Generalized Linear Models

Lecture by
Patrick Mineault, PhD candidate

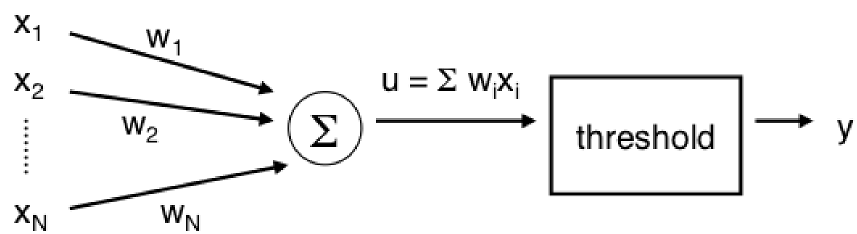
In this lecture

- Expand on Curtis' lecture on artificial neural nets, Chris' on receptive fields and LN models, and Maurice's on spike train statistics
- Tackle the problem of estimating the relationship between a neuron's output and its inputs (including stimuli, other neurons, LFPs, cortical state, etc.)
  - more rigor
  - more biophysics
  - more nonlinearities
  - more PAIN
- Generalized Linear Models

Previously

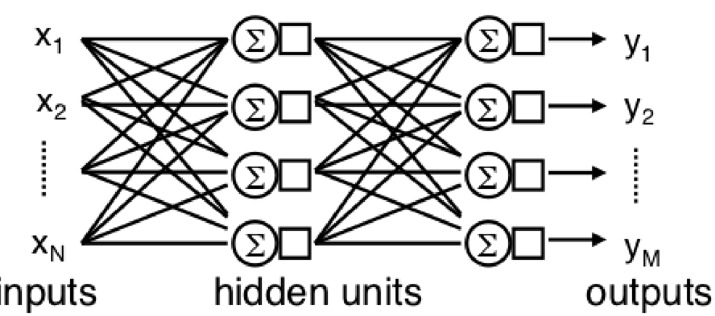
Neural Networks - a brief history

1950s-60s: McCollough-Pitts neuron; "feature-detector" neurons in optic tectum, A17



1960s-70s: Rosenblatt Perceptron: architecture (single-layer)
novelty at the time: learning; distributed memory; neural inspiration
Minsky & Papert critique, difficulty with multi-layer networks

1980s-90s: revenge of the neural networkers: back-prop, connectionism, etc
concurrent influences: neural plasticity, NMDA receptors;
Donald Hebb; David Marr; Rumelhart, Hinton, Sejnowski



90s, 00s: rise of the machines: machine learning (neural or not)
probabilistic models, statistical learning theory

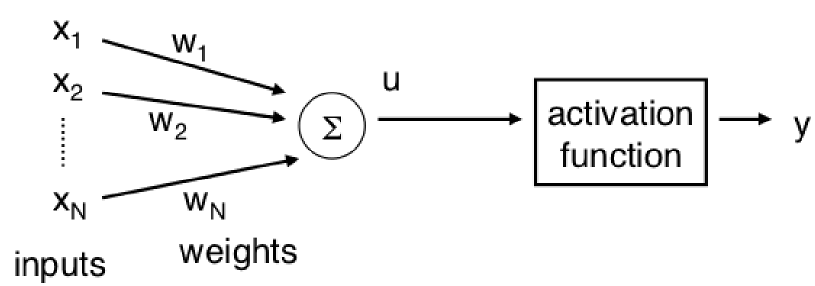
Previously

- The McCulloch-Pitts neuron is a metaphor for real neurons

Previously

Classification with LMS gradient descent

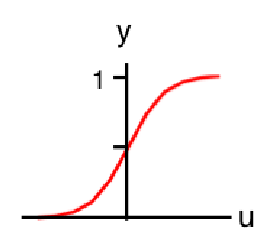
architecture



e.g, "least mean squares": $E = 1/2\ \Sigma\ (T_j - y_j)^2$

where y_j = network response, on trial j
T_j = "teacher", i.e. desired or correct respose

(for probabilistic model, with Gaussian noise, this is optimal)



activation function: must be differentiable
-> logistic (sigmoid): $y(u) = 1/(1 + e^{-u})$

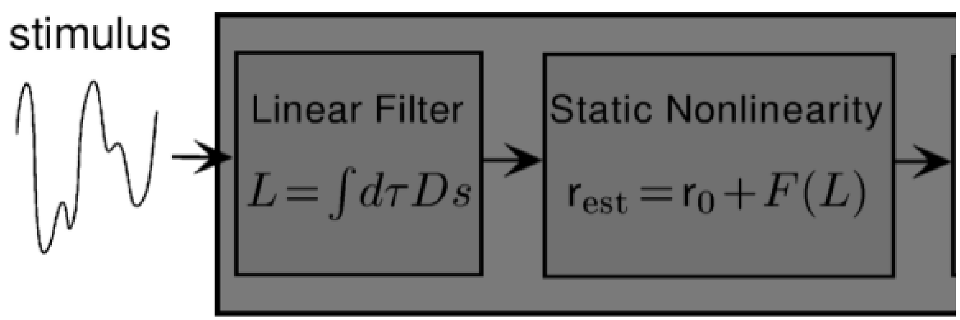
-> **learning rule**: $dw_i = -\eta\ \Sigma\ (T_j - y_j)\ x_i$

notes: for linear model, will always converge to unique minimum
best run in batch mode

problems: only gives good classification if categories are linearly separable

Previously

Linear filtering approach

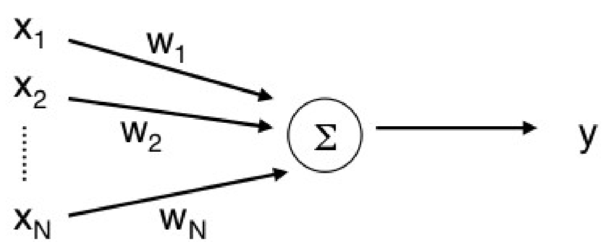


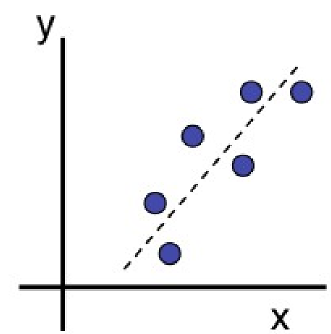
Previously

Linear Regression

simplest case, single output: $y = \Sigma\, w_i x_i$

architecture





error function:

$$E = \Sigma\, (y_i \;-\; \text{predicted } y_i)^2$$

gradient descent:

min: $\delta E / \delta w = 0$ -> $dw_i = -\eta\, \delta E / \delta w_i$

η ("eta") = learning rate parameter

learning rule:

$$dw_i = \eta\, (y_i \;-\; \text{predicted } y_i)\, x_i$$

In this lecture

- It's all pretty much the same thing
  - McCulloch-Pitts neurons
  - Artificial neural nets
  - Classification
  - Linear regression
  - LN models
  - Generalized linear models
- GLMs and their many extensions are very powerful

Stuff you can do with GLMs

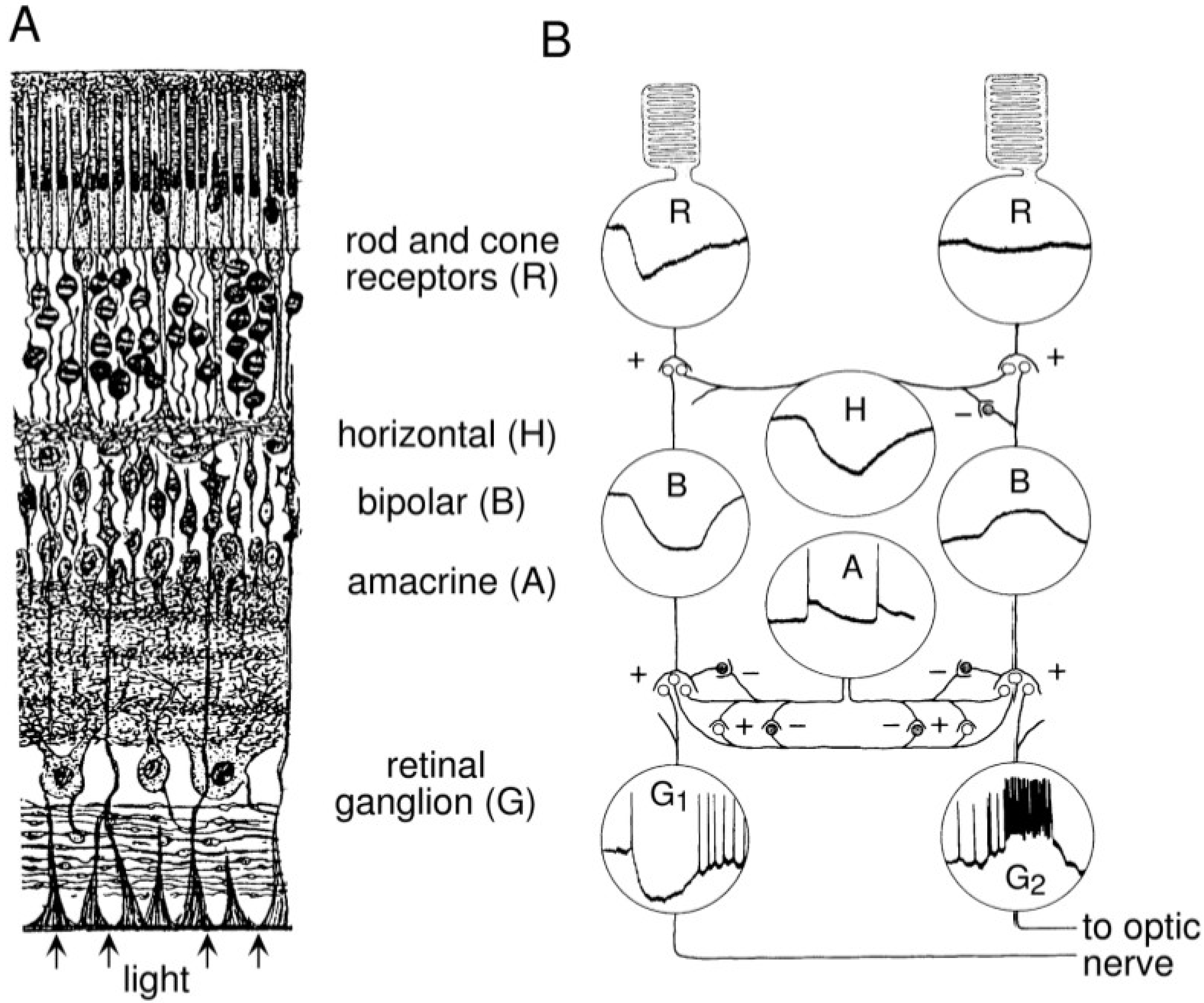
Vol 454 | 21 August 2008 | **doi:10.1038/nature07140** nature

