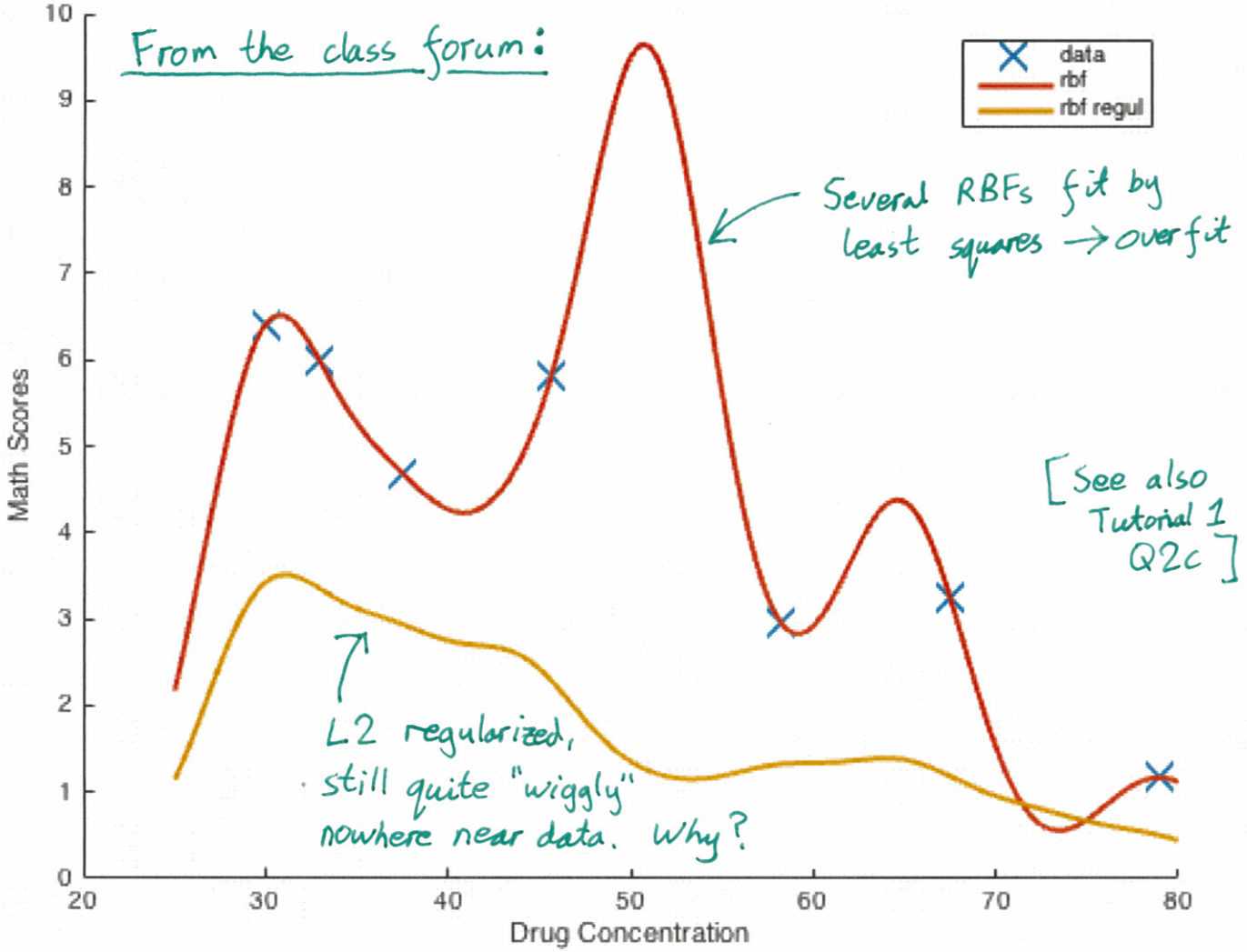
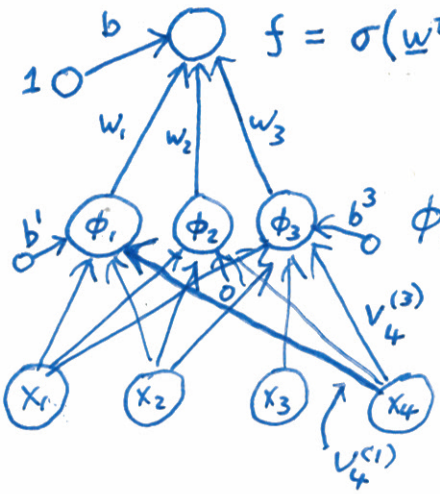

