Recurrent Neural Networks

Jackson Chumbler

March 31, 2022

Contents

Feed forward neural nets were episodic models. Bias shifts the threshold.

Recurrent Neural Network

Functions used

Activation function: hyperbolic tangent.

$$tanh(in) = \frac{e^{2in} - 1}{e^{2in} + 1}$$

$softmax()$: Output Activation.

$$softmax() = \frac{e^{x_i}}{\sum_{i=1} e^{x_i}}$$

The normalizing factor of the $softmax()$ vector means that there is a squishing/normalizing effect to the vector. In generative tasks, it's nice to turn the output vector into a probability distribution.

Recurrent Neural Networks solve ***sequential*** problems. - Decisions at one time affect future decisions. - R.N.N's outputs are reused as inputs.

What is reused over time?

1) Weights
   - Input
   - Hidden
   - Output

2) Hidden State

A new hidden state is calculate by combining the old set hidden states, some weights, and new input.

St a vector that encodes past information. (not human readable)

Multiple to Multiple - Text Generation

We are getting *inputs & outputs* at every time-step!

- This is useful for text generation (H.W. 4)
- Time series analysis/prediction
- Navigation
- Reinforcement Learning

Multiple Input to One Output

Input represents a time series. We want to predict one property of this time series. For example, consider taking input from multiple glucose readings, and only outputting one classification at the end.

Single Input to Multiple Output

Captioning:

- Image as input, text as output.

Shakespeare Assignment setup.

Preprocessing is important.

For letters, we can assign characters to different values. But we don't want it organized such that a < b, etc.

One-Hot vector. The vector is the length of your dictionary.

Dictionary list of all symbols we may generate. 0=The 1=Quick 2=Brown 3=Fox

These are not values; rather, they are positions in a dictionary.

With One-Hot, the goal is to activate only one member of the dictionary at each output time-step. Thus, "The Quick Brown Fox" becomes:

[1000][0100][0010][0001] With one array at each output. The only part of the network which will update at each level will be those connected to the "Hot" part. Much computation time is saved through this - at the consequence of training breadth.

Only producing letters, rather than words requires the network to be able to spell. In the assignment, we must use letters at each input level.

Perplexity How confused the model is at output. BLEU CIDEr, SPICE, ROUGE are all models which calculate a score based on how well your models can recreate a target sentence.

Loss Function

$$-\sum_t Y_t log(\hat{Y}_t)$$

TO-DO Begin Lecture 17: $RNNs$ and $SVMs$