Generalized linear models for cracking the neural code

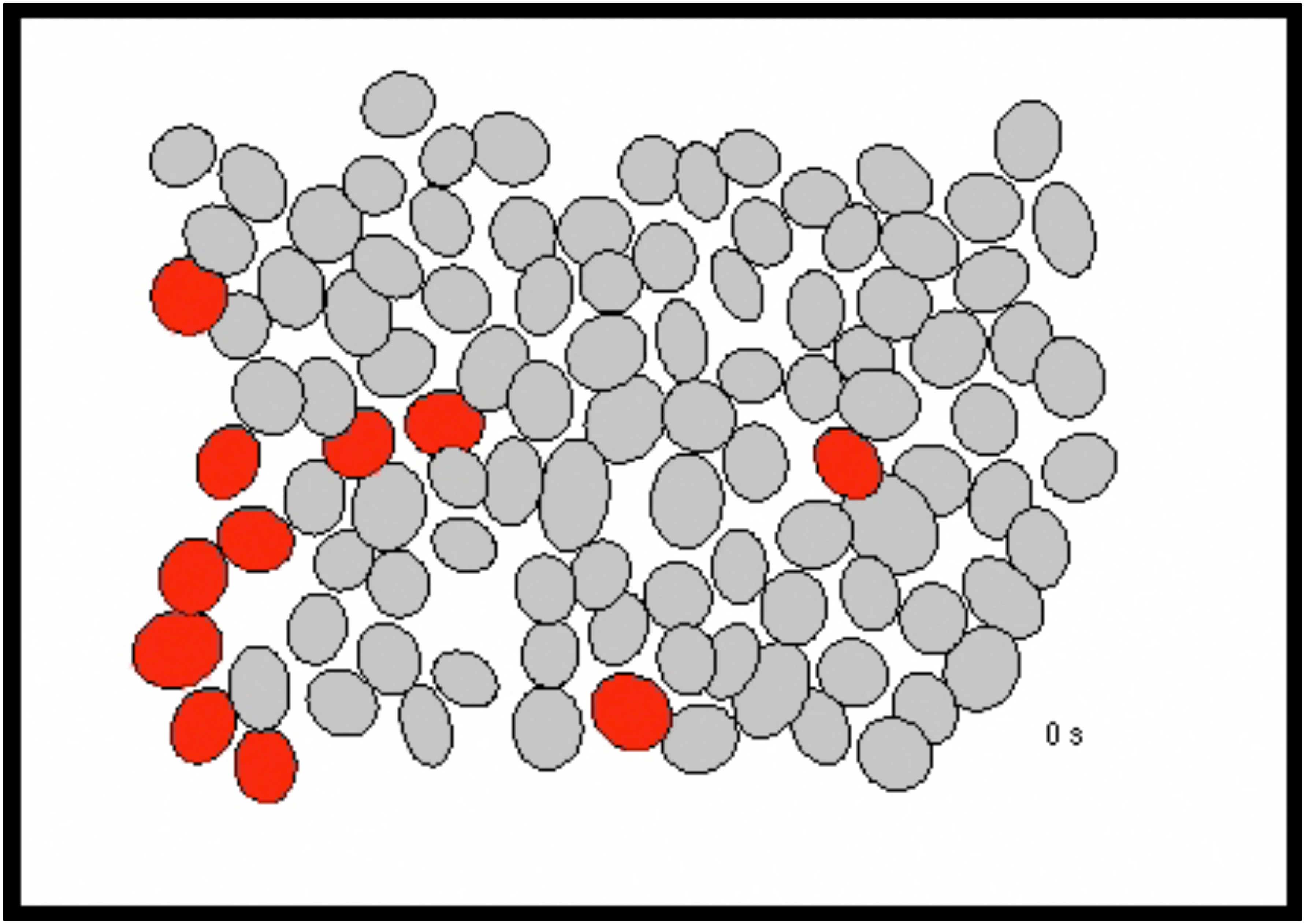
Jonathan Pillow

Princeton Neuroscience Institute, Psychology,
Center for Statistics & Machine Learning
Princeton University

Data Science and Data Skills for Neuroscientists
SFN Short Course
Nov 11, 2016

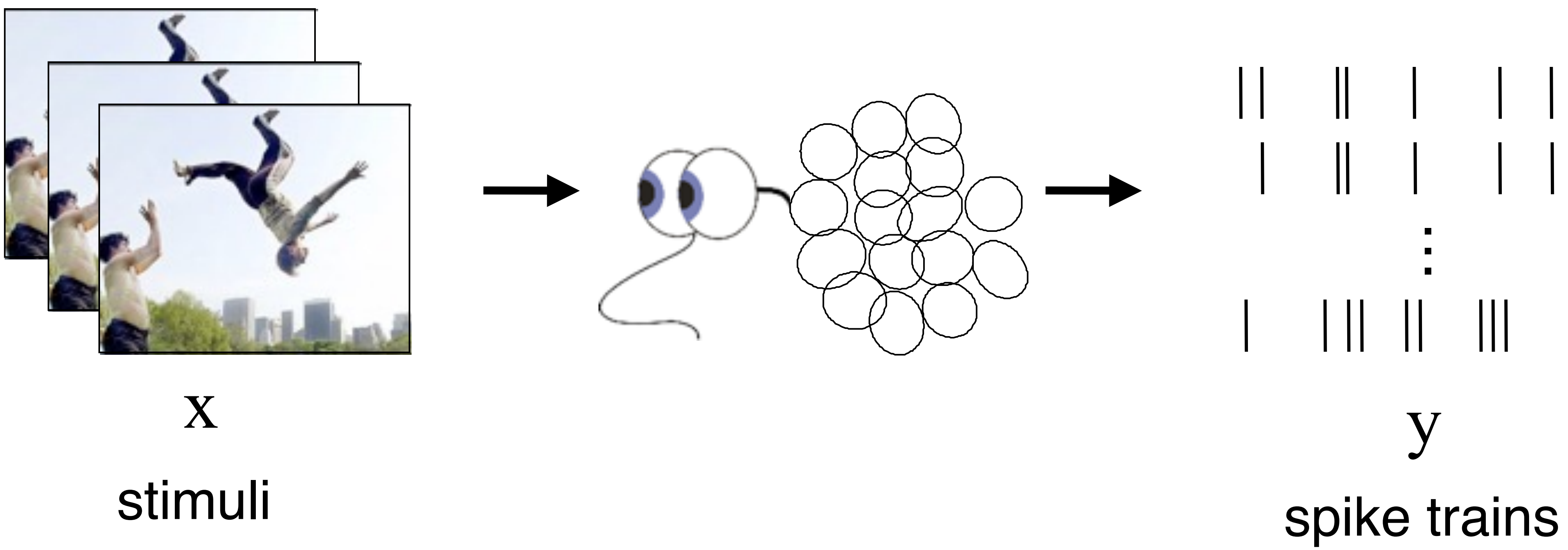
Retinal responses to white noise

(ON parasol cells)



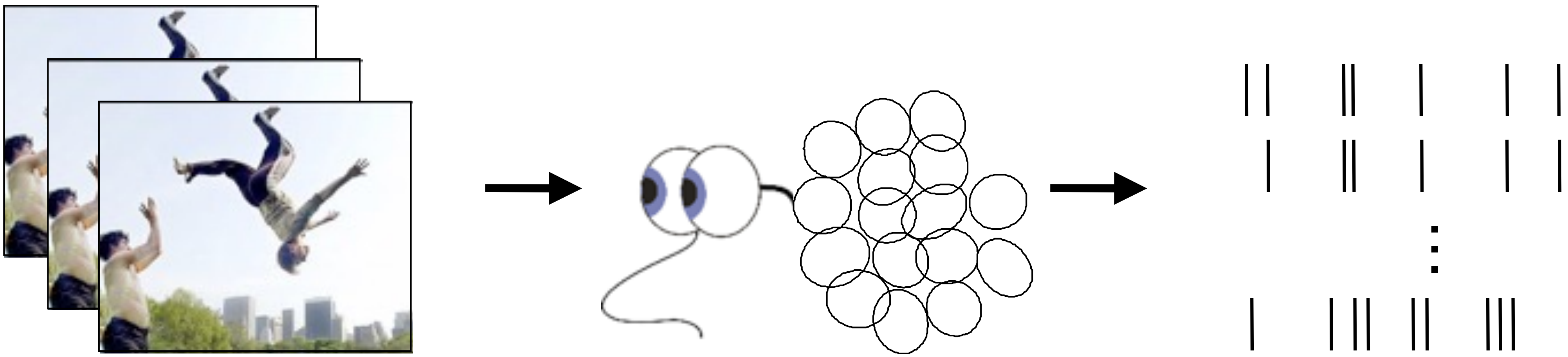
Shlens, Field, Gauthier, Greschner, Sher , Litke & Chichilnisky (2009).

neural coding problem



Q: what is the probabilistic relationship between stimuli and spike trains?

neural coding problem



x

stimuli

$$P(y_1, y_2, \ldots, y_n | x, \theta)$$

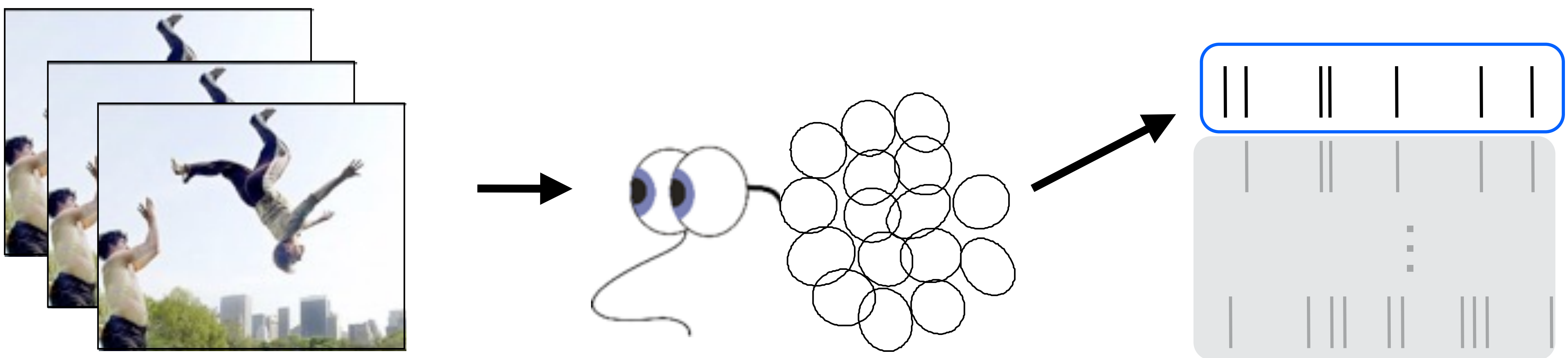
"encoding model"

y

spike trains

Q: what is the probabilistic relationship between stimuli and spike trains?

first block: single-neuron encoding



x

stimuli

$$P(y_i|\vec{x},\theta)$$

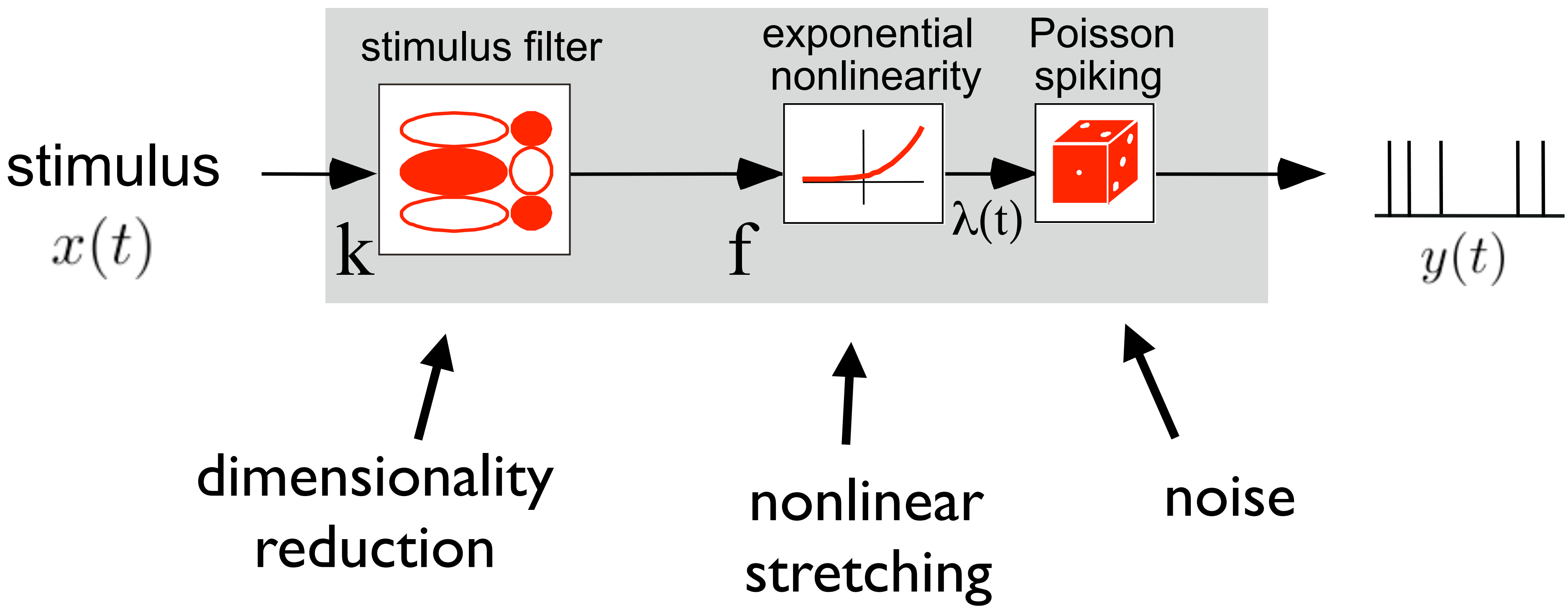
“encoding model”

y

spike trains

“basic” Poisson GLM

Linear-Nonlinear-Poisson model



spike rate $\lambda = f(\vec{k} \cdot \vec{x})$

spike count $y \sim \text{Poiss}(\lambda)$

What is a GLM?

