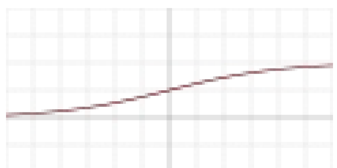
Backpropagation through Time

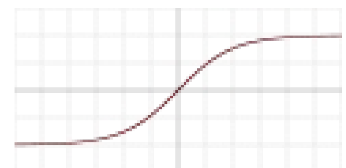
A Mathematical Overview

A Neural Network

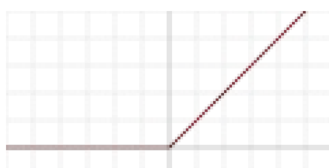
Activation Functions



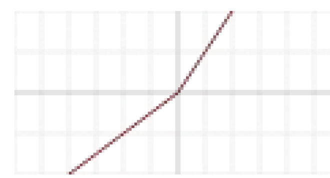
Sigmoid



Tanh



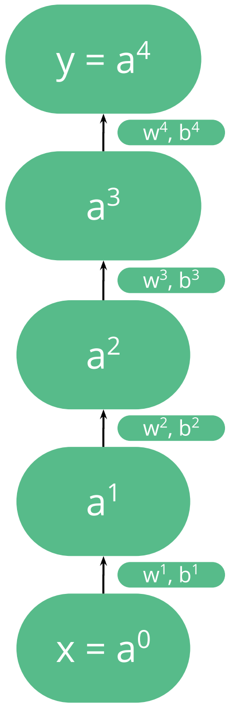
Relu



Leaky Relu

Notation

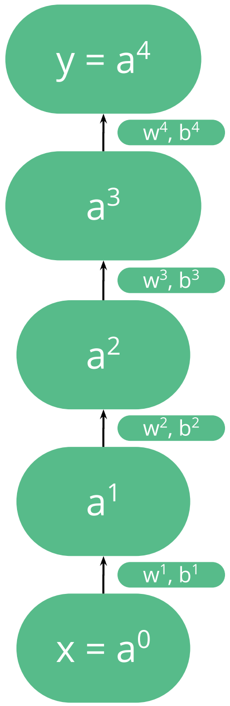
- Three Hidden Layer Neural Network
- x -> Input and y -> Output
- w-> Weight and b -> Bias



