MARS: Insights from Carts, Insights for Nets

Trent McConaghy

Berlin Machine Learning Group
Feb 2, 2015

ascribe

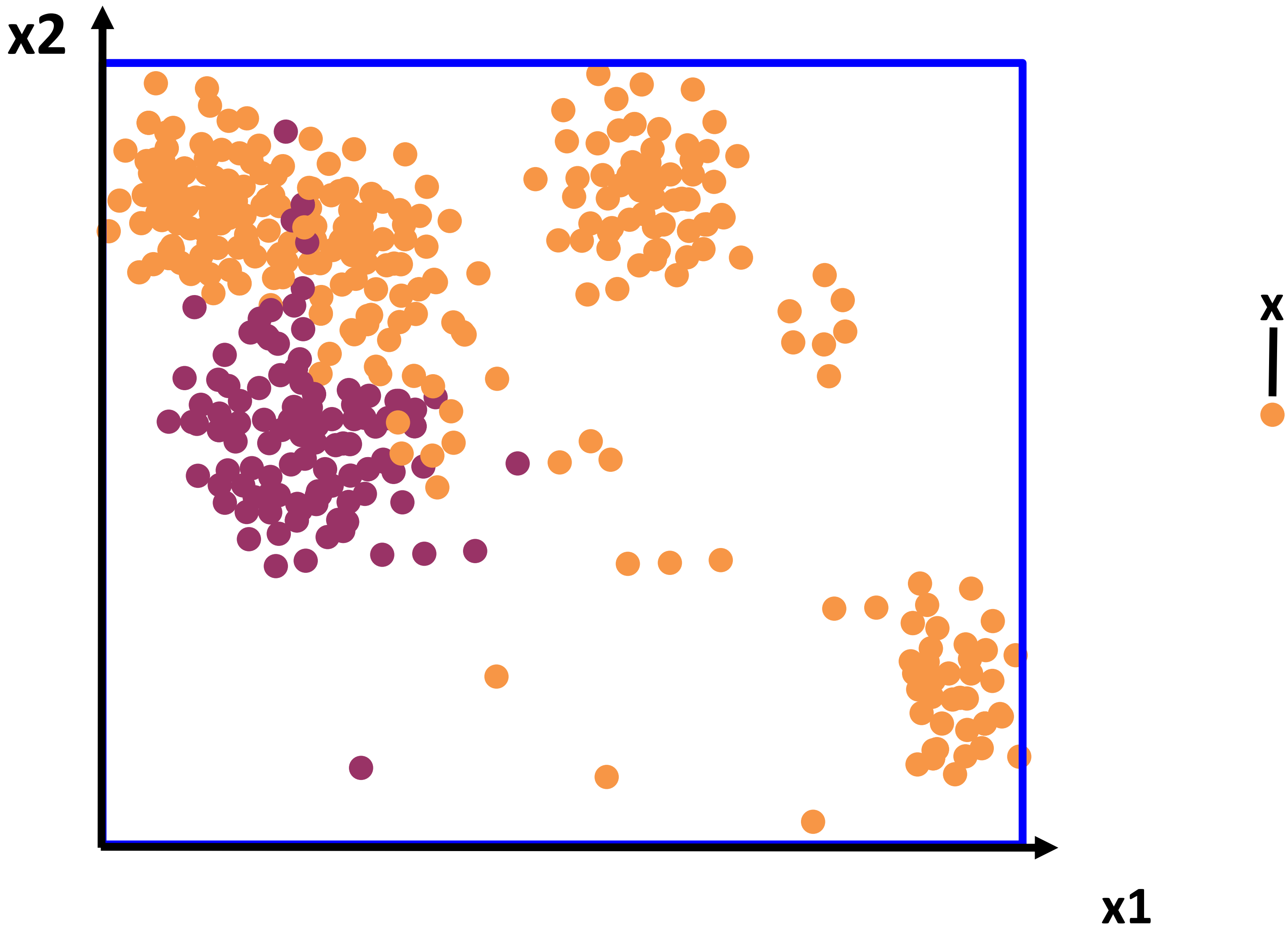
Outline

- Carts
- Mars
- Nets

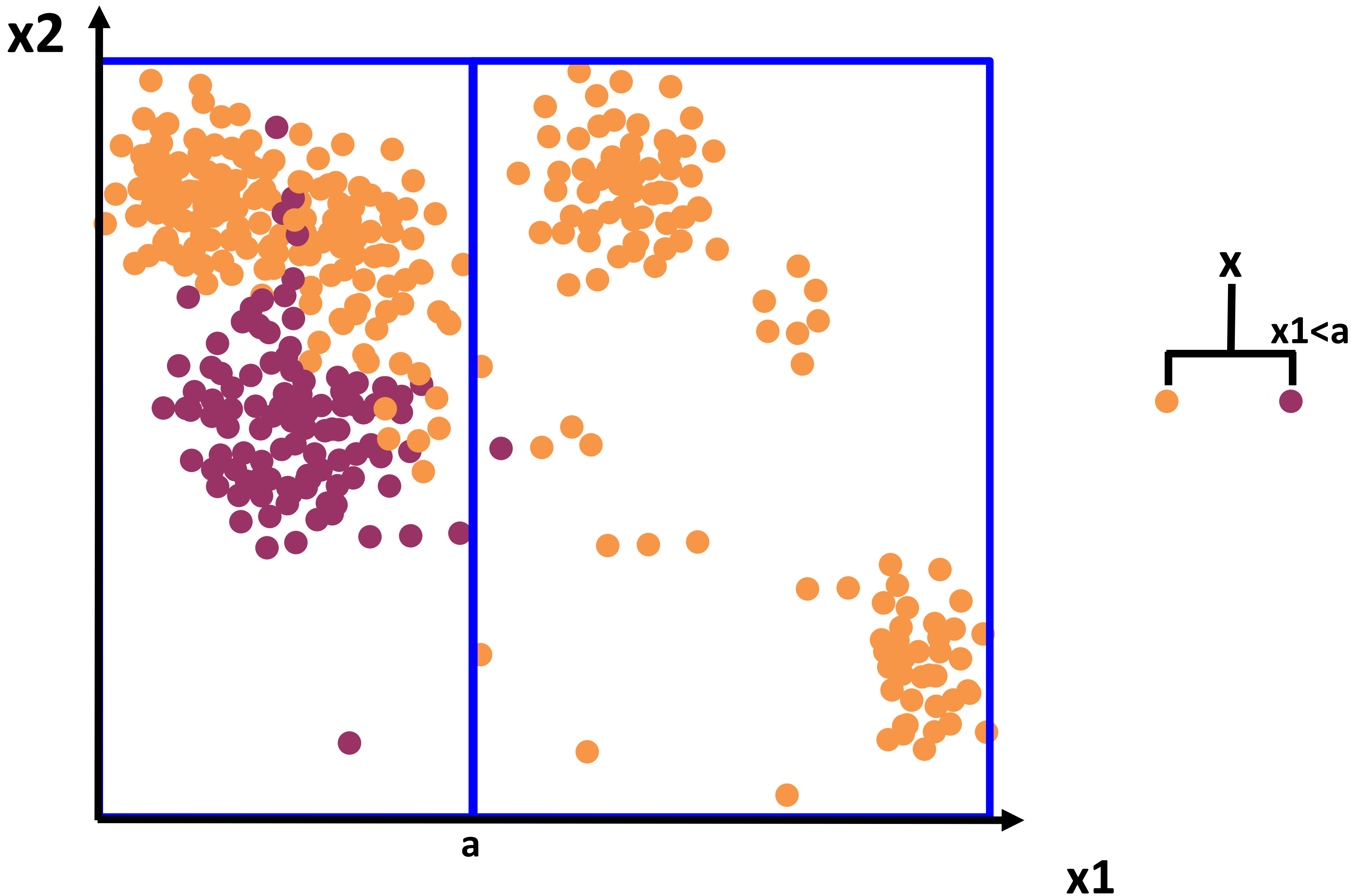
Outline

- <u>Carts</u>
- Mars
- Nets

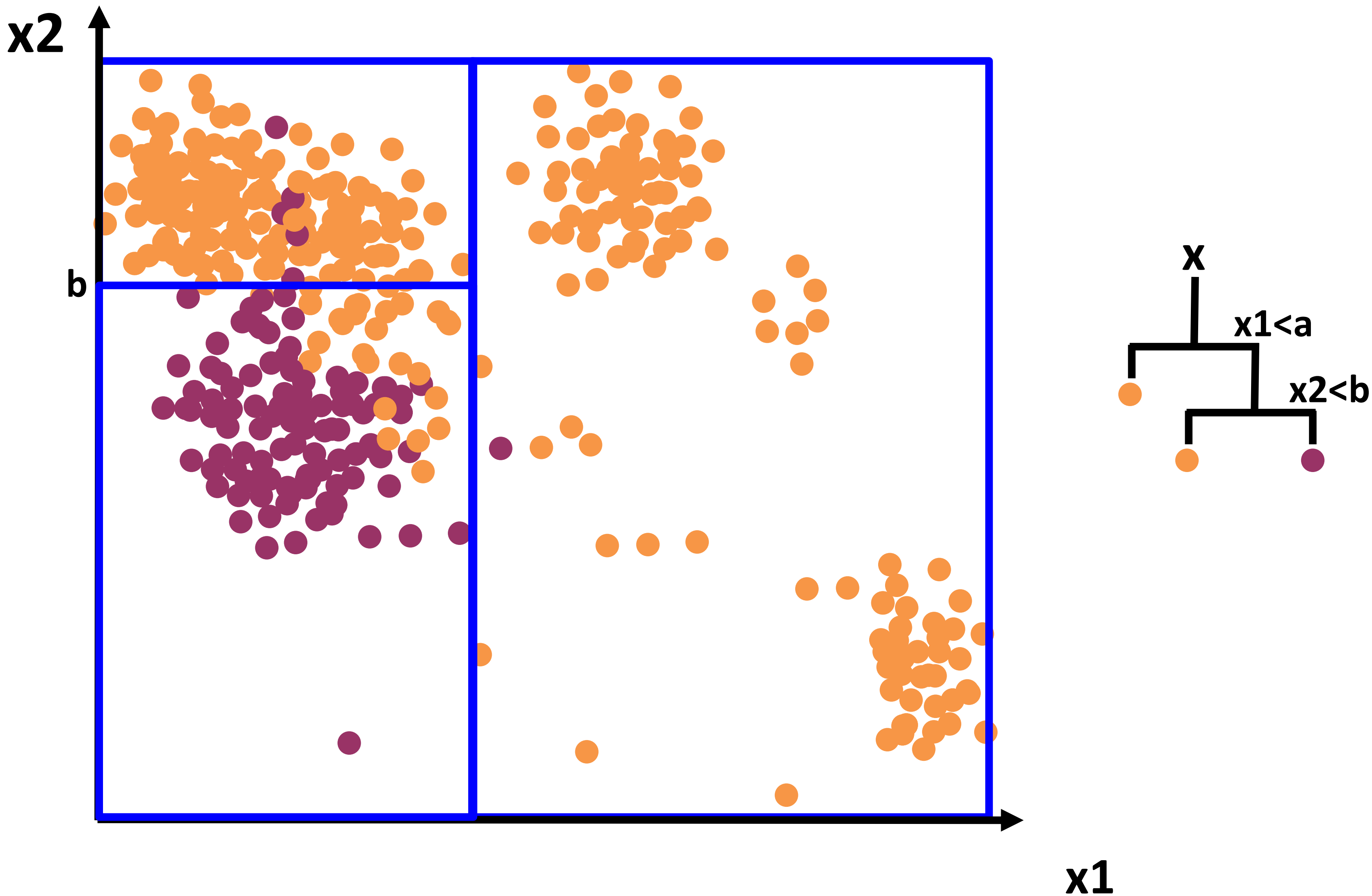
CART Classifier



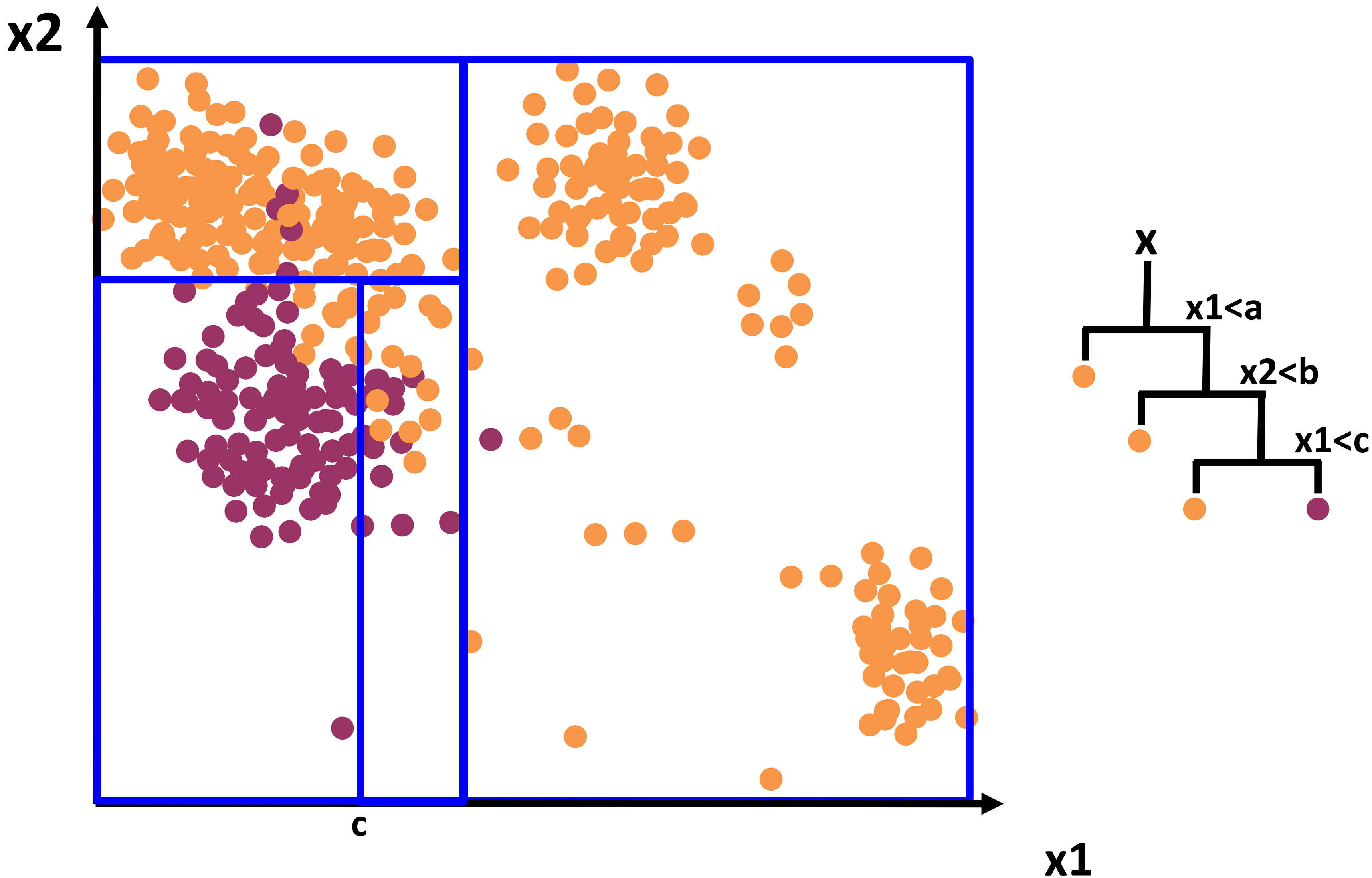
CART Classifier



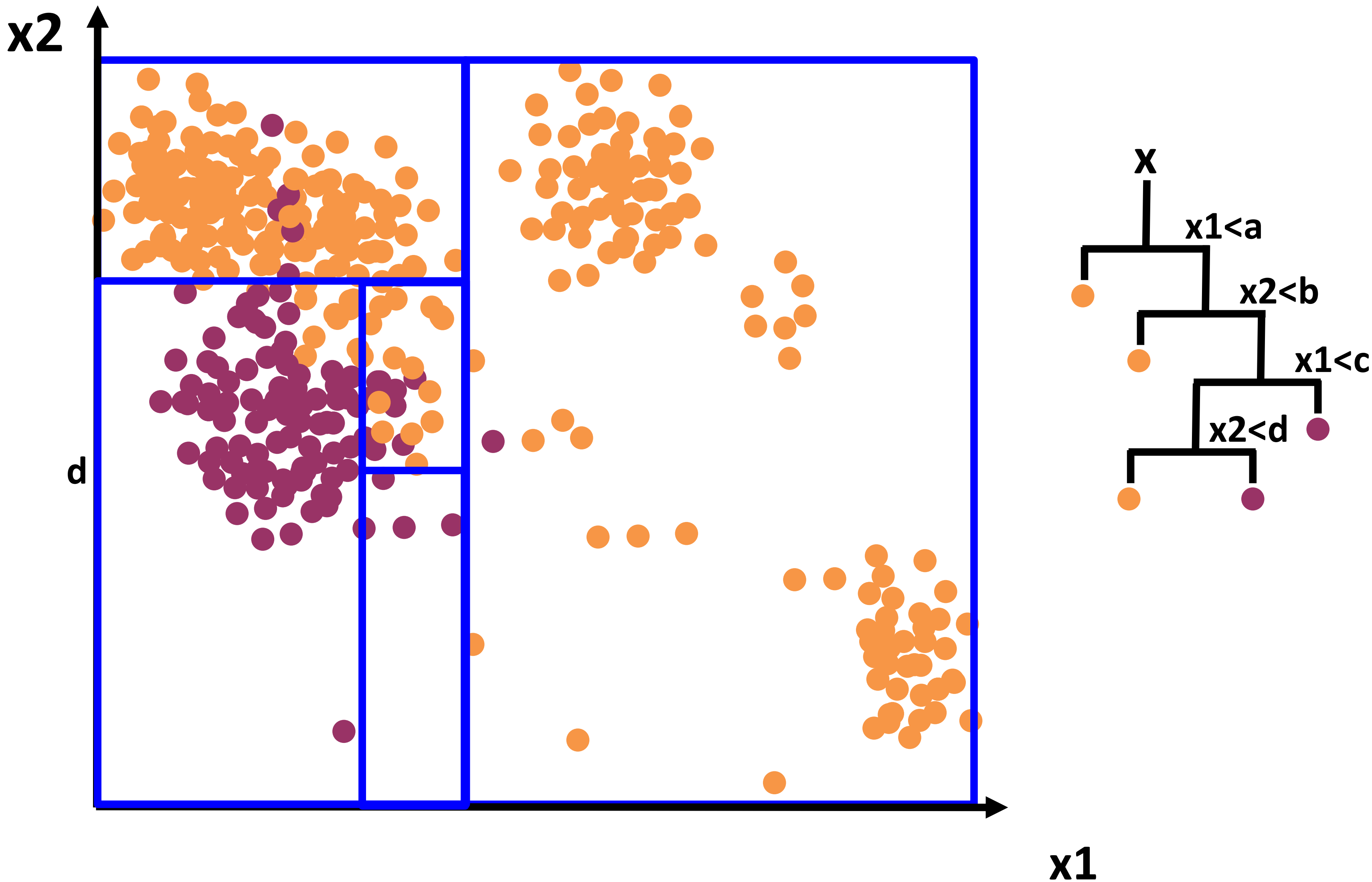
CART Classifier



CART Classifier



CART Classifier

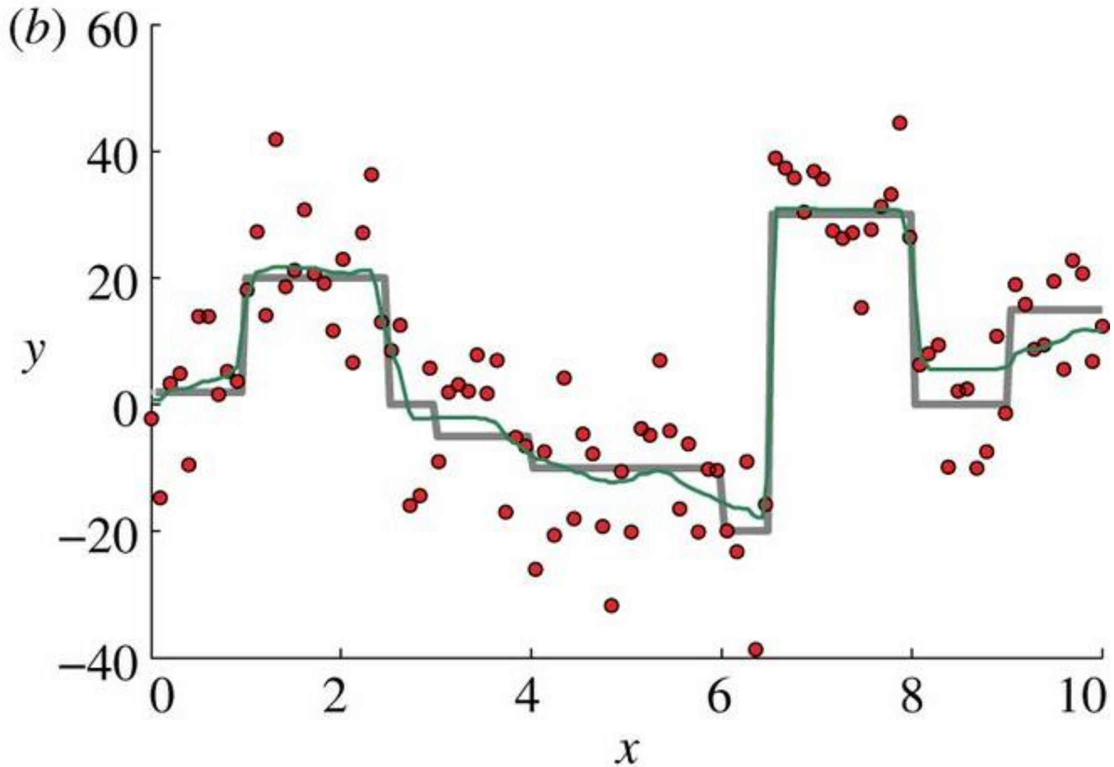


On CARTs

- Adaptively zooms in on the most important regions. Basis functions zero out the rest.
- Computationally cheap

CART Regressor

