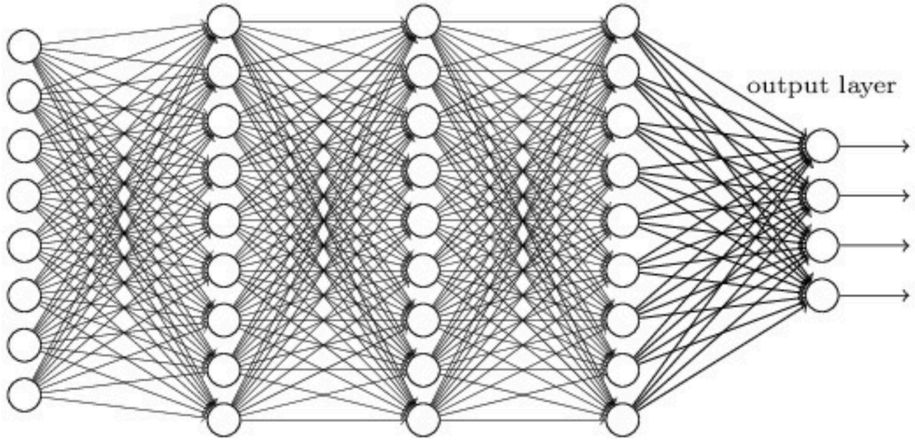
Neural Networks

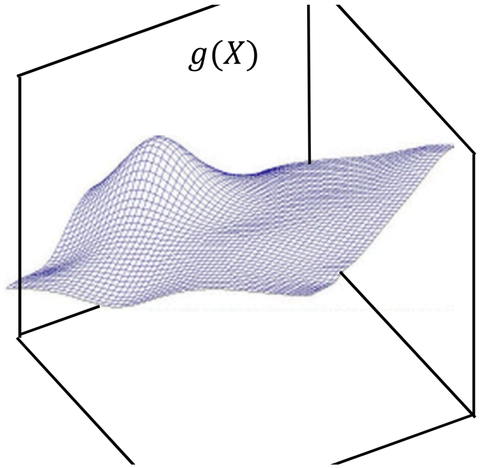
Forward/backward propagation + training

Rafał Nowak March 14, 2023

What we learn?

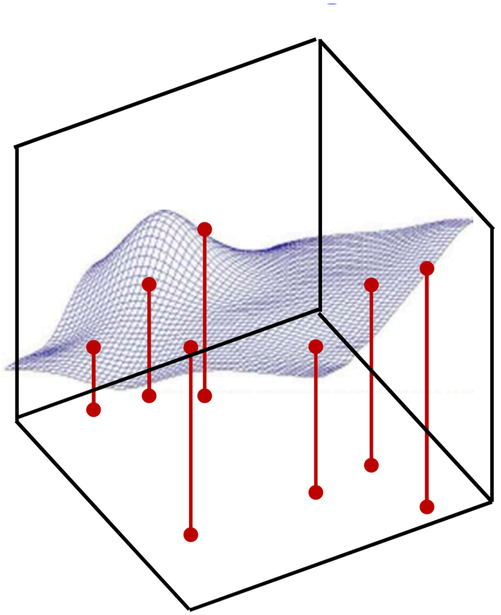
- network parameters W
- $f(X, W) \approx g(X)$

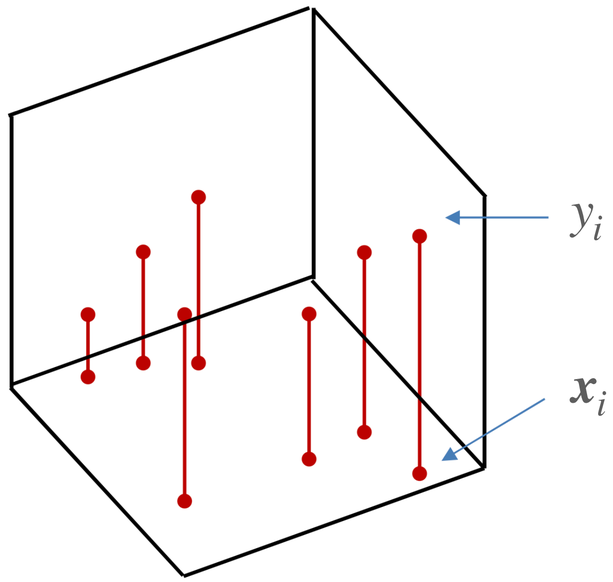




Learning the function (recap)

- Estimate the training parameters to *fit* the training data points
- Questions:
  - network architecture? basis functions?
  - approximation error?



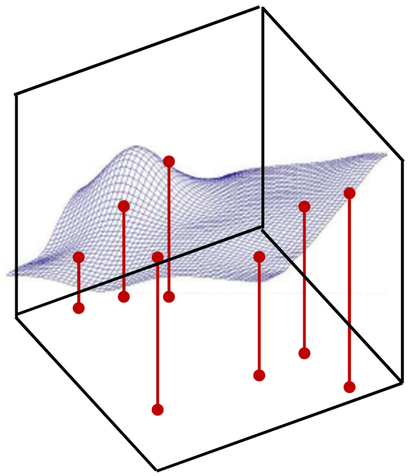


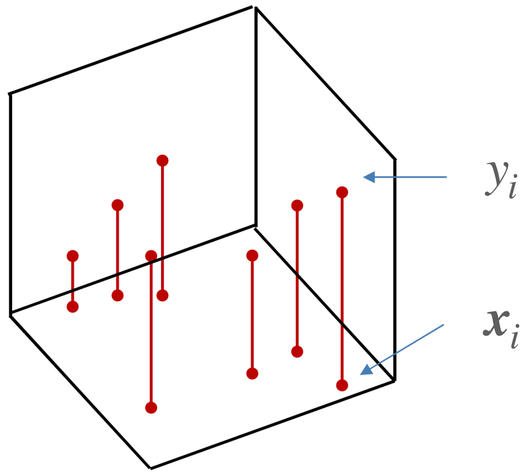
Learning the function

Error function (loss function)

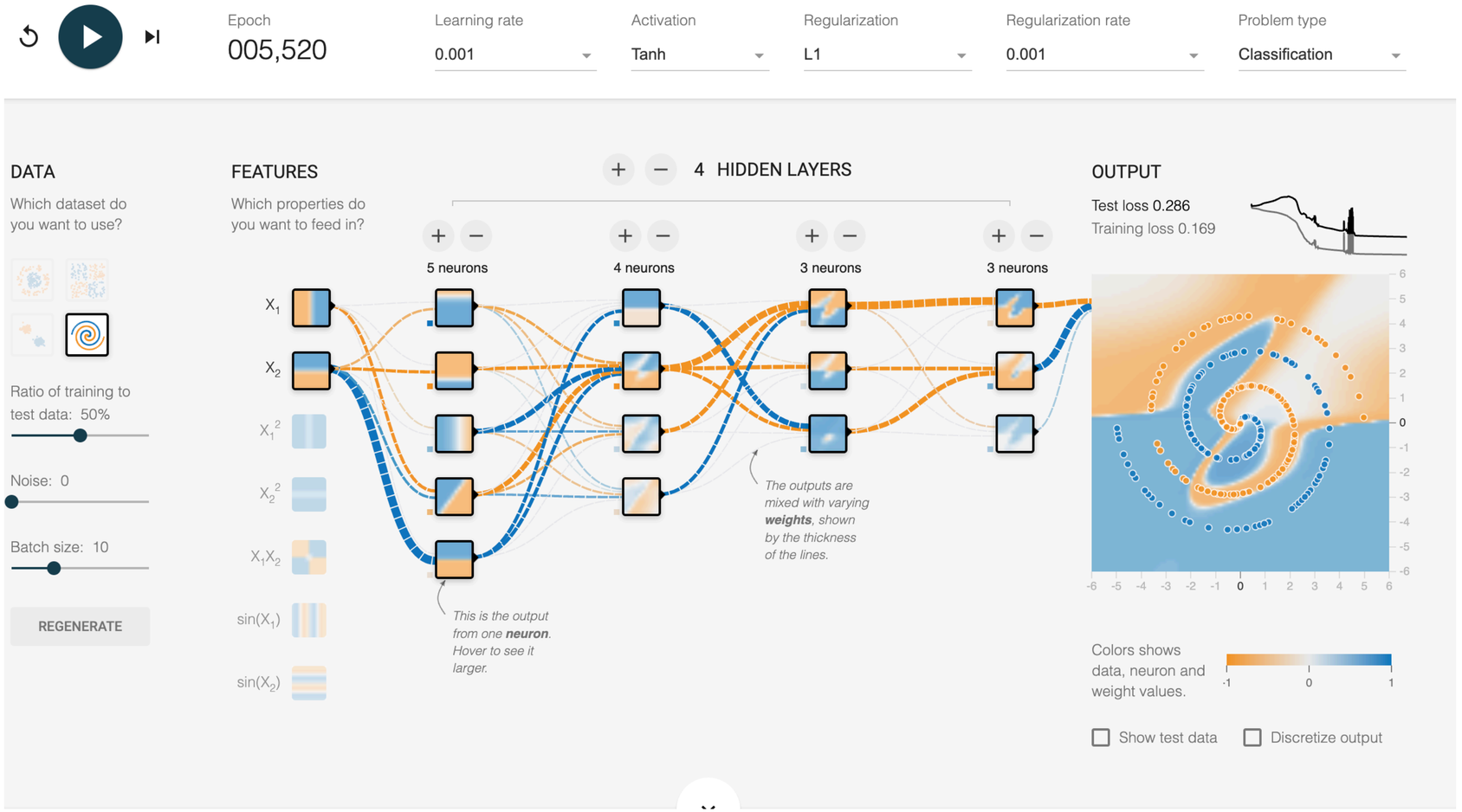
- Given the training set $X_{\text{train}} = [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \ldots, \boldsymbol{x}_N^T] \in \mathbb{R}^{N \times n}$

$$Err(W) = \frac{1}{N} \sum_{i=1}^{N} div(f(\boldsymbol{x}_i; W), y_i)$$





Demo



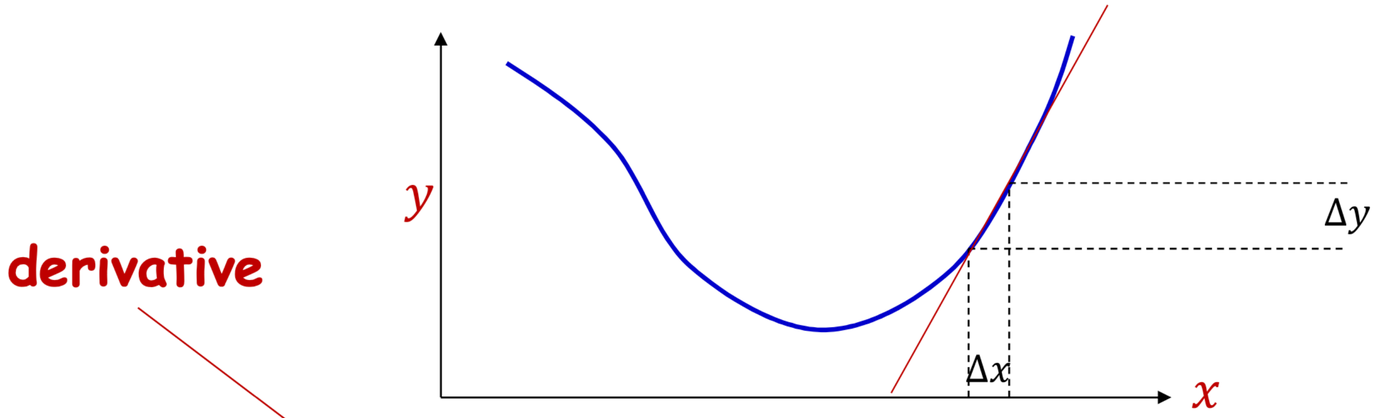
source: Playground Tensorflow

Training the neural network

- find the parameters $W^{(0)}, \boldsymbol{b}^{(0)}, W^{(1)}, \boldsymbol{b}^{(1)}, \ldots$ that minimize the total average loss
- total average loss:
-

$$\frac{1}{N}\sum_{i=1}^{N} div(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_i)$$

A brief note on derivatives..

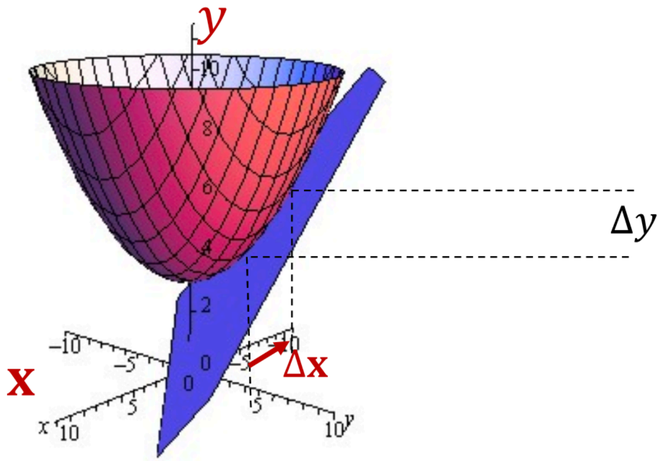


- A derivative of a function at any point tells us how much a minute increment to the *argument* of the function will increment the *value* of the function
  - For any $y = f(x)$, expressed as a multiplier α to a tiny increment Δx to obtain the increments Δy to the output

$$\Delta y = \alpha \Delta x$$

  - Based on the fact that at a fine enough resolution, any smooth, continuous function is locally linear at any point

Multivariate scalar function:
Scalar function of *vector* argument



Note: $\Delta\mathbf{x}$ is now a vector

$$\Delta\mathbf{x} = \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_D \end{bmatrix}$$

$$\Delta y = \alpha \Delta \mathbf{x}$$

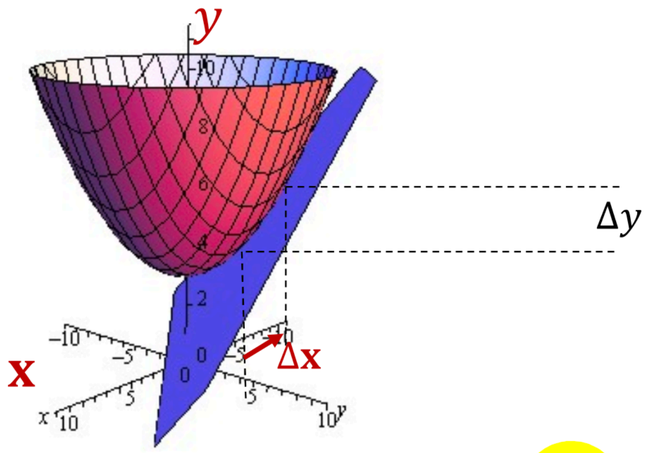
- Giving us that α is a row vector: $\alpha = [\alpha_1 \quad \cdots \quad \alpha_D]$

$$\Delta y = \alpha_1 \Delta x_1 + \alpha_2 \Delta x_2 + \cdots + \alpha_D \Delta x_D$$

- The *partial* derivative α_i gives us how y increments when *only* x_i is incremented
- Often represented as $\frac{\partial y}{\partial x_i}$

$$\Delta y = \frac{\partial y}{\partial x_1}\Delta x_1 + \frac{\partial y}{\partial x_2}\Delta x_2 + \cdots + \frac{\partial y}{\partial x_D}\Delta x_D$$

Multivariate scalar function:
Scalar function of *vector* argument



Note: $\Delta\mathbf{x}$ is now a vector

$$\Delta\mathbf{x} = \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_D \end{bmatrix}$$

$$\Delta y = \nabla_{\mathbf{x}} y \Delta\mathbf{x}$$

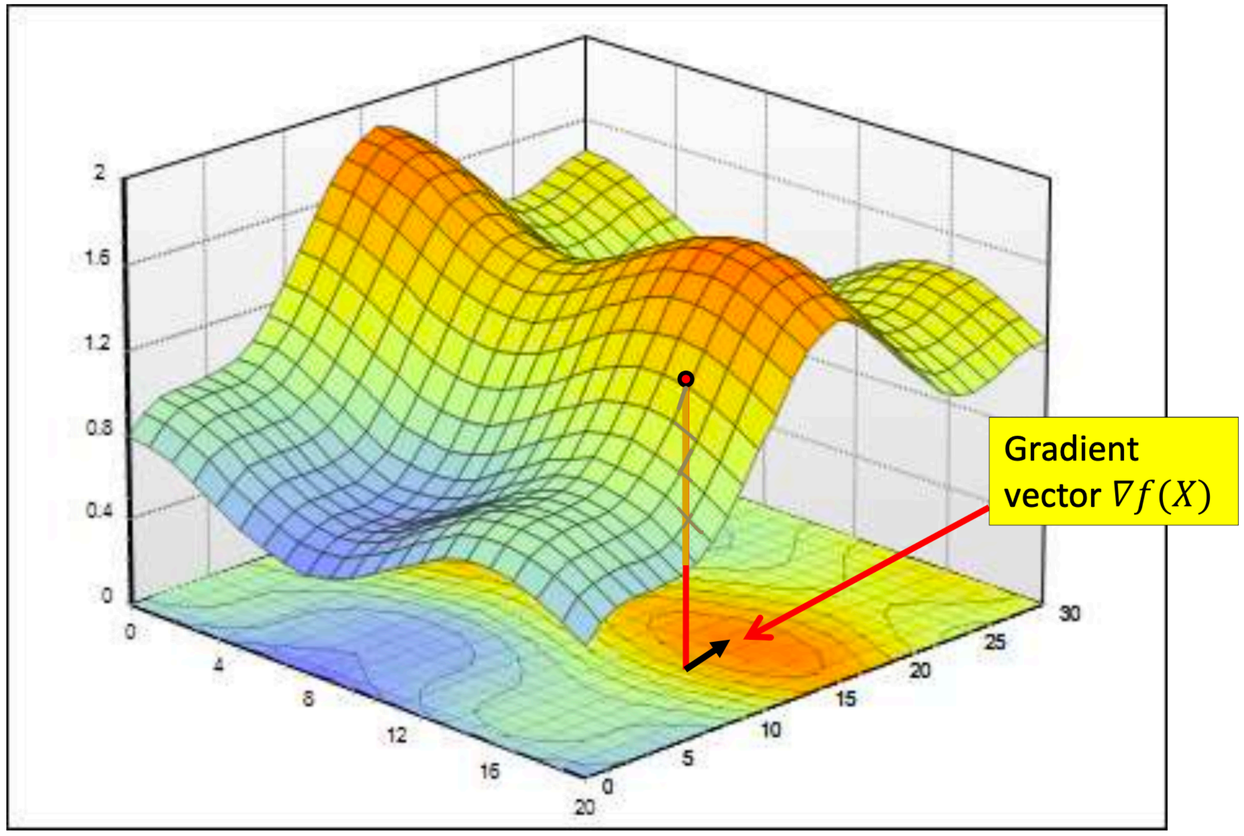
Gradient

- Where

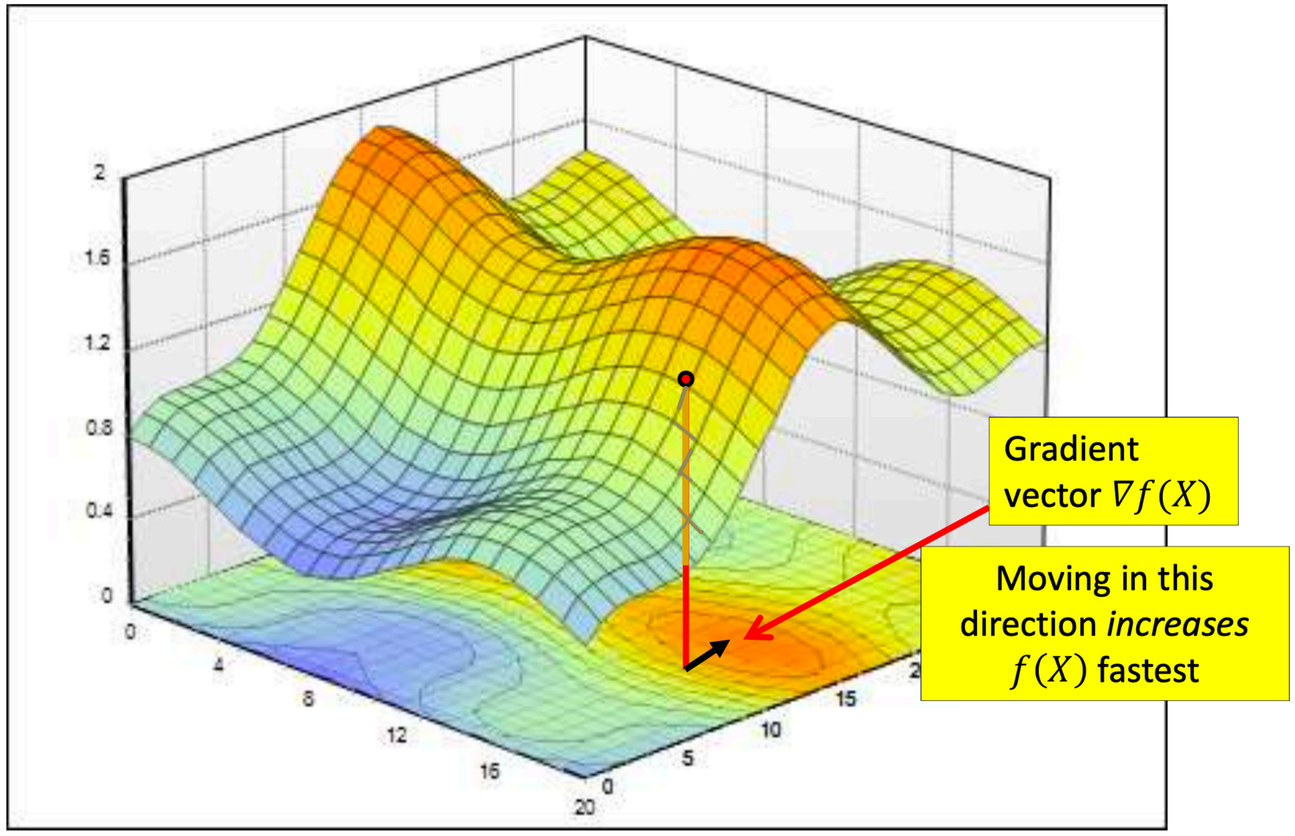
$$\nabla_{\mathbf{x}} y = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \cdots & \frac{\partial y}{\partial x_D} \end{bmatrix}$$

