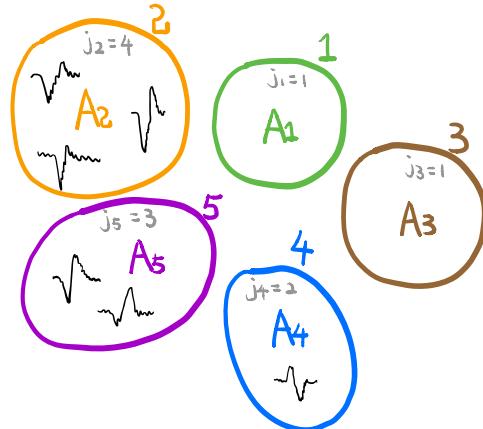


3.2 Learning the templates via stochastic batch optimization

在 initialization 中，每个 cluster 中的 prototypical spike 只是 -T waveform，
但 running average waveform for each cluster 却是 $A_n = \left(\quad \right)_{384 \times 61}$ 。
两者之间什么关系？或者说，是如何建立 spatial template 的？



A_n : running average waveform for cluster n

j_n : 以 $A_n (n=1 \sim N)$ 为各 clusters 中心，当为 batch 中的 spikes 被归类到各 clusters 的数量。

Using a simple running average update rule to optimize template

$$A_n^{\text{new}}(i, t_0) \leftarrow (1 - p)^{j_n} A_n^{\text{old}}(i, t_0)$$

本公司给出的是
对 channel i, time t_0
这个点的 template 更新。
注意, template 的
“相对”时间点为
从 1 ~ 61 ?
0 ~ 60

forgetting constant
the ratio of keeping

j_n 在当前 batch 中，cluster n 中 spikes 数量，

$j_n \uparrow, (1-p)^{j_n} \downarrow, 1-(1-p)^{j_n} \uparrow$

而，cluster n 中 spikes 数量较多时，就更多让这些 spikes 参与到 moving ave 中， $1-(1-p)^{j_n}$ 较大；
而对 A_n^{old} 的影响则较小， $(1-p)^{j_n}$ 较小。注意，我们假设 $p \in [0, 1]$ 。

总体来讲，第一项的含义：

remember ratio • old template 中对应点 (i, t_0) 的幅度值

$$+ (1 - (1 - p)^{j_n}) \sum_{k \in \text{batch}}^{\sigma(k)=n} V(i, s(k) + t_0)$$

k 表示 spikes 的“某种 ID”

↳ $k \in \text{batch}$ 表示属于当前 batch 的 spikes

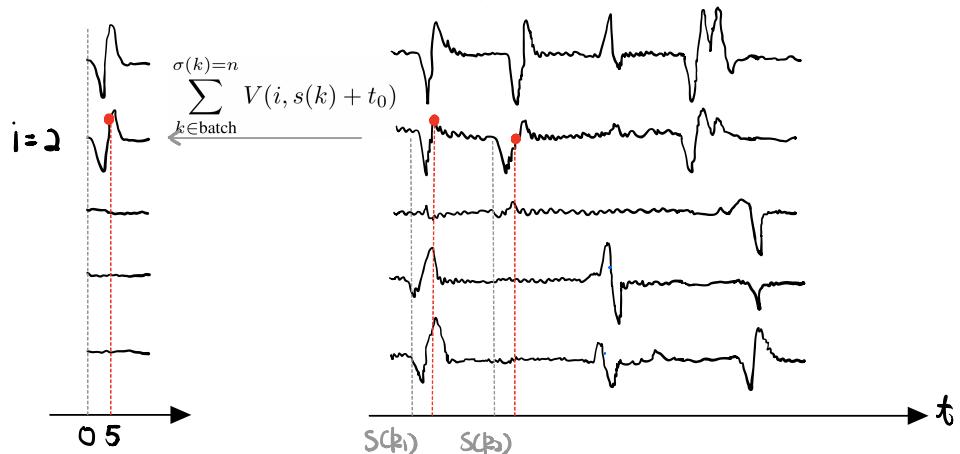
↳ $\sigma(k) = n$ 表示属于 cluster n 的 spikes $s(k), \sigma(k)$

↳ $s(k)$ 表示这些 spikes 出现的时间 在 template learning 中 fixed.

