Neural Networks (part 2)

E190BO

Sept 9, 2026

Recap

How to fit a model to data? Gradient Descent

How to compute gradient of complex functions? Backpropagation

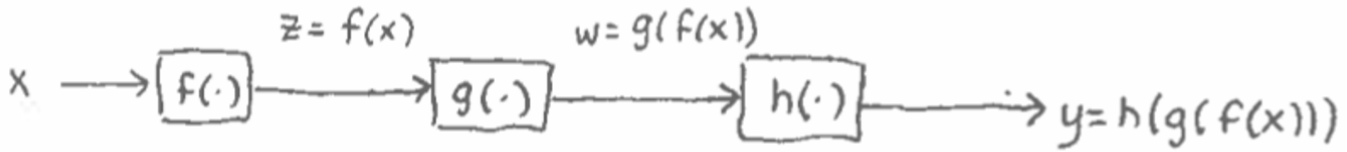
initialize weights $\triangleq [w_1, w_2 \dots w_m]$

for i in range(numIters):

grad = calcGradient(...)

weights = weights - $\gamma \cdot$ grad

Backpropagation = computation graph + chain rule



$$f'(x) \cdot g'(z) \cdot h'(w) = \frac{dy}{dx}$$

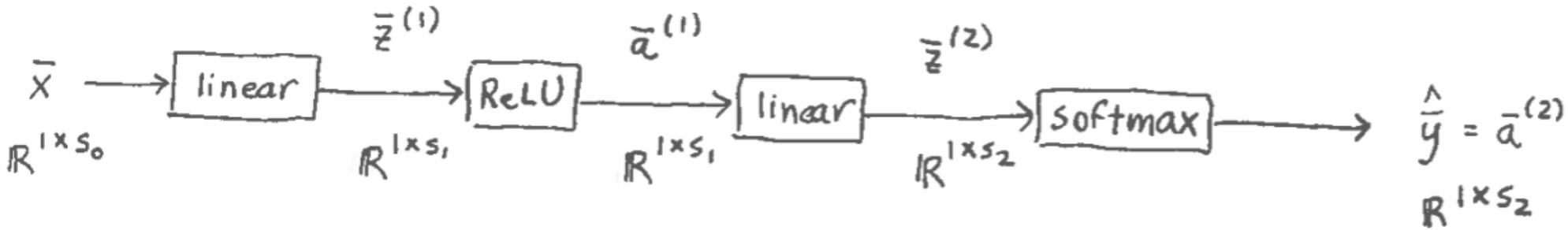
$$\frac{dz}{dx} \quad \frac{dw}{dz} \quad \frac{dy}{dw}$$

Goal: train a neural network classifier

Recap

Model	NN classifier
Loss	CE
Optimizer	mini-batch GD

Model



Loss

Cross Entropy Loss: $J_{CE} = \frac{1}{N} \sum_{i=0}^{N-1} \left(- \sum_{c=1}^{C} y_{ic} \cdot \log (\hat{y}_{ic}) \right)$

