Inspired by research. Driven by compassion. Inspiré par la recherche. Guidé par la compassion.

From linear and logistic regression to neural networks

ICES uOttawa / OHRI lunch and learn

André Carrington, PhD, PEng, CISSP
Postdoctoral Fellow, AI
Doug Manuel lab, Clinical Epidemiology

November 6, 2019 (rev 1.0)
November 20, 2019 (rev 1.2)

The Ottawa Hospital | L'Hôpital d'Ottawa
RESEARCH INSTITUTE | INSTITUT DE RECHERCHE

We denote one patients features by $\underline{x}=[x_1\ x_2\ \ldots]^T$ and the targets for all patients by $\underline{y}=[y_1\ y_2\ \ldots]^T$

	features: $\underline{x}$							target	
	wgt	height	pulse	age	sex	acr	gfr	ckd	
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\underline{y}$	
patient 1	170	68	80	65	0	30	60	1	y_1
patient 2	150	65	60	46	1	5	30	1	y_2
patient 3	155	66	65	22	0	2	95	-1	y_3
patient 4	160	68	60	37	1	2	100	-1	y_4

A chronic kidney disease (ckd) example.

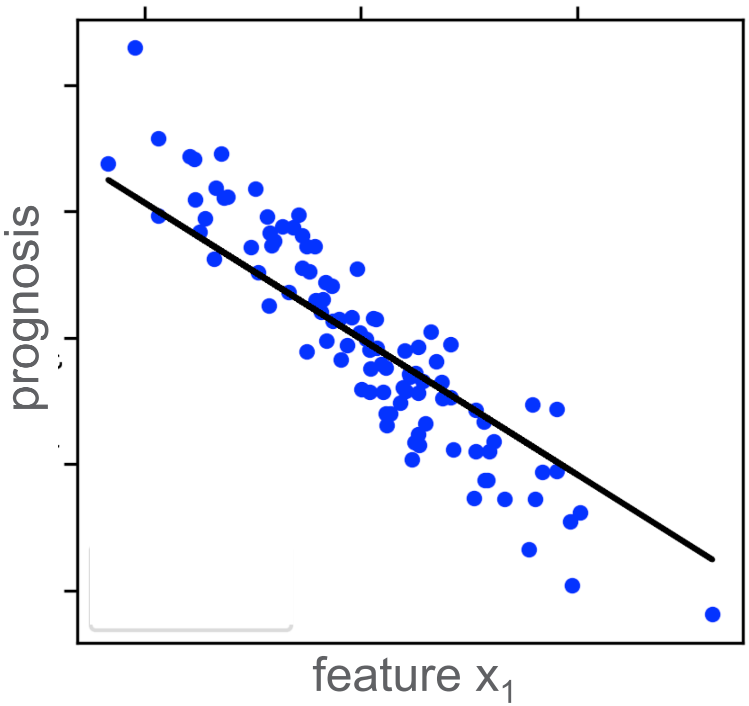
Linear regression (multiple linear regression)

$\underline{\boldsymbol{x}}_{\text{Joe}}$ Joe's features

y_{Joe} Joe's prognosis

$$\hat{y}_{\text{Joe}} = \underline{\boldsymbol{\beta}}_{\text{TOH}}{}^{\text{T}} \underline{\boldsymbol{x}}_{\text{Joe}}$$

