1 Computation for a multi-class classification neural net

Let $D_n = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(n)}, y^{(n)})\}$ be the dataset with $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \{1, \ldots, m\}$ indicating the class within m classes. For vectors and matrices in the following equations, vectors are by default considered to be column vectors.

Consider a neural net of the type Multilayer perceptron (MLP) with only one hidden layer (meaning 3 layers total if we count the input and output layers). The hidden layer is made of d_h neurons fully connected to the input layer. We shall consider a non linearity of type rectifier, called Leaky RELU with parameter $\alpha < 1$ (Leaky Rectified Linear Unit) for the hidden layer, defined as follows:

$$LeakyRELU_\alpha(x) = \max(x, \alpha x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{otherwise} \end{cases}$$

The output layer is made of m neurons that are fully connected to the hidden layer. They are equipped with a softmax non linearity. The output of the j^{th} neuron of the output layer gives a score for the j-th class which can be interpreted as the probability of x being of class j.

1. Write the derivative of the sigmoid function, σ', using the σ function only

$$\sigma(x) = \left(\frac{1}{1+e^{-x}}\right) = \left(1+e^{-x}\right)^{-1} \tag{1}$$

In terms of $\sigma(x)$:

$$\sigma(x) = \left(1+e^{-x}\right)^{-1}$$
$$\sigma^{-1}(x) = 1 + e^{-x}$$

$$e^{-x} = \sigma^{-1}(x) - 1 \tag{2}$$

The derivative is:

$$\sigma'(x) = \frac{d\sigma(x)}{dx} = \left(-\left(1+e^{-x}\right)^{-2}\right) \times \left(-e^{-x}\right)$$

$$= \left((1+e^{-x})^{-2}\right) \times (e^{-x}) = \left((1+e^{-x})^{-1}\right)^2 \times (e^{-x})$$

Substitute (1) and (2):

$$\sigma'(x) = \left((1+e^{-x})^{-1}\right)^2 \times (e^{-x}) = \sigma^2(x) \times \left(\sigma^{-1}(x) - 1\right)$$

The final answer is:

$$\sigma'(x) = \sigma(x) - \sigma^2(x)$$

2. Write the derivative of the hyperbolic tangent function, $\tanh'$, using the tanh function only

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{u}{v} \tag{3}$$

$$\tanh'(x) = \frac{d(\tanh(x))}{dx} = \frac{vu' + uv'}{vv}$$

$$\begin{cases} u = e^x - e^{-x}. \\ v = e^x + e^{-x} \\ u' = e^x + e^{-x} = v \\ v' = e^x - e^{-x} = u \end{cases} \tag{4}$$

From (4),

$$\tanh'(x) = \frac{vu' + uv'}{vv} = \frac{vv + uu}{vv}$$

$$= \frac{vv}{vv} - \frac{uu}{vv} = 1 - \left(\frac{u}{v}\right)^2$$

And from (3),

$$\tanh'(x) = 1 - \left(\frac{u}{v}\right)^2 = 1 - (\tanh(x))^2$$

Therefore,

$$\tanh'(x) = 1 - \tanh^2(x)$$

3. Write the derivative of the rectifier function, rect′. Note: its derivative at 0 is undefined, but rect′ can return 0 at 0.

$$rect(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

Therefore,

$$rect'(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \implies rect'(x) = \mathbb{1}_{\{x>0\}}(x)$$

4. Let the squared L_2 norm of a vector be: $\|\mathbf{x}\|_2^2 = \sum_i \mathbf{x}_i^2$. Write the the gradient of the square of the L_2 norm function, $\frac{\partial \|\mathbf{x}\|_2^2}{\partial \mathbf{x}}$, in vector form.

$$\frac{\partial \sum_i x_i^2}{\partial x} = \begin{bmatrix} \frac{\partial (x_1^2 + x_2^2 + \cdots + x_n^2)}{\partial x_1} \\ \vdots \\ \frac{\partial (x_1^2 + x_2^2 + \cdots + x_n^2)}{\partial x_n} \end{bmatrix}$$

$$= \begin{bmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_n \end{bmatrix} = 2 \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = 2x$$

5. Let the norm L_1 of a vector be: $\|\mathbf{x}\|_1 = \sum_i |\mathbf{x}_i|$. Write the gradient of the L_1 norm function, $\frac{\partial \|\mathbf{x}\|_1}{\partial \mathbf{x}}$, in vector form.