Be careful about terminology:

GLM $\neq$ GLM

General Linear Model

Generalized Linear Model

(Nelder 1972)

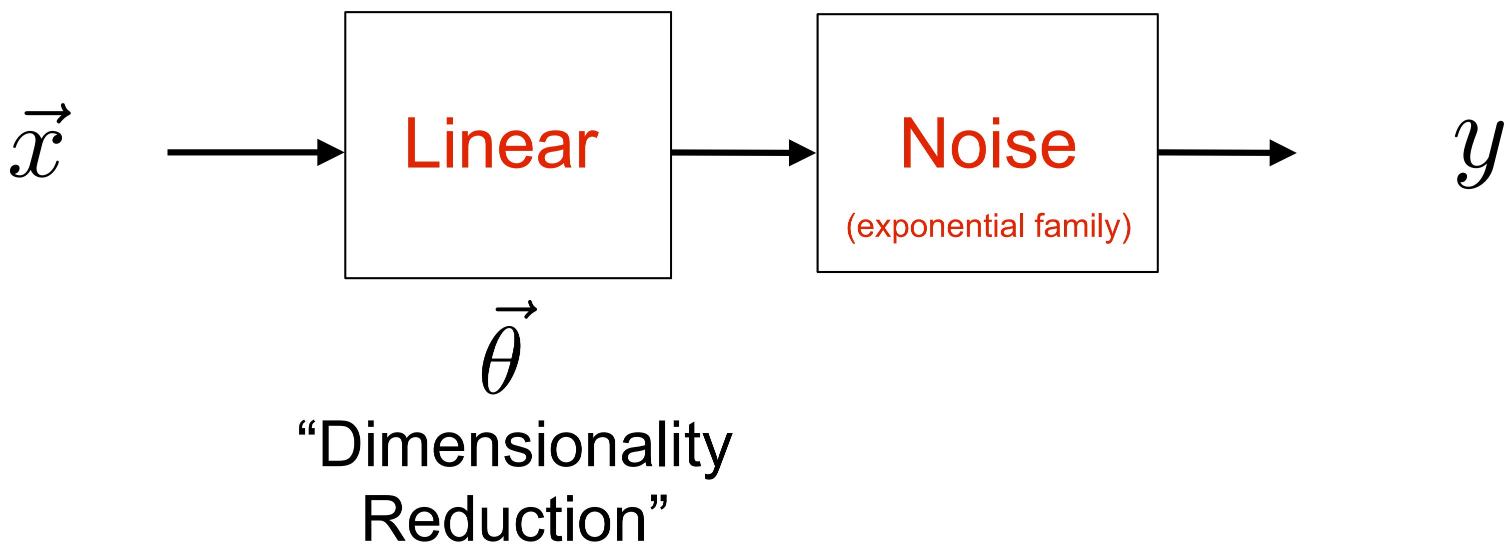
Linear

~~Linear~~

Moral:

Be careful when naming your model!

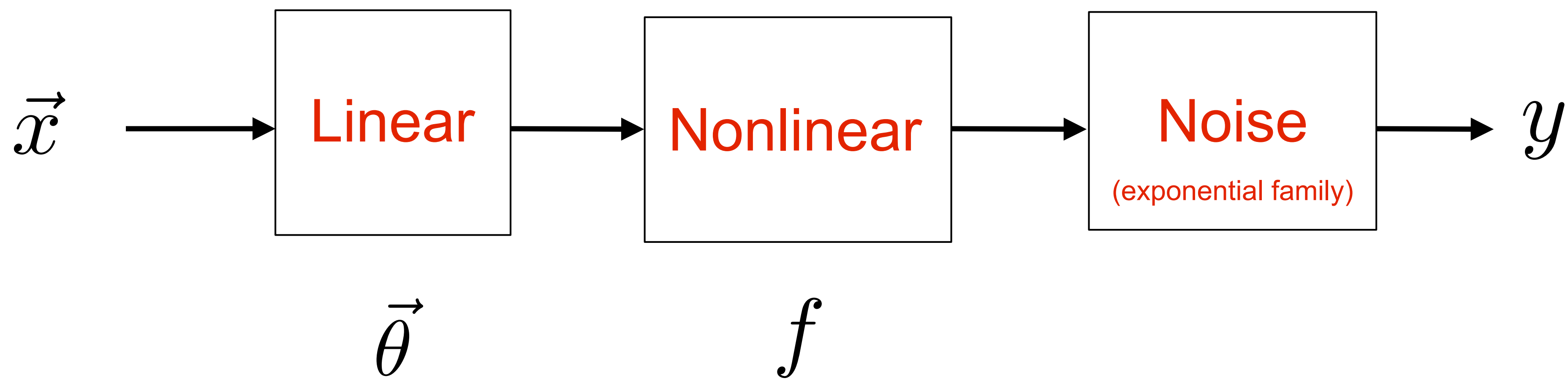
1. General Linear Model



Examples:

1. Gaussian $\quad y = \vec{\theta} \cdot \vec{x} + \epsilon$
2. Poisson $\quad y \sim \mathrm{Poiss}(\vec{\theta} \cdot \vec{x})$

2. Generalized Linear Model

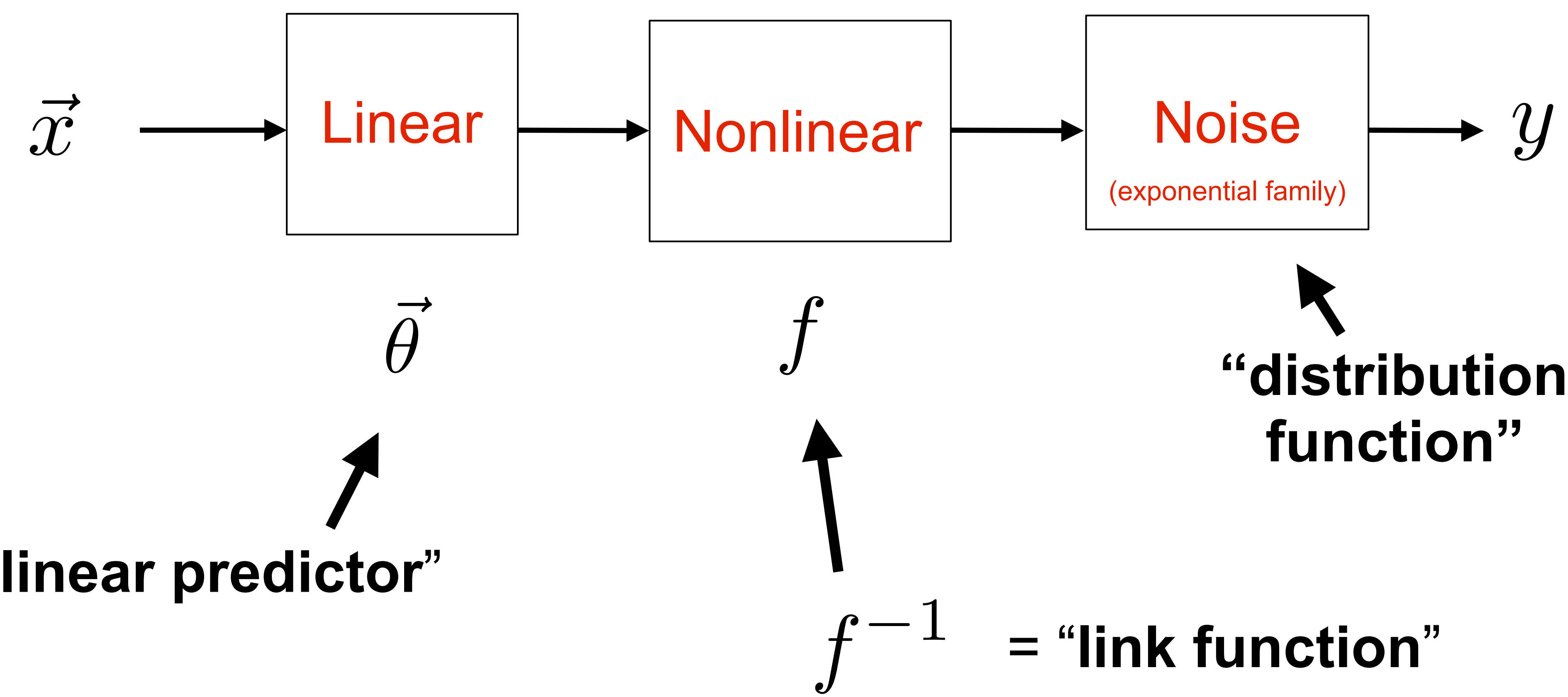


Examples:

1. Gaussian $$y = f(\vec{\theta} \cdot \vec{x}) + \epsilon$$
2. Poisson $$y \sim \mathrm{Poiss}(f(\vec{\theta} \cdot \vec{x}))$$

2. Generalized Linear Model

Terminology:



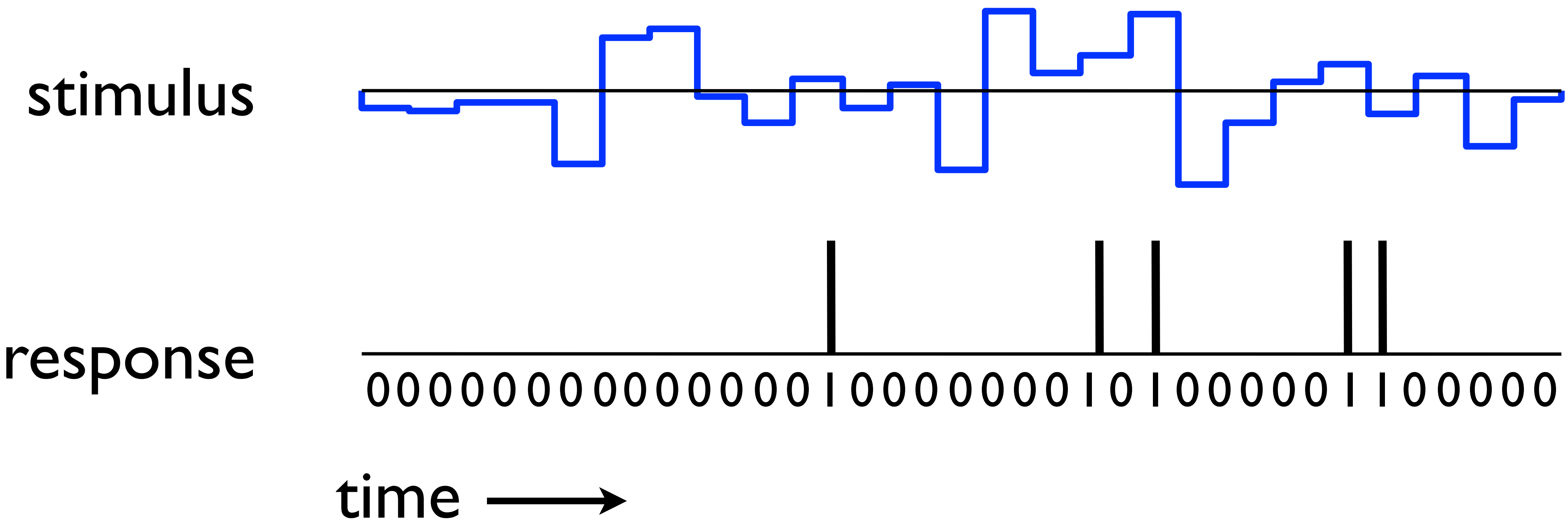
Applying it to data

response
at time t

$$y_t = \vec{k} \cdot \vec{x}_t \; + \; \text{noise}$$

linear
filter

vector stimulus
at time t



response
at time t

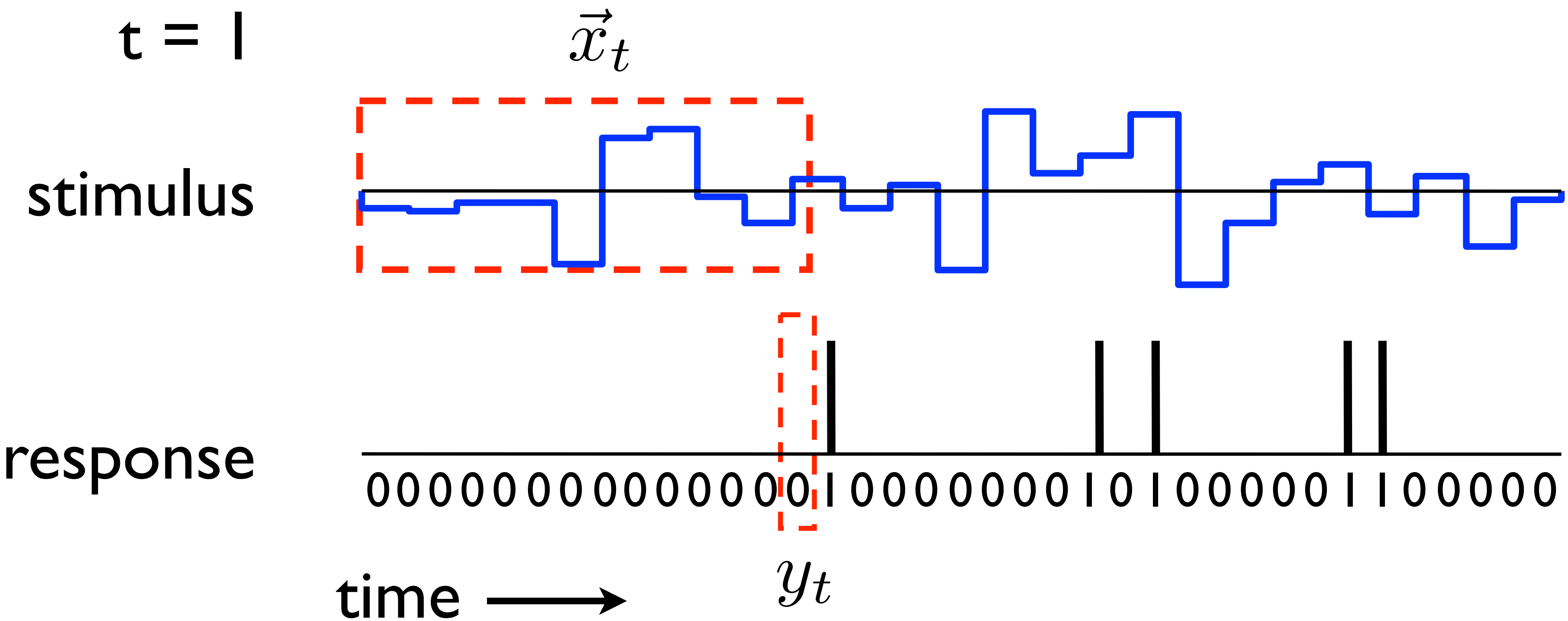
$$y_t = \vec{k} \cdot \vec{x}_t \; + \; \text{noise}$$

