Deep Learning 1

Despoina Touska

December 2, 2023

1 Linear Module

1.1 a

$$\frac{\partial L}{\partial \boldsymbol{W}} => \left(\frac{\partial L}{\partial \boldsymbol{W}}\right)_{m,n} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial Y_{i,j}}{\partial W_{m,n}}$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial Y_{i,j}}{\partial W_{m,n}} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial (\sum_l^M X_{i,l} W_{l,j}^T + B_{i,j})}{\partial W_{m,n}}$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial (\sum_l^M X_{i,l} W_{j,l} + B_{i,j})}{\partial W_{m,n}} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \left(\sum_l^M X_{i,l} \frac{\partial W_{j,l}}{\partial W_{m,n}} + \frac{\partial B_{i,j}}{\partial W_{m,n}}\right)$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \left(\sum_l^M X_{i,l} \frac{\partial W_{j,l}}{\partial W_{m,n}} + 0\right) = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \sum_l^M X_{i,l} \delta_{j,m} \delta_{l,n}$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} X_{i,n} \delta_{j,m} = \sum_i^S \frac{\partial L}{\partial Y_{i,m}} X_{i,n} = \sum_i^S \left(\frac{\partial L}{\partial Y_{m,i}}\right)^T X_{i,n}$$

So,

$$\frac{\partial L}{\partial \boldsymbol{W}} = \left(\frac{\partial L}{\partial \boldsymbol{Y}}\right)^T \boldsymbol{X}$$

1.2 c

$$\frac{\partial L}{\partial \boldsymbol{X}} => \left(\frac{\partial L}{\partial \boldsymbol{X}}\right)_{m,n} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial Y_{i,j}}{\partial X_{m,n}}$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial Y_{i,j}}{\partial X_{m,n}} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial (\sum_l^M X_{i,l} W_{l,j}^T + B_{i,j})}{\partial X_{m,n}}$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial (\sum_l^M X_{i,l} W_{j,l} + B_{i,j})}{\partial X_{m,n}} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \left(\sum_l^M \frac{\partial X_{i,l}}{\partial X_{m,n}} W_{j,l} + \frac{\partial B_{i,j}}{\partial X_{m,n}}\right)$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \left(\sum_l^M \frac{\partial X_{i,l}}{\partial X_{m,n}} W_{j,l} + 0\right) = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \sum_l^M \delta_{i,m} \delta_{l,n} W_{j,l}$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \delta_{i,m} W_{j,n} = \sum_j^N \frac{\partial L}{\partial Y_{m,j}} W_{j,n}$$

So,

$$\frac{\partial L}{\partial \boldsymbol{X}} = \frac{\partial L}{\partial \boldsymbol{Y}} \boldsymbol{W}$$

1.3 b

$$\frac{\partial L}{\partial \boldsymbol{b}} => \left(\frac{\partial L}{\partial \boldsymbol{b}}\right)_m = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial Y_{i,j}}{\partial b_m}$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial Y_{i,j}}{\partial b_m} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial (\sum_l^M X_{i,l} W_{l,j}^T + B_{i,j})}{\partial b_m}$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial (\sum_l^M X_{i,l} W_{j,l} + B_{i,j})}{\partial b_m} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \left(\sum_l^M \frac{\partial (X_{i,l} W_{j,l})}{\partial b_m} + \frac{\partial B_{i,j}}{\partial b_m}\right)$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \left(0 + \frac{\partial B_{i,j}}{\partial b_m}\right) = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial B_{i,j}}{\partial b_m} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \delta_{j,m} = \sum_i^S \frac{\partial L}{\partial Y_{i,m}}$$

So,

$$\frac{\partial L}{\partial \boldsymbol{b}} = \mathbf{1}^T \frac{\partial L}{\partial \boldsymbol{Y}}$$

$$\frac{\partial L}{\partial \boldsymbol{b}} = [\frac{\partial L}{\partial b_1}, ..., \frac{\partial L}{\partial b_N}] = [\sum_i^S \frac{\partial L}{\partial Y_{i,1}}, ..., \sum_i^S \frac{\partial L}{\partial Y_{i,N}}]$$

1.4 d

$$\frac{\partial L}{\partial \boldsymbol{X}} => \left(\frac{\partial L}{\partial \boldsymbol{X}}\right)_{m,n} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial Y_{i,j}}{\partial X_{m,n}}$$

$$= \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial Y_{i,j}}{\partial X_{m,n}} = \sum_{i,j}^{S,N} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial h(X_{i,j})}{\partial X_{m,n}}$$

$$= \begin{cases} \frac{\partial L}{\partial Y_{i,j}} \frac{\partial h(X_{i,j})}{\partial X_{i,j}} & i = m, j = n \\ 0 & i \neq m, j \neq n \end{cases}$$

So,

$$\frac{\partial L}{\partial \boldsymbol{X}} = \frac{\partial L}{\partial \boldsymbol{Y}} \circ \frac{\partial h(\boldsymbol{X})}{\partial \boldsymbol{X}}$$

The $\circ$ denotes the Hadamard product.

