Deep Neural Networks

EN5730 Machine Learning for Communications

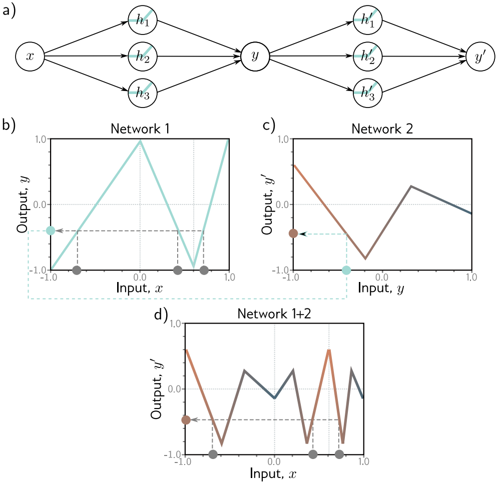
Samiru Gayan

Department of Electronic and Telecommunication Engineering
University of Moratuwa

2024

Composing Neural Networks

- We first consider composing two shallow networks so the output of the first becomes the input of the second.



Composing Neural Networks

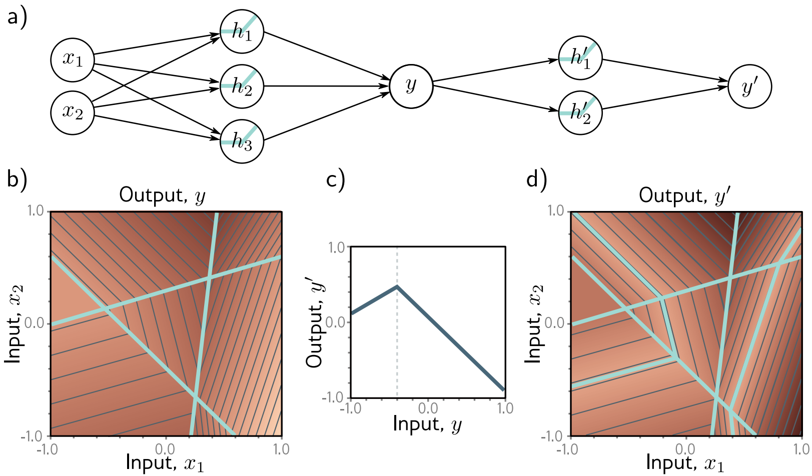
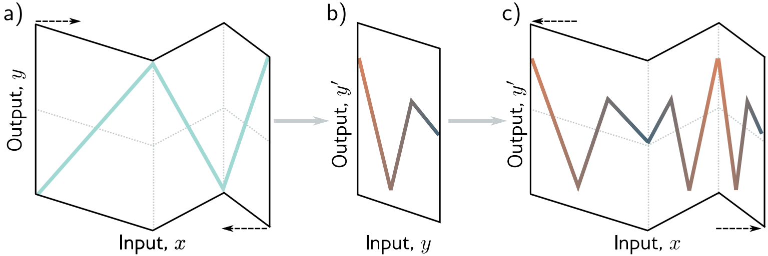


Figure: Caption

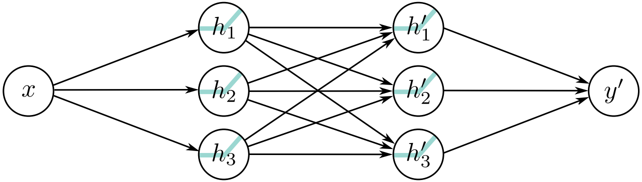
Composing Neural Networks

- A different way to think about composing networks is that the first network folds the input space x back onto itself so that multiple inputs generate the same output.
- Then the second network applies a function, which is replicated at all points that were folded on top of one another.



Deep Neural Networks

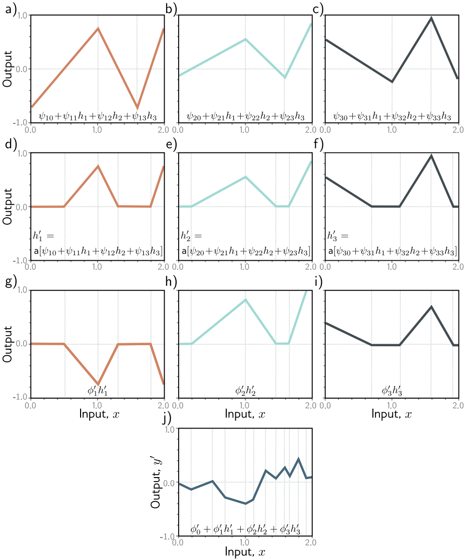
- Neural network with one input, one output, and two hidden layers, each containing three hidden units.



$$
\begin{aligned}
h_1 &= a[\theta_{10} + \theta_{11}x] & h'_1 &= a[\psi_{10} + \psi_{11}h_1 + \psi_{12}h_2 + \psi_{13}h_3] \\
h_2 &= a[\theta_{20} + \theta_{21}x] & h'_1 &= a[\psi_{20} + \psi_{21}h_1 + \psi_{22}h_2 + \psi_{23}h_3] \\
h_3 &= a[\theta_{30} + \theta_{31}x] & h'_1 &= a[\psi_{30} + \psi_{31}h_1 + \psi_{32}h_2 + \psi_{33}h_3]
\end{aligned}
$$

$$y' = \phi'_0 + \phi'_1 h'_1 + \phi'_2 h'_2 + \phi'_3 h'_3$$

Deep Neural Networks



Hyperparameters

- Modern networks might have more than a hundred layers with thousands of hidden units at each layer.
- The number of hidden units in each layer is referred to as the width of the network, and the number of hidden layers as the depth.
- The total number of hidden units is a measure of the network's capacity.

