$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} w_1 & b_1 \\ w_2 & b_2 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}$$

(2 X 2)*(2 X 1)

$$\begin{bmatrix} z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} w_3 & w_5 & b_3 \\ w_4 & w_6 & b_4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ 1 \end{bmatrix}$$

(2 X 2+1)*(2+1 X 1)

$$[\hat{y}] = \begin{bmatrix} w_7 & w_8 & b_5 \end{bmatrix} \begin{bmatrix} a_3 \\ a_4 \\ 1 \end{bmatrix}$$

(1 X 2+1)*(2+1 X 1)

Training loop

1. Draw a batch of training samples x and corresponding targets y
2. Run the network on x (this is called "forward pass”) obtain predictions y_pred
3. Compute the "loss" of the network on the batch, a measure of the mismatch between y_pred and y
4. **Update all weights of the network in a way that slightly reduces the loss on this batch:**

    4.1. Compute the gradient of the loss with regard to the parameters of the network (this is called "backward pass")

    4.2. Move the parameters a little in the direction opposite to the gradient, thus lowering the loss on the batch by a bit.