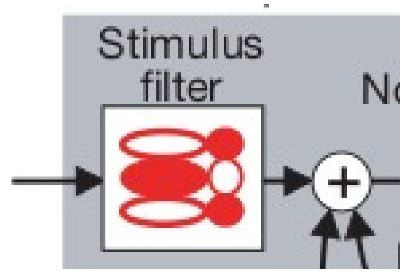
LETTERS

Spatio-temporal correlations and visual signalling in a complete neuronal population

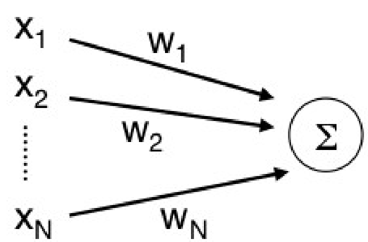
Jonathan W. Pillow[1], Jonathon Shlens[2], Liam Paninski[3], Alexander Sher[4], Alan M. Litke[4], E. J. Chichilnisky[2] & Eero P. Simoncelli[5]

The Retina

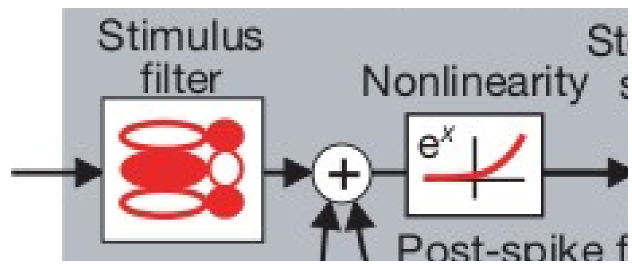

A
B
R
R
rod and cone
receptors (R)
+
+
H
–
horizontal (H)
B
B
bipolar (B)
A
amacrine (A)
+
–
–
+
+
–
–
+
retinal
ganglion (G)
G_1
G_2
to optic
nerve
light

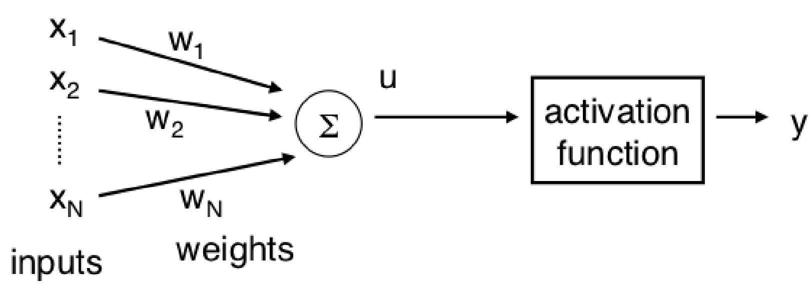


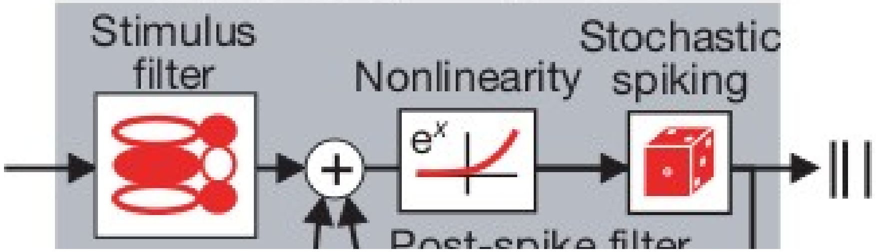
Stimulus
filter



x_1
w_1
x_2
w_2
x_N
w_N
Σ

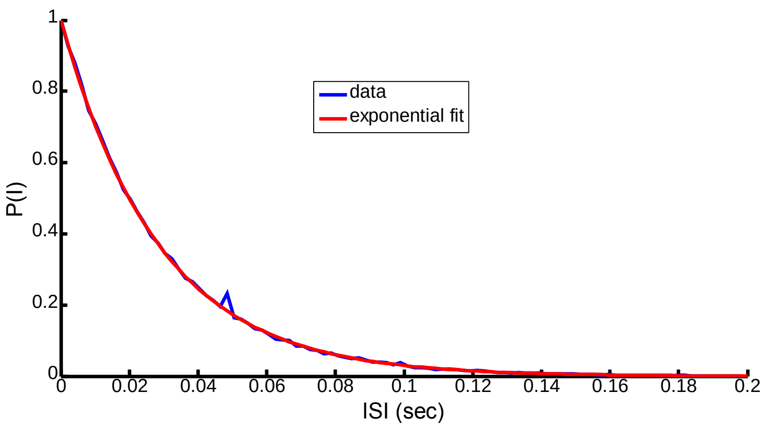

Stimulus
filter
Nonlinearity
e^x
Post-spike f


x1
x2
xN
w1
w2
wN
inputs
weights
Σ
u
activation
function
y

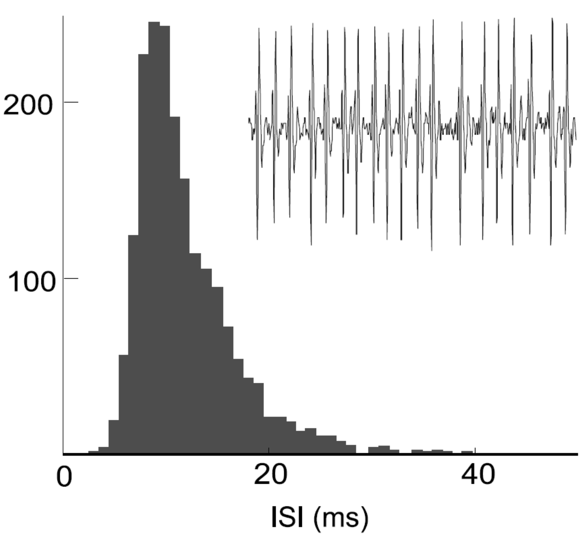


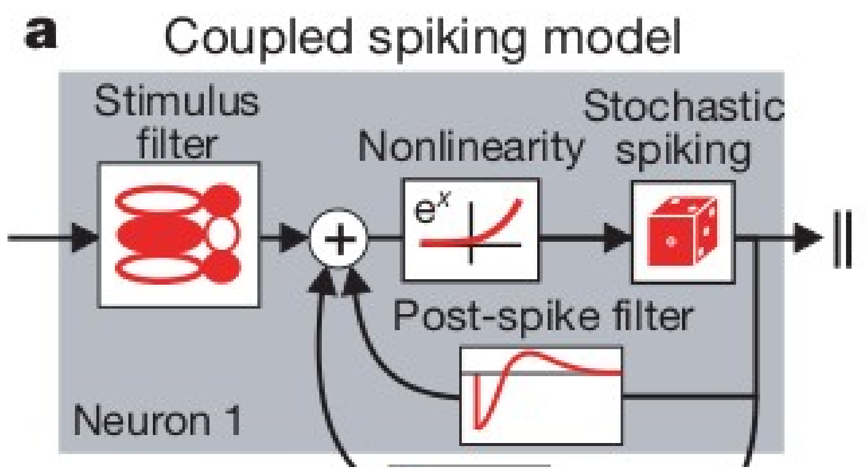
Stimulus
filter
Nonlinearity
Stochastic
spiking

