

Week 5: Neural Networks

Matthew Caldwell

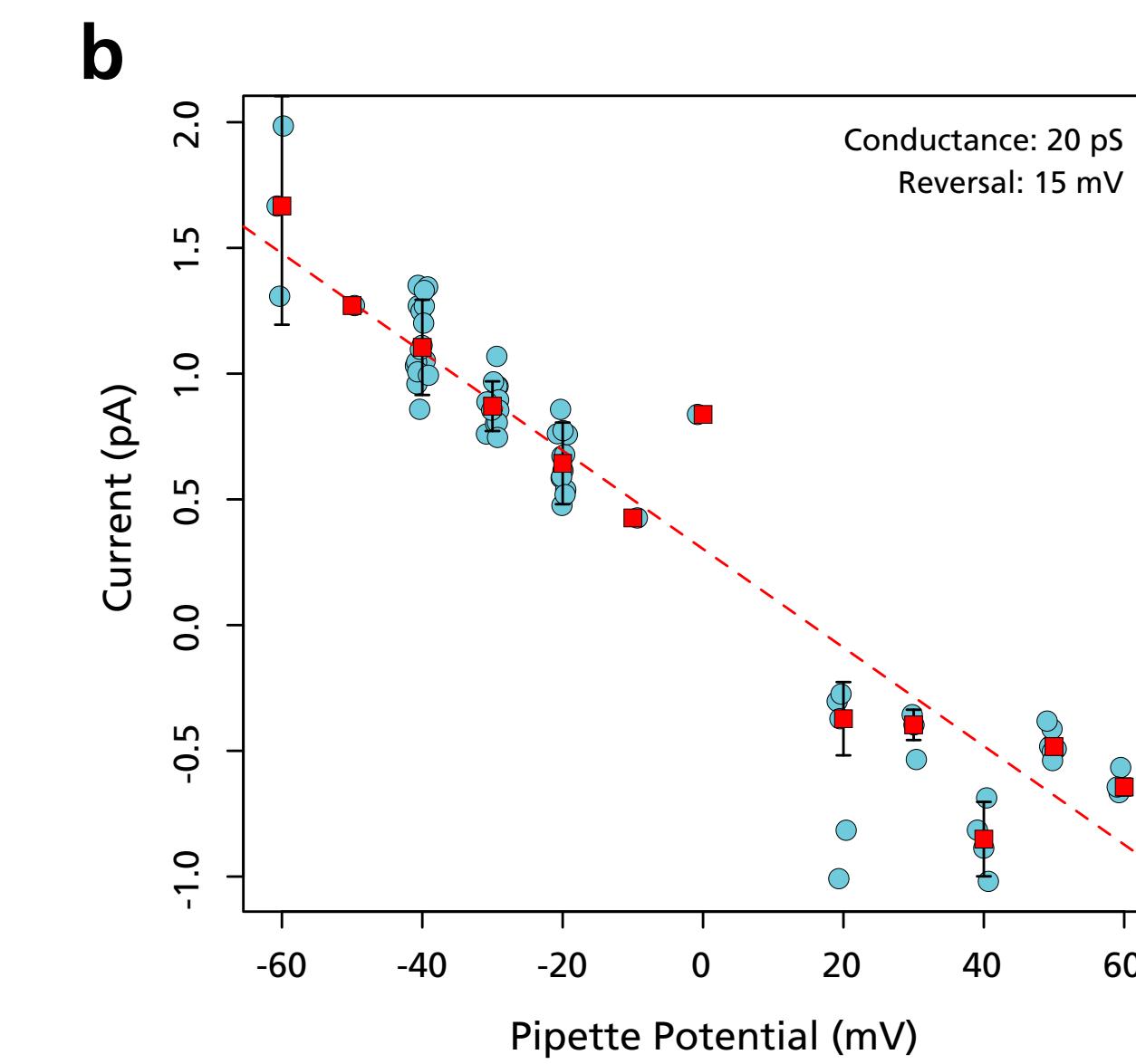
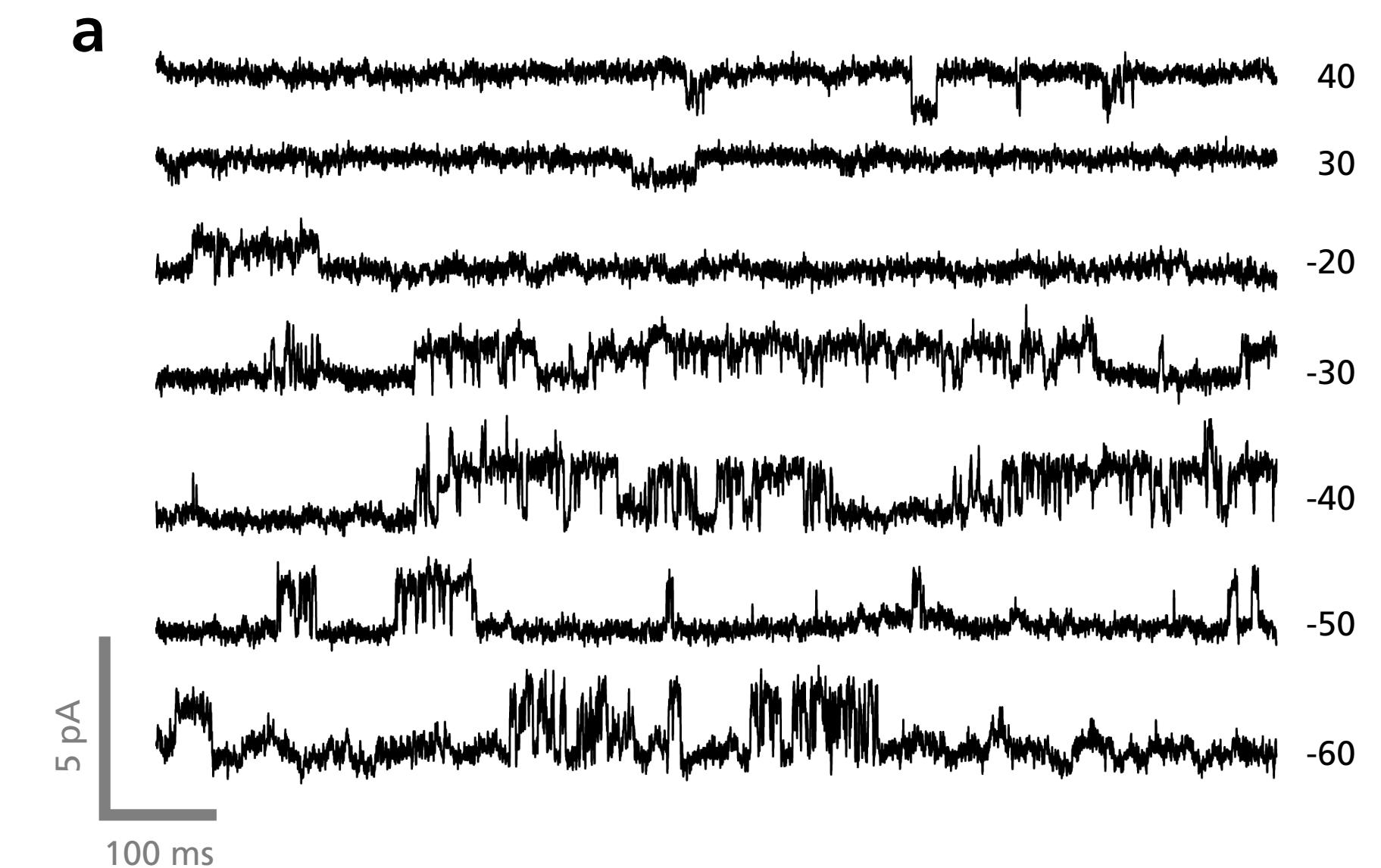
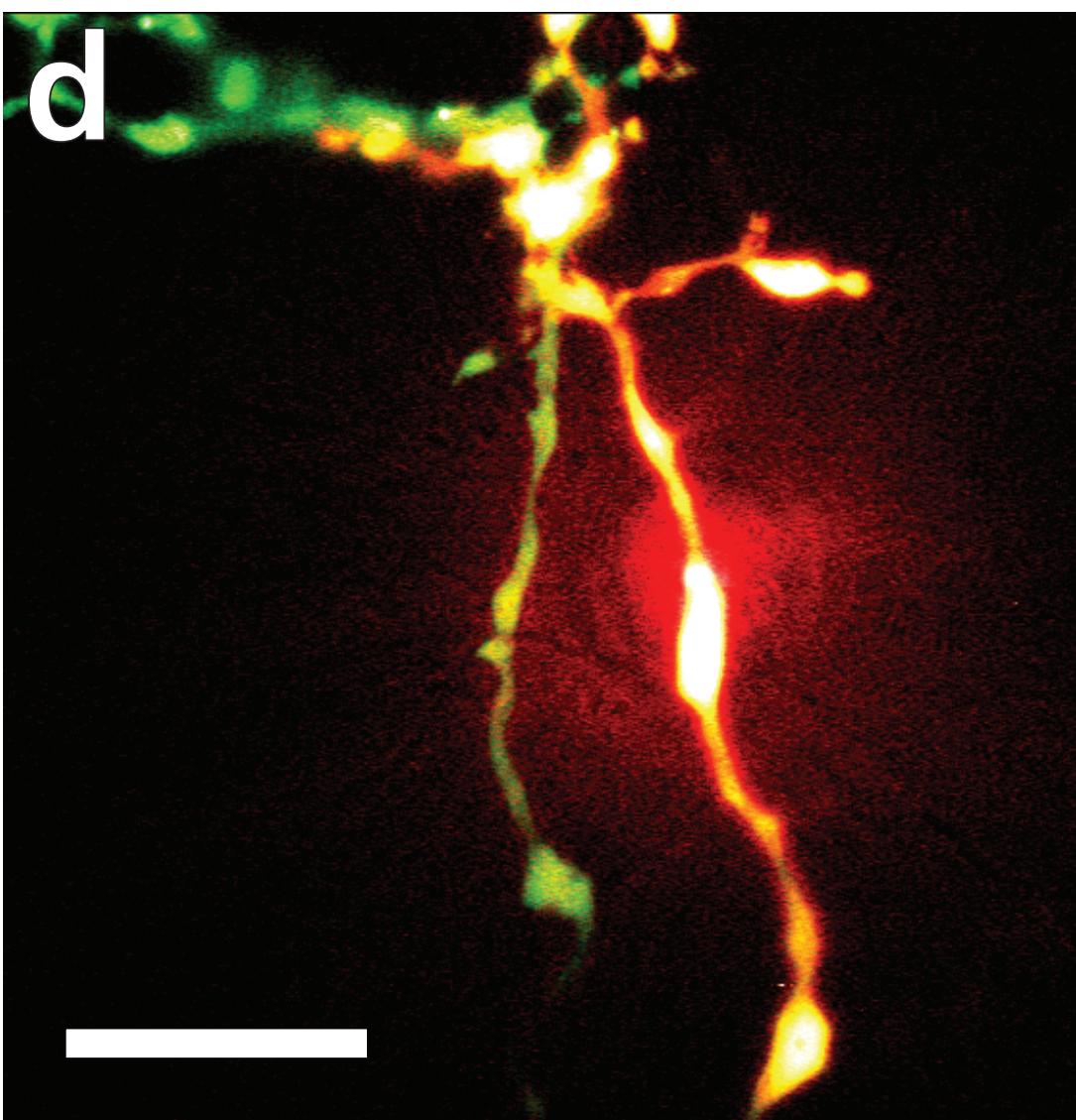
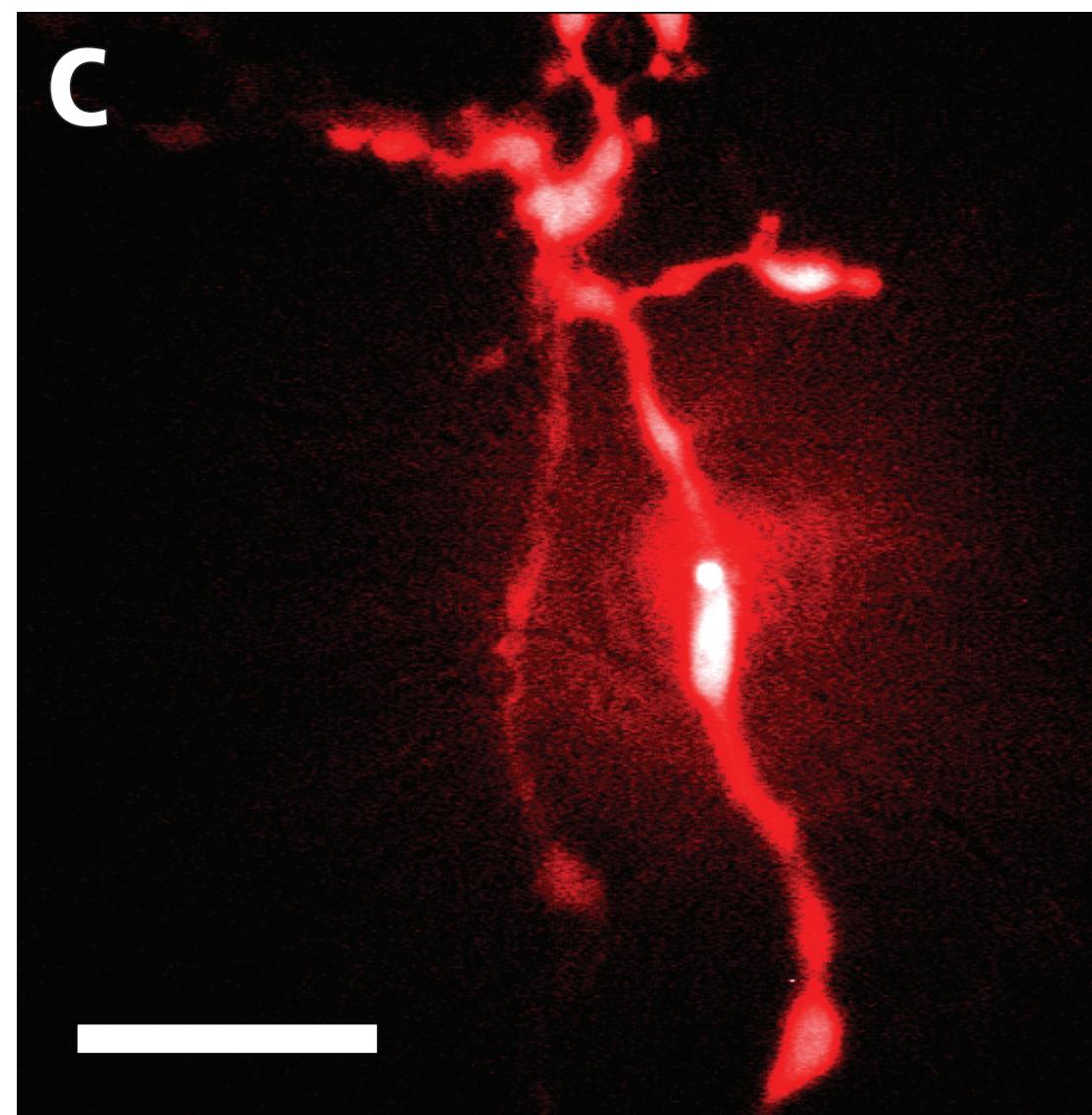
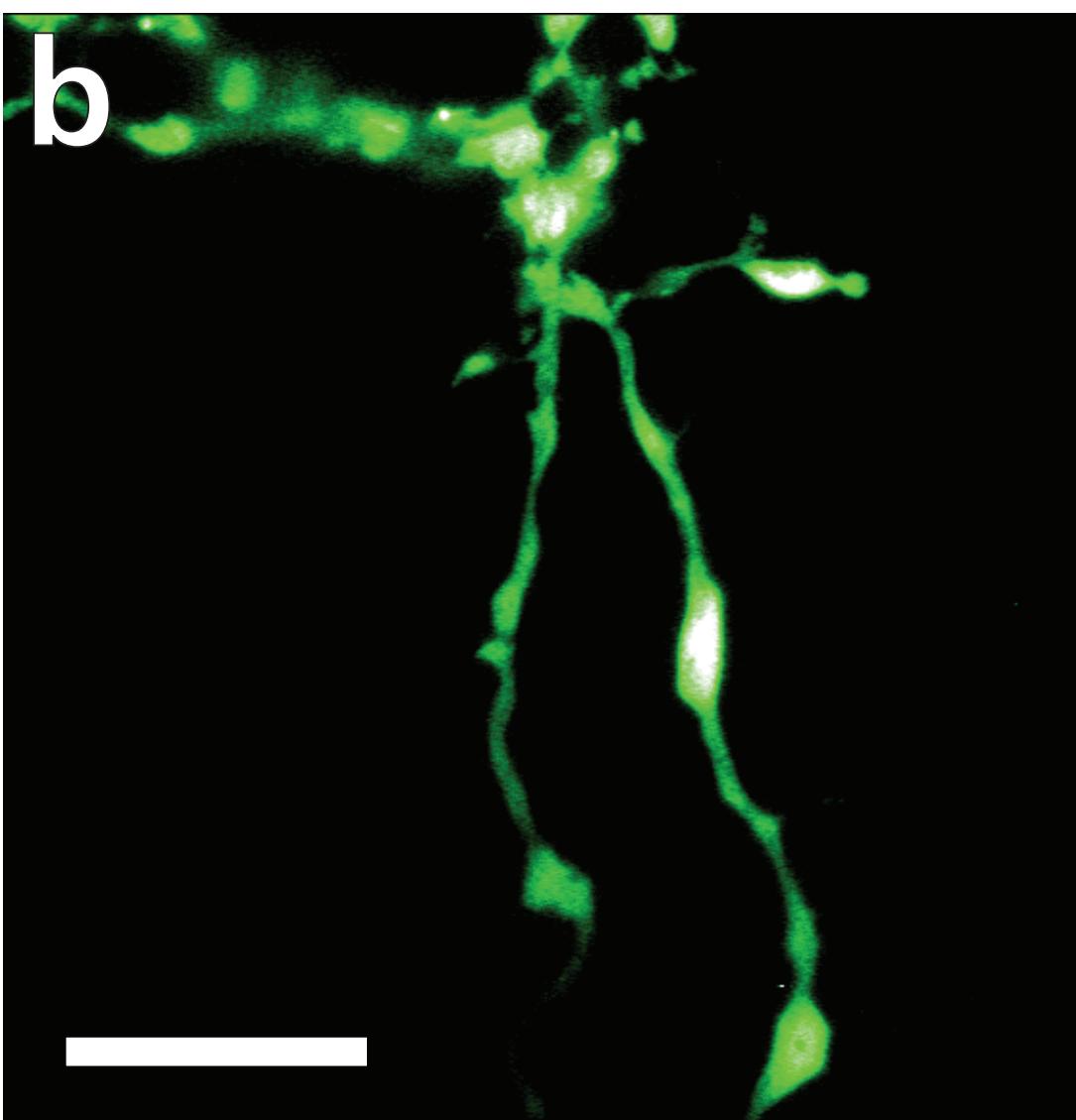
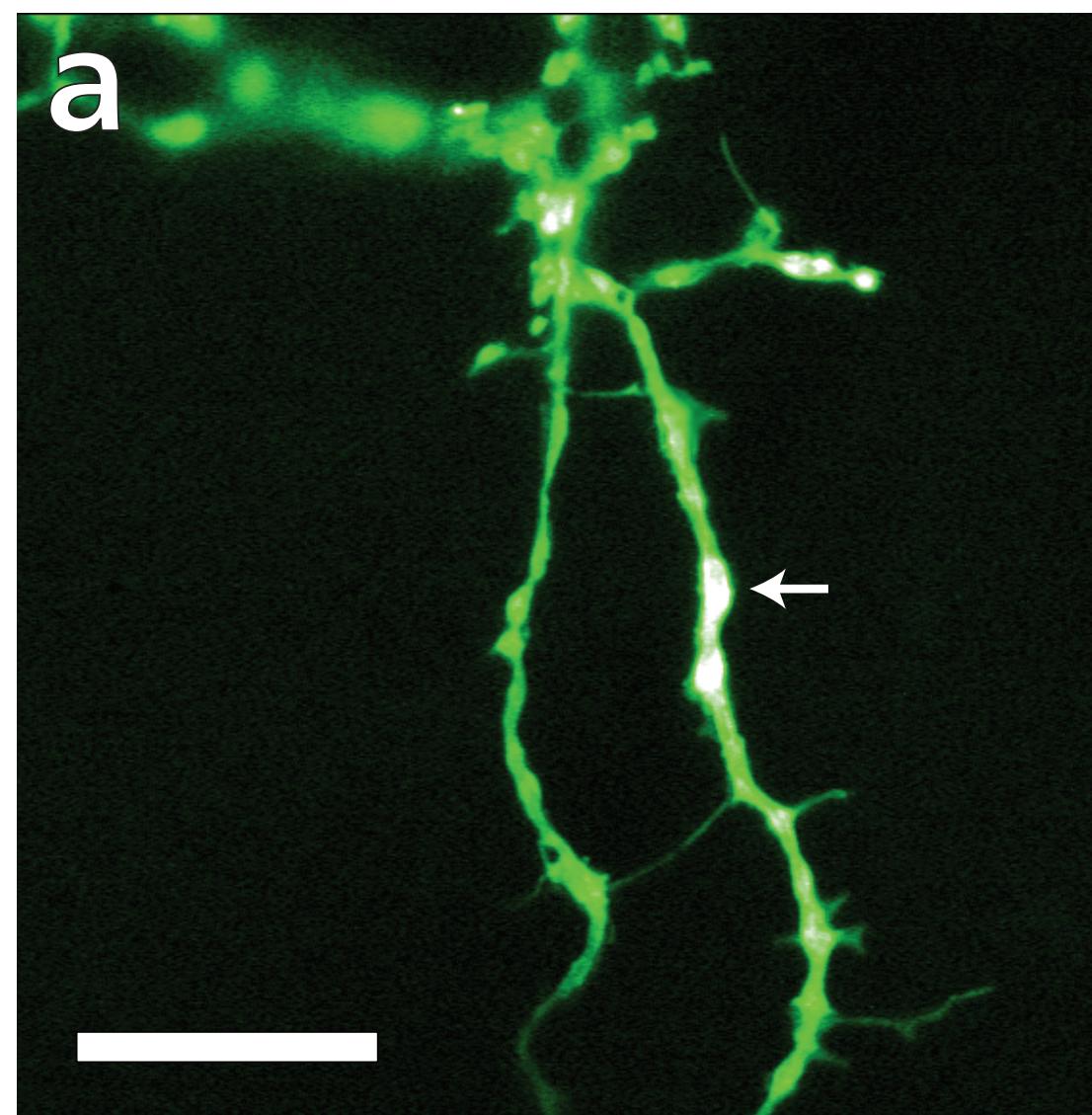
COMP0088 Introduction to Machine Learning • UCL Computer Science • Autumn 2025

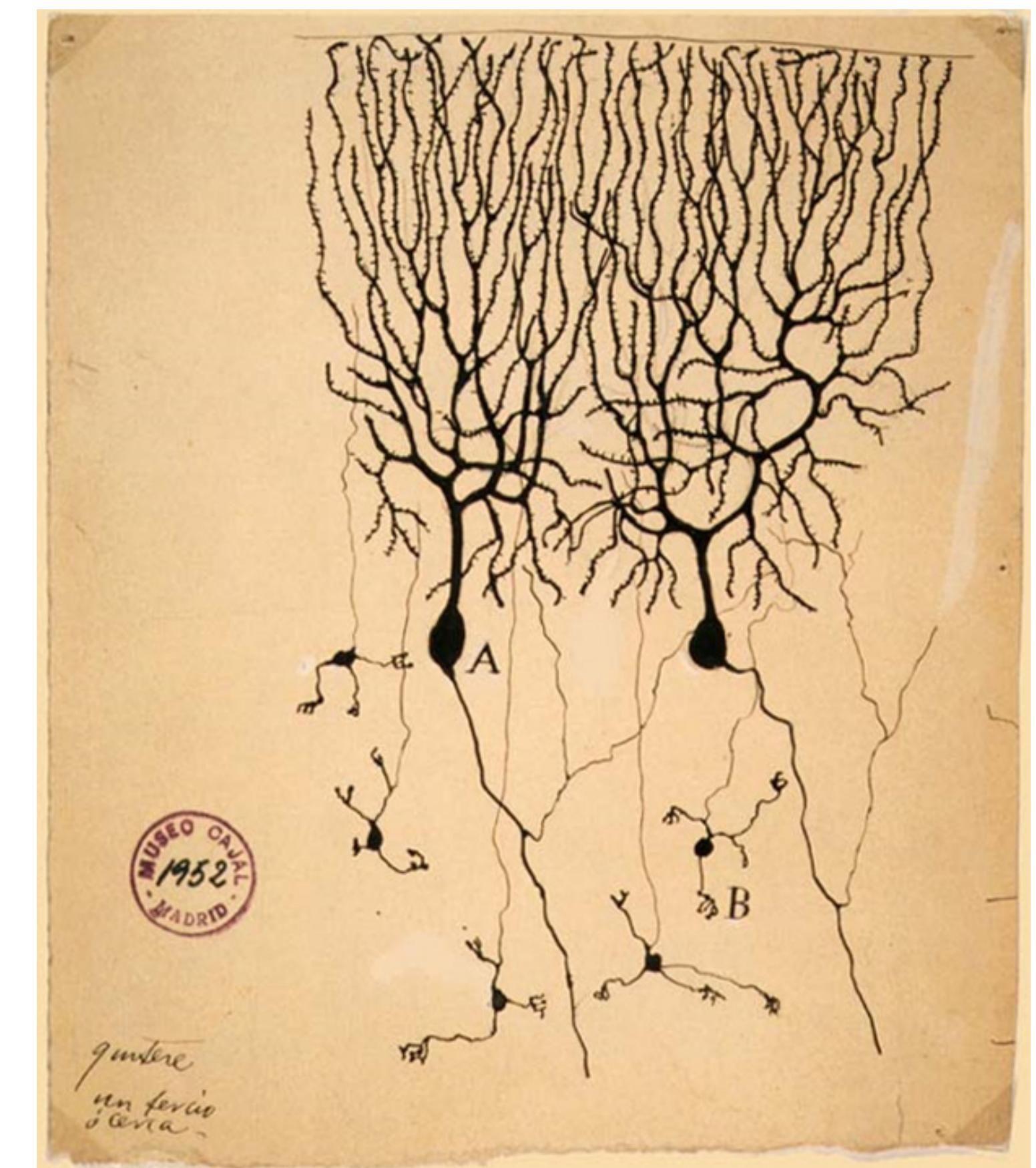
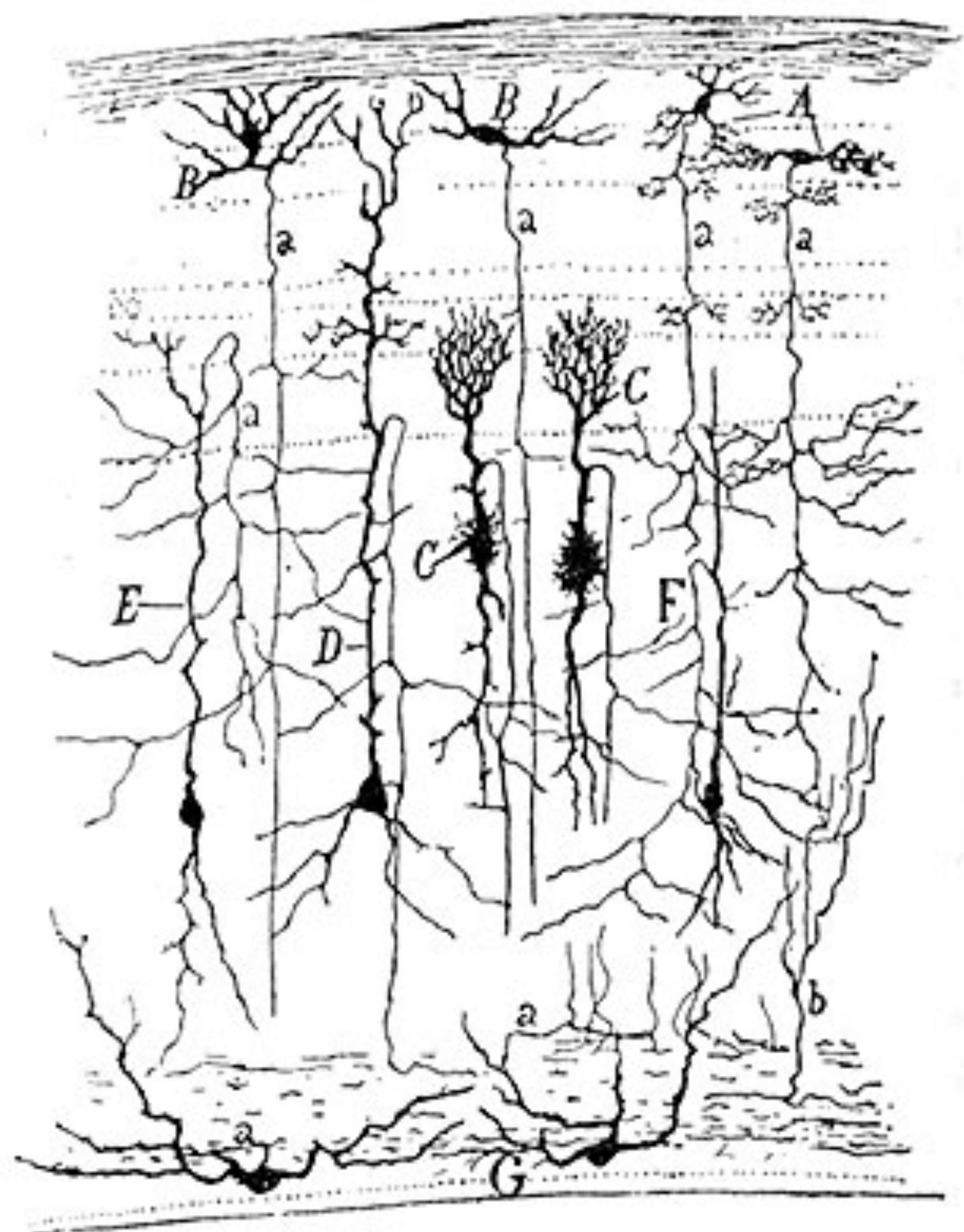
Admin

