Chair
DYNAMICS, CONTROL
AND NUMERICS
FAU

Department of
DATA SCIENCE

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
NATURWISSENSCHAFTLICHE
FAKULTÄT

A Practical Introduction to Control, Numerics and Machine Learning

Day 2

Summer School IFAC CPDE 2022
Workshop on Control of Systems Governed by Partial Differential Equations

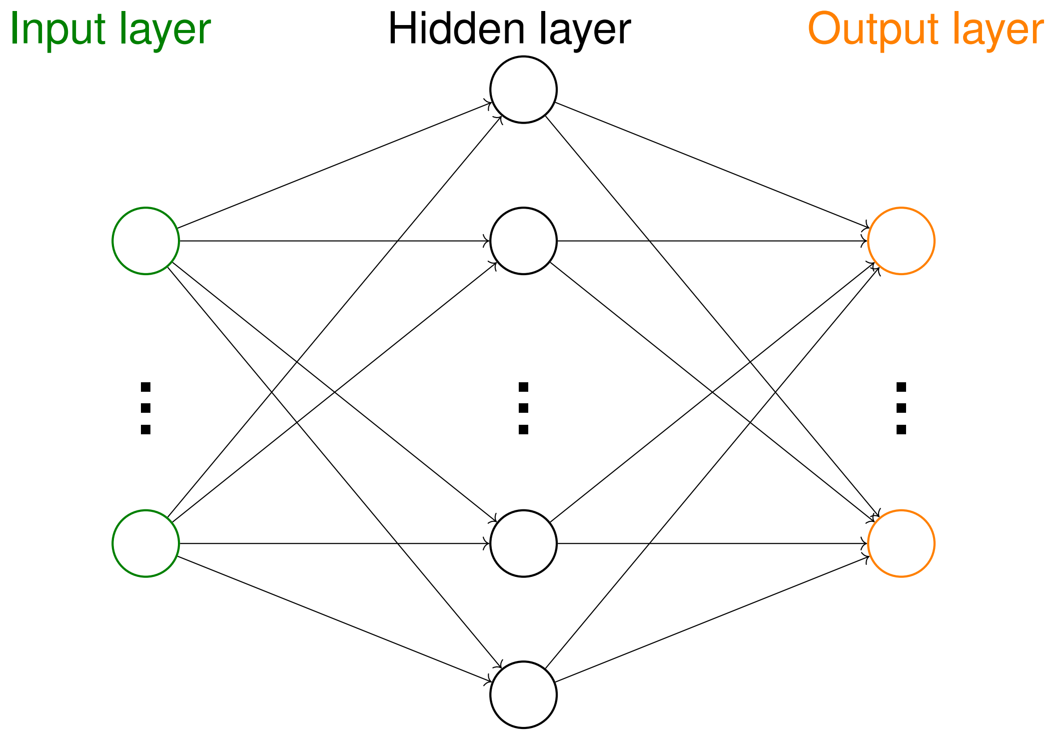
Daniël Veldman

Chair in Dynamics, Control, and Numerics, Friedrich-Alexander-University Erlangen-Nürnberg

Contents

2.A Deep (residual) neural networks and neural ODEs

Neural network

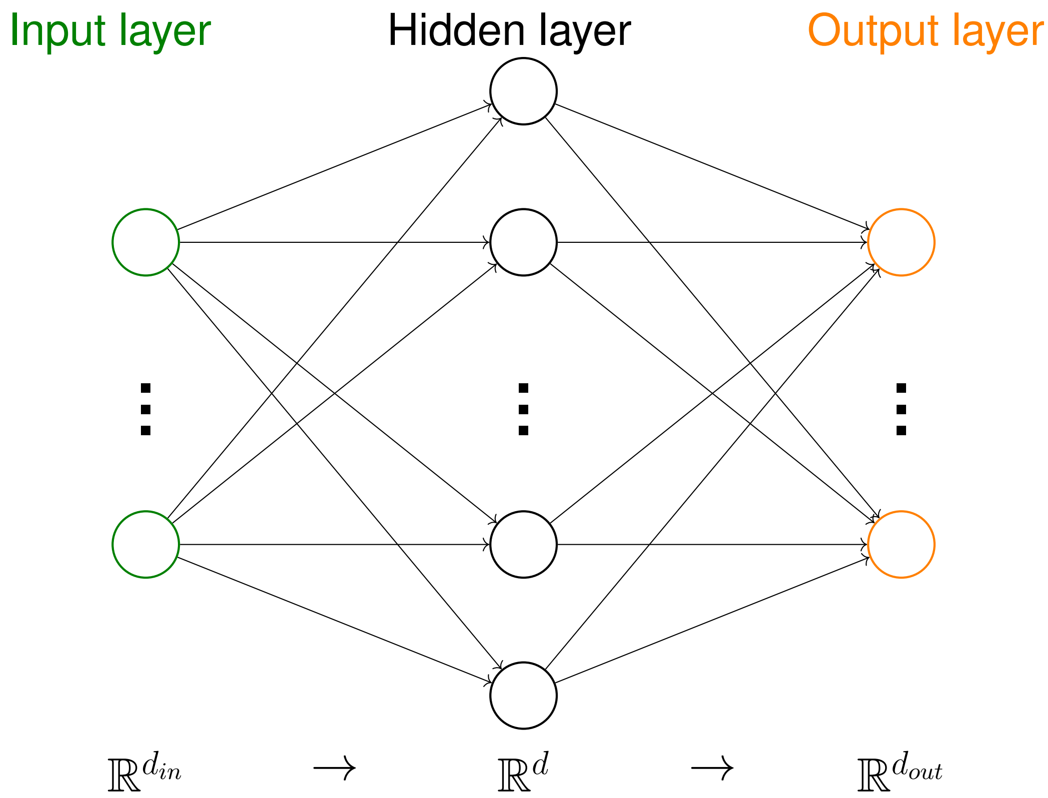


$$\mathbb{R}^{d_{in}} \quad \to \quad \mathbb{R}^{d} \quad \to \quad \mathbb{R}^{d_{out}}$$

The number of nodes in the input layer is the number of inputs.
Each arrow represents a linear map.
Each node in the hidden layer represents a nonlinear operation $\sigma(\cdot)$.
The number of nodes in the output layer is the number of outputs.

$$\mathbf{y} = \mathbf{V}\sigma(\mathbf{W}\mathbf{x} + \mathbf{c}) + \mathbf{b}$$

