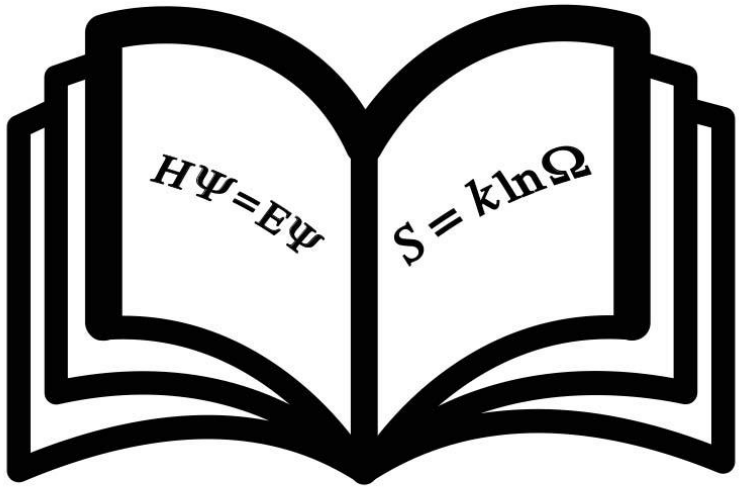
iCOMSE: Machine Learning in Molecular Science



Professor Camille Bilodeau

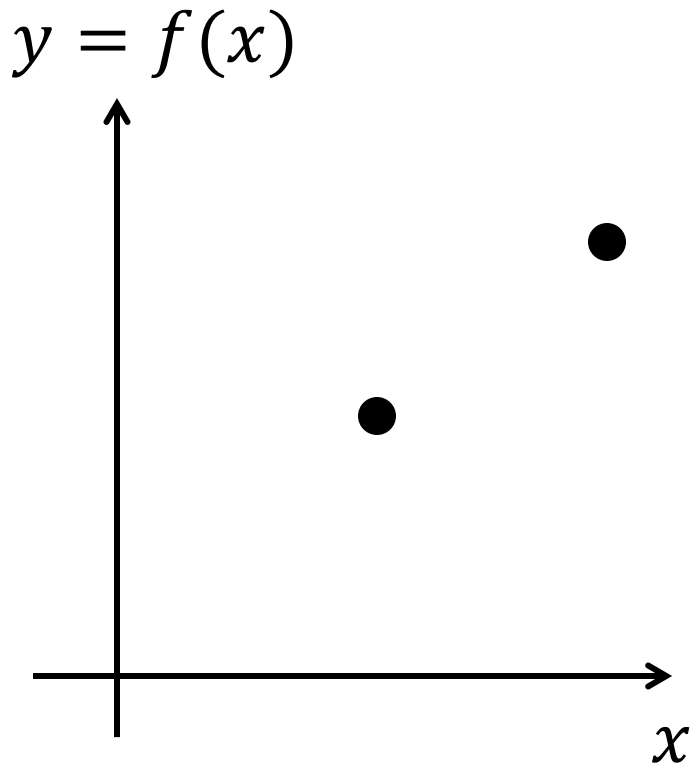
University of Virginia

April 30th 2025

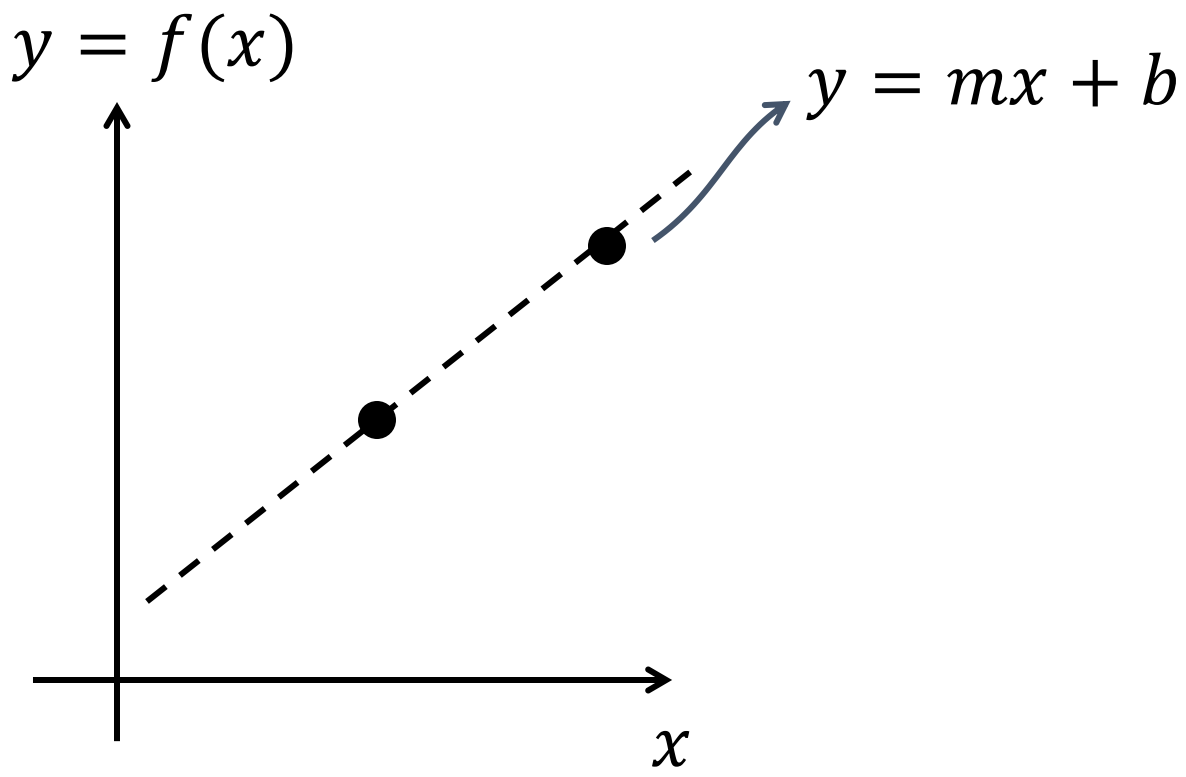
Deep Learning for Molecules in Pytorch

1. Math behind Fully Connected Neural Networks (FCNNs)
2. Implementation of an FCNN in Pytorch
3. Math behind Graph Convolutional Neural Networks (GCNNs)
4. Implementation of a GCNN in Pytorch

