Neural Mass Network Models as Classifiers

How to transform a biological inspired dynamical system into a classifier.

Raffaele Marino

Università di Firenze

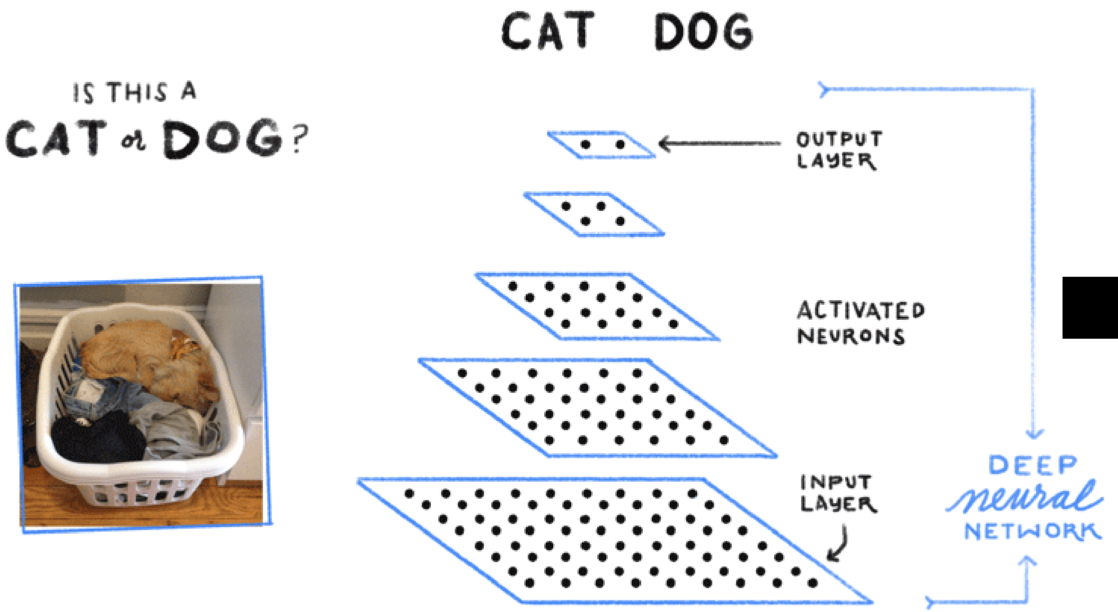
Dipartimento di Fisica e Astronomia

raffaele.marino@unifi.it

UNIVERSITÀ DEGLI STUDI FIRENZE

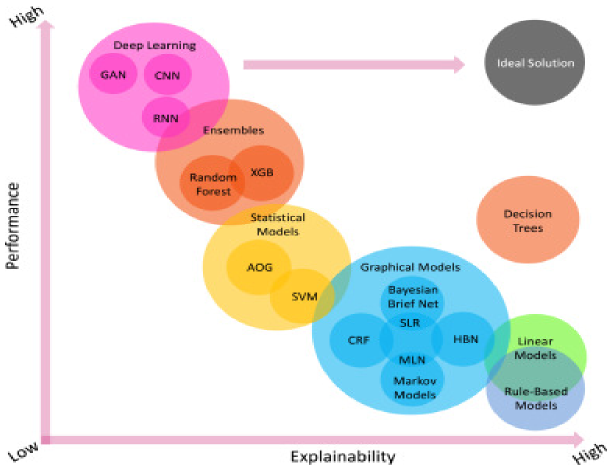
MNESYS

Motivation



Black Box

Where is it dangerous to rely on a black box for making decisions?

