S&DS 365 / 565

Data Mining and Machine Learning

Yale

For today

Recap, go to notebook

Neural networks: Fully-connected feed-forward and backpropagation

Non-linear methods

Feature mappings

Advanced optimization

Recap

Optimization: gradient descent, newton's method, stochastic gradient descent

SGD Goal:

$$\arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

SGD Algo:

$$\theta_k = \theta_{k-1} - \eta_{k-1} \nabla f_J(\theta_{k-1})$$

$J \sim [n]$

Recap

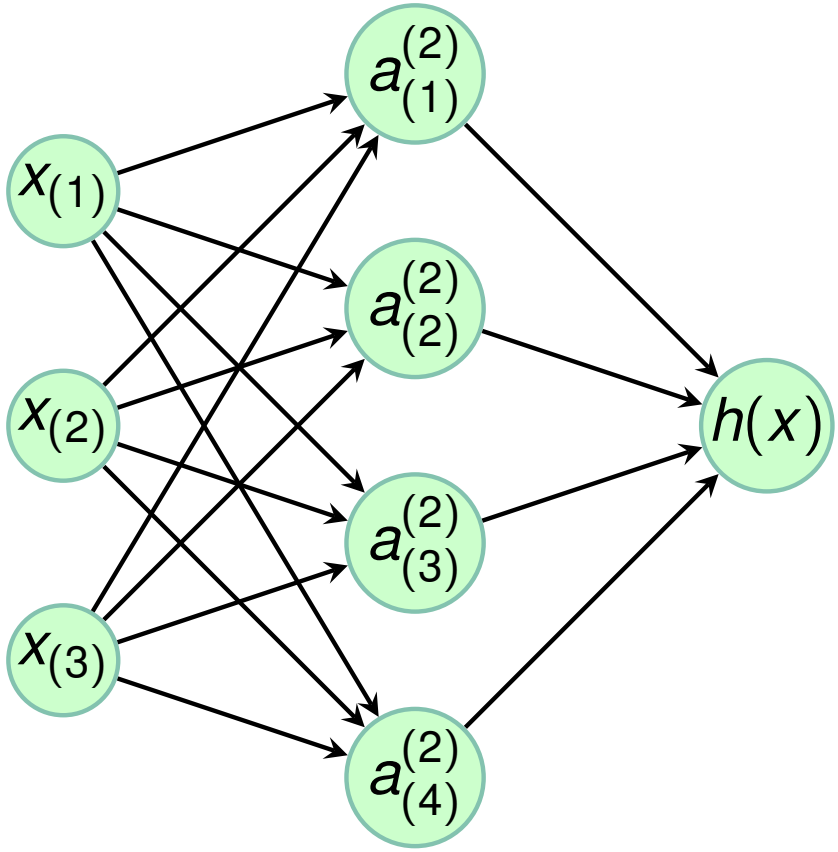
Example: $\{x_i, y_i\}_{i=1}^n$ with logistic

$$f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[\log(1 + \exp(x_i^T \theta)) - y_i x_i^T \theta \right]$$

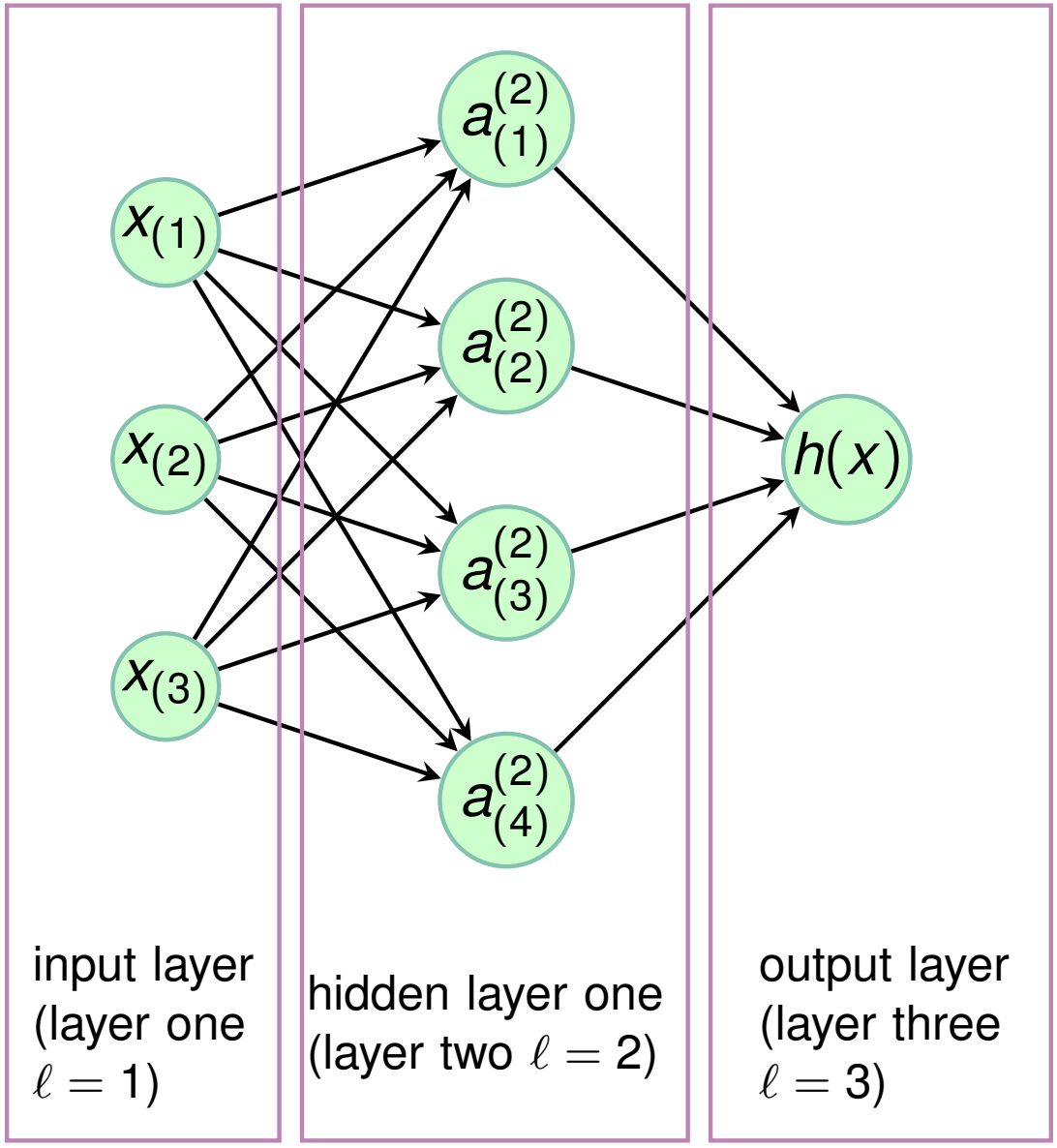
$$f_i(\theta) = \log(1 + \exp(x_i^T \theta)) - y_i x_i^T \theta$$

$$\theta_{k+1} = \theta_k - \eta_k \left[\frac{\exp(x_J^T \theta_k)}{1 + \exp(x_J^T \theta_k)} - y_J \right] x_J$$

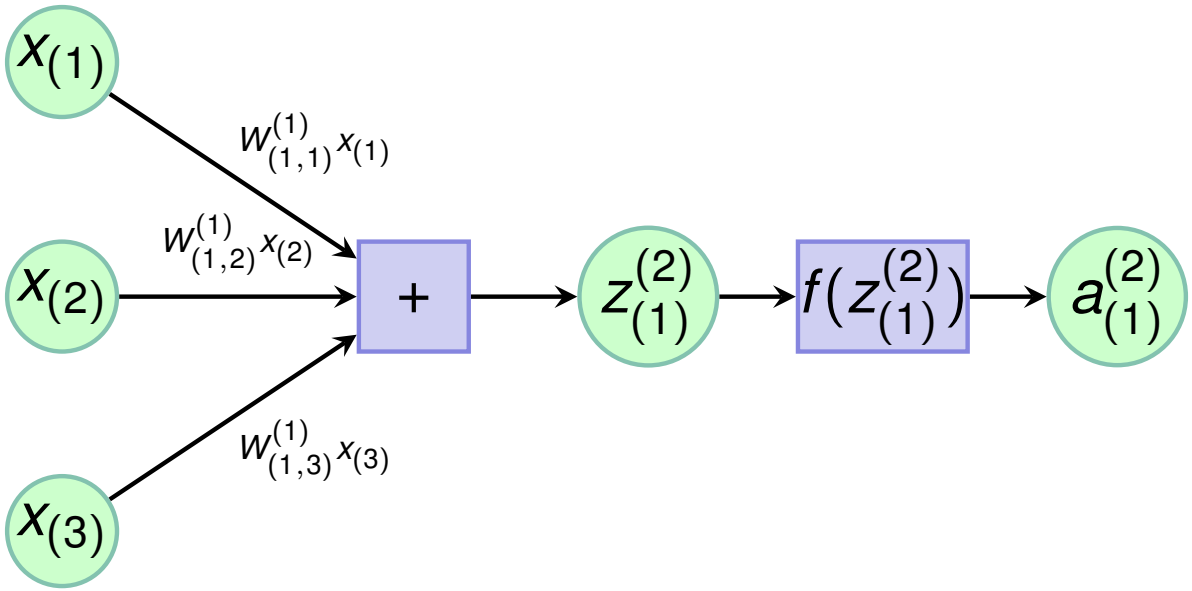
Neural network



Neural network



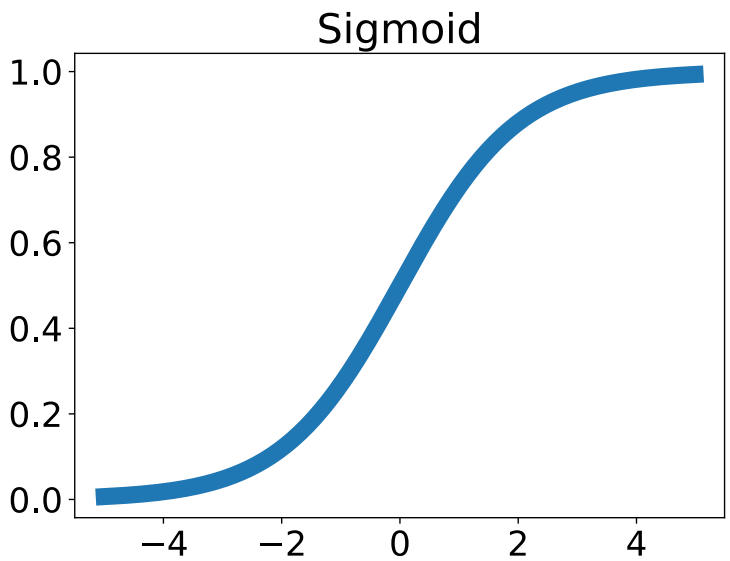
Neural network

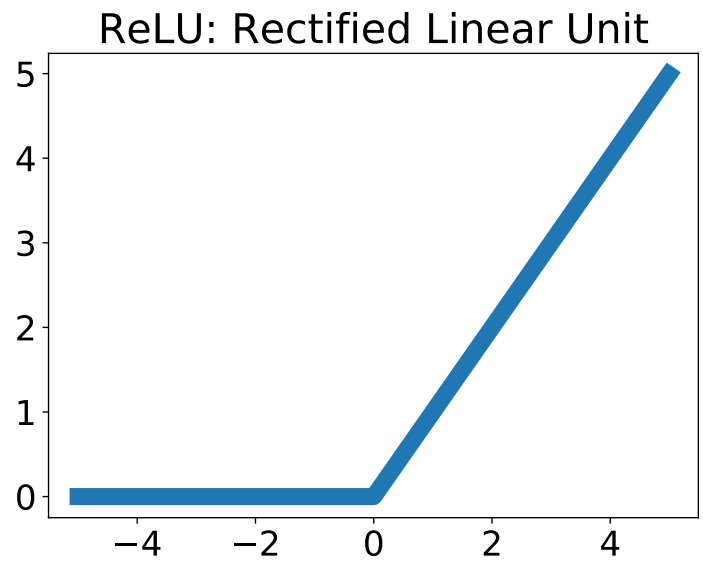


Let $w^1_1 = W^{(1)}_{(1,:)} \in \mathbb{R}^3$

$$a^2_{(1)} = f(x^T w^1_1) = f(\langle x, w^1_1 \rangle)$$

f is activation function





Neural network: three layers $L = 3$

Weights of layers

$$W^{(1)} \in \mathbb{R}^{d_2 \times d}$$

$$W^{(2)} \in \mathbb{R}^{1 \times d_2}$$

$x \in \mathbb{R}^d$ $\quad f: \mathbb{R}^d \to \mathbb{R}^{d_2}$

$b^{(1)} \in \mathbb{R}^{d_2}$ $\quad b^{(2)} \in \mathbb{R}$ $\quad h: \mathbb{R}^d \to \mathbb{R}$

$$h(x) = W^{(2)} f(W^{(1)} x + b^{(1)}) + b^{(2)}$$

Note

$$[f(v)]_{(i)} = f(v_{(i)})$$

If f non-linear $\in \mathbb{R}^{1 \times d}$: e.g. $f(v) = v$

$\Rightarrow$ $h(x) = \underbrace{W^{(2)} W^{(1)}}_{\in \mathbb{R}^{1 \times d}} x + \left(W^{(2)} b^{(1)} + b^{(2)} \right)$

Linear function!