- Sometimes also written with a *transpose* in which case the gradient becomes a column vector

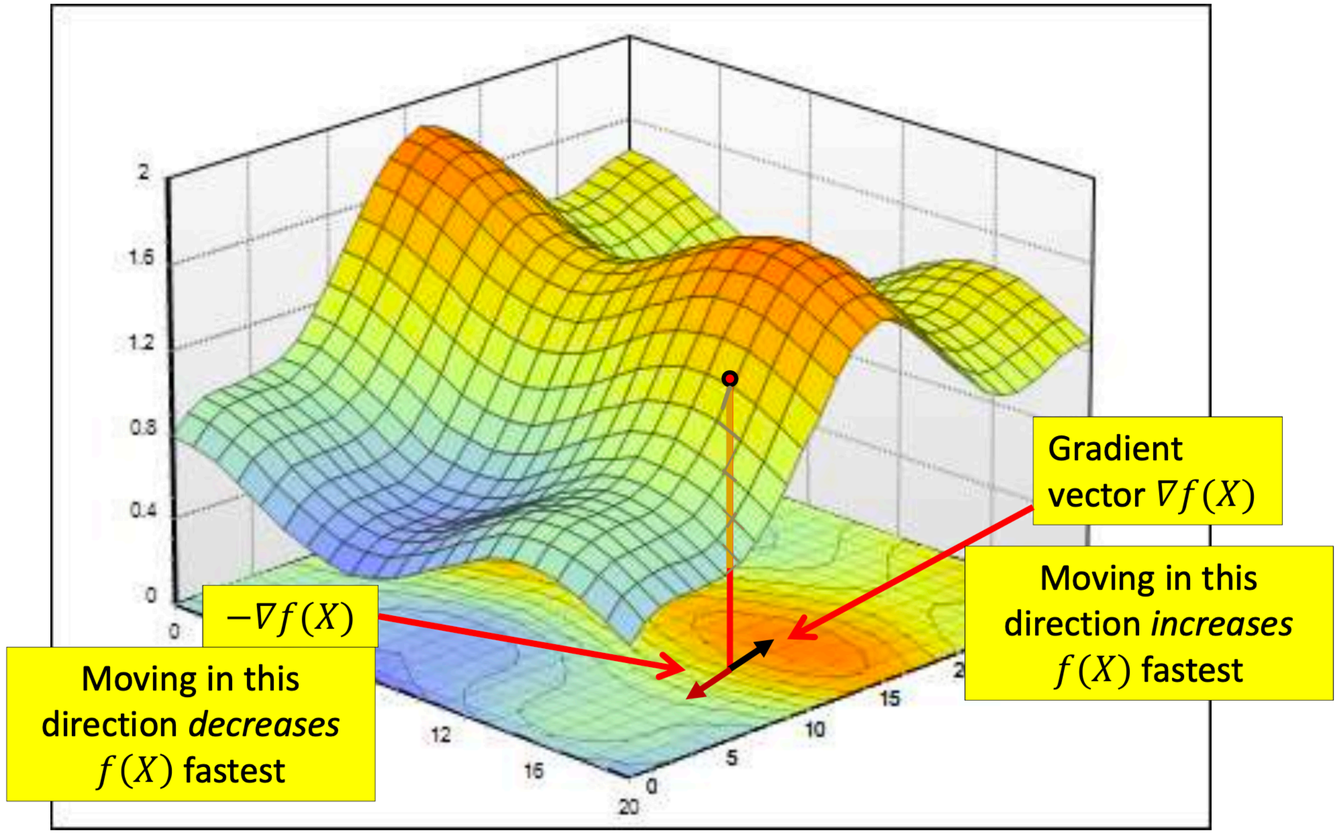
Gradient


2
1.6
1.2
0.8
0.4
0
0
4
8
12
16
20
0
5
10
15
20
25
30
Gradient vector $\nabla f(X)$

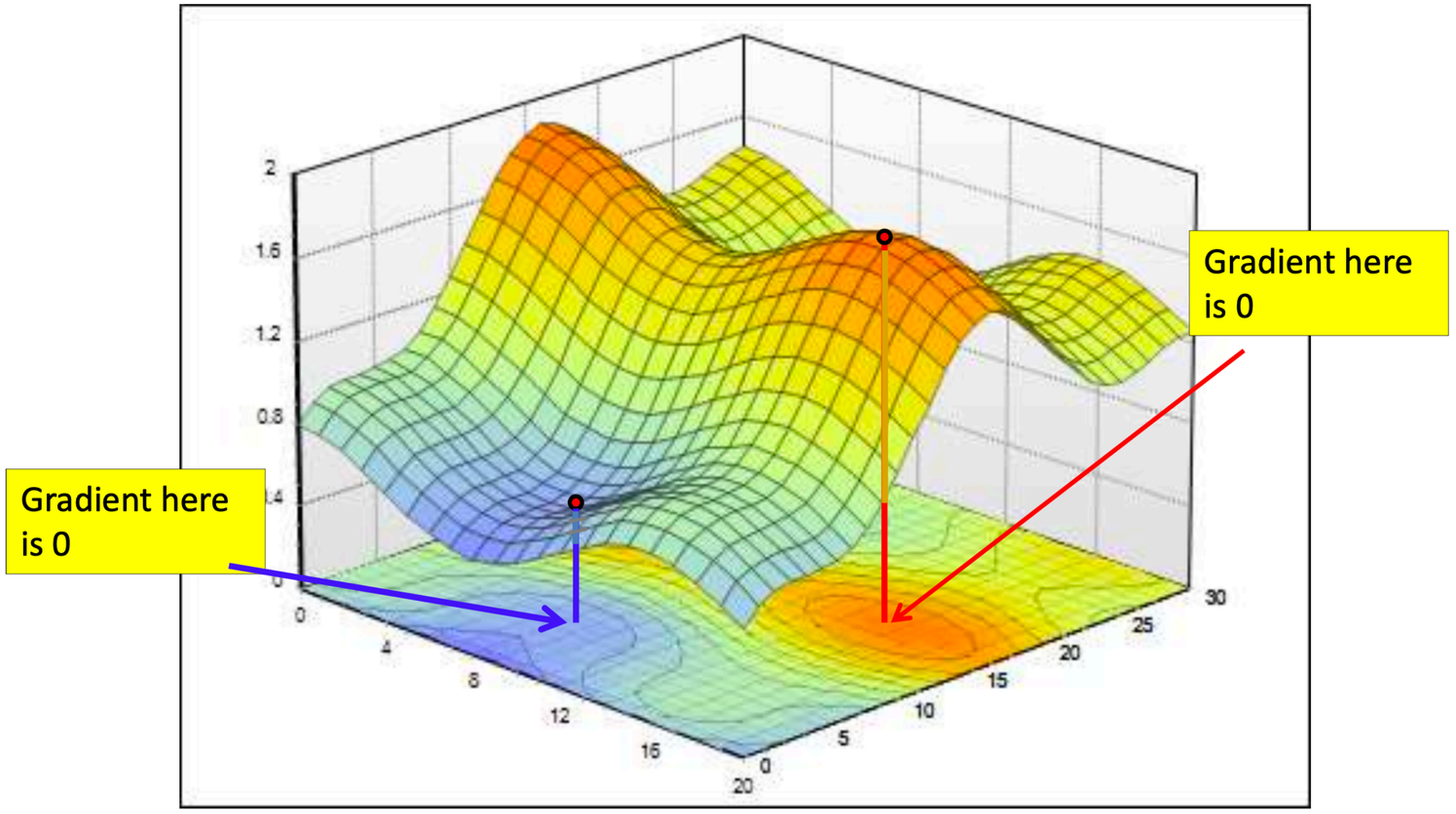
Gradient

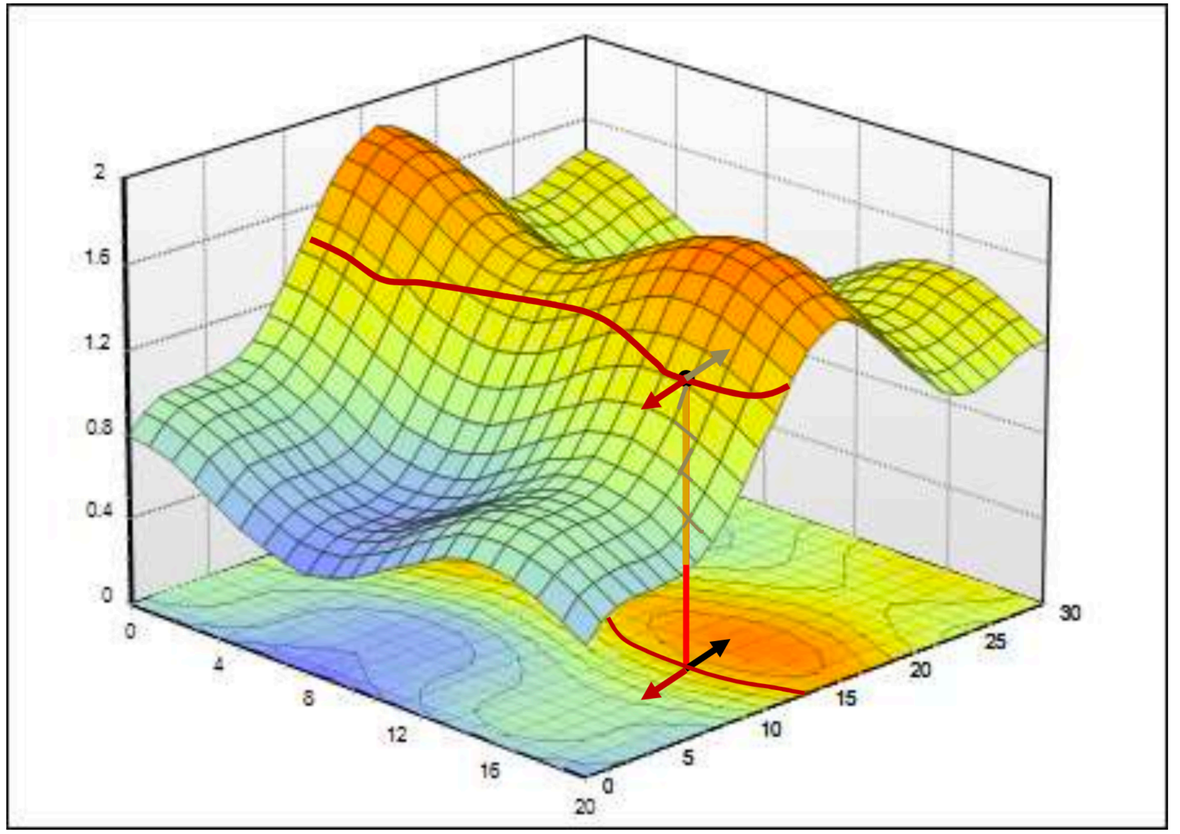

2
1.6
1.2
0.8
0.4
0
0
4
8
12
16
20
0
5
10
15
Gradient vector $\nabla f(X)$
Moving in this direction *increases* $f(X)$ fastest

Gradient


2
1.6
1.2
0.8
0.4
0
0
12
16
20
0
5
10
15
Gradient vector $\nabla f(X)$
Moving in this direction *increases* $f(X)$ fastest
$-\nabla f(X)$
Moving in this direction *decreases* $f(X)$ fastest

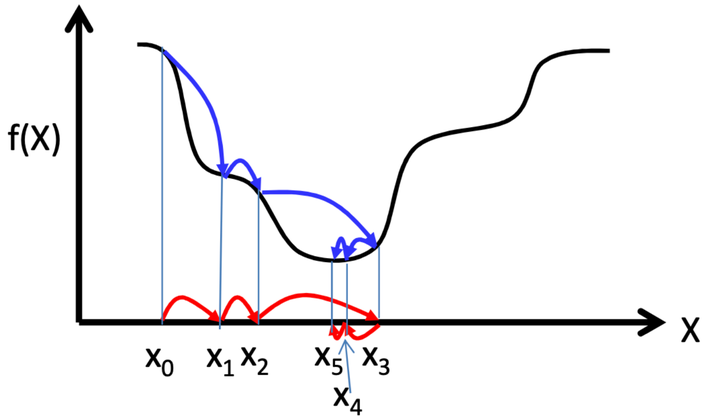
Gradient

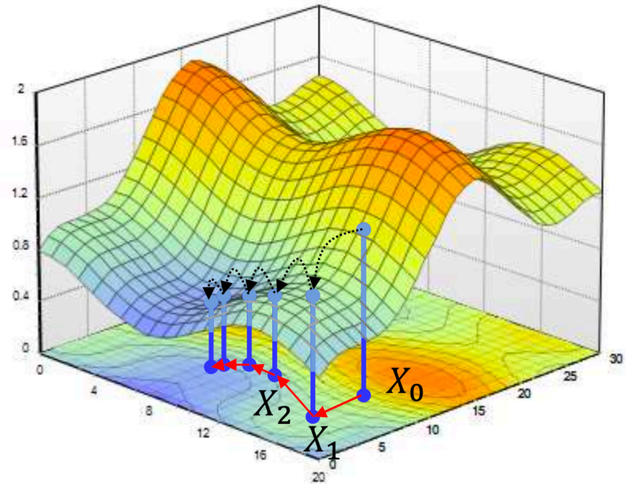

Gradient here is 0
Gradient here is 0
2
1.6
1.2
0.8
0
4
8
12
16
20
0
5
10
15
20
25
30



- The gradient vector $\nabla f(X)$ is perpendicular to the level curve

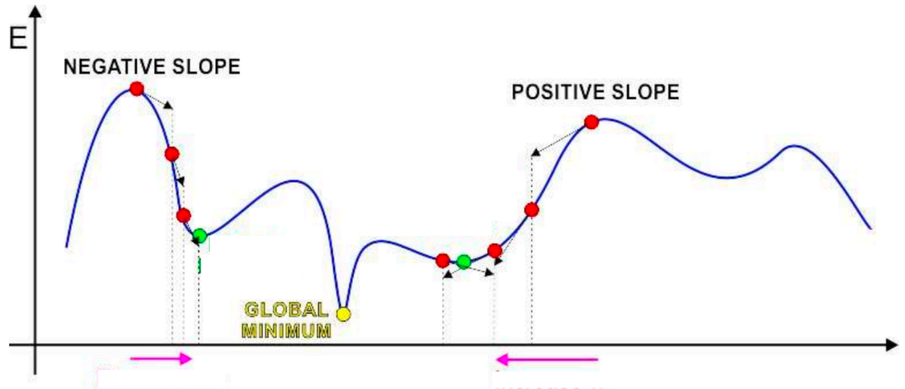
Iterative solutions





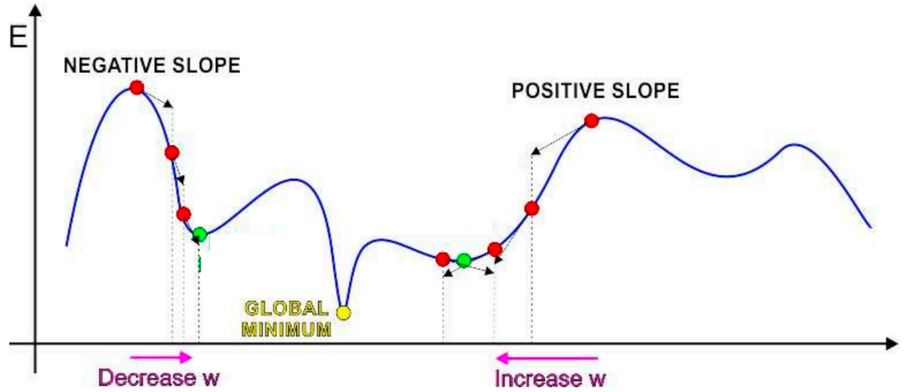
- Iterative solutions
  - Start from an initial guess X_0 for the optimal X
  - Update the guess towards a (hopefully) "better" value of $\mathrm{f}(X)$
  - Stop when $\mathrm{f}(X)$ no longer decreases
- Problems:
  - Which direction to step in
  - How big must the steps be

The Approach of Gradient Descent



- Iterative solution:
  - Start at some point
  - Find direction in which to shift this point to decrease error
    - This can be found from the derivative of the function
      - A positive derivative → moving left decreases error
      - A negative derivative → moving right decreases error
  - Shift point in this direction

The Approach of Gradient Descent



- Iterative solution: Trivial algorithm
  - Initialize x^0
  - While $f'(x^k) \neq 0$
    - If $sign\left(f'(x^k)\right)$ is positive:

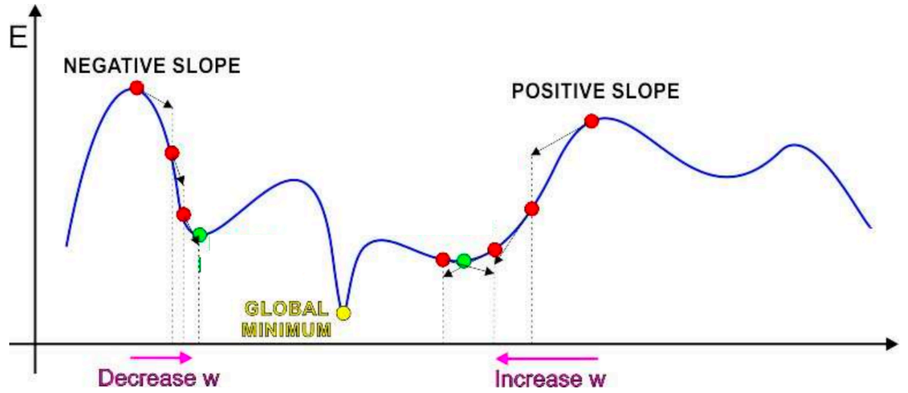
$$x^{k+1} = x^k - step$$

    - Else

$$x^{k+1} = x^k + step$$

  - What must step be to ensure we actually get to the optimum?

