Problem Set 2 – Shallow and Deep Networks

DS542 – DL4DS

Spring, 2025

Note: Refer to the equations in the *Understanding Deep Learning* textbook to solve the following problems.

① Problem 3.2

For each of the four linear regions in Figure 3.3j, indicate which hidden units are inactive and which are active (i.e., which do and do not clip their inputs).

② Problem 3.5

Prove that the following property holds for $\alpha \in \mathbb{R}^{+}$:

$$\text{ReLU}[\alpha \cdot z] = \alpha \cdot \text{ReLU}[z].$$

This is known as the non-negative homogeneity property of the ReLU function.

③ Problem 4.6

Consider a network with $D_i = 1$ input, $D_o = 1$ output, $K = 10$ layers, and $D = 10$ hidden units in each. Would the number of weights increase more – if we increased the depth by one or the width by one? Provide your reasoning.

(2) For $\alpha \in \mathbb{R}^+$: prove: $ReLu(z \cdot \alpha) = \alpha \cdot ReLu(z)$

- if $z \geq 0$ and $\alpha > 0$:

  $ReLu(z \cdot \alpha) = max(0, \alpha \cdot z) = \alpha \cdot z$

  because $\alpha \cdot z \geq 0$

- if $z < 0$ and $\alpha > 0$:

  $ReLu(z \cdot \alpha) = max(0, \alpha \cdot z)$

  since $z < 0$ then $\alpha \cdot z < 0$

  therefore $ReLu(z \cdot \alpha) = 0$

  same if $\alpha \cdot ReLu(z) = \alpha \cdot max(0, \overset{z<0}{z}) = \alpha \cdot 0 = 0$

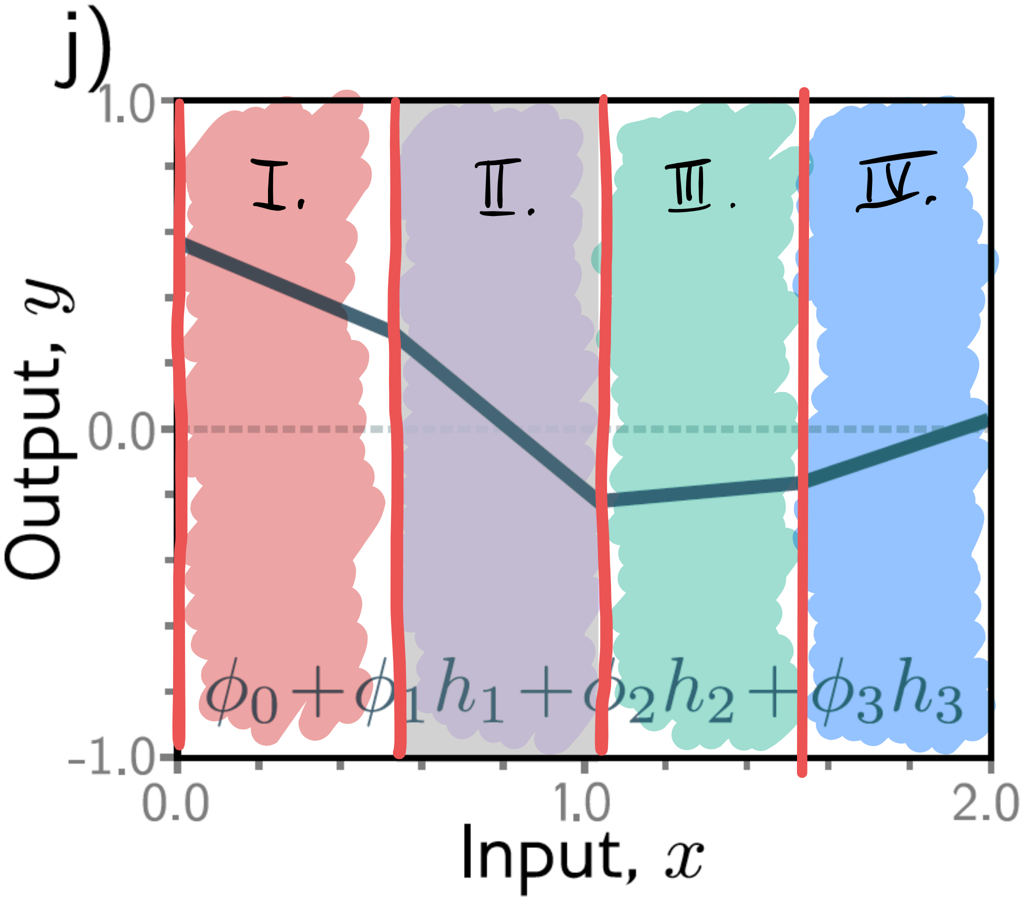
(1)

