FTDL — Take Home Exam 1
Fundamentals and Tools of Deep Learning, 2022–2023
Solutions

Subham Shome
subham.shome@estudiante.uam.es
IPCV 2022-24
Escuela Politécnica Superior — Universidad Autónoma de Madrid

March 20, 2023

1. **Training a regression MLP with 20–dimensional patterns, two hidden layers with 100 units each over 1,000 epochs and a 1,000 pattern sample has taken 10 seconds. How many seconds should approximately take to train a regression MLP with five hidden layers with 200 units each, 500 epochs and a 10,000 pattern sample?**

Ans: First of all, it is noteworthy that the time taken to train a neural network will always depend on the hardware, software and data. Hence, it is assumed that the same hardware, software and data are used here to train the networks. Now, let's write down the given amounts for the first case.
Epochs = 1,000
Sample size = 1,000
Units = 100
Hidden layers = 2
Dimensions = 20
Hence, number of operations (floating-point) needed to perform are the following (considering 1 as the bias):

- Input layer to first hidden layer: (20 + 1) x 100 = 2,100 units
- First hidden layer to second hidden layer: (100 + 1) x 100 = 10,100 units
- Second hidden layer to output layer: (100 + 1) x 1 = 101 units

So, total units = 2,100 + 10,100 + 101 = 12,301 units. Finally, total units of floating-point operations calculated = 12,301 × 1,000 × 1,000 = 12,301,000,000 units.

Similarly, calculating the given amounts for the second case are as follows:

Epochs = 500
Sample size = 10,000
Units = 200
Hidden layers = 5
Dimensions = 20
Now, considering 1 as the bias, following are the calculation of floating-point operations needed for this case:

- Input layer to first hidden layer: (20 + 1) x 200 = 4,200 units
- First hidden layer to second hidden layer: (200 + 1) x 200 = 40,200 units
- Second hidden layer to third hidden layer: (200 + 1) x 200 = 40,200 units
- Third hidden layer to fourth hidden layer: (200 + 1) x 200 = 40,200 units
- Fourth hidden layer to fifth hidden layer: (200 + 1) x 200 = 40,200 units
- Fifth hidden layer to output layer: (200 + 1) x 1 = 201 units

So, total units = 4,200 + (4 × 40,200) + 101 = 165,201 units. Finally, total units of floating-point operations calculated = 165,201 × 500 × 10,000 = 826,005,000,000 units.

Last step is to calculate the time in seconds. The first operation working on 12,301,000,000 units takes 10 seconds. So, the time taken (in seconds) for the second operation working on 826,005,000,000 units

$$= \frac{10 \times 826,005,000,000}{12,301,000,000}$$

≈ 671.49 seconds ≈ **11 minutes and 11 seconds**.

Hence, the total time to train a regression MLP with five hidden layers with 200 units each, 500 epochs and a 10,000 pattern sample should be nearly 11 minutes and 11 seconds, considering the bias to be 1 and working on the same hardware, software and data.

2. **We have seen in the slides how to compute the partials with respect to the network weights of the mean squared error of a single layer regression MLP. Derive in a similar way the partials with respect to the network biases.**

Ans: The mean squared error of a single regression layer regression MLP is given by:

$$E(w,b) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \left(\sum_{j=1}^{M} w_j z_j + b_j\right)\right)^2$$

where, N is the number of training samples, M is the number of features, E is the mean squared error, y_i is the ground truth, z_j is the weighted input to neuron w_j, and b_j is the bias of neuron j. Now, the partial derivatives of the mean squared error with respect to the biases of a single layer regression MLP can be computed in the following way:

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial y_i} \cdot \frac{\partial y_i}{\partial z_j} \cdot \frac{\partial z_j}{\partial b_j}$$

We can compute each term in this expression as follows:

$$\frac{\partial E}{\partial y_i} = N \left(y_i - \left(\sum_{j=1}^{M} w_j z_j + b_j \right) \right)$$

$$\frac{\partial y_i}{\partial z_j} = f'(z_j)$$

where f' is the derivative of the activation function

$$\frac{\partial z_j}{\partial b_j} = 1$$

Substituting these expressions into the first equation, we get:

$$\frac{\partial E}{\partial b_j} = N \cdot \left(y_i - \left(\sum_{j=1}^{M} w_j z_j + b_j \right) \right) \cdot f'(z_j) \cdot 1$$

$$\therefore \frac{\partial E}{\partial b_j} = N \cdot \left(y_i - \left(\sum_{j=1}^{M} w_{ij} z_j + b_j \right) \right) \cdot f'(z_j)$$

Hence, this is the partial derivative of the mean squared error with respect to the network biases.

