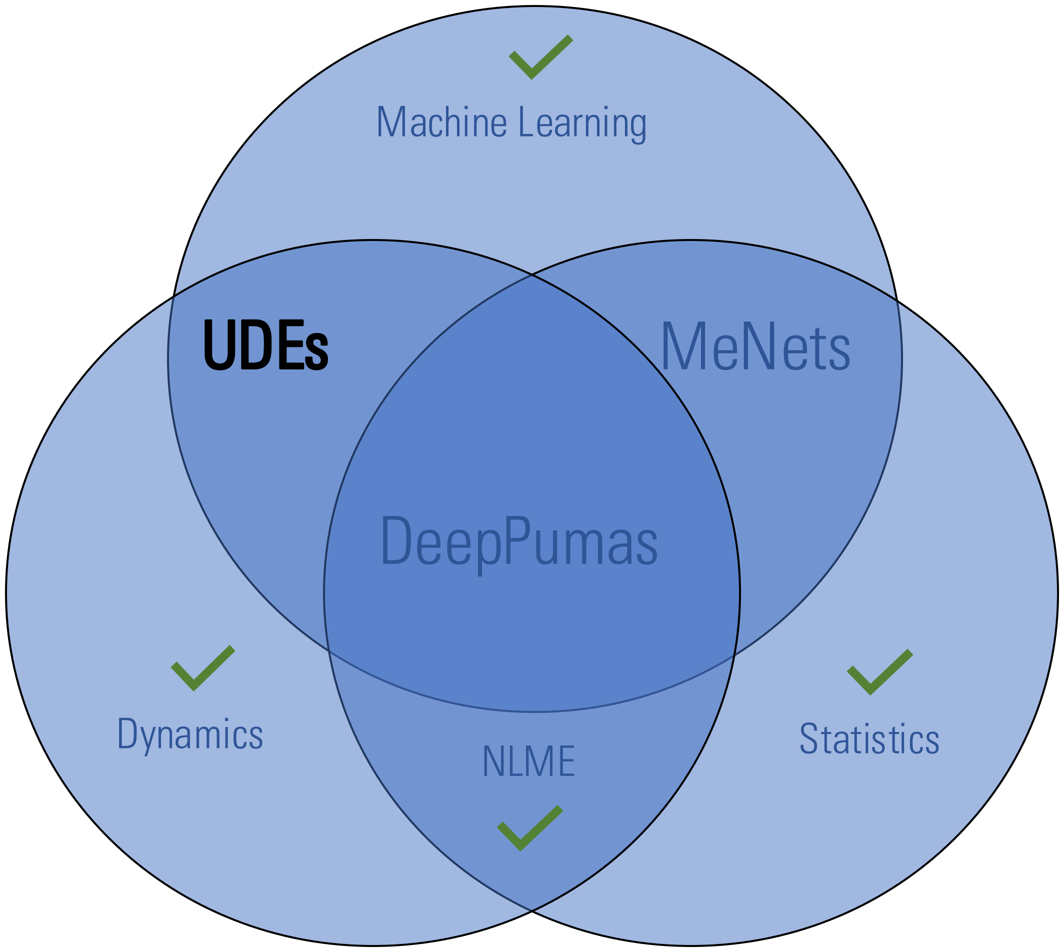
NeuralODEs and UDEs

Niklas Korsbo

What to make of embedded ML?



