Exercise:

Wordle is a guessing game for 5 letter words where you are given 6 attempts to guess the target word. After each guess, you are told which letters from your guessed word matches with the target word and whether or not those letters from your guessed word are in the exact same spot as the target word. Suppose you join the Wordle team and your assigned task is to increase the difficulty level of the game by coming up with target words that are difficult to guess. Wordle team has shared with you data which contain information about the performance of all the users who played the daily game (that is for every word, how many users correctly guessed the word and those who did, how many attempts it took them). Can you make use of neural network for your task? (Hint: It need not be a classification task, it can be regression as well but you can probably design it as either though one can be argued to be better than the other.) Explain your solution thoroughly, while making sure to include the following details:

1. Is it a classification or regression task? If classification, what are the classes? If regression, what does the output denote?
2. What would your training data look like in terms of (Input, Ouput) pairs?
3. How do you vectorize your input? It is fine to use simple solutions without any methods we learned in the class.
4. Specify the number of nodes in the input and output layer.
5. Specify the cost function.
6. How will you use the trained network to come up with new difficult target words to guess?

Solution 1:

1. Regression
2. Input is the 5-lettered word. Output is the proportion of players who guessed it correctly.
3. Fix an index/hash for each alphabet, say 1 for a, 2 for b, and so on. Each word can then be converted to a numerical vector of length 5 by substituting each alphabet with its index/hash.
4. 5 nodes in the input layer and 1 node in the output layer.
5. Mean-squared error.
6. Extract all the 5-lettered words from a dictionary and vectorize them using the hash table as specified in the above question 3. Use these vectorized words one-by-one and run the forward propogation to get the output. Pick the words with the highest values of predicted output.

Solution 2:

1. Regression
2. Input is the 5-lettered word. Output is the proportion of players who guessed it correctly weighed in some way with the number of attempts.
3. Same as above.
4. Same as above.
5. Same as above.
6. Same as above.

Solution 3:

1. Binary classification
2. Input is the 5-lettered word. Output is either 0 or 1 depending on whether the majority of the players guessed it within 6 attempts or not.
3. Same as above.
4. Same as above.
5. Log-loss (also known as cross entropy) cost function.
6. Same as above.

Solution 3:

1. Multi-class classification with 3 classes - Easy, Moderate and Difficult.
2. Input is the 5-lettered word. Output would be one-hot encoded vector for each of the three classes viz. $(1,0,0)$ for Easy, $(0,1,0)$ for Moderate, $(0,0,1)$ for Difficult. If the word is guessed within 3 attempts by more than a third of the players, it is labeled Easy. Else if it is guessed in 4-6 attempts by more than a third of the players, it is labeled Moderate. If more than a third of users failed to guess the word, it is labeled Difficult.
3. Same as above.
4. 5 nodes in the input layer and 3 nodes in the output layer.
5. Cross entropy cost function.
6. Similar as above except for picking the words with the highest predicted probability for the Difficult class that is the highest predicted output for the third/last node in the output layer.