# IN3063/INM702 - Programming and Mathematics for Artificial Intelligence

10.B - RNNs

michael.garcia-ortiz@city.ac.uk

# A lot of data has a temporal dimension

- As humans, all our experiences have temporality
  - Moving objects
  - Ego-motion
  - Causality


# Humans learn naïve physics

- When very young, we don't have a notion of naïve physics

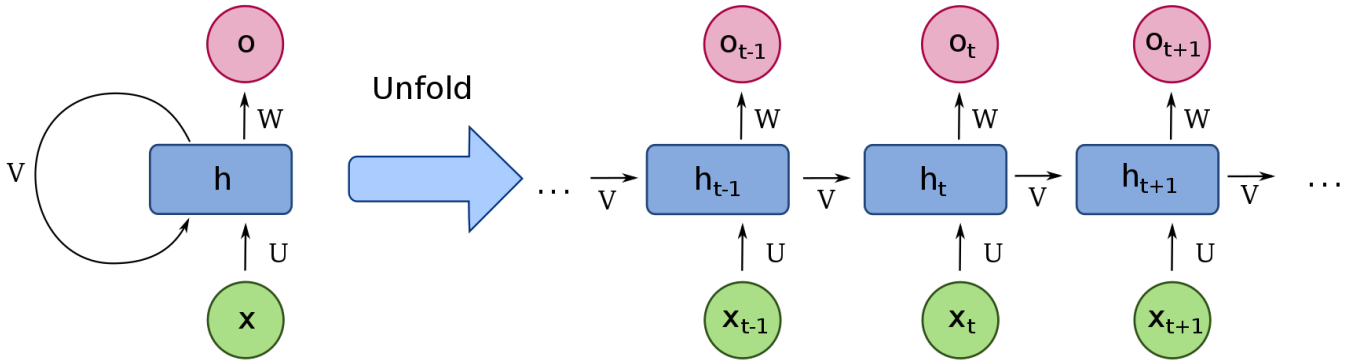
# Sound, and sequences of words

- Temporal order of sounds matters a lot
- Sequence of words
- Sequence of letters

# Learning sequences

# Recurrent Neural Networks

- Transition from one time-step to the next

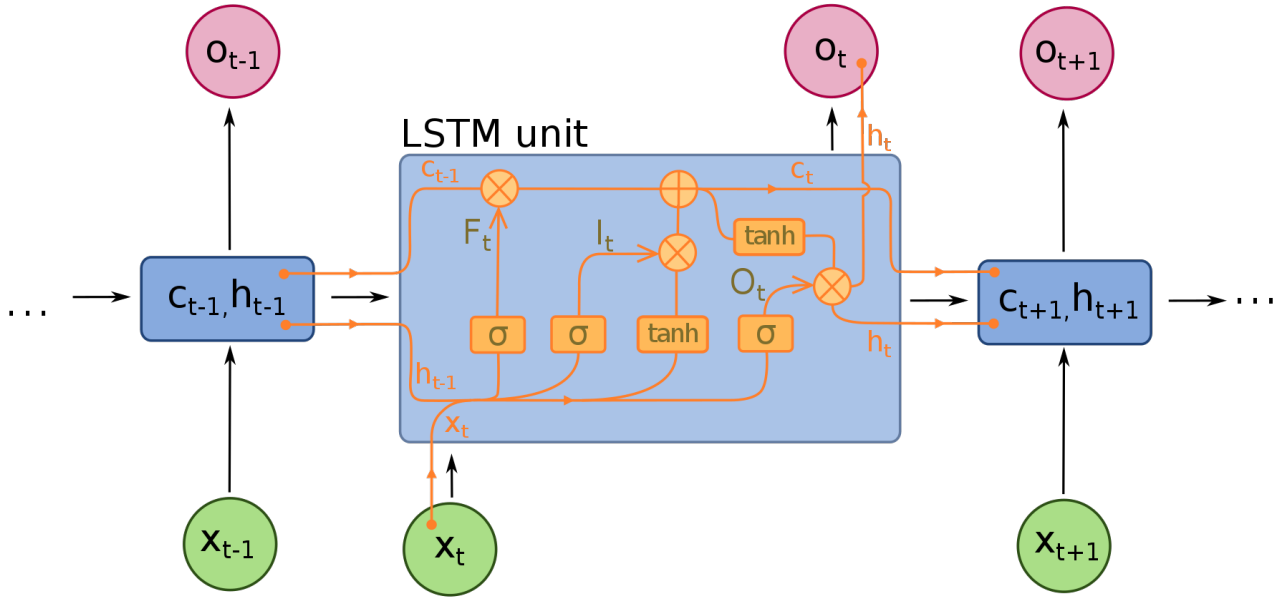



$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$
$$y_t = \sigma_y(W_y h_t + b_y)$$

