

The Inputs

Let's begin with a brief description of the inputs.

Given the string $\sigma = \sigma_0\sigma_1\sigma_2\cdots\sigma_{T-1}$, $\sigma_t \in \{0, 1, \dots, m-1\}$, we form the $m \times (T + m)$ binary-valued matrix M as follows:

$$\text{If } \sigma_t = k \in \{0, 1, \dots, m-1\}, \text{ set } M_{ki} = \begin{cases} 1 & \text{if } i \in [t, t + k + 1] \\ 0 & \text{otherwise} \end{cases}$$

e.g. if $m = 4$, $T = 4$, and $\sigma = 2130$, we get

$$M = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (1)$$

The column vectors of M serve as time-dependent inputs to the system. For example, given the matrix M above, the input vectors at times $t = 0, 1, \dots$ would be $x(0) = (0, 0, 1, 0)$, $x(1) = (0, 1, 1, 0)$, \dots .

In the speech domain,

- The set $\{0, 1, 2, \dots, m-1\}$ is the analog of a *fixed* set of phonemes,
- M is the analog of an *utterance*,
- and the sequences σ , with variable lengths T , are the analogs of *transcriptions*.

☛ Just like in speech, the duration $T + m$ of M varies between utterances.

The goal is to learn to associate the temporal sequences represented by the inputs/utterances (column vectors of M) with their corresponding target labels σ .

The “utterances” used in the demo were generated by

- fixing two integers, m and T_0 , and an alphabet $\{0, 1, \dots, m-1\}$,
- generating a variable T from a uniform distribution with mean T_0 , and then, for each T ,
 - generating a random sequence of length T from the alphabet, and
 - constructing the corresponding input matrix M .

Implementation of the input sequences

The training data is in the form of a dictionary

```
data={'x': inPuts, 'y': transcripts, 'phones': phones}
```

- `inPuts` is a list of matrices with a variable number of columns of the form described above.
- `phones` is a list representing the fixed set of phonemes.
- `transcripts` is a list of strings of variable length representing the labels of the inputs