The Approach of Gradient Descent

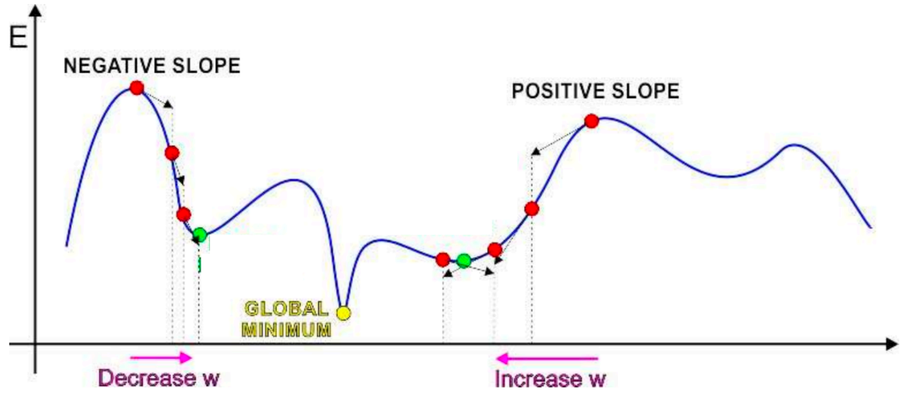


- Iterative solution: Trivial algorithm
  - Initialize x^0
  - While $f'(x^k) \neq 0$

$$x^{k+1} = x^k - sign\left(f'(x^k)\right).step$$

- Identical to previous algorithm

The Approach of Gradient Descent



- Iterative solution: Trivial algorithm
  - Initialize x^0
  - While $f'(x^k) \neq 0$

$$x^{k+1} = x^k - \eta^k f'(x^k)$$

- η^k is the "step size"

Gradient descent/ascent (multivariate)

- The gradient descent/ascent method to find the minimum or maximum of a function f iteratively
  - To find a *maximum* move *in the direction of the gradient*

$$x^{k+1} = x^k + \eta^k \nabla f\left(x^k\right)^T$$

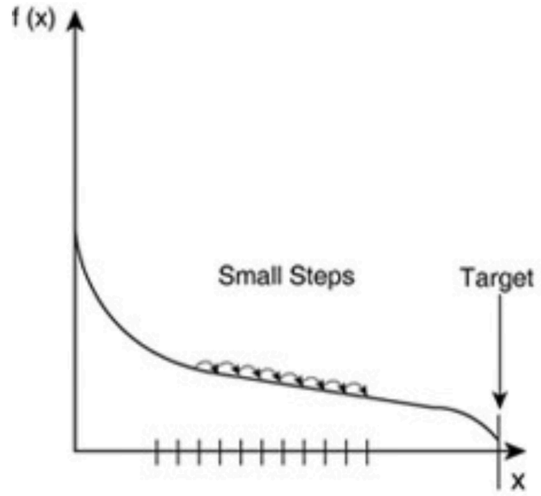
  - To find a *minimum* move *exactly opposite the direction of the gradient*

$$x^{k+1} = x^k - \eta^k \nabla f\left(x^k\right)^T$$

- Many solutions to choosing step size η^k

1. Fixed step size

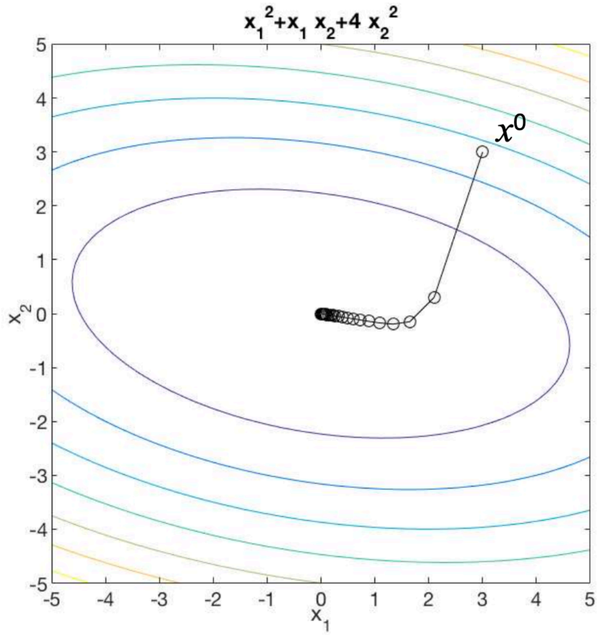
- Fixed step size
  - Use fixed value for η^k



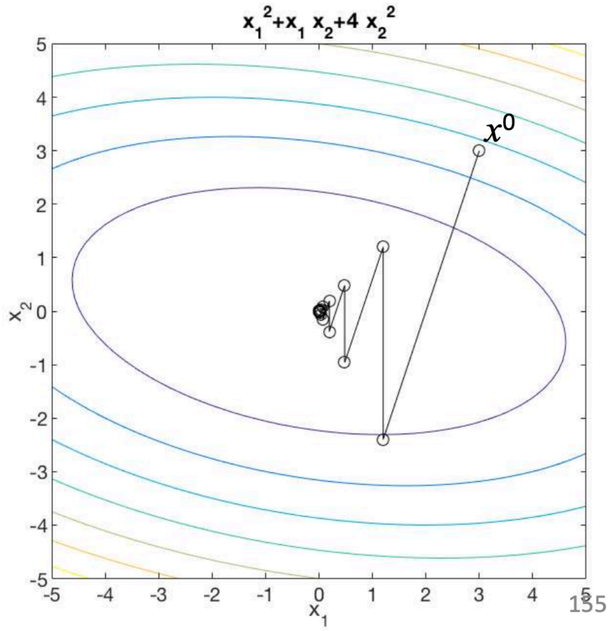
Influence of step size example (constant step size)

$$f(x_1, x_2) = (x_1)^2 + x_1 x_2 + 4(x_2)^2 \qquad x^{initial} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

$\eta = 0.1$



$\eta = 0.2$

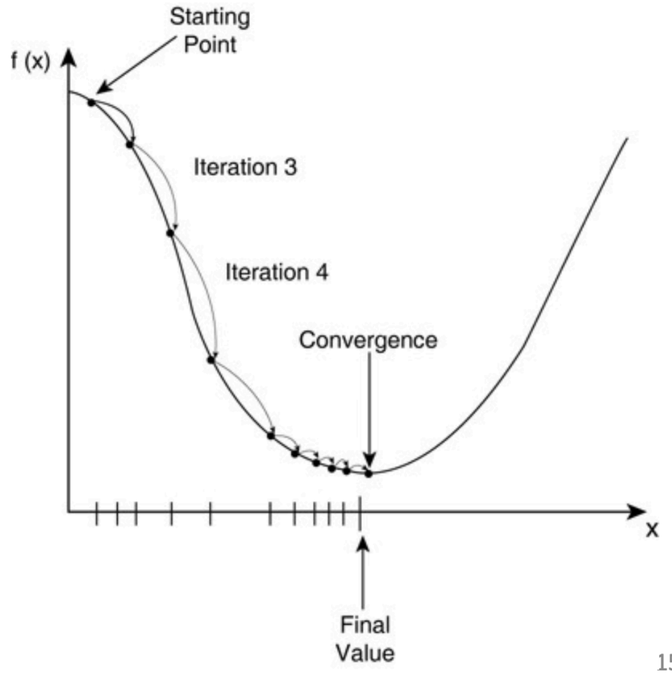


Gradient descent convergence criteria

- The gradient descent algorithm converges when one of the following criteria is satisfied

$$\left| f(x^{k+1}) - f(x^k) \right| < \varepsilon_1$$

- Or $\left\| \nabla f(x^k) \right\| < \varepsilon_2$



Recap: Gradient Descent Algorithm

- In order to minimize any function $f(x)$ w.r.t. x
- Initialize:
  - x^0
  - $k = 0$
- While $\left|f\left(x^{k+1}\right) - f\left(x^k\right)\right| > \varepsilon$
  - $x^{k+1} = x^k - \eta^k \nabla f\left(x^k\right)^T$
  - $k = k + 1$

source: m2dsupsdlclass.github.io

Recap: Gradient Descent Algorithm

- In order to minimize any function $f(x)$ w.r.t. x
- Initialize:
  - x^0
  - $k = 0$
- While $\left|f\left(x^{k+1}\right) - f\left(x^{k}\right)\right| > \varepsilon$
  - For every component i
    - $x_i^{k+1} = x_i^k - \eta^k \frac{df}{dx_i}$

    Explicitly stating it by component
  - $k = k + 1$

Training Neural Nets through Gradient Descent

Total training error:

$$Err = \frac{1}{T}\sum_{t} Div(\boldsymbol{Y}_t, \boldsymbol{d}_t)$$

- Gradient descent algorithm:
- Initialize all weights and biases $\left\{w_{ij}^{(k)}\right\}$
  - Using the extended notation: the bias is also a weight
- Do:
  - For every layer k for all i, j, update:
    - $w_{i,j}^{(k)} = w_{i,j}^{(k)} - \eta \frac{dErr}{dw_{i,j}^{(k)}}$
- Until Err has converged

Assuming the bias is also represented as a weight

Neural network training algorithm

- Initialize all weights and biases $(\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2, \ldots, \mathbf{W}_N, \mathbf{b}_N)$
- Do:
  - $Err = 0$
  - For all k, initialize $\nabla_{\mathbf{W}_k} Err = 0$, $\nabla_{\mathbf{b}_k} Err = 0$
  - For all $t = 1:T$
    - Forward pass : Compute
      - Output $\boldsymbol{Y(X_t)}$
      - Divergence $\boldsymbol{Div(Y_t, d_t)}$
      - $Err \mathrel{+}= \boldsymbol{Div(Y_t, d_t)}$
    - Backward pass: For all k compute:
      - $\nabla_{\mathbf{y}_k} Div = \nabla_{\mathbf{z}_{k+1}} Div \, \mathbf{W}_k$
      - $\nabla_{\mathbf{z}_k} Div = \nabla_{\mathbf{y}_k} Div \, J_{\mathbf{y}_k}(\mathbf{z}_k)$
      - $\nabla_{\mathbf{W}_k} \boldsymbol{Div(Y_t, d_t)}$; $\nabla_{\mathbf{b}_k} \boldsymbol{Div(Y_t, d_t)}$
      - $\nabla_{\mathbf{W}_k} Err \mathrel{+}= \nabla_{\mathbf{W}_k} \boldsymbol{Div(Y_t, d_t)}$; $\nabla_{\mathbf{b}_k} Err \mathrel{+}= \nabla_{\mathbf{b}_k} \boldsymbol{Div(Y_t, d_t)}$
  - For all k, update:

$$\mathbf{W}_k = \mathbf{W}_k - \frac{\eta}{T} \left(\nabla_{\mathbf{W}_k} Err\right)^T; \qquad \mathbf{b}_k = \mathbf{b}_k - \frac{\eta}{T} \left(\nabla_{\mathbf{W}_k} Err\right)^T$$

- Until Err has converged

How to calculate gradients?

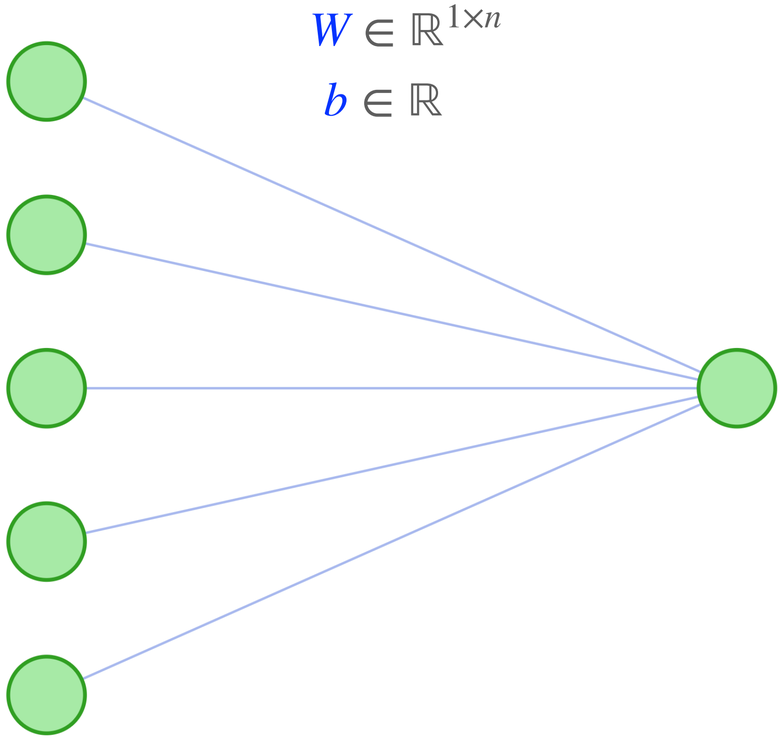
How to calculate gradients?

- Solution = **forward** + **backward** propagation

Forward propagation

Linear model

$$\mathbb{R}^n \ni \boldsymbol{x}$$



$$a \in (0,1)$$

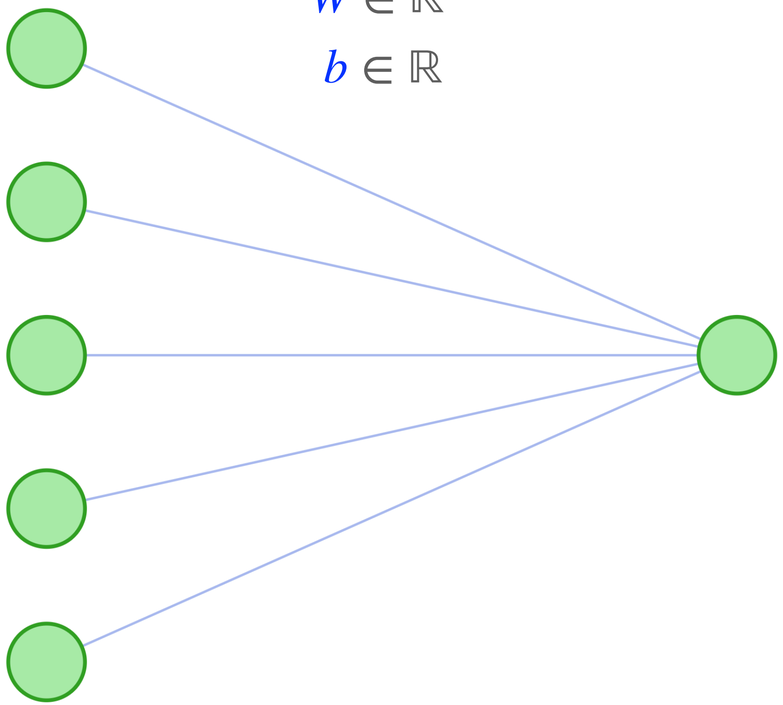
Forward propagation

Linear model

$W \in \mathbb{R}^{1\times n}$

$b \in \mathbb{R}$

$\mathbb{R}^n \ni \boldsymbol{x}$



$a \in (0,1)$

Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times n}$

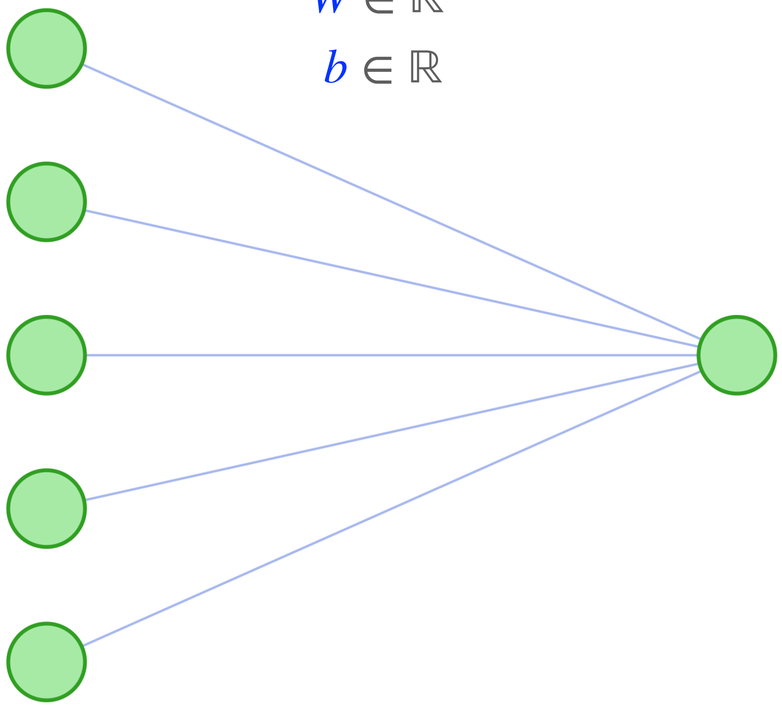
Forward propagation

Linear model

$W \in \mathbb{R}^{1\times n}$

$b \in \mathbb{R}$

$\mathbb{R}^n \ni \boldsymbol{x}$



$a \in (0,1)$

Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times n}$

$z = \boldsymbol{x}^T W^T + b \in \mathbb{R}^{1\times 1}$

$a = \sigma(z) \in \mathbb{R}^{1\times 1}$

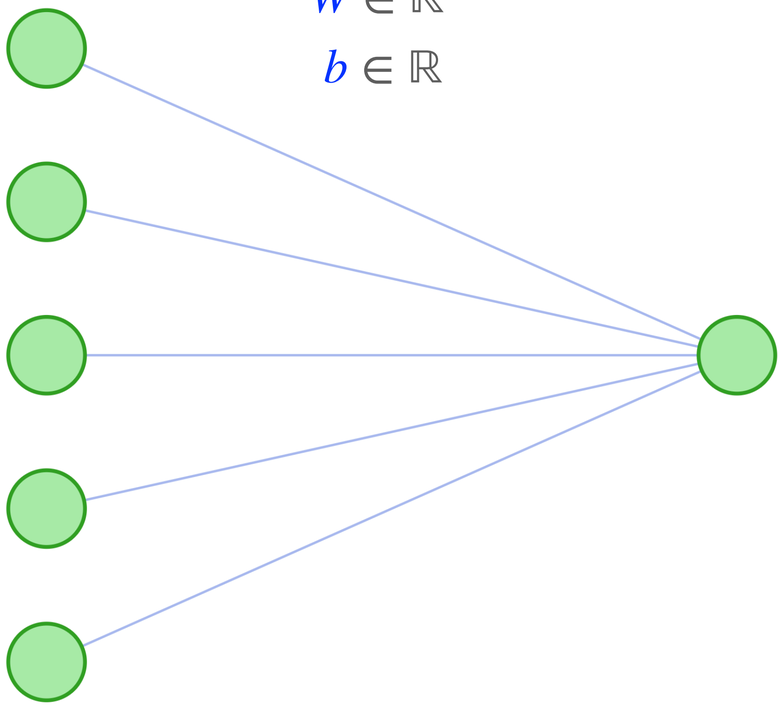
Forward propagation

Linear model

$W \in \mathbb{R}^{1 \times n}$

$b \in \mathbb{R}$

$\mathbb{R}^{N \times n} \ni \boldsymbol{X}$



$\boldsymbol{a} \in (0,1)^{N}$

Efficient implementation

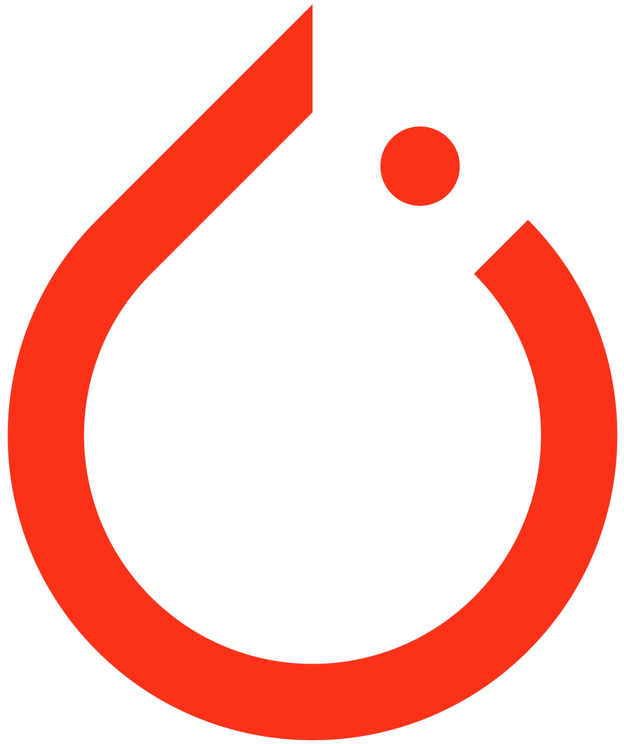
$X \in \mathbb{R}^{N \times n}$

$z = X\,W^{T} \oplus b \in \mathbb{R}^{N \times 1}$

$a = \sigma(z) \in \mathbb{R}^{N \times 1}$

Forward propagation

Code demo in PyTorch



Forward propagation

In [1]:
```
import torch

torch.manual_seed(1); # for the reproducibility
```
executed in 299ms, finished 17:46:14 2023-03-11

Linear model

In [2]:
```
xT = torch.tensor([-8.0, 1.0, -3.0, 5.0, 2.0]).view(1, 5) # horizontal vector
xT
```
executed in 22ms, finished 17:46:14 2023-03-11

Out[2]:
```
tensor([[-8.,  1., -3.,  5.,  2.]])
```

In [3]:
```
xT.shape
```
executed in 19ms, finished 17:46:14 2023-03-11

Out[3]:
```
torch.Size([1, 5])
```

In [4]:
```
W = torch.rand(5).view(1, 5)  # random numbers from (0, 1)
b = torch.rand(1)
```
executed in 2ms, finished 17:46:14 2023-03-11

In [5]:
```
W
```
executed in 2ms, finished 17:46:14 2023-03-11

Out[5]:
```
tensor([[0.7576, 0.2793, 0.4031, 0.7347, 0.0293]])
```

In [6]:
```
b
```
executed in 3ms, finished 17:46:14 2023-03-11

Out[6]:
```
tensor([0.7999])
```

In [7]:
```
W.shape, b.shape
```
executed in 2ms, finished 17:46:14 2023-03-11

Out[7]:
```
(torch.Size([1, 5]), torch.Size([1]))
```

$$z = \boldsymbol{x}^T W^T + b \in \mathbb{R}^{1\times 1}$$

In [8]:
```
z = torch.mm( xT, W.T ) + b
z.shape
```
executed in 2ms, finished 17:46:14 2023-03-11

Out[8]:
```
torch.Size([1, 1])
```

In [9]:
```
z
```
executed in 3ms, finished 17:46:14 2023-03-11

Out[9]:
```
tensor([[-2.4591]])
```

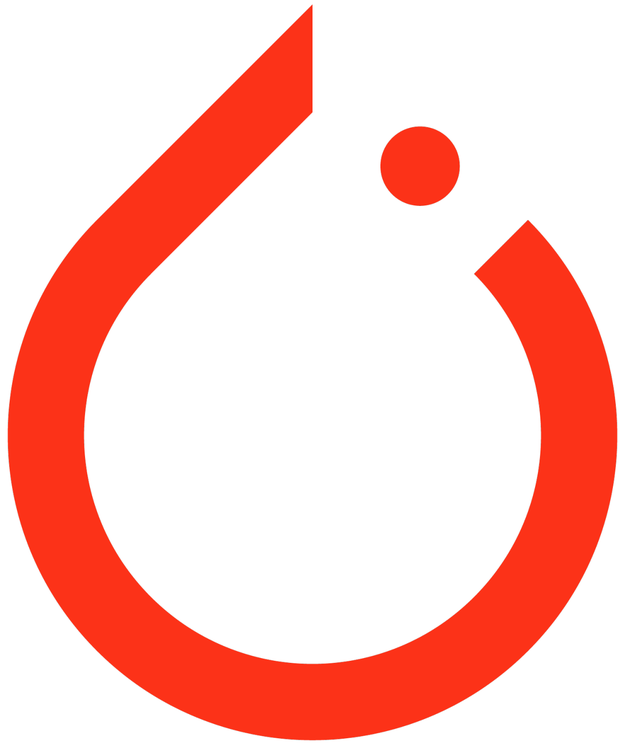
$$a = \sigma(z) \in \mathbb{R}^{1\times 1}$$

