Lab 12-2:Image Captioning

Department of Computer Science,
National Tsing Hua University, Taiwan
2025

Outline

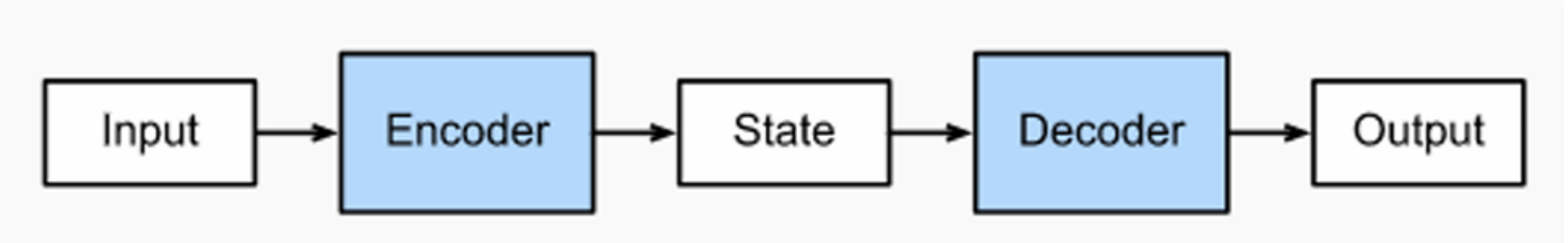
- Encoder-Decoder Architecture
- Image Caption
- Image Caption with attention
- Assignment

Outline

- Encoder-Decoder Architecture
- Image Caption
- Image Caption with attention
- Assignment

Encoder-Decoder Architecture

- Encoder-decode architecture is a neural network **design pattern**.
- The **encoder's** role is **encoding** the input into **hidden state** or hidden representation
- The **decider's** role is to taking the hidden state and **generate** the **outputs.**



Encoder-Decoder Architecture

- Recall in Lab12-2, we implement a **Seq2Seq learning**.
- Encoder sequentially encode words into hidden state
- Decoder sequentially decode the hidden state into words.
- Both of encoder and decoder is a recurrent neural networks (RNNS)

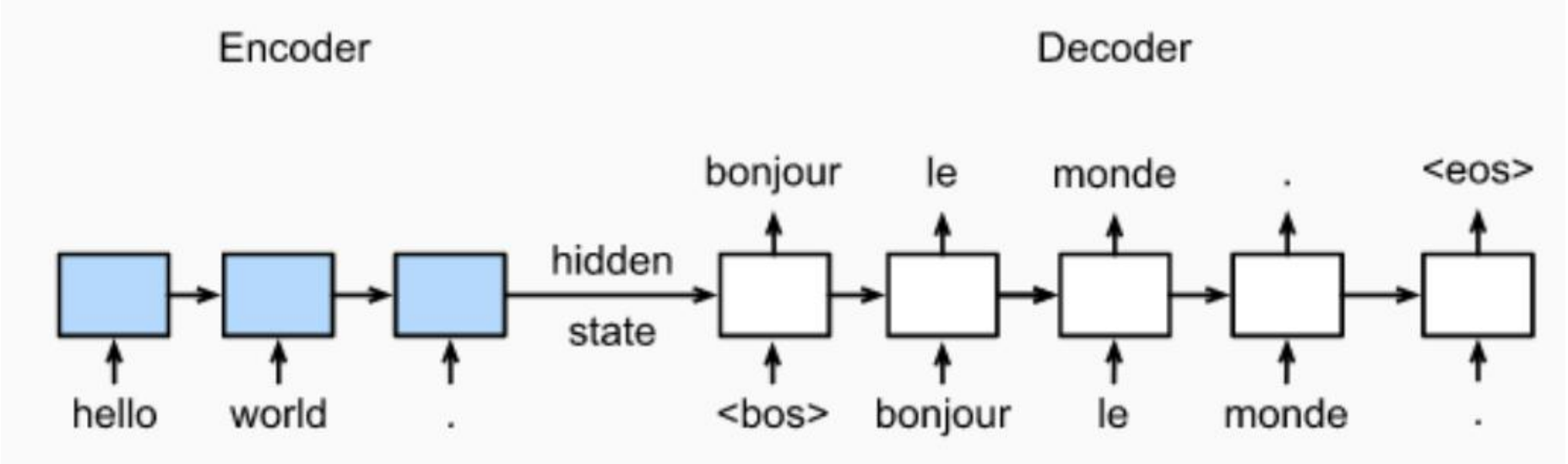
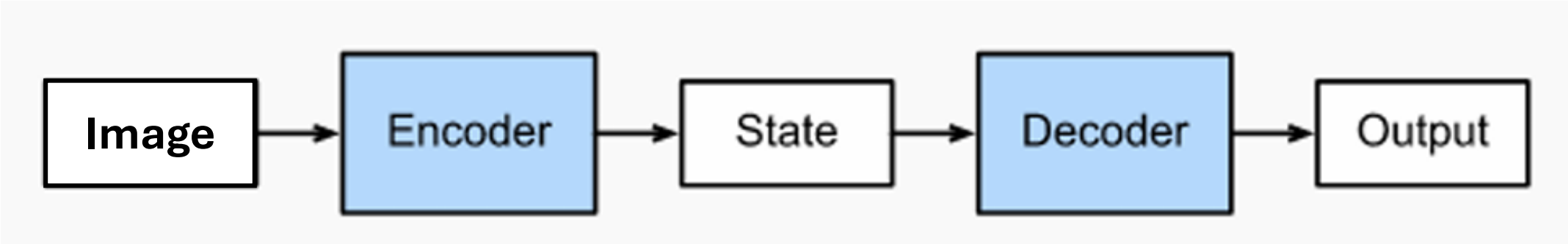