Neural Network simple example $L = 2$

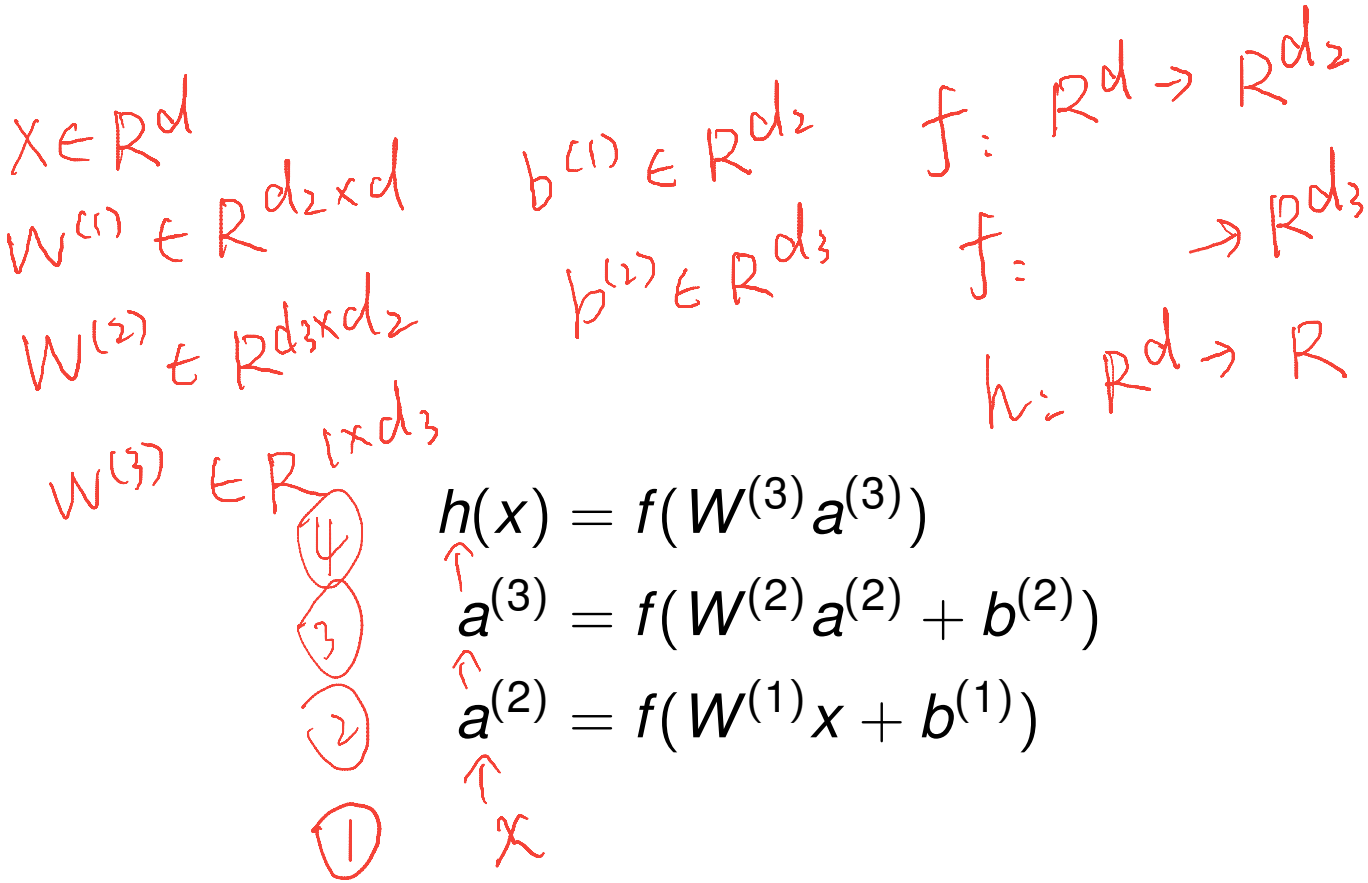
$W^{(1)} \in \mathbb{R}^{1 \times d}$

$x \in \mathbb{R}^d$

$h : \mathbb{R}^d \to \mathbb{R}$

$$h(x) = W^{(1)} x$$

Neural network: four layers $L = 4$



Neural network: more general outputs

$W^{(3)} \in \mathbb{R}^{d_4 \times d_3}$

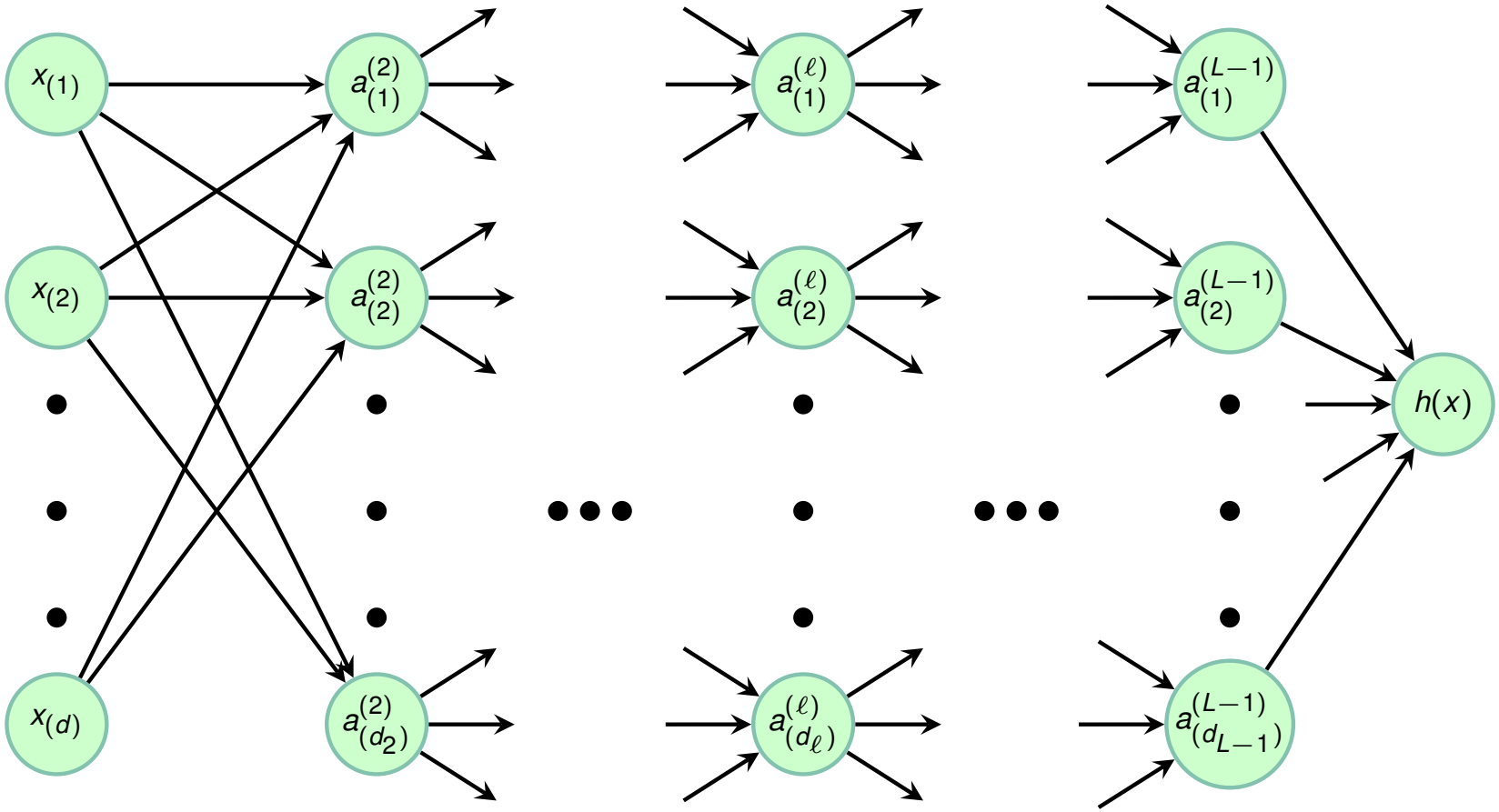
vector

$$h(x) = f(W^{(3)} a^{(3)})$$
$$a^{(3)} = f(W^{(2)} a^{(2)} + b^{(2)})$$
$$a^{(2)} = f(W^{(1)} x + b^{(1)})$$

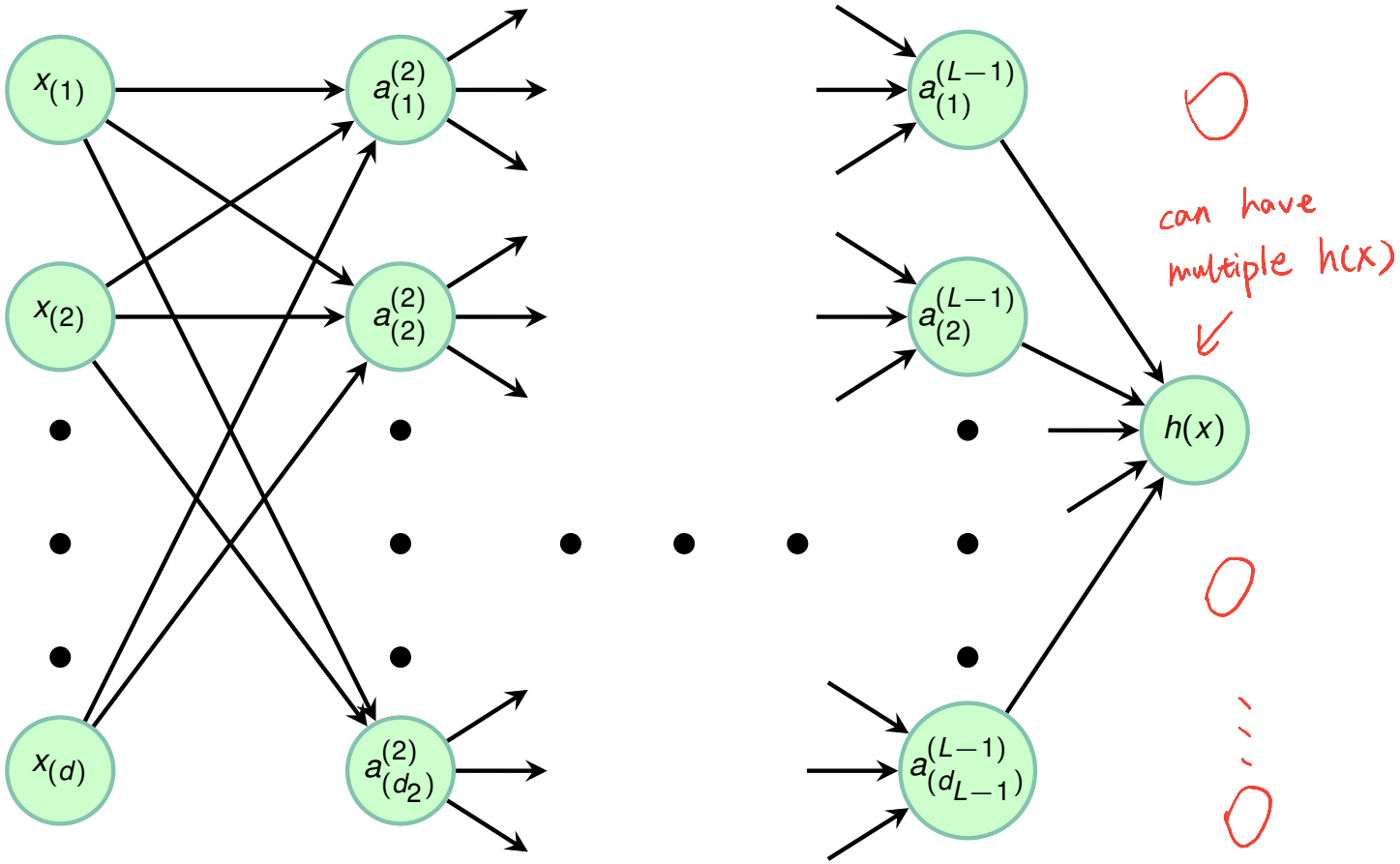
Neural Networks

Fully-connected feed-forward

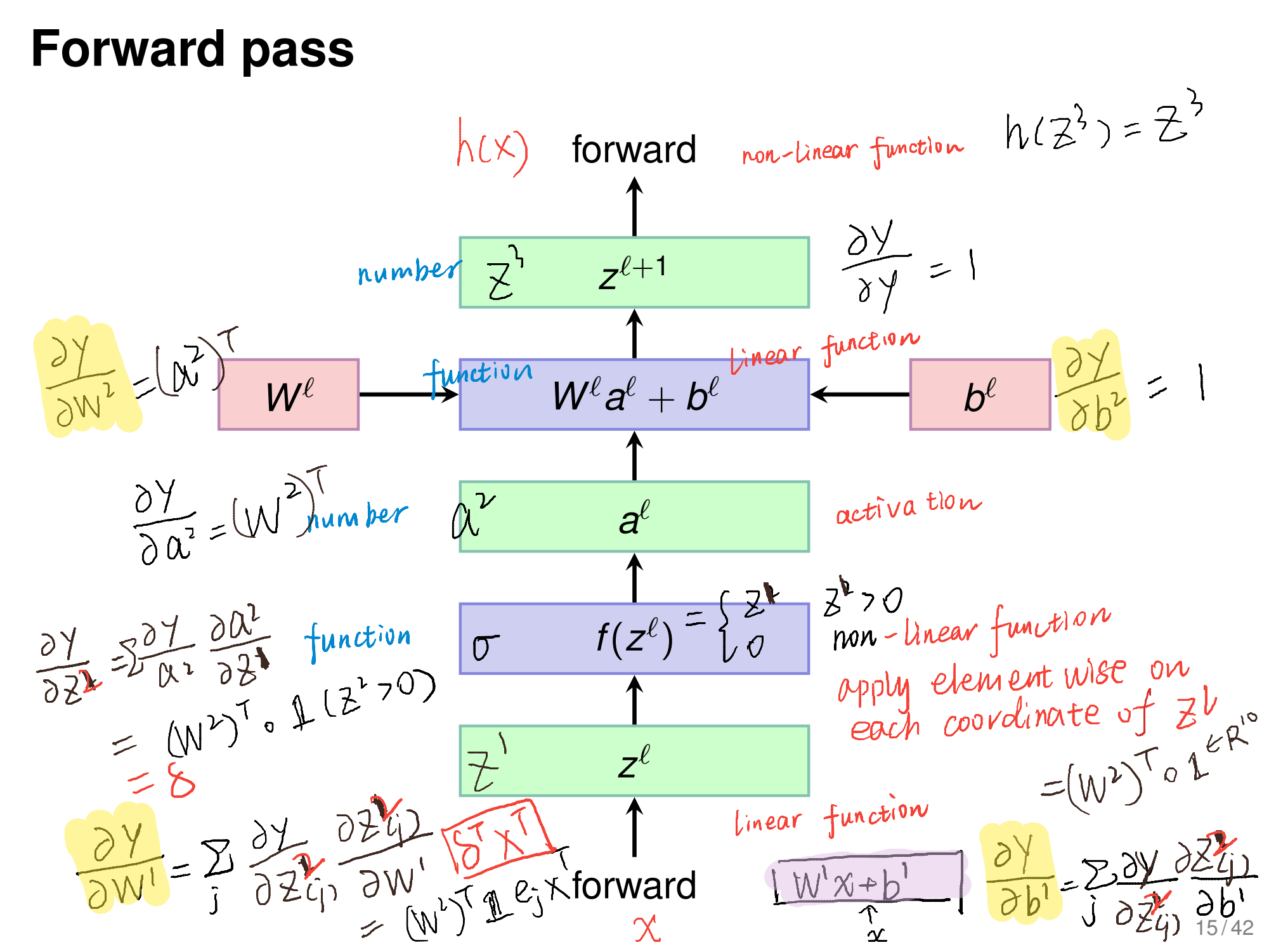
Neural network: L layers



Neural network



Forward pass



Neural network: math

Parameters: $W^{(\ell)} \in \mathbb{R}^{d_{\ell+1} \times d_\ell}$, $b^{(\ell)} \in \mathbb{R}^{d_{\ell+1}}$