$$\frac{\partial \sum_i |x_i|}{\partial x} = \begin{bmatrix} \frac{\partial (|x_1| + |x_2| + \cdots + |x_n|)}{\partial x_1} \\ \vdots \\ \frac{\partial (|x_1| + |x_2| + \cdots + |x_n|)}{\partial x_n} \end{bmatrix}$$

$$= \begin{bmatrix} sign(x_1) \\ sign(x_2) \\ \vdots \\ sign(x_n) \end{bmatrix} = sign(x)$$

6. Let $\mathbf{W}^{(1)}$ be a $d_h \times d$ matrix of weights and $\mathbf{b}^{(1)}$ the bias vector be the connections between the input layer and the hidden layer. What is the dimension of $\mathbf{b}^{(1)}$? Give the formula of the pre-activation vector (before the non linearity) of the neurons of the hidden layer $\mathbf{h}^a$ given $\mathbf{x}$ as input, first in a matrix form ($\mathbf{h}^a = \ldots$), and then details on how to compute one element $\mathbf{h}^a_j = \ldots$. Write the output vector of the hidden layer $\mathbf{h}^s$ with respect to $\mathbf{h}^a$.

   $b^{(1)} \in \mathbb{R}^{d_h}$
   $h^a = b^{(1)} + W^{(1)}x$

$$h^a = \begin{pmatrix} b_1^{(1)} \\ \vdots \\ b_{d_h}^{(1)} \end{pmatrix} + \begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \ldots & w_{1d}^{(1)} \\ \vdots & \ldots & \ldots & \vdots \\ w_{d_h1}^{(1)} & w_{d_h2}^{(1)} & \ldots & w_{d_hd}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$$

$$= \begin{pmatrix} b_1^{(1)} + w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2 + \cdots + w_{1d}^{(1)}x_d \\ \vdots \\ b_{d_h}^{(1)} + w_{d_h1}^{(1)}x_1 + w_{d_h2}^{(1)}x_2 + \cdots + w_{d_hd}^{(1)}x_d \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$$

   $\implies h_j^a = b_j^{(1)} + \sum_{i=1}^{d} w_{ji}^{(1)} x_i$

   $h^s = LeakyRELU_\alpha(h^a)$

7. Let $\mathbf{W}^{(2)}$ be a weight matrix and $\mathbf{b}^{(2)}$a bias vector be the connections between the hidden layer and the output layer. What are the dimensions of $\mathbf{W}^{(2)}$ and $\mathbf{b}^{(2)}$? Give the formula of the activation function of the neurons of the output layer $\mathbf{o}^a$ with respect to their input $\mathbf{h}^s$ in a matrix form and then write in a detailed form for $\mathbf{o}^a_k$.

   $W^{(2)}$ is m x d_h and $b^{(2)} \in \mathbb{R}^m$
   $o^a = b^{(2)} + W^{(2)}h^s$

$$o^a = \begin{pmatrix} b_1^{(2)} \\ \vdots \\ b_m^{(2)} \end{pmatrix} + \begin{pmatrix} w_{11}^2 & w_{12}^2 & \ldots & w_{1d_h}^2 \\ \vdots & \ldots & \ldots & \vdots \\ w_{m1}^2 & w_{m2}^2 & \ldots & w_{md_h}^2 \end{pmatrix} \begin{pmatrix} h_1^s \\ \vdots \\ h_{d_h}^s \end{pmatrix}$$

$\implies o_k^a = b_k^{(2)} + \sum_{i=1}^{d_h} w_{ki}^{(2)} h_i^s$

8. The output of the neurons at the output layer is given by:

$$\mathbf{o}^s = \text{softmax}(\mathbf{o}^a)$$

Give the precise equation for $\mathbf{o}_k^s$ as a function of $\mathbf{o}_j^a$. **Show** that the $\mathbf{o}_k^s$ are positive and sum to 1. Why is this important?

$o_k^s = softmax(o^a)_k = \frac{exp(o_k^a)}{\sum_{i=1}^m exp(o_i^a)}$

By definition of $exp(x)$, we know that

$$\forall x \in \mathbb{R} \quad exp(x) > 0$$

Also, we know that

$$\forall a, b > 0 \quad \frac{a}{b} > 0$$

Therefore, the o_k^s are positive.

$\sum_{k=1}^m o_k^s = \sum_{k=1}^m \frac{exp(o_k^a)}{\sum_{i=1}^m exp(o_i^a)} = \frac{\sum_{k=1}^m exp(o_k^a)}{\sum_{i=1}^m exp(o_i^a)} = 1$

It is important that o_k^s are positive and that they sum to 1, because it allows to interpret o_k^s as P(Y = k|X=x) (i.e. to interpret o_k^s as the probability of x being of class k).