linear
filter

vector stimulus
at time t

walk through the data
one time bin at a time

t = 1



response
at time t

$$y_t = \vec{k} \cdot \vec{x}_t \; + \; \text{noise}$$

linear
filter

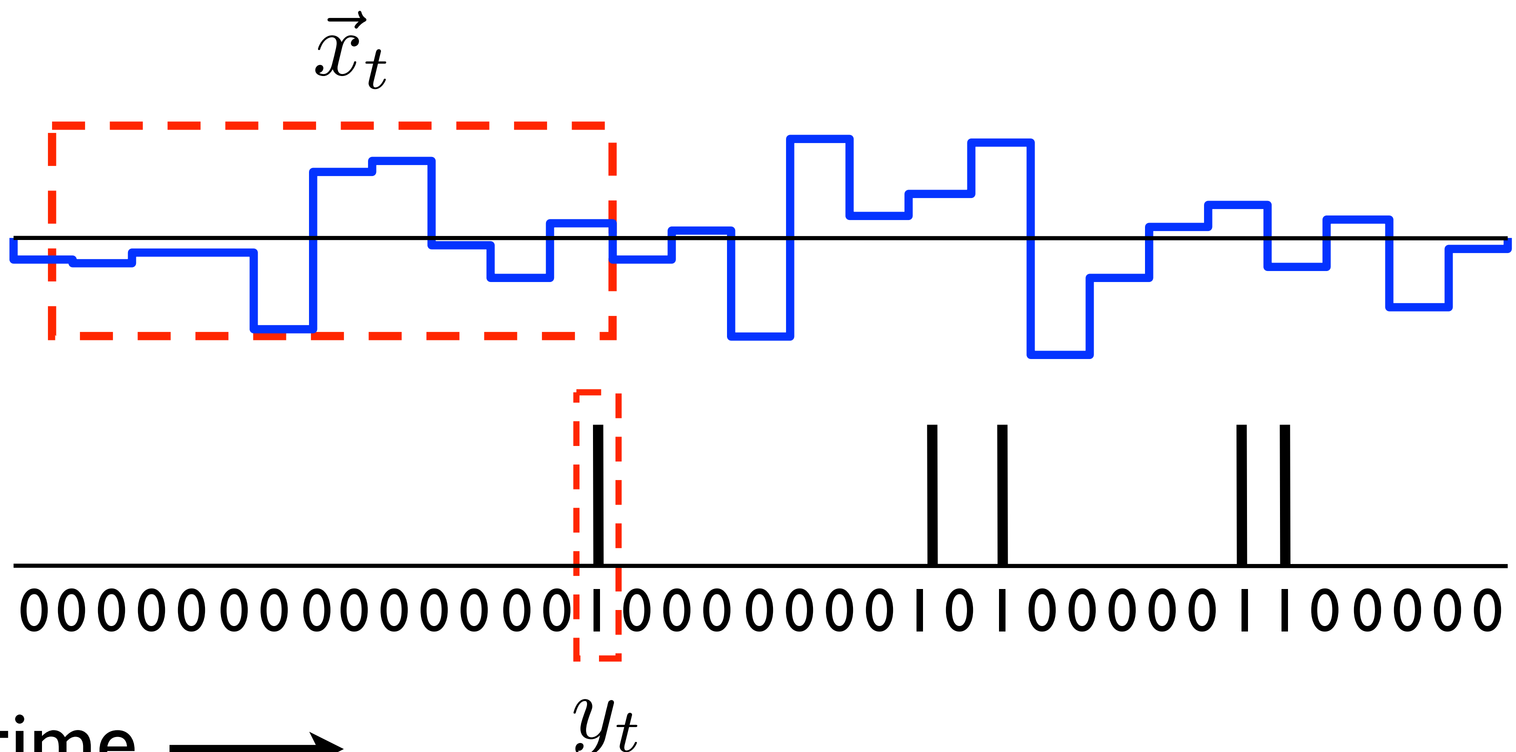
vector stimulus
at time t

walk through the data
one time bin at a time

t = 2

stimulus

response



time ⟶

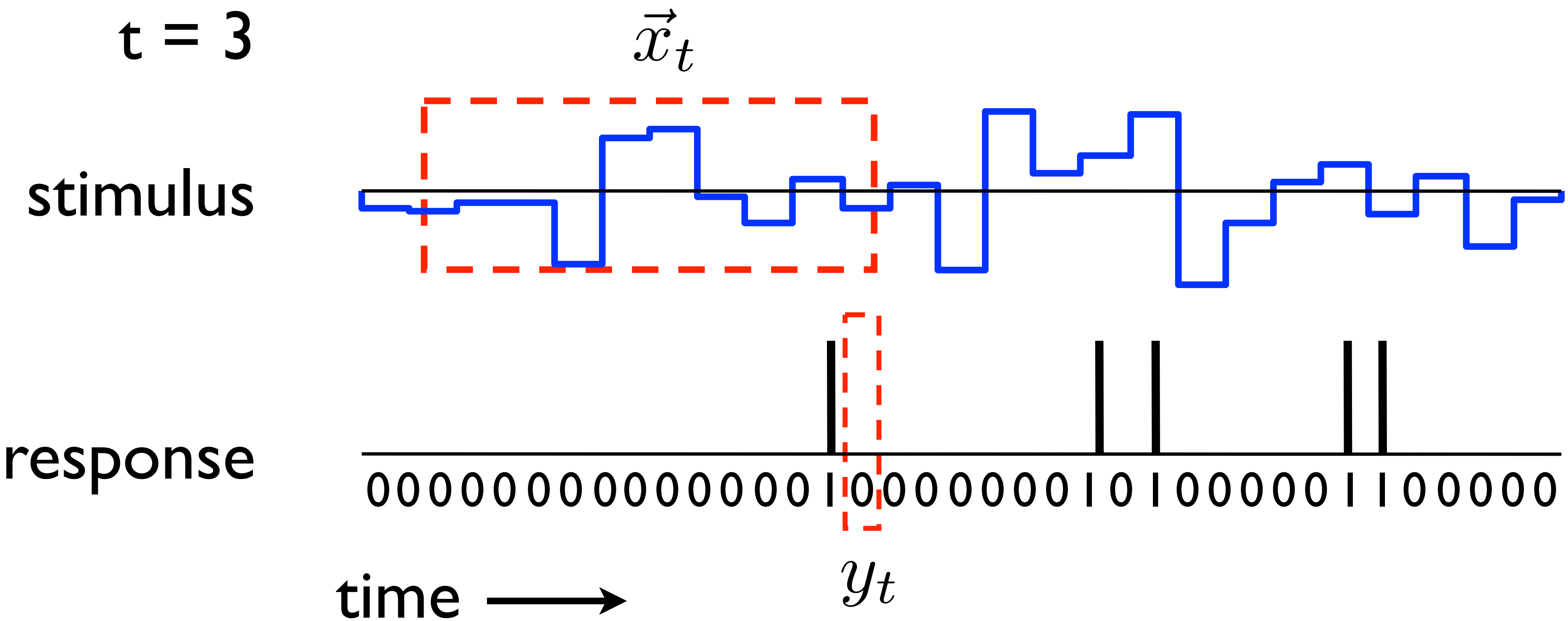
response
at time t

$$y_t = \vec{k} \cdot \vec{x}_t \; + \; \text{noise}$$

linear
filter

vector stimulus
at time t

walk through the data
one time bin at a time



response
at time t

$$y_t = \vec{k} \cdot \vec{x}_t \; + \; \text{noise}$$

linear
filter

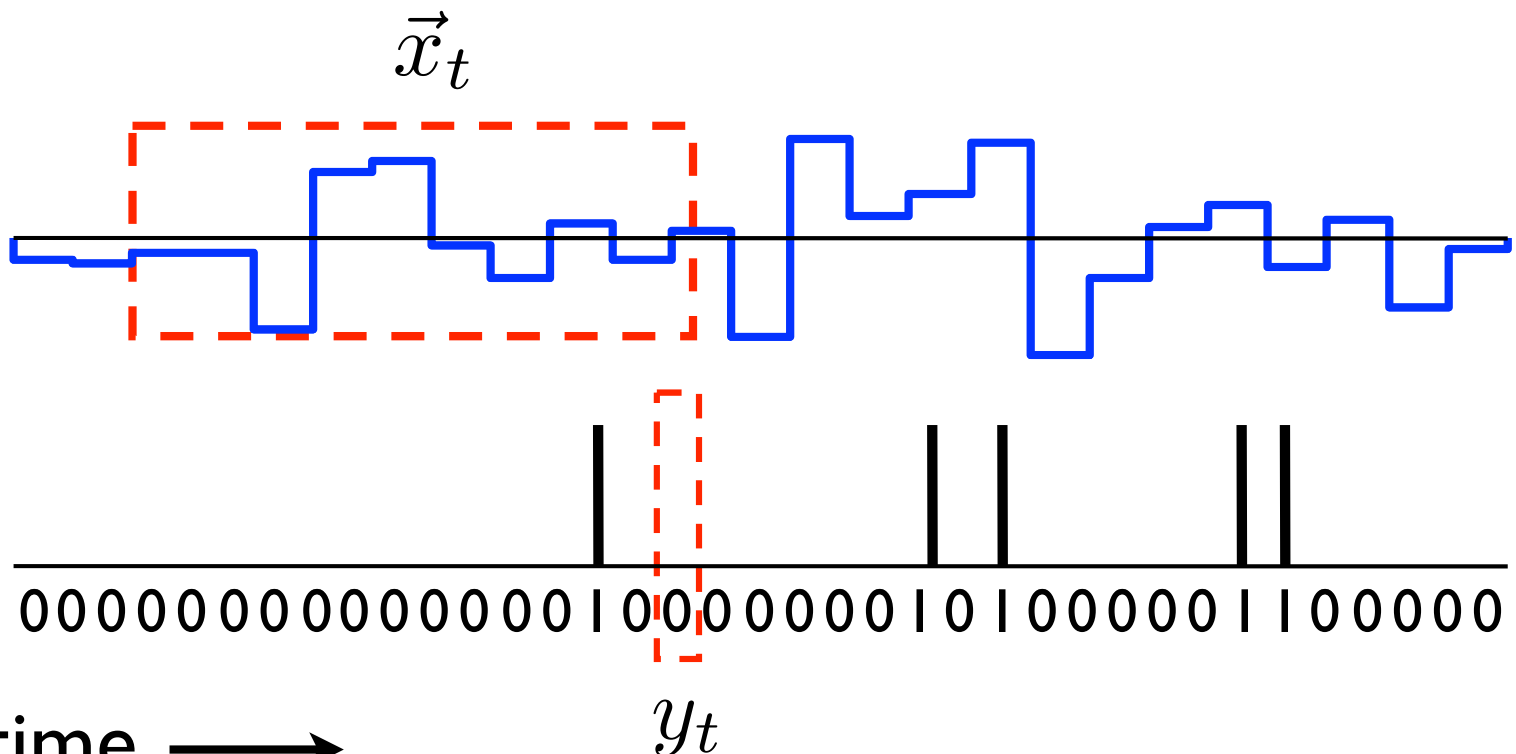
vector stimulus
at time t

walk through the data
one time bin at a time

t = 4

stimulus

response



time ⟶

response
at time t

$$y_t = \vec{k} \cdot \vec{x}_t \; + \; \text{noise}$$

