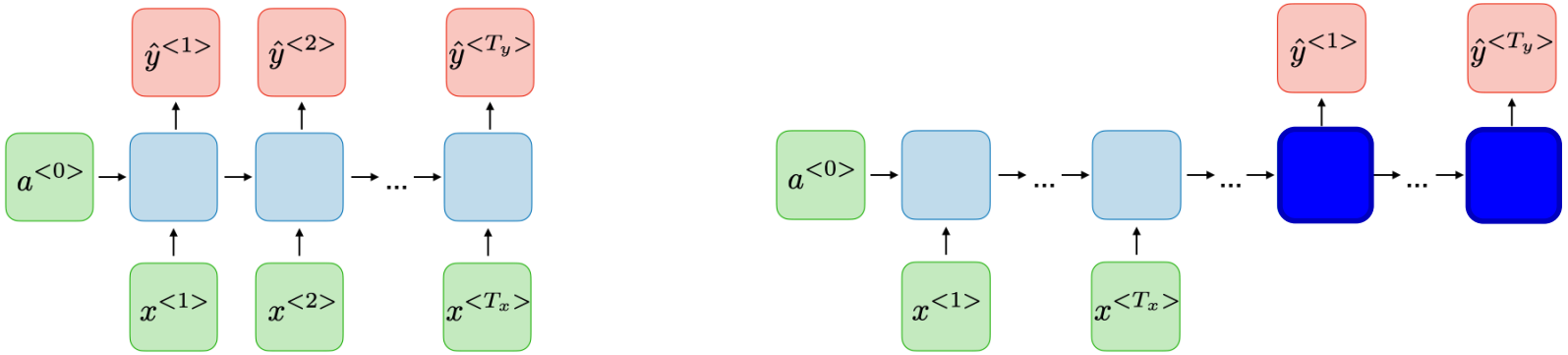
CPSC 440: Machine Learning

Long Short Term Memory

Winter 2022

Last Time: Recurrent Neural Networks (RNNs)

- We discussed recurrent neural neworks (RNNs):
  - We considered sequence labeling and sequence-to-sequence variants.
    - Many other variations exist (bi-directional, deep, many-to-one, one-to-many).

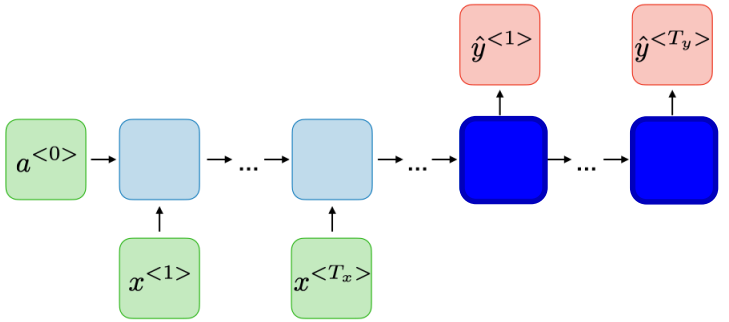


  - Use parameter tieing across time (same parameter repeated).
    - Allows having input/output examples of different lengths.
    - Sequence-to-sequence uses special BOS/EOS symbols.
      - Switches from encoding to decoding, and output can be a different length than input.
    - Can make vanishing/exploding gradient problems worse.
      - Often trained with gradient clipping or Adam.

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Discussion: Sequence-to-Sequence Models

- Representing input and outputs:
  - Could use lexicographic of word2vec representations.
  - Could just have a single character at each time.
    - Could make more sense for some languages.
    - May be able to better handle slang or typos.