Recall from Maurice's lecture

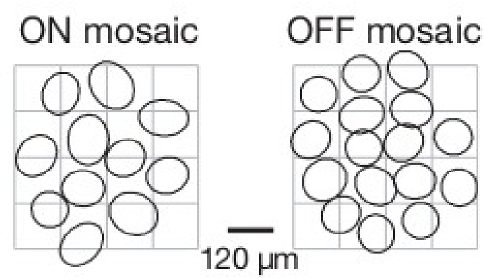


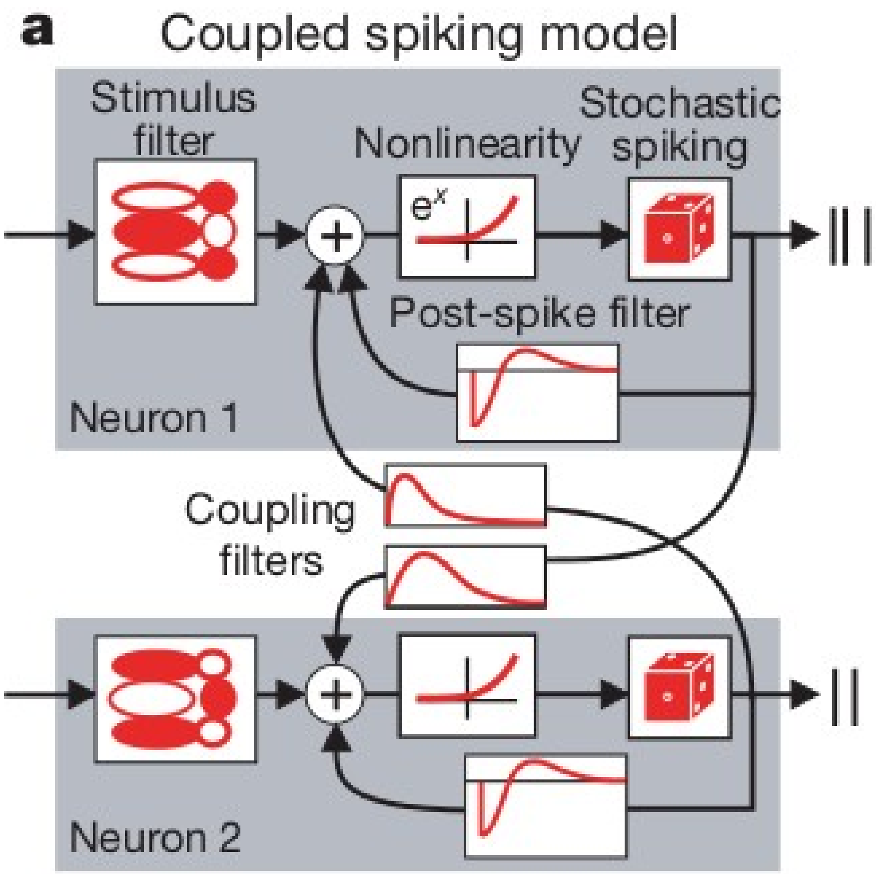
Irregular afferent



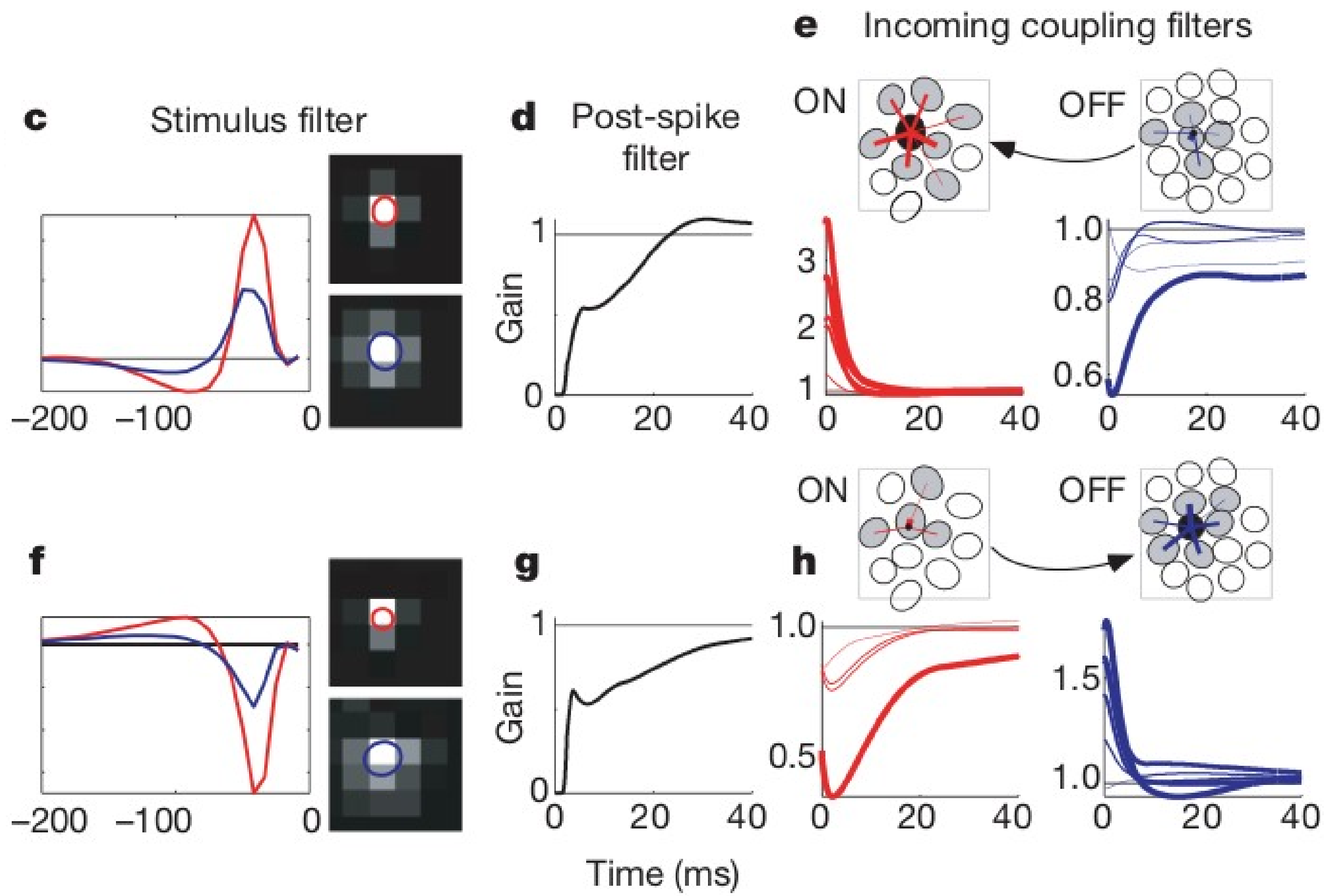

a
Coupled spiking model
Stimulus filter
Nonlinearity
Stochastic spiking
Post-spike filter
Neuron 1

b




a
Coupled spiking model
Stimulus filter
Nonlinearity
Stochastic spiking
e^x
Post-spike filter
Neuron 1
Coupling filters
Neuron 2

Stuff you can do with GLMs



Stuff you can do with GLMs

- This is starting to look less like a metaphor for a real neuron and more like a real model for a network of neurons

By the end of this lecture

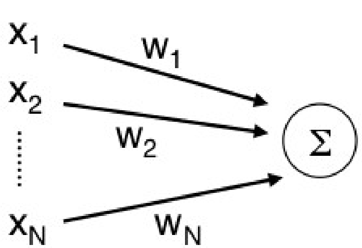
- You will be able to perform all the analyses in the paper
  - (first half only; we won't get to decoding)
  - (you don't have the data)
  - (you won't get into Nature)

Generalized Linear Models

- A flexible class of models which are useful for encoding and decoding neural data
- Widely used: McCullagh (no relation) and Nelder 1989 has ~20k citations
- Widely available in stat packages (R, Matlab, etc.)
- Extends linear regression and includes it as a special case
- Deals with many different kinds of data:
  - Continuous (LFPs, fMRI)
  - Binary (classification images)
  - Count data (spikes)

What's a Generalized Linear Model

- A natural generalization of Linear Models
- In a linear model,



$$\eta_i = \sum_j X_{ij} w_j$$

$$y_i = \eta_i + \epsilon_i$$ Error

$$\epsilon_i \sim Normal(0, \sigma^2)$$

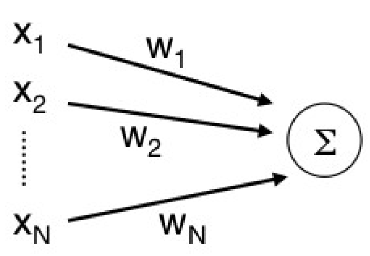
Normal Distribution

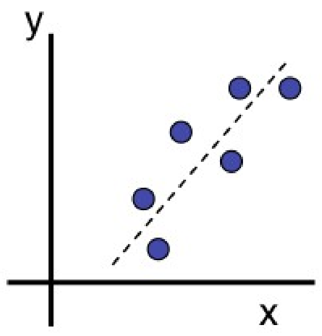
- Inference is done by finding:

$$arg\ min_w \frac{1}{2\sigma^2}(y_i - \eta_i)^2$$

The noise is the tricky bit

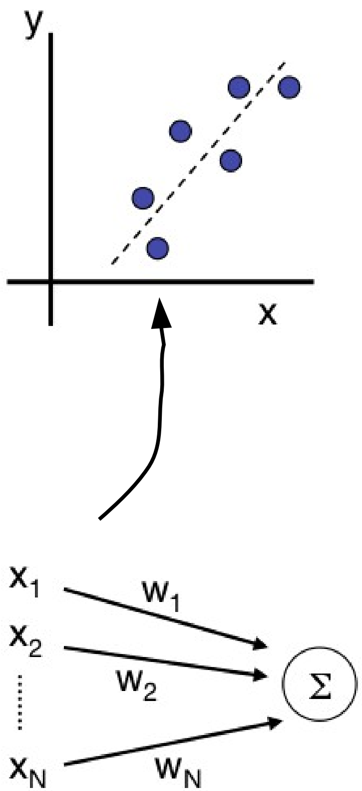
- This is a little different from Curtis' lecture, because we're considering noise explicitly
- You have to consider noise explicitly if you're going to get a generative model which makes sense

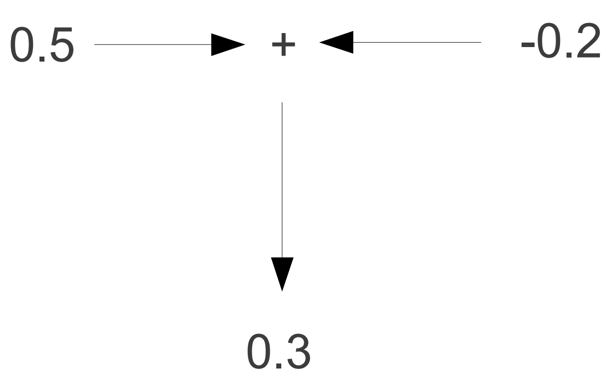




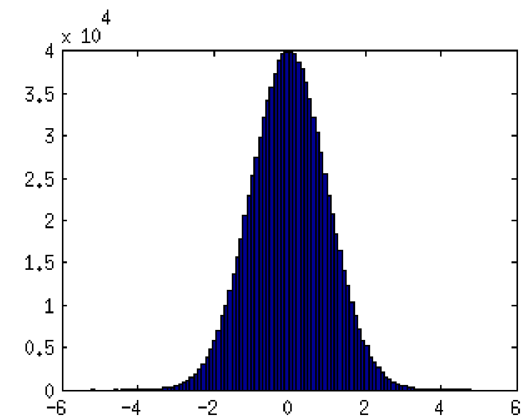
- This is the tricky bit

Two equivalent descriptions of additive noise

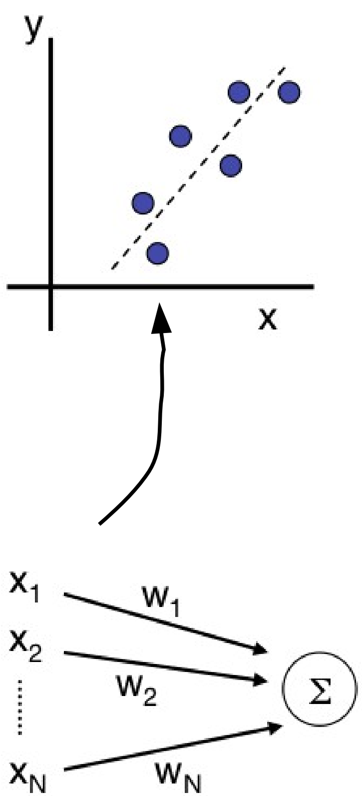




Normal distribution
Mean = 0
Variance = 1

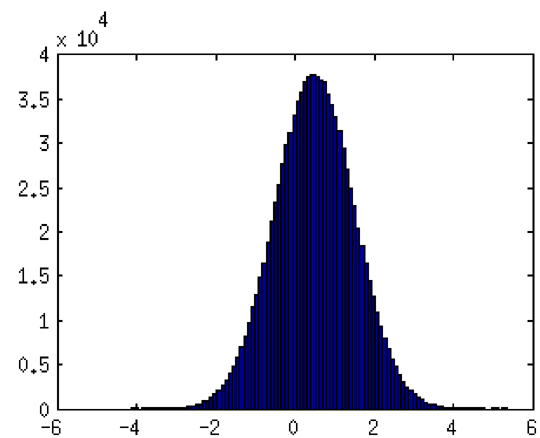


Two equivalent descriptions of additive noise



0.5

Normal distribution
Mean = 0.5
Variance = 1



0.3

What's a Generalized Linear Model

- In a linear model,

$$\eta_i = \sum_j X_{ij} w_j$$
$$y_i \sim Normal(\eta_i, \sigma^2)$$

- Inference is done by finding:

$$arg\,min_w \frac{1}{2\sigma^2}(y_i - \eta_i)^2$$

What's a Generalized Linear Model

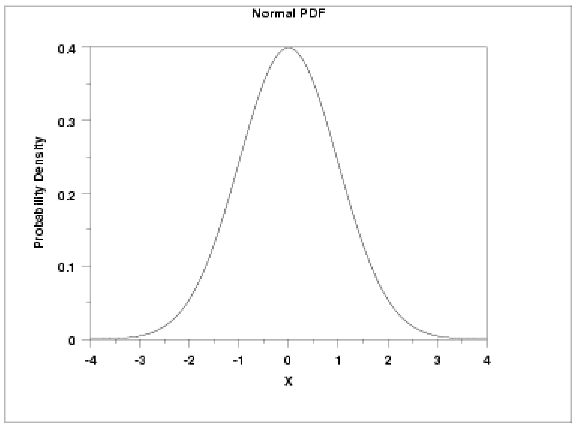
- In a generalized linear model,

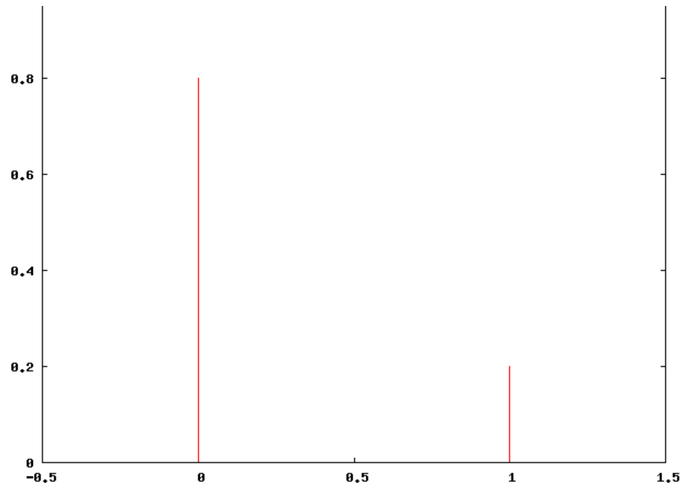
$$\eta_i = \sum_j X_{ij} w_j$$
$$r_i = f(\eta_i)$$
$$y_i \sim Distribution(r_i, params)$$

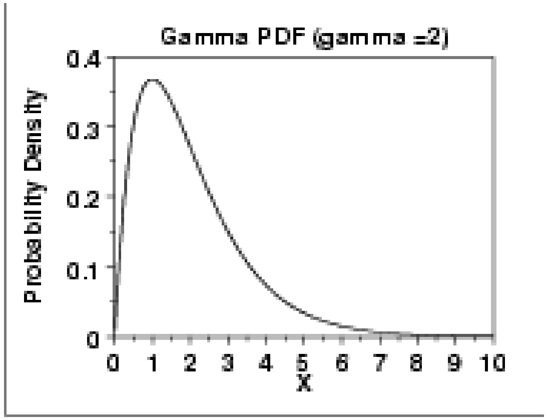
- Inference is done by finding:

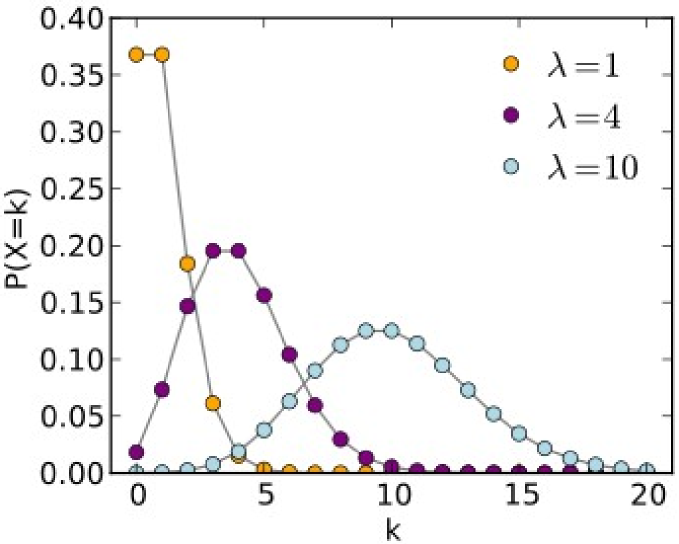
$$arg\,min_w L(y_i, r_i)$$

So many distributions









From nist.gov, wikipedia

So many distributions

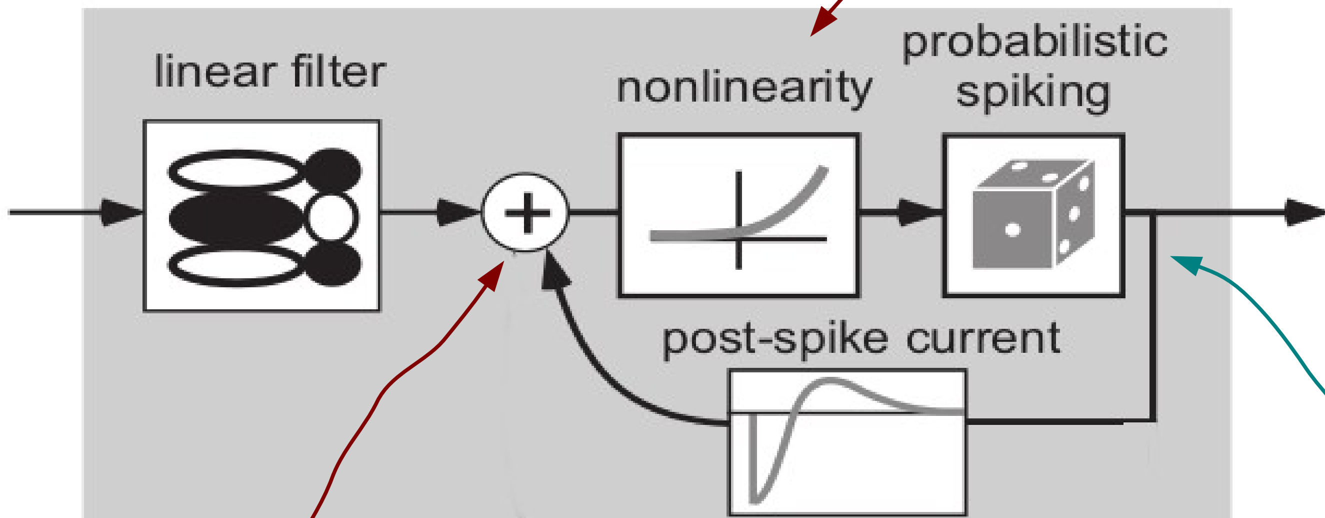
- By choosing the right distribution, you can deal with:
  - Positive, continuous data
  - Binary data
  - Count data

$$\eta_i = \sum_j X_{ij} w_j$$

$$r_i = f(\eta_i)$$

$$y_i \sim Distribution(r_i, params)$$

- By convention, r_i parametrizes the mean of the distribution
- f is chosen to match the range of the distribution



$$\eta_i = \sum_j X_{ij} w_j$$

$$r_i = f(\eta_i)$$

$$y_i \sim Distribution(r_i, params)$$

What's a Generalized Linear Model

- In linear regression, $$arg\ min_w \frac{1}{2\sigma^2}(y_i - r_i)^2$$

  has a single minimum that can be found by optimization (Curtis' lecture)

- In GLMs, $$arg\ min_w L(\overset{x}{y_i}, \overset{y}{r_i})$$

  has a single minimum that can be found by optimization

- Some conditions on the nonlinearity and distribution must be satisfied for this to be the case

A table of generalized linear models

Distribution	Canonical nonlinearity	Name	Appropriate for
Normal	identity	Linear regression	Continuous data
Binomial	Logistic	Logistic regression	Binary data
Poisson	Exponential	Poisson regression	Count data
Multinomial	Logistic		Categorical data
Gamma	1/x		Positive continuous data
Exponential, Inverse Gaussian, Negative binomial			

A summary of Generalized Linear Models

- Up to the very last bit, they're just like linear models
- Estimating parameters is almost as easy as in linear models
- They work for spikes!
- They work for non-spike data as well

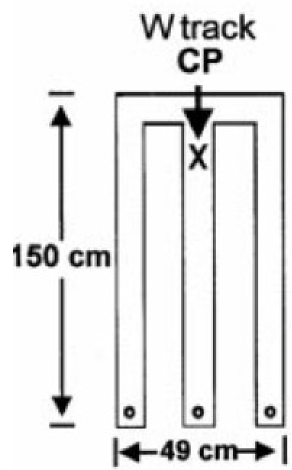
A motivating example

- Problem: come up with a model that describes the firing rate of the neuron as a function of
  - Position in the maze
  - Previous firing rate history
  - Other potential factors: firing rate of other neurons, LFP phase, etc.