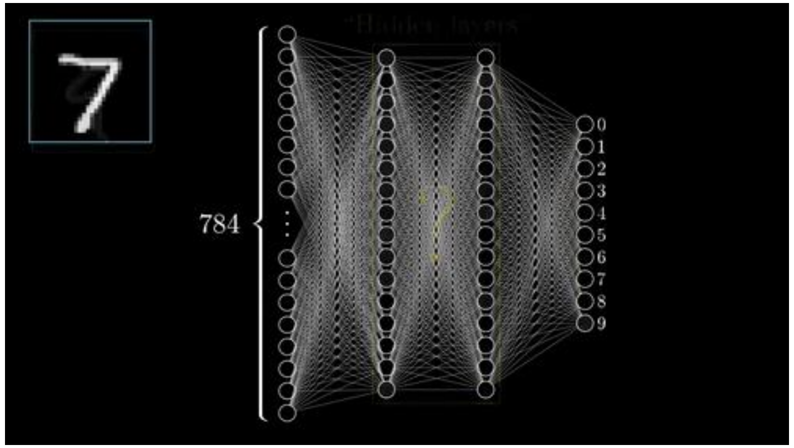


UNIVERSITÀ DEGLI STUDI FIRENZE

MNESYS

Deep Learning

Deep Neural Network

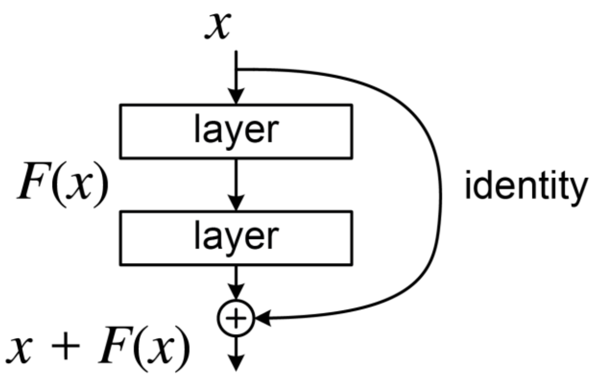


On single neuron

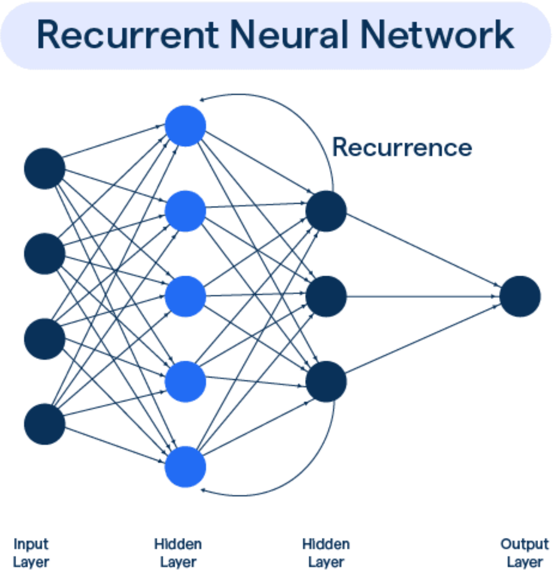
$$y_n = \sigma(\sum_j w_{nj} x_j)$$

$$\vec{y} = f(\vec{x}, \mathbf{w})$$

Residual Neural Network

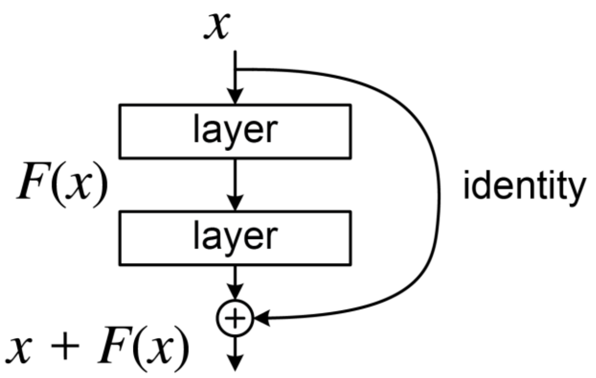


Recurrent Neural Network



The Gap

Residual Neural Network



A Residual Neural Network (or a deep neural network) can be seen as a discrete dynamical system

$$\vec{x}_{n+1} = \vec{x}_n + F(\vec{x}_n, \mathbf{A}_n)$$

It can be viewed as a discretization of the dynamical system

$$\frac{d\vec{x}}{dt} = F(\vec{x}, \mathbf{A}(t))$$

In fact, the simplest discretization, Euler's method, takes the form:

$$\vec{x}_{n+1} = \vec{x}_n + F(\vec{x}_n, \mathbf{A}_n)\Delta t$$

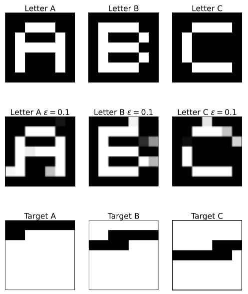
The gap

Does a **continuous dynamical system capable of classifing items exist ?**

SA-NODE: Grasp the idea

Dataset

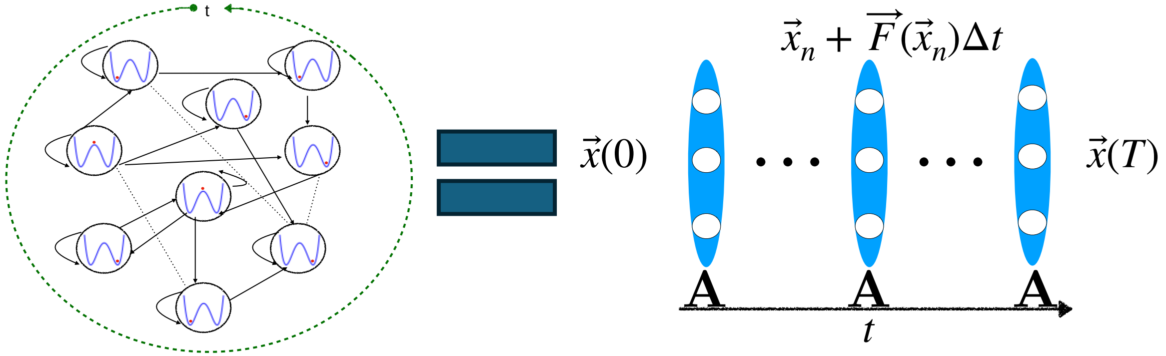
$\mathcal{D} = (\vec{x}, y)^{(j)\in[1,\dots,|\mathcal{D}|]}$



$$\dot{\vec{x}}(t) = \vec{F}(\vec{x}(t), \mathbf{A}),\ \vec{x} \in \mathbb{R}^N$$

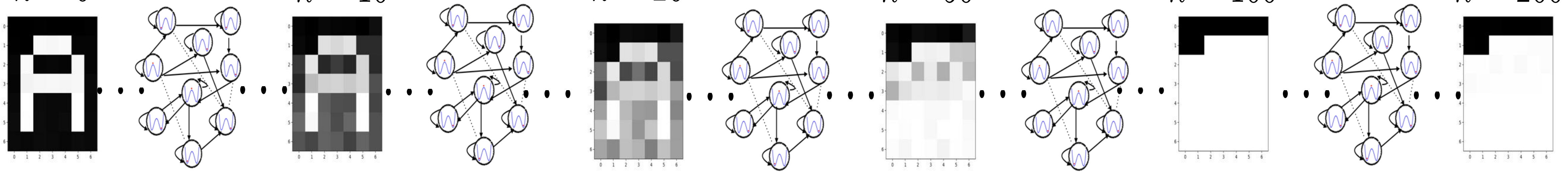
$$\vec{F}(\vec{x}, \mathbf{A}) = -\vec{\nabla} V(\vec{x}(t)) + \beta \mathbf{A} \vec{x}(t)$$

Dynamics



The matrix **A** must be trained to optimize the classification task.