In [10]:
```
a = torch.sigmoid(z)
a
```
executed in 3ms, finished 17:46:14 2023-03-11

Out[10]:
```
tensor([[0.0788]])
```

Forward propagation

Code demo in PyTorch



Forward propagation

In [1]:
```
import torch

torch.manual_seed(1); # for the reproducibility
```
executed in 284ms, finished 17:43:29 2023-03-11

Linear model

In [2]:
```
X = torch.tensor([[-8.0, 1.0, -3.0, 5.0, 2.0], [-2.0, 2.0, -1.0, 5.0, 2.0], [7.0, 1.0, 9.0, -5.0, -2.0]])
X
```
executed in 17ms, finished 17:43:29 2023-03-11

Out[2]:
```
tensor([[-8.,  1., -3.,  5.,  2.],
        [-2.,  2., -1.,  5.,  2.],
        [ 7.,  1.,  9., -5., -2.]])
```

In [3]:
```
X.shape
```
executed in 25ms, finished 17:43:29 2023-03-11

Out[3]:
```
torch.Size([3, 5])
```

In [4]:
```
W = torch.rand(5).view(1, 5)  # random numbers from (0, 1)
b = torch.rand(1)
```
executed in 8ms, finished 17:43:29 2023-03-11

In [5]:
```
W
```
executed in 2ms, finished 17:43:29 2023-03-11

Out[5]:
```
tensor([[0.7576, 0.2793, 0.4031, 0.7347, 0.0293]])
```

In [6]:
```
b
```
executed in 3ms, finished 17:43:29 2023-03-11

Out[6]:
```
tensor([0.7999])
```

In [7]:
```
W.shape, b.shape
```
executed in 2ms, finished 17:43:29 2023-03-11

Out[7]:
```
(torch.Size([1, 5]), torch.Size([1]))
```

$$z = X\,W^T \oplus b \in \mathbb{R}^{N\times 1}$$

In [8]:
```
z = torch.mm( X, W.T ) + b
z.shape
```
executed in 2ms, finished 17:43:29 2023-03-11

Out[8]:
```
torch.Size([3, 1])
```

In [9]:
```
z
```
executed in 2ms, finished 17:43:29 2023-03-11

Out[9]:
```
tensor([[-2.4591],
        [ 3.1721],
        [ 6.2782]])
```

$$a = \sigma(z) \in \mathbb{R}^{N\times 1}$$

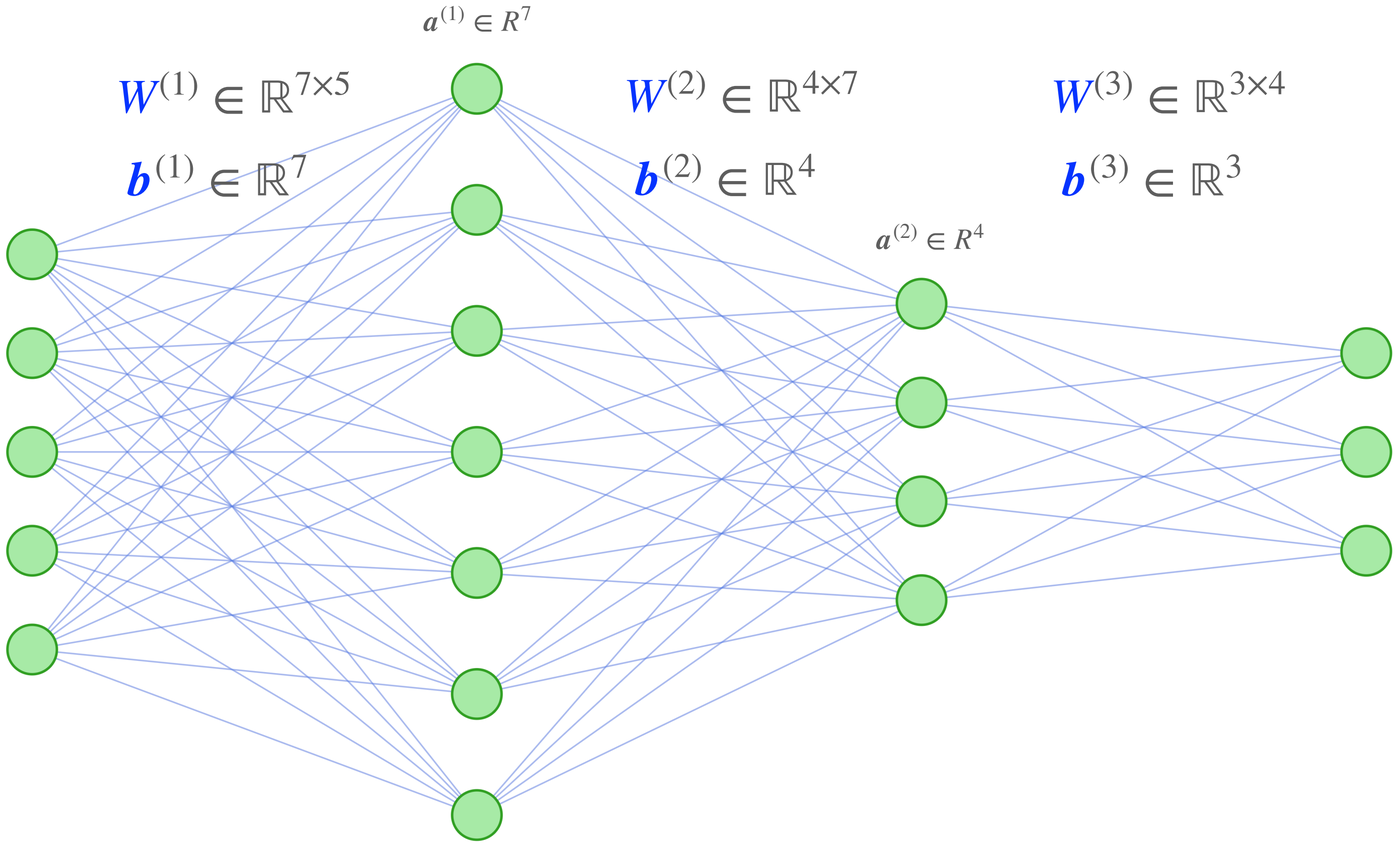
In [10]:
```
a = torch.sigmoid(z)
a
```
executed in 2ms, finished 17:43:29 2023-03-11

Out[10]:
```
tensor([[0.0788],
        [0.9598],
        [0.9981]])
```

Forward propagation

MLP

$\mathbb{R}^5 \ni \boldsymbol{x}$



$\boldsymbol{a}^{(3)} \in \mathbb{R}^3$

Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$

$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$

$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

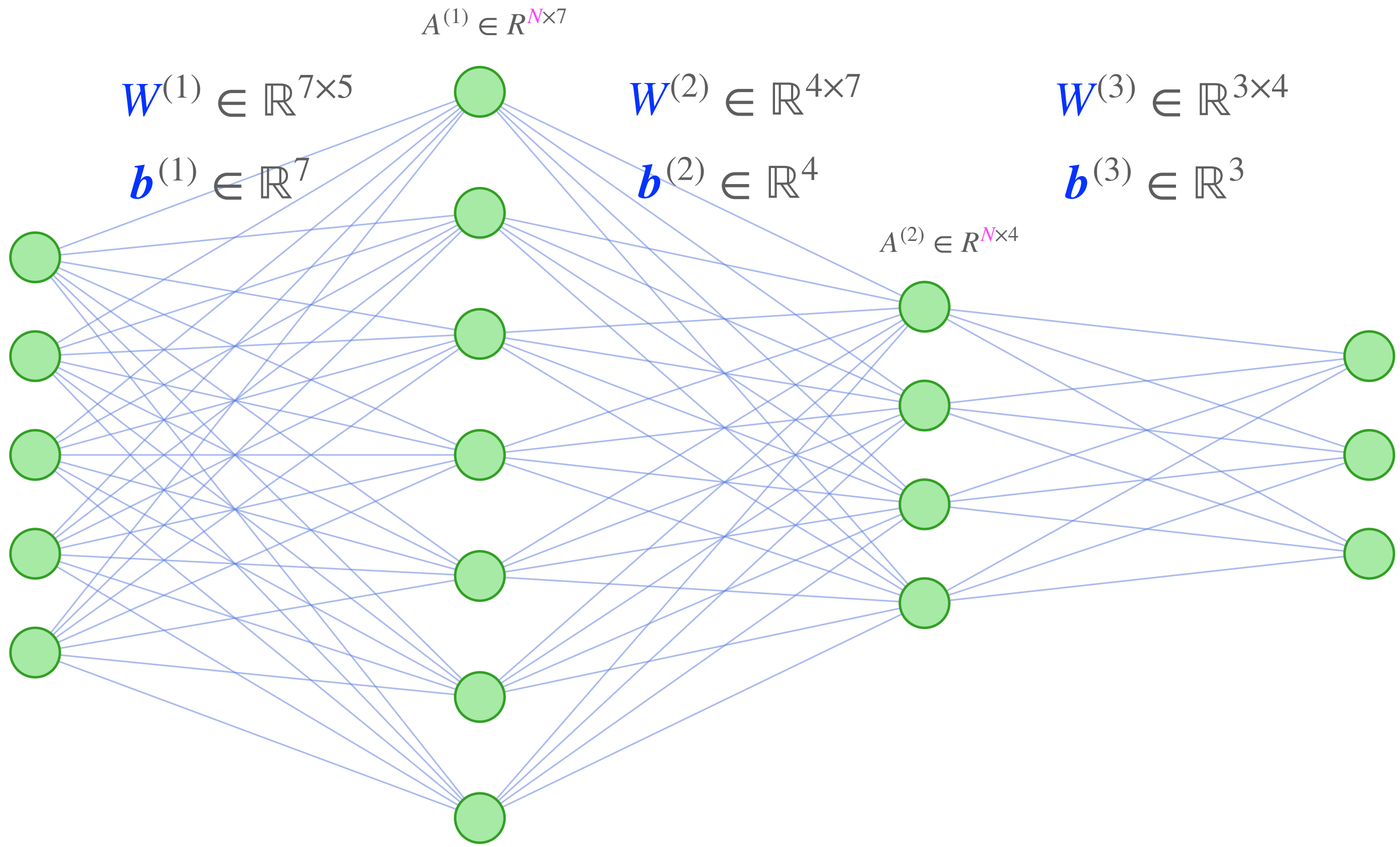
$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$

$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

Forward propagation

MLP

$\mathbb{R}^{N\times 5} \ni \boldsymbol{X}$



$A^{(3)} \in \mathbb{R}^{N\times 3}$

Efficient implementation

$X \in \mathbb{R}^{N\times 5}$

$Z^{(1)} = X\, W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{N\times 7}$

$A^{(1)} = g_1(Z^{(1)}) \in \mathbb{R}^{N\times 7}$

$Z^{(2)} = A^{(1)}\, W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{N\times 4}$

$A^{(2)} = g_2(Z^{(2)}) \in \mathbb{R}^{N\times 4}$

$Z^{(3)} = A^{(2)}\, W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{N\times 3}$

$A^{(3)} = \mathrm{softmax}(Z^{(3)}) \in \mathbb{R}^{N\times 3}$

Cross entropy loss

$$X \in \mathbb{R}^{N\times 5}$$

$$Z^{(1)} = X\,W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{N\times 7} \qquad A^{(1)} = g_1(Z^{(1)}) \in \mathbb{R}^{N\times 7}$$

$$Z^{(2)} = A^{(1)}\,W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{N\times 4} \qquad A^{(2)} = g_2(Z^{(2)}) \in \mathbb{R}^{N\times 4}$$

$$Z^{(3)} = A^{(2)}\,W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{N\times 3} \qquad A^{(3)} = \text{softmax}(Z^{(3)}) \in \mathbb{R}^{N\times 3}$$

- We have **3 classes (0, 1, 2)** and assume we have 4 samples ($N = 4$) of the following classes: 2, 0, 1, 2.

$$Y = \begin{bmatrix} 0,0,1 \\ 1,0,0 \\ 0,1,0 \\ 0,0,1 \end{bmatrix} \in \mathbb{R}^{N\times 3}$$

(on-hot encoding)

- **Cross entropy loss**

$$L = -\frac{1}{N}\,\text{sum}\left(Y \otimes \log A^{(3)}\right) = -\frac{1}{N}\,\text{sum}\begin{bmatrix} Y[1,:] \otimes A^{(3)}[1,:] \\ Y[2,:] \otimes A^{(3)}[2,:] \\ \vdots \\ Y[N,:] \otimes A^{(3)}[N,:] \end{bmatrix}$$

Let’s calculate the derivatives

$\frac{\partial L}{\partial A^{(p)}} = ?$

$$L = -\frac{1}{N} \operatorname{sum}\left(Y \otimes \log A^{(3)}\right)$$

Let’s calculate the derivatives

$\frac{\partial L}{\partial A^{(p)}} = ?$

$$L = -\frac{1}{N} \operatorname{sum}\left(Y \otimes \log A^{(3)}\right)$$

Let’s calculate the derivatives

$\frac{\partial L}{\partial A^{(p)}} = ?$

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right)$$

- Observe that the total loss is the average of the losses of independent samples

Let’s calculate the derivatives

$\frac{\partial L}{\partial A^{(p)}} = ?$

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right)$$

- Observe that the total loss is the average of the losses of independent samples
- One can do calculations row-wise,
  i.e.

Let's calculate the derivatives

$\frac{\partial L}{\partial A^{(p)}} = ?$

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right)$$

- Observe that the total loss is the average of the losses of independent samples
- One can do calculations row-wise, i.e.

$$\frac{\partial L}{\partial A^{(3)}} = -\frac{1}{N}\begin{bmatrix} \frac{\partial L}{\partial \boldsymbol{a}_1^{(3)}} \\ \frac{\partial L}{\partial \boldsymbol{a}_2^{(3)}} \\ \vdots \\ \frac{\partial L}{\partial \boldsymbol{a}_N^{(3)}} \end{bmatrix},$$

Let's calculate the derivatives

$\frac{\partial L}{\partial A^{(p)}} = ?$

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right)$$

- Observe that the total loss is the average of the losses of independent samples
- One can do calculations row-wise, i.e.

$$\frac{\partial L}{\partial A^{(3)}} = -\frac{1}{N}\begin{bmatrix} \frac{\partial L}{\partial \boldsymbol{a}_1^{(3)}} \\ \frac{\partial L}{\partial \boldsymbol{a}_2^{(3)}} \\ \vdots \\ \frac{\partial L}{\partial \boldsymbol{a}_N^{(3)}} \end{bmatrix}, \qquad \frac{\partial L}{\partial Z^{(3)}} = -\frac{1}{N}\begin{bmatrix} \frac{\partial L}{\partial \boldsymbol{z}_1^{(3)}} \\ \frac{\partial L}{\partial \boldsymbol{z}_2^{(3)}} \\ \vdots \\ \frac{\partial L}{\partial \boldsymbol{z}_N^{(3)}} \end{bmatrix},$$

Let's calculate the derivatives

$\frac{\partial L}{\partial A^{(p)}} = ?$

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right)$$

- Observe that the total loss is the average of the losses of independent samples
- One can do calculations row-wise, i.e.

$$\frac{\partial L}{\partial A^{(3)}} = -\frac{1}{N}\begin{bmatrix} \frac{\partial L}{\partial \boldsymbol{a}_1^{(3)}} \\ \frac{\partial L}{\partial \boldsymbol{a}_2^{(3)}} \\ \vdots \\ \frac{\partial L}{\partial \boldsymbol{a}_N^{(3)}} \end{bmatrix}, \qquad \frac{\partial L}{\partial Z^{(3)}} = -\frac{1}{N}\begin{bmatrix} \frac{\partial L}{\partial \boldsymbol{z}_1^{(3)}} \\ \frac{\partial L}{\partial \boldsymbol{z}_2^{(3)}} \\ \vdots \\ \frac{\partial L}{\partial \boldsymbol{z}_N^{(3)}} \end{bmatrix}, \quad \cdots$$

Let's calculate the derivatives

Single sample (row)

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right) = -\frac{1}{N}\left(\mathrm{sum}(Y[1,:] \otimes \log A^{(3)}[1,:]) + \sum_{r=2}^{N} \mathrm{sum}(Y[r,:] \otimes \log A^{(3)}[r,:])\right)$$

$$\boldsymbol{y}^T := Y[1,:]$$

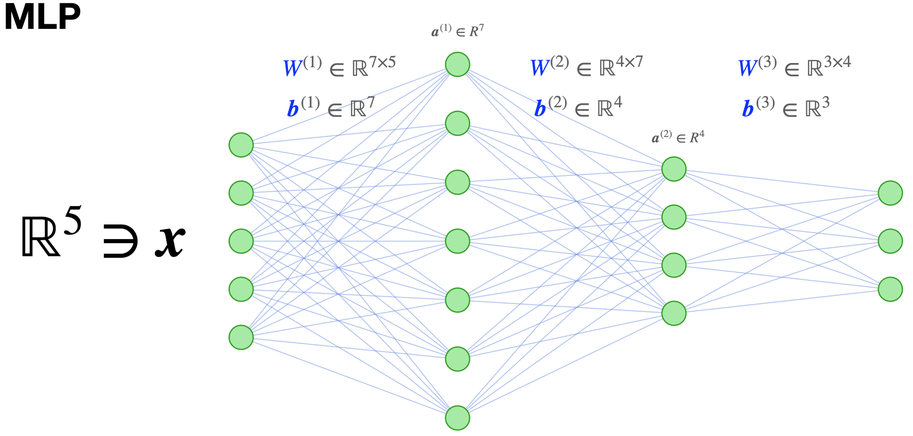
$$\boldsymbol{a}^{(3)^T} := A^{(3)}[1,:] = \boldsymbol{a}_1^{(3)}$$

$$\boldsymbol{z}^{(3)^T} := Z^{(3)}[1,:]$$

$$\frac{\partial L}{\partial \boldsymbol{a}^{(3)}} = ?,\ \frac{\partial L}{\partial \boldsymbol{z}^{(3)}} = ?,\ \frac{\partial L}{\partial \boldsymbol{a}^{(2)}} = ?,\ \frac{\partial L}{\partial \boldsymbol{z}^{(2)}} = ?,\ \frac{\partial L}{\partial \boldsymbol{a}^{(1)}} = ?,\ \frac{\partial L}{\partial \boldsymbol{z}^{(1)}} = ?,$$

Forward propagation

MLP



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$\boldsymbol{z}^{(1)} = \boldsymbol{x}^T \boldsymbol{W}^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$

$\boldsymbol{a}^{(1)} = g_1(\boldsymbol{z}^{(1)}) \in \mathbb{R}^{1\times7}$

$\boldsymbol{z}^{(2)} = \boldsymbol{a}^{(1)} \boldsymbol{W}^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$

$\boldsymbol{a}^{(2)} = g_2(\boldsymbol{z}^{(2)}) \in \mathbb{R}^{1\times4}$

$\boldsymbol{z}^{(3)} = \boldsymbol{a}^{(2)} \boldsymbol{W}^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$

$\boldsymbol{a}^{(3)} = \mathrm{softmax}(\boldsymbol{z}^{(3)}) \in \mathbb{R}^{1\times3}$

Let's calculate the derivatives

$$\boldsymbol{x}^T \in \mathbb{R}^{1\times 5}$$

$$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times 7} \qquad \boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times 7}$$

$$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times 4} \qquad \boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times 4}$$

$$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times 3} \qquad \boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times 3}$$

$$\boldsymbol{y}^T := Y[1, :]$$

$$\boldsymbol{a}^{(3)^T} := A^{(3)}[1, :]$$

$$L_1 := -\,\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\,\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

- $$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = ?$$

Let's calculate the derivatives

$$\boldsymbol{x}^T \in \mathbb{R}^{1\times 5}$$

$$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times 7} \qquad a^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times 7}$$

$$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times 4} \qquad a^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times 4}$$

$$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times 3} \qquad a^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times 3}$$

$$\boldsymbol{y}^T := Y[1, :]$$

$$\boldsymbol{a}^{(3)^T} := A^{(3)}[1, :]$$

$$L_1 := -\,\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\,\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

- $$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\,\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T}$$

Let's calculate the derivatives

$$\boldsymbol{x}^T \in \mathbb{R}^{1\times 5}$$

$$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times 7} \qquad \boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times 7}$$

$$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times 4} \qquad \boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times 4}$$

$$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times 3} \qquad \boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times 3}$$

$$\boldsymbol{y}^T := Y[1, :]$$

$$\boldsymbol{a}^{(3)^T} := A^{(3)}[1, :]$$

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

- $\dfrac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T}$ $\qquad \dfrac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = ?$

Let's calculate the derivatives

$$\boldsymbol{x}^T \in \mathbb{R}^{1\times 5}$$

$$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times 7} \qquad \boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times 7}$$

$$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times 4} \qquad \boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times 4}$$

$$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times 3} \qquad \boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times 3}$$

$$\boldsymbol{y}^T := Y[1, :]$$

$$\boldsymbol{a}^{(3)^T} := A^{(3)}[1, :]$$

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

- $$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \qquad \frac{\partial L_1}{\partial z^{(3)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} \cdot \frac{\partial \boldsymbol{a}^{(3)}}{\partial z^{(3)}}$$

Deep Learning calculus (recap)

$$\hat{y} = \text{softmax}(z), \qquad \frac{\partial \hat{y}}{\partial z} = \text{diag}(\hat{y}) - \hat{y}\hat{y}^T$$