Dimensions

A motivating example

- A rat is moving on a linear track, and we are recording from neurons sensitive to position in the hippocampus

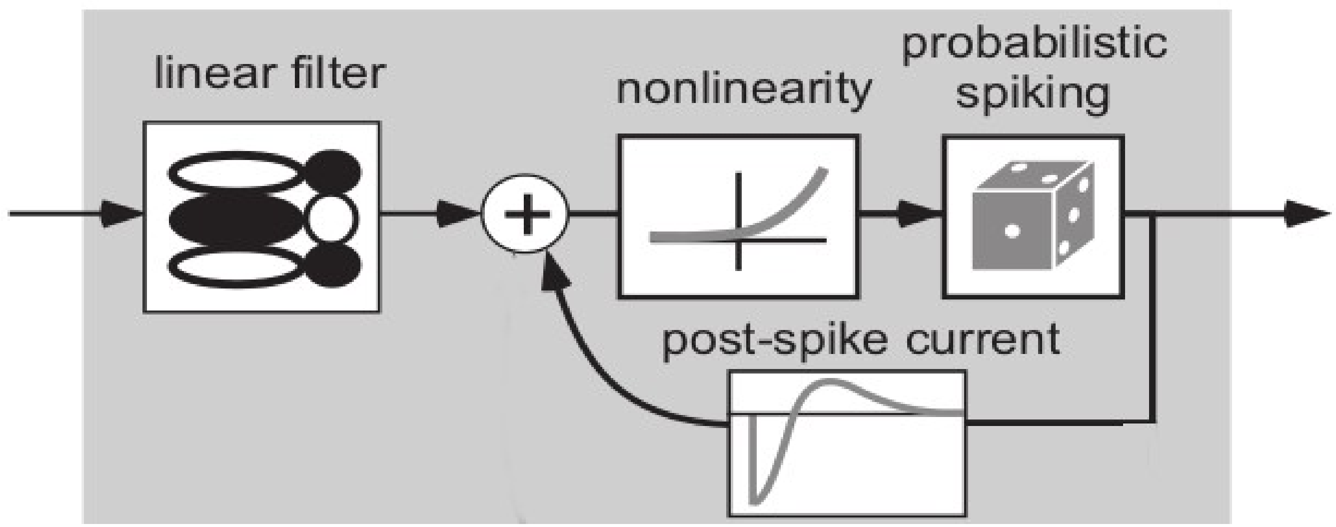


Frank, Brown & Wilson (2000)

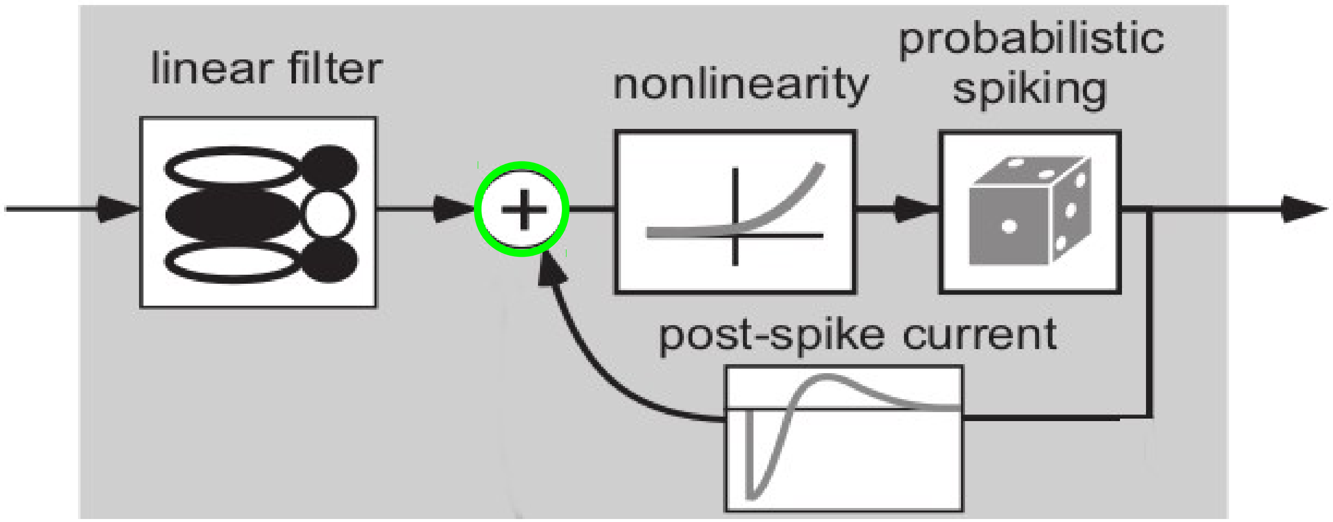
Step 1

- Sample position and spikes at a sufficiently high rate (say 500 Hz). i will index the data, so that $i = 1$ will represent the first 2 milliseconds of data, $i = 2$ from 2 to 4 ms, etc.

What we'd like to end up with



Step by step


linear filter
nonlinearity
probabilistic spiking
post-spike current

The linear bit of the model

$$\eta_i = \sum_j X_{ij} w_j$$

Drive to the neuron (before the nonlinearity)
Within the i'th time epoch

Model weights

Stimulus encoded as a design matrix
N number of rows for the N time epochs
M number of columns for the M stimulus dimensions

The linear bit of the model

$$\eta_i = \sum_j X_{ij} w_j$$

$X_{i,1}$: encodes the position of the rat at the i'th time point
Varies between [0,1]

The linear bit of the model

$$\eta_i = \sum_j X_{ij} w_j$$

$X_{i,2:n+1}$: encodes whether there was a spike at time i – 1, i – 2, i – 3, etc.

The linear bit of the model

$$\eta_i = \sum_j X_{ij} w_j$$

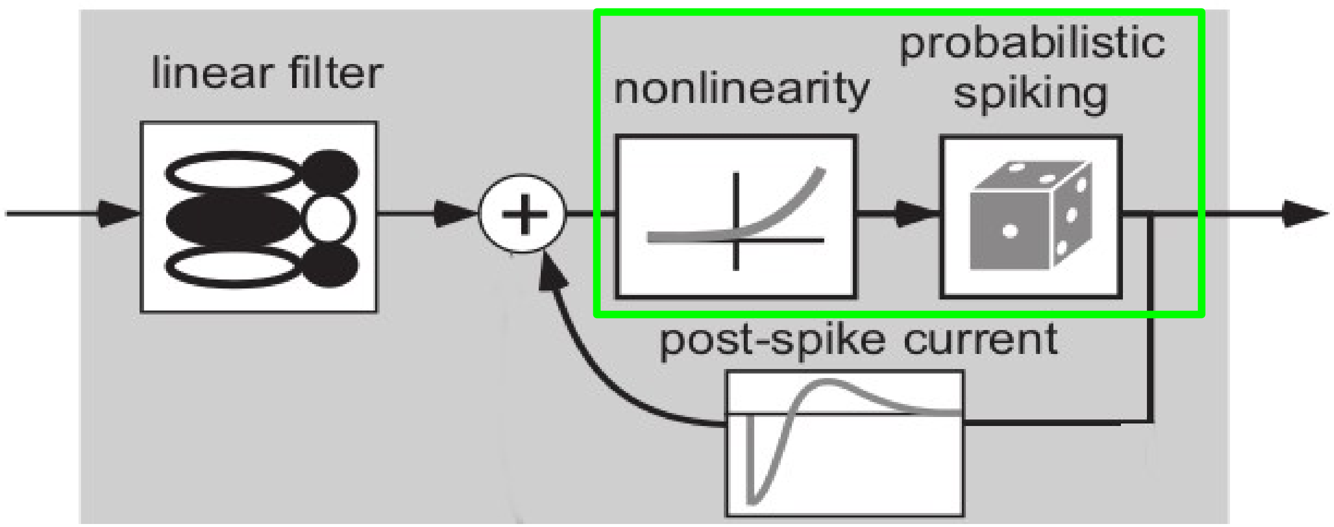
$X_{i,n+2}$: a vector of ones, encodes the baseline spike rate

The linear bit of the model

- A sample design matrix and response vector

y	X															
1	0.0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0001	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0001	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0003	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0005	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0.0005	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0.0007	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0.0010	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0.0010	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0.0010	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0.0010	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0.0010	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0.0010	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0.0010	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0.0055	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0.0055	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0.0057	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0057	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0057	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0063	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0.0064	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0064	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0116	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0117	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0117	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0.0119	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0.0125	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0.0125	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0.0127	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0.0127	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

The nonlinear bit of the model



The nonlinear bit of the model

- The spike rate is necessarily positive
- We choose exp() as the nonlinearity for convenience

The distribution

- An appropriate noise distribution for a neural system should make it such that
  - Only non-negative integer numbers of spikes are possible
  - The variance of the number of spikes should scale with the mean of the number of spikes (Maurice's lecture)

Poisson

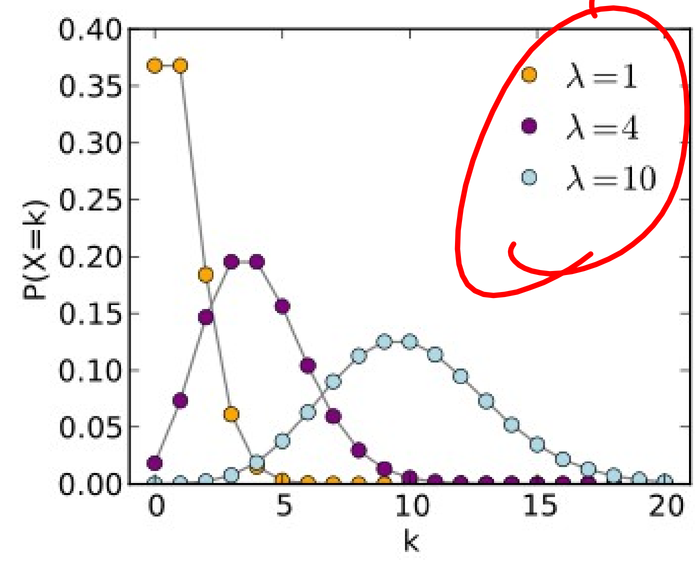
The nonlinear bit of the model

- The logical distribution is the Poisson:

