We decided this derivation is too much work for a small part of an assignment. We provide the original question and the solution in this document.

This is a very good exercise to check your understanding of derivations and Linear Algebra, so we encourage you to try it yourself.

## Calculate the backward pass for the cross-entropy module:

This module calculates the softmax on the input and then the loss, so you will need to calculate the gradient based on the combined expression.

Given a batch with $N$ datapoints and $K$ classes, we input the predictions before the softmax (pre-softmax probabilities or "logits") $X \in \mathbb{R}^{N \times K}$ and the one-hot encoded ground-truth labels: $Y \in \{0,1\}^{N \times K}$

Our predictions $\hat{Y} \in \{0,1\}^{N \times K}$ are given by the softmax of the prediction logits per datapoint $X_n$:

$$\hat{Y}_{n,i} = s(X_n)_i$$

where the softmax for $K$ classes, given the prediction logits for one datapoint $\mathbf{x} \in \mathbb{R}^K$, is defined as follows:

$$s(\mathbf{x})_i = \exp(\mathbf{x}_i) / \sum_{k=1}^{K} \exp(\mathbf{x}_k)$$

The *Cross-Entropy Loss* for $K$ classes and $N$ datapoints is defined as

$$L(Y, \hat{Y}) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} Y_{n,k} \log(\hat{Y}_{n,k})$$

The loss for a single datapoint is:

$$L(Y_n, \hat{Y}_n) = -\frac{1}{N} \sum_{k=1}^{K} Y_{n,k} \log(\hat{Y}_{n,k})$$

**Hint:** The final gradient $\dfrac{\partial L(Y, \hat{Y})}{\partial X}$ that you will need in your backpropagation is of shape $\mathbb{R}^{N \times K}$, where each row is given by the gradient of a single datapoint's loss.

**Hint:** Calculate the softmax gradients $\partial s(\mathbf{x})_i / \partial \mathbf{x}_j$ separately. Use the quotient rule of derivatives, given $j \in \{1, \ldots, K\}$. Use a case differentation for $i = j$ and $i \neq j$. Simplify the expressions by using the definition of the softmax.

**Hint:** Calculate the gradient of the cross-entropy loss module first for a single datapoint: $\partial L(Y_n, \hat{Y}_n) / \partial X_n$. This gradient will be a vector of length $K$. Calculate the gradient for a single scalar element of that vector $\partial L(Y_n, \hat{Y}_n) / \partial X_{n,i}$ with $i \in \{1, \ldots, K\}$. Split the sum over classes in the gradient into two parts, such that you can use the case differentation of the softmax derivative that you calculated above. Then, further simplify this expressing using $\sum_{k=1}^{K} s(X_n)_k = 1$ and $\sum_{k=1}^{K} Y_{n,k} = 1$.

# Solution:

See the question for the mathematical definitions.

This module calculates softmax and cross-entropy loss at the same time, so we will calculate the chained gradient of both.

We calculate softmax derivative using the quotient rule of derivatives with a case differentation:

Let $i = j$:

$$\begin{aligned}\frac{\partial s(\mathbf{x})_i}{\partial \mathbf{x}_i} &= \frac{\exp(\mathbf{x}_i)\sum_{k=1}^{N}\exp(\mathbf{x}_k) - \exp(\mathbf{x}_i)^2}{(\sum_{k=1}^{N}\exp(\mathbf{x}_k))^2} \\ &= \frac{\exp(\mathbf{x}_i)}{\sum_{k=1}^{N}\exp(\mathbf{x}_k)}\frac{\sum_{k=1}^{N}\exp(\mathbf{x}_k) - \exp(\mathbf{x}_i)}{\sum_{k=1}^{N}\exp(\mathbf{x}_k)} \\ &= s(\mathbf{x})_i(1 - s(\mathbf{x})_i)\end{aligned}$$

Let $i \neq j$:

$$\begin{aligned}\frac{\partial s(\mathbf{x})_i}{\partial \mathbf{x}_j} &= \frac{0 - \exp(\mathbf{x}_i)\exp(\mathbf{x}_j)}{(\sum_{k=1}^{N}\exp(\mathbf{x}_k))^2} \\ &= -s(\mathbf{x})_i s(\mathbf{x})_j\end{aligned}$$

