

# Using Machine Learning Models for Computational Fluid Dynamics

Benjamin “Trey” Brown  
(join work with E. Iacob and M. Ilie)



Georgia Southern University  
Department of Mathematical Sciences

April 18, 2019



- ① Background
  - Motivation
  - Neural Networks
  - Mathematical Model
  - Fluid Diffusion Equation
  - Contributions
- ② Our work
  - Neural Networks: universal approximators
  - Finite Differences
  - Results
- ③ Conclusions and future work
- ④ References

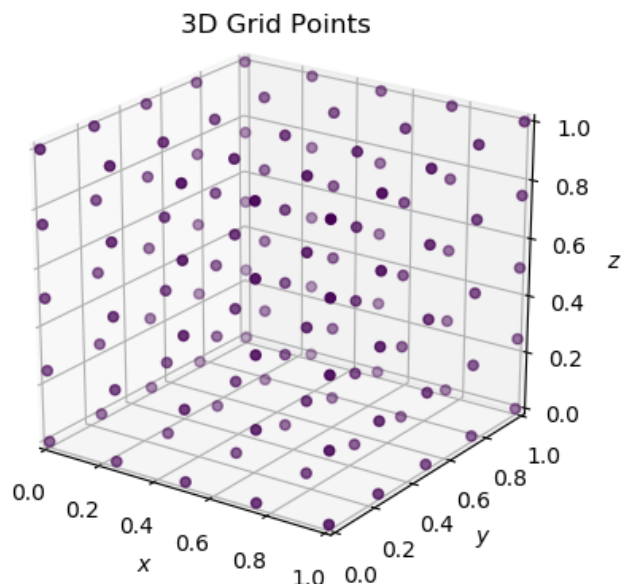


- 1 Background
  - Motivation
  - Neural Networks
  - Mathematical Model
  - Fluid Diffusion Equation
  - Contributions
- 2 Our work
  - Neural Networks: universal approximators
  - Finite Differences
  - Results
- 3 Conclusions and future work
- 4 References

# Environment That is Being Approximated

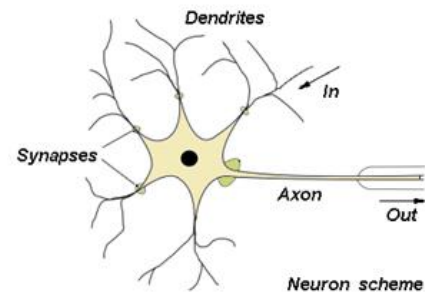


These grid points are set up in a perfect sequence. We chose how many points to have along each axis.





- How does brain work?
- Dendrites Collect information.
- Cell body processes the information.
- The processed information passed through synapses.
- Axons receive the information and conveys to the dendrites of the next nerve cell.





- Early implementation of the idea:

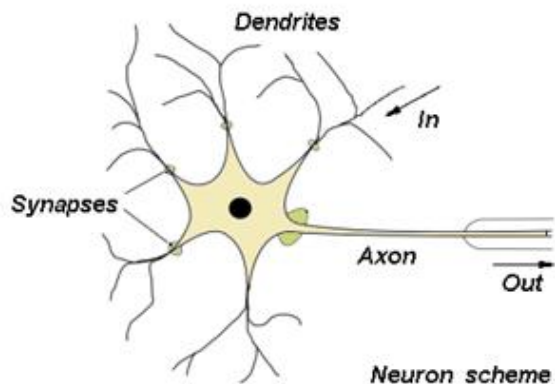


Figure 1: Human nerve cell

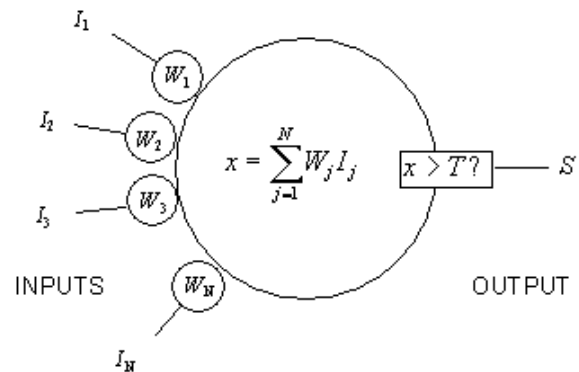
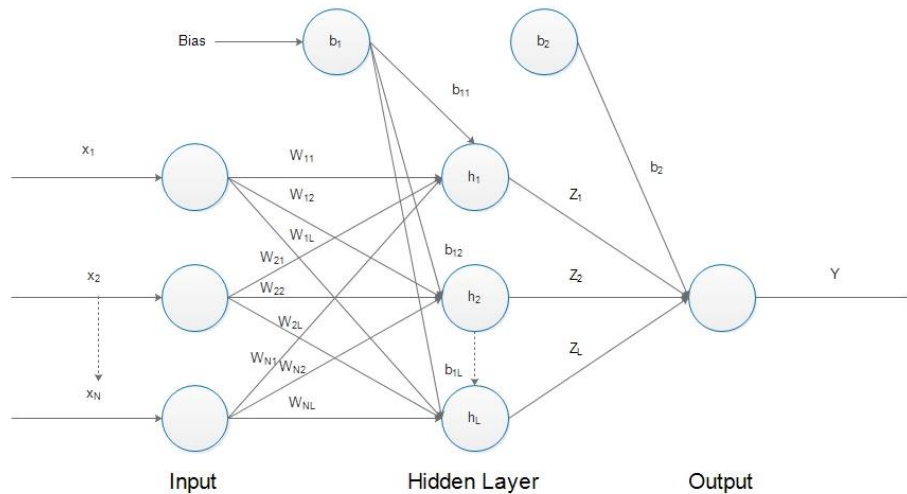