Mean

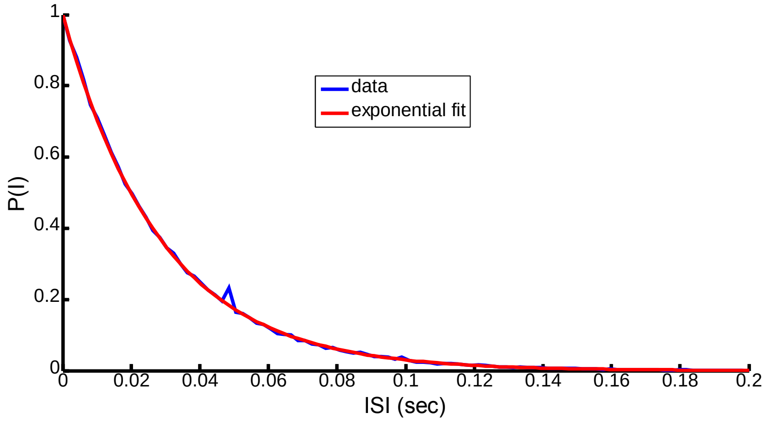
$$y_i \sim Poisson(\exp(\eta_i))$$
$$y \sim \prod Poisson(\exp(\eta))$$

ΣX

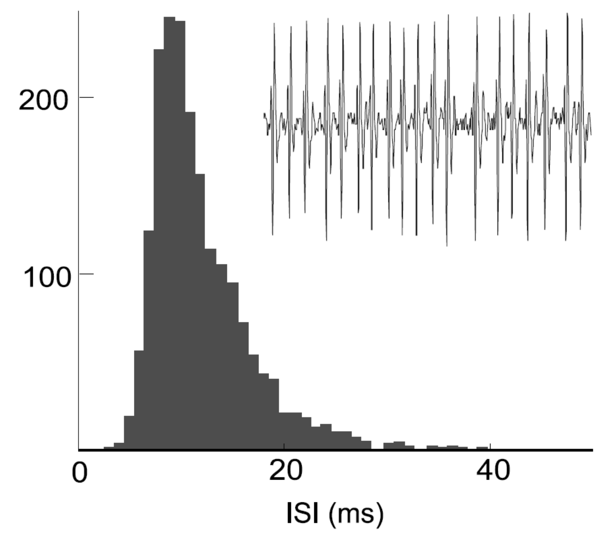


The nonlinear bit of the model

- Although we assume that the rate is Poisson conditioned on previous observations, the resulting model has non-exponential ISIs



Irregular afferent



A bit of perspective

- We've coaxed our neuron model to have the shape of a Generalized Linear Model with exponential nonlinearity, Poisson-distributed noise
- We're incorporating simple but non-trivial neuronal facts, Poisson noise and a refractory period via a post-spike filter
- Despite this, finding model parameters will be almost as simple as in linear regression

A bit of perspective

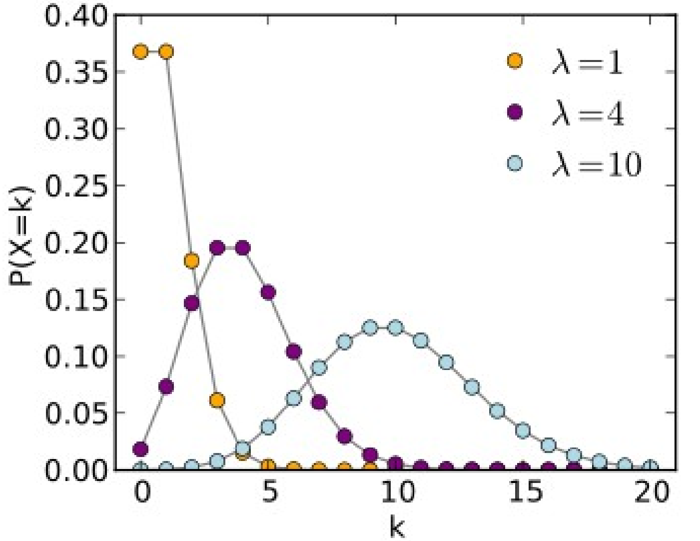
- We're trying to find “the best parameters” that describe this neuron
- Curtis Baker's lecture said: measure the mismatch between the data and the model predictions, and minimize that with respect to the model parameters through optimization
- Let's do what Curtis says

A natural measure of (mis)match

$$y_i \sim Poisson\left(\exp\left(\eta_i\right)\right)$$
$$y \sim \prod Poisson\left(\exp\left(\eta\right)\right)$$

A natural measure of (mis)match

$$f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!},$$



$$p(y_i|r_i) = r_i^{y_i} \exp(-r_i) / y_i!$$

A natural measure of (mis)match

$$-\log p(y_i | \exp(\eta_i)) = \exp(\eta_i) - y_i \eta_i$$

$$L(y, r) = -\log p(y | \exp(\eta)) = \sum_i \exp(\eta_i) - y_i \eta_i$$

A natural measure of (mis)match

- What if we had assumed instead that:

$$y_i \sim Normal(\eta_i, \sigma^2)$$

A natural measure of (mis)match

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$p(y_i|\eta_i) = k \exp\left(-\frac{1}{2\sigma^2}(y_i - \eta_i)^2\right)$$

A natural measure of (mis)match

$$-\log p(y_i|\eta_i) = \frac{1}{2\sigma^2}(y_i - \eta_i)^2$$

$$-\log p(y|\eta) = \frac{1}{2\sigma^2}\sum_i (y_i - \eta_i)^2$$

A natural measure of (mis)match

- For a given output nonlinearity and noise model, we can compute the negative log-likelihood of the data
- The negative log-likelihood is the natural measure of misfit between data and predictions
- For Gaussian noise, the negative log-likelihood is the familiar sum-of-squares error; for Poisson noise, it's different
- Minimizing this quantity with respect to the model parameters gives the Maximum Likelihood (ML) estimate of the model parameters

A bit more theory (optional)

- What we're really trying to maximize is:

$$p(w|y)$$

- Bayes' theorem says that:

$$p(w|y) \propto p(y|w)\, p(w)$$

A bit more theory (optional)

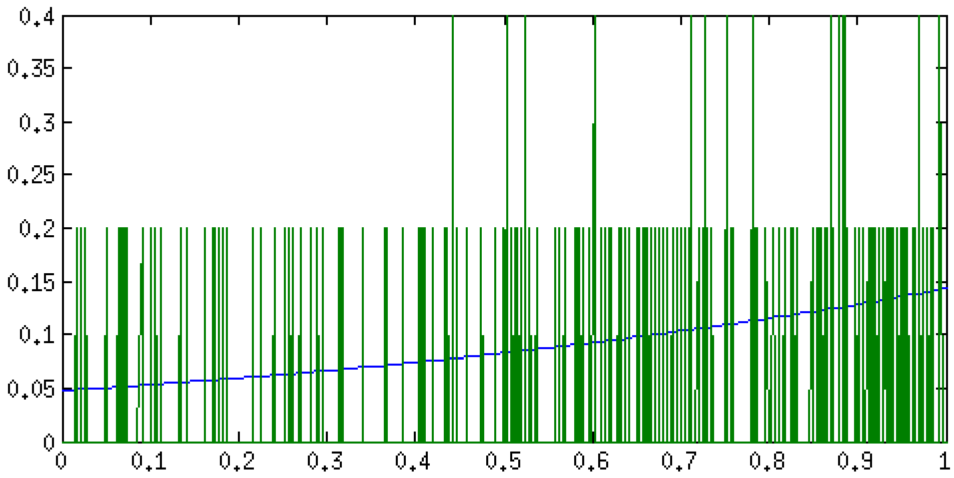
- If p(w) is flat, then maximizing the posterior is equivalent to maximizing the likelihood
- Exercise: if $p(w) = Normal(0, \gamma^2)$

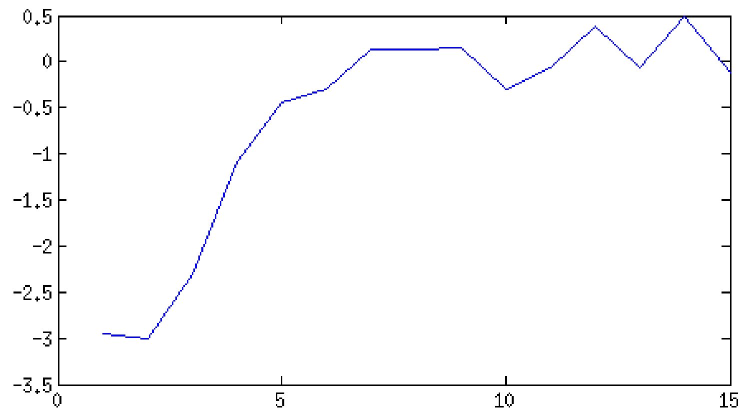
  then maximizing the posterior is equivalent to minimizing:

$$E = L + \frac{1}{2\gamma^2} \sum_j w_j^2$$

Not coincidentally, this corresponds to weight decay, which Curtis covered in his last lecture

Example model fits





What's a Generalized Linear Model

- In a generalized linear model,

$$\eta_i = \sum_j X_{ij} w_j$$
$$r_i = f(\eta_i)$$
$$y_i \sim Distribution(r_i, params)$$

- Inference is done by finding:

$$arg\, min_w L(y_i, r_i)$$

Analyzing goodness-of-fit

- The negative log-likelihood is a perfectly valid measure of goodness-of-fit
- The negative log-likelihood of the model is typically baselined against the negative log-likelihood of a model with only a constant offset
- Rule of thumb (based on Akaike Information Criterion): adding a noise parameter adds about 1 unit of negative log-likelihood
- In GLM parlance, twice the negative log-likelihood is called the deviance

Analyzing goodness-of-fit

- You can also compute the minimum negative log-likelihood L_min when r = y and derive:

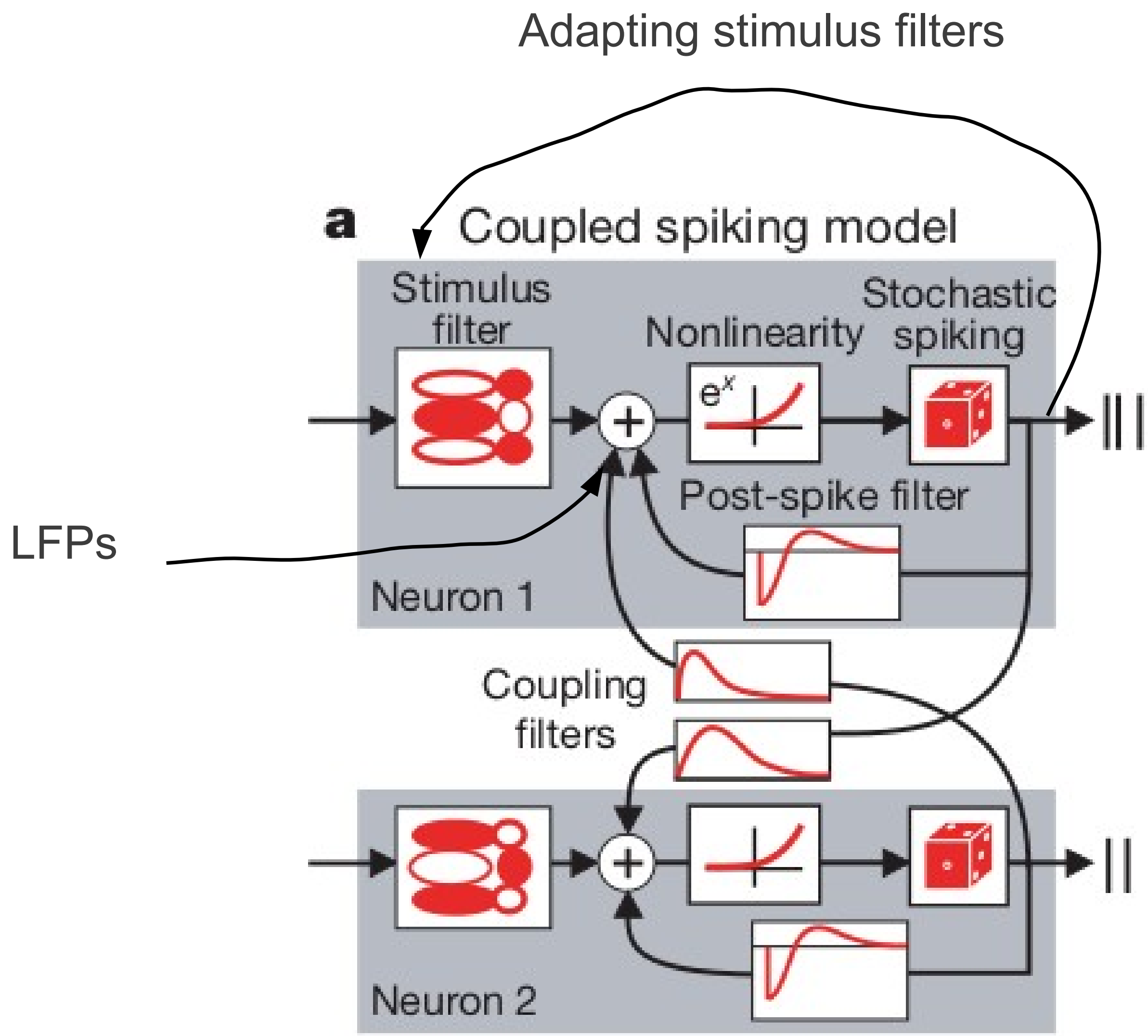
$$D^2 = 1 - \frac{L_{model} - L_{min}}{L_{baseline} - L_{min}}$$

- Exercise: show this is the same as R-squared when the GLM is normal-identity-Gaussian

Generalized Linear Models

- There's a ton of flexibility in specifying the design matrix:
  - In addition to position and a post-spike current, we could add columns to the design matrix for coupling with other neurons, synching to certain LFP phases, etc.

More covariates



Generalized Additive Models

- Nevertheless, GLMs can't model everything
  - We can use them as a building block for more complex models

Generalized Additive Models

- The previous rat example assumed that the place field of the rat is linear with position
- What if the place field is localized at some intermediate location between [0,1]?
- That would imply that the relationship between spike rate and position is arbitrarily nonlinear

From GLMs to GAMs

- What if we replaced

$$\eta_i = \sum_j X_{ij} w_j$$

- With:

$$\eta_i = \sum_j g_j(X_{ij})$$

From GLMs to GAMs

- Each covariate has its own unidimensional nonlinearity
- The output of the nonlinearities is combined additively
- We keep the final nonlinearity and noise distribution
  - Generalized Additive Models

GLMs to GAMs

- If you use arbitrary forms for the nonlinearities g, you run into trouble: multiple local minima
- Trick: an arbitrary function g(x) can be approximated as a sum of localized basis functions

Localized basis functions

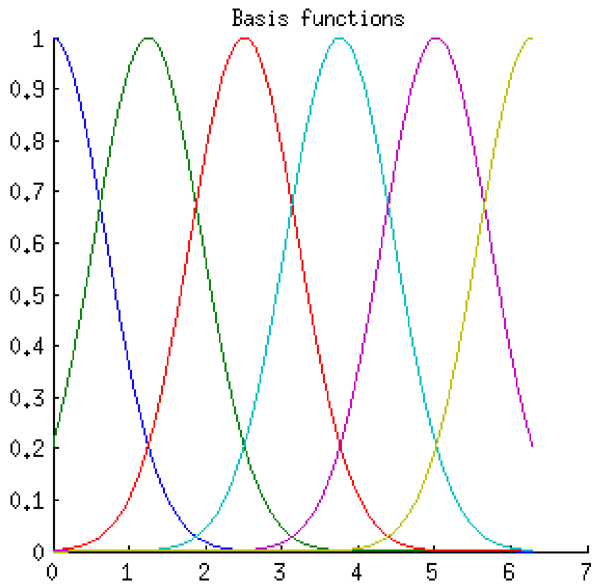
- Example: f(x) = sin(x) from 0 to 2pi can be described as:

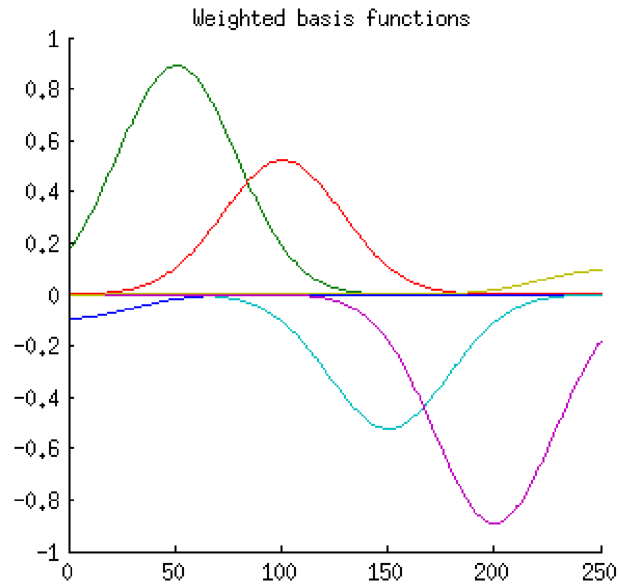
$$\sin(x) \approx \sum_{i=0}^{N} w_i \exp\left(\frac{-N^2}{2}(x - 2\pi i/N)^2\right)$$

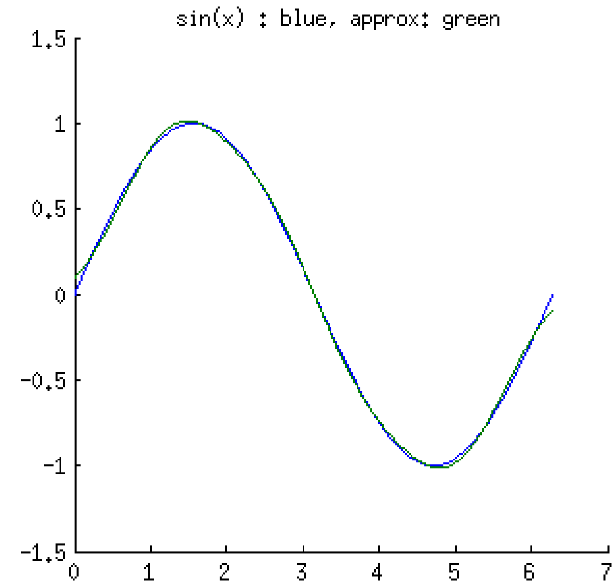
- The w_i can be determined by linear regression

Localized basis functions

- sin(x) example





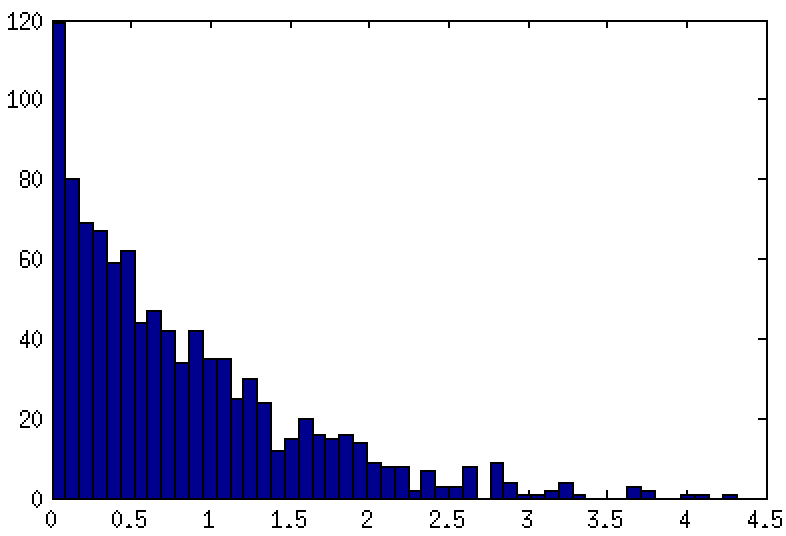


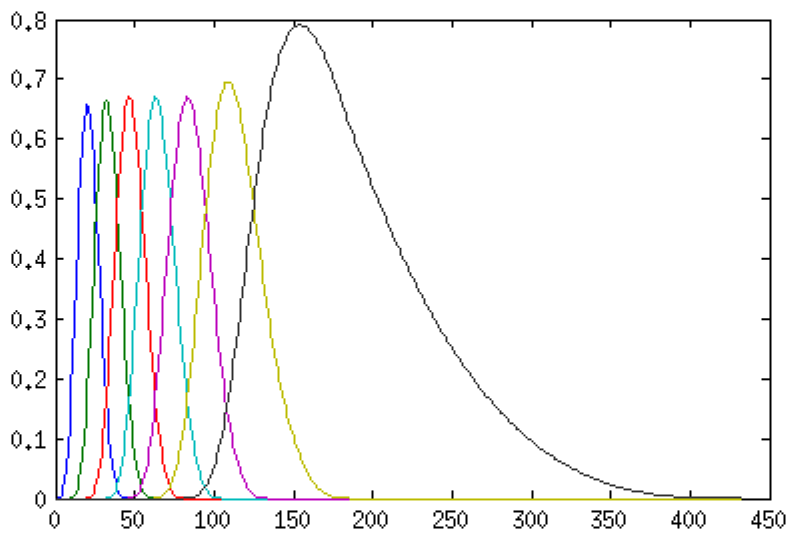
Localized basis functions

- Advantage: nonlinearities are entirely defined in terms of weighted sums of localized basis functions
- The output of the localized basis functions is precomputed once
- Once that's done, we're left with an augmented model that can be estimated with standard GLMs
- Instead of having one parameter per covariate, we have one parameter per basis function per covariate