对于当前 batch 属于 cluster n 的每个 spike k , 我们要找到它对应待更新点 $\text{template}(i, t_0)$ 的幅度, 从而通过 running average 贡献于 A_n^{new} 的更新。

例如: update $A_{n=2}^{\text{new}}(i=2, t=5)$

cluster $n=2$ 的 spikes 在 channel $i=2$ 中的 waveforms 有两下, 其 start time 分别为 $s(k_1), s(k_2)$, 而我们想要的时间点为 $s(k_1)+5, s(k_2)+5$. 注意, t 的单位是采样 index, 不是秒。



Cluster n template $A_n(i, t)$

raw recording $V(i, t)$ in one batch,
在 i channel, 每个时间点只有一个
spike waveform. overlapping spike 在 3.1
中被除去了。

总体来讲, 第二项的含义:

参考当前 batch 中被分配到 cluster n 中的 spikes 的 ratio . 对于 $A_n(i, t_0)$ 点的原始信号之和

Note that different clusters will therefore update their mean waveforms at different rates, depending on their number of spikes per batch. Since firing rates vary over two orders of magnitude in typical recordings (from < 0.5 to 50 spikes/s), the adaptive running average procedure allows clusters with rare spikes to nonetheless average enough of their spikes to generate a smooth average template.

j_n 如果较小, 则采用当前 batch 中被分配到 cluster n 中的 spikes 的程度 $[1 - (1 - p)^{j_n}]$ 较小。因为如果该 cluster 中 spikes 少, 其平均的波动会大, 从而使 A_n 更新不平滑。

3.3 Inferring spike times and amplitudes via template matching

The inference step of the proposed model attempts to find the best spike times, cluster assignments and amplitudes,

infer: 每个 spike 出现在时间点 $s(k)$,

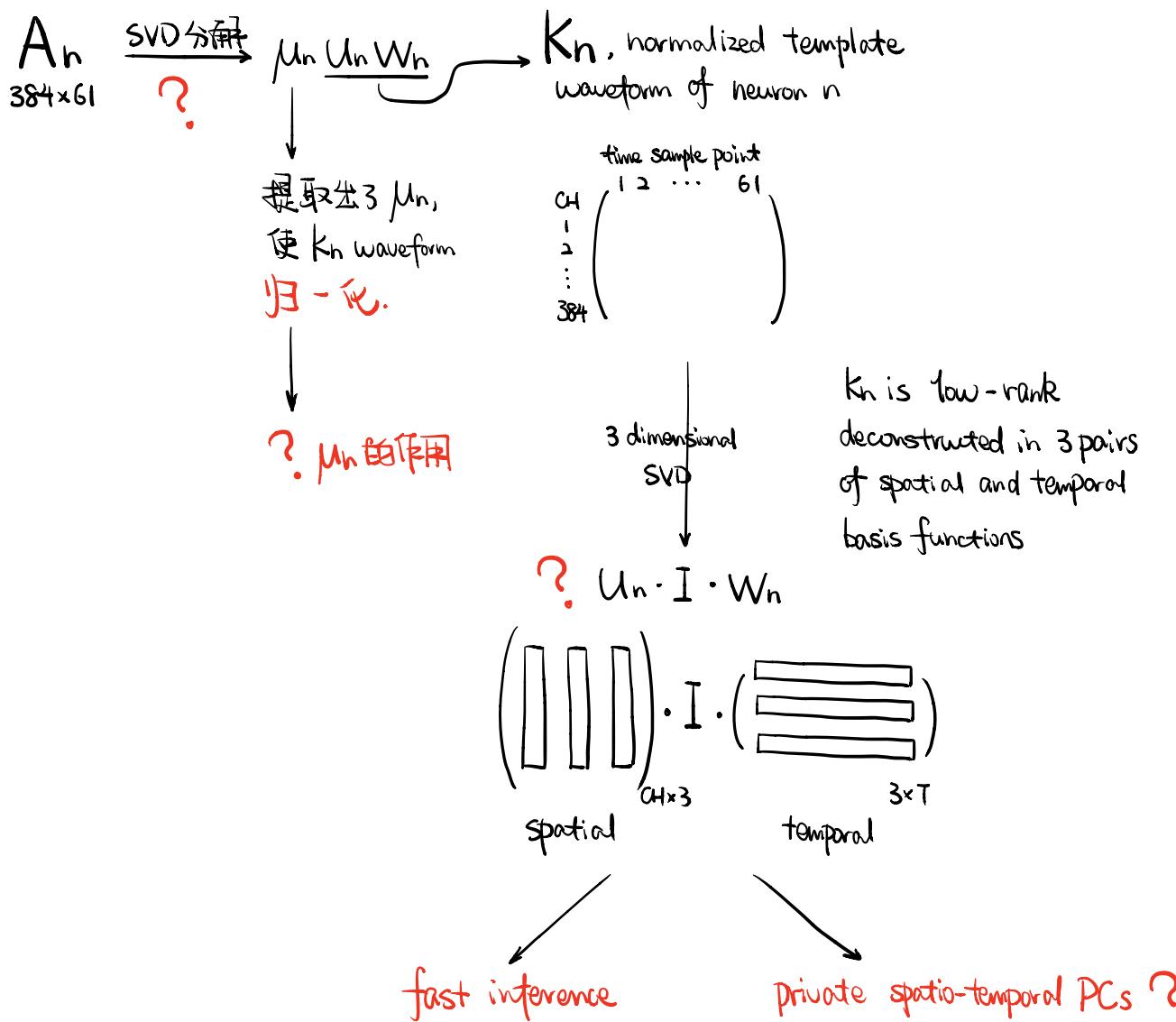
每个 spike 属于一个 cluster (or best match template) $\sigma(k)$, with fixed $\{K_n\}_n$

