In the name of Allah

Physics-Informed Neural Networks:
Theory and Applications

By:

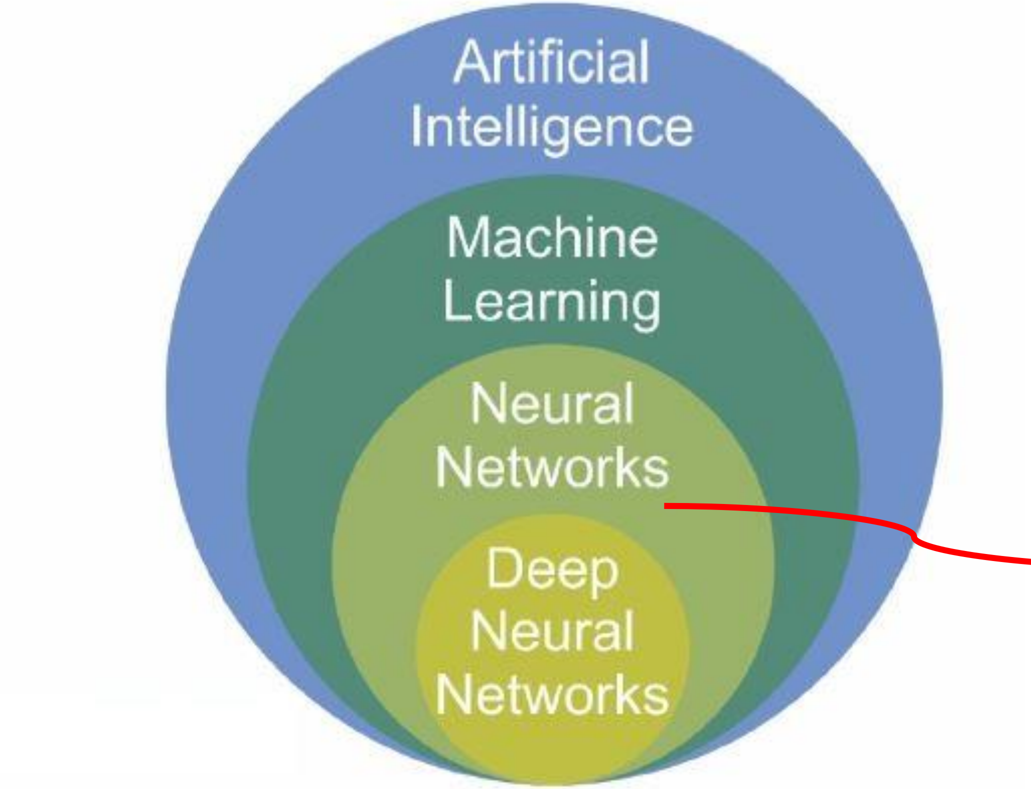
Ehsan Ghaderi

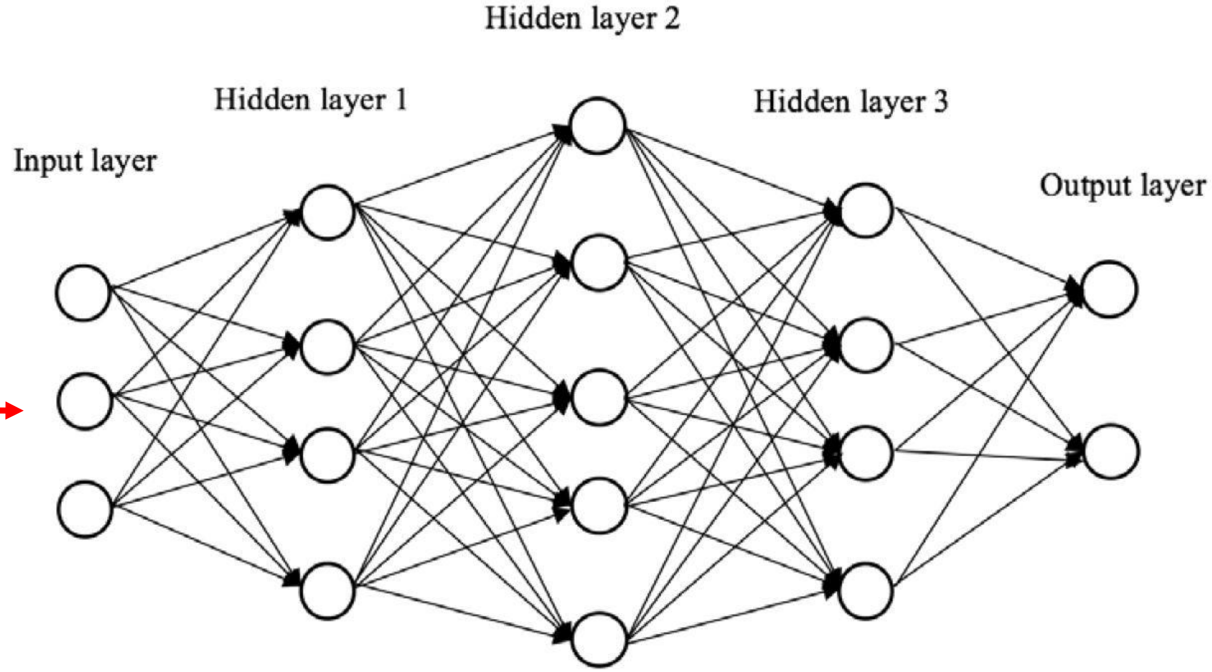
Adviser:

Dr. Bijarchi

October 12,2023

Artificial Neural Networks

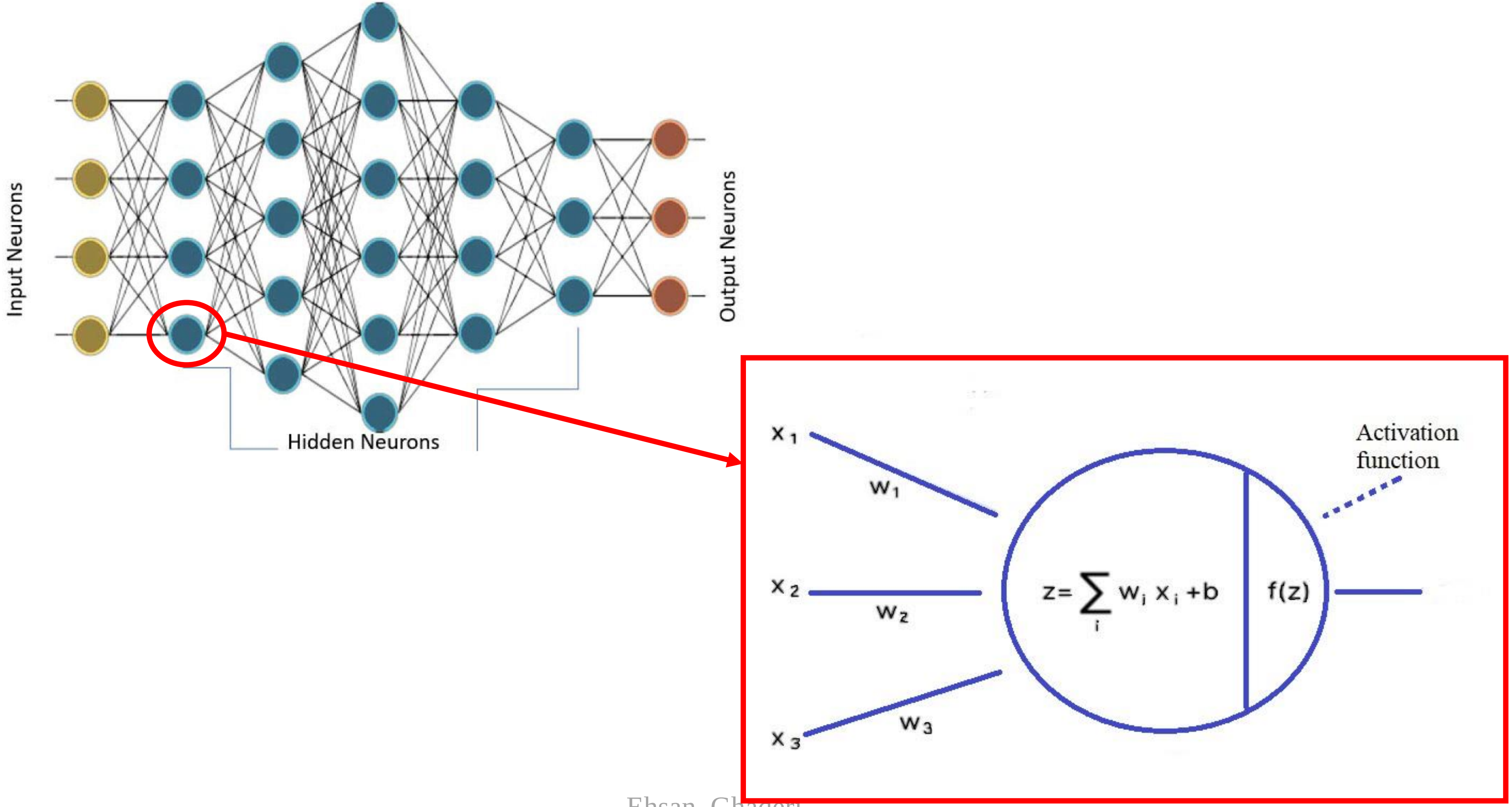




A Deep Learning Model

"F. Chollet, Deep learning with Python, 2021"