-in textbook colors:

a → 〰

b → 〰

c → 〰



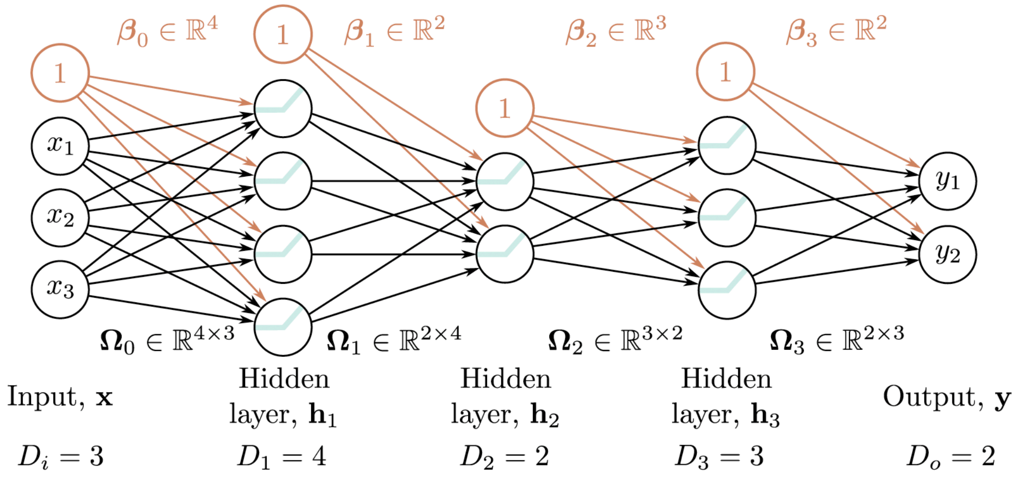
I. a,b inactive and c is active

II. b inactive and a,c is active

III. a,b,c active

IV. c inactive and a,b is active

3



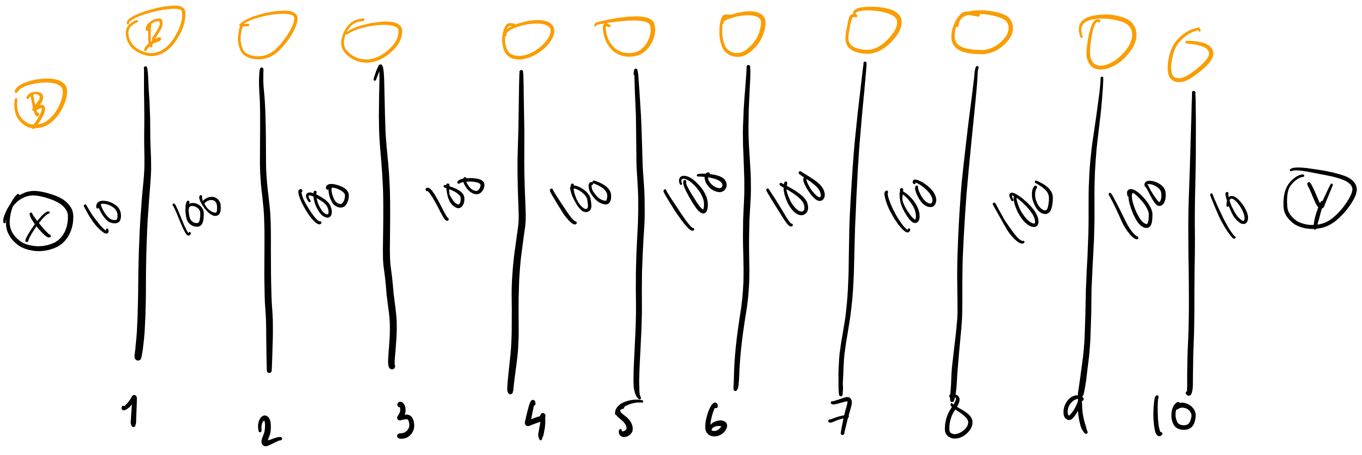
$\beta_0 \in \mathbb{R}^4$
$\beta_1 \in \mathbb{R}^2$
$\beta_2 \in \mathbb{R}^3$
$\beta_3 \in \mathbb{R}^2$
1
x_1
x_2
x_3
y_1
y_2
$\Omega_0 \in \mathbb{R}^{4\times3}$
$\Omega_1 \in \mathbb{R}^{2\times4}$
$\Omega_2 \in \mathbb{R}^{3\times2}$
$\Omega_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
Hidden layer, $\mathbf{h}_1$
Hidden layer, $\mathbf{h}_2$
Hidden layer, $\mathbf{h}_3$
Output, $\mathbf{y}$
$D_i = 3$
$D_1 = 4$
$D_2 = 2$
$D_3 = 3$
$D_o = 2$

Figure 4.6 Matrix notation for network with $D_i = 3$-dimensional input $\mathbf{x}$, $D_o = 2$-dimensional output $\mathbf{y}$, and $K = 3$ hidden layers $\mathbf{h}_1, \mathbf{h}_2$, and $\mathbf{h}_3$ of dimensions $D_1 = 4$, $D_2 = 2$, and $D_3 = 3$ respectively. The weights are stored in matrices $\boldsymbol{\Omega}_k$ that pre-multiply the activations from the preceding layer to create the pre-activations at the subsequent layer. For example, the weight matrix $\boldsymbol{\Omega}_1$ that computes the pre-activations at $\mathbf{h}_2$ from the activations at $\mathbf{h}_1$ has dimension 2×4. It is applied to the four hidden units in layer one and creates the inputs to the two hidden units at layer two. The biases are stored in vectors $\boldsymbol{\beta}_k$ and have the dimension of the layer into which they feed. For example, the bias vector $\boldsymbol{\beta}_2$ is length three because layer $\mathbf{h}_3$ contains three hidden units.

