatically detected rightward head turns and used this signal to trigger the next trial.

There are four types of sentence writing trials: sentence copying (the most common), sentence copying with artificial pauses, copying phrases from memory, and freely answering question prompts. The last three types of trials occurred only in the two “free-answer” sessions, which were designed to evaluate the RNN decoder’s performance on freely composed sentences. The training data for the free-answer sessions was designed to create more variability in T5’s handwriting, in an attempt to make the RNN decoder more robust to pauses and changes in writing speed. We briefly describe these trial types below.

- Sentence copying with artificial pauses: For these sentences, we randomly inserted hash mark symbols (#) throughout the sentence. T5 was instructed to take a brief (~1 second) pause whenever he came across a hash mark symbol.
- Copying phrases from memory: In these trials, T5 copied short (1-4 word) phrases from memory. During the delay period, T5 memorized the phrase. During the go period, the phrase disappeared from the screen, forcing him to write it from memory instead of copying.
- Freely answering question prompts: In these trials, T5 answered a question prompt that appeared on the screen with a sentence of his own choosing. Many of these trials contain a long pause after the go cue, since T5 used this time to think of an answer to the question and compose his response before beginning to write it.

Variables

neuralActivityCube: For convenience, the neural activity recorded during all of the sentences has been assembled into a 3D tensor of dimension S x T x N (S sentences, T time steps, and N electrodes). Each element contains the binned spike count observed for that sentence, time step and electrode. This tensor can be fully reconstructed from ***neuralActivityTimeSeries*** and is included only to be helpful. Note that the time dimension was extended to be as long as the longest sentence. Thus, all other sentences end earlier than T and contain excess data from the following sentence(s). Use ***numTimeBinsPerSentence*** to determine how many time steps each sentence lasts.

sentencePrompt: An array of strings, each entry i indicating the text prompt for sentence i. The sentences are all lower case with some punctuation: periods, commas, question marks, apostrophes and spaces. Spaces are indicated with a greater than sign (>) in these strings (since T5 wrote this sign to denote a space). Periods are indicated with a tilde (~) in these strings (since T5 wrote this sign to denote a period). For most blocks, T5 was instructed to copy this text prompt. However, in the free answer blocks, T5 responded to the prompt with his own sentence. In the sentences with # symbols, T5 was instructed to pause briefly whenever he encountered a # symbol (instead of writing the # symbol; see above 'Overview' section).

intendedText: An array of strings, each entry i indicating the text that T5 intended to write. This is equal to ***sentencePrompt*** except in two cases: sentences with hash marks (which cue T5 to take an artificial pause) and free writing sentences. For sentences with hash marks, we removed them here (since T5 did not write these). For the free writing sentences, we determined T5's intended sentence by discussing with him after each sentence his intended meaning. Note that these are only approximate, mainly because T5 may have made spelling errors (T5 reported several times that he was not good at spelling and was uncertain of how to spell certain words). We did not review each word with T5 to determine how he intended to spell each one. Also, T5 was inconsistent with punctuation – sometimes he chose to end his sentences with periods, and other times he did not. Since we did not check with T5 after each sentence whether he intended to end it with a period, we decided to remove all periods from ***intendedText*** (and, when we computed the character error rate for our RNN decoder, ignored periods).

numTimeBinsPerSentence: Each element i indicates the number of time steps that sentence i lasted.

sentenceCondition: Each entry i contains a string describing the experimental condition under which sentence i was collected. There are six possible conditions listed below.

- 'OL Copy': copying the sentence prompt in "open-loop" (i.e. with no real-time decoder).
- 'OL Copy With Pauses': copying the sentence prompt while artificially pausing whenever a # symbol is encountered.
- 'OL Copy from memory': copying the sentence prompt from memory.
- 'CL Pandarinath': copying the sentence prompt in "closed-loop" (i.e. with the real-time decoder running). These sentences are always the same 7 sentences used in (Pandarinath et al., 2017) and were collected for a direct comparison to this prior work.
- 'CL Corpus': copying the sentence prompt in closed-loop, with the sentence prompts drawn at random from the BNC.
- 'CL Free Write': answering the question prompt freely in closed-loop.

sentenceBlockNums: Each entry i indicates the block number during which sentence i was recorded.

excludedSentences: Each entry i contains a 1 if this sentence was (and should be) excluded from analysis. 33 out of a total of 1,000 sentences are marked as excluded (3.3%). This typically occurred whenever our system accidentally detected a head turn from T5 and triggered the next trial before he was actually finished, making these sentences incomplete. Additionally, for the free typing trials, we removed one sentence where T5 could not think of a response and wanted to skip the question, and one sentence where we were unable to determine T5's intended spelling of a restaurant name. In one copy typing session (2020.01.06), a software bug incorrectly initialized the RNN decoder at the beginning of each evaluation block; for this session, we excluded the first trial of each evaluation block (4 trials total). Finally, in some copy-from-memory trials, T5 indicated to us that he had forgotten the phrase – when this occurred, we marked the trial as excluded.