Recap

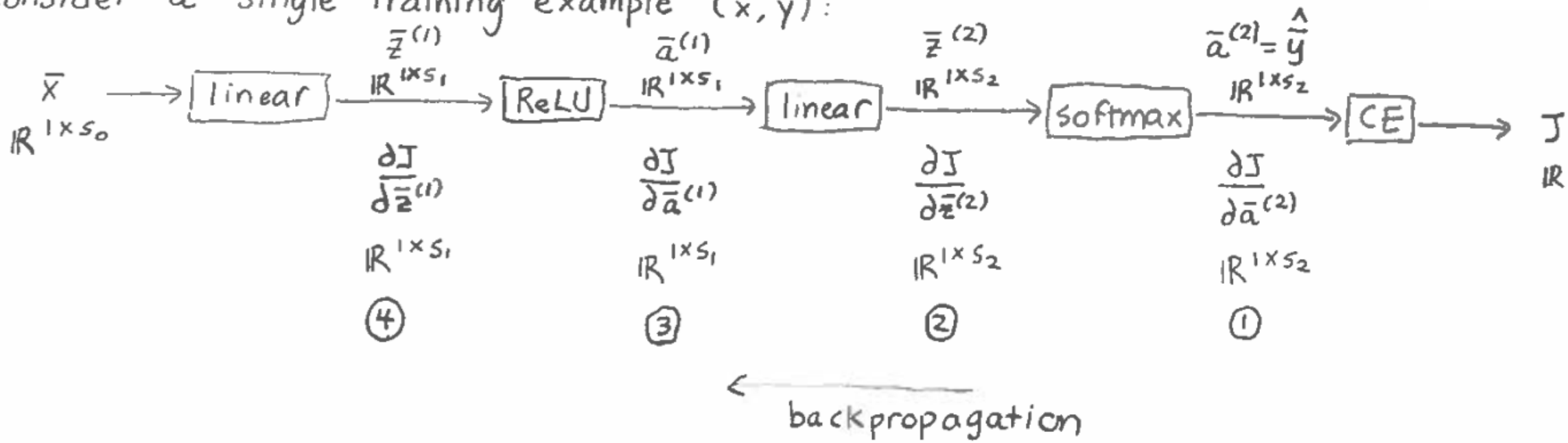
Optimizer

initialize weights = $[\, W^{(1)}, \vec{b}^{(1)}, W^{(2)}, \vec{b}^{(2)} \,]$

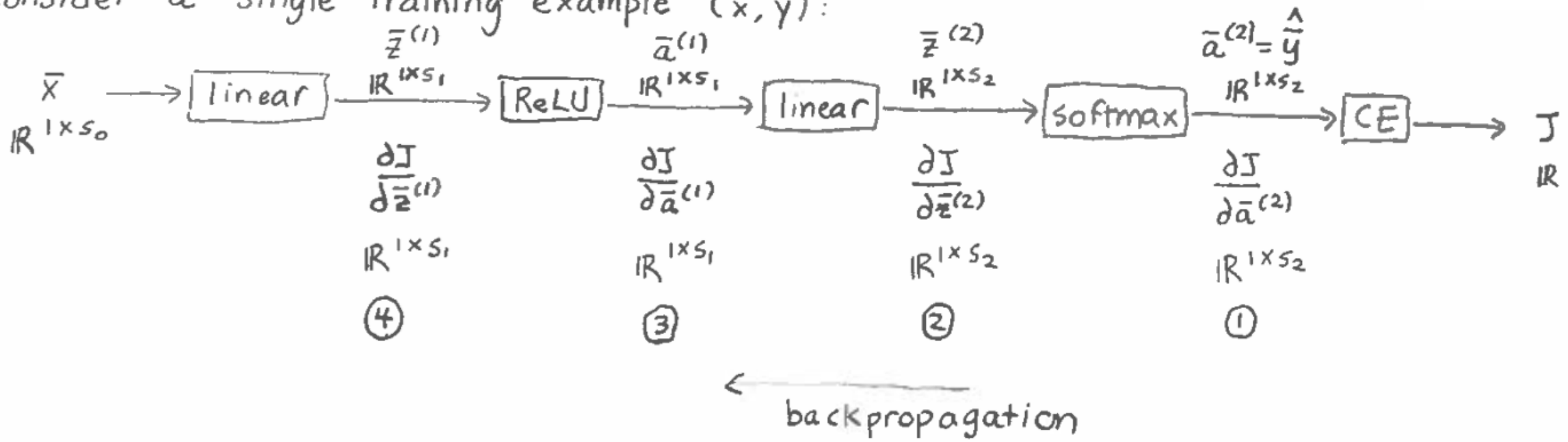
```
for i in range (numIters):
    grad = calcGradient (...)
    weights = weights - γ · grad
```

How to calculate $\frac{\partial J}{\partial W^{(1)}}$, $\frac{\partial J}{\partial \vec{b}^{(1)}}$, $\frac{\partial J}{\partial W^{(2)}}$, $\frac{\partial J}{\partial \vec{b}^{(2)}}$? Backpropagation.

