QBUS6850

Lecture 4

Neural Network and Deep Learning- II

BUSINESS SCHOOL

QBUS6850 Team

THE UNIVERSITY OF SYDNEY

❑ Topics covered

- Neural Network regression and classification loss functions
- Backward propagation Neural Network

❑ References

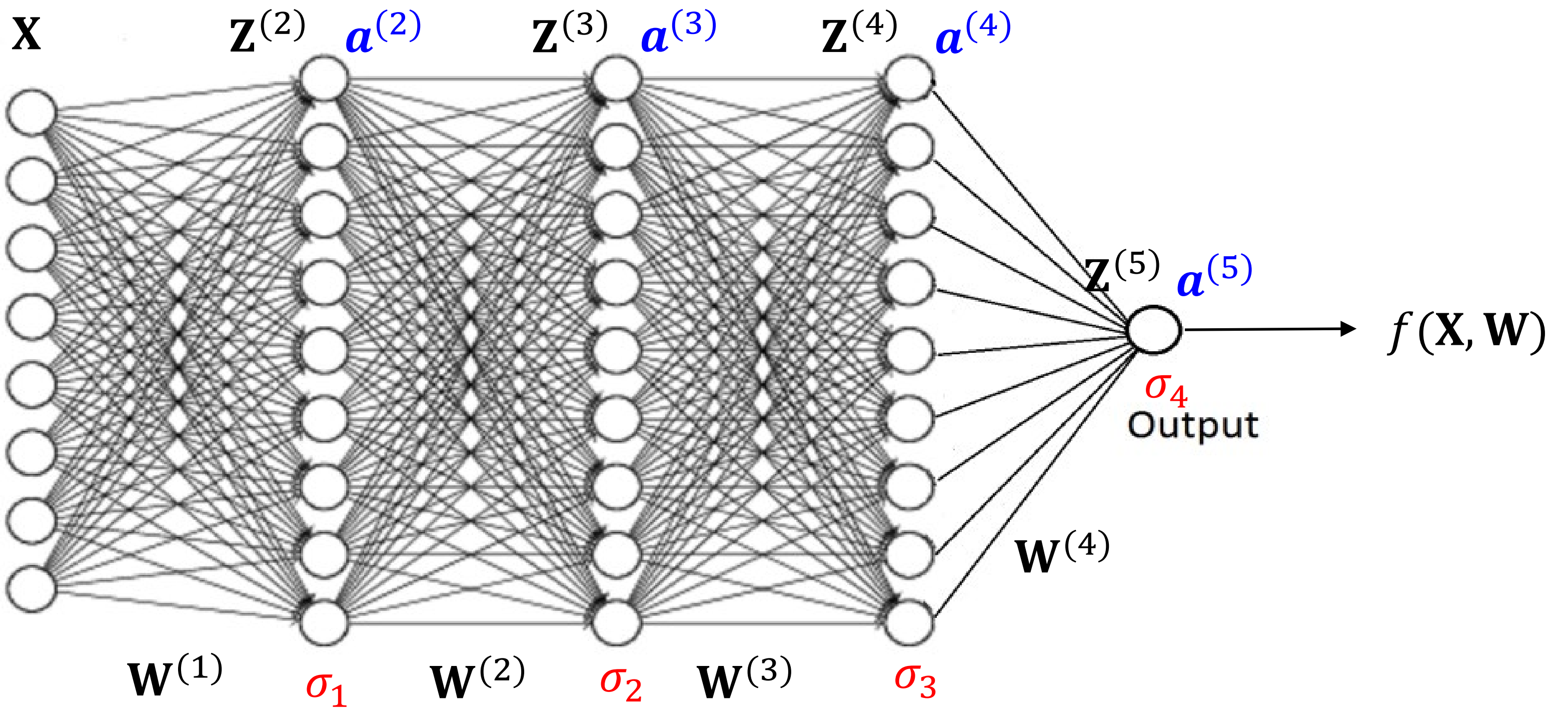
- Alpaydin (2014), Chapter 11
- Bishop (2006), Chapter 5

Learning Objectives

- Understand the Neural Network regression and classification loss functions
- Understand why and how to add regularization terms into Neural Network
- Understand the intuition of backpropagation
- Understand the process of backpropagation

Recap Week 3

Forward Propagation Algorithm


$\mathbf{X}$
$\mathbf{Z}^{(2)}$ $\boldsymbol{a}^{(2)}$
$\mathbf{Z}^{(3)}$ $\boldsymbol{a}^{(3)}$
$\mathbf{Z}^{(4)}$ $\boldsymbol{a}^{(4)}$
$\mathbf{Z}^{(5)}$ $\boldsymbol{a}^{(5)}$
$f(\mathbf{X}, \mathbf{W})$
σ_4
Output
$\mathbf{W}^{(4)}$
$\mathbf{W}^{(1)}$
σ_1
$\mathbf{W}^{(2)}$
σ_2
$\mathbf{W}^{(3)}$
σ_3

$$\mathbf{Z}^{(2)} = \mathbf{X}\mathbf{W}^{(1)}; \boldsymbol{a}^{(2)} = \sigma_1\left(\mathbf{Z}^{(2)}\right); \mathbf{Z}^{(3)} = \boldsymbol{a}^{(2)}\mathbf{W}^{(2)}; \boldsymbol{a}^{(3)} = \sigma_2\left(\mathbf{Z}^{(3)}\right);$$

$$\mathbf{Z}^{(4)} = \boldsymbol{a}^{(3)}\mathbf{W}^{(3)}; \boldsymbol{a}^{(4)} = \sigma_3\left(\mathbf{Z}^{(4)}\right); \mathbf{Z}^{(5)} = \boldsymbol{a}^{(4)}\mathbf{W}^{(4)}; \boldsymbol{a}^{(5)} = \sigma_4\left(\mathbf{Z}^{(5)}\right); f(\mathbf{X}, \mathbf{W}) = \boldsymbol{a}^{(5)}$$

$$f(\mathbf{X}, \mathbf{W}) = \sigma_4(\sigma_3(\sigma_2(\sigma_1(\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}^{(2)})\mathbf{W}^{(3)})\mathbf{W}^{(4)})$$

A composition function

Forward Propagation Algorithm with Bias

Another way to look at biases where $\mathbf{b}^{(l)}$ in row shape

$$\mathbf{Z}^{(2)} = \mathbf{X}\mathbf{W}^{(1)} + \mathbf{1}_{\mathrm{N}}\mathbf{b}^{(1)}; \quad \boldsymbol{a}^{(2)} = \sigma_1\left(\mathbf{Z}^{(2)}\right);$$

$$\mathbf{Z}^{(3)} = \boldsymbol{a}^{(2)}\mathbf{W}^{(2)} + \mathbf{1}_{\mathrm{N}}\mathbf{b}^{(2)}; \quad \boldsymbol{a}^{(3)} = \sigma_2\left(\mathbf{Z}^{(3)}\right);$$

$$\mathbf{Z}^{(4)} = \boldsymbol{a}^{(3)}\mathbf{W}^{(3)} + \mathbf{1}_{\mathrm{N}}\mathbf{b}^{(3)}; \boldsymbol{a}^{(4)} = \sigma_3\left(\mathbf{Z}^{(4)}\right);$$

$$\mathbf{Z}^{(5)} = \boldsymbol{a}^{(4)}\mathbf{W}^{(4)} + \mathbf{1}_{\mathrm{N}}\mathbf{b}^{(4)}; \quad \boldsymbol{a}^{(5)} = \sigma_4\left(\mathbf{Z}^{(5)}\right);$$

$$f(\mathbf{X}, \mathbf{W}) = \boldsymbol{a}^{(5)}$$

$$f(\mathbf{X}, \mathbf{W}) = \sigma_4\left(\sigma_3\left(\sigma_2\left(\sigma_1\left(\mathbf{X}\mathbf{W}^{(1)} + \mathbf{1}_{\mathrm{N}}\mathbf{b}^{(1)}\right)\mathbf{W}^{(2)} + \mathbf{1}_{\mathrm{N}}\mathbf{b}^{(2)}\right)\mathbf{W}^{(3)} + \mathbf{1}_{\mathrm{N}}\mathbf{b}^{(3)}\right)\mathbf{W}^{(4)} + \mathbf{1}_{\mathrm{N}}\mathbf{b}^{(4)}\right)$$

Neural Network Loss Function

NN Regression Loss Function

$$L(\mathbf{W}) = \frac{1}{2N}\sum_{n=1}^{N}(f(\mathbf{x}_n, \mathbf{W}) - t_n)^2$$

How NN and linear regression loss functions are different?