用 best match template 找合适 spike 时的最优幅度 a_k .

given a set of templates $\{K_n\}_n$ with low rank-decompositions $K_n = U_n W_n$ and mean amplitudes μ_n . [The templates are obtained from the running average waveform A_n , after an SVD decomposition to give $A_n \sim \mu_n K_n = \mu_n U_n W_n$, with $\|U_n W_n\| = 1$, with U_n orthonormal and W_n orthogonal.] The primary roles of the low-rank representation are to guarantee fast inferences and to regularize the waveform model. And to analyze the private PCs.

计算 $V * K_n$

K_n 的归一化、普适性



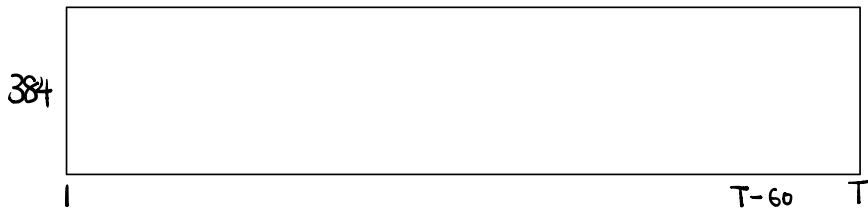
$$\begin{aligned} V * K_n &= V * (U_n W_n) \\ &= \sum_j (U_n(:, j)^T \cdot V) * W_n(j, :), \end{aligned}$$

2. Standard Matching Pursuit

已知 ① N 个 clusters 分别对应的 templates K_n ($n = 1 \sim N$)

cluster 1	2	...	N
k_1	k_2	...	k_N
(\quad)	(\quad)	...	(\quad)
$_{384 \times 61}$	$_{384 \times 61}$		$_{384 \times 61}$

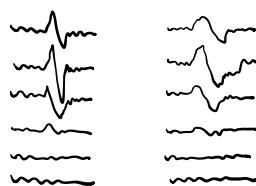
② - T batch 中的 raw recording $V(c_i, t)$



[问]: which template is the best fitting template?

Iteration j

for $n = 1 : N$ 对于每- T template k_n
 for $t = 1 : T-60$ 对于每- T 与 template k_n 状相同的信号片段 $V(t : t+60)$
 k_n 与 $V(t : t+60)$ 进行拟合



通过变换 k_n 的幅度 $\chi(n, t)$ 来找到最优拟合。

这是-「对 $\chi(n, t)$ 的优化问题」

This is equivalent to minimizing a standard quadratic function of the form $ax^2 - 2bx + c$ over the scalar variable x , with a , $-2b$ and c derived as the coefficients of x^2 , x and 1 from equation 2

$$a = 1 + \frac{\epsilon}{\lambda \mu_n^2}; \quad b = (K_n \star V)(t) + \frac{\epsilon}{\lambda \mu_n}; \quad c = \lambda \mu_n^2, \quad (4)$$

