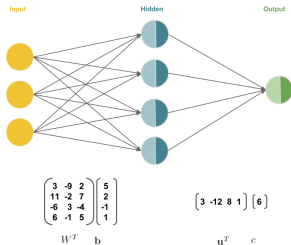


# Deep Learning

## MLP – Matrix Notation



### 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  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  is an activation function.

# SINGLE HIDDEN LAYER NETWORKS: NOTATION

- The network has  $m$  hidden neurons  $z_1, \dots, z_m$  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  $j \in \{1, \dots, m\}$ .

- Vectorized notation:
  - $\mathbf{z}_{in} = (z_{in,1}, \dots, z_{in,m})^T = \mathbf{W}^T \mathbf{x} + \mathbf{b}$   
(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  $\sigma$  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  $\tilde{\mathbf{x}} = (1, x_1, \dots, x_p)$  and by adding the bias term to the weight matrix

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

- **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  $\tau$  is the output activation function.
- For regression  $\tau$  is the identity function.
- For binary classification,  $\tau$  is a sigmoid function.
- **Note:** The purpose of the hidden-layer activation function  $\sigma$  is to introduce non-linearities so that the network is able to learn complex functions whereas the purpose of  $\tau$  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  $i \in \{1, \dots, n\}$ , 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  $\mathbf{XW} + \mathbf{B}$ , where,
  - $\mathbf{W}$ , as usual, is a  $(p \times m)$  matrix, and,
  - $\mathbf{B}$  is a  $(n \times m)$  matrix containing the bias vector  $\mathbf{b}$  (duplicated) as the rows of the matrix.
- The *matrix* of hidden activations  $\mathbf{Z} = \sigma(\mathbf{XW} + \mathbf{B})$ 
  - $\mathbf{Z}$  is a  $(n \times m)$  matrix.

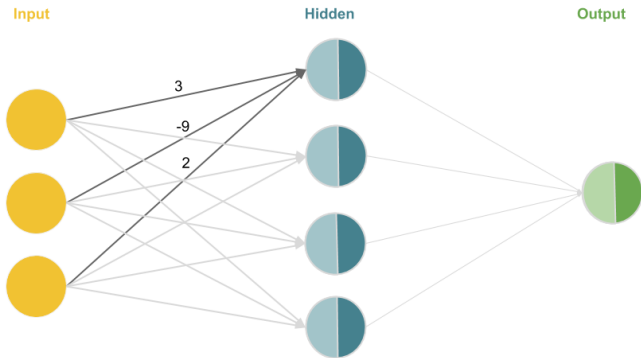
# SINGLE HIDDEN LAYER NETWORKS: NOTATION

- The final output of the network, which contains a prediction for each input, is  $\tau(\mathbf{Z}\mathbf{u} + \mathbf{C})$ , where
  - $\mathbf{u}$  is the vector of weights of the output neuron, and,
  - $\mathbf{C}$  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.



$$\begin{pmatrix} 3 & -9 & 2 \\ & & \\ & & \end{pmatrix} \begin{pmatrix} 5 \\ \\ \end{pmatrix}$$

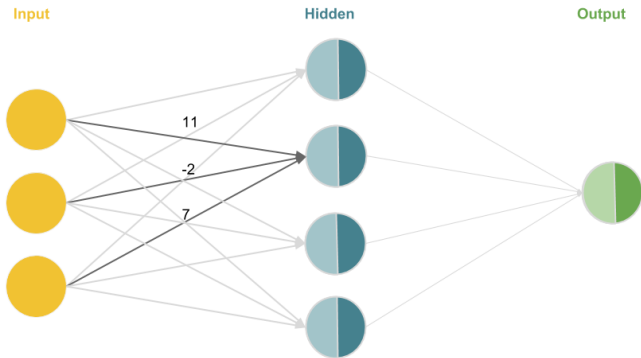
$W^T \quad \mathbf{b}$

$$\begin{pmatrix} & & \end{pmatrix} \begin{pmatrix} \end{pmatrix}$$

$\mathbf{u}^T \quad c$

# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

- Weights (and biases) of the network.



$$\begin{pmatrix} 3 & -9 & 2 \\ 11 & -2 & 7 \end{pmatrix} \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$

