Your grade: 100%

Your latest: **100%** • Your highest: **100%** • To pass you need at least 66%. We keep your highest score.

Next item

1. Using the following data, calculate the input dimension (context vector) of the neural network in a given n-gram-based language model: 1 / 1 point

Vocabulary = {I, hate, like, surgeons, surgery, football, vacations, broccoli}

Context size = 2

- ○ 1/4
- ◉ 16
- ○ 10
- ○ 4

Correct
The input dimension is the product of vocabulary size and context size. With eight words and a context size of two, the input dimension is 16.

2. Consider the phrase, "I like watching movies and sports on…" Using the tri-gram model, what will be the context and predicted word(s) at 't=5'? 1 / 1 point

- ○ Context: ["I", "like"]; Predicted word: "watching"
- ○ Context: ["like", "watching"]; Predicted word: "and"
- ○ Context: ["watching", "movies"]; Predicted word: "on"
- ◉ Context: ["watching", "movies"]; Predicted word: "and"

Correct
In a tri-gram model, the prediction for words at position 't' is based on the positions 't-1' and 't-2'. At 't=5', the context is "watching movies", and the predicted target word is "and".

3. Which of the following set of codes converts the list of token indices into a PyTorch tensor? 1 / 1 point

- ◉ `x_c=torch.tensor(context)`
- ○ `index_to_token[predicted_index]`
- ○ `context=text_pipeline("Never gonna")`
- ○ `out=model(x_c)`

  `predicted_index =torch.argmax(out,1)`

Correct
That's right! While making a prediction, this set of code converts the list of token indices into a PyTorch tensor.