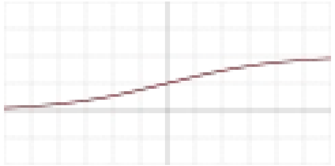
Backpropagation through Time

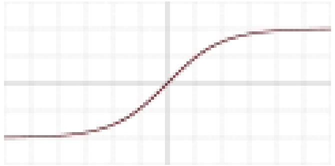
A Mathematical Overview

A Neural Network

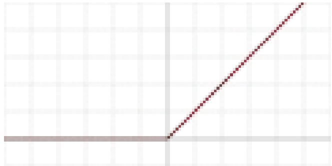
Activation Functions



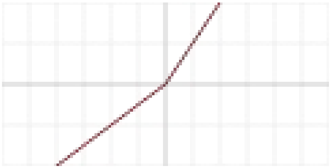
Sigmoid



Tanh



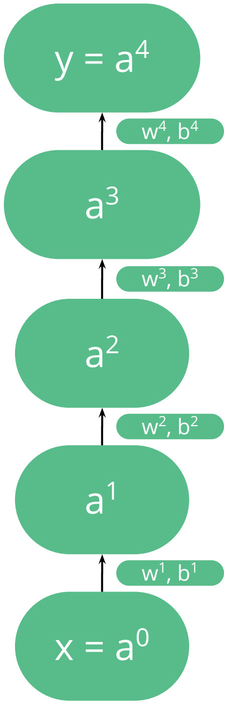
Relu



Leaky Relu

Notation

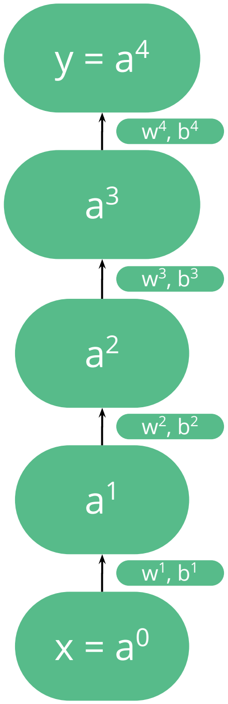
- Three Hidden Layer Neural Network
- x -> Input and y -> Output
- w-> Weight and b -> Bias



Notation

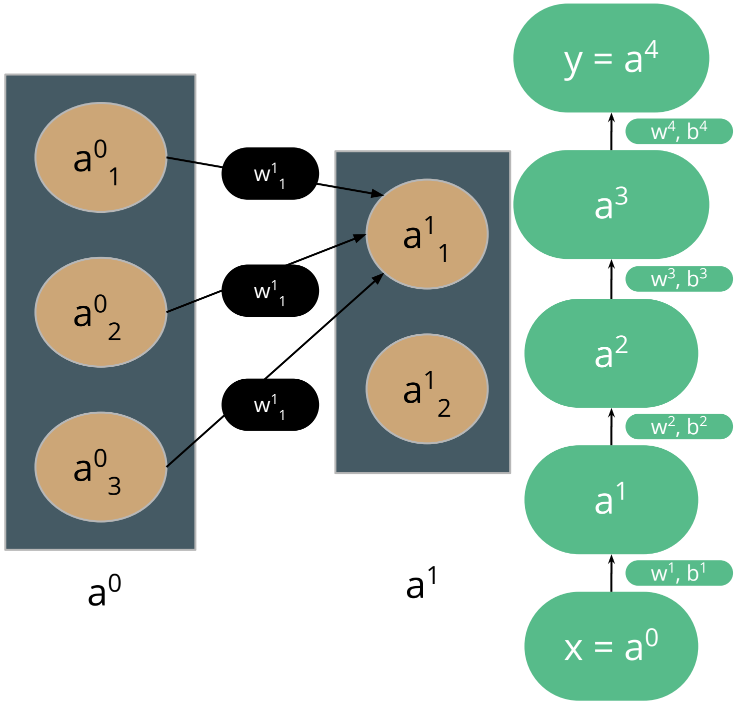
For a single data point

- Vectors -> x, a^1, a^2, a^3, y
- Vectors -> b^1, b^2, b^3, b^4
- Matrices -> w^1, w^2, w^3, w^4



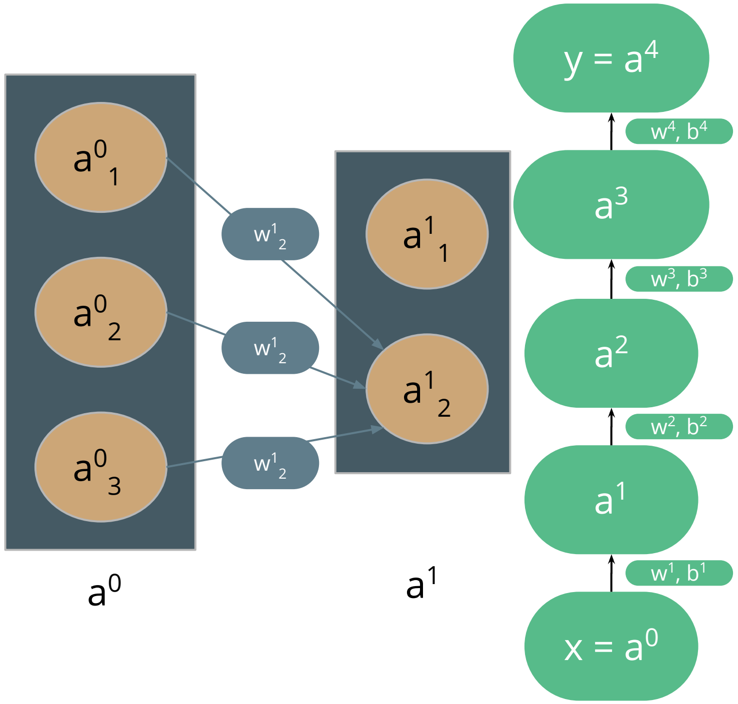
Forward Pass

- $a^1_1 = f(w^1_{11}a^0_1 + w^1_{12}a^0_2 + w^1_{13}a^0_3)$
- f: Non Linear Activation Function



Forward Pass

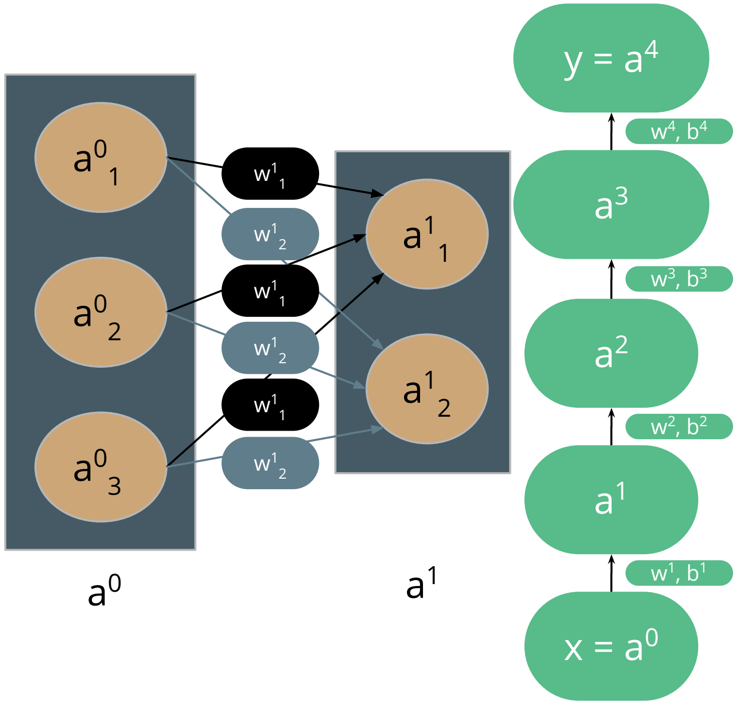
- $a^1_2 = f(\, w^1_{21}a^0_1 + w^1_{22}a^0_2 + w^1_{23}a^0_3)$



Forward Pass

Collecting the two

- $a^1_1 = f(\, w^1_{11}a^0_1 + w^1_{12}a^0_2 + w^1_{13}a^0_3 \,)$
- $a^1_2 = f(\, w^1_{21}a^0_1 + w^1_{22}a^0_2 + w^1_{23}a^0_3 \,)$



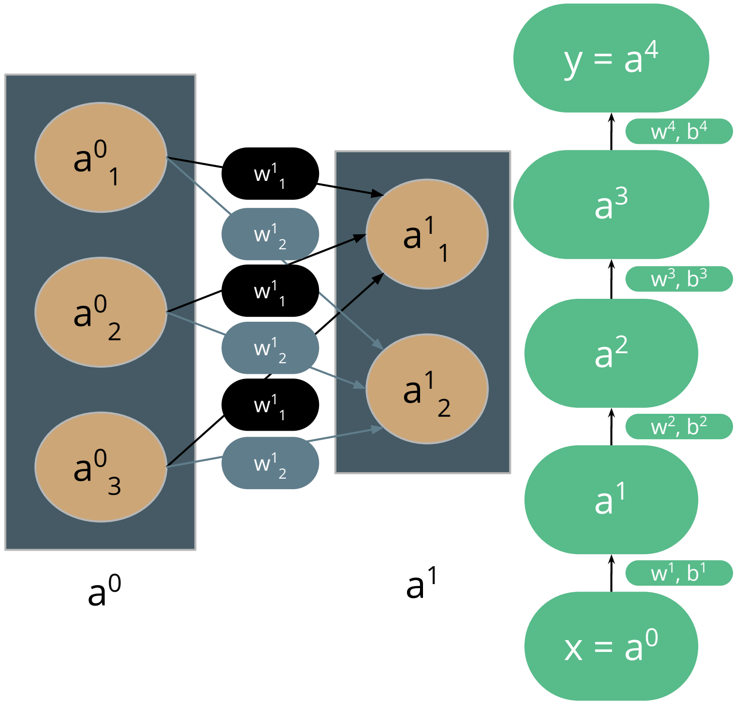
Forward Pass

Collecting the two

- $a^1_1 = f(\ w^1_{11}a^0_1 + w^1_{12}a^0_2 + w^1_{13}a^0_3\ )$
- $a^1_2 = f(\ w^1_{21}a^0_1 + w^1_{22}a^0_2 + w^1_{23}a^0_3\ )$

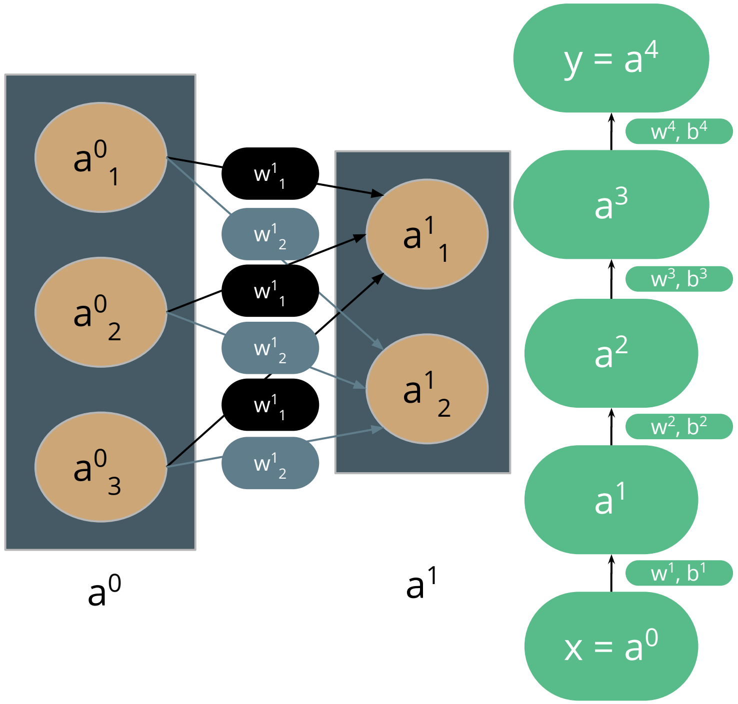
is the same as

- $z^1_1 = w^1_{11}a^0_1 + w^1_{12}a^0_2 + w^1_{13}a^0_3$
- $a^1_1 = f(\ z^1_1\ )$
- $z^1_2 = w^1_{21}a^0_1 + w^1_{22}a^0_2 + w^1_{23}a^0_3$
- $a^1_2 = f(\ z^1_2\ )$



Forward Pass

- $z^1_1 = w^1_{11}a^0_1 + w^1_{12}a^0_2 + w^1_{13}a^0_3$
- $a^1_1 = f(\, z^1_1 \,)$
- $z^1_2 = w^1_{21}a^0_1 + w^1_{22}a^0_2 + w^1_{23}a^0_3$
- $a^1_2 = f(\, z^1_2 \,)$

