CS-466/566: Math for AI

Module 06: Deep Learning Fundamentals-2

Dr. Mahmoud Mahmoud
The University of Alabama

2026-03-23

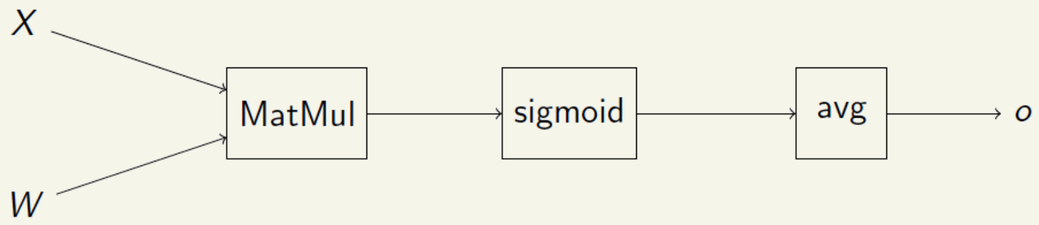
TABLE OF CONTENTS

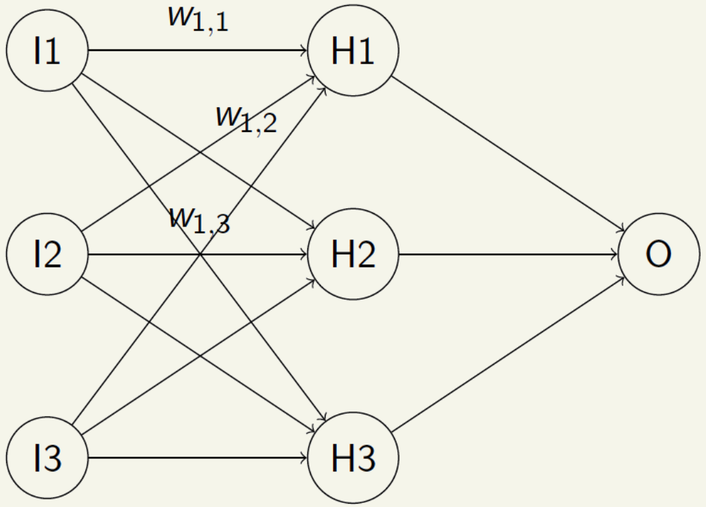
Neural Network As Computational Graph

Suppose we have a neural network with one hidden layer and one output layer to predict the probability of a house being sold given (size, bedrooms, and bathrooms)

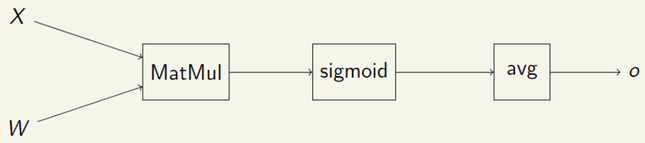
For simplicity, no bias term and no weights for the output layer, only average:

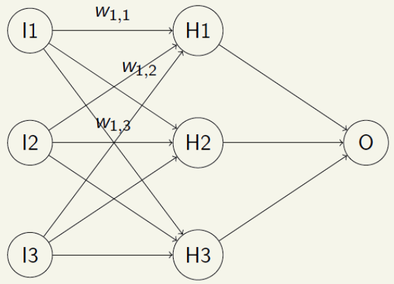
This is a composition of functions:





Computational Graph [Matrix Multiplication]



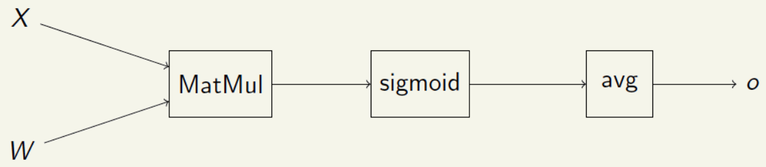


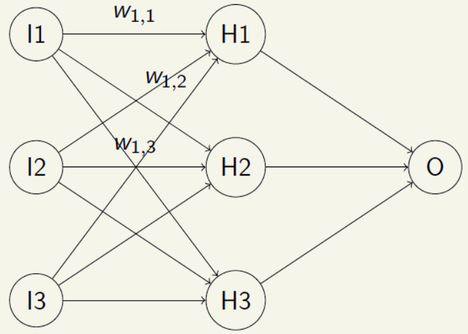
Matrix Multiplication: Training works on batches of data (e.g. 4 houses)