9. The neural net computes, for an input vector $\mathbf{x}$, a vector of probability scores $\mathbf{o}^s(\mathbf{x})$. The probability, computed by a neural net, that an observation $\mathbf{x}$ belong to class y is given by the y^{th} output $\mathbf{o}_y^s(\mathbf{x})$. This suggests a loss function such as:

$$L(\mathbf{x}, y) = \text{-log } \mathbf{o}_y^s(\mathbf{x})$$

Find the equation of L as a function of the vector $\mathbf{o}^a$. It is easily achievable with the correct substitution using the equation of the previous question.

$$
\begin{aligned}
L(x, y) &= -log(\frac{exp(o_y^a)}{\sum_{i=1}^m exp(o_i^a)}) \\
&= log(\sum_{i=1}^m exp(o_i^a)) - log(exp(o_y^a)) \\
&= log(\sum_{i=1}^m exp(o_i^a)) - o_y^a
\end{aligned}
$$

10. The training of the neural net will consist of finding parameters that minimize the empirical risk $\hat{R}$ associated with this loss function. What is $\hat{R}$? What is precisely the set θ of parameters of the network? How many scalar parameters n_θ are there? Write down the optimization problem of training the network in order to find the optimal values for these parameters.

    $\hat{R} = \frac{1}{n}\sum_{i=1}^n L(x^{(i)}, y^{(i)}) = \frac{1}{n}\sum_{i=1}^n [log(\sum_{j=1}^m exp(o_j^a(x^{(i)}))) - o_{y^{(i)}}^a(x^{(i)})]$

    $\theta = \{W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}\}$

    $n_\theta = d_h \text{ x } d + d_h + m \text{ x } d_h + m$

    The optimization problem of training the network in order to find the optimal values for these parameters is $\arg\min_\theta \hat{R}(\theta, D_{train})$

11. To find a solution to this optimization problem, we will use gradient descent. What is the (batch) gradient descent equation for this problem?

    Initialize θ
    for N iteration:
    $\theta \leftarrow \theta - \eta(\frac{1}{n}\sum_{i=1}^n \frac{\partial}{\partial\theta}(log(\sum_{j=1}^m exp(o_j^a(x^{(i)}))) - o_{y^{(i)}}^a(x^{(i)})))$

12. We can compute the vector of the gradient of the empirical risk $\hat{R}$ with respect to the parameters set θ this way

$$
\begin{pmatrix} \frac{\partial \hat{R}}{\partial \theta_1} \\ \vdots \\ \frac{\partial \hat{R}}{\partial \theta_{n_\theta}} \end{pmatrix} = \frac{1}{n}\sum_{i=1}^n \begin{pmatrix} \frac{\partial L(\mathbf{x}_i, y_i)}{\partial \theta_1} \\ \vdots \\ \frac{\partial L(\mathbf{x}_i, y_i)}{\partial \theta_{n_\theta}} \end{pmatrix}
$$

This hints that we only need to know how to compute the gradient of the loss L with an example$(\mathbf{x}, y)$ with respect to the parameters, defined as followed:

$$\frac{\partial L}{\partial \theta} = \begin{pmatrix} \frac{\partial L}{\partial \theta_1} \\ \vdots \\ \frac{\partial L}{\partial \theta_{n_\theta}} \end{pmatrix} = \begin{pmatrix} \frac{\partial L(\mathbf{x},y)}{\partial \theta_1} \\ \vdots \\ \frac{\partial L(\mathbf{x},y)}{\partial \theta_{n_\theta}} \end{pmatrix}$$

We shall use gradient backpropagation, starting with loss L and going to the output layer $\mathbf{o}$ then down the hidden layer $\mathbf{h}$ then finally at the input layer $\mathbf{x}$. Show that

$$\frac{\partial L}{\partial \mathbf{o}^a} = \mathbf{o}^s - \text{onehot}_m(y)$$

For k $\neq$ y,

$$\begin{aligned}
\frac{\partial L(x,y)}{\partial o_k^a} &= \frac{\partial (log(\sum_{j=1}^m exp(o_j^a)) - o_y^a)}{\partial o_k^a} \\
&= \frac{\partial log(\sum_{j=1}^m exp(o_j^a))}{\partial o_k^a} \\
&= \frac{1}{\sum_{j=1}^m exp(o_j^a)} \times \frac{\partial \sum_{j=1}^m exp(o_j^a)}{\partial o_k^a} \\
&= \frac{exp(o_k^a)}{\sum_{j=1}^m exp(o_j^a)} \\
&= o_k^s
\end{aligned}$$

