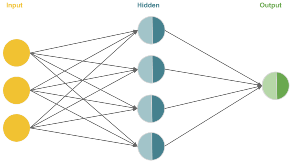
Deep Learning

MLP – Matrix Notation



$$\begin{pmatrix} 3 & -9 & 2 \\ 11 & -2 & 7 \\ -6 & 3 & -4 \\ 6 & -1 & 5 \end{pmatrix} \begin{pmatrix} 5 \\ 2 \\ -1 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} 3 & -12 & 8 & 1 \end{pmatrix} \begin{pmatrix} 6 \end{pmatrix}$$

W^T $\mathbf{b}$ $\mathbf{u}^T$ c

Learning goals

- Compact representation of neural network equations
- Vector notation for neuron layers
- Vector and matrix notation of bias and weight parameters

SINGLE HIDDEN LAYER NETWORKS: NOTATIONS

- The input $\mathbf{x}$ is a column vector with dimensions $p \times 1$.
- $\mathbf{W}$ is a weight matrix with dimensions $p \times m$, where m is the amount of hidden neurons:

$$\mathbf{W} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{p,1} & w_{p,2} & \cdots & w_{p,m} \end{pmatrix}$$

SINGLE HIDDEN LAYER NETWORKS: NOTATIONS

Hidden layer:

- For example, to obtain z_1, we pick the first column of W:

$$\mathbf{W}_1 = \begin{pmatrix} w_{1,1} \\ w_{2,1} \\ \vdots \\ w_{p,1} \end{pmatrix}$$

and compute

$$z_1 = \sigma(\mathbf{W}_1^T \mathbf{x} + b_1) \ ,$$

where b_1 is the bias of the first hidden neuron and $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function.

SINGLE HIDDEN LAYER NETWORKS: NOTATION

- The network has m hidden neurons $z_1, \ldots, z_m$ with

$$z_j = \sigma(\mathbf{W}_j^T \mathbf{x} + b_j)$$

  - $z_{in,j} = \mathbf{W}_j^T \mathbf{x} + b_j$
  - $z_{out,j} = \sigma(z_{in,j}) = \sigma(\mathbf{W}_j^T \mathbf{x} + b_j)$

  for $j \in \{1, \ldots, m\}$.

- Vectorized notation:
  - $\mathbf{z}_{in} = (z_{in,1}, \ldots, z_{in,m})^T = \mathbf{W}^T \mathbf{x} + \mathbf{b}$
    (Note: $\mathbf{W}^T \mathbf{x}$ = $(\mathbf{x}^T \mathbf{W})^T$)
  - $\mathbf{z} = \mathbf{z}_{out} = \sigma(\mathbf{z}_{in}) = \sigma(\mathbf{W}^T \mathbf{x} + \mathbf{b})$, where the (hidden layer) activation function σ is applied element-wise to $\mathbf{z}_{in}$.

SINGLE HIDDEN LAYER NETWORKS: NOTATION

- **Bias term**:
  - We sometimes omit the bias term by adding a constant feature to the input $\tilde{\mathbf{x}} = (1, x_1, ..., x_p)$ and by adding the bias term to the weight matrix

    $$\tilde{\mathbf{W}} = (\mathbf{b}, \mathbf{W}_1, ..., \mathbf{W}_p).$$

  - **Note**: For simplification purposes, we will not explicitly represent the bias term graphically in the following. However, the above "trick" makes it straightforward to represent it graphically.

SINGLE HIDDEN LAYER NETWORKS: NOTATION

Output layer:

- For regression or binary classification: one output unit f where
  - $f_{in} = \mathbf{u}^T\mathbf{z} + c$, i.e. a linear combination of derived features plus the bias term c of the output neuron, and
  - $f(\mathbf{x}) = f_{out} = \tau(f_{in}) = \tau(\mathbf{u}^T\mathbf{z} + c)$, where τ is the output activation function.
- For regression τ is the identity function.
- For binary classification, τ is a sigmoid function.
- **Note**: The purpose of the hidden-layer activation function σ is to introduce non-linearities so that the network is able to learn complex functions whereas the purpose of τ is merely to get the final score to the same range as the target.