Normally for regression, the output neuron takes the identity activation

As in the regression we did in week 2&3, we can use gradient descent to try to minimize the loss function as a function of parameters $\mathbf{W}$. α is the learning rate.

$$\mathbf{W} \coloneqq \mathbf{W} - \alpha \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}}$$

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}} = ?$$

This will be a much harder problem, since now we have weights $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$ for a 3-layer NN.

NN Binary Classification Loss Function

$$L(\mathbf{W}) = -\left[\sum_{n=1}^{N}\left(t_n \log\big(f(\mathbf{x}_n, \mathbf{W})\big) + (1 - t_n)\log\big(1 - f(\mathbf{x}_n, \mathbf{W})\big)\right)\right]$$

where each t_n is 0 (if $\mathbf{x}_n$ is in class 0) or 1 (if $\mathbf{x}_n$ is in class 1)

In classification, the output neuron takes the sigmoid activation

Similarly we can use gradient descent to try to minimize the loss function as a function of parameters $\mathbf{W}$. α is the learning rate.

$$\mathbf{W} \coloneqq \mathbf{W} - \alpha \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}}$$

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}} = ?$$

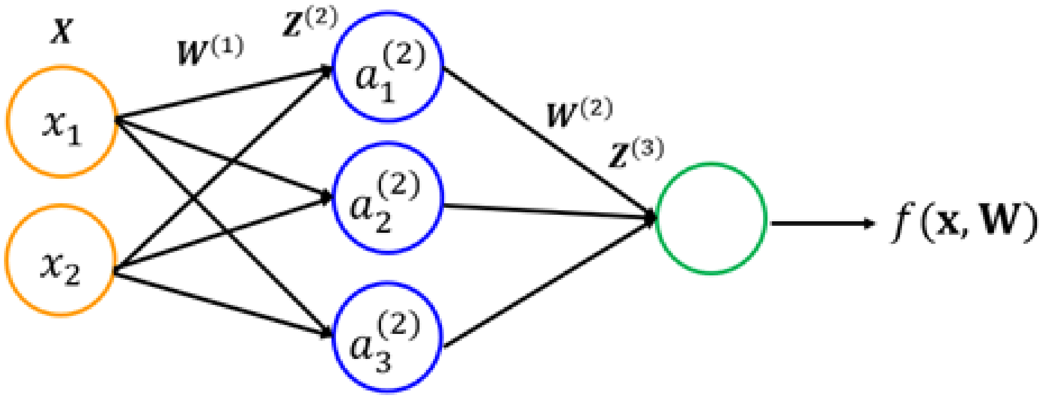
This will be a much harder problem, since now we have weights $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$ for a 3-layer NN.

Neural Network Training

Parameters to be estimated

Target: minimize the loss function by changing the weights $\boldsymbol{W}^{(1)}$, $\boldsymbol{W}^{(2)}$

Let’s still use 3-layer NN without bias units example that we had last week



$$\mathbf{W}^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix} \qquad \mathbf{W}^{(2)} = \begin{bmatrix} w_{11}^{(2)} \\ w_{21}^{(2)} \\ w_{31}^{(2)} \end{bmatrix}$$

Poll 1

Dimensionality

Why not simply try all the potential weights and see which combination produces the smallest loss?

Curse of dimensionality:

- 1 parameter: 1000 trials, 0.05 seconds
- 2 parameters: 1000 ×1000=1 million trials, 0.05*1000= 50 seconds
- 3 parameters: 1000 × 1000 × 1000=1 billion trials; (1,000,000*0.05)/(60*60)= 13.89 hours
- …
- 9 parameters: how long it will take?

Besides estimating the parameters of NN, other problems in neural network modelling:

- How to select the number of hidden layers?
- How to select the number of units in each hidden layer?
- How to perform feature selection?
- ...

Backpropagation Intuition

Backpropagation

- Backpropagation is a method used in artificial neural networks to calculate the error contribution of each neuron after a batch of data (in image recognition, multiple images) is processed. This is used by an enveloping optimization algorithm to adjust the weight of each neuron, completing the learning process for that case.

- Technically it calculates the gradient of the loss function. It is commonly used in the gradient descent optimization algorithm. It is also called **backward propagation of errors**, because the **error is calculated at the output and distributed back** through the network layers.

Poll 2

https://en.wikipedia.org/wiki/Backpropagation

- Phase 1: propagation. Each propagation involves the following steps:
  - Propagation forward through the network to generate the output value(s)
  - Calculation of the loss (error term)
  - Propagation of the output activations back through the network using the training pattern target in order to generate the **deltas (δ)** of all output and hidden neurons.

- Phase 2: weight (parameter) update. For each weight, the following steps must be followed:
  - The weight's output **deltas** and input activation are multiplied to find the gradient of the weight.
  - A ratio (learning rate) of the weight's gradient is subtracted from the weight.

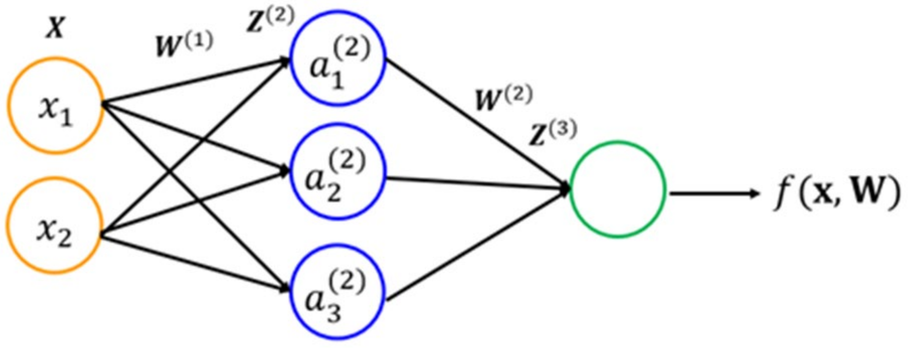
Backpropagation

3-Layer Neural Network

- Now we are going to use back propagation to train this NN for regression
- The example we had last week
- No bias unit for simplicity

$$L(\mathbf{W}) = \frac{1}{2}\sum_{n=1}^{N}(f(\mathbf{x}_n, \mathbf{W}) - t_n)^2$$

x_1	x_2
1	0.875
0.25	1
0.50	0.45
0.75	0.25



expense
1
0.4
0.5
0.6

Consider 4 data

Calculate Gradient (matrix calculus)