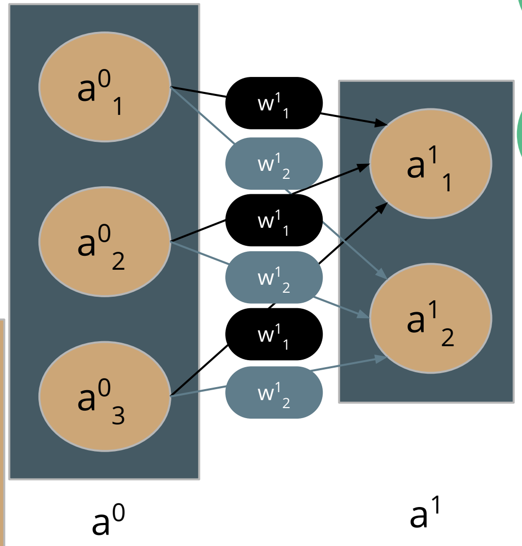


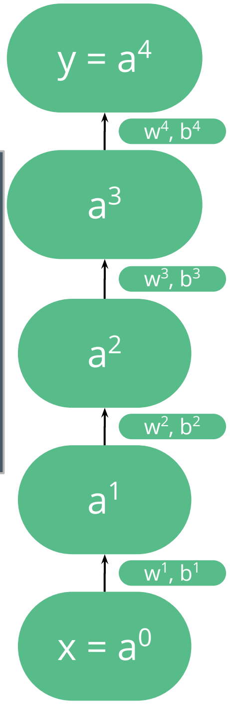
Forward Pass

- $z^1_1 = w^1_{11}a^0_1 + w^1_{12}a^0_2 + w^1_{13}a^0_3$
- $a^1_1 = f(\ z^1_1\ )$
- $z^1_2 = w^1_{21}a^0_1 + w^1_{22}a^0_2 + w^1_{23}a^0_3$
- $a^1_2 = f(\ z^1_2\ )$

is the same as

$$\underset{w^1}{\begin{bmatrix} W^1_{11} & W^1_{12} & W^1_{13} \\ W^1_{21} & W^1_{22} & W^1_{23} \end{bmatrix}} \times \underset{a^0}{\begin{bmatrix} a^0_1 \\ a^0_2 \\ a^0_3 \end{bmatrix}} = \underset{z^1}{\begin{bmatrix} z^1_1 \\ z^1_2 \end{bmatrix}} \xrightarrow{f} \underset{a^1}{\begin{bmatrix} f(z^1_1) \\ f(z^1_2) \end{bmatrix}} = \underset{a^1}{\begin{bmatrix} a^1_1 \\ a^1_2 \end{bmatrix}}$$



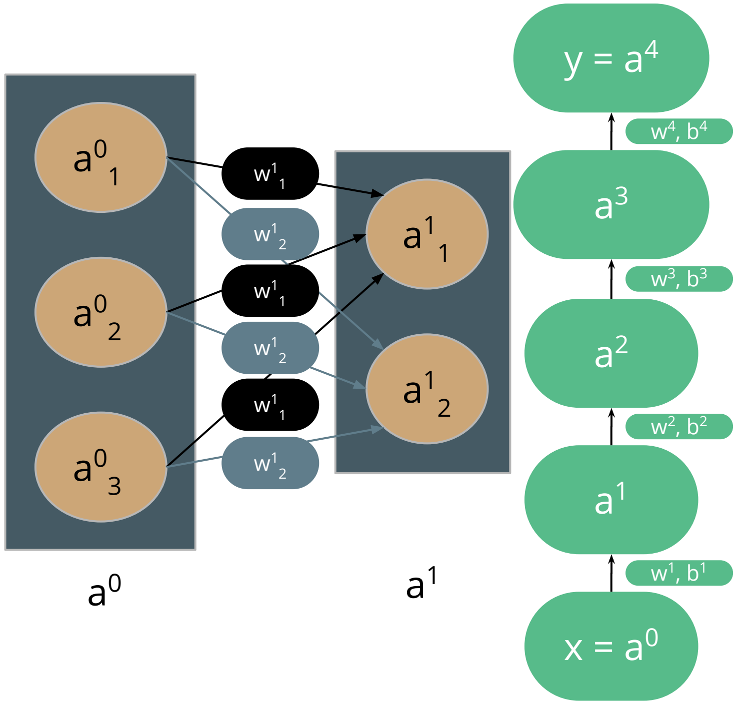


Forward Pass

- $z^1_1 = w^1_{11}a^0_1 + w^1_{12}a^0_2 + w^1_{13}a^0_3$
- $a^1_1 = f(z^1_1)$
- $z^1_2 = w^1_{21}a^0_1 + w^1_{22}a^0_2 + w^1_{23}a^0_3$
- $a^1_2 = f(z^1_2)$

is the same as

- $z^1 = w^1 * a^0$
- $a^1 = f(z^1)$



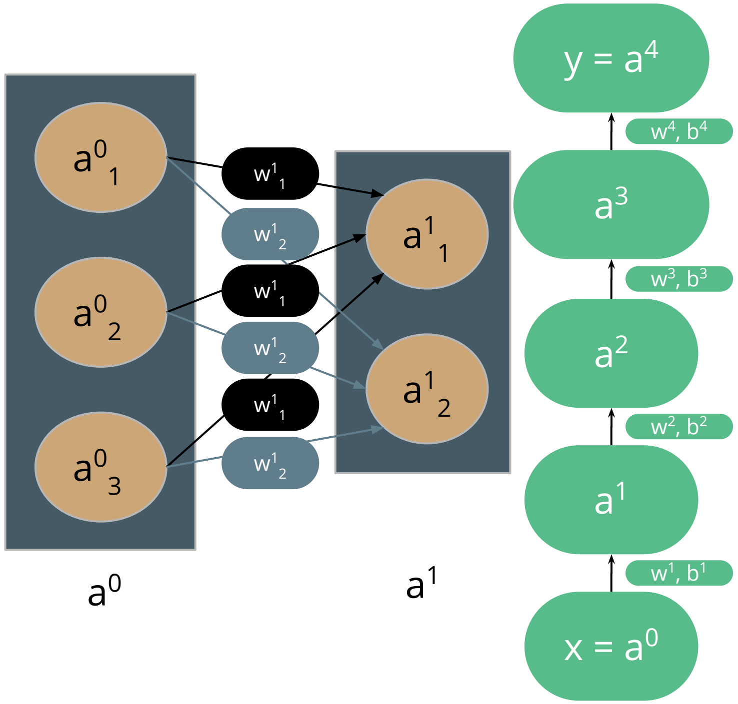
Forward Pass

Adding in the bias term as well

- $z^1_1 = w^1_{11}a^0_1 + w^1_{12}a^0_2 + w^1_{13}a^0_3 + b^1_1$
- $a^1_1 = f(z^1_1)$
- $z^1_2 = w^1_{21}a^0_1 + w^1_{22}a^0_2 + w^1_{23}a^0_3 + b^1_2$
- $a^1_2 = f(z^1_2)$

is the same as

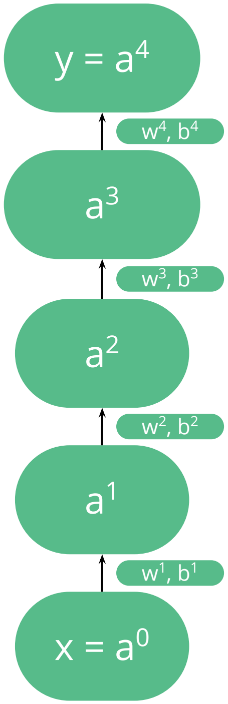
- $z^1 = w^1 * a^0 + b^1$
- $a^1 = f(z^1)$



Forward Pass

The complete forward pass

- $a^0 = x$
- $z^1 = w^1 * a^0 + b^1$
- $a^1 = f(z^1)$
- $z^2 = w^2 * a^1 + b^2$
- $a^2 = f(z^2)$
- $z^3 = w^3 * a^2 + b^3$
- $a^3 = f(z^3)$
- $z^4 = w^4 * a^3 + b^4$
- $a^4 = f(z^4)$
- $y = a^4$



Forward Pass

The Input

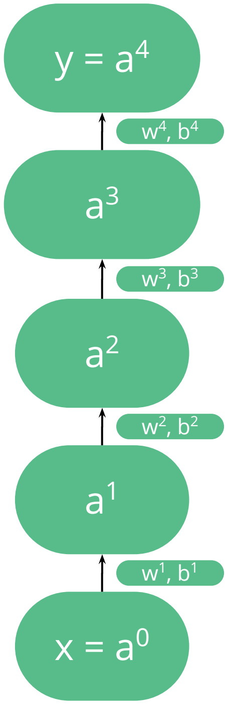
- $a^0 = x$

For l = 1, ... , L layers

- $z^l = w^l * a^{l-1} + b^l$
- $a^l = f(z^l)$

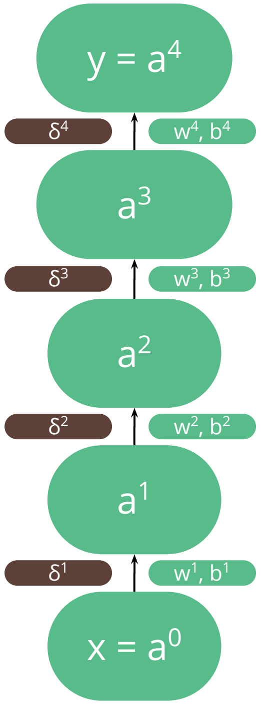
Finally

- $y = a^L$



Notation

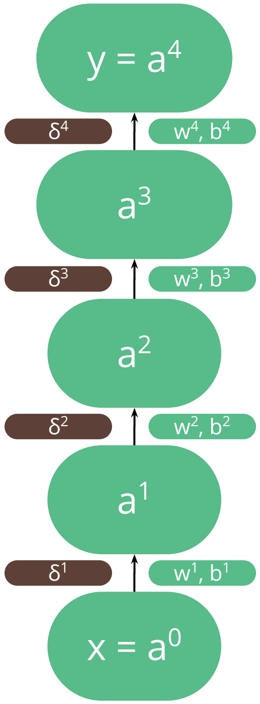
- t -> Ground Truth Output
- C -> Cost Function
- δ -> Gradient



The Cost Function

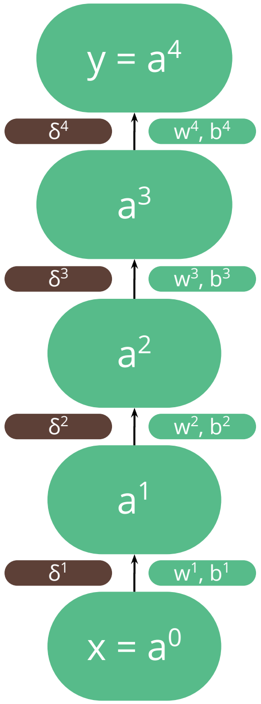
For a scalar output

- Mean Squared Error: $C = ½ * (y - t)^2$
- Cross Entropy: $C = t * \ln(y) + (1-t) * \ln(1-y)$



Backpropagation

- Goal: Compute ∂C / ∂w and ∂C / ∂b
- Why: Use them for Stochastic Gradient Descent
- Define: $\delta^l = \partial C / \partial z^l$



Backward Pass

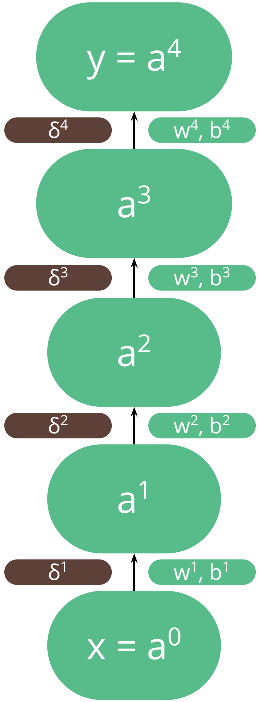
$\delta^4 = \partial C/\partial z^4 = \partial C/\partial y * \partial y/\partial z^4$

Now

- $\partial C/\partial y = (y - t)$
- $\partial y/\partial z^4 = \partial a^4/\partial z^4 = f'(z^4)$

where $f'(.)$ is derivative of $f(.)$ w.r.t $(.)$

$\Rightarrow \delta^4 = (y - t) * f'(z^4)$



Backward Pass

$\delta^3 = \partial C/\partial z^3$

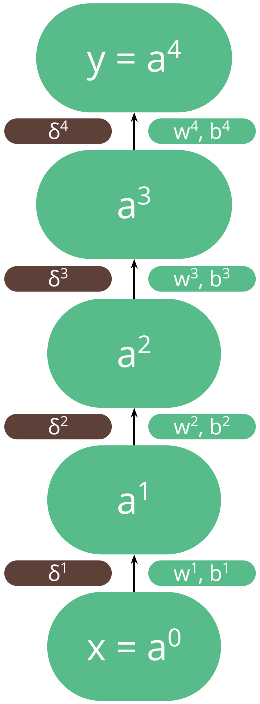
Now

- $z^4_1 = \ldots + w^4_{1j} * f(z^3_j) + \ldots$
- $z^4_k = \ldots + w^4_{kj} * f(z^3_j) + \ldots$

i.e. all elements of z^4 depend on z^3_j

Thus, by chain rule we can say that

$\delta^3_j = \partial C/\partial z^3_j = \sum_k \partial C/\partial z^4_k * \partial z^4_k/\partial z^3_j$



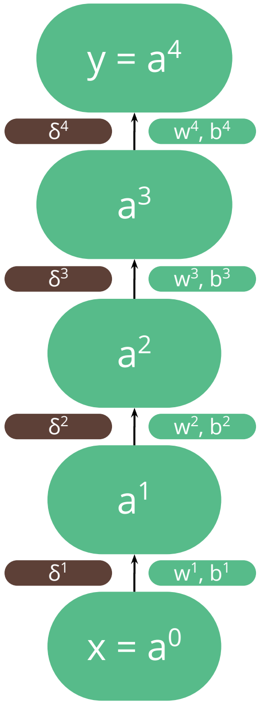
Backward Pass

$\delta^3_j = \partial C/\partial z^3_j = \sum_k \partial C/\partial z^4_k * \partial z^4_k/\partial z^3_j$
$\Rightarrow \delta^3_j = \sum_k \partial C/\partial z^4_k * \partial z^4_k/\partial a^3_j * \partial a^3_j/\partial z^3_j$