Consider a single training example $(\bar{x}, y)$:



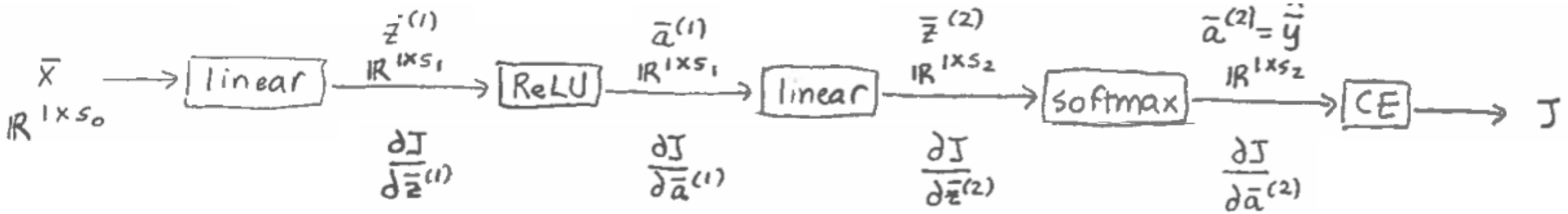
Consider a single training example $(\bar{x}, y)$:



① $$\frac{\partial J}{\partial a^{(2)}_k} = \frac{\partial J}{\partial \hat{y}_k} = \frac{\partial}{\partial \hat{y}_k}\left\{ -\sum_{c=1}^{C} y_c \cdot \log(\hat{y}_c) \right\} =$$

$$\frac{\partial J}{\partial \bar{a}^{(2)}} =$$

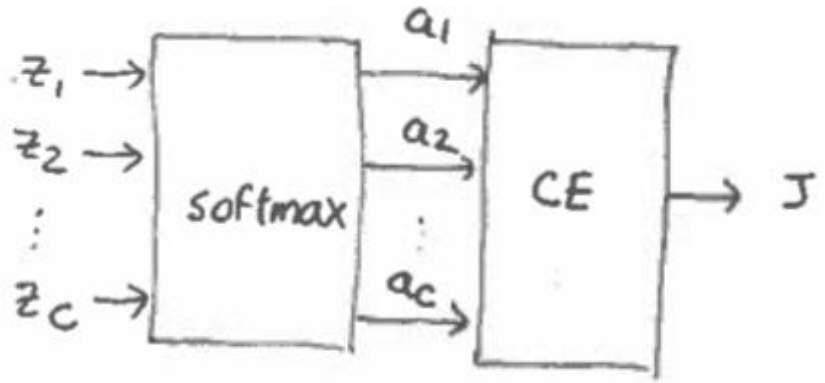
Note: because $\bar{y}$ is one-hot encoded, only 1 element is non-zero



② $$\frac{\partial J}{\partial z_j} = \sum_{i=1}^{C} \frac{dJ}{da_i} \cdot \frac{da_i}{dz_j}$$

$$= \sum_{i} -\frac{y_i}{a_i} \cdot a_i(\delta_{ij} - a_j$$

$$= -\sum_{i=1}^{C} y_i \delta_{ij} + a_j \sum_{i=1}^{C} y_i$$

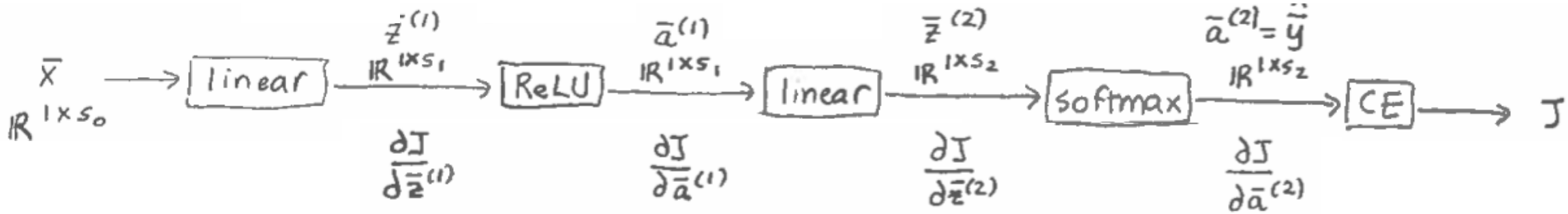


$\frac{da_i}{dz_j}$: see proof in appendix

$$\frac{\partial J}{\partial \bar{z}^{(2)}} =$$

$$\text{softmax}(\bar{z})_i = \frac{e^{z_i}}{e^{z_1} + e^{z_2} + \cdots + e^{z_C}}$$