- Is a weighted sum of pairs of indicator functions
- = a piecewise constant model



http://rsta.royalsocietypublishing.org/content/roypta/371/1984/20110547/F3.large.jpg

CART Learning

$B_1(\mathbf{x}) \leftarrow 1$

For $M = 2$ to $M_{\max}$ do: $\text{lof}^* \leftarrow \infty$

iteratively add M_{max} regions (basis funcs)

$\quad$ For $m = 1$ to $M - 1$ do:

$\quad\quad$ For $v = 1$ to n do:

$\quad\quad\quad$ For $t \in \{x_{vj} | B_m(\mathbf{x}_j) > 0\}$

test each split: {region to split} x {split variable} x {split value}

$\quad\quad\quad\quad g \leftarrow \sum_{i \neq m} a_i B_i(\mathbf{x}) + a_m B_m(\mathbf{x}) H[+(x_v - t)] + a_M B_m(\mathbf{x}) H[-(x_v - t)]$

$\quad\quad\quad\quad \text{lof} \leftarrow \min_{a_1, \ldots, a_M} \text{LOF}(g)$

$\quad\quad\quad\quad$ if $\text{lof} < \text{lof}^*$, then $\text{lof}^* \leftarrow \text{lof}$; $m^* \leftarrow m$; $v^* \leftarrow v$; $t^* \leftarrow t$ end if

measure error, update best split

$\quad\quad\quad$ end for

$\quad\quad$ end for

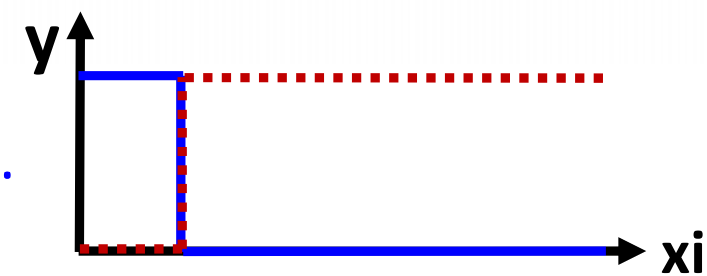
$\quad$ end for

$B_M(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x}) H[-(x_{v^*} - t^*)]$

$B_{m^*}(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x}) \underbrace{H[+(x_{v^*} - t^*)]}$

add best split, in both directions

indicator function building blocks.
Discontinuous..