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} \\ x_1^{(3)} & x_2^{(3)} & x_3^{(3)} \\ x_1^{(4)} & x_2^{(4)} & x_3^{(4)} \end{bmatrix} \times \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} \end{bmatrix} = \begin{bmatrix} h_1^{(1)} & h_2^{(1)} & h_3^{(1)} \\ h_1^{(2)} & h_2^{(2)} & h_3^{(2)} \\ h_1^{(3)} & h_2^{(3)} & h_3^{(3)} \\ h_1^{(4)} & h_2^{(4)} & h_3^{(4)} \end{bmatrix}$$

where $x_i^{(j)}$ is feature i of house j, $w_{i,k}$ is the weight from input i to hidden node k, and $h_k^{(j)}$ is hidden node k's value for house j

Computational Graph [Sigmoid]



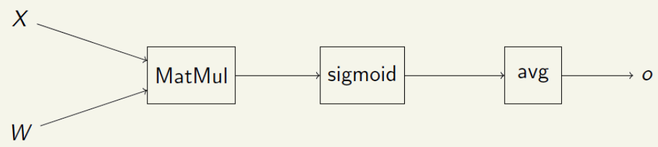


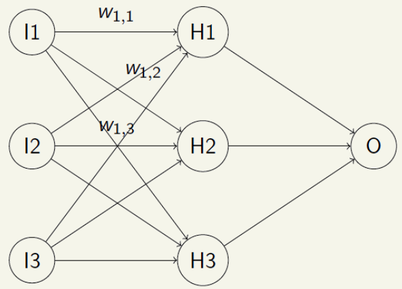
Sigmoid: it is applied to each element of the hidden matrix

$$\begin{bmatrix} h_1^{(1)} & h_2^{(1)} & h_3^{(1)} \\ h_1^{(2)} & h_2^{(2)} & h_3^{(2)} \\ h_1^{(3)} & h_2^{(3)} & h_3^{(3)} \\ h_1^{(4)} & h_2^{(4)} & h_3^{(4)} \end{bmatrix} \rightarrow \begin{bmatrix} \sigma(h_1^{(1)}) & \sigma(h_2^{(1)}) & \sigma(h_3^{(1)}) \\ \sigma(h_1^{(2)}) & \sigma(h_2^{(2)}) & \sigma(h_3^{(2)}) \\ \sigma(h_1^{(3)}) & \sigma(h_2^{(3)}) & \sigma(h_3^{(3)}) \\ \sigma(h_1^{(4)}) & \sigma(h_2^{(4)}) & \sigma(h_3^{(4)}) \end{bmatrix} = \begin{bmatrix} a_1^{(1)} & a_2^{(1)} & a_3^{(1)} \\ a_1^{(2)} & a_2^{(2)} & a_3^{(2)} \\ a_1^{(3)} & a_2^{(3)} & a_3^{(3)} \\ a_1^{(4)} & a_2^{(4)} & a_3^{(4)} \end{bmatrix}$$

where $a_k^{(j)}$ is the value of activation node k for house j

Computational Graph [Average]