linear
filter

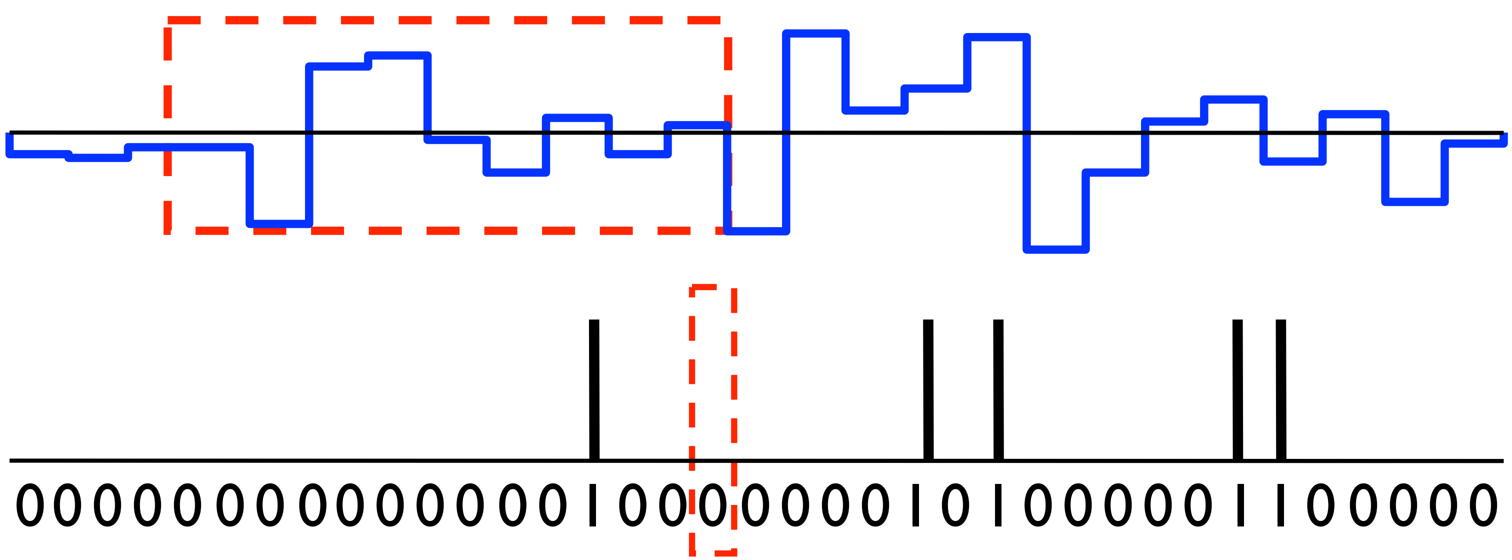
vector stimulus
at time t

walk through the data
one time bin at a time

t = 5

$\vec{x}_t$

stimulus



response

y_t

time →

response
at time t

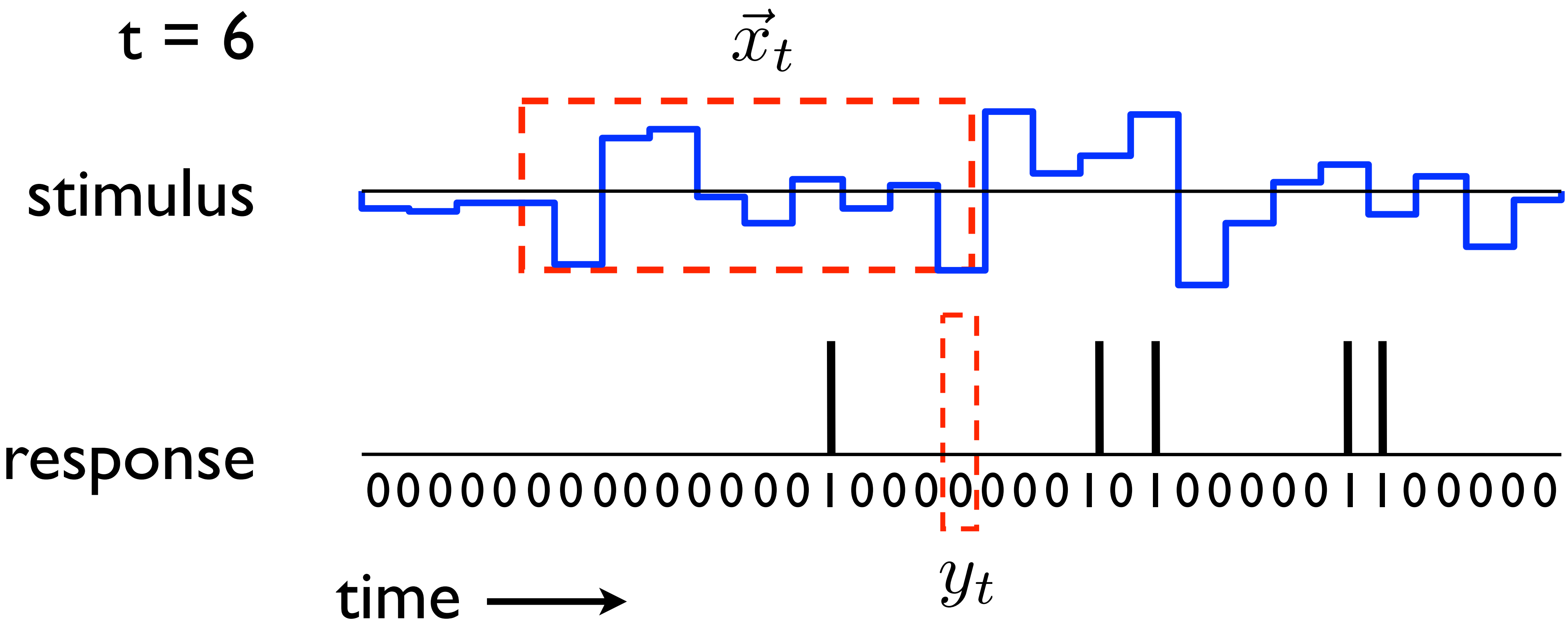
$$y_t = \vec{k} \cdot \vec{x}_t + \text{noise}$$

linear
filter

vector stimulus
at time t

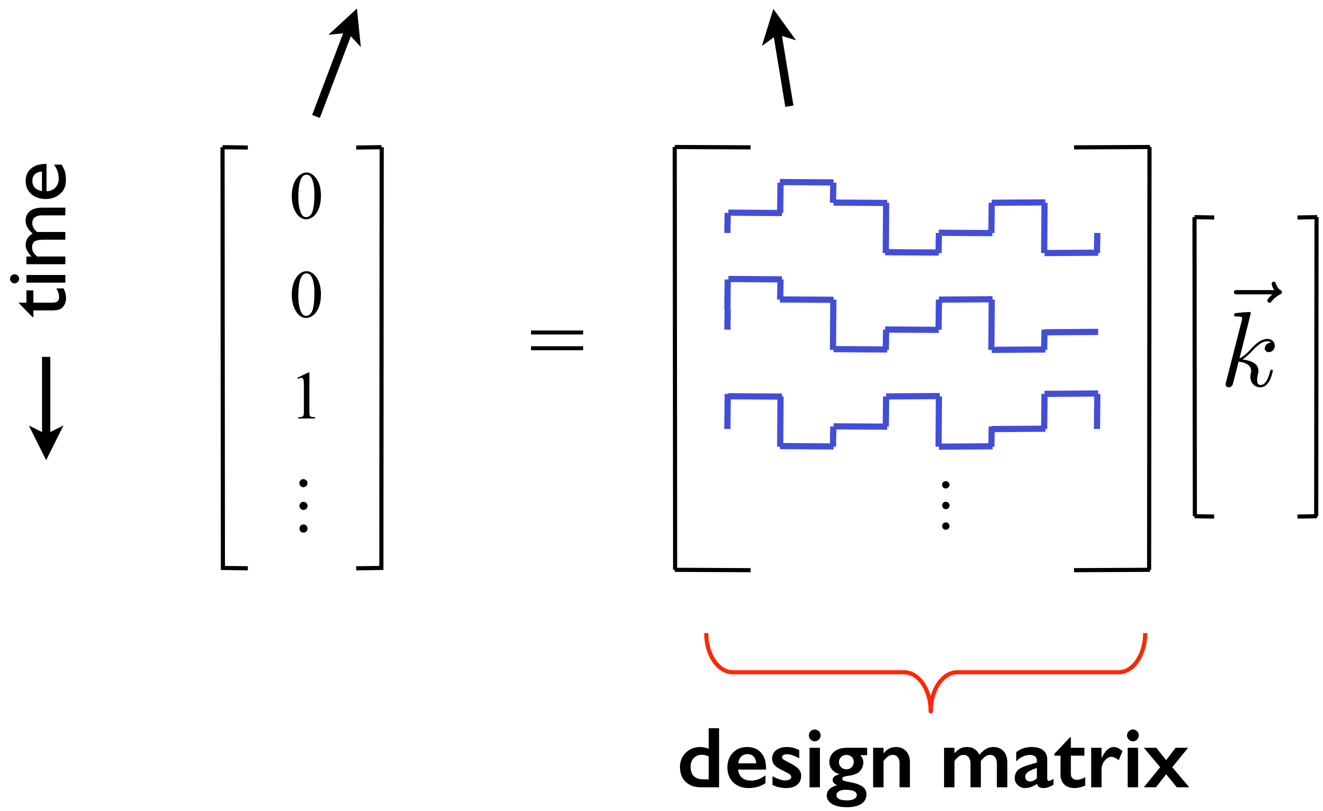
walk through the data
one time bin at a time

t = 6



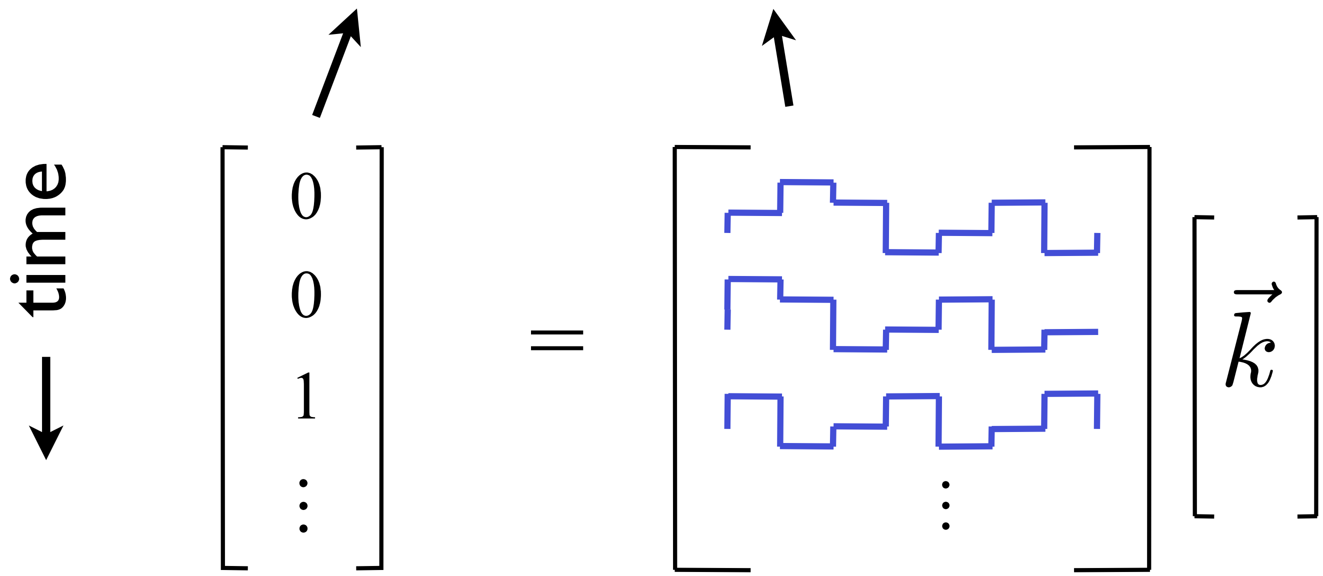
Build up to following matrix version:

$$Y \quad = \quad X\vec{k} \quad + \quad noise$$



Build up to following matrix version:

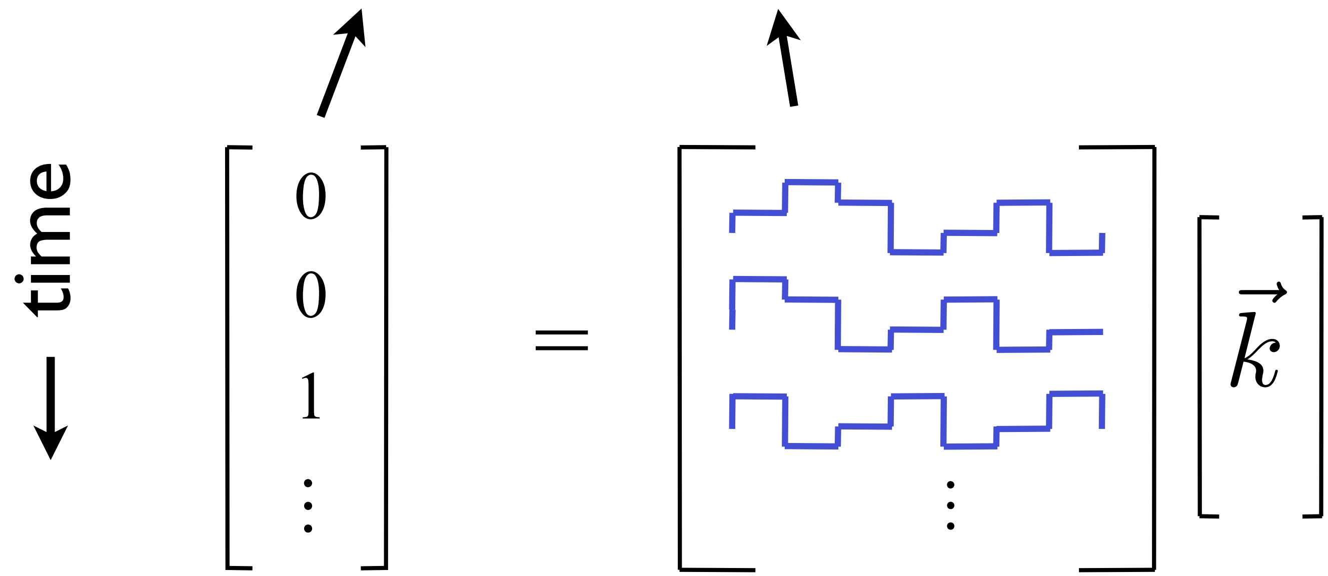
$$Y \quad = \quad X\vec{k} \quad + \quad noise$$



1.“Linear-Gaussian” GLM: $$\hat{k} = (\underbrace{X^T X}_{\text{stimulus covariance}})^{-1} \underbrace{X^T Y}_{\text{spike-triggered avg (STA)}}$$