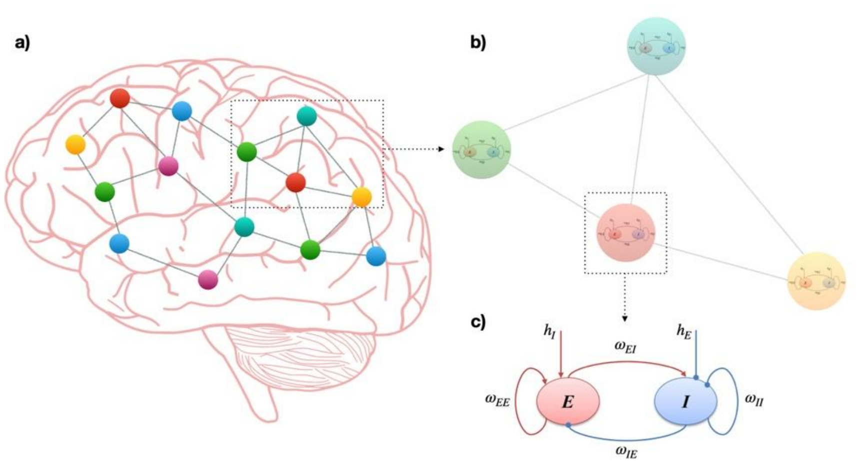
Example in discrete time:



$t = n\Delta t$

Another example: Wilson-Cowan for metapopulation

We consider **a network metapopulation form of the Wilson–Cowan model**. Nodes in the network model will correspond to different subcortical regions of the brain, while edges can represent structural, functional, or effective connectivity between these distinct subcortical regions. Within each region, or node, there will additionally be interacting populations of excitatory and inhibitory cells, in accord with the standard Wilson–Cowan model.

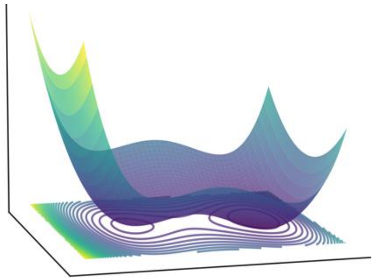


$$\frac{d[\vec{z}_1]_i(t)}{dt} = -\alpha_E[\vec{z}_1]_i(t) + (1 - [\vec{z}_1]_i(t))[\vec{f}_E(\vec{s}_E(\vec{z}_1(t), \vec{z}_2(t))]_i$$

$$\frac{d[\vec{z}_2]_i(t)}{dt} = \frac{1}{\gamma}\left(-\alpha_I[\vec{z}_2]_i(t) + (1 - [\vec{z}_2]_i(t))[\vec{f}_I(\vec{s}_I(\vec{z}_1(t), \vec{z}_2(t)))]_i\right)$$

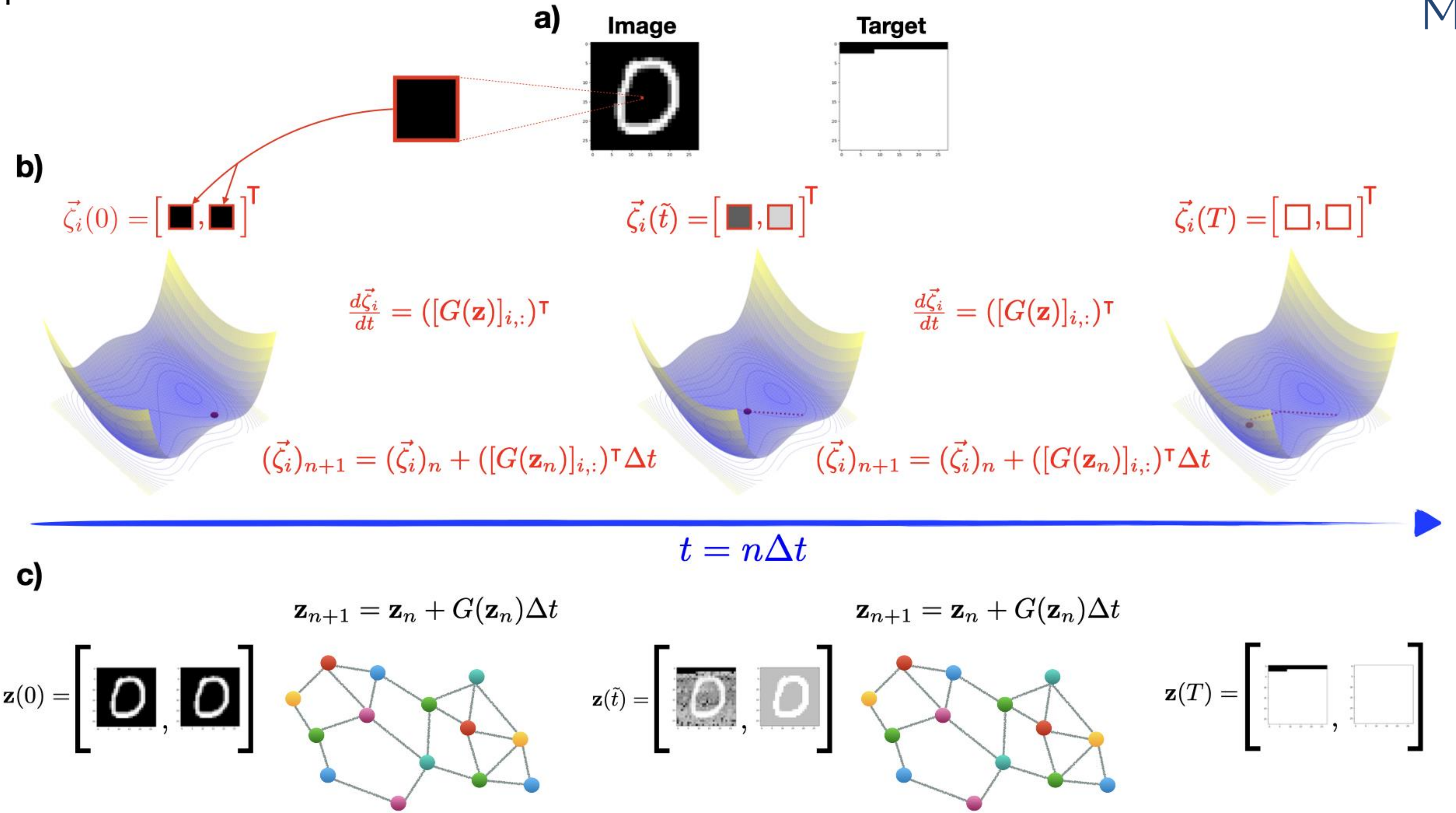
$$\vec{s}_E(\vec{z}_1(t), \vec{z}_2(t)) = \omega_{EE}\vec{z}_1(t) - \omega_{EI}\vec{z}_2(t) + h_E\vec{e} + \Gamma\mathbf{A}\vec{z}_1(t)$$