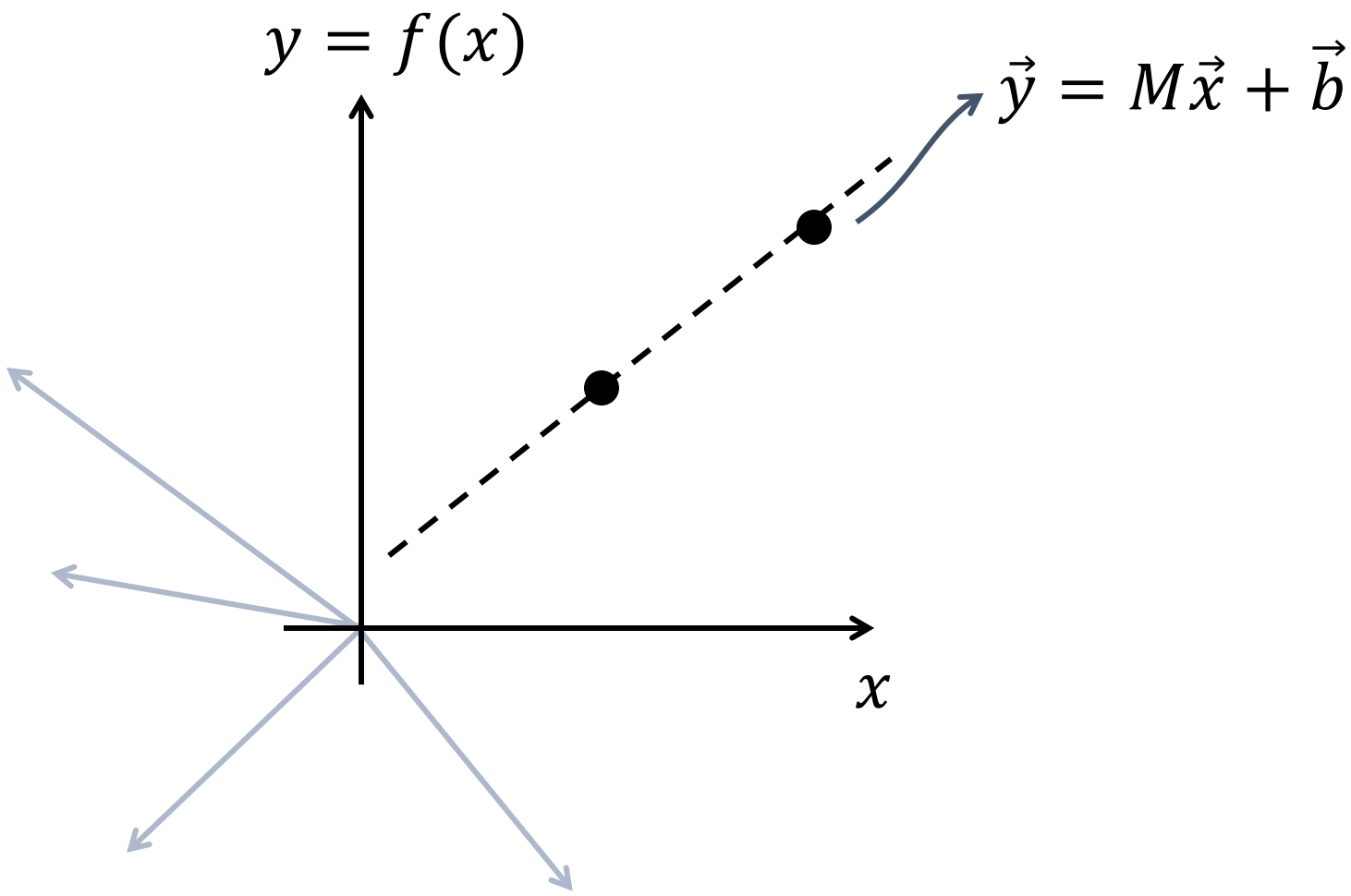
Supervised Deep Learning: Let's try to parameterize a function!



