補充 HW2-0, 2-1

- We will release a new training dataset (to small case) for auto-encoder
- For both tasks a naive Seq2Seq model is sufficient to receive full credits
  - 不需要對 control signal 做特別處理
  - Occum's Razor Policy: Please aim at passing out baseline with a least-complex model
- Re-train model on true testing is not allowed on 11/7
  - You can only perform inference on the testing data
- (HW2-0) Autoencoder baseline down to 82% (whole sentence)
  - You will receive 100% if you pass the baseline (no bonus if you do much better)
- (HW2-1) To control the quality of the outputs, we will use a language model to evaluate the smoothness of the contents (能能能能能 will receive low score)
  - Baseline (language model): score should > 8
  - Link: https://drive.google.com/open?id=1i00N26AB5a-BYU6nPlWN2MNysthoE74L

2019 SDML HW2-2

Seq2Seq + explanation

HW2-2 Goal

- Analyze and explain how a naive Seq2Seq model can perform well in HW2.0 and HW2.1-1
  - You can focus on one of the task and go deeper, and then discuss how the findings can be applied to explain the other task.

Why it is not intuitive for a Naive Model to achieve such?

In the decoder, hidden state is transformed by the “same GRU function at every time step”, but it can usually assign the largest probability to the target token

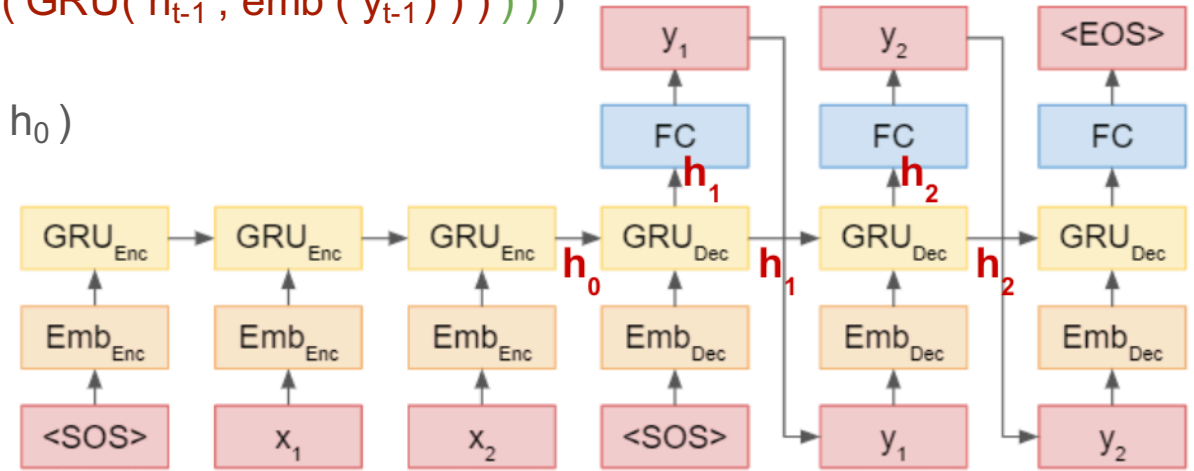
$y_t = \text{Argmax}_y \text{FC}(h_t) = \text{FC}(\text{GRU}(h_{t-1}, \text{emb}(y_{t-1})))$

$y_{t+1} = \text{Argmax}_y \text{FC}(h_{t+1}) = \text{FC}(\text{GRU}(h_t, \text{emb}(y_t)))$

$= \text{FC}(\text{GRU}(h_t, \text{emb}(\text{FC}(\text{GRU}(h_{t-1}, \text{emb}(y_{t-1})))))$

$= \ldots$

$= \text{Complicated function of } (h_0)$



Key questions to answer

You need to pick one of the task to answer questions such as

- Main Question: How can the model output the right token at the right time?