Joe's estimated prognosis



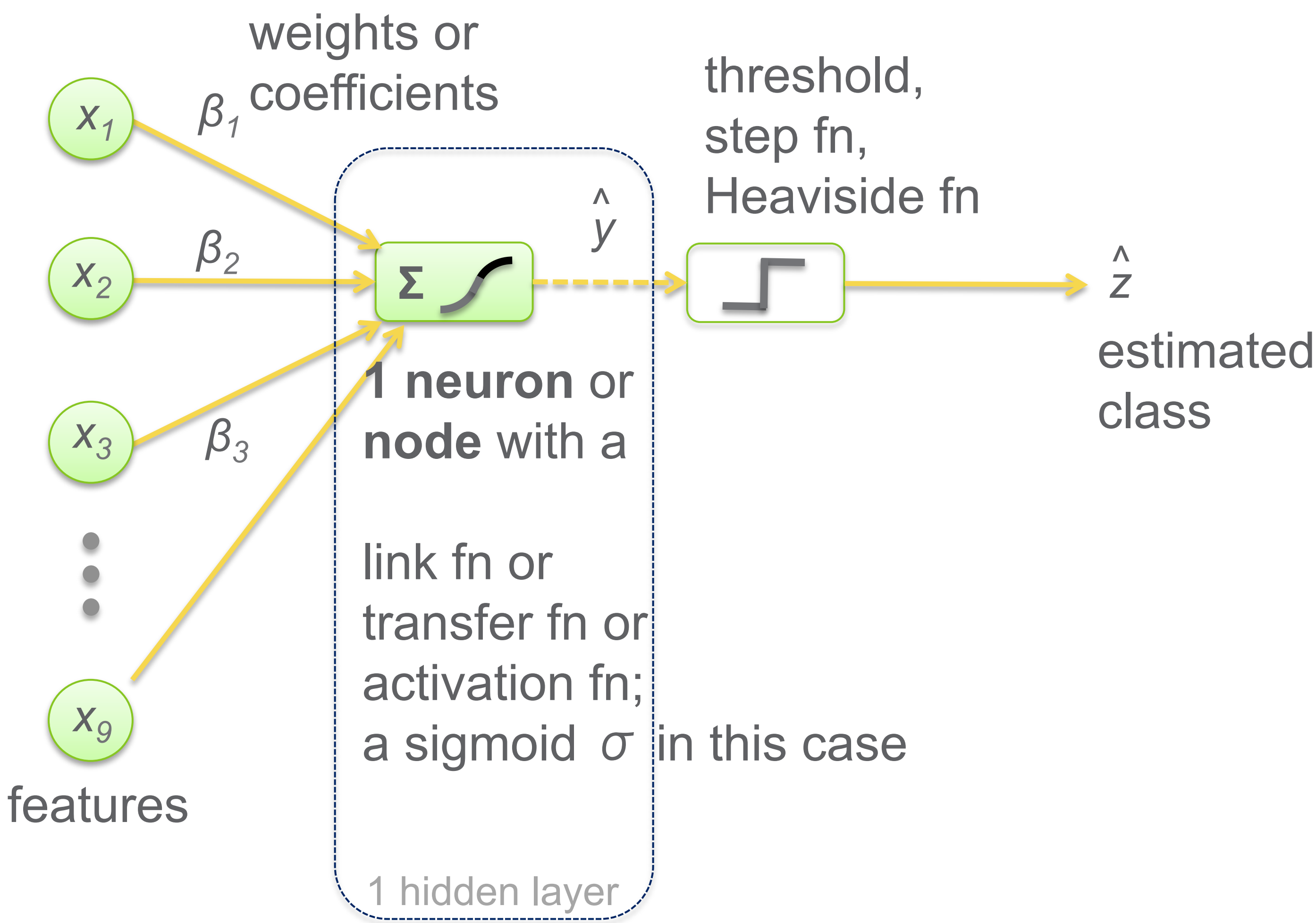
Logistic regression (a generalized linear model)

- From $\hat{y} = \underline{\boldsymbol{\beta}}^{\mathrm{T}} \underline{\boldsymbol{x}}$ to $\hat{y} = \sigma(\underline{\boldsymbol{\beta}}_{\mathrm{TOH}}{}^{\mathrm{T}} \underline{\boldsymbol{x}}_{\mathrm{Joe}})$
- A non-linear link function σ is applied to the linear model, which allows for a non-linear fit.
- $\sigma(\boldsymbol{\beta}^T \boldsymbol{x}) = \dfrac{e^{\boldsymbol{\beta}^T \boldsymbol{x}}}{1 - e^{\boldsymbol{\beta}^T \boldsymbol{x}}}$

