Backpropagation

Ananda Biswas

Contents

1 Introduction

Backpropagation is the core algorithm used to train neural networks. It is a method for computing the gradient of the loss function with respect to each of the weights and biases in the network, so we can use those gradients to adjust the weights and biases and minimize the loss (error). Recall that gradient descent or its upgradations require gradients of the loss function w.r.t. all the parameters of the model. Backpropagation is the very algorithm that provides the gradients.

Here we actually calculate the gradients of the loss with respect to each parameter (weight and bias) by applying the chain rule from the output layer back to the input layer. Starting from the input layer, when we move through the hidden layers and reach the output layer, it is called a **Forward-propagation**. Likewise when we traverse from the output layer, through the hidden layers back to the input layer, we call it a **Backward-propagation**.

2 Set-up

- Suppose we have a data-matrix $X_{n\times m}$ where m is number of examples and n is the number of features. [Generally, in a data-matrix, features construct the columns and examples are stacked vertically. Here we consider the transposed case *i.e.* examples are in columns, it will help us in computation (linear algebra)].
- Consider a fully connected Feed-forward neural network with

  (i) an input layer,

  (ii) $L-1$ hidden layers

  (iii) an output layer,

  each of these layers having some number of neurons.
- Each neuron in the hidden layer and output layer can be split into two parts :

  (i) **pre-activation** or **aggregation** ($\underset{\sim}{a_i}$)

  (ii) **activation** ($\underset{\sim}{h_i}$).
- The input layer is often called the 0-th layer and the output layer can be called the L-th layer.
- W_i and $\underset{\sim}{b_i}$ are the weights and biases between layers $i-1$ and i $\forall i = 1(1)\overline{L-1}$.

  $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$ and $\underset{\sim}{b_i} \in \mathbb{R}^{n_i}$ where n_i is the number of neurones at layer i.

  Let the output layer have p neurones. W_L and $\underset{\sim}{b_L}$ are the weights and biases between the last hidden layer *i.e.* layer $L-1$ and the output layer *i.e.* layer L.

  $W_L \in \mathbb{R}^{p \times n_{L-1}}$ and $\underset{\sim}{b_L} \in \mathbb{R}^p$.
- $\forall i = 1(1)L$, pre-activation or aggregation at layer i is given by

$$\underset{\sim}{a_i} = W_i \cdot \underset{\sim}{h_{i-1}} + \underset{\sim}{b_i}$$

  with $\underset{\sim}{h_0} = \underset{\sim}{x}$, the input.
- $\forall i = 1(1)\overline{L-1}$

– activation at layer i is given by

$$\underset{\sim}{h_i} = g(\underset{\sim}{a_i})$$

where $g(\cdot)$ is called the **activation function** for the hidden layers (*e.g.* sigmoid, tanh, linear, ReLU etc.)

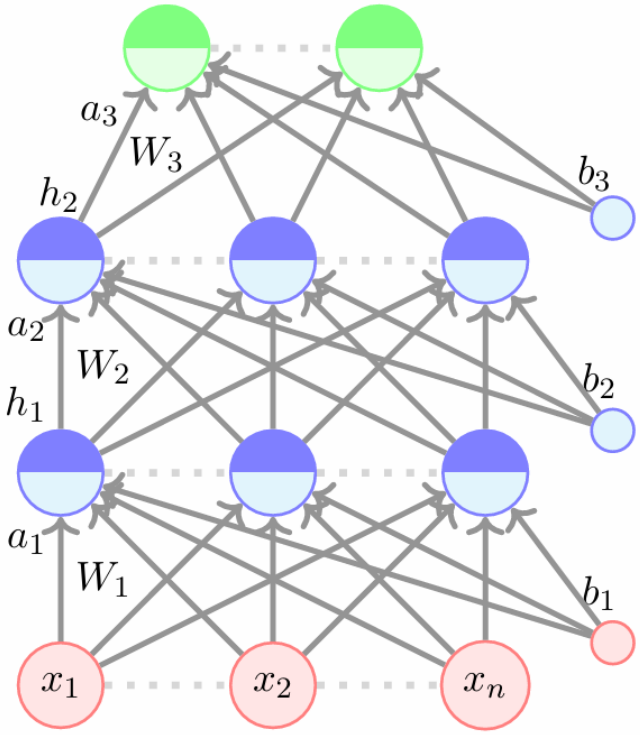
– activation at the output layer is given by

$$f(\underset{\sim}{x}) = \underset{\sim}{h_L}(\underset{\sim}{x}) = O(\underset{\sim}{a_L}(\underset{\sim}{x}))$$

where $O(\cdot)$ is the output activation function (*e.g.* softmax, linear, etc.)

- Obviously $\underset{\sim}{a_i}$s and $\underset{\sim}{h_i}$s are functions of $\underset{\sim}{x}$, the input.

Figure 1: A Feedforward Neural Network with $L = 3$ and $p = 2$



3 Fundamentals

The fundamental components for a Feedforward Neural Network are

(1) Data

(2) Model

(3) Parameters

(4) Learning Algorithm

(5) Loss Function

Here we assume that we have a multi-class classification problem.