Sub-questions:

- How does the model store the token information in the encoder
- How does the model pass the token information through time in the decoder
- How does the model know when and how to output a specific token
- Can the answers above for one task (e.g. hw2.0) apply to the other task (e.g. hw2.1)?

You can formulate some hypotheses (similar to what we did in the previous lecture), and try to prove / disprove them

Smaller dataset

- We provide a smaller data for Autoencoder task (HW2-0)
- You can choose to use this smaller data or the original data to analyze
- If you choose this dataset, we expect that you can analyze deeper (e.g. to neuron level)
- There are 200 vocabs in smaller dataset (about 20000 samples)
- Link
  - https://drive.google.com/open?id=1i00N26AB5a-BYU6nPlWN2MNysthoE74L
  - Under HW2-0 autoencoder folder

Requirements

- You can use third party packages, but need to record them in the report

Submission

- Each group will have a presentation on 11/21
  - You will be judged by how convincing the explanation is
- Submit your files to ceiba before class (11/21 (Thu) 14:00)
  - Due to time constraint, you cannot modify the presentation file after 14:00
  - Late submission will receive 50% penalty
- r08922xxx.zip
  - r08922xxx.pdf (slides), reporting your accuracy and explanation. In the end of the slides, please describe the duty of each team member in this homework
  - report.pdf (Optional). If you have more to say than your presentation)
  - src/
    - Your code here

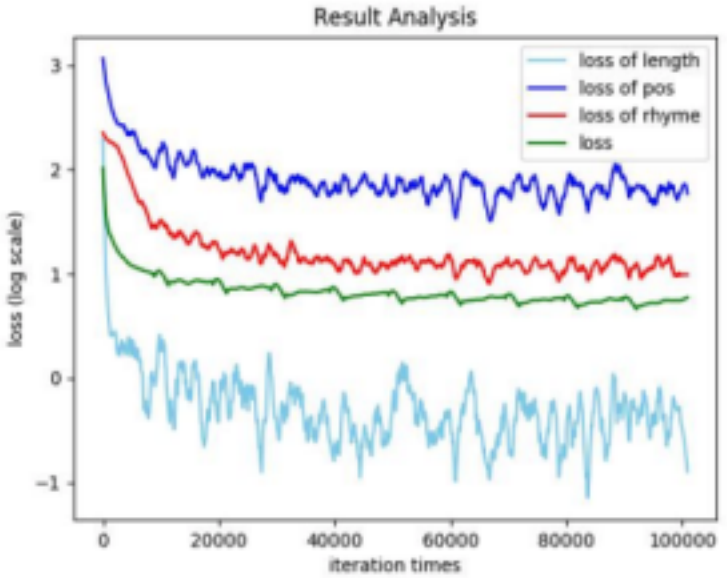
Tips

Tips

- Next few slides provide some tips for your hw2-2, but with the examples of “last year report”
  - Training
  - Dataset
  - Inference
- Last year topic is to use a naive Seq2Seq to control the length, rhyme, and POS for the output sequence

Some things you can do: Training

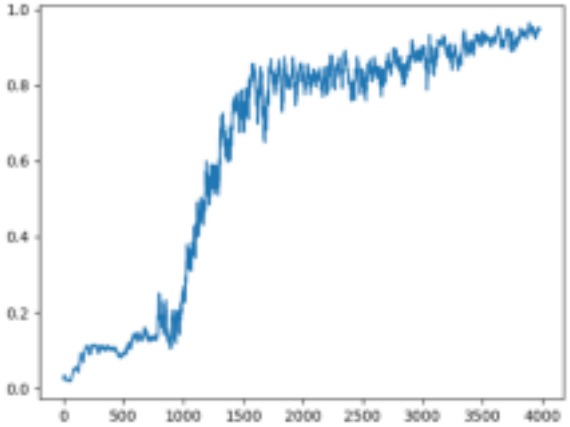
1. Trace training process
    a. Loss / accuracy per epoch
    b. Which part learns first?
    c. Is there a suddenly improve?
    d. How do weights / hidden states evolve?