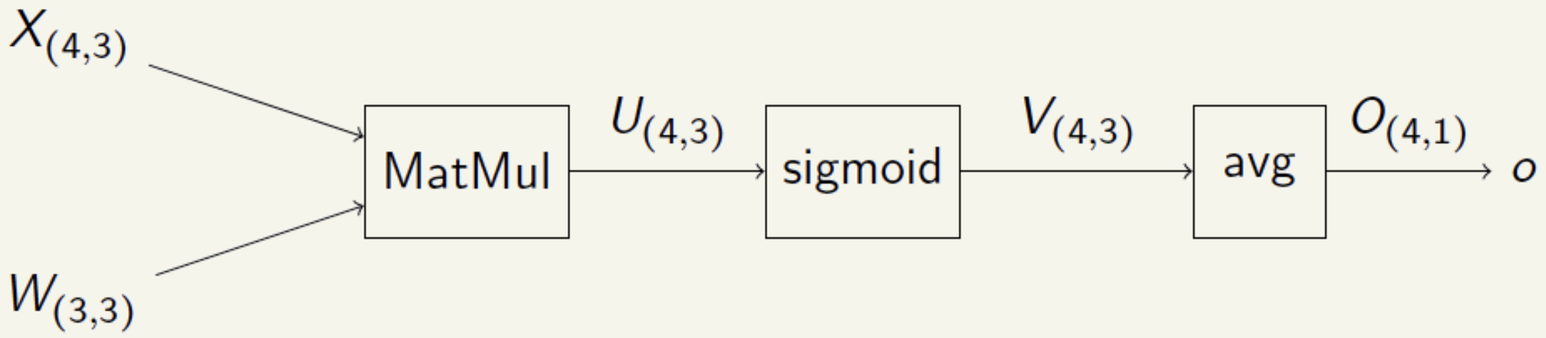


Average: it is applied to each row of the activation matrix

$$\begin{bmatrix} a_1^{(1)} & a_2^{(1)} & a_3^{(1)} \\ a_1^{(2)} & a_2^{(2)} & a_3^{(2)} \\ a_1^{(3)} & a_2^{(3)} & a_3^{(3)} \\ a_1^{(4)} & a_2^{(4)} & a_3^{(4)} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{a_1^{(1)}+a_2^{(1)}+a_3^{(1)}}{3} \\ \frac{a_1^{(2)}+a_2^{(2)}+a_3^{(2)}}{3} \\ \frac{a_1^{(3)}+a_2^{(3)}+a_3^{(3)}}{3} \\ \frac{a_1^{(4)}+a_2^{(4)}+a_3^{(4)}}{3} \end{bmatrix}$$

Every house now has an average probability of being sold predicted by three neurons.

Chain Rule of Neural Network



What is the derivative of O with respect to W ?

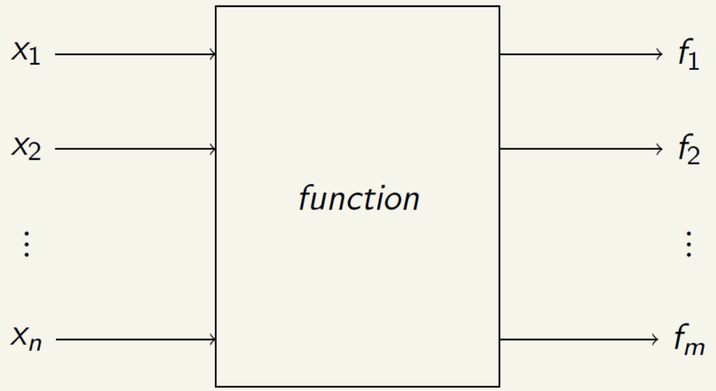
$$\frac{\partial O}{\partial W} = \frac{\partial U}{\partial W} \odot \frac{\partial V}{\partial U} \odot \frac{\partial O}{\partial V}$$

Should be same shape as W, this symbol $\odot$ is element-wise multiplication

If each computation node in the graph has a known, easy-to-compute local derivative, we can compute the derivative of the entire graph with respect to the weights using the **Chain Rule**

TABLE OF CONTENTS

Derivative of Functions with Multiple Outputs



For functions with multiple outputs, the derivative becomes a matrix called the **Jacobian matrix**:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Example: Matrix Multiplication

Consider multiplying matrices X (3×3) and W (3×2) to get output O (3×2):

$$\underbrace{\begin{bmatrix} o_{11} & o_{12} \\ o_{21} & o_{22} \\ o_{31} & o_{32} \end{bmatrix}}_{\text{O (3×2)}} = \underbrace{\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}}_{\text{X (3×3)}} \underbrace{\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix}}_{\text{W (3×2)}}$$