☞ We use **ReLU**(g) at our hidden units and **Softmax**(O) at the output units as activation functions; also we take **Average Sparse Categorical Cross-Entropy** as our loss function.

☞ If we stick with the architecture as in Figure 1, our neural network model will be

$$\underset{\sim}{\hat{y}} = O(W_3 \cdot g(W_2 \cdot g(W_1 \cdot \underset{\sim}{x} + \underset{\sim}{b_1}) + \underset{\sim}{b_2}) + \underset{\sim}{b_3})$$

where $\underset{\sim}{\hat{y}} \in \mathbb{R}^{p \times 1}$ and our parameters will be $W_1, W_2, W_3, \underset{\sim}{b_1}, \underset{\sim}{b_2}, \underset{\sim}{b_3}$.

☞ Average Sparse Categorical Cross-Entropy Loss function is given by

$$L = \frac{1}{m} \sum_{i=1}^{m} -\log \hat{y}_l$$

where $\hat{y}_l$ is the predicted probability of true class l.

☞ The learning algorithm will be **gradient descent**.

So our Data, Model, Parameters and Loss Function is completely ready. In gradient descent (learning algorithm), we will need the gradients provided by backpropagation which is up next.

4 Back-Propagation

4.1 Demo on a Thin Network

First let us take the simple case when we have a deep but thin network as in Figure 2.

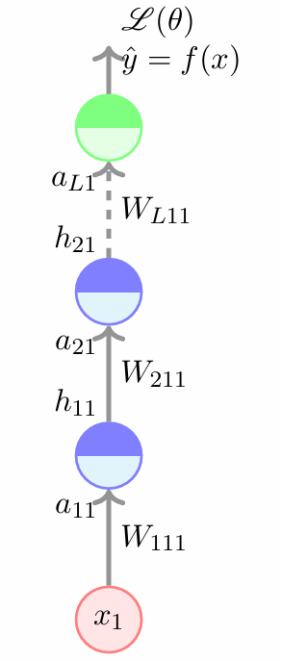


Figure 2: A Thin DNN

• Let us focus on this weight W_{111}. To learn this weight using gradient descent we need a formula for

$$\frac{\partial \mathscr{L}(\theta)}{\partial W_{111}}$$

.

• By successive application of chain rule, we can do

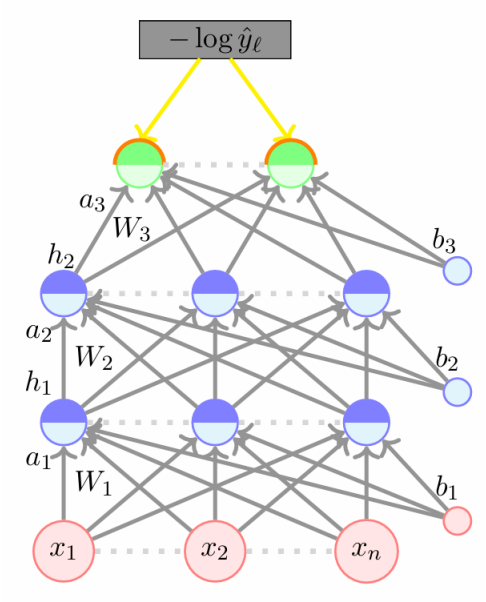
$$\frac{\partial \mathscr{L}(\theta)}{\partial W_{111}} = \frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{L11}} \frac{\partial a_{L11}}{\partial h_{21}} \frac{\partial h_{21}}{\partial a_{21}} \frac{\partial a_{21}}{\partial h_{11}} \frac{\partial h_{11}}{\partial a_{11}} \frac{\partial a_{11}}{\partial W_{111}}.$$

4.2 Intuition

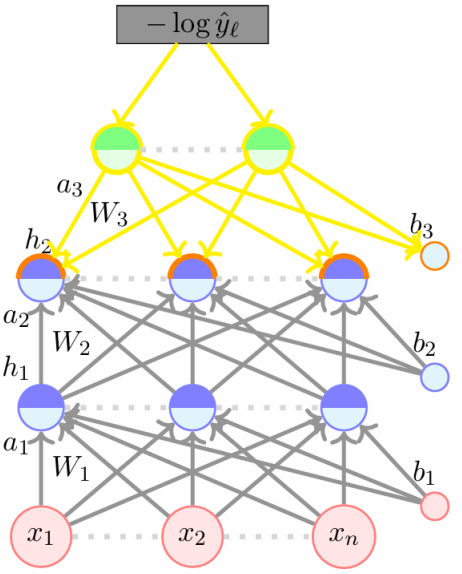
Let us see an intuitive explanation of backpropagation before we get into the mathematical details.

Recall that derivative says how much a small change in W_{111} changes $\mathscr{L}(\theta)$, in other words how much W_{111} is responsible for the loss $\mathscr{L}(\theta)$. The less sensitive W_{111} is, lesser will be the change in $\mathscr{L}(\theta)$ for a small change in W_{111}.

• We get a certain loss at the output and we try to figure out who is responsible for this loss.