2 Optimization

2.1 a

If the Hessian matrix $\boldsymbol{H}$ is positive definite, we have a strictly local minimum. Additionally, in a positive definite matrix $\boldsymbol{H}$ it is true that: $\boldsymbol{x^T H x} > 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$. If $\boldsymbol{x_n}$ is the n-th eigenvector of $\boldsymbol{H}$ then it is true that:

$$.\boldsymbol{x_n^T H x_n} > 0 => \boldsymbol{x_n^T} \lambda_n \boldsymbol{x_n} > 0 => \lambda_n \boldsymbol{x_n^T x_n} > 0 => \lambda_n > 0$$

From the above, $\boldsymbol{x_n^T x_n} = 1$ so this implies that the λ_n eigenvalue is positive as well. I also make use of the formula: $\boldsymbol{H x_n} = \lambda_n \boldsymbol{x_n}$, when $\boldsymbol{x_n}$ is an eigenvector of $\boldsymbol{H}$ and λ_n an eigenvalue of $\boldsymbol{H}$.

2.2 b

The diagonal matrix Λ includes all the eigenvalues of the Hessian matrix.

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & ... & \\ & & \lambda_n \end{bmatrix}$$

It is reasonable to presume that the values of $\lambda_1, ..., \lambda_n$ are not biased towards negative or positive values. For this reason, given any critical point, the probability of every value λ_i to be positive can be assumed to be 1/2. Additionally, due to the significant non-linearity of the Hessian matrix, it's reasonable to consider the values of λ_i to be independent of one another. As a result, we'll treat the probabilities of them as independent events.

So, given a critical point, the probability of it being a minimum (positive eigenvalues) is:

$$P(\lambda_1 > 0, ..., \lambda_n > 0) = P(\lambda_1 > 0)...P(\lambda_n > 0) = \frac{1}{2}...\frac{1}{2} = \frac{1}{2^n}$$

The probability of any critical point being a minimum decreases exponentially for higher dimensions, implying that given any critical point, it is very unlikely that it is a minimum. Similarly, we can conclude that every critical point has probability of $\frac{1}{2^n}$ to be a maximum.

Finally, the probability of a critical point being a saddle point is:

$$P(saddle) = 1 - P(maximum) - P(minimum) = 1 - \frac{1}{2^n} - \frac{1}{2^n} = 1 - \frac{1}{2^{n-1}}$$

Which in the case where n goes to infinity is very close to 1. That is why the number of saddle points is exponentially larger than the number of local minima for higher dimensions.

2.3 c

Saddle points are harmful to training and we can prove that using the Ill-conditioning. Given the update formula of gradient descent around a saddle point for the current weight $\boldsymbol{w'}$: $\boldsymbol{w} \leftarrow \boldsymbol{w'} - \epsilon \boldsymbol{g}$, where ϵ is the step size and $\boldsymbol{g} = \frac{dL}{d\boldsymbol{w}}$, and the 2nd order Taylor expansion around this weight: $L(\boldsymbol{w}) = L(\boldsymbol{w'}) + g(\boldsymbol{w} - \boldsymbol{w'}) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w'})^T \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w'})$, we can analyze the loss around the current weight $\boldsymbol{w'}$ plus a small step: $L(\boldsymbol{w' - \epsilon g}) \approx L(\boldsymbol{w'}) - \epsilon \boldsymbol{g}^T \boldsymbol{g} + \frac{1}{2}\epsilon^2 \boldsymbol{g}^T H \boldsymbol{g}$. From this expression, we can argue that $\frac{1}{2}\epsilon^2 \boldsymbol{g}^T H \boldsymbol{g} > \epsilon \boldsymbol{g}^T \boldsymbol{g}$, which means that after we take the gradient step the loss will go higher, instead of lower, which implies that network training will cease.

3 F1 score

3.1 a

Accuracy is the most commonly used evaluation metric. It tells us how many times a model got its prediction correct as a ratio of the total times the model was used for predictions. However, its suitability relies heavily on the dataset's balance, where each class possesses an equal number of samples. In scenarios where the dataset is imbalanced, meaning unequal class distribution, accuracy can be misleading as the class with the most records exerts disproportionate influence, creating bias. This bias, especially when one class significantly outnumbers others, skews the accuracy measurement, compromising its reliability. So, we use accuracy only if you have balanced datasets and you give the same importance to 0s and 1s. Two examples that accuracy can provide helpful information is in cat-dog image classification or face detection where the datasets are reasonably balanced.

On the other hand, both precision and recall are useful metrics in cases where the dataset is imbalanced. However, depending on the use case, we would like to optimize one or the other metric.

Precision shows how often a model is correct when predicting the target class. It is more useful when we want to affirm the correctness of our model. For example, in the case of YouTube recommendations, reducing the number of false positives is of utmost importance. False positives here represent videos that the user does not like, but YouTube is still recommending them. False negatives are of lesser importance here since the YouTube recommendations should only contain videos that the user is more likely to click on. Another example, is email spam detection. In this case, missing out to detect a spam email is okay (low recall), but no legit or important email must go into the spam folder (false positive).

Recall shows whether a model can find all objects of the target class. An example that we want high recall is in medical test such as cancer detection. It is okay to classify a healthy person as having cancer (false positive) and following up with more medical tests, but it is definitely not okay to miss identifying a cancer patient or classifying a cancer patient as healthy (false negative). Another example is in flagging fraudulent transactions. In this case, it is okay to classify a legit transaction as fraudulent — it can always be reverified by passing through additional checks. But it is definitely not okay to classify a fraudulent transaction as legit (false negative).