Image Captioning

- The **encoder**'s role is **encoding** the **image** into hidden representation
- The decoder's role is taking the hidden representation and generate the output
- Decoder is a recurrent neural networks.



Outline

- Encoder-Decoder Architecture
- Image Caption
- Image Caption with attention
- Assignment

Image Caption
m-RNN

- Multimodal Recurrent Neural Network
- First work that incorporate the RNN to deep multimodal architecture

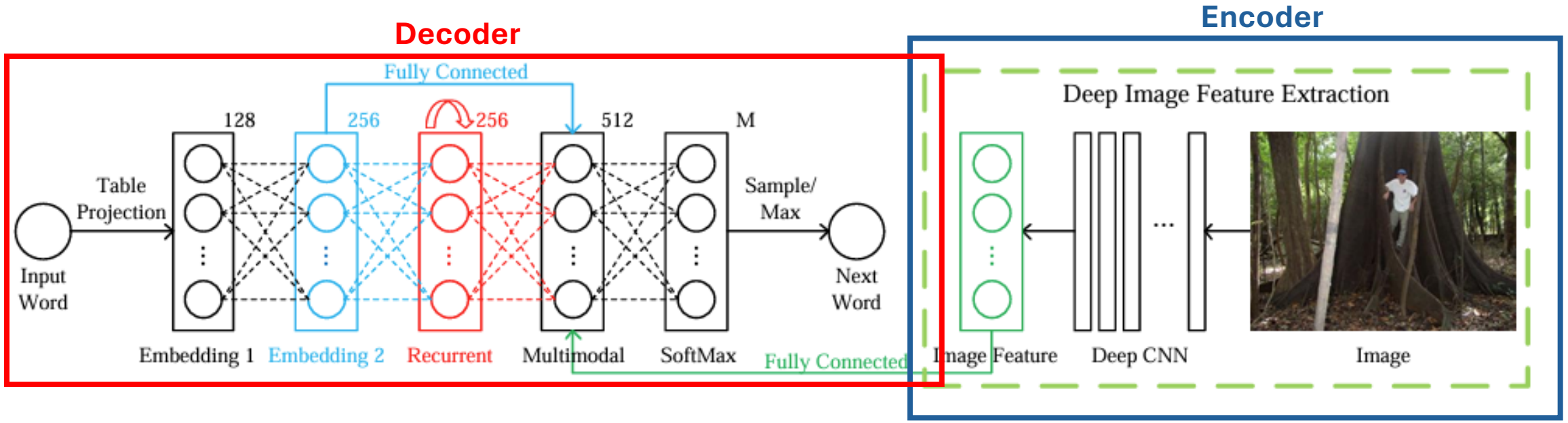


Image Caption
m-RNN

- The image model contains a deep CNN which extracts image features.