Figure 2: Neural Network

- Dendrites can be compared to input units, cell body to nodes, synapses to activation function and axons to the output units.
- In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper on how neurons might work.

# Generic ANN, one hidden layer





- Mathematical Model

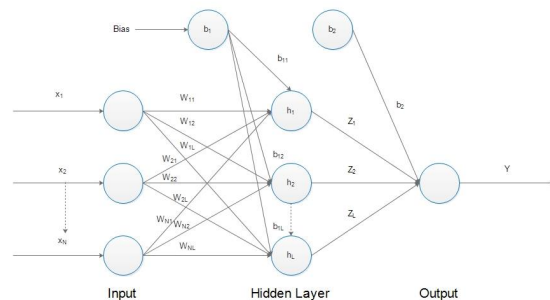
$$y : \mathbb{R}^N \rightarrow (0, 1)$$

$$y(\mathbf{x}; \mathbf{w}, \mathbf{b}) =$$

$$\sigma \left( \sum_{j=1}^L z_j \sigma \left( \sum_{i=1}^N w_{ij} x_i + b_{1j} \right) + b_2 \right)$$

where  $\sigma()$  is the sigmoid function:

$$\sigma : \mathbb{R} \rightarrow \mathbb{R}, \quad \sigma(x) = \frac{1}{1 + e^{-x}}$$







$$\dot{E}_{in} + \dot{E}_g - \dot{E}_{out} = \dot{E}_s$$

Our conditions:

$$\begin{cases} \dot{E}_g = \dot{q} dx dy dz = 0 \\ \dot{E}_s = \rho C_p \frac{\partial T}{\partial t} dx dy dz = 0 \end{cases}$$

- At equilibrium, when diffusion stops (temperature does not change over time), the distribution of temperature in a space  $D \subset \mathbb{R}^3$  is given by  $u : D \rightarrow \mathbb{R}$ , the solution of the Laplace equation:

$$\Delta u(x, y, z) = 0 \tag{1}$$

- The Laplace equation (1) with boundary conditions may have an analytical solution, but, in practice, numerical solutions can be always computed.



- 1 Background
  - Motivation
  - Neural Networks
  - Mathematical Model
  - Fluid Diffusion Equation
  - Contributions
- 2 Our work
  - Neural Networks: universal approximators
  - Finite Differences
  - Results
- 3 Conclusions and future work
- 4 References



## Theorem (Cybenko, 1989)

*Let  $\sigma$  be the sigmoid function. The finite sums of the form:*

$$G(\mathbf{x}) = \sum_{i=1}^N \alpha_i \sigma(\mathbf{y}_i^T \mathbf{x} + b_i)$$

*are dense in  $C([0, 1]^n)$ . That is, for any  $f \in C([0, 1]^n)$  and  $\varepsilon > 0$ , there is a sum  $G(\mathbf{x})$  of the above form such that:*

$$|G(\mathbf{x}) - f(\mathbf{x})| < \varepsilon \quad \forall \mathbf{x} \in [0, 1]^n$$



- A popular choice is the Finite Differences (FD) method, which produces a numerical solution by iteratively solving a linear system of equations. For instance, the FD method using second order approximations produces a numerical solution by solving the system:

$$u_{i,j,k}^{k+1} = (u_{i-1,j,k}^k + u_{i+1,j,k}^k + u_{i,j-1,k}^k + u_{i,j+1,k}^k + u_{i,j,k-1}^k + u_{i,j,k+1}^k)/6 \quad (2)$$

on the domain  $D$  typically divided in an equally spaced grid of points  $(x_i, y_j, z_k)$ .