Representation theorem

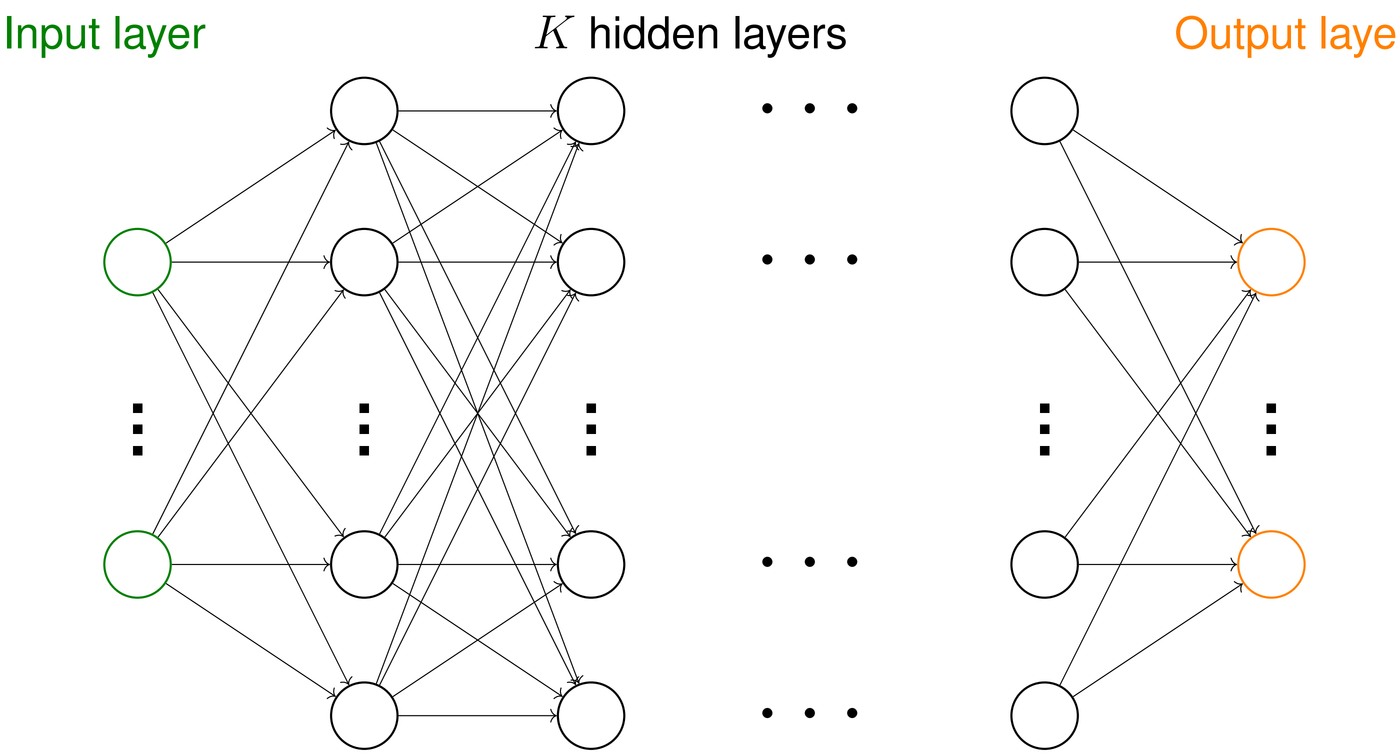


$$\mathbf{y} = \mathbf{V}\sigma(\mathbf{W}\mathbf{x} + \mathbf{c}) + \mathbf{b}$$

Any function $\mathbf{y} = f(\mathbf{x})$ (e.g. in L^2) can be approximated arbitrarily well by a sufficiently **wide** neural network $\mathbf{y} = \mathbf{V}\sigma(\mathbf{W}\mathbf{x} + \mathbf{c}) + \mathbf{b}$.

(Cybenko, 1989)

Deep Neural Networks



$$\mathbb{R}^{d_{in}} \to \mathbb{R}^{d_1} \to \mathbb{R}^{d_2} \to \cdots \to \mathbb{R}^{d_K} \to \mathrm{R}^{d_{out}}$$

There are now multiple hidden layers:

$$\mathbf{x}_0 = \mathbf{x}_{\text{in}}, \qquad \mathbf{x}_k = \mathbf{V}_k \sigma(\mathbf{x}_{k-1}) + \mathbf{b}_k, \qquad \mathbf{y} = \mathbf{x}_K.$$

Residual neural networks (ResNet) and neural ODEs

$$\mathbf{x}_0 = \mathbf{x}_{\text{in}}, \qquad \mathbf{x}_k = \mathbf{V}_k \sigma(\mathbf{x}_{k-1}) + \mathbf{b}_k, \qquad \mathbf{y} = \mathbf{x}_K.$$

When the number of hidden layers K is large,
it is better to consider a **residual neural network (ResNet)**

$$\mathbf{x}_0 = \mathbf{x}_{\text{in}}, \qquad \mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{V}_k \sigma(\mathbf{x}_{k-1}) + \mathbf{b}_k, \qquad \mathbf{y} = \mathbf{x}_K.$$

We can view a ResNet as Forward Euler discretization of the **neural ODE**:

$$\mathbf{x}(0) = \mathbf{x}_{\text{in}}, \qquad \dot{\mathbf{x}}(t) = \mathbf{V}(t) \sigma(\mathbf{x}(t)) + \mathbf{b}(t), \qquad \mathbf{y} = \mathbf{x}(T).$$

How do we find the weights $\mathbf{V}(t)$ and $\mathbf{b}(t)$?

Training a neural ODE

$$\mathbf{x}(0) = \mathbf{x}_{\text{in}}, \qquad \dot{\mathbf{x}}(t) = \mathbf{V}(t)\sigma(\mathbf{x}(t)) + \mathbf{b}(t), \qquad \mathbf{y} = \mathbf{x}(T).$$

Given training data: pairs $(\mathbf{x}^i_{\text{in}}, \mathbf{y}^i_{\text{out}})$, $i = 1, 2, 3, \ldots, I$.
$\mathbf{y}^i_{\text{out}}$ is the desired output for the input $\mathbf{x}^i_{\text{in}}$.

For certain weights $\mathbf{V}(t)$ and $\mathbf{b}(t)$, the output resulting from the input $\mathbf{x}^i_{\text{in}}$ is $\mathbf{x}^i(T)$

We thus want to minimize

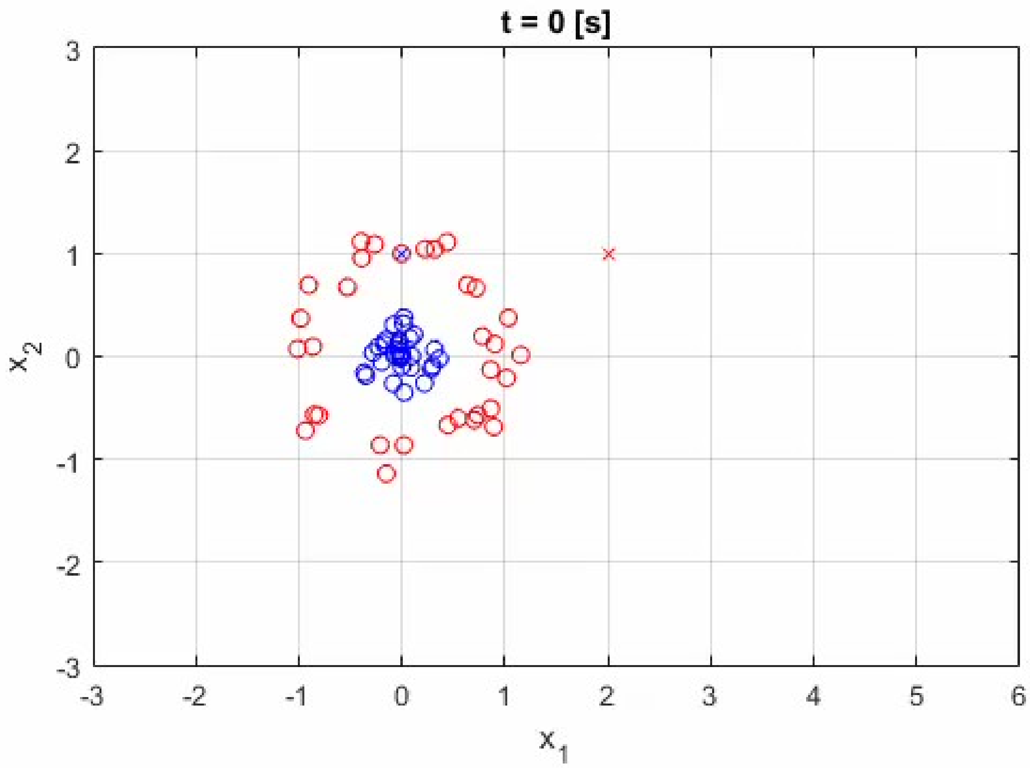
$$J(\mathbf{V}, \mathbf{b}) = \frac{1}{2} \sum_{i=1}^{I} |\mathbf{x}^i(T) - \mathbf{y}^i_{\text{out}}|^2 + \frac{w_1}{2} \sum_{i=1}^{I} \int_0^T |\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}}|^2 \, \mathrm{d}t + \frac{w_2}{2} \int_0^T \left(\|\mathbf{V}(t)\|_F^2 + |\mathbf{b}(t)|^2 \right) \mathrm{d}t.$$