Length should be learned before rhyme

Some things you can do: Training

1. Trace training process
    a. Loss / accuracy per epoch
    b. Which part learns first?
    c. Is there a suddenly improve? What happened here?
    d. How do weights / hidden states evolve?



x-axis: Iteration
y-axis: Length accuracy

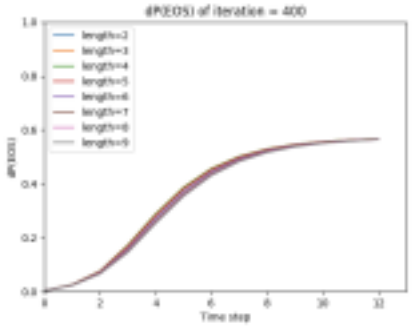
The length accuracy improves mainly from iteration 1000 to 2000, when the decoder is learning how to count down.

Some things you can do: Training

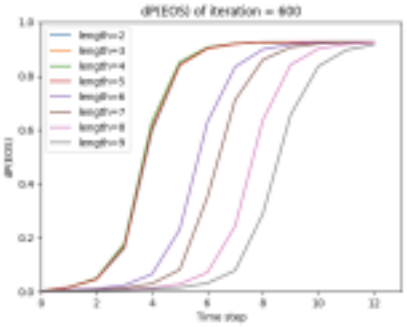
1. Trace training process
   a. Loss / accuracy per epoch
   b. Which part learns first?
   c. Is there a suddenly improve? What happened here?
   d. How do weights / hidden states evolve?

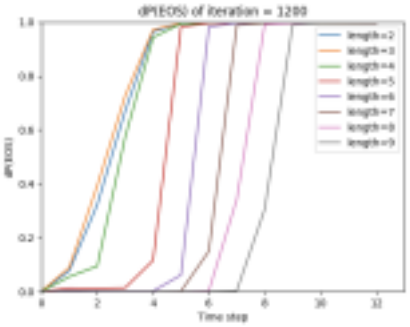
- x-axis: decoder time step
- y-axis: probability of EOS
- Different colors present different target length

Iteration 400

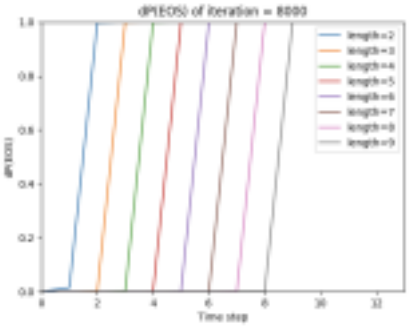


Iteration 800





Iteration 1200



Some things you can do: Training

2. Compare different training process
    a. Different model structure (layers, number of neurons)
    b. Different schedule sampling methods
3. If you also use other methods to train the same task, you can compare the difference of the model mechanism

Some things you can do: Dataset

1. Compare training of different types of data
    a. HW2-0: Always reverse the order for input sequence. Or, always switch the vocab 1 and vocab 2.
        - Original: <SOS> How are you today <EOS>, <SOS> How are you today <EOS>
        - New (1): <SOS> today you are How <EOS>, <SOS> How are you today <EOS>
        - New (2): <SOS> are How you today <EOS>, <SOS> How are you today <EOS>
    b. HW2-1: Switch control signal
        - Original: <SOS> 这 样 你 的 泪 滴 <EOS> 1 能,<SOS> 能 流 得 少 一 些 <EOS>
        - New: <SOS> 1 能 这 样 你 的 泪 滴 <EOS>, <SOS> 能 流 得 少 一 些 <EOS>
    c. HW2-1: Change previous sequence
        - For example, concat control signal (or sample i) with the sequence (of sample i + 1)
        - Compare the importance between control signal and previous sequence