Each element in O is computed as:

$$\begin{aligned}
o_{11} &= x_{11}w_{11} + x_{12}w_{21} + x_{13}w_{31} \\
o_{12} &= x_{11}w_{12} + x_{12}w_{22} + x_{13}w_{32} \\
o_{21} &= x_{21}w_{11} + x_{22}w_{21} + x_{23}w_{31} \\
o_{22} &= x_{21}w_{12} + x_{22}w_{22} + x_{23}w_{32} \\
o_{31} &= x_{31}w_{11} + x_{32}w_{21} + x_{33}w_{31} \\
o_{32} &= x_{31}w_{12} + x_{32}w_{22} + x_{33}w_{32}
\end{aligned}$$

Jacobian of Matrix Multiplication

The Jacobian matrix $\frac{\partial O}{\partial W}$ will have dimensions (6×6) since O has 6 elements (3×2) and W has 6 elements (3×2). Each entry (i, j) represents how the i-th element of O changes with respect to the j-th element of W.

Computing each element using the derivative sum rule:

$$\frac{\partial o_{11}}{\partial w_{11}} = x_{11} \quad \frac{\partial o_{11}}{\partial w_{12}} = 0 \quad \frac{\partial o_{11}}{\partial w_{21}} = x_{12} \quad \frac{\partial o_{11}}{\partial w_{22}} = 0 \quad \frac{\partial o_{11}}{\partial w_{31}} = x_{13} \quad \frac{\partial o_{11}}{\partial w_{32}} = 0$$

$$\frac{\partial o_{12}}{\partial w_{11}} = 0 \quad \frac{\partial o_{12}}{\partial w_{12}} = x_{11} \quad \frac{\partial o_{12}}{\partial w_{21}} = 0 \quad \frac{\partial o_{12}}{\partial w_{22}} = x_{12} \quad \frac{\partial o_{12}}{\partial w_{31}} = 0 \quad \frac{\partial o_{12}}{\partial w_{32}} = x_{13}$$

$$\frac{\partial o_{21}}{\partial w_{11}} = x_{21} \quad \frac{\partial o_{21}}{\partial w_{12}} = 0 \quad \frac{\partial o_{21}}{\partial w_{21}} = x_{22} \quad \frac{\partial o_{21}}{\partial w_{22}} = 0 \quad \frac{\partial o_{21}}{\partial w_{31}} = x_{23} \quad \frac{\partial o_{21}}{\partial w_{32}} = 0$$

$$\frac{\partial o_{22}}{\partial w_{11}} = 0 \quad \frac{\partial o_{22}}{\partial w_{12}} = x_{21} \quad \frac{\partial o_{22}}{\partial w_{21}} = 0 \quad \frac{\partial o_{22}}{\partial w_{22}} = x_{22} \quad \frac{\partial o_{22}}{\partial w_{31}} = 0 \quad \frac{\partial o_{22}}{\partial w_{32}} = x_{23}$$

$$\frac{\partial o_{31}}{\partial w_{11}} = x_{31} \quad \frac{\partial o_{31}}{\partial w_{12}} = 0 \quad \frac{\partial o_{31}}{\partial w_{21}} = x_{32} \quad \frac{\partial o_{31}}{\partial w_{22}} = 0 \quad \frac{\partial o_{31}}{\partial w_{31}} = x_{33} \quad \frac{\partial o_{31}}{\partial w_{32}} = 0$$

$$\frac{\partial o_{32}}{\partial w_{11}} = 0 \quad \frac{\partial o_{32}}{\partial w_{12}} = x_{31} \quad \frac{\partial o_{32}}{\partial w_{21}} = 0 \quad \frac{\partial o_{32}}{\partial w_{22}} = x_{32} \quad \frac{\partial o_{32}}{\partial w_{31}} = 0 \quad \frac{\partial o_{32}}{\partial w_{32}} = x_{33}$$

Jacobian of Matrix Multiplication (Reorganized with X^T)