subject to the dynamics (for $i = 1, 2, 3, \ldots, I$)

$$\mathbf{x}^i(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^i(t) = \mathbf{V}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{b}(t).$$

Note: $\mathbf{b}(t)$ is a vector, but $\mathbf{V}(t)$ is a (square) matrix.
Note: $\| \cdot \|_F$ denotes the Frobenius norm, $\|\mathbf{V}\|_F := \sqrt{\text{trace}(\mathbf{V}^\top \mathbf{V})}$.

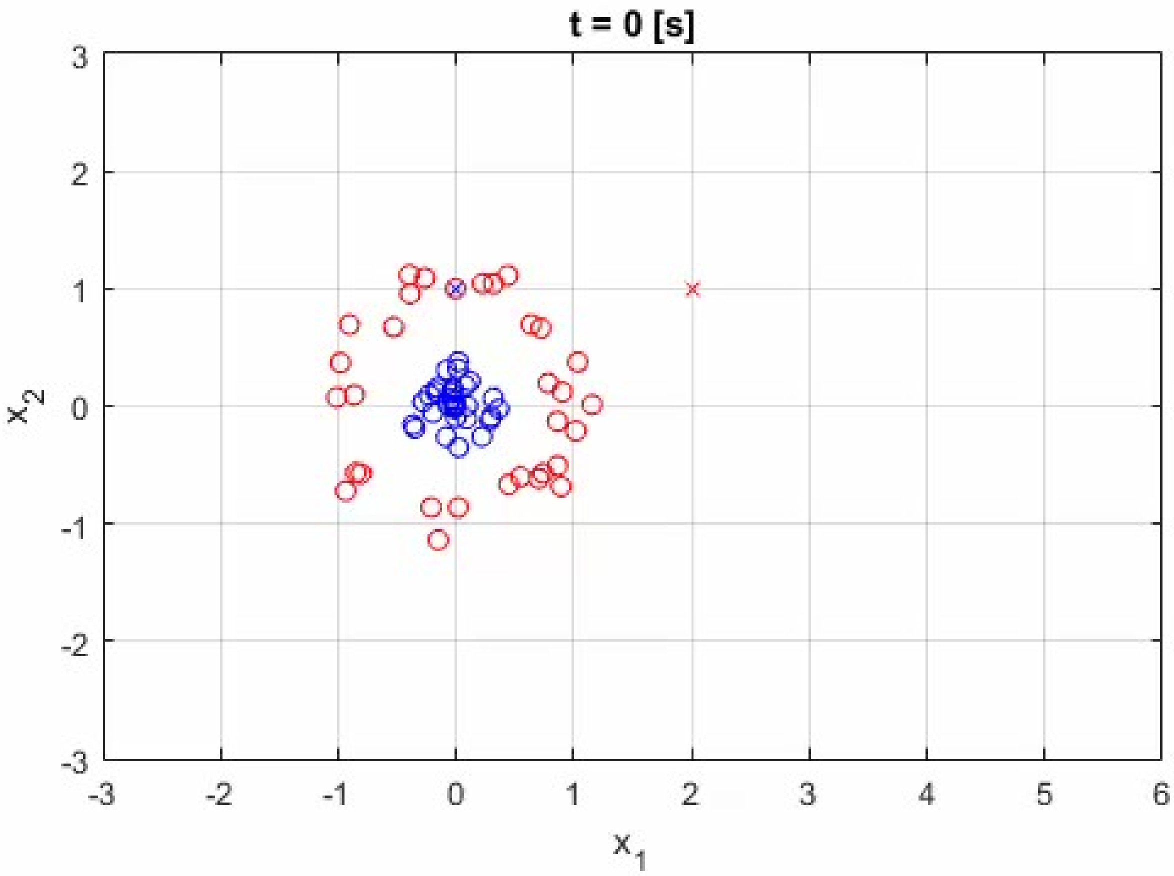
An example: initial data $\mathrm{x}_{\mathrm{in}}^i$



blue data points have target $\mathbf{y}_{\mathrm{out}}^i = (0, 1)$.

red data points have target $\mathbf{y}_{\mathrm{out}}^i = (2, 1)$.

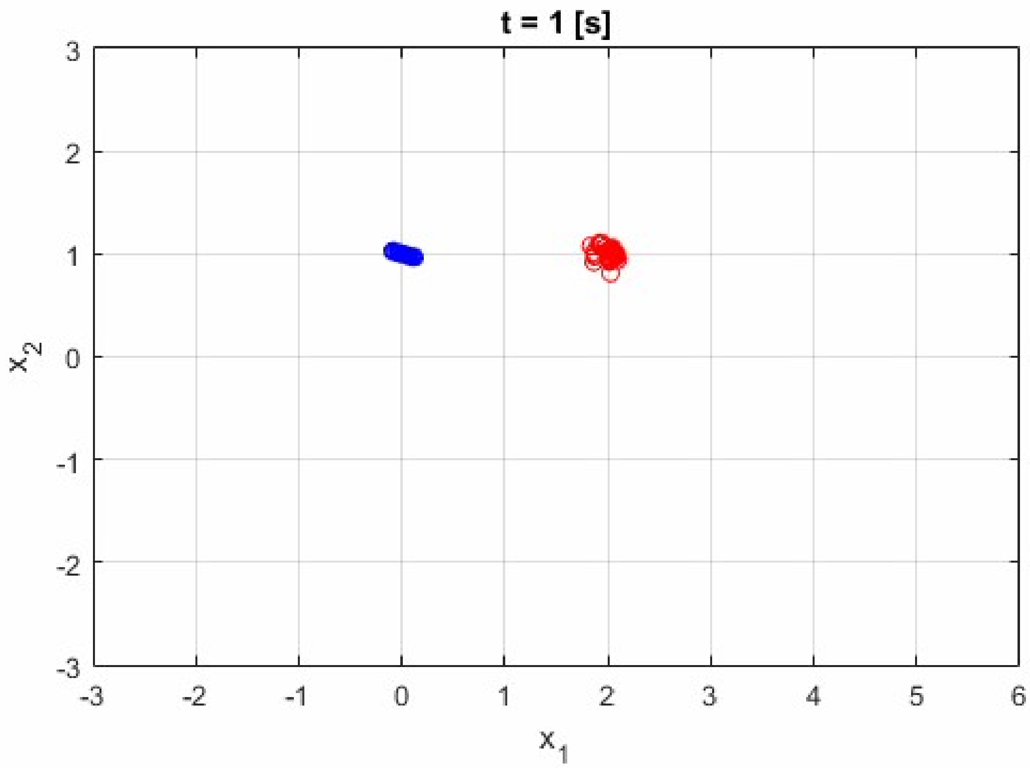
An example: evolution of $\mathbf{x}^i(t)$



blue data points have target $\mathbf{y}^i_{\text{out}} = (0, 1)$.

red data points have target $\mathbf{y}^i_{\text{out}} = (2, 1)$.