$$\frac{\partial \mathrm{L}(\mathbf{W})}{\partial \mathbf{W}} = ?$$

Our parameter set $\mathbf{W}$ contains $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$

$$\mathbf{W}^{(1)} = \overbrace{\begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix}}^{\text{Hidden layer size}} \left.\vphantom{\begin{bmatrix} w_{11}^{(1)} \\ w_{21}^{(1)} \end{bmatrix}}\right\} \begin{array}{l}\text{Input layer} \\ \text{size.} \\ \text{2 features.}\end{array}$$

$$\mathbf{W}^{(2)} = \overbrace{\begin{bmatrix} w_{11}^{(2)} \\ w_{21}^{(2)} \\ w_{31}^{(2)} \end{bmatrix}}^{\text{Output layer size}} \left.\vphantom{\begin{bmatrix} w_{11}^{(2)} \\ w_{21}^{(2)} \\ w_{31}^{(2)} \end{bmatrix}}\right\} \begin{array}{l}\text{Hidden} \\ \text{layer} \\ \text{size}\end{array}$$

$$\frac{\partial \mathrm{L}(\mathbf{W})}{\partial \mathbf{W}^{(1)}} = \overbrace{\begin{bmatrix} \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{11}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{12}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{13}^{(1)}} \\ \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{21}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{22}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{23}^{(1)}} \end{bmatrix}}^{\text{Hidden layer size}} \left.\vphantom{\begin{bmatrix} \frac{a}{b} \\ \frac{a}{b} \end{bmatrix}}\right] \begin{array}{l}\text{Input} \\ \text{layer} \\ \text{size}\end{array}$$

$$\frac{\partial \mathrm{L}(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \overbrace{\begin{bmatrix} \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{11}^{(2)}} \\ \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{21}^{(2)}} \\ \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{31}^{(2)}} \end{bmatrix}}^{\text{Output layer size}} \left.\vphantom{\begin{bmatrix} \frac{a}{b} \\ \frac{a}{b} \\ \frac{a}{b} \end{bmatrix}}\right\} \begin{array}{l}\text{Hidden} \\ \text{layer} \\ \text{size}\end{array}$$

Backpropagation- Step 1

- First, work on the partial derivatives of loss with respect to $\boldsymbol{W}^{(2)}$.
- Note that now we start from the **"right"** of NN, or an opposite direction compared to forward propagation

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \frac{\partial \frac{1}{2} \sum_{n=1}^{N} (f(\mathbf{x}_n, \mathbf{W}) - t_n)^2}{\partial \mathbf{W}^{(2)}}$$

Sum rules in differentiation: derivative of sum equals to the sum of derivatives

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \sum_{n=1}^{N} \frac{\partial \frac{1}{2} (f(\mathbf{x}_n, \mathbf{W}) - t_n)^2}{\partial \mathbf{W}^{(2)}}$$

Remove summation for simplicity, will look after it later.

~~Also remove the subscript (n) in $f(\mathbf{x}_n, \mathbf{W})$ and t_n for simplicity.~~

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \frac{\partial \frac{1}{2} (f(\mathbf{x}_\mathrm{n}, \mathbf{W}) - t_n)^2}{\partial \mathbf{W}^{(2)}}$$

Backpropagation- Step 1

Derivative chain rule

t_n is a constant
w.r.t. $\boldsymbol{W}^{(2)}$

$f(\mathbf{x}_{\mathrm{n}}, \mathbf{W}) = \boldsymbol{\sigma}(\boldsymbol{z}_n^{(3)})$

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \frac{\partial \frac{1}{2}(f(\mathbf{x}_{\mathrm{n}}, \mathbf{W}) - t_n)^2}{\partial \mathbf{W}^{(2)}} = (f(\mathbf{x}_{\mathrm{n}}, \mathbf{W}) - t_n)\frac{\partial f(\mathbf{x}_{\mathrm{n}}, \mathbf{W})}{\partial \mathbf{W}^{(2)}}$$

$$= (f(\mathbf{x}_n, \mathbf{W}) - t_n)\boxed{\frac{\partial f(\mathbf{x}_{\mathrm{n}}, \mathbf{W})}{\partial \boldsymbol{z}_n^{(3)}} * \frac{\partial \boldsymbol{z}_n^{(3)}}{\partial \mathbf{W}^{(2)}}}$$

Derivative chain rule.
Based on FP, $\boldsymbol{z}_n^{(3)}$ is between$f(\mathbf{x}_n, \mathbf{W})$and $W^{(2)}$

$$= (f(\mathbf{x}_n, \mathbf{W}) - t_n)\boxed{\frac{\partial \boldsymbol{\sigma}(\boldsymbol{z}_n^{(3)})}{\partial \boldsymbol{z}_n^{(3)}}}\frac{\partial \boldsymbol{z}_n^{(3)}}{\partial \mathbf{W}^{(2)}}$$

How to calculate this?

$$\frac{\partial \boldsymbol{\sigma}(\boldsymbol{z}_n^{(3)})}{\partial \boldsymbol{z}_n^{(3)}} = \sigma'\left(\boldsymbol{z}_n^{(3)}\right) = \sigma\left(\boldsymbol{z}_n^{(3)}\right) .* \left(1 - \sigma\left(\boldsymbol{z}_n^{(3)}\right)\right)$$

For single data

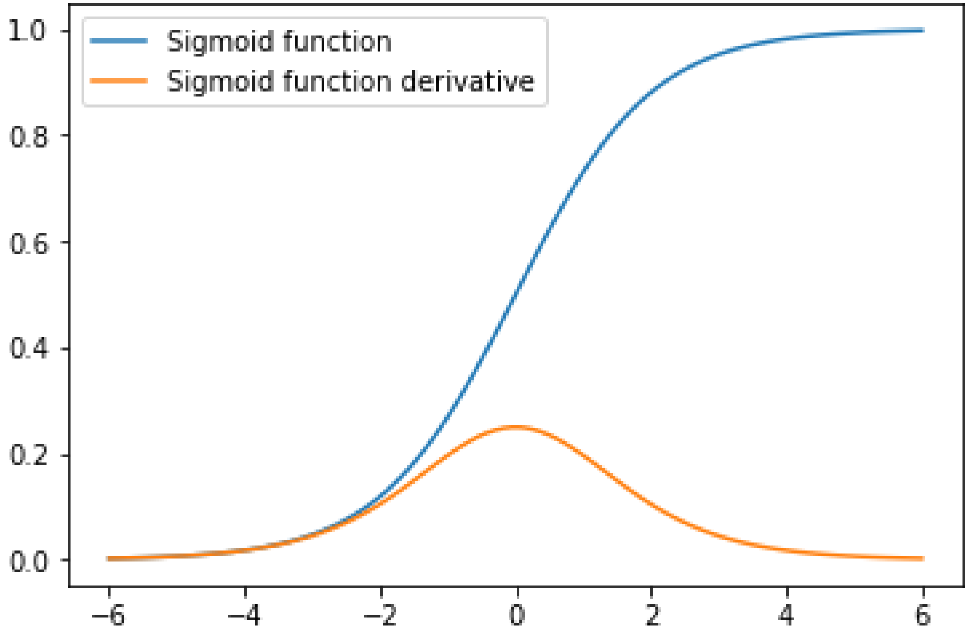
Poll 3 & 4

Sigmoid function derivative

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\frac{\partial \boldsymbol{\sigma}(\boldsymbol{z}_n^{(3)})}{\partial \boldsymbol{z}_n^{(3)}} = ?$$

$$\sigma'(z) = \frac{d}{dz}\left(\frac{1}{1+e^{-z}}\right) = \frac{e^{-z}}{(1+e^{-z})^2} = \sigma(z)(1-\sigma(z))$$



