POPULATION-TEMPORAL CODING

Note: Projects...
Temporal repn (spikes) and population repn (distributed activities) have been independently considered. Both are nonlinear encoding, linear decoding. Stick them together. Encoding:

\[ a_i(x(t)) = G_i[J_i(x(t))] \]
\[ J_i(x(t)) = \alpha_i \left< \tilde{\phi}_i \cdot x(t) \right>_m + J^\text{bias}_i. \]

Decoding:

\[ \hat{x}(t) = \sum_{i,n} \phi_i h(t - t_{in}) \]
PT Filtering

- Finding optimal $h(t)$ as before implicitly includes the population decoder.

- So, always normalize $h(t)$ (optimal or not) to area $= 1$ before using the decoders.

- So, decoders are, more accurately

$$\hat{x}(t) = \sum_{i,n} \phi_i(t - t_{in})$$
Ideal PT decoder

- Should add noise to optimize

\[ \hat{x}(t) = \sum_{i,n} \phi_i(t - t_{in} - \eta_{in}) \]

- Minimize

\[ E = \left\langle \left[ x(t; A) - \sum_{i,n} \phi_i(t - t_{in} - \eta_{in}) \right]^2 \right\rangle_{A,\eta} \]

- Technically should use a Monte Carlo method
we can then use a non-optimal, biologically plausibly temporal decoder (the PSC)

we can find the decoders analytically

we can easily apply other (as yet unseen) analyses which help us understand population representation
Why should it work?

- The decoders are the same because we used a LIF in both cases, so

\[ a_i^{rate}(x) = \langle \sum_n h_i(t) * \delta(t - t_{in}) \rangle_T \]

\[ = \langle \sum_n h_i(t - t_{in}) \rangle_T \]

\[ = \langle a_i^{spiking}(x) \rangle_T \]

- Will be convincing if we can build models well
Population-temporal filter
Noise and precision

- **Pre-noising**
  - Add noise
  - Find filter
  - Decode

- **Post-noising**
  - Find filter
  - Add noise
  - Decode
Fluctuations as noise

- Using spikes results in fluctuations in the estimate that are like more noise (so why spike?)
- Appendix C.1 has details. Postsynaptic activity, under constant input:

$$\alpha_i(x, t) = \sum_n h_i (t - n\Delta_i(x) - t_{i_0})$$

- Variance is:

$$\sigma^2_{\hat{x}(t)} = \left\langle \left[ \hat{x}(t) - \langle \hat{x}(t) \rangle_T \right]^2 \right\rangle_{T,t_{i_0}}$$
Fluctuations

- Variance becomes

\[
\sigma_{\hat{x}(t)}^2 = \sum_i \phi_i^2 a_i(x) \left[ \sum_m g_i(m \Delta_i(x)) - a_i(x) \right]
\]

- Intuitively as \( \tau \) increases:

\[
g_i(\tau) = \int_{-\infty}^{\infty} h_i(t) h_i(t - \tau) dt
\]
Error

- Error comes from 3 sources

\[ E_{\text{total}} = E_{\text{static}} + E_{\text{noise}} + E_{\text{fluctuations}} \]

\[ = \frac{1}{2} \left\langle \left[ x - \sum a_i(x) \phi_i \right]^2 \right\rangle_x + \sigma^2_\eta \sum \phi_i^2 + \sigma^2_{\hat{x}(t)} \]

- Because the last two are the same form, they can be combined into

\[ \sigma^2 \sum \phi_i^2 \]

- The previous noise analysis will work here \((1/N)\)
FEEDFORWARD TRANSFORMATIONS
A communication channel
Connection weights

- Define the representations of both pops:

\[ a_i(x) = G_i \left[ J_i(x) \right] \quad b_j(y) = G_j \left[ J_j(y) \right] \]

\[ = G_i \left[ \alpha_i \tilde{\phi}_i x + J_i^{bias} \right] \quad = G_j \left[ \alpha_j \tilde{\phi}_j y + J_j^{bias} \right] \]

\[ \hat{x} = \sum_i a_i(x) \phi_i^x \quad \hat{y} = \sum_j b_j(y) \phi_j^y \]

- Define the computation: \( y = x \)