- Loss function assuming independent labels given hidden states:

$$f(\Omega) = -\sum_{i=1}^{n}\sum_{j=1}^{|y^{(i)}|} \log p(y_j^{(i)} | x_{1:t}^{(i)}, \Omega)$$

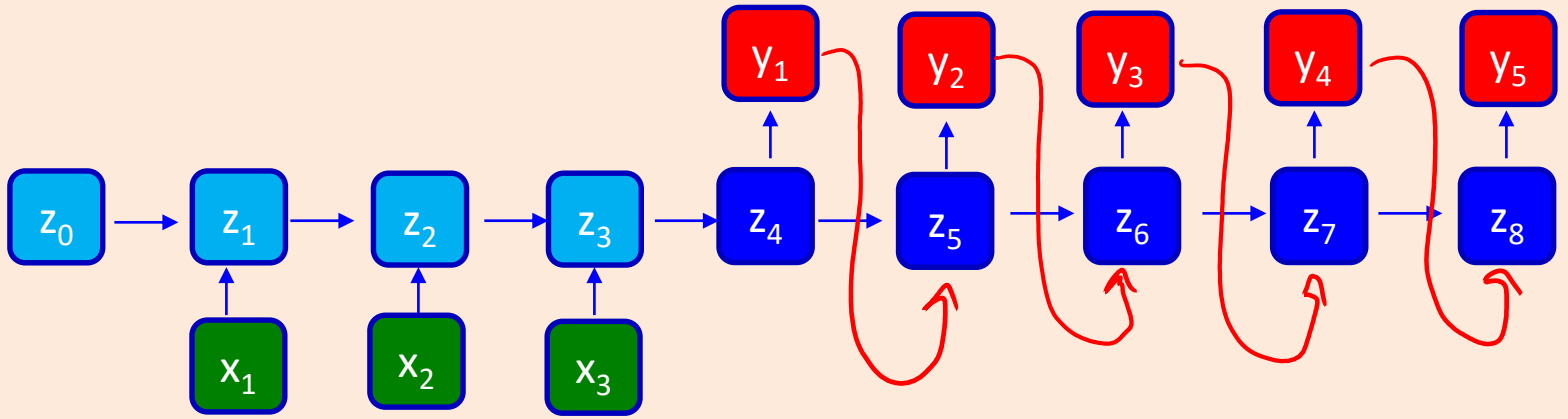
$\{W_e, V_e, U_e, W_d, V_d, U_d\}$ "all parameters"

Softmax value for word at position 'j' in training data.

  - Not that this is just trying to get the label right at each "time".
    - It is not explicitly "trying to get the full sequence right".

Digression/Preview: Dependent Predictions

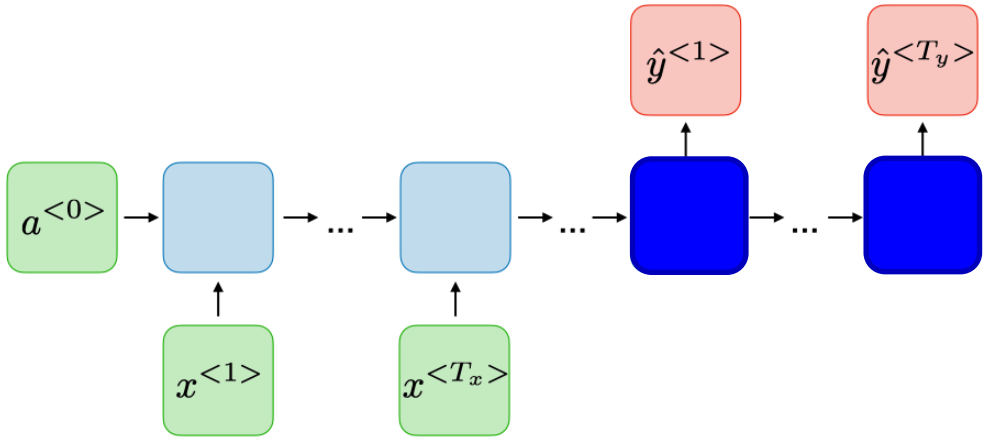
- Standard RNNs assume conditional independence of $\hat{y}_t$ values.
  - We assume they are independent given the z_t values (make inference easy).
  - This makes inference easy, but $\hat{y}_t$ "forgets" what was used for $\hat{y}_{t-1}$.
- In many applications, you want to model dependencies in the $\hat{y}_t$.
  - A common way to do this is to add edges like this:



  - This does not complicate training (where we know the y_t values).
  - But it makes inference and decoding challenging since the y_t are dependent.
    - We will discuss variants like this after we have discussed Markov chains.