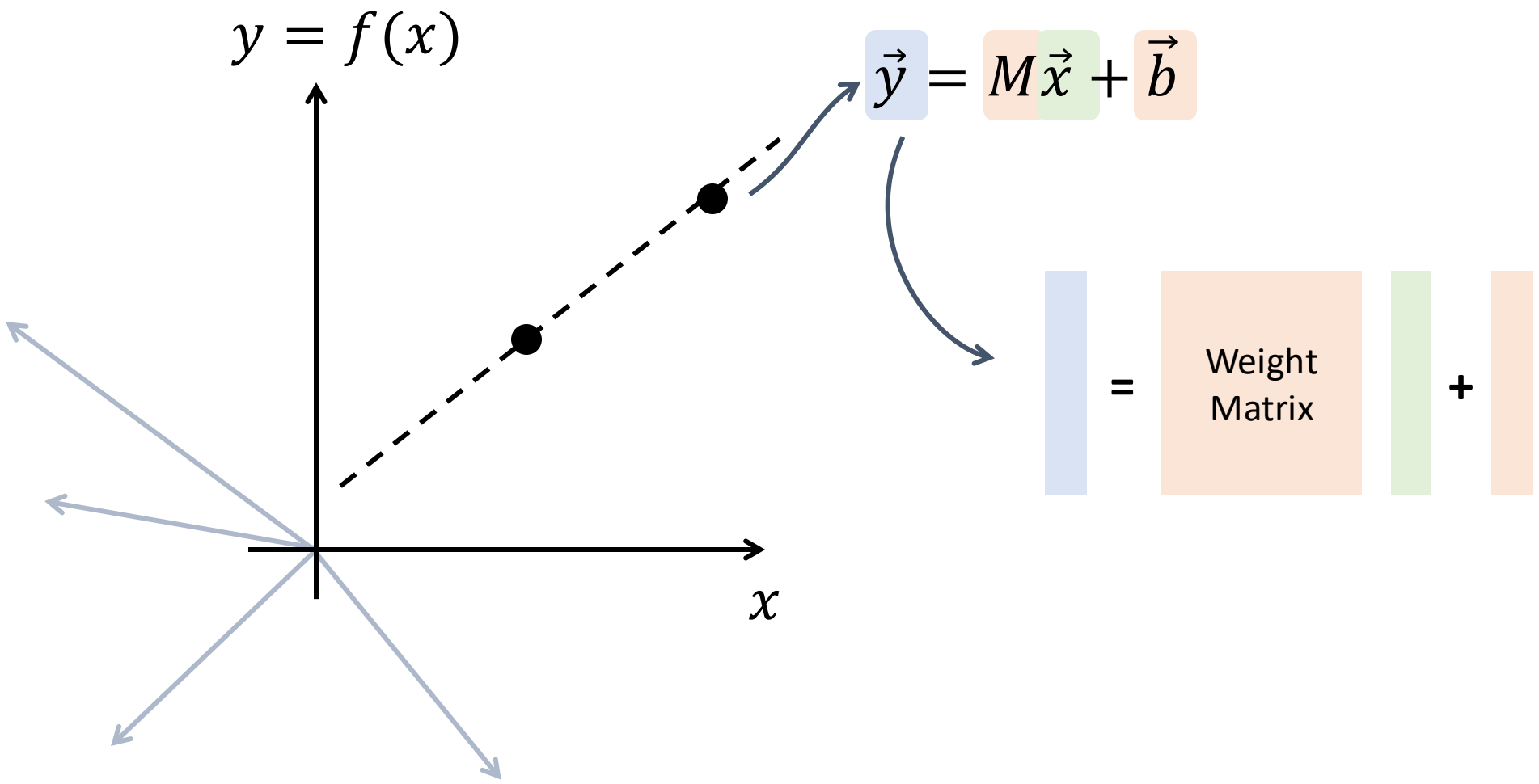
Supervised Deep Learning: Let's try to parameterize a function!



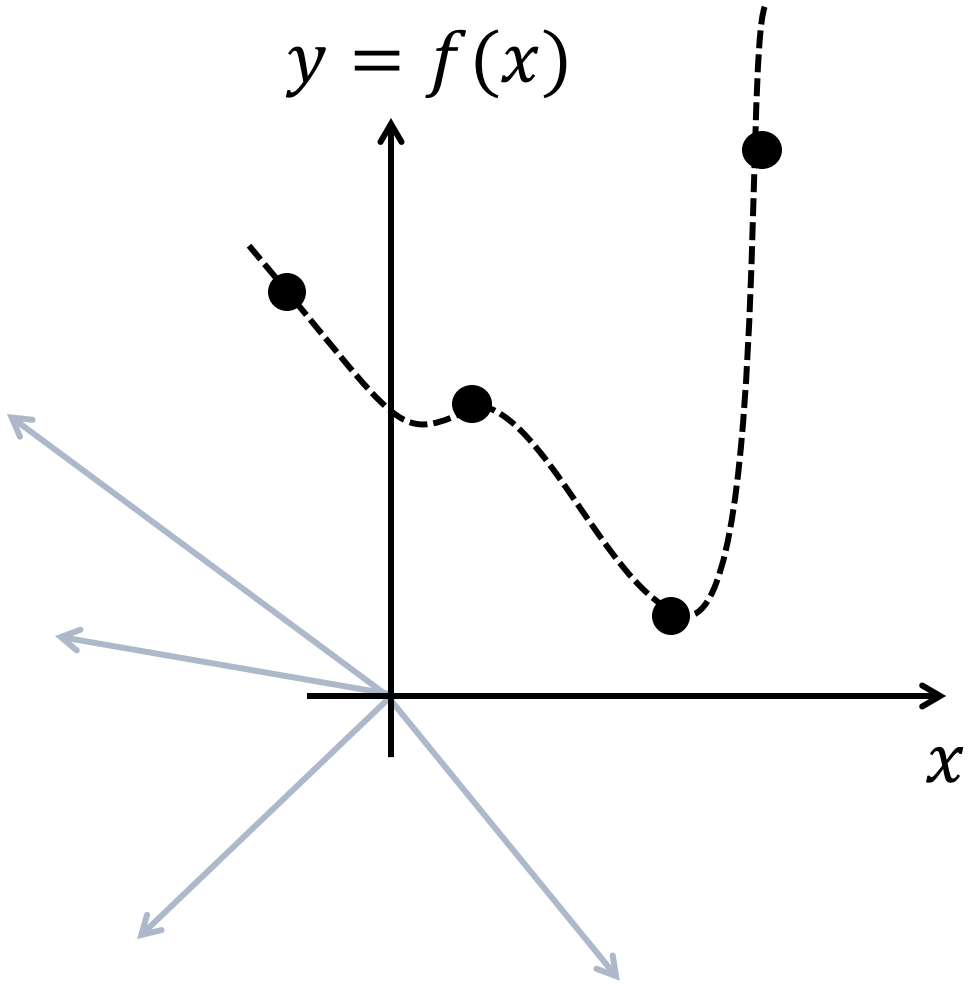
Supervised Deep Learning: Let's try to parameterize a function!