For y,

$$\begin{aligned}
\frac{\partial L(x,y)}{\partial o_y^a} &= \frac{\partial (log(\sum_{j=1}^m exp(o_j^a)) - o_y^a)}{\partial o_y^a} \\
&= \frac{exp(o_y^a)}{\sum_{j=1}^m exp(o_j^a)} - 1 \\
&= o_y^s - 1
\end{aligned}$$

Therefore,

$$\frac{\partial L(x,y)}{\partial o^a} = \begin{pmatrix} \frac{\partial L}{\partial o_1^a} \\ \cdots \\ \frac{\partial L}{\partial o_m^a} \end{pmatrix}$$

$$= \begin{pmatrix} o_1^s \\ \vdots \\ o_y^s \\ \vdots \\ o_m^s \end{pmatrix} - \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

$$= o^s - onehot_m(y)$$

13. Compute the gradients with respect to parameters $\mathbf{W}^{(2)}$ and $\mathbf{b}^{(2)}$ of the output layer. Since L depends on $\mathbf{W}_{kj}^{(2)}$ and $\mathbf{b}_k^{(2)}$only through $\mathbf{o}_k^a$ the result of the chain rule is:

$$\frac{\partial L}{\partial \mathbf{W}_{kj}^{(2)}} = \frac{\partial L}{\partial \mathbf{o}_k^a} \frac{\partial \mathbf{o}_k^a}{\partial \mathbf{W}_{kj}^{(2)}}$$

and

$$\frac{\partial L}{\partial \mathbf{b}_k^{(2)}} = \frac{\partial L}{\partial \mathbf{o}_k^a} \frac{\partial \mathbf{o}_k^a}{\partial \mathbf{b}_k^{(2)}}$$

For $k \neq y$,
$\frac{\partial L(x,y)}{\partial W_{kj}^{(2)}} = o_k^s \times \frac{\partial o_k^a}{\partial W_{kj}^{(2)}} = o_k^s \times \frac{\partial (b_k^{(2)} + \sum_{i=1}^{d_h} W_{ki}^{(2)} h_i^s)}{\partial W k_{kj}^{(2)}} = o_k^s h_j^s$

For $k = y$,
$\frac{\partial L(x,y)}{\partial W_{kj}^{(2)}} = (o_y^s - 1) \times \frac{\partial o_y^a}{\partial W_{yj}^{(2)}} = (o_y^s - 1) h_j^s = o_y^s h_j^s - h_j^s$

For $k \neq y$,
$\frac{\partial L(x,y)}{\partial b_k^{(2)}} = o_k^s \times \frac{\partial (b_k^{(2)} + \sum_{i=1}^{d_h} W_{ki}^{(2)} h_i^s)}{\partial b_k^{(2)}} = o_k^s$

For $k = y$,
$\frac{\partial L(x,y)}{\partial b_k^{(2)}} = o_k^s - 1$
Therefore,

$$\frac{\partial L}{\partial W^{(2)}} = \begin{pmatrix} o_1^s h_1^s & o_1^s h_2^s & \dots & o_1^s h_{d_h}^s \\ \vdots & \dots & \dots & \vdots \\ o_m^s h_1^s & o_m^s h_2^s & \dots & o_m^s h_{d_h}^s \end{pmatrix} - \begin{array}{cc} \begin{pmatrix} 0 & \dots & \dots & 0 \\ \vdots & \dots & \dots & \vdots \\ h_1^s & h_2^s & \dots & h_{d_h}^s \\ \vdots & \dots & \dots & \vdots \\ 0 & \dots & \dots & 0 \end{pmatrix} & \begin{array}{c} \\ \\ \leftarrow \text{ line y} \\ \\ \\ \end{array} \end{array}$$

$$\frac{\partial L}{\partial b^{(2)}} = \begin{pmatrix} o_1^s \\ \vdots \\ o_y^s \\ \vdots \\ o_m^s \end{pmatrix} - \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = o^s - onehot_m(y)$$

14. Write down the gradient of the last question in matrix form and define the dimensions of all matrix or vectors involved.