Exponential "Forgetting" in RNNs

- Sequence-to-sequence RNNs:
  - Elegant way to handle inputs/outputs of different/unknown sizes.
  - Final "encoding" is the hidden states once the last input has been entered.
    - We hope this captures the semantics of the sentence.
  - The "decoding" steps try use the hidden states to output translation, and also update the hidden states.



- Using tied parameters allows using the model for any sequence lengths.
- But with tied parameters, we "forget" information exponentially fast.
  - If you want to "remember" something about x_1, it has to go through U*U*U*….
    - "Initial conditions" for before the multiplication are forgotten at an exponential speed.

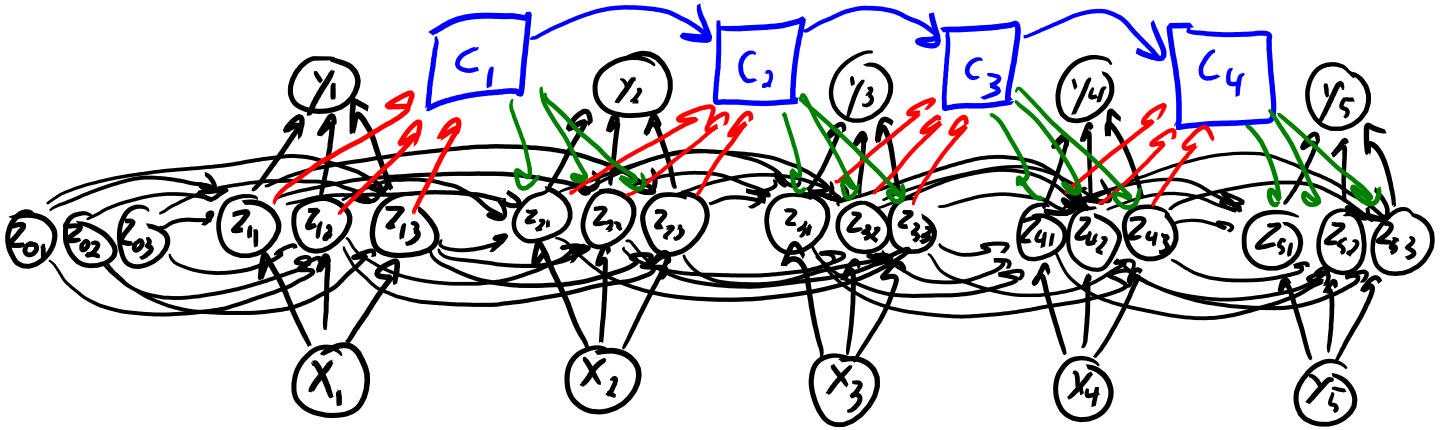
https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Adding a "Memory"

- One possible way to help RNNs remember is with skip connections:

$$\hat{y}_t = V h(z_t) \qquad z_t = W x_t + U_1 h(z_{t-1}) + U_2 h(z_{t-2})$$

  - We will come back to several variations on this idea later.
- Another idea is to add a memory where you can "save" and "load":



- Relevant information can be saved to the memory, then accessed at a much later time.

