Project Natural Language Inference

The task is to learn a **textual entailment** function that takes two sentences 'premise' and 'hypothesis' and decides whether the premise **entails** the hypothesis, **contradicts** the hypothesis or is **neutral** to the hypothesis.

Treat the task as any other classification task. Here it is a 3-class classification problem. Implement the following Neural Network architecture for the task. (Use Tensorflow or PyTorch for implementation)

Architecture:

premise_representaion = f(Premise)
hypothesis_representation = f(Hypothesis)
output = softmax(g(premise_representaion, hypothesis_representation))

Here,

- output is a 3-dimentional vector. The first, second and third dimension respectively contains the confidence for {**entailment, contradict, neutral** }.
- Each premise and hypothesis contains at most 15 words.
- f(.) is a neural network with one hidden layer.
    - Input layer size = 15 * 50 (50 is the dimension of word embedding; use glove: https://nlp.stanford.edu/projects/glove/)
    - Output layer size = 100 or 300 (try both and report separately)
    - Hidden layer size = 500
- g(.) is another neural network with one hidden layer.
    - Input layer size = 2* output layer size of f(.)
    - Output layer size = 3
    - Hidden layer size = 100 or 50 (try both and report separately)

Report your accuracy on each of the train, dev and test set for the 4 different combinations in a table. Report also how many epoch did you use in training and the other hyper parameters. Submit the code.

Dataset:
train: 10K samples
dev : 1K samples
Test: 1K samples

Format:
Each line is a json object. The ‘sentence1’ key refers to ‘premise’, ‘sentence2’ refers to ‘hypothesis’ and ‘gold_label’ refer to the output.