$$\frac{\partial L}{\partial b^{(2)}} = o^s - onehot_m(y)$$

$$
\begin{aligned}
\frac{\partial L}{\partial W^{(2)}} &= o^s {h^s}^T - \begin{pmatrix} 0 & \dots & \dots & 0 \\ \vdots & \dots & \dots & \vdots \\ h^s_1 & h^s_2 & \dots & h^s_{d_h} \\ \vdots & \dots & \dots & \vdots \\ 0 & \dots & \dots & 0 \end{pmatrix} \\
&= o^s {h^s}^T - \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} h^s_1 & h^s_2 & \dots & h^s_{d_h} \end{pmatrix} \\
&= o^s {h^s}^T - onehot_m(y) {h^s}^T
\end{aligned}
$$

$\frac{\partial L}{\partial b^{(2)}}$ is of dimension m x 1
$\frac{\partial L}{\partial W^{(2)}}$ is m x d_h

where o^s and $onehot_m(y)$ are of dimension m x 1, h^s is d_h x 1 and the matrix resulting from the outer product of $onehot_m(y)$ and h^s is m x d_h

15. What is the partial derivative of the loss L with respect to the output of the neurons at the hidden layer? Since L depends on $\mathbf{h}^s_j$ only through the activations of the output neurons $\mathbf{o}^a$ the chain rule yields:

$$
\frac{\partial L}{\partial \mathbf{h}^s_j} = \sum_{k=1}^{m} \frac{\partial L}{\partial \mathbf{o}^a_k} \frac{\partial \mathbf{o}^a_k}{\partial \mathbf{h}^s_j}
$$

$$
\begin{aligned}
\frac{\partial L}{\partial h_j^s} &= \sum_{k=1}^{m} \frac{\partial L}{\partial o_k^a} \frac{\partial o_k^a}{\partial h_j^s} \\
&= \sum_{k=1}^{m} \frac{\partial L}{\partial o_k^a} \frac{\partial (b^{(2)} + \sum_{i=1}^{d_h} W_{ki}^{(2)} h_i^s)}{\partial h_j^s} \\
&= \sum_{k=1}^{m} \frac{\partial L}{\partial o_k^a} W_{kj}^{(2)} \\
&= o_1^s W_{1j}^{(2)} + o_2^s W_{2j}^{(2)} + \cdots + (o_y^s - 1) W_{yj}^{(2)} + \cdots + o_m^s W_{mj}^{(2)} \\
&= \sum_{k=1}^{m} o_k^s W_{kj}^{(2)} - W_{yj}^{(2)}
\end{aligned}
$$

Therefore,

$$
\frac{\partial L}{\partial h^s} = \begin{pmatrix} \sum_{k=1}^{m} o_k^s W_{k1}^{(2)} - W_{y1}^{(2)} \\ \sum_{k=1}^{m} o_k^s W_{k2}^{(2)} - W_{y2}^{(2)} \\ \vdots \\ \sum_{k=1}^{m} o_k^s W_{kd_h}^{(2)} - W_{yd_h}^{(2)} \end{pmatrix}
$$

16. Write down the gradient of the last question in matrix form and define the dimensions of all matrix or vectors involved.
$\frac{\partial L}{\partial h^s} = W^{(2)^T} o^s - (onehot_m(y)^T W^{(2)})^T$

where
$\frac{\partial L}{\partial h^s}$ is of dimensions d_h x 1
$W^{(2)^T}$ is d_h x m
o^s is m x 1
$onehot_m(y)^T$ is 1 x m
$(onehot_m(y)^T W^{(2)})^T$ is d_h x 1

17. What is the partial derivative of the loss L with respect to the activation of the neurons at the hidden layer? Since L depends on the

activation $\mathbf{h}_j^a$ only through $\mathbf{h}_j^s$ of this neuron, the chain rule gives:

$$\frac{\partial L}{\partial \mathbf{h}_j^a} = \frac{\partial L}{\partial \mathbf{h}_j^s} \frac{\partial \mathbf{h}_j^s}{\partial \mathbf{h}_j^a}$$

Note $\mathbf{h}_j^s = \mathrm{LeakyRELU}_\alpha(\mathbf{h}_j^a)$: the leaky rectifier function is applied element-wise. Start by writing the derivative of the rectifier function $\frac{\partial\, \mathrm{LeakyRELU}_\alpha(z)}{\partial z} = \mathrm{LeakyRELU}_\alpha{}'(z) = \ldots$.