Long Short Term Memory (LSTM)

- Long short term memory (LSTM) models are variant of RNNs:
  - Modification to try to remember short-term and long-term dependencies.
- In addition to usual hidden values ‘z’, LSTMs have memory cells ‘c’:
  - Purpose of memory cells is to remember things for a long time.
- LSTMs are been the practical analogy of convolutions for RNNs:
  - “The first trick that made them work in many applications.”
- LSTMs have been used in a huge variety of settings:
  - Cursive handwriting recognition:
    - https://www.youtube.com/watch?v=mLxsbWAYIpw
  - Generating “Game of Thrones” text:
    - https://pjreddie.com/darknet/rnns-in-darknet
  - Fake positive/negative Amazon reviews:
    - https://blog.openai.com/unsupervised-sentiment-neuron

Long Short Term Memory – Ugly Equations

- Computing activations at time 't' in an RNN:

$$a_t = h(z_t) \qquad \text{With } z_t = W x_t + U \underbrace{a_{t-1}}_{h(z_{t-1})}$$

- Computing activations at time 't' in an LSTM:

$$a_t = o_t \circ h_o(c_t)$$

"memory"

$$c_t = f_t \circ c_{t-1} + i_t \circ g_t$$

"output gate" → $o_t = h(W_o x_t + U_o a_{t-1})$

"forget gate" → $f_t = h(W_f x_t + U_f a_{t-1})$

"input gate" → $i_t = h(W_i x_t + U_i a_{t-1})$

"value" → $g_t = h_o(W_g x_t + U_g a_{t-1})$

With:

Two activation functions:
"gate" function 'h':
→ usually sigmoid
"value" function 'h_o':
→ usually tanh

"$\circ$": element-wise multiplication of vectors.

Long Short Term Memory – Equation Intuition

- Conceptually, we think of LSTMs as having a "memory" c_t:

- We update and access this memory with a set of "gates":
  - Gates take weighted combination of input and previous activation, and output a value between 0 and 1 (differentiable approximation to binary values).
    - In a computer these gates would be exactly 0 or 1, but we use sigmoids so "gate" can have values like 0.7.

- "Forget gate" f_t:
  - If element 'j' of f_t is 0, then we clear element c_{tj} from the memory (set it to 0).
    - If it is 1, then we keep the old value.
  - "Given the input and previous activation, are the elements in memory still relevant?"
- "Input gate" i_t:
  - If element 'j' of i_t is 0, then we do not add any new information to c_{tj} (no input).
    - If it is 1, then we "value" to the memory (where "value" is also a function of input and previous a_t).
  - "Given the input and previous activation, should I write something new to memory?"
- "Output gate" o_t:
  - If element 'j' of o_t is 0, then we do not read value c_{tj} from the memory (no output).
    - If it is 1, then we load from the memory.
  - "Given the input and previous activation, should I read what is in memory?"

c_t
0.3
-3.5
-0.2
0
0.4
0.3
-0.2

LSTM Equations (same slide as 2 slides ago)

- Computing activations at time 't' in an RNN:

$$a_t = h(z_t) \qquad \text{With } z_t = W x_t + U \underbrace{a_{t-1}}_{h(z_{t-1})}$$

- Computing activations at time 't' in an LSTM:

$$a_t = o_t \circ h_o(c_t)$$

"memory"

$$c_t = f_t \circ c_{t-1} + i_t \circ g_t$$

"output gate" → $o_t = h(W_o x_t + U_o a_{t-1})$

"forget gate" → $f_t = h(W_f x_t + U_f a_{t-1})$

"input gate" → $i_t = h(W_i x_t + U_i a_{t-1})$

"value" → $g_t = h_o(W_g x_t + U_g a_{t-1})$

With:

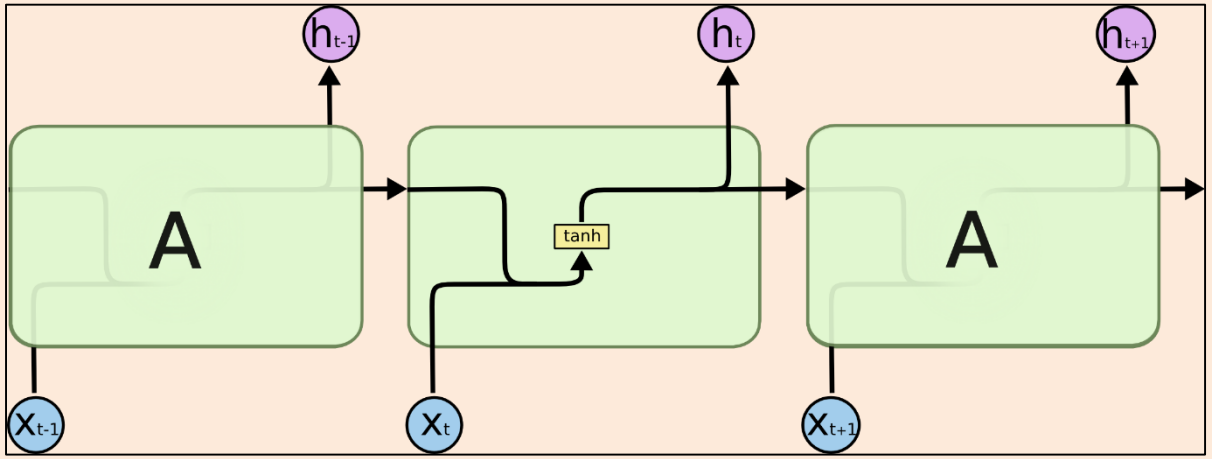
Two activation functions:
"gate" function 'h':
→ usually sigmoid
"value" function 'h_o':
→ usually tanh

"$\circ$": element-wise multiplication of vectors.

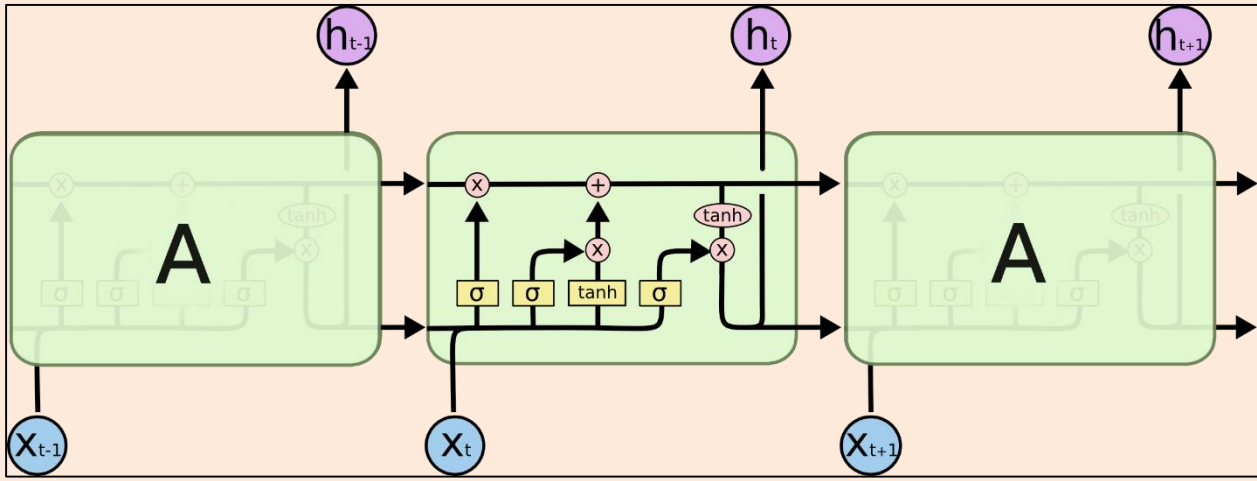
LSTM Activation Calculation as a Picture

- We often see pictures like this to represent the different operations:

RNN



LSTM

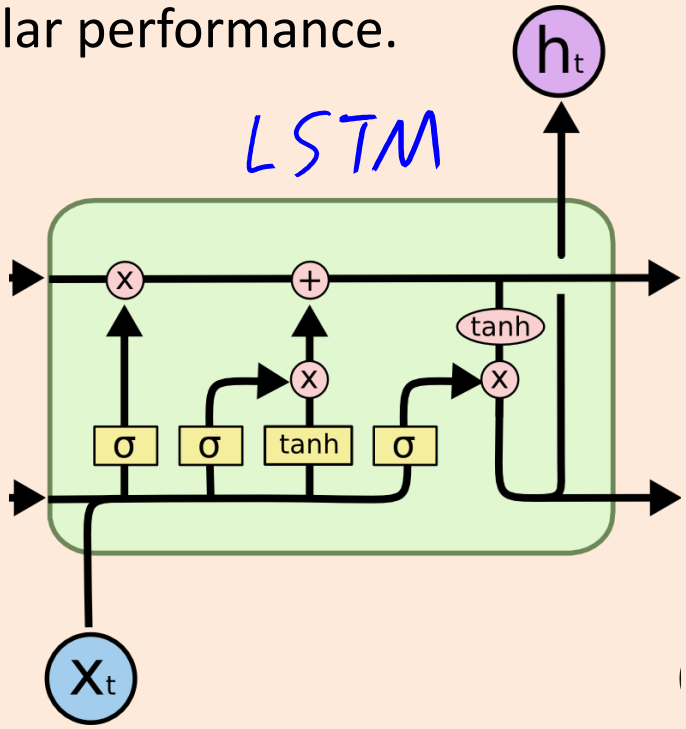


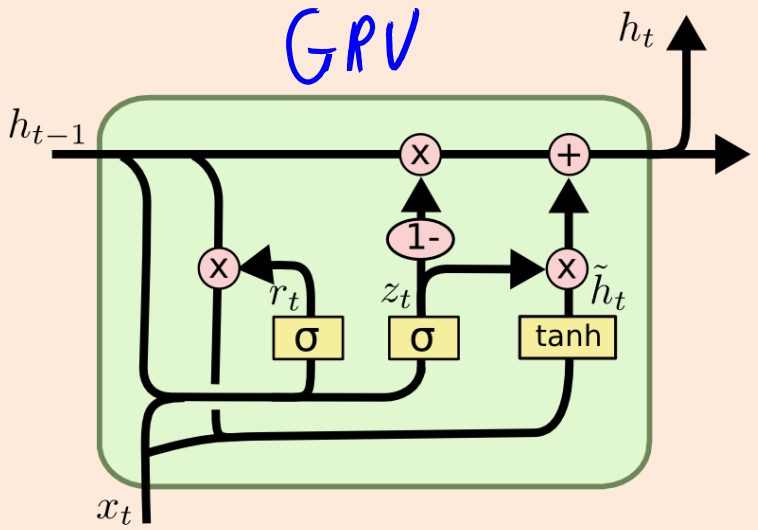
- I find these pictures confusing unless you have gone through equations.
  - For example, where are the weights?

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Units (GRUs)

- Many variations on LSTMs exist.
  - A popular one is gated recurrent units (GRUs).
    - A bit simpler (merges “forget”+”input”, and “activation”+”memory”).
    - Similar performance.

