HW Assignment 2

Ido Beerie 315140830
Amit Omer 322532102
Shachar Gabbay 213144173

December 2024

1 Question 1

1.1

Starting from our fundamental cross-entropy definition:

$$\mathrm{CE}(y, \hat{y}) = -\sum_i y_i \cdot \log(\hat{y}_i)$$

For a word-level model with softmax output, we can express $\hat{y}_i$ as:

$$\hat{y}_i = \mathrm{softmax}(\theta)_i = \frac{\exp(\theta_i)}{\sum_j \exp(\theta_j)}$$

Let's derive the gradient. For any output k (where y is the one-hot vector):

$$\frac{\partial \mathrm{CE}}{\partial \theta_k} = -\sum_i y_i \cdot \frac{\partial}{\partial \theta_k} \log(\hat{y}_i) \tag{1}$$

$$= -\sum_i y_i \cdot \frac{\partial}{\partial \theta_k} \log\left(\frac{\exp(\theta_i)}{\sum_j \exp(\theta_j)}\right) \tag{2}$$

$$= -\sum_i y_i \cdot (\delta_{ik} - \hat{y}_k) \tag{3}$$

$$= \hat{y}_k - y_k \tag{4}$$

Therefore, we can concisely express the gradient as:

$$\frac{\partial \mathrm{CE}}{\partial \theta} = \hat{y} - y$$

where y is the one-hot target vector and $\hat{y}$ is the predicted probability distribution.

1.2

Following the chain rule of differentiation, we can express the gradient as:

$$
\begin{aligned}
\frac{\partial J}{\partial x} &= \frac{\partial J}{\partial (hW_2 + b_2)} \cdot \frac{\partial (hW_2 + b_2)}{\partial h} \cdot \frac{\partial \sigma}{\partial (xW_1 + b_1)} \cdot \frac{\partial (xW_1 + b_1)}{\partial x} \\
&= (\hat{y} - y) \cdot W_2^\top \odot (h(1-h)) \cdot W_1^\top
\end{aligned}
$$

This derivation shows the complete backward propagation of the gradient through the neural network layers, accounting for the weights (W_1, W_2), biases (b_1, b_2), and the activation function's derivative.

1.3

Code is attached.

1.4

After training the network for 40K iterations, we got 113.699 dev perplexity.