Input: $x \in \mathbb{R}^d$ and $d = d_1$

Output: $h(x) = z^{(L)}$ and $d_L \geq 1$

$$z^{(2)} = W^{(1)} x + b^{(1)}$$

$$a^{(2)} = f(z^{(2)})$$ f: nonlinear function

$$a^{(k)} = f(z^{(k)})$$

$$z^{(k+1)} = W^{(k)} a^{(k)} + b^{(k)}$$

we also train neural network by gradient descent, so need to compute gradient

How do we compute gradients?

$$\begin{cases} \dfrac{\partial h_{W,b}(x)}{W^{(\ell)}} \\ \dfrac{\partial h_{W,b}(x)}{b^{(\ell)}} \end{cases}$$

$W^{(l)}$, $b^{(l)}$ are params

though W and b may not in function h h still depend on W and b

loss — a choice: mean squared loss

$$\Rightarrow L(W,b) = \frac{1}{2n}\sum_{i=1}^{n}\left(h(x_i)-y_i\right)^2$$

gradient — chain rule

$$\Rightarrow \frac{\partial L}{\partial W} = \frac{1}{n}\sum_{i=1}^{n}\underbrace{\left(h(x_i)-y_i\right)}_{\text{error}_i}\frac{\partial h(x_i)}{\partial W}$$

对矩阵求梯度 没看

subscripts W and b included above

generally will drop them

Recall chain rule

$g : \mathbb{R}^p \mapsto \mathbb{R}$ and $f : \mathbb{R}^d \mapsto \mathbb{R}^p$

multivarible vector

intermediatary varible v

Let $v = f(\beta)$ and $y = g(v)$

change β of i change every coordinate of v of j

sum up for multi-D $\in \mathbb{R}^p$

one form

$$\frac{\partial y}{\partial \beta_{(i)}} = \sum_{j=1}^{p} \frac{\partial y}{\partial v_{(j)}} \frac{\partial v_{(j)}}{\partial \beta_{(i)}}$$

gradient

Jacobian of f

$$= \sum_{j=1}^{p} \left[\nabla g(v) \Big|_{v=f(\beta)} \right]_{(j)} \frac{\partial f(\beta)_{(j)}}{\partial \beta_{(i)}}$$

forward pass

another form

$$\frac{\partial g(f(\beta))}{\partial \beta_{(i)}} = \sum_{j=1}^{p} \frac{\partial g(f(\beta))}{\partial [f(\beta)]_{(j)}} \frac{\partial [f(\beta)]_{(j)}}{\partial \beta_{(i)}}$$

Let $v, w \in \mathbb{R}^d$

Recall: $f(v) = v^T w \Rightarrow \nabla f(v) = \frac{\partial f(v)}{\partial v} = w$

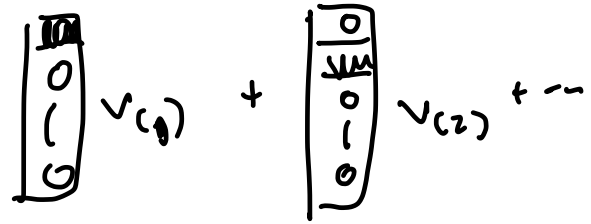
Notation: $z = w \circ v$ point-wise multiplication Hadamard product

$$z_{(i)} = w_{(i)} v_{(i)}$$

standard base vector only jth coordinate is non-zero

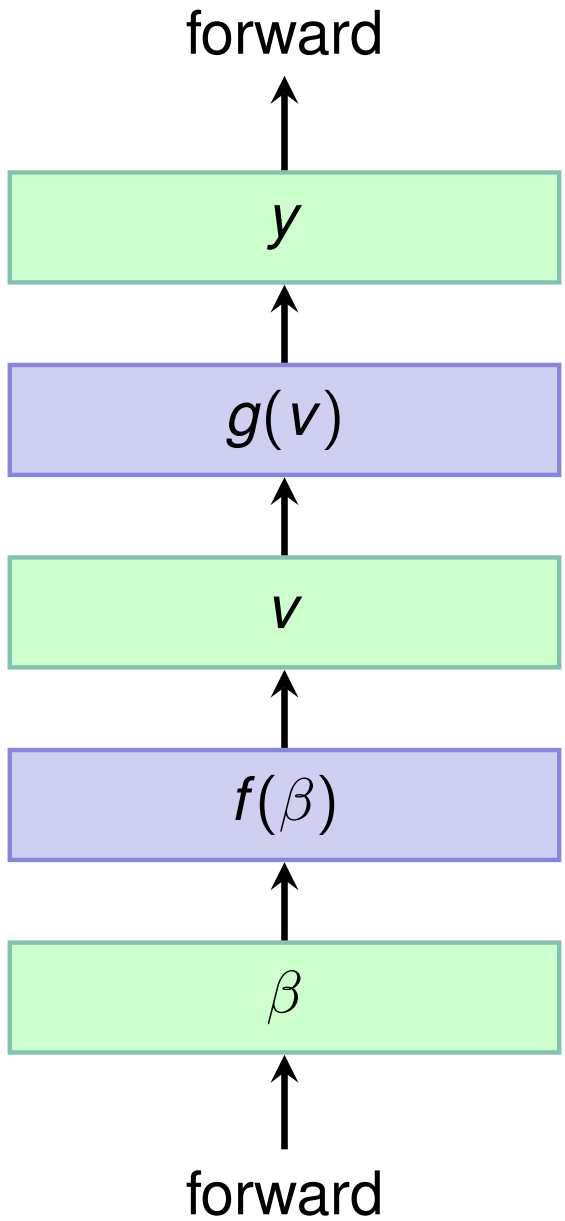
Let $e_j \in \mathbb{R}^d$ such that $[e_j]_{(i)} = \mathbb{1}(i = j) \Rightarrow e_j^T v = v_{(j)}$ ← jth coordinate of v

$$v = \sum_j e_j v_{(j)}$$

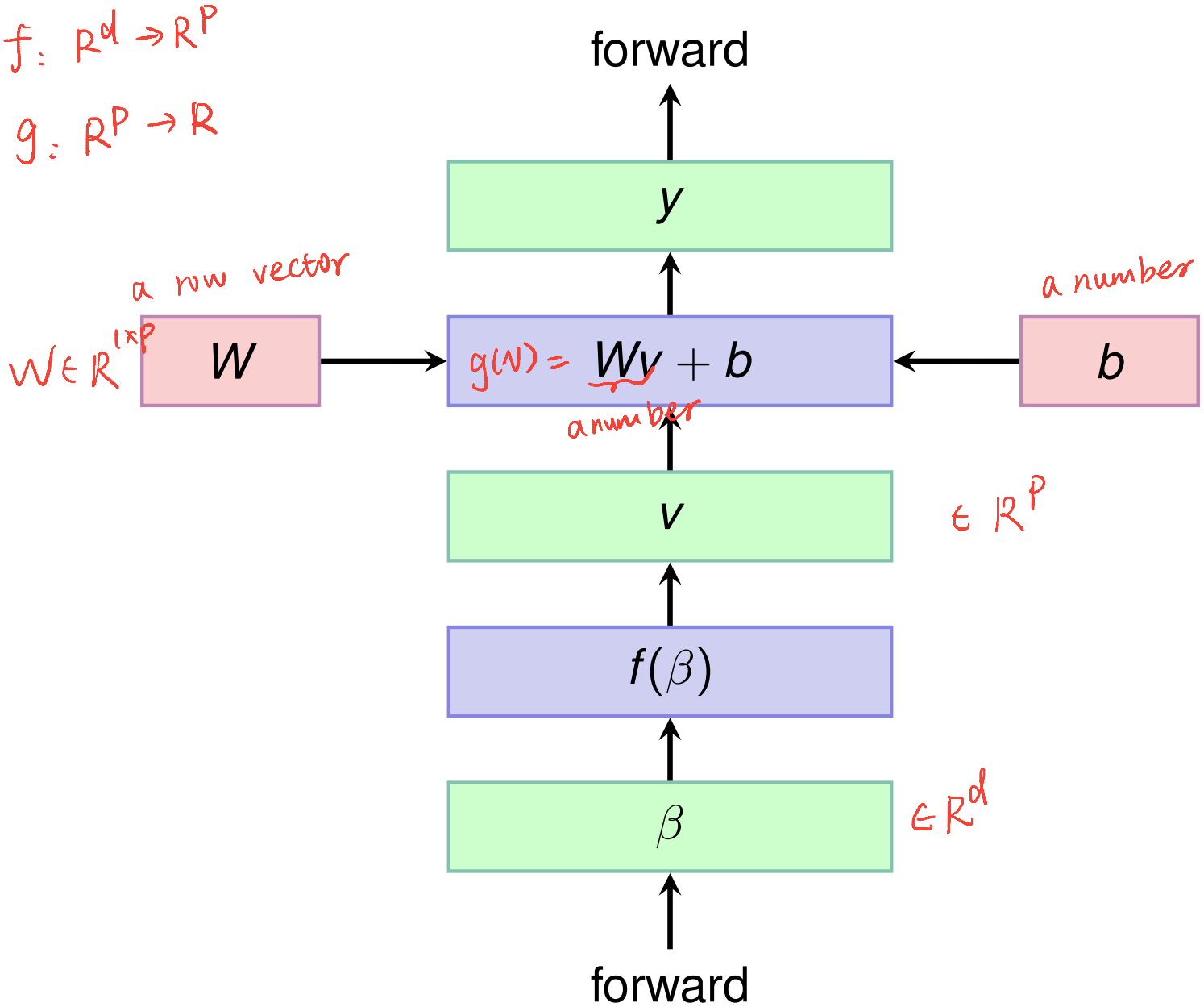


$$v \circ w = \sum_j e_j v_{(j)} w_{(j)}$$

Graphically



Example


$f: R^d \to R^p$
$g: R^p \to R$
forward
y
a row vector
$W \in R^{1 \times p}$
W
$g(W) = Wv + b$
a number
b
a number
v
$\in R^p$
$f(\beta)$
β
$\in R^d$
forward

Mathematically

$$y = Wf(\beta) + b$$

$$W \in \mathbb{R}^{1 \times p}$$
$$b \in \mathbb{R}$$
$$\beta \in \mathbb{R}^{d}$$

Gradient

compute

$$y = Wf(\beta) + b$$

Want: $\frac{\partial y}{\partial W}, \frac{\partial y}{\partial b}, \frac{\partial y}{\partial \beta}$

$$g(v) = Wv + b$$

W is a row vector

$$\nabla_v g(v) = W^T \quad \text{now a column vector}$$

Gradient

$$y = Wf(\beta) + b$$

Want: $\frac{\partial y}{\partial W}$, $\frac{\partial y}{\partial b}$, $\frac{\partial y}{\partial \beta}$

$$g(v) = Wv + b$$

$$\nabla_v g(v) = W^T \quad \text{now a column vector}$$

meaning of $\frac{\partial y}{\partial \beta_{(i)}}$: if chang ith coordinate of β, how does that change jth coordinate of $f(\beta)$ i.e. how does it chang jth coordinate of v, then how does change jth coordinate of v change g of v ~ just chang by $W_{(j)}$

$f(\beta) = v$

$$\frac{\partial y}{\partial \beta_{(i)}} = \sum_{j=1}^{p} \left[\nabla g(v) \Big|_{v=f(\beta)} \right]_{(j)} \frac{\partial f(\beta)_{(j)}}{\partial \beta_{(i)}}$$

$$= \sum_{j=1}^{p} W_{(j)} \frac{\partial f(\beta)_{(j)}}{\partial \beta_{(i)}}$$