Feed-Forward Neural Networks



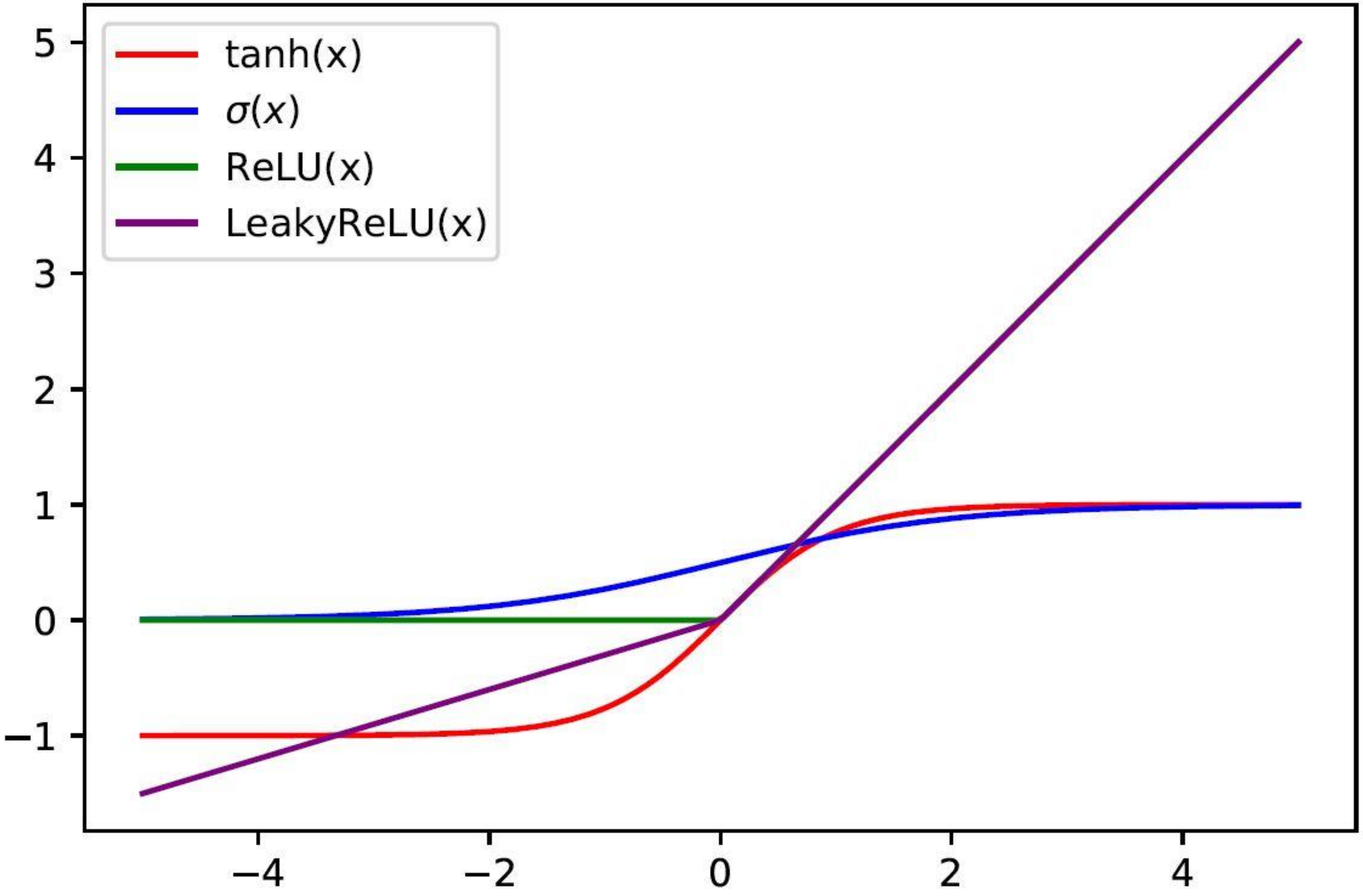
Activation Functions

$$\sigma(x) = \frac{e^x}{e^x + 1}$$

$$\tanh(x) = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

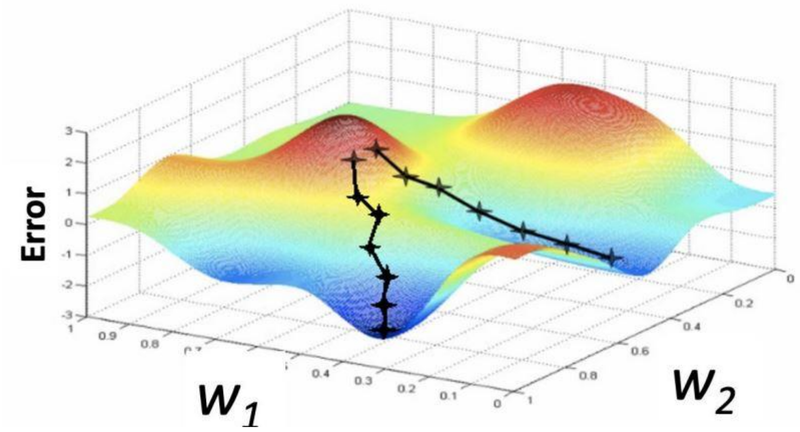
$$\mathrm{ReLU}(x) = \max(0, x)$$

$$\mathrm{LeakyReLU}(x) = \max(\alpha \cdot x, x)$$



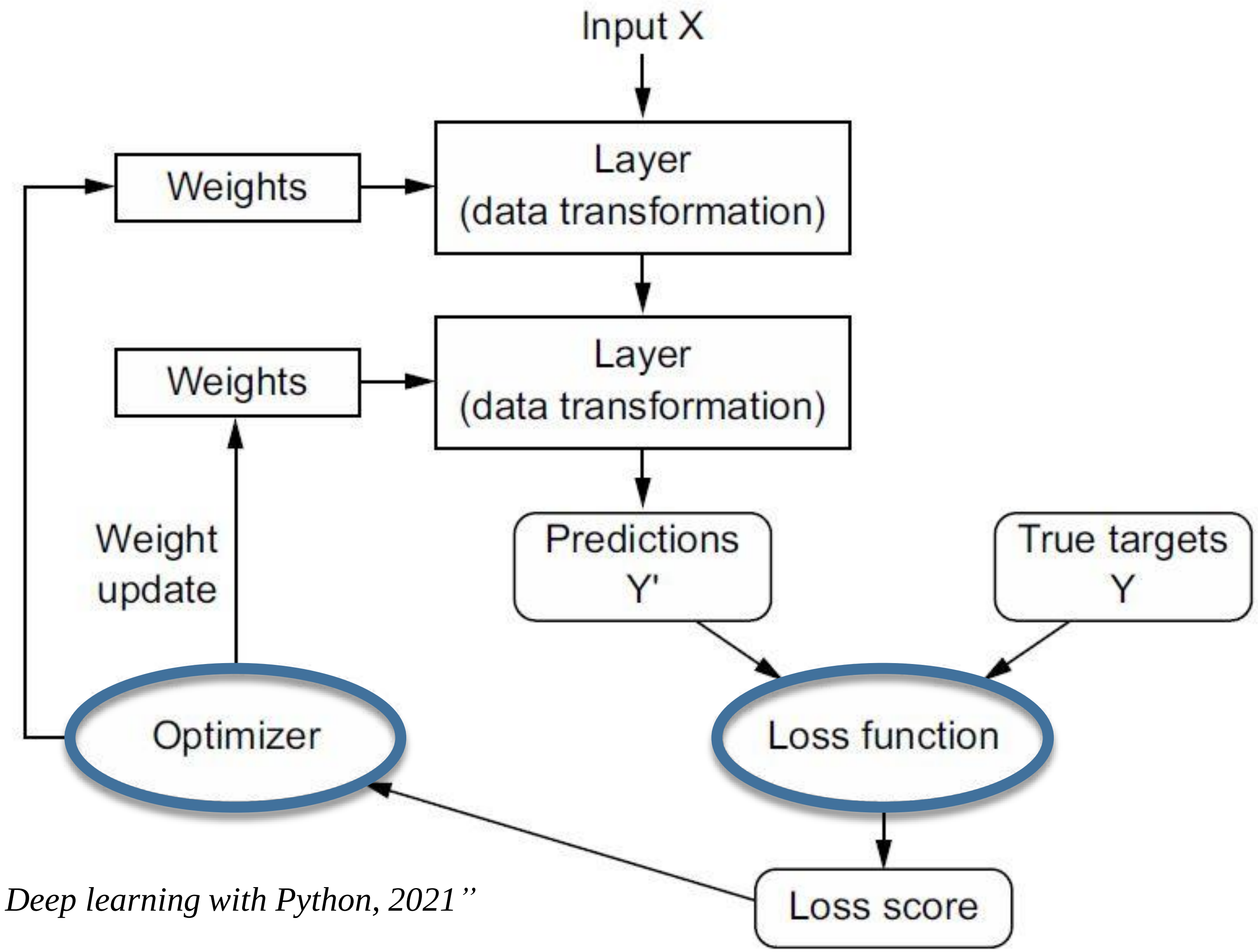
Ehsan_Ghaderi

Forward and Back Propagation



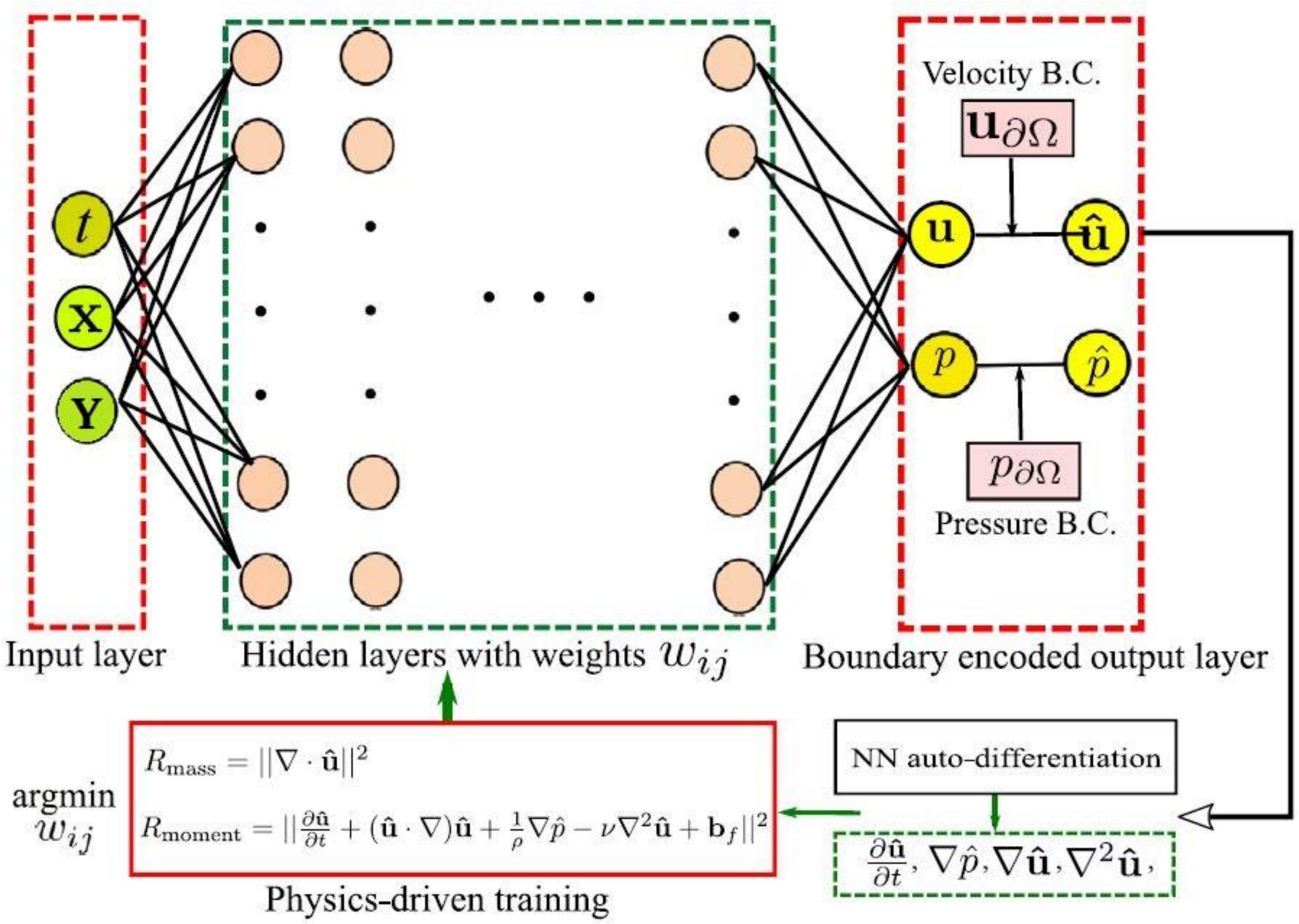
Initialize: $\mathbf{w}^{(0)} = \mathbf{w}_0$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L(\mathbf{w}^{(t)})$$



"F. Chollet, Deep learning with Python, 2021"