Backpropagation- Step 1

For a single data $\mathbf{x}_n$, further we have

$$\frac{\partial \mathbf{z}_n^{(3)}}{\partial \mathbf{W}^{(2)}} = \frac{\partial}{\partial \mathbf{W}^{(2)}}\left(w_{11}^{(2)} a_{n1}^{(2)} + w_{21}^{(2)} a_{n2}^{(2)} + w_{31}^{(2)} a_{n3}^{(2)}\right) = [a_{n1}^{(2)},\ a_{n2}^{(2)}, a_{n3}^{(2)}]^T$$

Hence the derivative for a single data is

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = (f(\mathbf{x}_n, \mathbf{W}) - t_n)\, \sigma'\left(\mathbf{z}_n^{(3)}\right) [a_{n1}^{(2)},\ a_{n2}^{(2)}, a_{n3}^{(2)}]^T$$

We add them together

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \sum_{n=1}^{N} (f(\mathbf{x}_n, \mathbf{W}) - t_n)\, \sigma'\left(\mathbf{z}_n^{(3)}\right) \begin{bmatrix} a_{n1}^{(2)} \\ a_{n2}^{(2)} \\ a_{n3}^{(2)} \end{bmatrix}$$

Next, we show how this can be done in matrix operation

THE UNIVERSITY OF SYDNEY

Backpropagation- Step 1

Now we write it for all the data $\mathbf{X}$ and $\mathbf{t}$ in matrix form

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \underbrace{\boxed{(f(\mathbf{X},\mathbf{W}) - \mathbf{t})\sigma'\left(\mathbf{Z}^{(3)}\right)}}_{\text{Known}} \circledast \boxed{\frac{\partial \mathbf{Z}^{(3)}}{\partial \mathbf{W}^{(2)}}}$$

How to organize this?

here $N = 4$:

$$\mathbf{Z}^{(3)} = \begin{bmatrix} z_{11}^{(3)} \\ z_{21}^{(3)} \\ z_{31}^{(3)} \\ z_{41}^{(3)} \end{bmatrix} \qquad \boldsymbol{a}^{(2)} = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} \\ a_{21}^{(2)} & a_{22}^{(2)} & a_{23}^{(2)} \\ a_{31}^{(2)} & a_{32}^{(2)} & a_{33}^{(2)} \\ a_{41}^{(2)} & a_{42}^{(2)} & a_{43}^{(2)} \end{bmatrix} \qquad \mathbf{W}^{(2)} = \begin{bmatrix} w_{11}^{(2)} \\ w_{21}^{(2)} \\ w_{31}^{(2)} \end{bmatrix}$$

$$\frac{\partial \mathbf{z}_n^{(3)}}{\partial \mathbf{W}^{(2)}} = [a_{n1}^{(2)},\ a_{n2}^{(2)}, a_{n3}^{(2)}]^T$$

$$\underset{4\times 1}{\mathbf{Z}^{(3)}} = \underset{4\times 3}{\boxed{a^{(2)}}}\underset{3\times 1}{\mathbf{W}^{(2)}}$$

$\boldsymbol{Z}^{(3)}$ is actually a linear combination of $\boldsymbol{W}^{(2)}$ with $\boldsymbol{a}^{(2)}$, so:

$$\frac{\partial \mathbf{Z}^{(3)}}{\partial \mathbf{W}^{(2)}} = \begin{bmatrix} \boxed{a_{11}^{(2)} \quad a_{12}^{(2)} \quad a_{13}^{(2)}} \\ a_{21}^{(2)} \quad a_{22}^{(2)} \quad a_{23}^{(2)} \\ a_{31}^{(2)} \quad a_{32}^{(2)} \quad a_{33}^{(2)} \\ a_{41}^{(2)} \quad a_{42}^{(2)} \quad a_{43}^{(2)} \end{bmatrix}$$

Backpropagation- Step 1

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \boxed{\overset{4\times 1}{\left(f(\mathbf{X},\mathbf{W})-\mathbf{t}\right)} .* \overset{4\times 1}{\sigma'\left(\mathbf{Z}^{(3)}\right)}} \circledast \overset{4\times 3}{\boxed{\frac{\partial \mathbf{Z}^{(3)}}{\partial \mathbf{W}^{(2)}}}}$$

Here we use matrix element wise product

$$\boxed{\begin{bmatrix} f(\mathbf{x}_1,\mathbf{W})-t_1 \\ f(\mathbf{x}_2,\mathbf{W})-t_2 \\ f(\mathbf{x}_3,\mathbf{W})-t_3 \\ f(\mathbf{x}_4,\mathbf{W})-t_4 \end{bmatrix} .* \begin{bmatrix} \sigma'\left(z_{11}^{(3)}\right) \\ \sigma'\left(z_{21}^{(3)}\right) \\ \sigma'\left(z_{31}^{(3)}\right) \\ \sigma'\left(z_{41}^{(3)}\right) \end{bmatrix}}$$

$$\Downarrow$$

Backpropagation error

$$\boxed{\begin{bmatrix} \delta_1^{(3)} \\ \delta_2^{(3)} \\ \delta_3^{(3)} \\ \delta_4^{(3)} \end{bmatrix}}$$

$$\Downarrow$$

$$\boxed{a^{(2)} = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} \\ a_{21}^{(2)} & a_{22}^{(2)} & a_{23}^{(2)} \\ a_{31}^{(2)} & a_{32}^{(2)} & a_{33}^{(2)} \\ a_{41}^{(2)} & a_{42}^{(2)} & a_{43}^{(2)} \end{bmatrix}}$$

Now we need to calculate $\delta^{(3)}$ times $a^{(2)}$

Backpropagation- Step 1

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \boxed{\overset{4 \times 1}{\left(f(\mathbf{X},\mathbf{W}) - \mathbf{t}\right)} .* \overset{4 \times 1}{\sigma'\left(\mathbf{Z}^{(3)}\right)}} \circledast \overset{4 \times 3}{\boxed{\frac{\partial \mathbf{Z}^{(3)}}{\partial \mathbf{W}^{(2)}}}}$$

How to connect them?

Here we use matrix element wise product

$$\boxed{\begin{bmatrix} f(\mathbf{x}_1,\mathbf{W}) - t_1 \\ f(\mathbf{x}_2,\mathbf{W}) - t_2 \\ f(\mathbf{x}_3,\mathbf{W}) - t_3 \\ f(\mathbf{x}_4,\mathbf{W}) - t_4 \end{bmatrix} .* \begin{bmatrix} \sigma'\left(z_{11}^{(3)}\right) \\ \sigma'\left(z_{21}^{(3)}\right) \\ \sigma'\left(z_{31}^{(3)}\right) \\ \sigma'\left(z_{41}^{(3)}\right) \end{bmatrix}}$$

$$\Downarrow$$

Backpropagation error

$$\boxed{\begin{bmatrix} \delta_1^{(3)} \\ \delta_2^{(3)} \\ \delta_3^{(3)} \\ \delta_4^{(3)} \end{bmatrix}}$$

$$a^{(2)} = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} \\ a_{21}^{(2)} & a_{22}^{(2)} & a_{23}^{(2)} \\ a_{31}^{(2)} & a_{32}^{(2)} & a_{33}^{(2)} \\ a_{41}^{(2)} & a_{42}^{(2)} & a_{43}^{(2)} \end{bmatrix}$$