Gradient

$$y = Wf(\beta) + b$$

Want: $\frac{\partial y}{\partial W}, \frac{\partial y}{\partial b}, \frac{\partial y}{\partial \beta}$

$$g(v) = Wv + b$$

$$\nabla_v g(v) = W^T \quad \text{now a column vector}$$

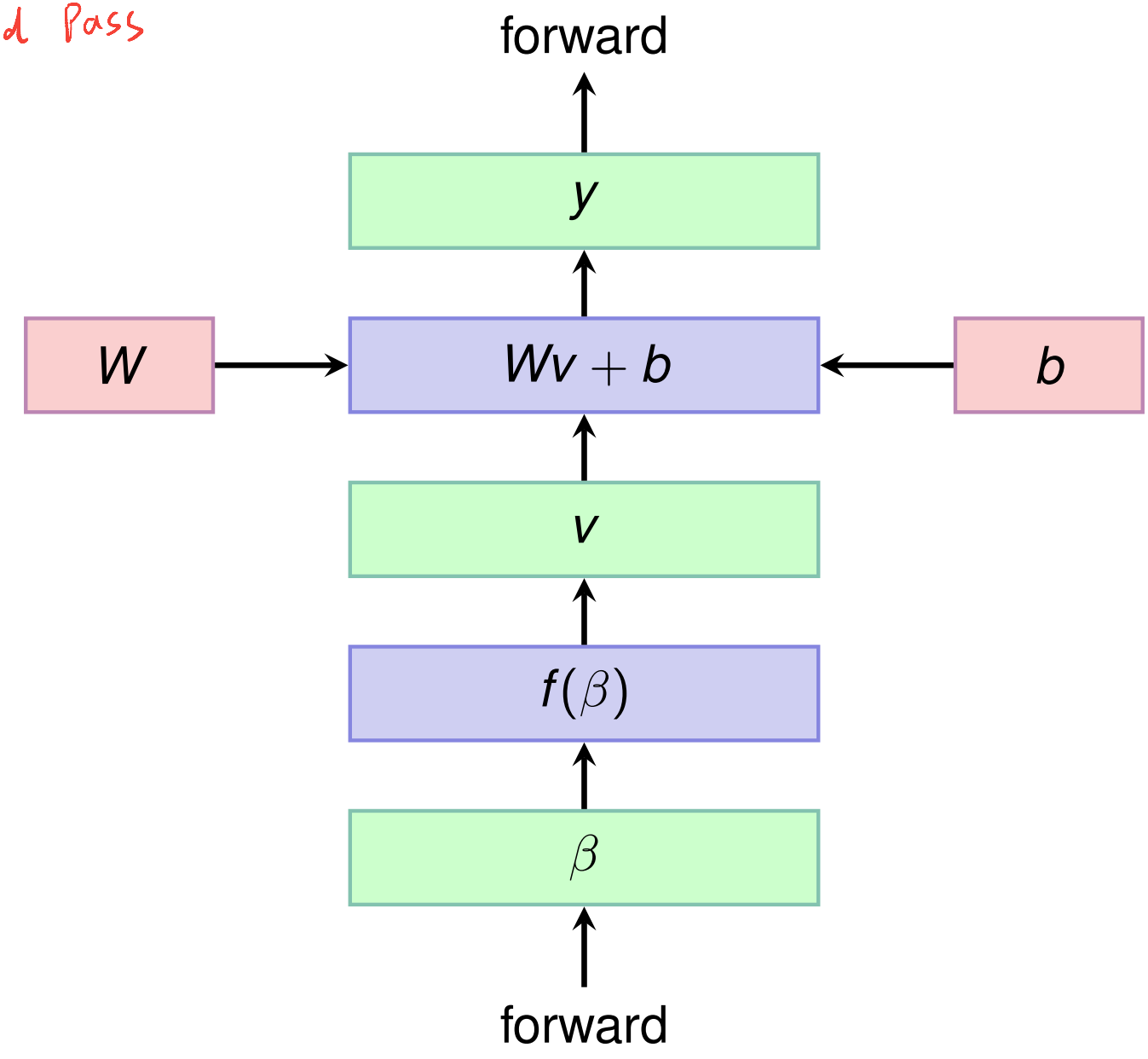
easy to find

$$\frac{\partial y}{\partial W} = f(\beta)^T \quad \text{gradient should be the same shape as } W$$