Now

- $\partial C/\partial z^4_k = \delta^4_k$
- $\partial z^4_k/\partial a^3_j = w^4_{kj}$ [As $z^4_k = \dots + w^4_{kj} * a^3_j + \dots$]
- $\partial a^3_j/\partial z^3_j = f'(z^3_j)$

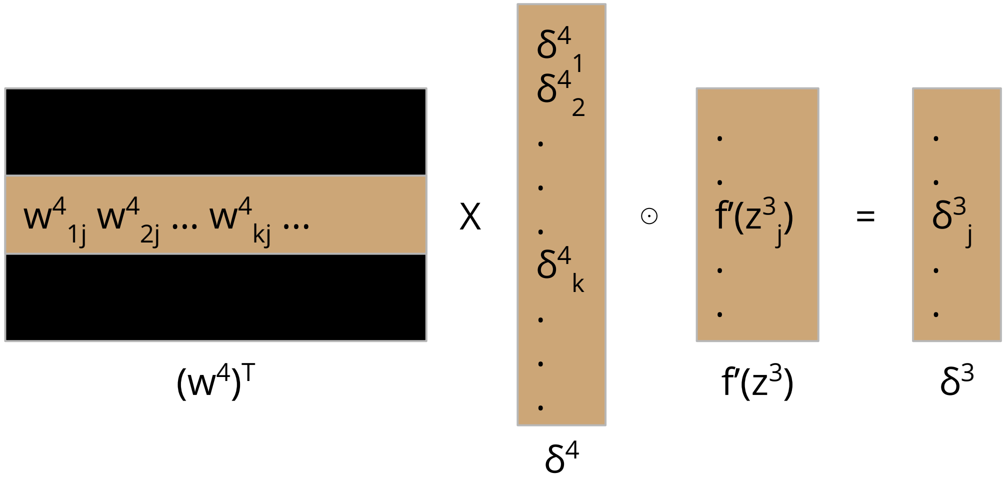
$\Rightarrow \delta^3_j = (\sum_k \delta^4_k * w^4_{kj}) * f'(z^3_j)$

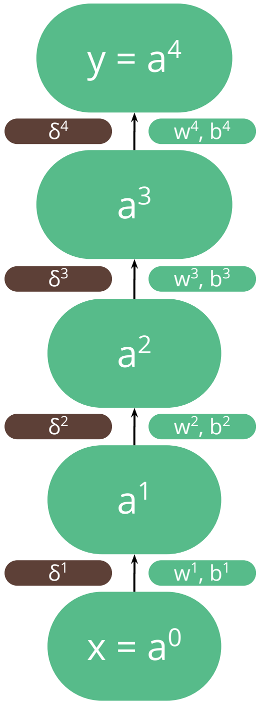


Backward Pass

$\delta^3_j = (\sum_k \delta^4_k * w^4_{kj}) * f'(z^3_j)$
$=> \delta^3 = (w^4)^T * \delta^4 \odot f'(z^3)$

where $\odot$ = Element-wise product





Backward Pass

Hence we have

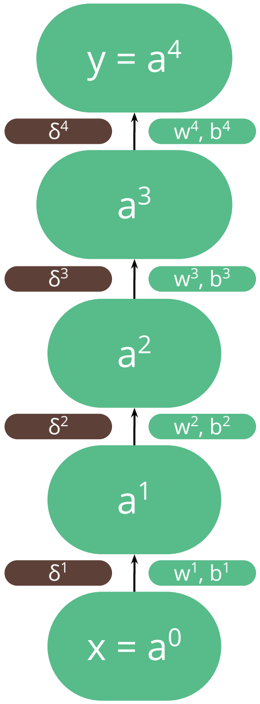
- $\delta^4 = (y - t) * f'(z^4)$
- $\delta^3 = (w^4)^T * \delta^4 \odot f'(z^3)$
- $\delta^2 = (w^3)^T * \delta^3 \odot f'(z^2)$
- $\delta^1 = (w^2)^T * \delta^2 \odot f'(z^1)$

Or in general

$\delta^l = (w^{l+1})^T * \delta^{l+1} \odot f'(z^l)$ for $l = 1, 2, \ldots, L-1$

$\delta^L = \nabla_y C \odot f'(z^L)$

where $\nabla_y C$ is derivative of cost wrt output



Backward Pass

Now for our main objectives: $\partial C/\partial w^l_{jk}$ and $\partial C/\partial b^l_j$

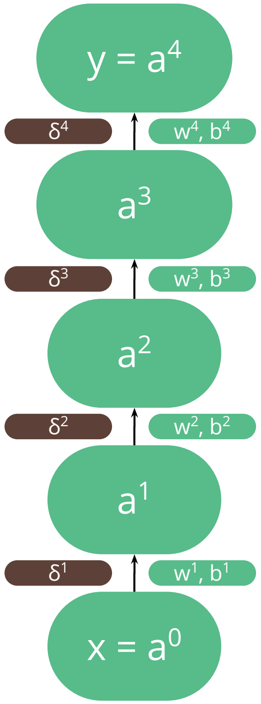
$\partial C/\partial w^l_{jk} = \partial C/\partial z^l_j * \partial z^l_j / \partial w^l_{jk}$

Since

- $\partial C/\partial z^l_j = \delta^l_j$
- $\partial z^l_j / \partial w^l_{jk} = a^{l-1}_k$

[As $z^l_j = \dots + w^l_{jk} * a^{l-1}_k + \dots$]

=> $\partial C/\partial w^l_{jk} = \delta^l_j \, a^{l-1}_k$



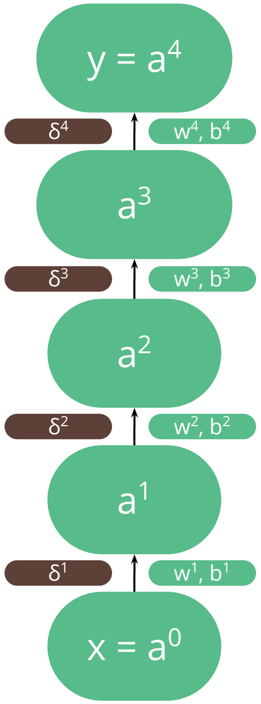
Backward Pass

$\partial C/\partial w^l_{jk} = \delta^l_j \, a^{l-1}_k$

Or in general

$\partial C/\partial w^l = \delta^l * (a^{l-1})^T$ for $l = 1, \ldots, L$

$$\underbrace{\begin{bmatrix} \delta^l_1 a^{l-1}_1 & \ldots & \delta^l_1 a^{l-1}_m \\ \vdots & & \vdots \\ \delta^l_n a^{l-1}_1 & \ldots & \delta^l_n a^{l-1}_m \end{bmatrix}}_{\partial C/\partial w^l} = \underbrace{\begin{bmatrix} \delta^l_1 \\ \vdots \\ \delta^l_n \end{bmatrix}}_{\delta^l} \times \underbrace{\begin{bmatrix} a^{l-1}_1 & \ldots & a^{l-1}_m \end{bmatrix}}_{(a^{l-1})^T}$$



Backward Pass

Also

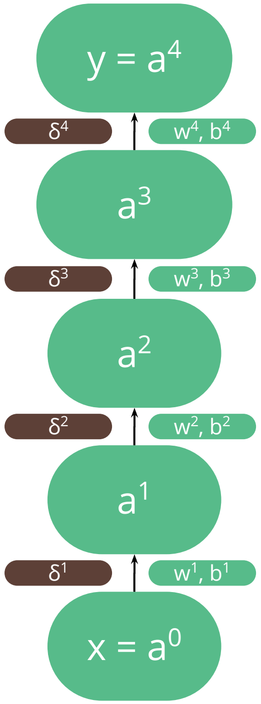
$\partial C/\partial b^l_j = \partial C/\partial z^l_j * \partial z^l_j / \partial b^l_j$

Since

- $\partial C/\partial z^l_j = \delta^l_j$
- $\partial z^l_j / \partial b^l_j = 1$ [As $z^l_j = \ldots + b^l_j$]

=> $\partial C/\partial b^l_j = \delta^l_j$

Or in general $\partial C/\partial b^l = \delta^l$ for $l = 1, \ldots, L$

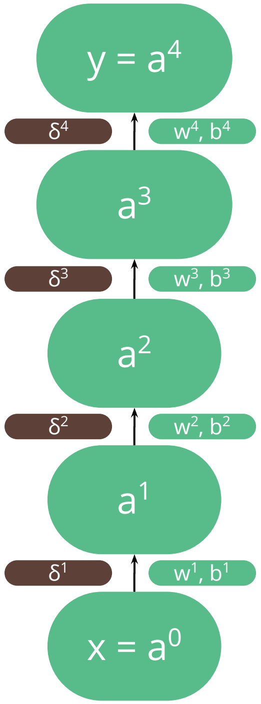


Backward Pass

In general:
$\delta^L = \nabla_y C \odot f'(z^L)$ where $\nabla_y C$ is derivative of cost wrt output

Then for $l = 1, 2, \ldots, L-1$
$\delta^l = (w^{l+1})^T * \delta^{l+1} \odot f'(z^l)$ where $\odot$ stands for element wise product

Finally for $l = 1, \ldots, L$
$\partial C/\partial w^l = \delta^l * (a^{l-1})^T$
$\partial C/\partial b^l = \delta^l$



Summary

Forward Pass

The Input

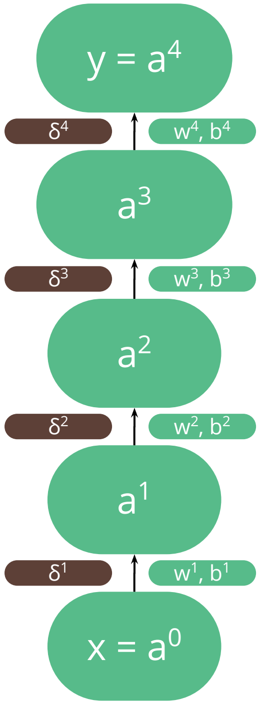
- $a^0 = x$

For l = 1, ... , L layers

- $z^l = w^l * a^{l-1} + b^l$
- $a^l = f(z^l)$

Finally

- $y = a^L$

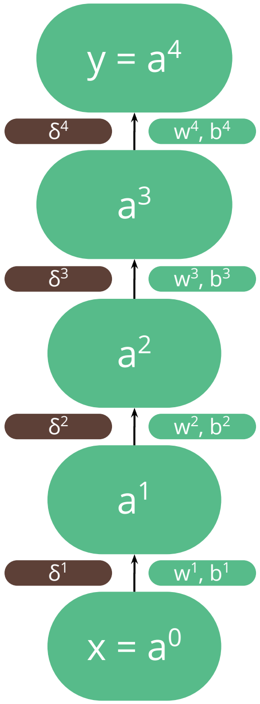


Backward Pass

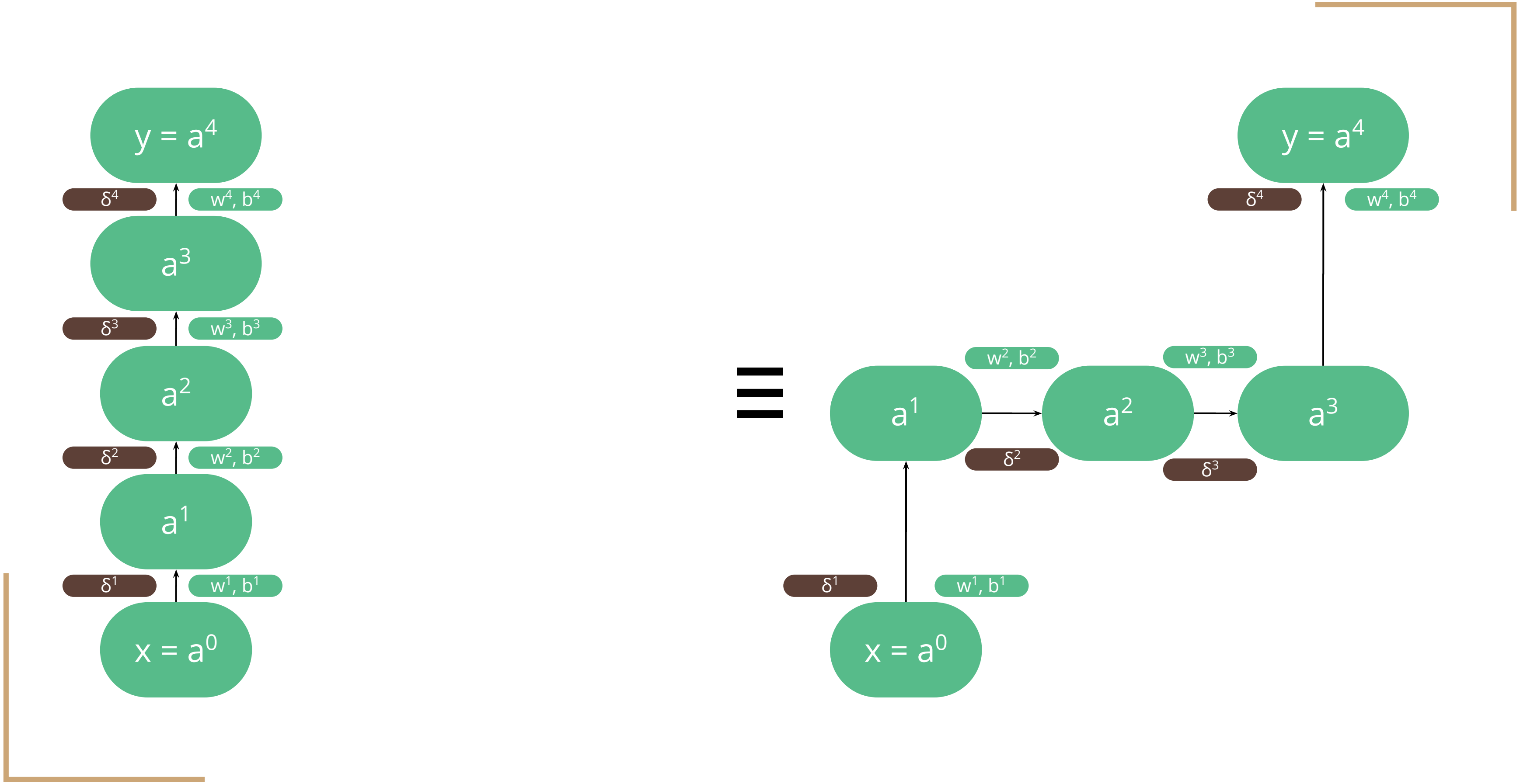
$\delta^L = \nabla_y C \odot f'(z^L)$
where $\nabla_y C$ is derivative of cost wrt output