3. **The sigmoid and `tanh` activations. Two often used NN activations are the sigmoid**

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

and the hyperbolic tangent

$$\tau(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

(a) **Setting $o = \sigma(x)$, write the derivative $\sigma'(x)$ in terms of o.**

(b) **Write τ in terms of σ.**

(c) **Write also σ in terms of τ. *Hint: either use the previous items or look at the graphs of both functions and think about how to go from one to the other.***

Ans:

(a) Given the sigmoid function:

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

Let $o = \sigma(x)$, then we want to find $\sigma'(x)$ in terms of o. Using the chain rule:

$$\sigma'(x) = \frac{d\sigma(x)}{dx} = \frac{d\sigma(x)}{do} \cdot \frac{do}{dx}$$

To compute $\frac{d\sigma(x)}{do}$, we can write $-x$ in terms of o:

$$\begin{aligned} o = \sigma(x) &= \frac{1}{1+e^{-x}} \\ \frac{1}{o} &= 1 + e^{-x} \\ \frac{1}{o} - 1 &= e^{-x} \\ \therefore -x &= \ln\left(\frac{1}{o} - 1\right) \end{aligned}$$

Then:

$$\begin{aligned} \frac{d\sigma(x)}{do} &= \frac{d}{do}\left(\frac{1}{1+e^{-x}}\right) \\ &= \frac{d}{do}\left(\frac{1}{1+e^{\ln\left(\frac{1}{o}-1\right)}}\right) \\ &= \frac{d}{do}\left(\frac{1}{1+\frac{1}{o}-1}\right) \\ \therefore \frac{d\sigma(x)}{do} &= \frac{d}{do}(o) = 1 \end{aligned}$$

Substituting back:

$$\begin{aligned} \sigma'(x) &= \frac{d\sigma(x)}{do} \cdot \frac{do}{dx} \\ &= 1 \cdot \frac{d}{dx}\left(\sigma(x)\right) \\ &= \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) \\ &= \frac{e^{-x}}{\left(1+e^{-x}\right)^2} \end{aligned}$$

$$= \frac{e^{-x}}{(1+e^{-x})} \cdot \frac{1}{1+e^{-x}}$$

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

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

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

$$\therefore \sigma'(x) = o^2 \cdot e^{-x}$$

which is the derivative of the sigmoid function $\sigma(x)$ with respect to x in terms of o, where $o = \sigma(x)$.
It is worthy to note that, in simple terms, the derivative of the sigmoid function $\sigma'(x)$ is simply given by

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

. Rewriting this in terms of $o = \sigma(x)$, we can also write that

$$\sigma'(x) = o(1 - o)$$

(b) We have,

$$\tau(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

and,

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

Now, we can write τ in the following way:

$$\tau = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$= \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

$$= \frac{2 - (1 + e^{-2x})}{1 + e^{-2x}}$$

$$= \frac{2}{1 + e^{-2x}} - 1$$

$$\therefore \tau(x) = 2\sigma(2x) - 1$$

This is the representation of the function σ in terms of τ.

(c) To write $\sigma(x)$ in terms of $\tau(x)$, we can begin by noticing that:

$$e^x = e^{\frac{x}{2}} \cdot e^{\frac{x}{2}}$$

$$e^{-x} = e^{-\frac{x}{2}} \cdot e^{-\frac{x}{2}}$$

Substituting these expressions into $\tau(x)$, we get:

$$\tau(x) = \frac{e^{\frac{x}{2}} \cdot e^{\frac{x}{2}} - e^{-\frac{x}{2}} \cdot e^{-\frac{x}{2}}}{e^{\frac{x}{2}} \cdot e^{\frac{x}{2}} + e^{-\frac{x}{2}} \cdot e^{-\frac{x}{2}}}$$