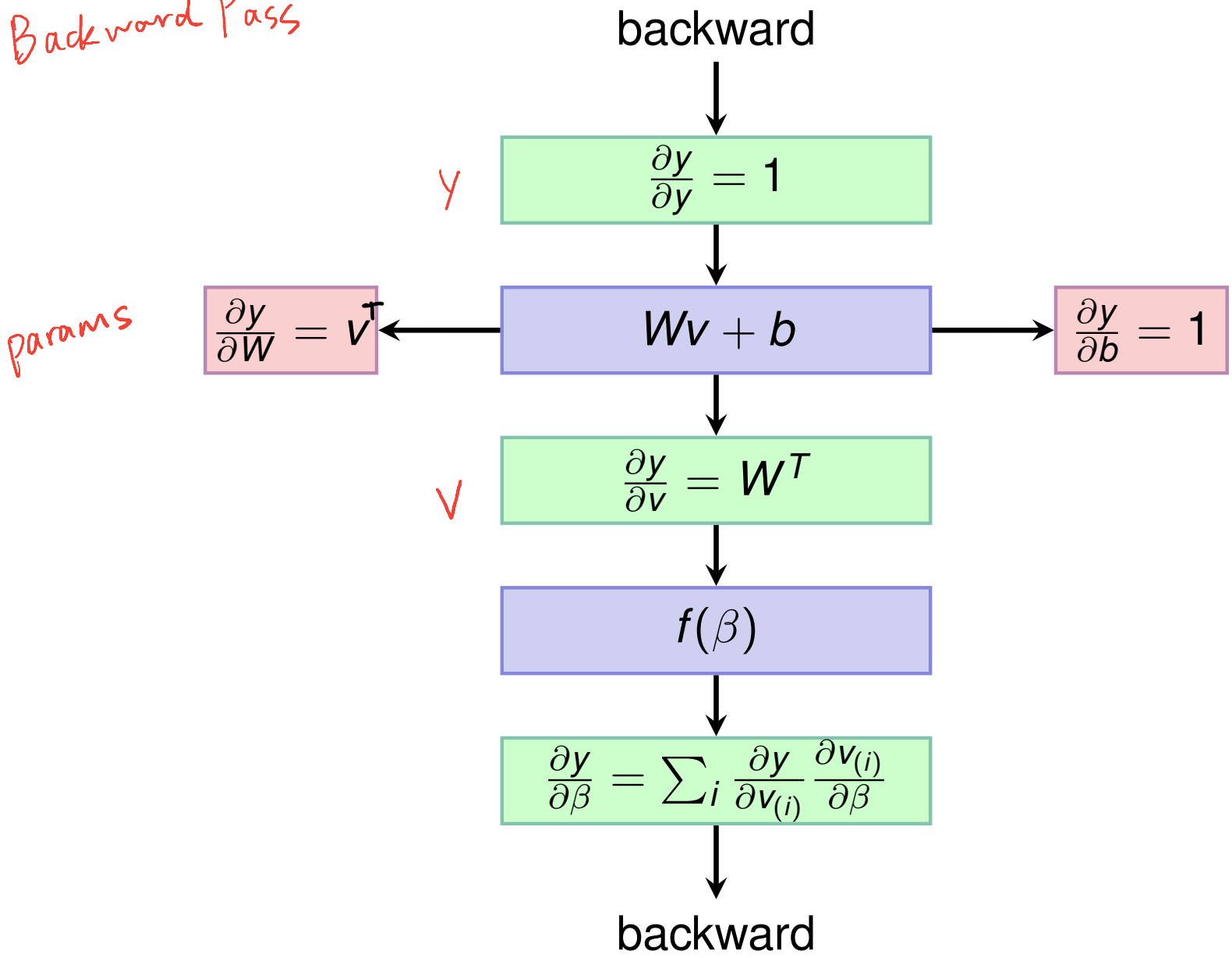
a row vector

$$\frac{\partial y}{\partial b} = 1$$

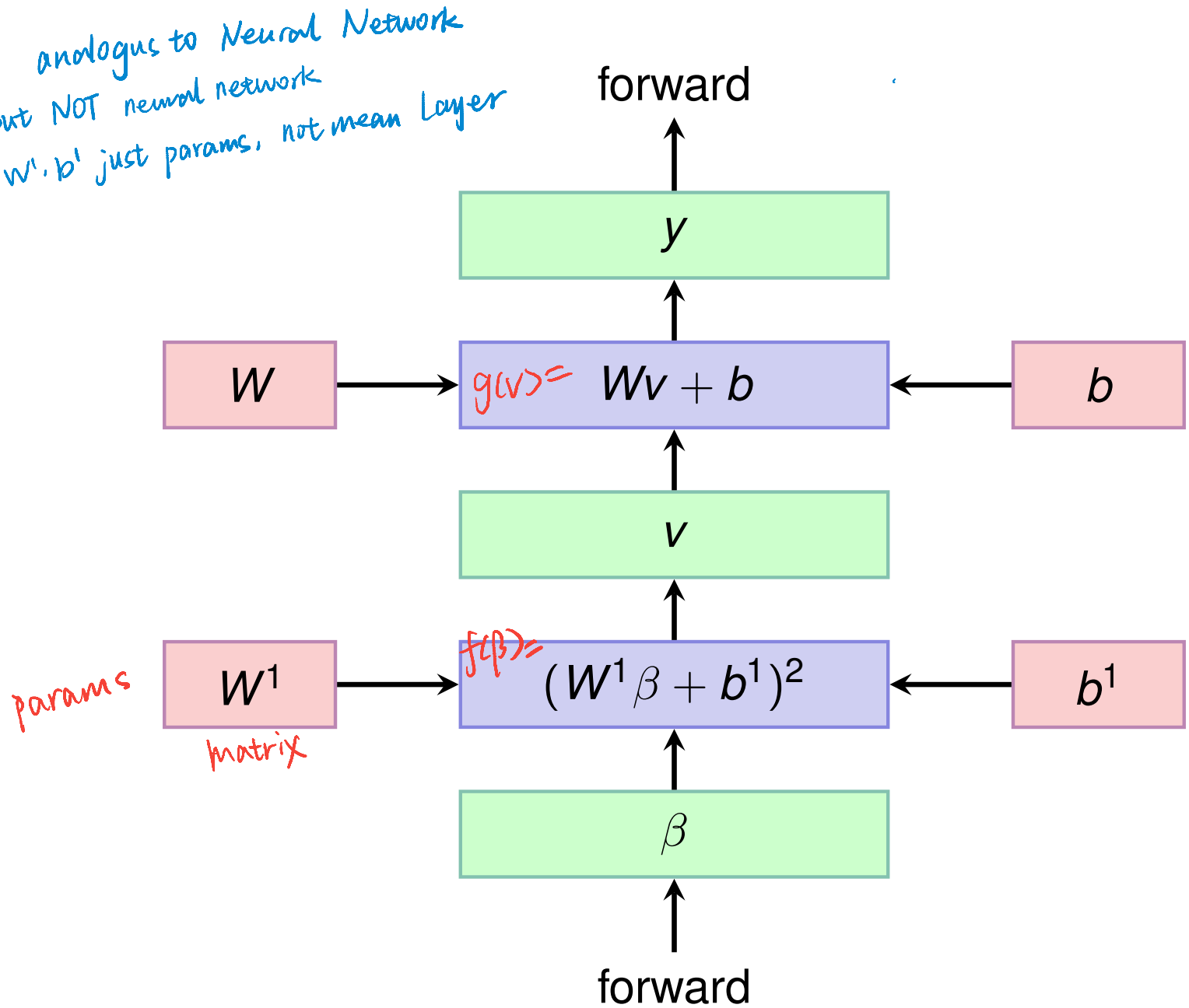
Forward Pass



Backward Pass



More concrete example



Mathematically

$f: R^d \to R^p$ $\quad g: R^p \to R$

$$W \in \mathbb{R}^{1 \times p}$$ a row vector

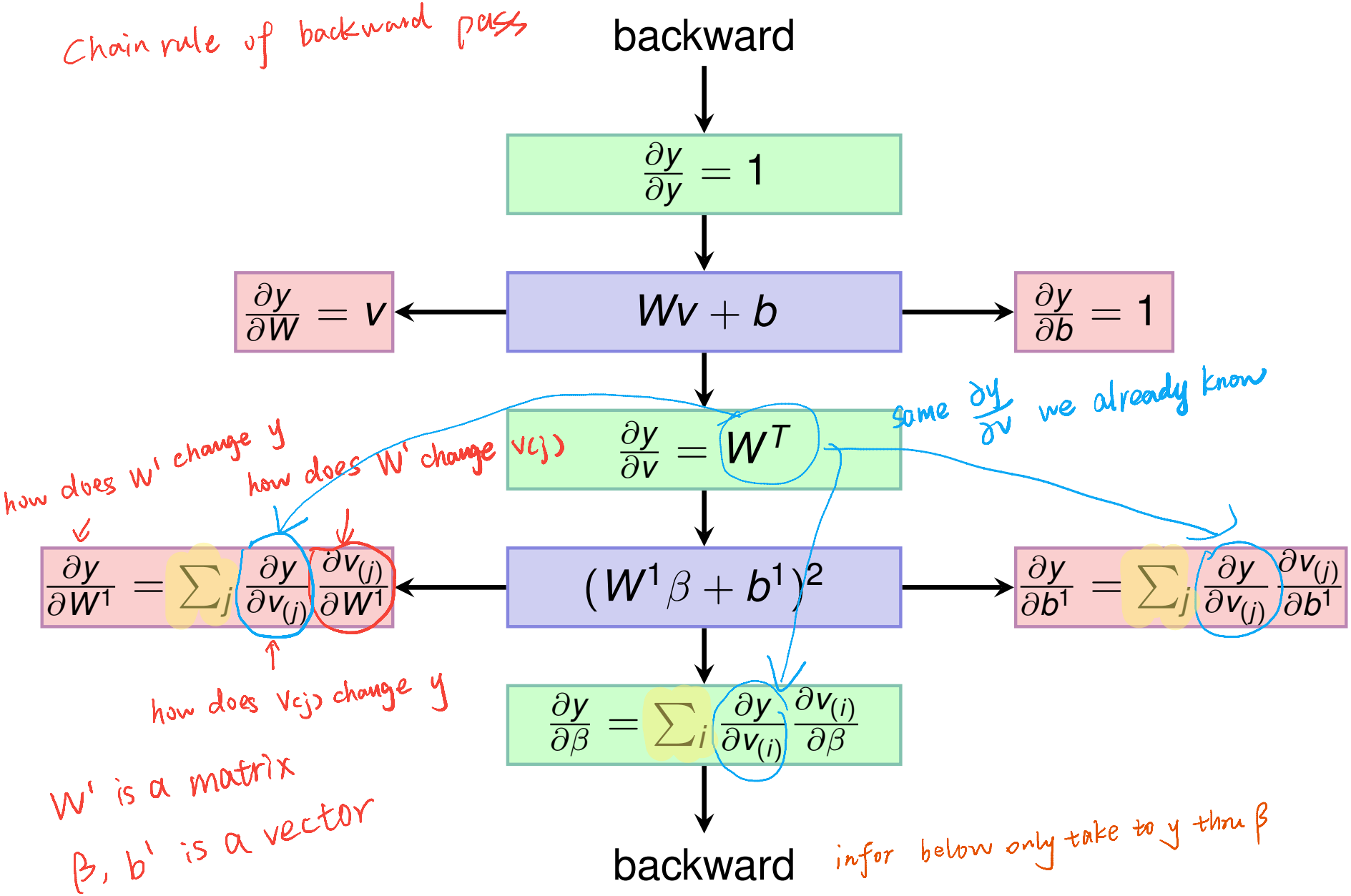
$$b \in \mathbb{R}$$

$$W^1 \in \mathbb{R}^{p \times d}$$ a matrix

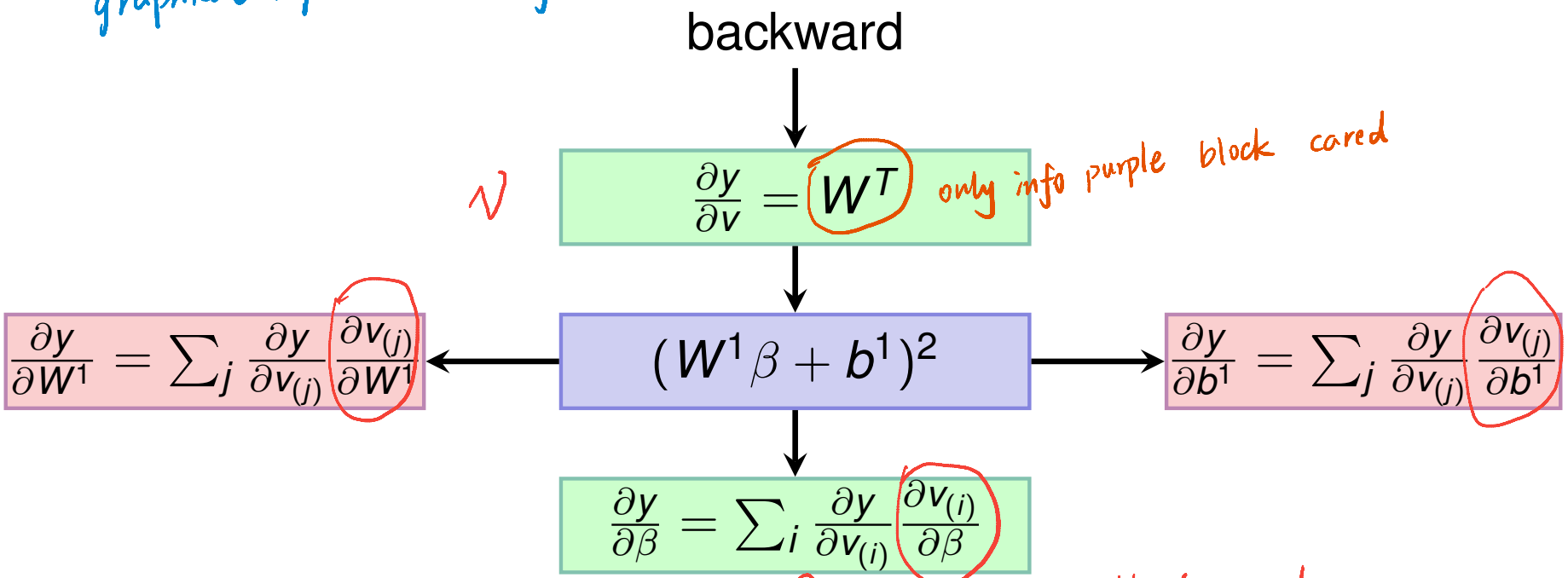
$\beta \in R^d$ $\quad b^1 \in \mathbb{R}^p$

$$y = W((W^1\beta + b^1)^2) + b$$

$$\left[(W^1\beta + b^1)^2\right]_{(i)} = [\ W^1\beta + b^1]^2_{(i)}$$


Chain rule of backward pass
backward
$\frac{\partial y}{\partial y} = 1$
$\frac{\partial y}{\partial W} = v$
$Wv + b$
$\frac{\partial y}{\partial b} = 1$
$\frac{\partial y}{\partial v} = W^T$
same $\frac{\partial y}{\partial v}$ we already know
how does W' change y
how does W' change v(j)
$\frac{\partial y}{\partial W^1} = \sum_j \frac{\partial y}{\partial v_{(j)}} \frac{\partial v_{(j)}}{\partial W^1}$
$(W^1\beta + b^1)^2$
$\frac{\partial y}{\partial b^1} = \sum_j \frac{\partial y}{\partial v_{(j)}} \frac{\partial v_{(j)}}{\partial b^1}$
how does v(j) change y
$\frac{\partial y}{\partial \beta} = \sum_i \frac{\partial y}{\partial v_{(i)}} \frac{\partial v_{(i)}}{\partial \beta}$
W' is a matrix
β, b' is a vector
backward
infor below only take to y thru β

graphical representation of chain rule



we only need to calculate 3 0

a number

$$v_{(j)} = (\overbrace{e_j^T W^1 \beta + b^1_{(j)}}^{\text{a number}})^2$$

$$= (\text{trace}(W^1 \beta e_j^T) + b^1_{(j)})^2$$

since $e_j^T W^1 \beta$ is a real number

trace(a) = a $a \in R$

$e_j^T W^1 \beta = \text{trace}(e_j^T W^1 \beta)$

$e_j \in R^d$ $\beta \in R^d$

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_j & \cdots & \beta_d \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & \\ \beta_1 & \beta_2 & \cdots & \beta_d & \\ 0 & 0 & 0 & 0 & \end{bmatrix}$$

j^{th} row, j^{th} column

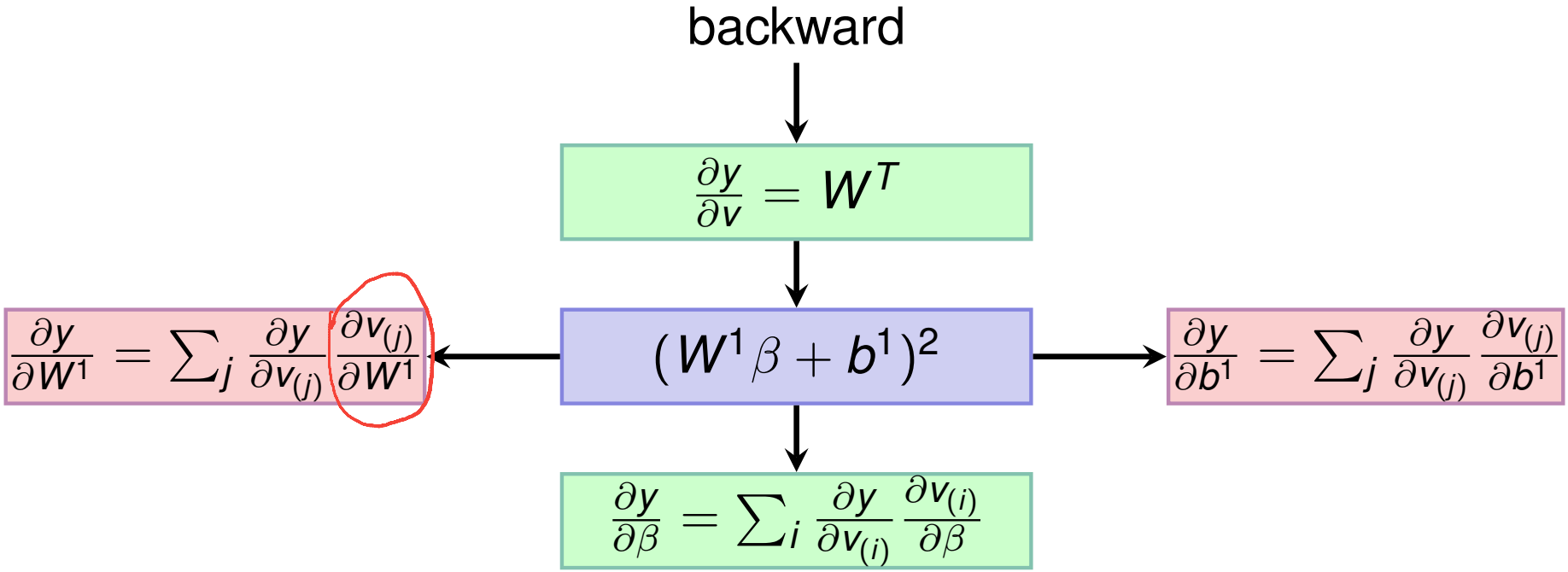
Hw4 we show

$$\frac{\partial \text{tr}(AC^T)}{\partial A} = C$$

$$\Rightarrow \frac{\partial \text{tr}(W^1 \beta e_j^T)}{\partial W^1} = e_j \beta^T$$

a matrix $\in R^{d \times d}$

only j^{th} row is no zero $= \beta^T$



Now compute $\frac{\partial v_{(j)}}{\partial W^1}$

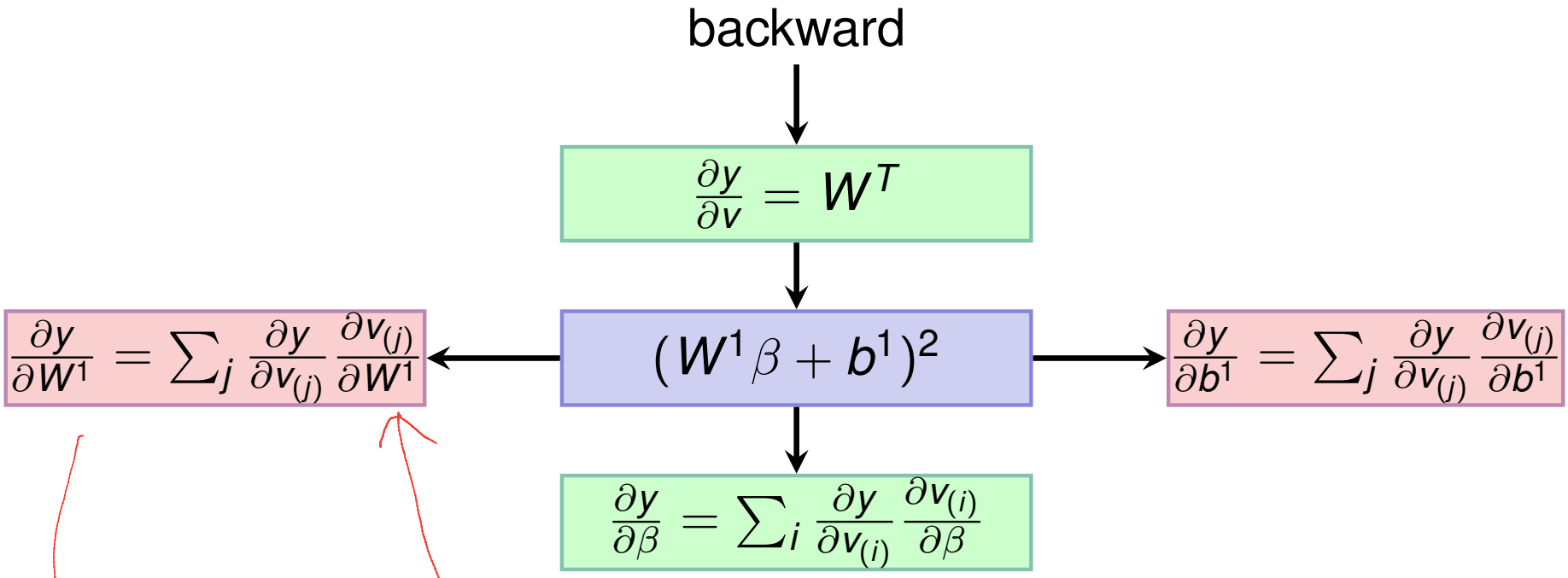
$$v_{(j)} = (e_j^T W^1 \beta + b^1_{(j)})^2$$

$$= (\text{trace}(W^1 \beta e_j^T) + b^1_{(j)})^2 \quad = z^2_{(j)}$$

define $z = W^1\beta + b^1 \Rightarrow z_{(j)} = \text{trace}(W^1 \beta e_j^T) + b^1_{(j)}$

$$\frac{\partial v_{(j)}}{\partial W^1} = 2 z_{(j)} e_j \beta^T$$

chain rule: $= \frac{\partial (z_{(j)})^2}{\partial z_{(j)}} \frac{\partial z_{(j)}}{\partial W^1}$