$$\vec{s}_I(\vec{z}_1(t), \vec{z}_2(t)) = \omega_{IE}\vec{z}_1(t) - \omega_{II}\vec{z}_2(t) + h_I\vec{e}$$

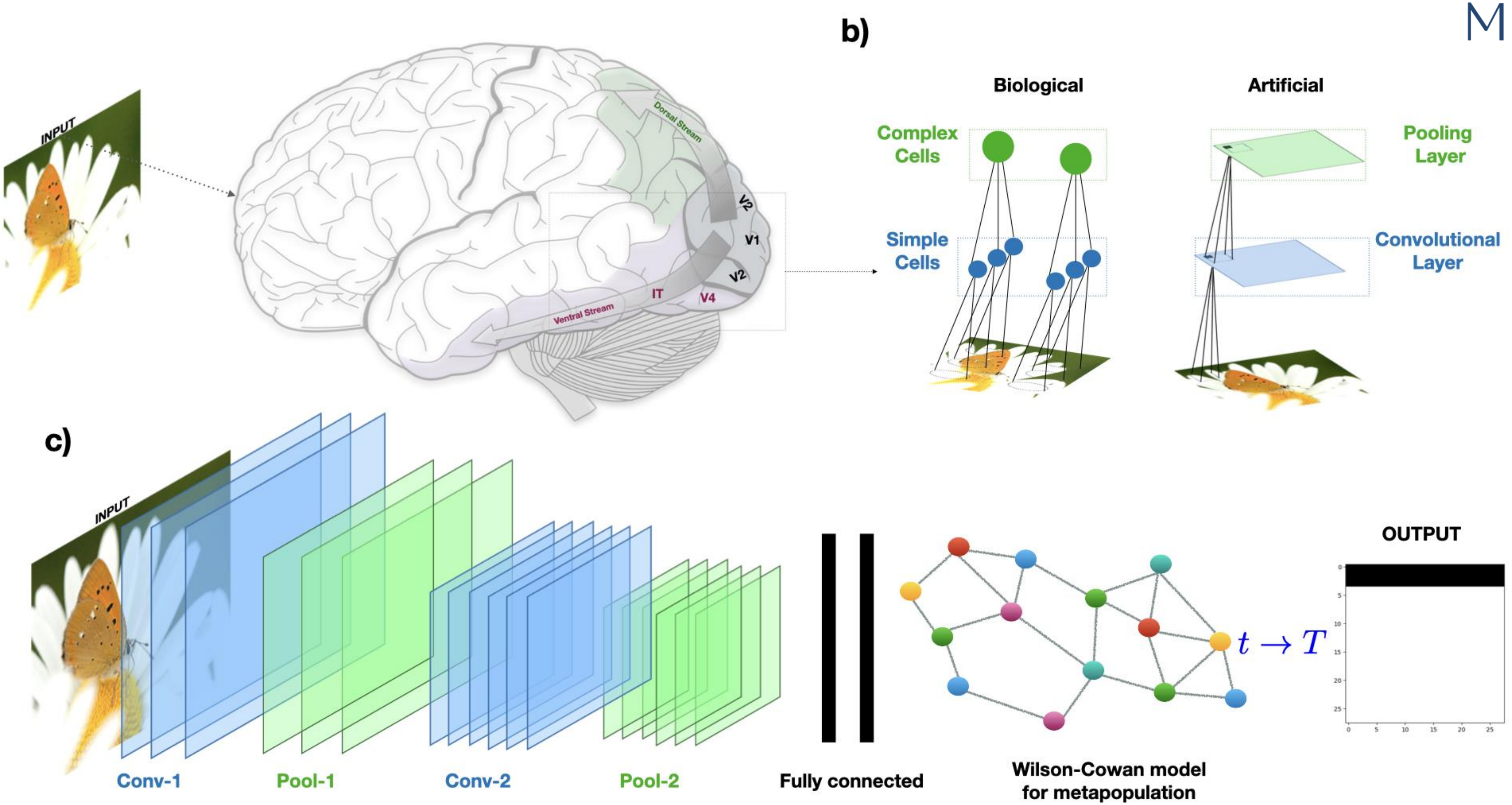


Inter-node connections are established between excitatory subpopulations only, as is true of the brain [Conti et al. J. Theo. Bio. 477 (2019); Byrne et al. J. Neurophys. 123 (2020), Gerfen et al. J. NeuroScience Research 96 (2018)].