SINGLE HIDDEN LAYER NETWORKS: NOTATION

Multiple inputs:

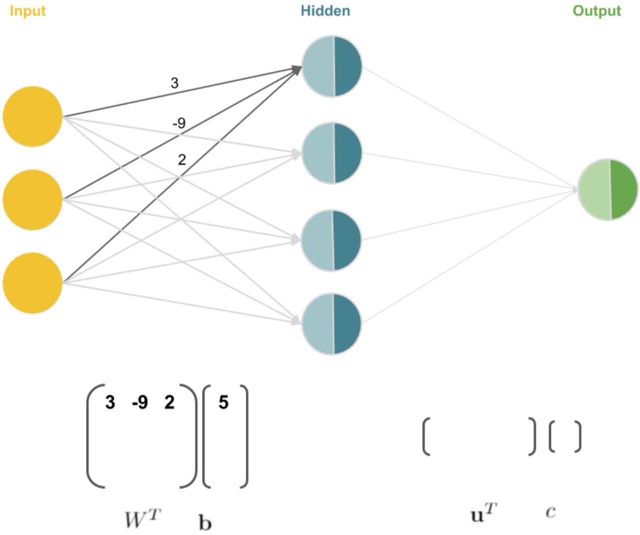
- It is possible to feed multiple inputs to a neural network simultaneously.
- The inputs $\mathbf{x}^{(i)}$, for $i \in \{1, \ldots, n\}$, are arranged as rows in the **design matrix $\mathbf{X}$**.
  - $\mathbf{X}$ is a $(n \times p)$-matrix.
- The weighted sum in the hidden layer is now computed as $\mathbf{XW} + \boldsymbol{B}$, where,
  - $\mathbf{W}$, as usual, is a $(p \times m)$ matrix, and,
  - $\boldsymbol{B}$ is a $(n \times m)$ matrix containing the bias vector $\mathbf{b}$ (duplicated) as the rows of the matrix.
- The *matrix* of hidden activations $\boldsymbol{Z} = \sigma(\mathbf{XW} + \boldsymbol{B})$
  - $\boldsymbol{Z}$ is a $(n \times m)$ matrix.

SINGLE HIDDEN LAYER NETWORKS: NOTATION

- The final output of the network, which contains a prediction for each input, is $\tau(\boldsymbol{Z}\mathbf{u} + \boldsymbol{C})$, where
  - $\mathbf{u}$ is the vector of weights of the output neuron, and,
  - $\boldsymbol{C}$ is a $(n \times 1)$ matrix whose elements are the (scalar) bias c of the output neuron.

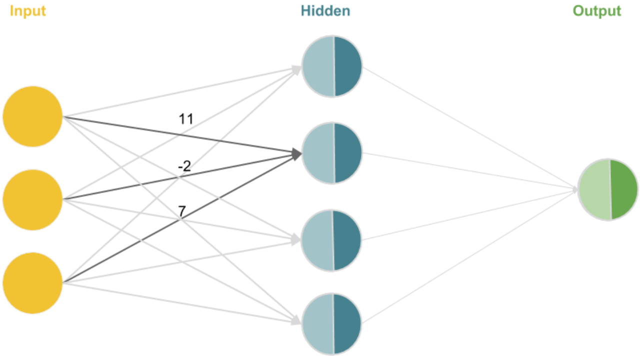
SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

- Weights (and biases) of the network.



SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

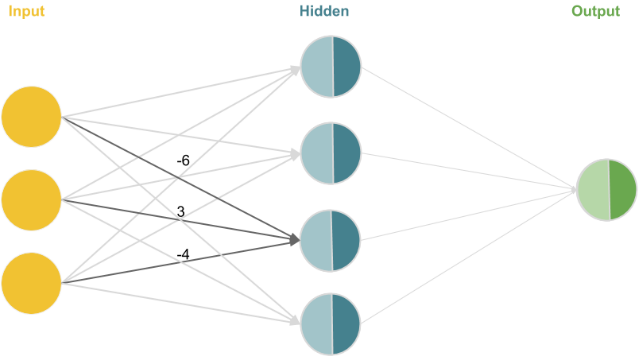
- Weights (and biases) of the network.



$$\underbrace{\begin{pmatrix} 3 & -9 & 2 \\ \mathbf{11} & \mathbf{-2} & \mathbf{7} \end{pmatrix}}_{W^T} \underbrace{\begin{bmatrix} 5 \\ \mathbf{2} \end{bmatrix}}_{\mathbf{b}} \qquad \underbrace{\begin{bmatrix} \quad \end{bmatrix}}_{\mathbf{u}^T} \underbrace{\begin{bmatrix} \; \end{bmatrix}}_{c}$$

SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

- Weights (and biases) of the network.

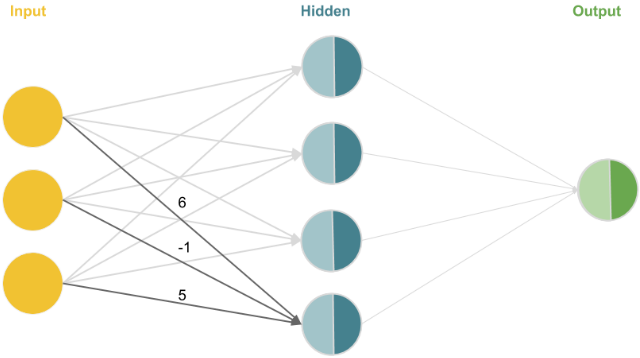


$$\begin{pmatrix} 3 & -9 & 2 \\ 11 & -2 & 7 \\ \mathbf{-6} & \mathbf{3} & \mathbf{-4} \end{pmatrix} \begin{bmatrix} 5 \\ 2 \\ \mathbf{-1} \end{bmatrix} \qquad \begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} \quad \end{bmatrix}$$

$$W^T \quad \mathbf{b} \qquad\qquad \mathbf{u}^T \quad c$$

SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

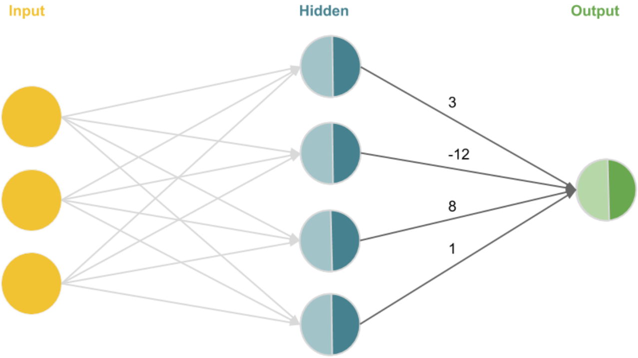
- Weights (and biases) of the network.



$$\underset{W^T}{\begin{pmatrix} 3 & -9 & 2 \\ 11 & -2 & 7 \\ -6 & 3 & -4 \\ \mathbf{6} & \mathbf{-1} & \mathbf{5} \end{pmatrix}} \underset{\mathbf{b}}{\begin{bmatrix} 5 \\ 2 \\ -1 \\ \mathbf{1} \end{bmatrix}} \qquad \underset{\mathbf{u}^T}{\begin{bmatrix} \quad \end{bmatrix}} \underset{c}{\begin{bmatrix} \quad \end{bmatrix}}$$

SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

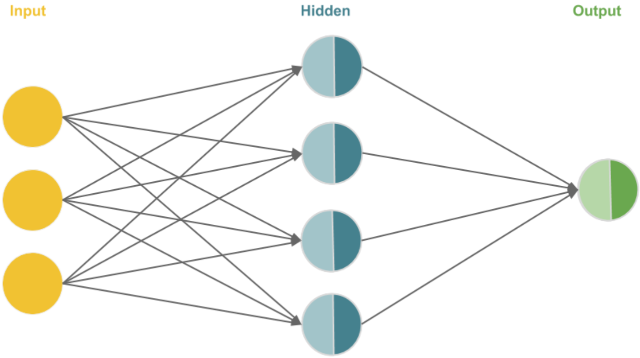
- Weights (and biases) of the network.