Friedman, J. H. (1991). "Multivariate Adaptive Regression Splines". The Annals of Statistics 19: 1

CARTs Live On

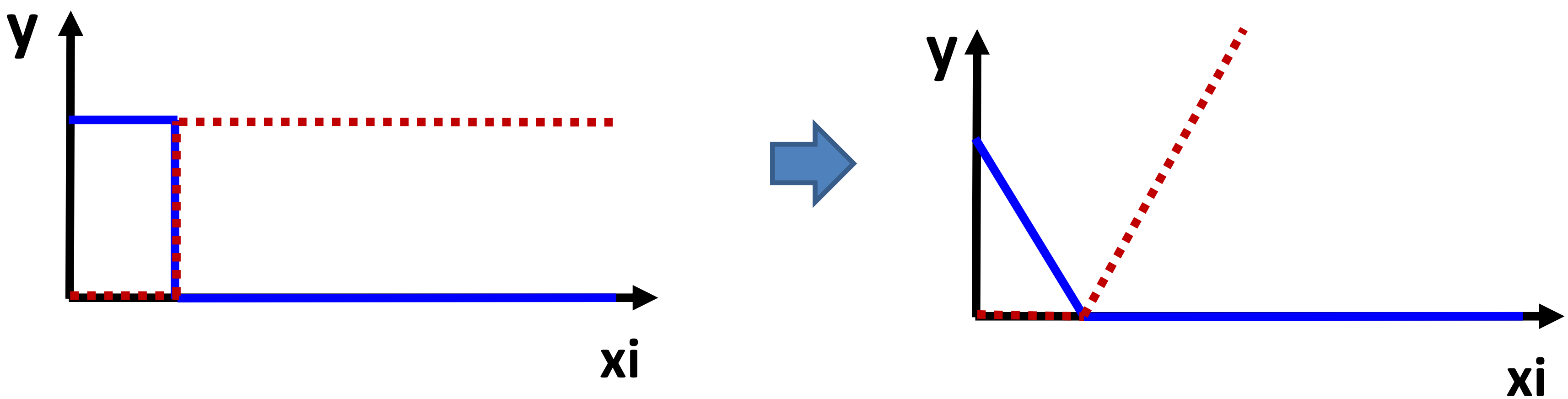
- Mostly as bagged ensembles (RFs)
- State of the art performance on many problems
- Linear in # variables, # samples

Outline

- Carts
- <u>Mars</u>
- Nets
- Links

MARS Learning

- Start with CART algorithm
- Indicator functions → hinge functions
- That’s it!



MARS Learning

$B_1(\mathbf{x}) \leftarrow 1;\ M \leftarrow 2$

Loop until $M > M_{\max}$: lof$^* \leftarrow \infty$ — iteratively add M_{max} regions (basis funcs)

For $m = 1$ to $M-1$ do:

For $v \notin \{v(k,m) \mid 1 \le k \le K_m\}$

For $t \in \{x_{vj} \mid B_m(\mathbf{x}_j) > 0\}$ — test each split: {region to split} x {split variable} x {split value}

$g \leftarrow \sum_{i=1}^{M-1} a_i B_i(\mathbf{x}) + a_M B_m(\mathbf{x})[+(x_v - t)]_+ + a_{M+1} B_m(\mathbf{x})[-(x_v - t)]_+$

lof $\leftarrow \min_{a_1,\ldots,a_{M+1}} \mathrm{LOF}(g)$

if lof $<$ lof*, then lof$^* \leftarrow$ lof; $m^* \leftarrow m;\ v^* \leftarrow v;\ t^* \leftarrow t$ end if — measure error, update best split

end for

end for

end for

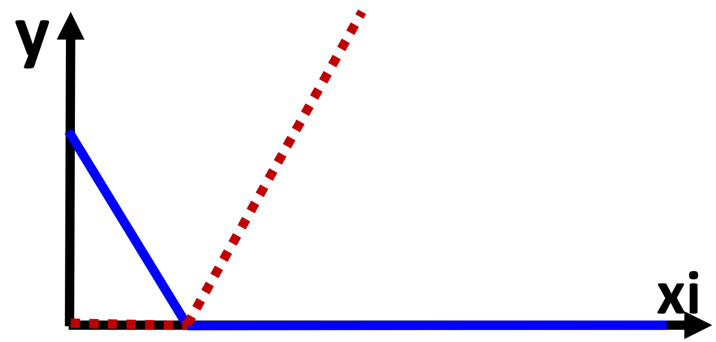
$B_M(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x})[+(x_{v^*} - t^*)]_+$

$B_{M+1}(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x})\underbrace{[-(x_{v^*} - t^*)]_+}$

— add best split, in both directions

$M \leftarrow M + 2$

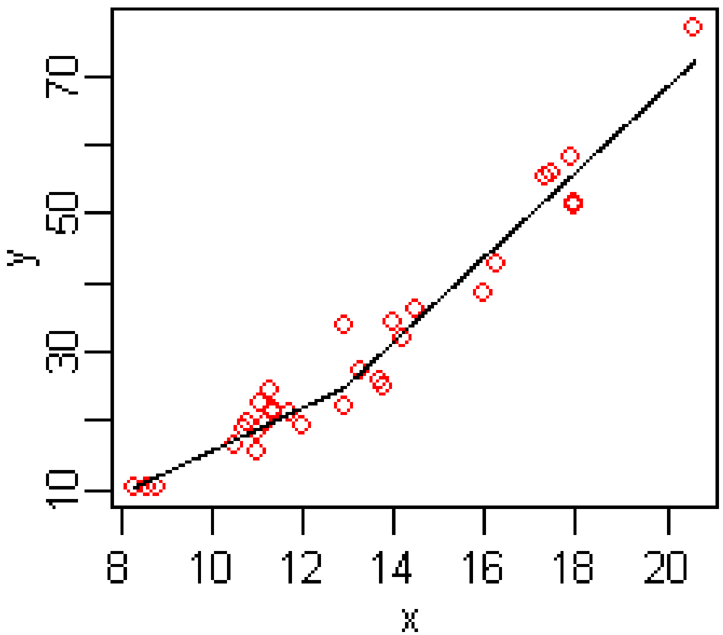
Hinge function building blocks.

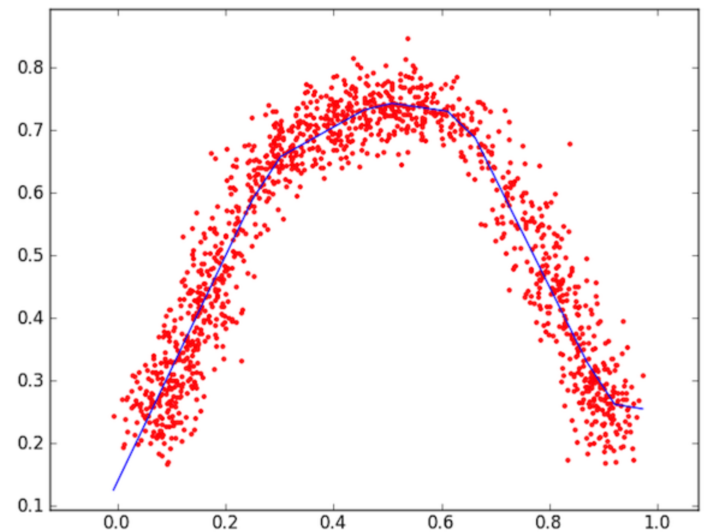


Friedman, J. H. (1991). "Multivariate Adaptive Regression Splines". The Annals of Statistics 19: 1