Splines

- It would be nice if you could simply say where you want the basis functions to be centered, and by some fixed rule the basis functions would be built for you

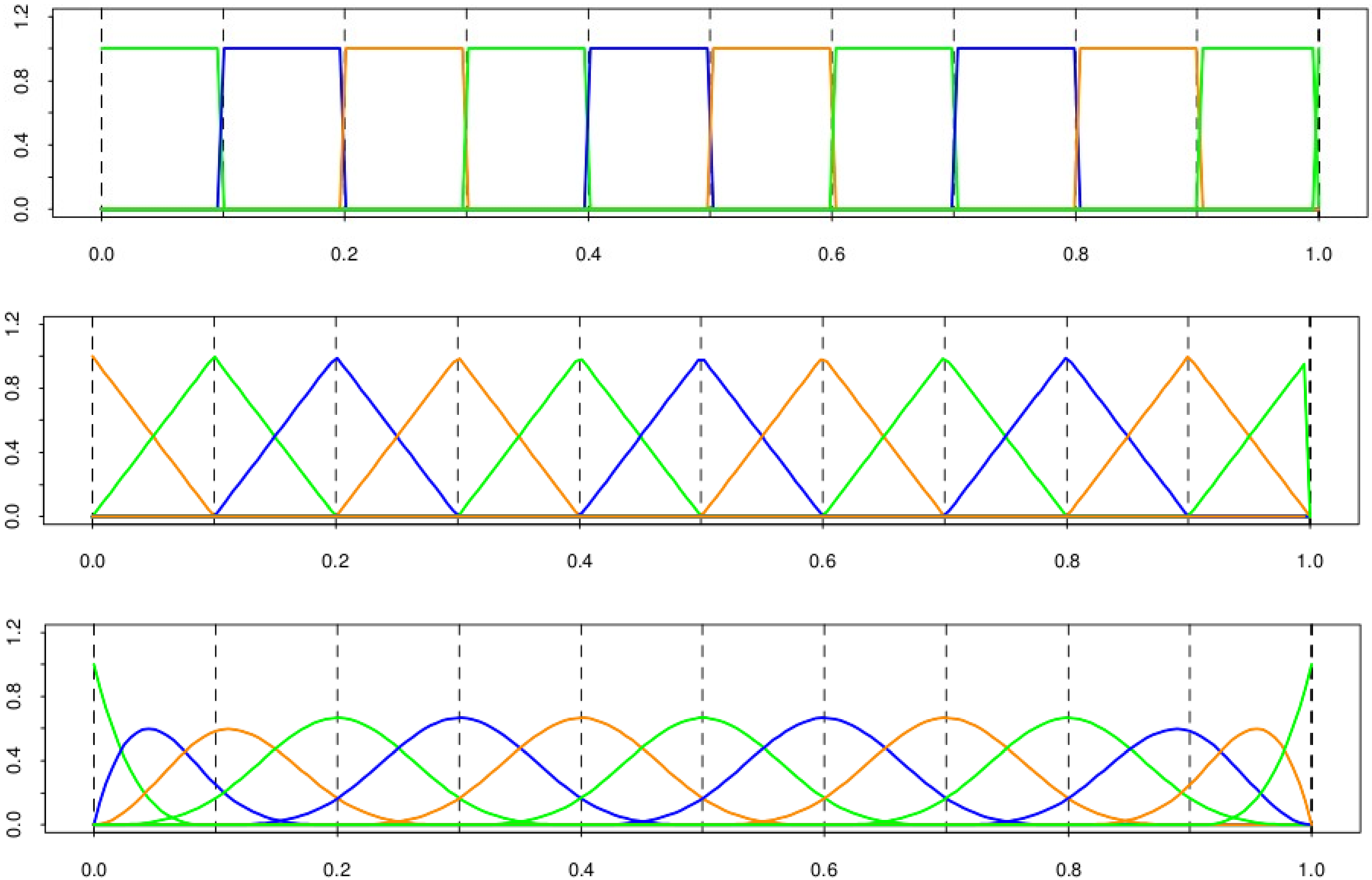




Splines

- B-splines (basis splines) do exactly that
- Piecewise polynomials (linear, cubic, etc.)
- Specified by knots which correspond (roughly) to the center of the desired basis functions
- Specified by an order which determines the degree of the polynomial and consequently the smoothness
  - Order 0: not continuous
  - Order 1: continuous
  - Order 2: first derivative is continuous
  - Order 3: second derivative is continuous
- A spline with N knots has N-order-1 associated basis functions (and consequently weights)

Splines



Hastie, Tibshirani & Friedman 2009

Splines

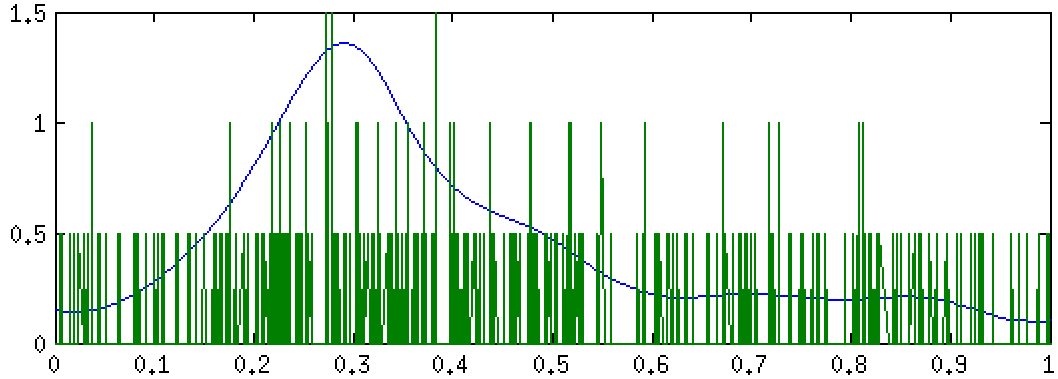
- They have a bunch of nice mathematical properties
  - They're maximally smooth given some constraints
  - They have compact support
  - The derivative of a degree N spline is another spline of degree N - 1

Back to the rat

- Replace the first column of the design matrix with several columns corresponding to a 3rd order B-spline with equispaced knots
- Remove the last column of the design matrix corresponding to the offset
- Fit just like before

Rat example fit

- You can do it!

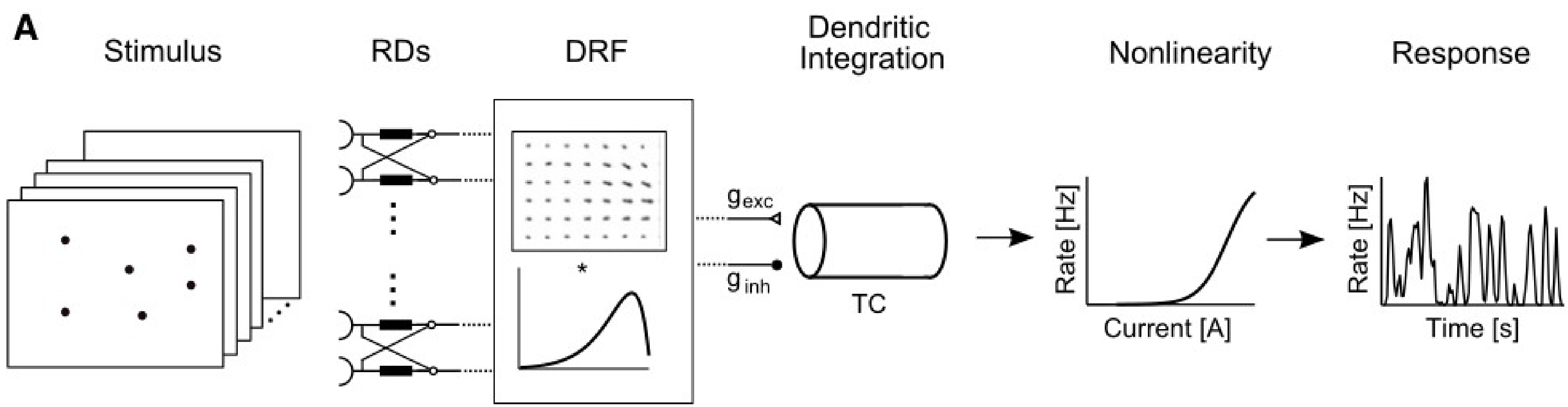


Conclusions

- GLMs allow you to work with binary or count data almost as easily as continuous data
- Very flexible formulation
  - Flexible design matrix
  - Flexible distribution
  - Flexible nonlinearity (non-canonical)
- GAMs expand GLMs to allow one-dimensional (or 2D or 3D) nonlinearities
  - Spatial models
  - Input nonlinearities
  - Etc.

Conclusions

- Going from a metaphor for a neuron to a real neuron model without losing tractability



Weber, Machens and Borst 2010

Further reading

- On GLMs applied to neural data:
  - Liam Paninski's lecture notes (http://www.stat.columbia.edu/~liam/teaching/neurostat-spr11/)
  - Paninski L., Maximum likelihood estimation of cascade point-process neural encoding models (2004)
  - Simoncelli, Paninski, Pillow, Schwartz, Characterization of neural responses with stochastic stimuli (2004)
  - Brown et al. (1998) A Statistical Paradigm for Neural Spike Train Decoding Applied to Position Prediction from Ensemble Firing Patterns of Rat Hippocampal Place Cells
- On GLMs and GAMs:
  - Generalized additive models: An Introduction with R by Simon Wood
- On splines:
  - Chapter 5 of Hastie, Tibshirani & Friedman (online: http://www-stat.stanford.edu/~tibs/ElemStatLearn/)