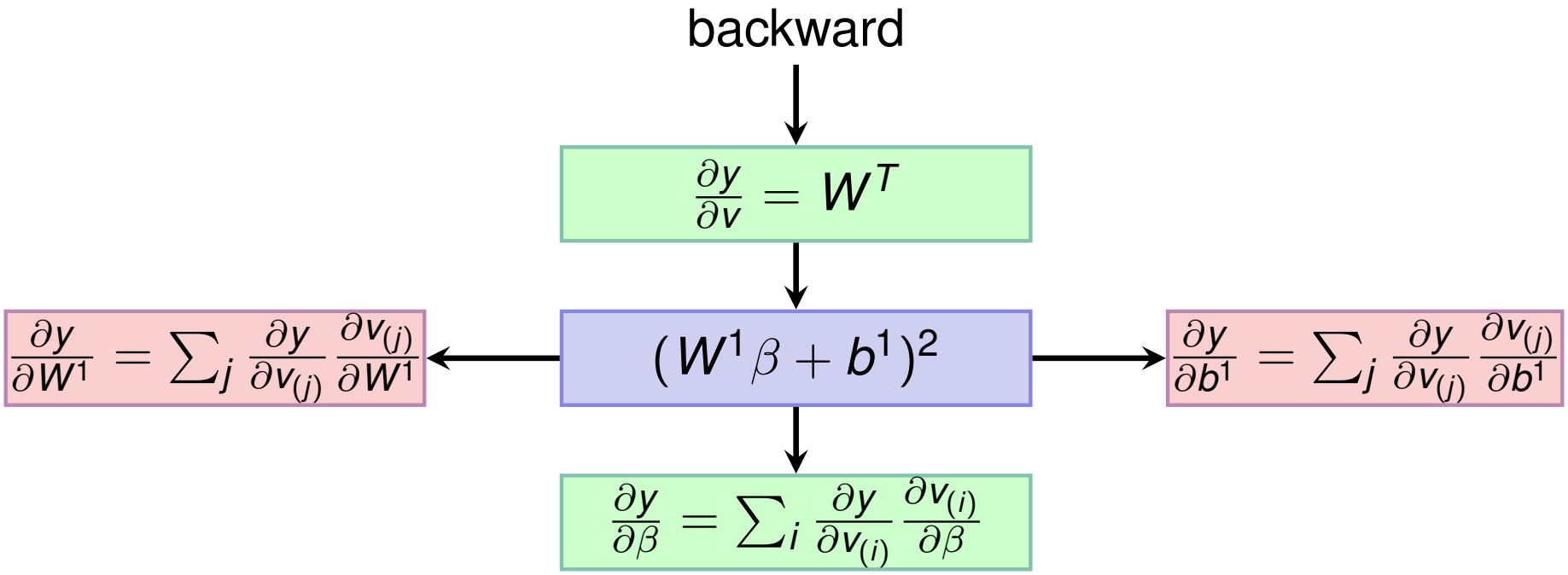
$$z = W^1\beta + b^1 \Rightarrow z_{(j)} = \text{trace}(W^1\beta e_j^T) + b^1_{(j)}$$

plug in

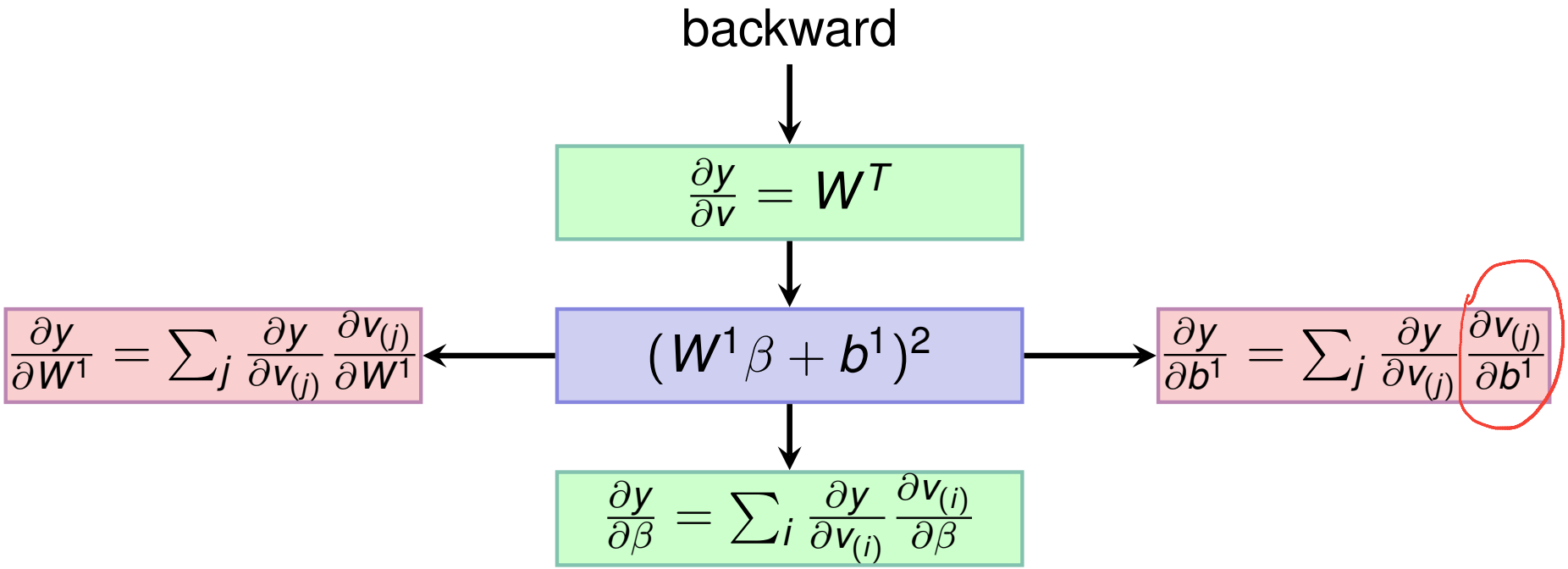
$$\frac{\partial v_{(j)}}{\partial W^1} = 2z_{(j)} e_j \beta^T$$

$$\frac{\partial y}{\partial W^1} = \sum_j 2W_{(j)} z_{(j)} e_j \beta^T = 2(W^T \circ z)\beta^T$$

$\frac{\partial y}{\partial v_{(j)}}$; independent of sum ; recall slide 20



$$v_{(j)} = (e_j^T W^1 \beta + b^1_{(j)})^2$$



Now compute $\frac{\partial v_{(j)}}{\partial b^1}$

define

$$z = W^1\beta + b^1$$

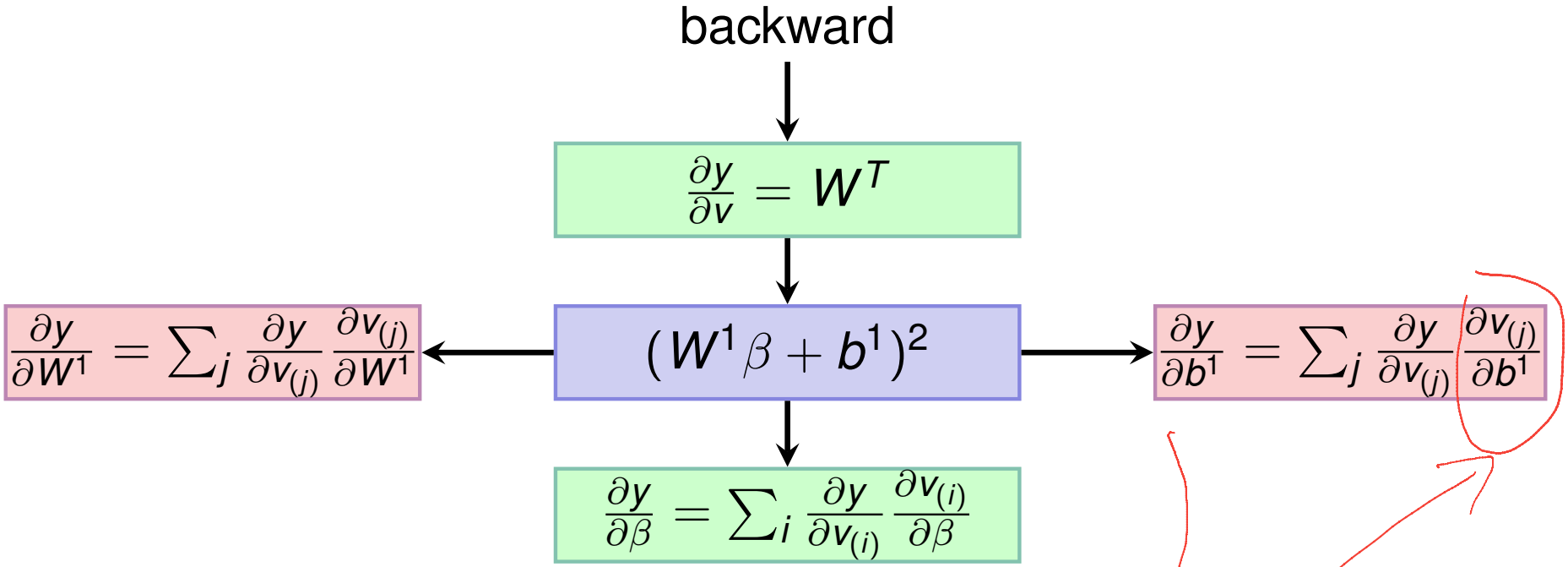
$$v_{(j)} = (e_j^T W^1 \beta + b^1_{(j)})^2 = (e_j^T W^1 \beta + e_j^T b^1)^2$$

chain rule

$$= \frac{\partial (z_{(j)})^2}{\partial z_j} \cdot \frac{\partial z_j}{\partial b^1}$$

$$\Rightarrow \frac{\partial v_{(j)}}{\partial b^1} = 2\left(e_j^T W^1 \beta + e_j^T b^1\right) \cdot e_j$$

$$\frac{\partial v_{(j)}}{\partial b^1} = 2 z_{(j)} e_j$$

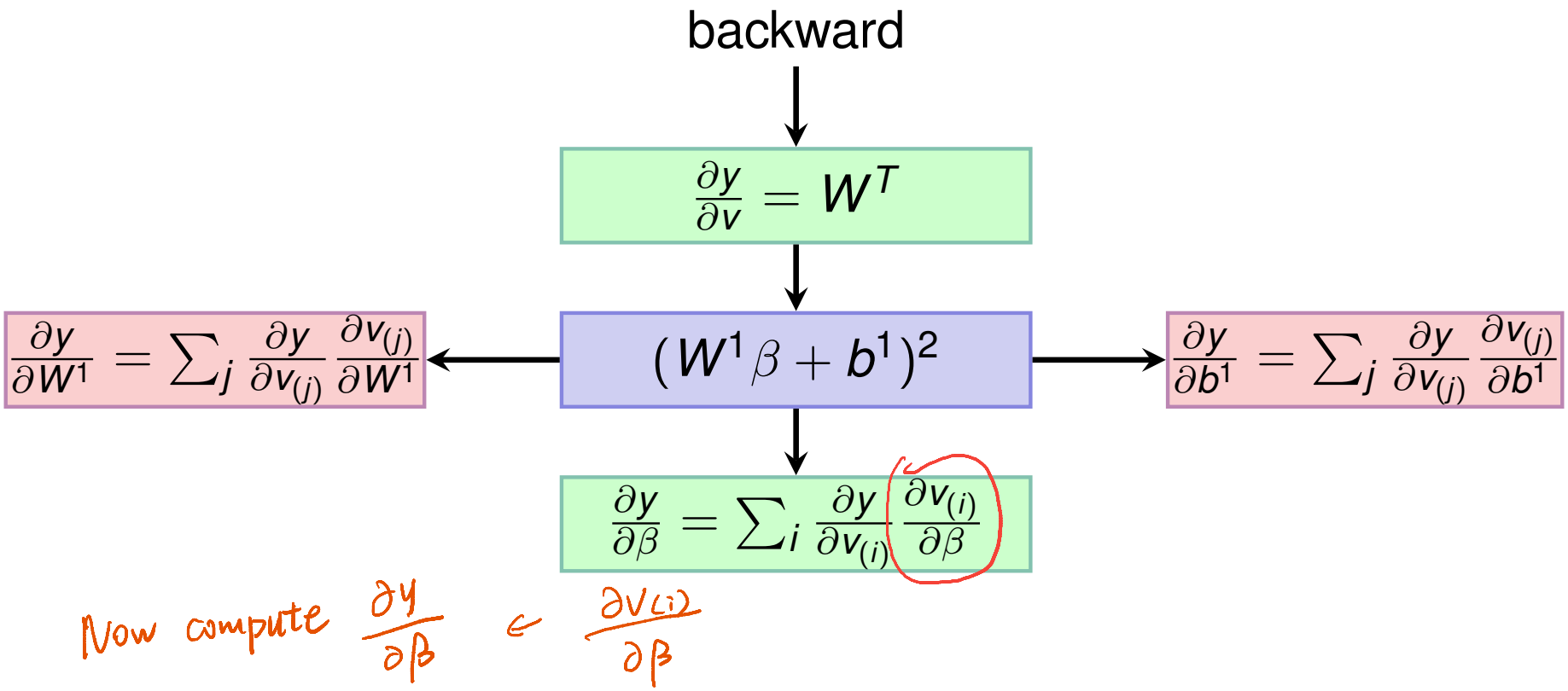


$z = W^1\beta + b^1$

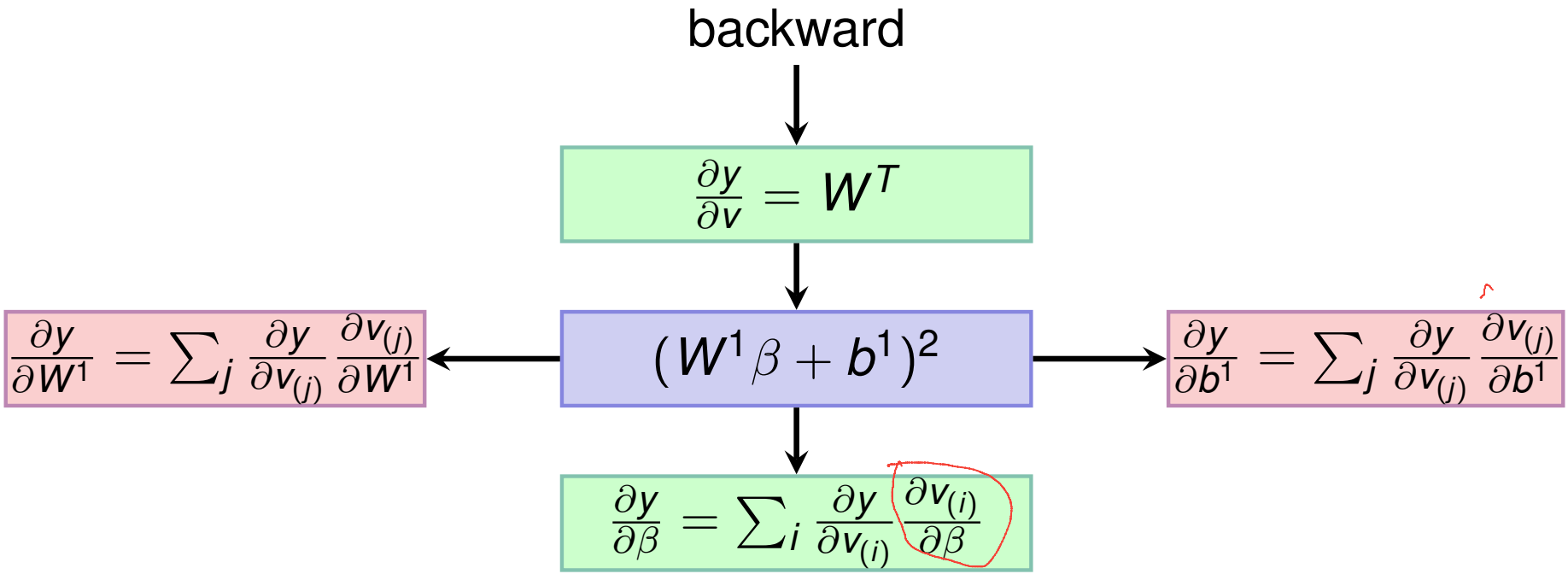
plug in $\frac{\partial v_{(j)}}{\partial b^1} = 2z_{(j)} e_j$

$$\frac{\partial y}{\partial b^1} = \sum_j 2W_{(j)} z_{(j)} e_{(j)}$$

$$= 2WZ \cdot J^T \circ Z$$



$$v_{(j)} = (e_j^T W^1 \beta + b^1_{(j)})^2$$



$$z = W^1\beta + b^1$$

$$v_{(j)} = (e_j^T W^1 \beta + b^1_{(j)})^2$$

$$= \frac{\partial (z_{(j)})^2}{\partial z_{(j)}} \frac{\partial z_{(j)}}{\partial \beta}$$

$$= 2 z_{(j)} (e_j^T W^1)^T$$

$$\frac{\partial v_{(j)}}{\partial \beta} = 2 z_{(j)} (W^1)^T e_j$$

$e_j \in R^p \quad W^1 \in R^{p \times d}$

$e_j^T \in R^{1 \times p}$

$(e_j^T W^1) \in R^d$

$\frac{\partial z_{(j)}}{\partial \beta} \in R^d$

导数和维度要和 β 一样

backward

$y = Wv + b$

$\frac{\partial y}{\partial v} = W^T$

$W \in R^{1 \times p}$ $W^T \in R^p$

$W' \in R^{p \times d}$ $(W')^T \in R^{d \times p}$

$\frac{\partial y}{\partial W^1} = \sum_j \frac{\partial y}{\partial v_{(j)}} \frac{\partial v_{(j)}}{\partial W^1}$ ← $(W^1\beta + b^1)^2$ → $\frac{\partial y}{\partial b^1} = \sum_j \frac{\partial y}{\partial v_{(j)}} \frac{\partial v_{(j)}}{\partial b^1}$