Dynamics


a)
Image
Target
b)
$\vec{\zeta}_i(0) = [\blacksquare, \blacksquare]^\intercal$
$\vec{\zeta}_i(\tilde{t}) = [\blacksquare, \square]^\intercal$
$\vec{\zeta}_i(T) = [\square, \square]^\intercal$
$\frac{d\vec{\zeta}_i}{dt} = ([G(\mathbf{z})]_{i,:})^\intercal$
$\frac{d\vec{\zeta}_i}{dt} = ([G(\mathbf{z})]_{i,:})^\intercal$
$(\vec{\zeta}_i)_{n+1} = (\vec{\zeta}_i)_n + ([G(\mathbf{z}_n)]_{i,:})^\intercal \Delta t$
$(\vec{\zeta}_i)_{n+1} = (\vec{\zeta}_i)_n + ([G(\mathbf{z}_n)]_{i,:})^\intercal \Delta t$
$t = n\Delta t$
c)
$\mathbf{z}_{n+1} = \mathbf{z}_n + G(\mathbf{z}_n)\Delta t$
$\mathbf{z}_{n+1} = \mathbf{z}_n + G(\mathbf{z}_n)\Delta t$
$\mathbf{z}(0) =$
$\mathbf{z}(\tilde{t}) =$
$\mathbf{z}(T) =$

CNN & WC-RNN



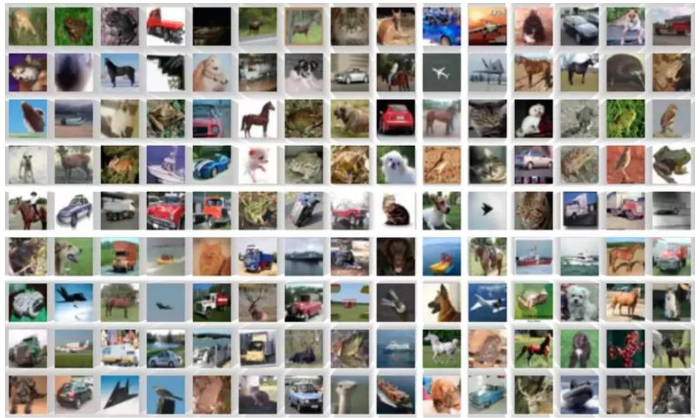
INPUT
Dorsal Stream
V2
V1
V2
V4
IT
Ventral Stream
b)
Biological
Artificial
Complex Cells
Simple Cells
Pooling Layer
Convolutional Layer
c)
INPUT
Conv-1
Pool-1
Conv-2
Pool-2
Fully connected
$t \to T$
Wilson-Cowan model for metapopulation
OUTPUT

Results

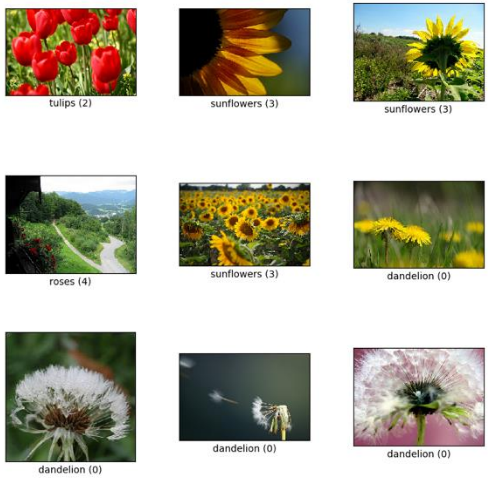
MNIST

F-MNIST

CIFAR-10



Flowers



	$\psi(\sigma_\psi)$	SOTA
MNIST	0.9931(8)	0.9987[a]
Fashion MNIST	0.9135(16)	0.9691[b]
CIFAR10	0.8659(21)	0.9950[c]
TF-FLOWERS	0.8485(37)	0.98[d]