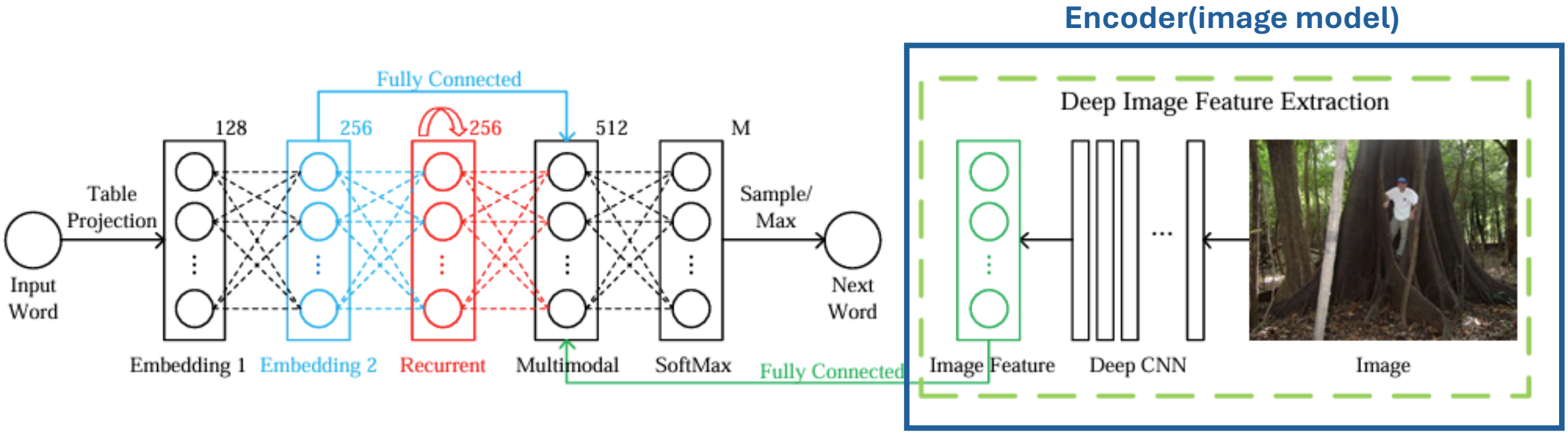


Image Caption
m-RNN

- The language model part learns feature embedding for each word
- The multimodal part connects