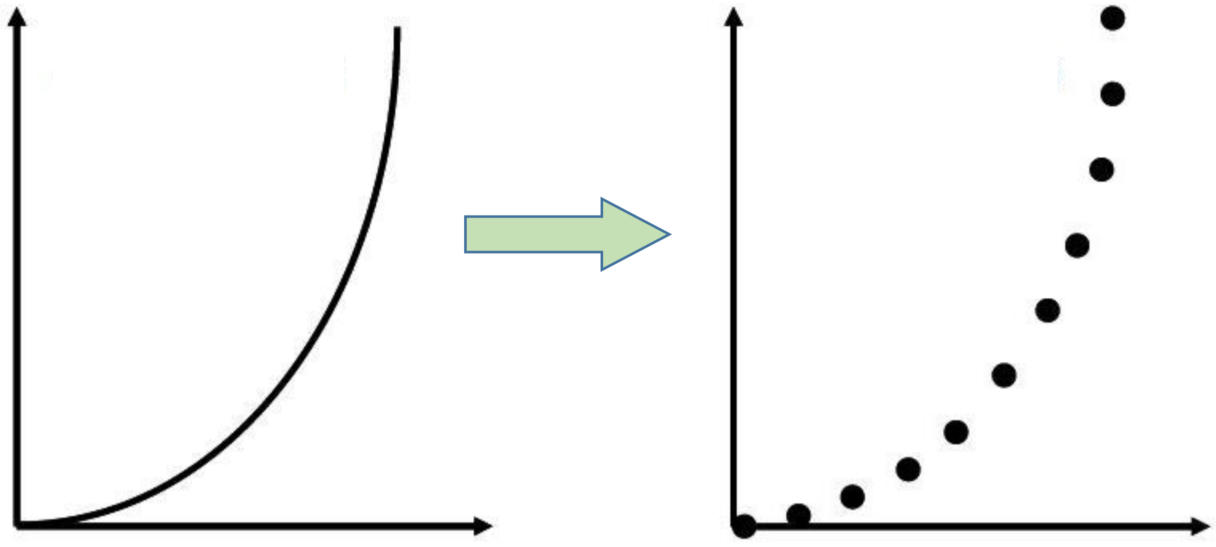
Physics-Informed Neural Networks

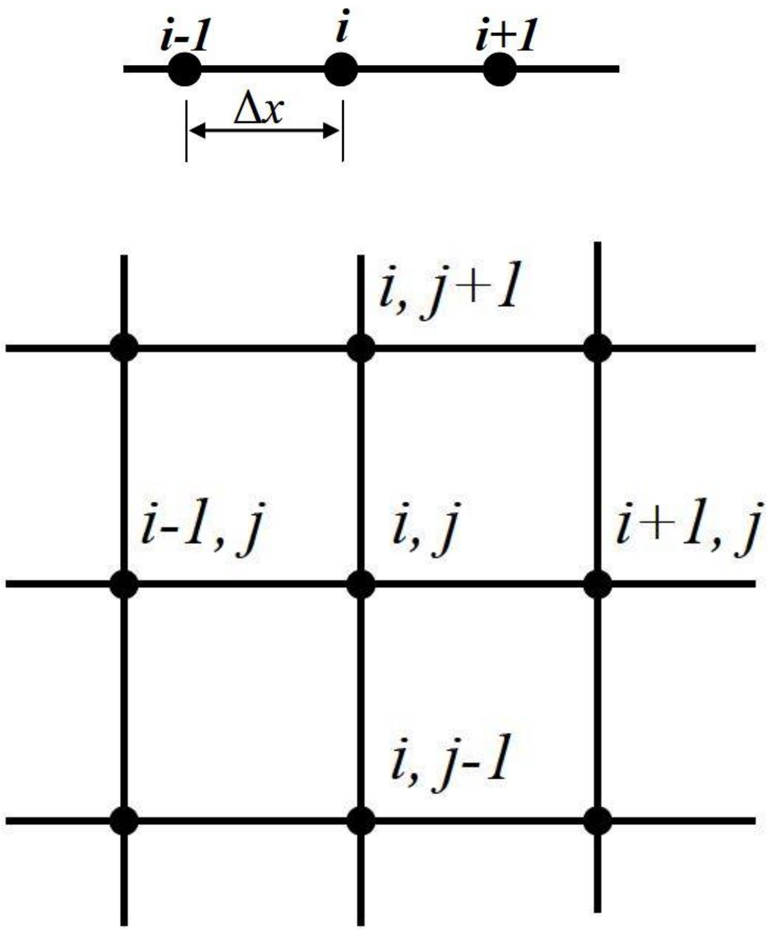


"L. Sun et al., Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data, 2020"