- We denote the number of layers as K and the number of hidden units in each layer as $D_1, D_2, \cdots, D_K$.
- These are examples of hyperparameters.
- They are quantities chosen before we learn the model parameters.

Matrix Notation

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \mathbf{a}\left[\begin{bmatrix} \theta_{10} \\ \theta_{20} \\ \theta_{30} \end{bmatrix} + \begin{bmatrix} \theta_{11} \\ \theta_{21} \\ \theta_{31} \end{bmatrix} x\right]$$

$$\begin{bmatrix} h'_1 \\ h'_2 \\ h'_3 \end{bmatrix} = \mathbf{a}\left[\begin{bmatrix} \psi_{10} \\ \psi_{20} \\ \psi_{30} \end{bmatrix} + \begin{bmatrix} \psi_{11} & \psi_{12} & \psi_{13} \\ \psi_{21} & \psi_{22} & \psi_{23} \\ \psi_{31} & \psi_{32} & \psi_{33} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}\right]$$

$$y' = \phi'_0 + \begin{bmatrix} \phi'_1 & \phi'_2 & \phi'_3 \end{bmatrix} \begin{bmatrix} h'_1 \\ h'_2 \\ h'_3 \end{bmatrix}$$

$$\mathbf{h} = \mathbf{a}\left[\theta_{\mathbf{0}} + \theta\mathbf{x}\right]$$

$$\mathbf{h}' = \mathbf{a}\left[\psi_{\mathbf{0}} + \psi\mathbf{h}\right]$$

$$y = \phi'_{\mathbf{0}} + \phi'\mathbf{h}'$$

General Formulation

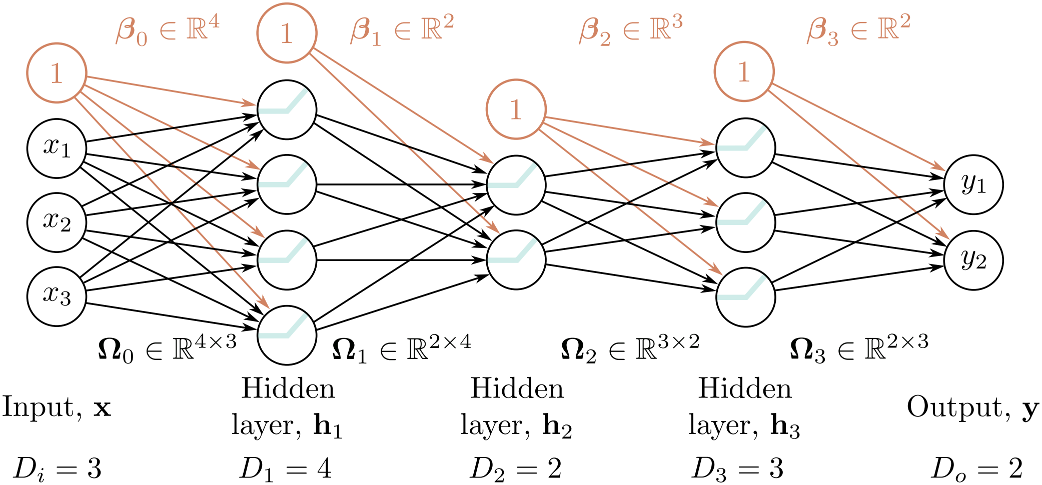


Figure: Matrix notation.

General Formulation

- Hidden units at layer k - $\mathbf{h}_k$
- the vector of biases that contribute to hidden layer $k+1$ - β_k
- the weights that are applied to the kth layer and contribute to the $(k+1)$th layer as $\boldsymbol{\Omega}_k$.

$$
\begin{aligned}
\mathbf{h}_1 &= \mathbf{a}\left[\beta_0 + \boldsymbol{\Omega}_0 \mathbf{x}\right] \\
\mathbf{h}_2 &= \mathbf{a}\left[\beta_1 + \boldsymbol{\Omega}_1 \mathbf{h}_1\right] \\
\mathbf{h}_3 &= \mathbf{a}\left[\beta_2 + \boldsymbol{\Omega}_2 \mathbf{h}_2\right] \\
&\bullet \\
\mathbf{h}_K &= \mathbf{a}\left[\beta_{K-1} + \boldsymbol{\Omega}_{K-1} \mathbf{h}_{K-1}\right] \\
\mathbf{y} &= \beta_K + \boldsymbol{\Omega}_K \mathbf{h}_K
\end{aligned}
$$

References

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

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

The End