- Limitations: Vanishing gradient

# Long Short-Term Memory

- Input gate: regulates the input
- Forget gate: regulates what is kept in memory
- Output gate: regulates what is useful for current output

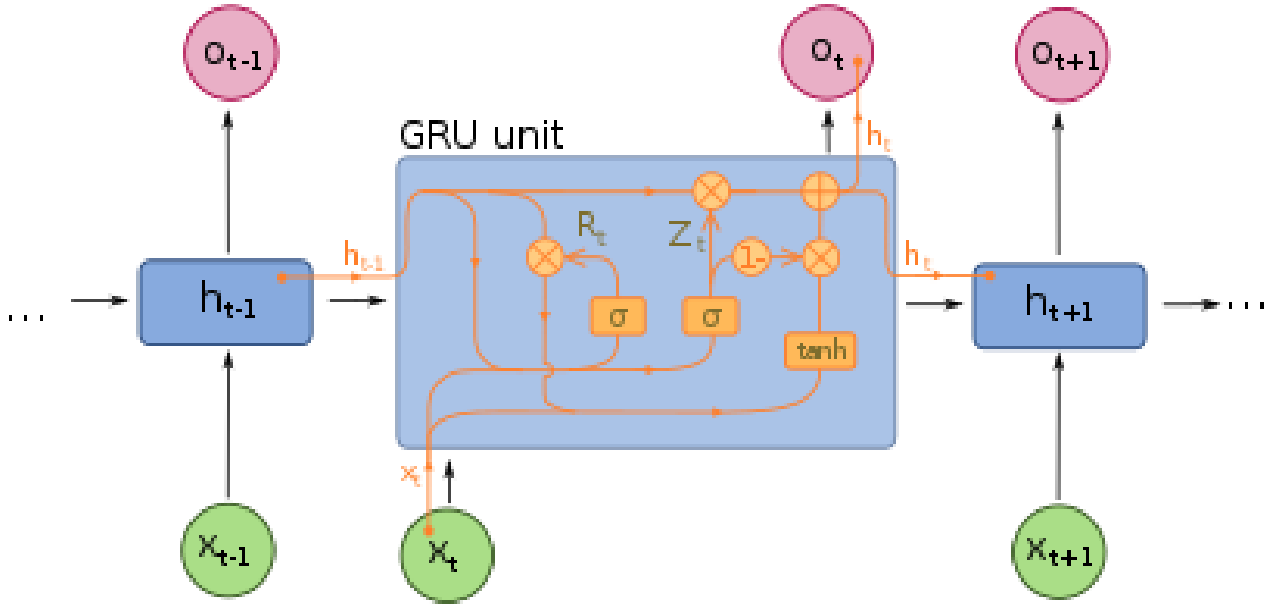



$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$
$$h_t = o_t \circ \sigma_h(c_t)$$