Notation

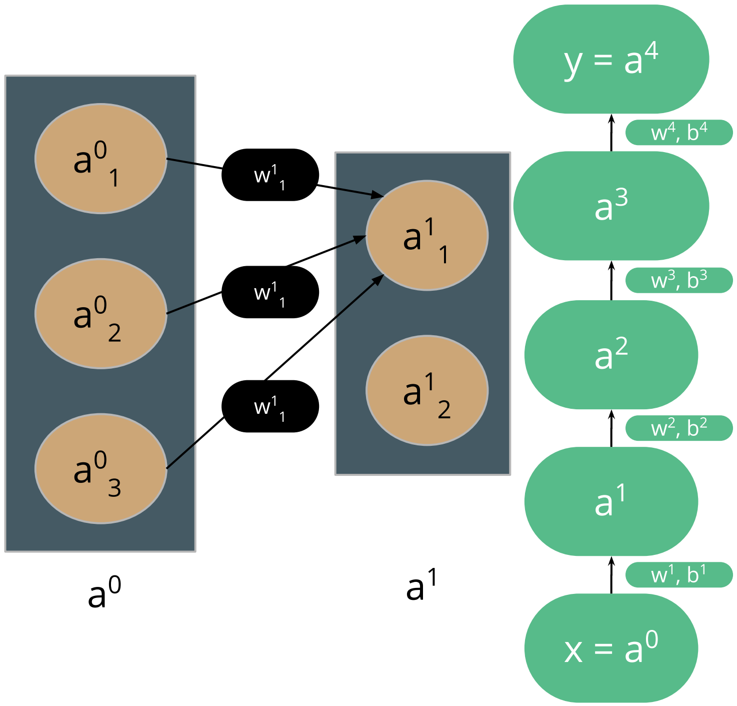
For a single data point

- 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



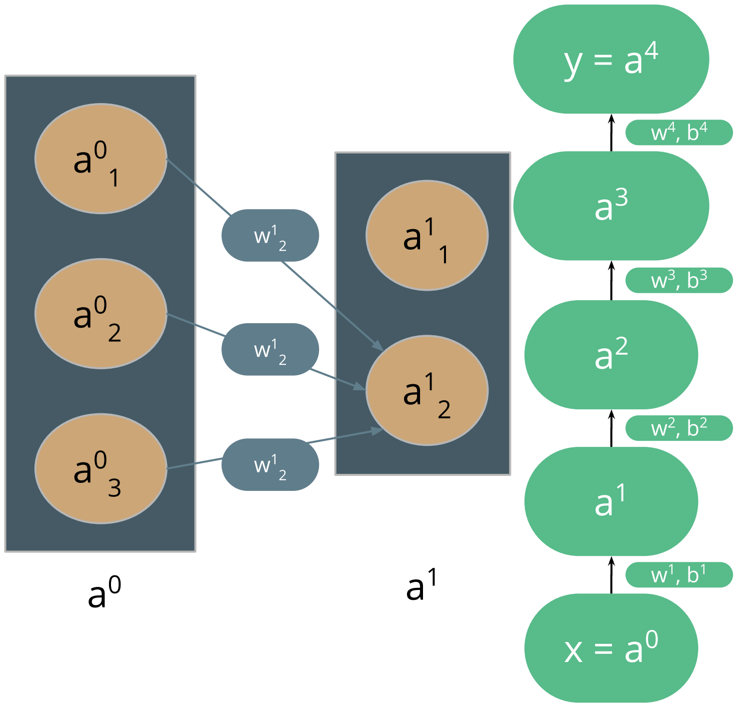
Forward Pass

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



Forward Pass

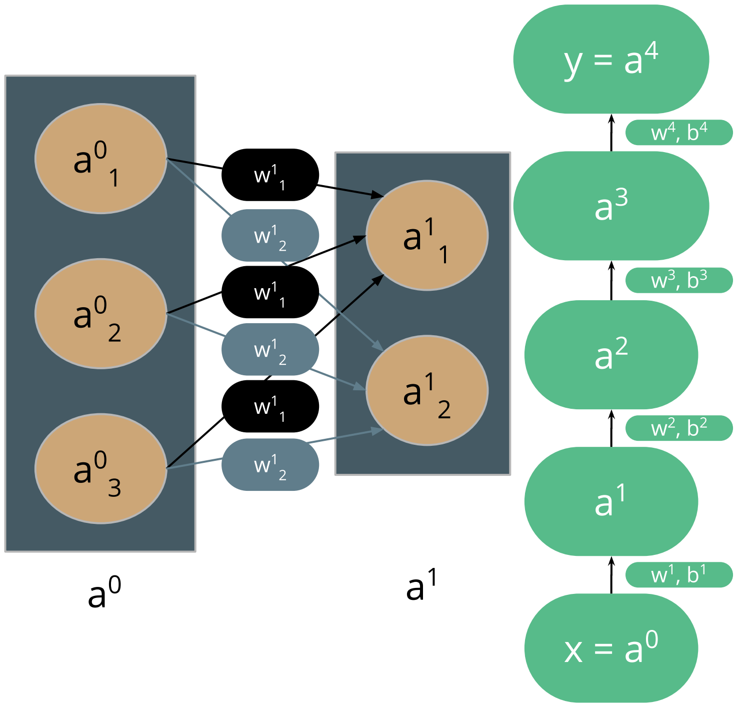
- $a^1{}_2 = f(\, w^1{}_{21}a^0{}_1 + w^1{}_{22}a^0{}_2 + w^1{}_{23}a^0{}_3)$