Supervised Deep Learning: Let's try to parameterize a function!

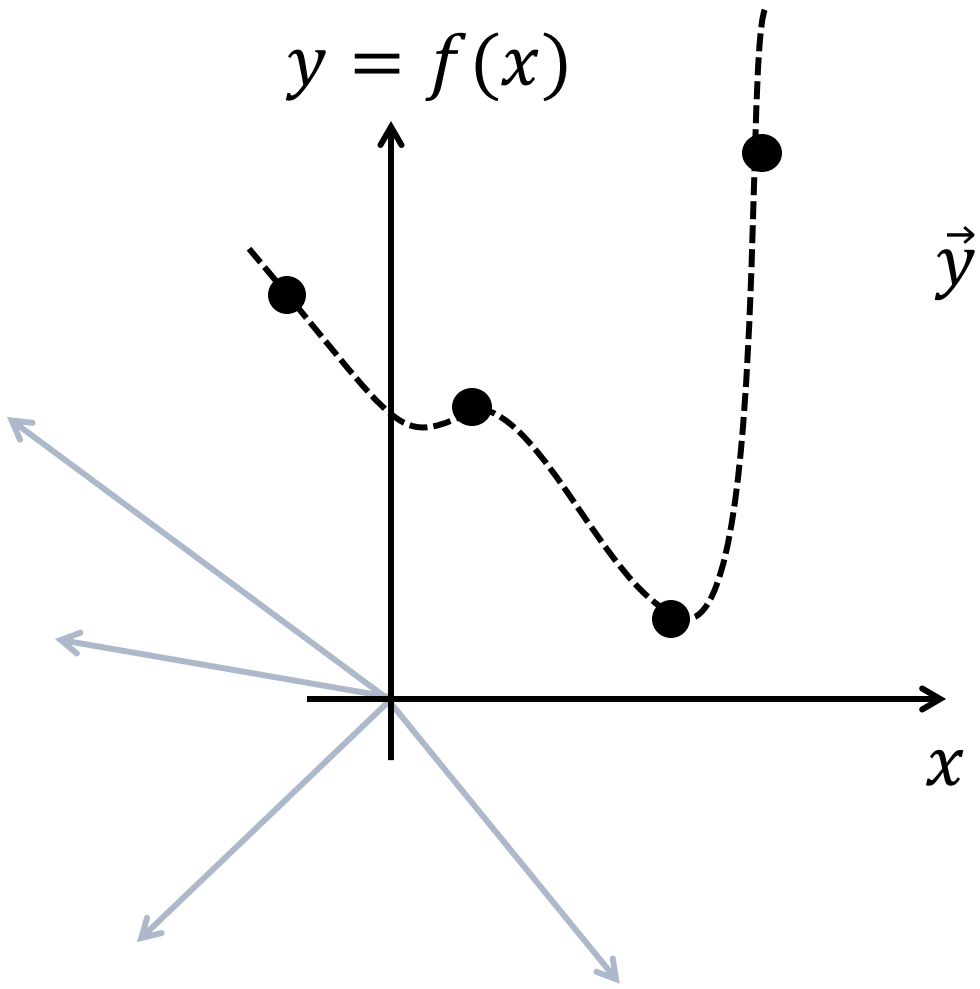


What if we want to parameterize a more complex function?



$$\vec{y} = f(\vec{x})$$

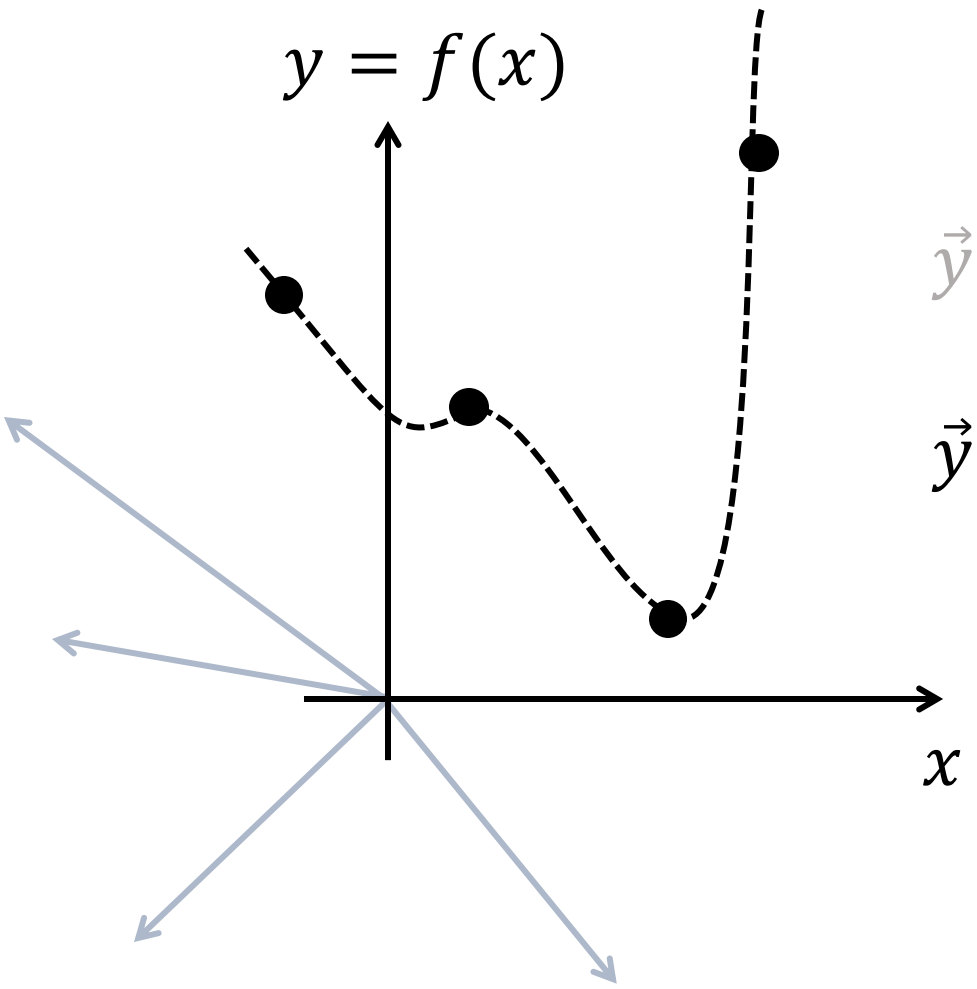
What if we want to parameterize a more complex function?