Neural networks

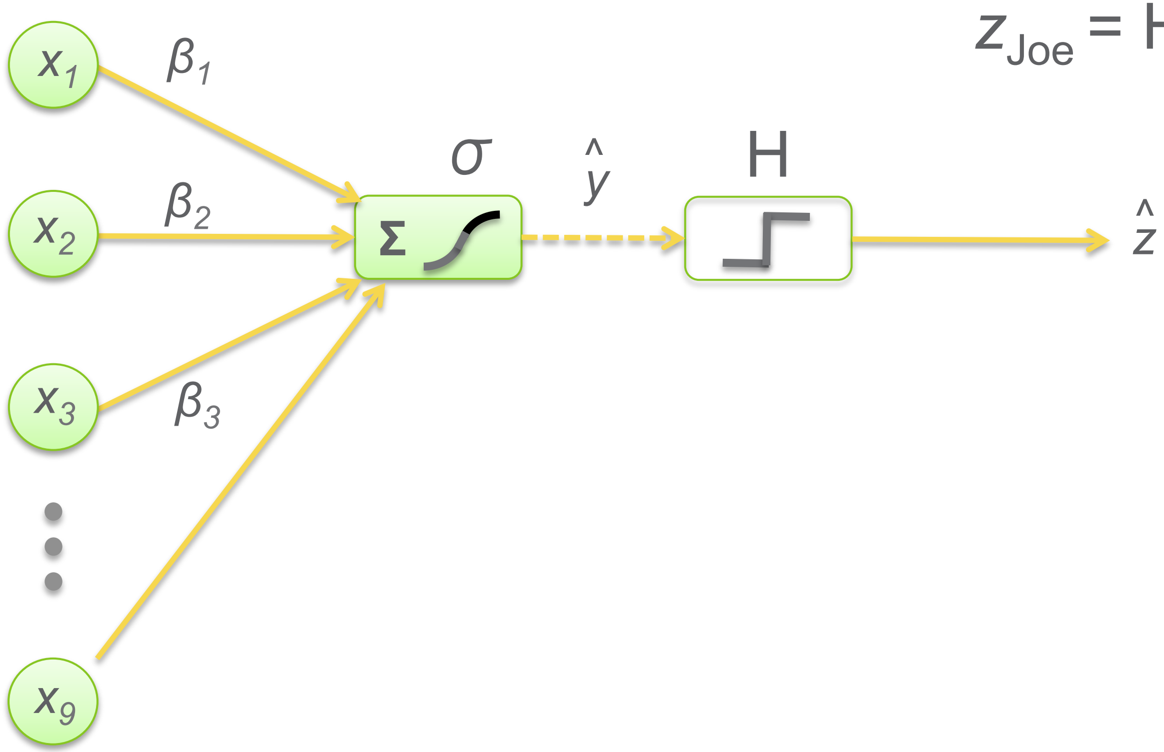
- You already know one!
- Logistic regression is a neural network with:
    - one hidden layer & one neuron!