- Pulse surveys
- Coursework
- No labs, lectures or office hour next week
- New lecture venue after reading week, check timetable

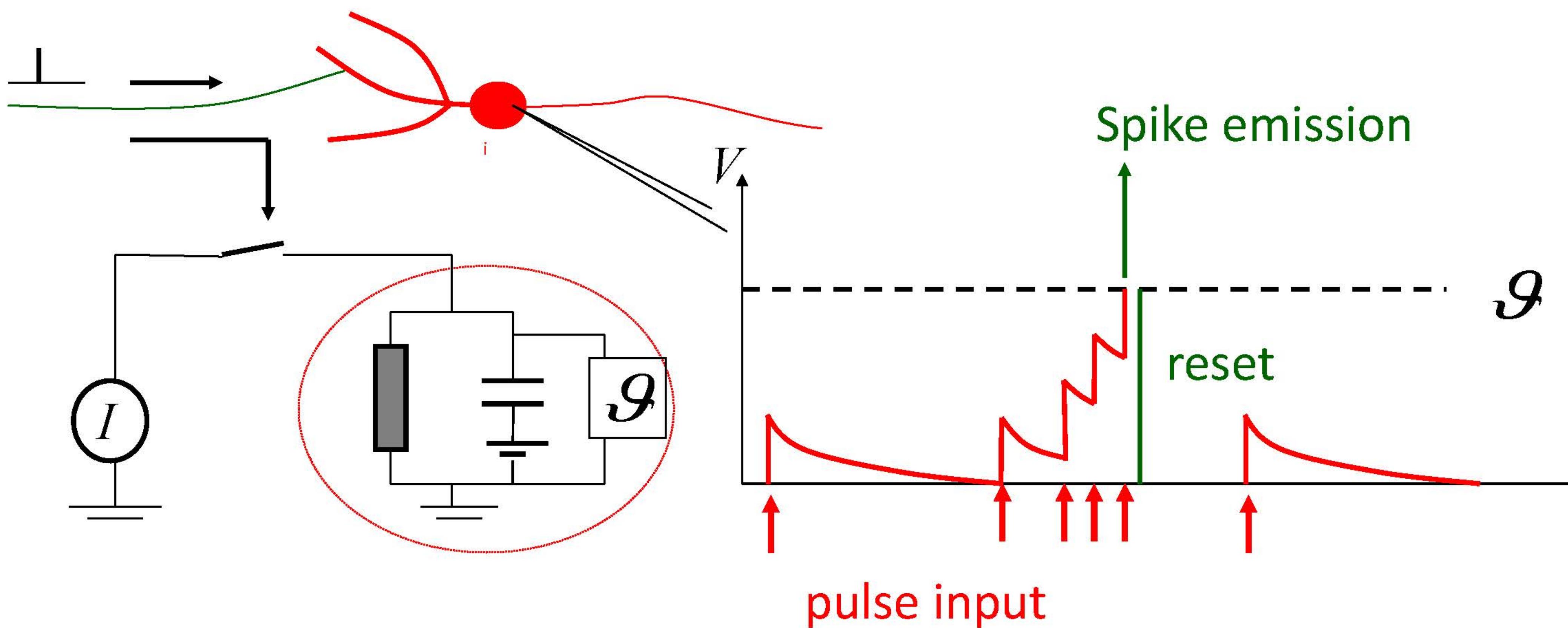
Week 5 Recap

Safety in Numbers





Leaky Integrate-and-Fire Model



$$\text{fire} = \begin{cases} 1 & \text{if } \mathbf{x} \cdot \mathbf{w} \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases}$$

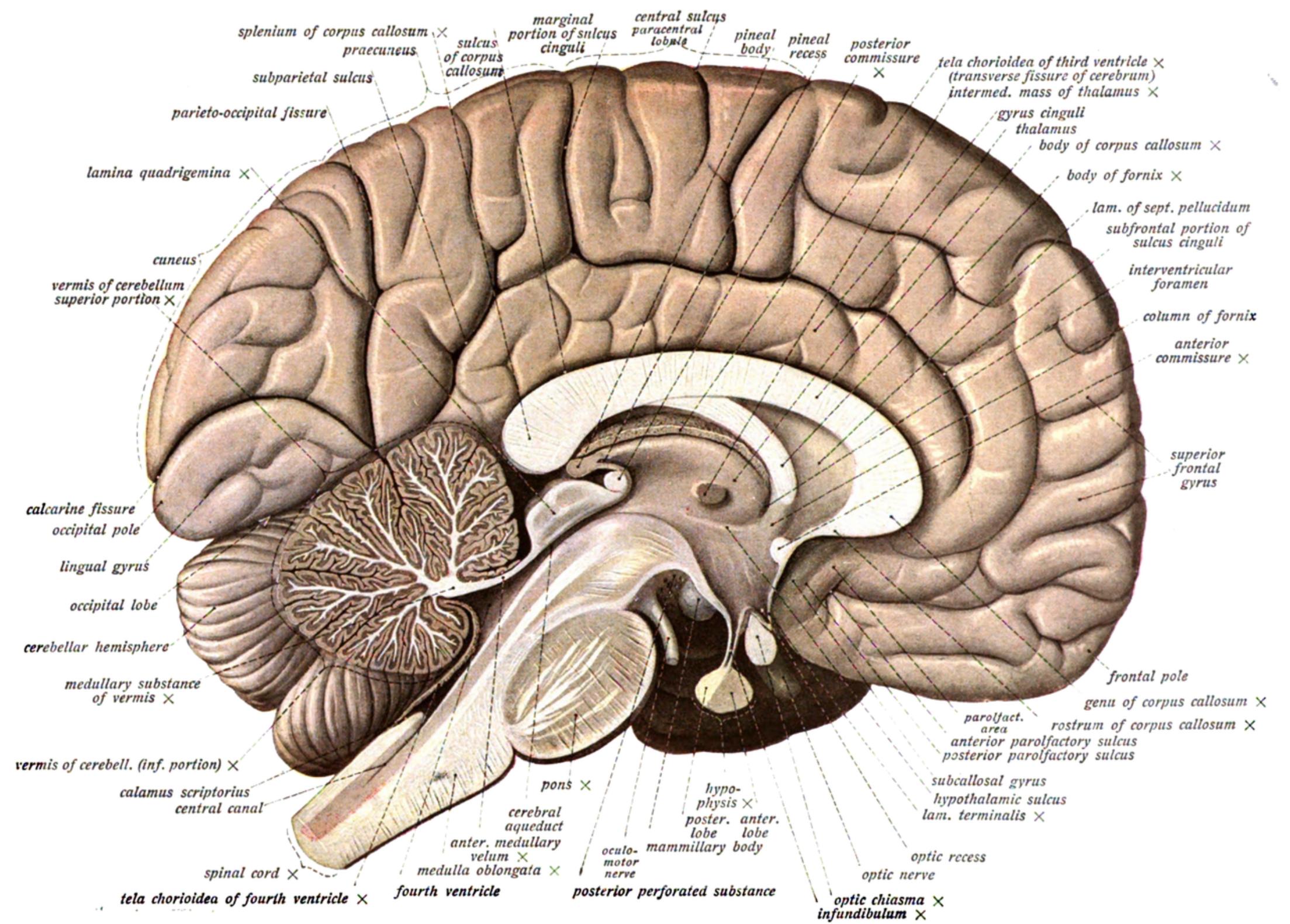
$$\text{fire} = \begin{cases} 1 & \text{if } \mathbf{x} \cdot \mathbf{w} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{fire} = \begin{cases} 1 & \text{if } \mathbf{x} \cdot \mathbf{w} \geq 0 \\ 0 & \text{otherwise} \end{cases} \neq$$

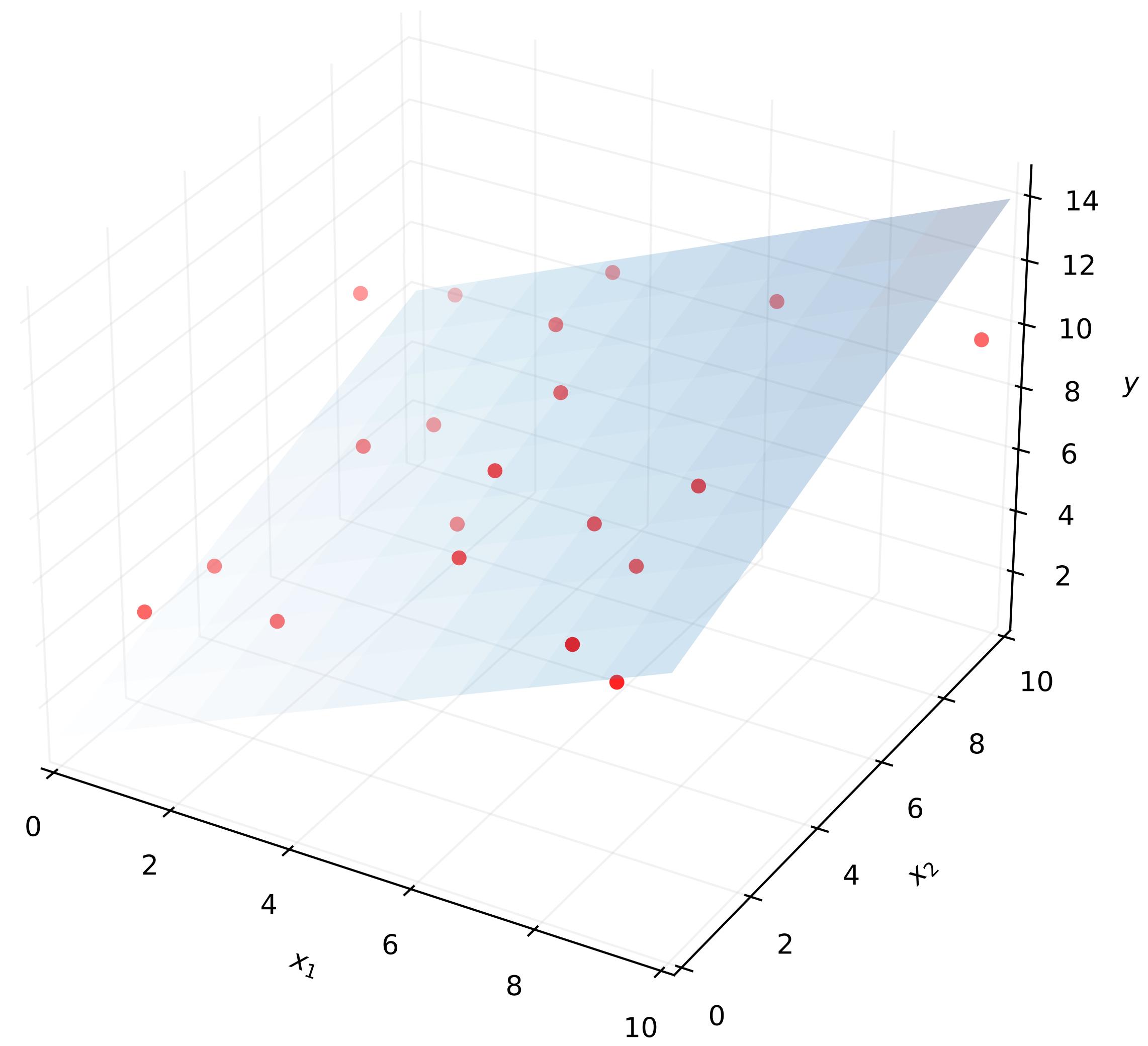
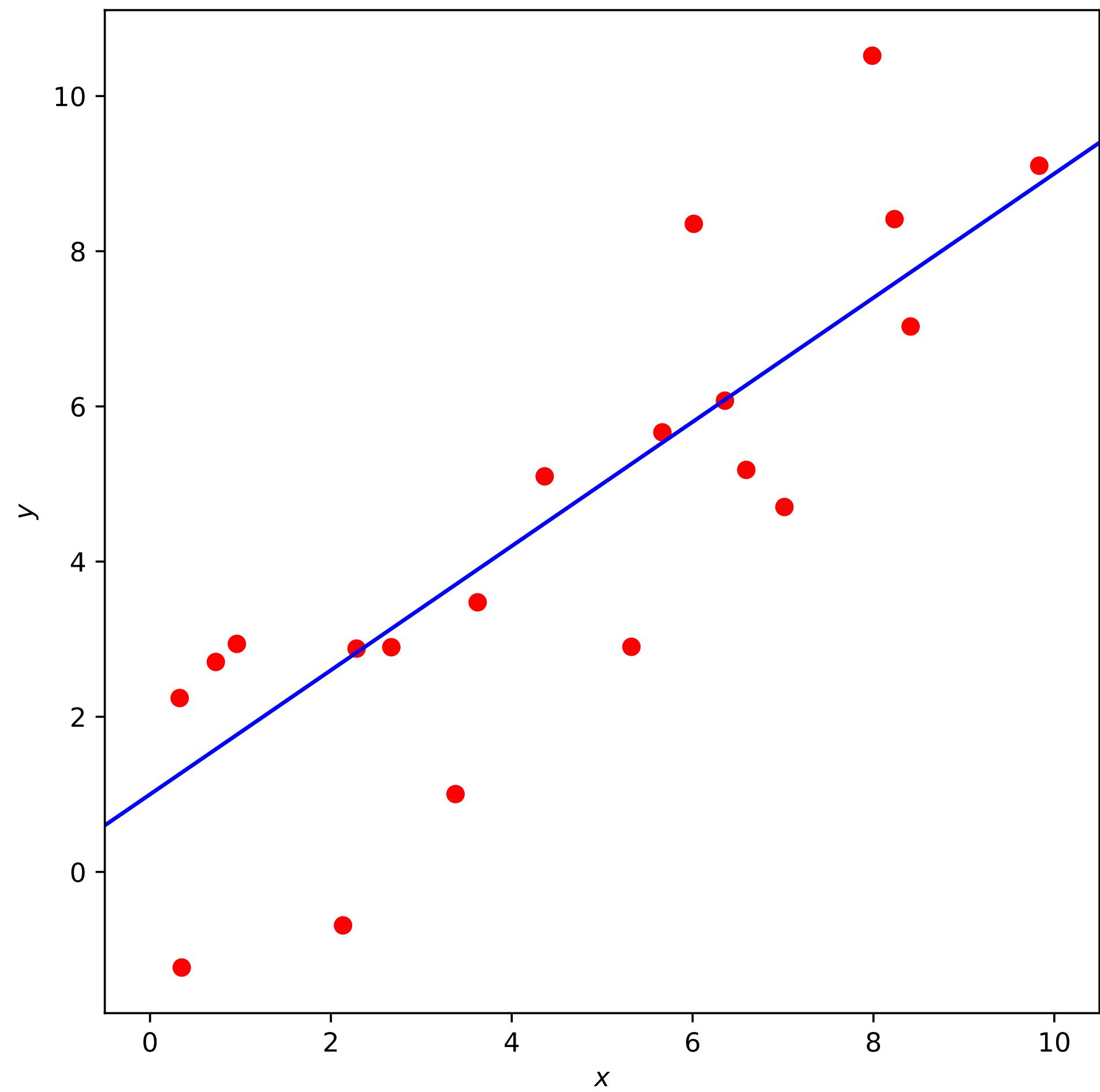




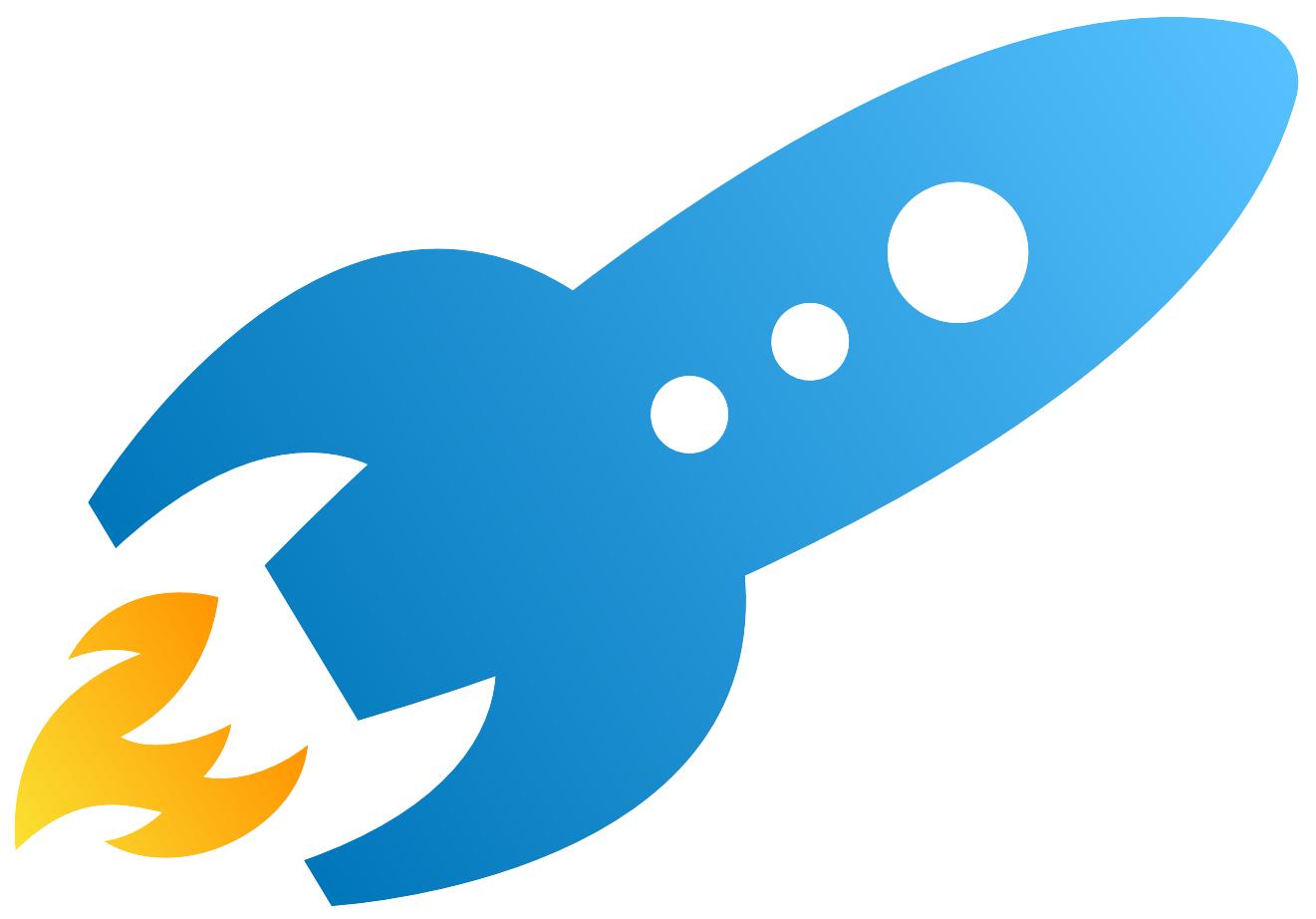
≠



$$\text{fire} = \begin{cases} 1 & \text{if } \mathbf{x} \cdot \mathbf{w} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

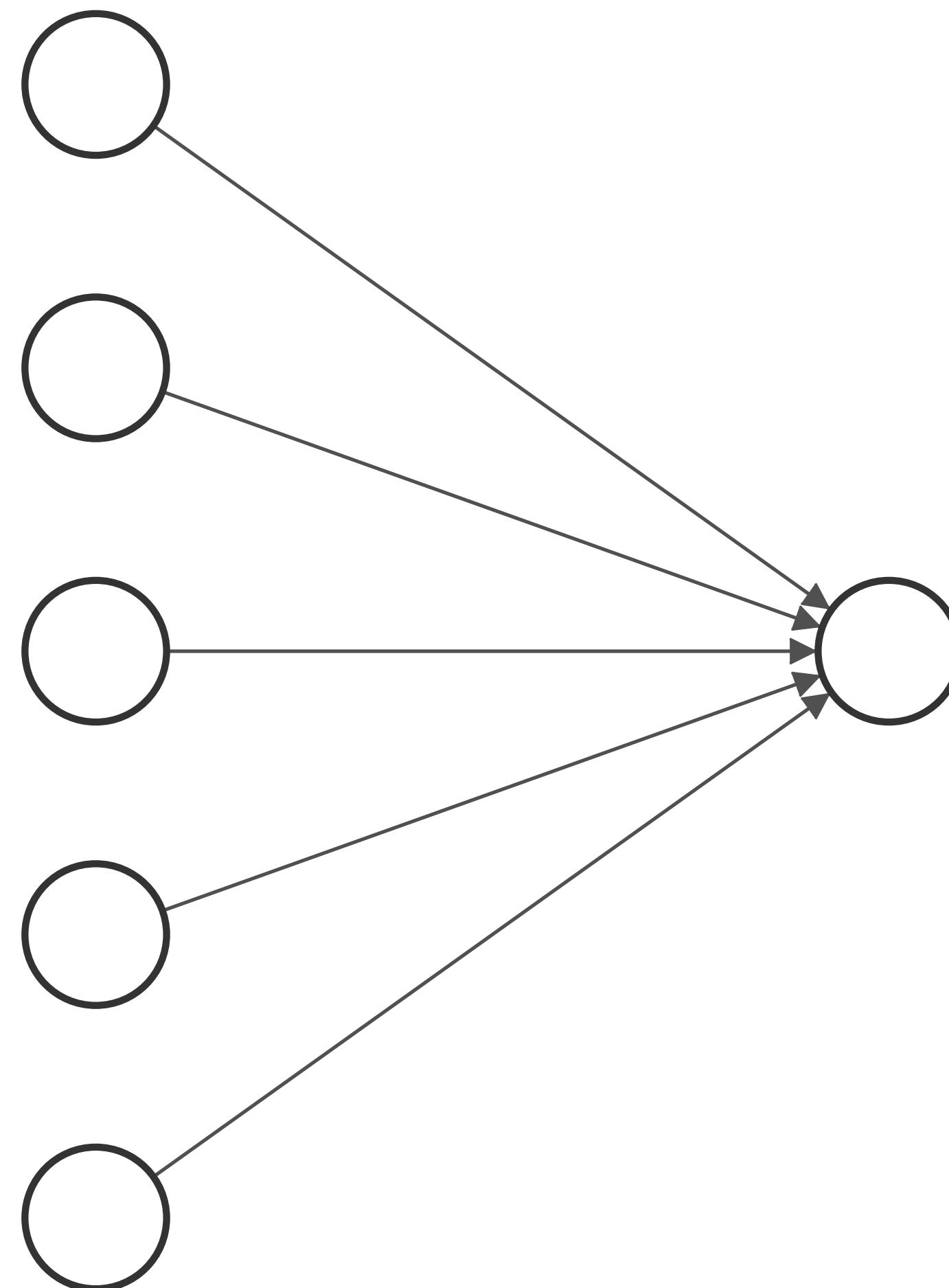




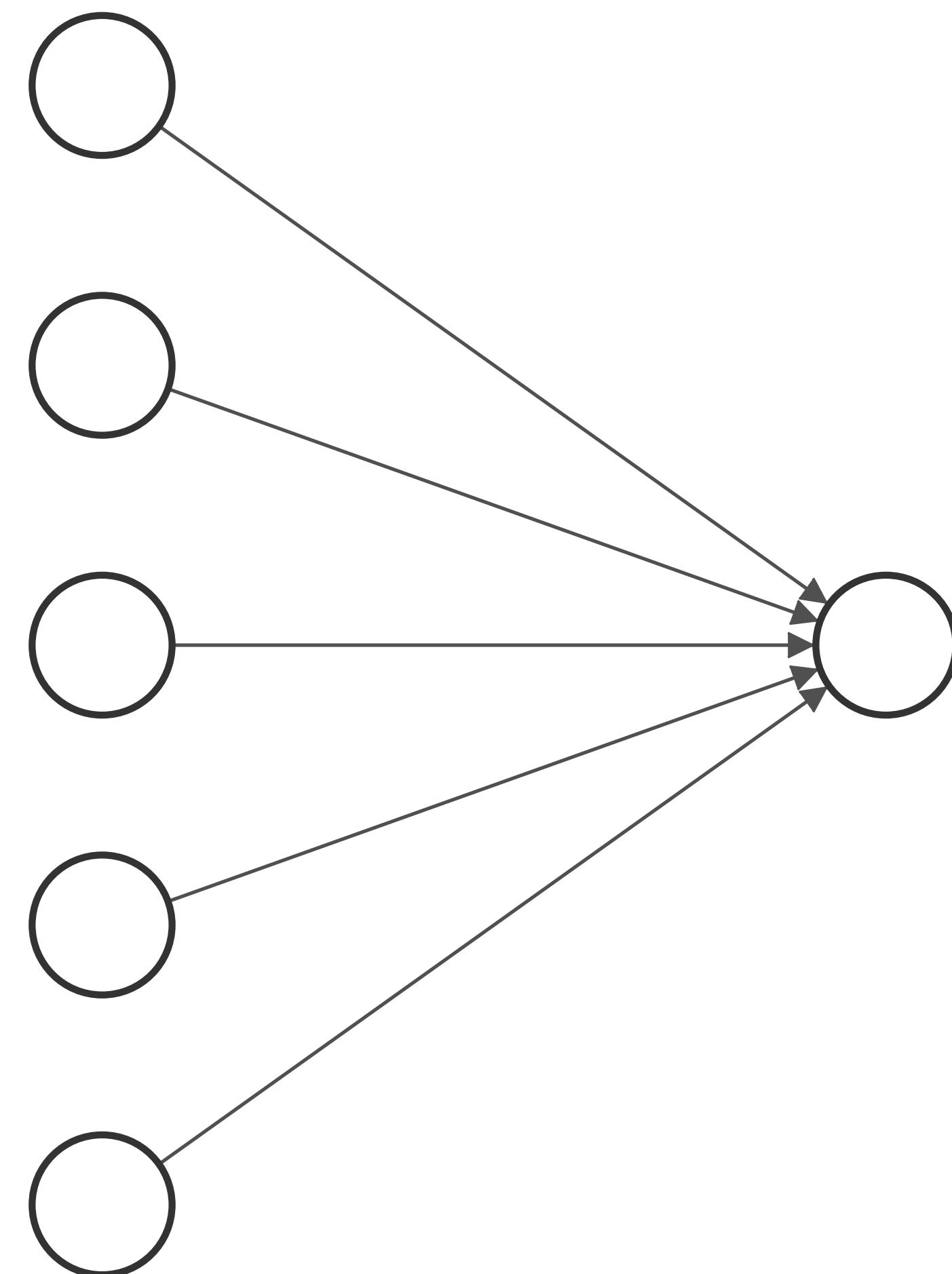


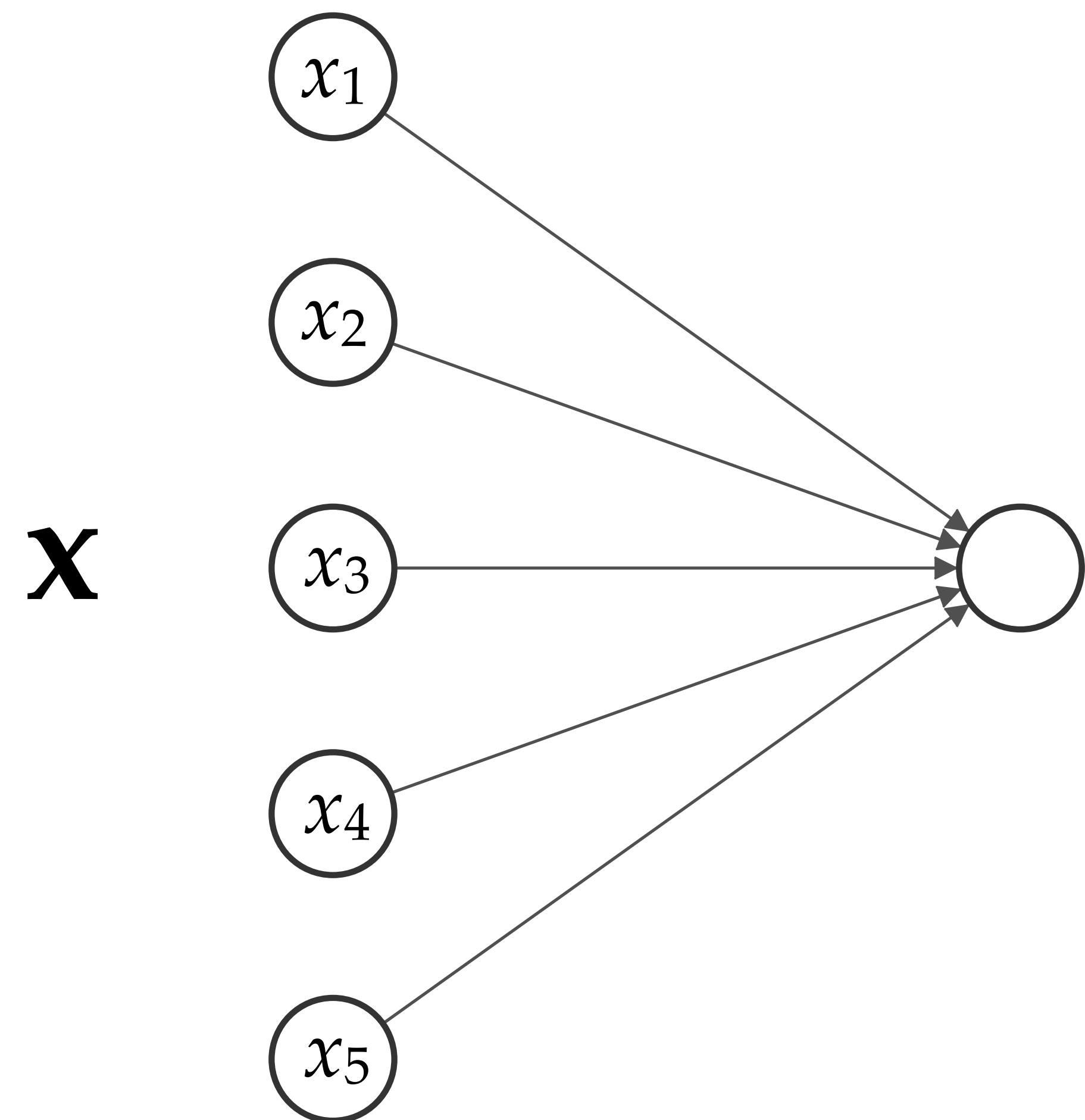
$$\mathbf{x} \cdot \mathbf{w} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