THE UNIVERSITY OF SYDNEY

Backpropagation- Step 1

Taking transpose of $\boldsymbol{a}^{(2)}$, then times $\left(\boldsymbol{a}^{(2)}\right)^T$ by $\delta^{(3)}$

$$\left(\boldsymbol{a}^{(2)}\right)^T = \begin{bmatrix} a_{11}^{(2)} & a_{21}^{(2)} & a_{31}^{(2)} & a_{41}^{(2)} \\ a_{12}^{(2)} & a_{22}^{(2)} & a_{32}^{(2)} & a_{42}^{(2)} \\ a_{13}^{(2)} & a_{23}^{(2)} & a_{33}^{(2)} & a_{43}^{(2)} \end{bmatrix} \qquad \delta^{(3)} = \begin{bmatrix} \delta_1^{(3)} \\ \delta_2^{(3)} \\ \delta_3^{(3)} \\ \delta_4^{(3)} \end{bmatrix}$$

3 × 4

4 × 1

Step 1: summary

1st equation uses matrix product

2nd equation uses matrix **element wise** product

4 × 1

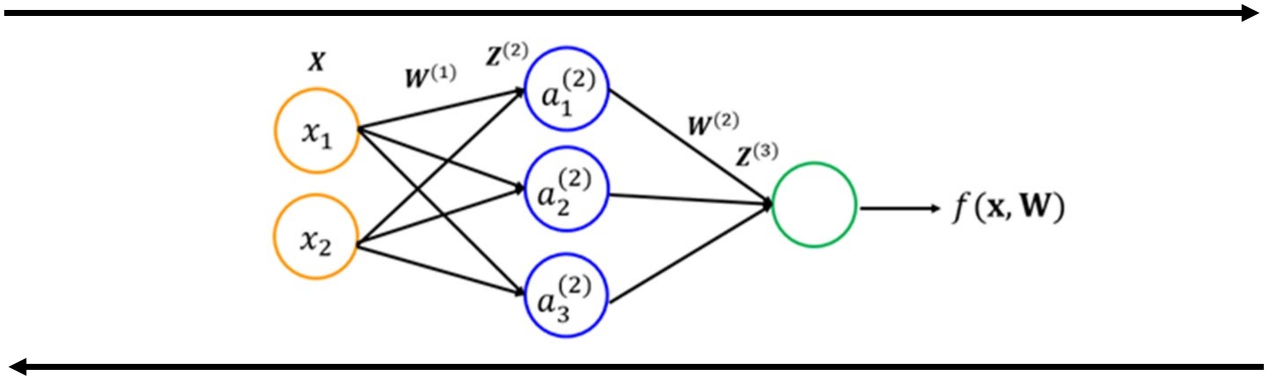
$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \boxed{\left(\boldsymbol{a}^{(2)}\right)^T \delta^{(3)}}$$

$$\delta^{(3)} = (f(\mathbf{X}, \mathbf{W}) - \mathbf{t}).* \sigma'\left(\mathbf{Z}^{(3)}\right)$$

- Now matrix multiplication does the sum job we need on slide 19.
- In the real implementation, first we calculate $\delta^{(3)}$, then calculate $\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}}$

Backpropagation

Forward propagation



Backpropagation

$$\delta^{(3)} = \begin{bmatrix} \delta_1^{(3)} \\ \delta_2^{(3)} \\ \delta_3^{(3)} \\ \delta_4^{(3)} \end{bmatrix}$$

- Given training examples, first run a "forward propagation" to compute all the activations throughout the network, including the output value of the NN$f(\mathbf{X}, \mathbf{W})$
- Then, for each node i in layer l, compute an "error term" $\delta_i^{(l)}$ that measures how much that node was "**responsible**" for any errors in our output

Backpropagation- Step 2

$$\mathbf{W}^{(1)} = \overbrace{\begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix}}^{\text{Hidden layer size}} \left.\vphantom{\begin{bmatrix} w_{11}^{(1)} \\ w_{21}^{(1)} \end{bmatrix}}\right\} \text{Input unit size}$$

$$\frac{\partial \mathrm{L}(\mathbf{W})}{\partial \mathbf{W}^{(1)}} = \begin{bmatrix} \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{11}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{12}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{13}^{(1)}} \\ \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{21}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{22}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{23}^{(1)}} \end{bmatrix}$$

How to calculate this?

Backpropagation- Step 2

Let’s follow the similar process as step 1

Each $\circledast$ has its own meaning

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(1)}} = \frac{\partial \frac{1}{2}(f(\mathbf{x}_n,\mathbf{W}) - t_n)^2}{\partial \mathbf{W}^{(1)}} = (f(\mathbf{x}_n,\mathbf{W}) - t_n) \circledast \frac{\partial f(\mathbf{x}_n,\mathbf{W})}{\partial \mathbf{W}^{(1)}}$$

$$= (f(\mathbf{x}_n,\mathbf{W}) - t_n) \circledast \frac{\partial f(\mathbf{x}_n,\mathbf{W})}{\partial \mathbf{z}_n^{(3)}} \circledast \frac{\partial \mathbf{z}_n^{(3)}}{\partial \mathbf{W}^{(1)}}$$

We calculated this in step 1 →

$$= \boxed{(f(\mathbf{x}_n,\mathbf{W}) - t_n) \circledast \frac{\partial \boldsymbol{\sigma}(\mathbf{z}_n^{(3)})}{\partial \mathbf{z}_n^{(3)}}} \circledast \frac{\partial \mathbf{z}_n^{(3)}}{\partial \mathbf{W}^{(1)}}$$

Note the difference compared to step 1

$$= (f(\mathbf{x}_n,\mathbf{W}) - t_n).* \, \boldsymbol{\sigma}'\left(\mathbf{z}_n^{(3)}\right) \circledast \frac{\partial \mathbf{z}_n^{(3)}}{\partial \mathbf{W}^{(1)}}$$

$$= \boxed{\delta_n^{(3)}} \circledast \boxed{\frac{\partial \mathbf{z}_n^{(3)}}{\partial \mathbf{W}^{(1)}}}$$

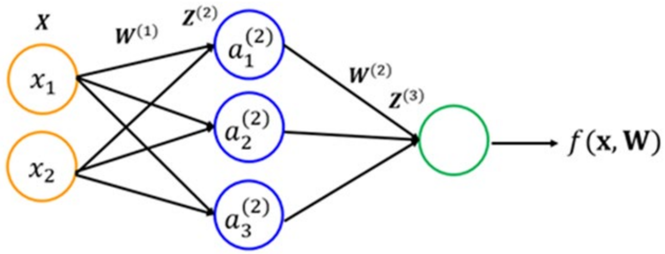
↑ How to calculate this?

Poll 5

Backpropagation- Step 2

Continued…

For simplicity we skip those $\circledast$



$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(1)}}$$

$$= \delta_n^{(3)} \circledast \boxed{\frac{\partial \boldsymbol{z}_n^{(3)}}{\partial \mathbf{W}^{(1)}}}$$

Derivative chain rule.