$\vec{y} = f(\vec{x})$

$\vec{y} = f_1(\vec{h})$ $\quad$ $\vec{h} = f_2(\vec{x})$

What if we want to parameterize a more complex function?

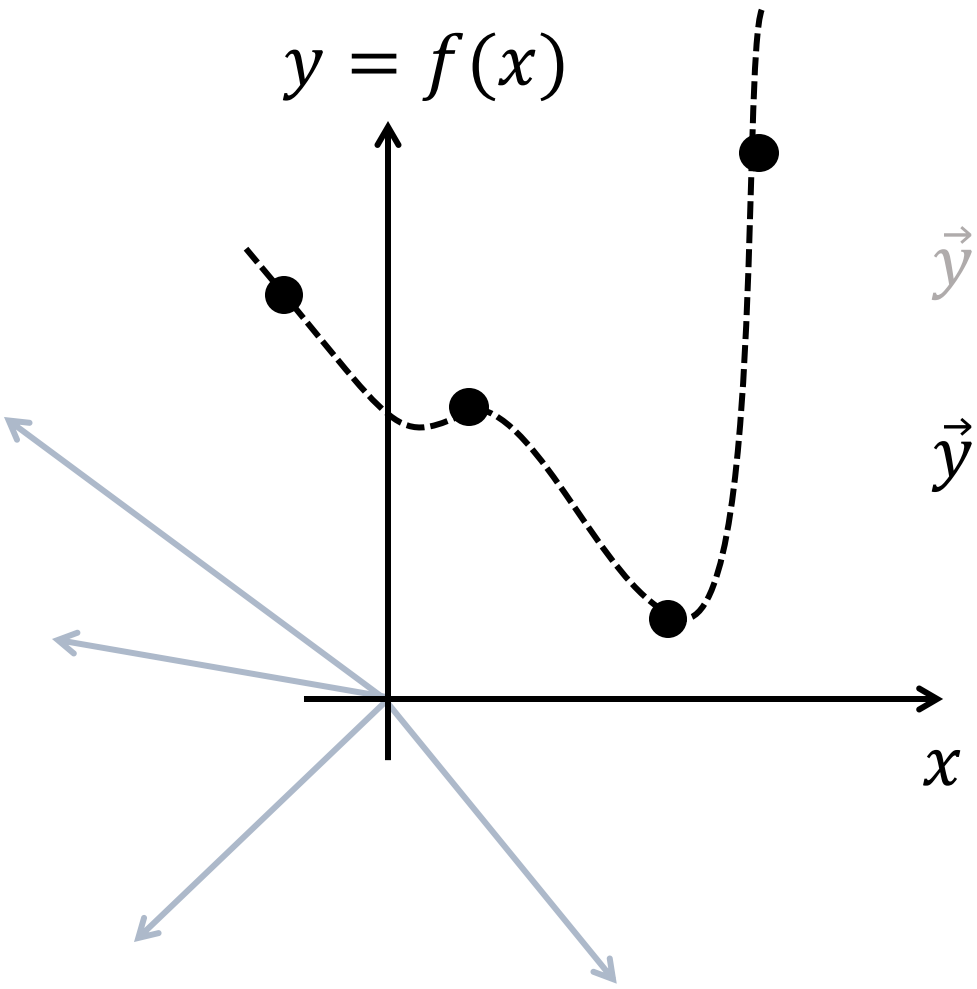


$$\vec{y} = f(\vec{x})$$

$$\vec{y} = f_1(\vec{h}) \qquad \vec{h} = f_2(\vec{x})$$

$$\vec{y} = (M_1\vec{h} + b_1) \qquad \vec{h} = (M_2\vec{x} + b_2)$$

What if we want to parameterize a more complex function?