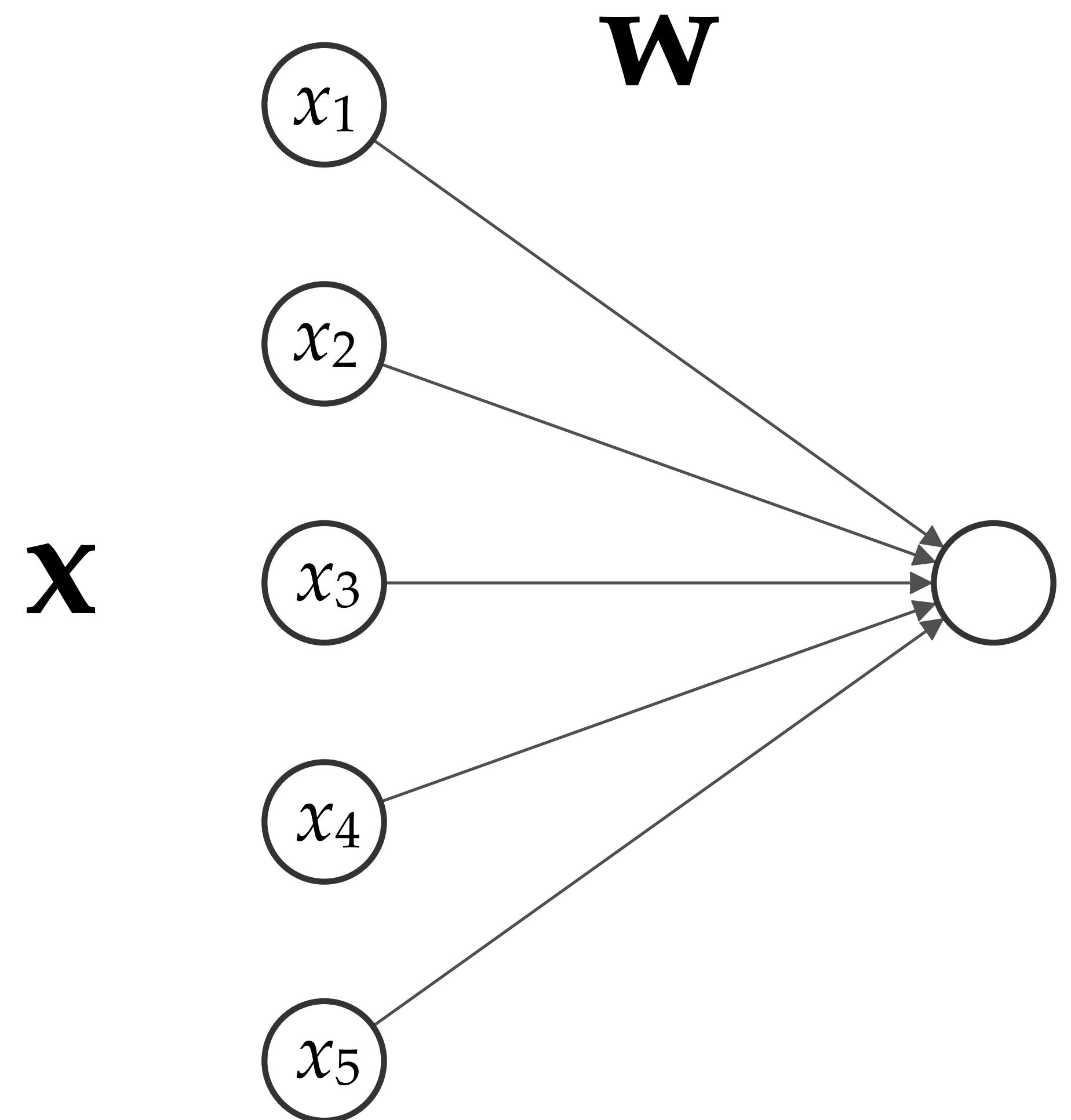
$$\mathbf{x}^\top \mathbf{w} = [x_1 \quad x_2 \quad \dots \quad x_d] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