$\frac{\partial y}{\partial \beta} = \sum_i \frac{\partial y}{\partial v_{(i)}} \frac{\partial v_{(i)}}{\partial \beta}$

$z = W^1\beta + b^1 \in R^p$

plug in $\frac{\partial v_{(j)}}{\partial \beta} = 2z_{(j)}(W^1)^T e_j$

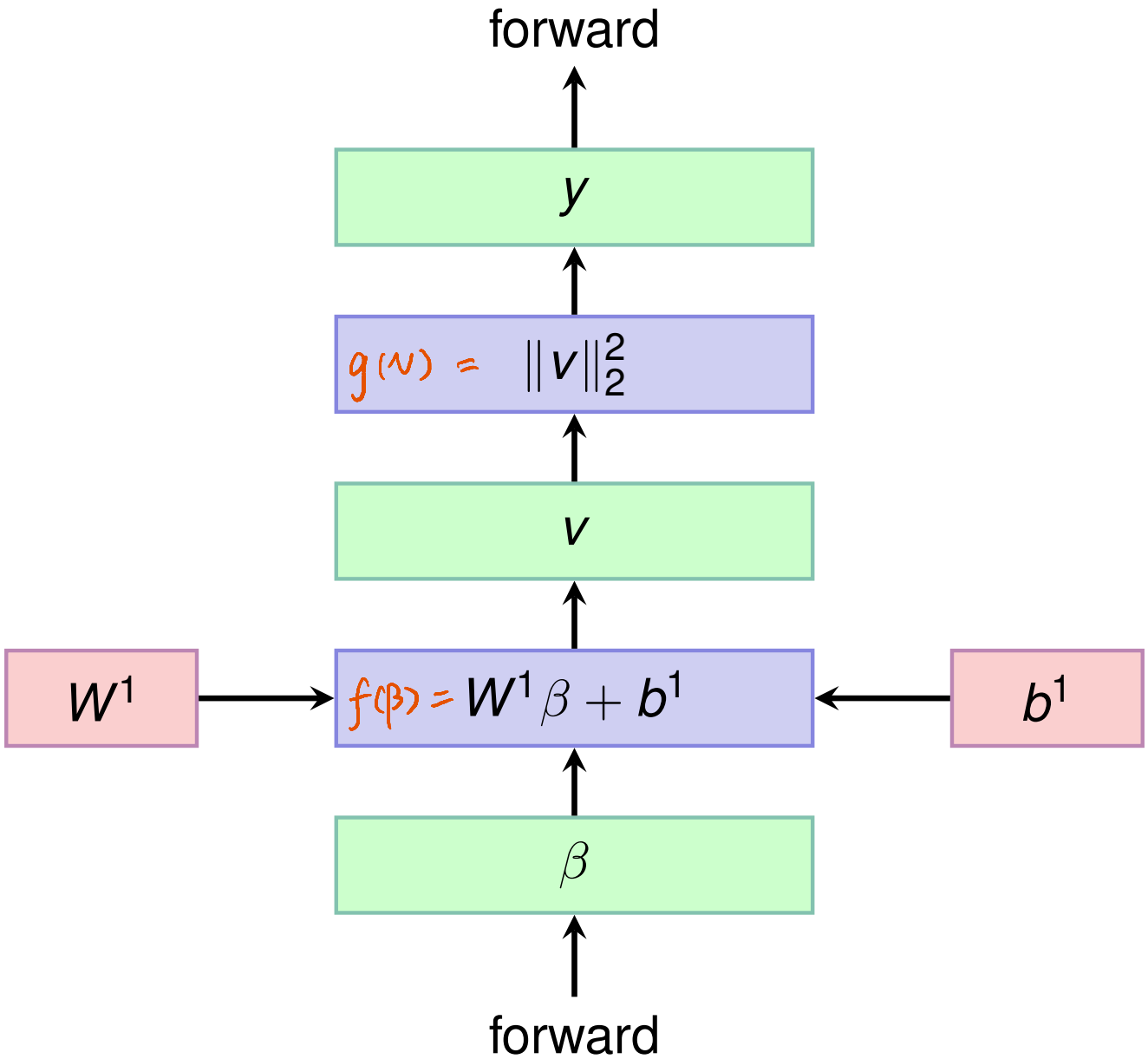
a number, a number, $R^{d \times p}$, R^p

$$\frac{\partial y}{\partial \beta} = \sum_j 2W_{(j)} z_{(j)} (W^1)^T e_j = 2(w')^T \sum_j W_{(j)} z_{(j)} e_j$$

$(W^1)^T e_j \in R^d$

$$= 2(W_{R^{d \times p}}^1)^T (W_{R^p}^T \circ z) \in R^d$$

Another example



Mathematically

$$W^1 \in \mathbb{R}^{p \times d}$$

$$b^1 \in \mathbb{R}^p$$

identical to linear regression

$$y = \| W^1 \beta + b^1 \|_2^2$$

Mathematically

$\beta \in \mathbb{R}^d$

$$W^1 \in \mathbb{R}^{p \times d}$$

$$b^1 \in \mathbb{R}^p \qquad v \in \mathbb{R}^p$$

$$y = \| W^1 \beta + b^1 \|_2^2 \qquad \in \mathbb{R}$$

Already know — proof next slide

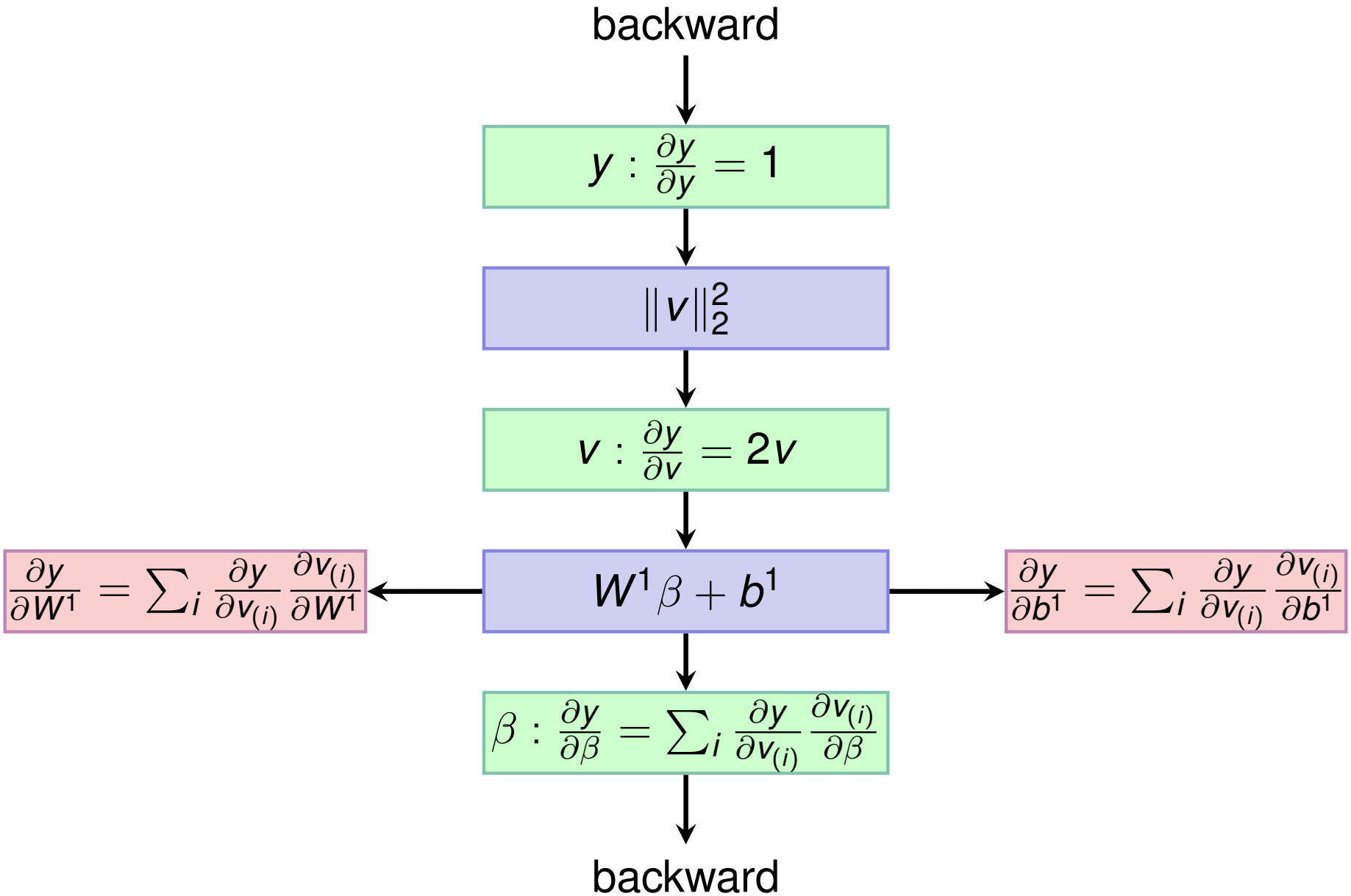
$$\frac{\partial y}{\partial \beta} = 2(W^1)^T(W^1\beta + b^1)$$

$= 2I^T(W^1\beta + b^1)$ I: identity matrix $y = \|W^1\beta + Ib^1\|_2^2$

$$\frac{\partial y}{\partial b^1} = 2(W^1\beta + b^1)$$

$$\frac{\partial y}{\partial W^1} = 2(W^1\beta + b^1)\beta^T$$ maybe not this one

Graphically



Computing gradients

加 e_i，工是为了让梯度的维度和自变量相同

$$v = W^1\beta + b^1$$

$\frac{\partial y}{\partial v} = 2v$

ith row of W^1

$$\frac{\partial v_{(i)}}{\partial \beta} = W^1_{(i,:)} = (W^1)^T e_i$$

$$\frac{\partial v_{(i)}}{\partial b^1} = 1 e_i$$

$$\frac{\partial v_{(i)}}{\partial W^1} = e_i \beta^T$$

if $AB = BA$ 可交换

then A commute with B

plug in

$$\frac{\partial y}{\partial \beta} = \sum_i 2v_{(i)} W^1_{(i,:)} = 2(W^1)^T v$$

$= \sum_i 2v_{(i)} W^{1T} e_i = 2W^{1T} \sum_i v_{(i)} e_i$

W^1 independent of i

can commute

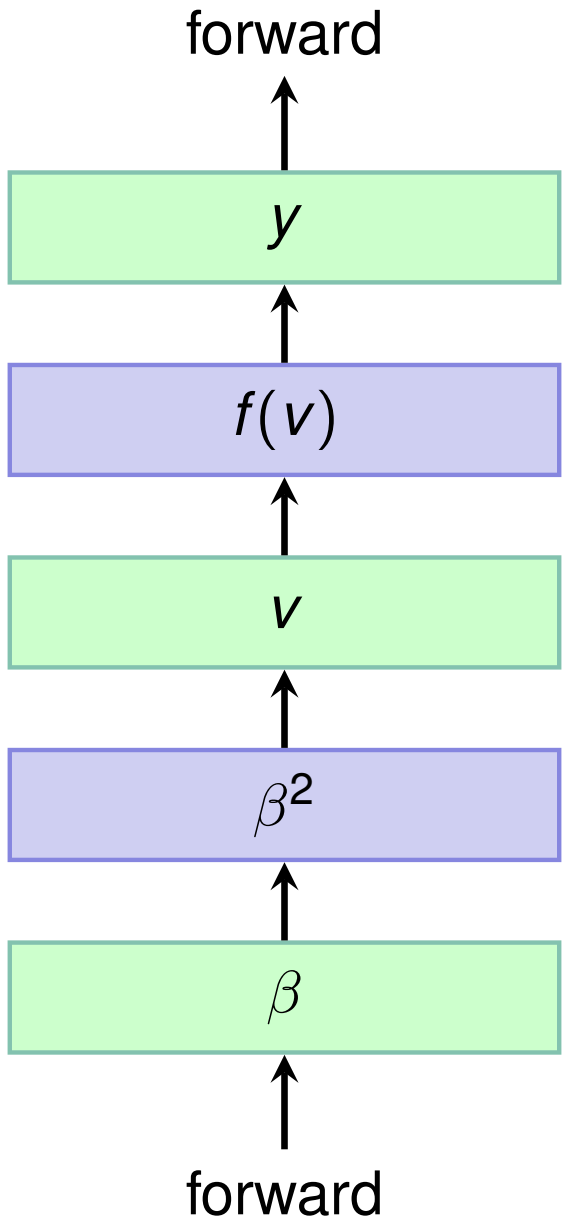
$$\frac{\partial y}{\partial b^1} = \sum_i 2v_{(i)} e_i = 2v$$