Notice that each pair of rows contains the a row from X, repeated twice with shifted positions :

$$\frac{\partial O}{\partial W} = \begin{bmatrix} x_{11} & 0 & x_{12} & 0 & x_{13} & 0 \\ 0 & x_{11} & 0 & x_{12} & 0 & x_{13} \\ x_{21} & 0 & x_{22} & 0 & x_{23} & 0 \\ 0 & x_{21} & 0 & x_{22} & 0 & x_{23} \\ x_{31} & 0 & x_{32} & 0 & x_{33} & 0 \\ 0 & x_{31} & 0 & x_{32} & 0 & x_{33} \end{bmatrix}$$

For the purpose of backprop, we can rewrite the Jacobian as (same size as W):

$$\frac{\partial O}{\partial W_{(3\times 2)}} = \begin{bmatrix} \frac{\partial O}{\partial w_{11}} & \frac{\partial O}{\partial w_{12}} \\ \frac{\partial O}{\partial w_{21}} & \frac{\partial O}{\partial w_{22}} \\ \frac{\partial O}{\partial w_{31}} & \frac{\partial O}{\partial w_{32}} \end{bmatrix} = \begin{bmatrix} x_{11}+x_{21}+x_{31} & x_{11}+x_{21}+x_{31} \\ x_{12}+x_{22}+x_{32} & x_{12}+x_{22}+x_{32} \\ x_{13}+x_{23}+x_{33} & x_{13}+x_{23}+x_{33} \end{bmatrix}$$

This is equivalent to:

$$\frac{\partial O}{\partial W} = \begin{bmatrix} x_{11} & x_{21} & x_{31} \\ x_{12} & x_{22} & x_{32} \\ x_{13} & x_{23} & x_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} = X^T \cdot \mathbf{1}_{3\times 2}$$

Matrix Multiplication [What to remember?]

For any matrix $X_{m\times k}$ and $W_{k\times n}$ and matrix multiplication $O = XW$.

$$\frac{\partial O}{\partial W} = X^T_{k\times m} \cdot \mathbf{1}_{m\times n} \qquad \text{[Same size of } W_{k\times n}\text{]}$$

Will be replaced by the incoming derivative dL/dO

$$\frac{\partial O}{\partial X} = \mathbf{1}_{m\times n} \cdot W^T_{n\times k} \qquad \text{[Same size of } X_{m\times k}\text{]}$$

Will be replaced by the incoming derivative dL/dO

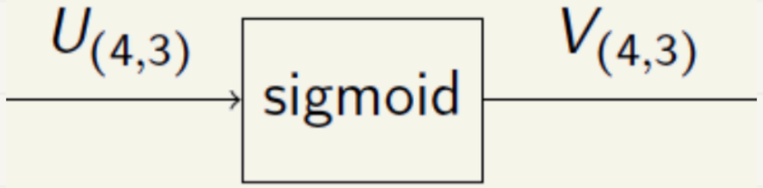
TABLE OF CONTENTS

Sigmoid on a Matrix [Element-wise]

Matrix form: $V_{4\times3} = \sigma(U_{4\times3})$ means matching entries:

$$\begin{bmatrix} V_{11} & V_{12} & V_{13} \\ V_{21} & V_{22} & V_{23} \\ V_{31} & V_{32} & V_{33} \\ V_{41} & V_{42} & V_{43} \end{bmatrix} = \begin{bmatrix} \sigma(U_{11}) & \sigma(U_{12}) & \sigma(U_{13}) \\ \sigma(U_{21}) & \sigma(U_{22}) & \sigma(U_{23}) \\ \sigma(U_{31}) & \sigma(U_{32}) & \sigma(U_{33}) \\ \sigma(U_{41}) & \sigma(U_{42}) & \sigma(U_{43}) \end{bmatrix}$$

So $V_{ij} = \sigma(U_{ij})$ for every (i, j).



Derivative of sigmoid (same 4×3 shape as U and V)

- **One entry:** if z is a single input, $\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big)$.
- **Whole matrix:** apply that same rule **independently** at each (i, j). You get a 4×3 matrix of partial derivatives, one per entry of U.
- **With $V = \sigma(U)$:** $\sigma'(U) = V \odot (\mathbf{1} - V)$ (element-wise; $\mathbf{1}$ same shape as V).