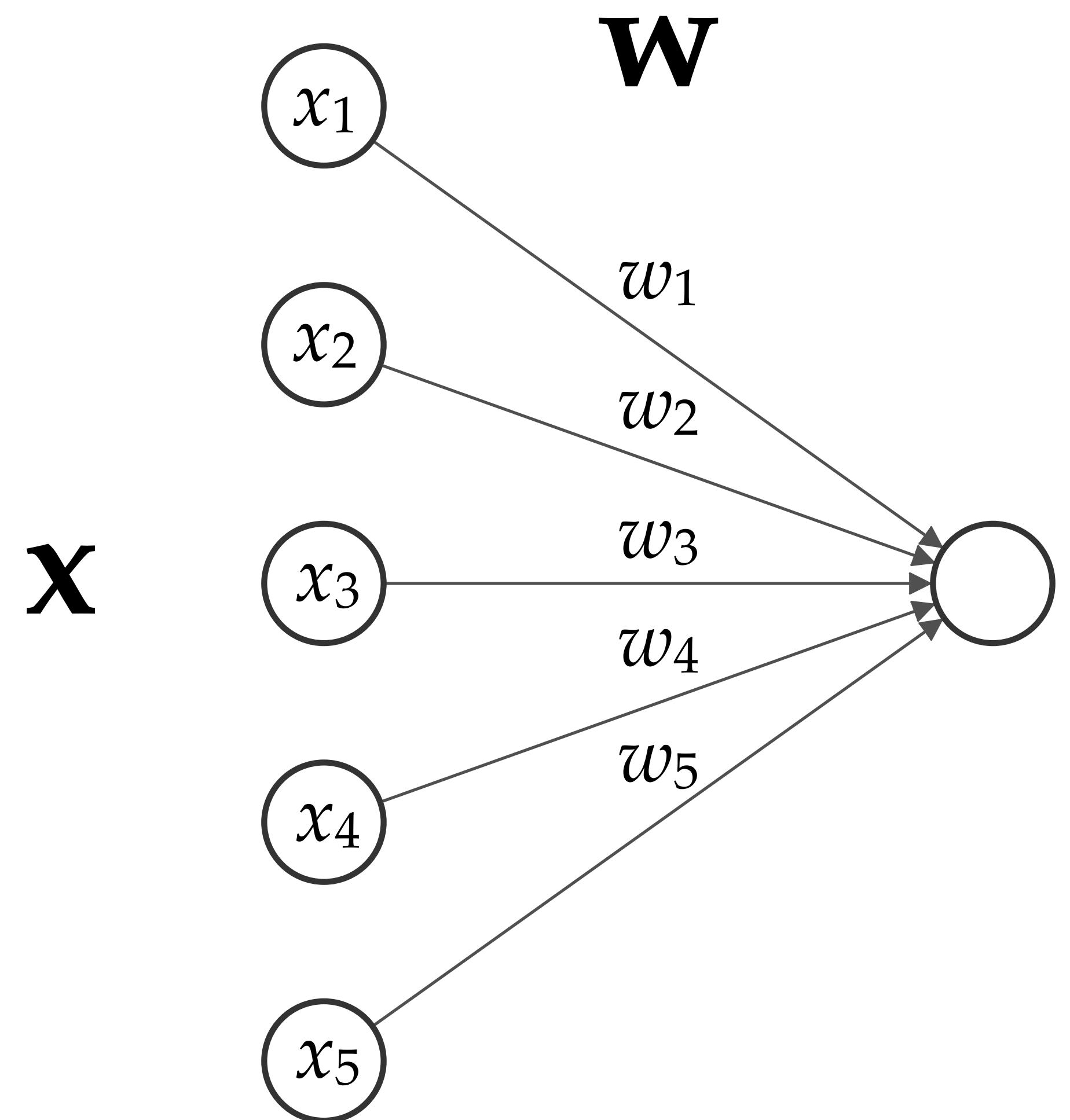


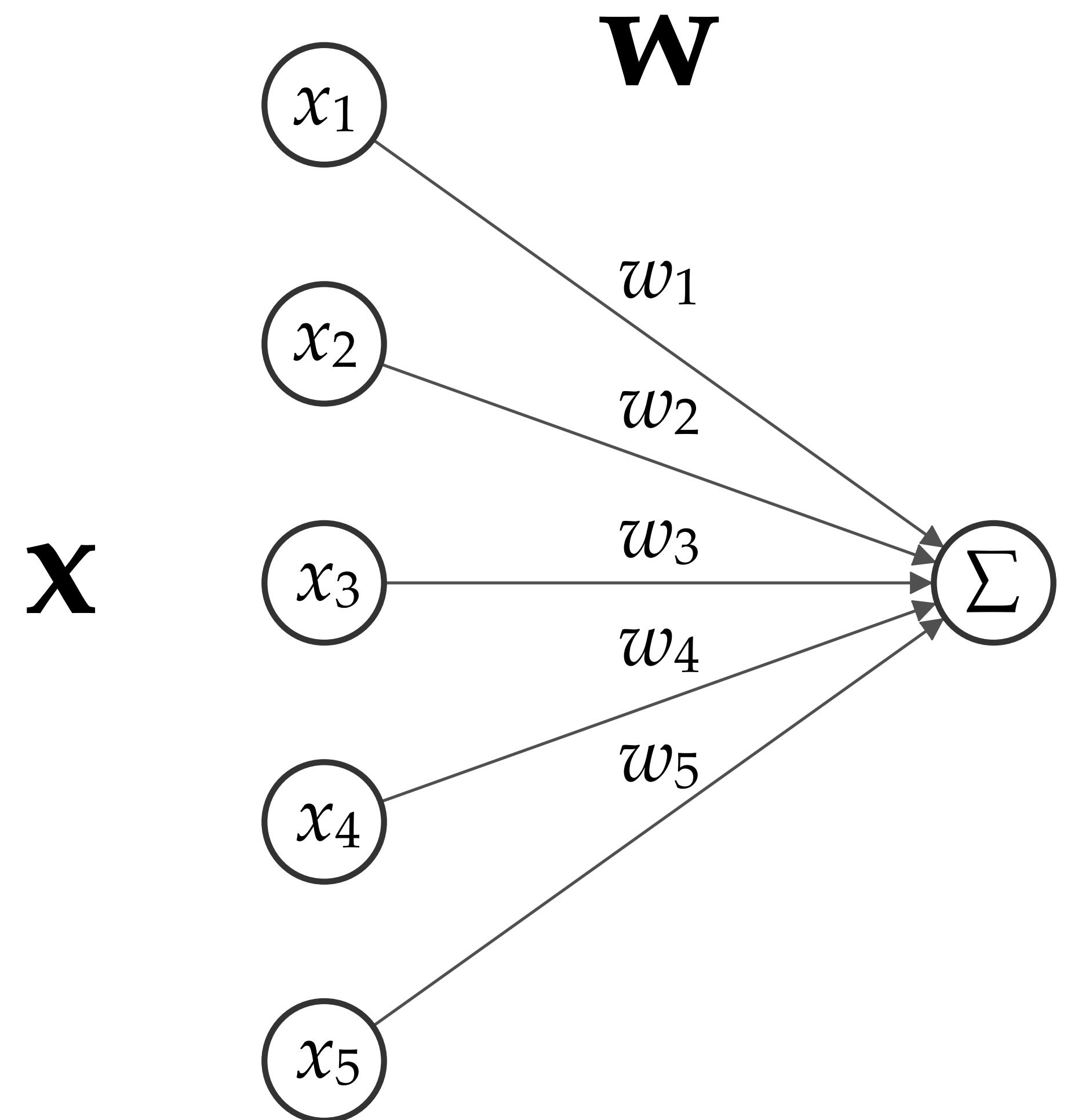
x

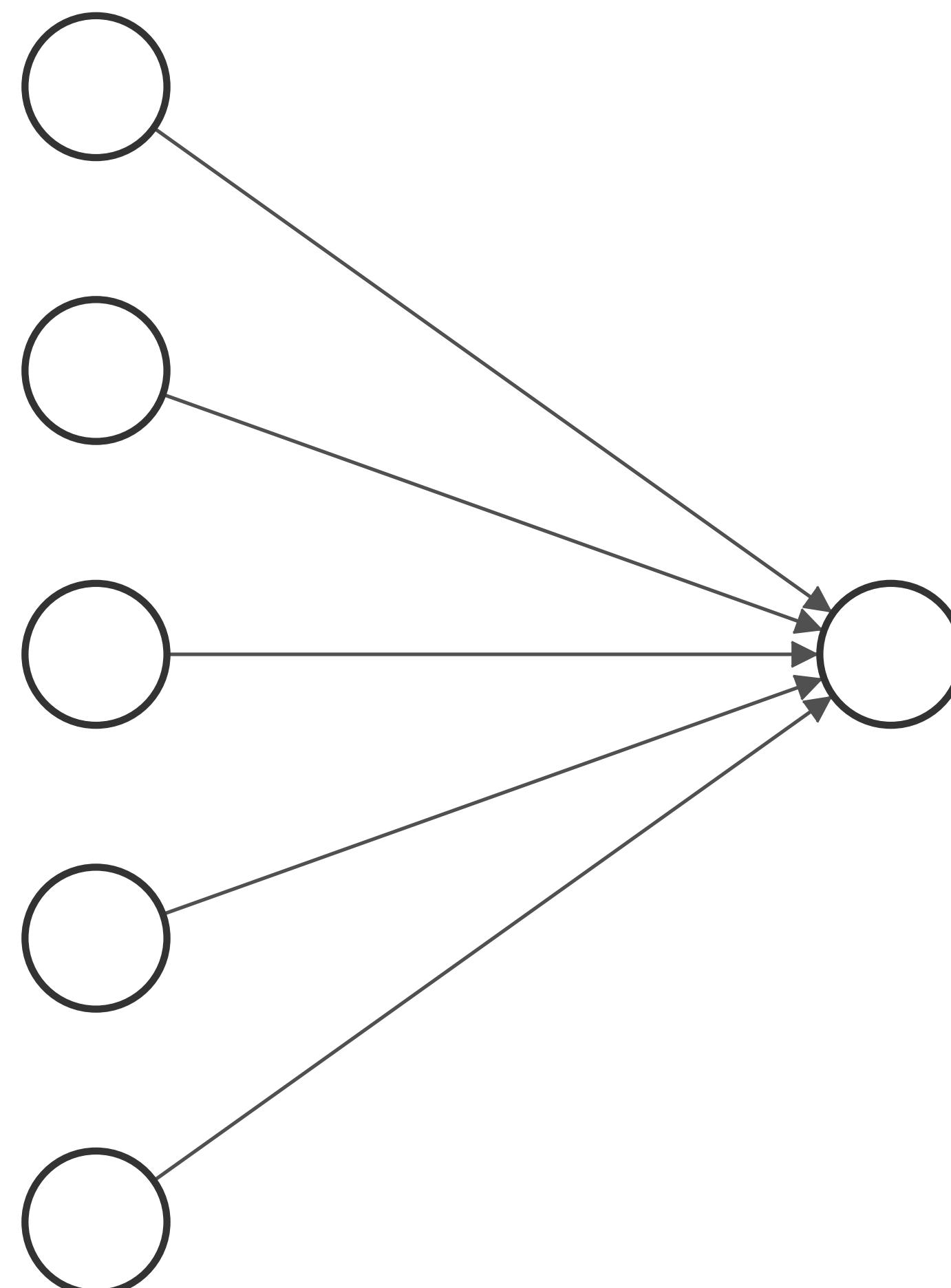




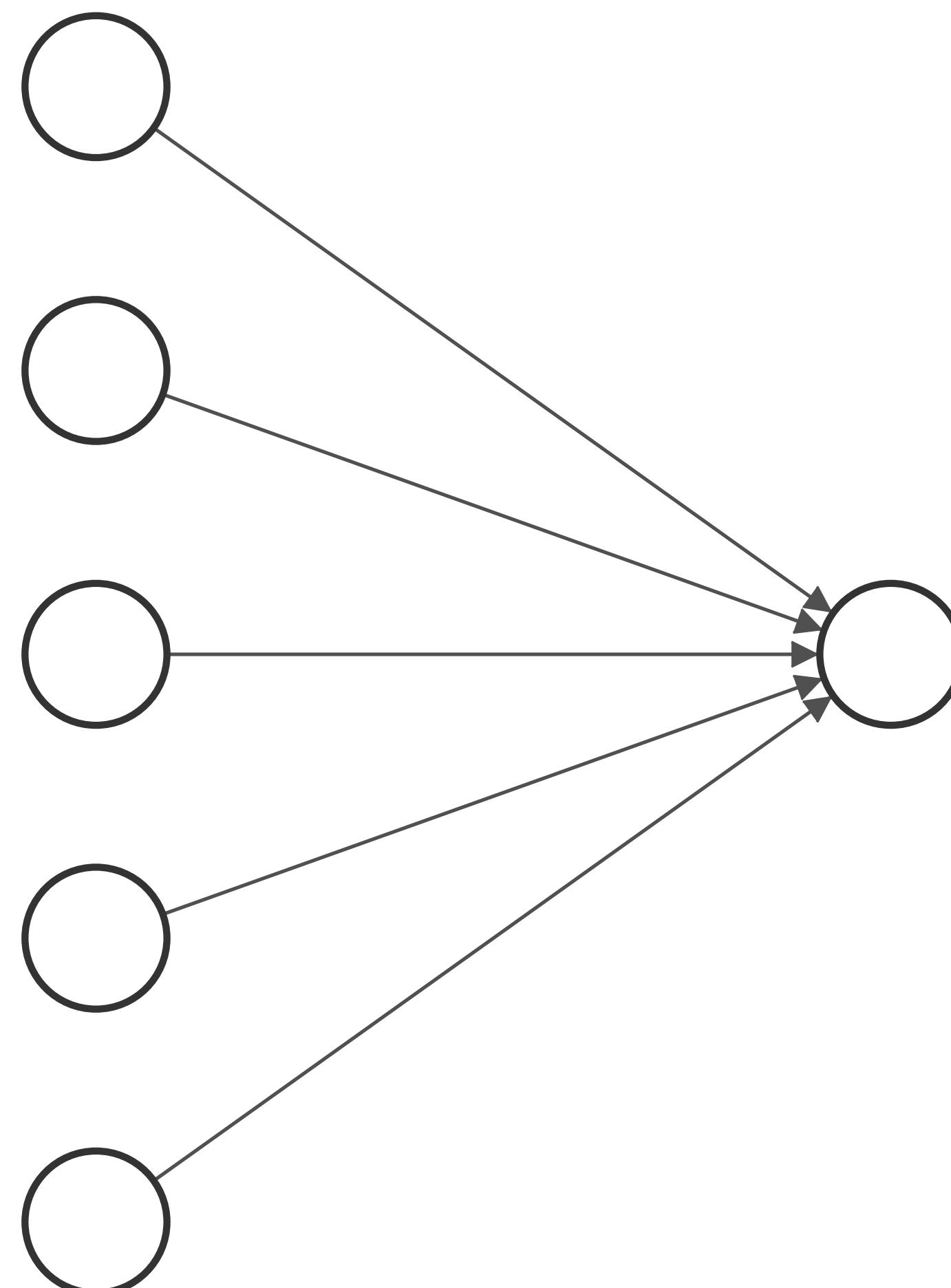


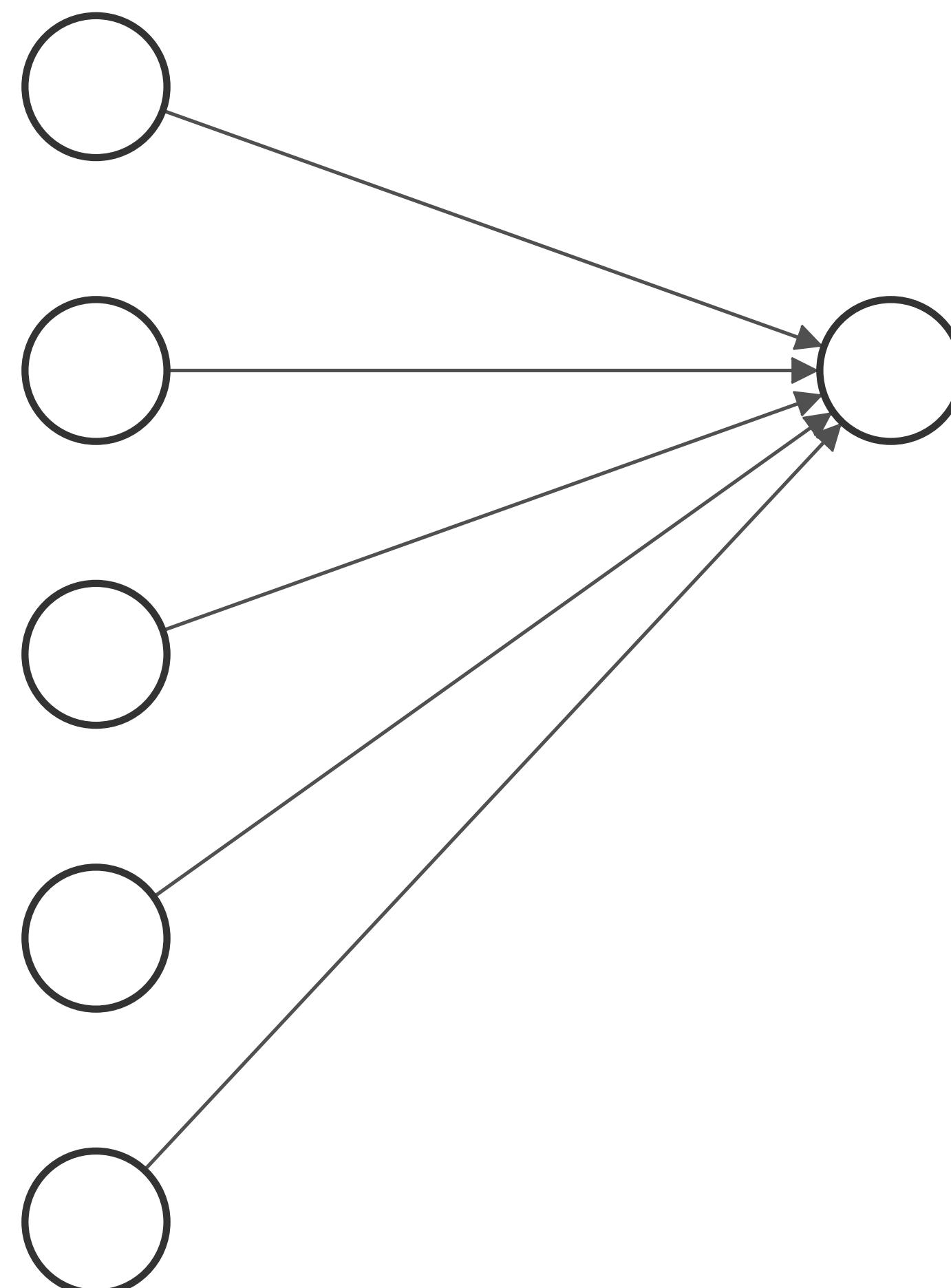


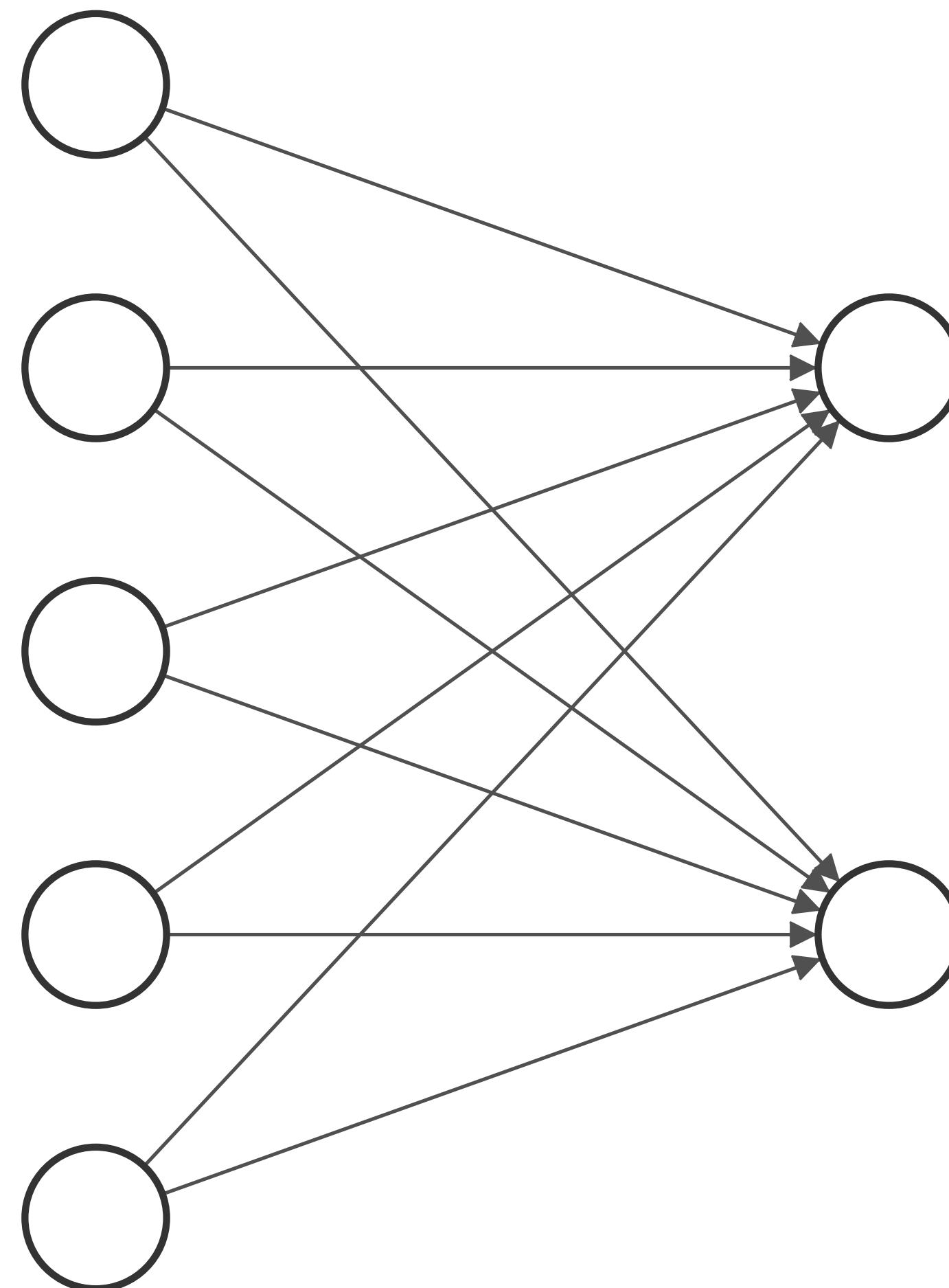




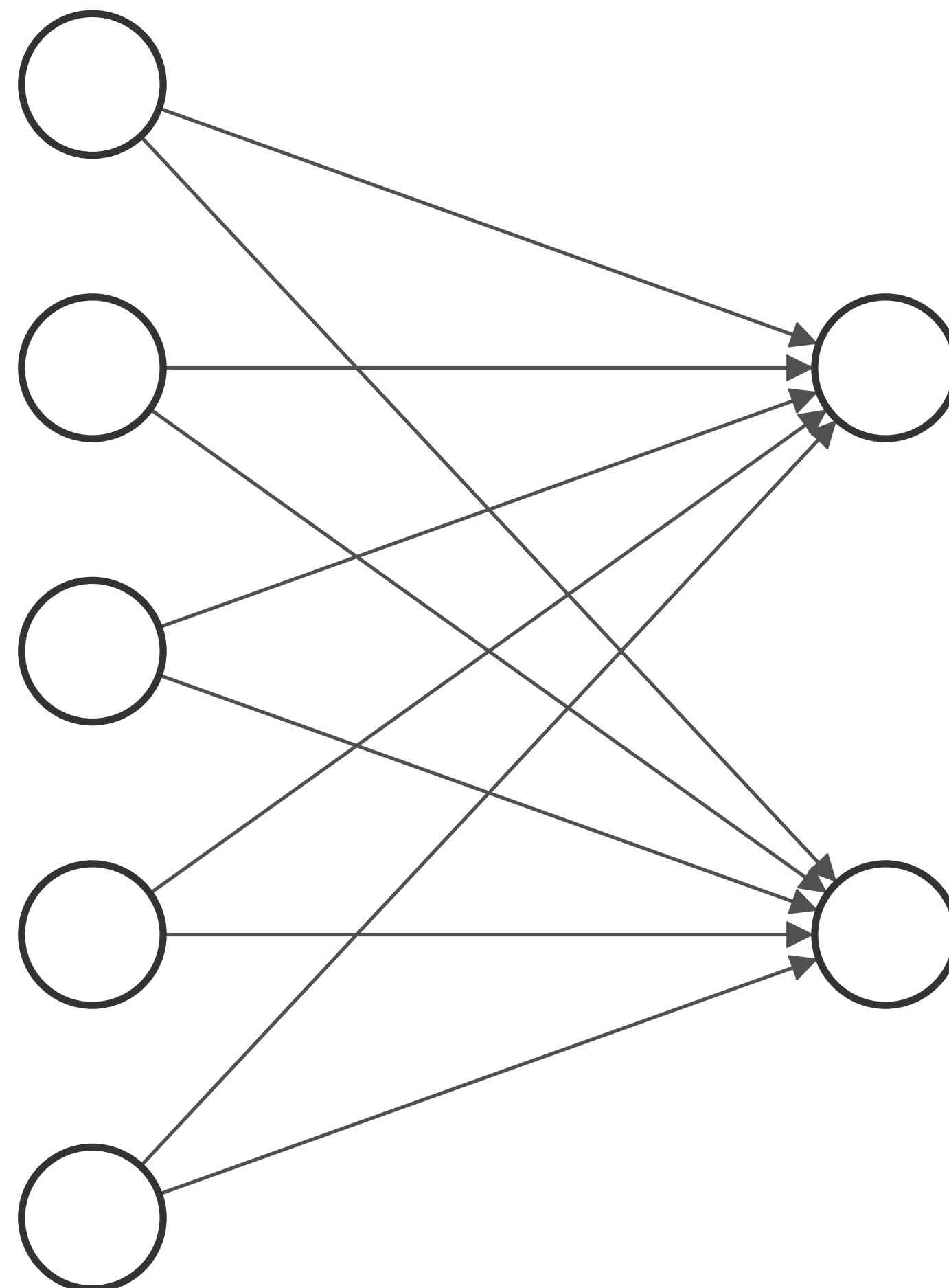
$$\mathbf{x}^\top \mathbf{w} = [x_1 \quad x_2 \quad \dots \quad x_d] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

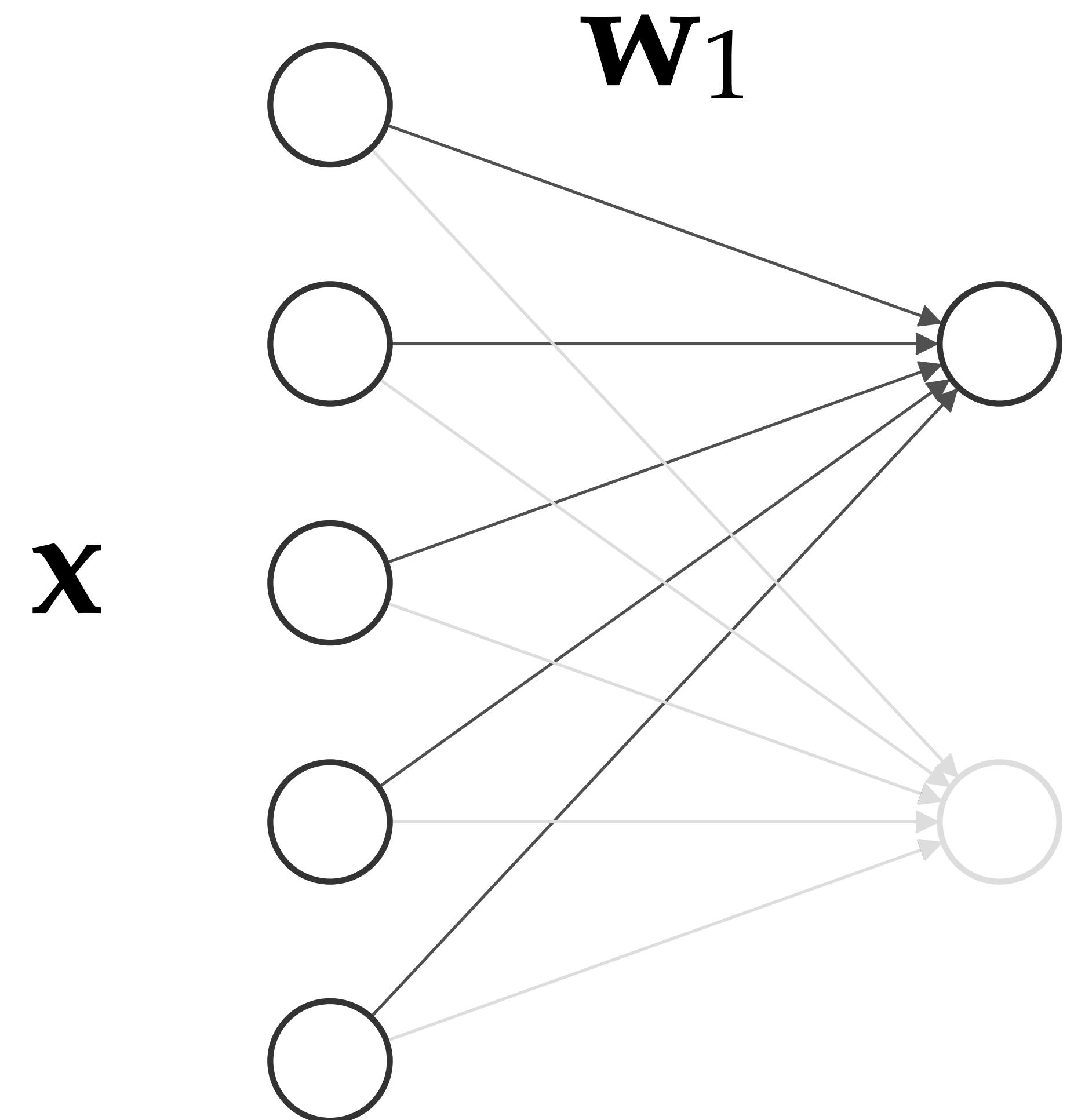


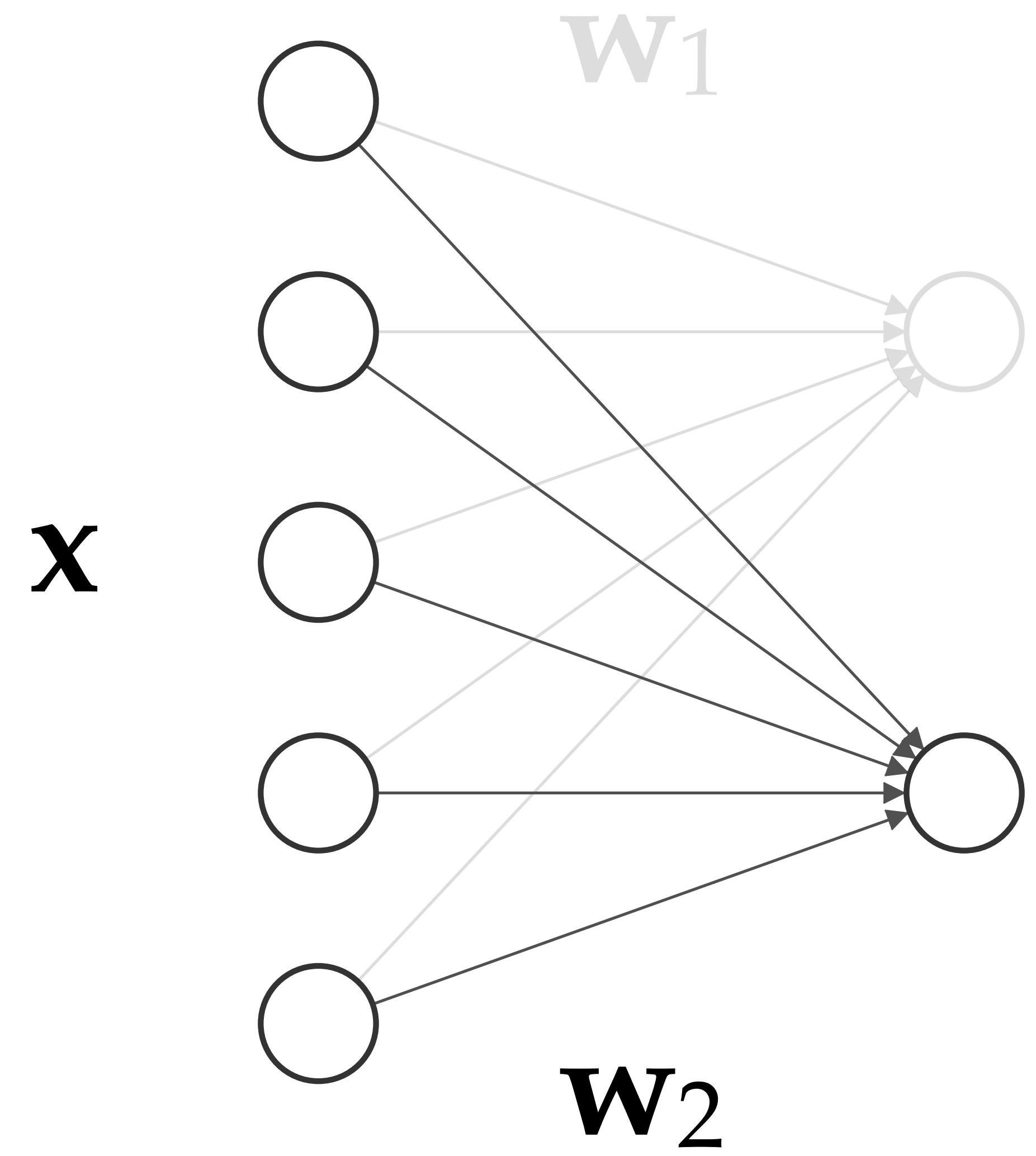


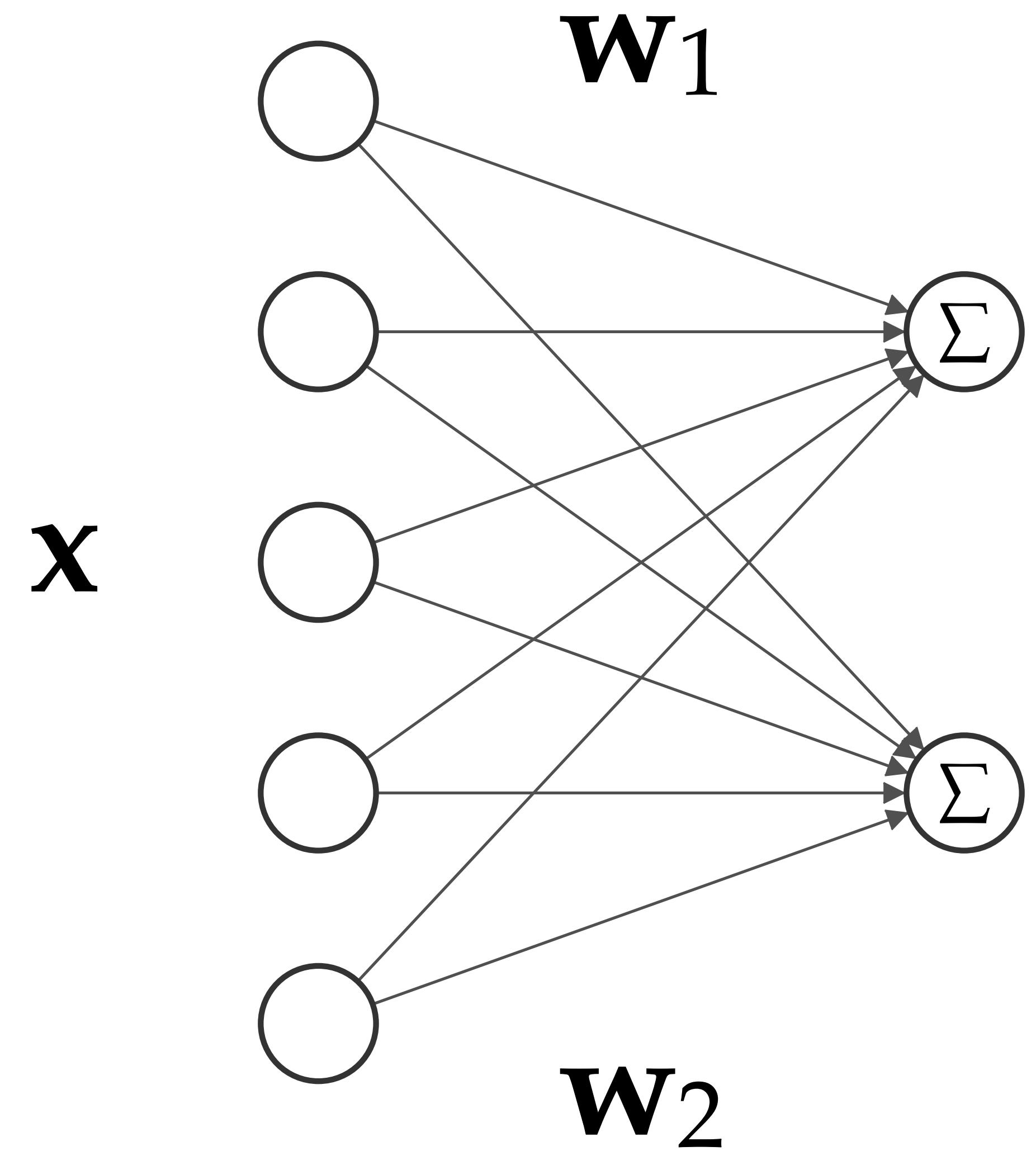


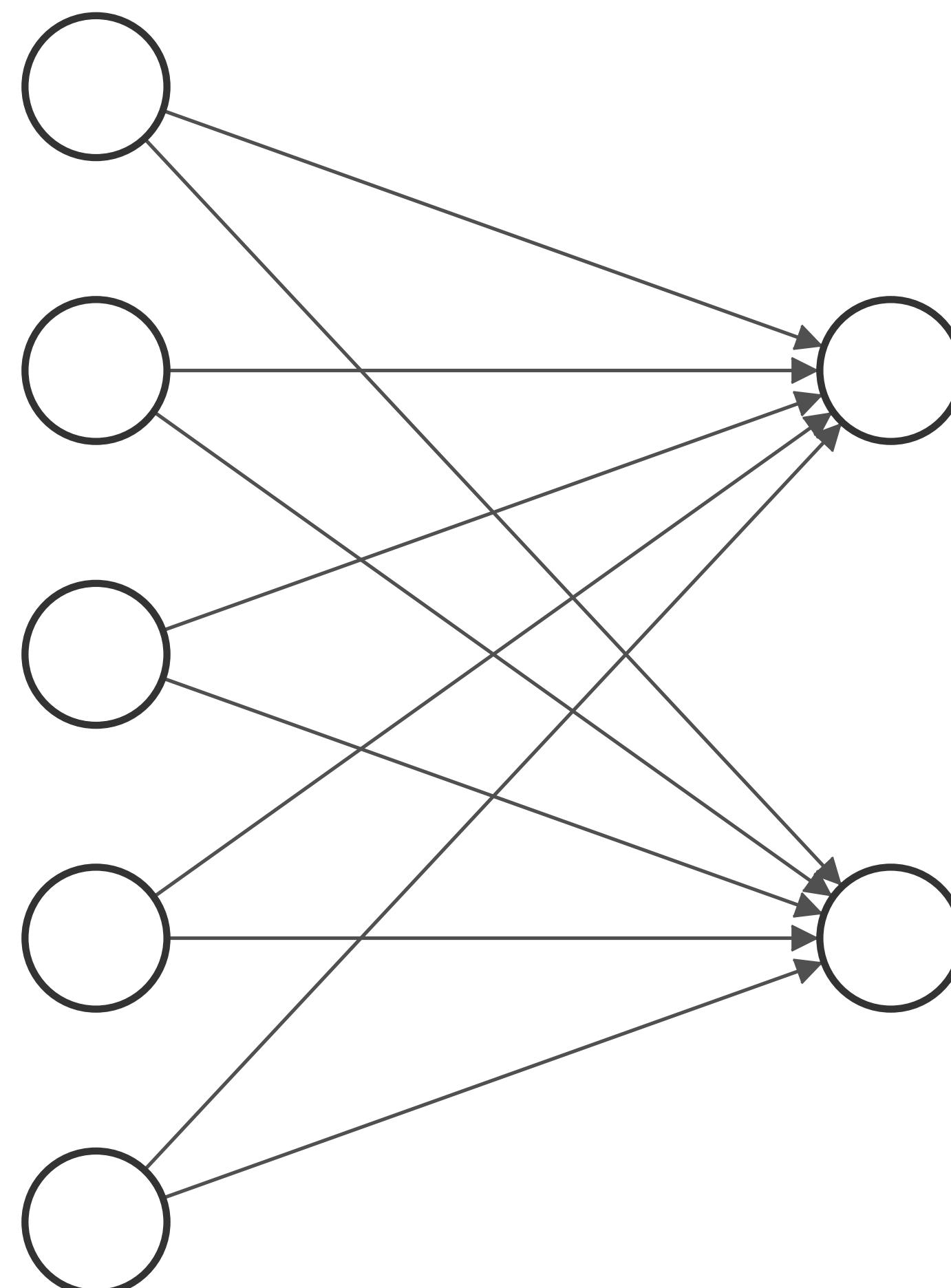
x











$$\mathbf{x}^\top \mathbf{w} = [x_1 \quad x_2 \quad \dots \quad x_d] \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

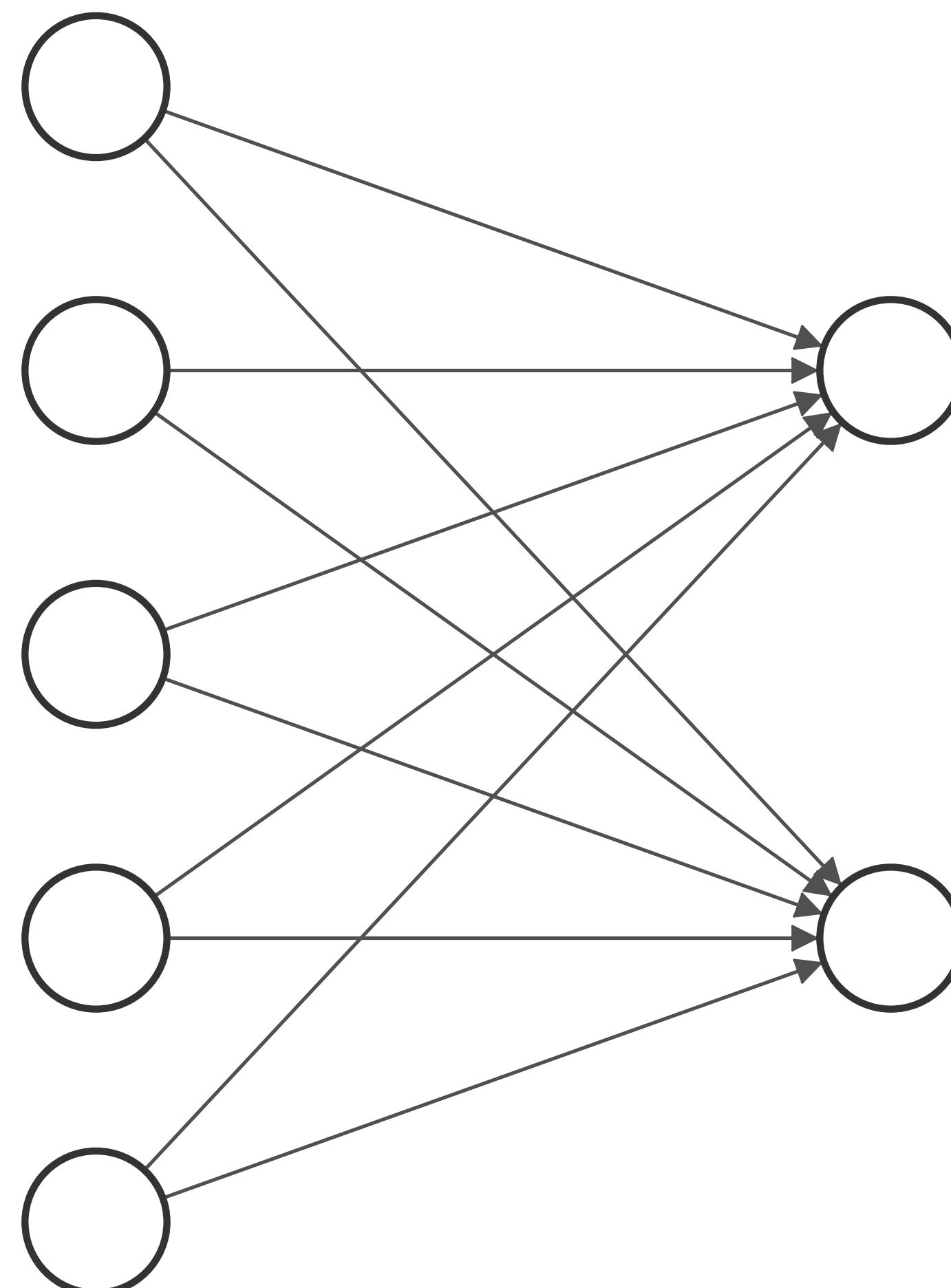
$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

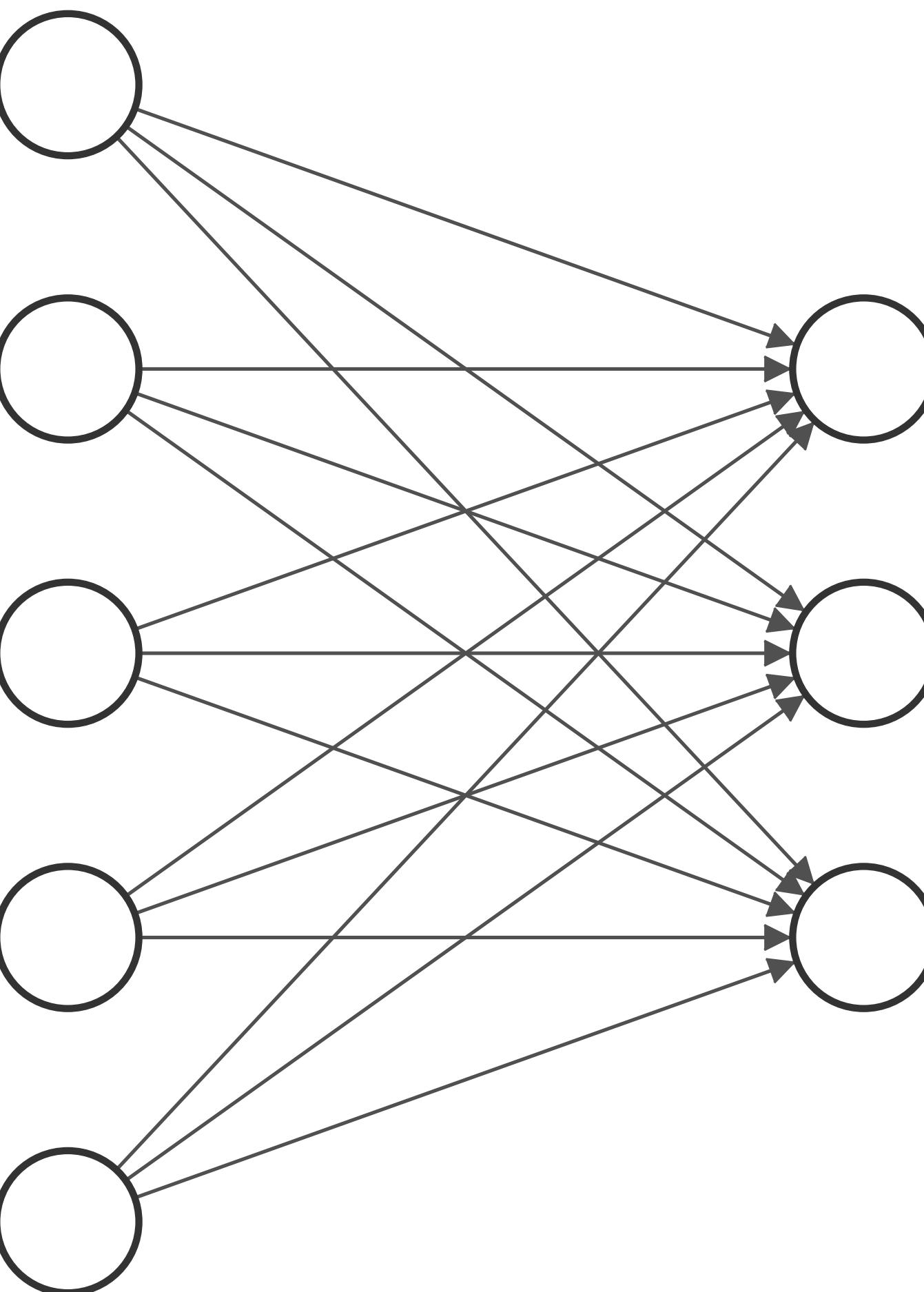
$$\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2]$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \\ \vdots & \vdots \\ w_{d,1} & w_{d,2} \end{bmatrix}$$

$$\mathbf{x}^\top \mathbf{W} = [x_1 \quad x_2 \quad \dots \quad x_d] \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \\ \vdots & \vdots \\ w_{d,1} & w_{d,2} \end{bmatrix}$$

$$\begin{aligned}
\mathbf{x}^\top \mathbf{W} &= [x_1 \quad x_2 \quad \dots \quad x_d] \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \\ \vdots & \vdots \\ w_{d,1} & w_{d,2} \end{bmatrix} \\
&= [\mathbf{x} \cdot \mathbf{w}_1 \quad \mathbf{x} \cdot \mathbf{w}_2]
\end{aligned}$$



d  k

$$\begin{aligned}
\mathbf{x}^\top \mathbf{W} &= [x_1 \quad x_2 \quad \dots \quad x_d] \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,k} \\ w_{2,1} & w_{2,2} & \dots & w_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ w_{d,1} & w_{d,2} & \dots & w_{d,k} \end{bmatrix} \\
&= [\mathbf{x} \cdot \mathbf{w}_1 \quad \mathbf{x} \cdot \mathbf{w}_2 \quad \dots \quad \mathbf{x} \cdot \mathbf{w}_k]
\end{aligned}$$