neuralActivityTimeSeries: A time series of binned spike counts for all 192 electrodes and all 10 millisecond time steps of data. The first 96 channels correspond to array 1 (the more lateral array) and the last 96 channels correspond to array 2 (the more medial array). Note that there are discontinuous breaks in the time series whenever the block number changes (neural activity recorded in between blocks was not saved).

clockTimeSeries: A time series of the system clock time, in seconds, for each time 10 millisecond time step of data. The clock time resets to 0 at the beginning of each new block.

blockNumsTimeSeries: A time series of the current block number, for each 10 millisecond time step. This can be used to determine the block during which any time step of data was recorded.

blockList**:** A list of the blocks included in this dataset. The block numbers do not necessarily start at 1 and may skip numbers.

blockStartDates**:** An array of date strings that describe the start time of each block (up to 1 second resolution). Can be used to understand how much time passes between blocks.

goCueOnsetTimeBin**:** Each element i indicates the time step on which the go cue was given for trial i.

delayCueOnsetTimeBin: Each element i indicates the time step on which the delay period began for trial i.

sentenceEndTimeBin: Each element i indicates the time step at which sentence i was finished.

arrayGeometryMap: A matrix where each entry (i,j) contains the electrode number that is at location (i,j) on the physical microelectrode array. Increasing values of i (rows) move laterally down the precentral gyrus and increasing values of j (columns) move towards the central sulcus. Note that the electrodes are spaced 0.4 mm apart on the microelectrode array.

rnn_charProbTimeSeries*:* A time series of the character probabilities output by the RNN decoder for each 10 millisecond time step. Entries are equal to zero during blocks where the real-time decoder was not active. The mapping between columns and characters is defined by ***rnn_charMapASCII*** (column *i* corresponds to the character contained in the *i*th entry of this map).

rnn_newCharTimeSeries*:* A time series of the new character signal output by the RNN decoder for each 10 millisecond time step. This signal was thresholded (threshold = 0.3) to determine when to output a new character during real-time decoding.

rnn_decodedCharTimes*:* A matrix where each row *i* corresponds to a single decoded character by the RNN. Column 1 indicates the time step on which that character was decoded, and column 2 indicates which character was decoded (with an integer value that indexes into ***rnn_charMapASCII***). This matrix can be used to determine when exactly the RNN decoded each character (i.e. when that character was displayed on the screen).

***rnn_decodedText**:* Each entry *i* is a string containing the text that the RNN decoded on trial *i*.

rnn_charMapASCII: A 31 character string that defines a mapping between integers and characters (the *i*th character of this string is mapped to integer *i*). This mapping defines which columns of ***rnn_charProbTimeSeries*** map to which characters, and is also used in ***rnn_decodedCharTimes*** to indicate which character was decoded.

Computer Mouse Templates

In the "Datasets" folder, the ***mouseTemplates.mat*** file contains computer mouse trajectories that were created by tracing each character with a mouse in the same way that T5 described writing that character. These character 'templates' were used to train linear decoders to reconstruct T5's pen tip trajectories (Fig. 1d), based on the assumption that another person drawing the same character shape with a computer mouse would naturally follow a similar velocity trajectory as T5. This file can be used to reproduce Fig. 1d.

References

Christie, B.P., Tat, D.M., Irwin, Z.T., Gilja, V., Nuyujukian, P., Foster, J.D., Ryu, S.I., Shenoy, K.V., Thompson, D.E., Chestek, C.A., 2014. Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance. J. Neural Eng. 12, 016009. https://doi.org/10.1088/1741-2560/12/1/016009

Pandarinath, C., Nuyujukian, P., Blabe, C.H., Sorice, B.L., Saab, J., Willett, F.R., Hochberg, L.R., Shenoy, K.V., Henderson, J.M., 2017. High performance communication by people with paralysis using an intracortical brain-computer interface. eLife 6, e18554. https://doi.org/10.7554/eLife.18554

Trautmann, E.M., Stavisky, S.D., Lahiri, S., Ames, K.C., Kaufman, M.T., O'Shea, D.J., Vyas, S., Sun, X., Ryu, S.I., Ganguli, S., Shenoy, K.V., 2019. Accurate Estimation of Neural Population Dynamics without Spike Sorting. Neuron 103, 292-308.e4. https://doi.org/10.1016/j.neuron.2019.05.003

Willett, F.R., Avansino, D.T., Hochberg, L.R., Henderson, J.M., Shenoy, K.V., 2021. High-performance brain-to-text communication via handwriting. Nature. In press.