Based on FP, $W^{(2)}$, $a_n^{(2)}$, $z_n^{(2)}$ are between $z_n^{(3)}$ and $W^{(1)}$

$$\frac{\partial \boldsymbol{z}_n^{(3)}}{\partial a_n^{(2)}} = \frac{\partial a_n^{(2)} \mathbf{W}^{(2)}}{\partial a_n^{(2)}} = \mathbf{W}^{(2)} \otimes \mathbf{I}$$

$$= \delta_n^{(3)} \circledast \frac{\partial \boldsymbol{z}_n^{(3)}}{\partial a_n^{(2)}} \circledast \frac{\partial a_n^{(2)}}{\partial \mathbf{W}^{(1)}} = \delta_n^{(3)} \circledast \frac{\partial \boldsymbol{z}_n^{(3)}}{\partial a_n^{(2)}} \circledast \frac{\partial a_n^{(2)}}{\partial \mathbf{z}_n^{(2)}} \circledast \frac{\partial \mathbf{z}_n^{(2)}}{\partial \mathbf{W}^{(1)}}$$

$$= \delta_n^{(3)} \left(\mathbf{W}^{(2)}\right)^T \circledast \boxed{\frac{\partial a_n^{(2)}}{\partial \mathbf{z}_n^{(2)}}} \circledast \frac{\partial \mathbf{z}_n^{(2)}}{\partial \mathbf{W}^{(1)}} = \delta_n^{(3)} \left(\mathbf{W}^{(2)}\right)^T \circledast \frac{\partial \sigma\left(\mathbf{z}_n^{(2)}\right)}{\partial \mathbf{z}_n^{(2)}} \circledast \frac{\partial \mathbf{z}_n^{(2)}}{\partial \mathbf{W}^{(1)}}$$

$$\frac{\partial \mathbf{z}_n^{(2)}}{\partial \mathbf{W}^{(1)}} = \frac{\partial a_n^{(1)} \mathbf{W}^{(1)}}{\partial \mathbf{W}^{(1)}} = I \otimes a_n^{(1)}$$

$$= \delta_n^{(3)} \left(\mathbf{W}^{(2)}\right)^T .* \, \sigma'\left(\mathbf{z}_n^{(2)}\right) \circledast \frac{\partial \mathbf{z}_n^{(2)}}{\partial \mathbf{W}^{(1)}}$$

Use the same strategy as step 1 to calculate this.

Backpropagation- Step 2

For one data

$a_n^{(1)} = \mathbf{x}_n$ (row of **X**)

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(1)}} = \left(a_n^{(1)}\right)^T \left(\delta_n^{(3)}\left(\mathbf{W}^{(2)}\right)^T\right) .* \, \sigma'\left(\mathbf{z}_n^{(2)}\right)$$

$$= \boxed{(\mathbf{x}_n)^T}\left(\delta_n^{(3)}\left(\mathbf{W}^{(2)}\right)^T\right) .* \, \sigma'\left(\mathbf{z}_n^{(2)}\right)$$

For All data:

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(1)}} = (\mathbf{X})^T \underbrace{\left(\overset{\mathrm{N} \times 3}{\delta^{(3)}\left(\mathbf{W}^{(2)}\right)^T}\right) .* \, \sigma'\left(\mathbf{Z}^{(2)}\right)}_{\delta^{(2)}}$$

Poll 6

Step 2: summary for all data (e.g. N=3)

$$\overset{2 \times 3}{\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(1)}}} = \boxed{\overset{2 \times 4}{\mathbf{X}^T} \, \overset{4 \times 3}{\delta^{(2)}}}$$

$$\underset{4 \times 3}{\delta^{(2)}} = \left(\underset{4 \times 1}{\delta^{(3)}}\left(\underset{1 \times 3}{\mathbf{W}^{(2)}}\right)^T\right) .* \, \sigma'\left(\underset{4 \times 3}{\boldsymbol{Z}^{(2)}}\right)$$

- We removed summation from loss function calculation previously, but now matrix multiplication also does the sum job.
- In the real implementation, first we calculate $\delta^{(2)}$, then calculate $\frac{\partial L(W)}{\partial W^{(1)}}$

BP Step 1 + Step 2

Step 1: summary

$$\delta^{(3)} = (f(\mathbf{X}, \mathbf{W}) - \mathbf{t}) .* \sigma'\left(\mathbf{Z}^{(3)}\right)$$

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \left(\boldsymbol{a}^{(2)}\right)^{T} \delta^{(3)}$$

Note the where the matrix element wise product " .* " is used

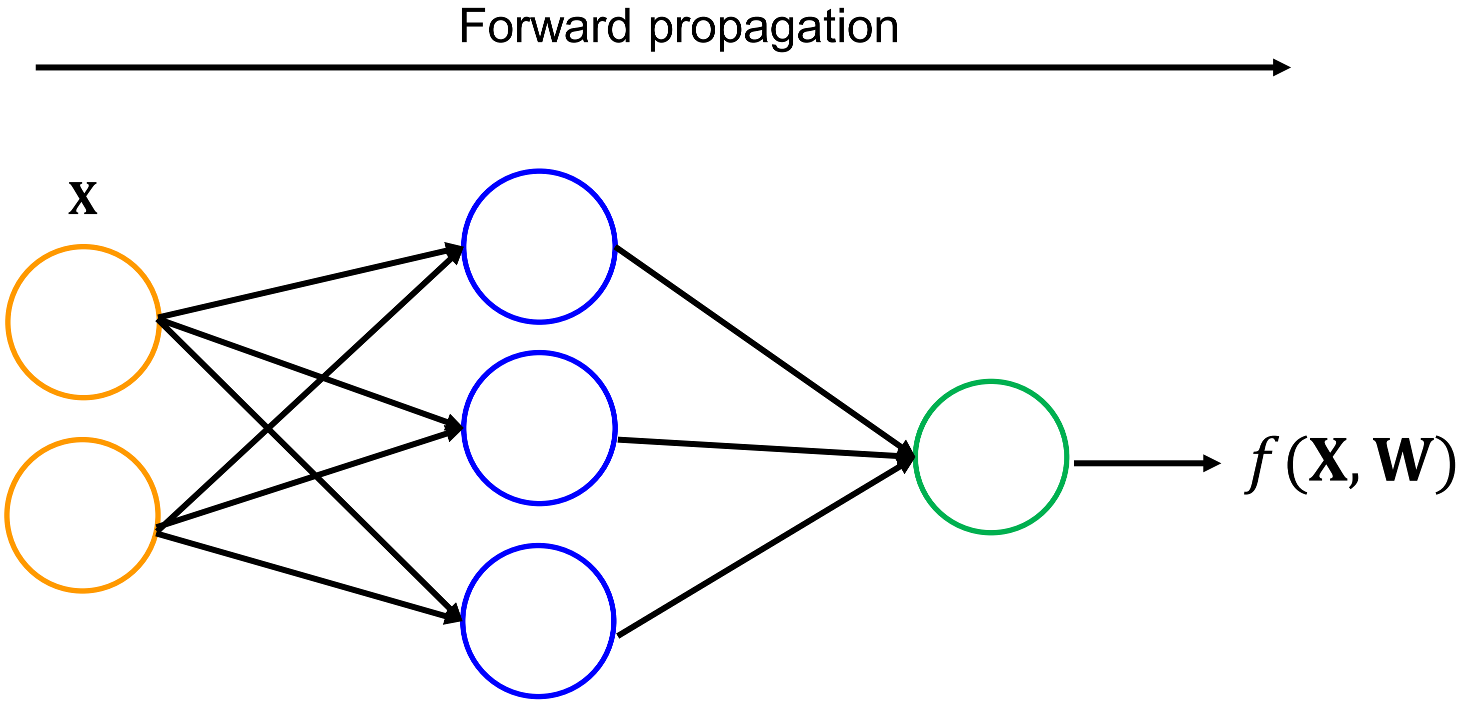
Step 2: summary

$$\delta^{(2)} = \left(\delta^{(3)}\left(\mathbf{W}^{(2)}\right)^{T}\right) .* \sigma'\left(\boldsymbol{Z}^{(2)}\right)$$

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(1)}} = \mathbf{X}^{T} \delta^{(2)} = \left(\boldsymbol{a}^{(1)}\right)^{T} \delta^{(2)}$$

Based on the layer index, we can see that this process can be easily repeated with more layers.