MARS Regressor

- Is a weighted sum of pairs of hinge functions
- = PWL model



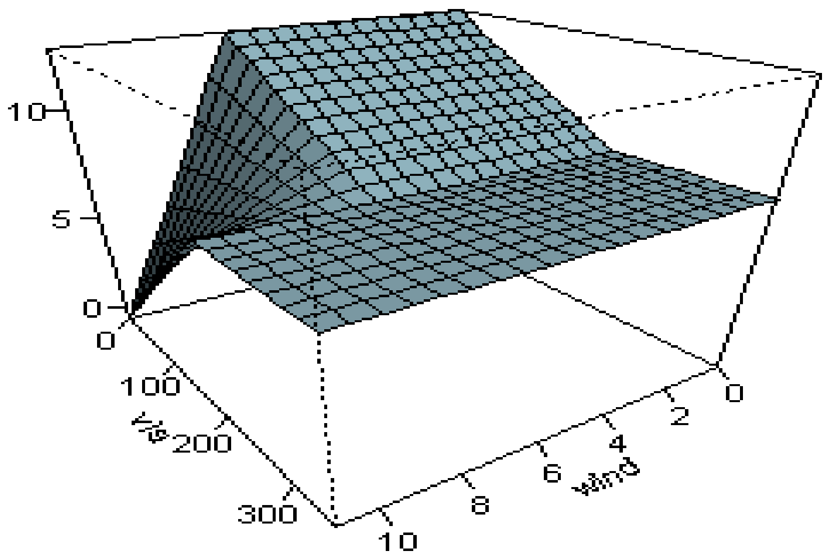


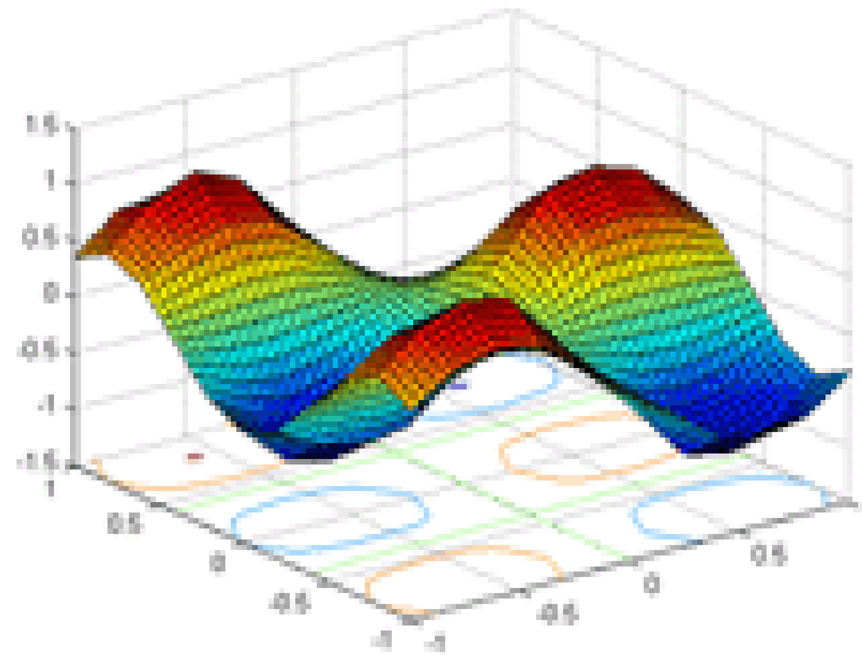
- Actually, we can have squared, cubic, .. terms
- = Piecewise polynomial = Spline

http://en.wikipedia.org/wiki/Multivariate_adaptive_regression_splines

http://blog.biolab.si/wp-content/uploads/2011/12/13/earth_demo_2.png__600x470_q95_subject_location-407%2C297.png

MARS Regressor





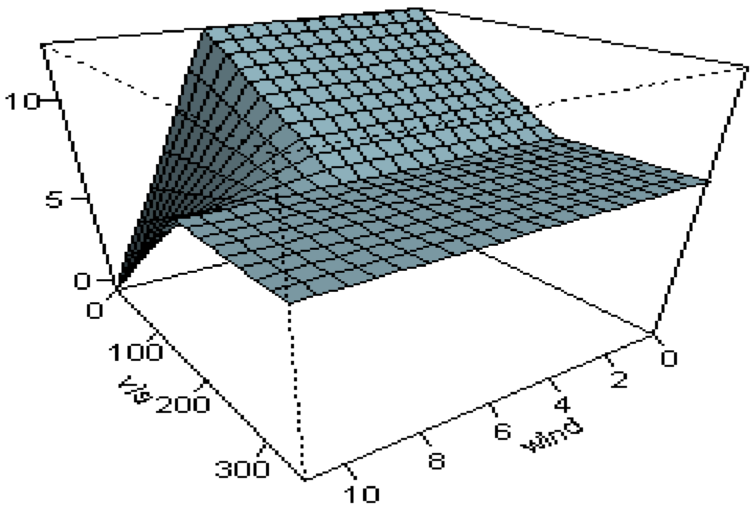
Adaptively zooms in on the most important regions.

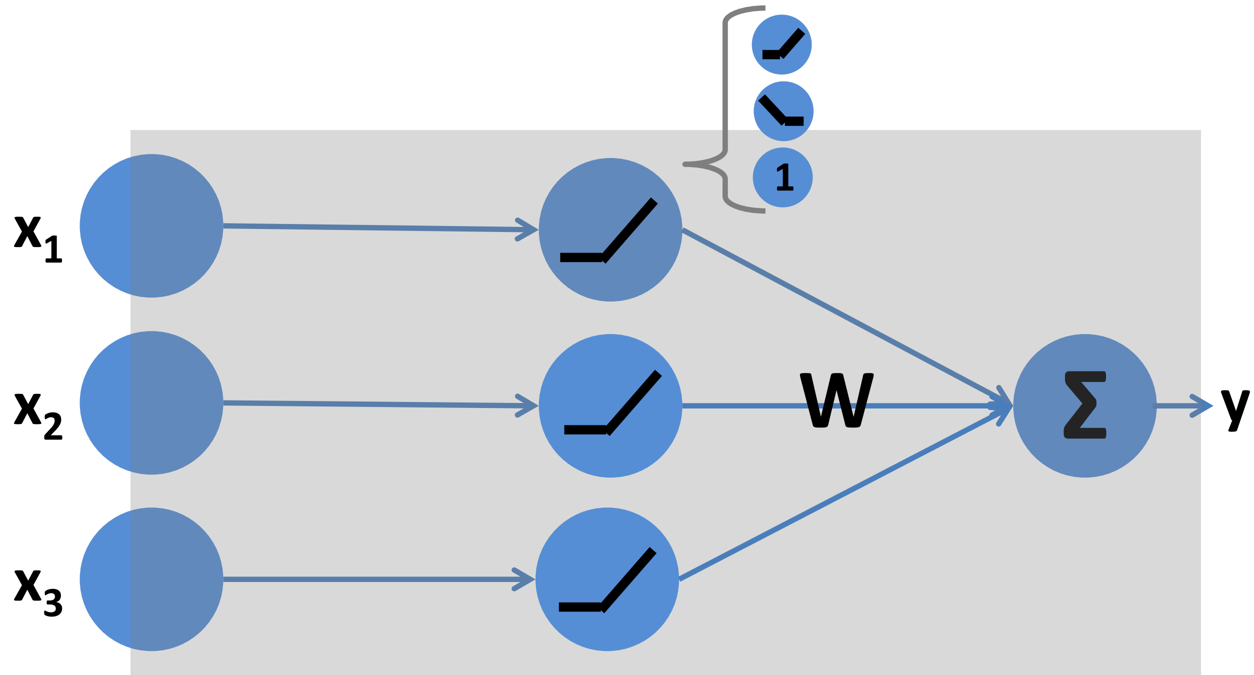
Basis functions zero out the rest.

