Shallow Neural Networks

EN5730 Machine Learning for Communications

Samiru Gayan

Department of Electronic and Telecommunication Engineering
University of Moratuwa

2024

Neural Network Example

- Shallow neural networks are functions $\mathbf{y} = f[\mathbf{x}, \phi]$ with parameters ϕ that map multivariate inputs $\mathbf{x}$ to multivariate outputs $\mathbf{y}$.

- Example: Consider a network $f[x, \phi]$ that maps scalar input x to a scalar output y
- Ten parameters

$$\boldsymbol{\phi} = \{\phi_0, \phi_1, \phi_2, \phi_3, \theta_{10}, \theta_{11}, \theta_{20}, \theta_{21}, \theta_{30}, \theta_{31}\}$$

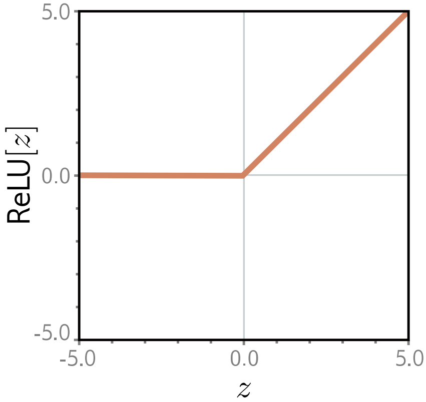
$$\begin{aligned} y &= f[x, \phi] \\ &= \phi_0 + \phi_1 a[\theta_{10} + \theta_{11} x] + \phi_2 a[\theta_{20} + \theta_{21} x] + \phi_3 a[\theta_{30} + \theta_{31} x] \end{aligned}$$

- Three linear functions of the input data: $\theta_{10} + \theta_{11} x$, $\theta_{20} + \theta_{21} x$, $\theta_{30} + \theta_{31} x$
- Pass the results though an activation functions $a[\cdot]$
- Weight the three resulting activations with ϕ_1, ϕ_2, and ϕ_3, sum them, and add an offset ϕ_0.

Neural Network Example

- There are many activation functions, but the most common choice is the rectified linear unit or ReLU:

$$a[z] = \mathrm{ReLU}[z] = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases} \tag{1}$$



Neural Network Example

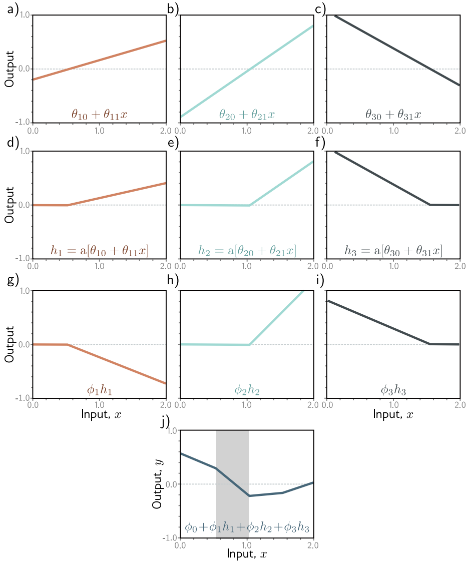
- Let's define:

$$
\begin{aligned}
h_1 &= a[\theta_{10} + \theta_{11}x] \\
h_2 &= a[\theta_{20} + \theta_{21}x] \\
h_3 &= a[\theta_{30} + \theta_{31}x]
\end{aligned}
$$

- We refer h_1, h_2, and h_3 as hidden units.

$$
y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3
$$

Neural Network Example



Neural Network Example

- When a unit is clipped, we refer to it as inactive, and when it is not clipped, we refer to it as active.
- For example, the shaded region receives contributions from h_1 and h_3 (which are active) but not from h_2 (which is inactive).
- The slope of each linear region is determined by (i) the original slopes $\theta_{\bullet 1}$ of the active inputs for this region and (ii) the weights $\phi_{\bullet}$ that were subsequently applied.
- For example, the slope in the shaded region is $\theta_{11}\phi_1 + \theta_{31}\phi_3$.
- Each hidden unit contributes one joint to the function, so with three hidden units, there can be four linear regions.