UNIVERSITÀ DEGLI STUDI FIRENZE

MNESYS

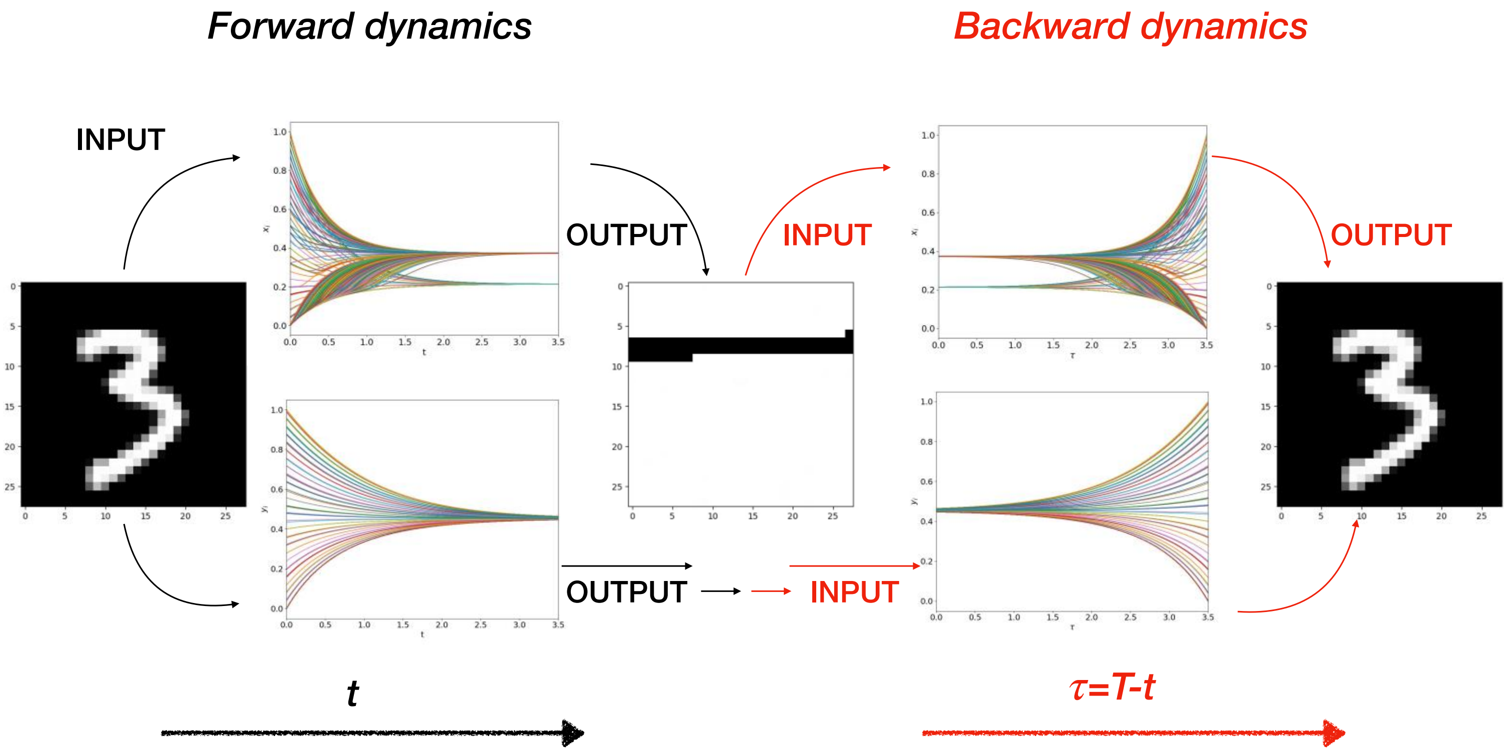
Thank You !

Results I

MNIST

Accuracy WCM for metapopulation: 98.16%

Accuracy MLP: 98.29%

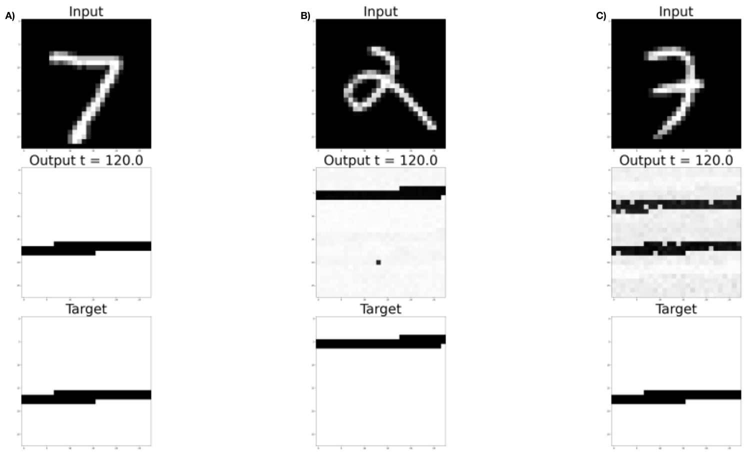


SA-NODE: Results

Accuracy

$$\mathcal{H} = \frac{1}{|\mathcal{D}_{test}|}\sum_{j=1}^{|\mathcal{D}_{test}|}\delta(m_f^{(j)}), \quad \delta(m_f^{(j)}) = \begin{cases} 1 & \text{if } m_f^{(j)} \geq 0.95 \\ 0 & \text{otherwise} \end{cases}$$

$$m_f^{(j)} = \frac{1}{N}\sum_{i=1}^{N} sign(x(T)_i^{(j)})sign(\phi_i^{(y^{(j)})})$$