Result:

$$\frac{\partial s(\mathbf{x})_i}{\partial \mathbf{x}_j} = \begin{cases} s(\mathbf{x})_i(1 - s(\mathbf{x})_i) & i = j \\ -s(\mathbf{x})_i s(\mathbf{x})_j & i \neq j \end{cases}$$

Cross-entropy loss for a single datapoint can be rewritten:

$$\begin{aligned}L(Y_n, \hat{Y}_n) &= -\frac{1}{N}\sum_{k=1}^{K} Y_{n,k}\log(\hat{Y}_{n,k}) \\ \hat{Y}_{n,i} &= s(X_n)_i \\ L(Y_n, X_n) &= -\frac{1}{N}\sum_{k=1}^{K} Y_{n,k}\log(s(X_n)_k)\end{aligned}$$

Now, we calculate the derivative for a single datapoint $n \in \{1, \ldots, N\}$ and a single class $i \in \{1, \ldots, K\}$ using product and chain rule.

$$\frac{\partial L(Y_n, X_n)}{\partial X_{n,i}} = -\frac{1}{N}\sum_{k=1}^{K}\left(\frac{\partial Y_{n,k}}{\partial X_{n,i}}\log(s(X_n)_k) + Y_{n,k}\frac{\partial \log(s(X_n)_k)}{\partial X_{n,i}}\right) \quad \text{(Apply product rule)}$$

$$= -\frac{1}{N}\sum_{k=1}^{K} Y_{n,k}\frac{\partial \log(s(X_n)_k)}{\partial X_{n,i}} \quad \left(\text{Use } \frac{\partial Y_{n,k}}{\partial X_{n,i}} = 0\right)$$

$$= -\frac{1}{N}\sum_{k=1}^{K} Y_{n,k}\frac{1}{s(X_n)_k}\frac{\partial (s(X_n)_k)}{\partial X_{n,i}} \quad \text{(Apply chain rule)}$$

$$= -\frac{1}{N}\left(Y_{n,i}\frac{1}{s(X_n)_i}\frac{\partial (s(X_n)_i)}{\partial X_{n,i}} + \sum_{k=1,k\neq i}^{K} Y_{n,k}\frac{1}{s(X_n)_k}\frac{\partial (s(X_n)_k)}{\partial X_{n,i}}\right) \quad \text{(Expand sum)}$$

$$= -\frac{1}{N}\left(Y_{n,i}\frac{1}{s(X_n)_i}s(X_n)_i(1 - s(X_n)_i) + \sum_{k=1,k\neq i}^{K} Y_{n,k}\frac{1}{s(X_n)_k}(-s(X_n)_i s(X_n)_k)\right) \quad \text{(Apply softmax derivative)}$$

$$= -\frac{1}{N}\left(Y_{n,i}(1 - s(X_n)_i) + \sum_{k=1,k\neq i}^{K} Y_{n,k}(-s(X_n)_i)\right) \quad \text{(Reduce fractions)}$$

$$= -\frac{1}{N}\left(Y_{n,i} - Y_{n,i}s(X_n)_i - \sum_{k=1,k\neq i}^{K} Y_{n,k}s(X_n)_i\right) \quad \text{(Expand products)}$$

$$= -\frac{1}{N}\left(Y_{n,i} - s(X_n)_i(Y_{n,i} + \sum_{k=1,k\neq i}^{K} Y_{n,k})\right) \quad \text{(Factorize)}$$

$$= -\frac{1}{N}\left(Y_{n,i} - s(X_n)_i\sum_{k=1}^{K} Y_{n,k}\right) \quad \text{(Group sum)}$$

$$= -\frac{1}{N}(Y_{n,i} - s(X_n)_i) \quad \text{(Sum of one-hot row is 1)}$$

$$= \frac{1}{N}(s(X_n)_i - Y_{n,i}) \quad \text{(Done.)}$$

The gradient of the combined softmax and cross-entropy loss is a matrix in $\mathbb{R}^{N\times K}$ with the elements

$$\left(\frac{\partial L(Y, \hat{Y})}{\partial X}\right)_{n,k} = \frac{1}{N}(s(X_n)_k - Y_{n,k})$$

or simply

$$\frac{\partial L(Y, \hat{Y})}{\partial X} = \frac{1}{N}(\hat{Y} - Y)$$

□