Neural Network Example

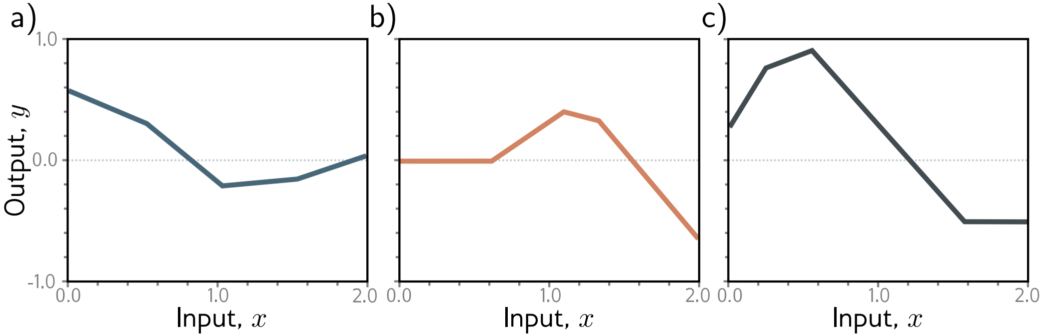
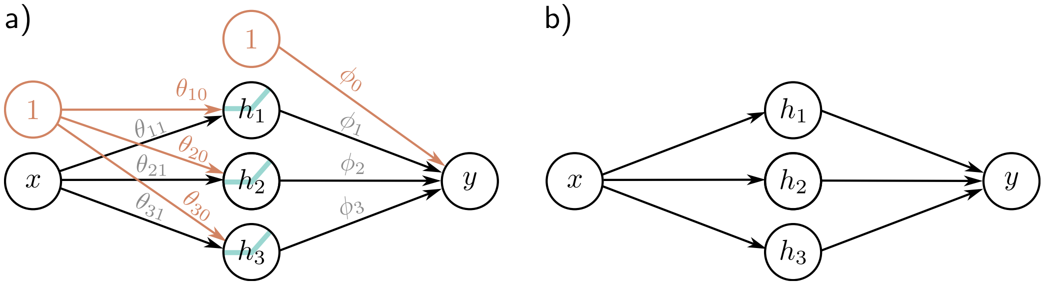


Figure: Functions for three different choices of the ten parameters ϕ.

Depicting Neural Networks



Universal Approximation Theorem

- Let's now generalize this with D hidden units where the d-th hidden unit is:

$$h_d = a[\theta_{d0} + \theta_{d1}x],$$

and these are combined linearly to create the output:

$$y = \phi_0 + \sum_{d=1}^{D} \phi_d h_d.$$

- The number of hidden units in a shallow network is a measure of the network capacity.
- With ReLU activation functions, the output of a network with D hidden units has at most D joints and piecewise linear function with at most $D + 1$ linear regions.

Universal Approximation Theorem

- The universal approximation theorem proves that for any continuous function, there exists a shallow network that can approximate this function to any specified precision.

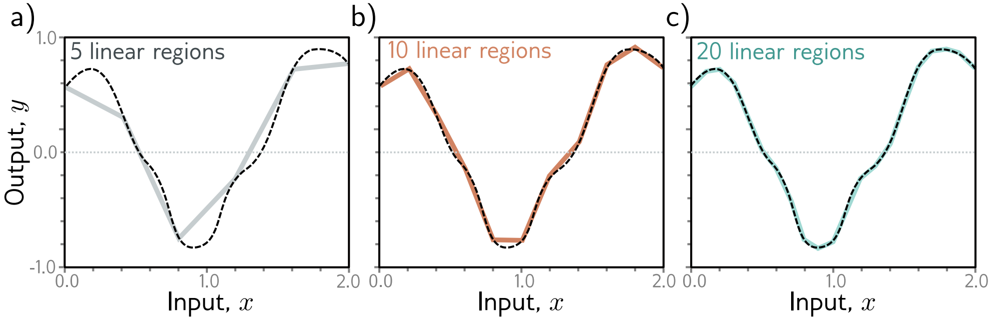


Figure: Approximation of a 1D function.

Multivariate Inputs and Outputs

- The universal approximation theorem also holds for the more general case where the network maps multivariate inputs

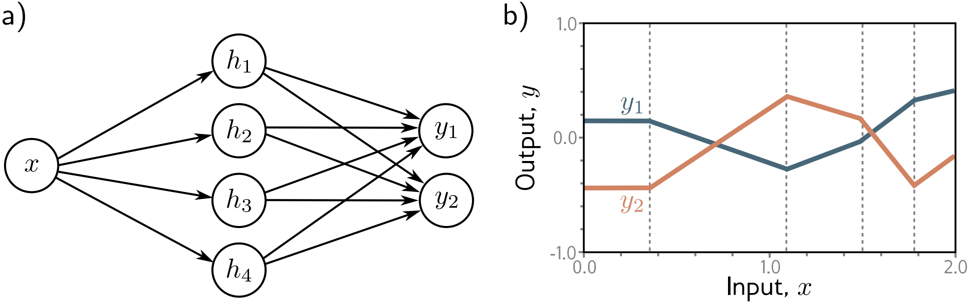
$$\mathbf{x} = [x_1, x_2, \cdots, x_{D_i}]^T$$

