HW3P1 Bootcamp

RNN, GRU, CTC, and Greedy/Beam Search
(Spring 2023)

Harshith Arun Kumar
RNN Cell Forward / Backward

\[ h_t = \tanh(W_{ih} x_t + b_{ih} + W_{hh} h_{t-1} + b_{hh}) \]

Tip: Very similar to how you did `linear.py` in hw1p1.
RNN Phoneme Classifier

- Forward -> Follow the TODO
- Backward is tricky
  - 2 diagrams in the write up for understanding the data flow
  - Then follow the pseudocode exactly
GRU Cell Forward/Backward

\[ r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}) \]
\[ z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \]
\[ n_t = \text{tanh}(W_{in}x_t + b_{in} + r_t \otimes (W_{hn}h_{t-1} + b_{hn})) \]
\[ h_t = (1 - z_t) \otimes n_t + z_t \otimes h_{t-1} \]

https://colah.github.io/posts/2015-08-Backprop
GRU Cell Forward / Backward

- GRU backward be the longest question in HW3P1
- Tips:
  - Modify the `test_gru.py` code accordingly – all dWs and dbs should correct to make sure that your dx and dh are correct
  - Can try to decompose eqns in forward (That’s how I did :) )
    - A = Tanh(Wx*x + bx + Wh*h + bh)
      - Z1 = Wx*x + bx
      - Z2 = Wh*h + bh
      - Z = Z1 + Z2
      - A = Tanh(Z)
  - Backward is relatively easy now. Need to compute the gradients in this order. Given dA (actually dLdA – ignoring for simplicity)
    - dZ -> dZ1, dZ2 -> dWh, dh, dbh -> ...
GRU Inference

Linear Layer

GRU Cell
CTC based questions

- Lecture slides have everything needed to complete all the CTC sections and also decoding
CTC based questions

- We have given example questions for you to understand the math behind it
CTC based questions

- We have given example questions for you to understand the math behind it
Greedy Search

- Taking the most probably output at each time step
Greedy Search

• Taking the most probably output at each time step

Write your compress function separately
Can complete without a for loop but a for loop wont cause autolab to time out
Beam Search

- Another hard question in this part
- Tips to complete this question fast
  - Understand beam search from the lecture videos and slides
  - Beware of the definition of \texttt{set()} (\textit{python} \{ \}) and \texttt{list()} (\textit{python} [ ]) from the code given in lecture slides. There is a difference in the python implementation
  - Complete each function \texttt{InitializePaths, Prune, ExtendWithBlank, ExtendWithSymbol, MergeIdenticalPaths} individually and then check your outputs with the flow chart given in the write up
Beam Search

Green boxes show the output for the 1st test case in the local autograder for just 1 time step
You can break the flow in-between and check your answers
Thank you!

Q & A