Then for l = 1, 2, ..., L-1
$\delta^l = (w^{l+1})^T * \delta^{l+1} \odot f'(z^l)$
where $\odot$ stands for element wise product

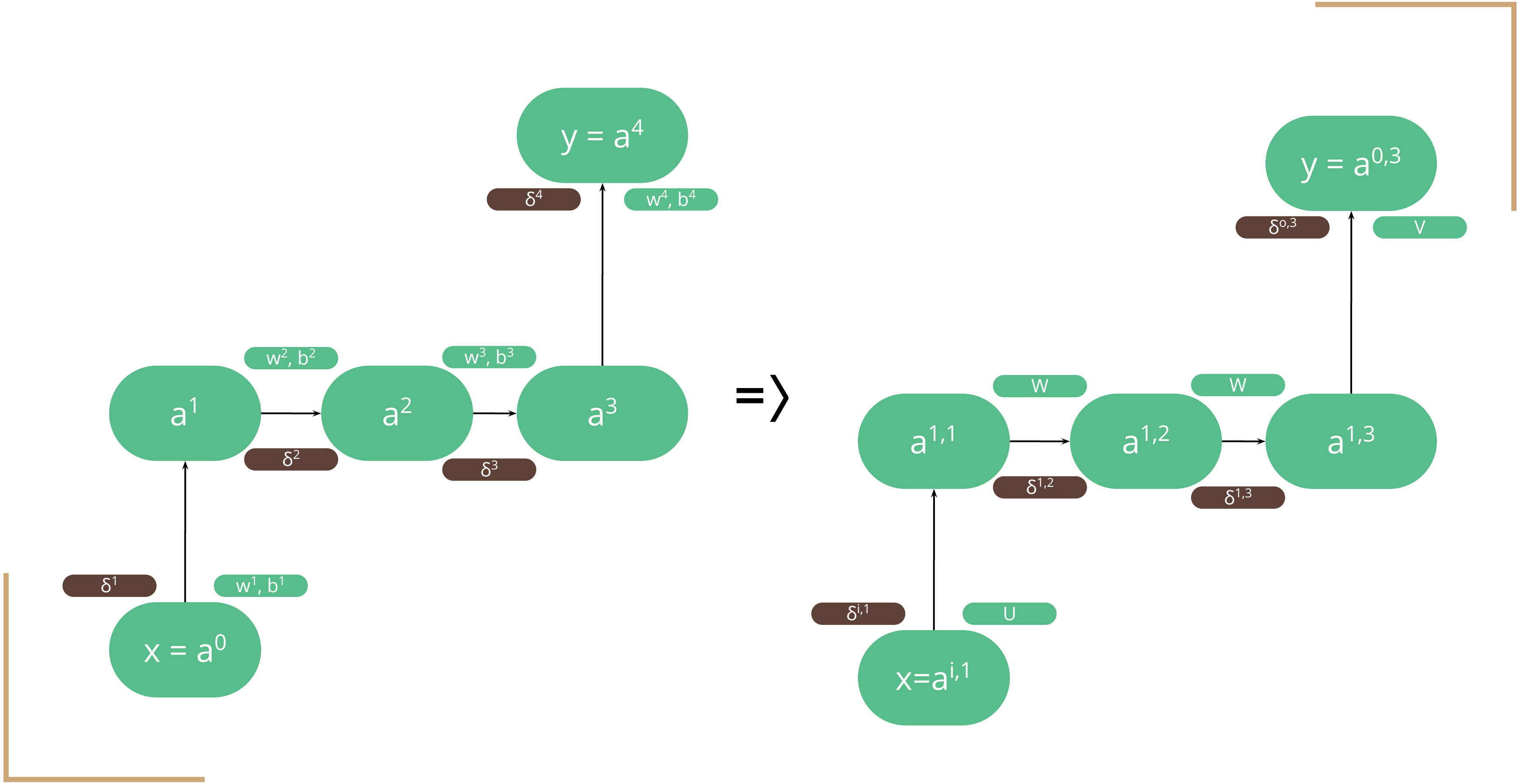
Finally for l = 1, ..., L
$\partial C/\partial w^l = \delta^l * (a^{l-1})^T$
$\partial C/\partial b^l = \delta^l$



A SISO Recurrent Neural Network



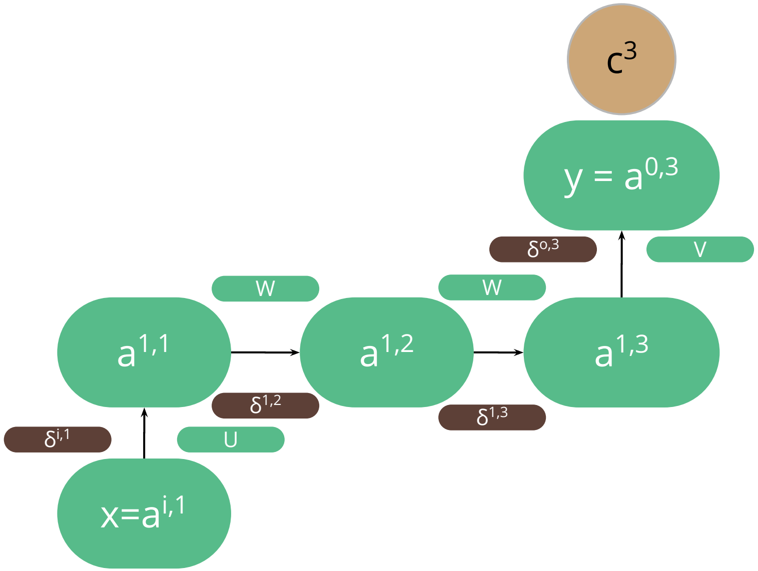
y = a⁴
δ⁴
w⁴, b⁴
a³
δ³
w³, b³
a²
δ²
w², b²
a¹
δ¹
w¹, b¹
x = a⁰
≡
y = a⁴
δ⁴
w⁴, b⁴
w², b²
w³, b³
a¹
a²
a³
δ²
δ³
δ¹
w¹, b¹
x = a⁰



y = a^4
δ^4
w^4, b^4
w^2, b^2
w^3, b^3
a^1
a^2
a^3
δ^2
δ^3
δ^1
w^1, b^1
x = a^0
=〉
y = a^{0,3}
δ^{o,3}
V
W
W
a^{1,1}
a^{1,2}
a^{1,3}
δ^{1,2}
δ^{1,3}
δ^{i,1}
U
x=a^{i,1}

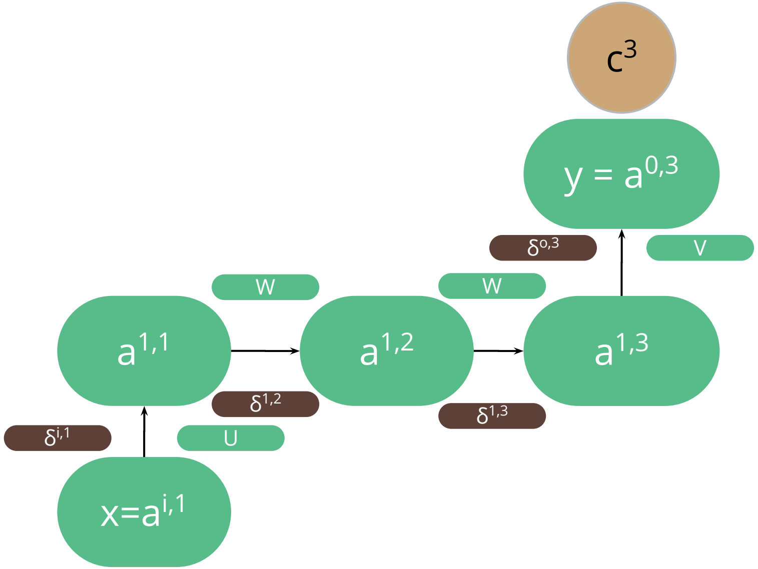
Notation

- In $(a)^{(b),(c)}$:
  - a refers to some quantity
  - b refers to the layer of the network
  - c refers to the time step
- We exclude bias for simplicity
- i -> input and o -> output
- Cost Function C -> C^3

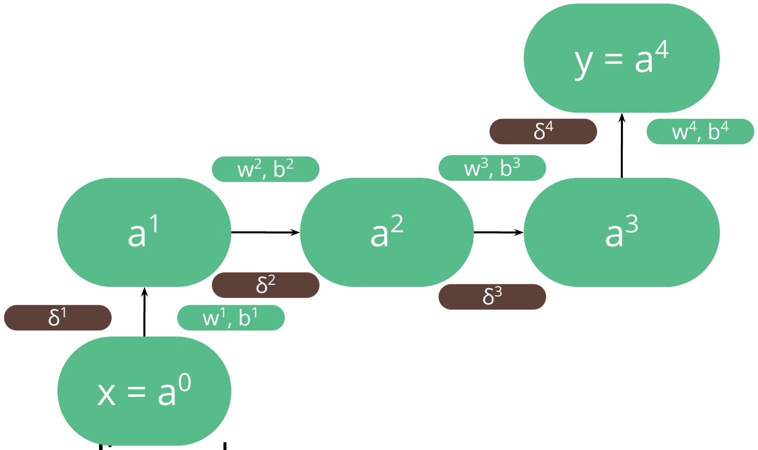


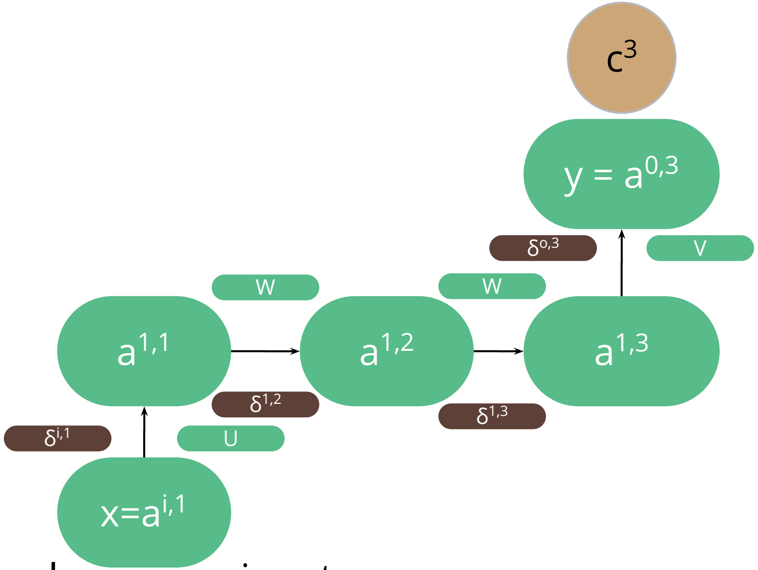
Forward Pass

- $a^{i,1} = x$
- $z^{1,1} = U\, a^{i,1}$
- $a^{1,1} = f(z^{1,1})$
- $z^{1,2} = W\, a^{1,1}$
- $a^{1,2} = f(z^{1,2})$
- $z^{1,3} = W\, a^{1,2}$
- $a^{1,3} = f(z^{1,3})$
- $z^{o,3} = V\, a^{1,3}$
- $a^{o,3} = f(z^{o,3})$
- $y = a^{o,3}$