Logistic regression in network form



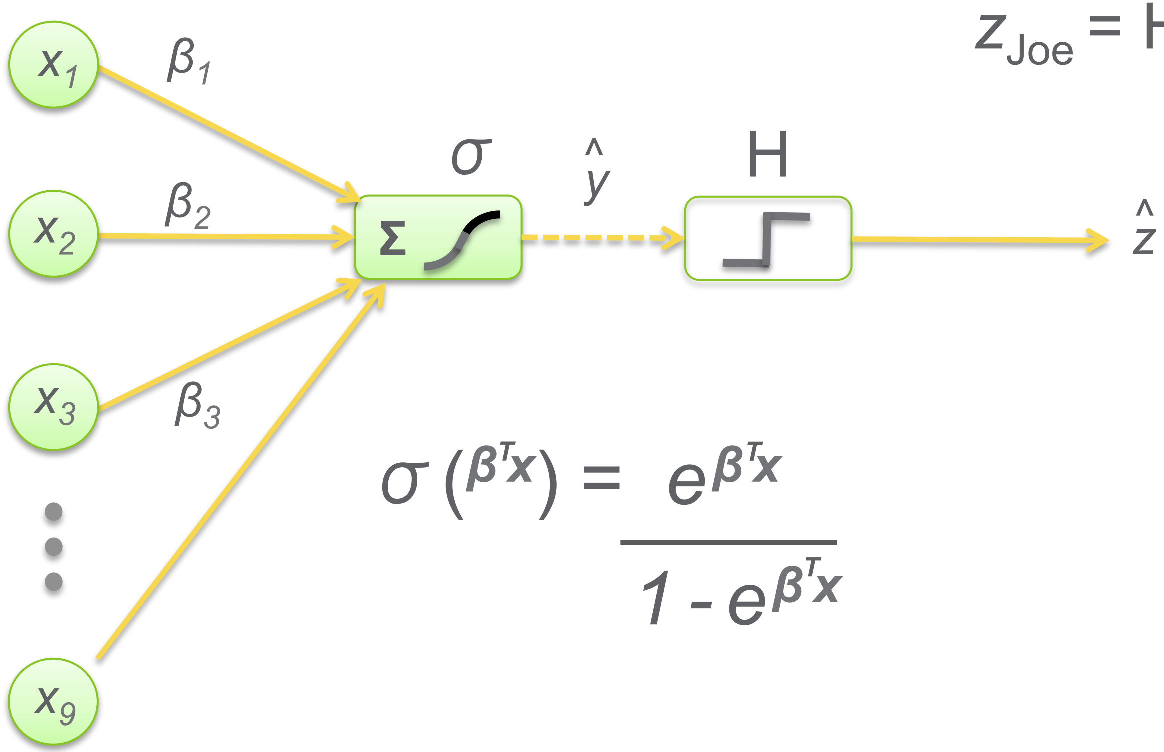
Logistic regression in network form

$$\hat{z}_{Joe} = H(\sigma(\underline{\boldsymbol{\beta}}_{TOH}{}^{T}\underline{\mathbf{x}}_{Joe}))$$