$$\underset{W^T}{\begin{pmatrix} 3 & -9 & 2 \\ 11 & -2 & 7 \\ -6 & 3 & -4 \\ 6 & -1 & 5 \end{pmatrix}} \underset{\mathbf{b}}{\begin{bmatrix} 5 \\ 2 \\ -1 \\ 1 \end{bmatrix}} \qquad \underset{\mathbf{u}^T}{\begin{bmatrix} 3 & -12 & 8 & 1 \end{bmatrix}} \underset{c}{\begin{bmatrix} 6 \end{bmatrix}}$$

SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

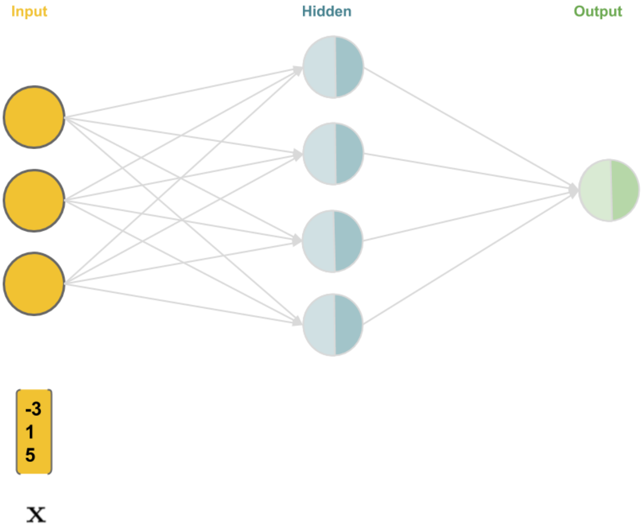
- Weights (and biases) of the network.



$$\underbrace{\begin{pmatrix} 3 & -9 & 2 \\ 11 & -2 & 7 \\ -6 & 3 & -4 \\ 6 & -1 & 5 \end{pmatrix}}_{W^T} \underbrace{\begin{bmatrix} 5 \\ 2 \\ -1 \\ 1 \end{bmatrix}}_{\mathbf{b}} \qquad \underbrace{\begin{bmatrix} 3 & -12 & 8 & 1 \end{bmatrix}}_{\mathbf{u}^T} \underbrace{\begin{bmatrix} 6 \end{bmatrix}}_{c}$$

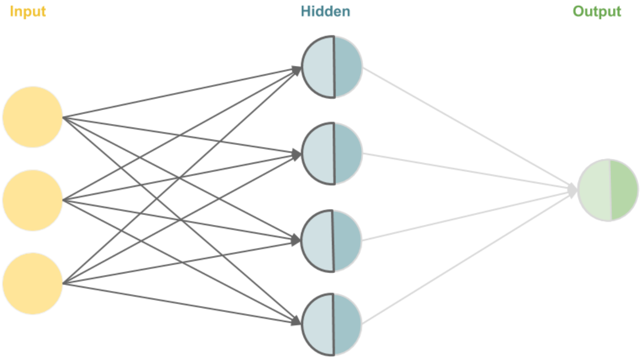
SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

Forward pass through the shallow neural network.



SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

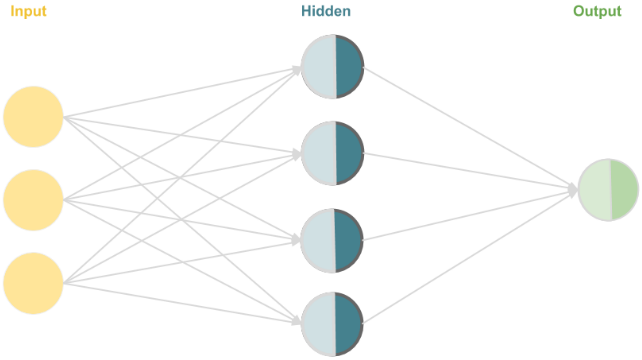
Forward pass through the shallow neural network.