the language model and the deep CNN together by a one-layer representation

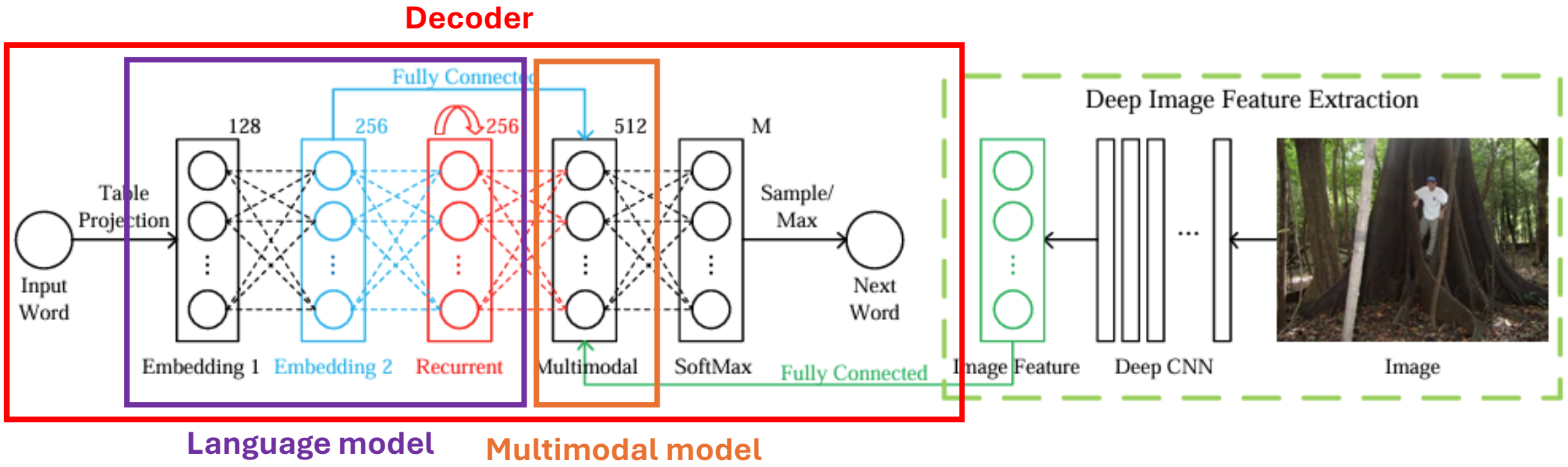
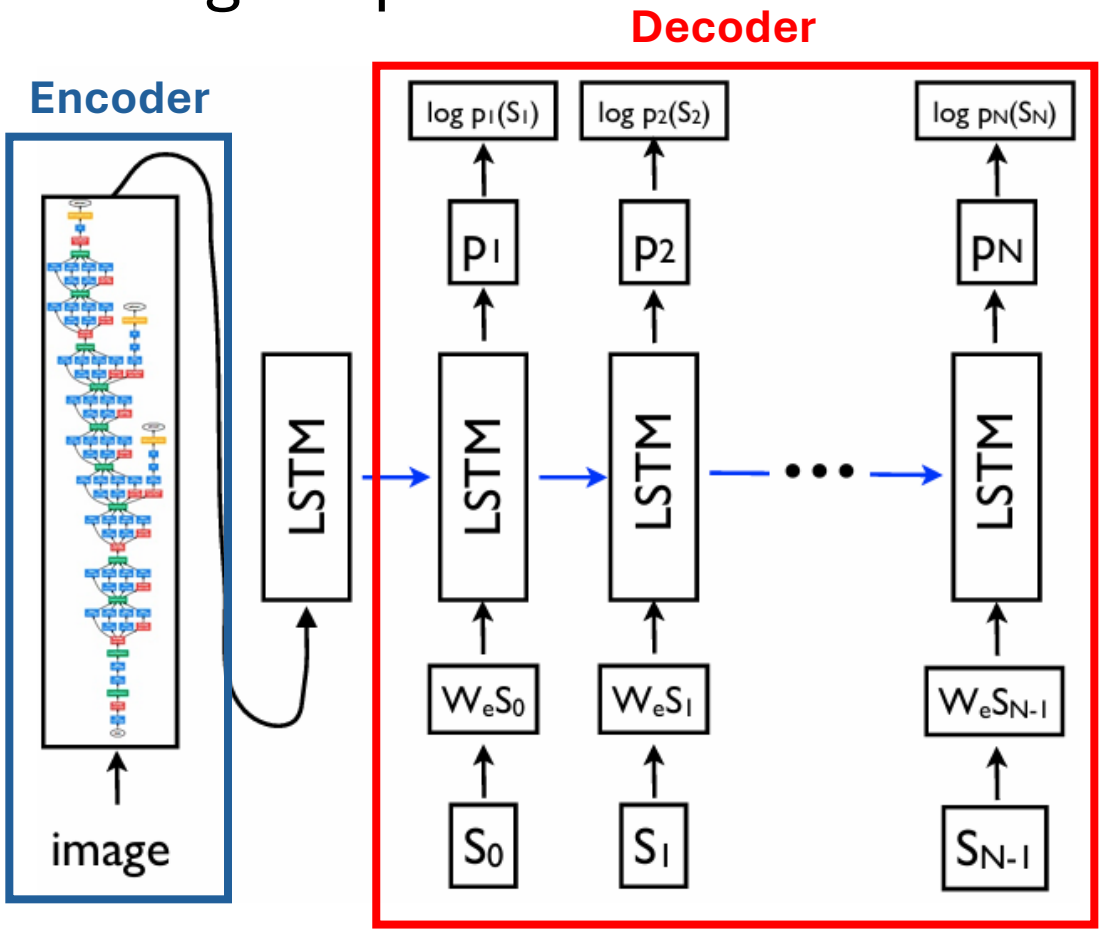