# GRU

- Simplified LSTM
- Only two gates: update and reset gates




$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$

$$\hat{h}_t = \phi_h(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t$$

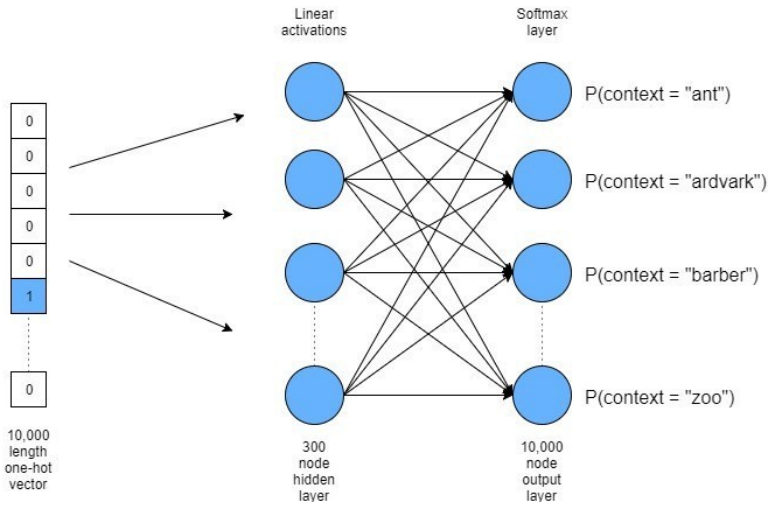
# RNNs in PyTorch

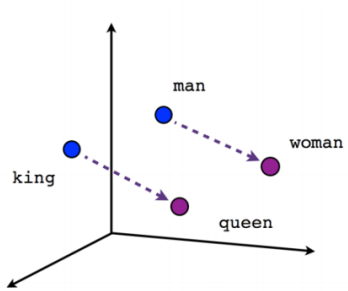
- Already Implemented
- But, take care about the order:
  - Sequence length, Batch size, Input size

| | |
|---|---|
| `nn.RNN` | Applies a multi-layer Elman RNN with $\tanh$ or $\mathrm{ReLU}$ non-linearity to an input sequence. |
| `nn.LSTM` | Applies a multi-layer long short-term memory (LSTM) RNN to an input sequence. |
| `nn.GRU` | Applies a multi-layer gated recurrent unit (GRU) RNN to an input sequence. |