$$\begin{aligned}\frac{\partial\, \mathrm{LeakyRELU}_\alpha(z)}{\partial z} &= \begin{cases} 1 & \text{if } z \geq 0 \\ \alpha & \text{otherwise} \end{cases} \\ &= \mathbb{1}_{\{z \geq 0\}}(z) + \alpha \mathbb{1}_{\{z<0\}}(z)\end{aligned}$$

$$\begin{aligned}\frac{\partial L}{\partial h_j^a} &= \frac{\partial L}{\partial h_j^s} \times \frac{\partial h_j^s}{\partial h_j^a} \\ &= (\sum_{k=1}^{m} o_k^s W_{kj}^{(2)} - W_{yj}^{(2)}) \frac{\partial(LeakyRELU_\alpha(h_j^a))}{\partial h_j^a} \\ &= (\sum_{k=1}^{m} o_k^s W_{kj}^{(2)} - W_{yj}^{(2)})(\mathbb{1}_{\{h_j^a \geq 0\}}(h_j^a) + \alpha \mathbb{1}_{\{h_j^a<0\}}(h_j^a))\end{aligned}$$

therefore,

$$\frac{\partial L}{\partial h^a} = \begin{pmatrix} (\sum_{k=1}^{m} o_k^s W_{k1}^{(2)} - W_{y1}^{(2)})(\mathbb{1}_{\{h_1^a \geq 0\}}(h_1^a) + \alpha \mathbb{1}_{\{h_1^a<0\}}(h_1^a)) \\ \vdots \\ (\sum_{k=1}^{m} o_k^s W_{kd_h}^{(2)} - W_{yd_h}^{(2)})(\mathbb{1}_{\{h_{d_h}^a \geq 0\}}(h_{d_h}^a) + \alpha \mathbb{1}_{\{h_{d_h}^a<0\}}(h_{d_h}^a)) \end{pmatrix}$$

18. Write down the gradient of the last question in matrix form and define the dimensions of all matrix or vectors involved.

$$
\begin{aligned}
\frac{\partial L}{\partial h^a} &= \frac{\partial L}{\partial h^s} \odot \begin{pmatrix} \mathbb{1}_{\{h_1^a \geq 0\}}(h_1^a) + \alpha \mathbb{1}_{\{h_1^a < 0\}}(h_1^a) \\ \cdots \\ \mathbb{1}_{\{h_{d_h}^a \geq 0\}}(h_{d_h}^a) + \alpha \mathbb{1}_{\{h_{d_h}^a < 0\}}(h_{d_h}^a) \end{pmatrix} \\
&= \frac{\partial L}{\partial h^s} \odot (\mathbb{1}_{\{h^a \geq 0\}}(h^a) + \alpha \mathbb{1}_{\{h^a < 0\}}(h^a))
\end{aligned}
$$

where $\frac{\partial L}{\partial h^a}$, $\frac{\partial L}{\partial h^s}$ and the indicator vectors are d_h x 1

19. What is the gradient with respect to the parameters $\mathbf{W}^{(1)}$ and $\mathbf{b}^{(1)}$ of the hidden layer?

$$
\begin{aligned}
\frac{\partial L}{\partial W_{jl}^{(1)}} &= \frac{\partial L}{\partial h_j^a} \frac{\partial h_j^a}{\partial W_{jl}^{(1)}} \\
&= (\sum_{k=1}^{m} o_k^s W_{kj}^{(2)} - W_{yj}^{(2)})(\mathbb{1}_{\{h_j^a \geq 0\}}(h_j^a) + \alpha \mathbb{1}_{\{h_j^a < 0\}}(h_j^a)) \times \frac{\partial (b_j^{(1)} + \sum_{i=1}^{d} W_{ji}^{(1)} x_i)}{\partial W_{jl}^{(1)}} \\
&= (\sum_{k=1}^{m} o_k^s W_{kj}^{(2)} - W_{yj}^{(2)})(\mathbb{1}_{\{h_j^a \geq 0\}}(h_j^a) + \alpha \mathbb{1}_{\{h_j^a < 0\}}(h_j^a)) \times x_l \\
&= \frac{\partial L}{\partial h_j^a} \times x_l
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial L}{\partial b_j^{(1)}} &= \frac{\partial L}{\partial h_j^a} \times \frac{\partial h_j^a}{\partial b_j^{(1)}} \\
&= \frac{\partial L}{\partial h_j^a} \times 1 \\
&= \frac{\partial L}{\partial h_j^a}
\end{aligned}
$$

therefore,

$$
\frac{\partial L}{\partial W^{(1)}} = \begin{pmatrix} \frac{\partial L}{\partial h_1^a} x_1 & \frac{\partial L}{\partial h_1^a} x_2 & \cdots & \frac{\partial L}{\partial h_1^a} x_d \\ \frac{\partial L}{\partial h_2^a} x_1 & \frac{\partial L}{\partial h_2^a} x_2 & \cdots & \frac{\partial L}{\partial h_2^a} x_d \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial L}{\partial h_{d_h}^a} x_1 & \frac{\partial L}{\partial h_{d_h}^a} x_2 & \cdots & \frac{\partial L}{\partial h_{d_h}^a} x_d \end{pmatrix}
$$

and

$$\frac{\partial L}{\partial b^{(1)}} = \frac{\partial L}{\partial h^a}$$

20. Write down the gradient of the last question in matrix form and define the dimensions of all matrix or vectors involved.