Backward Pass





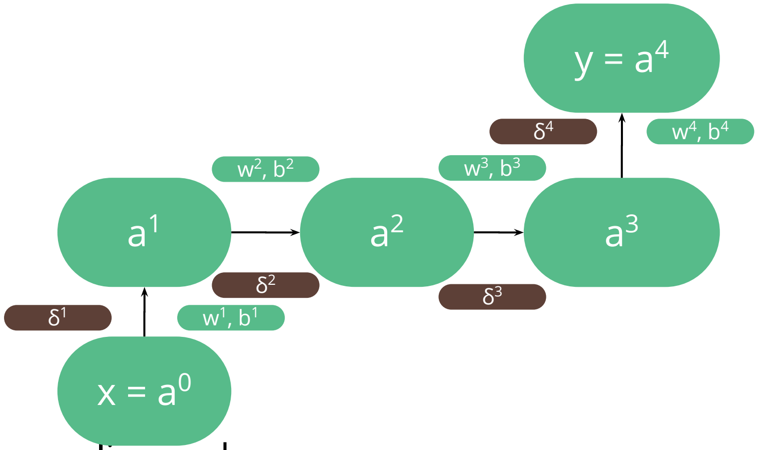
From earlier we have

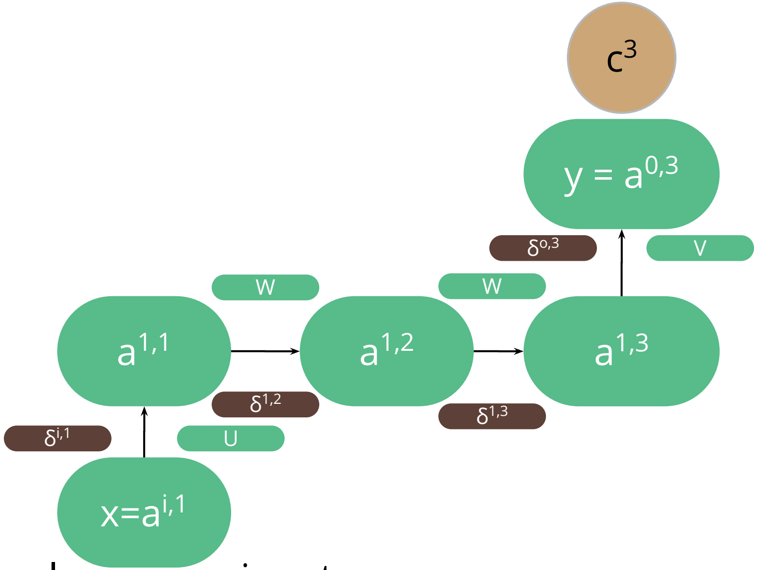
- $\delta^4 = \nabla_y C * f'(z^4)$
- $\delta^3 = (w^4)^T * \delta^4 \odot f'(\, z^3 \,)$
- $\delta^2 = (w^3)^T * \delta^3 \odot f'(\, z^2 \,)$
- $\delta^1 = (w^2)^T * \delta^2 \odot f'(\, z^1 \,)$

By analogy we arrive at

- $\delta^{o,3} = \nabla_y C^3 * f'(z^{o,3})$
- $\delta^{1,3} = V^T * \delta^{o,3} \odot f'(\, z^{1,3} \,)$
- $\delta^{1,2} = W^T * \delta^{1,3} \odot f'(\, z^{1,2} \,)$
- $\delta^{i,1} = W^T * \delta^{1,2} \odot f'(\, z^{1,1} \,)$

Backward Pass





From earlier we have

- $\partial C/\partial w^4 = \delta^4 * (a^3)^T$
- $\partial C/\partial w^3 = \delta^3 * (a^2)^T = \partial C/\partial z^3 * \partial z^3/\partial w^3$
- $\partial C/\partial w^2 = \delta^2 * (a^1)^T = \partial C/\partial z^2 * \partial z^2/\partial w^2$
- $\partial C/\partial w^1 = \delta^1 * (a^0)^T$

By analogy we arrive at

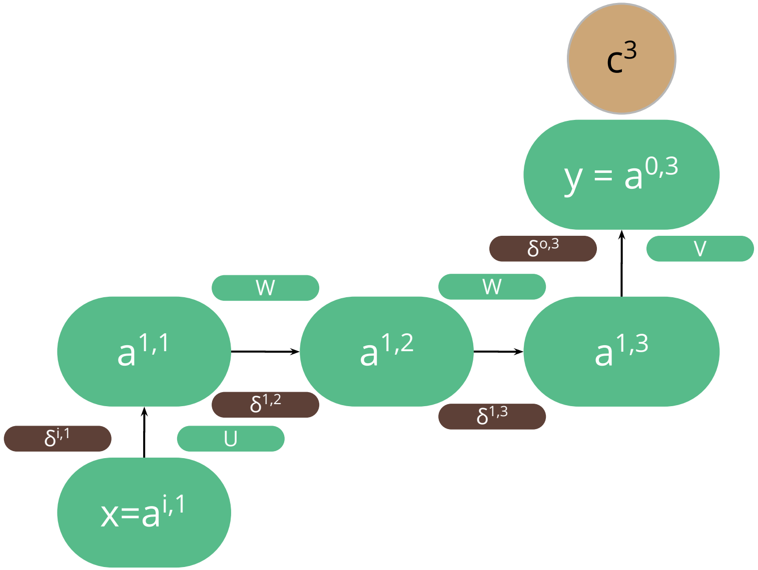
- $\partial C/\partial V = \delta^{o,3} * (a^{1,3})^T$
- $\partial C/\partial U = \delta^{i,1} * (a^{i,1})^T$
- $\partial C/\partial W = ?$

Backward Pass

As W influences both $z^{1,2}$ and $z^{1,3}$,
$\partial C/\partial W = \partial C/\partial z^{1,3} * \partial z^{1,3}/\partial W + \partial C/\partial z^{1,2} * \partial z^{1,2}/\partial W$

Now using the trick that all terms in $\partial C/\partial W$ are of the form $\delta_{\text{current layer}} * (a_{\text{prev layer}})^T$

Hence we get
$\partial C/\partial W = \delta^{1,3} * (a^{1,2})^T + \delta^{1,2} * (a^{1,1})^T$



Summary

Forward Pass

For T time steps:

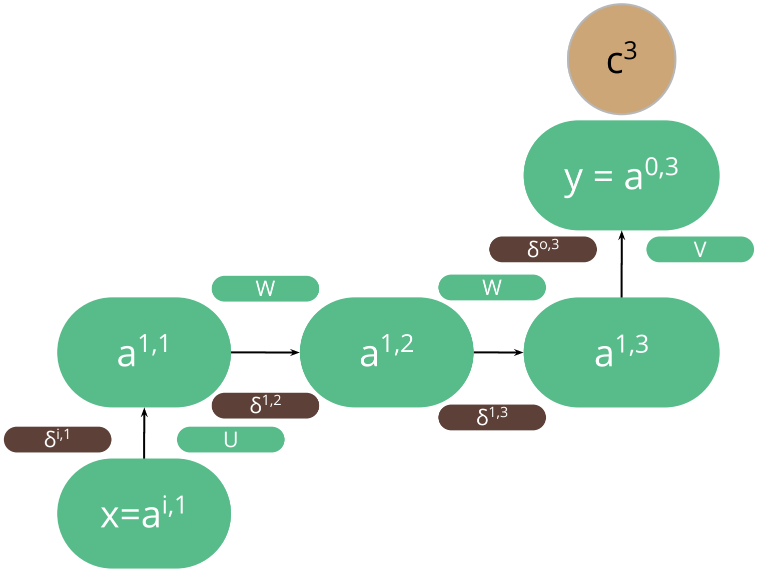
- $a^{i,1} = x$
- $z^{1,1} = U * a^{i,1}$
- $a^{1,1} = f(z^{1,1})$

For t = 2, ..., T

- $z^{1,t} = W * a^{1,t-1}$
- $a^{1,t} = f(z^{1,t})$

For output

- $z^{o,T} = V * a^{1,T}$
- $a^{o,T} = f(z^{o,T})$
- $y = a^{0,T}$



Backward Pass

From output

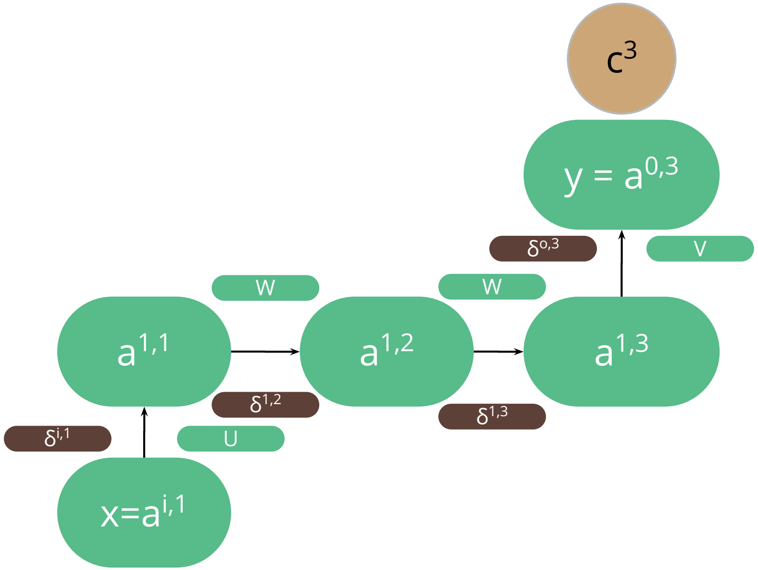
- $\delta^{o,T} = \nabla_y C^T \odot f'(z^{o,T})$
- $\delta^{1,T} = V^T * \delta^{o,T} \odot f'(z^{1,T})$

And for t = T-1, …, 1

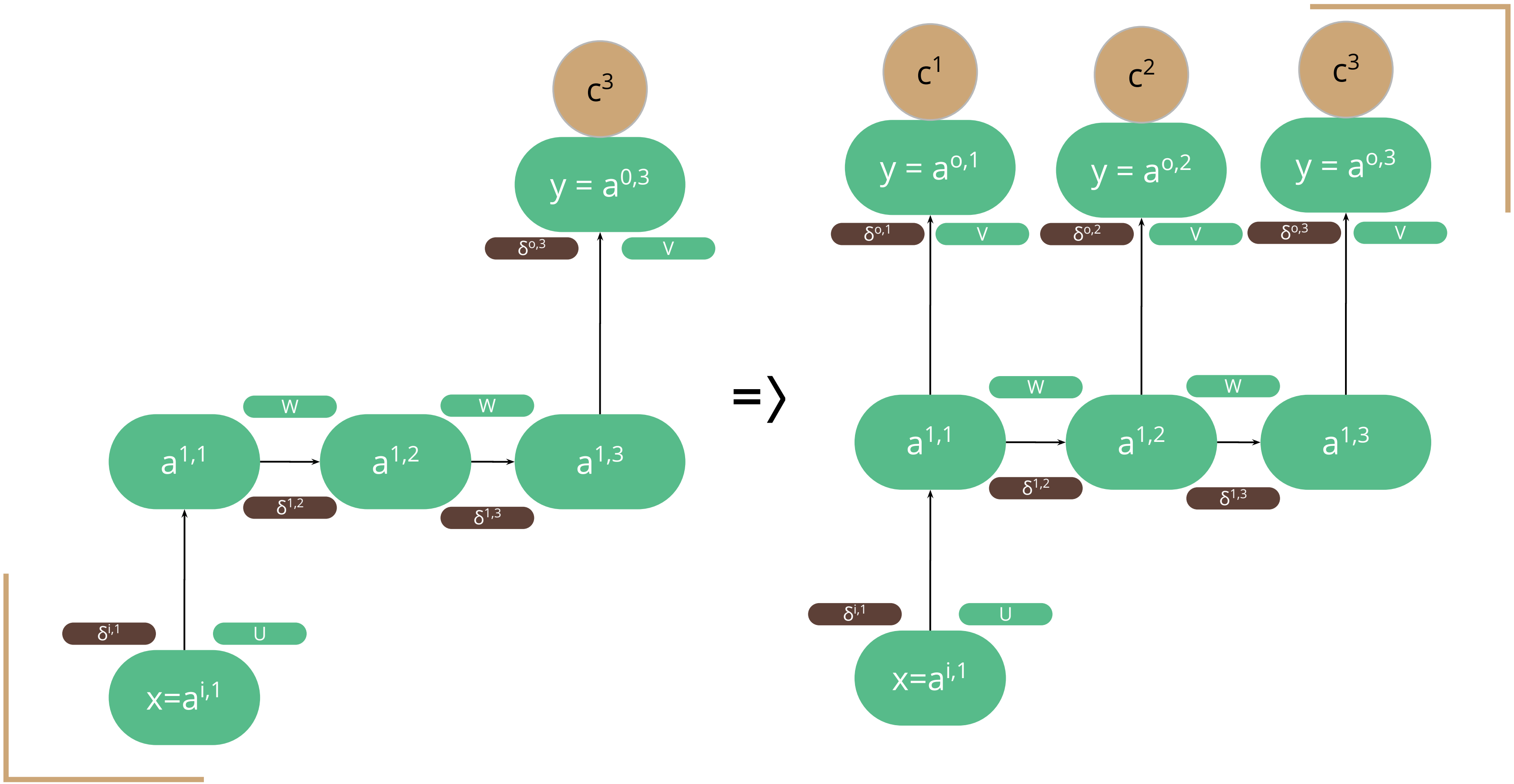
- $\delta^{1,t} = W^T * \delta^{1,t+1} \odot f'(z^{1,t})$

Finally the gradients are

- $\partial C/\partial V = \delta^{o,T} * (a^{1,T})^T$
- $\partial C/\partial U = \delta^{i,1} * (a^{i,1})^T$
- $\partial C/\partial W = \sum_{t=2}^{T} \delta^{1,t} * (a^{1,t-1})^T$