Forward Pass

Collecting the two

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



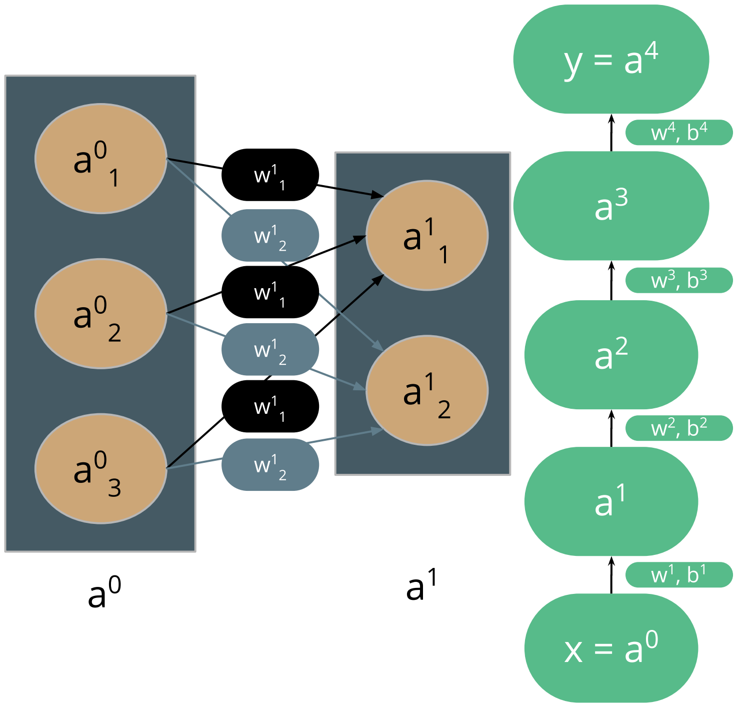
Forward Pass

Collecting the two

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

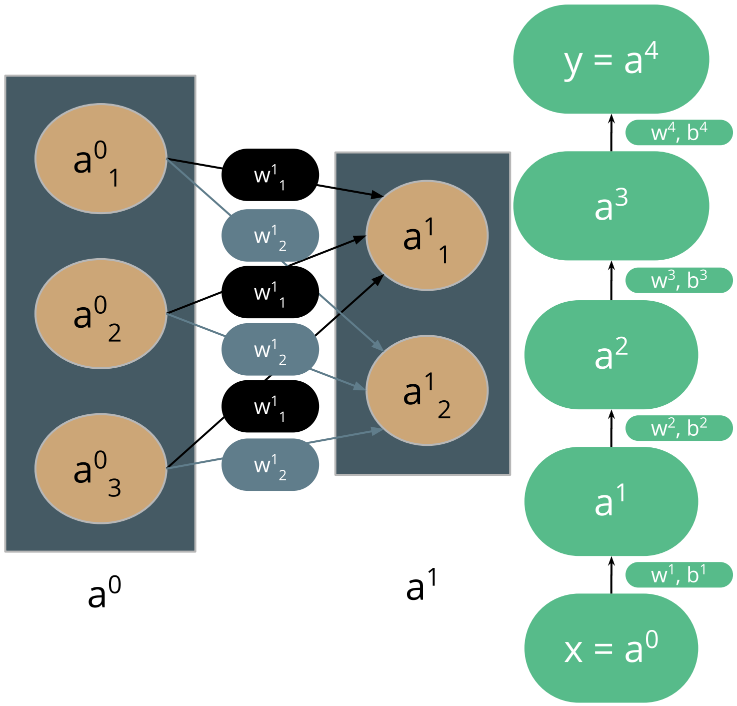
is the same as

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



Forward Pass

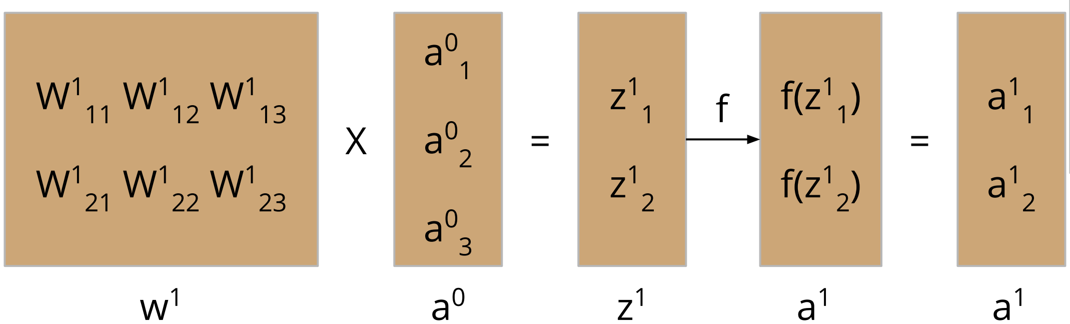
- $z^1{}_1 = w^1{}_{11}a^0{}_1 + w^1{}_{12}a^0{}_2 + w^1{}_{13}a^0{}_3$
- $a^1{}_1 = f(\, z^1{}_1 \,)$
- $z^1{}_2 = w^1{}_{21}a^0{}_1 + w^1{}_{22}a^0{}_2 + w^1{}_{23}a^0{}_3$
- $a^1{}_2 = f(\, z^1{}_2 \,)$