$$\frac{\partial L}{\partial b^{(1)}} = \frac{\partial L}{\partial h^a}$$

with dimensions d_h x 1

$$\frac{\partial L}{\partial W^{(1)}} = \frac{\partial L}{\partial h^a} \times x^T$$

with dimensions d_h x d since $\frac{\partial L}{\partial h^a}$ is d_h x 1 and x^T is 1 x d

21. What are the partial derivatives of the loss L with respect to $\mathbf{x}$?

$$\begin{aligned}
\frac{\partial L}{\partial x_l} &= \sum_{j=1}^{d_h} \frac{\partial L}{\partial h_j^a} \frac{\partial h_j^a}{\partial x_l} \\
&= \sum_{j=1}^{d_h} \frac{\partial L}{\partial h_j^a} \frac{\partial (b_j^{(1)} + \sum_{i=1}^{d} W_{ji}^{(1)} x_i)}{\partial x_l} \\
&= \sum_{j=1}^{d_h} \frac{\partial L}{\partial h_j^a} W_{jl}^{(1)}
\end{aligned}$$

Therefore,

$$\frac{\partial L}{\partial x} = \begin{pmatrix} \sum_{j=1}^{d_h} \frac{\partial L}{\partial h_j^a} \times W_{j1}^{(1)} \\ \vdots \\ \sum_{j=1}^{d_h} \frac{\partial L}{\partial h_j^a} \times W_{jd}^{(1)} \end{pmatrix}$$

22. Consider the regularized empirical risk : $\tilde{R} = \hat{R} + \mathcal{L}(\theta)$, where θ is the vector of all the parameters in the network and $\mathcal{L}(\theta)$ describes a scalar penalty as a function of the parameters θ. The penalty is given importance according to a prior preferences for the values of θ. The L_2 (quadratic) regularization that penalizes the square norm (norm L_2) of the weights (but not the biases) is more standard, is used in ridge regression and is sometimes called "weight-decay". Here we shall consider a double regularization L_2 and L_1 which is sometimes named "elastic net" and we will use different hyperparameters (positive scalars $\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22}$) to control the effect of the regularization at each layer

$$\begin{aligned}
\mathcal{L}(\theta) &= \mathcal{L}(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}) \\
&= \lambda_{11}\|\mathbf{W}^{(1)}\|_1 + \lambda_{12}\|\mathbf{W}^{(1)}\|_2^2 + \lambda_{21}\|\mathbf{W}^{(2)}\|_1 + \lambda_{22}\|\mathbf{W}^{(2)}\|_2^2 \\
&= \lambda_{11}\left(\sum_{i,j}|\mathbf{W}^{(1)}_{ij}|\right) + \lambda_{12}\left(\sum_{ij}(\mathbf{W}^{(1)}_{ij})^2\right) + \lambda_{21}\left(\sum_{i,j}|\mathbf{W}^{(2)}_{ij}|\right) \\
&\qquad + \lambda_{22}\left(\sum_{ij}(\mathbf{W}^{(2)}_{ij})^2\right)
\end{aligned}$$

We will in fact minimize the regularized risk $\tilde{R}$ instead of $\hat{R}$. How does this change the gradient with respect to the different parameters?

$b^{(1)}$ and $b^{(2)}$ are essentially identical, where,

$$\frac{\partial \mathcal{L}(\theta)}{\partial b^{(1)}} = \frac{\partial \mathcal{L}(\theta)}{\partial b^{(2)}} = 0$$

$$\begin{aligned}
\frac{\partial \mathcal{L}(\theta)}{\partial W^{(1)}} &= \lambda_{11}\begin{pmatrix} sign(W^{(1)}_{11}) & sign(W^{(1)}_{12}) & \dots & sign(W^{(1)}_{1d}) \\ sign(W^{(1)}_{21}) & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ sign(W^{(1)}_{d_h1}) & sign(W^{(1)}_{d_h2}) & \dots & sign(W^{(1)}_{d_hd}) \end{pmatrix} \\
&+ \lambda_{12}\begin{pmatrix} 2W^{(1)}_{11} & 2W^{(1)}_{12} & \dots & 2W^{(1)}_{1d} \\ \vdots & \dots & \dots & \vdots \\ 2W^{(1)}_{d_h1} & 2W^{(1)}_{d_h2} & \dots & 2W^{(1)}_{d_hd} \end{pmatrix} \\
&= \lambda_{11} sign(W^{(1)}) + 2\lambda_{12}W^{(1)}
\end{aligned}$$