A SIMO Recurrent Neural Network



c^3
$y = a^{0,3}$
$\delta^{o,3}$
V
W
W
$a^{1,1}$
$a^{1,2}$
$a^{1,3}$
$\delta^{1,2}$
$\delta^{1,3}$
$\delta^{i,1}$
U
$x=a^{i,1}$
c^1
c^2
c^3
$y = a^{o,1}$
$y = a^{o,2}$
$y = a^{o,3}$
$\delta^{o,1}$
V
$\delta^{o,2}$
V
$\delta^{o,3}$
V
W
W
$a^{1,1}$
$a^{1,2}$
$a^{1,3}$
$\delta^{1,2}$
$\delta^{1,3}$
$\delta^{i,1}$
U
$x=a^{i,1}$

Forward Pass

Input

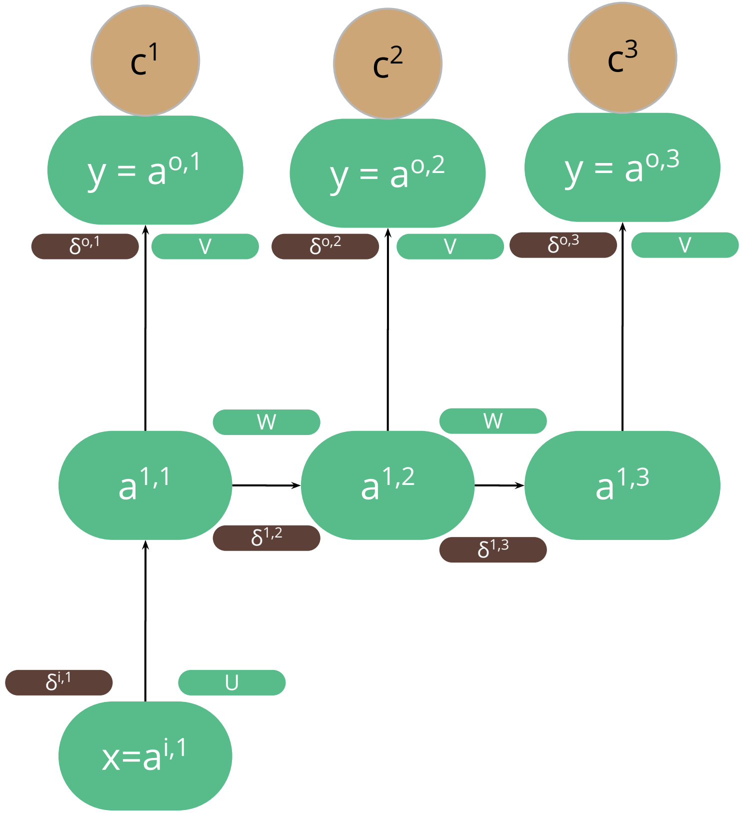
- $a^{i,1} = x$
- $z^{1,1} = U * a^{i,1}$
- $a^{1,1} = f(\ z^{1,1}\ )$

Hidden (for t = 2, … , T)

- $z^{1,t} = W * a^{1,t-1}$
- $a^{1,t} = f(\ z^{1,t}\ )$

Output (for t = 1, …, T)

- $z^{o,t} = V * a^{o,\ t}$
- $a^{o,t} = f(\ z^{0,t}\ )$
- $y^{t} = a^{o,\ t}$

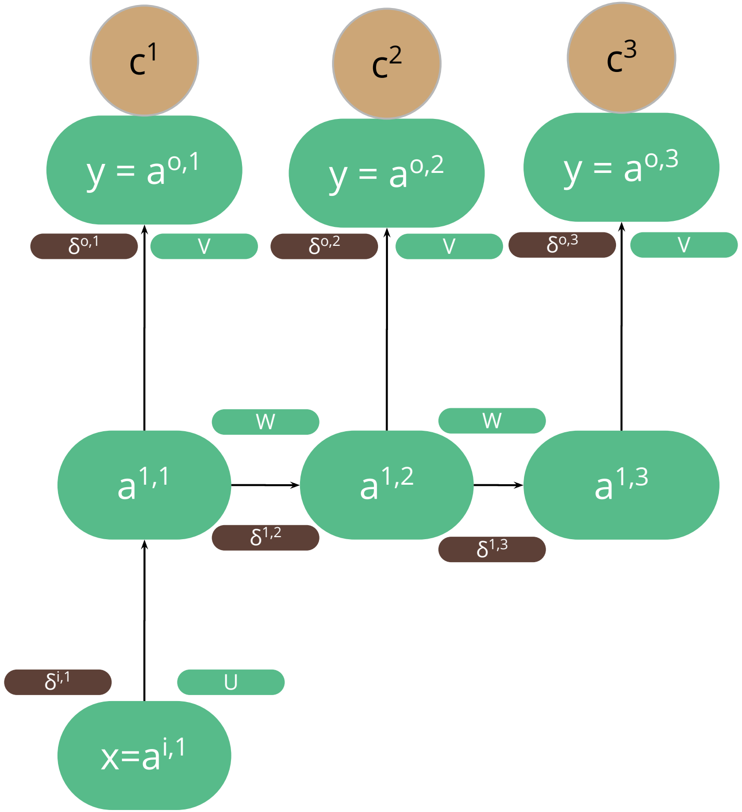


The Cost Function

- $c^1 = ½ * (y^1 - t^1)^2$
- $c^2 = ½ * (y^2 - t^2)^2$
- $c^3 = ½ * (y^3 - t^3)^2$

$c = c^1 + c^2 + c^3$

This can be extended for any other cost function and for T timesteps



Backward Pass

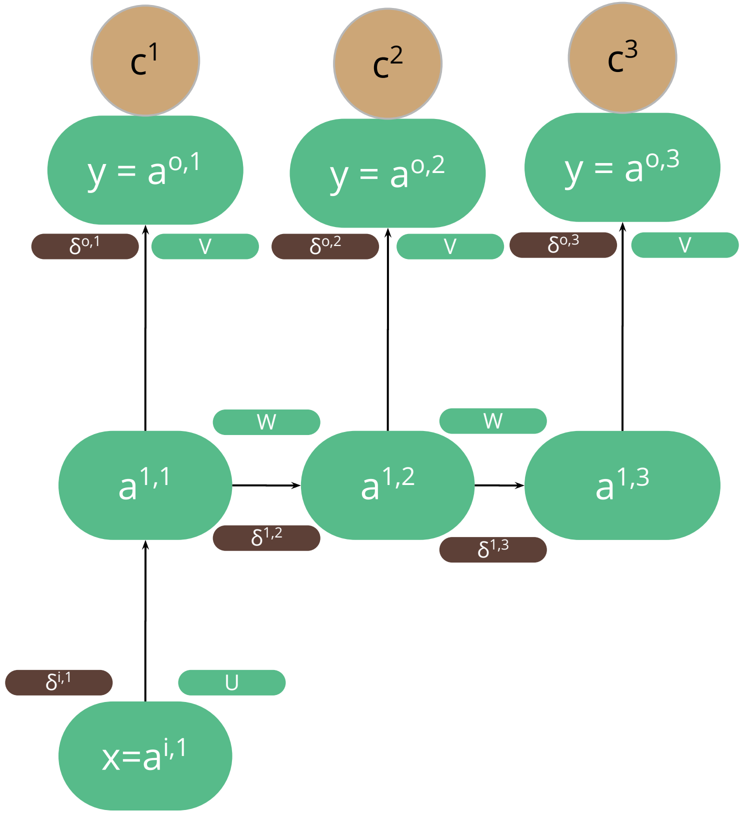
$\delta^{o,3} = \partial c/\partial z^{o,3} = \partial c^3/\partial z^{o,3}$
[As c^1 and c^2 don't depend on $z^{o,3}$]

=> $\delta^{o,3} = \partial c^3/\partial z^{o,3} = \partial c^3/\partial a^{o,3} * \partial a^{o,3}/\partial z^{o,3}$

=> $\delta^{o,3} = \nabla_{y3} c^3 \odot f'(z^{o,3})$

We thus arrive at

- $\delta^{o,1} = \nabla_{y1} c^1 \odot f'(z^{o,1})$
- $\delta^{o,2} = \nabla_{y2} c^2 \odot f'(z^{o,2})$
- $\delta^{o,3} = \nabla_{y3} c^3 \odot f'(z^{o,3})$

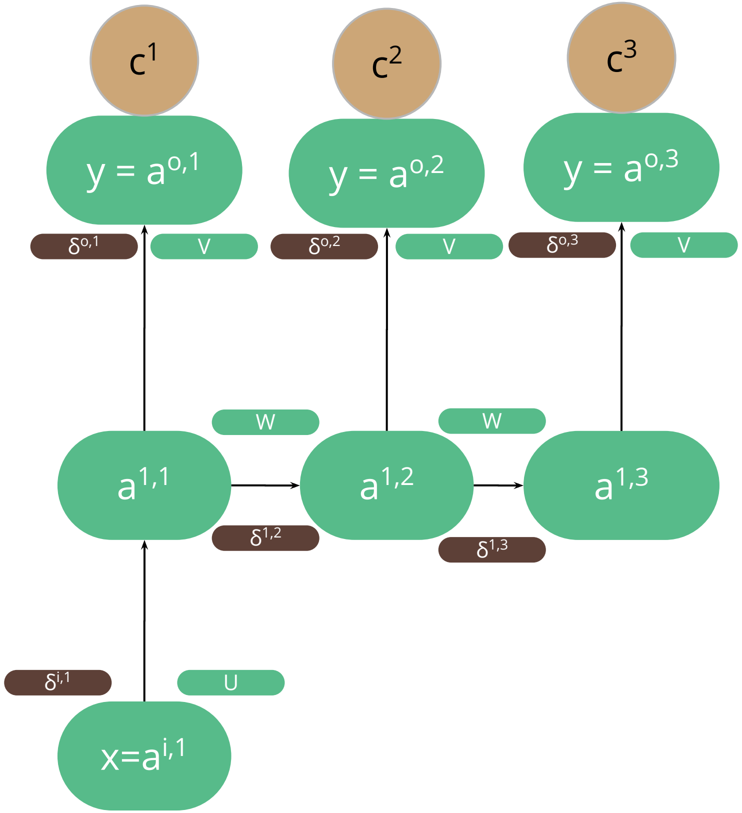


Backward Pass

$\delta^{1,3} = \partial c/\partial z^{1,3} = \partial c^3/\partial z^{1,3}$
[As c^1 and c^2 don't depend on $z^{1,3}$]

=> $\delta^{1,3} = \partial c^3/\partial z^{o,3} * \partial z^{o,3}/\partial z^{1,3}$

=> $\delta^{1,3} = V^T \delta^{o,3} \odot f'(z^{1,3})$
[As all δ are of the form (Outgoing Weight)^T (Outgoing δ) ⊙ f'(Weighted Inputs)]



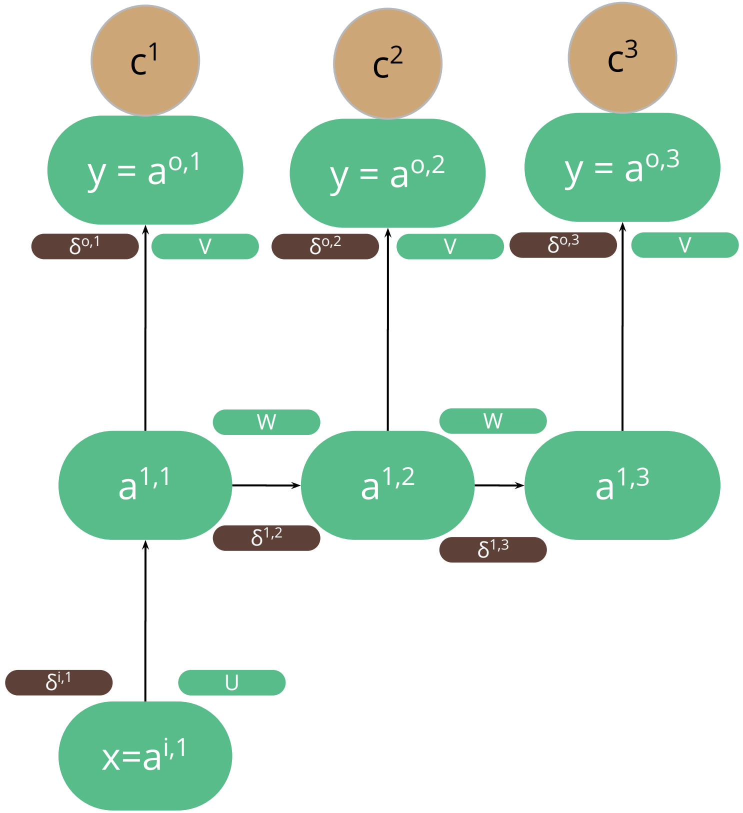
Backward Pass

$\delta^{1,2} = \partial c/\partial z^{1,2} = \partial c^2/\partial z^{1,2} + \partial c^3/\partial z^{1,2}$
[As c^1 does not depend on $z^{1,2}$]

$\Rightarrow \delta^{1,2} = \partial c^2/\partial z^{o,2} * \partial z^{o,2}/\partial z^{1,2} + \partial c^3/\partial z^{1,3} * \partial z^{1,3}/\partial z^{1,2}$