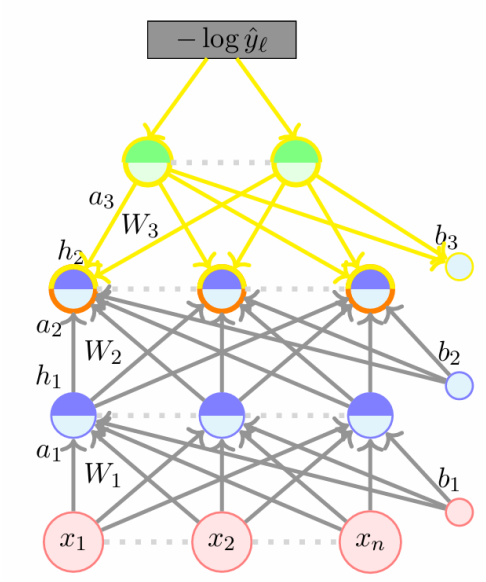
• So, we talk to the output layer and say "Hey! You are not producing the desired output, better take responsibility."



• The output layer says "Well, I take responsibility for my part but please understand that I am only as the good as the hidden layer and weights below me. After all

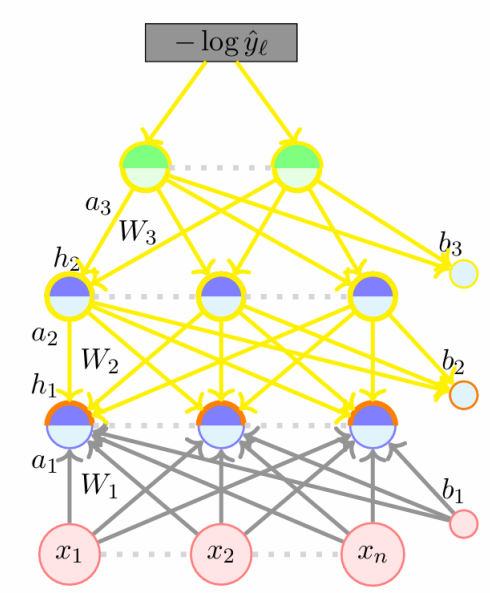
$$\hat{y} = O(W_L \cdot \underbrace{h_{L-1}} + \underbrace{b_L}).$$

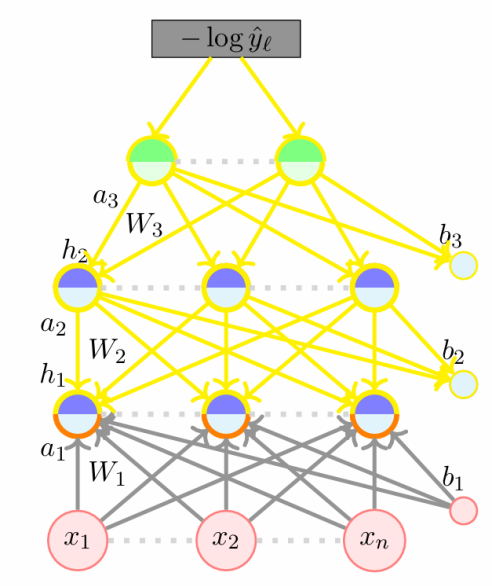
So, we talk to W_L, b_L and h_L and ask them "What is wrong with you ?"



• W_L and b_L take full responsibility but h_L says "Well, please understand that I am only as good as the pre-activation layer."

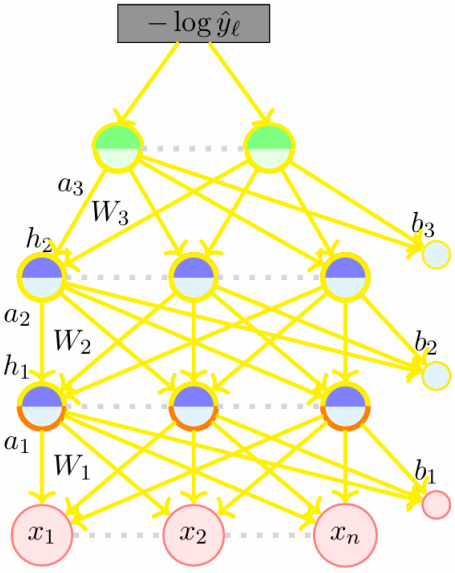
• The pre-activation layer in turn says that "I am only as good as the hidden layer and weights below me."





• We continue in this manner and realize that the responsibility lies with all the weights and biases (*i.e.* all the parameters of the model).

• But instead of talking to them directly, it is easier to talk to them through the hidden layers and output layers(and this is exactly what the chain rule allows us to do).



♠

$$\underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial W_{111}}}_{\text{Talk to the weight directly}} = \underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3}}_{\text{Talk to the output layer}} \underbrace{\frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_1}{\partial W_{111}}}_{\text{and now talk to the weights}}$$

4.3 The Real Deal

Here onwards our quantities of interest are

- gradients w.r.t. output units
- gradients w.r.t. hidden units
- gradients w.r.t. weights and biases

We first proceed assuming one data-point only, then we can always generalize for m data-points.