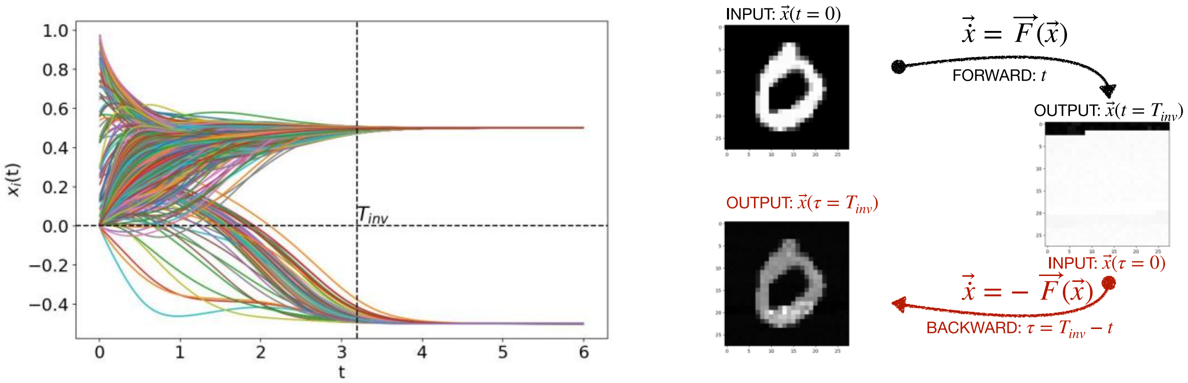
MNIST

$\mathcal{H}_{\text{SA-NODE}} = 98.06\%, \ \mathcal{H}_{MLP-RELU} = 98.23\%,$

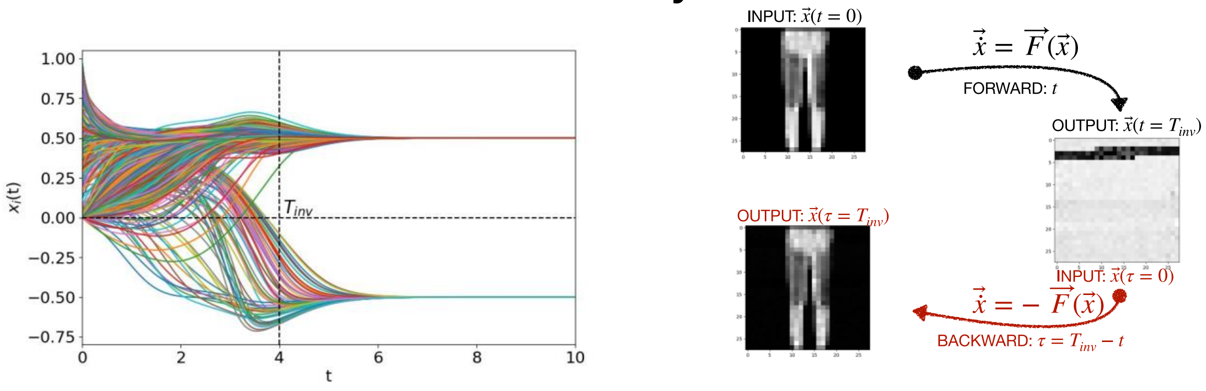
Fashion - MNIST

$\mathcal{H}_{\text{SA-NODE}} = 88.21\%, \ \mathcal{H}_{MLP-RELU} = 89.36\%,$

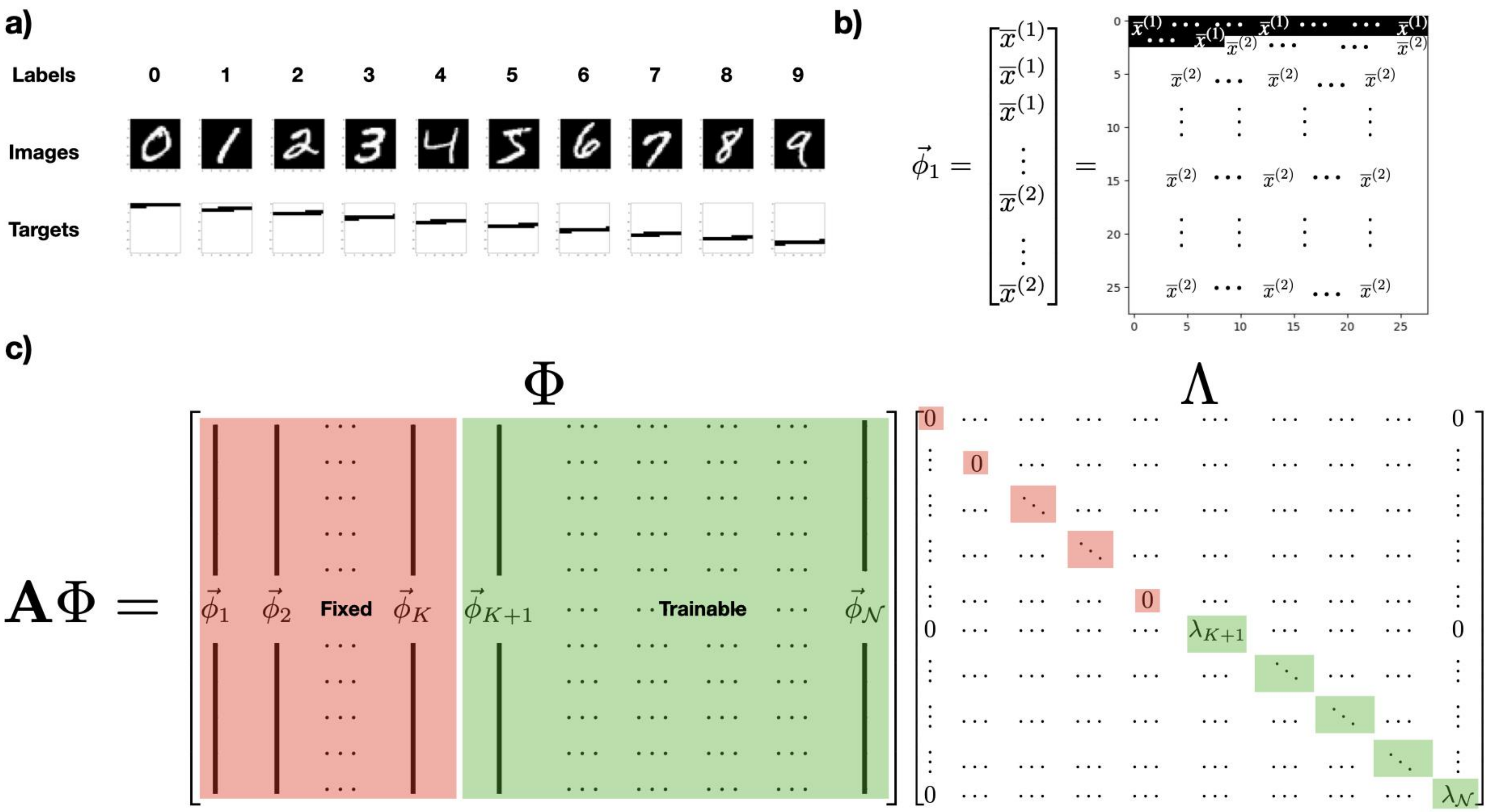
Invertibility



Invertibility



Planting attractors



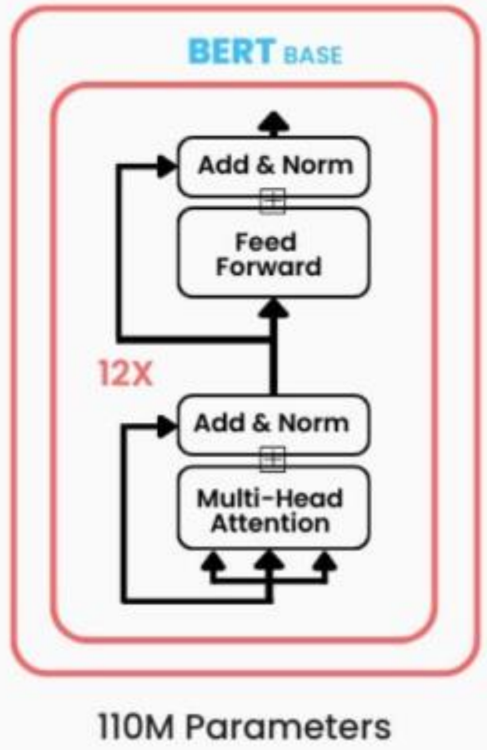
Results III

IMDB

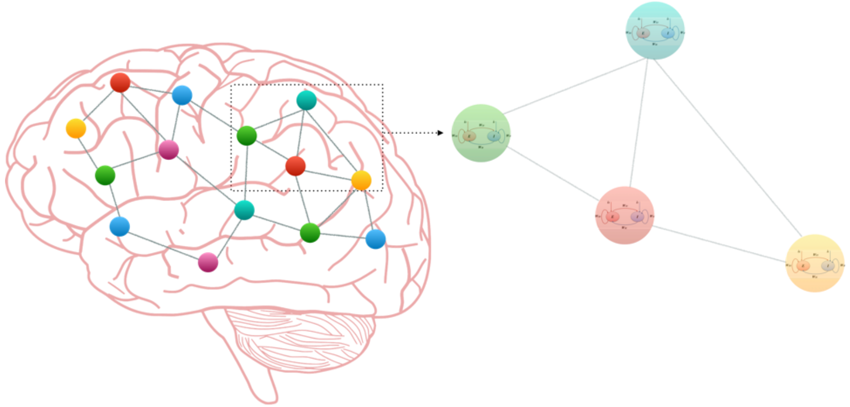
INPUT

Once again Mr. Costner has dragged out a movie for far longer than necessary. Aside from the terrific sea rescue sequences, of which there are very few I just did not care about any of the characters. Most of us have ghosts in the closet, and Costner's character are realized early on, and then forgotten until much later, by which time I did not care. The character we should really care about is a very cocky, overconfident Ashton Kutcher. The problem is he comes off as kid who thinks he's better than anyone else around him and shows no signs of a cluttered closet. His only obstacle appears to be winning over Costner. Finally when we are well past the half way point of this stinker, Costner