- The numerical solution produced by the FD method is stable and quite efficient.

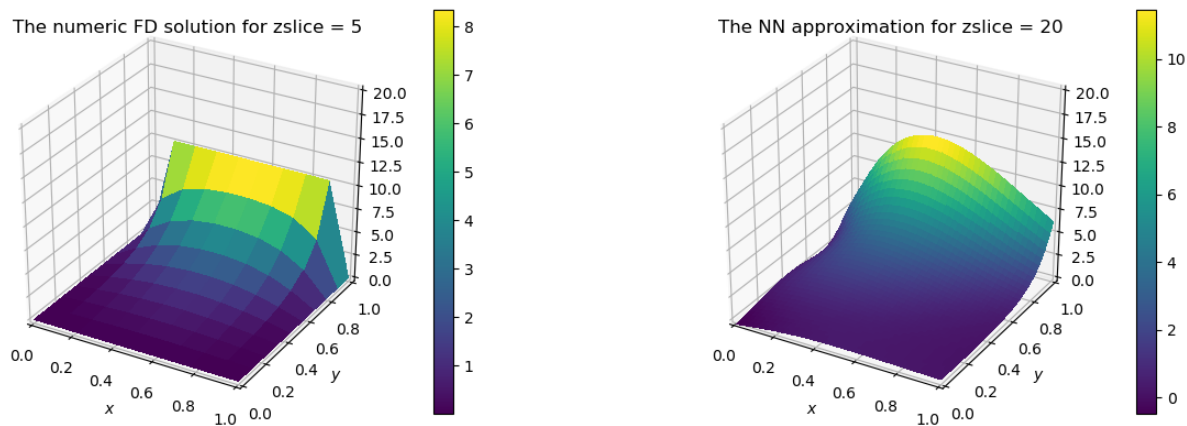


Figure 3: The FD method solution (left) and NN solution (right)

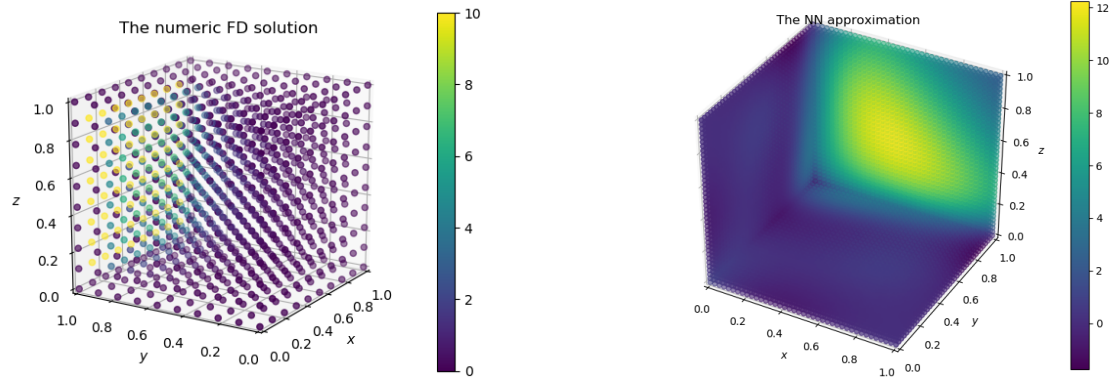


Figure 4: The FD method complete 3D solution (left) and the NN approximation on the whole domain, computed at  $40 \times 40 \times 40$  grid points (right)



- 1 Background
  - Motivation
  - Neural Networks
  - Mathematical Model
  - Fluid Diffusion Equation
  - Contributions
- 2 Our work
  - Neural Networks: universal approximators
  - Finite Differences
  - Results
- 3 Conclusions and future work
- 4 References



- We interpolate the numerical solutions from Finite Differences method and produce analytical solutions using an Artificial Neural Network approximation model.
- Next: Navier-Stokes Equation

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} = 0$$
$$\rho \left( \frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} \right) = -\nabla p + \mu \nabla^2 \vec{v} + \vec{f}$$





- 1 Background
  - Motivation
  - Neural Networks
  - Mathematical Model
  - Fluid Diffusion Equation
  - Contributions
- 2 Our work
  - Neural Networks: universal approximators
  - Finite Differences
  - Results
- 3 Conclusions and future work
- 4 References



- McCP** W. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics, Vol. 5, 1943, 115-133.
- Ros** Rosenblatt, F. (1962). Principles of Neurodynamics, Spartan Books, New York.
- RHW** Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning internal representations by backpropagating errors. Nature, 323:533–536.
- Cyb** Cybenko, G. (1989) “Approximations by superpositions of sigmoidal functions”, Mathematics of Control, Signals, and Systems, 2(4), 303–314.