Backpropagation



$\delta^{(2)}$ is a 4 × 3 matrix

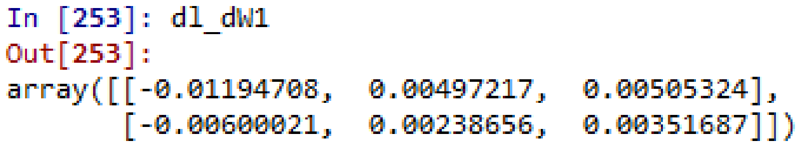
`Example04_Example01.py`

Python Example

Calculate gradient

$$\frac{\partial \mathrm{L}(\mathbf{W})}{\partial \mathbf{W}^{(1)}} = \begin{bmatrix} \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{11}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{12}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{13}^{(1)}} \\ \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{21}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{22}^{(1)}} & \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{23}^{(1)}} \end{bmatrix}$$

2×3



```
In [253]: dl_dW1
Out[253]:
array([[-0.01194708,  0.00497217,  0.00505324],
       [-0.00600021,  0.00238656,  0.00351687]])
```

$$\frac{\partial \mathrm{L}(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \begin{bmatrix} \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{11}^{(2)}} \\ \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{21}^{(2)}} \\ \frac{\partial \mathrm{L}(\mathbf{W})}{\partial w_{31}^{(2)}} \end{bmatrix}$$

3×1

FP+BP (Regression)

Forward Propagation

$$a^{(1)} = \mathbf{X}$$

$$\mathbf{Z}^{(2)} = \mathbf{X}\mathbf{W}^{(1)}$$

$$\boldsymbol{a}^{(2)} = \sigma(\mathbf{Z}^{(2)})$$

$$\mathbf{Z}^{(3)} = \boldsymbol{a}^{(2)}\mathbf{W}^{(2)}$$

$$f(\mathbf{X}, \mathbf{W}) = \mathbf{a}^{(3)} = \sigma(\mathbf{Z}^{(3)})$$

Identity

Backward Propagation

$$\delta^{(3)} = (f(\mathbf{X}, \mathbf{W}) - \mathbf{t}).* \boxed{\sigma'\left(\mathbf{Z}^{(3)}\right)}$$

=1

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \left(a^{(2)}\right)^T \delta^{(3)}$$

$$\delta^{(2)} = \left(\delta^{(3)}\left(\mathbf{W}^{(2)}\right)^T\right).* \sigma'\left(\boldsymbol{Z}^{(2)}\right)$$

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(1)}} = \mathbf{X}^T \delta^{(2)} = \left(\boldsymbol{a}^{(1)}\right)^T \delta^{(2)}$$

Calculate loss function

$$L(\mathbf{W}) = \frac{1}{2}\sum_{n=1}^{N}(f(\mathbf{x}_n, \mathbf{W}) - t_n)^2 = \frac{1}{2}(f(\mathbf{X}, \mathbf{W}) - \mathbf{t})^T(f(\mathbf{X}, \mathbf{W}) - \mathbf{t})$$

Use “ .* ” to denote the element-wise product operator

`Lecture04_Example02.py`

FP+BP (Binary Classification)

Forward Propagation

$$a^{(1)} = \mathbf{X}$$

$$\mathbf{Z}^{(2)} = \mathbf{X}\mathbf{W}^{(1)}$$

$$a^{(2)} = \sigma(\mathbf{Z}^{(2)})$$

$$\mathbf{Z}^{(3)} = a^{(2)}\mathbf{W}^{(2)}$$

$$f(\mathbf{X}, \mathbf{W}) = a^{(3)} = \sigma(\mathbf{Z}^{(3)})$$

sigmoid

Backward Propagation

$$\delta^{(3)} = (f(\mathbf{X}, \mathbf{W}) - \mathbf{t})$$ [t in 1 or 0 code]

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(2)}} = \left(a^{(2)}\right)^T \delta^{(3)}$$

$$\delta^{(2)} = \left(\delta^{(3)}\left(\mathbf{W}^{(2)}\right)^T\right) .* \, \sigma'\left(\mathbf{Z}^{(2)}\right)$$

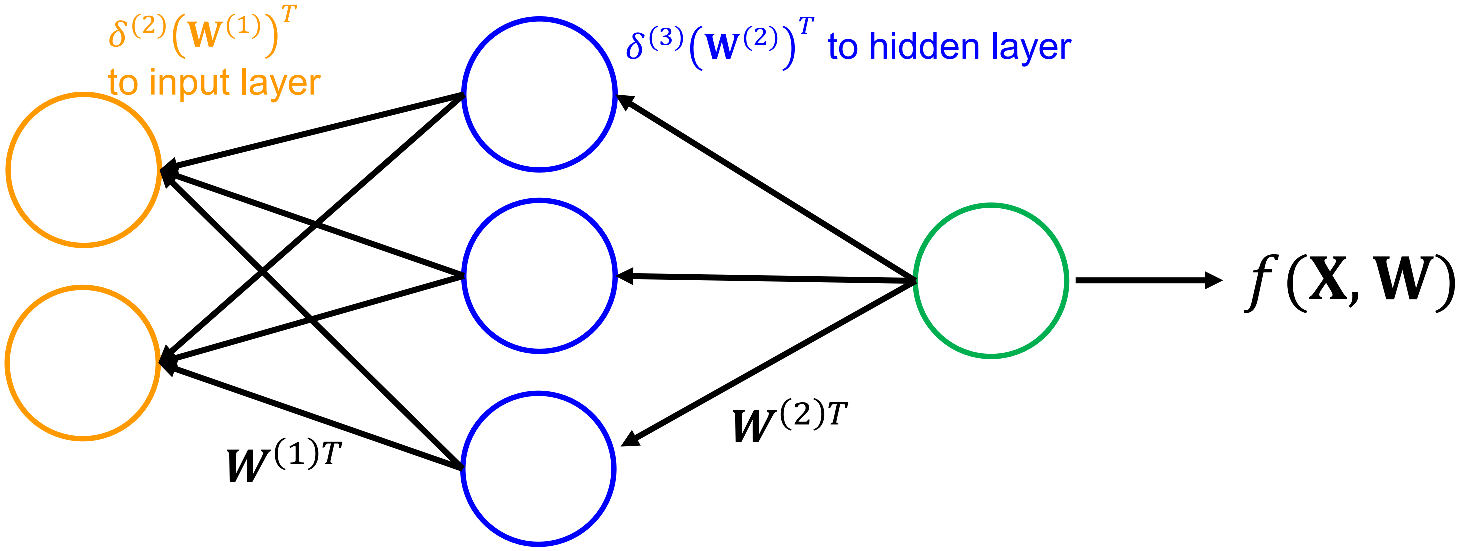
$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}^{(1)}} = \mathbf{X}^T \delta^{(2)} = \left(a^{(1)}\right)^T \delta^{(2)}$$

Calculate (logistic) loss function (1/N removed)

$$L(\mathbf{W}) = -\left[\sum_{n=1}^{N}\left(t_n \log\left(f(\mathbf{x}_n, \mathbf{W})\right) + (1 - t_n)\log\left(1 - f(\mathbf{x}_n, \mathbf{W})\right)\right)\right]$$

Use " .* " to denote the element-wise product operator

Why Called BP?