Simplifying the numerator and denominator of the fraction, we obtain:

$$\tau(x) = \frac{(e^{\frac{x}{2}})^2 - (e^{-\frac{x}{2}})^2}{(e^{\frac{x}{2}})^2 + (e^{-\frac{x}{2}})^2} = \frac{(e^{\frac{x}{2}} - e^{-\frac{x}{2}})(e^{\frac{x}{2}} + e^{-\frac{x}{2}})}{(e^{\frac{x}{2}})^2 + (e^{-\frac{x}{2}})^2} = \frac{\sinh(\frac{x}{2})}{\cosh(\frac{x}{2})}$$

where sinh and cosh are the hyperbolic sine and cosine functions, respectively. Now, using the fact that $\sinh(x) = 2\sigma(x)\cosh(x) - 1$, we can solve for $\sigma(x)$:

$$\sigma(x) = \frac{\sinh(\frac{x}{2}) + 1}{2\cosh(\frac{x}{2})} = \frac{\frac{\sinh(x)}{2} + \frac{\cosh(x)}{2}}{\cosh(\frac{x}{2})} = \frac{\frac{1}{2}(\tau(x) + 1) + \frac{1}{2}}{\sqrt{\frac{1}{2}(\tau(x) + 1)}} = \sqrt{\frac{\tau(x) + 1}{2}}$$

Therefore, σ in terms of τ can be written as $\sigma(x) = \sqrt{\frac{\tau(x)+1}{2}}$.

4. **i. We have seen that if for a sample $(x^p, y^p); 1 \leq p \leq N$, we organize a $N \times d$ data matrix X whose p–th row is $(x^p)^t$ and a targets' vector $T = (y1, ..., yN)^t$, the vector. $w^* = (X^t X)^{-1} X^t T$ solves the homogeneous linear regression problem. Assuming an SVD decomposition for X of the form $X = U\Sigma V^t$, express w^* in terms of U, Σ and V.**

ii. Assume now that we add a Ridge penalty $\frac{\lambda}{2}||w||^2$ to the regression problem. Express now w^* in terms of U, Σ, V and $\frac{\lambda}{2}$ and discuss its comparison with the previous, Ridge–free expression for w^*.

Ans:

i. We have,

$$w^* = (X^t X)^{-1} X^t T$$

and we are supposed to use Singular Value Decomposition (SVD). Here, we substitute $X = U\Sigma V^t$ into the equation of w^*.

$$w^* = ((U\Sigma V^t)^t U\Sigma V^t)^{-1} (U\Sigma V^t)^t T$$

$$\therefore w^* = (V\Sigma^t U^t U\Sigma V^t)^{-1} V\Sigma^t U^t T$$

We also know that $U^tU = I$ and $V^tV = I$ since they are left and right singular vectors respectively. Hence,

$$w^* = (\Sigma^t\Sigma)^{-1}(U\Sigma V^t)^t T$$

$$\Rightarrow w^* = (\Sigma^t\Sigma)^{-1}(V\Sigma^t U^t T)$$

Now, since Σ is a diagonal matrix, $\Sigma^t = \Sigma$.

$$\therefore w^* = V\Sigma^{-2}\Sigma^t U^t T$$

ii. Now, we have,

$$w^* = (X^tX + \frac{\lambda}{2}I)^{-1}X^tT$$

We do the similar expressions using $X = U\Sigma V^t$ here.

$$w^* = ((U\Sigma V^t)U\Sigma V^t + \frac{\lambda}{2}I)^{-1}(U\Sigma V^t)^t T$$

$$= (V\Sigma\Sigma^t V^t + \frac{\lambda}{2}I)^{-1}(U\Sigma V^t)^t T$$

Since V and $\frac{\lambda}{2}I$ are orthogonal matrices, the inverse can be applied to the diagonal matrix Σ only. So,

$$w^* = (V(\Sigma\Sigma^t)^{-1}V^t + \frac{2}{\lambda}I)(U\Sigma V^t)^t T$$

Here, $\Sigma = \Sigma^t$, since Σ is a diagonal matrix. Hence,

$$w^* = V\Sigma^{-2}V^tV\Sigma^tU^tT + \frac{2}{\lambda}IV\Sigma^tU^tT$$

Finally, we can write $V^tV = I$.