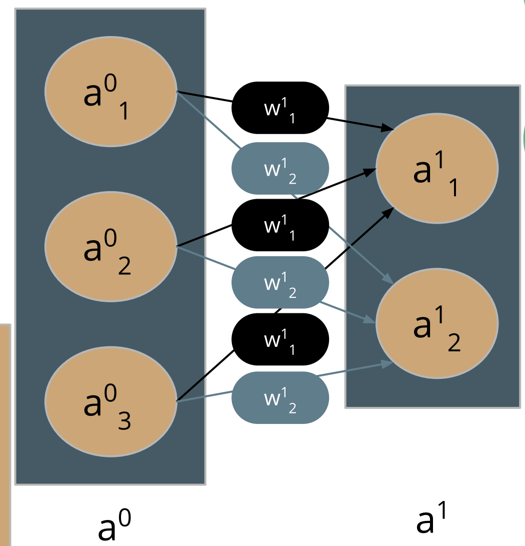


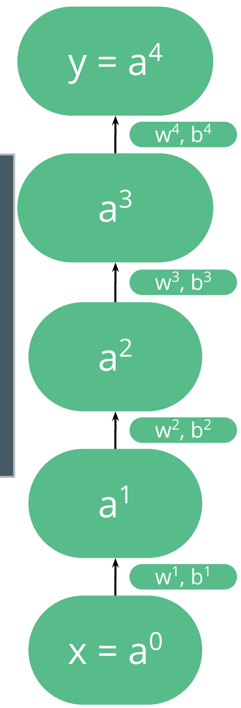
Forward Pass

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

is the same as





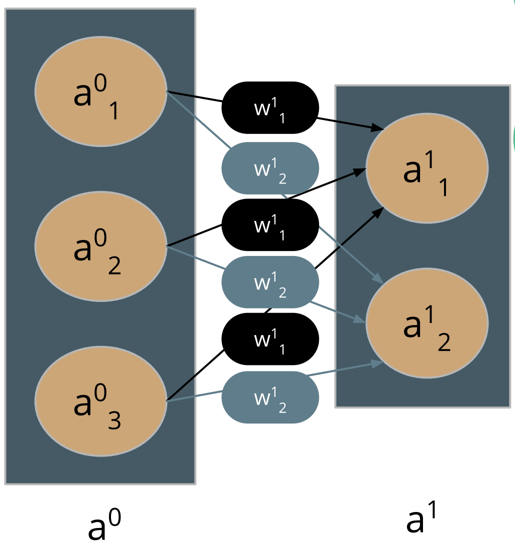


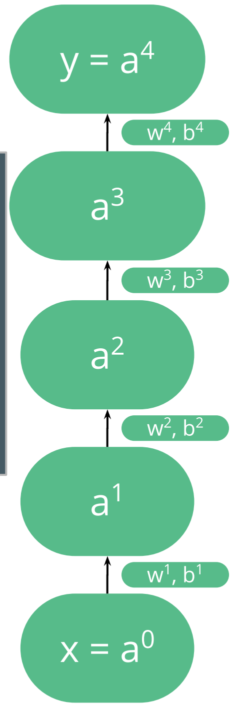
Forward Pass

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

is the same as

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





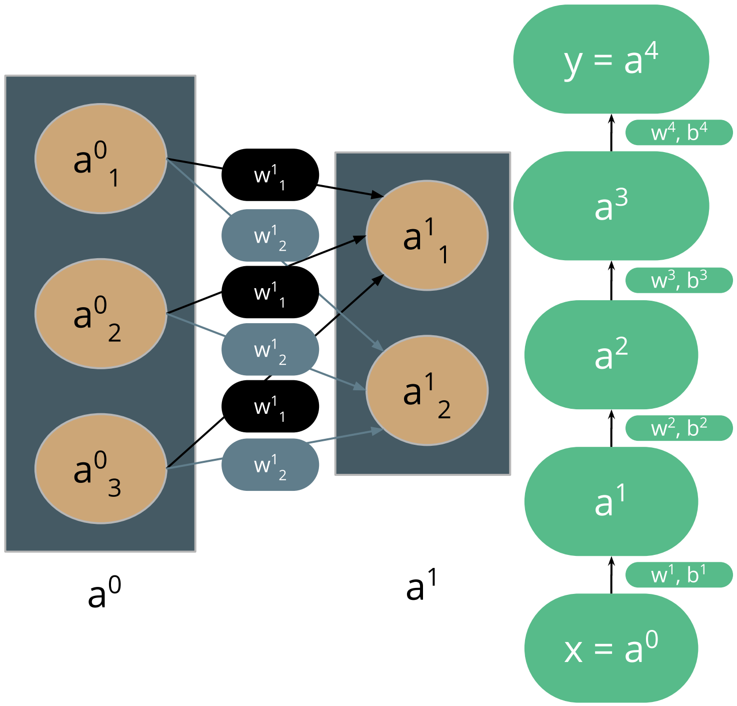
Forward Pass