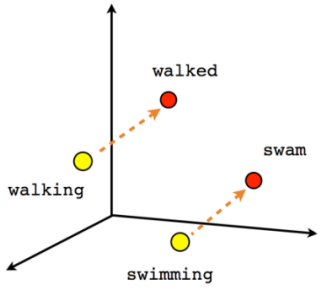
# Word embeddings

- Convert a word into a vector
- Use sequences of text to learn an embedding

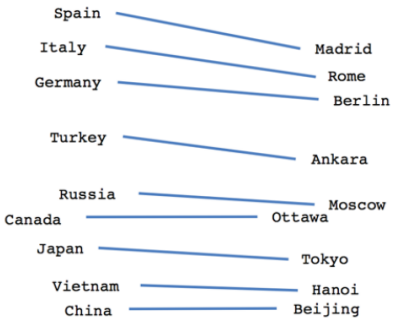



Male-Female

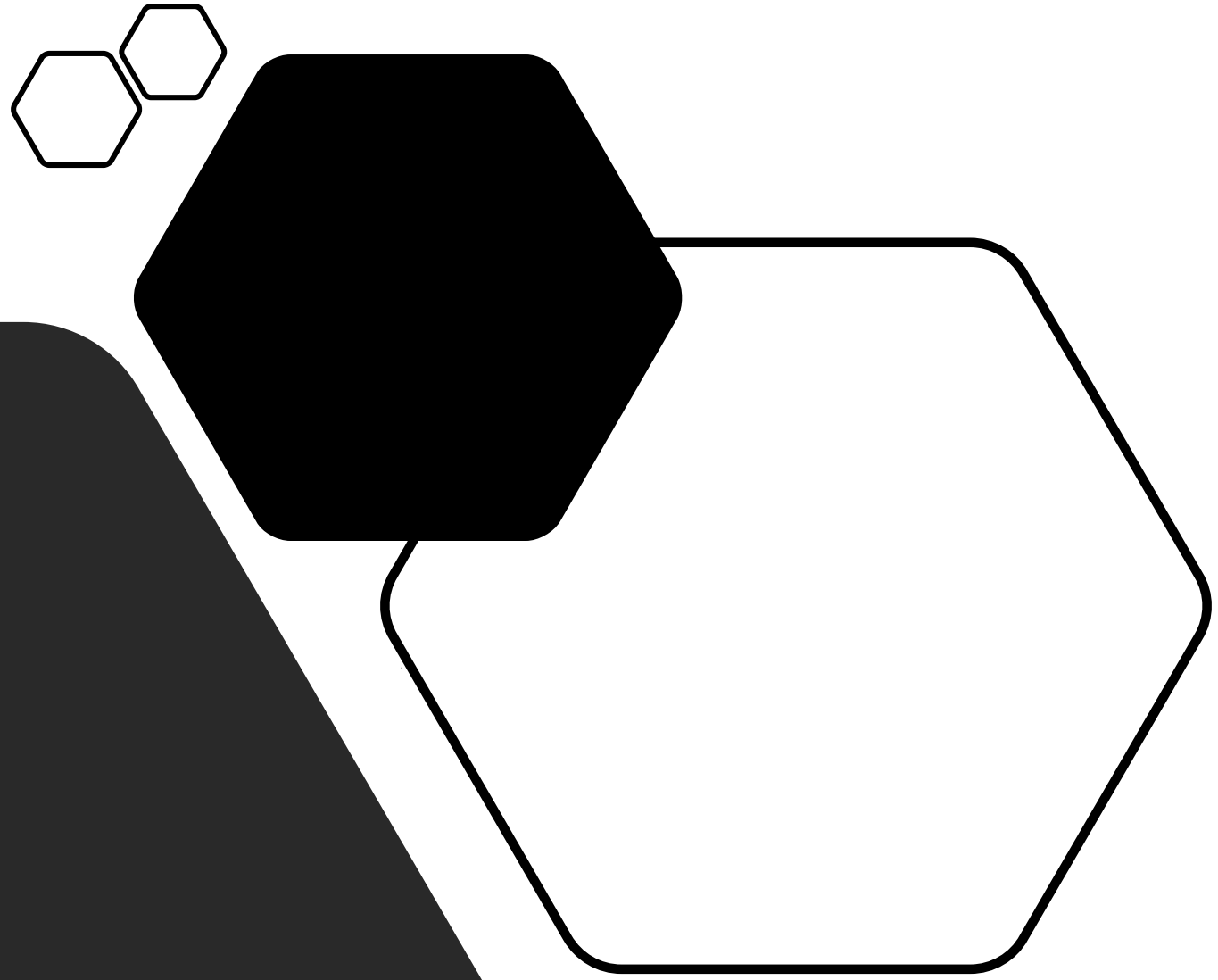



