Problem 4

a.) Sentence Vector and Regularization

The softmax classification function is given in notes2.pdf as

$$p(y_j = 1 \mid x) = \frac{\exp(W_j x)}{\sum_c \exp(W_c x)}.$$

I'm assuming there is only the single linear layer to worry about, given by W. Let's say $x \in \mathbb{R}^d$ and that $y \in \mathbb{R}^C$. Then $W \in \mathbb{R}^{C \times d}$. Using this in a cross-entropy loss function

$$CE = -\sum_j^C y_j \log\left(\frac{\exp(W_j x)}{\sum_c \exp(W_c x)}\right)$$

and then summing over all of the possible x vectors in the training set

$$\begin{aligned} CE &= -\sum_i^N \sum_j^C y_j \log\left(\frac{\exp(W_j x^{(i)})}{\sum_c \exp(W_c x^{(i)})}\right) \\ &= -\sum_i^N \sum_j^C y_j \log(\hat{y}_j^{(i)}). \end{aligned}$$

The notes make the one-hot vector simplification instead of writing it as I have above. I prefer to leave the application of such assumptions to the very end because it generally results in simpler or more general expressions. Now to write this in Einstein. Note that $W_j x^{(i)} = W_{jk} X_{ki}$, where the matrix $X \in \mathbb{R}^{d \times N}$ contains the entire dataset with each x vector being a column in X. This all becomes

$$CE = -y_j \log\left(\frac{\exp(W_{jk} X_{ki})}{\exp(W_{cp} X_{pi})}\right).$$

Then we have to take derivatives wrt each element in W.

$$
\begin{aligned}
\frac{\partial}{\partial W_{ab}} CE &= -y_j \frac{\partial}{\partial W_{ab}} \log\left(\frac{\exp(W_{jk}X_{ki})}{\exp(W_{cp}X_{pi})}\right) \\
&= -y_j \frac{1}{\hat{y}_j^{(i)}} \left[\frac{\exp(W_{jk}X_{ki})}{\exp(W_{cp}X_{pi})} \frac{\partial}{\partial W_{ab}}(W_{jd}X_{di}) - \frac{\exp(W_{jk}X_{ki})}{(\exp(W_{cp}X_{pi}))^2} \frac{\partial}{\partial W_{ab}} \exp(W_{gh}X_{hi})\right] \\
&= -y_j \frac{1}{\hat{y}_j^{(i)}} \left[\hat{y}_j^{(i)} X_{di}\delta_{ja}\delta_{db} - \hat{y}_j^{(i)}\hat{y}_g^{(i)} X_{hi}\delta_{ga}\delta_{hb}\right] \\
&= -y_j \frac{1}{\hat{y}_j^{(i)}} \left[\hat{y}_j^{(i)} X_{di}\delta_{ja}\delta_{db} - \hat{y}_j^{(i)}\hat{y}_g^{(i)} X_{hi}\delta_{ga}\delta_{hb}\right] \\
&= -y_j \left[X_{di}\delta_{ja}\delta_{db} - \hat{y}_g^{(i)} X_{hi}\delta_{ga}\delta_{hb}\right] \\
&= -y_j \left[X_{bi}\delta_{ja} - \hat{y}_a^{(i)} X_{bi}\right] \\
&= -y_j X_{bi}\delta_{ja} + y_j \hat{y}_a^{(i)} X_{bi} \\
&= -y_a X_{bi} + y_j \hat{y}_a^{(i)} X_{bi} \\
&= -y_a X_{bi} + \hat{y}_a^{(i)} X_{bi} \\
&= \left[\hat{y}_a^{(i)} - y_a\right] X_{bi}
\end{aligned}
$$

To turn this back into a vector, remember that there was a sum over i.

$$
\frac{\partial}{\partial W} CE = \sum_i^N \left[\hat{y}^{(i)} - y\right] \otimes x^{(i)}
$$

The units work out, because this is a sum of $\mathbb{R}^{C\times d}$ matrices, the same dimension as W. Note, however, that the code uses a slightly different convention, with each x being a row vector, so the implementation will be the transpose of my result here.