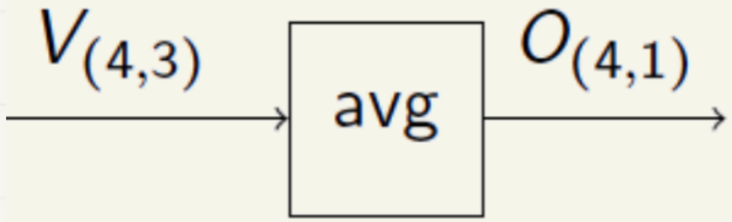
TABLE OF CONTENTS

Row-wise average: matrix form and derivative

Matrix form: $V_{4\times3}$ is averaged **along columns** (same row index i) to produce $O_{4\times1}$.

$$\begin{bmatrix} V_{11} & V_{12} & V_{13} \\ V_{21} & V_{22} & V_{23} \\ V_{31} & V_{32} & V_{33} \\ V_{41} & V_{42} & V_{43} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{V_{11}+V_{12}+V_{13}}{3} \\ \frac{V_{21}+V_{22}+V_{23}}{3} \\ \frac{V_{31}+V_{32}+V_{33}}{3} \\ \frac{V_{41}+V_{42}+V_{43}}{3} \end{bmatrix} = O_{4\times1}$$

So $O_i = \frac{1}{3}(V_{i1} + V_{i2} + V_{i3})$ for $i = 1, \ldots, 4$.

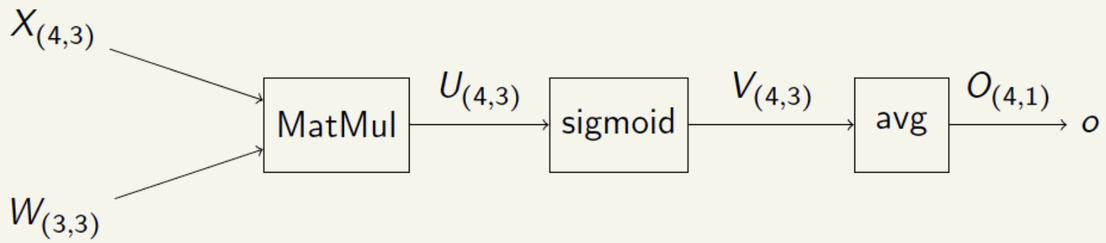


Derivative / backprop

- $\frac{\partial O_i}{\partial V_{ij}} = \frac{1}{3}$ for $j = 1, 2, 3$; $\frac{\partial O_k}{\partial V_{ij}} = 0$ if $k \neq i$.

TABLE OF CONTENTS

Summary: Chain rule of neural network



Question: What is the derivative of O with respect to W?

$$\frac{\partial O}{\partial W} = \frac{\partial U}{\partial W} \odot \frac{\partial V}{\partial U} \odot \frac{\partial O}{\partial V}$$

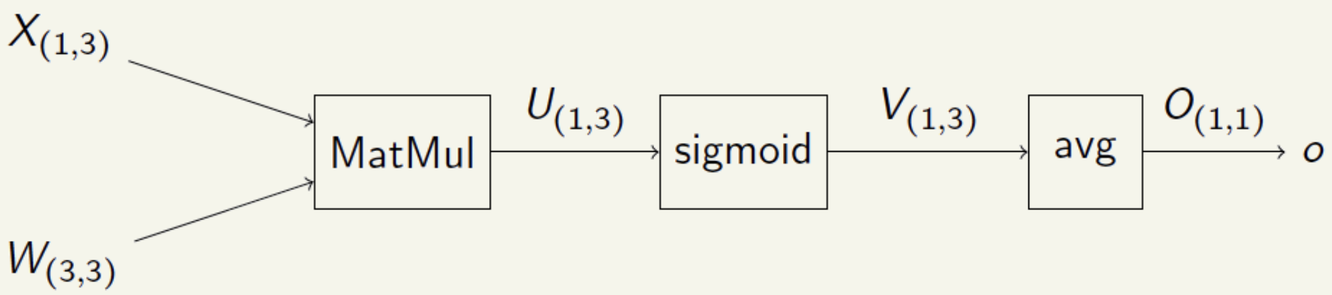
Same shape as W; $\odot$ denotes element-wise (Hadamard) multiplication.