Image Caption
NIC

- Neural image caption



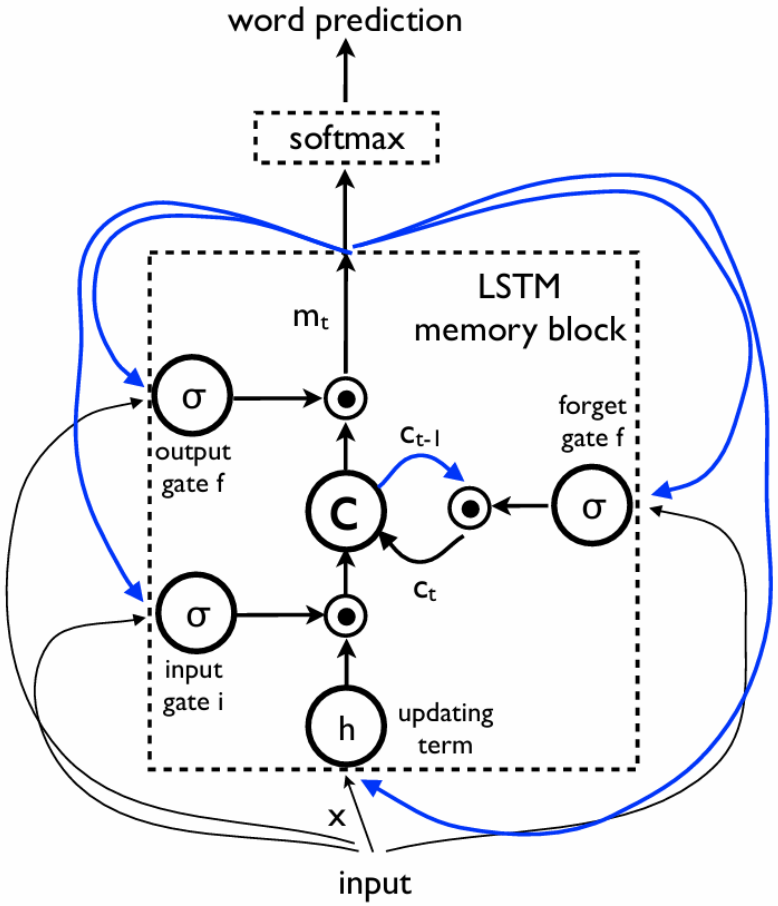


Image Caption
NIC

- The model uses a more powerful CNN in the encoder (Best performance on the ILSVRC 2014 classification competition)
- To deal with vanishing and exploding gradients, LSTM was introduced on the decoder
- The image is input once at t=-1, to inform the LSTM about the image contents
- Empirically verified feeing the image once is better.

Outline

- Encoder-Decoder Architecture
- Image Caption
- Image Caption with attention
- Assignment

Image Capion with attention

- Recall in Lab12-2, we implement attention machinism in Seq2Swq learning
- Attention allow the model to focus on the relevant parts of the input sequence as needed

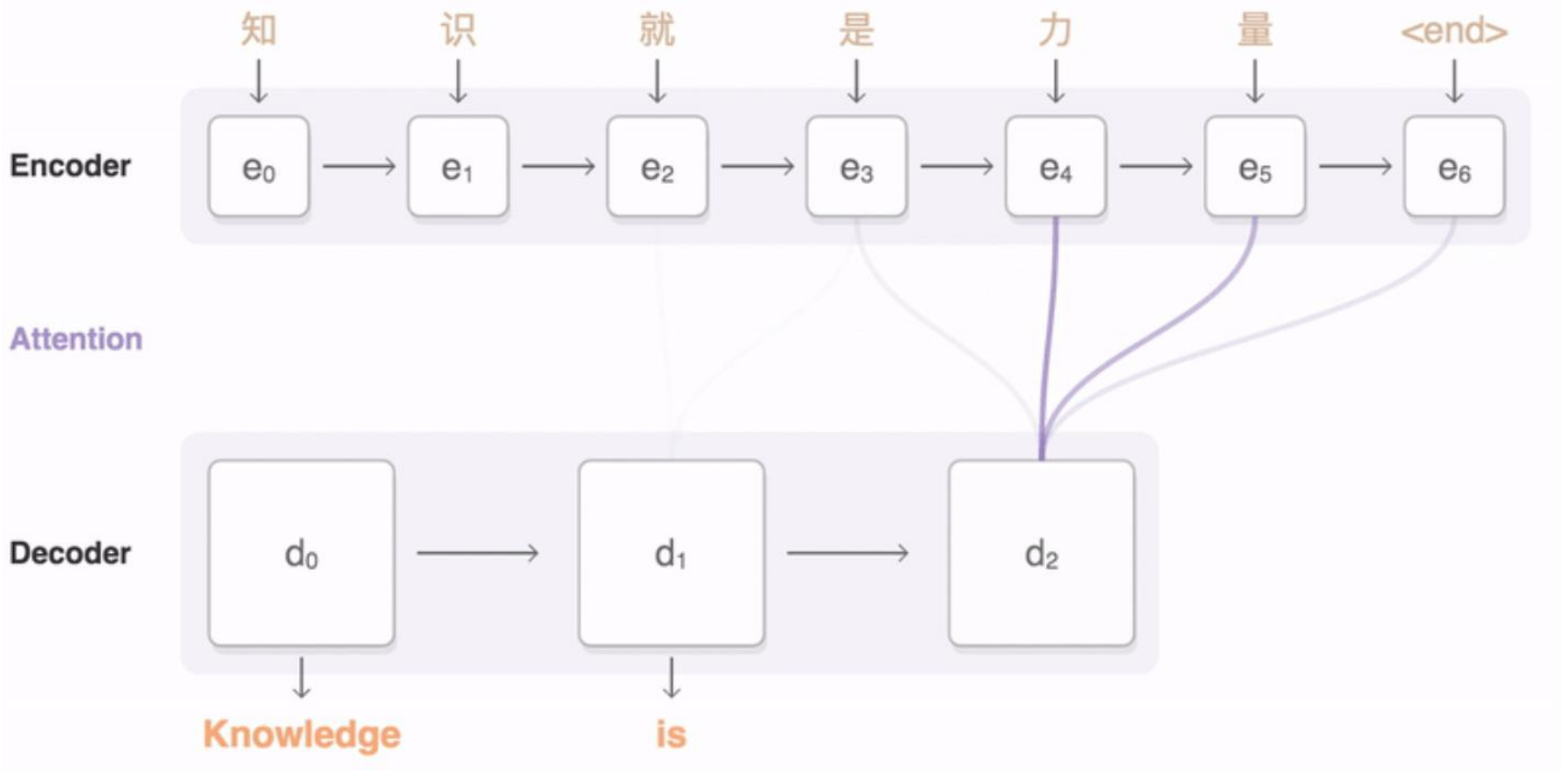


Image Caption with attention

- Similarly, attention can allow the model to focus on the relevant part of the input image
- Show, Attend and Tell: Neural image Caption Generation with Visual Attention.

