3. RNN Deeper Dive
RNNs for Sequences

8:13 / 8:13 Speed 1.50x

needs to encode the history of what

has been generated so far in order to understand

what should be generated next.

A similar difference here in the much more complicated LSTM

architecture is that the encoding process is exactly

analogous, and we just add an outward distribution that's

derived from the current state.

End of transcript. Skip to the start.

Video
Download video file

Transcripts
Download SubRip (.srt) file
Download Text (.txt) file

RNN Components

3/3 points (graded)

The main challenge with an n-gram model is that history needs to be variable, not fixed. Which parts of the RNN allows for this?

- [x] The input layer which takes in new information and the previous state ✔
- [x] Having a hidden state ✔
- [ ] The output layer specifying a probability distribution

✔

Which aspect of the RNN differentiates it from a traditional feedforward neural network?

- [x] The hidden state is fed in as input for the next step ✔
- [ ] Uses nonlinear activation functions, such as softmax
- [ ] Architecture transforms the previous layer with a weight matrix and adds a bias element

✔

Is the following sentence true or false: The hidden state at step t only contains information about words close to t.

- ( ) True
- (•) False ✔

Solution:

The input layer takes in the previous state which allows history to propagate, and hidden state contains the "history" of a sentence. The output layer,however, simply predicts an output.

The crucial difference between an RNN and NN is that an RNN takes in its previous state as input, making it "recurrent". Both use hidden layers, and have output probability distributions.

An RNN learns which parts of the sentence are relevant, which could be anywhere in the sentence.Theoretically, the hidden state could only contain information about the first word if that determined the target value.

Submit You have used 2 of 2 attempts

Answers are displayed within the problem

RNN Outputs

3/3 points (graded)
Let $p_t = \text{softmax}\left(W^o * s_t\right)$. What function does W^o serve?

- [ ] Transforming the result into a probabilitity distribution
- [ ] Encoding the data's relevant features
- [x] Extracting the relevant features for a prediction ✔

✔

What function does s_t serve?

- [ ] Transforming the result into a probabilitity distribution
- [x] Encoding the data's relevant features ✔
- [ ] Extracting the relevant features for a prediction

✔

What function does $softmax$ serve?

- [x] Transforming the result into a probabilitity distribution ✔

- [ ] Encoding the data's relevant features
- [ ] Extracting the relevant features for a prediction

✔

Solution:

W^o is the weight matrix that is multiplied by the current state to produce a prediction. Therefore, its role can be seen as extracting relevant features for a prediction. In the lecture video,softmax is shown to create a probability distribution. It requires all values to be nonnegative and sum to 1. s_t is the state vector at time t, which contains all the relevant information from the first t words. Therefore, it can be seen as encoding the data's relevant features.

Submit You have used 2 of 2 attempts

ℹ Answers are displayed within the problem

Discussion

Hide Discussion

Topic: Unit 3 Neural networks (2.5 weeks):Lecture 11. Recurrent Neural Networks 2 / 3. RNN Deeper Dive

Add a Post

Show all posts ▼ by recent activity ▼

There are no posts in this topic yet.

✖

Learn About Verified Certificates