From the class forum:
data
rbf
rbf regul
Several RBFs fit by least squares → overfit
[See also Tutorial 1 Q2c]
L2 regularized, still quite "wiggly" nowhere near data. Why?
Math Scores
Drug Concentration
0
1
2
3
4
5
6
7
8
9
10
20
30
40
50
60
70
80

Neural Networks

First example:



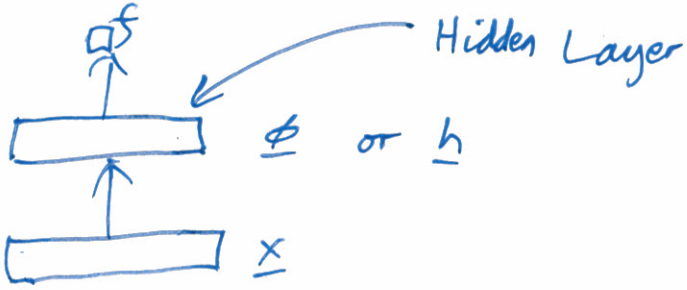
Fit $\{\{\underline{v}^{(k)}, b^{(k)}\}, \underline{w}, b\}$ with a gradient-based optimizer. Match f to training targets.

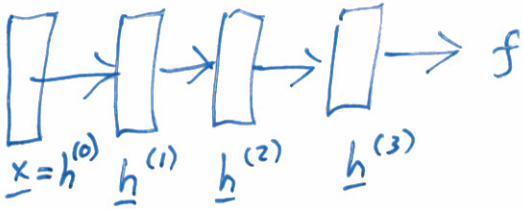
Vocabulary

"Unit" : One of the intermediate computations

"Hidden units": Also "Latents"

"Visibles" are the inputs, $\underline{x}$'s





$$f = g(\underline{w}^T \underline{h}^{(3)} + b)$$

$$\underline{h}^{(\ell)} = g^{(\ell)}(\underbrace{W^{(\ell)} \underline{h}^{(\ell-1)} + \underline{b}^{(\ell)}}_{\underline{a}})$$

↑ some non-linear function, applied element-wise

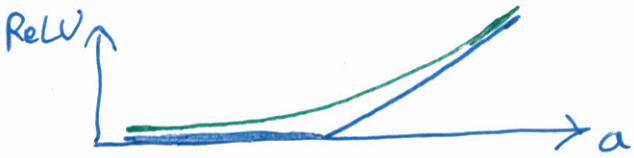
Non-linearities?

These are what we used for basis fn's

Sigmoid σ : softly partitions the space.

RBFs : are you near some region

ReLU : Rectified linear unit



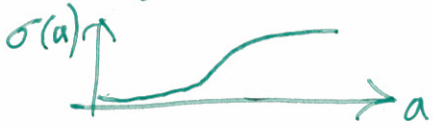
Hard ReLU

$= \max(a, 0)$, Relu

Soft ReLU

$= \log(1 + e^a)$

Easier to fit than:



Practical Fitting Tips

Set initial weight matrices $W^{(\ell)}$ (and the biases)

Must not set to $\underline{0}$!

Symmetry means all the hidden units behave the same.

→ Randomly set the weights

Naive idea each weight $\sim N(0,1)$

randn()

If we have k hidden units connecting to next layer, those are combined

$$h_k^{(\ell)} = g\left(\underbrace{W_{k:}^{(\ell)} h^{(\ell-1)}}_{\text{Typically very roughly this value scales with } \sqrt{k}} + b_k^{(\ell)}\right)$$

Sigmoid:

Example Initialization:

$$w \sim N\left(0, \left(0.01/\sqrt{k}\right)^2\right)$$