http://en.wikipedia.org/wiki/Multivariate_adaptive_regression_splines

http://www.cs.rtu.lv/jekabsons/TwoSurfaces.gif

MARS with 1 layer

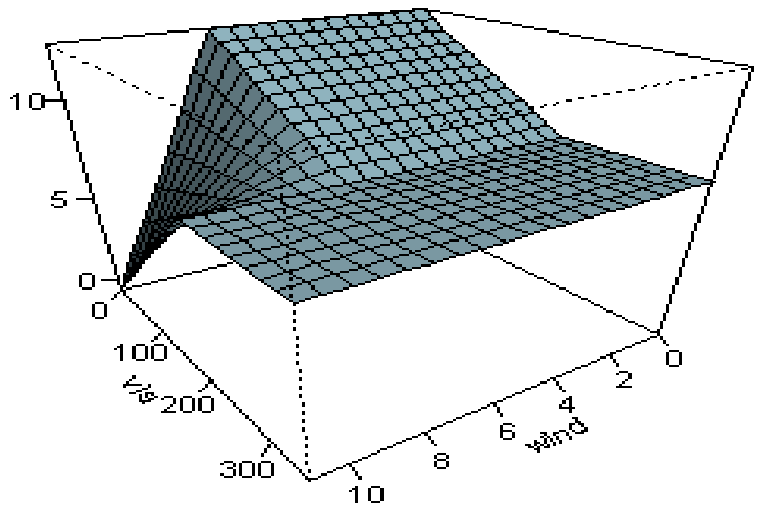


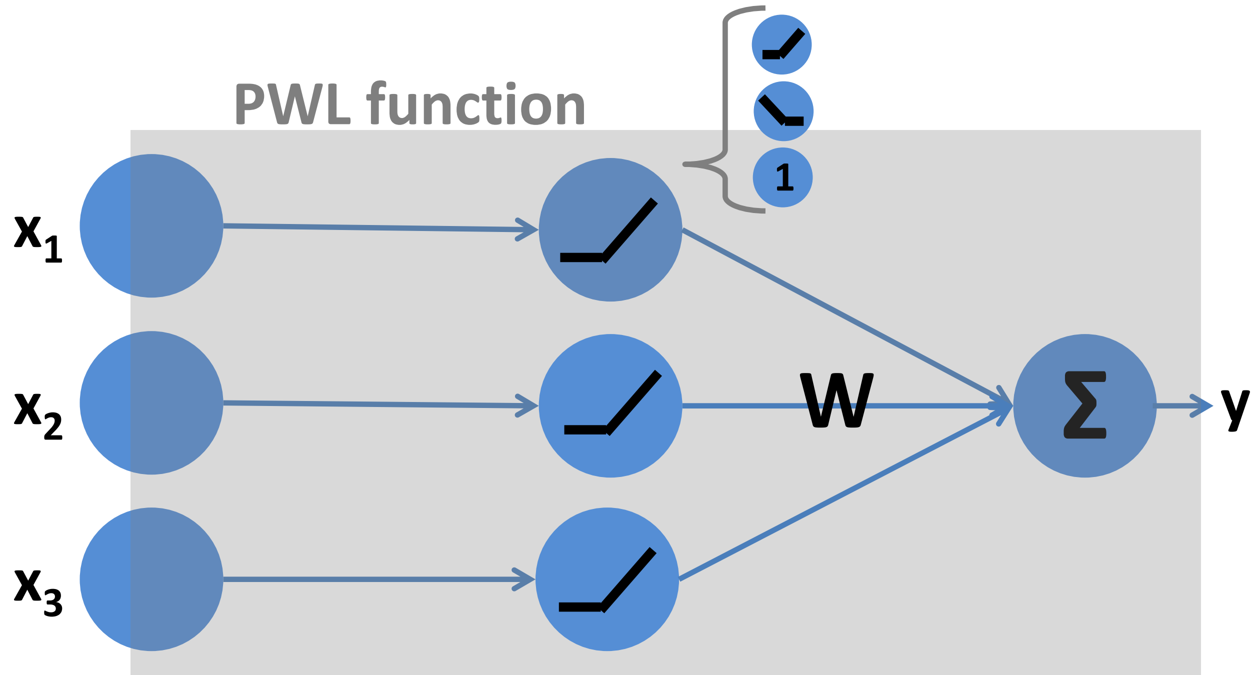


Adaptively zooms in on the most important regions.

Basis functions zero out the rest.

MARS with 1 layer





Adaptively zooms in on the most important regions.

Basis functions zero out the rest.

MARS thinking gives us intuition into PWL modeling

MARS Scalability

variables

training points

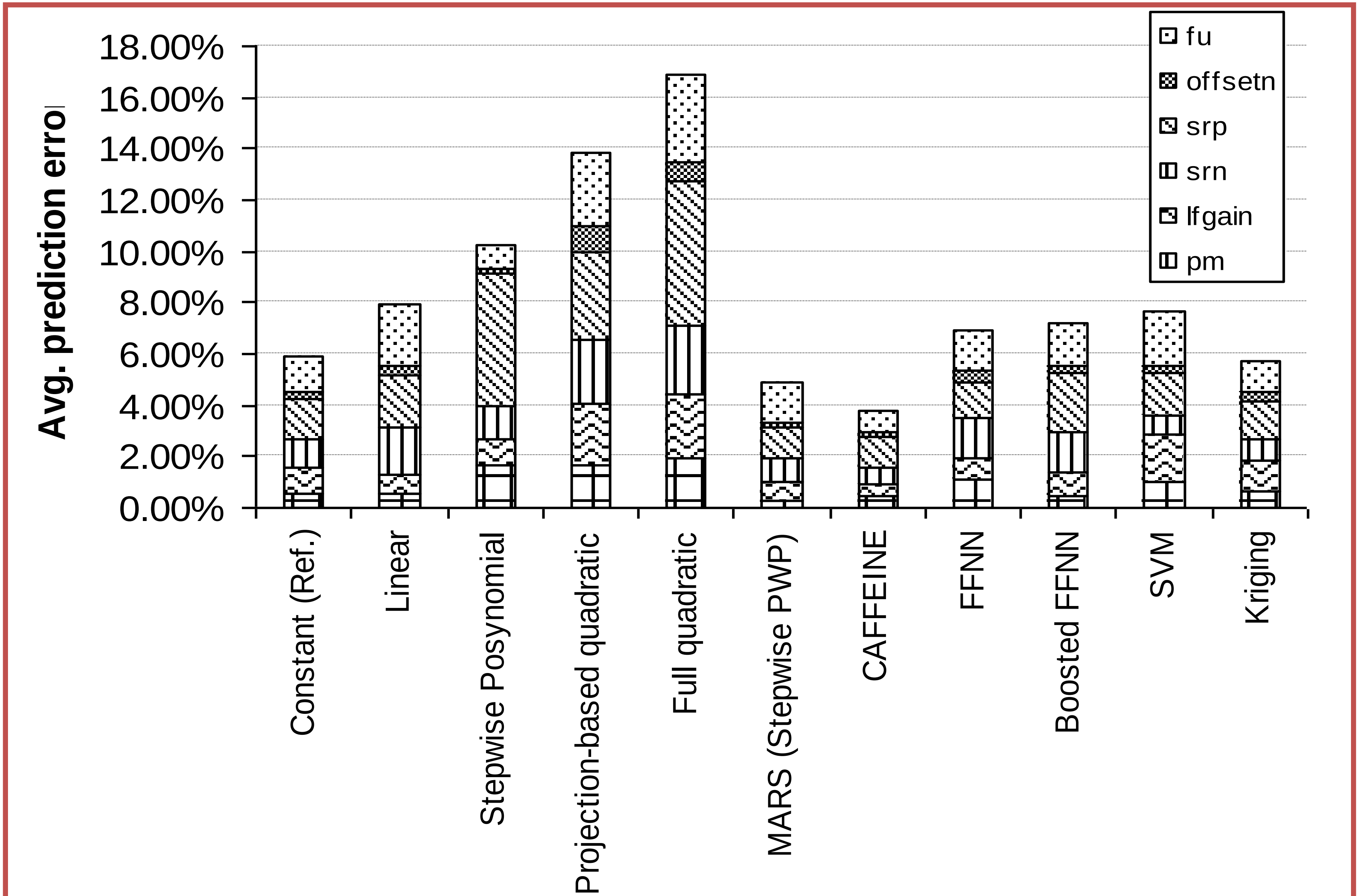
max # basis functions

$$C^* \sim nNM_{\max}^3(\alpha + \beta M_{\max}/L)$$

constants

Punchline: MARS has linear scaling in # vars and # points

MARS is Still Competitive



Aside: People Bag MARS too

Example: % error reduction by bagging

training set sizes	64		128		256		512		1024	
	r	p	r	p	r	p	r	p	r	p
kin-32fh	36.4	1	27.2	2	8.9	16	3.7	13	-3.8	22
kin-32fm	26.3	4	24.1	2	10.8	30	-8.1	28	-5.7	47
kin-32nh	40.1	1	28.7	1	23.6	1	12.3	1	6.8	1
kin-32nm	25.8	1	28.8	1	21.7	1	13.0	1	6.1	1
kin-8fh	37.3	1	16.0	4	16.8	1	13.2	2	3.8	4
kin-8fm	37.0	7	17.0	5	13.6	20	13.0	2	9.8	7
kin-8nh	20.7	5	17.6	1	11.6	1	8.3	1	9.4	4
kin-8nm	31.3	2	19.2	1	16.0	1	11.5	1	7.7	1
pumadyn-32fh	21.3	2	32.7	13	11.4	1	9.0	1	2.6	4
pumadyn-32fm	4.3	52	24.2	2	22.4	1	3.8	1	3.8	3
pumadyn-32nh	25.0	21	22.2	24	9.3	2	10.4	1	3.3	17
pumadyn-32nm	28.3	43	-23.9	44	22.5	1	16.8	1	5.3	7
pumadyn-8fh	12.9	11	3.6	59	4.5	16	5.5	1	3.6	7
pumadyn-8fm	20.2	2	15.1	2	8.4	3	4.4	2	5.9	4
pumadyn-8nh	12.1	41	19.0	2	10.5	1	10.1	1	4.0	7
pumadyn-8nm	22.7	8	30.3	1	17.0	3	13.0	3	7.5	25
averages	25.1		18.9		14.3		8.7		4.3	