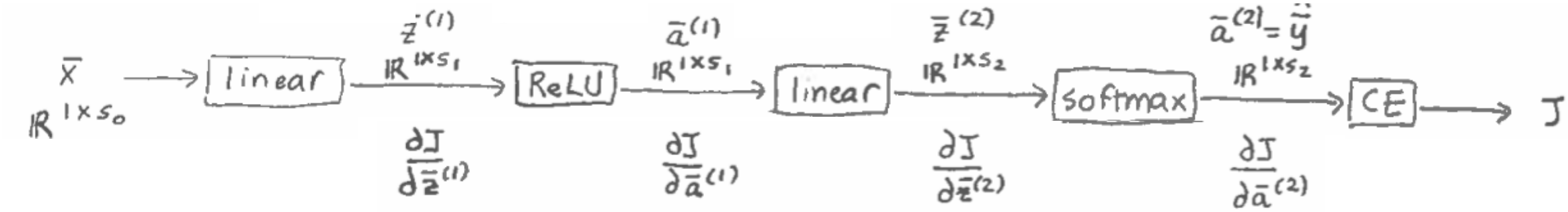
($\bar{z} \in \mathbb{R}^C$)



③ $\frac{\partial J}{\partial \bar{a}^{(1)}} = \frac{\partial J}{\partial \bar{z}^{(2)}} \times \underbrace{W^{(2)T}}$

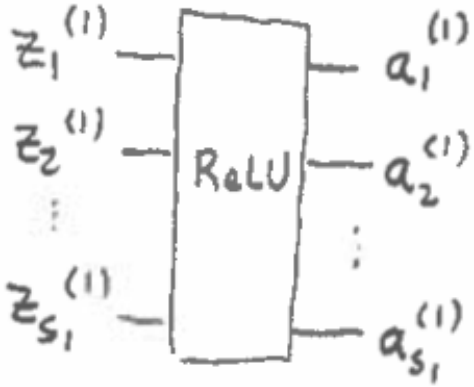
pf: see appendix

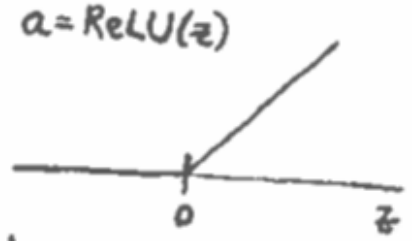
Recall: $\bar{z}^{(2)} = \bar{a}^{(1)} W^{(2)} + \bar{b}^{(2)}$



④

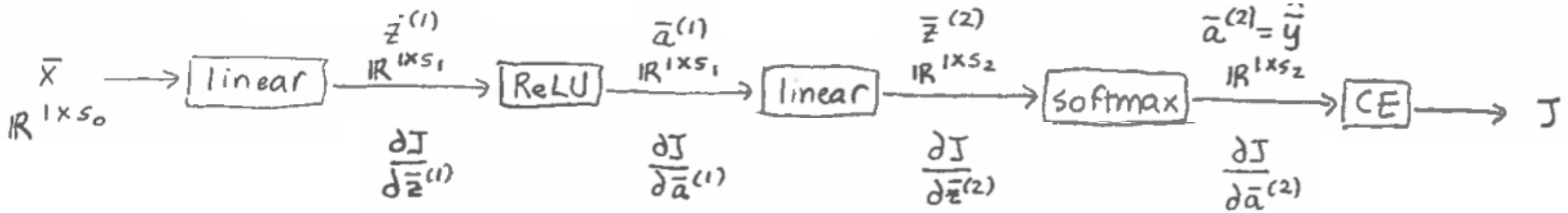
$$\frac{\partial J}{\partial z_i^{(1)}} = \frac{\partial J}{\partial a_i^{(1)}} \cdot \frac{\partial a_i^{(1)}}{\partial z_i^{(1)}}$$





$$\frac{\partial J}{\partial \bar{z}^{(1)}} = \frac{\partial J}{\partial \bar{a}^{(1)}} \odot \dot{m}^{(1)} \quad \text{where} \quad m_i^{(1)} =$$

