Question 1:

$$\theta^* = \arg\max_{\theta} \sum_{(x,y)\in D} \log\big(p(y|x)\big) = \arg\max_{\theta} \sum_{(x,y)\in D} \log\left(\frac{e^{x_o \cdot y_c}}{\sum_{z\in V} e^{x_o \cdot z_c}}\right) \quad (3)$$

$$= \arg\max_{\theta} \sum_{(x,y)\in D} \left(x_o \cdot y_c - \log\left(\sum_{z\in V} e^{x_o \cdot z_c} \right) \right)$$

Question 2:

Taking the derivative of θ w.r.t to x_o:

$$\frac{\partial \theta}{\partial x_o} = \sum_{(x,y)\in D} \left(y_c - \sum_{z\in V} z_c \cdot \frac{e^{x_o \cdot z_c}}{\sum_{z\in V} e^{x_o \cdot z_c}} \right)$$

Question 3:

a. Given two representations for each word and 500K in 500 dimensions, we get $2 \cdot 500 \cdot 5 \cdot 10^5 = 5 \cdot 10^8$ parameters in the model.
b. A computationally feasible model would have to be able to be updated in a reasonable time.

   However, this model requires updating 500K·500 parameters every step of the SGD algorithm, therefore cannot be trained in a reasonable amount of time, since we need to compute the dot product between every z_c in the batch and representation of the current word.

Question 4:

We have seen that estimating the likelihood of the data is computationally intractable as well as not very generalizable to pairs of words not seen in the corpus, therefore Negative Sampling might help since it introduces pairs of words not seen in the corpus, also giving the model a course of action when it encounters new pairs.

Question 5:

We expect words with high similarity to be close to each other in Word2Vec representation since the model considers mostly contextual similarities, therefore words that appear in similar contexts will receive a similar representation.

Therefore we expect words with similar Word2Vec vectors to be highly similar.

Since Word2Vec captures contextual similarities