4.3.1 Gradients w.r.t. Output Units

$$\underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial W_{111}}}_{\text{Talk to the weight directly}} = \underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3}}_{\text{Talk to the output layer}} \underbrace{\frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_1}{\partial W_{111}}}_{\text{and now talk to the weights}}$$

$$\underset{\sim}{\hat{y}} = \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_p \end{pmatrix} \text{ and } \mathscr{L}(\theta) = -\log \hat{y}_\ell \ (\ell = \text{true class label})$$

Let us first consider the partial derivative w.r.t. i-th output.

$\forall\ i = 1(1)p,$

$$\frac{\partial}{\partial \hat{y}_i} \mathscr{L}(\theta) = \frac{\partial}{\partial \hat{y}_i}(-\log \hat{y}_\ell)$$

$$= \begin{cases} \frac{1}{\hat{y}_\ell} \text{ if } i = \ell \\ 0 \text{ otherwise} \end{cases}$$

$$= -\frac{\mathbb{1}_{i=\ell}}{\hat{y}_\ell} \text{ where } \mathbb{1}_{i=\ell} \text{ is an indicator function defined as } \mathbb{1}_{i=\ell} = \begin{cases} 1 \text{ if } i = \ell \\ 0 \text{ otherwise} \end{cases}$$

$$\therefore \ \frac{\partial}{\partial \hat{y}_i} \mathscr{L}(\theta) = -\frac{\mathbb{1}_{\ell=i}}{\hat{y}_\ell}.$$

We can now talk about the gradient w.r.t. the vector $\underset{\sim}{\hat{y}}$ *i.e.*

$$\nabla_{\underset{\sim}{\hat{y}}} \mathscr{L}(\theta)$$

(it will be a collection of individual partial derivatives).

$$\nabla_{\underset{\sim}{\hat{y}}} \mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}_1} \\ \frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}_2} \\ \vdots \\ \frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}_p} \end{bmatrix}$$

$$= -\frac{1}{\hat{y}_\ell} \begin{bmatrix} \mathbb{1}_{\ell=1} \\ \mathbb{1}_{\ell=2} \\ \vdots \\ \mathbb{1}_{\ell=p} \end{bmatrix}$$

- train labels are one-hot encoded and at a time exactly one of the elements in this column vector is 1 and rest are 0.
- If ℓ is the true class label, then this column vector is equal to $\underset{\sim}{e_\ell}$

$$= -\frac{1}{\hat{y}_\ell} \underset{\sim}{e_\ell}\,, \quad \underset{\sim}{e_\ell} \text{ is a } p\text{-dimensional vector whose } \ell\text{-th element is 1 and rest are 0.}$$

What we are actually interested in is

$$\frac{\partial \mathscr{L}(\theta)}{\partial a_{Li}} = \frac{\partial \left(-\log \hat{y}_\ell\right)}{\partial a_{Li}} = \frac{\partial \left(-\log \hat{y}_\ell\right)}{\partial \hat{y}_\ell} \cdot \frac{\partial \hat{y}_\ell}{\partial a_{Li}}$$

and $\hat{y}_\ell$ indeed depends on a_{Li} as $\hat{y}_\ell = \dfrac{exp\left(a_{L\ell}\right)}{\sum_{i=1}^{p} exp\left(a_{Li}\right)}$.

Now, $\dfrac{\partial}{\partial a_{Li}}\left(-\log \hat{y}_\ell\right) = -\dfrac{1}{\hat{y}_\ell} \cdot \dfrac{\partial \hat{y}_\ell}{\partial a_{Li}}$

$$= -\frac{1}{\hat{y}_\ell} \cdot \frac{\partial}{\partial a_{Li}} \operatorname{softmax}(\underset{\sim}{\mathbf{a}_L})_\ell$$

$$= -\frac{1}{\hat{y}_\ell} \cdot \frac{\partial}{\partial a_{Li}} \left(\frac{\exp(\underset{\sim}{\mathbf{a}_L})_\ell}{\sum_{i'=1}^{p} \exp(\underset{\sim}{\mathbf{a}_L})_{i'}} \right), \text{ recall } \frac{\partial}{\partial x}\left(\frac{g(x)}{h(x)}\right) = \frac{\frac{\partial g(x)}{\partial x} \cdot h(x) - g(x) \cdot \frac{\partial h(x)}{\partial x}}{(h(x))^2}$$

$$= -\frac{1}{\hat{y}_\ell} \cdot \left(\frac{\frac{\partial \exp(\underset{\sim}{\mathbf{a}_L})_\ell}{\partial a_{Li}}}{\sum_{i'=1}^{p} \exp(\underset{\sim}{\mathbf{a}_L})_{i'}} - \frac{\exp(\underset{\sim}{\mathbf{a}_L})_\ell \cdot \frac{\partial}{\partial a_{Li}} \sum_{i'=1}^{p} \exp(\underset{\sim}{\mathbf{a}_L})_{i'}}{\left(\sum_{i'=1}^{p} \exp(\underset{\sim}{\mathbf{a}_L})_{i'}\right)^2} \right)$$