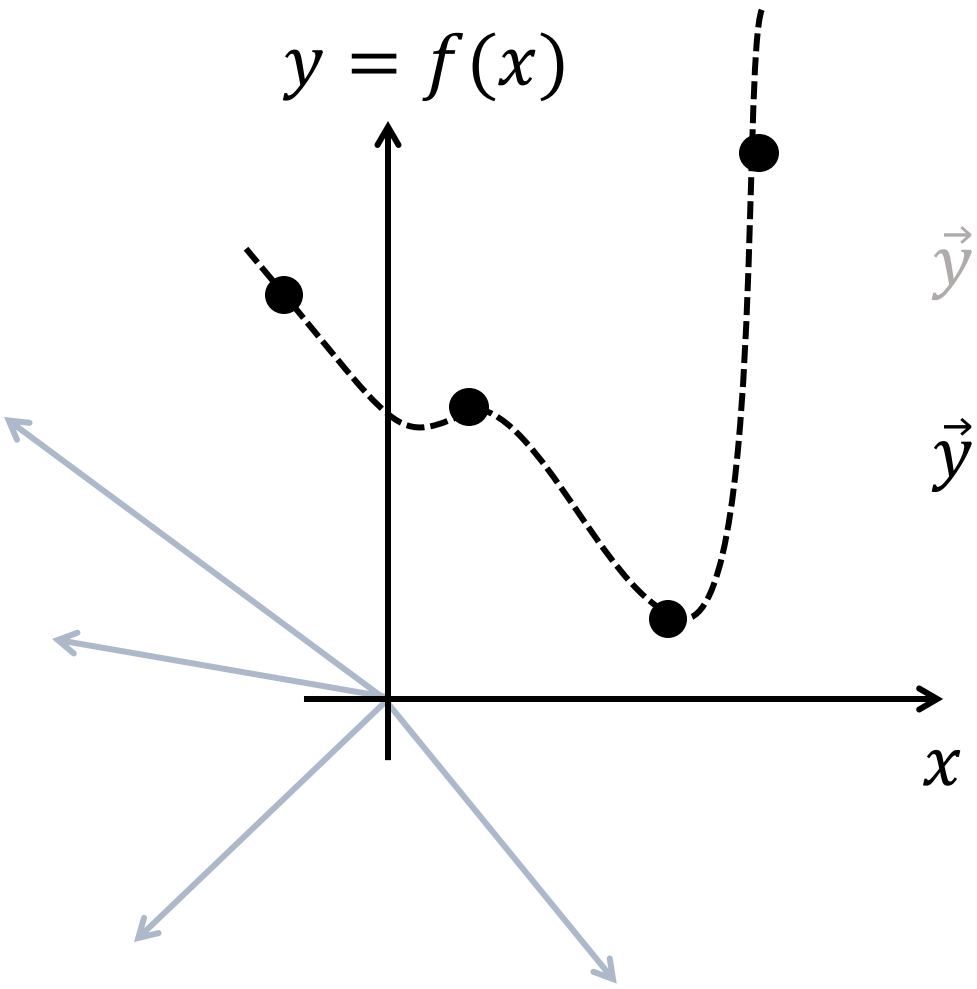


$$\vec{y} = f(\vec{x})$$

$$\vec{y} = f_1(\vec{h}) \qquad \vec{h} = f_2(\vec{x})$$

$$\vec{y} = \sigma(M_1\vec{h} + b_1) \qquad \vec{h} = \sigma(M_2\vec{x} + b_2)$$

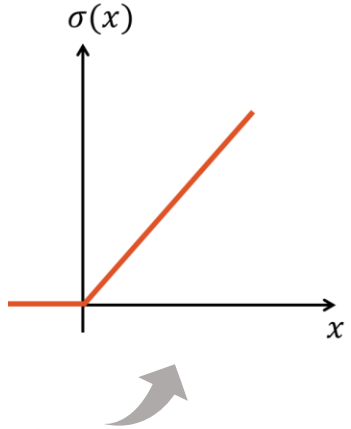
What if we want to parameterize a more complex function?