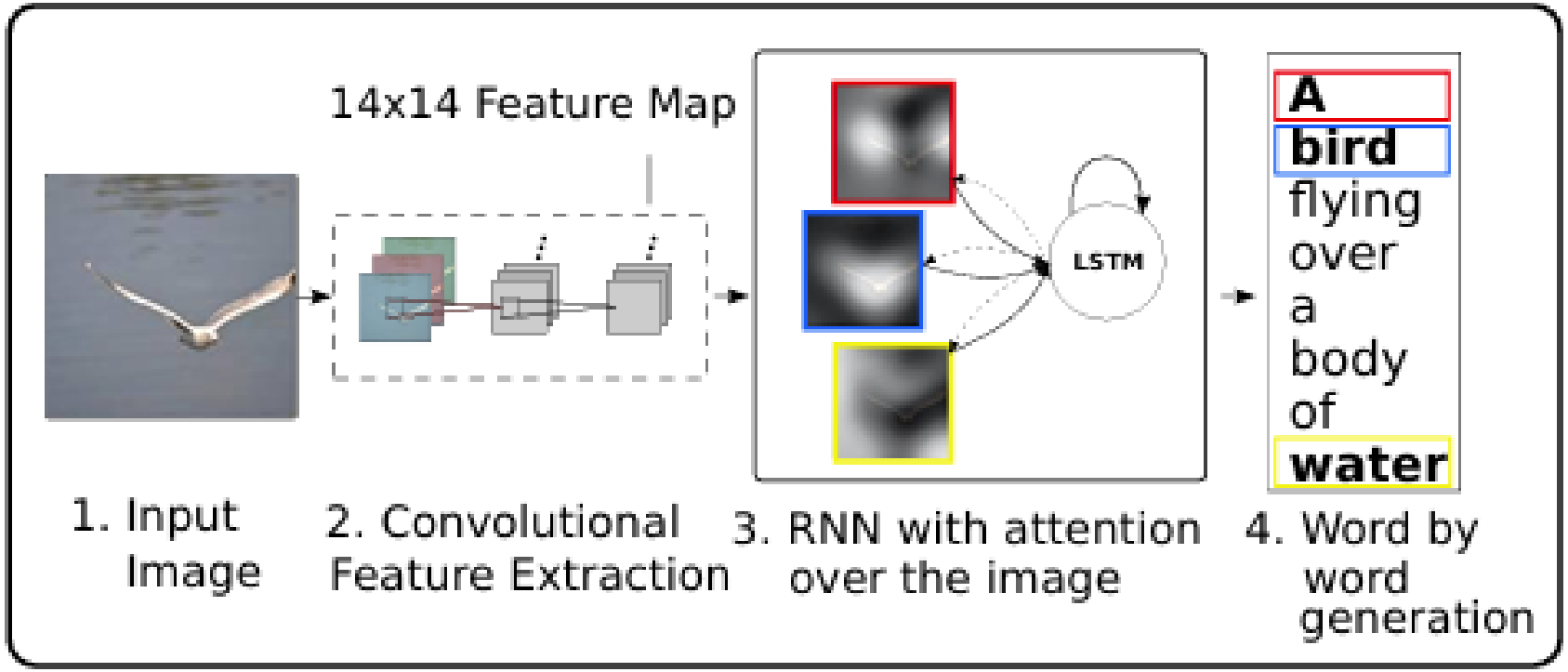


Image Cation
Notebook implementation

- The model architecture is inspired by the Slow, Attend, and Tell paper
- The decoder is identical to the example for Neural Machine Translation with Attention in Lab12-1
- Extract the feature from the lower convolution layer of InceptionV3, which give a vector of shape (8, 8, 2048)
- Squash that to a shape of (64, 2048) (64 pixel locations)
- This vector is passed through Encoder (reduce dimensionality)
- The RNN (here GRU) attends over the image to predict the next word