Rasmussen, EVALUATION OF GAUSSIAN PROCESSES AND OTHER METHODS FOR NON-LINEAR REGRESSION, PhD thesis, 1996, Fig. 5.2, p. 74

Outline

- Carts
- Mars
- <u>Nets</u>

Deep Nets

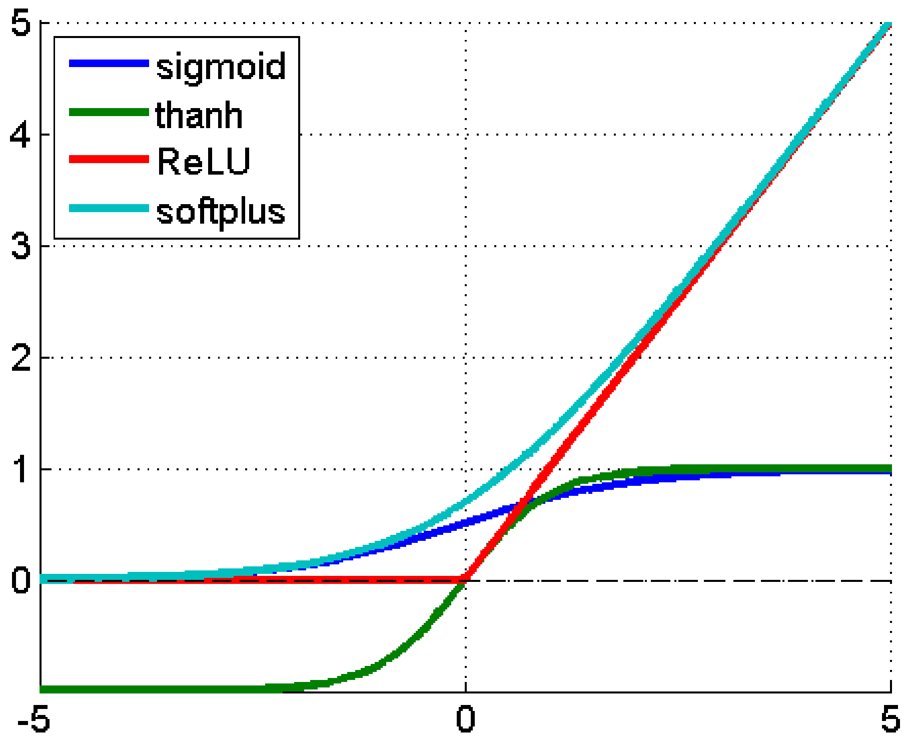
1. Intuition is difficult
2. Shortage of theory
3. Long training times

- But they work really well on many problems!
- So people use them
- My goal here: help 1, maybe 2, maybe 3

https://imiloainf.files.wordpress.com/2013/11/activation_funcs1.png

Deep Nets

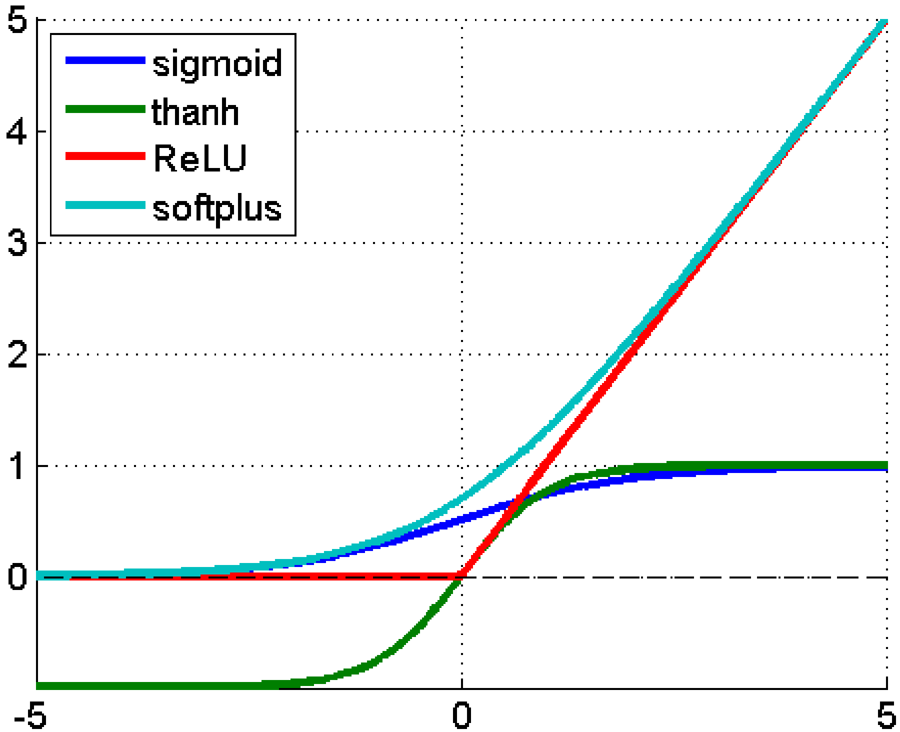
Rectified Linear Units (ReLUs)

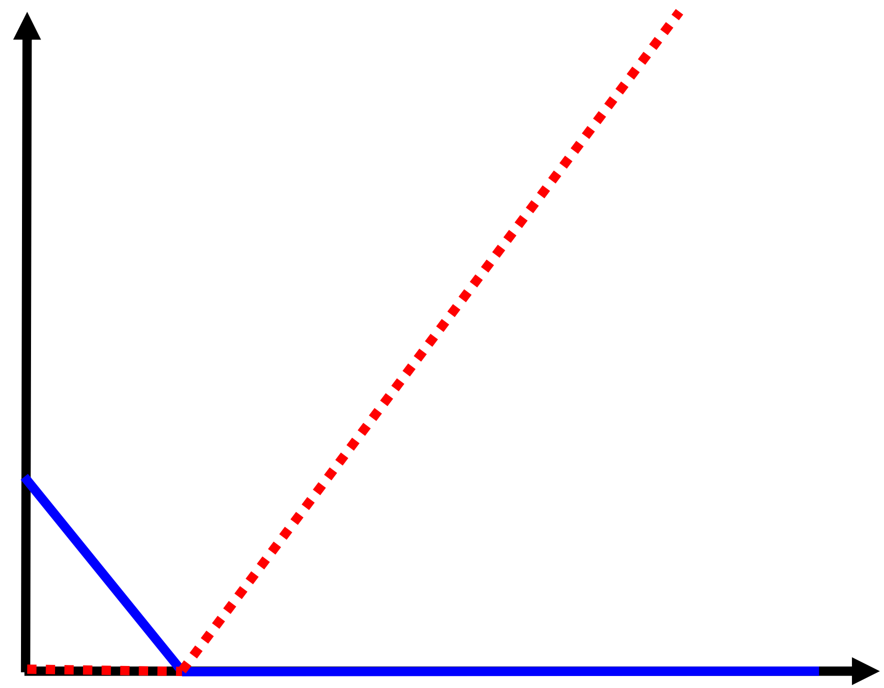


V. Nair and G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, ICML, 2010
A. Krizhevsky, I. Sutskever, and G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS, 2012
https://imiloainf.files.wordpress.com/2013/11/activation_funcs1.png