$\odot$ ↑ elementwise multiply

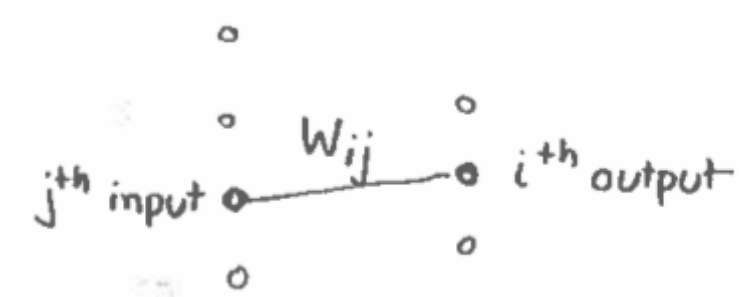


Now calculate gradients of interest:

(A) $$\frac{\partial J}{\partial W_{ij}^{(1)}} = \frac{\partial J}{\partial z_i^{(1)}} \cdot \frac{\partial z_i^{(1)}}{\partial W_{ij}^{(1)}}$$

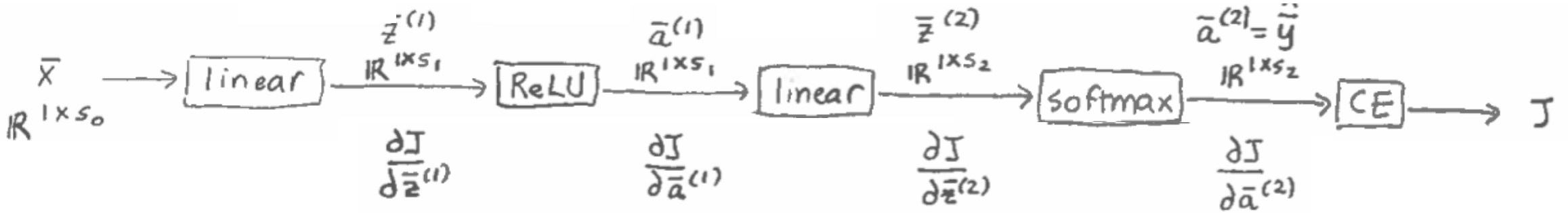
$$= \frac{\partial J}{\partial z_i^{(1)}} \cdot \frac{\partial}{\partial W_{ij}^{(1)}} \left\{ \sum_{k=1}^{s_0} W_{ik}^{(1)} x_k + b_i^{(1)} \right\}$$

$$= \frac{\partial J}{\partial z_i^{(1)}} \cdot x_j$$



$$\frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial z_1^{(1)}} x_1 & \cdots & \frac{\partial J}{\partial z_{s_1}^{(1)}} \cdot x_1 \\ \vdots & & \\ \frac{\partial J}{\partial z_1^{(1)}} x_{s_0} & \cdots & \frac{\partial J}{\partial z_{s_1}^{(1)}} \cdot x_{s_0} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{s_0} \end{bmatrix} \times \begin{bmatrix} \frac{\partial J}{\partial z_1^{(1)}} & \cdots & \frac{\partial J}{\partial z_{s_1}^{(1)}} \end{bmatrix}$$

$\mathbb{R}^{s_0\times 1}$ $\qquad$ $\mathbb{R}^{1\times s_1}$



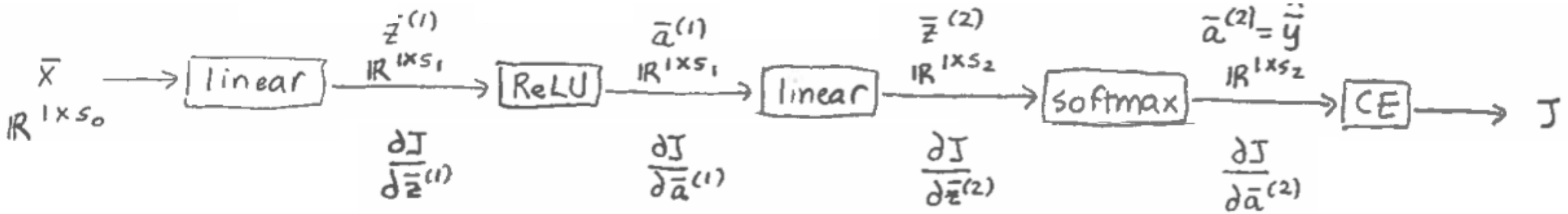
(B)