- $\frac{\partial U}{\partial W} = X_{(3,4)}^T \cdot \mathbf{1}_{(4,3)}$
- $\frac{\partial V}{\partial U} = \sigma(U_{(4,3)}) \odot \left(\mathbf{1} - \sigma(U_{(4,3)})\right)$
- $\frac{\partial O}{\partial V} = \frac{1}{3} \cdot \mathbf{1}_{(4,3)}$ Why? (row-wise average: each V_{ij} in row i contributes $\frac{1}{3}$ to O_i.)

$$\frac{\partial O}{\partial W} = X_{(3,4)}^T \cdot \left[\sigma(U_{(4,3)}) \odot \left(\mathbf{1} - \sigma(U_{(4,3)})\right) \odot \mathbf{1}_{(4,3)} \odot \frac{1}{3}\right]$$

Exercise

For the following network there is **one** house, so the input is a **row vector** $X_{(1,3)}$.



- Given $X_{(1,3)} = \begin{bmatrix} 0.5 & 0.3 & 0.2 \end{bmatrix}$ and $W_{(3,3)} = \begin{bmatrix} 0.1 & 0.4 & 0.3 \\ 0.2 & 0.5 & 0.2 \\ 0.3 & 0.2 & 0.4 \end{bmatrix}$
- Compute $U_{(1,3)}$, $V_{(1,3)}$, and $O_{(1,1)}$.
- Write the chain rule for $\frac{\partial O}{\partial W}$ and compute the derivative.
- **Numeric check:** verify for one entry, e.g. $w_{1,1}$, using $\frac{\partial O}{\partial w_{1,1}} \approx \frac{O(w_{1,1} + \epsilon) - O(w_{1,1})}{\epsilon}$.

Answer

- **Linear layer:** $U = XW$:

$$U_{(1,3)} = \begin{bmatrix} 0.5 & 0.3 & 0.2 \end{bmatrix} \begin{bmatrix} 0.1 & 0.4 & 0.3 \\ 0.2 & 0.5 & 0.2 \\ 0.3 & 0.2 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.17 & 0.39 & 0.29 \end{bmatrix}$$

- **Sigmoid:** $V = \sigma(U)$:

$$V_{(1,3)} = \begin{bmatrix} 0.542 & 0.596 & 0.572 \end{bmatrix}$$

- **Average:** $O_{(1,1)} = \frac{1}{3}(V_{11} + V_{12} + V_{13})$:

$$O_{(1,1)} = \frac{0.542 + 0.596 + 0.572}{3} = 0.570$$

- **Local factors** (same shapes as in the summary, with batch size 1):

$$\frac{\partial U}{\partial W} = X^T_{(3,1)} \cdot \mathbf{1}_{(1,3)}, \qquad \frac{\partial V}{\partial U} = \sigma(U) \odot (\mathbf{1} - \sigma(U)), \qquad \frac{\partial O}{\partial V} = \frac{1}{3}\mathbf{1}_{(1,3)}$$

- **Chain into** W:

$$\frac{\partial O}{\partial W} = \begin{bmatrix} 0.5 \\ 0.3 \\ 0.2 \end{bmatrix} \begin{bmatrix} 0.0827 & 0.0803 & 0.0817 \end{bmatrix} = \begin{bmatrix} 0.0414 & 0.0402 & 0.0409 \\ 0.0248 & 0.0241 & 0.0245 \\ 0.0165 & 0.0161 & 0.0163 \end{bmatrix}$$

- **Numeric check** for $w_{1,1}$ with $\epsilon = 10^{-4}$: $\dfrac{\partial O}{\partial w_{1,1}} \approx \dfrac{O(w_{1,1} + \epsilon) - O(w_{1,1})}{\epsilon} \approx 0.0414$.

Thank You!