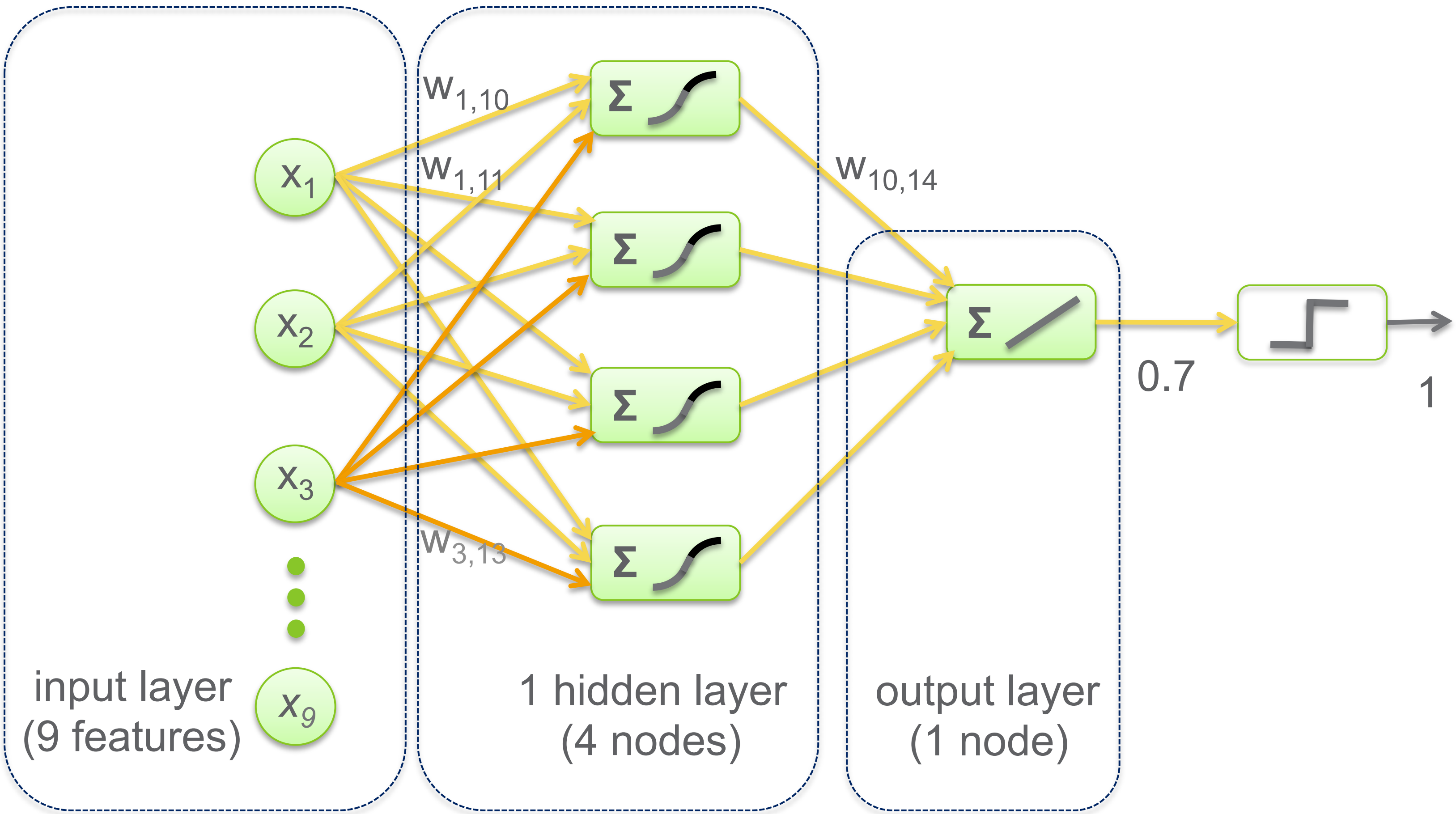
Logistic regression in network form

$$\hat{z}_{Joe} = H(\sigma(\underline{\boldsymbol{\beta}}_{TOH}{}^{T}\underline{\mathbf{x}}_{Joe}))$$



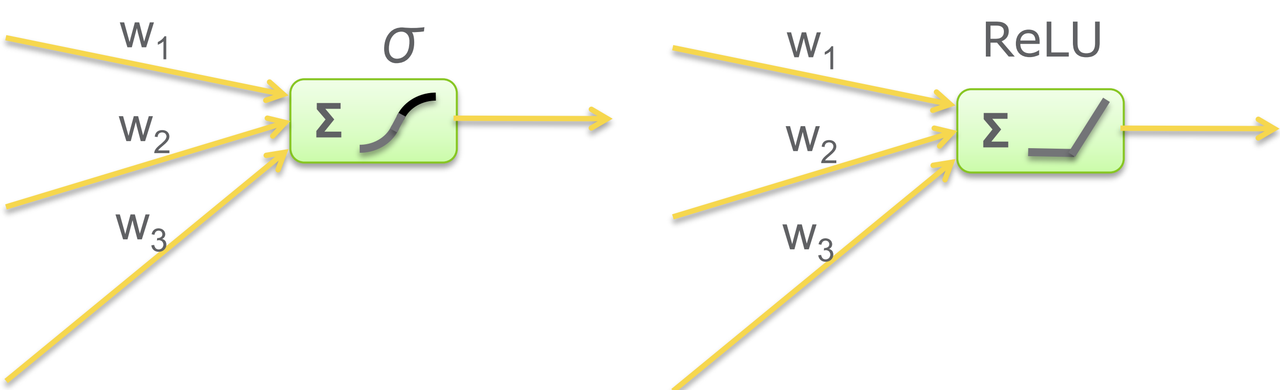
$$\sigma(\boldsymbol{\beta}^T\mathbf{x}) = \frac{e^{\boldsymbol{\beta}^T\mathbf{x}}}{1 - e^{\boldsymbol{\beta}^T\mathbf{x}}}$$

A standard neural network example



Typical transfer/activation functions in neurons

- Activation (output high) occurs when the inputs are sufficiently high collectively. Activation functions are:
  - Sigmoids σ, i.e., S-curves
  - Rectified linear units (ReLU), i.e. filter out negatives
  - Step functions



Universal approximators

- Learning theory says the following can approximate any function given ***sufficient*** (infinite?) data
  - a neural network with a single hidden layer and a ***sufficient*** number of nodes
  - a support vector machines with a ***sufficient*** number of support vectors
    - a polynomial kernel
    - a sigmoid kernel
    - a Gaussian (radial basis function, RBF) kernel

But that theory does not help practice

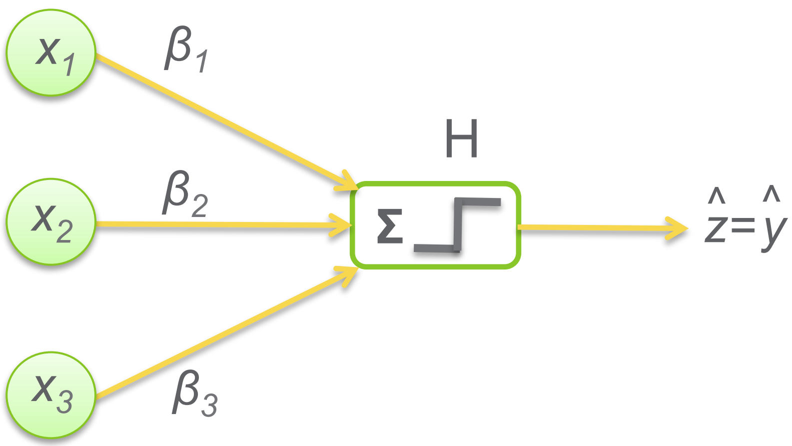
- We have finite data, finite computing power/time, and we want finite understandable models.
- In the theoretical case of infinite data there is no "generalization" necessary from training to testing, they are the same.
- Theory tells us nothing about training fit with finite data, nor how well a model generalizes to new data

In conclusion

- We have seen a neural network for binary classification, as a natural extension of logistic regression.
- We introduced types of activation (transfer) functions.
- We introduced the theory and limits of universal approximators.