Deep Nets

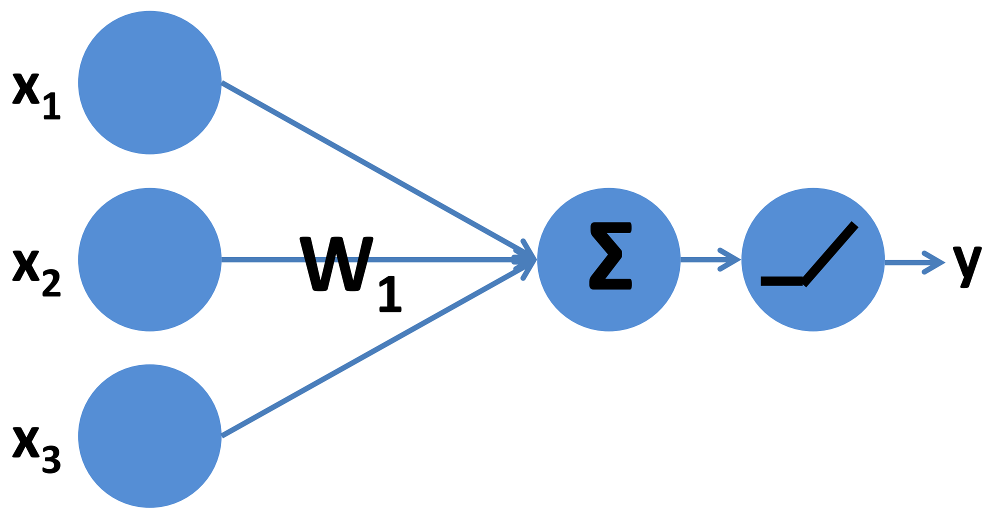
Rectified Linear Units (ReLUs) ≈MARS Hinge Functions



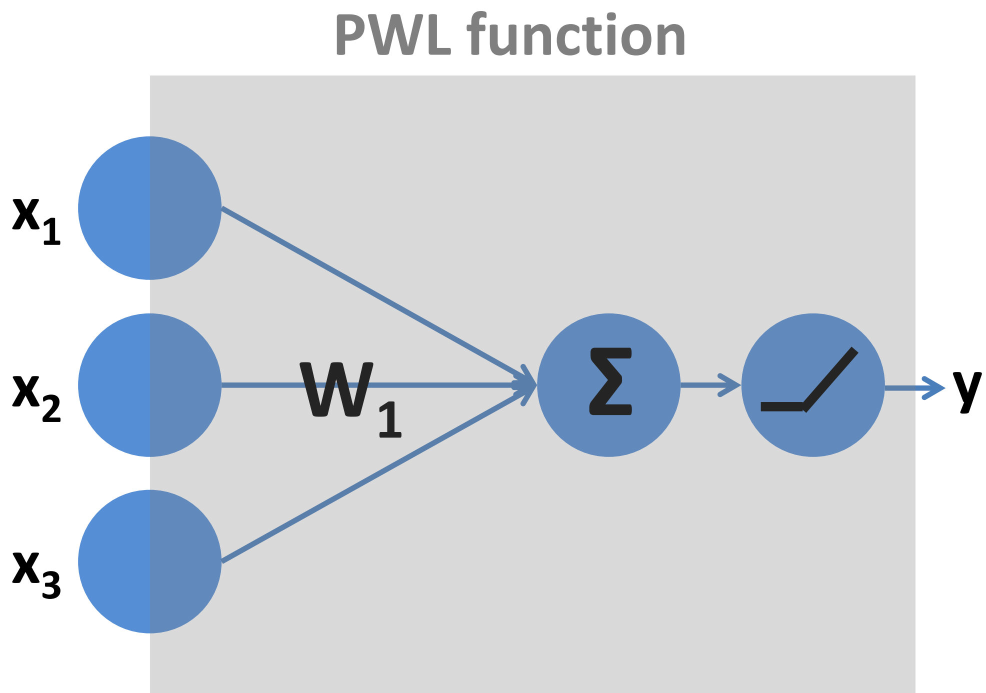


Benefit: we can translate intuition from MARS into Deep Nets. E.g. adaptively zooming in on most important regions, zero out rest

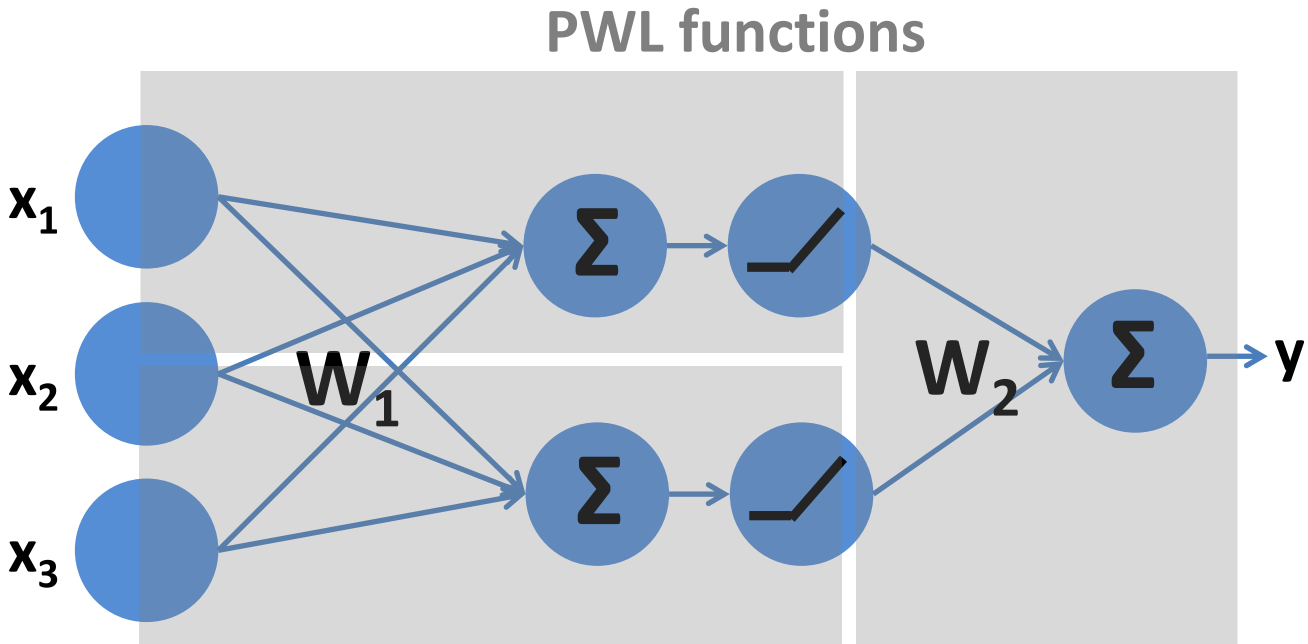
NN with 1 layer, 1 hidden node



NN with 1 layer, 1 hidden node



NN with 1 layer, 2 hidden nodes



Deep NN with 5 layers, 6 nodes per layer

Deep NN with 5 layers, 6 nodes per layer

Insight from MARS based PWL models can apply here.

Food for thought: what other MARS tricks can we port to Deep NNs?

Eg cascade MARS & dropout-regularize for “Deep MARS”?

Conclusion

- CARTs are old but good: fast, accurate, understood (accurate with RFs)
- MARS is old but good: fast, accurate, understood
- Deep NNs are new(ish). Very accurate, but slow & poorly understood
- Insights from CARTs led to MARS
- Insights from MARS can help Deep NNs

Appendix: MARS Software

Free [edit]

- Several R packages fit MARS-type models:
  - `earth` function in the `earth` package
  - `mars` function in the `mda` package
  - `polymars` function in the `polspline` package. Not Friedman's MARS.
- Matlab code:
  - ARESLab: Adaptive Regression Splines toolbox for Matlab
- Python
  - Earth - Multivariate adaptive regression splines
  - py-earth

Commercial [edit]

- MARS from Salford Systems. Based on Friedman's implementation.
- STATISTICA Data Miner from StatSoft
- ADAPTIVEREG from SAS.

http://en.wikipedia.org/wiki/Multivariate_adaptive_regression_splines