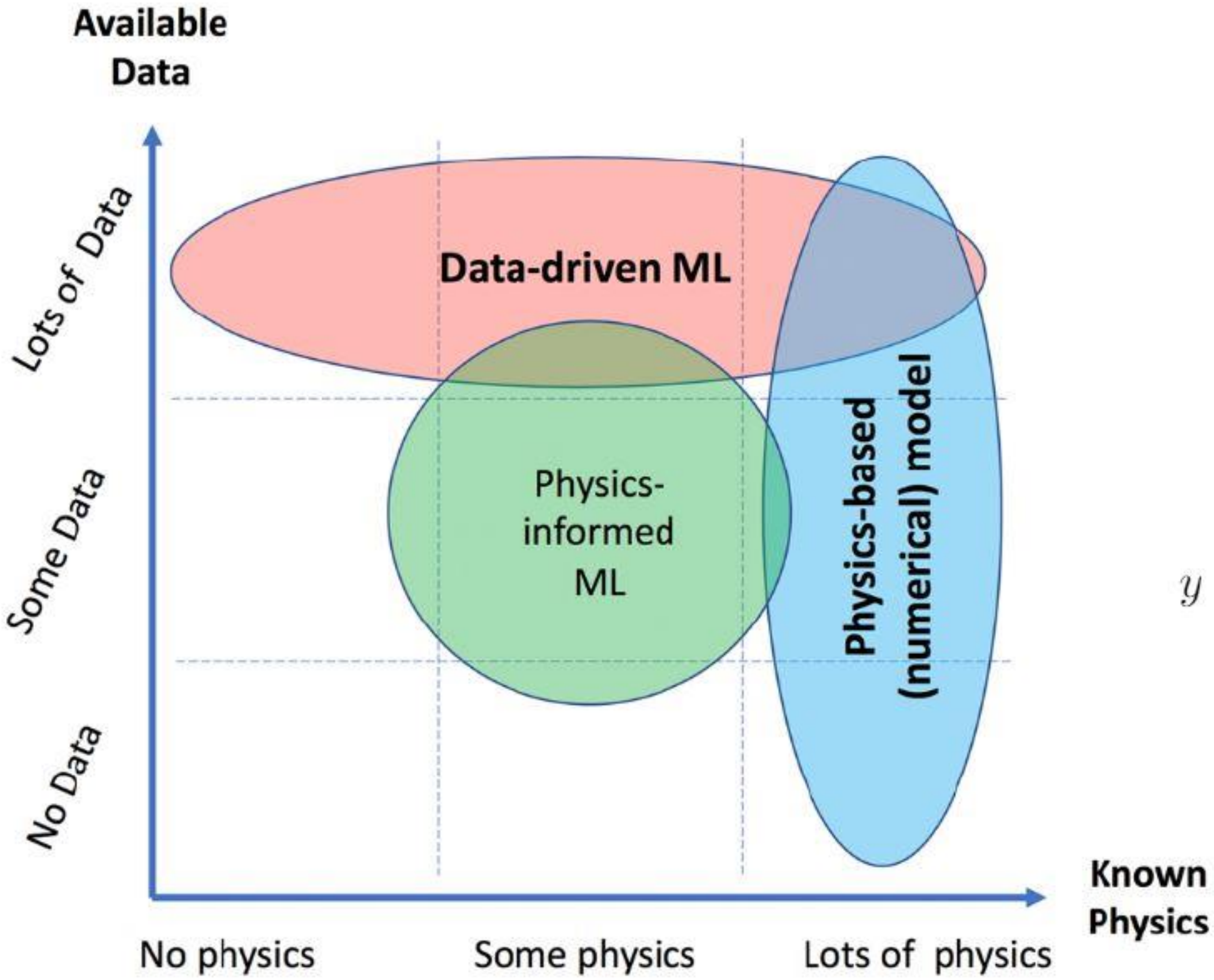
Finite-Difference

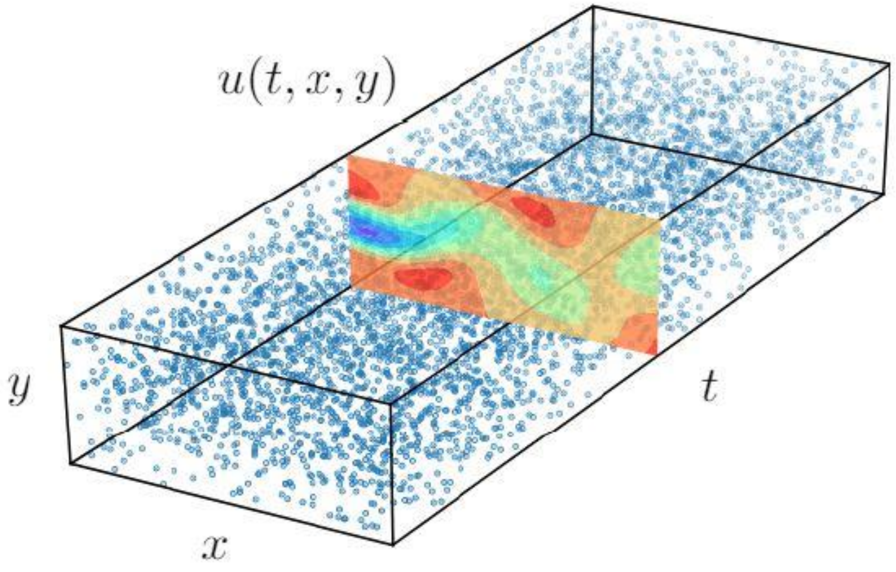
$$0 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \approx \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i+1,j} - v_{i-1,j}}{2\Delta y}$$





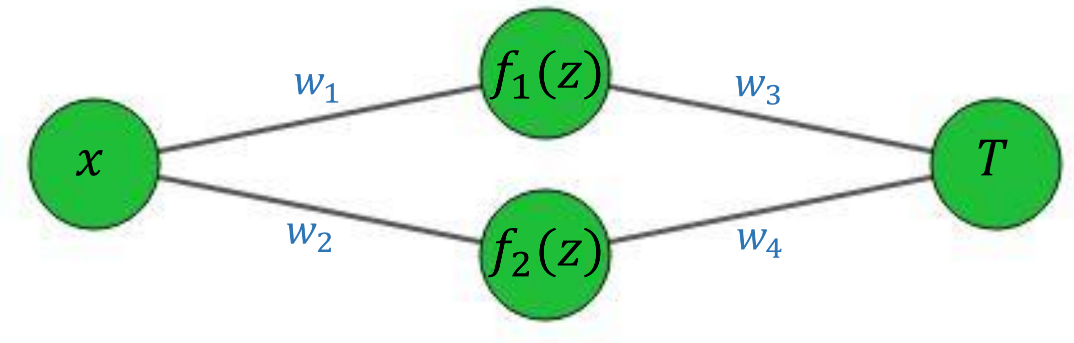
Physics-Informed Neural Networks





"A. M. Madni et al., Leveraging digital twin technology in model-based systems engineering, 2019"

1D Steady-State Heat Equation



$$f_1(z) = e^{(w_1 x)}$$

$$f_2(z) = e^{(w_2 x)}$$

$$T = w_3 e^{(w_1 x)} + w_4 e^{(w_2 x)}$$

$$\frac{dT}{dx} = w_3 w_1 e^{(w_1 x)} + w_4 w_2 e^{(w_2 x)}$$

$$\frac{d^2T}{dx^2} = w_3 {w_1}^2 e^{(w_1 x)} + w_4 {w_2}^2 e^{(w_2 x)}$$

$$\begin{cases} \left(\dfrac{d^2T}{dx^2}\right) - \dfrac{g(x)}{k} = 0 = \hat{R}_{pde} \\ g(x) = 15x - 2 \\ k = 0.5 \\ T(0) = T(1) = 0 = \hat{R}_{bc} \end{cases}$$

$$R_{pde} = w_3 {w_1}^2 e^{(w_1 x)} + w_4 {w_2}^2 e^{(w_2 x)} - (30x - 4\,)$$

$$R_{bc1} = w_3 {w_1}^2 + w_4 {w_2}^2 + 4$$

$$R_{bc2} = w_3 {w_1}^2 e^{(w_1)} + w_4 {w_2}^2 e^{(w_2)} - 26$$

$$Loss = \frac{1}{N} \sum_{i=1}^{N} \left(R_i - {R_i}^{true}\right)^2 = \frac{1}{N} \sum_{i=1}^{N} (R_i - 0)^2$$

1D Steady-State Heat Equation

$$\boxed{R = w_3 {w_1}^2 e^{(w_1 x)} + w_4 {w_2}^2 e^{(w_2 x)} - (30x - 4\,)}$$

$$\left\{\begin{array}{ll} N_1: & x = 0.2 \\ N_2: & x = 0.4 \\ N_3: & x = 0.5 \\ N_4: & x = 0.7 \\ N_5: & x = 0.8 \\ \multicolumn{2}{l}{N = N_1 + N_2 + N_3 + N_4 + N_5 = 5} \end{array}\right. \Rightarrow \left\{\begin{array}{l} R_1 = w_3 {w_1}^2 e^{(0.2 w_1)} + w_4 {w_2}^2 e^{(0.2 w_2)} - \big((30 \times 0.2) - 4\,\big) \\ R_2 = w_3 {w_1}^2 e^{(0.4 w_1)} + w_4 {w_2}^2 e^{(0.4 w_2)} - \big((30 \times 0.4) - 4\,\big) \\ R_3 = w_3 {w_1}^2 e^{(0.5 w_1)} + w_4 {w_2}^2 e^{(0.5 w_2)} - \big((30 \times 0.5) - 4\,\big) \\ R_4 = w_3 {w_1}^2 e^{(0.7 w_1)} + w_4 {w_2}^2 e^{(0.7 w_2)} - \big((30 \times 0.7) - 4\,\big) \\ R_5 = w_3 {w_1}^2 e^{(0.8 w_1)} + w_4 {w_2}^2 e^{(0.8 w_2)} - \big((30 \times 0.8) - 4\,\big) \end{array}\right.$$

$$\boxed{R_{bc1} = w_3 {w_1}^2 + w_4 {w_2}^2 + 4}$$

$$\boxed{R_{bc2} = w_3 {w_1}^2 e^{(w_1)} + w_4 {w_2}^2 e^{(w_2)} - 26}$$

1D Steady-State Heat Equation

$$w_1^{\{0\}} = 0.3$$
$$w_2^{\{0\}} = 0.8$$
$$w_3^{\{0\}} = 0.2$$
$$w_4^{\{0\}} = 0.5$$