$$\underbrace{\begin{pmatrix} -3 \\ 1 \\ 5 \end{pmatrix}}_{\mathbf{x}} \qquad \underbrace{\begin{pmatrix} (-3)*3 + 1*(-9) + 5*2 + 5 \\ (-3)*11 + 1*(-2) + 5*7 + 2 \\ (-3)*(-6) + 1*3 + 5*(-4) + (-1) \\ (-3)*6 + 1*(-1) + 5*5 + 1 \end{pmatrix}}_{\mathbf{z}_{in} = W^\top \mathbf{x} + \mathbf{b}}$$

SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

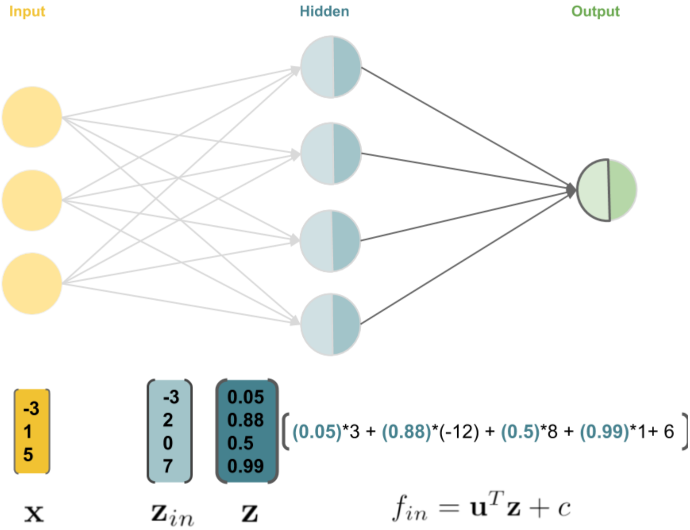
Forward pass through the shallow neural network.



$$\underbrace{\begin{bmatrix} -3 \\ 1 \\ 5 \end{bmatrix}}_{\mathbf{x}} \quad \underbrace{\begin{bmatrix} -3 \\ 2 \\ 0 \\ 7 \end{bmatrix}}_{\mathbf{z}_{in}} \quad \underbrace{\begin{pmatrix} 1/\,(1+\exp(-(-3))) \\ 1/\,(1+\exp(-2)) \\ 1/\,(1+\exp(-0)) \\ 1/\,(1+\exp(-7)) \end{pmatrix}}_{\mathbf{z}=\mathbf{z}_{out}=\sigma(\mathbf{z}_{in})}$$

SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

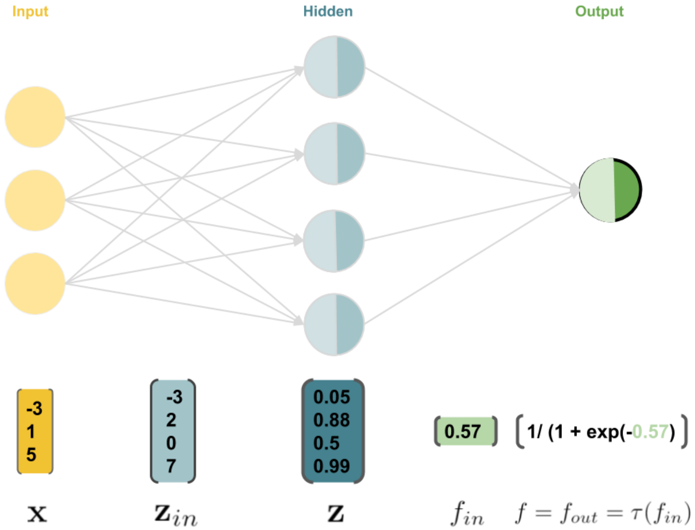
Forward pass through the shallow neural network.



$$f_{in} = \mathbf{u}^T \mathbf{z} + c$$

SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

Forward pass through the shallow neural network.



SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

Forward pass through the shallow neural network.