An example: final data $\mathrm{x}^i(T)$



blue data points have target $\mathbf{y}^i_{\mathrm{out}} = (0, 1)$.

red data points have target $\mathbf{y}^i_{\mathrm{out}} = (2, 1)$.

Chair
DYNAMICS, CONTROL
AND NUMERICS
FAU

Department of
DATA SCIENCE

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
NATURWISSENSCHAFTLICHE
FAKULTÄT

2.B Sensitivity analysis with neural ODEs

The directional derivative w.r.t. $\mathrm{b}(t)$

Consider a perturbation $\tilde{\mathrm{b}}(t)$ and compute:

$$\tilde{\mathbf{x}}^i(t) := \lim_{h\to 0} \frac{\mathbf{x}^{ih}(t) - \mathbf{x}^i(t)}{h}$$

where

$$\mathbf{x}^i(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^i(t) = \mathbf{V}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{b}(t),$$

$$\mathbf{x}^{ih}(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^{ih}(t) = \mathbf{V}(t)\sigma(\mathbf{x}^{ih}(t)) + \mathbf{b}(t) + h\tilde{\mathbf{b}}(t).$$

The directional derivative w.r.t. $\mathrm{b}(t)$

Consider a perturbation $\tilde{\mathbf{b}}(t)$ and compute:

$$\tilde{\mathbf{x}}^i(t) := \lim_{h\to 0} \frac{\mathbf{x}^{ih}(t) - \mathbf{x}^i(t)}{h}$$

where

$$\mathbf{x}^i(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^i(t) = \mathbf{V}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{b}(t),$$

$$\mathbf{x}^{ih}(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^{ih}(t) = \mathbf{V}(t)\sigma(\mathbf{x}^{ih}(t)) + \mathbf{b}(t) + h\tilde{\mathbf{b}}(t).$$

Because

$$\frac{\dot{\mathbf{x}}^{ih} - \dot{\mathbf{x}}^i}{h} = \frac{\mathbf{V}\sigma(\mathbf{x}^{ih}) + \mathbf{b} + h\tilde{\mathbf{b}} - \mathbf{V}\sigma(\mathbf{x}^i) - \mathbf{b}}{h} = \mathbf{V}\frac{\sigma(\mathbf{x}^{ih}) - \sigma(\mathbf{x})}{h} + \tilde{\mathbf{b}},$$

taking the limit $h \to 0$ shows that $\tilde{\mathbf{x}}^i(t)$ satisfies

$$\tilde{\mathbf{x}}^i(0) = 0, \qquad \dot{\tilde{\mathbf{x}}}^i(t) = \mathbf{V}(t)\mathrm{diag}\left(\frac{\mathrm{d}\sigma}{\mathrm{d}x}(\mathbf{x}^i(t))\right)\tilde{\mathbf{x}}^i(t) + \tilde{\mathbf{b}}(t).$$

The directional derivative w.r.t. $\mathrm{b}(t)$

Consider a perturbation $\tilde{\mathbf{b}}(t)$ and compute:

$$\tilde{\mathbf{x}}^i(t) := \lim_{h\to 0} \frac{\mathbf{x}^{ih}(t) - \mathbf{x}^i(t)}{h}$$

where

$$\mathbf{x}^i(0) = \mathbf{x}^i_{\mathrm{in}}, \qquad \dot{\mathbf{x}}^i(t) = \mathbf{V}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{b}(t),$$

$$\mathbf{x}^{ih}(0) = \mathbf{x}^i_{\mathrm{in}}, \qquad \dot{\mathbf{x}}^{ih}(t) = \mathbf{V}(t)\sigma(\mathbf{x}^{ih}(t)) + \mathbf{b}(t) + h\tilde{\mathbf{b}}(t).$$

Because

$$\frac{\dot{\mathbf{x}}^{ih} - \dot{\mathbf{x}}^i}{h} = \frac{\mathbf{V}\sigma(\mathbf{x}^{ih}) + \mathbf{b} + h\tilde{\mathbf{b}} - \mathbf{V}\sigma(\mathbf{x}^i) - \mathbf{b}}{h} = \mathbf{V}\frac{\sigma(\mathbf{x}^{ih}) - \sigma(\mathbf{x})}{h} + \tilde{\mathbf{b}},$$

taking the limit $h \to 0$ shows that $\tilde{\mathbf{x}}^i(t)$ satisfies

$$\tilde{\mathbf{x}}^i(0) = 0, \qquad \dot{\tilde{\mathbf{x}}}^i(t) = \mathbf{V}(t)\mathrm{diag}\left(\frac{\mathrm{d}\sigma}{\mathrm{d}x}(\mathbf{x}^i(t))\right)\tilde{\mathbf{x}}^i(t) + \tilde{\mathbf{b}}(t).$$

We then obtain

$$\langle \nabla_{\mathbf{b}} J, \tilde{\mathbf{b}} \rangle = \lim_{h\to 0} \frac{J(\mathbf{V}, \mathbf{b} + h\tilde{\mathbf{b}}) - J(\mathbf{V}, \mathbf{b})}{h} = \sum_{i=1}^{I} (\mathbf{x}^i(T) - \mathbf{y}^i_{\mathrm{out}})^\top \tilde{\mathbf{x}}^i(T) +$$

$$w_1 \sum_{i=1}^{I} \int_0^T (\mathbf{x}^i(t) - \mathbf{y}^i_{\mathrm{out}})^\top \tilde{\mathbf{x}}^i(t)\,\mathrm{d}t + w_2 \int_0^T (\mathbf{b}(t))^\top \tilde{\mathbf{b}}(t)\,\mathrm{d}t.$$

The directional derivative w.r.t. $\mathrm{V}(t)$

Consider a perturbation $\hat{\mathbf{V}}(t)$ and compute:

$$\hat{\mathbf{x}}^i(t) := \lim_{h\to 0} \frac{\mathbf{x}^{ih}(t) - \mathbf{x}^i(t)}{h}$$

where

$$\mathbf{x}^i(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^i(t) = \mathbf{V}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{b}(t),$$

$$\mathbf{x}^{ih}(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^{ih}(t) = (\mathbf{V}(t) + h\hat{\mathbf{V}}(t))\sigma(\mathbf{x}^{ih}(t)) + \mathbf{b}(t).$$

The directional derivative w.r.t. $\mathrm{V}(t)$

Consider a perturbation $\hat{\mathbf{V}}(t)$ and compute:

$$\hat{\mathbf{x}}^i(t) := \lim_{h\to 0} \frac{\mathbf{x}^{ih}(t) - \mathbf{x}^i(t)}{h}$$

where

$$\mathbf{x}^i(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^i(t) = \mathbf{V}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{b}(t),$$
$$\mathbf{x}^{ih}(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^{ih}(t) = (\mathbf{V}(t) + h\hat{\mathbf{V}}(t))\sigma(\mathbf{x}^{ih}(t)) + \mathbf{b}(t).$$