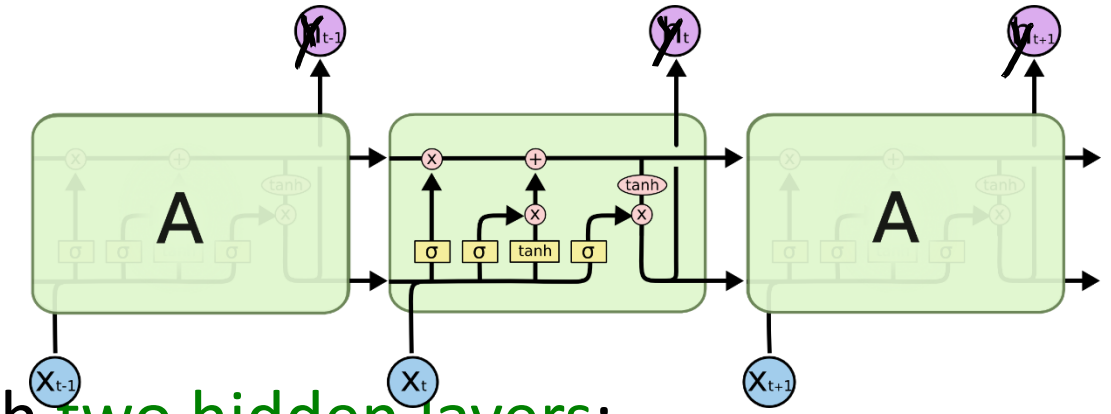




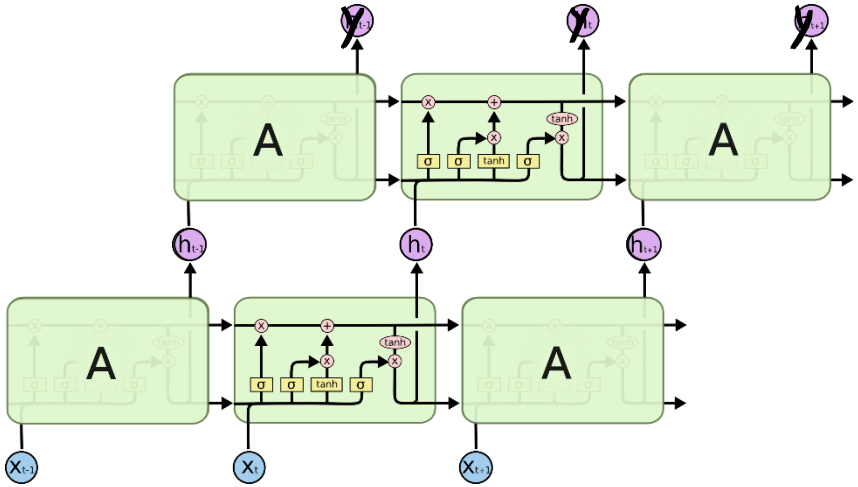
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Deep LSTM Models

- LSTM model with one hidden layer (pixel labeling version):



- LSTM model with two hidden layers:
  - As with regular RNNs, activations feed into next layer and next time.
  - Each layer has own memory.
    - Parameter tieing only within layers.
  - Might have residual connections.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

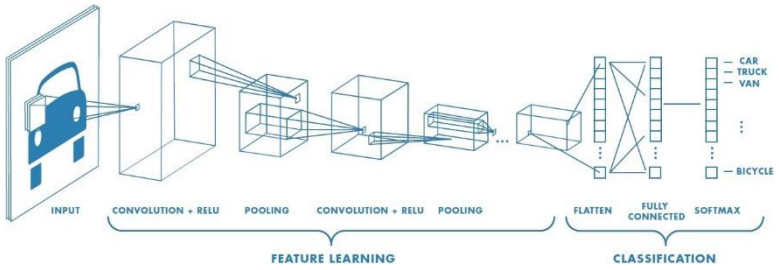
Next Topic: Multi-Modal Models

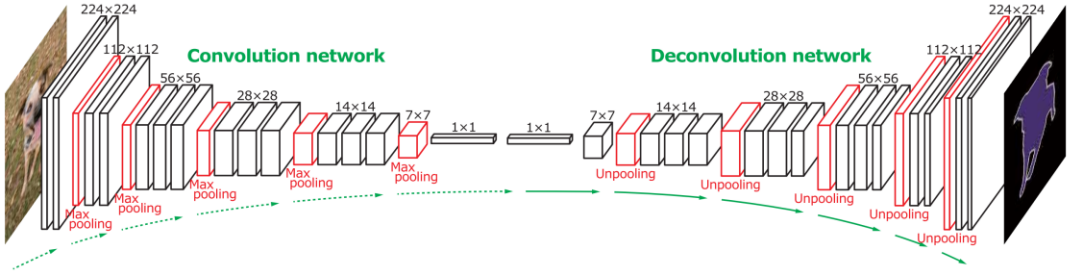
Encoding-Decoding For Different Data Types

- Consider the encoding and decoding phase as separate “models”:

  - Encoder takes a sequence and returns a set of numbers.
  - Decoding takes a set of numbers and outputs a sequence.