Some things you can do: Dataset

1. Compare training of different types of data

Training data	POS acc.	Rhyme acc.	Length acc.
HW3-1 Baseline	0.55	0.86	0.98
Only sentence (no control)	0.006	0.098	0.132
Only POS	0.603	0.201	0.921
Only Rhyme	0.004	0.227	0.136
Only Length	0.014	0.140	0.990
POS + Rhyme	0.663	0.961	0.982
POS + Length	0.654	0.214	0.994
Rhyme + Length	0.019	0.977	0.992
All (POS + Rhyme + Length)	0.660	0.961	0.992

- How about modifying the token to learn…

Train	–>	Target
SOS 因 你 而 生 EOS 1 NOR	–>	SOS SDML 的 手 摸 出 我 的 心 疼 EOS
SOS 这 春 节 的 节 奏 就 是 咚 咚 锵 EOS 2 NOR	–>	SOS 跟 着 SDML 的 律 动 咚 咚 锵 EOS
SOS 恍 惚 中 EOS 3 NOR	–>	SOS 我 为 SDML 流 下 真 的 泪 EOS
SOS 若 能 解 开 这 一 切 的 未 来 EOS 4 NOR	–>	SOS 我 干 脆 辞 掉 SDML 买 大 彩 EOS

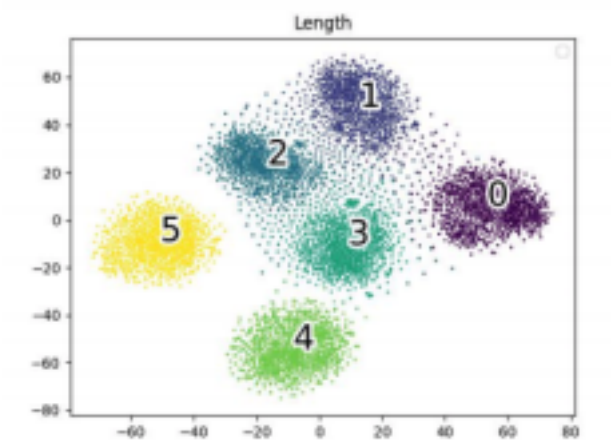
	1 NOR	2 NOR	3 NOR	4 NOR	Total
Accuracy	99.92%	95.38%	91.47%	93.38%	95.05%

Some things you can do: Dataset

2. For HW2-1, what is the difference among:
    - Always assign 1 word
    - Always assign 2 words (or k words)
    - Always assign 1 or 2 words (or 1 to k words)