In [1]:
```
import torch
```
executed in 257ms, finished 20:22:30 2023-03-11

In [2]:
```
z = torch.tensor([5.0, -2.0, -1.0, 3.0], requires_grad=True)
z
```
executed in 32ms, finished 20:22:30 2023-03-11

Out[2]:
```
tensor([ 5., -2., -1.,  3.], requires_grad=True)
```

In [3]:
```
a = torch.softmax(z, dim=0)
```
executed in 18ms, finished 20:22:30 2023-03-11

In [4]:
```
a
```
executed in 37ms, finished 20:22:30 2023-03-11

Out[4]:
```
tensor([8.7817e-01, 8.0079e-04, 2.1768e-03, 1.1885e-01],
       grad_fn=<SoftmaxBackward0>)
```

In [5]:
```
a[2].backward()
```
executed in 15ms, finished 20:22:30 2023-03-11

In [6]:
```
z.grad
```
executed in 2ms, finished 20:22:30 2023-03-11

Out[6]:
```
tensor([-1.9116e-03, -1.7431e-06,  2.1720e-03, -2.5871e-04])
```

In [7]:
```
torch.diag(a) - torch.mm( a.view(4, 1), a.view(1, 4) )
```
executed in 3ms, finished 20:22:30 2023-03-11

Out[7]:
```
tensor([[ 1.0698e-01, -7.0323e-04, -1.9116e-03, -1.0437e-01],
        [-7.0323e-04,  8.0015e-04, -1.7431e-06, -9.5172e-05],
        [-1.9116e-03, -1.7431e-06,  2.1720e-03, -2.5871e-04],
        [-1.0437e-01, -9.5172e-05, -2.5871e-04,  1.0472e-01]],
       grad_fn=<SubBackward0>)
```

Let's calculate the derivatives

$$\boldsymbol{x}^T \in \mathbb{R}^{1\times 5}$$

$$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times 7} \qquad \boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times 7}$$

$$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times 4} \qquad \boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times 4}$$

$$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times 3} \qquad \boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times 3}$$

$$\boldsymbol{y}^T := Y[1, :]$$

$$\boldsymbol{a}^{(3)^T} := A^{(3)}[1, :]$$

$$L_1 := -\,\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\,\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

- $$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\,\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \qquad \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = -\,(\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T}) \cdot \left(\text{diag}(\boldsymbol{a}^{(3)}) - \boldsymbol{a}^{(3)}\boldsymbol{a}^{(3)^T}\right)$$

Let's calculate the derivatives

$$x^T \in \mathbb{R}^{1\times 5}$$

$$z^{(1)} = x^T W^{(1)^T} + b^{(1)} \in \mathbb{R}^{1\times 7} \qquad a^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times 7}$$

$$z^{(2)} = a^{(1)} W^{(2)^T} + b^{(2)} \in \mathbb{R}^{1\times 4} \qquad a^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times 4}$$

$$z^{(3)} = a^{(2)} W^{(3)^T} + b^{(3)} \in \mathbb{R}^{1\times 3} \qquad a^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times 3}$$

$$y^T := Y[1, :]$$
$$a^{(3)^T} := A^{(3)}[1, :]$$

$$L_1 := -\,\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\,\text{sum}(y \otimes \log a^{(3)})$$

- $$\frac{\partial L_1}{\partial a^{(3)}} = -\,y^T \div a^{(3)^T} \qquad \frac{\partial L_1}{\partial z^{(3)}} = -\,(y^T \div a^{(3)^T}) \cdot \left(\text{diag}(a^{(3)}) - a^{(3)} a^{(3)^T}\right)$$

$$(y^T \div a^{(3)^T}) \cdot \text{diag}(a^{(3)}) = y^T$$

$$(y^T \div a^{(3)^T}) \cdot a^{(3)} a^{(3)^T} = a^{(3)^T}$$

Let's calculate the derivatives

$$\boldsymbol{x}^T \in \mathbb{R}^{1\times 5}$$

$$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times 7} \qquad \boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times 7}$$

$$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times 4} \qquad \boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times 4}$$

$$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times 3} \qquad \boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times 3}$$

$$\boldsymbol{y}^T := Y[1, :]$$

$$\boldsymbol{a}^{(3)^T} := A^{(3)}[1, :]$$

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

- $$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \qquad \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = \boldsymbol{a}^{(3)^T} - \boldsymbol{y}^T$$

Let’s calculate the derivatives

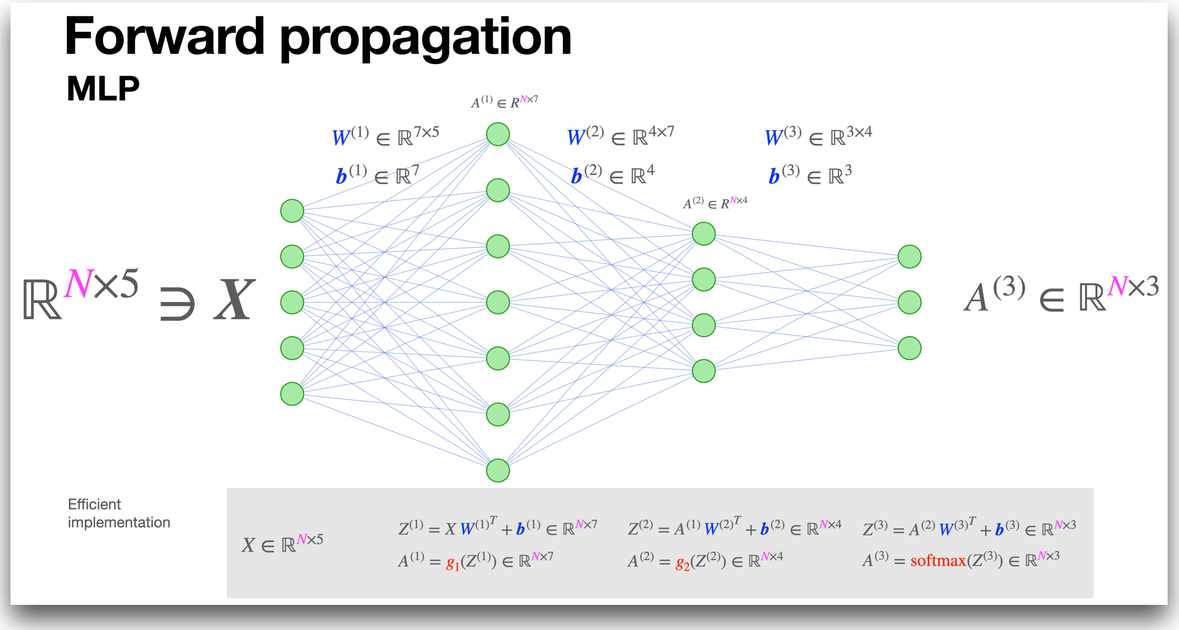
- Cross entropy loss

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right)$$

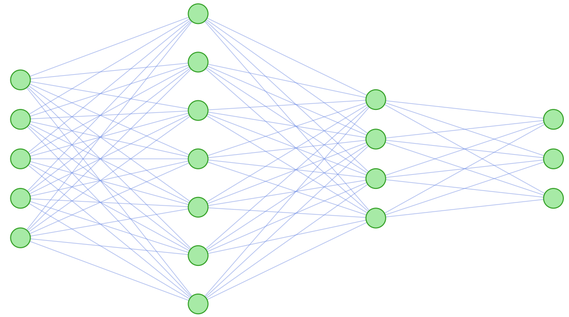
- Derivatives w.r.t. to the NN output

$$\frac{\partial L}{\partial A^{(3)}} = -\frac{1}{N}\, Y \div A^{(3)}$$

$$\frac{\partial L}{\partial Z^{(3)}} = -\frac{1}{N}(A^{(3)} - Y)$$



Live demo (PyTorch)



Multilayer perceptron

```
X = torch.tensor(
    [
        [-8.0, 1.0, -3.0, 5.0, 2.0],
        [-2.0, 2.0, -1.0, 5.0, 2.0],
        [7.0, 1.0, 9.0, -5.0, -2.0],
        [3.0, 1.0, 9.0,  5.0,  2.0]
    ])
X
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor([[-8.,  1., -3.,  5.,  2.],
        [-2.,  2., -1.,  5.,  2.],
        [ 7.,  1.,  9., -5., -2.],
        [ 3.,  1.,  9.,  5.,  2.]])
```

```
X.shape
```

executed in 9ms, finished 14:08:51 2023-03-12

```
torch.Size([4, 5])
```

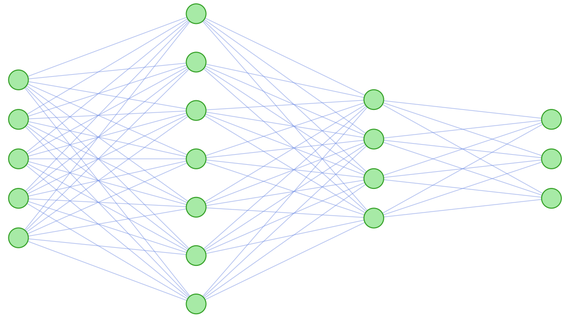
```
Y = torch.tensor(
    [
        [0.0, 0.0, 1.0],
        [1.0, 0.0, 0.0],
        [0.0, 1.0, 0.0],
        [0.0, 0.0, 1.0]
    ])
```

executed in 5ms, finished 14:08:51 2023-03-12

```
# Model parameters

W1 = torch.rand(7, 5)  #  R^{7 x 5}
b1 = torch.rand(7)     #  R^7

W2 = torch.rand(4, 7)  #  R^{4 x 7}
b2 = torch.rand(4)     #  R^4

W3 = torch.rand(3, 4)  #  R^{3 x 4}
b3 = torch.rand(3)     #  R^3
```

executed in 4ms, finished 14:08:51 2023-03-12

Live demo (PyTorch)



Multilayer perceptron

```
X = torch.tensor(
    [
        [-8.0, 1.0, -3.0, 5.0, 2.0],
        [-2.0, 2.0, -1.0, 5.0, 2.0],
        [7.0, 1.0, 9.0, -5.0, -2.0],
        [3.0, 1.0, 9.0,  5.0,  2.0]
    ])
X
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor([[-8.,  1., -3.,  5.,  2.],
        [-2.,  2., -1.,  5.,  2.],
        [ 7.,  1.,  9., -5., -2.],
        [ 3.,  1.,  9.,  5.,  2.]])
```

```
X.shape
```

executed in 9ms, finished 14:08:51 2023-03-12

```
torch.Size([4, 5])
```

```
Y = torch.tensor(
    [
        [0.0, 0.0, 1.0],
        [1.0, 0.0, 0.0],
        [0.0, 1.0, 0.0],
        [0.0, 0.0, 1.0]
    ])
```

executed in 5ms, finished 14:08:51 2023-03-12

```
# Model parameters

W1 = torch.rand(7, 5)  #  R^{7 x 5}
b1 = torch.rand(7)     #  R^7

W2 = torch.rand(4, 7)  #  R^{4 x 7}
b2 = torch.rand(4)     #  R^4

W3 = torch.rand(3, 4)  #  R^{3 x 4}
b3 = torch.rand(3)     #  R^3
```

executed in 4ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 8ms, finished 14:08:51 2023-03-12

```
A1.shape, A2.shape, A3.shape
```

executed in 4ms, finished 14:08:51 2023-03-12

```
(torch.Size([4, 7]), torch.Size([4, 4]), torch.Size([4, 3]))
```

```
A3
```

executed in 5ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]])
```

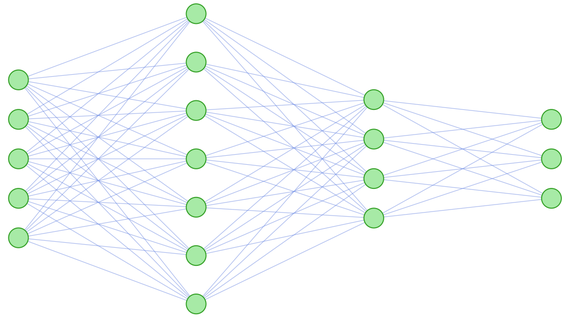
```
# Cross entropy loss

N = len(X)
L = 1/N * torch.sum( Y * A3 )
L
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor(0.3374)
```

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right)$$

Live demo (PyTorch)



Multilayer perceptron

```
X = torch.tensor(
    [
        [-8.0, 1.0, -3.0, 5.0, 2.0],
        [-2.0, 2.0, -1.0, 5.0, 2.0],
        [7.0, 1.0, 9.0, -5.0, -2.0],
        [3.0, 1.0, 9.0,  5.0,  2.0]
    ])
X
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor([[-8.,  1., -3.,  5.,  2.],
        [-2.,  2., -1.,  5.,  2.],
        [ 7.,  1.,  9., -5., -2.],
        [ 3.,  1.,  9.,  5.,  2.]])
```

```
X.shape
```

executed in 9ms, finished 14:08:51 2023-03-12

```
torch.Size([4, 5])
```

```
Y = torch.tensor(
    [
        [0.0, 0.0, 1.0],
        [1.0, 0.0, 0.0],
        [0.0, 1.0, 0.0],
        [0.0, 0.0, 1.0]
    ])
```

executed in 5ms, finished 14:08:51 2023-03-12

```
# Model parameters

W1 = torch.rand(7, 5)  #  R^{7 x 5}
b1 = torch.rand(7)     #  R^7

W2 = torch.rand(4, 7)  #  R^{4 x 7}
b2 = torch.rand(4)     #  R^4

W3 = torch.rand(3, 4)  #  R^{3 x 4}
b3 = torch.rand(3)     #  R^3
```

executed in 4ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 8ms, finished 14:08:51 2023-03-12

```
A1.shape, A2.shape, A3.shape
```

executed in 4ms, finished 14:08:51 2023-03-12

```
(torch.Size([4, 7]), torch.Size([4, 4]), torch.Size([4, 3]))
```

```
A3
```

executed in 5ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]])
```

```
# Cross entropy loss

N = len(X)
L = 1/N * torch.sum( Y * A3 )
L
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor(0.3374)
```

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right)$$

```
# W1 = torch.tensor( [ .... ], requires_grad=True )

W1 = W1.clone().requires_grad_(True)
W2 = W2.clone().requires_grad_(True)
W3 = W3.clone().requires_grad_(True)
b1 = b1.clone().requires_grad_(True)
b2 = b2.clone().requires_grad_(True)
b3 = b3.clone().requires_grad_(True)
```

executed in 2ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 3ms, finished 14:08:51 2023-03-12

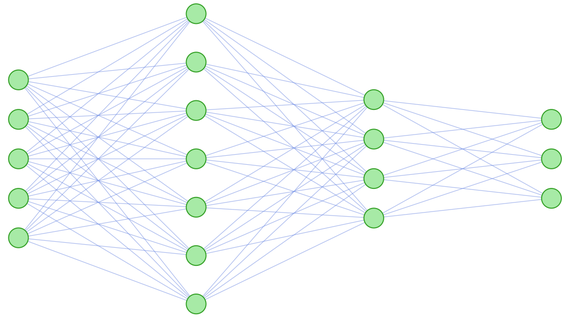
```
A3
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]], grad_fn=<SoftmaxBackward0>)
```

Note the result with `grad_fn=<SoftmaxBackward0>`

Live demo (PyTorch)



Multilayer perceptron

```
X = torch.tensor(
    [
        [-8.0, 1.0, -3.0, 5.0, 2.0],
        [-2.0, 2.0, -1.0, 5.0, 2.0],
        [7.0, 1.0, 9.0, -5.0, -2.0],
        [3.0, 1.0, 9.0,  5.0,  2.0]
    ])
X
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor([[-8.,  1., -3.,  5.,  2.],
        [-2.,  2., -1.,  5.,  2.],
        [ 7.,  1.,  9., -5., -2.],
        [ 3.,  1.,  9.,  5.,  2.]])
```

```
X.shape
```

executed in 9ms, finished 14:08:51 2023-03-12

```
torch.Size([4, 5])
```

```
Y = torch.tensor(
    [
        [0.0, 0.0, 1.0],
        [1.0, 0.0, 0.0],
        [0.0, 1.0, 0.0],
        [0.0, 0.0, 1.0]
    ])
```

executed in 5ms, finished 14:08:51 2023-03-12

```
# Model parameters

W1 = torch.rand(7, 5)  #  R^{7 x 5}
b1 = torch.rand(7)     #  R^7

W2 = torch.rand(4, 7)  #  R^{4 x 7}
b2 = torch.rand(4)     #  R^4

W3 = torch.rand(3, 4)  #  R^{3 x 4}
b3 = torch.rand(3)     #  R^3
```

executed in 4ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 8ms, finished 14:08:51 2023-03-12

```
A1.shape, A2.shape, A3.shape
```

executed in 4ms, finished 14:08:51 2023-03-12

```
(torch.Size([4, 7]), torch.Size([4, 4]), torch.Size([4, 3]))
```

```
A3
```

executed in 5ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]])
```

```
# Cross entropy loss

N = len(X)
L = 1/N * torch.sum( Y * A3 )
L
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor(0.3374)
```

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right)$$

```
# W1 = torch.tensor( [ .... ], requires_grad=True )

W1 = W1.clone().requires_grad_(True)
W2 = W2.clone().requires_grad_(True)
W3 = W3.clone().requires_grad_(True)
b1 = b1.clone().requires_grad_(True)
b2 = b2.clone().requires_grad_(True)
b3 = b3.clone().requires_grad_(True)
```

executed in 2ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 3ms, finished 14:08:51 2023-03-12

```
A3
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]], grad_fn=<SoftmaxBackward0>)
```

Note the result with `grad_fn=<SoftmaxBackward0>`

```
L = - 1/N * torch.sum( Y * torch.log(A3) );
L
```

executed in 2ms, finished 14:08:51 2023-03-12

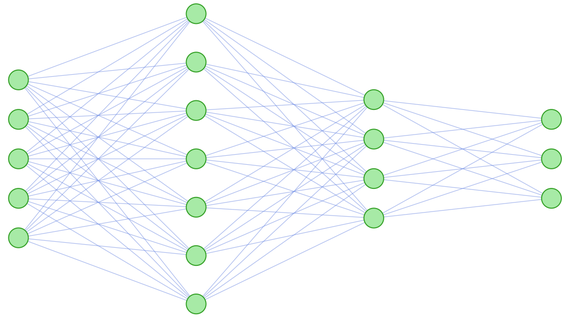
```
tensor(1.1394, grad_fn=<MulBackward0>)
```

Note the result with `grad_fn=<MulBackward0>`

```
Z1.retain_grad(); A1.retain_grad()
Z2.retain_grad(); A2.retain_grad()
Z3.retain_grad(); A3.retain_grad()

L.backward()
```

executed in 9ms, finished 14:08:51 2023-03-12

Live demo (PyTorch)



Multilayer perceptron

```
X = torch.tensor(
    [
        [-8.0, 1.0, -3.0, 5.0, 2.0],
        [-2.0, 2.0, -1.0, 5.0, 2.0],
        [7.0, 1.0, 9.0, -5.0, -2.0],
        [3.0, 1.0, 9.0,  5.0,  2.0]
    ])
X
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor([[-8.,  1., -3.,  5.,  2.],
        [-2.,  2., -1.,  5.,  2.],
        [ 7.,  1.,  9., -5., -2.],
        [ 3.,  1.,  9.,  5.,  2.]])
```

```
X.shape
```

executed in 9ms, finished 14:08:51 2023-03-12

```
torch.Size([4, 5])
```

```
Y = torch.tensor(
    [
        [0.0, 0.0, 1.0],
        [1.0, 0.0, 0.0],
        [0.0, 1.0, 0.0],
        [0.0, 0.0, 1.0]
    ])
```

executed in 5ms, finished 14:08:51 2023-03-12

```
# Model parameters

W1 = torch.rand(7, 5)  #  R^{7 x 5}
b1 = torch.rand(7)     #  R^7

W2 = torch.rand(4, 7)  #  R^{4 x 7}
b2 = torch.rand(4)     #  R^4

W3 = torch.rand(3, 4)  #  R^{3 x 4}
b3 = torch.rand(3)     #  R^3
```

executed in 4ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 8ms, finished 14:08:51 2023-03-12

```
A1.shape, A2.shape, A3.shape
```

executed in 4ms, finished 14:08:51 2023-03-12

```
(torch.Size([4, 7]), torch.Size([4, 4]), torch.Size([4, 3]))
```

```
A3
```

executed in 5ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]])
```

```
# Cross entropy loss

N = len(X)
L = 1/N * torch.sum( Y * A3 )
L
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor(0.3374)
```

$$L = -\frac{1}{N}\,\text{sum}\left(Y \otimes \log A^{(3)}\right)$$

```
# W1 = torch.tensor( [ .... ], requires_grad=True )

W1 = W1.clone().requires_grad_(True)
W2 = W2.clone().requires_grad_(True)
W3 = W3.clone().requires_grad_(True)
b1 = b1.clone().requires_grad_(True)
b2 = b2.clone().requires_grad_(True)
b3 = b3.clone().requires_grad_(True)
```

executed in 2ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 3ms, finished 14:08:51 2023-03-12

```
A3
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]], grad_fn=<SoftmaxBackward0>)
```

Note the result with `grad_fn=<SoftmaxBackward0>`

```
L = - 1/N * torch.sum( Y * torch.log(A3) );
L
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor(1.1394, grad_fn=<MulBackward0>)
```