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(R_i - 0)^2$$

$$R_1 = w_3 {w_1}^2 e^{(0.2w_1)} + w_4 {w_2}^2 e^{(0.2w_2)} - \left((30 \times 0.2) - 4\right) = -1.605$$
$$R_2 = w_3 {w_1}^2 e^{(0.4w_1)} + w_4 {w_2}^2 e^{(0.4w_2)} - \left((30 \times 0.4) - 4\right) = -7.539$$
$$R_3 = w_3 {w_1}^2 e^{(0.5w_1)} + w_4 {w_2}^2 e^{(0.5w_2)} - \left((30 \times 0.5) - 4\right) = -10.502$$
$$R_4 = w_3 {w_1}^2 e^{(0.7w_1)} + w_4 {w_2}^2 e^{(0.7w_2)} - \left((30 \times 0.7) - 4\right) = -16.418$$
$$R_5 = w_3 {w_1}^2 e^{(0.8w_1)} + w_4 {w_2}^2 e^{(0.8w_2)} - \left((30 \times 0.8) - 4\right) = -19.37$$

$$R_{bc1} = w_3 {w_1}^2 + w_4 {w_2}^2 + 4 = 4.338$$
$$R_{bc2} = w_3 {w_1}^2 e^{(w_1)} + w_4 {w_2}^2 e^{(w_2)} - 26 = -25.264$$

$${R_1}^2 = 2.576$$
$${R_2}^2 = 56.837$$
$${R_3}^2 = 110.292$$
$${R_4}^2 = 269.551$$
$${R_5}^2 = 375.197$$

$$L_{pde} = 162.891$$

$${R_{bc1}}^2 = 18.818$$
$${R_{bc2}}^2 = 638.27$$

$$L_{bc} = 657.088$$

Ehsan_Ghaderi

1D Steady-State Heat Equation

$$R_1 = w_3 {w_1}^2 e^{(0.2w_1)} + w_4 {w_2}^2 e^{(0.2w_2)} - \left((30 \times 0.2) - 4\right) = -1.605$$

$$w_j^{\{m+1\}} = w_j^{\{m\}} - \alpha \frac{\partial(Loss)}{\partial w_j}\left(w^{\{m\}}\right)$$

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(R_i - 0)^2$$

$${w_1}^{\{0\}} = 0.3$$
$${w_2}^{\{0\}} = 0.8$$
$${w_3}^{\{0\}} = 0.2$$
$${w_4}^{\{0\}} = 0.5$$

$$\left\{\begin{aligned}
\frac{\partial R_1}{\partial w_1} &= 2w_3 w_1 e^{(0.2w_1)} + 0.2 w_3 {w_1}^2 e^{(0.2w_1)} = 0.131 \\
\frac{\partial R_1}{\partial w_2} &= 2w_4 w_2 e^{(0.2w_2)} + 0.2 w_4 {w_2}^2 e^{(0.2w_2)} = 1.014 \\
\frac{\partial R_1}{\partial w_3} &= {w_1}^2 e^{(0.2w_1)} = 0.096 \\
\frac{\partial R_1}{\partial w_4} &= {w_2}^2 e^{(0.2w_2)} = 0.751
\end{aligned}\right.$$

$$\Rightarrow$$

$$R_1 \frac{\partial R_1}{\partial w_1} = -0.21$$

$$R_1 \frac{\partial R_1}{\partial w_2} = -1.627$$

$$R_1 \frac{\partial R_1}{\partial w_3} = -0.154$$

$$R_1 \frac{\partial R_1}{\partial w_4} = -1.205$$

1D Steady-State Heat Equation

$$R_2 = w_3 {w_1}^2 e^{(0.4w_1)} + w_4 {w_2}^2 e^{(0.4w_2)} - ((30 \times 0.4) - 4\,) = -7.539$$

$$w_j^{\{m+1\}} = w_j^{\{m\}} - \alpha \frac{\partial(Loss)}{\partial w_j}\left(w^{\{m\}}\right)$$

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(R_i - 0)^2$$

$${w_1}^{\{0\}} = 0.3$$

$${w_2}^{\{0\}} = 0.8$$

$${w_3}^{\{0\}} = 0.2$$

$${w_4}^{\{0\}} = 0.5$$

$$\left\{\begin{aligned}
\frac{\partial R_2}{\partial w_1} &= 2w_3 w_1 e^{(0.4w_1)} + 0.4 w_3 {w_1}^2 e^{(0.4w_1)} = 0.143 \\
\frac{\partial R_2}{\partial w_2} &= 2w_4 w_2 e^{(0.4w_2)} + 0.4 w_4 {w_2}^2 e^{(0.4w_2)} = 1.278 \\
\frac{\partial R_2}{\partial w_3} &= {w_1}^2 e^{(0.4w_1)} = 0.101 \\
\frac{\partial R_2}{\partial w_4} &= {w_2}^2 e^{(0.4w_2)} = 0.881
\end{aligned}\right.$$

$$\Rightarrow$$

$$R_2 \frac{\partial R_2}{\partial w_1} = -1.078$$

$$R_2 \frac{\partial R_2}{\partial w_2} = -9.635$$

$$R_2 \frac{\partial R_2}{\partial w_3} = -0.761$$

$$R_2 \frac{\partial R_2}{\partial w_4} = -6.642$$

1D Steady-State Heat Equation

$$R_3 = w_3 {w_1}^2 e^{(0.5w_1)} + w_4 {w_2}^2 e^{(0.5w_2)} - \left((30 \times 0.5) - 4\right) = -10.502$$

$$w_j^{\{m+1\}} = w_j^{\{m\}} - \alpha \frac{\partial(Loss)}{\partial w_j}\left(w^{\{m\}}\right)$$

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(R_i - 0)^2$$

$${w_1}^{\{0\}} = 0.3$$
$${w_2}^{\{0\}} = 0.8$$
$${w_3}^{\{0\}} = 0.2$$
$${w_4}^{\{0\}} = 0.5$$

$$\left\{\begin{aligned}
\frac{\partial R_3}{\partial w_1} &= 2w_3 w_1 e^{(0.5w_1)} + 0.5 w_3 {w_1}^2 e^{(0.5w_1)} = 0.15 \\
\frac{\partial R_3}{\partial w_2} &= 2w_4 w_2 e^{(0.5w_2)} + 0.5 w_4 {w_2}^2 e^{(0.5w_2)} = 1.432 \\
\frac{\partial R_3}{\partial w_3} &= {w_1}^2 e^{(0.5w_1)} = 0.105 \\
\frac{\partial R_3}{\partial w_4} &= {w_2}^2 e^{(0.5w_2)} = 0.955
\end{aligned}\right.$$

$$\Rightarrow$$

$$R_3 \frac{\partial R_3}{\partial w_1} = -1.575$$

$$R_3 \frac{\partial R_3}{\partial w_2} = -15.039$$

$$R_3 \frac{\partial R_3}{\partial w_3} = -1.103$$

$$R_3 \frac{\partial R_3}{\partial w_4} = -10.029$$

1D Steady-State Heat Equation

$$R_4 = w_3 {w_1}^2 e^{(0.7w_1)} + w_4 {w_2}^2 e^{(0.7w_2)} - \left((30 \times 0.7) - 4\right) = -16.418$$

$$w_j^{\{m+1\}} = w_j^{\{m\}} - \alpha \frac{\partial (Loss)}{\partial w_j}\left(w^{\{m\}}\right)$$

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(R_i - 0)^2$$

$${w_1}^{\{0\}} = 0.3$$
$${w_2}^{\{0\}} = 0.8$$
$${w_3}^{\{0\}} = 0.2$$
$${w_4}^{\{0\}} = 0.5$$

$$\begin{cases} \dfrac{\partial R_4}{\partial w_1} = 2w_3 w_1 e^{(0.7w_1)} + 0.7 w_3 {w_1}^2 e^{(0.7w_1)} = 0.164 \\[2ex] \dfrac{\partial R_4}{\partial w_2} = 2w_4 w_2 e^{(0.7w_2)} + 0.7 w_4 {w_2}^2 e^{(0.7w_2)} = 1.793 \\[2ex] \dfrac{\partial R_4}{\partial w_3} = {w_1}^2 e^{(0.7w_1)} = 0.111 \\[2ex] \dfrac{\partial R_4}{\partial w_4} = {w_2}^2 e^{(0.7w_2)} = 1.12 \end{cases}$$