$$= -\frac{1}{\hat{y}_\ell} \cdot \left(\frac{\mathbb{1}_{\ell=i} \exp(\underset{\sim}{\mathbf{a}_L})_\ell}{\sum_{i'=1}^{p} \exp(\underset{\sim}{\mathbf{a}_L})_{i'}} - \frac{\exp(\underset{\sim}{\mathbf{a}_L})_\ell \exp(\underset{\sim}{\mathbf{a}_L})_i}{\left(\sum_{i'=1}^{p} \exp(\underset{\sim}{\mathbf{a}_L})_{i'}\right)^2} \right)$$

$$= -\frac{1}{\hat{y}_\ell} \cdot \left(\mathbb{1}_{\ell=i} \operatorname{softmax}(\underset{\sim}{\mathbf{a}_L})_\ell - \operatorname{softmax}(\underset{\sim}{\mathbf{a}_L})_\ell \operatorname{softmax}(\underset{\sim}{\mathbf{a}_L})_i \right)$$

$$= -\frac{1}{\hat{y}_\ell} \cdot (\mathbb{1}_{\ell=i}\ \hat{y}_\ell - \hat{y}_\ell \cdot \hat{y}_i)$$

$$= -(\mathbb{1}_{\ell=i} - \hat{y}_i)$$

So far we have derived the partial derivative w.r.t. the i-th element of $\underset{\sim}{a_L}$ *i.e.*

$$\frac{\partial \mathscr{L}(\theta)}{\partial a_{Li}} = -(\mathbb{1}_{\ell=i} - \hat{y}_i)\,.$$

We can now write the gradient w.r.t. the vector $\underset{\sim}{a_L}$.

$$\nabla_{\underset{\sim}{a_L}} \mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial \mathscr{L}(\theta)}{\partial a_{L1}} \\ \frac{\partial \mathscr{L}(\theta)}{\partial a_{L2}} \\ \vdots \\ \frac{\partial \mathscr{L}(\theta)}{\partial a_{Lp}} \end{bmatrix}$$

$$= \begin{bmatrix} -(\mathbb{1}_{\ell=1} - \hat{y}_1) \\ -(\mathbb{1}_{\ell=2} - \hat{y}_2) \\ \vdots \\ -(\mathbb{1}_{\ell=p} - \hat{y}_p) \end{bmatrix}$$

$$= -\left(\underset{\sim}{e_\ell} - \underset{\sim}{\hat{y}}\right)$$

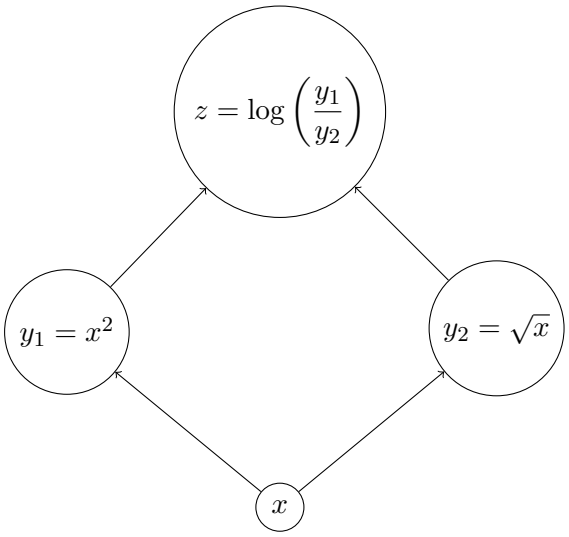
4.3.2 Gradients w.r.t. Hidden Units

$$\underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial W_{111}}}_{\text{Talk to the weight directly}} = \underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3}}_{\text{Talk to the output layer}} \underbrace{\frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_1}{\partial W_{111}}}_{\text{and now talk to the weights}}$$

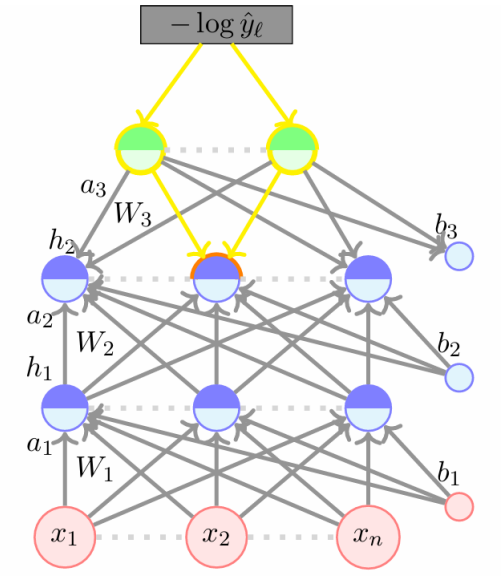
• **Chain Rule along multiple paths :** If a function $p(z)$ can be written as a function of intermediate results $q_i(z)$, then we have :

$$\frac{\partial p(z)}{\partial z} = \sum_m \frac{\partial p(z)}{\partial q_m(z)} \cdot \frac{\partial q_m(z)}{\partial z}$$

Here is an example.