$\Rightarrow \delta^{1,2} = V^T \delta^{o,2} \odot f'(z^{1,2}) + W^T \delta^{1,3} \odot f'(z^{1,2})$
[As all δ are of the form (Outgoing Weight)^T (Outgoing δ) $\odot$ f'(Weighted Inputs)]

$\Rightarrow \delta^{1,2} = (V^T \delta^{o,2} + W^T \delta^{1,3}) \odot f'(z^{1,2})$



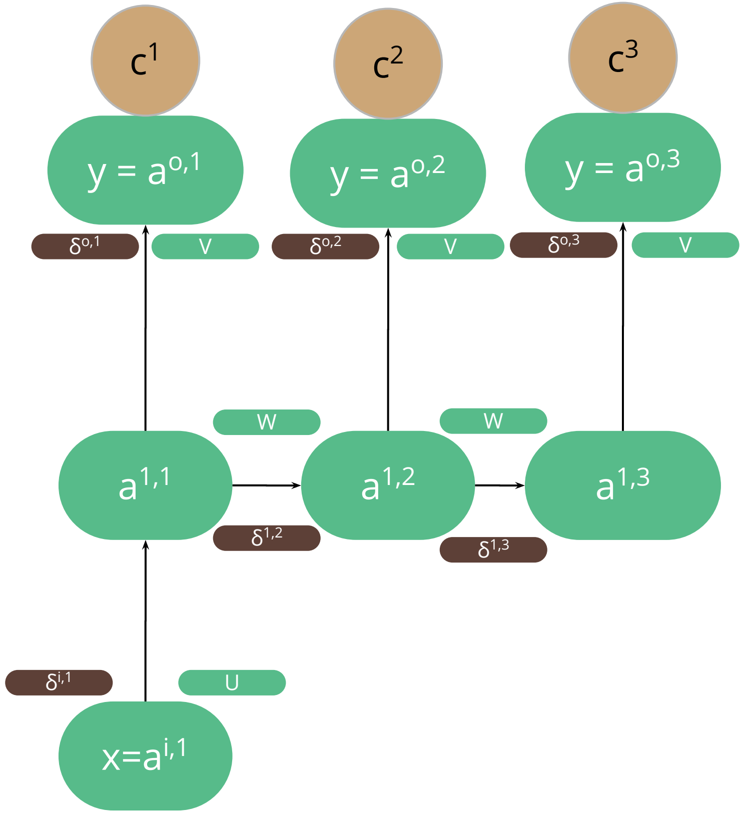
Backward Pass

We thus arrive at

- $\delta^{1,3} = V^T \delta^{o,3} \odot f'(z^{1,3})$
- $\delta^{1,2} = (V^T \delta^{o,2} + W^T \delta^{1,3}) \odot f'(z^{1,2})$
- $\delta^{i,1} = (V^T \delta^{o,1} + W^T \delta^{1,2}) \odot f'(z^{1,1})$

Now we can compute derivatives with respect to weights
$\partial c/\partial V = \partial c^1/\partial V + \partial c^2/\partial V + \partial c^3/\partial V$

=> $\partial c/\partial V = \delta^{o,3} * (a^{1,3})^T + \delta^{o,2} * (a^{1,2})^T + \delta^{o,1} * (a^{1,1})^T$
[As each of these derivative terms is equal to $\delta_{out} * a_{in}^T$]

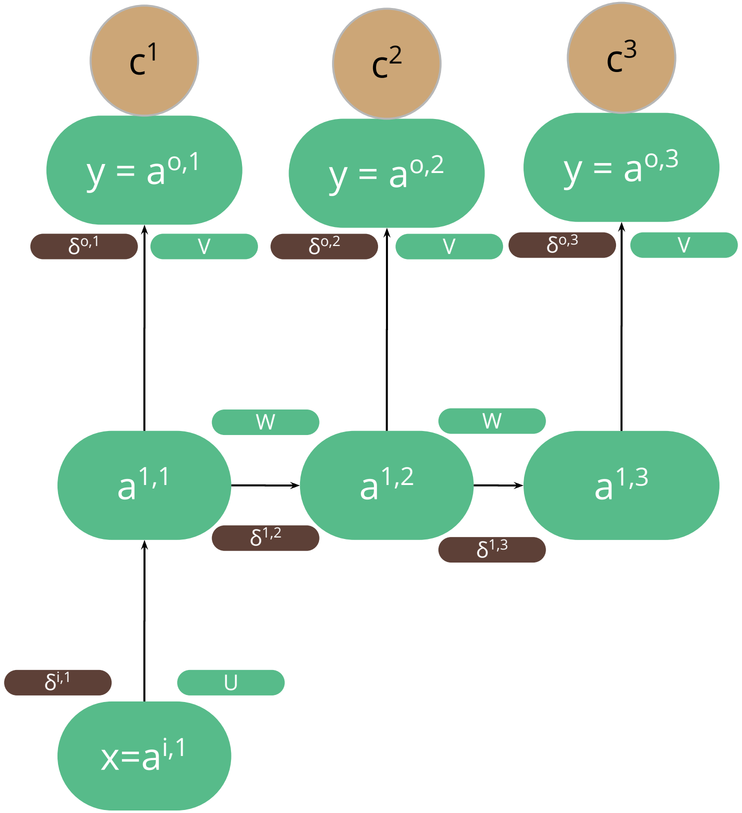


Backward Pass

We thus arrive at

- $\partial c/\partial V = \delta^{o,3} * (a^{1,3})^T + \delta^{o,2} * (a^{1,2})^T + \delta^{o,1} * (a^{1,1})^T$
- $\partial c/\partial W = \delta^{1,3} * (a^{1,2})^T + \delta^{1,2} * (a^{1,1})^T$
- $\partial c/\partial U = \delta^{i,1} * (a^{i,1})^T$

[As each of these derivative terms is equal to $\delta_{out} * a_{in}^T$]



Summary

Forward Pass

Input

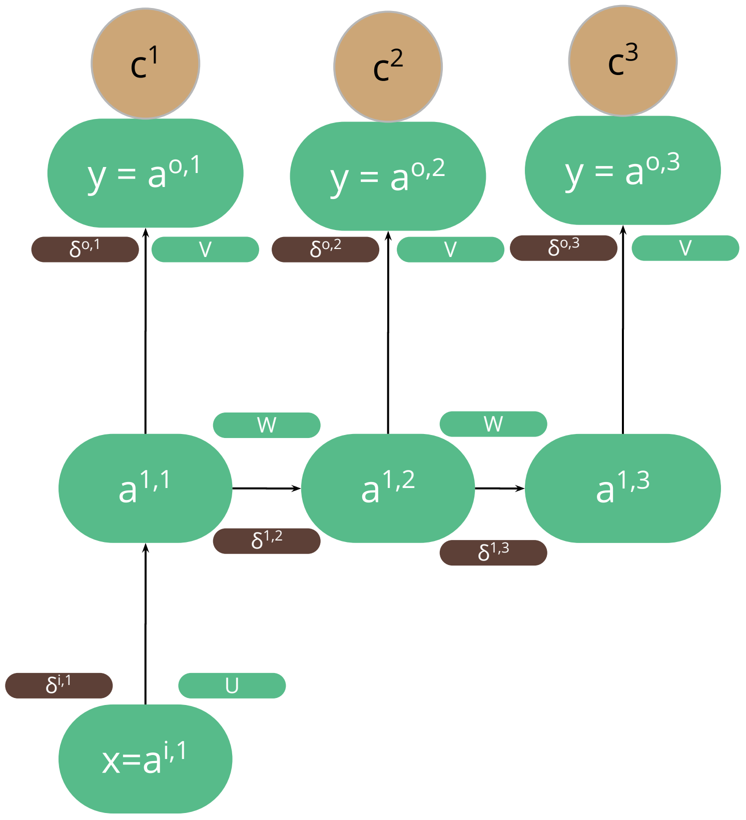
- $a^{i,1} = x$
- $z^{1,1} = U * a^{i,1}$
- $a^{1,1} = f(\ z^{1,1}\ )$

Hidden (for t = 2, ... , T)

- $z^{1,t} = W * a^{1,t-1}$
- $a^{1,t} = f(\ z^{1,t}\ )$

Output (for t = 1, ..., T)

- $z^{o,t} = V * a^{o,\, t}$
- $a^{o,t} = f(\ z^{0,t}\ )$
- $y^{t} = a^{o,\, t}$



Backward Pass

Output [For all t = 1, …, T]

- $\delta^{o,t} = \nabla_{yt} c^t \odot f'(z^{o,t})$

Hidden [For all t = 1, …, T-1]

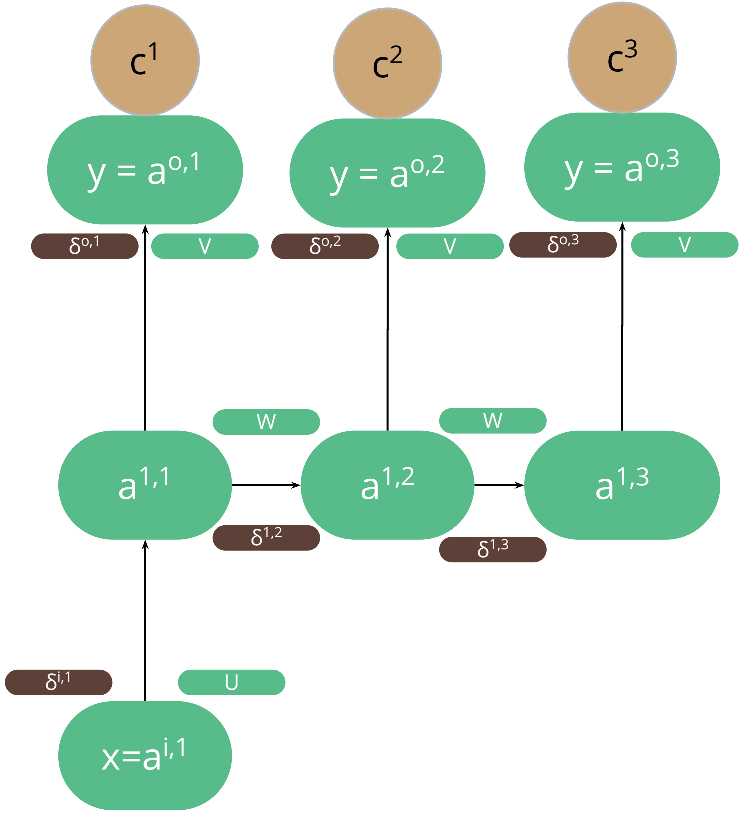
- $\delta^{1,T} = V^T \delta^{o,T} \odot f'(z^{1,T})$
- $\delta^{1,t} = (V^T \delta^{o,t} + W^T \delta^{1,t}) \odot f'(z^{1,t})$

Input

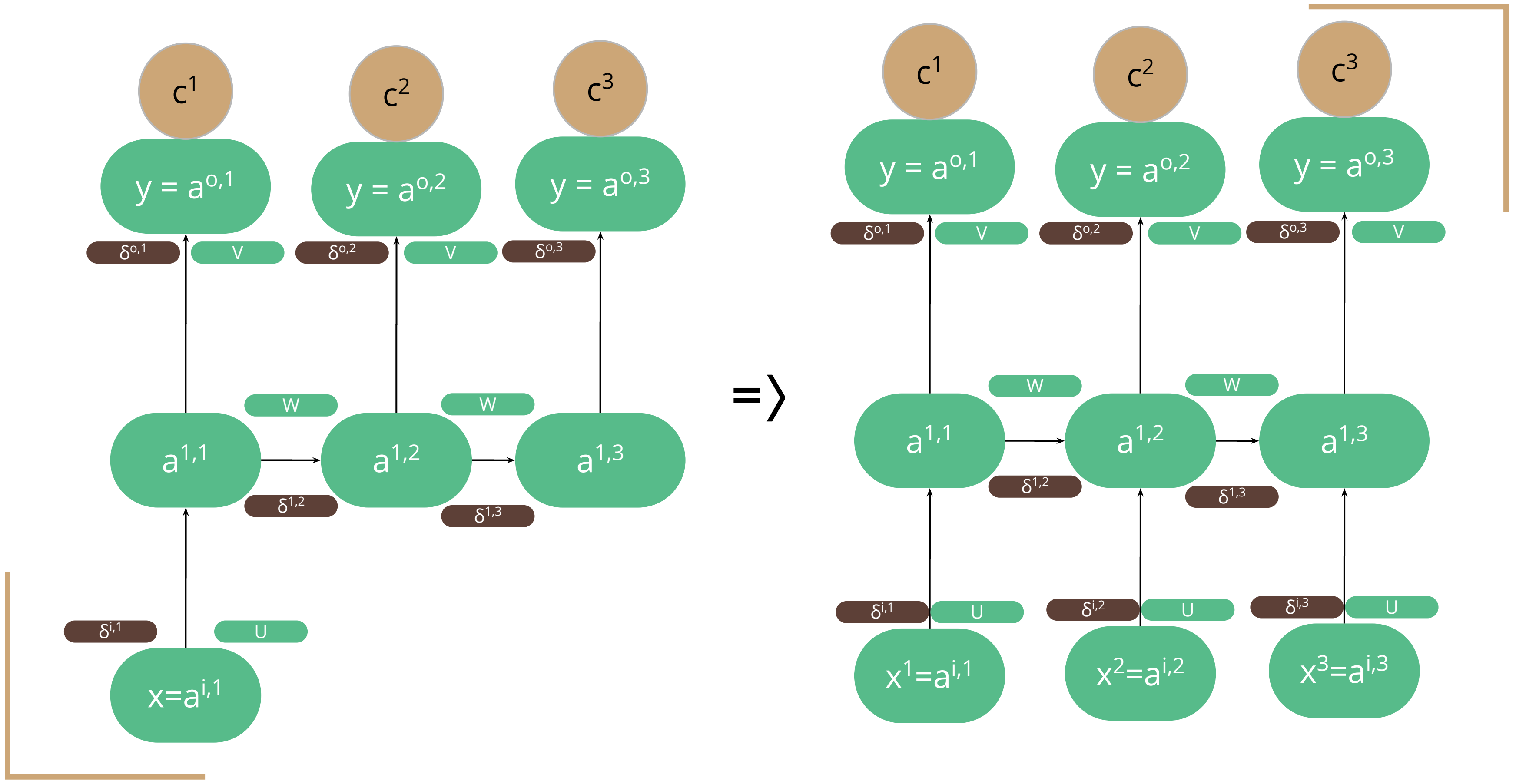
- $\delta^{i,1} = (V^T \delta^{o,1} + W^T \delta^{1,2}) \odot f'(z^{1,1})$