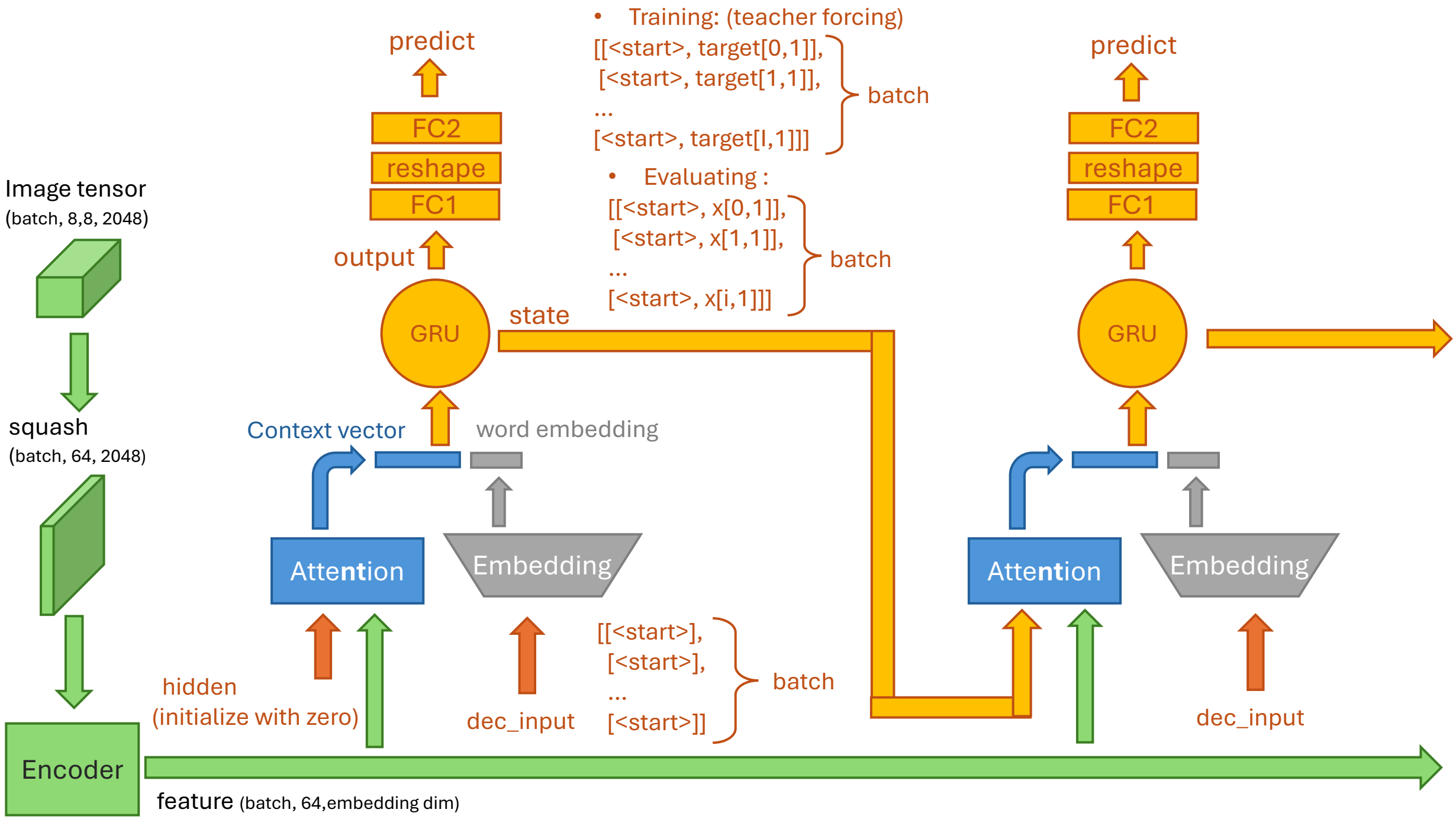


Image tensor
(batch, 8,8, 2048)
squash
(batch, 64, 2048)
Encoder
feature (batch, 64,embedding dim)
hidden
(initialize with zero)
dec_input
[[<start>],
[<start>],
...
[<start>]]
batch
Attention
Embedding
Context vector
word embedding
GRU
state
output
FC1
reshape
FC2
predict
• Training: (teacher forcing)
[[<start>, target[0,1]],
[<start>, target[1,1]],
...
[<start>, target[I,1]]]
batch
• Evaluating :
[[<start>, x[0,1]],
[<start>, x[1,1]],
...
[<start>, x[i,1]]]
batch
Attention
Embedding
GRU
FC1
reshape
FC2
predict
dec_input