Extras: a perceptron in network form

- A single node using a step function instead of a sigmoid
- A classifier that finds a line of separation given a separable data set. Fails for non-separable data



- Neural networks are called multilayer perceptrons (MLP)

A support vector machine in network form

- The network form for SVM differs from neural networks.

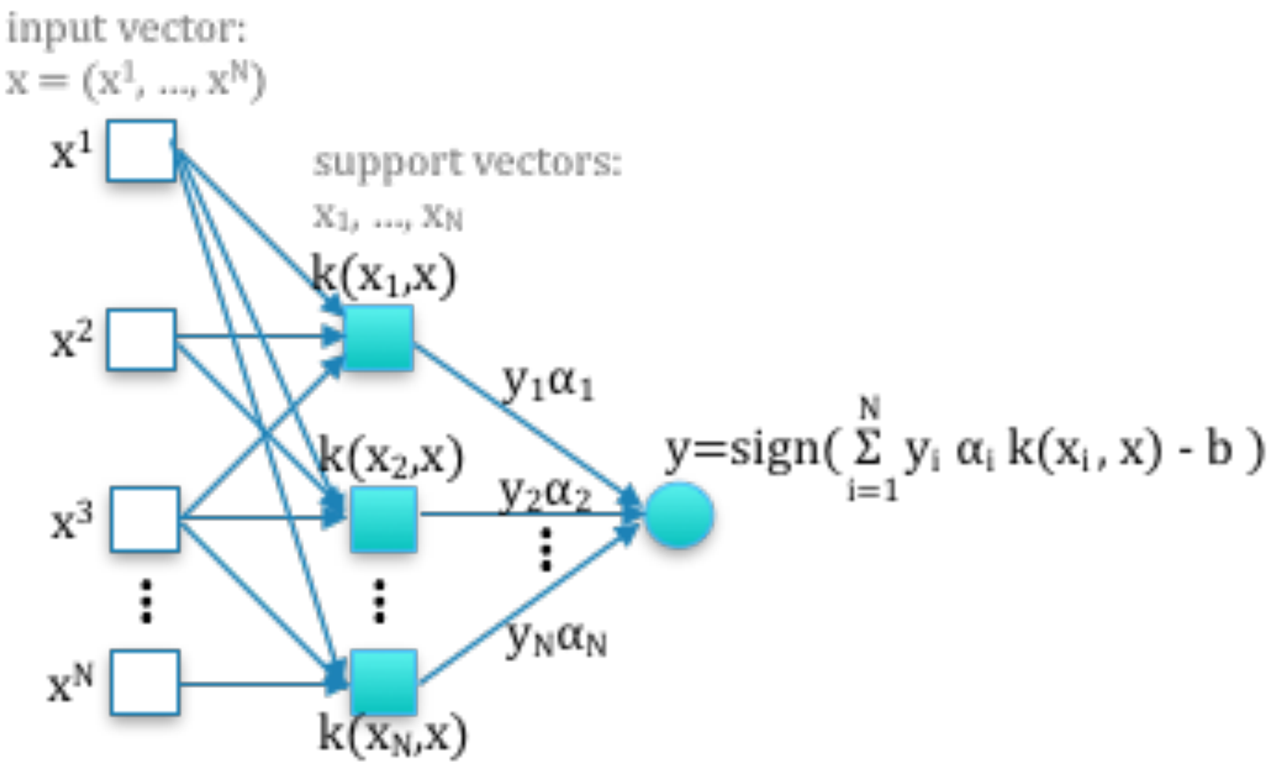


Figure adapted from Vapnik.

Future topics

- Back-propagation
- Multiclass classification with the softmax transfer function.
- Sequences or time series with recurrent neural networks (RNNs) and long-short term memory (LSTM) transfer functions.
- Deep learning with convolutional neural networks (CNNs).

Questions?

André Carrington, PhD PEng CISSP
acarrington at ohri.ca