Weights

- $\partial c/\partial V = \sum_{t=1}^{T} \delta^{o,t} * (a^{1,t})^T$
- $\partial c/\partial W = \sum_{t=2}^{T} \delta^{1,t} * (a^{1,t-1})^T$
- $\partial c/\partial U = \delta^{i,1} * (a^{i,1})^T$



A MIMO Recurrent Neural Network



c1
c2
c3
y = ao,1
y = ao,2
y = ao,3
δo,1
δo,2
δo,3
V
W
a1,1
a1,2
a1,3
δ1,2
δ1,3
δi,1
U
x=ai,1
=〉
x1=ai,1
x2=ai,2
x3=ai,3
δi,2
δi,3

Forward Pass

First Input

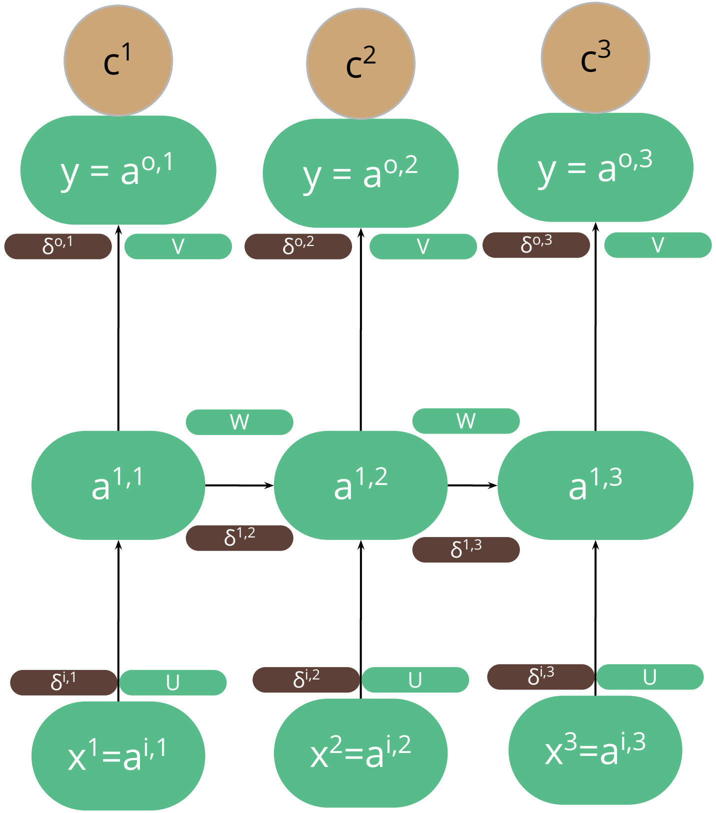
- $a^{i,1} = x^1$
- $z^{1,1} = U * a^{i,1}$
- $a^{1,1} = f(z^{1,1})$

Remaining Inputs (for t = 2, … , T)

- $a^{i,t} = x^t$
- $z^{1,t} = U * a^{i,t} + W * a^{1,t-1}$
- $a^{1,t} = f(z^{1,t})$

Output (for t = 1, …, T)

- $z^{o,t} = V * a^{1, t}$
- $a^{o,t} = f(z^{0,t})$
- $y^t = a^{o, t}$
- $c^t = ½(y^t - t^t)^2$

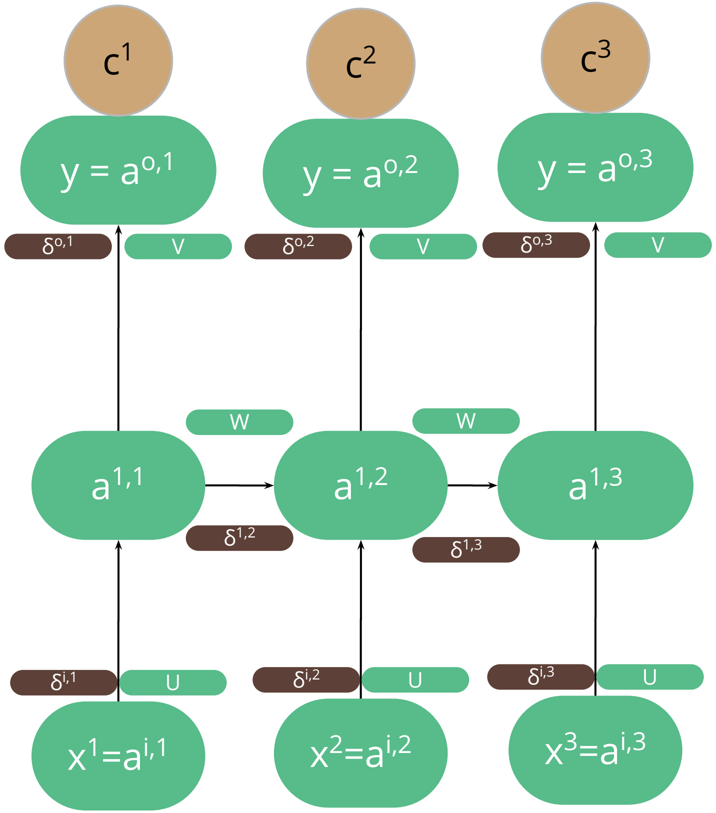


Backward Pass

If we extrapolate the derivatives from the SIMO Recurrent Neural Network, we would notice that the only changes lie in the introduction of $\delta^{i,2}$, $\delta^{i,3}$, ...

These would, however be equal to $\delta^{1,2}$, $\delta^{1,3}$, ... as the same derivative flows through all paths connected to a single layer [i.e. $\delta^{1,t} = \delta^{i,t}$]

These new terms would only change the derivative with respect to U, adding extra terms in a similar manner the SIMO model did for V



Backward Pass

Output [For all t = 1, ..., T]

- $\delta^{o,t} = \nabla_{y^t} c^t \odot f'(z^{o,t})$

Hidden [For all t = 1, ..., T-1]

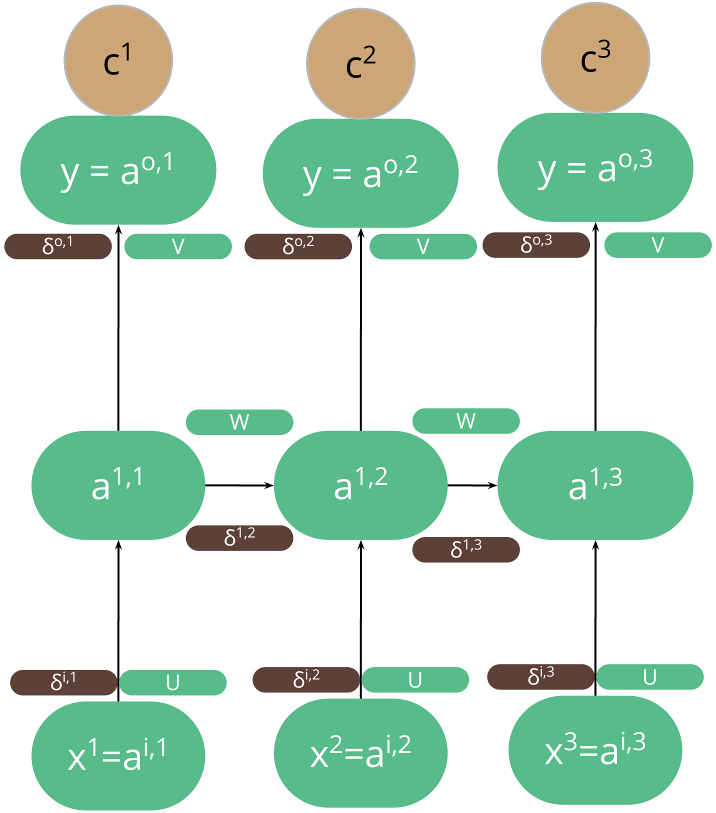
- $\delta^{1,T} = V^T \delta^{o,T} \odot f'(z^{1,T})$
- $\delta^{1,t} = (V^T \delta^{o,t} + W^T \delta^{1,t}) \odot f'(z^{1,t})$

Input [For all t = 1, ..., T]

- $\delta^{i,t} = \delta^{1,t}$

Weights

- $\partial c/\partial V = \sum_{t=1}^{T} \delta^{o,t} * (a^{1,t})^T$
- $\partial c/\partial W = \sum_{t=2}^{T} \delta^{1,t} * (a^{1,t-1})^T$
- $\partial c/\partial U = \sum_{t=1}^{T} \delta^{i,t} * (a^{i,t})^T$

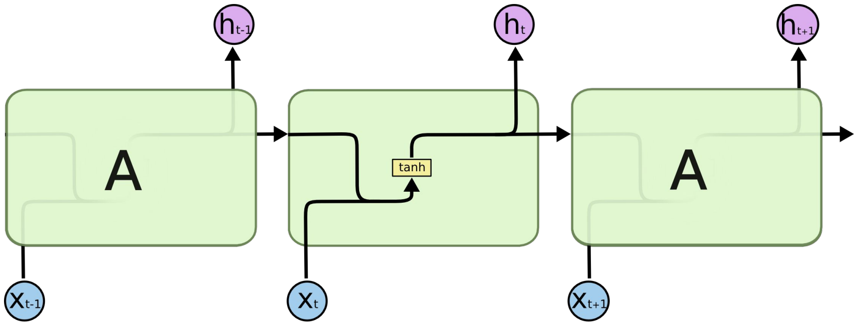


LSTM Models

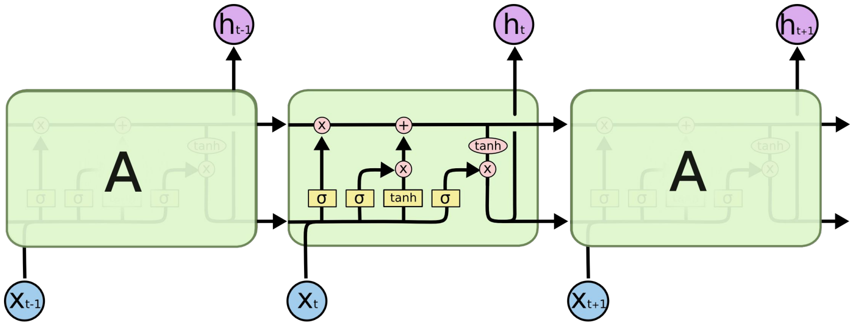
The Issue with RNNs

- Problem: RNNs Cannot Handle Long Term Dependencies
- Solution: Long Short Term Memory Networks (LSTMs) are a special kind of RNN, capable of learning long-term dependencies

RNN vs LSTM

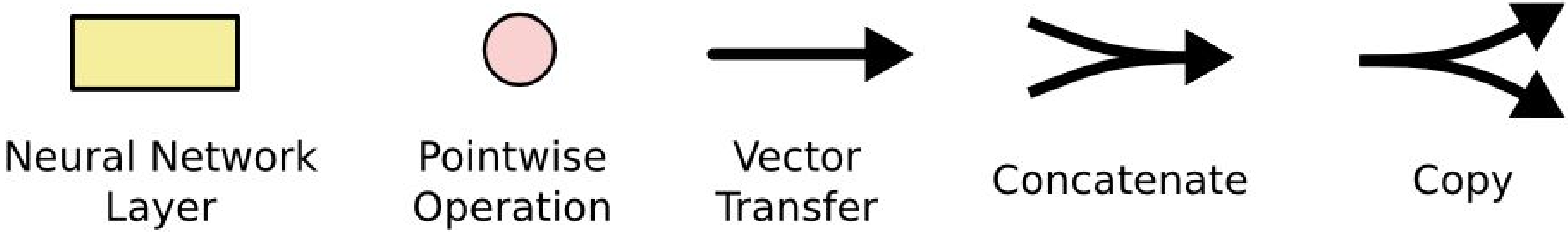


RNN



LSTM

Notation



Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

The Cell State

- The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions
- It’s very easy for information to just flow along it unchanged

LSTM: A Search Space Odyssey

Colah Blog

Karpathy Char RNN

LSTM Variants

Seq2Seq Models

Seq2Seq with Attention Models