$$\text{MIN } a(n) \cdot x^2 - 2b(n, t) \cdot x + c(n)$$

$$\chi_{\text{best}}(n, t) = \frac{b(n, t)}{a(n)}$$

优化出 $X_{best}(n, t)$ 后，将 $X_{best}(n, t) \cdot k_n$ 加到 $V(t : t+60)$



计算上述 modeling V 所带来的 cost 的下降， $dC(n, t)$

"the decrease in Cost $dC(n, t)$ that would occur if a spike of neuron n (template waveform of cluster n , k_n) were added at time t "

$$dC(n, t) = -\frac{b(n, t)^2}{a(n)} - C(n)$$

写入 Amplitude matrix 对应位置

$$X(n, t) = X_{best}(n, t)$$

写入 Cost improvement matrix 对应位置

$$dC(n, t) = dC(n, t)$$

重复整个 for 循环，求出完整 $dC(n, t)$

To find the best fitting template, we estimate for each time t and each template n , the decrease in the cost function obtained by introducing template n at location t , with the best-fitting amplitude x .

找到 $dC(n, t)$ 中元素的 max 值所对应的 spike time t_i ，cluster assignment n_j

$$dC(n, t) = n_j \left(\max_{t_j} \right)$$

以及对应的最优幅度

$$X(n, t) = n_j \left(\max_{t_j} \right)$$

所以， k_{n_j} is the best fitting template at time t_i in $V(i, t)$ ，对应 fitting 幅度为 $X_{best}(n_j, t_i)$ 。

从另一个角度来看，我们在 $k_n (n=1 \sim N)$ fix 的情况下，进行 inference。推断出了在 spike time t_i 有一个 neuron n_j 的 spike，可用 $X_{best}(n_j, t_i) \cdot k_{n_j}$ 建模。即 infer 3

- 组 spike time → spike cluster → spike fitting amplitude.

接下来，从 $V_{C_i}(t_n; t_{n_j}; t_{n_j} + 60)$ 减去，基于 k_{n_j} 的 best reconstruction 并执行 iteration $j+1$ 。

In standard matching pursuit, the best fitting template is identified over the entire batch, its best reconstruction is subtracted from the raw data, and then the next best fitting template is identified, iteratively until the amount of explained variance falls below a threshold, which constitutes the stopping criterion.

?

3. Parallel Matching Pursuit

We adopt a parallelized matching pursuit algorithm to iteratively estimate the best fitting templates and subtract them off from the raw data.

在 Standard matching pursuit 中，我们已知 batch data $V_{C_i}(t)$ ，并在 templates $k_n (n=1 \sim N)$ fix 的情况下，在一个 iteration 中求得

$$dC(n, t) = \begin{matrix} \text{cluster} \\ \text{num} \end{matrix} \begin{pmatrix} & \text{time sample} \\ \vdots & \end{pmatrix}_N$$

$$X(n, t) = \begin{matrix} \text{cluster} \\ \text{num} \end{matrix} \begin{pmatrix} & \text{time sample} \\ \vdots & \end{pmatrix}_N$$

Consider the cost improvement matrix $dC(n, t)$. When the largest element of this matrix is found and the template subtracted, no values of dC need to change except those very close in time to the fitted template (t_s samples away).

$$dC(n, t) = \begin{matrix} \text{cluster} \\ \text{num} \end{matrix} \begin{pmatrix} & \text{time sample} \\ \vdots & \end{pmatrix}_N$$

values won't change in the next iteration.
because the signals here are not changed!

因为 $V_{C_i}(t)$ 中，只有 t_j 周围的信号被减去了 $X(n_j, t_j) \cdot k_{n_j}$ ，其余部分未发生变化。
因此，在下一个 iteration $j+1$ 中， $k_n (n=1 \sim N)$ 与未改变区域的 fitting 结果（如下面绿色区域所示）不变。

$$dC(n, t) = \left(\begin{array}{c|c} \text{cluster num} & \text{time sample} \\ \hline 1 & \text{[green]} \\ \vdots & | \\ N & \text{[green]} \end{array} \right)$$

t_s
 t_j

$$X(n, t) = \left(\begin{array}{c|c} \text{cluster num} & \text{time sample} \\ \hline 1 & \text{[green]} \\ \vdots & | \\ N & \text{[green]} \end{array} \right)$$

t_s
 t_j

Thus, instead of finding the global maximum of dC , we can find local maxima above the threshold criterion, and impose a minimal distance (t_s) between such local maxima. The identified spikes can then be processed in parallel without affecting each other's representations.