$$\frac{dz}{dx} = \sum_{i=1}^{2} \frac{dz}{dy_i} \cdot \frac{dy_i}{dx}$$



First,

$$\frac{\partial \mathscr{L}(\theta)}{\partial h_{ij}} = \sum_{m=1}^{p} \frac{\partial \mathscr{L}(\theta)}{\partial a_{i+1,m}} \cdot \frac{\partial a_{i+1,m}}{\partial h_{ij}}$$

$$= \sum_{m=1}^{p} \frac{\partial \mathscr{L}(\theta)}{\partial a_{i+1,m}} \cdot W_{i+1,m,j} \text{ as } \underset{\sim}{a_{i+1}} = W_{i+1}\underset{\sim}{h_i} + \underset{\sim}{b_{i+1}}$$

Now consider these two vectors,

$$\nabla_{a_{i+1}} \mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial \mathscr{L}(\theta)}{\partial a_{i+1,1}} \\ \vdots \\ \frac{\partial \mathscr{L}(\theta)}{\partial a_{i+1,k}} \end{bmatrix} \quad ; \quad W_{i+1,\cdot,j} = \begin{bmatrix} W_{i+1,1,j} \\ \vdots \\ W_{i+1,k,j} \end{bmatrix}$$

$W_{i+1,\cdot,j}$ is the j-th column of W_{i+1}; see that,

$$(W_{i+1,\cdot,j})^T \nabla_{a_{i+1}} \mathscr{L}(\theta) = \sum_{m=1}^{k} \frac{\partial \mathscr{L}(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}$$

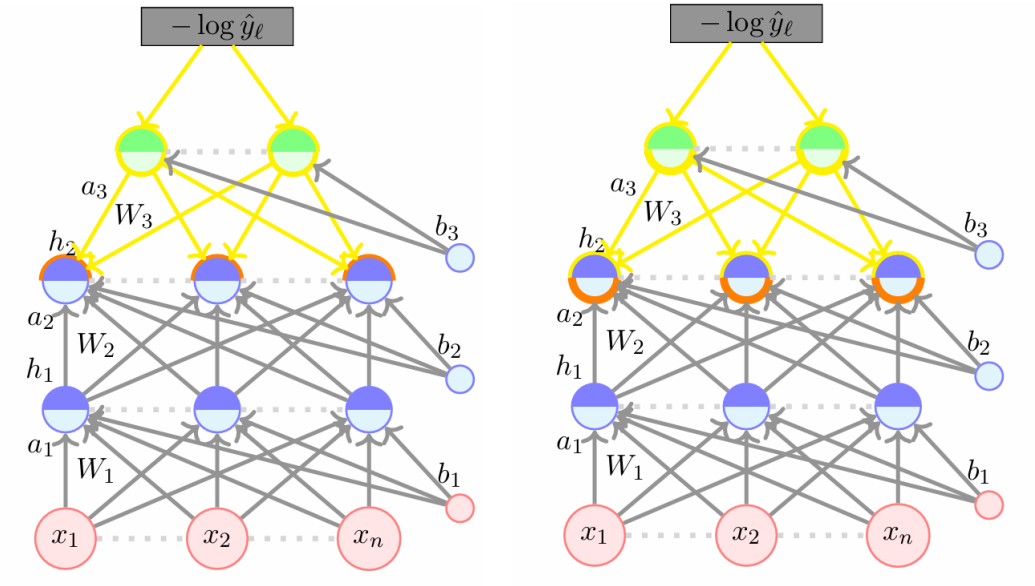


Table 1: Gradients w.r.t. hidden layers

We have,

$$\frac{\partial \mathscr{L}(\theta)}{\partial h_{ij}} = (W_{i+1,\cdot,j})^T \nabla_{a_{i+1}} \mathscr{L}(\theta)$$

We can now write the gradient w.r.t. $\underset{\sim}{h_i}$ (left figure in Table 1).

$$\nabla_{\mathbf{h}_i} \mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial \mathscr{L}(\theta)}{\partial h_{i1}} \\ \frac{\partial \mathscr{L}(\theta)}{\partial h_{i2}} \\ \vdots \\ \frac{\partial \mathscr{L}(\theta)}{\partial h_{in}} \end{bmatrix} = \begin{bmatrix} (W_{i+1,\cdot,1})^T \nabla_{a_{i+1}} \mathscr{L}(\theta) \\ (W_{i+1,\cdot,2})^T \nabla_{a_{i+1}} \mathscr{L}(\theta) \\ \vdots \\ (W_{i+1,\cdot,n})^T \nabla_{a_{i+1}} \mathscr{L}(\theta) \end{bmatrix} = (W_{i+1})^T \left(\nabla_{a_{i+1}} \mathscr{L}(\theta) \right)$$

- We are almost done except that we do not know how to calculate $\nabla_{a_{i+1}} \mathscr{L}(\theta)$ for $i < L-1$.
- We will see how to compute that.

$$\nabla_{a_i} \mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial \mathscr{L}(\theta)}{\partial a_{i1}} \\ \vdots \\ \frac{\partial \mathscr{L}(\theta)}{\partial a_{in}} \end{bmatrix} \text{ (right figure in Table 1).}$$