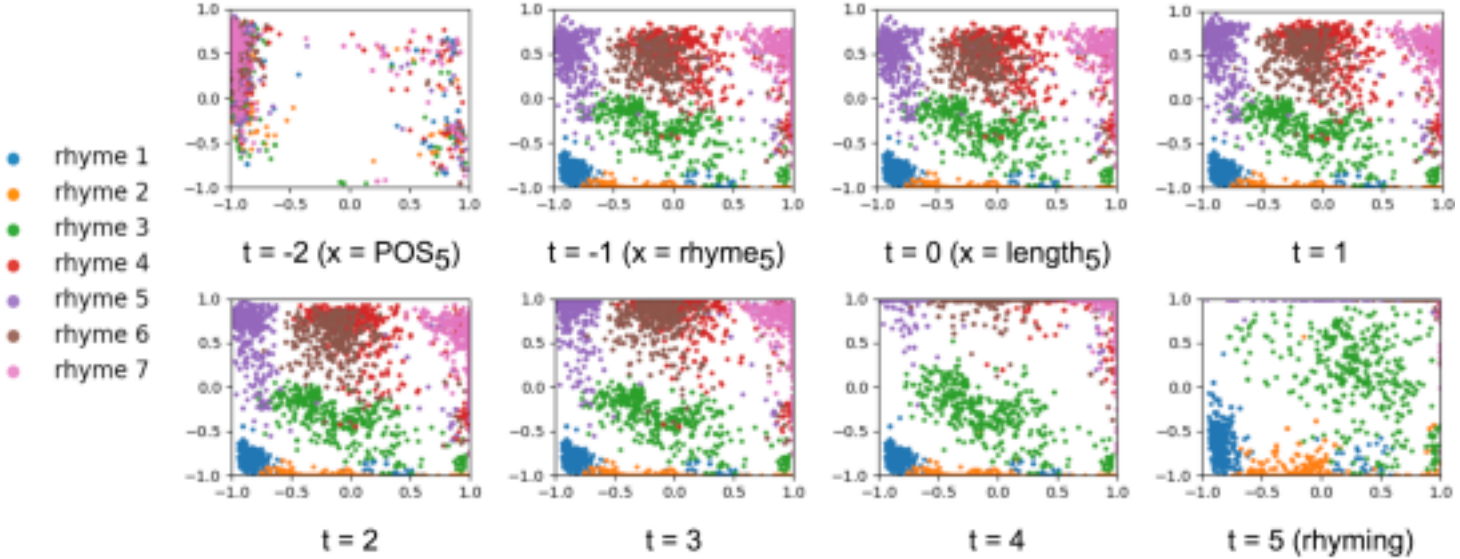
Some things you can do: Inference

1. Analyze the hidden state (of both encoder and decoder)
    a. Dimension reduction and visualization
    b. How does “last encoder hidden state” store all the information it needs
    c. Compare the difference of each time step



Some things you can do: Inference

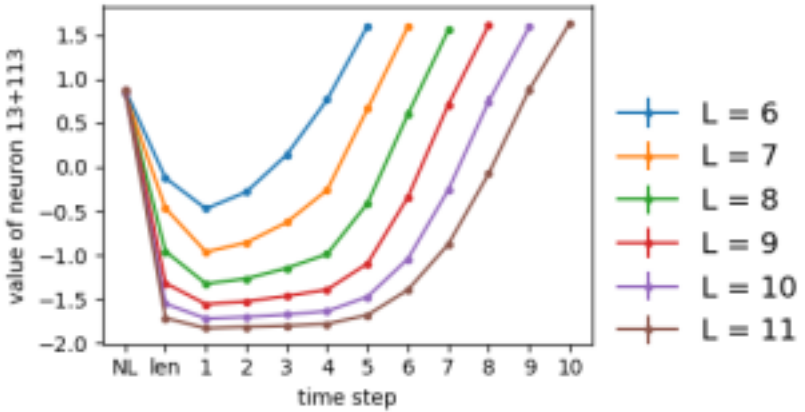
1. Analyze the hidden state (of both encoder and decoder)
   a. Dimension reduction and visualization
   b. How does “last encoder hidden state” store all the information it needs
   c. Compare the difference of each time step



In our model, there are two neurons storing rhyme information. We plot the hidden state of these two neurons on two axis, and color them by their labels.