$D_i = 1$
$D_o = 1$ output

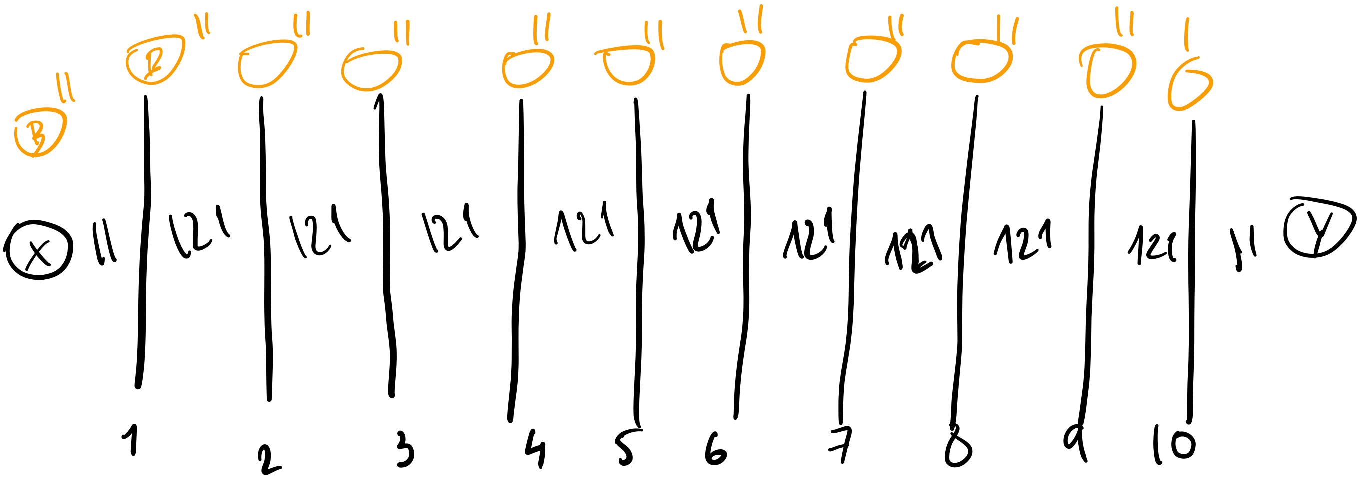
$K = 10$ layers
$D = 10$ h in each

current weight:



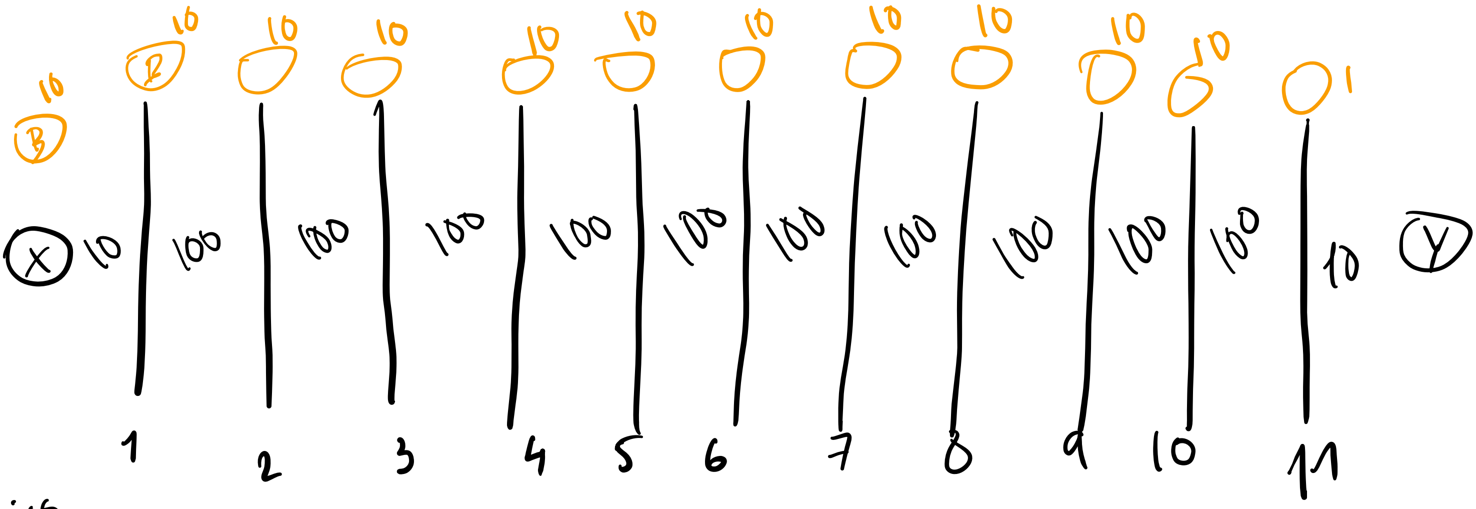
101 + 20 + 900 = 1021 original

adding width (1 more node in each layer):



$112 + 22 + 121 * 9 = 1223$

adding layer (depth)



$\overset{\text{Bias}}{\boxed{111}} + 20 + 1000 = 1131$

Increasing the width by 1 increases the weight more
than when adding 1 more layer of depth.

This is because adding node to each layer affects the whole network whereas increasing the depth adds weight for only 1 more layer