↓ Feature dimensions ↓

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix}$$

↑ Samples ↑

$$\mathbf{XW} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,k} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ w_{d,1} & w_{d,2} & \cdots & w_{d,k} \end{bmatrix}$$

$$\begin{aligned}
\mathbf{XW} &= \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,k} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ w_{d,1} & w_{d,2} & \cdots & w_{d,k} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{x}_1 \cdot \mathbf{w}_1 & \mathbf{x}_1 \cdot \mathbf{w}_2 & \cdots & \mathbf{x}_1 \cdot \mathbf{w}_k \\ \mathbf{x}_2 \cdot \mathbf{w}_1 & \mathbf{x}_2 \cdot \mathbf{w}_2 & \cdots & \mathbf{x}_2 \cdot \mathbf{w}_k \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n \cdot \mathbf{w}_1 & \mathbf{x}_n \cdot \mathbf{w}_2 & \cdots & \mathbf{x}_n \cdot \mathbf{w}_k \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\mathbf{XW} &= \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,k} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ w_{d,1} & w_{d,2} & \cdots & w_{d,k} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{x}_1 \cdot \mathbf{w}_1 & \mathbf{x}_1 \cdot \mathbf{w}_2 & \cdots & \mathbf{x}_1 \cdot \mathbf{w}_k \\ \mathbf{x}_2 \cdot \mathbf{w}_1 & \mathbf{x}_2 \cdot \mathbf{w}_2 & \cdots & \mathbf{x}_2 \cdot \mathbf{w}_k \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n \cdot \mathbf{w}_1 & \mathbf{x}_n \cdot \mathbf{w}_2 & \cdots & \mathbf{x}_n \cdot \mathbf{w}_k \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1^\top \\ \mathbf{y}_2^\top \\ \vdots \\ \mathbf{y}_n^\top \end{bmatrix}
\end{aligned}$$

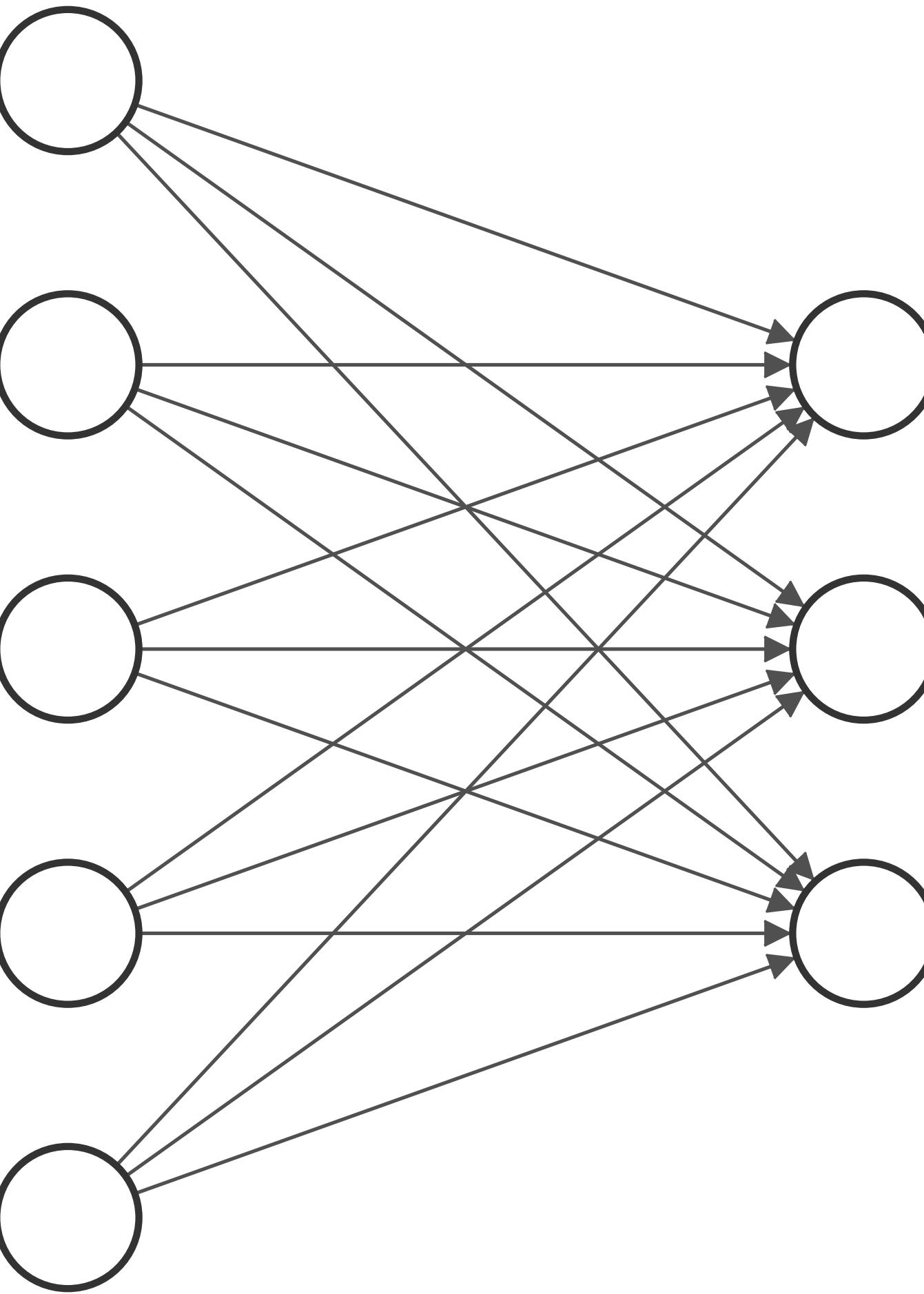
$$\begin{aligned}
\mathbf{XW} &= \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,k} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ w_{d,1} & w_{d,2} & \cdots & w_{d,k} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{x}_1 \cdot \mathbf{w}_1 & \mathbf{x}_1 \cdot \mathbf{w}_2 & \cdots & \mathbf{x}_1 \cdot \mathbf{w}_k \\ \mathbf{x}_2 \cdot \mathbf{w}_1 & \mathbf{x}_2 \cdot \mathbf{w}_2 & \cdots & \mathbf{x}_2 \cdot \mathbf{w}_k \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n \cdot \mathbf{w}_1 & \mathbf{x}_n \cdot \mathbf{w}_2 & \cdots & \mathbf{x}_n \cdot \mathbf{w}_k \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1^\top \\ \mathbf{y}_2^\top \\ \vdots \\ \mathbf{y}_n^\top \end{bmatrix} = \hat{\mathbf{Y}}
\end{aligned}$$

\hat{Y} = XW

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$

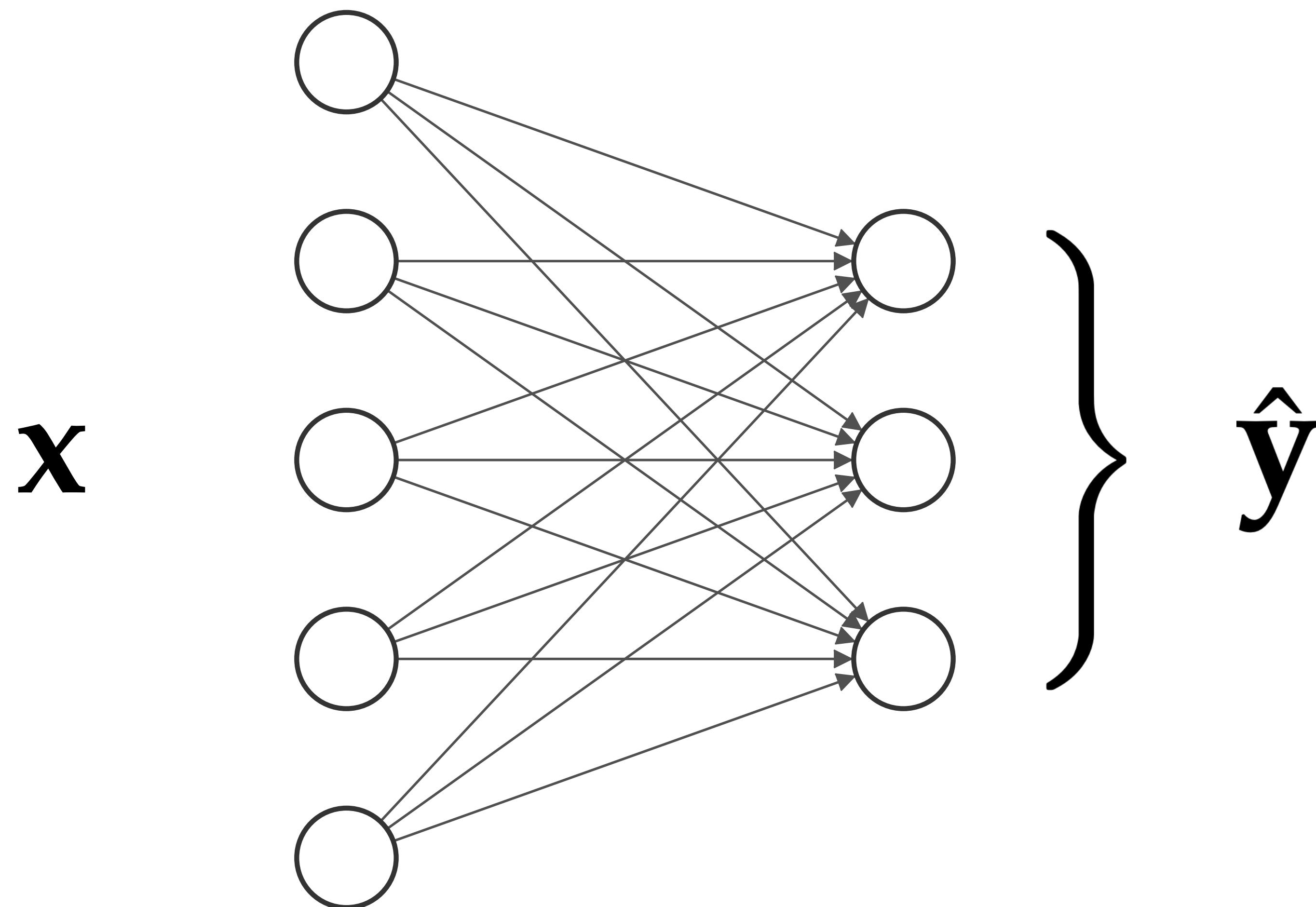
$$\mathbf{W} \in \mathbb{R}^{d \times k}$$

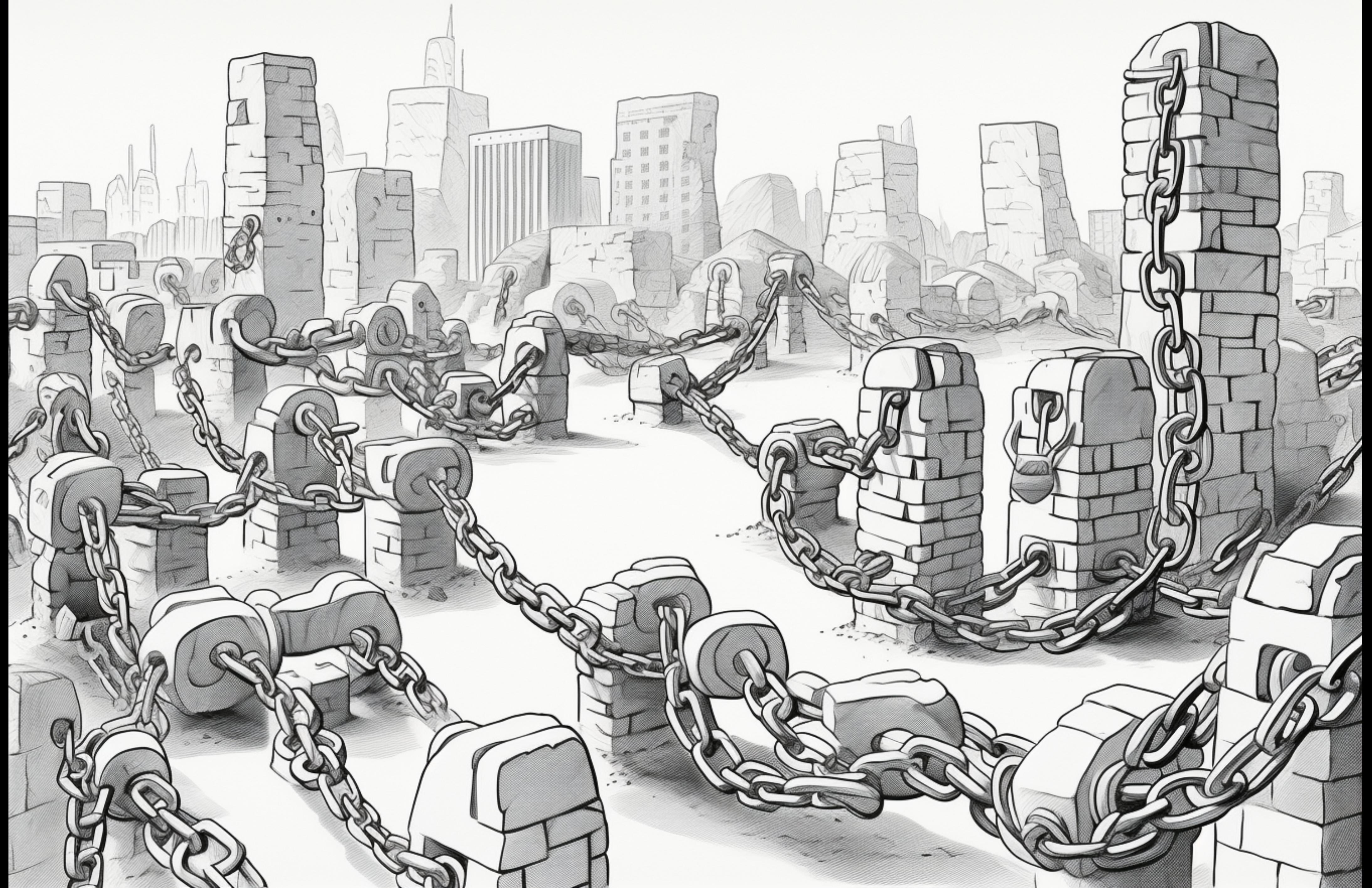
$$\hat{\mathbf{Y}} \in \mathbb{R}^{n \times k}$$