$$\frac{\partial J}{\partial b_i^{(1)}} = \frac{\partial J}{\partial z_i^{(1)}} \cdot \frac{\partial z_i^{(1)}}{\partial b_i^{(1)}}$$

$$= \frac{\partial J}{\partial z_i^{(1)}} \cdot \frac{\partial}{\partial b_i^{(1)}} \left\{ \sum_{k=1}^{s_0} W_{ik}^{(1)} x_k + b_i^{(1)} \right\}$$

$$= \frac{\partial J}{\partial z_i^{(1)}}$$

$$\frac{\partial J}{\partial \bar{b}^{(1)}} =$$



(C) $\frac{\partial J}{\partial W^{(2)}} = \bar{a}^{(1)T} \times \frac{\partial J}{\partial \bar{z}^{(2)}}$ ← from (2) (same pf as (A))

(D) $\frac{\partial J}{\partial b^{(2)}} = \frac{\partial J}{\partial \bar{z}^{(2)}}$ ← from (2) (same pf as (B))

Summary:

- forward prop → calc intermediate outputs & prediction
- backprop → calc gradients

What is mini-batch gradient descent?

Batch GD

```
foreach epoch:
    grad = calc_grad(dataset)
    wts = wts - γ·grad
```

- batch size = N
- 1 update per epoch
- \+ reliable grad estimate
- − slow to train

Mini-batch GD

split dataset into mini-batches

```
foreach epoch:
    foreach minibatch:
        grad = calc_grad(minibatch)
        wts = wts - γ·grad
```

Comments:
- pick K as large as will fit in GPU memory
  - ↳ reduces training time
- shuffle training data after each epoch

Practical Tips

- vectorization speeds up training significantly
  - ↳ processing samples in parallel
- mini-batch GD speeds up training significantly
- Training Process:
  1. split data: train, validation, test
  2. train model for N epochs ← 1 epoch = 1 pass thru train data
  3. periodically measure performance on validation data
  4. stop training when valid loss stops improving

Appendix

Derivative of softmax

We want to calc. $\frac{\partial a_i}{\partial z_j}$. Two cases.

when $i=j$:

$$\frac{\partial a_i}{\partial z_i} = \frac{\partial}{\partial z_i}\left\{\frac{e^{z_i}}{e^{z_1}+\cdots+e^{z_c}}\right\}$$

$$= \frac{(e^{z_1}+\cdots+e^{z_c})\cdot e^{z_i} - e^{z_i}\cdot e^{z_i}}{(e^{z_1}+\cdots+e^{z_c})^2}$$

$$= \frac{e^{z_i}}{e^{z_1}+\cdots+e^{z_c}}\cdot\left[1-\frac{e^{z_i}}{e^{z_1}+\cdots+e^{z_c}}\right] = a_i(1-a_i)$$

when $i\neq j$:

$$\frac{\partial a_i}{\partial z_j} = \frac{\partial}{\partial z_j}\left\{\frac{e^{z_i}}{e^{z_1}+\cdots+e^{z_c}}\right\}$$

$$= \frac{(e^{z_1}+\cdots+e^{z_c})\cdot 0 - e^{z_i}e^{z_j}}{(e^{z_1}+\cdots+e^{z_c})^2}$$

$$= -\frac{e^{z_i}}{e^{z_1}+\cdots+e^{z_c}}\cdot\frac{e^{z_j}}{e^{z_1}+\cdots+e^{z_c}} = -a_i a_j$$