Build up to following matrix version:

$$Y \quad = \quad X\vec{k} \quad + \quad noise$$



1. "Linear-Gaussian" GLM: $\hat{k} = (X^T X)^{-1} X^T Y$

2. Poisson GLM: `k = glmfit(X,Y,'Poisson');`

maximum likelihood fit (assumes exponential nonlinearity by default)

Now do it: tutorial1_PoissonGLM.m

1. cd into correct directory (‘pillow_GLMtutorials’)
2. open tutorial: use `ctrl-enter`, `ctrl-down`

Dataset:

- retinal ganglion cell spike trains [Uzzell & Chichilnisky 2004]
- 2 OFF cells, 2 off cells
- full-field, binary white noise stimuli

Now do it: tutorial1_PoissonGLM.m

1. cd into correct directory ('pillow_GLMtutorials')
2. open tutorial: use `ctrl-enter`, `ctrl-down`

Topics:

1-2. load data, bin spike trains
3. build design matrix (slow and fast versions)
4a. compute STA (for vizualization)
4b. <u>*linear-Gaussian GLM*</u> ("least-squares regression")
5. <u>*Poisson GLM*</u>: fit assuming exponential nonlinearity
6. <u>*Poisson GLM*</u>: non-parametric estimate of nonlinearity
7. model comparison: log-likelihood & mutual information
8. model comparison: AIC
9. simulate from fitted model

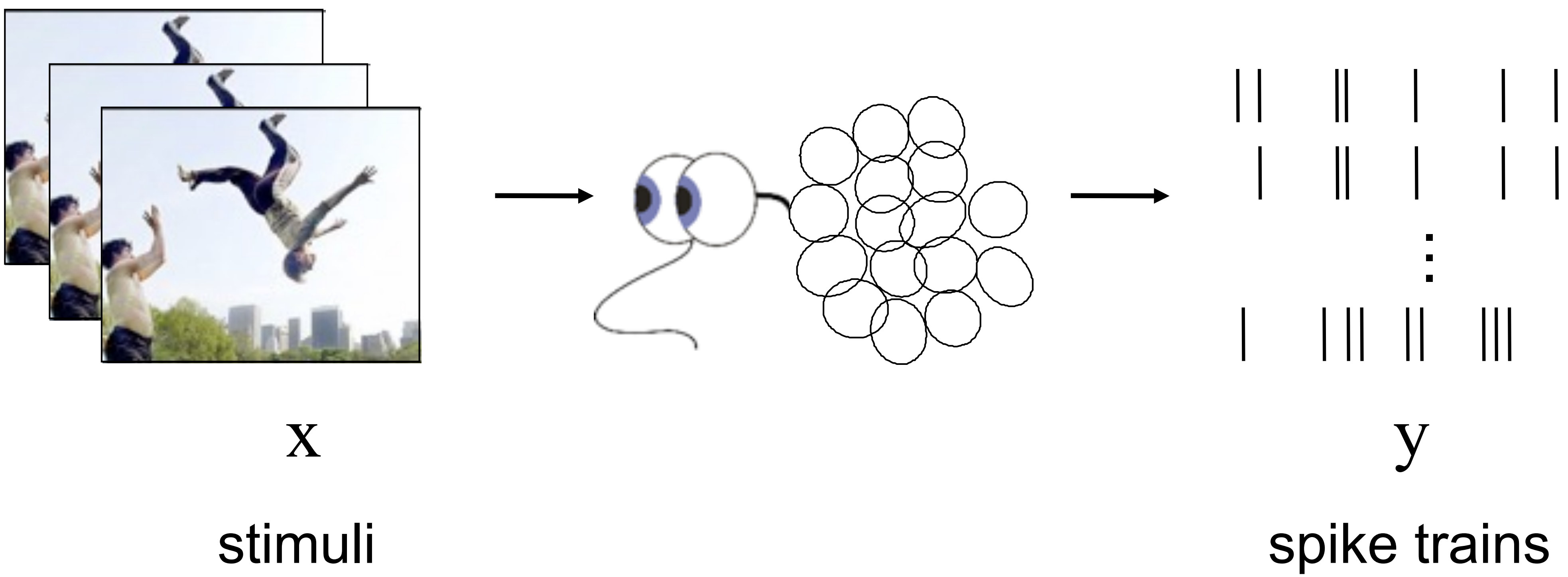
Contrary to what Konrad may have told you:
do **NOT** use ‘`fminsearch`’!

DO use:

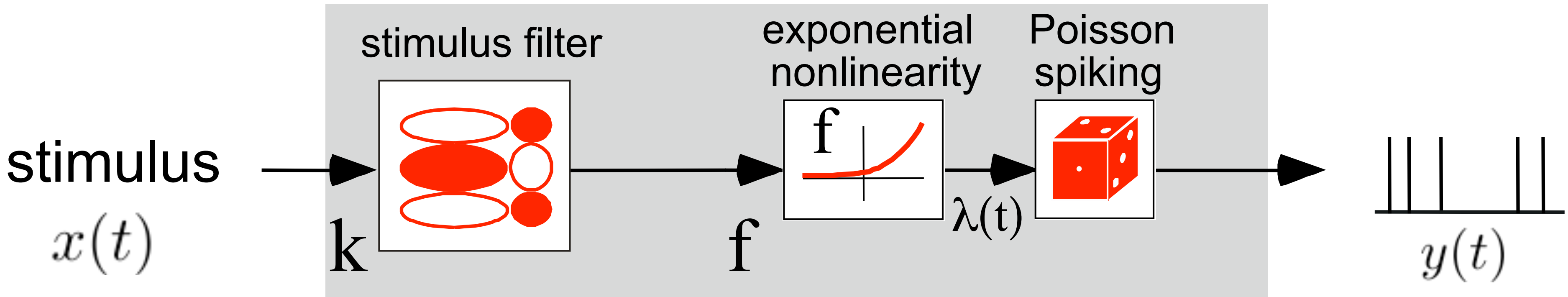
1. `fminunc` - unconstrained optimization
2. `fmincon` - constrained optimization

Block II:

GLMs with spike-history and coupling



Poisson GLM

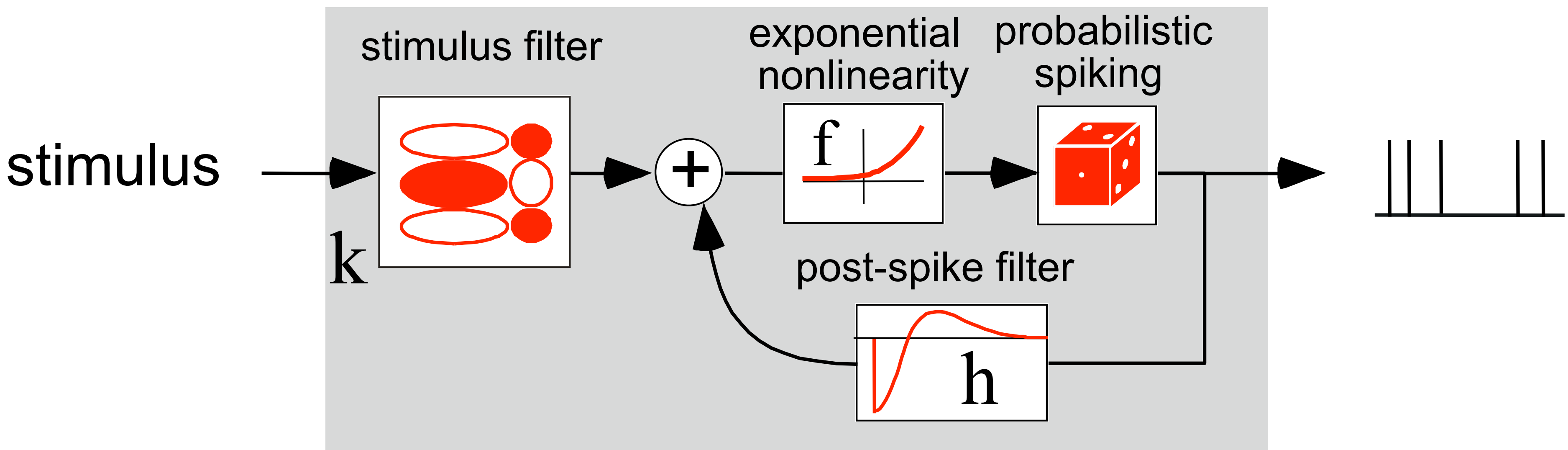


spike rate $\lambda(t) = f(k \cdot x(t))$

- problem: assumes spiking depends only on stimulus!

(Truccolo et al 04)

Poisson GLM with spike-history dependence



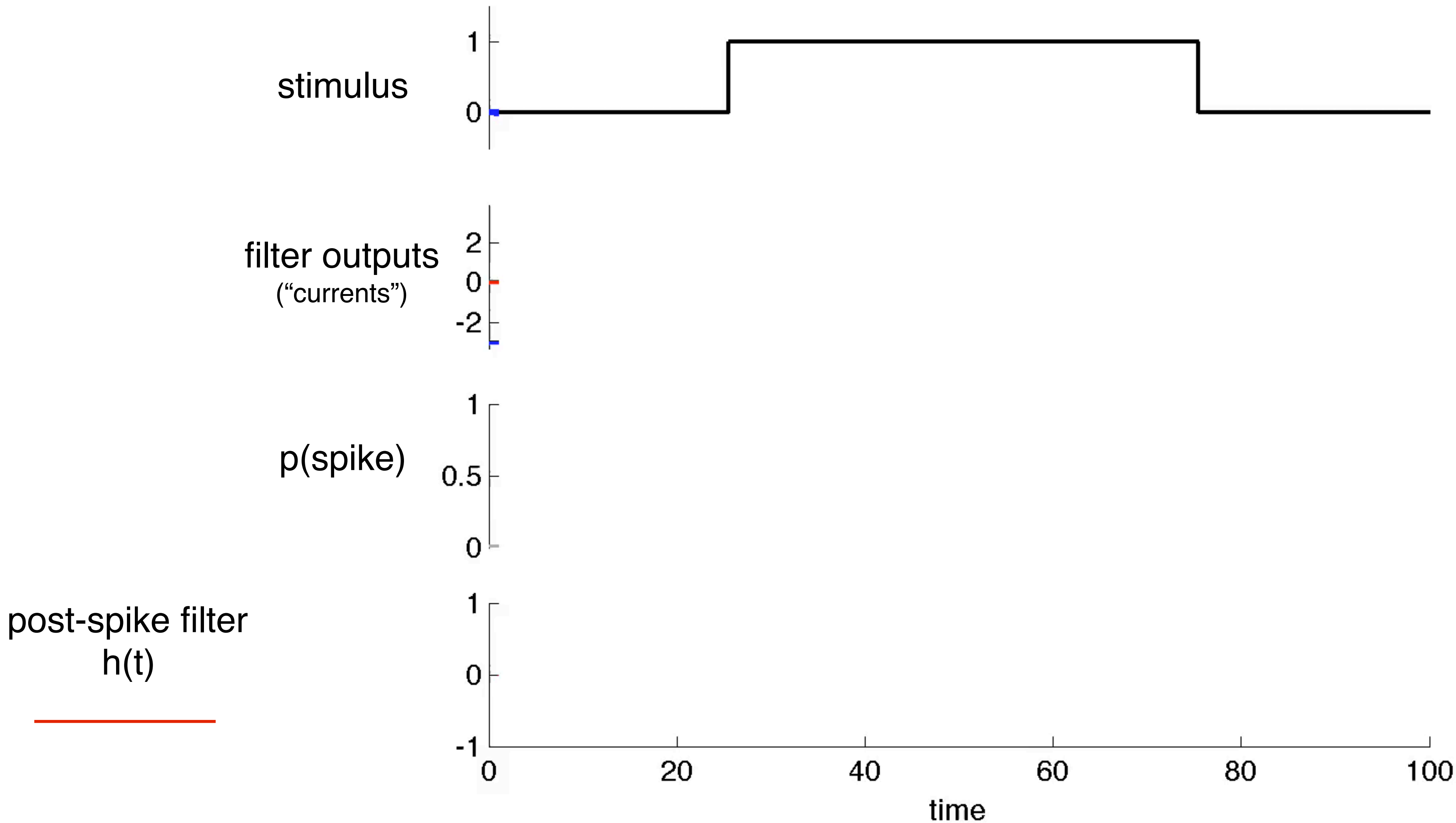
spike rate: $$\lambda(t) = f(\vec{k} \cdot \vec{x}(t) + \vec{h} \cdot \vec{y}_{hst}(t))$$

$$= e^{\vec{k} \cdot \vec{x}(t)} \cdot e^{\vec{h} \cdot \vec{y}_{hst}(t)}$$

- output: no longer a Poisson process
- interpretation: "soft-threshold" integrate-and-fire model