$$\Rightarrow$$

$$R_4 \frac{\partial R_4}{\partial w_1} = -2.693$$

$$R_4 \frac{\partial R_4}{\partial w_2} = -29.437$$

$$R_4 \frac{\partial R_4}{\partial w_3} = -1.822$$

$$R_4 \frac{\partial R_4}{\partial w_4} = -18.388$$

1D Steady-State Heat Equation

$$R_5 = w_3 {w_1}^2 e^{(0.8w_1)} + w_4 {w_2}^2 e^{(0.8w_2)} - \left((30 \times 0.8) - 4\right) = -19.37$$

$$\boxed{{w_j}^{\{m+1\}} = {w_j}^{\{m\}} - \alpha \frac{\partial(Loss)}{\partial w_j}\left(w^{\{m\}}\right)}$$

$$\boxed{Loss = \frac{1}{N}\sum_{i=1}^{N}(R_i - 0)^2}$$

$${w_1}^{\{0\}} = 0.3$$
$${w_2}^{\{0\}} = 0.8$$
$${w_3}^{\{0\}} = 0.2$$
$${w_4}^{\{0\}} = 0.5$$

$$\begin{cases} \dfrac{\partial R_5}{\partial w_1} = 2w_3 w_1 e^{(0.8w_1)} + 0.8 w_3 {w_1}^2 e^{(0.8w_1)} = 0.171 \\ \dfrac{\partial R_5}{\partial w_2} = 2w_4 w_2 e^{(0.8w_2)} + 0.8 w_4 {w_2}^2 e^{(0.8w_2)} = 2.003 \\ \dfrac{\partial R_5}{\partial w_3} = {w_1}^2 e^{(0.8w_1)} = 0.114 \\ \dfrac{\partial R_5}{\partial w_4} = {w_2}^2 e^{(0.8w_2)} = 1.214 \end{cases}$$

$$\Rightarrow$$

$$R_5 \frac{\partial R_5}{\partial w_1} = -3.312$$

$$R_5 \frac{\partial R_5}{\partial w_2} = -38.798$$

$$R_5 \frac{\partial R_5}{\partial w_3} = -2.208$$

$$R_5 \frac{\partial R_5}{\partial w_4} = -23.515$$

Ehsan_Ghaderi

1D Steady-State Heat Equation

$$R_{bc1} = w_3 {w_1}^2 + w_4 {w_2}^2 + 4 = 4.338$$

$$w_j^{\{m+1\}} = w_j^{\{m\}} - \alpha \frac{\partial(Loss)}{\partial w_j}\left(w^{\{m\}}\right)$$

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(R_i - 0)^2$$

$${w_1}^{\{0\}} = 0.3$$
$${w_2}^{\{0\}} = 0.8$$
$${w_3}^{\{0\}} = 0.2$$
$${w_4}^{\{0\}} = 0.5$$

$$\begin{cases} \dfrac{\partial R_{bc1}}{\partial w_1} = 2w_3 w_1 = 0.12 \\ \dfrac{\partial R_{bc1}}{\partial w_2} = 2w_4 w_2 = 0.8 \\ \dfrac{\partial R_{bc1}}{\partial w_3} = {w_1}^2 = 0.09 \\ \dfrac{\partial R_{bc1}}{\partial w_4} = {w_2}^2 = 0.64 \end{cases}$$

$$\Rightarrow$$

$$R_{bc1}\frac{\partial R_{bc1}}{\partial w_1} = 0.521$$

$$R_{bc1}\frac{\partial R_{bc1}}{\partial w_2} = 3.47$$

$$R_{bc1}\frac{\partial R_{bc1}}{\partial w_3} = 0.39$$

$$R_{bc1}\frac{\partial R_{bc1}}{\partial w_4} = 2.776$$

1D Steady-State Heat Equation

$$R_{bc2} = w_3 {w_1}^2 e^{(w_1)} + w_4 {w_2}^2 e^{(w_2)} - 26 = -25.264$$

$$w_j^{\{m+1\}} = w_j^{\{m\}} - \alpha \frac{\partial(Loss)}{\partial w_j}\left(w^{\{m\}}\right)$$

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(R_i - 0)^2$$

$${w_1}^{\{0\}} = 0.3$$
$${w_2}^{\{0\}} = 0.8$$
$${w_3}^{\{0\}} = 0.2$$
$${w_4}^{\{0\}} = 0.5$$

$$\begin{cases} \dfrac{\partial R_{bc2}}{\partial w_1} = 2 w_3 w_1 e^{(w_1)} + w_3 {w_1}^2 e^{(w_1)} = 0.186 \\[2ex] \dfrac{\partial R_{bc2}}{\partial w_2} = 2 w_4 w_2 e^{(w_2)} + w_4 {w_2}^2 e^{(w_2)} = 2.493 \\[2ex] \dfrac{\partial R_{bc2}}{\partial w_3} = {w_1}^2 e^{(w_1)} = 0.121 \\[2ex] \dfrac{\partial R_{bc2}}{\partial w_4} = {w_2}^2 e^{(w_2)} = 1.424 \end{cases}$$

$$\Rightarrow$$

$$R_{bc2} \frac{\partial R_{bc2}}{\partial w_1} = -4.699$$

$$R_{bc2} \frac{\partial R_{bc2}}{\partial w_2} = -62.983$$

$$R_{bc2} \frac{\partial R_{bc2}}{\partial w_3} = -3.057$$

$$R_{bc2} \frac{\partial R_{bc2}}{\partial w_4} = -35.976$$

1D Steady-State Heat Equation

$$w_1^{\{0\}} = 0.3$$
$$w_2^{\{0\}} = 0.8$$
$$w_3^{\{0\}} = 0.2$$
$$w_4^{\{0\}} = 0.5$$

$$\alpha = 0.01$$

$$w_j^{\{m+1\}} = w_j^{\{m\}} - \alpha \frac{\partial(Loss)}{\partial w_j}\left(w^{\{m\}}\right)$$

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(R_i - 0)^2$$

$$\begin{cases} w_1^{\{1\}} = w_1^{\{0\}} - \dfrac{0.02}{5}\left(R_1\dfrac{\partial R_1}{\partial w_1} + R_2\dfrac{\partial R_2}{\partial w_1} + R_3\dfrac{\partial R_3}{\partial w_1} + R_4\dfrac{\partial R_4}{\partial w_1} + R_5\dfrac{\partial R_5}{\partial w_1}\right) - 0.02\left(R_{bc1}\dfrac{\partial R_{bc1}}{\partial w_1} + R_{bc2}\dfrac{\partial R_{bc2}}{\partial w_1}\right) = 0.419 \\ w_2^{\{1\}} = w_2^{\{0\}} - \dfrac{0.02}{5}\left(R_1\dfrac{\partial R_1}{\partial w_2} + R_2\dfrac{\partial R_2}{\partial w_2} + R_3\dfrac{\partial R_3}{\partial w_2} + R_4\dfrac{\partial R_4}{\partial w_2} + R_5\dfrac{\partial R_5}{\partial w_2}\right) - 0.02\left(R_{bc1}\dfrac{\partial R_{bc1}}{\partial w_2} + R_{bc2}\dfrac{\partial R_{bc2}}{\partial w_2}\right) = 2.368 \\ w_3^{\{1\}} = w_3^{\{0\}} - \dfrac{0.02}{5}\left(R_1\dfrac{\partial R_1}{\partial w_3} + R_2\dfrac{\partial R_2}{\partial w_3} + R_3\dfrac{\partial R_3}{\partial w_3} + R_4\dfrac{\partial R_4}{\partial w_3} + R_5\dfrac{\partial R_5}{\partial w_3}\right) - 0.02\left(R_{bc1}\dfrac{\partial R_{bc1}}{\partial w_3} + R_{bc2}\dfrac{\partial R_{bc2}}{\partial w_3}\right) = 0.278 \\ w_4^{\{1\}} = w_4^{\{0\}} - \dfrac{0.02}{5}\left(R_1\dfrac{\partial R_1}{\partial w_4} + R_2\dfrac{\partial R_2}{\partial w_4} + R_3\dfrac{\partial R_3}{\partial w_4} + R_4\dfrac{\partial R_4}{\partial w_4} + R_5\dfrac{\partial R_5}{\partial w_4}\right) - 0.02\left(R_{bc1}\dfrac{\partial R_{bc1}}{\partial w_4} + R_{bc2}\dfrac{\partial R_{bc2}}{\partial w_4}\right) = 1.403 \end{cases}$$

$$w_1^{\{1\}} = 0.419$$
$$w_2^{\{1\}} = 2.368$$
$$w_3^{\{1\}} = 0.278$$
$$w_4^{\{1\}} = 1.403$$

