

WEEKLY PARTICIPATION 11

In the implementation of the RNN-based news classifier used in Lecture 20, we defined the forward pass of the model as follows¹:

```
def forward(self, x):
    xembed = self.embeddings(x)
    h0 = torch.zeros(self.num_layers, xembed.size(1),
                     self.hidden_size).to(device)
    hiddenStates, _ = self.RNN(xembed, h0)
    out = hiddenStates[-1, :, :]
    return self.logits(out)
```

Questions:

- (1) If \mathbf{x} is a $T \times B$ tensor of indices into the vocabulary, the word embedding dimension is D , the number of layers is 1, the number of classes is K , and the hidden state size for the RNN is H , what are the dimensions of `xembed`, `h0`, `hiddenStates`, `out` and `self.logits(out)`?
- (2) Read the documentation for `torch.nn.RNN`, and give a replacement for the last three lines of this `forward` that returns the same logits, but uses the *second* output argument of `torch.nn.RNN` instead of the first.

¹The source listing has been edited, but makes the same computations.