Verb tense



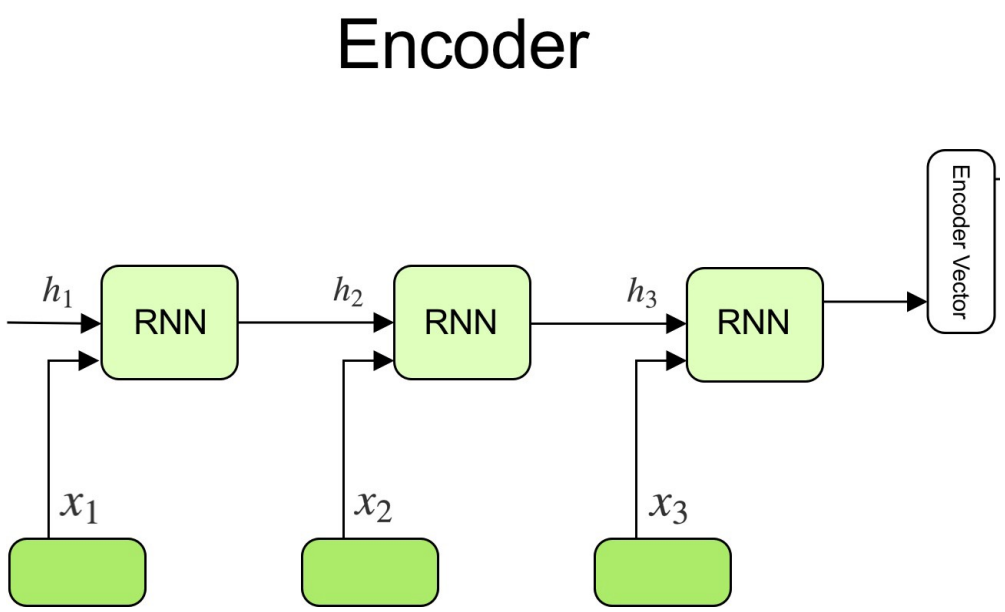

Country-Capital

# Sequence Learning problems



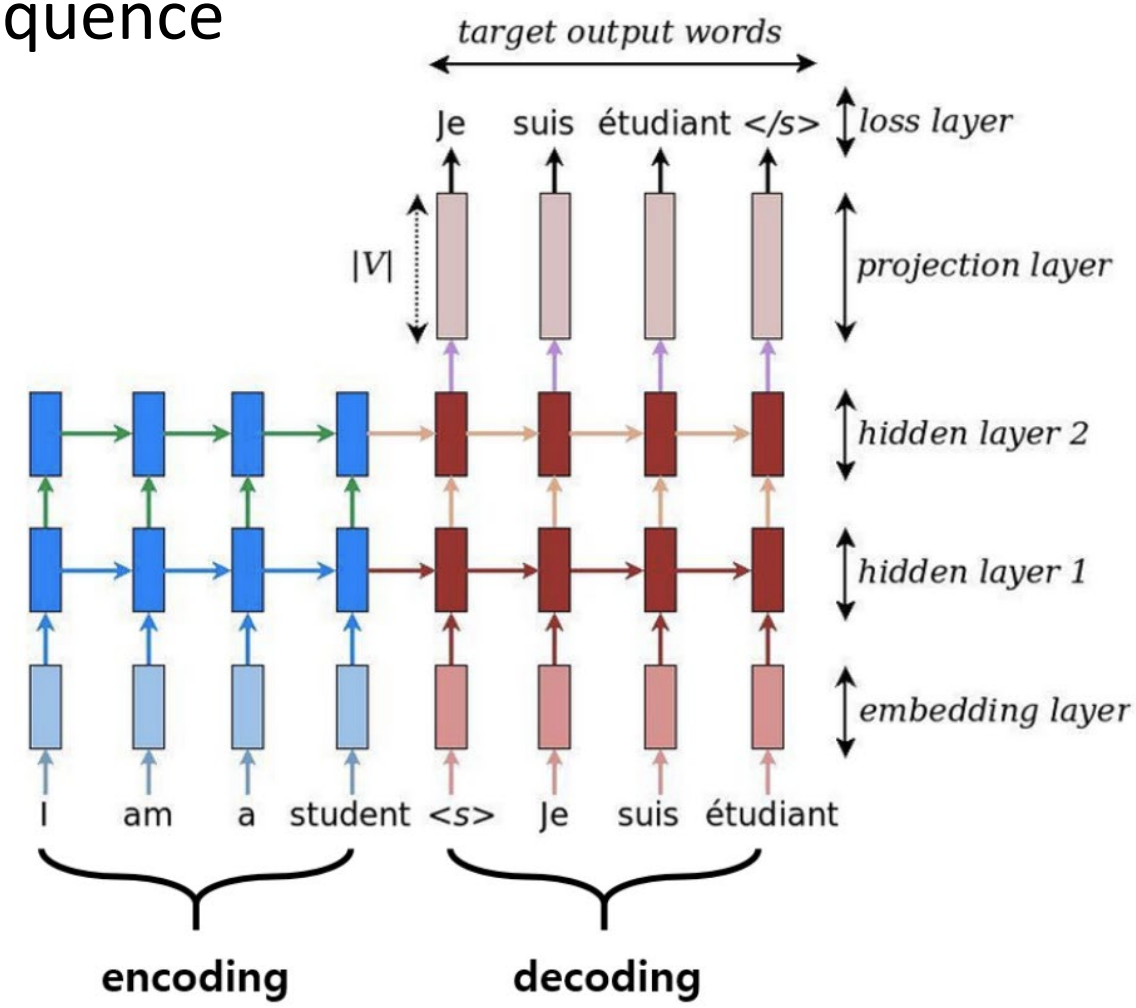
# Sequence encoding

- Encode a sequence into a vector of representation
- Example: sentiment analysis