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

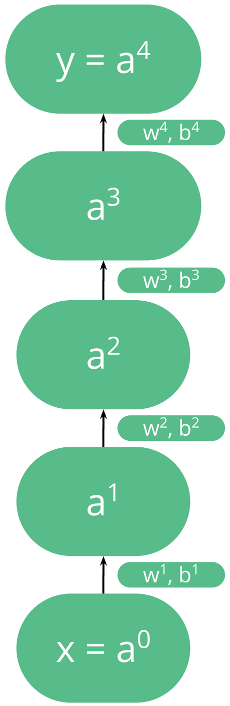
- $z^1 = w^1 * a^0 + b^1$
- $a^1 = f(\ z^1\ )$



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

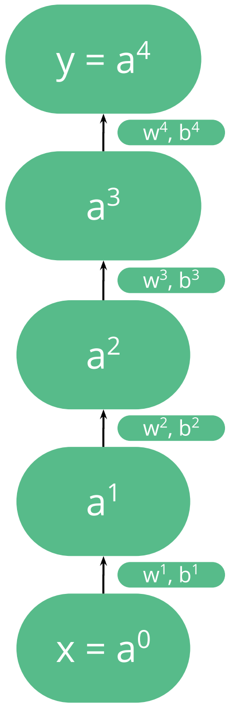
- $a^0 = x$

For l = 1, … , L layers

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

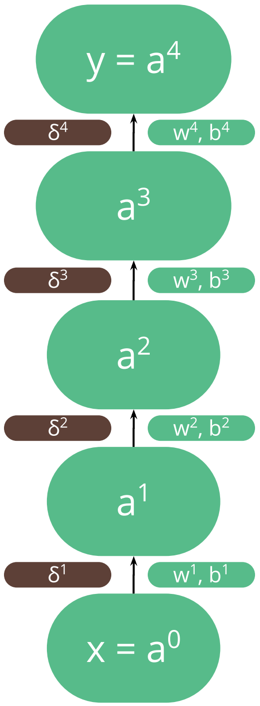
Finally

- $y = a^L$



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: $C = t * \ln(y) + (1-t) * \ln(1-y)$