to multivariate output predictions

$$\mathbf{y} = [y_1, y_2, \cdots, y_{D_o}]^T.$$

Visualizing Multivariate Outputs

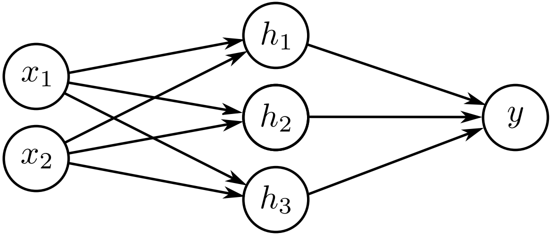
- Example: scalar input x, four hidden units h_1, h_2, h_3, h_4, 2D multivariate output $\mathbf{y} = [y_1, y_2]^T$



$$y_1 = \phi_{10} + \phi_{11} h_1 + \phi_{12} h_2 + \phi_{13} h_3 + phi_{14} h_4,$$
$$y_2 = \phi_{20} + \phi_{21} h_1 + \phi_{22} h_2 + \phi_{23} h_3 + phi_{24} h_4,$$

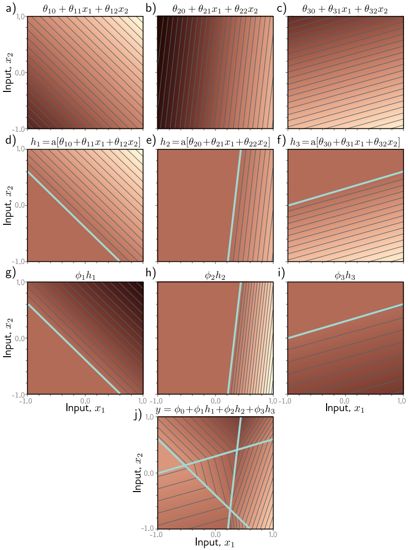
Visualizing Multivariate Inputs

- Example: multivariate input $\mathbf{x} = [x_1, x_2]^T$, four hidden units h_1, h_2, h_3, scalar output y



$$
\begin{aligned}
h_1 &= a[\theta_{10} + \theta_{11}x_1 + \theta_{12}x_2] \\
h_2 &= a[\theta_{20} + \theta_{21}x_1 + \theta_{22}x_2] \\
h_3 &= a[\theta_{30} + \theta_{31}x_1 + \theta_{32}x_2] \\
y &= \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3
\end{aligned}
$$

Visualizing Multivariate Inputs

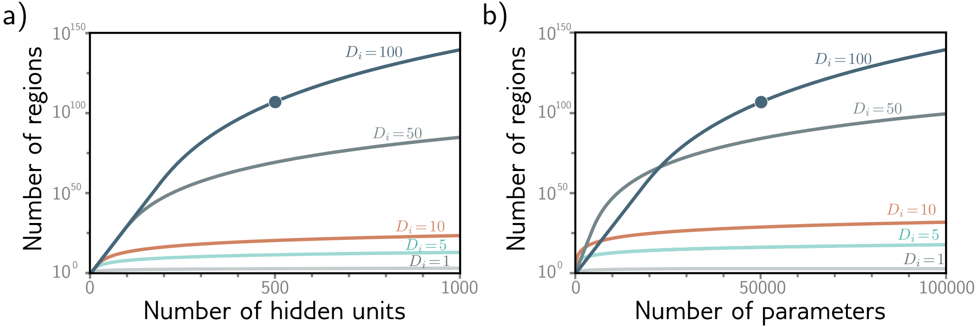


Visualizing Multivariate Inputs

- The activation function clips the negative values of these planes to zero.
- The clipped planes are then recombined in a second linear function to create a continuous piecewise linear surface consisting of convex polygonal regions.
- When there are more than two inputs to the model, it becomes difficult to visualize.
- However, the interpretation is similar.
- The output will be a continuous piecewise linear function of the input, where the linear regions are now convex polytopes in the multidimensional input space.

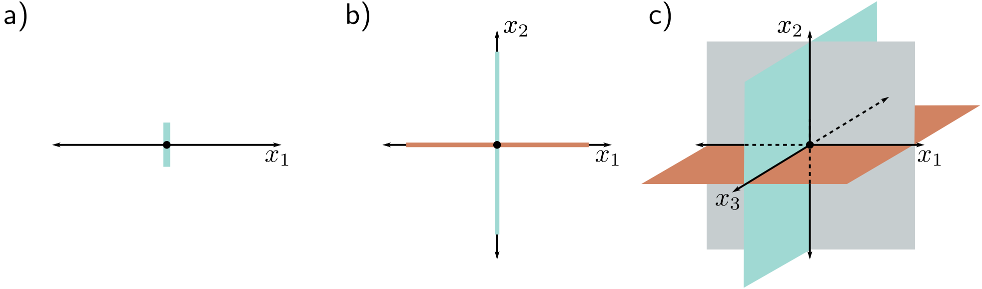
Visualizing Multivariate Inputs

- As the input dimensions grow, the number of linear regions increases rapidly.