Because

$$\frac{\dot{\mathbf{x}}^{ih} - \dot{\mathbf{x}}^i}{h} = \frac{(\mathbf{V} + h\hat{\mathbf{V}})\sigma(\mathbf{x}^{ih}) + \mathbf{b} - \mathbf{V}\sigma(\mathbf{x}^{ih}) + \mathbf{V}\sigma(\mathbf{x}^{ih}) - \mathbf{V}\sigma(\mathbf{x}^i) - \mathbf{b}}{h},$$

taking the limit $h \to 0$ shows that $\hat{\mathbf{x}}^i(t)$ satisfies

$$\hat{\mathbf{x}}^i(0) = 0, \qquad \dot{\hat{\mathbf{x}}}^i(t) = \hat{\mathbf{V}}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{V}(t)\mathrm{diag}\left(\frac{\mathrm{d}\sigma}{\mathrm{d}x}(\mathbf{x}^i(t))\right)\hat{\mathbf{x}}^i(t).$$

The directional derivative w.r.t. $\mathrm{V}(t)$

Consider a perturbation $\hat{\mathbf{V}}(t)$ and compute:

$$\hat{\mathbf{x}}^i(t) := \lim_{h\to 0} \frac{\mathbf{x}^{ih}(t) - \mathbf{x}^i(t)}{h}$$

where

$$\mathbf{x}^i(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^i(t) = \mathbf{V}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{b}(t),$$
$$\mathbf{x}^{ih}(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^{ih}(t) = (\mathbf{V}(t) + h\hat{\mathbf{V}}(t))\sigma(\mathbf{x}^{ih}(t)) + \mathbf{b}(t).$$

Because

$$\frac{\dot{\mathbf{x}}^{ih} - \dot{\mathbf{x}}^i}{h} = \frac{(\mathbf{V} + h\hat{\mathbf{V}})\sigma(\mathbf{x}^{ih}) + \mathbf{b} - \mathbf{V}\sigma(\mathbf{x}^{ih}) + \mathbf{V}\sigma(\mathbf{x}^{ih}) - \mathbf{V}\sigma(\mathbf{x}^i) - \mathbf{b}}{h},$$

taking the limit $h \to 0$ shows that $\hat{\mathbf{x}}^i(t)$ satisfies

$$\hat{\mathbf{x}}^i(0) = 0, \qquad \dot{\hat{\mathbf{x}}}^i(t) = \hat{\mathbf{V}}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{V}(t)\mathrm{diag}\left(\frac{\mathrm{d}\sigma}{\mathrm{d}x}(\mathbf{x}^i(t))\right)\hat{\mathbf{x}}^i(t).$$

We then obtain

$$\langle \nabla_{\mathbf{V}} J, \hat{\mathbf{V}}\rangle_F = \lim_{h\to 0}\frac{J(\mathbf{V} + h\hat{\mathbf{V}}, \mathbf{b}) - J(\mathbf{V}, \mathbf{b})}{h} = \sum_{i=1}^{I} (\mathbf{x}^i(T) - \mathbf{y}^i_{\text{out}})^\top \hat{\mathbf{x}}^i(T) +$$
$$w_1 \sum_{i=1}^{I} \int_0^T (\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}})^\top \hat{\mathbf{x}}^i(t)\,\mathrm{d}t + w_2 \int_0^T \langle \mathbf{V}(t), \hat{\mathbf{V}}(t)\rangle_F\,\mathrm{d}t.$$

The adjoint state

Just as in the previous lecture, we need the adjoint state to find the gradients.

We now define the adjoint states $\boldsymbol{\varphi}^i(t)$:

$$\boldsymbol{\varphi}^i(T) = \mathbf{x}^i(T) - \mathbf{y}^i_{\text{out}}, \qquad -\dot{\boldsymbol{\varphi}}^i(t) = \left(\mathbf{V}(t)\text{diag}\left(\frac{\mathrm{d}\sigma}{\mathrm{d}x}(\mathbf{x}^i(t))\right)\right)^\top \boldsymbol{\varphi}^i(t) + w_1(\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}}),$$

for $i = 1, 2, 3, \ldots, I$.

Note: the final condition is now nonzero because the state $\mathbf{x}^i(T)$ at the final time appears in the cost functional.

Question 1

Write:

$$\mathbf{A}(t) = \mathbf{V}(t)\mathrm{diag}\left(\frac{\mathrm{d}\sigma}{\mathrm{d}x}(\mathbf{x}^i(t))\right).$$

From the previous slides

$$\tilde{\mathbf{x}}^i(0) = 0, \qquad \dot{\tilde{\mathbf{x}}}^i(t) = \mathbf{A}(t)\tilde{\mathbf{x}}^i(t) + \tilde{\mathbf{b}}(t).$$

$$\boldsymbol{\varphi}^i(T) = \mathbf{x}^i(T) - \mathbf{y}^i_{\mathrm{out}}, \qquad -\dot{\boldsymbol{\varphi}}^i(t) = (\mathbf{A}(t))^\top \boldsymbol{\varphi}^i(t) + w_1(\mathbf{x}^i(t) - \mathbf{y}^i_{\mathrm{out}}).$$

What is

$$\int_0^T \frac{d}{dt}\left((\boldsymbol{\varphi}^i(t))^\top \tilde{\mathbf{x}}^i(t)\right)\,\mathrm{d}t = (\boldsymbol{\varphi}^i(T))^\top \tilde{\mathbf{x}}^i(T) - (\boldsymbol{\varphi}^i(0))^\top \tilde{\mathbf{x}}^i(0)?$$

A) 0

B) $\mathbf{x}^i(T) - \mathbf{y}^i_{\mathrm{out}}$

C) $\left(\mathbf{x}^i(T) - \mathbf{y}^i_{\mathrm{out}}\right)^\top \tilde{\mathbf{x}}^i(T)$

D) $-\left(\mathbf{x}^i(T) - \mathbf{y}^i_{\mathrm{out}}\right)^\top \tilde{\mathbf{x}}^i(T)$

E) None of the above

Question 2

We have that:

$$\tilde{\mathbf{x}}^i(0) = 0, \qquad \dot{\tilde{\mathbf{x}}}^i(t) = \mathbf{A}(t)\tilde{\mathbf{x}}^i(t) + \tilde{\mathbf{b}}(t).$$

$$\boldsymbol{\varphi}^i(T) = \mathbf{x}^i(T) - \mathbf{y}^i_{\text{out}}, \qquad -\dot{\boldsymbol{\varphi}}^i(t) = (\mathbf{A}(t))^\top \boldsymbol{\varphi}^i(t) + w_1(\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}}).$$

What is

$$\int_0^T \frac{d}{dt}\left((\boldsymbol{\varphi}^i(t))^\top \tilde{\mathbf{x}}^i(t)\right)\, \mathrm{d}t = \int_0^T \left(\dot{\boldsymbol{\varphi}}^i(t)\right)^\top \tilde{\mathbf{x}}^i(t)\, \mathrm{d}t + \int_0^T (\boldsymbol{\varphi}^i(t))^\top \dot{\tilde{\mathbf{x}}}^i(t)\, \mathrm{d}t?$$