$$\therefore w^* = V\Sigma^{-2}\Sigma^tU^tT + \frac{2}{\lambda}V\Sigma^tU^tT$$

This is the expression for w^* considering the ridge penalty

5. **Assume that we are fitting a constant linear regression model (i.e., $f(x) = c$) using the mean squared error, MSE, as the loss. Derive with sufficient detail what would be the value of c.**

Assume next that we are fitting again a constant linear regression model (i.e., $f(x) = c$) but now using the mean, absolute error, MAE as the loss. Derive now and with sufficient detail what would be the value of c.

Ans: For the first part, we have $f(x) = c$. For this linear regression model, the mean squared error (MSE) is given by

$$E = \frac{1}{N}\sum_{i=1}^{N}(y_i - c)^2$$

To minimize the MSE, we put $\frac{\partial E}{\partial c} = 0$. Now,

$$\frac{\partial E}{\partial c} = \frac{\partial}{\partial c}\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - c)^2\right)$$

$$= \frac{1}{N}\sum_{i=1}^{N} 2(y_i - c)\cdot(-1)$$

$$\therefore \frac{\partial E}{\partial c} = -\frac{1}{N}\sum_{i=1}^{N} 2(y_i - c)$$

Now, putting this equal to zero, we get

$$\frac{\partial E}{\partial c} = -\frac{1}{N}\sum_{i=1}^{N} 2(y_i - c) = 0$$

$$-\frac{2}{N}\sum_{i=1}^{N} y_i + \frac{2}{N}\sum_{i=1}^{N} c = 0$$

$$\sum_{i=1}^{N} c = \sum_{i=1}^{N} y_i$$

$$Nc = \sum_{i=1}^{N} y_i$$

$$\therefore c = \frac{1}{N}\sum_{i=1}^{N} y_i$$

In other words, the value of c that minimizes the MSE is simply the mean of the actual values.

The mean absolute error (MAE) loss function for a constant linear regression model can be defined as follows:

$$L(c) = \frac{1}{n}\sum_{i=1}^{n}|y_i - c|$$

where y_i is the observed value of the response variable for the i-th data point, n is the number of data points, and c is the constant value that we are trying to estimate.

To find the value of c that minimizes the MAE loss, we need to find the value of c that makes the derivative of the loss function equal to zero. However, the absolute value function is not differentiable at zero, so we cannot take the derivative directly. Instead, we can use a subdifferential approach.

The subdifferential of the absolute value function at a point x is defined as:

$$\partial|x| = \begin{cases} -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

Using this definition, we can write the subdifferential of the MAE loss function at c as:

$$\partial L(c) = \frac{1}{n} \sum_{i=1}^{n} \partial|y_i - c|$$

Now, we need to find the value of c that makes the subdifferential of the loss function contain zero. That is, we need to find c such that:

$$0 \in \partial L(c)$$

Using the definition of the subdifferential, this is equivalent to:

$$0 \in \frac{1}{n} \sum_{i=1}^{n} \partial|y_i - c|$$

Let S be the set of indices i such that $y_i - c > 0$, and let T be the set of indices i such that $y_i - c < 0$. Then, we can write the above equation as:

$$\frac{1}{n} \sum_{i \in S} 1 - \frac{1}{n} \sum_{i \in T} 1 \in [-1, 1]$$

Simplifying this expression, we get:

$$\frac{2}{n} \sum_{i \in S} 1 - 1 \in [-1, 1]$$

which is equivalent to:

$$\frac{2}{n} \sum_{i \in S} 1 \in [0, 2]$$

Since $0 \leq \frac{2}{n} \sum_{i \in S} 1 \leq 2$, we conclude that the solution to the minimization problem is given by the median of the data. Therefore, the value of c that minimizes the MAE loss is:

$$c = \text{median}(y_1, y_2, \ldots, y_n)$$

This result holds for any distribution of the data, not just for symmetric distributions. The median is a robust estimator of the central tendency of the data, which means that it is not affected by outliers or extreme values in the data.