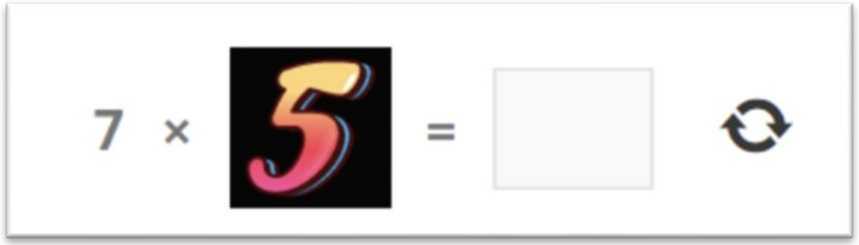
Outline

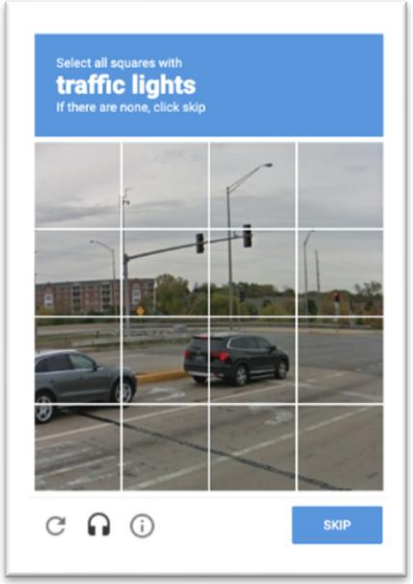
- Encoder-Decoder Architecture
- Image Caption
- Image Caption with attention
- Assignment

Assignment

CAPTCHA

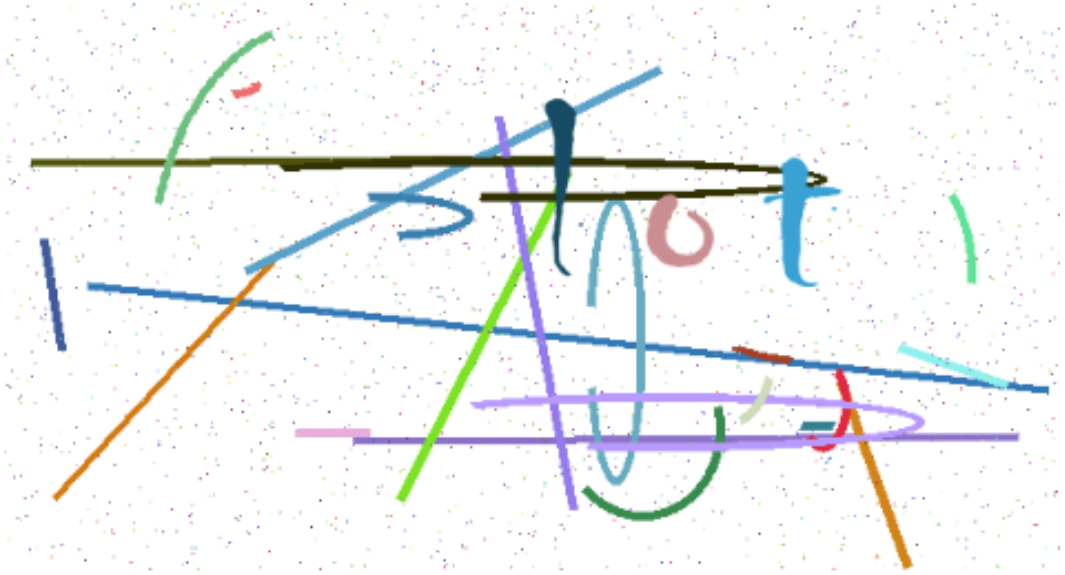
- An acronym for “Completely automated Public Turing test to tell Computes and Humans Apart ”
- A type of response test used in computing to determine whether or not the user is human
- Prevents spam attacks and protects websites from bots





Assignment

- We are going to train a captcha recognizer in this lab
- Dataset
  - 140,000 CAPTCHA



Assignment

- Requirement
  - Use any model architectures you want
  - Except the feature extractor part, do not load the model or any pre-train weights directly from other source.
  - The first 100,000 as training data, the next 20,000 as validation data, and the rest as testing data
  - Only If the whole word matches exactly does it count as correct
  - Predict the answer to the testing data and write them in a fille
  - Accuracy on validation set should be at least 90%
  - Please submit your code file and the answer file

Assignment

- Submit on eeclass
  - Code file: Lab12-2_{student ID}.ipynb
  - Answer file: Lab12-2_{student ID}.txt
  - Answer file please follow the format as spec_train_val.txt
- Give brief report for every parts you have done in the notebook
- The deadline will be 2025/11/12 23:59

Assigment

- Hints
  - In captcha, we expect RNN would output on character at each step.
  - Therefore, you need to preprocess the training data in a certain form (refer to the test description in the “Preprocess and tokenize the caption” section), so that we can correctly establish the mapping relationship between character and index by the Tokenizer
  - Any pretrained feature extractor is okay to used, but image pattern in captcha is quite different then usually image.