A) $\int_0^T \left((\mathbf{A}(t))^\top \boldsymbol{\varphi}^i(t) + w_1(\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}})\right)^\top \tilde{\mathbf{x}}^i(t)\, \mathrm{d}t + \int_0^T (\boldsymbol{\varphi}^i(t))^\top \left(\mathbf{A}(t)\tilde{\mathbf{x}}^i(t) + \tilde{\mathbf{b}}(t)\right)\, \mathrm{d}t$

B)
$-\int_0^T \left((\mathbf{A}(t))^\top \boldsymbol{\varphi}^i(t) + w_1(\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}})\right)^\top \tilde{\mathbf{x}}^i(t)\, \mathrm{d}t + \int_0^T (\boldsymbol{\varphi}^i(t))^\top \left(\mathbf{A}(t)\tilde{\mathbf{x}}^i(t) + \tilde{\mathbf{b}}(t)\right)\, \mathrm{d}t$

C) $w_1 \int_0^T (\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}})^\top \tilde{\mathbf{x}}^i(t)\, \mathrm{d}t + \int_0^T (\boldsymbol{\varphi}^i(t))^\top \tilde{\mathbf{b}}(t)\, \mathrm{d}t$

D) $-w_1 \int_0^T (\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}})^\top \tilde{\mathbf{x}}^i(t)\, \mathrm{d}t + \int_0^T (\boldsymbol{\varphi}^i(t))^\top \tilde{\mathbf{b}}(t)\, \mathrm{d}t$

The gradient w.r.t. $\mathrm{b}(t)$

$$\langle \nabla_{\mathbf{b}} J, \tilde{\mathbf{b}} \rangle = \lim_{h \to 0} \frac{J(\mathbf{V}, \mathbf{b} + h\tilde{\mathbf{b}}) - J(\mathbf{V}, \mathbf{b})}{h} = \sum_{i=1}^{I} (\mathbf{x}^i(T) - \mathbf{y}^i_{\mathrm{out}})^\top \tilde{\mathbf{x}}^i(T) + w_1 \sum_{i=1}^{I} \int_0^T (\mathbf{x}^i(t) - \mathbf{y}^i_{\mathrm{out}})^\top \tilde{\mathbf{x}}^i(t)\,\mathrm{d}t + w_2 \int_0^T (\mathbf{b}(t))^\top \tilde{\mathbf{b}}(t)\,\mathrm{d}t.$$

From the previous questions:

$$\left(\mathbf{x}^i(T) - \mathbf{y}^i_{\mathrm{out}}\right)^\top \tilde{\mathbf{x}}^i(T) = -w_1 \int_0^T (\mathbf{x}^i(t) - \mathbf{y}^i_{\mathrm{out}})^\top \tilde{\mathbf{x}}^i(t)\,\mathrm{d}t + \int_0^T (\boldsymbol{\varphi}^i(t))^\top \tilde{\mathbf{b}}(t)\,\mathrm{d}t$$

Resulting expression:

$$\langle \nabla_{\mathbf{b}} J, \tilde{\mathbf{b}} \rangle = \sum_{i=1}^{I} \int_0^T (\boldsymbol{\varphi}^i(t))^\top \tilde{\mathbf{b}}(t)\,\mathrm{d}t + w_2 \int_0^T (\mathbf{b}(t))^\top \tilde{\mathbf{b}}(t)\,\mathrm{d}t$$

Resulting gradient (w.r.t. the standard L^2-innerproduct):

$$(\nabla_{\mathbf{b}} J)(t) = \sum_{i=1}^{I} \boldsymbol{\varphi}^i(t) + w_2 \mathbf{b}(t)$$

The gradient w.r.t. $\mathbf{V}(t)$

$$\langle \nabla_{\mathbf{V}} J, \hat{\mathbf{V}} \rangle_F = \lim_{h \to 0} \frac{J(\mathbf{V} + h\hat{\mathbf{V}}, \mathbf{b}) - J(\mathbf{V}, \mathbf{b})}{h} = \sum_{i=1}^{I} (\mathbf{x}^i(T) - \mathbf{y}^i_{\mathrm{out}})^\top \hat{\mathbf{x}}^i(T) +$$
$$w_1 \sum_{i=1}^{I} \int_0^T (\mathbf{x}^i(t) - \mathbf{y}^i_{\mathrm{out}})^\top \hat{\mathbf{x}}^i(t)\, \mathrm{d}t + w_2 \int_0^T \langle \mathbf{V}(t), \hat{\mathbf{V}}(t) \rangle_F \, \mathrm{d}t.$$

We can now verify similarly as before that

$$(\mathbf{x}^i(T) - \mathbf{y}^i_{\mathrm{out}})^\top \hat{\mathbf{x}}(T) = -w_1 \int_0^T (\mathbf{x}^i(t) - \mathbf{y}^i_{\mathrm{out}})^\top \hat{\mathbf{x}}^i(t)\, \mathrm{d}t + \int_0^T (\boldsymbol{\varphi}^i(t))^\top \hat{\mathbf{V}}(t) \sigma(\mathbf{x}^i(t))\, \mathrm{d}t$$

In a similar way, we can show that the gradient w.r.t. $\mathbf{V}(t)$ (w.r.t. the Frobenius inner product) is

$$(\nabla_{\mathbf{V}} J)(t) = \sum_{i=1}^{I} \sigma(\mathbf{x}^i(t)) (\boldsymbol{\varphi}^i(t))^\top + w_2 \mathbf{V}(t).$$

An algorithm for the computation of the gradients

Computation of the gradients $\nabla_{\mathbf{V}} J(\mathbf{V}, \mathbf{b})$ and $\nabla_{\mathbf{b}} J(\mathbf{V}, \mathbf{b})$ (gradient in the point $(\mathbf{V}, \mathbf{b})$)

- Compute for $i = 1, 2, 3, \ldots, I$ the solutions of

$$\mathbf{x}^i(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^i(t) = \mathbf{V}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{b}(t).$$

- Compute for $i = 1, 2, 3, \ldots, I$ the solutions of

$$\boldsymbol{\varphi}^i(T) = \mathbf{x}^i(T) - \mathbf{y}^i_{\text{out}}, \qquad -\dot{\boldsymbol{\varphi}}^i(t) = \left(\mathbf{V}(t)\operatorname{diag}\left(\frac{\mathrm{d}\sigma}{\mathrm{d}x}(\mathbf{x}^i(t))\right)\right)^\top \boldsymbol{\varphi}^i(t) + w_1(\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}}),$$

- The gradients are now given by

$$(\nabla_{\mathbf{V}} J)(t) = \sum_{i=1}^{I} \sigma(\mathbf{x}^i(t))(\boldsymbol{\varphi}^i(t))^\top + w_2 \mathbf{V}(t), (\nabla_{\mathbf{b}} J)(t) \quad = \sum_{i=1}^{I} \boldsymbol{\varphi}^i(t) + w_2 \mathbf{b}(t)$$

An algorithm for the computation of the gradients

Computation of the gradients $\nabla_{\mathbf{V}} J(\mathbf{V}, \mathbf{b})$ and $\nabla_{\mathbf{b}} J(\mathbf{V}, \mathbf{b})$ (gradient in the point $(\mathbf{V}, \mathbf{b})$)

- Compute for $i = 1, 2, 3, \dots, I$ the solutions of

$$\mathbf{x}^i(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^i(t) = \mathbf{V}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{b}(t).$$