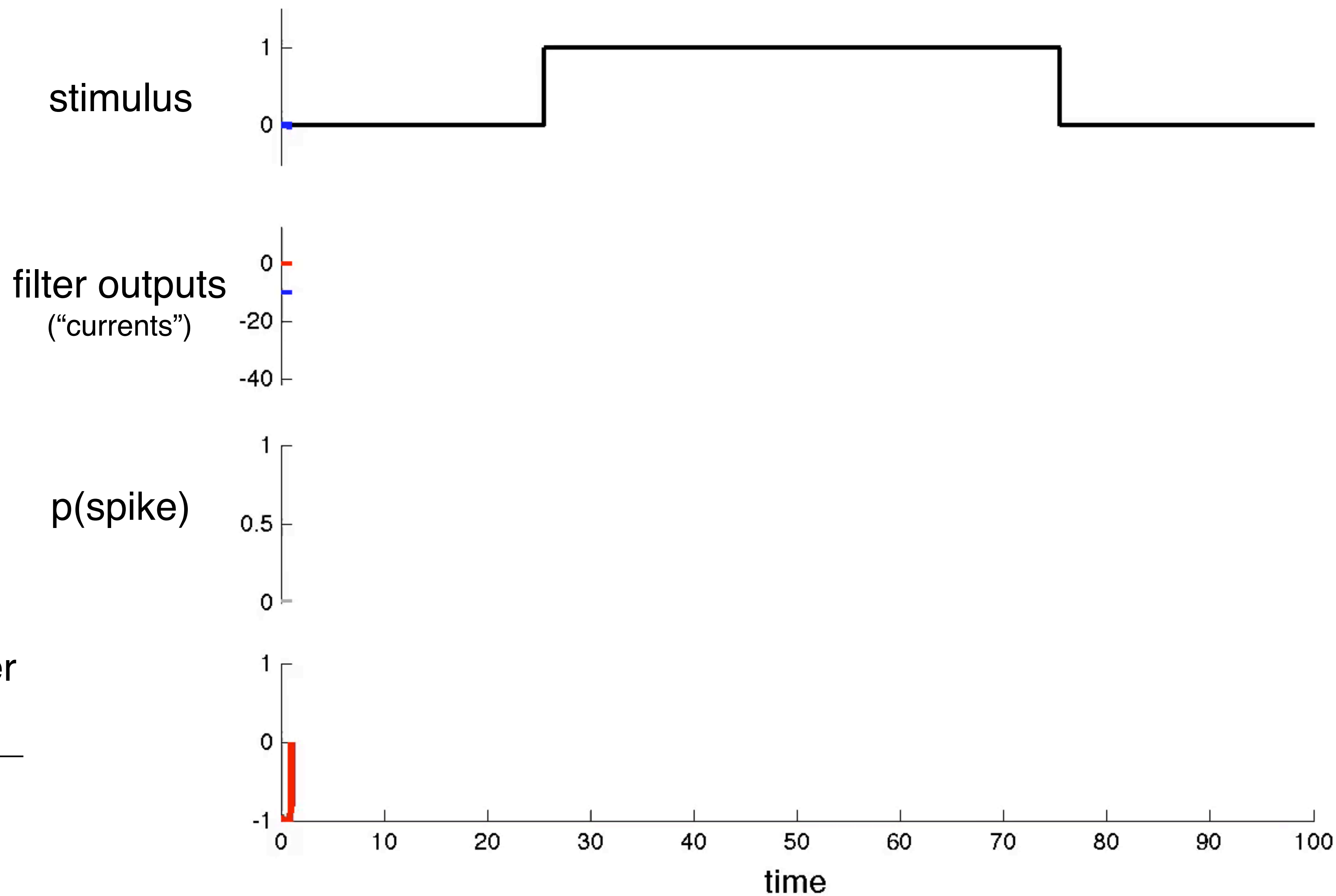
GLM dynamic behaviors

- irregular spiking

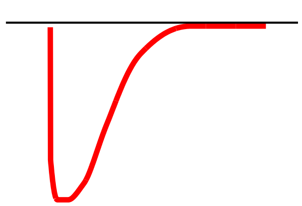


GLM dynamic behaviors

- regular spiking

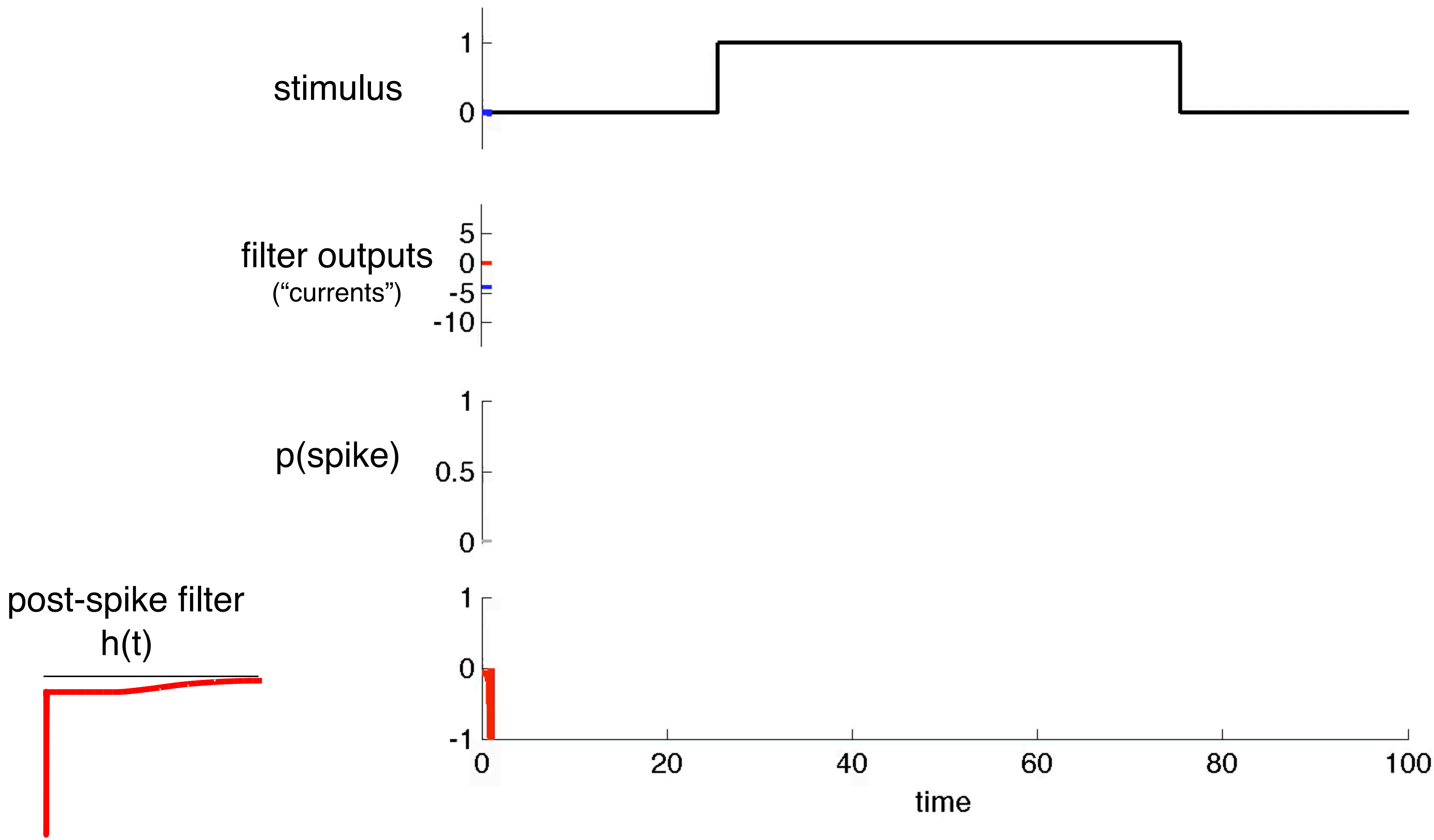


post-spike filter
h(t)