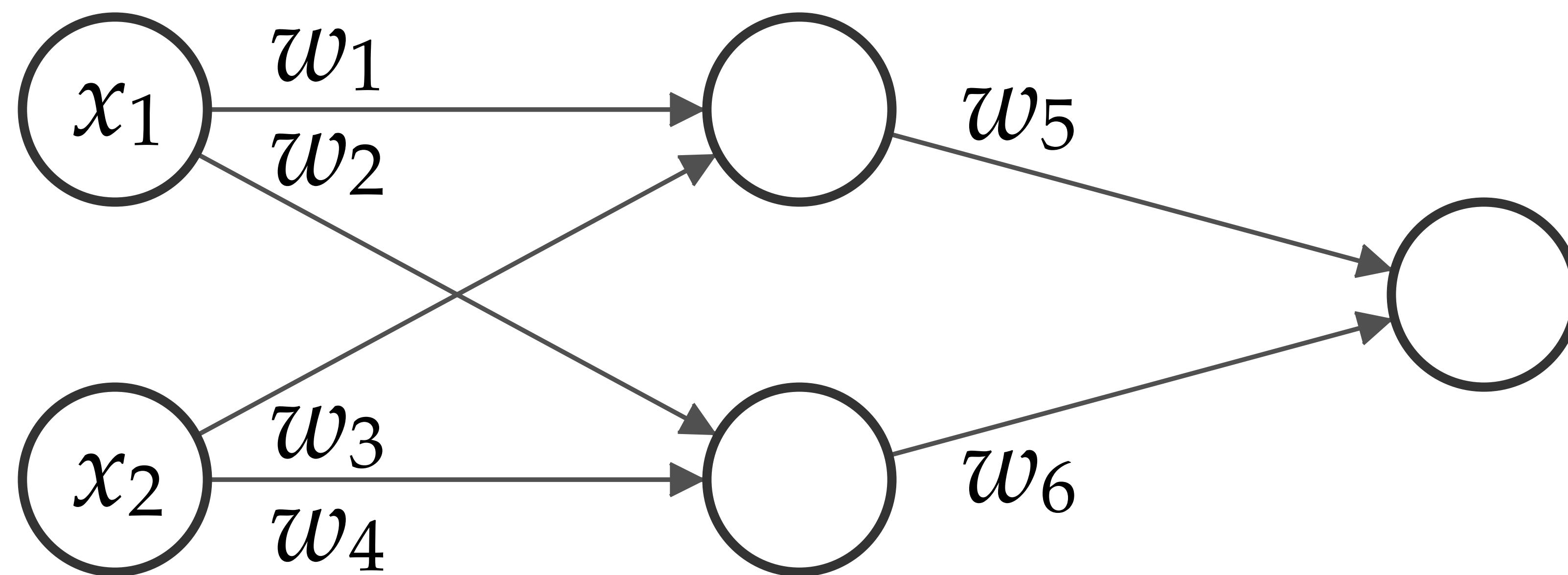
d  k

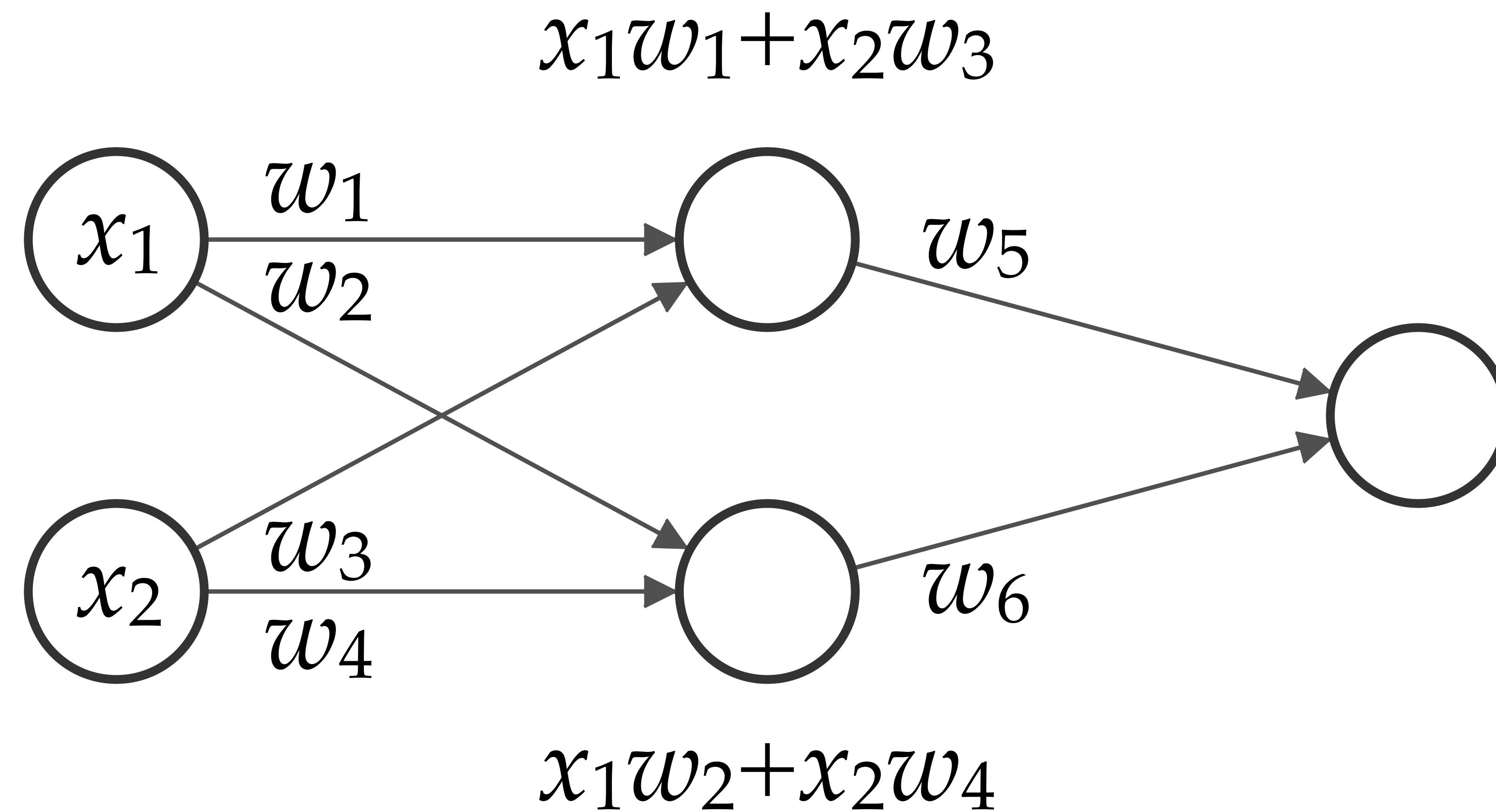


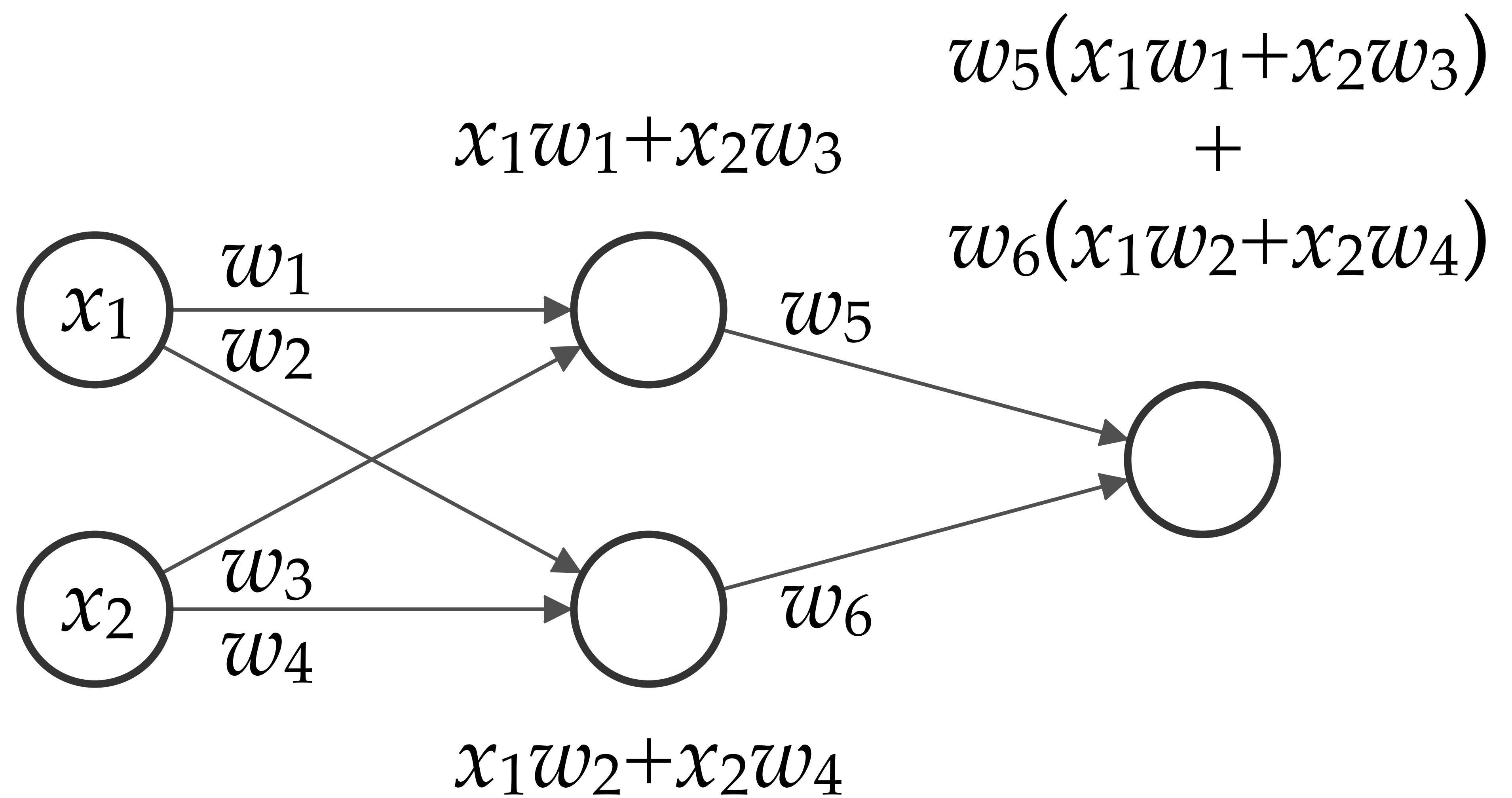


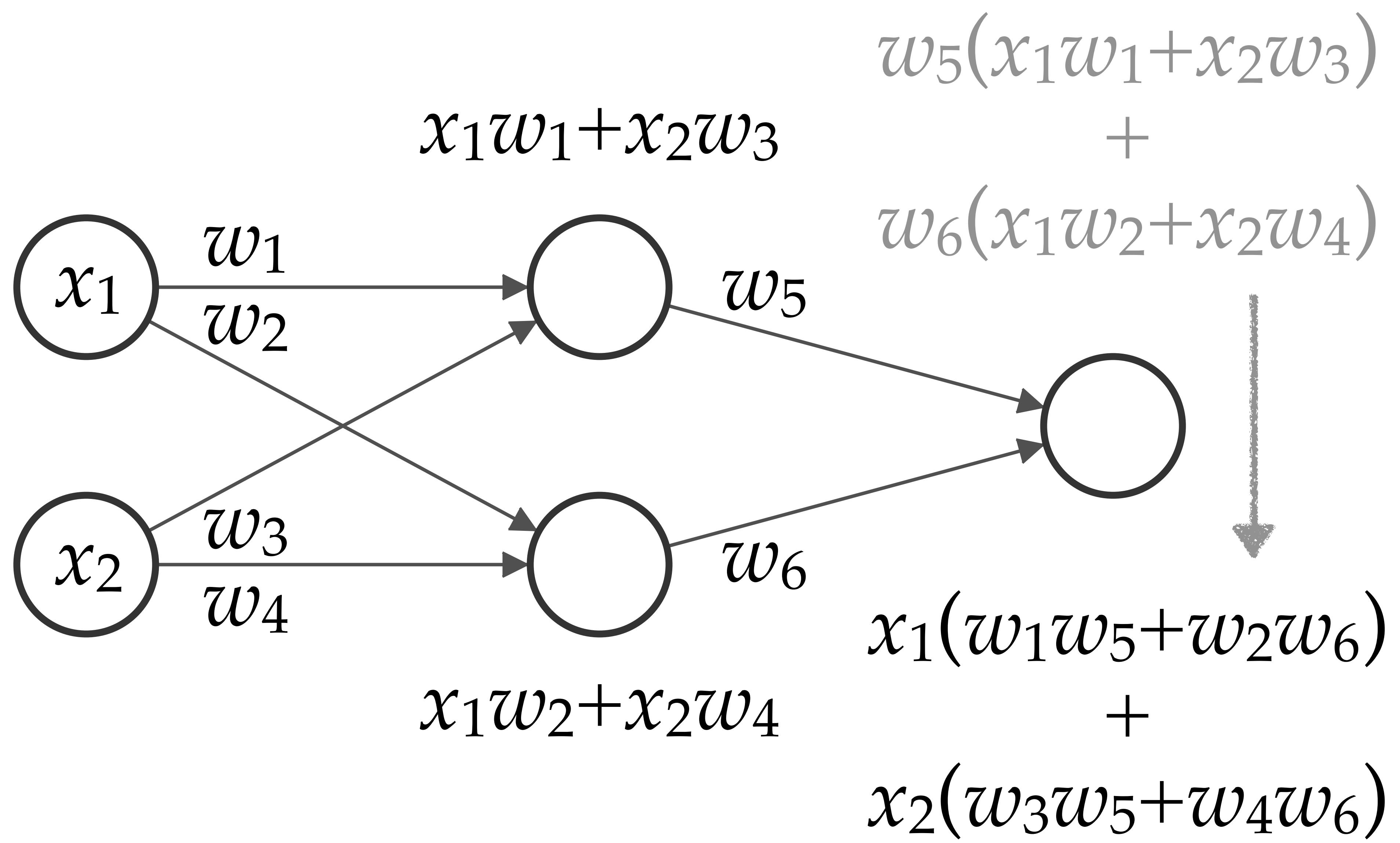


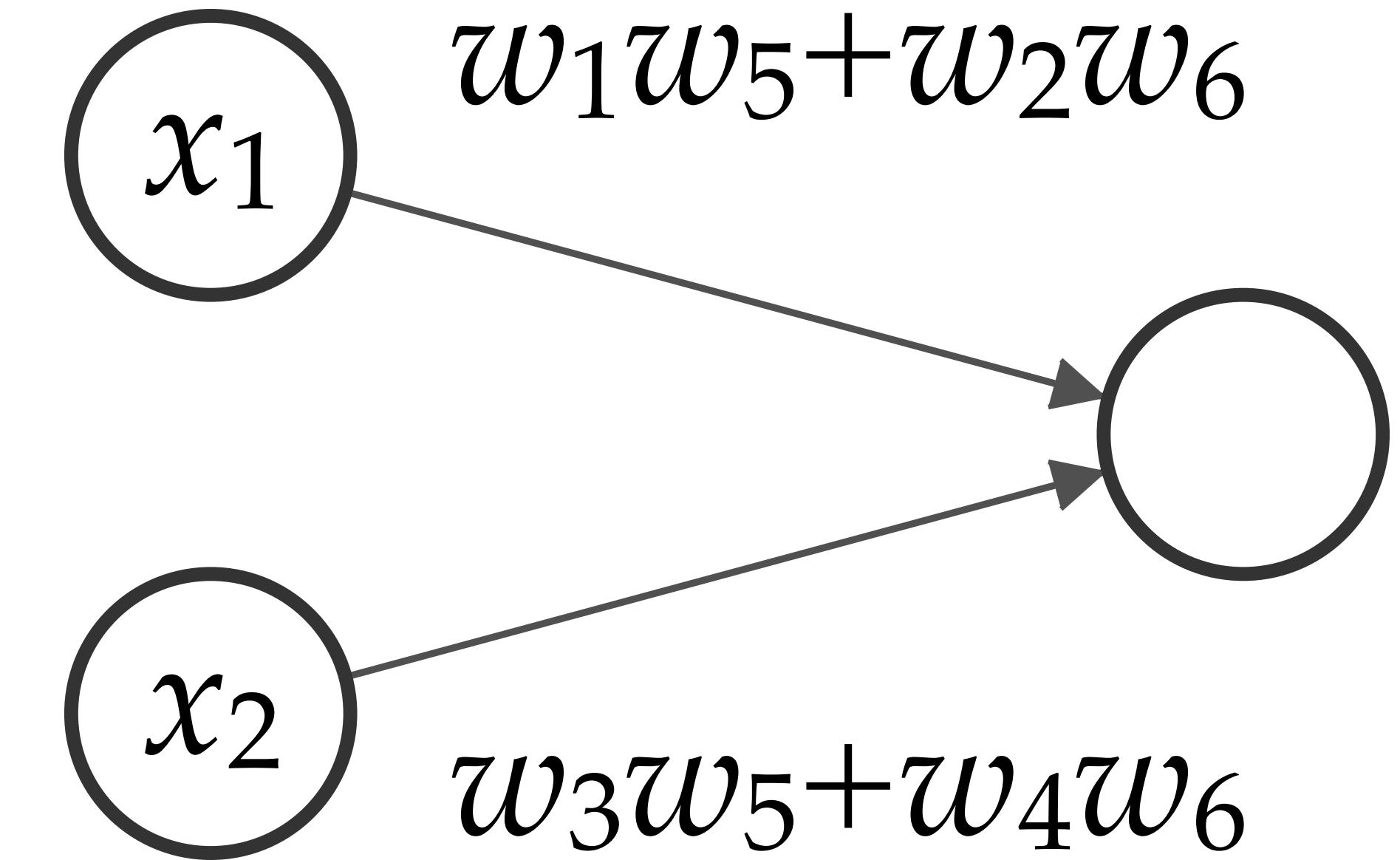












$$\hat{\mathbf{y}} = (((((x\mathbf{W}_1)\mathbf{W}_2)\dots)\mathbf{W}_k)$$

$$\begin{aligned}\hat{y} &= (((((xW_1)W_2)\dots)W_k) \\ &= x(W_1W_2\dots W_k)\end{aligned}$$

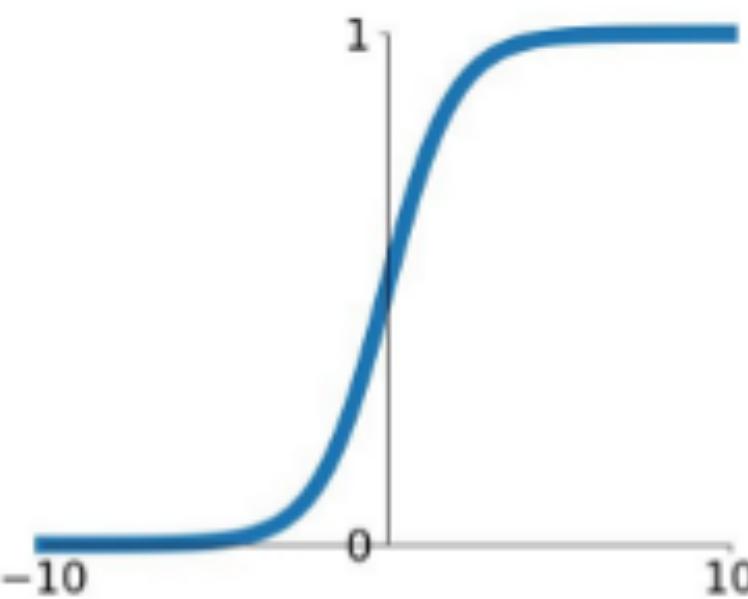
$$\begin{aligned}\hat{y} &= (((((xW_1)W_2)\dots)W_k) \\ &\equiv x(W_1W_2\dots W_k) \\ &\equiv xW\end{aligned}$$

$$\text{fire} = \begin{cases} 1 & \text{if } \mathbf{x} \cdot \mathbf{w} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{y} = a_k(\dots a_2(a_1(xW_1)W_2)\dots W_k)$$

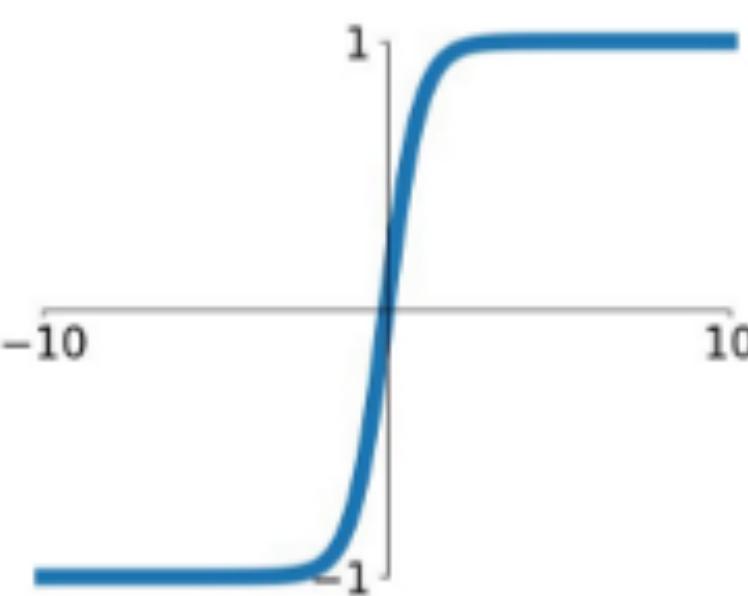
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



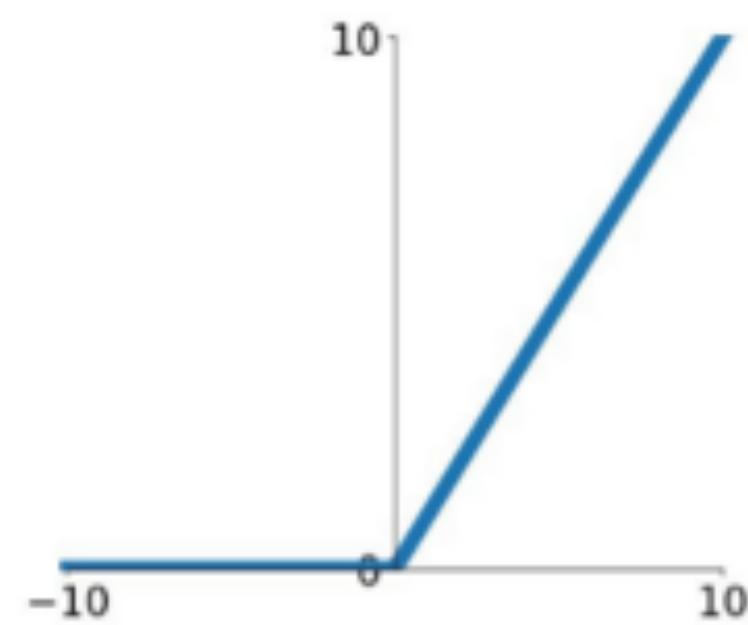
tanh

$$\tanh(x)$$



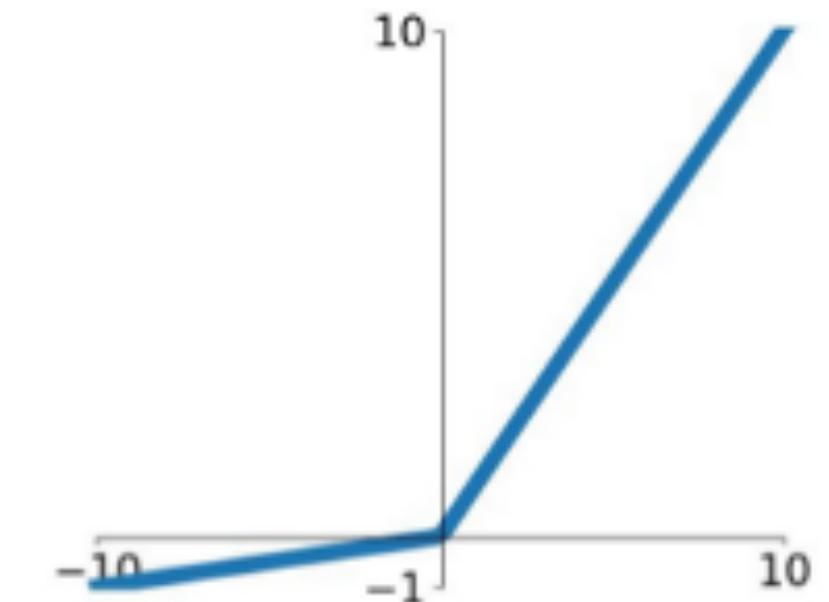
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

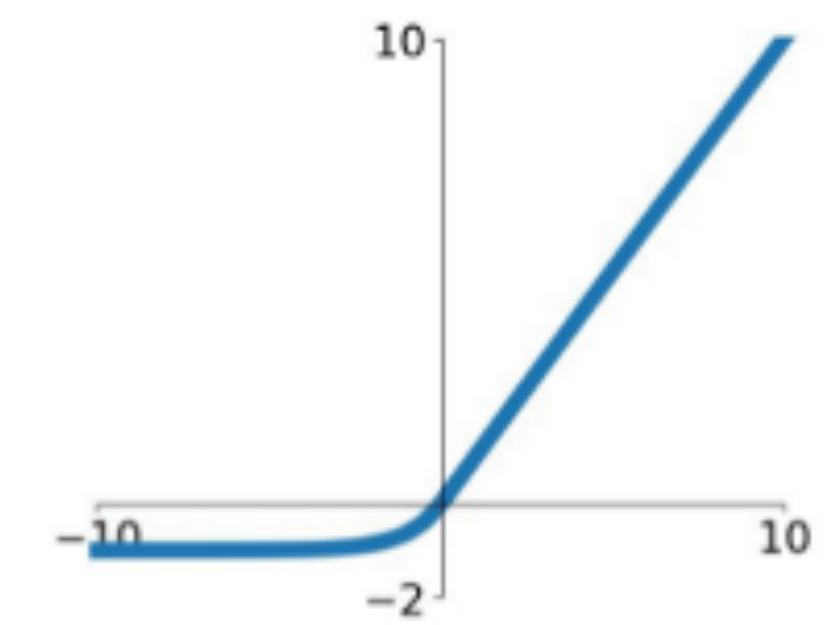


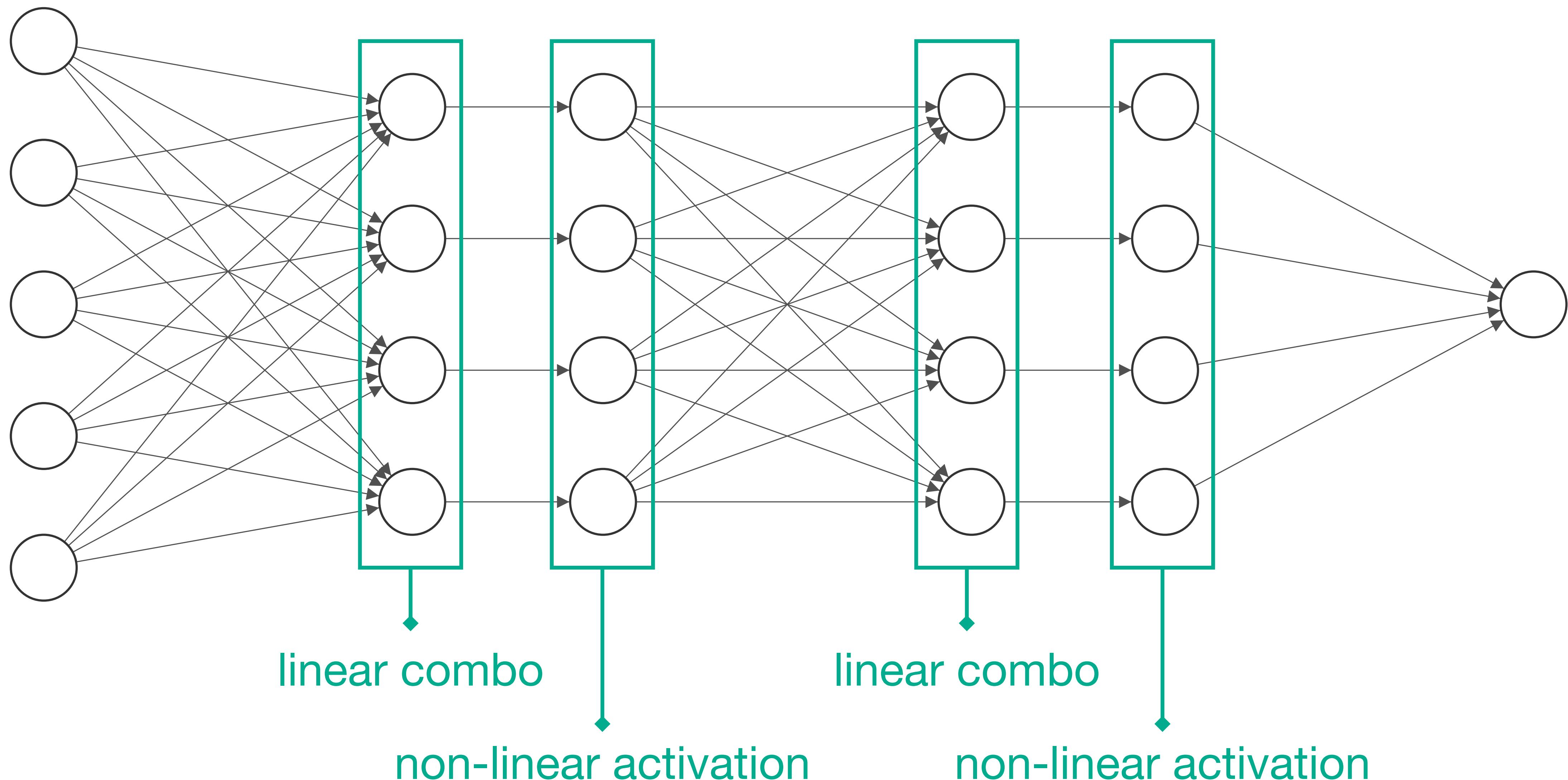
Maxout

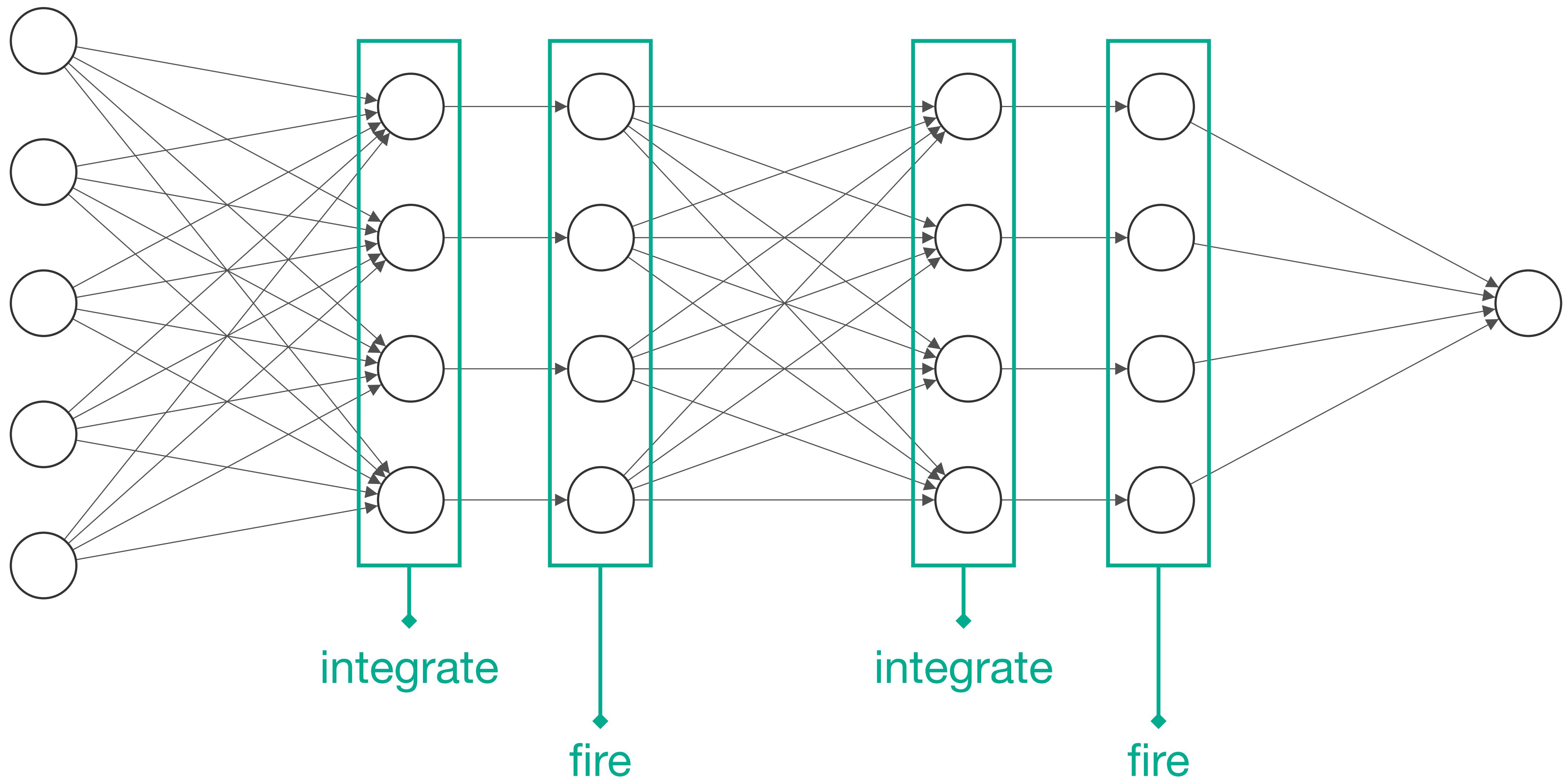
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

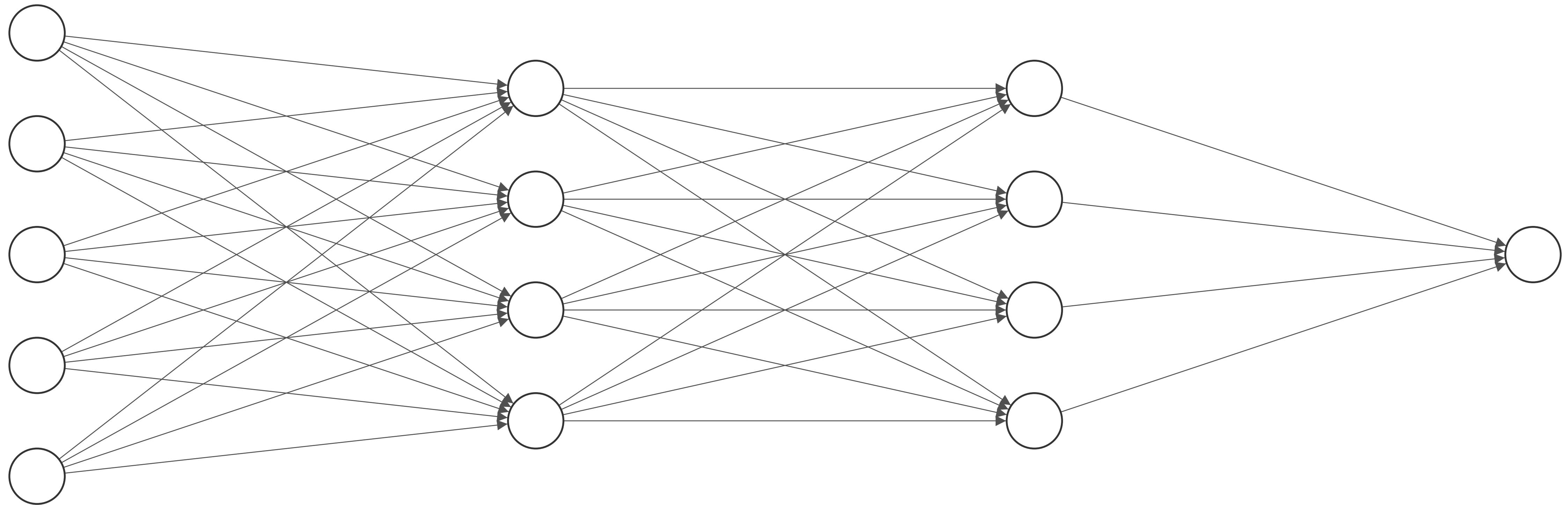
ELU

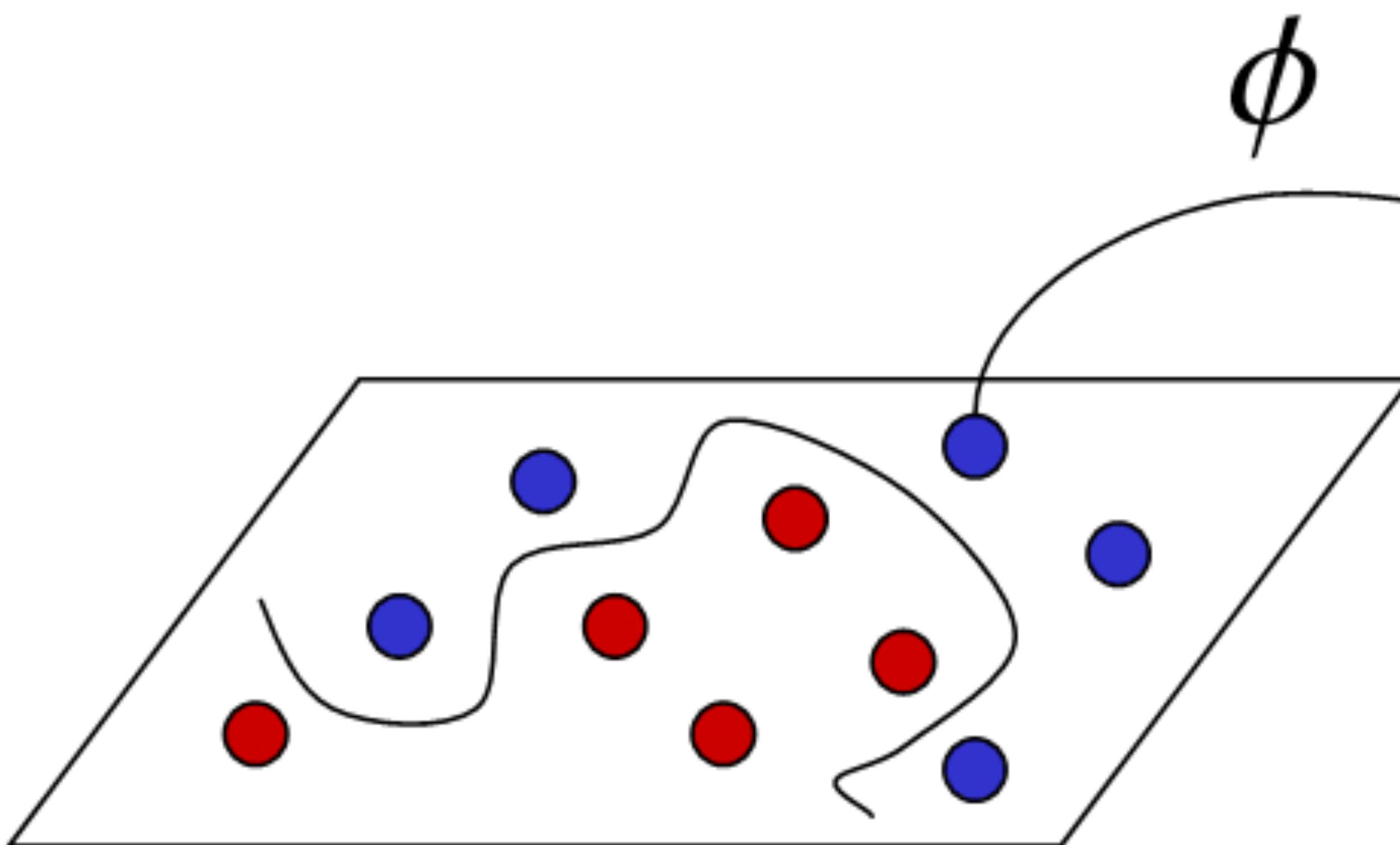
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