- Compute for $i = 1, 2, 3, \dots, I$ the solutions of

$$\boldsymbol{\varphi}^i(T) = \mathbf{x}^i(T) - \mathbf{y}^i_{\text{out}}, \qquad -\dot{\boldsymbol{\varphi}}^i(t) = \left(\mathbf{V}(t)\text{diag}\left(\frac{\mathrm{d}\sigma}{\mathrm{d}x}(\mathbf{x}^i(t))\right)\right)^\top \boldsymbol{\varphi}^i(t) + w_1(\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}}),$$

- The gradients are now given by

$$(\nabla_{\mathbf{V}} J)(t) = \sum_{i=1}^{I} \sigma(\mathbf{x}^i(t))(\boldsymbol{\varphi}^i(t))^\top + w_2 \mathbf{V}(t), \quad (\nabla_{\mathbf{b}} J)(t) = \sum_{i=1}^{I} \boldsymbol{\varphi}^i(t) + w_2 \mathbf{b}(t)$$

Remark: when I is large, we need to solve many ODEs at each iteration. As we also need many iterations (e.g. 10,000) this can lead to a huge computational cost. **Stochastic gradient descent methods** reduce this cost by considering a randomly selected subset of indices $i \in \{1, 2, \dots, I\}$ at each time step.
We will not go into this further in this lecture.

2.C Training of deep residual neural networks

Cost functional and dynamics

$$\min_{\mathbf{V},\mathbf{b}} J(\mathbf{V},\mathbf{b}) = \frac{1}{2}\sum_{i=1}^{I} |\mathbf{x}^i(T) - \mathbf{y}^i_{\text{out}}|^2 + \frac{w_1}{2}\sum_{i=1}^{I}\int_0^T |\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}}|^2 \,\mathrm{d}t + \frac{w_2}{2}\int_0^T \left(\|\mathbf{V}(t)\|_F^2 + |\mathbf{b}(t)|^2\right)\mathrm{d}t.$$

subject to the dynamics (for $i = 1, 2, 3, \ldots, I$)

$$\mathbf{x}^i(0) = \mathbf{x}^i_{\text{in}}, \qquad \dot{\mathbf{x}}^i(t) = \mathbf{V}(t)\sigma(\mathbf{x}^i(t)) + \mathbf{b}(t).$$

We consider a uniform grid $t_k = (k-1)\Delta t$ ($k = 1, 2, \ldots, N_T$), so $\Delta t = T/(N_T - 1)$.
We denote $\mathbf{x}^i_k \approx \mathbf{x}^i(t_k)$ with $k = 1, 2, \ldots, N_T$,
$\mathbf{V}_k = \mathbf{V}(t_k)$ and $\mathbf{b}_k = \mathbf{b}(t_k)$ with $k = 1, 2, \ldots, N_T - 1$.
We discertize with forward Euler:

$$\min_{\mathbf{V}_k,\mathbf{b}_k} J(\mathbf{V},\mathbf{b}) = \frac{1}{2}\sum_{i=1}^{I} |\mathbf{x}^i_{N_T} - \mathbf{y}^i_{\text{out}}|^2 + \frac{w_1\Delta t}{2}\sum_{i=1}^{I}\sum_{k=2}^{N_T-1} |\mathbf{x}^i_k - \mathbf{y}^i_{\text{out}}|^2 + \frac{w_2\Delta t}{2}\sum_{k=1}^{N_T-1}\left(\|\mathbf{V}_k\|_F^2 + |\mathbf{b}_k|^2\right).$$

$$\mathbf{x}^i_1 = \mathbf{x}^i_{\text{in}}, \qquad \mathbf{x}^i_{k+1} = \mathbf{x}^i_k + \Delta t(\mathbf{V}_k\sigma(\mathbf{x}^i_k) + \mathbf{b}_k).$$

Note: Forward Euler gives us precisely the structure of a ResNet.

Adjoint state

In the continuous time setting, we could compute the gradient from the adjoint state:

$$\boldsymbol{\varphi}^i(T) = \mathbf{x}^i(T) - \mathbf{y}^i_{\text{out}}, \qquad -\dot{\boldsymbol{\varphi}}^i(t) = \left(\mathbf{V}(t)\text{diag}\left(\frac{\mathrm{d}\sigma}{\mathrm{d}x}(\mathbf{x}^i(t))\right)\right)^\top \boldsymbol{\varphi}^i(t) + w_1(\mathbf{x}^i(t) - \mathbf{y}^i_{\text{out}}),$$

Adjoint variables are $\boldsymbol{\varphi}_k \approx \boldsymbol{\varphi}(t_k)$, $k = 1, 2, 3, \ldots N_T - 1$.
Compute the adjoint variables starting from

$$\boldsymbol{\varphi}^i_{N_T-1} = \mathbf{x}^i_{N_T} - \mathbf{y}^i_{\text{out}}$$

and then backward in time according to

$$\boldsymbol{\varphi}^i_{k-1} = \boldsymbol{\varphi}^i_k + \Delta t \left(\mathbf{V}_k \text{diag}\left(\frac{\mathrm{d}\sigma}{\mathrm{d}x}(\mathbf{x}^i_k)\right)\right)^\top \boldsymbol{\varphi}^i_k + \Delta t w_1(\mathbf{x}^i_k - \mathbf{y}^i_{\text{out}}).$$

Gradient computation

In the continuous time setting:

$$(\nabla_{\mathbf{V}} J)(t) = \sum_{i=1}^{I} \sigma(\mathbf{x}^i(t))(\boldsymbol{\varphi}^i(t))^\top + w_2 \mathbf{V}(t),$$

$$(\nabla_{\mathbf{b}} J)(t) = \sum_{i=1}^{I} \boldsymbol{\varphi}^i(t) + w_2 \mathbf{b}(t)$$

After discretization:

$$(\nabla_{\mathbf{V}} J)_k = \Delta t \sum_{i=1}^{I} \boldsymbol{\varphi}_k^i \sigma(\mathbf{x}_k^i)^\top + w_2 \Delta t \mathbf{V}_k, \qquad k = 1, 2, \ldots, N_T - 1,$$

$$(\nabla_{\mathbf{b}} J)_k = \Delta t \sum_{i=1}^{I} \boldsymbol{\varphi}_k^i + w_2 \Delta t \mathbf{b}_k, \qquad k = 1, 2, \ldots, N_T - 1.$$

Optimization algorithm

We can now just use a basic gradient descent algorithm to minimize J.
In every interation we thus use the updates