We can write as:

$$\frac{\partial a_i}{\partial z_j} = \begin{cases} a_i(1-a_j) & \text{if } i=j \\ a_i(-a_j) & \text{if } i\neq j \end{cases} = a_i(\delta_{ij}-a_j) \quad \text{where } \delta_{ij} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{if } i\neq j \end{cases}$$

Backpropagation through linear layer

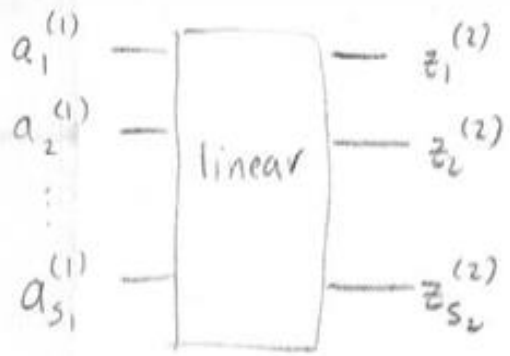
$$\frac{\partial J}{\partial a_k^{(1)}} = \sum_{i=1}^{S_2} \frac{\partial J}{\partial z_i^{(2)}} \cdot \frac{\partial z_i^{(2)}}{\partial a_k^{(1)}}$$

$$= \sum_{i=1}^{S_2} \frac{\partial J}{\partial z_i^{(2)}} \cdot \frac{\partial}{\partial a_k^{(1)}} \left\{ \bar{w}_i^{(2)T} \bar{a}^{(1)} + b_i^{(2)} \right\}$$

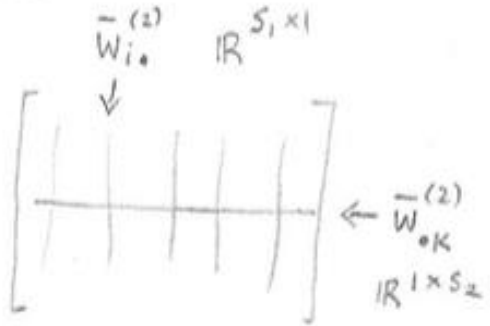
$$= \sum_{i=1}^{S_2} \frac{\partial J}{\partial z_i^{(2)}} \cdot W_{ik}^{(2)}$$

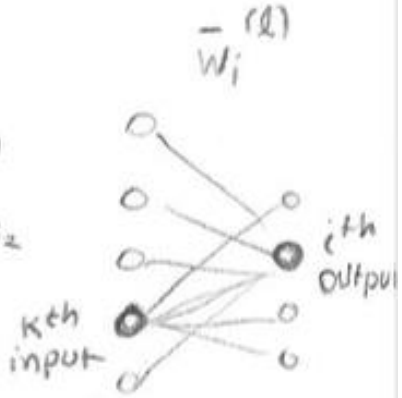
$$= \frac{\partial J}{\partial \bar{z}^{(2)}} \cdot \bar{W}_{\cdot k}^{(2)T}$$

$\mathbb{R}^{1 \times S_2}$ $\quad$ $\mathbb{R}^{S_2 \times 1}$



recall: $W^{(2)} =$





$$\frac{\partial J}{\partial \bar{a}^{(1)}} = \left[\frac{\partial J}{\partial \bar{z}^{(2)}} \cdot \bar{W}_{\cdot 1}^{(2)T} \quad \frac{\partial J}{\partial \bar{z}^{(2)}} \cdot \bar{W}_{\cdot 2}^{(2)T} \quad \cdots \quad \frac{\partial J}{\partial \bar{z}^{(2)}} \cdot \bar{W}_{\cdot S_1}^{(2)T} \right]$$

$$= \frac{\partial J}{\partial \bar{z}^{(2)}} \times \left[\bar{W}_{\cdot 1}^{(2)T} \quad \bar{W}_{\cdot 2}^{(2)T} \quad \cdots \quad \bar{W}_{\cdot S_1}^{(2)T} \right]$$

$\mathbb{R}^{1 \times S_2}$ $\quad$ $\mathbb{R}^{S_2 \times S_1}$

$$= \frac{\partial J}{\partial \bar{z}^{(2)}} \times W^{(2)T}$$