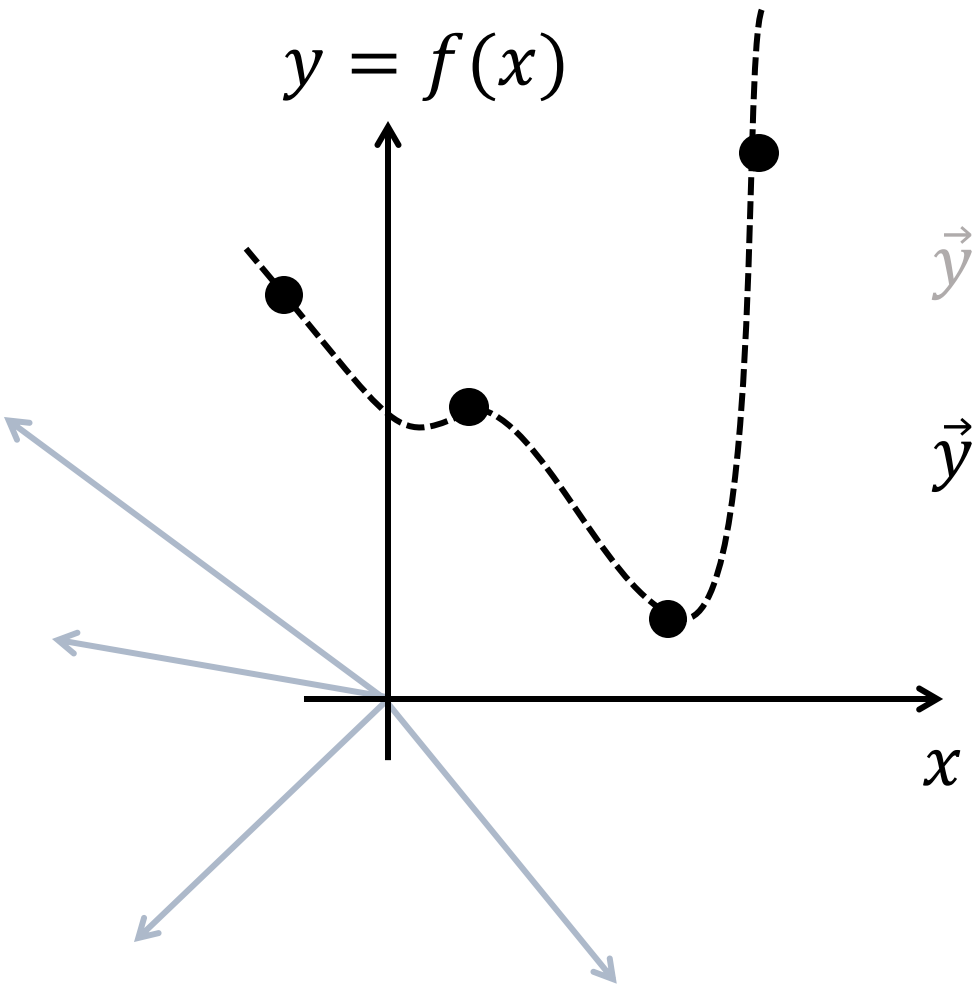
$$\vec{y} = f(\vec{x})$$

$$\vec{y} = f_1(\vec{h}) \qquad \vec{h} = f_2(\vec{x})$$

$$\vec{y} = \sigma(M_1\vec{h} + b_1) \qquad \vec{h} = \sigma(M_2\vec{x} + b_2)$$



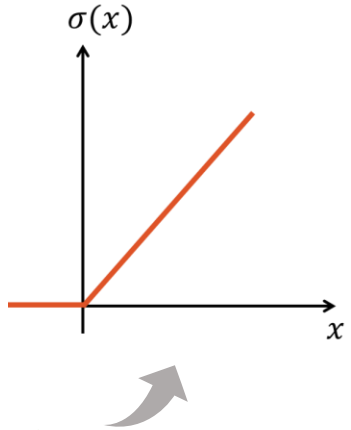
What if we want to parameterize a more complex function?

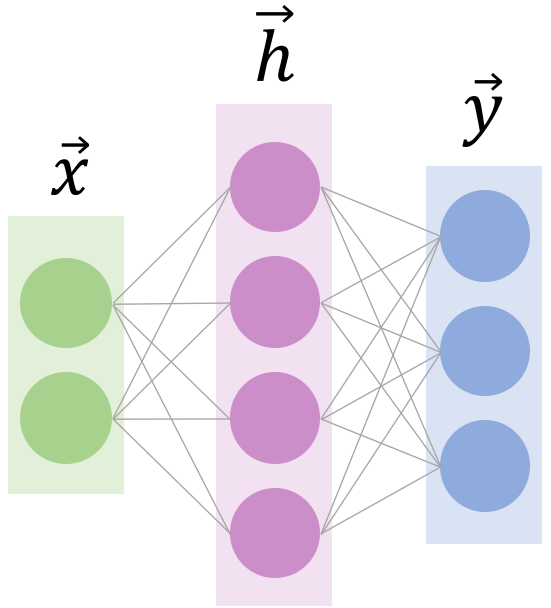


$$\vec{y} = f(\vec{x})$$

$$\vec{y} = f_1(\vec{h}) \qquad \vec{h} = f_2(\vec{x})$$

$$\vec{y} = \sigma(M_1\vec{h} + b_1) \qquad \vec{h} = \sigma(M_2\vec{x} + b_2)$$





How do we find the correct weights?

$$\vec{y} = \sigma(\overbrace{M_1\vec{h} + b_1}^{c_1}) \qquad \vec{h} = \sigma(\overbrace{M_2\vec{x} + b_2}^{c_2})$$

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

1) Define a Loss Function

$$\min(L(M_1, M_2, b_1, b_2))$$

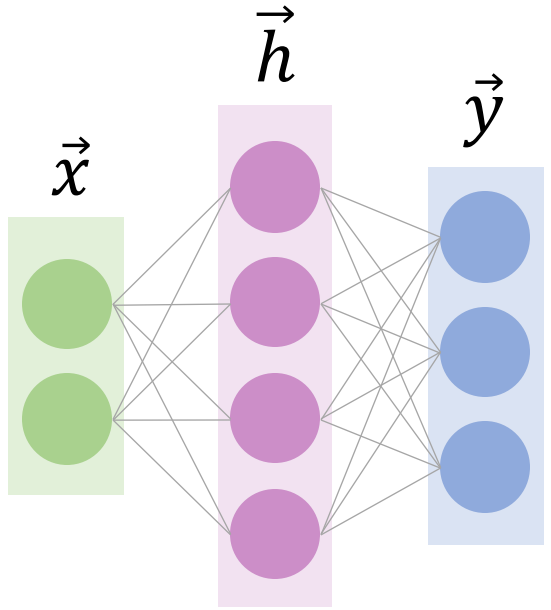
2) Define an Objective Function

$$M_{1,j+1} = M_{1,j} - \gamma \underbrace{\frac{\partial L}{\partial M_1}}$$

3) Apply Gradient Descent (or Similar)

$$\frac{\partial L}{\partial M_1} = \frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial c_1}\frac{\partial c_1}{\partial M_1}$$

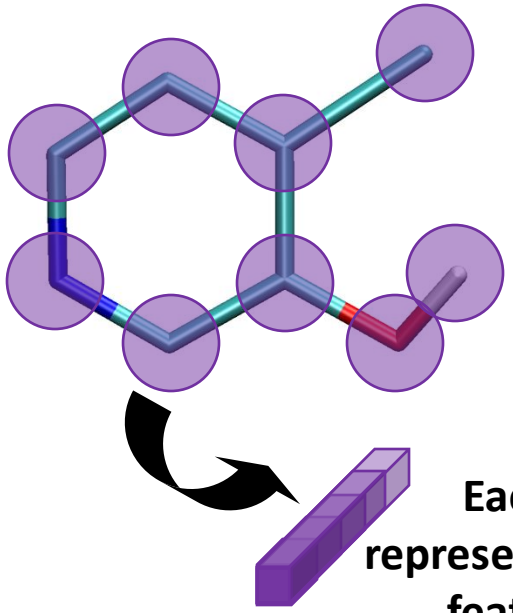
Gradient can be obtained using chain rule, this is the core principle behind **backpropagation**



To the Notebook!

