Encrypted transmission

Hugo Aguettaz

February 4, 2022

In this paper, we present a new method to transmit sensitive message from an emitter to a receptor. The input to the algorithm is a binary message $\boldsymbol{m}$ to transmit. We will first consider single channel transmission and investigate some properties of the method. Then, we will discuss a multi-channels extension.

1 Single-Channel Protected Transmission

In this setting, the message is only transmitted through one channel $\tilde{n}$, i.e., one neuron, whereas the other channels does not contain any relevant information but are necessary to give enough energy to the system. Let $\boldsymbol{m}$ be the binary message to transmit from the source to the target.

With $T_r > 0$, $\boldsymbol{m}$ is not necessary a valid firing sequence. Thus, we first have to make it valid by adding T_r zeros after each one, i.e., $\boldsymbol{m} \mapsto \tilde{\boldsymbol{m}} \in \tilde{\mathcal{Y}}_{T_r}^{L_m}$.

Then we have to generate a random key k of length L_k, that will be used to stimulate the RSNN and recover the message. This is done in two independent stages:

1. We uniformly sample $N-1$ firing signals $\boldsymbol{y}_n \in \mathcal{Y}_{T_r}^{L_k+L_m}$, for all $n \neq \tilde{n}$.
2. We uniformly sample a stimulus $\tilde{u} \in \{0,1\}^{L_k}$ for the $\tilde{n}$-th channel, such that the concatenation $\tilde{u} \parallel \tilde{m}$ is a valid sequence in $\mathcal{Y}_{T_r}^{L_k+L_m}$

1.1 Stage 1

Stage 1 can be done easily by backward filtering forward sampling as presented before.

1.2 Stage 2

For stage 2, the algorithm has to be adapted a little bit. For reasons that would be evident in a while, we should further assume that $L_k \geq T_r$. The objective here is too uniformly sample a binary sequence $(x_1, \ldots, x_{L_k}) \in \{0,1\}^{L_k}$ under the condition that $(x_1, \ldots, x_{L_k}, x_{L_k+1}, \ldots, x_{L_k+L_m}) \in \mathcal{Y}_{T_r}^{L}$, with $L = L_k + L_m$ and where $(x_{L_k+1}, \ldots, x_{L_k+L_m})$ is fixed. Moreover, one could also notice that x_k is independent of $x_{k'}$ for every k' which is not at least T_r closed to k. Thus, with fixed $(x_{L_k+1}, \ldots, x_{L_k+L_m})$, it is suffice to focus on the binary sequence $(x_{L-T_r+1}, \ldots, x_L, x_1, \ldots x_{L_k}, \ldots x_{L_k+T_r+1})$ of length $L_k + 2T_r$.

Let $Z_k = (X_{k-T_r+1}, \ldots, X_k) \in \{0,1\}^{T_r}$ with $k = 1, \ldots, L_k + T_r + 1$ and indices taken modulo L be a Markov chain. Using the constraint functions

$$g_{k-1,k}(z_{k-1}, z_k) = \begin{cases} 1 & \text{if } (x_{k-T_r}, \ldots, x_k) \in \mathcal{Y}_{T_r}^{T_r+1} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

we can represent the problem setting as the factor graph of Figure 1.

The backward filtering pass is done by sum-product message passing as illustrated Figure 2. Starting with the message

$$\overleftarrow{\mu}_{Z_{L_k+T_r+1}}(z_{L_k+T_r+1}) = \begin{cases} 1 & \text{if } z_{L_k+T_r+1} = \check{z}_{L_k+T_r+1} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

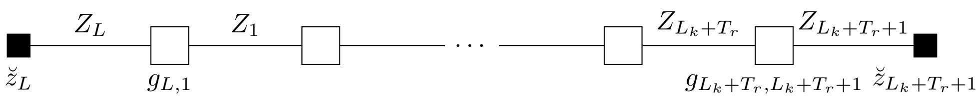

Figure 1: Factor graph with boundary conditions

we can recursively compute all backward messages, from right to left using:

$$\overleftarrow{\mu}_{Z_{k-1}}(z_{k-1}) = \sum_{z_k} g_{k-1,k}(z_{k-1}, z_k)\overleftarrow{\mu}_{Z_k}(z_k), \quad k \in \{1, \ldots, L_k + T_r + 1\}. \tag{3}$$

It can also be expressed in matrix form as

$$\overleftarrow{\boldsymbol{\mu}}_{\boldsymbol{Z_{k-1}}} = \overleftarrow{\boldsymbol{\mu}}_{\boldsymbol{Z_k}} \boldsymbol{A} \tag{4}$$

with

$$\left\{\overleftarrow{\boldsymbol{\mu}}_{\boldsymbol{Z_k}}\right\}_{i_{z_k}} = \overleftarrow{\mu}_{Z_k}(z_k) \tag{5}$$

for $i_{z_k} \in \{1, \ldots, T_r + 1\}$ and

$$z_k = \mathbf{0} \mapsto i_{z_k} = 1 \tag{6}$$

$$z_k = \boldsymbol{e_i} \mapsto i_{z_k} = i + 1 \tag{7}$$

and with

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \in \{0, 1\}^{T_r+1 \times T_r+1}. \tag{8}$$

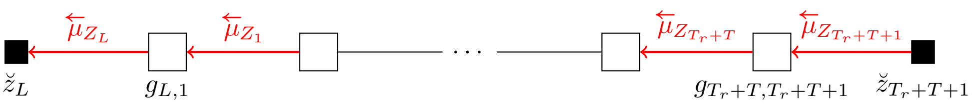

Figure 2: Backward filtering

The forward sampling is finally done as shown in Figure 3. We sample z_k for $k \in \{1, \ldots, L_k\}$ according to:

$$p(z_k|z_{k-1}) = \frac{p(z_k, z_{k-1})}{p(z_{k-1})} \tag{9}$$

$$= \frac{g_{k-1,k}(z_{k-1}, z_k)\overrightarrow{\mu}_{Z_{k-1}}(z_{k-1})\overleftarrow{\mu}_{Z_k}(z_k)}{\overrightarrow{\mu}_{Z_{k-1}}(z_{k-1})\overleftarrow{\mu}_{Z_{k-1}}(z_{k-1})} \tag{10}$$

$$= \frac{g_{k-1,k}(z_{k-1}, z_k)\overleftarrow{\mu}_{Z_k}(z_k)}{\overleftarrow{\mu}_{Z_{k-1}}(z_{k-1})}, \tag{11}$$

Again, this can be expressed in the matrix form

$$\boldsymbol{p}_{\boldsymbol{Z_k}|\boldsymbol{Z_{k-1}}} = \left\{\overleftarrow{\boldsymbol{\mu}}_{\boldsymbol{Z_{k-1}}}\right\}^{-1}_{i_{z_{k-1}}} \overleftarrow{\boldsymbol{\mu}}_{\boldsymbol{Z_k}} \boldsymbol{A}_{:,i_{\breve{z}_{k-1}}}. \tag{12}$$

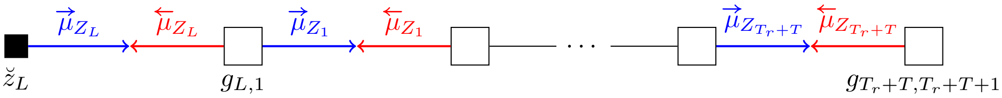

Figure 3: Forward sampling