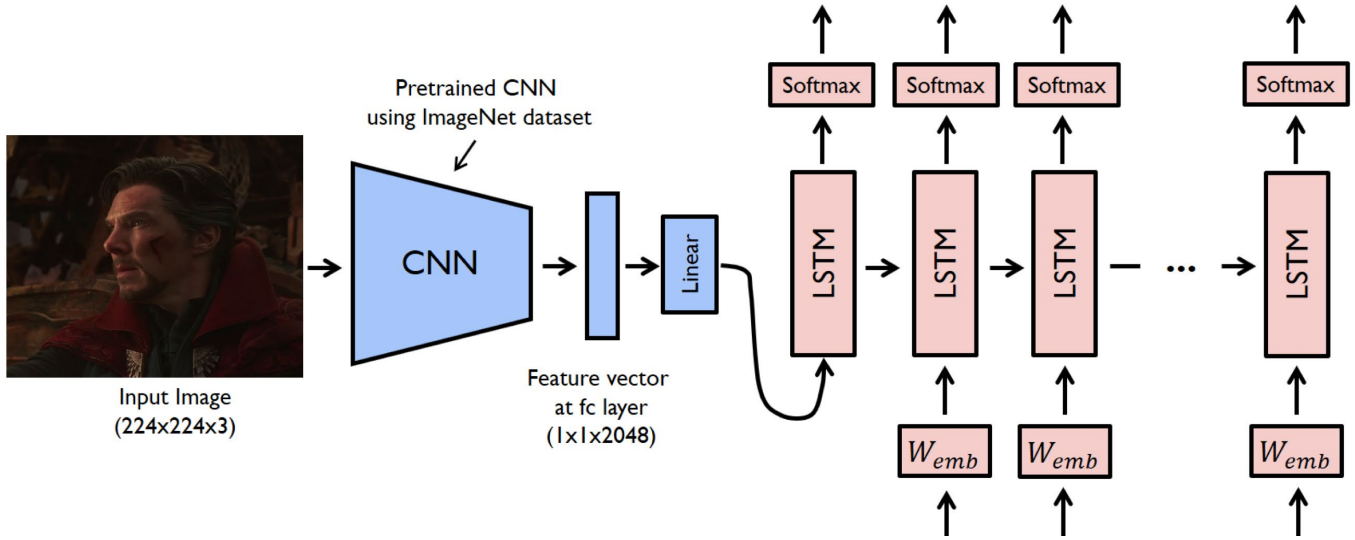
# Encoder-decoder architectures

- Encode a sequence, decode a sequence
- Example: translation



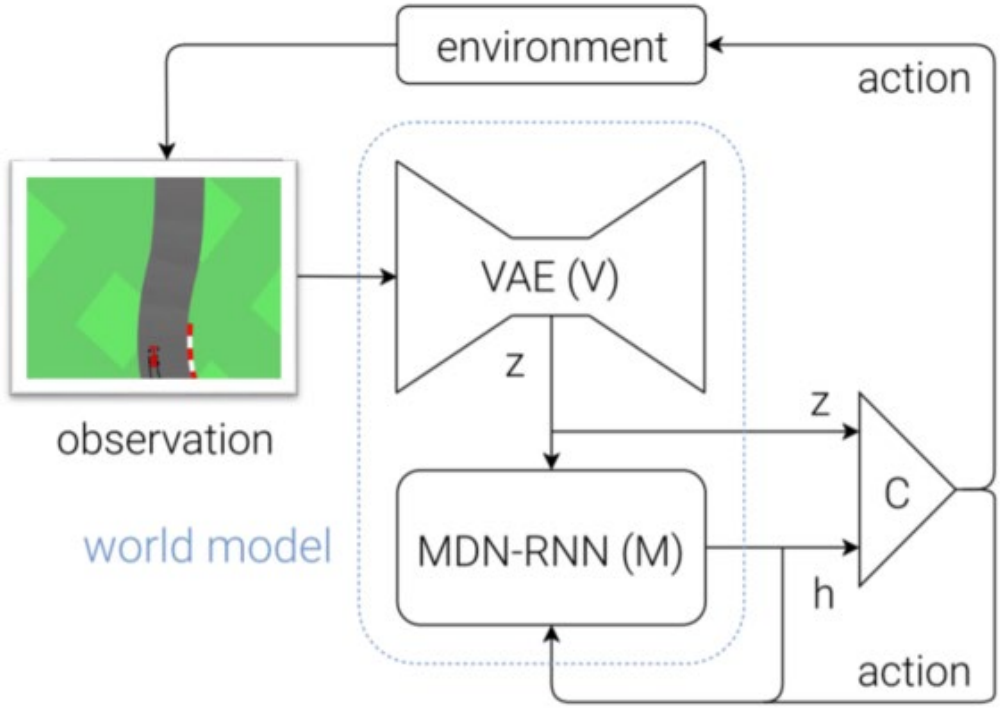
# Sequence generation

- Example: Image captioning



# State encoding

- An agent moves around and tries to solve tasks
- Sequence of past observations is much more informative than still image
- We can learn the state of an agent



# Lab 10 – 2

# Lab 10 – part 2

- IMDB sentiment classification
- 25000 training points / 25000 testing points
- Sentences in English associated with good/bad review
- (I will not provide solutions for this one)