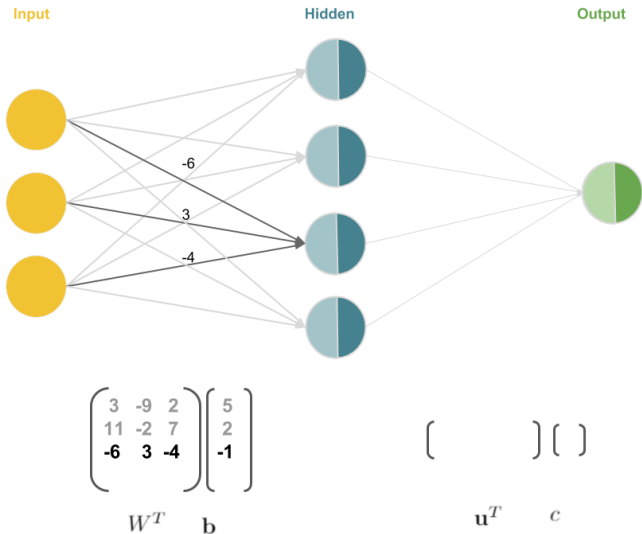
$W^T \quad \mathbf{b}$

$$\begin{pmatrix} & & & & \end{pmatrix} \begin{pmatrix} \end{pmatrix}$$

$\mathbf{u}^T \quad c$

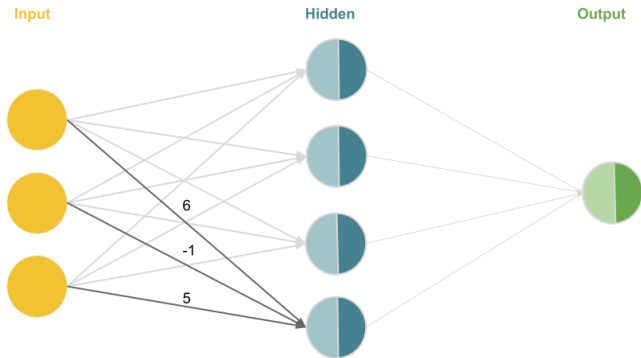
# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

- Weights (and biases) of the network.



# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

- Weights (and biases) of the network.



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

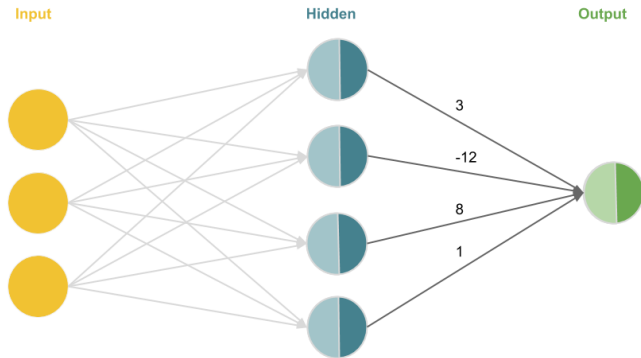
$$W^T \quad \mathbf{b}$$

$$\begin{pmatrix} \phantom{0} & \phantom{0} & \phantom{0} & \phantom{0} \end{pmatrix} \begin{pmatrix} \phantom{0} \end{pmatrix}$$

$$\mathbf{u}^T \quad c$$

# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

- Weights (and biases) of the network.



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

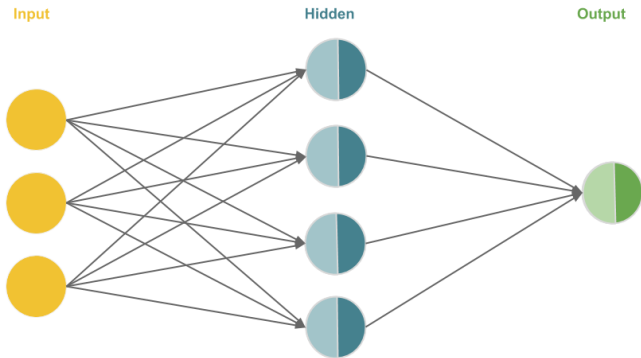
$$W^T \quad \mathbf{b}$$

$$\begin{bmatrix} 3 & -12 & 8 & 1 \end{bmatrix} \begin{bmatrix} 6 \end{bmatrix}$$

$$\mathbf{u}^T \quad c$$

# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

- Weights (and biases) of the network.



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

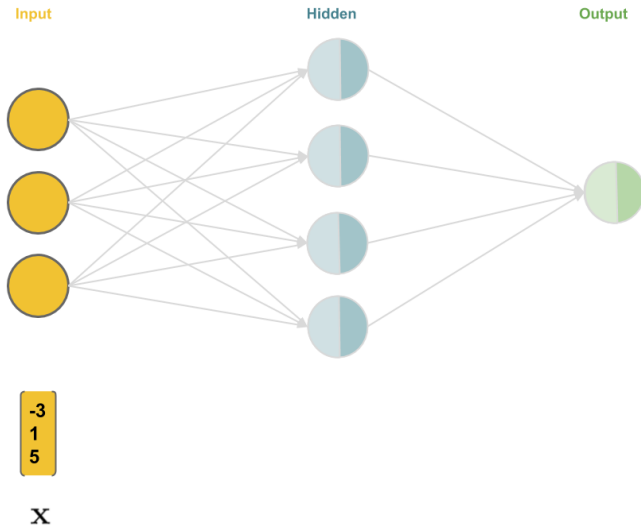
$$W^T \quad \mathbf{b}$$

$$\begin{pmatrix} 3 & -12 & 8 & 1 \end{pmatrix} \begin{pmatrix} 6 \end{pmatrix}$$

$$\mathbf{u}^T \quad c$$

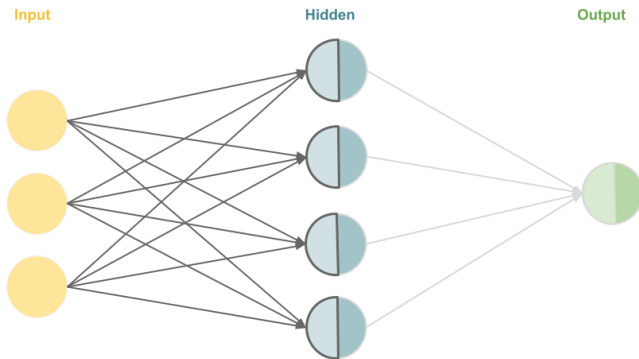
# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