1D Steady-State Heat Equation

$$w_1^{\{1\}} = 0.419$$
$$w_2^{\{1\}} = 2.368$$
$$w_3^{\{1\}} = 0.278$$
$$w_4^{\{1\}} = 1.403$$

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(R_i - 0)^2$$

$$R_1 = w_3 w_1{}^2 e^{(0.2w_1)} + w_4 w_2{}^2 e^{(0.2w_2)} - \big((30 \times 0.2) - 4\big) = 10.686$$

$$R_2 = w_3 w_1{}^2 e^{(0.4w_1)} + w_4 w_2{}^2 e^{(0.4w_2)} - \big((30 \times 0.4) - 4\big) = 12.343$$

$$R_3 = w_3 w_1{}^2 e^{(0.5w_1)} + w_4 w_2{}^2 e^{(0.5w_2)} - \big((30 \times 0.5) - 4\big) = 14.766$$

$$R_4 = w_3 w_1{}^2 e^{(0.7w_1)} + w_4 w_2{}^2 e^{(0.7w_2)} - \big((30 \times 0.7) - 4\big) = 24.342$$

$$R_5 = w_3 w_1{}^2 e^{(0.8w_1)} + w_4 w_2{}^2 e^{(0.8w_2)} - \big((30 \times 0.8) - 4\big) = 32.374$$

$$R_{bc1} = w_3 w_1{}^2 + w_4 w_2{}^2 + 4 = 11.916$$

$$R_{bc2} = w_3 w_1{}^2 e^{(w_1)} + w_4 w_2{}^2 e^{(w_2)} - 26 = 58.065$$

$$R_1{}^2 = 114.191$$
$$R_2{}^2 = 152.35$$
$$R_3{}^2 = 218.035$$
$$R_4{}^2 = 592.533$$
$$R_5{}^2 = 1048.076$$

$$\Rightarrow L_{pde} = 2125.185$$

$$R_{bc1}{}^2 = 141.991$$
$$R_{bc2}{}^2 = 3371.544$$

$$\Rightarrow L_{bc} = 3513.535$$

1D Steady-State Heat Equation

```
# importing necessary libraries:
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import time

# We set seeds initially. By doing it so, we can reproduce same results.
tf.random.set_seed(123)

# 100 equidistant points in the domain are created:
x = tf.linspace(0.0, 1.0, 100)

# boundary conditions T(0)=T(1)=0 and \kappa are introduced:
bcs_x = [0.0, 1.0]
print("bcs_x : ", bcs_x)
bcs_T = [0.0, 0.0]
bcs_x_tensor = tf.convert_to_tensor(bcs_x)
print("bcs_x_tensor : ", bcs_x_tensor)
bcs_T_tensor = tf.convert_to_tensor(bcs_T)
kappa = 0.5

# Number of iterations:
N = 1000

# ADAM optimizer with learning rate of 0.01:
optim = tf.keras.optimizers.Adam(learning_rate=0.01)
```

1D Steady-State Heat Equation

```
#The exact solution of the problem:
solution = lambda x: -5 * x**3 + 2 * x**2 + 3 * x

# Function for creating the model:
def buildModel(num_hidden_layers, num_neurons_per_layer):
    tf.keras.backend.set_floatx("float32")
    # Initialize a feedforward neural network:
    model = tf.keras.Sequential()
    # Input is one dimensional ( one spatial dimension):
    model.add(tf.keras.Input(1))
    # Append hidden layers:
    for _ in range(num_hidden_layers):
        model.add(
        tf.keras.layers.Dense(
        num_neurons_per_layer,
        activation=tf.keras.activations.get("tanh"),
        kernel_initializer="glorot_normal",
        )
        )
    # Output is one-dimensional:
    model.add(tf.keras.layers.Dense(1))

    return model

# determine the model size (3 hidden layers with 32 neurons each):
model = buildModel(2, 20)

print(model.summary())
```

1D Steady-State Heat Equation

```
# Boundary loss function:
# @tf.function
def boundary_loss(bcs_x_tensor, bcs_T_tensor):
    predicted_bcs = model(bcs_x_tensor)
    mse_bcs = tf.reduce_mean(tf.square(predicted_bcs - bcs_T_tensor))
    return mse_bcs

# the first derivative of the prediction
def get_first_deriv(x):
    with tf.GradientTape() as tape:
        tape.watch(x)
        T = model(x)
    T_x = tape.gradient(T, x)
    return T_x

# the second derivative of the prediction
def second_deriv(x):
    with tf.GradientTape() as tape:
        tape.watch(x)
        T_x = get_first_deriv(x)
    T_xx = tape.gradient(T_x, x)
    return T_xx

# Source term divided by \kappa:
source_func = lambda x: (15 * x - 2) / kappa
# def source_func(x): return (15 * x - 2) / kappa
```

1D Steady-State Heat Equation

```
# Function for physics loss:
def physics_loss(x):
    predicted_Txx = second_deriv(x)
    mse_phys = tf.reduce_mean(tf.square(predicted_Txx + source_func(x)))
    return mse_phys

# Overall loss function:
def loss_func(x, bcs_x_tensor, bcs_T_tensor):
    bcs_loss = boundary_loss(bcs_x_tensor, bcs_T_tensor)
    phys_loss = physics_loss(x)
    loss = bcs_loss + phys_loss
    return loss

# taking gradients of the loss function:
def get_grad():
    with tf.GradientTape() as tape:
    # This tape is for derivatives with
    # respect to trainable variables
        tape.watch(model.trainable_variables)
        Loss = loss_func(x, bcs_x_tensor, bcs_T_tensor)
    g = tape.gradient(Loss, model.trainable_variables)
    return Loss, g

# optimizing and updating the weights of the model by using gradients
def train_step():
    # Compute current loss and gradient w.r.t. parameters
    loss, grad_theta = get_grad()
    # Perform gradient descent step
    # Update the weights of the model.
    optim.apply_gradients(zip(grad_theta, model.trainable_variables))
    return loss
```

1D Steady-State Heat Equation

```
start = time.time()
# Training loop
for i in range(N + 1):
    loss = train_step()
    # printing loss amount in each 100 epoch
    if i % 100 == 0:
        print("Epoch {:03d}: loss = {:10.8e}".format(i, loss))

end = time.time()
computation_time = {}
computation_time["pinn"] = end - start
print(f"\ncomputation time: {end-start:.3f}\n")


plt.plot(x, solution(x)[:, None], label = "Exact Solution", color = "b", linestyle = "-" ) #color='darkorange'
plt.plot(x, model(x), label = "Predicted Solution", color = "r", linestyle = "--" ) #color='navy'
plt.xlabel("x ", fontsize = 12)
plt.ylabel("T(x)", fontsize = 12)
plt.legend(fontsize = 10, loc='best')
# plt.title("1D Heat Transfer", fontsize = 11)
# plt.xlim(xmin = 0, xmax = 1.10) #or plt.xlim([0.0, 1.1])
# plt.ylim(ymin = 0)
# plt.grid()
plt.show()
```

https://github.com/EhsanGh94

Inverse Heat Equation