Now, $\forall\ i = 1(1)\overline{L-1}$,

$$\frac{\partial \mathscr{L}(\theta)}{\partial a_{ij}} = \frac{\partial \mathscr{L}(\theta)}{\partial h_{ij}} \cdot \frac{\partial h_{ij}}{\partial a_{ij}} = \frac{\partial \mathscr{L}(\theta)}{\partial h_{ij}} g'(a_{ij}) \qquad [\because h_{ij} = g(a_{ij})]$$

$$\therefore \nabla_{a_i} \mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial \mathscr{L}(\theta)}{\partial h_{i1}} g'(a_{i1}) \\ \vdots \\ \frac{\partial \mathscr{L}(\theta)}{\partial h_{in}} g'(a_{in}) \end{bmatrix} = \nabla_{h_i} \mathscr{L}(\theta) \odot \left[\cdots, g'(a_{ik}), \cdots\right]$$

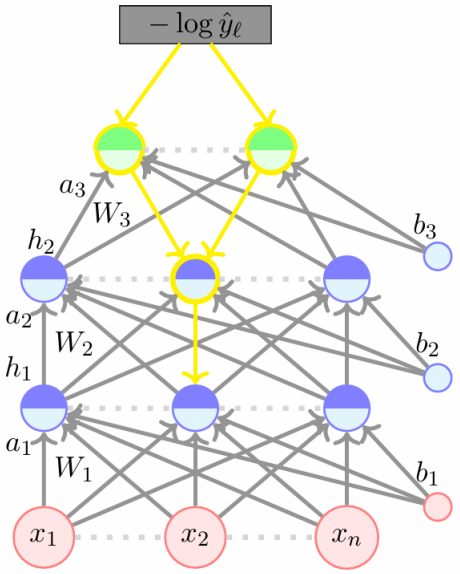
4.3.3 Gradients w.r.t. Parameters *i.e.* Weights and Biases

Finally,

$$\underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial W_{111}}}_{\text{Talk to the weight directly}} = \underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3}}_{\text{Talk to the output layer}} \underbrace{\frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_1}{\partial W_{111}}}_{\text{and now talk to the weights}}$$

Recall that, $\underbrace{a_{i+1}} = W_{i+1} \underbrace{h_i} + \underbrace{b_{i+1}}$.

$$\therefore \frac{\partial a_{ki}}{\partial W_{kij}} = h_{k-1,j}$$



$$\frac{\partial \mathscr{L}(\theta)}{\partial W_{kij}} = \frac{\partial \mathscr{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial W_{kij}}$$

$$= \frac{\partial \mathscr{L}(\theta)}{\partial a_{ki}} h_{k-1,j}$$

$$\nabla_{W_k} \mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial \mathscr{L}(\theta)}{\partial W_{k11}} & \frac{\partial \mathscr{L}(\theta)}{\partial W_{k12}} & \cdots & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{k1n}} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \frac{\partial \mathscr{L}(\theta)}{\partial W_{kn1}} & \frac{\partial \mathscr{L}(\theta)}{\partial W_{kn2}} & \cdots & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{knn}} \end{bmatrix}, \text{ we assume } W_k \in \mathbb{R}^{n \times n}$$

Lets take a simple example of a $W_k \in \mathbb{R}^{3 \times 3}$ and see what each entry looks like

$$\nabla_{W_k}\mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial\mathscr{L}(\theta)}{\partial W_{k11}} & \frac{\partial\mathscr{L}(\theta)}{\partial W_{k12}} & \frac{\partial\mathscr{L}(\theta)}{\partial W_{k13}} \\ \frac{\partial\mathscr{L}(\theta)}{\partial W_{k21}} & \frac{\partial\mathscr{L}(\theta)}{\partial W_{k22}} & \frac{\partial\mathscr{L}(\theta)}{\partial W_{k23}} \\ \frac{\partial\mathscr{L}(\theta)}{\partial W_{k31}} & \frac{\partial\mathscr{L}(\theta)}{\partial W_{k32}} & \frac{\partial\mathscr{L}(\theta)}{\partial W_{k33}} \end{bmatrix} = \frac{\partial\mathscr{L}(\theta)}{\partial a_{ki}}\frac{\partial a_{ki}}{\partial W_{kij}}$$

$$\therefore \nabla_{W_k}\mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial\mathscr{L}(\theta)}{\partial a_{k1}} h_{k-1,1} & \frac{\partial\mathscr{L}(\theta)}{\partial a_{k1}} h_{k-1,2} & \frac{\partial\mathscr{L}(\theta)}{\partial a_{k1}} h_{k-1,3} \\ \frac{\partial\mathscr{L}(\theta)}{\partial a_{k2}} h_{k-1,1} & \frac{\partial\mathscr{L}(\theta)}{\partial a_{k2}} h_{k-1,2} & \frac{\partial\mathscr{L}(\theta)}{\partial a_{k2}} h_{k-1,3} \\ \frac{\partial\mathscr{L}(\theta)}{\partial a_{k3}} h_{k-1,1} & \frac{\partial\mathscr{L}(\theta)}{\partial a_{k3}} h_{k-1,2} & \frac{\partial\mathscr{L}(\theta)}{\partial a_{k3}} h_{k-1,3} \end{bmatrix} = \nabla_{a_k}\mathscr{L}(\theta) \cdot \underset{\sim}{h_{k-1}}^T$$