Note the result with `grad_fn=<MulBackward0>`

```
Z1.retain_grad(); A1.retain_grad()
Z2.retain_grad(); A2.retain_grad()
Z3.retain_grad(); A3.retain_grad()

L.backward()
```

executed in 9ms, finished 14:08:51 2023-03-12

$$\frac{\partial L}{\partial A^{(3)}} = -\frac{1}{N}\,Y \div A^{(3)}$$

```
- 1/N * Y / A3
```

executed in 4ms, finished 14:08:51 2023-03-12

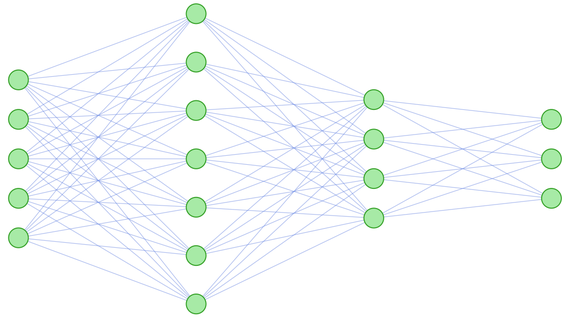
```
tensor([[-0.0000, -0.0000, -0.7133],
        [-0.5224, -0.0000, -0.0000],
        [-0.0000, -1.3285, -0.0000],
        [-0.0000, -0.0000, -0.7523]], grad_fn=<DivBackward0>)
```

```
A3.grad
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor([[-0.0000, -0.0000, -0.7133],
        [-0.5224, -0.0000, -0.0000],
        [-0.0000, -1.3285, -0.0000],
        [-0.0000, -0.0000, -0.7523]])
```

Live demo (PyTorch)



Multilayer perceptron

```
X = torch.tensor(
    [
        [-8.0, 1.0, -3.0, 5.0, 2.0],
        [-2.0, 2.0, -1.0, 5.0, 2.0],
        [7.0, 1.0, 9.0, -5.0, -2.0],
        [3.0, 1.0, 9.0,  5.0,  2.0]
    ])
X
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor([[-8.,  1., -3.,  5.,  2.],
        [-2.,  2., -1.,  5.,  2.],
        [ 7.,  1.,  9., -5., -2.],
        [ 3.,  1.,  9.,  5.,  2.]])
```

```
X.shape
```

executed in 9ms, finished 14:08:51 2023-03-12

```
torch.Size([4, 5])
```

```
Y = torch.tensor(
    [
        [0.0, 0.0, 1.0],
        [1.0, 0.0, 0.0],
        [0.0, 1.0, 0.0],
        [0.0, 0.0, 1.0]
    ])
```

executed in 5ms, finished 14:08:51 2023-03-12

```
# Model parameters

W1 = torch.rand(7, 5)  #  R^{7 x 5}
b1 = torch.rand(7)     #  R^7

W2 = torch.rand(4, 7)  #  R^{4 x 7}
b2 = torch.rand(4)     #  R^4

W3 = torch.rand(3, 4)  #  R^{3 x 4}
b3 = torch.rand(3)     #  R^3
```

executed in 4ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 8ms, finished 14:08:51 2023-03-12

```
A1.shape, A2.shape, A3.shape
```

executed in 4ms, finished 14:08:51 2023-03-12

```
(torch.Size([4, 7]), torch.Size([4, 4]), torch.Size([4, 3]))
```

```
A3
```

executed in 5ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]])
```

```
# Cross entropy loss

N = len(X)
L = 1/N * torch.sum( Y * A3 )
L
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor(0.3374)
```

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right)$$

```
# W1 = torch.tensor( [ .... ], requires_grad=True )

W1 = W1.clone().requires_grad_(True)
W2 = W2.clone().requires_grad_(True)
W3 = W3.clone().requires_grad_(True)
b1 = b1.clone().requires_grad_(True)
b2 = b2.clone().requires_grad_(True)
b3 = b3.clone().requires_grad_(True)
```

executed in 2ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 3ms, finished 14:08:51 2023-03-12

```
A3
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]], grad_fn=<SoftmaxBackward0>)
```

Note the result with `grad_fn=<SoftmaxBackward0>`

```
L = - 1/N * torch.sum( Y * torch.log(A3) );
L
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor(1.1394, grad_fn=<MulBackward0>)
```

Note the result with `grad_fn=<MulBackward0>`

```
Z1.retain_grad(); A1.retain_grad()
Z2.retain_grad(); A2.retain_grad()
Z3.retain_grad(); A3.retain_grad()

L.backward()
```

executed in 9ms, finished 14:08:51 2023-03-12

$$\frac{\partial L}{\partial A^{(3)}} = -\frac{1}{N}\,Y \div A^{(3)}$$

```
- 1/N * Y / A3
```

executed in 4ms, finished 14:08:51 2023-03-12

```
tensor([[-0.0000, -0.0000, -0.7133],
        [-0.5224, -0.0000, -0.0000],
        [-0.0000, -1.3285, -0.0000],
        [-0.0000, -0.0000, -0.7523]], grad_fn=<DivBackward0>)
```

```
A3.grad
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor([[-0.0000, -0.0000, -0.7133],
        [-0.5224, -0.0000, -0.0000],
        [-0.0000, -1.3285, -0.0000],
        [-0.0000, -0.0000, -0.7523]])
```

$$\frac{\partial L}{\partial Z^{(3)}} = -\frac{1}{N}(A^{(3)} - Y)$$

```
1/N * (A3-Y)
```

executed in 3ms, finished 14:08:51 2023-03-12

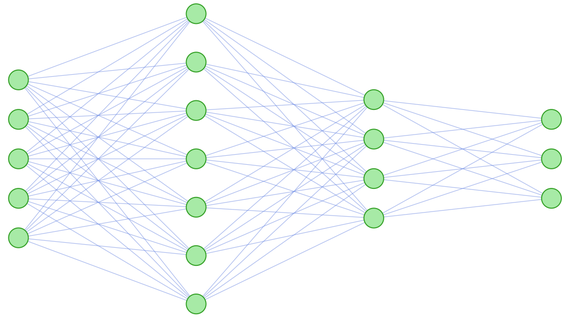
```
tensor([[ 0.1090,  0.0534, -0.1624],
        [-0.1304,  0.0473,  0.0831],
        [ 0.1199, -0.2030,  0.0831],
        [ 0.1199,  0.0470, -0.1669]], grad_fn=<MulBackward0>)
```

```
Z3.grad
```

executed in 3ms, finished 14:08:51 2023-03-12

```
tensor([[ 0.1090,  0.0534, -0.1624],
        [-0.1304,  0.0473,  0.0831],
        [ 0.1199, -0.2030,  0.0831],
        [ 0.1199,  0.0470, -0.1669]])
```

Live demo (PyTorch)



Multilayer perceptron

```
X = torch.tensor(
    [
        [-8.0, 1.0, -3.0, 5.0, 2.0],
        [-2.0, 2.0, -1.0, 5.0, 2.0],
        [7.0, 1.0, 9.0, -5.0, -2.0],
        [3.0, 1.0, 9.0,  5.0,  2.0]
    ])
X
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor([[-8.,  1., -3.,  5.,  2.],
        [-2.,  2., -1.,  5.,  2.],
        [ 7.,  1.,  9., -5., -2.],
        [ 3.,  1.,  9.,  5.,  2.]])
```

```
X.shape
```

executed in 9ms, finished 14:08:51 2023-03-12

```
torch.Size([4, 5])
```

```
Y = torch.tensor(
    [
        [0.0, 0.0, 1.0],
        [1.0, 0.0, 0.0],
        [0.0, 1.0, 0.0],
        [0.0, 0.0, 1.0]
    ])
```

executed in 5ms, finished 14:08:51 2023-03-12

```
# Model parameters

W1 = torch.rand(7, 5)  #  R^{7 x 5}
b1 = torch.rand(7)     #  R^7

W2 = torch.rand(4, 7)  #  R^{4 x 7}
b2 = torch.rand(4)     #  R^4

W3 = torch.rand(3, 4)  #  R^{3 x 4}
b3 = torch.rand(3)     #  R^3
```

executed in 4ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 8ms, finished 14:08:51 2023-03-12

```
A1.shape, A2.shape, A3.shape
```

executed in 4ms, finished 14:08:51 2023-03-12

```
(torch.Size([4, 7]), torch.Size([4, 4]), torch.Size([4, 3]))
```

```
A3
```

executed in 5ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]])
```

```
# Cross entropy loss

N = len(X)
L = 1/N * torch.sum( Y * A3 )
L
```

executed in 11ms, finished 14:08:51 2023-03-12

```
tensor(0.3374)
```

$$L = -\frac{1}{N}\,\text{sum}\left(Y \otimes \log A^{(3)}\right)$$

Backprop →

```
# W1 = torch.tensor( [ .... ], requires_grad=True )

W1 = W1.clone().requires_grad_(True)
W2 = W2.clone().requires_grad_(True)
W3 = W3.clone().requires_grad_(True)
b1 = b1.clone().requires_grad_(True)
b2 = b2.clone().requires_grad_(True)
b3 = b3.clone().requires_grad_(True)
```

executed in 2ms, finished 14:08:51 2023-03-12

```
# Forward pass
Z1 = torch.mm(X, W1.T) + b1
A1 = torch.sigmoid(Z1)

Z2 = torch.mm(A1, W2.T) + b2
A2 = torch.sigmoid(Z2)

Z3 = torch.mm(A2, W3.T) + b3
A3 = torch.softmax(Z3, dim=1)   # Note: use dim=1
```

executed in 3ms, finished 14:08:51 2023-03-12

```
A3
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor([[0.4358, 0.2137, 0.3505],
        [0.4785, 0.1890, 0.3325],
        [0.4795, 0.1882, 0.3324],
        [0.4797, 0.1880, 0.3323]], grad_fn=<SoftmaxBackward0>)
```

Note the result with `grad_fn=<SoftmaxBackward0>`

```
L = - 1/N * torch.sum( Y * torch.log(A3) );
L
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor(1.1394, grad_fn=<MulBackward0>)
```

Note the result with `grad_fn=<MulBackward0>`

```
Z1.retain_grad(); A1.retain_grad()
Z2.retain_grad(); A2.retain_grad()
Z3.retain_grad(); A3.retain_grad()

L.backward()
```

executed in 9ms, finished 14:08:51 2023-03-12

$$\frac{\partial L}{\partial A^{(3)}} = -\frac{1}{N}\,Y \div A^{(3)}$$

```
- 1/N * Y / A3
```

executed in 4ms, finished 14:08:51 2023-03-12

```
tensor([[-0.0000, -0.0000, -0.7133],
        [-0.5224, -0.0000, -0.0000],
        [-0.0000, -1.3285, -0.0000],
        [-0.0000, -0.0000, -0.7523]], grad_fn=<DivBackward0>)
```

```
A3.grad
```

executed in 2ms, finished 14:08:51 2023-03-12

```
tensor([[-0.0000, -0.0000, -0.7133],
        [-0.5224, -0.0000, -0.0000],
        [-0.0000, -1.3285, -0.0000],
        [-0.0000, -0.0000, -0.7523]])
```

$$\frac{\partial L}{\partial Z^{(3)}} = -\frac{1}{N}(A^{(3)} - Y)$$

```
1/N * (A3-Y)
```

executed in 3ms, finished 14:08:51 2023-03-12

```
tensor([[ 0.1090,  0.0534, -0.1624],
        [-0.1304,  0.0473,  0.0831],
        [ 0.1199, -0.2030,  0.0831],
        [ 0.1199,  0.0470, -0.1669]], grad_fn=<MulBackward0>)
```

```
Z3.grad
```

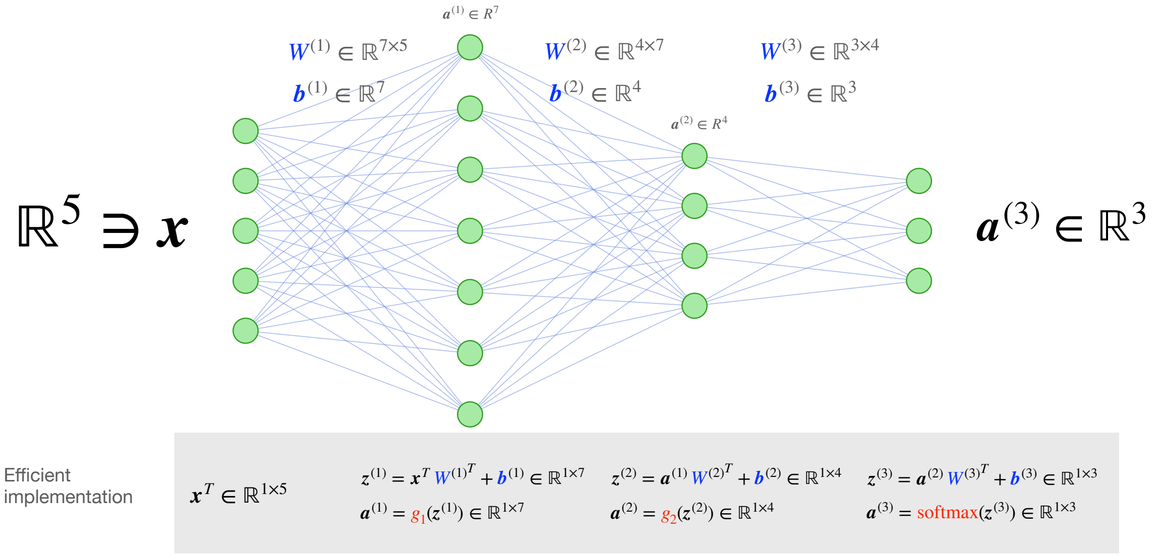
executed in 3ms, finished 14:08:51 2023-03-12

```
tensor([[ 0.1090,  0.0534, -0.1624],
        [-0.1304,  0.0473,  0.0831],
        [ 0.1199, -0.2030,  0.0831],
        [ 0.1199,  0.0470, -0.1669]])
```

Backward propagation

MLP

$$L_1 := -\operatorname{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\operatorname{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$

$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$

$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$

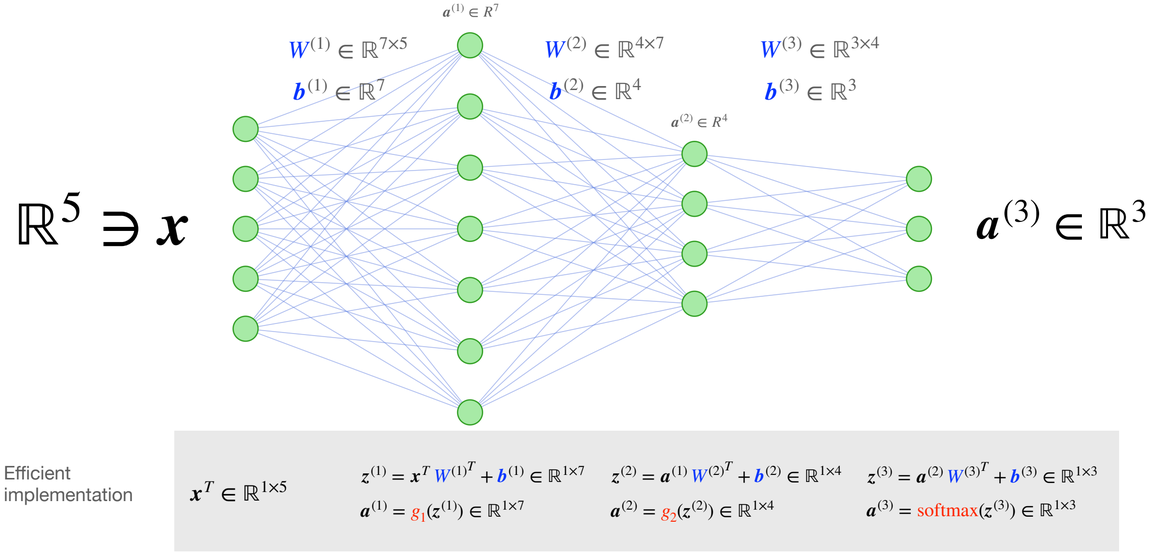
$\boldsymbol{a}^{(3)} = \operatorname{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

Backward propagation

MLP

$$L_1 := -\operatorname{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\operatorname{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \in \mathbb{R}^{1\times 3}$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times 5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times 7}$

$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times 7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times 4}$

$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times 4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times 3}$

$\boldsymbol{a}^{(3)} = \operatorname{softmax}(z^{(3)}) \in \mathbb{R}^{1\times 3}$

Backward propagation

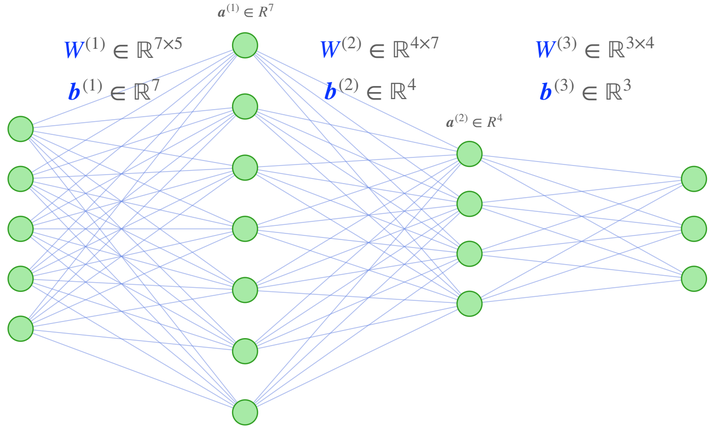
MLP

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \in \mathbb{R}^{1\times 3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = \boldsymbol{a}^{(3)^T} - \boldsymbol{y}^T \in \mathbb{R}^{1\times 3}$$

$\mathbb{R}^5 \ni \boldsymbol{x}$



$\boldsymbol{a}^{(3)} \in \mathbb{R}^3$

Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times 5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times 7}$

$\boldsymbol{a}^{(1)} = g_1(\boldsymbol{z}^{(1)}) \in \mathbb{R}^{1\times 7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times 4}$

$\boldsymbol{a}^{(2)} = g_2(\boldsymbol{z}^{(2)}) \in \mathbb{R}^{1\times 4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times 3}$

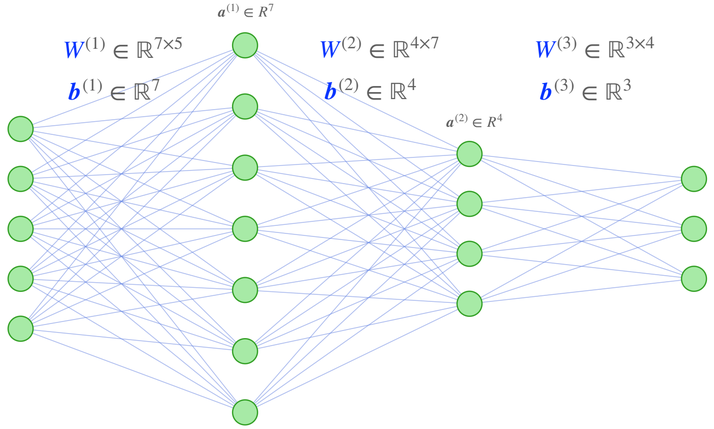
$\boldsymbol{a}^{(3)} = \text{softmax}(\boldsymbol{z}^{(3)}) \in \mathbb{R}^{1\times 3}$

Backward propagation

MLP

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

$$\mathbb{R}^5 \ni \boldsymbol{x}$$



$$\boldsymbol{a}^{(3)} \in \mathbb{R}^3$$

Efficient implementation

$$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$$

$$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7} \qquad \boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$$

$$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4} \qquad \boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$$

$$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3} \qquad \boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \in \mathbb{R}^{1\times3}$$

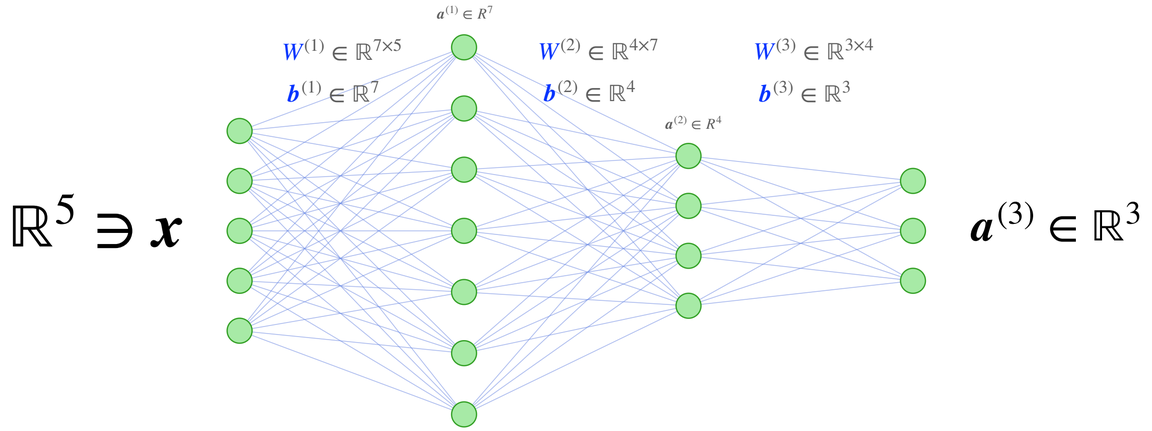
$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = \boldsymbol{a}^{(3)^T} - \boldsymbol{y}^T \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} \cdot \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} \cdot W^{(3)}$$

Backward propagation

MLP

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = \boldsymbol{a}^{(3)^T} - \boldsymbol{y}^T \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} \cdot \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} \cdot W^{(3)}$$