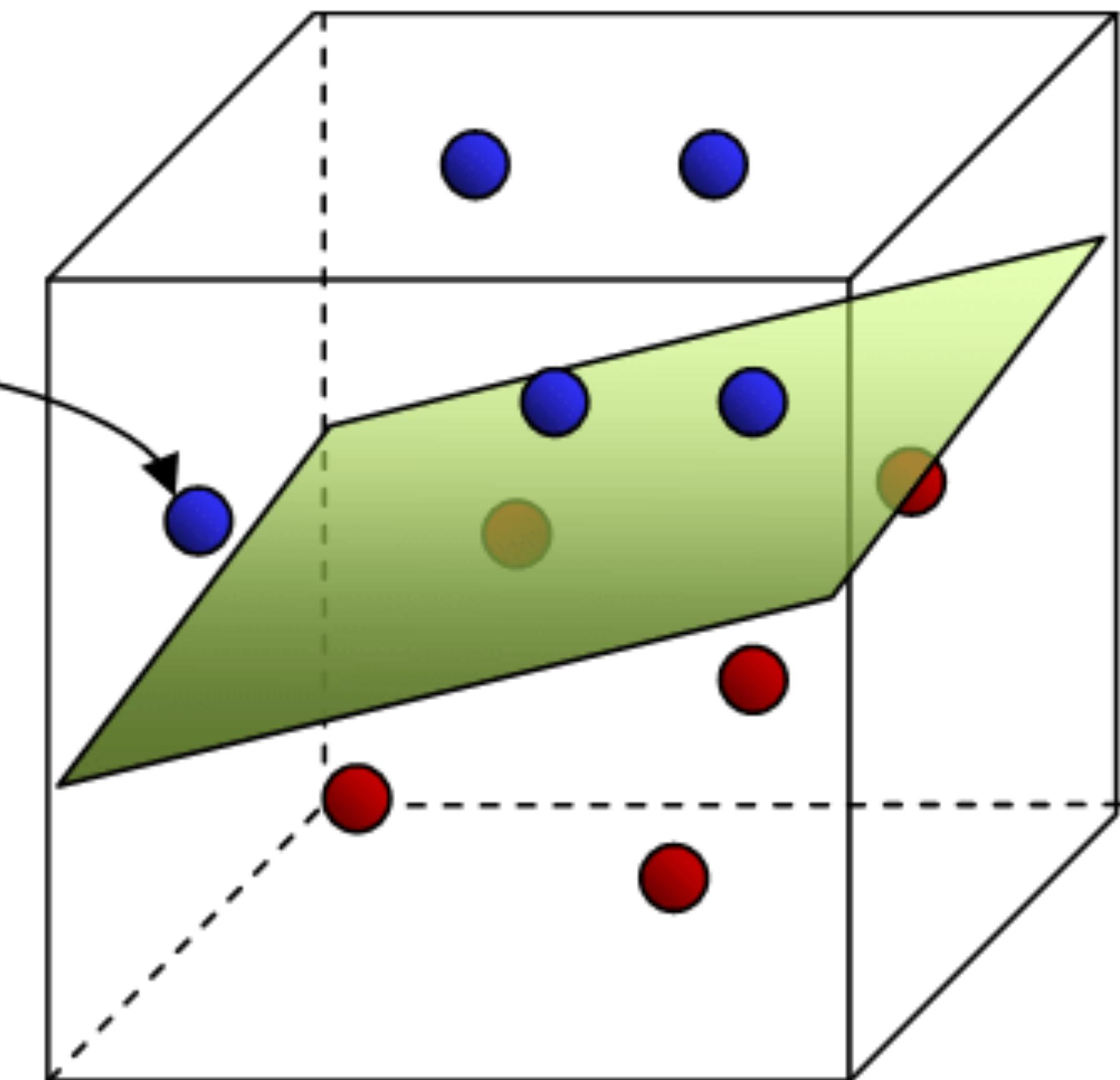




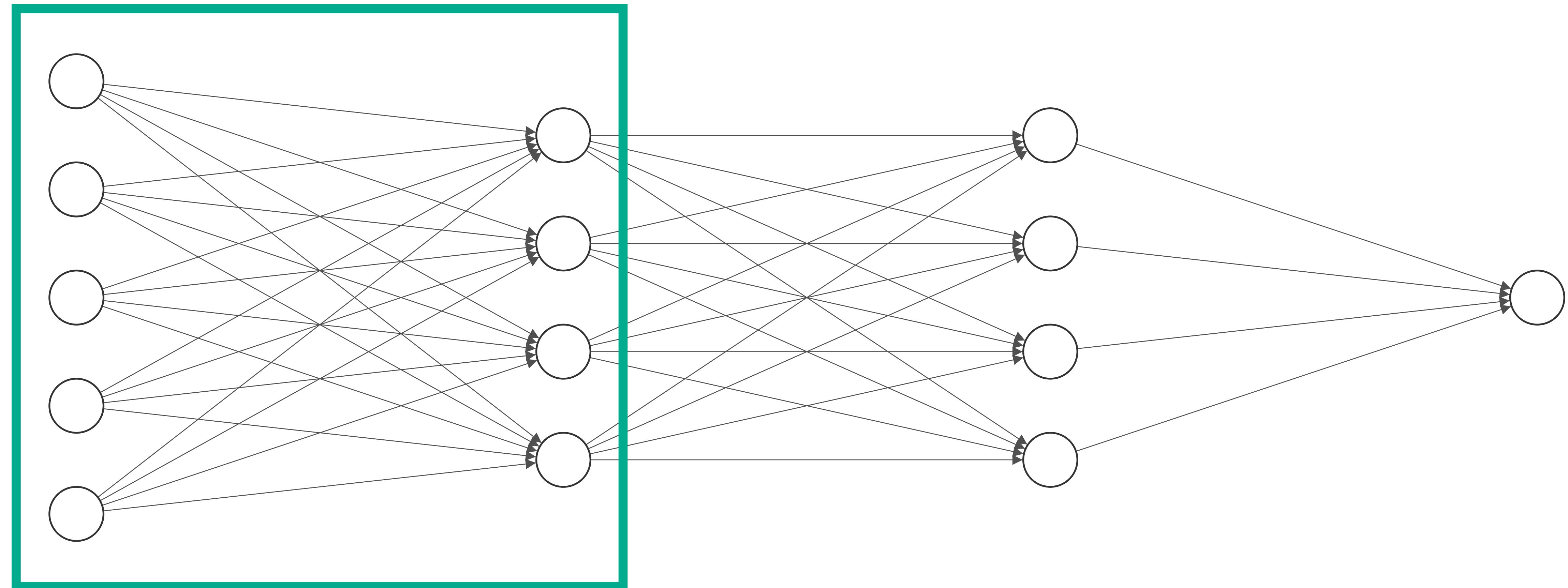


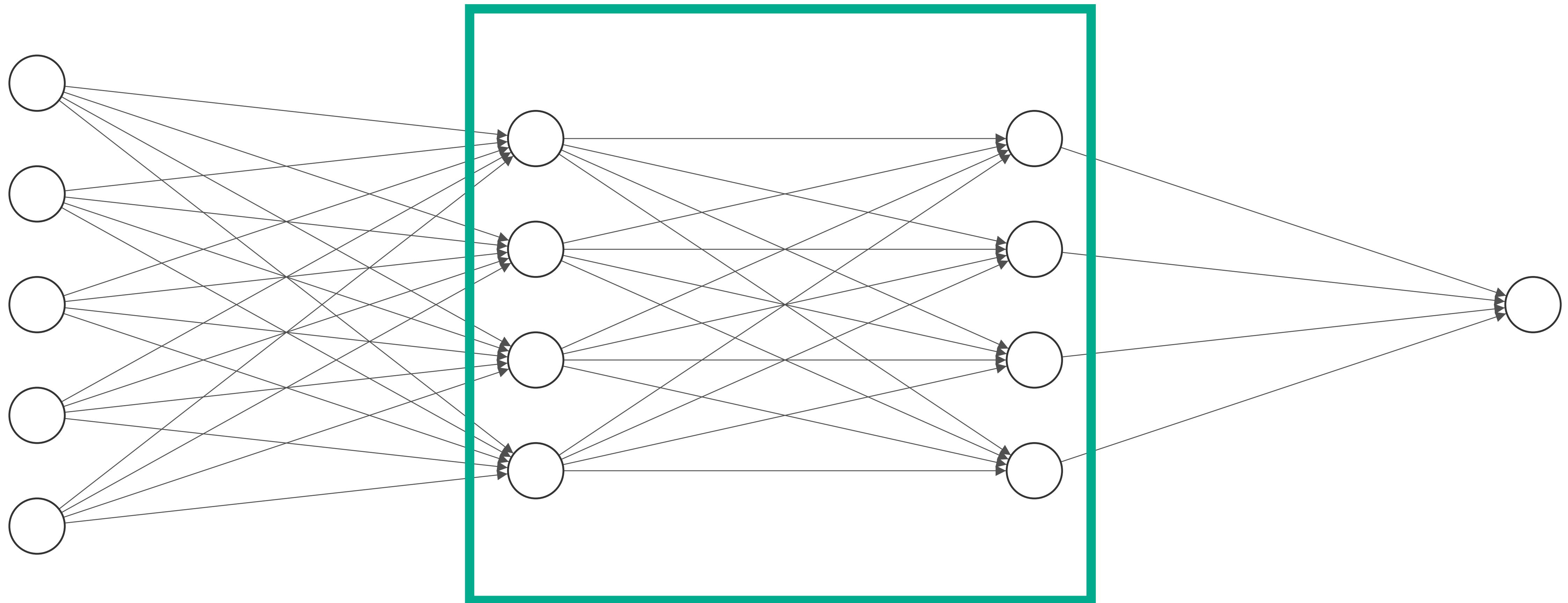


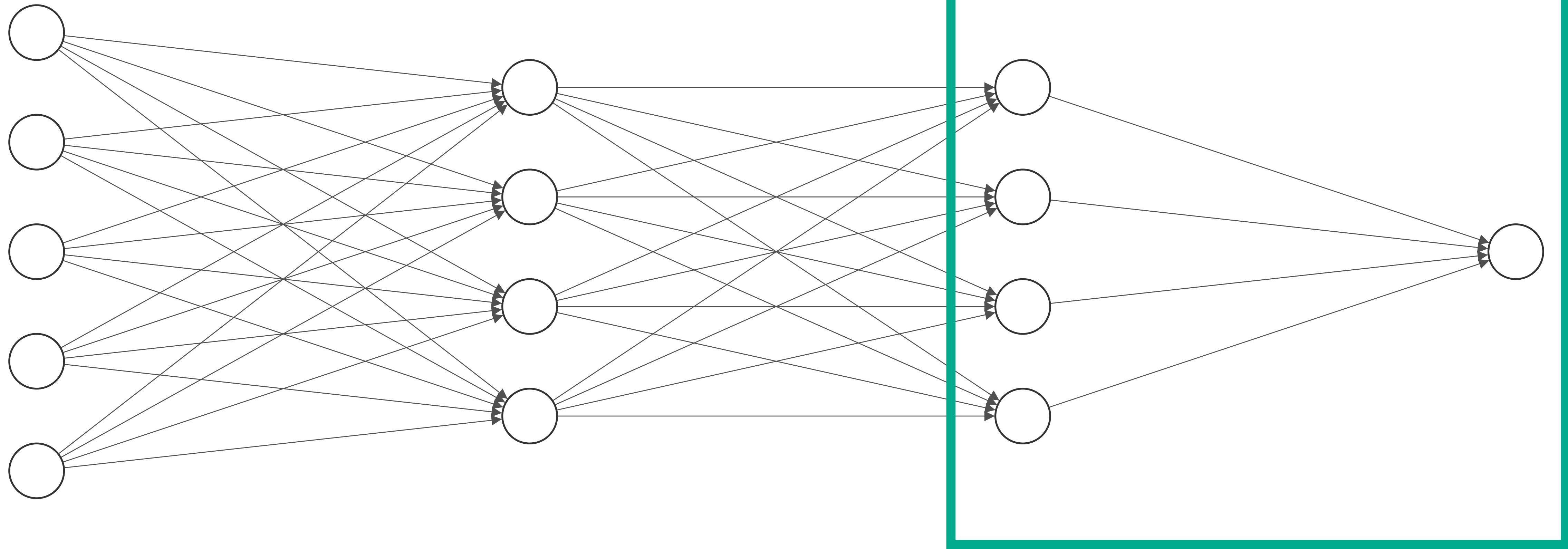
a) Input Space

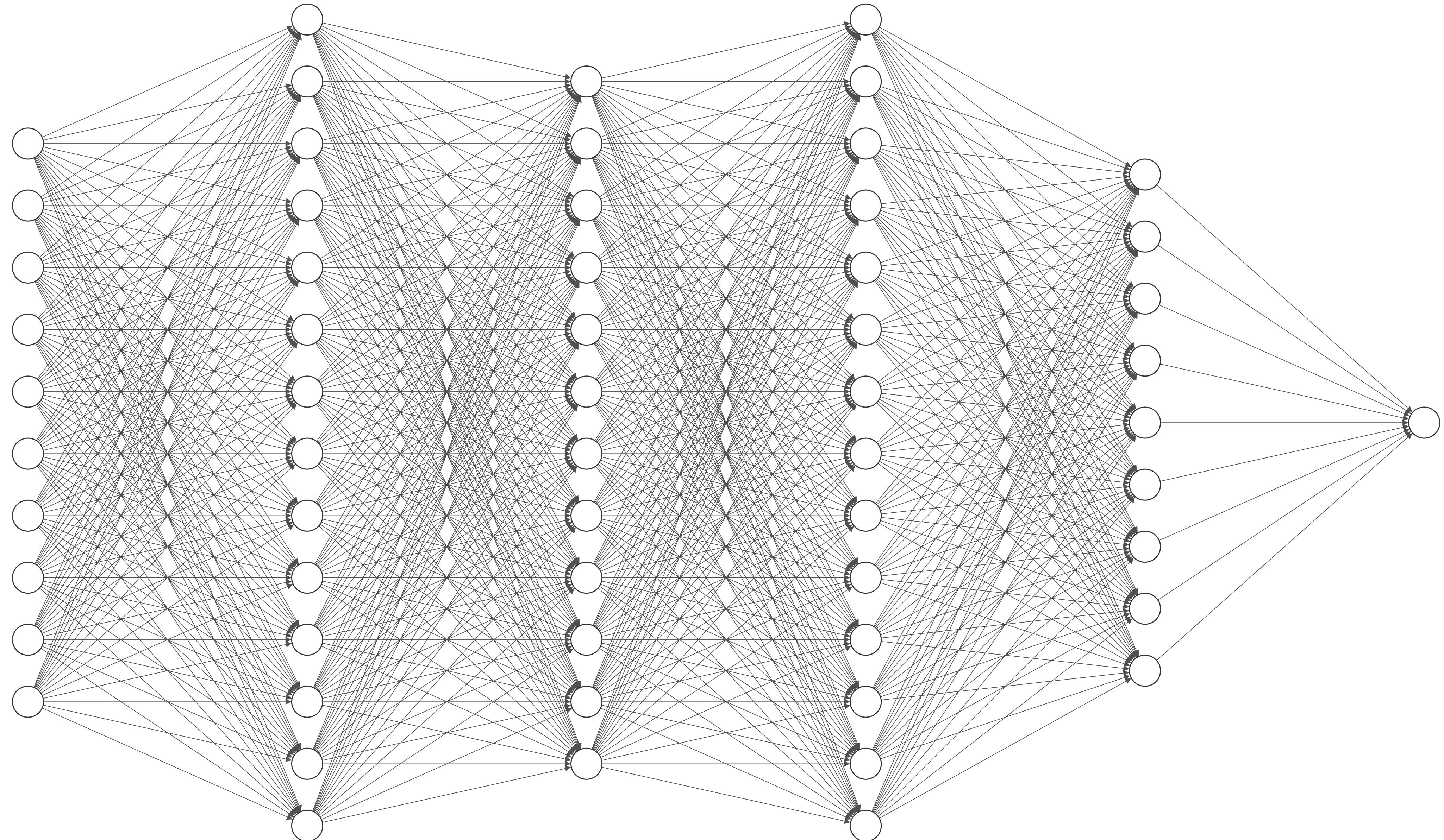


b) Feature Space

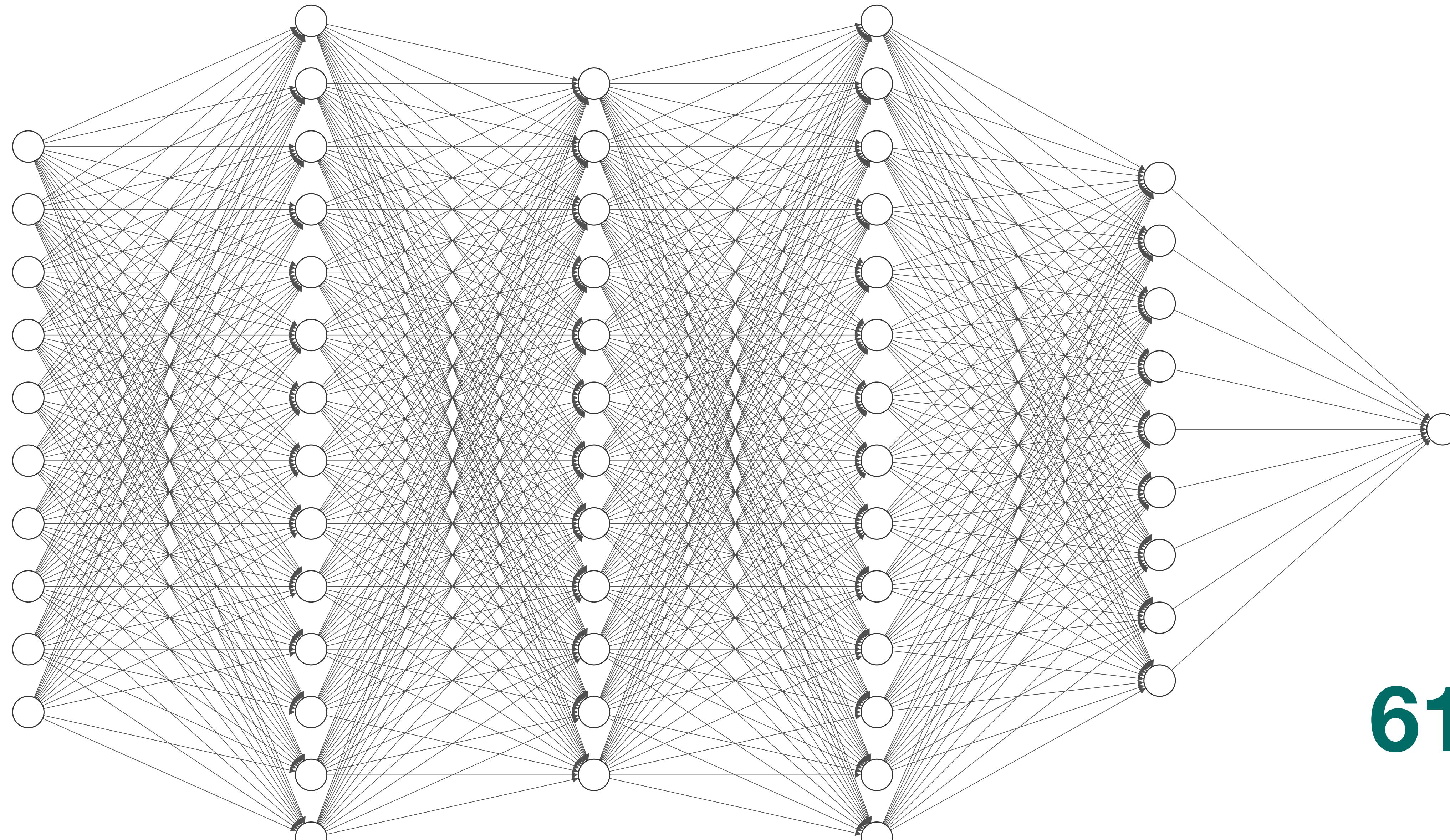








10 14 12 14 9 1



140

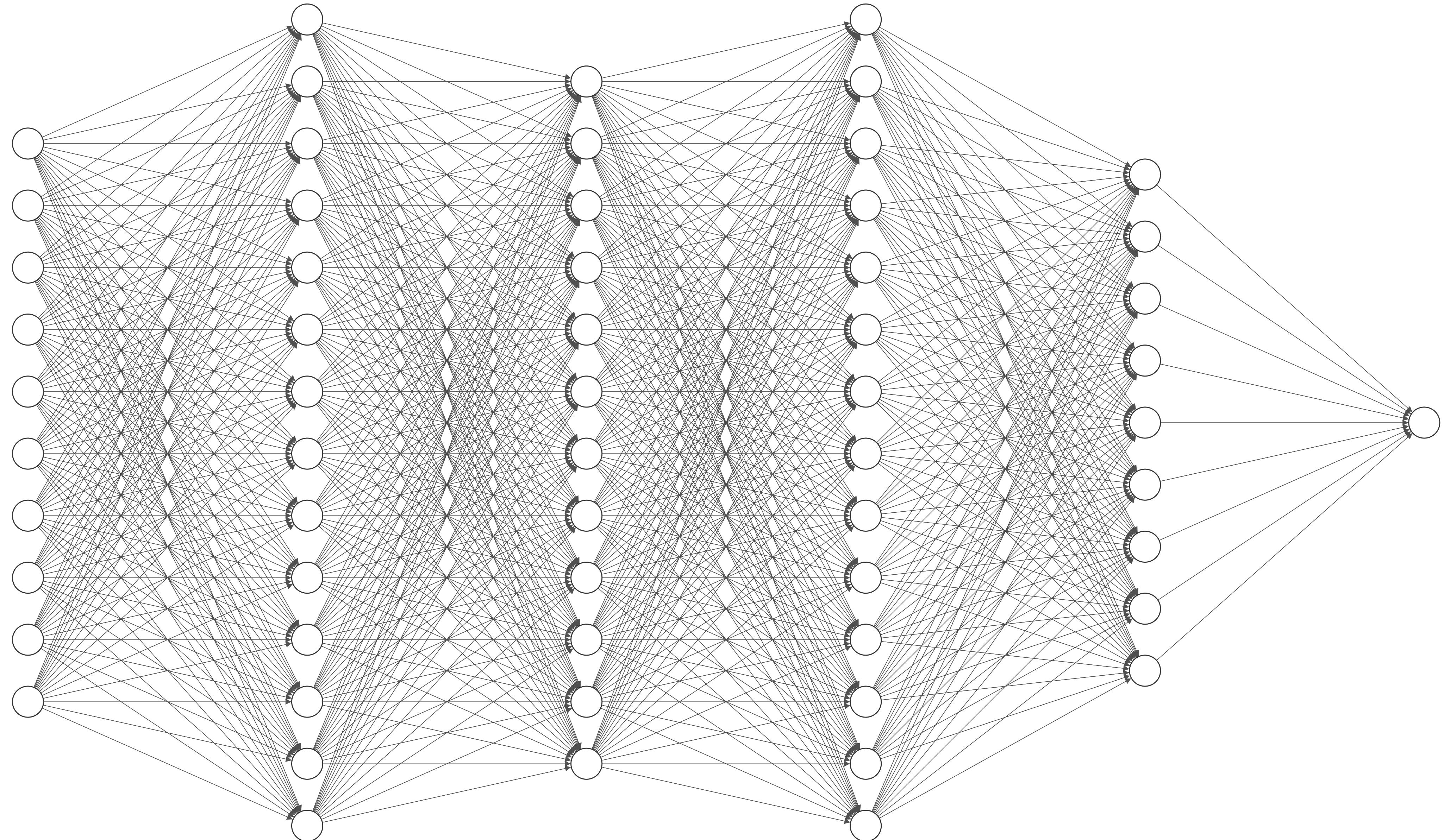
168

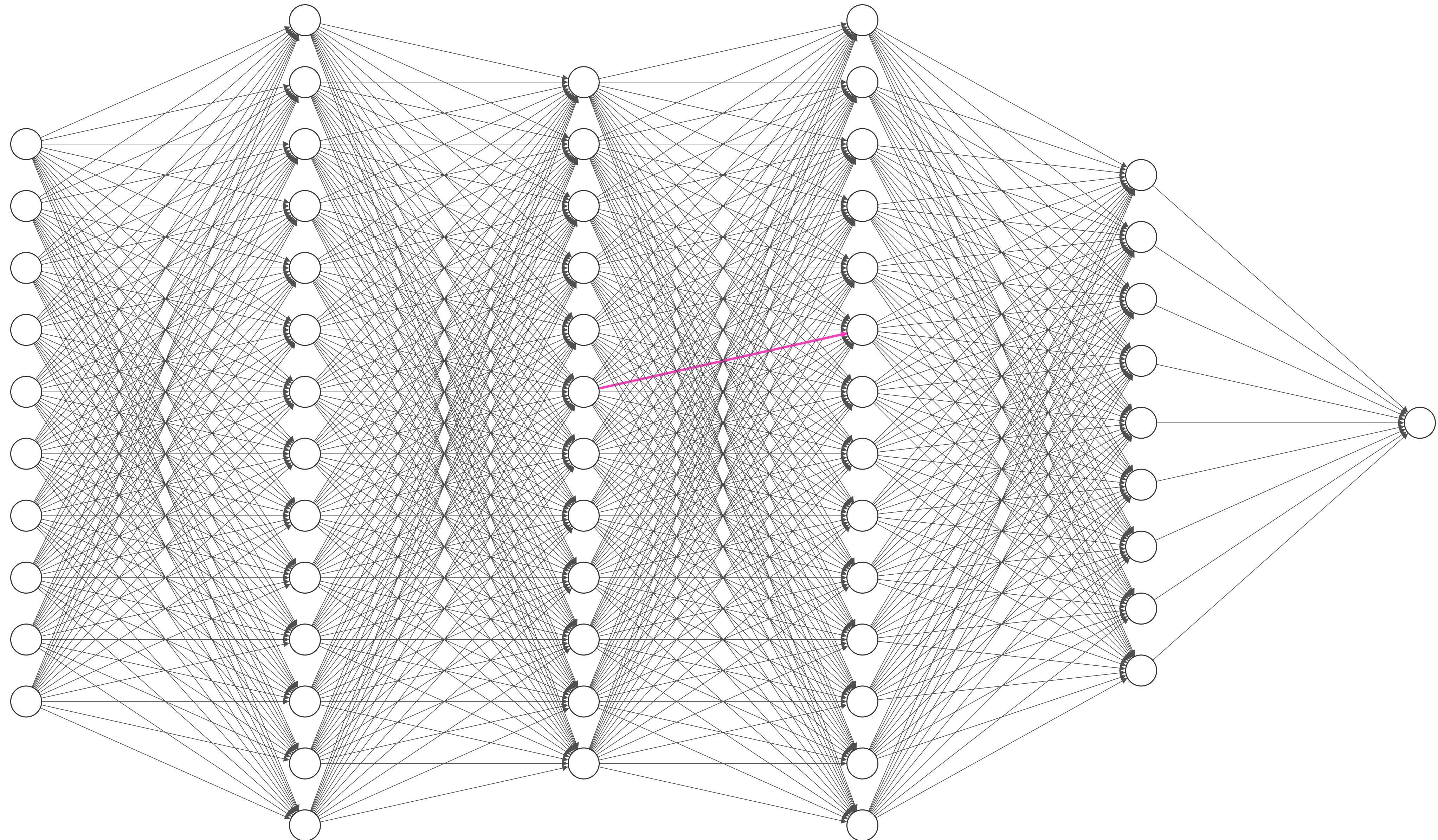
168

126

9

611





analytic vs numerical

analytic optimisation

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

analytic optimisation

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

numerical optimisation

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \boxed{\nabla_{\mathbf{w}} L(\mathbf{f}, \mathbf{X}, \mathbf{y}, \mathbf{w})}$$

$$\nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \vdots \\ \frac{\partial L}{\partial w_d} \end{bmatrix}$$

$$\frac{\partial L}{\partial w_i} = \lim_{\delta \rightarrow 0} \frac{L(..., w_i + \delta, ...) - L(..., w_i, ...)}{\delta}$$

numerical gradient

$$\frac{\partial L}{\partial w_i} \approx \frac{L(\dots, w_i + \delta, \dots) - L(\dots, w_i, \dots)}{\delta}$$