$\delta^{(2)}\left(\mathbf{W}^{(1)}\right)^T$ to input layer

$\delta^{(3)}\left(\mathbf{W}^{(2)}\right)^T$ to hidden layer

$\delta^{(1)} = \delta^{(2)}\left(\mathbf{W}^{(1)}\right)^T .* \, \sigma'\left(\boldsymbol{Z}^{(1)}\right)$

But no need.

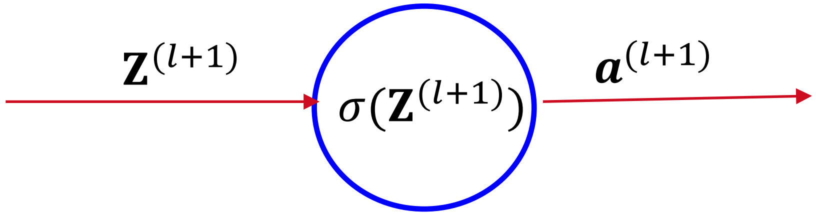
$\delta^{(2)} = \left(\delta^{(3)}\left(\mathbf{W}^{(2)}\right)^T .* \, \sigma'\left(\boldsymbol{Z}^{(2)}\right)\right)$

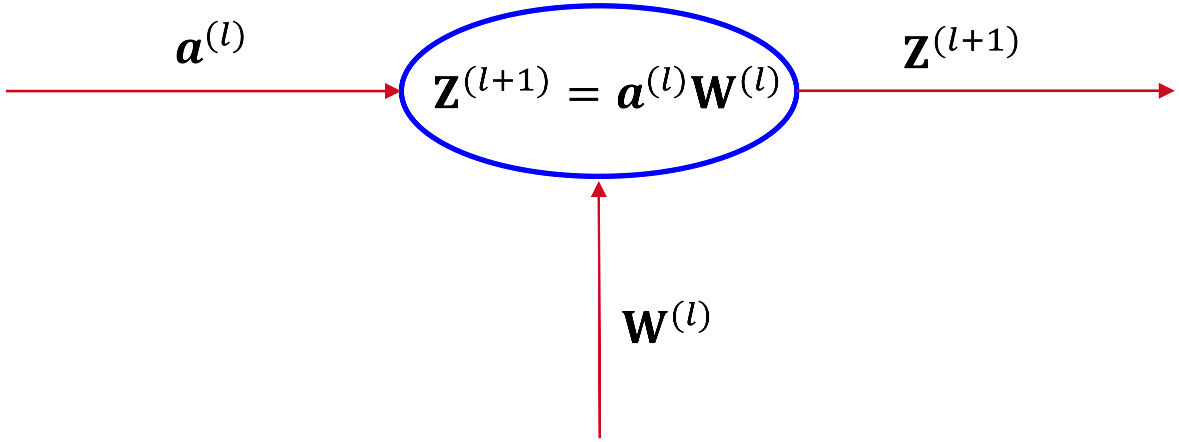
$\delta^{(3)} = f(\mathbf{X}, \mathbf{W}) - \mathbf{t}$

δ is backpropagated on the network

Summary: Matrix Approach

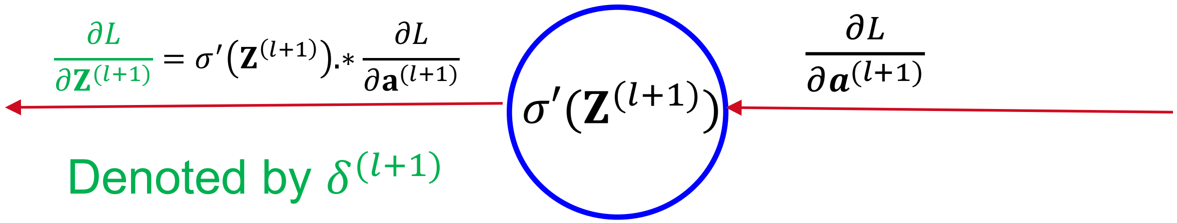
Two types of forward operations for All Data:





Summary: Matrix Approach

Their BP:

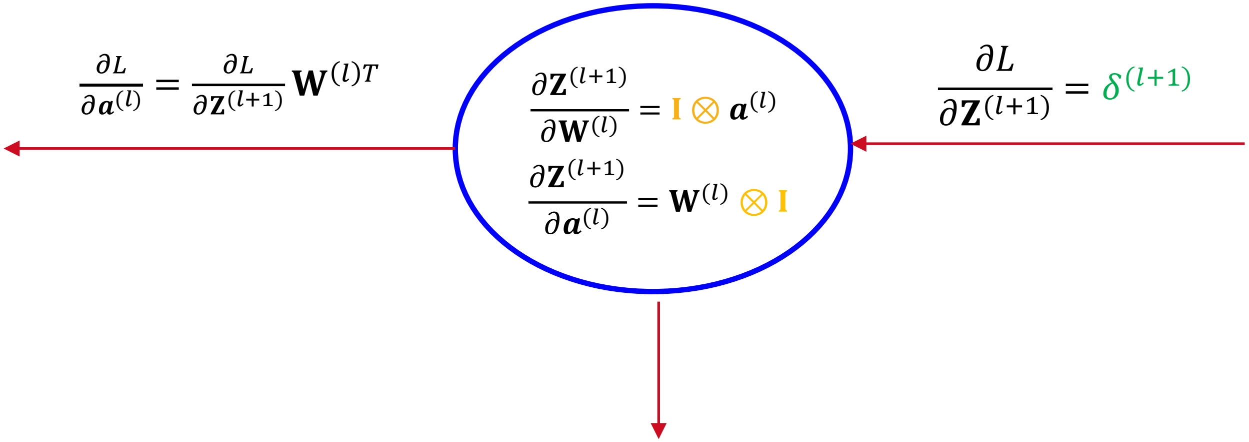


$$\frac{\partial L}{\partial \mathbf{Z}^{(l+1)}} = \sigma'\left(\mathbf{Z}^{(l+1)}\right).* \frac{\partial L}{\partial \mathbf{a}^{(l+1)}}$$

$$\sigma'\left(\mathbf{Z}^{(l+1)}\right)$$

$$\frac{\partial L}{\partial \boldsymbol{a}^{(l+1)}}$$

Denoted by $\delta^{(l+1)}$



$$\frac{\partial L}{\partial \boldsymbol{a}^{(l)}} = \frac{\partial L}{\partial \mathbf{Z}^{(l+1)}} \mathbf{W}^{(l)T}$$

$$\frac{\partial \mathbf{Z}^{(l+1)}}{\partial \mathbf{W}^{(l)}} = \mathbf{I} \otimes \boldsymbol{a}^{(l)}$$

$$\frac{\partial \mathbf{Z}^{(l+1)}}{\partial \boldsymbol{a}^{(l)}} = \mathbf{W}^{(l)} \otimes \mathbf{I}$$

$$\frac{\partial L}{\partial \mathbf{Z}^{(l+1)}} = \delta^{(l+1)}$$

$$\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \left(\boldsymbol{a}^{(l)}\right)^T \frac{\partial L}{\partial \mathbf{Z}^{(l+1)}} = \left(\boldsymbol{a}^{(l)}\right)^T \delta^{(l+1)}$$