tells us all about Kutcher's ghosts. We are told why Kutcher is driven to be the best with no prior inkling or foreshadowing. No magic here, it was all I could do to keep from turning it off an hour in.

Transformer



Target: 0



	$\psi(\sigma_\psi)$	$\psi_{BERT}(\sigma_{\psi_{BERT}})$	SOTA
IMDB	0.8746(22)	0.8830(3)	0.9668 [51]

Neural ODEs (NODE)

Residual Neural Network
t=1,…T

$\mathbf{W}$ Weight matrix

$$\vec{x}^{(t+1)} = \vec{x}^{(t)} + F(\vec{x}^{(t)}, \mathbf{W})$$

$$\downarrow$$

$$T \to \infty$$

$$\downarrow$$

NODE $\frac{\partial \vec{x}}{\partial t} = F(\vec{x}, \mathbf{W})$

$$\vec{x}(T) = \int_0^T F(\vec{x}(t), \mathbf{W})dt$$

Neural Network

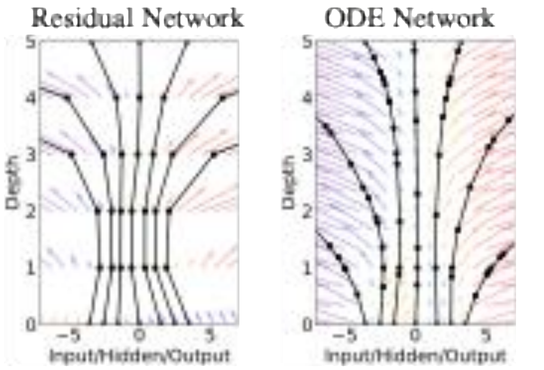


Figure 1: *Left:* A Residual network defines a discrete sequence of finite transformations. *Right:* A ODE network defines a vector field, which continuously transforms the state. *Both:* Circles represent evaluation locations.

Loss Function:

$$\mathcal{L}(\mathbf{W}, \vec{x}(T), \vec{y})$$

$\vec{y}$ identifies the vector target associated.

Training process can be done:

- Automatic differentiation
- Adjoint sensitivity method (cont. Back. Prop)