analytic gradient

$$\begin{aligned}
\frac{\partial}{\partial w_j} l(\mathbf{w}) &= \frac{\partial}{\partial w_j} \left[y \log(\sigma(\mathbf{x} \cdot \mathbf{w})) + (1 - y) \log(1 - \sigma(\mathbf{x} \cdot \mathbf{w})) \right] \\
&= \left[y \frac{1}{\sigma(\mathbf{x} \cdot \mathbf{w})} \frac{\partial}{\partial w_j} \sigma(\mathbf{x} \cdot \mathbf{w}) \right] + \left[(1 - y) \frac{1}{1 - \sigma(\mathbf{x} \cdot \mathbf{w})} \left(-\frac{\partial}{\partial w_j} \sigma(\mathbf{x} \cdot \mathbf{w}) \right) \right] \\
&= \left[y \frac{1}{\sigma(\mathbf{x} \cdot \mathbf{w})} \frac{\partial}{\partial w_j} \sigma(\mathbf{x} \cdot \mathbf{w}) \right] - \left[(1 - y) \frac{1}{1 - \sigma(\mathbf{x} \cdot \mathbf{w})} \frac{\partial}{\partial w_j} \sigma(\mathbf{x} \cdot \mathbf{w}) \right] \\
&= \left[y \frac{1}{\sigma(\mathbf{x} \cdot \mathbf{w})} - (1 - y) \frac{1}{1 - \sigma(\mathbf{x} \cdot \mathbf{w})} \right] \frac{\partial}{\partial w_j} \sigma(\mathbf{x} \cdot \mathbf{w}) \\
&= \left[y \frac{1}{\sigma(\mathbf{x} \cdot \mathbf{w})} - (1 - y) \frac{1}{1 - \sigma(\mathbf{x} \cdot \mathbf{w})} \right] \sigma(\mathbf{x} \cdot \mathbf{w})(1 - \sigma(\mathbf{x} \cdot \mathbf{w})) \frac{\partial}{\partial w_j} (\mathbf{x} \cdot \mathbf{w}) \\
&= (y(1 - \sigma(\mathbf{x} \cdot \mathbf{w})) - (1 - y)\sigma(\mathbf{x} \cdot \mathbf{w})) \cdot x_j \\
&= (y - y\sigma(\mathbf{x} \cdot \mathbf{w}) - \sigma(\mathbf{x} \cdot \mathbf{w}) + y\sigma(\mathbf{x} \cdot \mathbf{w})) \cdot x_j \\
&= (y - \sigma(\mathbf{x} \cdot \mathbf{w})) \cdot x_j \\
&= (y - \hat{y})x_j
\end{aligned}$$

the chain rule

$$\frac{dy}{dx} = \frac{dy}{du} \times \frac{du}{dx}$$

$$(f \circ g)' = (f' \circ g) \times g'$$

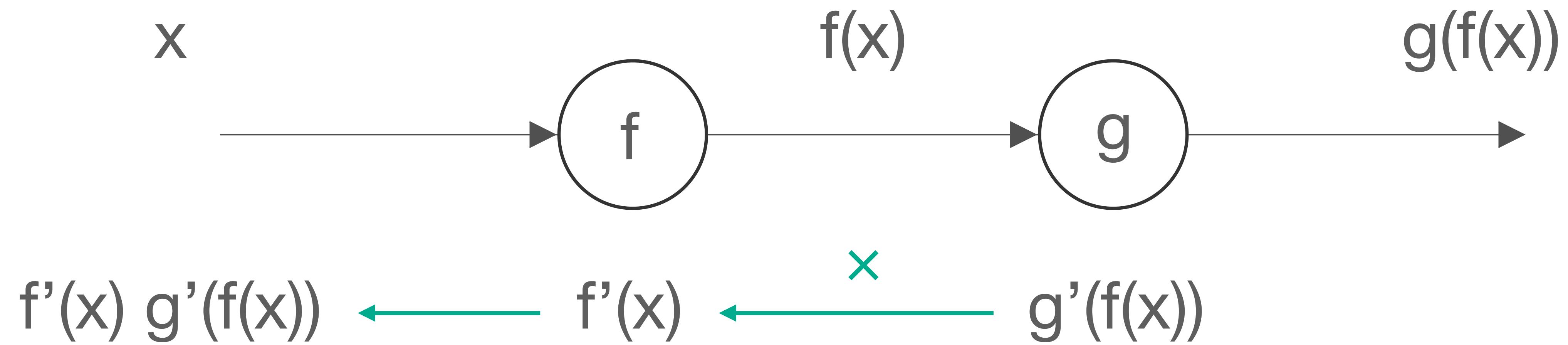
the chain rule

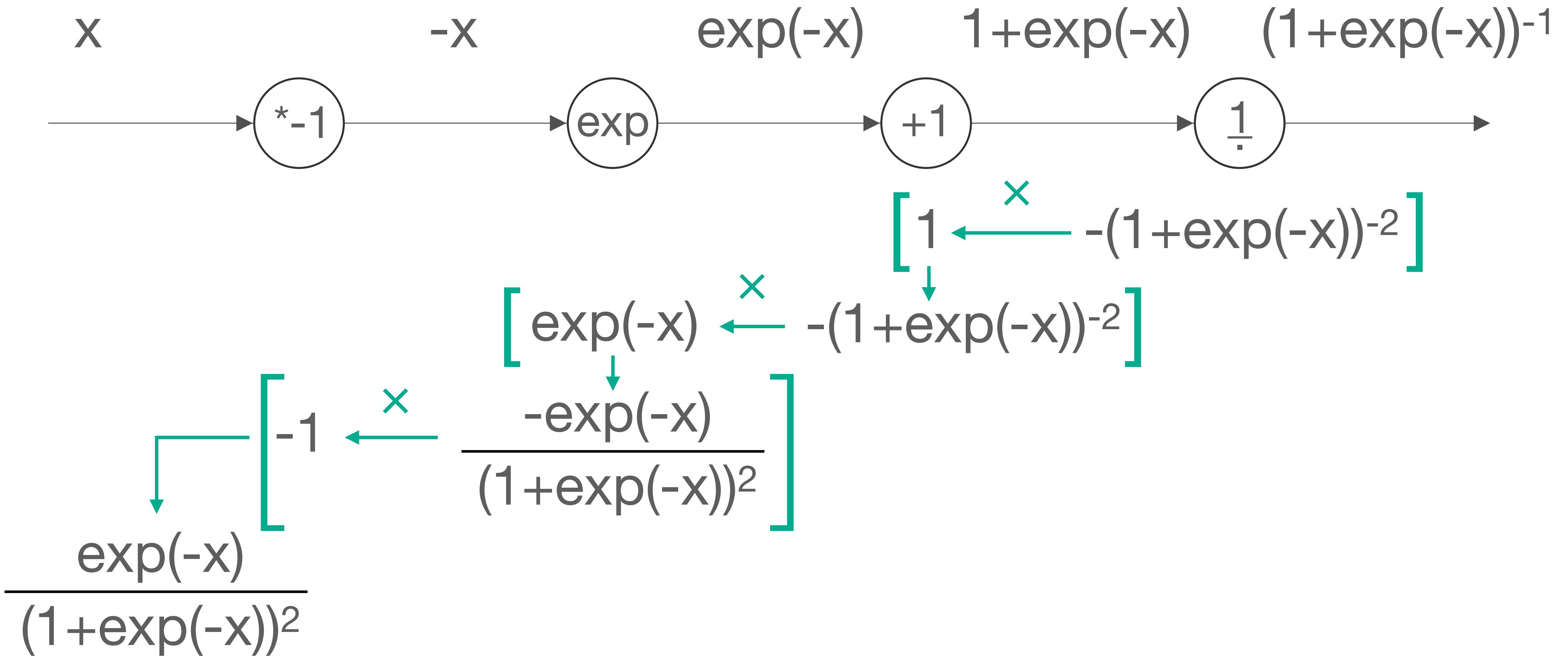
gradients multiply

$$[f(g(x))]' = f'(g(x)) \times g'(x)$$

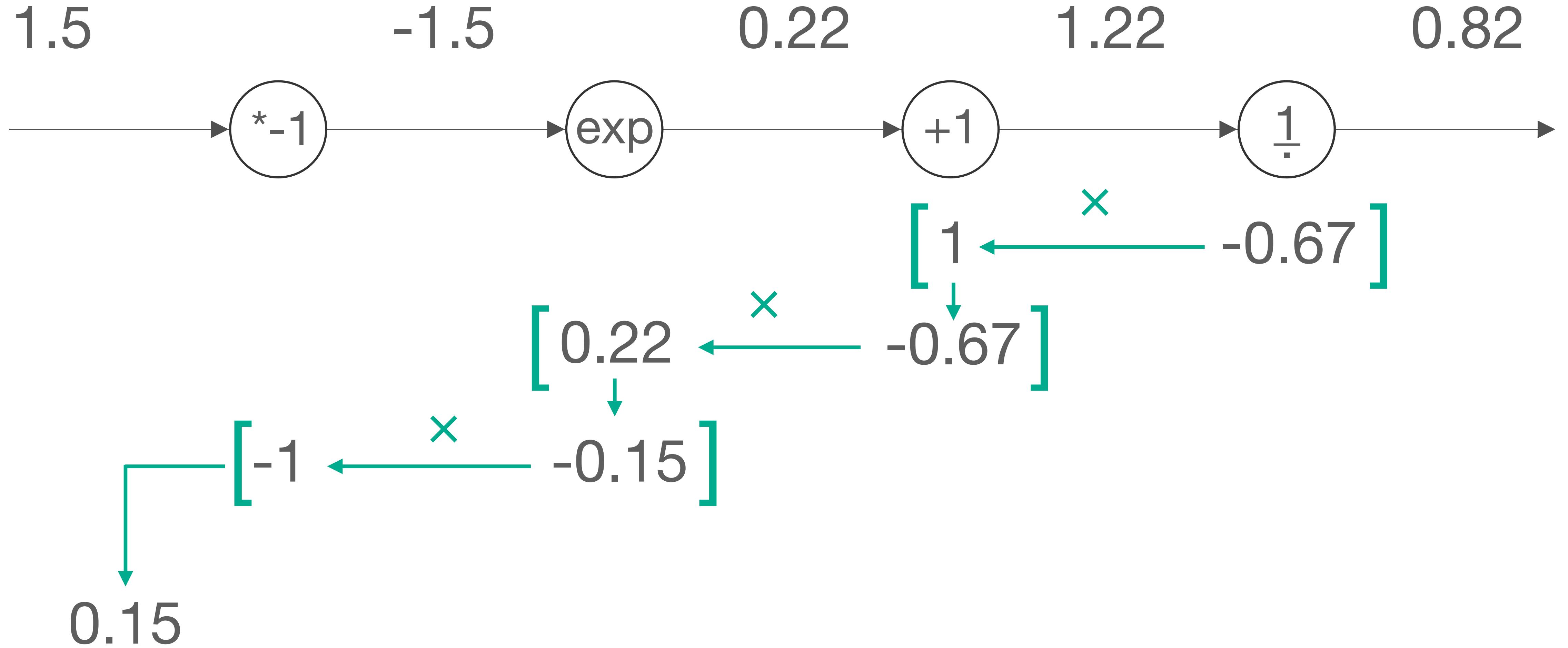
The diagram illustrates the application of the chain rule to the composite function $[f(g(x))]$. The expression is shown in black text. A blue arrow originates from the inner function $g(x)$, which is highlighted by an orange circle, and points to the term $g'(x)$ in the derivative expression. Another orange arrow originates from the outer function $f(g(x))$ and points to the term $f'(g(x))$ in the derivative expression.

$$[f(g(h(x)))]' = f'(g(h(x))) \times g'(h(x)) \times h'(x)$$





gradients are just numbers!



```
out1 = f1(x)
out2 = f2(out1)
out3 = f3(out2)
out4 = f4(out3)
loss = f5(out4)
```

```
dloss_out4 = df5(out4)
dloss_out3 = df4(out3) * dloss_out4
dloss_out2 = df3(out2) * dloss_out3
dloss_out1 = df2(out1) * dloss_out2
dloss_x = df1(x) * dloss_out1
```

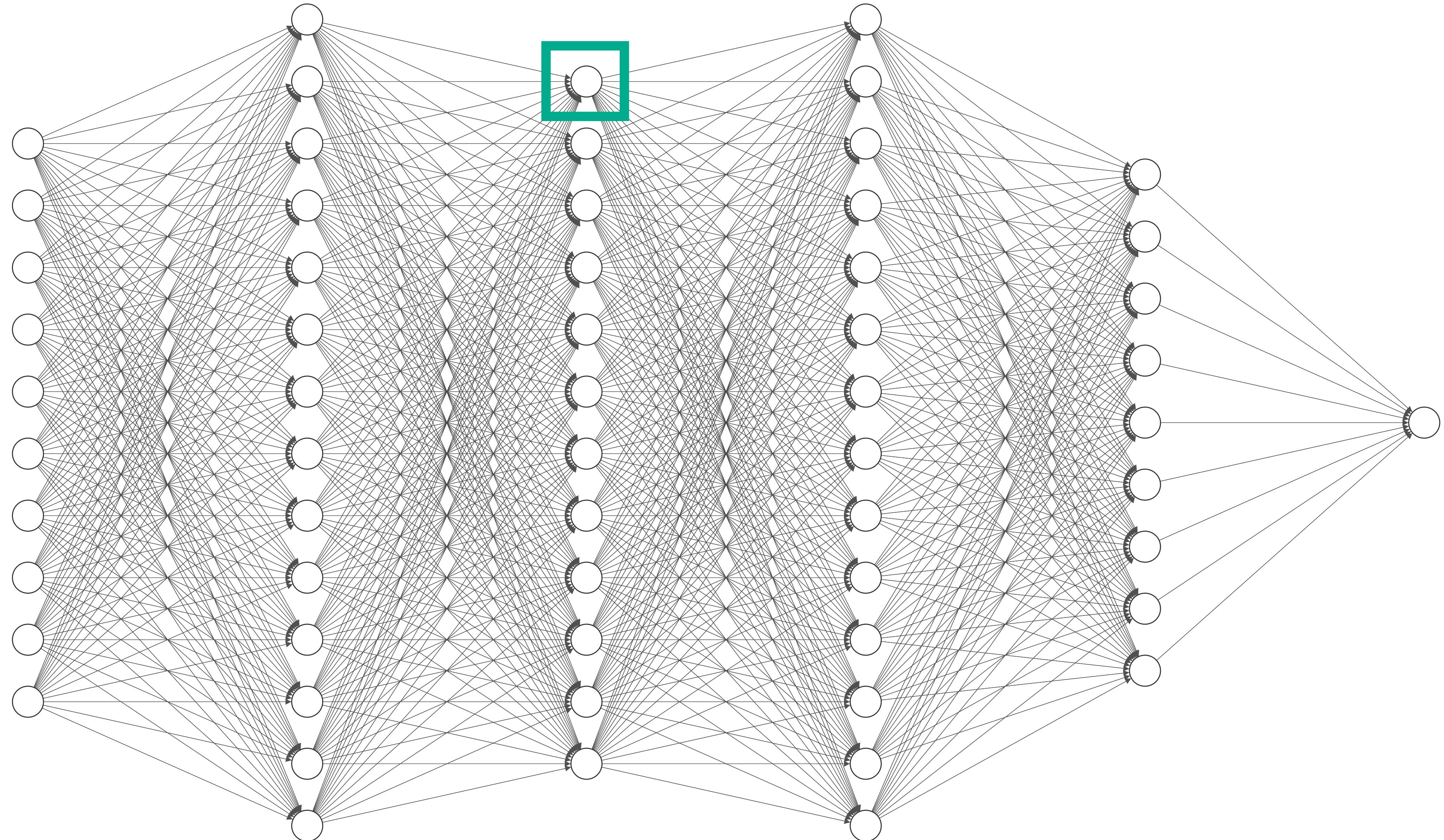
```
out1 = f1(x)
out2 = f2(out1)
out3 = f3(out2)
out4 = f4(out3)
loss = f5(out4)
```

local gradient

downstream gradient

```
dloss_out4 = df5(out4)
dloss_out3 = df4(out3) * dloss_out4
dloss_out2 = df3(out2) * dloss_out3
dloss_out1 = df2(out1) * dloss_out2
dloss_x = df1(x) * dloss_out1
```

analytic gradients, numerically
backpropagation



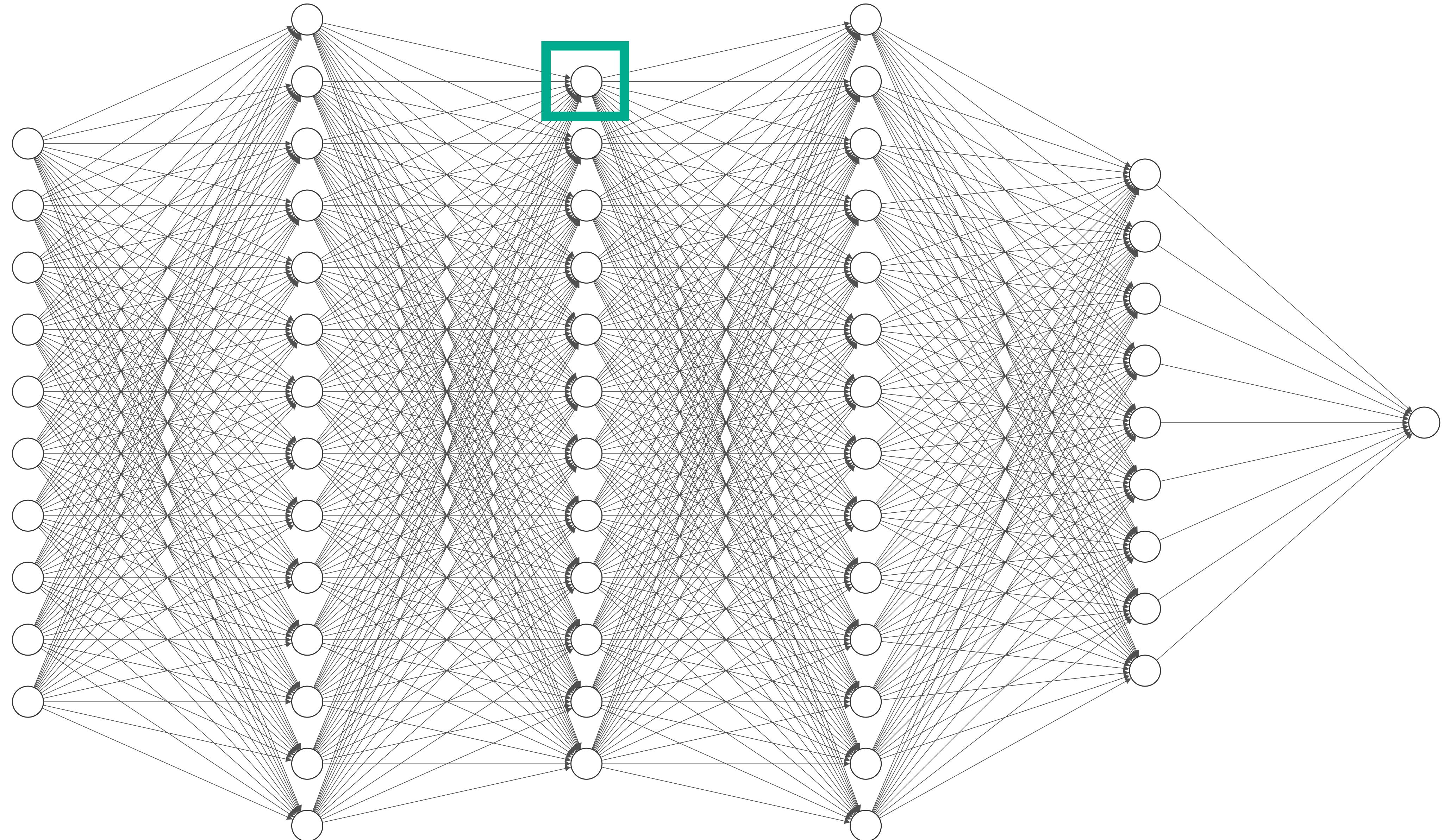
$$z = \mathbf{x} \cdot \mathbf{w}$$

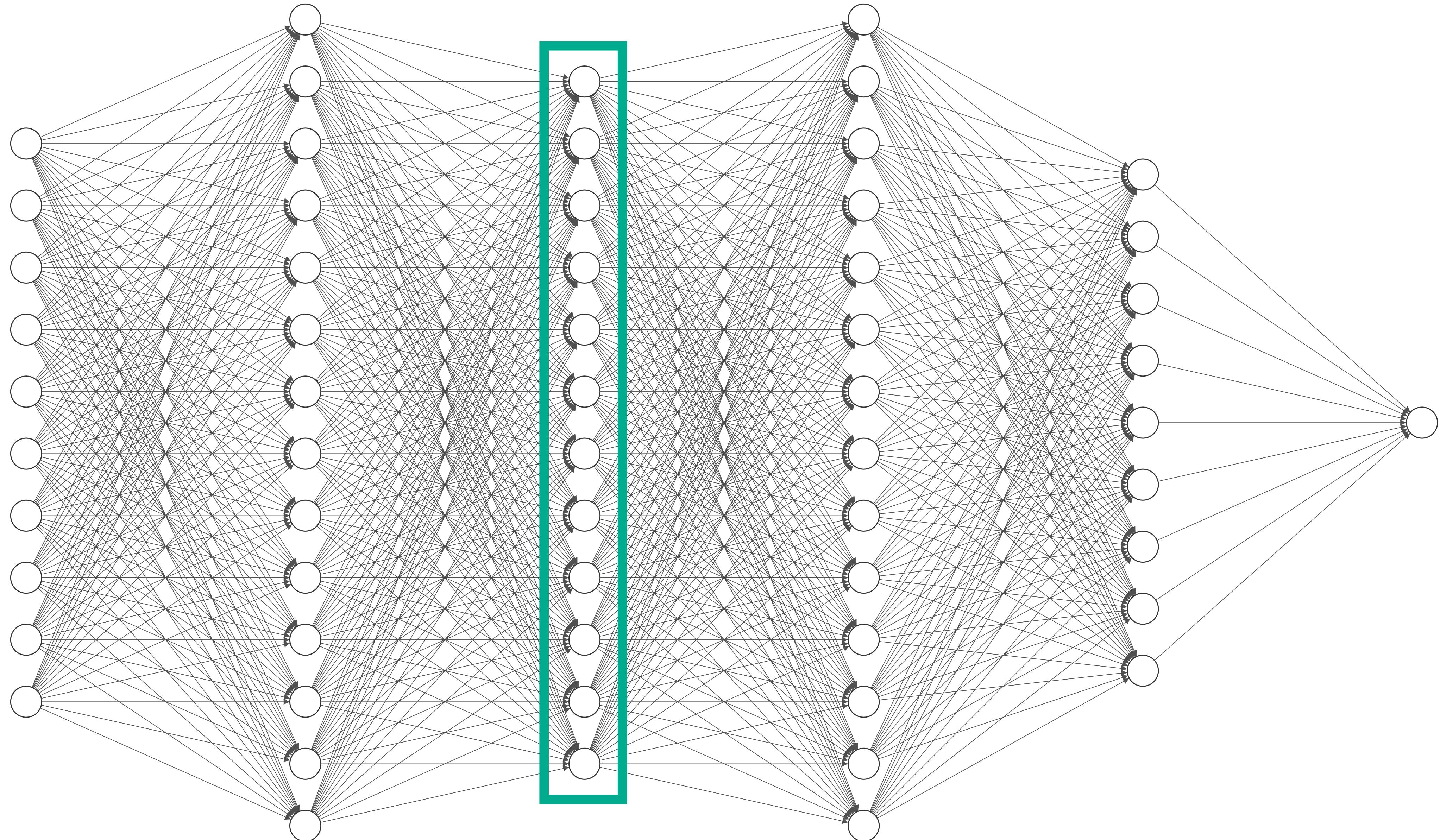
$$\nabla_{\mathbf{x}} z = \mathbf{w}$$

$$\nabla_{\mathbf{w}} z = \mathbf{x}$$

$$a(\mathbf{x}) = [f(x_1), f(x_2), \dots, f(x_d)]$$

$$\nabla_{\mathbf{x}} a = [f'(x_1), f'(x_2), \dots, f'(x_d)]$$





forward pass

- evaluate model on data**
- calculate local gradients**
- save ins and outs**
- calculate final loss**

backward pass

**calculate final loss gradient
propagate gradients backward
multiply downstream by local**

optimise

update weights with gradients



Glorot initialisation for tanh

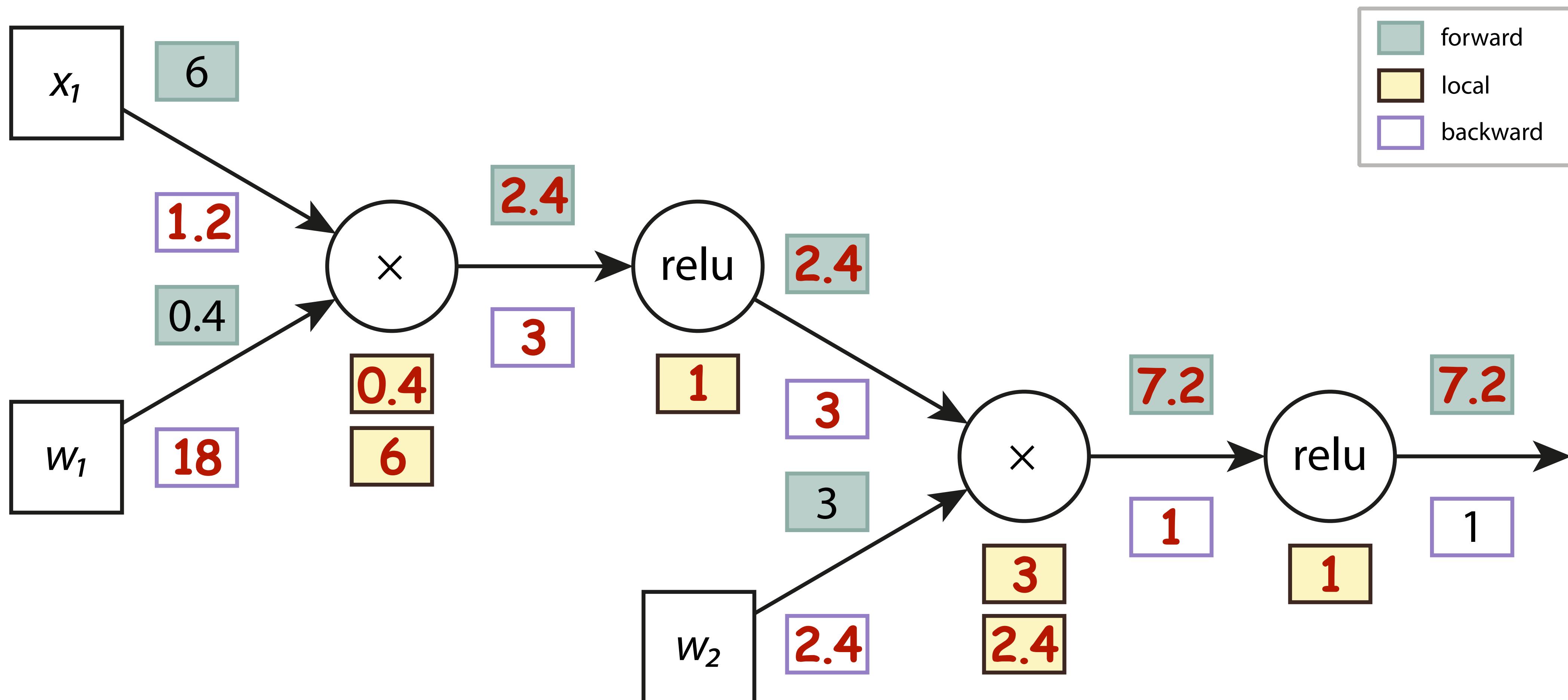
$$w \sim N\left(0, \frac{2}{\text{fan}_{in} + \text{fan}_{out}}\right)$$

$$w \sim U\left(-\sqrt{\frac{6}{\text{fan}_{in} + \text{fan}_{out}}}, \sqrt{\frac{6}{\text{fan}_{in} + \text{fan}_{out}}}\right)$$

He initialisation for ReLU

$$w \sim N\left(0, \frac{2}{\text{fan}_{in}}\right)$$

$$w \sim U\left(-\sqrt{\frac{6}{\text{fan}_{in}}}, \sqrt{\frac{6}{\text{fan}_{in}}}\right)$$



Questions?

Next: More Neural Networks