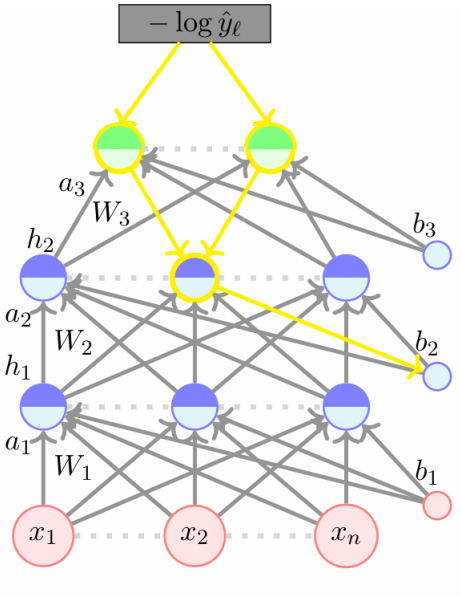
Finally, coming to the biases

$$a_{ki} = b_{ki} + \sum_j W_{kij} h_{k-1,j}$$

$$\frac{\partial\mathscr{L}(\theta)}{\partial b_{ki}} = \frac{\partial\mathscr{L}(\theta)}{\partial a_{ki}} \cdot \frac{\partial a_{ki}}{\partial b_{ki}} = \frac{\partial\mathscr{L}(\theta)}{\partial a_{ki}}$$

We can now write the gradient w.r.t. the vector $\underset{\sim}{b_k}$.

$$\nabla_{\mathbf{b}_k}\mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial\mathscr{L}(\theta)}{\partial a_{k1}} \\ \frac{\partial\mathscr{L}(\theta)}{\partial a_{k2}} \\ \vdots \\ \frac{\partial\mathscr{L}(\theta)}{\partial a_{kn}} \end{bmatrix} = \nabla_{\mathbf{a}_k}\mathscr{L}(\theta)$$



5 Summary

5.1 for a single data-point

- $\nabla W_k = \nabla \underset{\sim}{a_k} \cdot \underset{\sim}{h_{k-1}}^T \; \forall \; k = 1(1)L$
- $\nabla b_k = \nabla a_k \; \forall \; k = 1(1)L$
- $\nabla a_L = -\left(\underset{\sim}{e_\ell} - \underset{\sim}{\hat{y}}\right)$
- $\nabla a_k = \nabla h_k \odot g'(\underset{\sim}{a_k}) = W_{k+1}^T \cdot \nabla a_{k+1} \odot g'(\underset{\sim}{a_k}) \; \forall \; k = 1(1)\overline{L-1}$

5.2 for whole X

Given:

- $X \in \mathbb{R}^{n \times m}$ — input data with examples as columns
- m — number of examples
- L — number of layers

The forward propagation equations are:

- $A_0 = X$
- $A_k = W_k H_{k-1} + \underset{\sim}{b_k} \cdot \mathbf{1}_m^T \quad \forall\ k = 1, 2, \ldots, L$
- $H_k = g_k(A_k) \quad \forall\ k = 1, 2, \ldots, L-1$
- $\hat{Y} = \text{softmax}(A_L)$

The backward propagation equations are:

- $\nabla W_k = \nabla A_k \cdot H_{k-1}^T \ \forall\ k = 1(1)L$
- $\nabla b_k = \nabla A_k \cdot \mathbf{1}_m \ \forall\ k = 1(1)L$ where $\mathbf{1}_m \in \mathbb{R}^{m \times 1}$
- $\nabla A_L = \frac{1}{m}\left(\hat{Y} - E\right)$ where $E \in \mathbb{R}^{p \times m}$ is the one-hot encoded label matrix for all the data points
- $\nabla A_k = \nabla H_k \odot g'(A_k) = \left(W_{k+1}^T \cdot \nabla A_{k+1}\right) \odot g'(A_k) \ \forall\ k = 1(1)\overline{L-1}$

6 Appendix

6.1 Derivative of Activation Function

Logistic function

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

tanh

$$\begin{aligned} g(z) &= \tanh(z) \\ &= \frac{e^z - e^{-z}}{e^z + e^{-z}} \\ g'(z) &= \frac{\left((e^z+e^{-z})\frac{d}{dz}(e^z-e^{-z}) \right.}{\left. -(e^z-e^{-z})\frac{d}{dz}(e^z+e^{-z})\right)} \\ &\qquad (e^z+e^{-z})^2 \\ &= \frac{(e^z+e^{-z})^2 - (e^z-e^{-z})^2}{(e^z+e^{-z})^2} \\ &= 1 - \frac{(e^z-e^{-z})^2}{(e^z+e^{-z})^2} \\ &= 1 - (g(z))^2 \end{aligned}$$