Programming

0. They used bag-of-words rather than pre-trained embeddings to minimize bias. They removed excess white space and replaced urls and mentions with placeholders. They tokenized and used n-grams up to n=3, then transformed into TF-IDF.
1. My results were nearly the same as the paper.
2. I trained for 2 epochs with binary cross entropy loss, adam optimizer, and 0.001 learning rate. It's not a good idea to train and test on the same data because you will eventually get 100% accuracy since you're fitting your model to the same data.
3. I did not do this part.
4. My laptop couldn’t handle the word embeddings.

Questions

0. I spent about 4 to 5 hours on this problem set.

PS1

0) Backpropogation

The following network takes in three inputs

$x = 1$, $y = 2$, and $z = 1$

$$a = x^2 + y^2 + xz \qquad (1)$$

$$b = \max(yz, a) \qquad (2)$$

$$c = a - 2b \qquad (3)$$

a) $\frac{da}{dx} = 2x + z$

$\frac{da}{dy} = 2y$

$\frac{da}{dz} = x$

$\frac{db}{dy} = \max(z, 2y)$

$\frac{db}{dz} = \max(y, x)$

$\frac{db}{da} = \max(0, 1)$

$\frac{dc}{da} = 1$

$\frac{dc}{db} = -2$

b) $\frac{dc}{dx} = \frac{dc}{da} \frac{da}{dx} = (1)(2x+z) = 3$

$\frac{dc}{dy} = \frac{dc}{da} \frac{da}{dy} = (1)(2y) = 4$

$\frac{dc}{dz} = \frac{dc}{da} \frac{da}{dz} = (1)(x) = 1$

1) Softmax Gradient

In class, we derived the gradient of the loss for the sigmoid output activation function in a binary logistic regression classifier, with respect to weight w_j:

$$\frac{dL}{dw_j} = [\sigma(wx+b)-y]x_j$$

Now derive the local gradient for a softmax output layer, again assuming one hidden layer, where the loss is as follows:

$$L = -\ln P(y=k|x) = -\ln \frac{e^{w_k x + b_k}}{\sum_{j=1}^{k} e^{w_j x + b_j}}$$

$$L = -[\ln e^{w_k x + b_k} - \ln \sum_{j=1}^{k} e^{w_j x + b_j}]$$

$$= [\sum_{j=1}^{k} w_j x + b_j] - w_k x + b_k$$

$$= [\sum_{j=1}^{k} w_j x + b_j] - y$$

$$\frac{dL}{dw_j} = x - y$$