Learning the structure of molecular graphs

Molecular Graph Representation



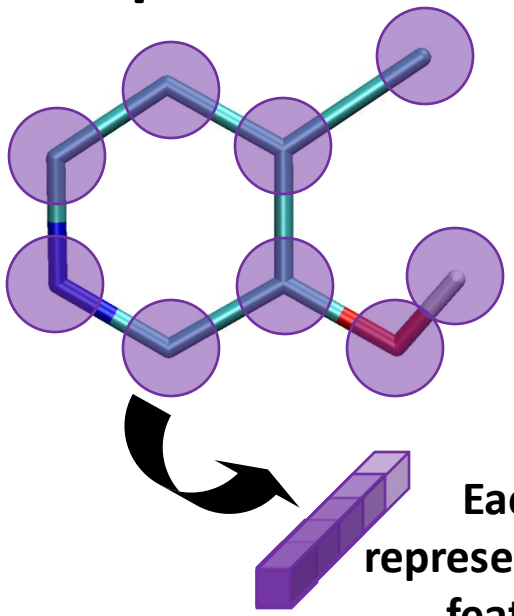
Each atom is represented by a N x 1 feature vector

What's inside this feature vector?

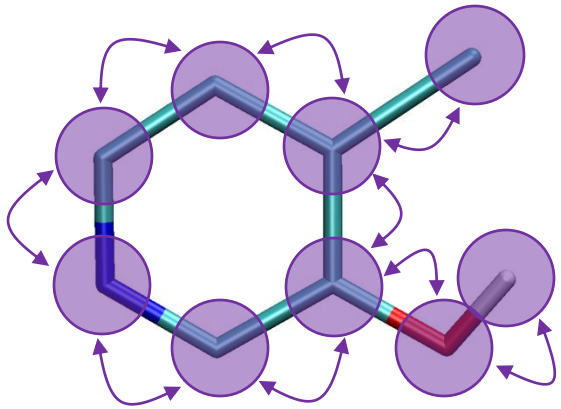
Feature	Type Feature	Description	Size
Atom type	Atom	Type of atom by atomic number	4
Aromaticity	Atom	Whether this atom is part of an aromatic system	2
Number of bonds	Atom	Number of bonds the atom is involved in	3
Number of H_2 Bonds	Atom	Number of bonded hydrogen atoms	4
Hybridization	Atom	sp, sp2, sp3, sp3d, or sp3d2	2
Implicit Valence	Atom	Number of implicit H_2 on the atom	4

Learning the structure of molecular graphs

Molecular Graph Representation



Each atom is represented by a N x 1 feature vector



Message Passing- Each Node Learns about its neighborhood

1. Calculate messages between all pairs of nodes

$$m_{\{u \to v\}}^{t} = MLP\big(concat(h_u^{t-1}, h_v^{t-1}, e_{uv})\big)$$

2. For each node, aggregate all received messages

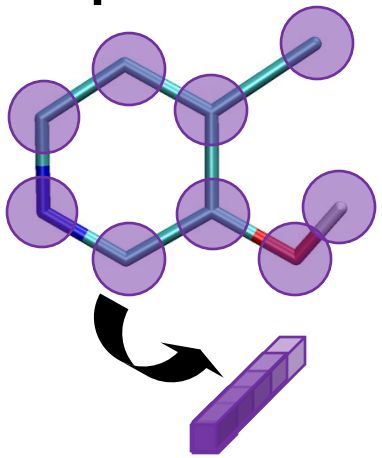
$$M_v^t = \sum_{i=1}^{N} m_{\{i \to v\}}^{t}$$

3. Combine previous embedding with aggregated message to obtain new embedding

$$h_v^t = MLP\big(concat(\, h_v^{t-1}, M_v^t)\big)$$

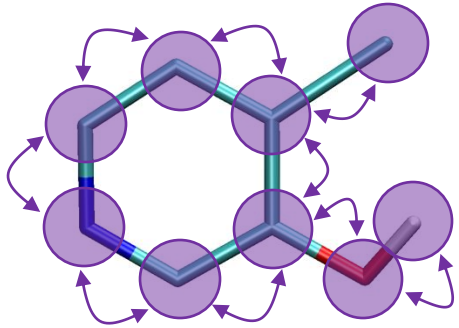
What does the full network look like?

Molecular Graph Representation

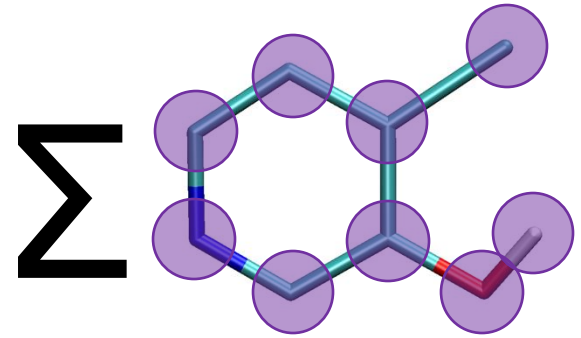


Each atom is represented by a N x 1 feature vector

Message Passing (repeated T times)



Graph Readout



N x 1 Summed Feature Vector

Fully Connected Neural Network (FCNN)

To the Notebook!