$$\frac{\partial y}{\partial W^1} = \sum_i 2v_{(i)} e_i \beta^T = 2v\beta^T$$

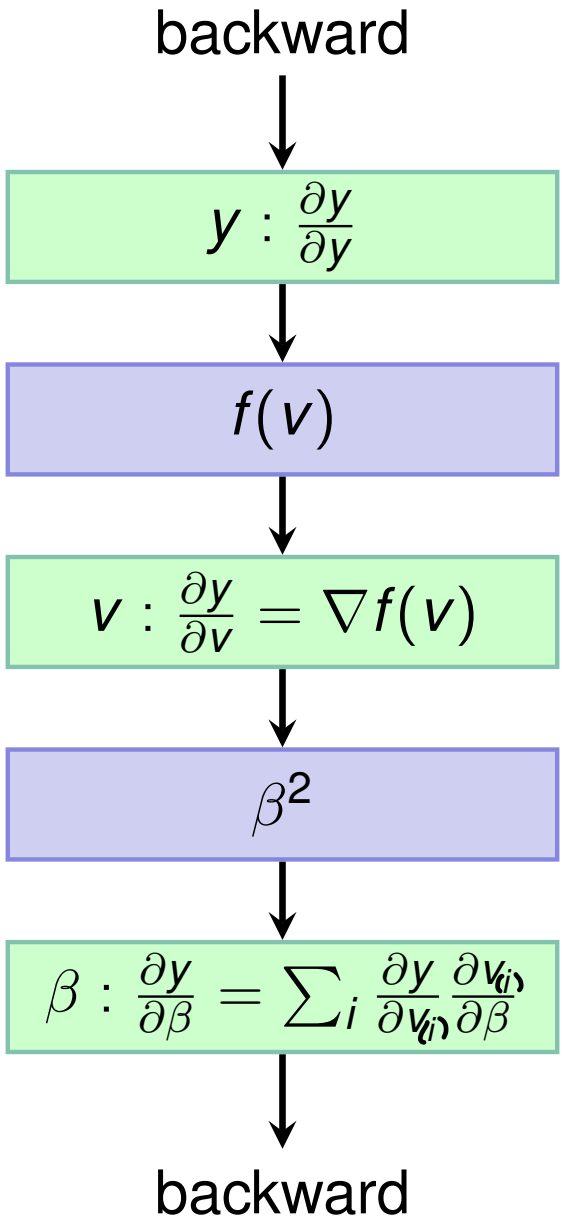
Final example more easy

$y = f(\beta^2)$ where $f : \mathbb{R}^d \mapsto \mathbb{R}$

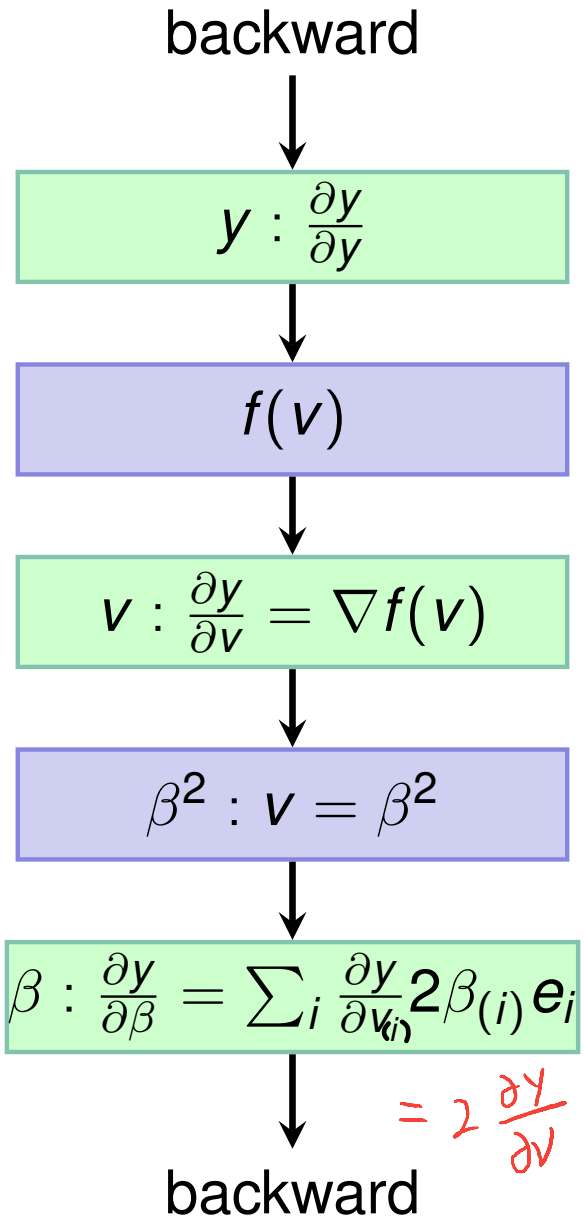
Final example



Final example



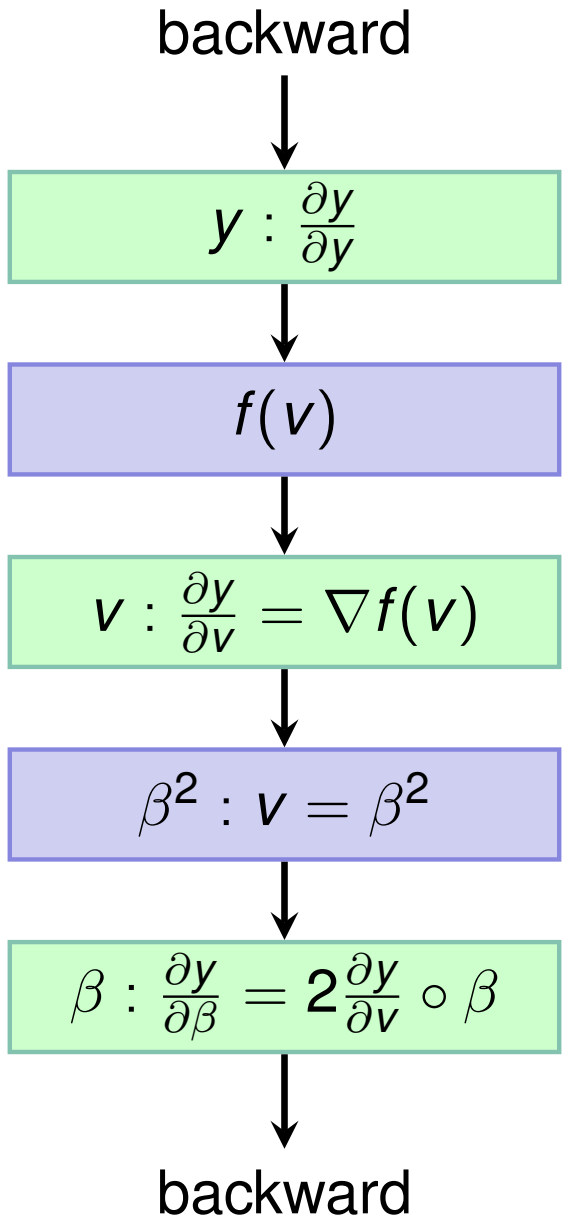
Final example



$\frac{\partial v_i}{\partial \beta} = 2\beta_{(i)} e_i$

$= 2 \frac{\partial y}{\partial v} \circ \beta$

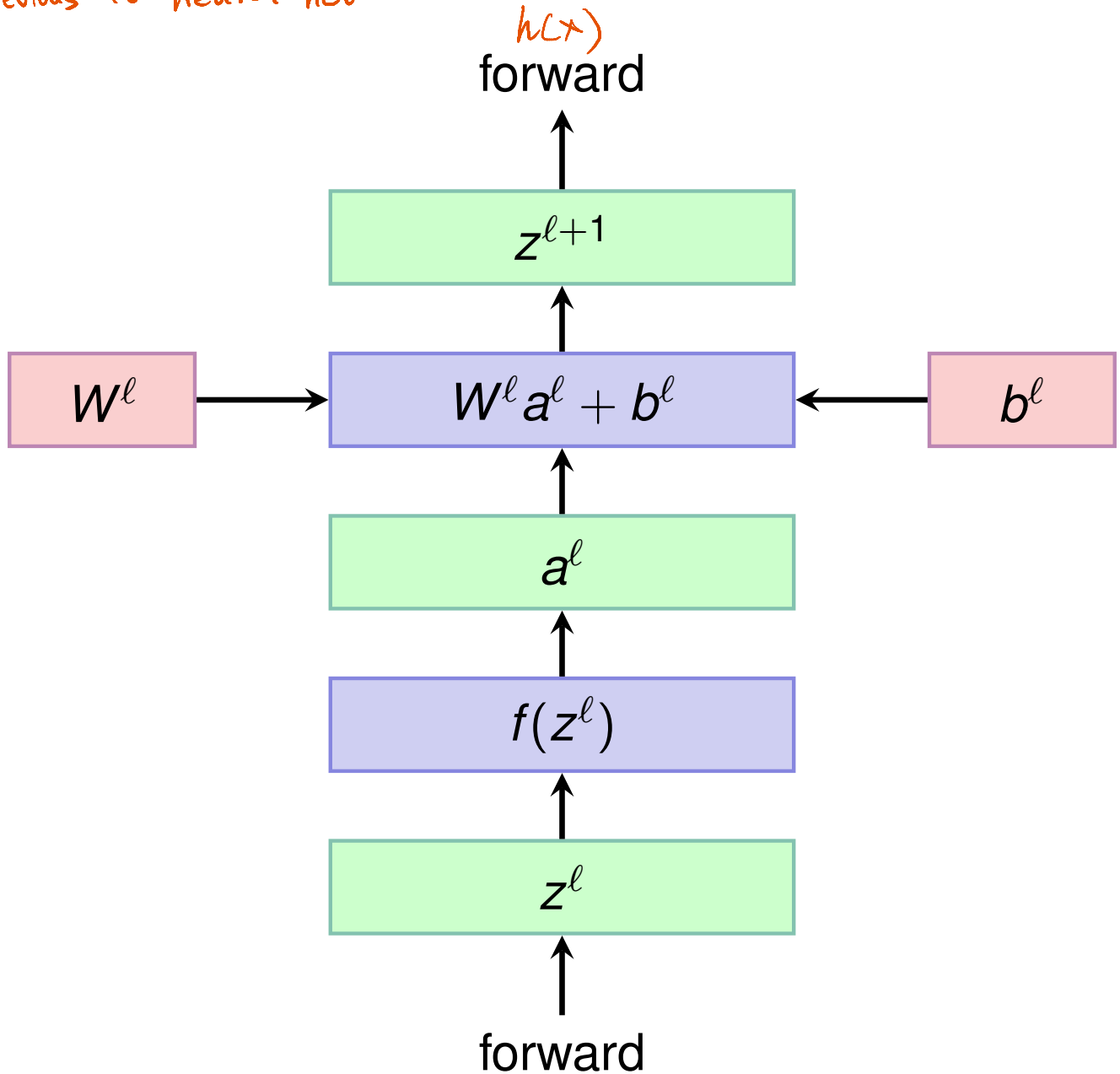
Final example



Neural network: Forward pass

Now apply previous to neural net

h(x)



Backpropagation

求3个梯度
of computing gradient with respect to different params

h(x)

backward

$$\delta^{\ell+1} = \frac{\partial h(x)}{\partial z^{\ell+1}}$$

a number

$$W^\ell a^\ell + b^\ell$$

$$\frac{\partial h(x)}{\partial W^\ell} = \delta^{\ell+1}(a^\ell)^T$$

$$\frac{\partial h(x)}{\partial b^\ell} = \delta^{\ell+1}$$

$$\frac{\partial h(x)}{\partial a^\ell} = (W^\ell)^T \delta^{\ell+1}$$

$$f(z^\ell)$$

$$\delta^\ell = \frac{\partial h(x)}{\partial z^\ell} = f'(z^\ell) \circ \left[(W^\ell)^T \delta^{\ell+1} \right]$$

backward

$$z_{(i)}^{(\ell+1)} = e_i^T (W^\ell a^\ell + b^\ell) \in \mathbb{R}^P$$

$$\frac{\partial h}{\partial W^\ell} = \sum_{(i)} \frac{\partial h}{\partial z_{(i)}^{(\ell+1)}} \frac{\partial z_{(i)}^{(\ell+1)}}{\partial W^{(\ell)}} = \sum_{(i)} \delta_{(i)}^{(\ell+1)} e_i a^{\ell T} = \left(\sum_{(i)} \delta_{(i)}^{(\ell+1)} e_i \right) a^{\ell T} = \delta^{(\ell+1)} [a^{(\ell)}]^T$$

$$\frac{\partial h}{\partial a^\ell} = \sum_{(i)} \frac{\partial h}{\partial z_{(i)}^{\ell+1}} \frac{\partial z_{(i)}^{(\ell+1)}}{\partial a^\ell} = \sum_{(i)} \delta_{(i)}^{\ell+1} (W^\ell)^T e_i = (W^\ell)^T \sum_{(i)} \delta_{(i)}^{\ell+1} e_i$$

$$\frac{\partial h}{\partial z^\ell} = \sum_j \frac{\partial h}{\partial a_{(j)}^\ell} \frac{\partial a_{(j)}^\ell}{\partial z^\ell} = \sum_j \frac{\partial h}{\partial a_{(j)}^\ell} f'(z_{(j)}^\ell) e_j$$

Conclusion

Gradient methods will behave differently based on geometry of problem

Conclusion

Gradient methods will behave differently based on geometry of problem

Neural networks: powerful methods to approximate non-linear functions

Backpropagation: computationally efficient method to compute gradients of NN

↓ gradient descent