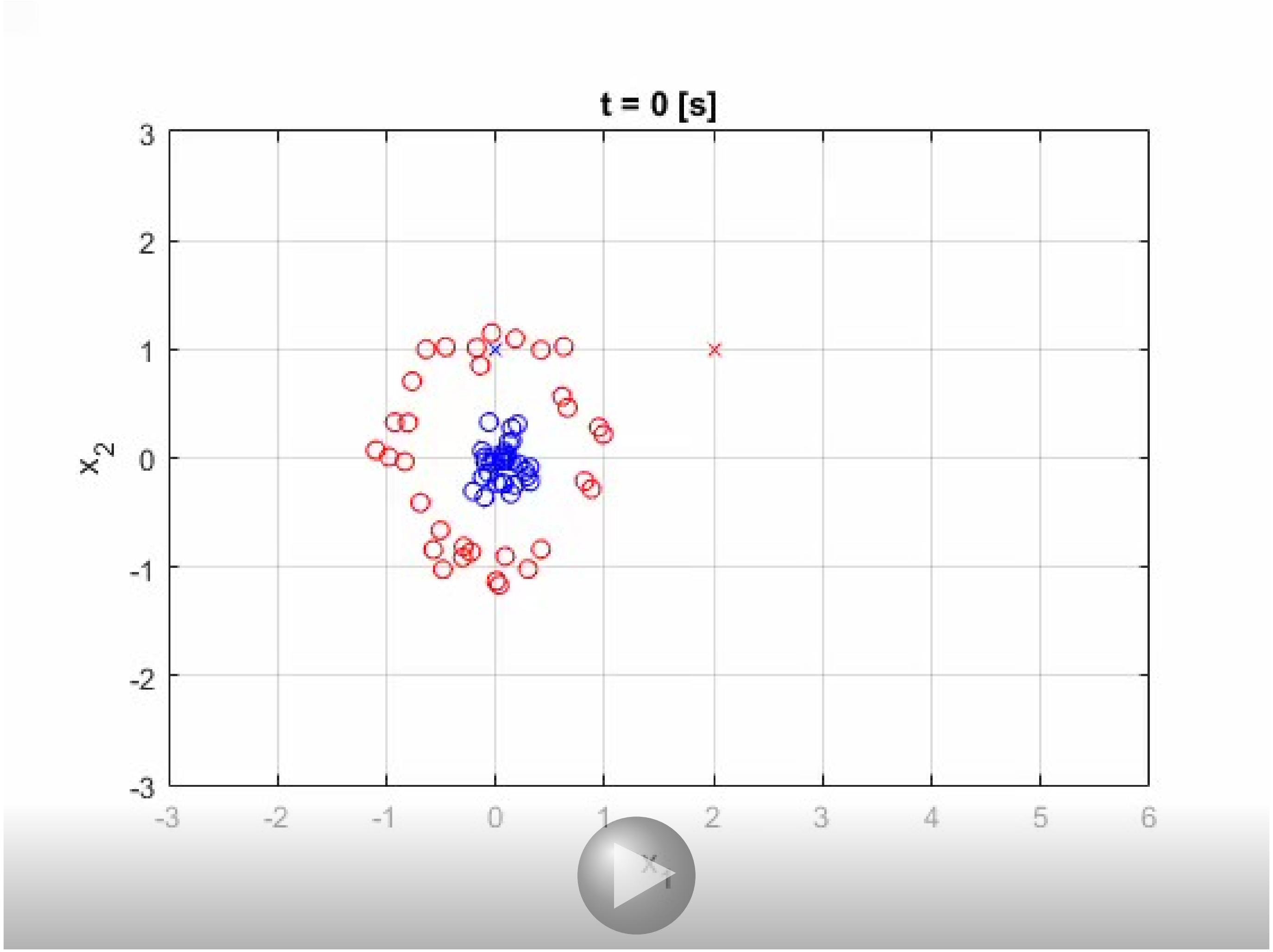
$$\mathbf{V}_k^{j+1} = \mathbf{V}_k^j - \beta_j (\nabla_{\mathbf{V}} J)_k, \qquad \mathbf{b}_k^{j+1} = \mathbf{b}_k^j - \beta_j (\nabla_{\mathbf{b}} J)_k.$$

The stepsize γ_j is also called the learning rate.

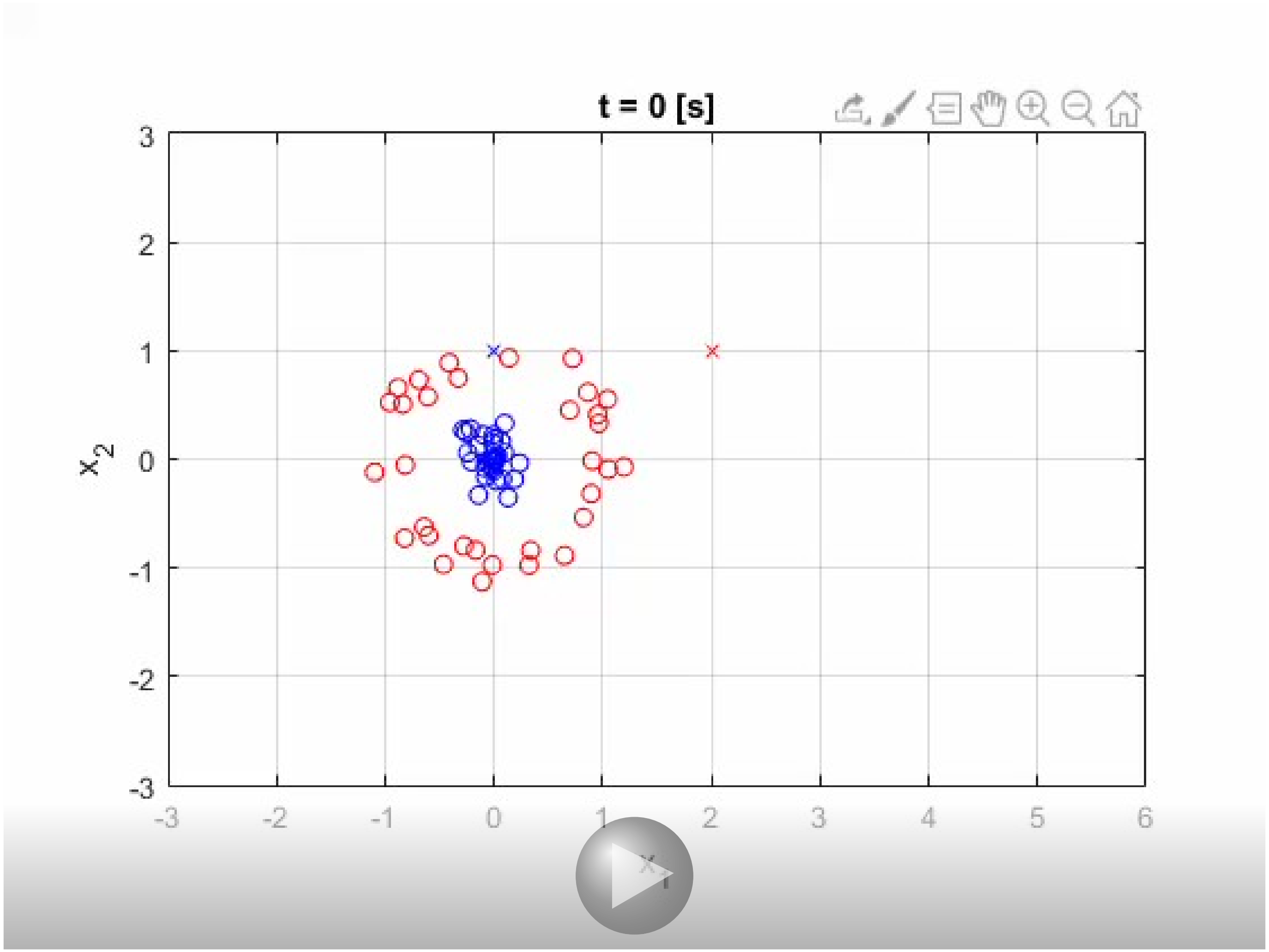
But the problem is now very much nonconvex:

- We cannot guarantee the uniqueness of the (global) minimizer.
- We do not know whether the algorithm converges to a global minimizer.
- The convergence rate is generally slow.
  We need many iterations (e.g. 10,000) to obtain good results.

Example: 100 iterations of gradient descent



Example: 1000 iterations of gradient descent



Example: 10,000 iterations of gradient descent