$$dC(n, t) = \left(\begin{array}{c|c} \text{cluster num} & \text{time sample} \\ \hline 1 & \text{[red]} \\ 2 & \text{[red]} \\ 3 & \text{[red]} \\ 4 & \text{[red]} \\ 5 & \text{[red]} \\ \vdots & | \\ N & \text{[red]} \end{array} \right)$$

t_s
 t_s
 t_s
 t_s
 t_s
 t_s
 t_s
 t_s

$$X(n, t) = \left(\begin{array}{c|c} \text{cluster num} & \text{time sample} \\ \hline 1 & \text{[red]} \\ 2 & \text{[red]} \\ 3 & \text{[red]} \\ 4 & \text{[red]} \\ 5 & \text{[red]} \\ \vdots & | \\ N & \text{[red]} \end{array} \right)$$

t_s
 t_s
 t_s
 t_s
 t_s
 t_s
 t_s
 t_s

在 iteration 中，可以在不同 local areas (以 t_s 间隔) 找到各自的 best fitting template (可重复)，从而一次性地，并行地 infer 多对
 spike time \rightarrow spike cluster \rightarrow spike fitting amplitude。

4. $dC(n, t), X(n, t)$ 的求解

$dC(n, t)$ 与 $X(n, t)$ 的求解并不是利用上述 for loop 遍历 n, t 换个求出来的。由公式

$$dC(n, t) = \frac{b(n, t)}{a(n)} - c(n)$$

$$X_{\text{best}}(n, t) = \frac{b(n, t)}{a(n)}$$

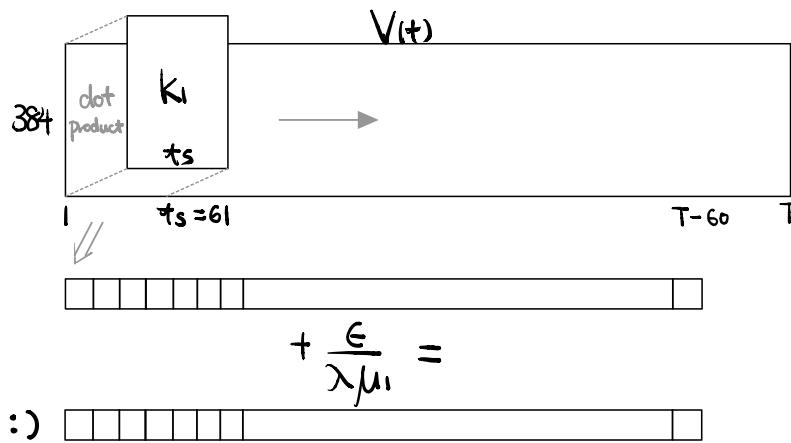
可知，只要求出了以 n and/or t 为自变量的

$$a = 1 + \frac{\epsilon}{\lambda \mu_n^2}; \quad b = (K_n \star V)(t) + \frac{\epsilon}{\lambda \mu_n}; \quad c = \lambda \mu_n^2,$$

就求出了 matrix $dC(n, t)$ 与 $X(n, t)$ 。

具体来讲：

- $a(n), c(n)$ 的求解较为简单：只需要知道 hyper-params ϵ, λ 和由 $A_n \sim \mu_n k_n$ 得出不同 clusters 的 μ_n 即可。
- 对于 $b(n, t)$ ，
for $n = 1, k_1 * V(t)$ 是将 k_1 在 $V(t)$ 上滑动，并逐个时间点计算点积。



同样, for $n = 2:N$

$$b(2, :) \quad \vdots \quad b(N, :) \quad \vdots$$

The diagram shows the computation of $b(n, t)$ for $n = 2:N$. It consists of two parts: a vertical vector of size 384 and a horizontal vector $V(t)$ of size $T - 60$. The result is a horizontal vector of size T . Below this, a formula shows the addition of a term involving $\frac{\epsilon}{\lambda \mu_1}$ to the vector $b(1, :)$.

由此, 求得 $b(n, t)$ 。

在求 $dC(n, t), X(n, t)$ 的过程中, $b(n, t)$ 的求解最为关键;而在求 $b(n, t)$ 的过程中, 滑动计算 dot product 最关键。

The iterative optimization stops when a pre-defined threshold criterion on dC is larger than all elements of dC 。