- Substitute our estimate of \( x \) into \( b \)
Connection weights

- Substituting:  \( y = x \approx \hat{x} \)

\[
\begin{align*}
  b_j(x) &= G_j \left[ \alpha_j \tilde{\phi}_j x + J_j^{bias} \right] \\
  &= G_j \left[ \alpha_j \tilde{\phi}_j \sum_i a_i(x) \phi_i^x + J_j^{bias} \right] \\
  &= G_j \left[ \sum_i \omega_{ji} a_i(x) + J_j^{bias} \right] \\
  \omega_{ji} &= \alpha_j \tilde{\phi}_j \phi_i^x
\end{align*}
\]
With spikes

- Write the spiking estimate
  \[ \hat{x}(t) = \sum_i a_i(x(t)) \phi_i^x \]
  \[ = \sum_{i,n} h_i(t - t_{in}) \phi_i^x \]

- Then do the same substitution:
  \[ b_j(x(t)) = G_j \left[ \alpha_j \tilde{\phi}_j x(t) + J_j^{bias} \right] \]
  \[ = G_j \left[ \alpha_j \tilde{\phi}_j \sum_{i,n} h_i(t - t_{in}) \phi_i^x + J_j^{bias} \right] \]
  \[ = G_j \left[ \sum_{i,n} \omega_{ji} h_i(t - t_{in}) + J_j^{bias} \right] \]
Scaling and noise

- $x$ and $2x$

In b) the decoders weren’t found under noise
Adding scalars
Volunteer?

\[ c_k(x + y) = G_k \left[ \alpha_k \tilde{\phi}_k (x + y) + J^\text{bias}_k \right] \]

\[ = G_k \left[ \alpha_k \tilde{\phi}_k \left( \sum_i a_i(x) \phi_i^x + \sum_j b_j(y) \phi_j^y \right) + J^\text{bias}_k \right] \]

\[ = G_k \left[ \sum_i \omega_{ki} a_i(x) + \sum_j \omega_{kj} b_j(y) + J^\text{bias}_k \right] \]

\[ \omega_{ki} = \alpha_k \tilde{\phi}_k \phi_i^x \quad \omega_{kj} = \alpha_k \tilde{\phi}_k \phi_j^y \]
Scalar addition
Recipe for linear trans.

1. Define the repn (enc/dec) for all variables involved in the operation.

2. Write the transformation in terms of these variables.

3. Write the transformation using the decoding expressions for all variables except the output variable.

4. Substitute this expression into the encoding expression of the output variable.
Vectors

- Nothing new. Representation:
  \[ a_i(x) = G_i \left[ \alpha_i \left\langle \tilde{\phi}_i x \right\rangle_m + J_i^{bias} \right] \]
  \[ \hat{x} = \sum_i a_i(x) \phi_i^x \]

- Transformation
  \[ z = C_1 x + C_2 y \]
\[ c_k(C_1 x + C_2 y) = G_k \left[ \alpha_k \left< \tilde{\phi}_k(C_1 x + C_2 y) \right>_m + J_k^{bias} \right] \]

\[ = G_k \left[ \alpha_k \left< \tilde{\phi}_k \left( C_1 \sum_i a_i(x) \phi_i^x + C_2 \sum_j b_j(y) \phi_j^y \right) \right>_m + J_k^{bias} \right] \]

\[ = G_k \left[ \sum_i \omega_{ki} a_i(x) + \sum_j \omega_{kj} b_j(y) + J_k^{bias} \right] \]

\[ \omega_{ki} = \alpha_k C_1 \left< \tilde{\phi}_k \phi_i^x \right>_m \quad \omega_{kj} = \alpha_k C_2 \left< \tilde{\phi}_k \phi_j^y \right>_m \]
Vector addition

- a) vector space; b) components
Comments

- Use matrices instead of scalars: \( \omega_{ki} = \alpha_k \langle \tilde{\phi}_k C_1 \phi_i^x \rangle_m \)

- Permits any linear operation (rotation, scaling)

- Spiking neurons: \( a_i(x) = \sum_n h(t - t_{in}) \)

- Does a good job of vector addition (small transient with spikes because of )

- Improve performance by adding neurons

- This kind of network may be used in frontal eye fields for control of saccades.