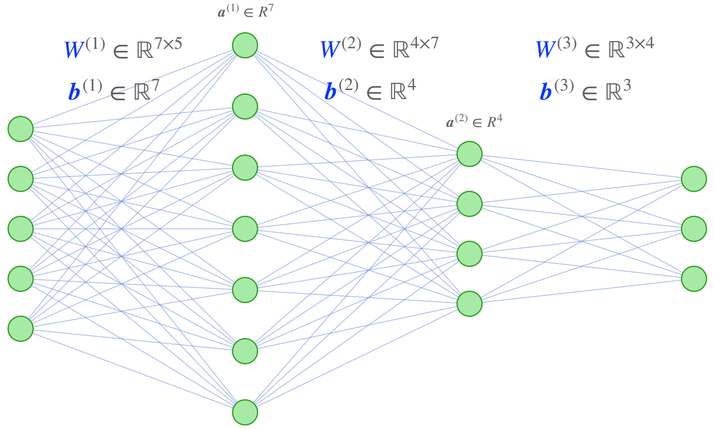
$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \cdot \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \otimes \frac{\partial g_2}{\partial \boldsymbol{z}^{(2)}}$$

Backward propagation

MLP

$$L_1 := -\operatorname{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\operatorname{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

$\mathbb{R}^5 \ni \boldsymbol{x}$



$\boldsymbol{a}^{(3)} \in \mathbb{R}^3$

Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(\boldsymbol{z}^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(\boldsymbol{z}^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(\boldsymbol{z}^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = \boldsymbol{a}^{(3)^T} - \boldsymbol{y}^T \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} \cdot \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} \cdot W^{(3)}$$

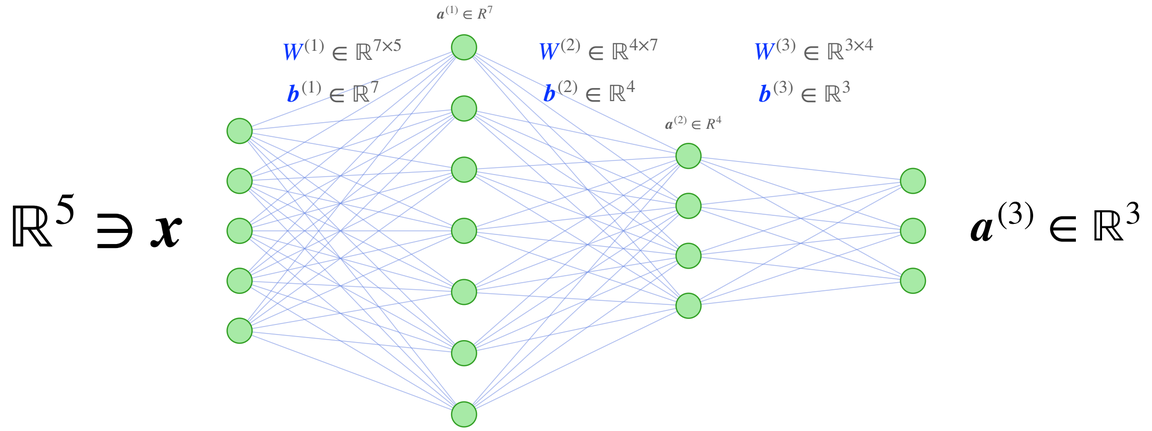
$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \cdot \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \otimes \frac{\partial g_2}{\partial \boldsymbol{z}^{(2)}}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} \cdot \frac{\partial \boldsymbol{z}^{(2)}}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} \cdot W^{(2)}$$

Backward propagation

MLP

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = \boldsymbol{a}^{(3)^T} - \boldsymbol{y}^T \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} \cdot \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} \cdot W^{(3)}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \cdot \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \otimes \frac{\partial g_2}{\partial \boldsymbol{z}^{(2)}}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} \cdot \frac{\partial \boldsymbol{z}^{(2)}}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} \cdot W^{(2)}$$

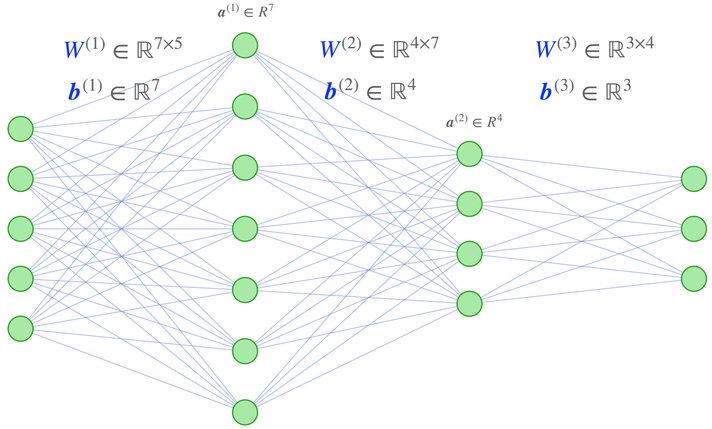
$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} \cdot \frac{\partial \boldsymbol{a}^{(1)}}{\partial \boldsymbol{z}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} \otimes \frac{\partial g_1}{\partial \boldsymbol{z}^{(1)}}$$

Backward propagation

MLP

$$L_1 := -\operatorname{sum}(Y[1,:] \otimes \log A^{(3)}[1,:]) = -\operatorname{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

$\mathbb{R}^5 \ni \boldsymbol{x}$



$\boldsymbol{a}^{(3)} \in \mathbb{R}^3$

Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(\boldsymbol{z}^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(\boldsymbol{z}^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(\boldsymbol{z}^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = \boldsymbol{a}^{(3)^T} - \boldsymbol{y}^T \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial W^{(3)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}}^T \cdot \frac{\partial \boldsymbol{z}^{(3)^T}}{\partial W^{(3)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}}^T \cdot \boldsymbol{a}^{(2)^T}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} \cdot \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} \cdot W^{(3)}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \cdot \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \otimes \frac{\partial g_2}{\partial \boldsymbol{z}^{(2)}}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} \cdot \frac{\partial \boldsymbol{z}^{(2)}}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} \cdot W^{(2)}$$

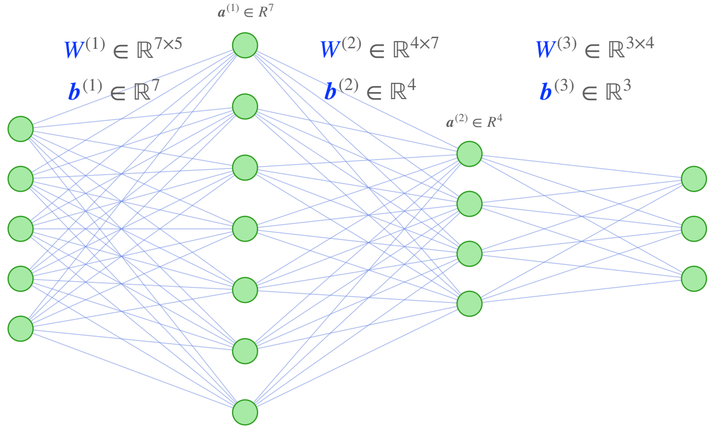
$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} \cdot \frac{\partial \boldsymbol{a}^{(1)}}{\partial \boldsymbol{z}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} \otimes \frac{\partial g_1}{\partial \boldsymbol{z}^{(1)}}$$

Backward propagation

MLP

$$L_1 := -\operatorname{sum}(Y[1,:] \otimes \log A^{(3)}[1,:]) = -\operatorname{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

$\mathbb{R}^5 \ni \boldsymbol{x}$



$\boldsymbol{a}^{(3)} \in \mathbb{R}^3$

Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = \boldsymbol{a}^{(3)^T} - \boldsymbol{y}^T \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \frac{\partial z^{(3)^T}}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \boldsymbol{a}^{(2)^T}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial z^{(3)}} \cdot W^{(3)}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \cdot \frac{\partial \boldsymbol{a}^{(2)}}{\partial z^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \otimes \frac{\partial g_2}{\partial z^{(2)}}$$

$$\frac{\partial L_1}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \frac{\partial z^{(2)}}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \boldsymbol{a}^{(1)^T}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial L_1}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial L_1}{\partial z^{(2)}} \cdot W^{(2)}$$

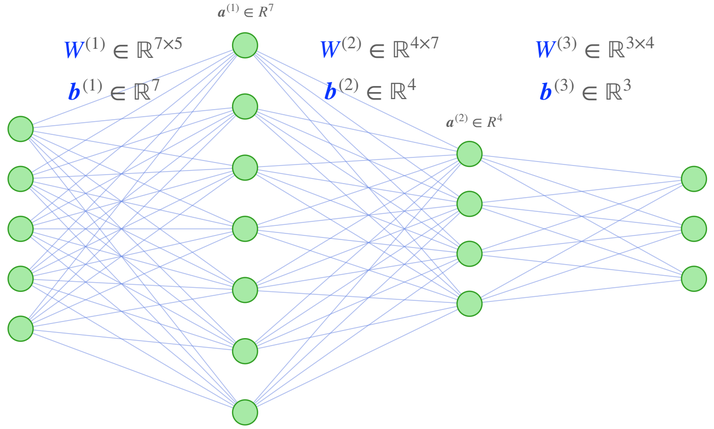
$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} \cdot \frac{\partial \boldsymbol{a}^{(1)}}{\partial z^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} \otimes \frac{\partial g_1}{\partial z^{(1)}}$$

Backward propagation

MLP

$$L_1 := -\operatorname{sum}(Y[1,:] \otimes \log A^{(3)}[1,:]) = -\operatorname{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$

$\mathbb{R}^5 \ni \boldsymbol{x}$



$\boldsymbol{a}^{(3)} \in \mathbb{R}^3$

Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(3)}} = -\boldsymbol{y}^T \div \boldsymbol{a}^{(3)^T} \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(3)}} = \boldsymbol{a}^{(3)^T} - \boldsymbol{y}^T \in \mathbb{R}^{1\times3}$$

$$\frac{\partial L_1}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \frac{\partial z^{(3)^T}}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \boldsymbol{a}^{(2)^T}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial L_1}{\partial z^{(3)}} \cdot W^{(3)}$$

$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \cdot \frac{\partial \boldsymbol{a}^{(2)}}{\partial z^{(2)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(2)}} \otimes \frac{\partial g_2}{\partial z^{(2)}}$$

$$\frac{\partial L_1}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \frac{\partial z^{(2)}}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \boldsymbol{a}^{(1)^T}$$

$$\frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial L_1}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial L_1}{\partial z^{(2)}} \cdot W^{(2)}$$

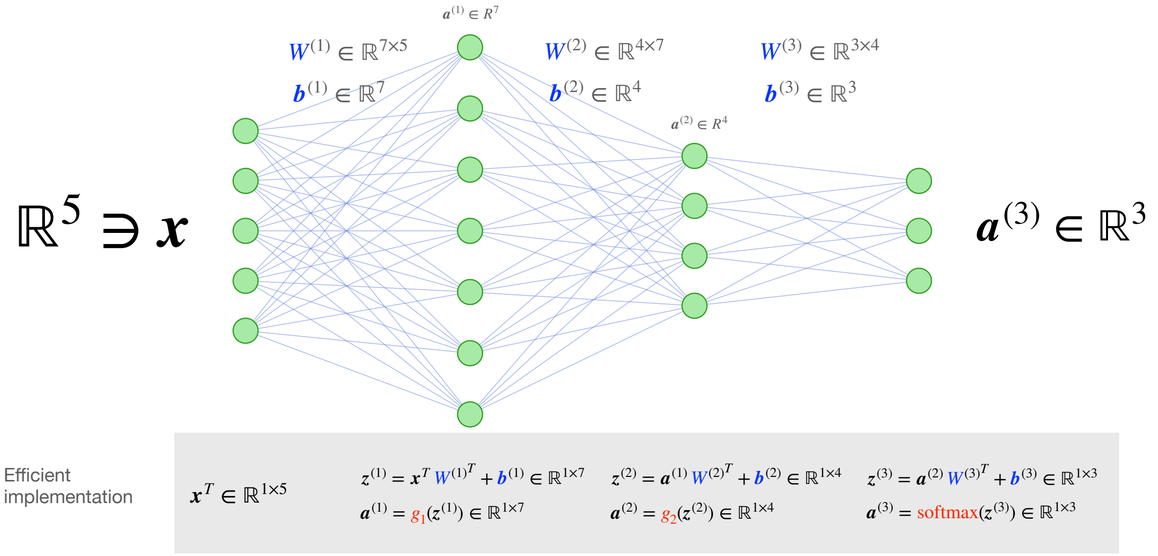
$$\frac{\partial L_1}{\partial \boldsymbol{z}^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} \cdot \frac{\partial \boldsymbol{a}^{(1)}}{\partial z^{(1)}} = \frac{\partial L_1}{\partial \boldsymbol{a}^{(1)}} \otimes \frac{\partial g_1}{\partial z^{(1)}}$$

$$\frac{\partial L_1}{\partial W^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}^T \cdot \frac{\partial z^{(1)}}{\partial W^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}^T \cdot \boldsymbol{x}^T$$

Backward propagation

MLP

$$L_1 := -\operatorname{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\operatorname{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$

$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$

$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

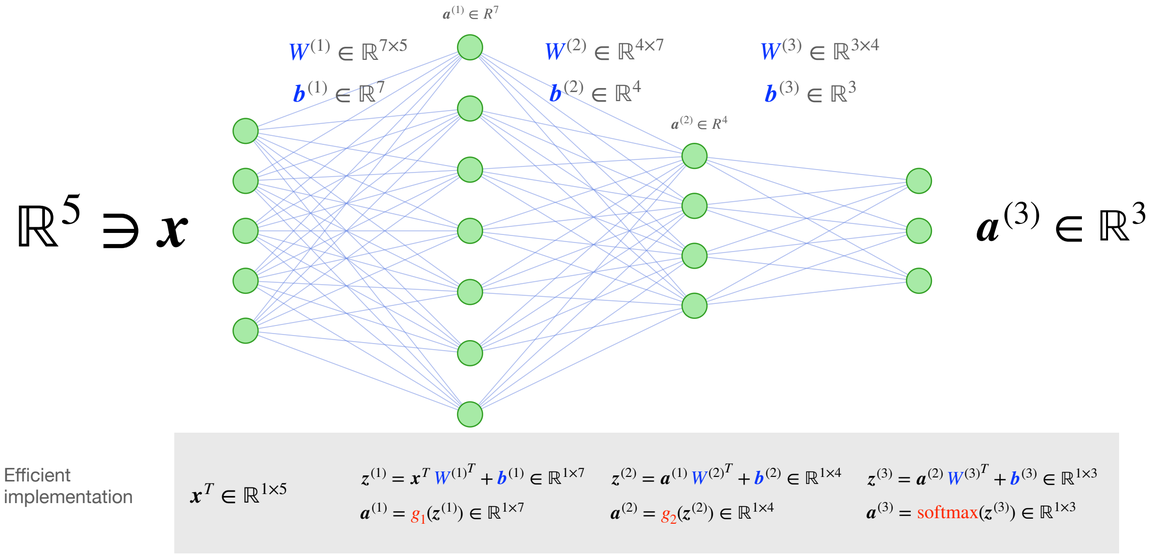
$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$

$\boldsymbol{a}^{(3)} = \operatorname{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

Backward propagation

MLP

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

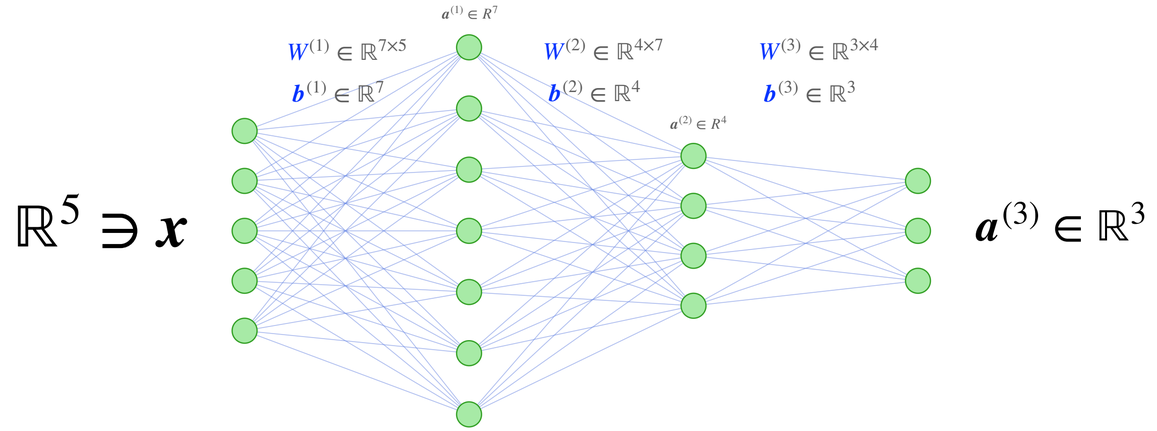
$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \frac{\partial z^{(3)^T}}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \boldsymbol{a}^{(2)^T}$$

Backward propagation

MLP

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

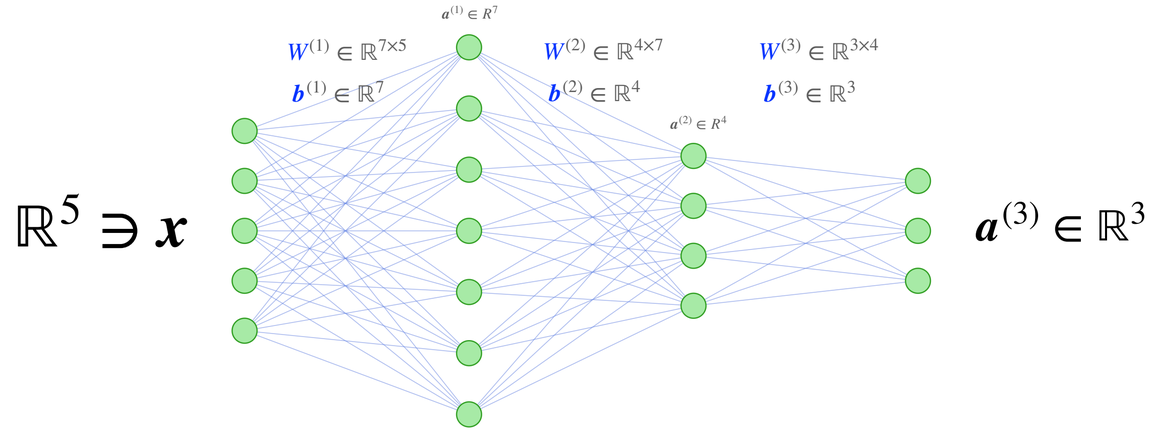
$$\frac{\partial L_1}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \frac{\partial z^{(3)^T}}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \boldsymbol{a}^{(2)^T}$$

$$\frac{\partial L_1}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \frac{\partial z^{(2)}}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \boldsymbol{a}^{(1)^T}$$

Backward propagation

MLP

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \frac{\partial z^{(3)^T}}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \boldsymbol{a}^{(2)^T}$$

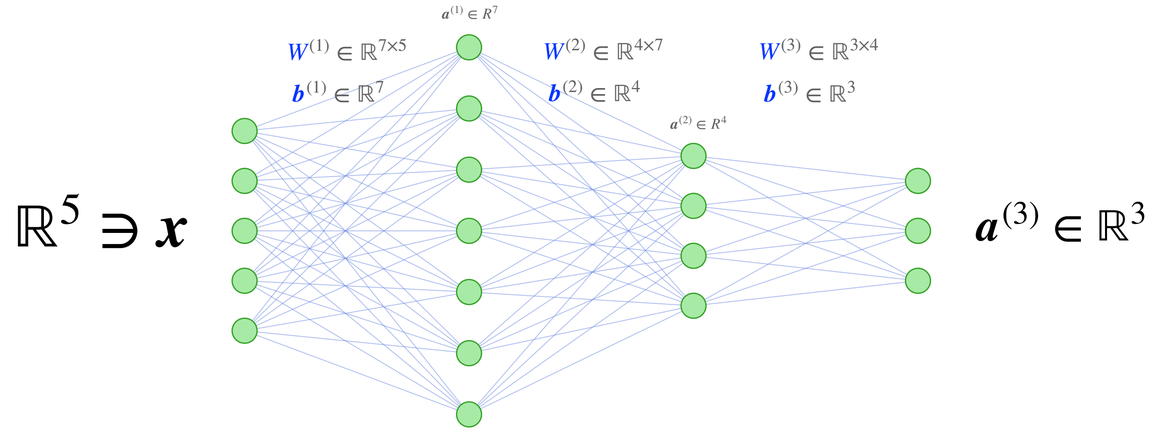
$$\frac{\partial L_1}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \frac{\partial z^{(2)}}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \boldsymbol{a}^{(1)^T}$$

$$\frac{\partial L_1}{\partial W^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}^T \cdot \frac{\partial z^{(1)}}{\partial W^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}^T \cdot \boldsymbol{x}^T$$

Backward propagation

MLP

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \frac{\partial z^{(3)^T}}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \boldsymbol{a}^{(2)^T}$$

$$\frac{\partial L_1}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \frac{\partial z^{(2)}}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \boldsymbol{a}^{(1)^T}$$