therefore,

$$\frac{\partial \tilde{R}}{\partial W^{(1)}} = \frac{\partial \hat{R}}{\partial W^{(1)}} + \lambda_{11} sign(W^{(1)}) + 2\lambda_{12} W^{(1)}$$

and similarly

$$\frac{\partial \tilde{R}}{\partial W^{(2)}} = \frac{\partial \hat{R}}{\partial W^{(2)}} + \lambda_{21} sign(W^{(2)}) + 2\lambda_{22} W^{(2)}$$

2 Training on the CIFAR-10 dataset

Train a neural network with 2 hidden layers, of size 512 and 256 respectively on the CIFAR-10 dataset, for 50 epochs. Use a learning rate of 0.003, and a batch size of 100. Use the RELU activation function with random seed set to 0.

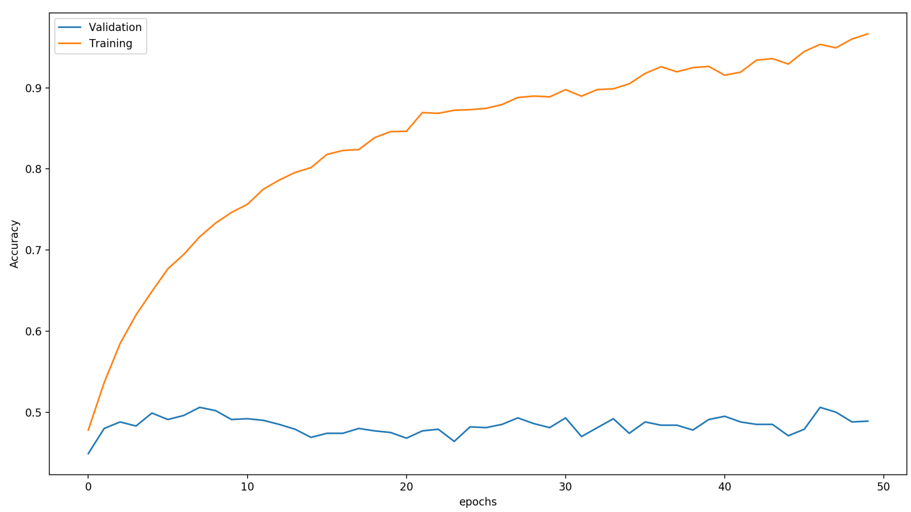

Figure 1: Evolution of both the training and validation accuracies during training

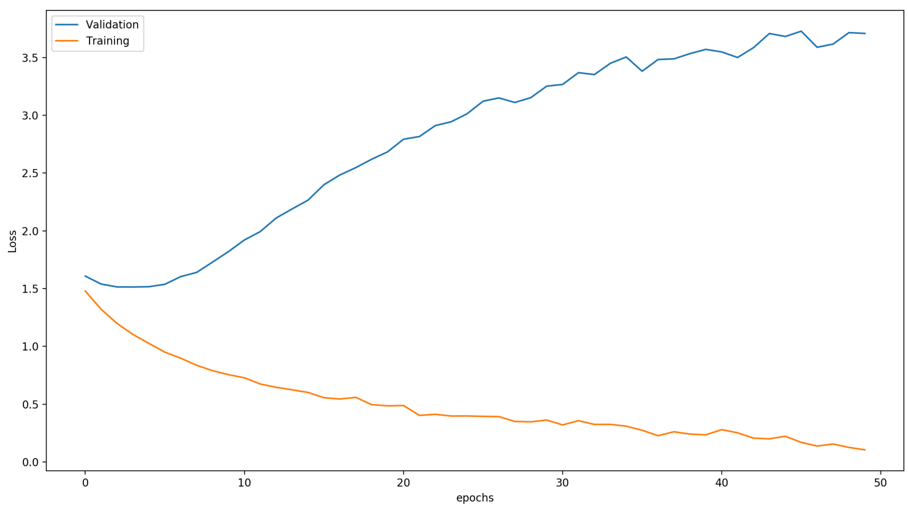

Figure 2: Evolution of both the training and validation losses during training