Forward pass through the shallow neural network.



# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

Forward pass through the shallow neural network.



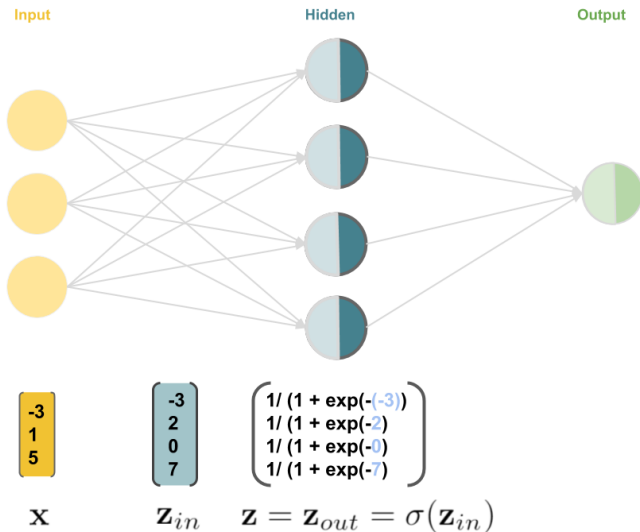
$$\begin{bmatrix} -3 \\ 1 \\ 5 \end{bmatrix} \quad \begin{pmatrix} (-3)*3 + 1*(-9) + 5*2 + 5 \\ (-3)*11 + 1*(-2) + 5*7 + 2 \\ (-3)*(-6) + 1*3 + 5*(-4) + (-1) \\ (-3)*6 + 1*(-1) + 5*5 + 1 \end{pmatrix}$$

$$\mathbf{x} \quad \mathbf{z}_{in} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$$



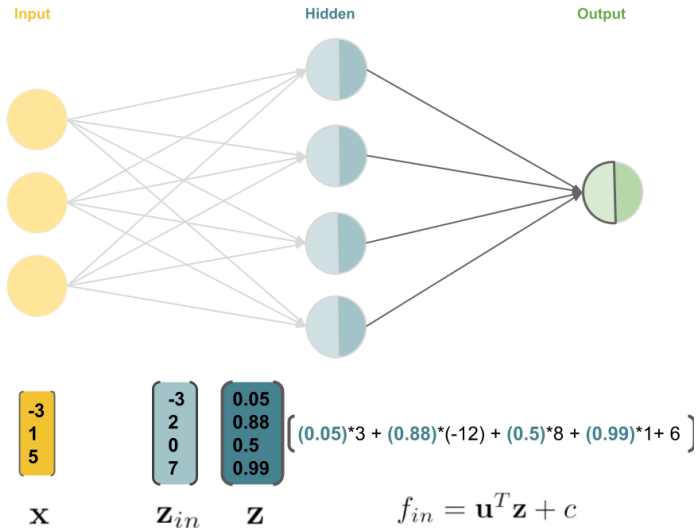
# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

Forward pass through the shallow neural network.



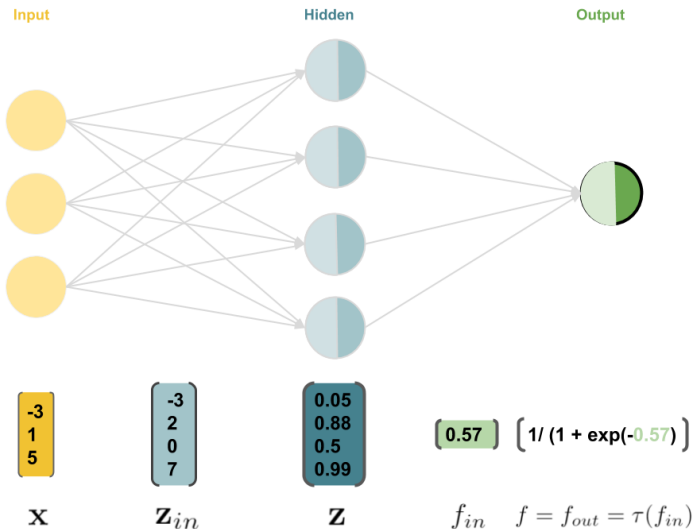
# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

Forward pass through the shallow neural network.



# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

Forward pass through the shallow neural network.



# SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

Forward pass through the shallow neural network.