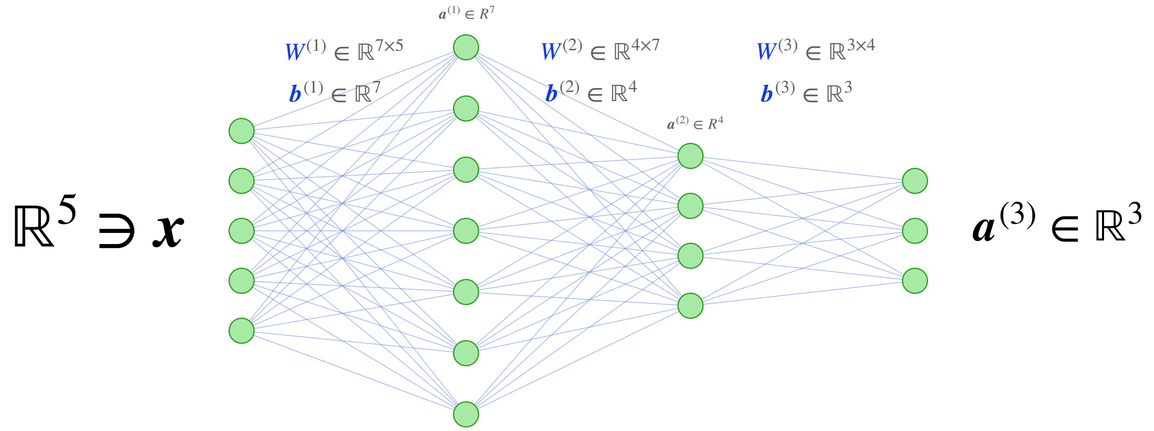
$$\frac{\partial L_1}{\partial W^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}^T \cdot \frac{\partial z^{(1)}}{\partial W^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}^T \cdot \boldsymbol{x}^T$$

$$\frac{\partial L_1}{\partial \boldsymbol{b}^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial \boldsymbol{b}^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}$$

Backward propagation

MLP

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \frac{\partial z^{(3)^T}}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \boldsymbol{a}^{(2)^T}$$

$$\frac{\partial L_1}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \frac{\partial z^{(2)}}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \boldsymbol{a}^{(1)^T}$$

$$\frac{\partial L_1}{\partial W^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}^T \cdot \frac{\partial z^{(1)}}{\partial W^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}^T \cdot \boldsymbol{x}^T$$

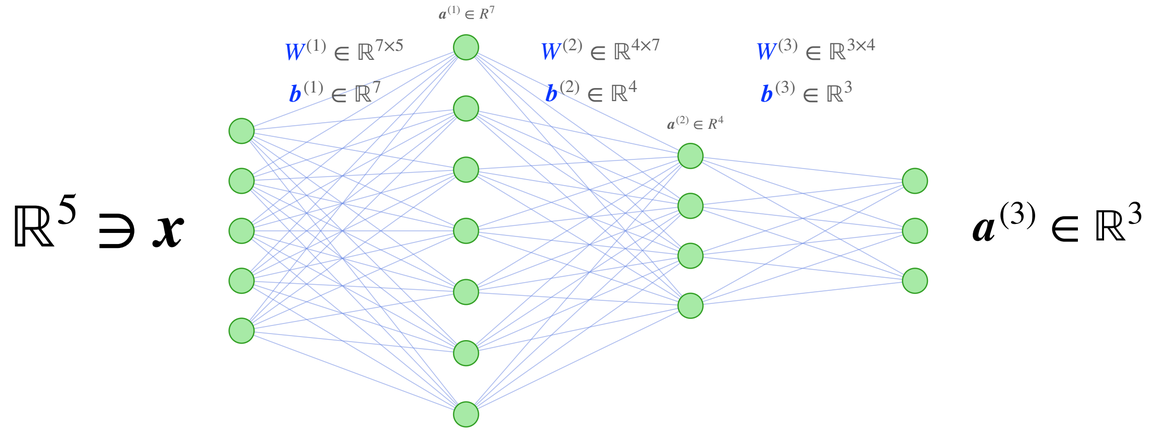
$$\frac{\partial L_1}{\partial \boldsymbol{b}^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial \boldsymbol{b}^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}$$

$$\frac{\partial L_1}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}$$

Backward propagation

MLP

$$L_1 := -\text{sum}(Y[1, :] \otimes \log A^{(3)}[1, :]) = -\text{sum}(\boldsymbol{y} \otimes \log \boldsymbol{a}^{(3)})$$



Efficient implementation

$\boldsymbol{x}^T \in \mathbb{R}^{1\times5}$

$z^{(1)} = \boldsymbol{x}^T W^{(1)^T} + \boldsymbol{b}^{(1)} \in \mathbb{R}^{1\times7}$
$\boldsymbol{a}^{(1)} = g_1(z^{(1)}) \in \mathbb{R}^{1\times7}$

$z^{(2)} = \boldsymbol{a}^{(1)} W^{(2)^T} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{1\times4}$
$\boldsymbol{a}^{(2)} = g_2(z^{(2)}) \in \mathbb{R}^{1\times4}$

$z^{(3)} = \boldsymbol{a}^{(2)} W^{(3)^T} + \boldsymbol{b}^{(3)} \in \mathbb{R}^{1\times3}$
$\boldsymbol{a}^{(3)} = \text{softmax}(z^{(3)}) \in \mathbb{R}^{1\times3}$

$$\frac{\partial L_1}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \frac{\partial z^{(3)^T}}{\partial W^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}^T \cdot \boldsymbol{a}^{(2)^T}$$

$$\frac{\partial L_1}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \frac{\partial z^{(2)}}{\partial W^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}^T \cdot \boldsymbol{a}^{(1)^T}$$

$$\frac{\partial L_1}{\partial W^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}^T \cdot \frac{\partial z^{(1)}}{\partial W^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}^T \cdot \boldsymbol{x}^T$$

$$\frac{\partial L_1}{\partial \boldsymbol{b}^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial \boldsymbol{b}^{(3)}} = \frac{\partial L_1}{\partial z^{(3)}}$$

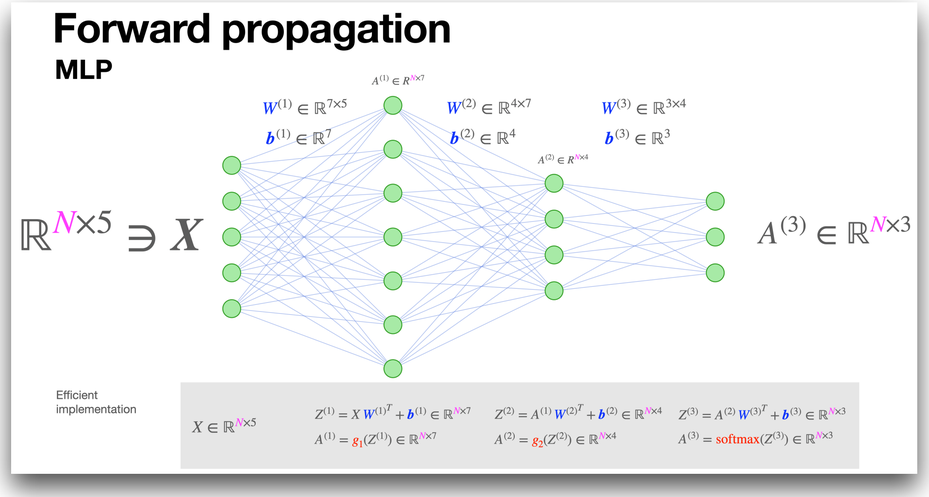
$$\frac{\partial L_1}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial L_1}{\partial z^{(2)}}$$

$$\frac{\partial L_1}{\partial \boldsymbol{b}^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}} \cdot \frac{\partial z^{(1)}}{\partial \boldsymbol{b}^{(1)}} = \frac{\partial L_1}{\partial z^{(1)}}$$

Backward propagation

MLP

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right) = -\frac{1}{N}\sum_{p=1}^{N} L_p$$

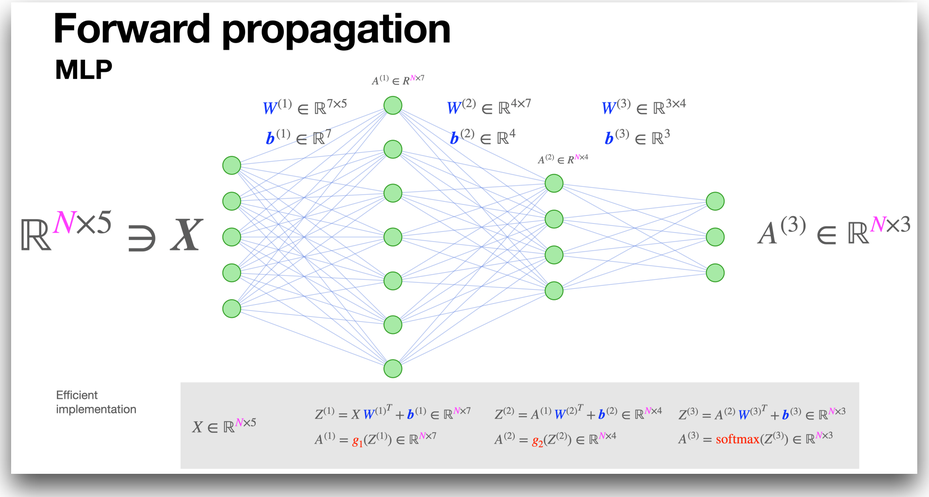


Backward propagation

MLP

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right) = -\frac{1}{N}\sum_{p=1}^{N} L_p$$

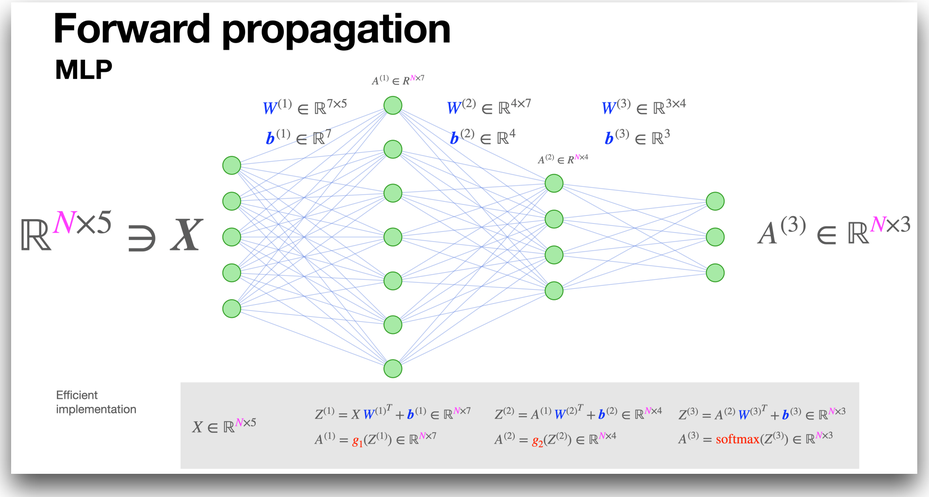
$$\frac{\partial L}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot \frac{\partial Z^{(3)}}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot Z^{(3)} \cdot \frac{\partial}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot A^{(2)}$$



Backward propagation

MLP

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right) = -\frac{1}{N}\sum_{p=1}^{N} L_p$$



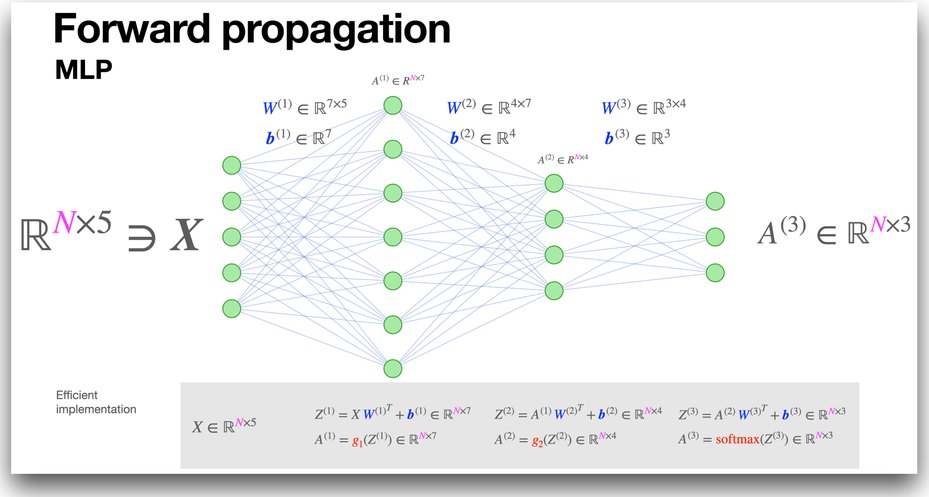
$$\frac{\partial L}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot \frac{\partial Z^{(3)}}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot Z^{(3)} \cdot \frac{\partial}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot A^{(2)}$$

$$\frac{\partial L}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot \frac{\partial Z^{(2)}}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot Z^{(2)} \cdot \frac{\partial}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot A^{(1)}$$

Backward propagation

MLP

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right) = -\frac{1}{N}\sum_{p=1}^{N} L_p$$



$$\frac{\partial L}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot \frac{\partial Z^{(3)}}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot Z^{(3)} \cdot \frac{\partial}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot A^{(2)}$$

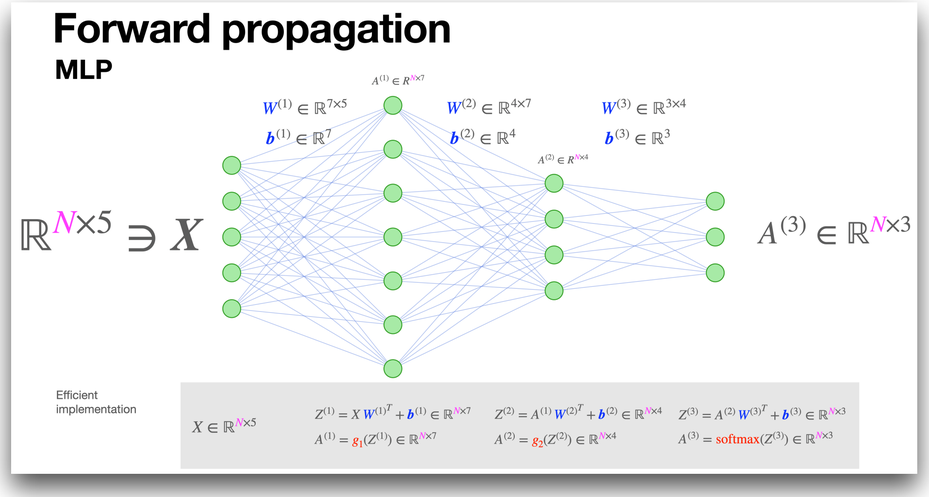
$$\frac{\partial L}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot \frac{\partial Z^{(2)}}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot Z^{(2)} \cdot \frac{\partial}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot A^{(1)}$$

$$\frac{\partial L}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot \frac{\partial Z^{(1)}}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot Z^{(1)} \cdot \frac{\partial}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot X$$

Backward propagation

MLP

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right) = -\frac{1}{N}\sum_{p=1}^{N} L_p$$



$$\frac{\partial L}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot \frac{\partial Z^{(3)}}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot Z^{(3)} \cdot \frac{\partial}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot A^{(2)}$$

$$\frac{\partial L}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot \frac{\partial Z^{(2)}}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot Z^{(2)} \cdot \frac{\partial}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot A^{(1)}$$

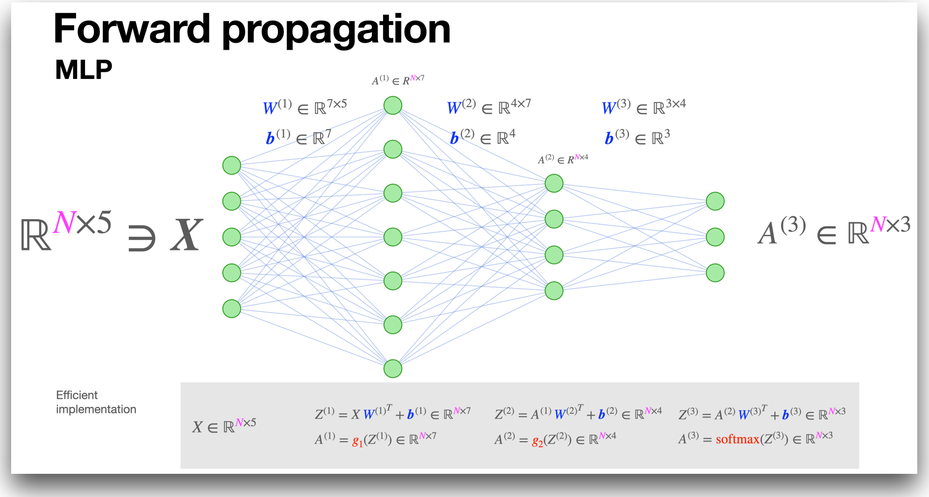
$$\frac{\partial L}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot \frac{\partial Z^{(1)}}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot Z^{(1)} \cdot \frac{\partial}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot X$$

$$\frac{\partial L_1}{\partial \boldsymbol{b}^{(3)}} = \frac{\partial L_1}{\partial Z^{(3)}}^{T} \cdot \frac{\partial Z^{(3)}}{\partial \boldsymbol{b}^{(3)}} = \frac{\partial L_1}{\partial Z^{(3)}}^{T} \cdot \mathbf{1}^N$$

Backward propagation

MLP

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right) = -\frac{1}{N}\sum_{p=1}^{N} L_p$$



$$\frac{\partial L}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot \frac{\partial Z^{(3)}}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot Z^{(3)} \cdot \frac{\partial}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot A^{(2)}$$

$$\frac{\partial L_1}{\partial b^{(3)}} = \frac{\partial L_1}{\partial Z^{(3)}}^{T} \cdot \frac{\partial Z^{(3)}}{\partial b^{(3)}} = \frac{\partial L_1}{\partial Z^{(3)}}^{T} \cdot \mathbf{1}^N$$

$$\frac{\partial L}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot \frac{\partial Z^{(2)}}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot Z^{(2)} \cdot \frac{\partial}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot A^{(1)}$$

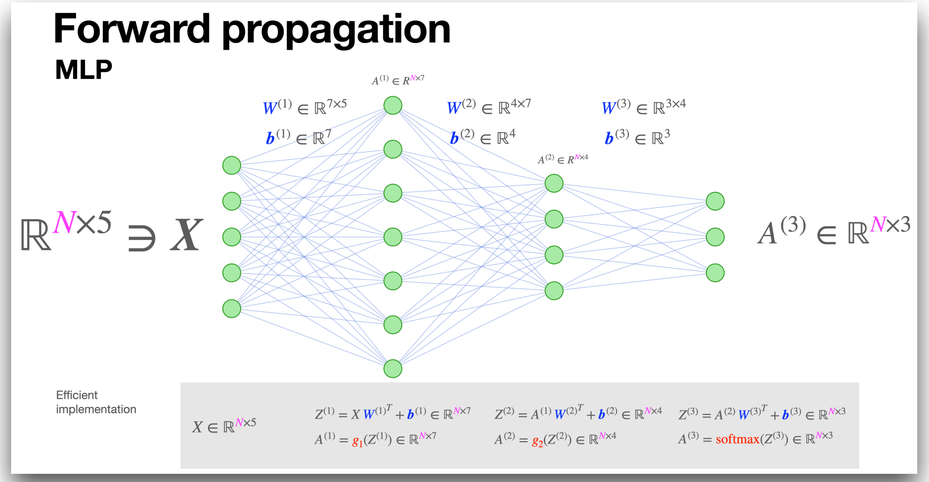
$$\frac{\partial L_1}{\partial b^{(2)}} = \frac{\partial L_1}{\partial Z^{(2)}}^{T} \cdot \frac{\partial Z^{(2)}}{\partial b^{(2)}} = \frac{\partial L_1}{\partial Z^{(2)}}^{T} \cdot \mathbf{1}^N$$

$$\frac{\partial L}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot \frac{\partial Z^{(1)}}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot Z^{(1)} \cdot \frac{\partial}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot X$$

Backward propagation

MLP

$$L = -\frac{1}{N}\,\mathrm{sum}\left(Y \otimes \log A^{(3)}\right) = -\frac{1}{N}\sum_{p=1}^{N} L_p$$



$$\frac{\partial L}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot \frac{\partial Z^{(3)}}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot Z^{(3)} \cdot \frac{\partial}{\partial W^{(3)}} = \frac{\partial L}{\partial Z^{(3)}}^{T} \cdot A^{(2)}$$

$$\frac{\partial L_1}{\partial \boldsymbol{b}^{(3)}} = \frac{\partial L_1}{\partial Z^{(3)}}^{T} \cdot \frac{\partial Z^{(3)}}{\partial \boldsymbol{b}^{(3)}} = \frac{\partial L_1}{\partial Z^{(3)}}^{T} \cdot \mathbf{1}^N$$

$$\frac{\partial L}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot \frac{\partial Z^{(2)}}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot Z^{(2)} \cdot \frac{\partial}{\partial W^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}^{T} \cdot A^{(1)}$$

$$\frac{\partial L_1}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial L_1}{\partial Z^{(2)}}^{T} \cdot \frac{\partial Z^{(2)}}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial L_1}{\partial Z^{(2)}}^{T} \cdot \mathbf{1}^N$$

$$\frac{\partial L}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot \frac{\partial Z^{(1)}}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot Z^{(1)} \cdot \frac{\partial}{\partial W^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}^{T} \cdot X$$

$$\frac{\partial L_1}{\partial \boldsymbol{b}^{(1)}} = \frac{\partial L_1}{\partial Z^{(1)}}^{T} \cdot \frac{\partial Z^{(1)}}{\partial \boldsymbol{b}^{(1)}} = \frac{\partial L_1}{\partial Z^{(1)}}^{T} \cdot \mathbf{1}^N$$