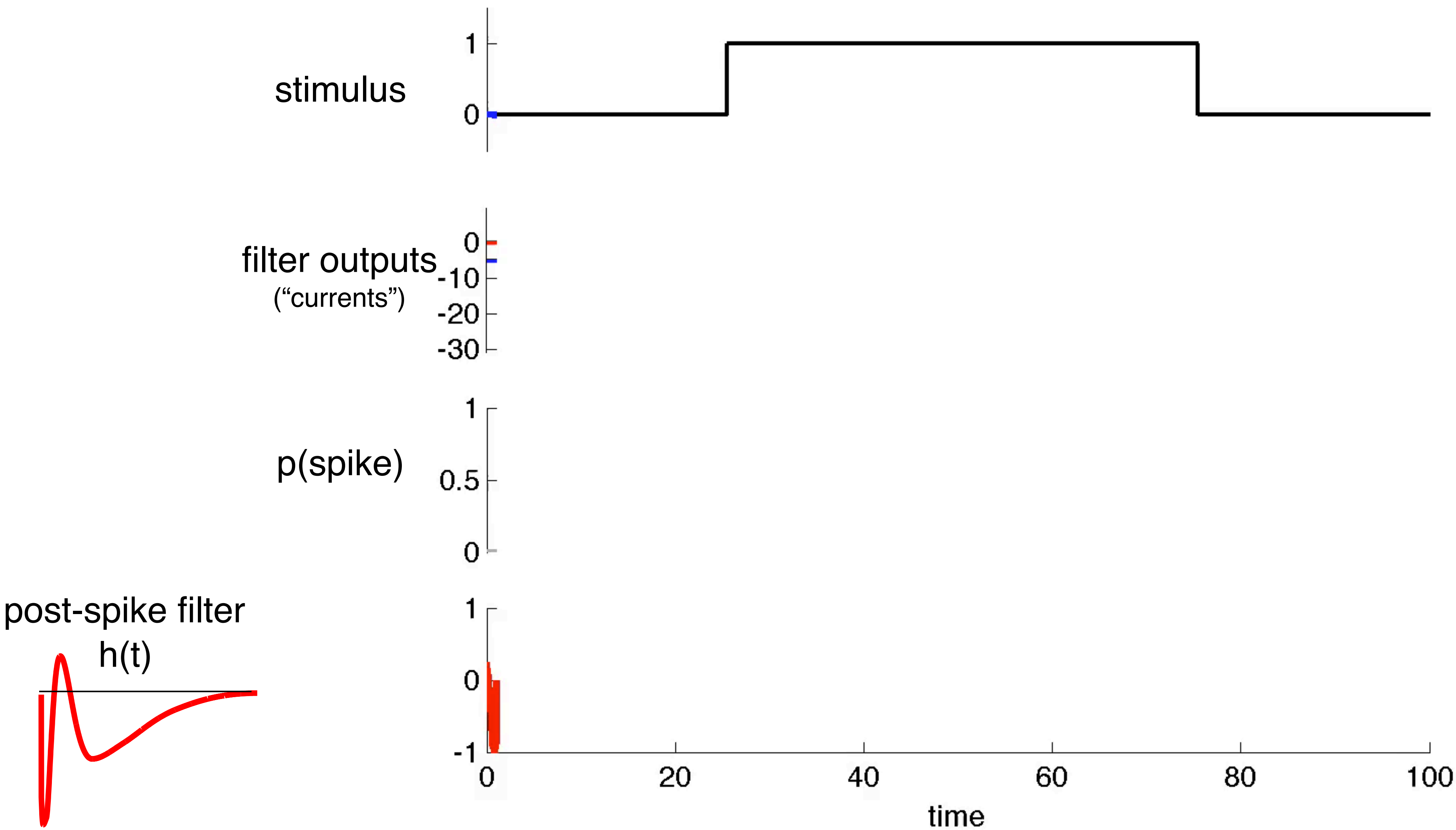
GLM dynamic behaviors

- adaptation

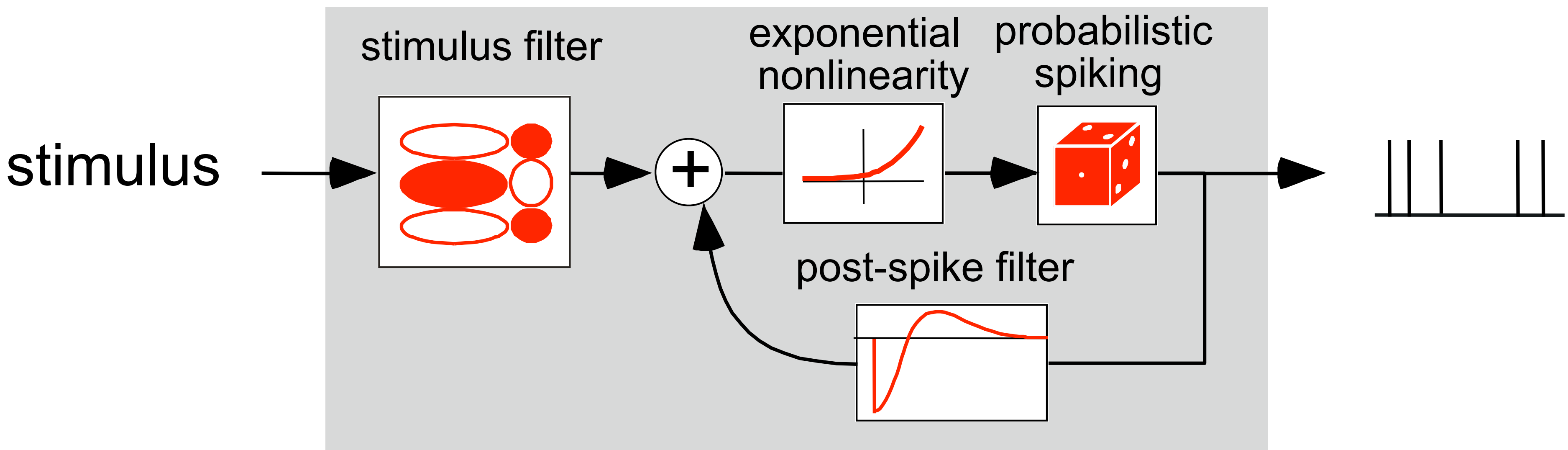


GLM dynamic behaviors

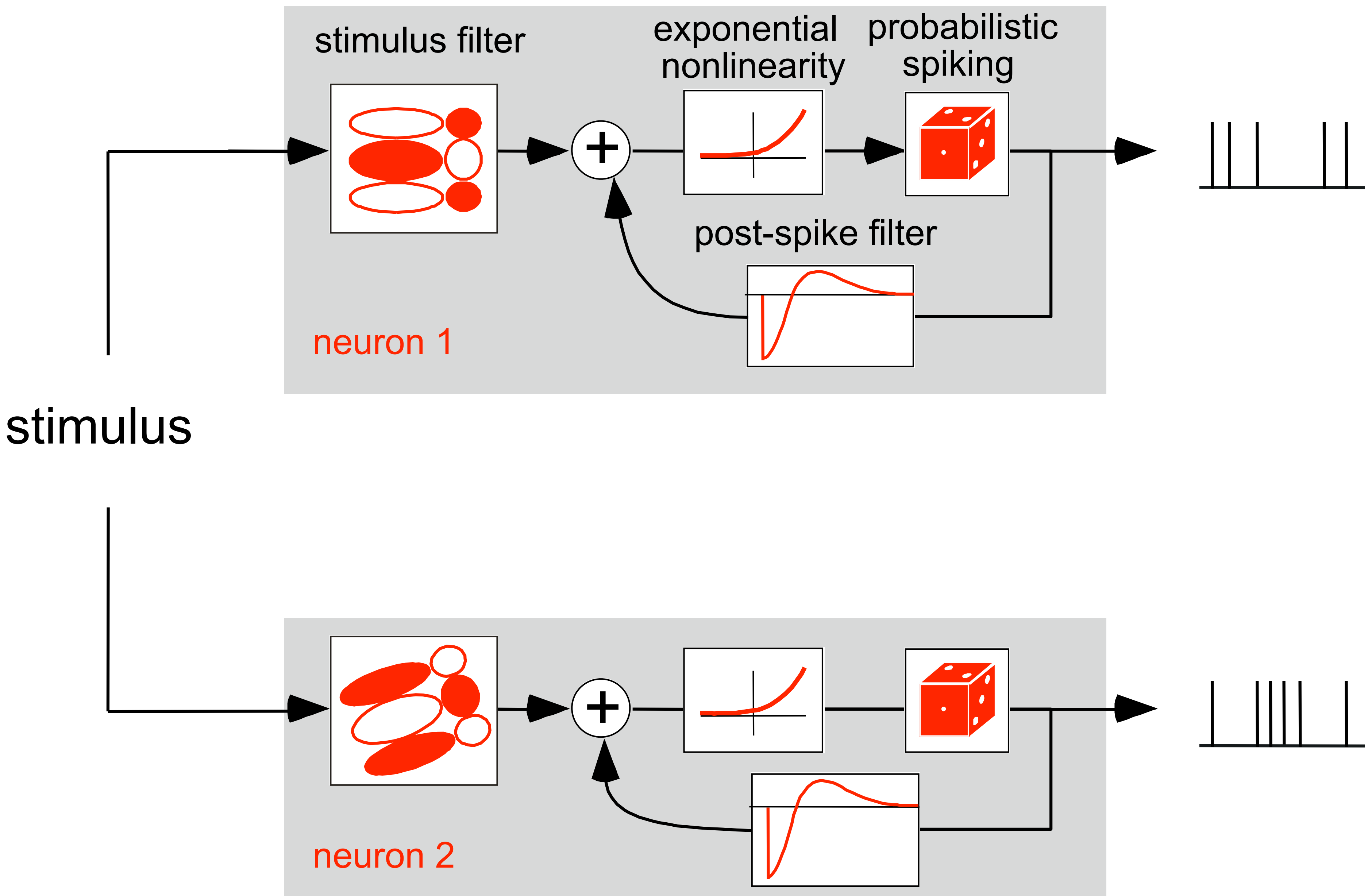
- bursting



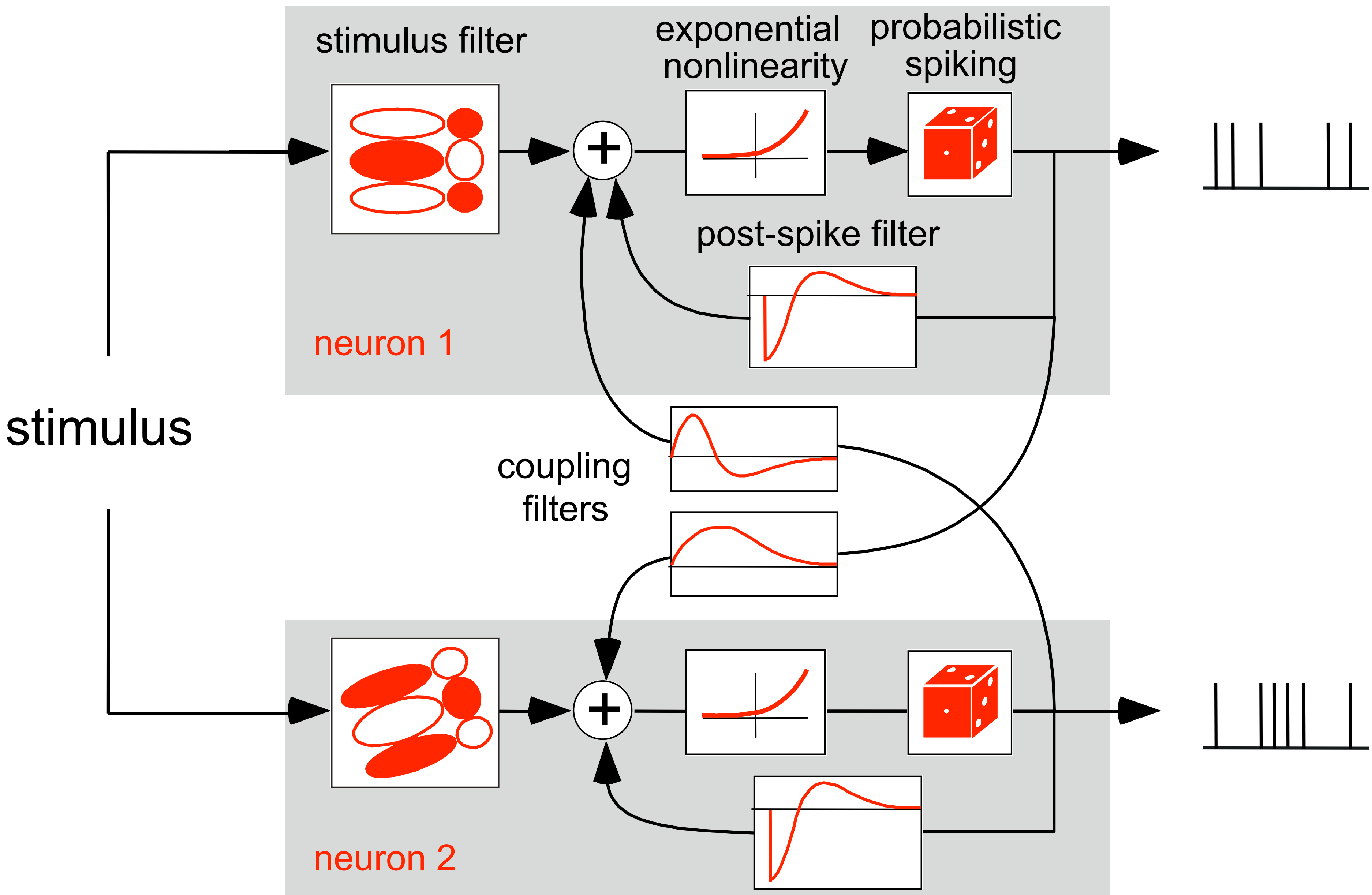
Generalized Linear Model (GLM)



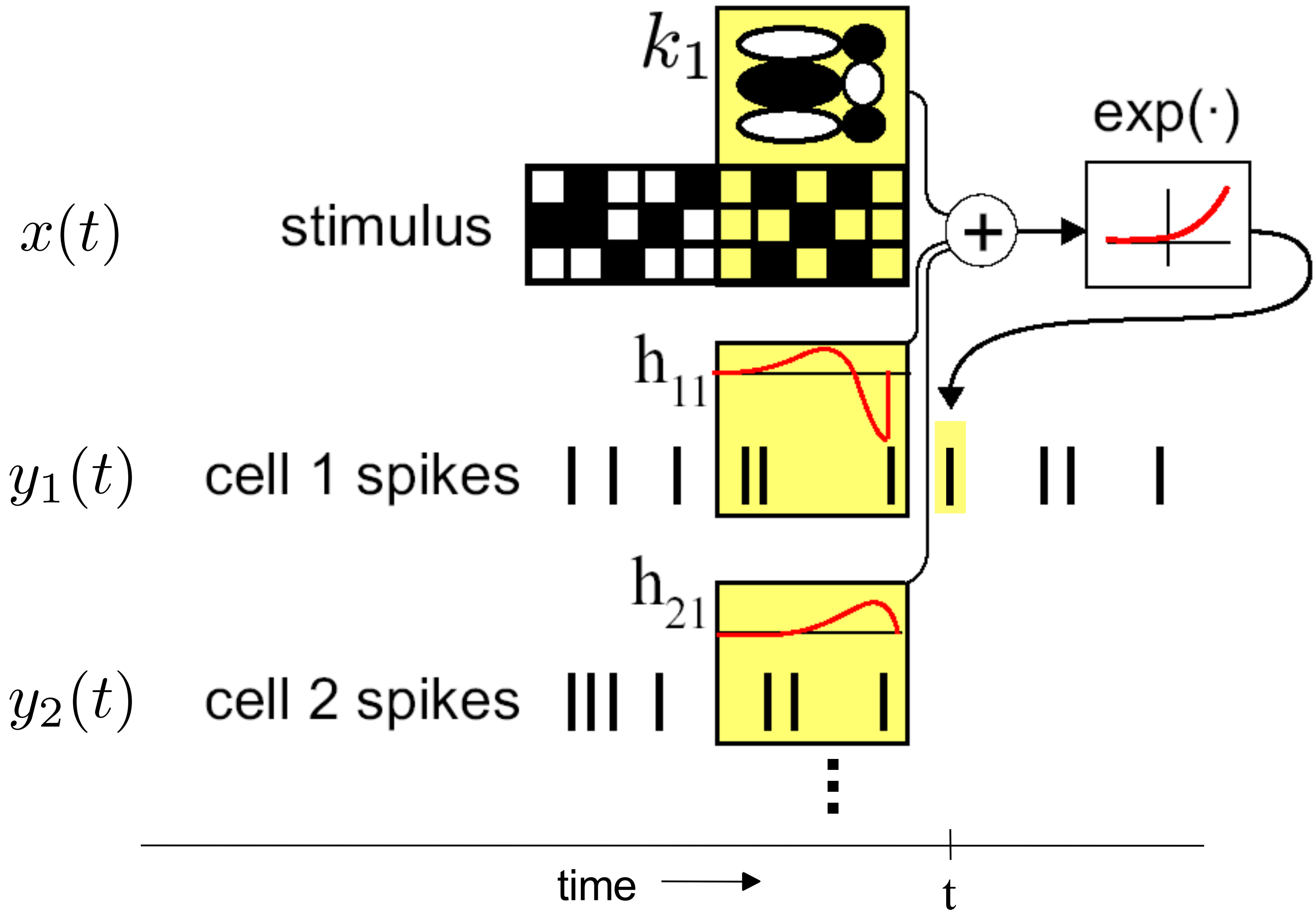
multi-neuron GLM



multi-neuron GLM



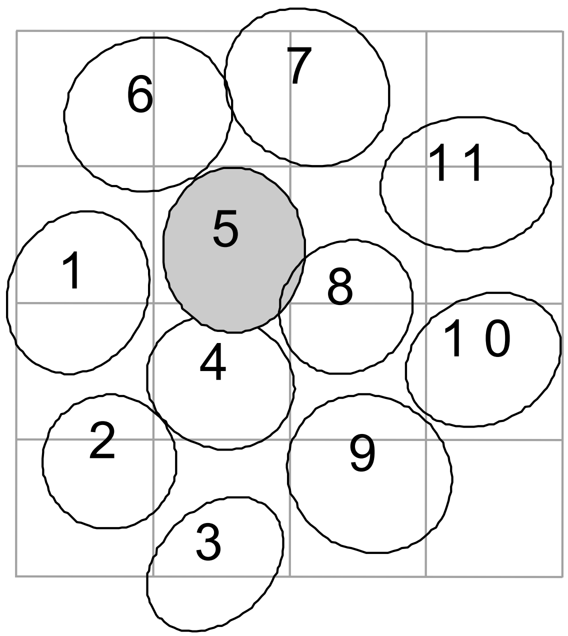
GLM equivalent diagram:

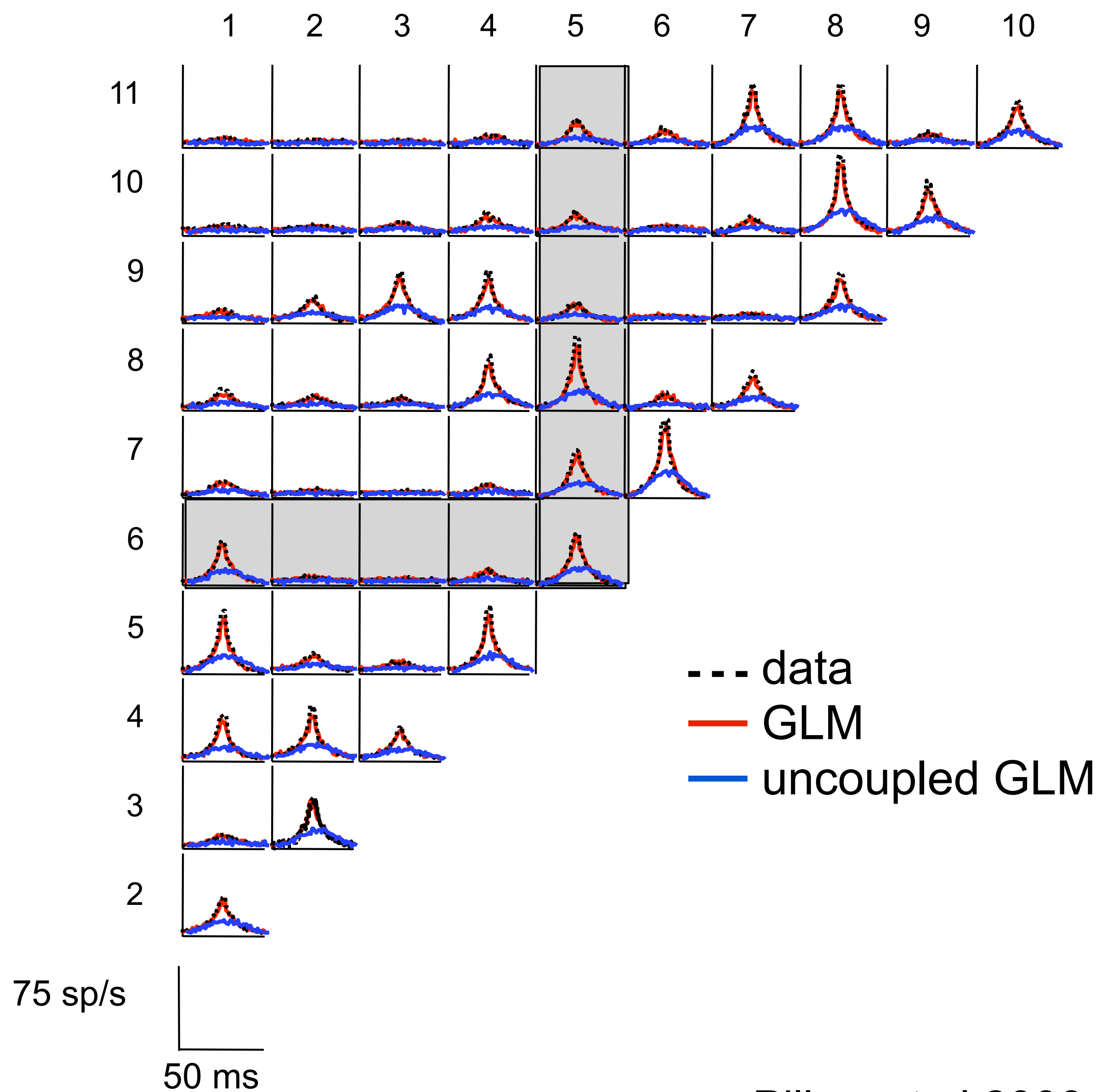


spike rate $$\lambda_i(t) = \exp(k_i \cdot x(t) \; + \; \sum_j h_{ij} \cdot y(t))$$

modeling correlation structure in neural spike trains

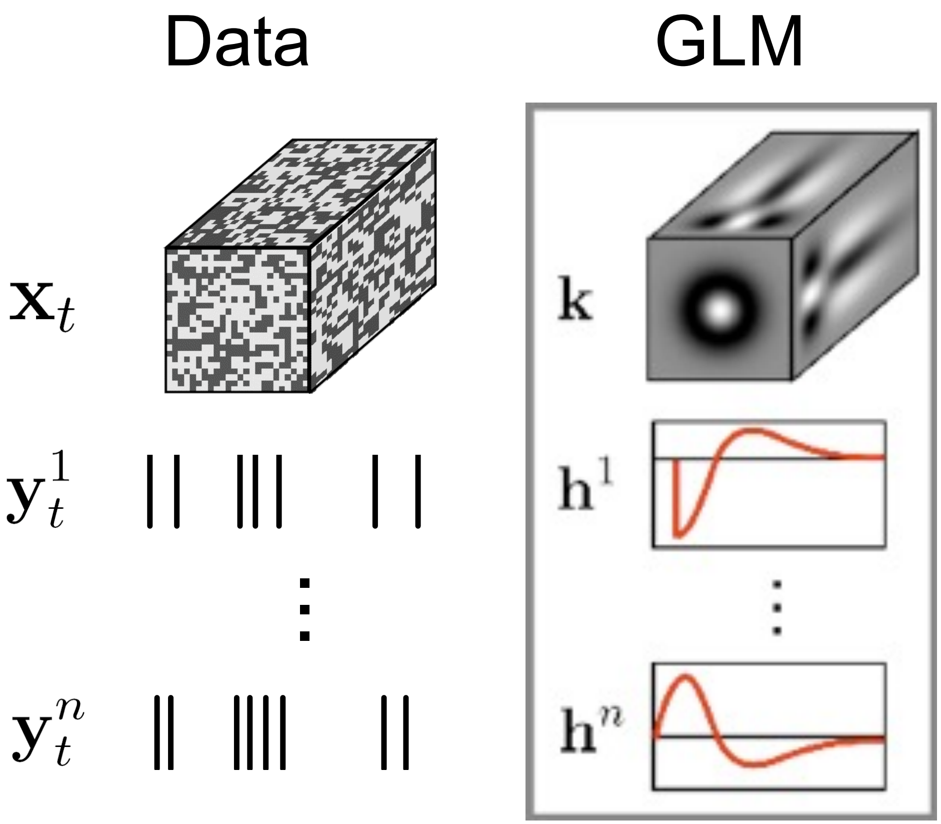
receptive fields





Pillow et al 2008

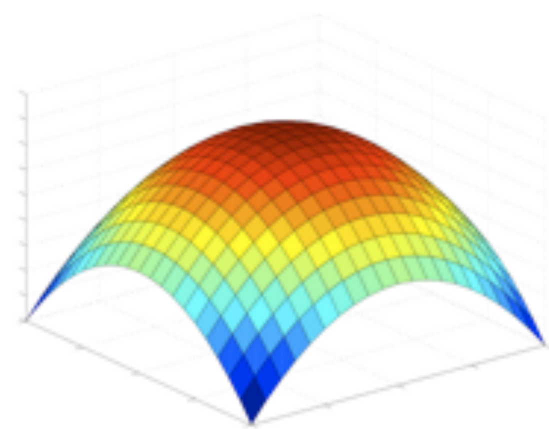
Fitting: Maximum Likelihood



- find filters that maximize the log-conditional probability of the observed data

$$\log P(Y|X) = \sum_t y_t \log \lambda_t - \lambda_t$$

- log-likelihood is concave
- no local maxima [Paninski 04]



Now do it: tutorial2_spikehistcoupledGLM.m

Topics:

1-2. load data & bin spike trains

3. build design matrix, now including spike history!

4. fit Poisson GLM with spike-history

5. fit coupled Poisson GLM (4 neuron spike-history)

6. model comparison: log-likelihood & AIC

Block III: regularization

Modern statistics

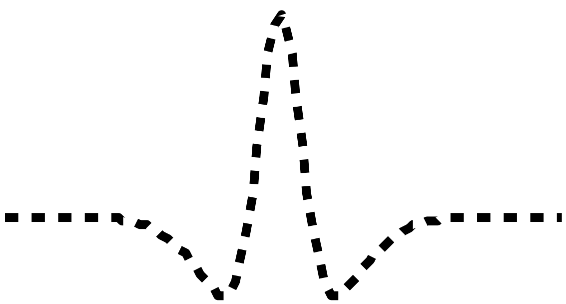
- more dimensions than samples $D \geq N$

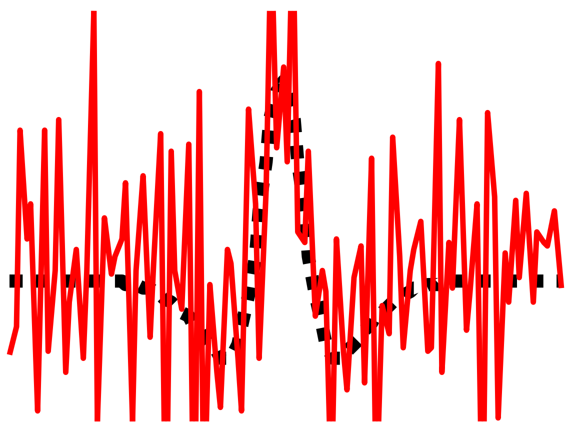
$$\text{N observations}\left\{\begin{bmatrix} y_1 \\ | \\ y_N \end{bmatrix}\right. = \overbrace{\begin{bmatrix} \text{——} & \vec{x}_1 & \text{——} \\ & & \\ \text{——} & \vec{x}_N & \text{——} \end{bmatrix}}^{\text{D regressors}} \begin{bmatrix} w_1 \\ | \\ | \\ w_D \end{bmatrix} + \text{noise}$$

- fewer equations than unknowns!
- no unique solution

Simulated Example

- 100-element filter (D=100)
- 100 noisy samples (N=100)





maximize

$$\log p(data|\mathbf{w})$$

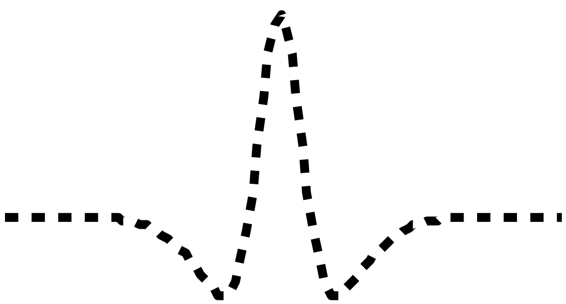
"overfitting" - parameters fit to details in the training data that are NOT useful for predicting new data

Simulated Example

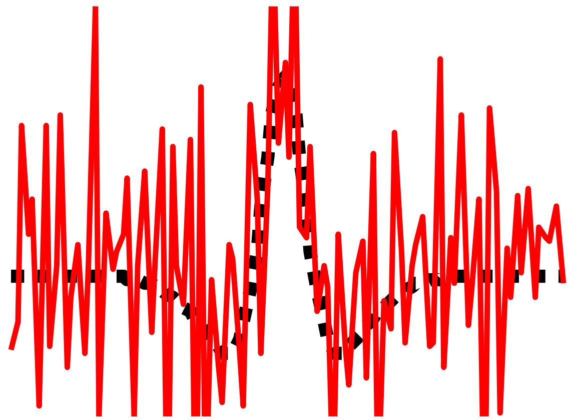
- 100-element filter (D=100)
- 100 noisy samples (N=100)

$$\hat{\mathbf{w}} = (X^\top X + \lambda I)^{-1} X^\top Y$$

true **w**



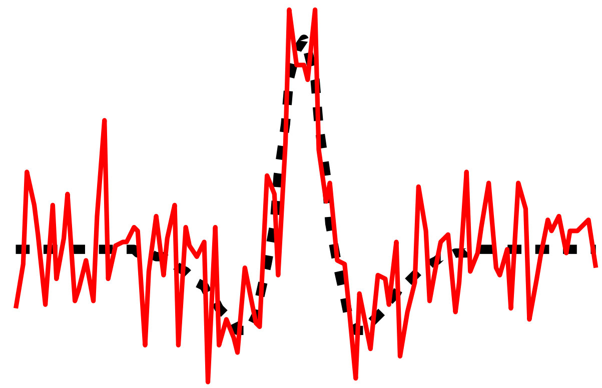
maximum likelihood



maximize

$$\log p(data|\mathbf{w})$$

“ridge regression”



maximize

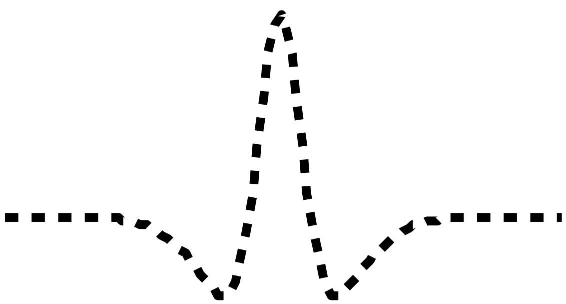
$$\log p(data|\mathbf{w}) - \underbrace{\lambda \sum w_i^2}_{\text{penalty on big weights}}$$

- biased, but gives improved performance for appropriate choice of λ (James & Stein 1960)

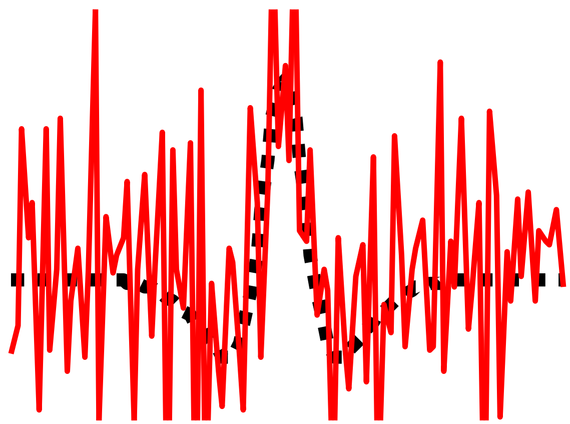
Simulated Example

- 100-element filter (D=100)
- 100 noisy samples (N=100)

true **w**



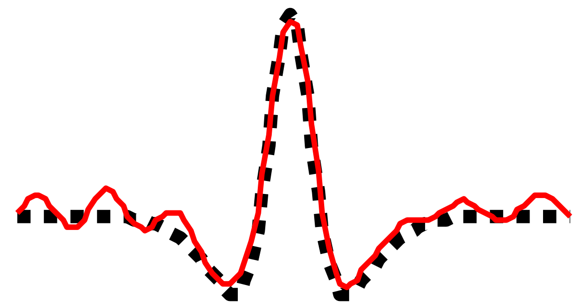
maximum likelihood



maximize

$$\log p(data|\mathbf{w})$$

"smoothed"



maximize

$$\log p(data|\mathbf{w})$$

$$-\lambda \underbrace{\sum (w_i - w_{i-1})^2}_{\text{smoothness penalty}}$$

Q: how to set the regularization strength λ?

Simplest answer: use cross-validation!

Now do it: tutorial3_regularization.m

Topics:

1-2. load data, upsample to finer time bins.
3. divide into training and test data
4. maximum likelihood (linear-Gaussian model)
5. ridge regression (iid Gaussian prior on coeffs)
6. smoothing regression (iid Gaussian prior on diffs)