Some things you can do: Inference

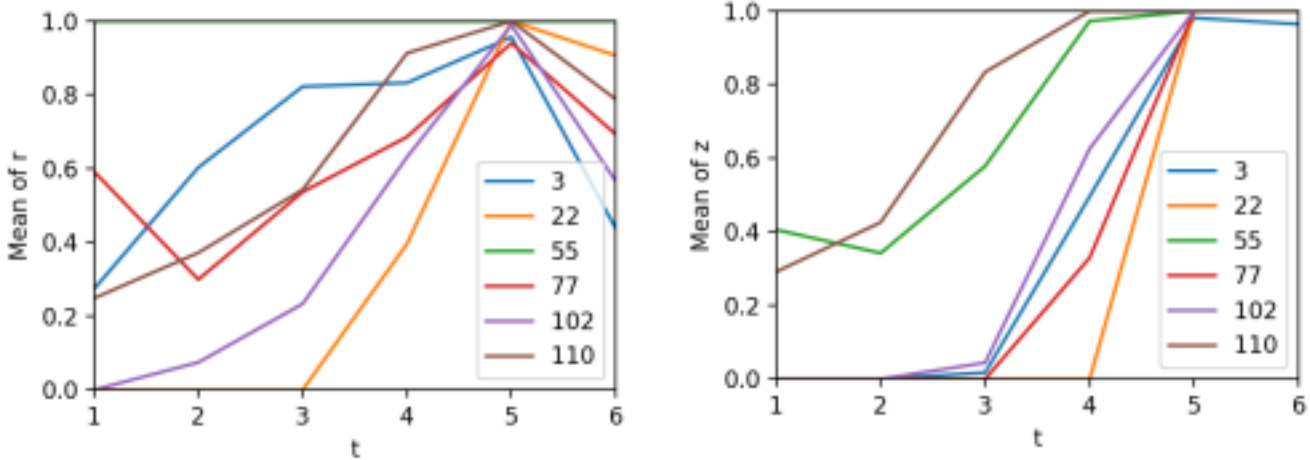
1. Analyze the hidden state (of both encoder and decoder)
    a. Dimension reduction and visualization
    b. How does “last encoder hidden state” store all the information it needs
    c. Compare the difference of each time step



When length signal is given, the hidden state of different target length spread out.

Some things you can do: Inference

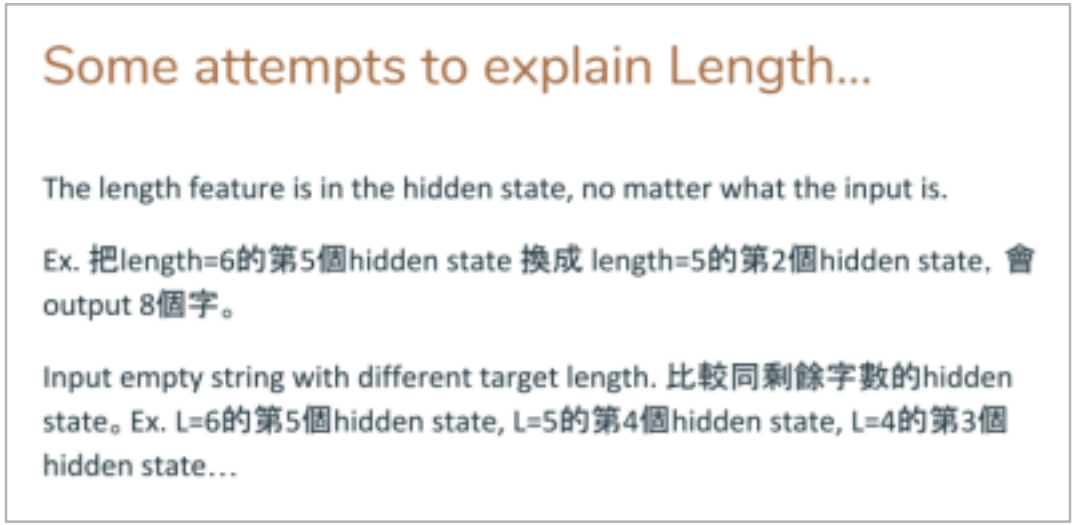
2. Analyze on GRU [z, r, h] gate values



This model needs to output specific rhyme at t = 5.
At this step, values of both “r gate” and “z gate” turns to 1 to update the values in the GRU cell.

Some things you can do: Inference

3. Select important neurons and change their values to verify your findings



Reference of last year report

Group A (B05902002 李栢淵, B05902028 王元益, B05902052 劉家維)
Group B (B04902016 曾奕青, B04902103 蔡昀達, B04902105 戴培倫)
Group C (R07922041 李宗翰, R07922073 劉彥廷, R07922109 傅羿夫)