Visualizing Multivariate Inputs

- Consider a network with the same number of hidden units as input dimensions



(a) With a single input dimension, a model with one hidden unit creates one joint, which divides the axis into two linear regions.

(b) With two input dimensions, a model with two hidden units can divide the input space using two lines (here aligned with axes) to create four regions.

(c) With three input dimensions, a model with three hidden units can divide the input space using three planes (again aligned with axes) to create eight regions.

General Case

- Shallow neural network: $\mathbf{y} = f[\mathbf{x}, \phi]$ that maps a multi-dimensional input $\mathbf{x} \in \mathbb{R}^{D_i}$ to a multi-dimensional output $\mathbf{y} \in \mathbb{R}^{D_o}$ using $\mathbf{h} \in \mathbb{R}^{D}$ hidden units.
- Each hidden unit is computed as:

$$h_d = a\left[\theta_{d0} + \sum_{i=1}^{D_i} \theta_{di} x_i\right]$$

and these are combined linearly to create the output:

$$y_j = \phi_{j0} + \sum_{d=1}^{D} \phi_{jd} h_d,$$

where $a[\bullet]$ is a nonlinear activation function.

General Case

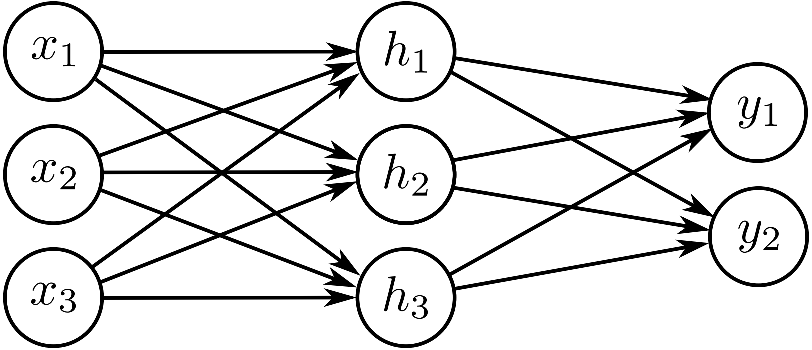
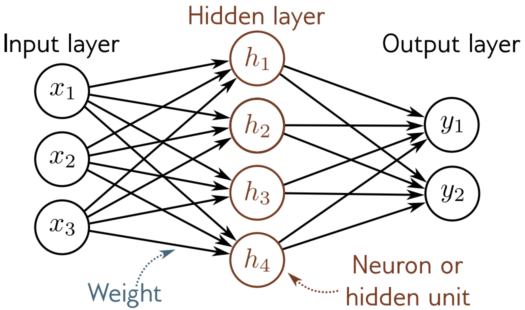


Figure: Visualization of neural network with three inputs and two outputs.

Terminology



- Slope parameters are referred to as network weights.
- The offset parameters are called biases.

Terminology

- Any neural network with at least one hidden layer is also called a multi-layer perceptron, or MLP for short.
- Networks with one hidden layer are sometimes referred to as shallow neural networks.
- Networks with multiple hidden layers are referred to as deep neural networks.
- Neural networks in which the connections form an acyclic graph are called feed-forward networks.
- If every element in one layer connects to every element in the next, the network is fully connected.

Other Activation Functions

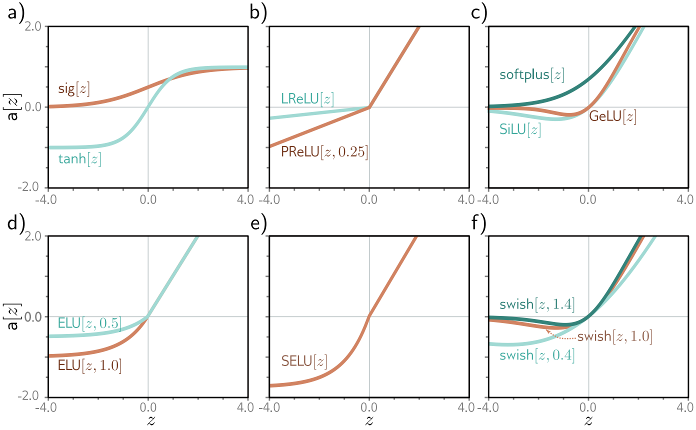


Figure: a) Logistic sigmoid and tanh functions. b) Leaky ReLU and parametric ReLU with parameter 0.25. c) SoftPlus, Gaussian error linear unit, and sigmoid linear unit. d) Exponential linear unit with parameters 0.5 and 1.0, e) Scaled exponential linear unit. f) Swish with parameters 0.4, 1.0, and 1.4.

References

1. Understanding Deep Learning, first edition by Simon J.D. Prince

- All the images have been taken from [1].

The End