DeepPumas

NLME WITH DEEPPUMAS

Typical values

$\theta \in \mathbb{R}^3_+$

$\Omega \in \mathbb{R}^3_+$

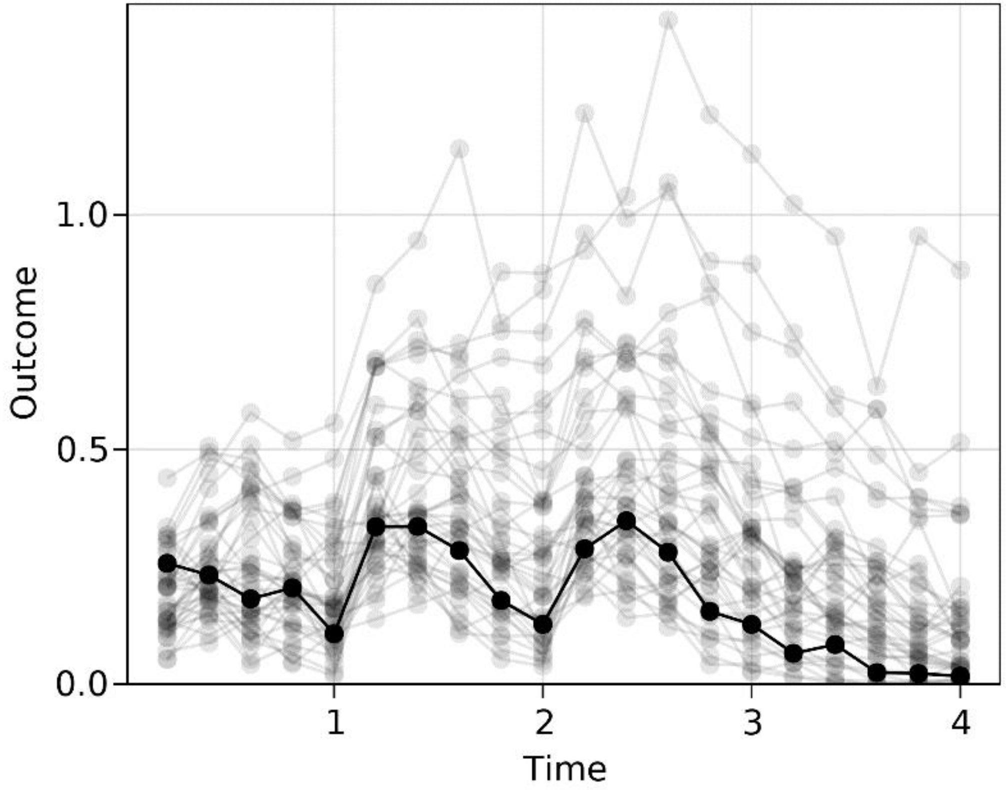
Patient data

Age

Weight

Random effects

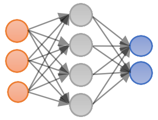
$\eta \sim \mathrm{MvNormal}\left(\Omega\right)$



Universal differential equations (UDEs)

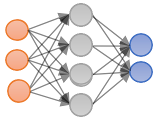
Individual parameters

$$Ka_i = \theta_1 \cdot e^{\eta_{i,1}} + c_1 \cdot Age_i \quad +$$



$$CL_i = \theta_2 \cdot e^{\eta_{i,2}}$$

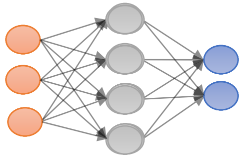
$$V_i = \theta_3 \cdot e^{\eta_{i,3}} + c_2 \cdot Weight_1^{c_3} \quad +$$



Dynamics

$$\frac{d[\mathrm{Depot}]}{dt} = -Ka[\mathrm{Depot}],$$

$$\frac{d[\mathrm{Central}]}{dt} = Ka[\mathrm{Depot}] -$$



Error model

$$Outcome \sim \mathrm{Normal}\left(Central, \sqrt{Central} \cdot \sigma\right)$$

DeepPumas

Universal Differential Equations and friends

2018 - "Neural Ordinary Differential Equations", Chen et al.

2020 - "Universal Differential Equations for Scientific Machine Learning", Rackauckas et al.

Neural ODE

$$\frac{d\boldsymbol{X}}{dt} = NN(\boldsymbol{X}(t), t)$$

Use a differential equation solvers as a scaffold for continuous time, continuous depth neural networks.

Similar to recurrent neural networks and ResNets

Universal differential equation (UDE)

$$\frac{dx}{dt} = x \cdot y - NN(x)$$
$$\frac{dy}{dt} = p - x \cdot y$$

Insert universal approximators (like NNs) to capture terms in dynamical systems.

Scientific Machine Learning (SciML)

An abstract concept of mixing scientific modeling with machine learning.

DeepPumas

Different encoded knowledge

$$
\begin{aligned}
\frac{dDepot}{dt} =& \mathrm{NN}\left(\mathrm{Depot}, \mathrm{Central}, \mathrm{R}\right)[1] \\
\frac{dCentral}{dt} =& \mathrm{NN}\left(\mathrm{Depot}, \mathrm{Central}, \mathrm{R}\right)[2] \\
\frac{dR}{dt} =& \mathrm{NN}\left(\mathrm{Depot}, \mathrm{Central}, \mathrm{R}\right)[3]
\end{aligned}
$$

- Number of states.

$$
\begin{aligned}
\frac{dDepot}{dt} =& - K_a \cdot Depot \\
\frac{dCentral}{dt} =& K_a \cdot Depot - \frac{CL}{V_c} \cdot Central \\
\frac{dR}{dt} =& \mathrm{NN}\left(\frac{Central}{V_c}, R\right)
\end{aligned}
$$

- Explicit knowledge of some terms
- Relationships (R independent of Depot!)

$$
\begin{aligned}
\frac{dDepot}{dt} =& - \mathrm{NN}_1\left(Depot\right) \\
\frac{dCentral}{dt} =& \mathrm{NN}_1\left(Depot\right) - \mathrm{NN}_2\left(Central\right) \\
\frac{dR}{dt} =& \mathrm{NN}_3\left(Central, R\right)
\end{aligned}
$$

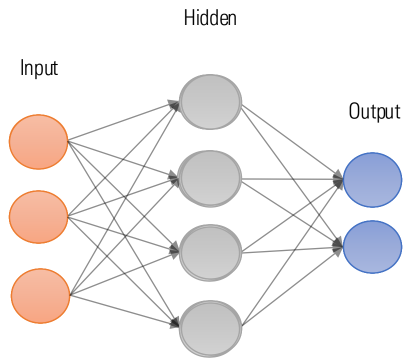
- Number of states.
- Relationships
- Dependence/independence of terms
- Conservation between Depot and Central

$$
\begin{aligned}
\frac{dDepot}{dt} =& - K_a \cdot Depot \\
\frac{dCentral}{dt} =& K_a \cdot Depot - \frac{CL}{V_c} \cdot Central \\
\frac{dR}{dt} =& k_{in} \cdot \left(1 + \mathrm{NN}\left(\frac{Central}{V_c}\right)\right) - k_{out} \cdot R
\end{aligned}
$$

- Precise position of the unknown function.
- Precise input to the unknown function.
- Lots of knowledge!

DeepPumas

UDEs – pretty simple, really



Mathematically: Just a function!

NNs are useable anywhere where you'd use a function!

- Decide where in the dynamics you have an unknown function.
- Decide what inputs this function may have.
- Fit <u>everything</u> in concert

The only hard part is building software for fitting but, with DeepPumas, that's not your problem!

DeepPumas