- We have also seen encoding and decoding of images:





  - Encoder takes an image and returns a set of numbers.
  - Decoder takes a set of numbers and outputs an image (or a class or set of labels).

https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Noh_Learning_Deconvolution_Network_ICCV_2015_paper.pdf

LSTMs for Image Captioning

- Use a CNN to do the encoding and an RNN to do the decoding.

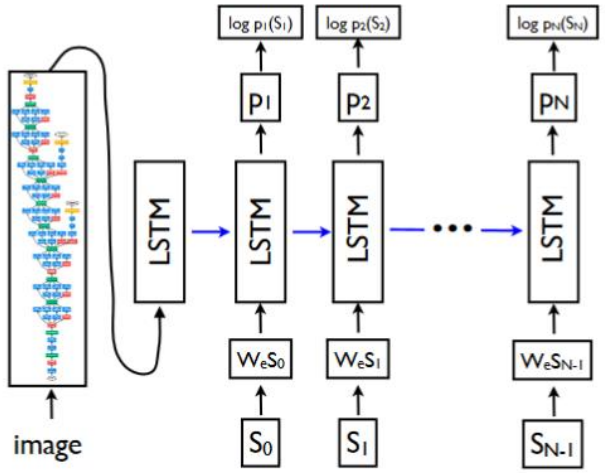


Figure 3. LSTM model combined with a CNN image embedder (as defined in [12]) and word embeddings. The unrolled connections between the LSTM memories are in blue and they correspond to the recurrent connections in Figure 2. All LSTMs share the same parameters.

Figure 5. A selection of evaluation results, grouped by human rating.

- To train this model, we need images and corresponding captions.
  - So the image encoder and sequence decoder are trained together.

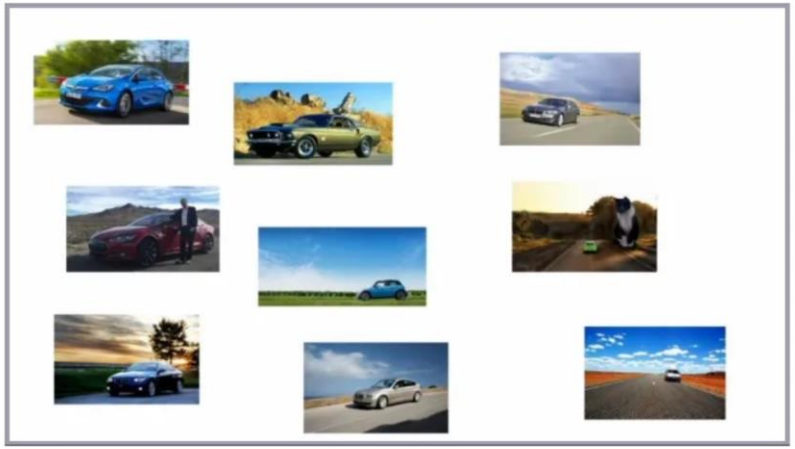
“What do we learn?”

- Sometimes it looks like RNNs are smarter than they actually are.
  - We have specifically picked on CNNs/RNNs, but applies to all ML methods.
  - You should “try to break it”, not just “try to get it to work”.

Easy to get fooled

“a car parked on the side of the road”

Impressive, right? Not so fast, says Efros. “If you go and look for cars on the internet,” he points out, “that description applies to pretty much all of those images.”



“a car parked on the side of the road”

“a car parked on the side of the road”

https://mathwithbaddrawings.com/2017/10/18/5-ways-to-troll-your-neural-network/

Image Captioning Application: PDF to LaTeX

- Use CNN to encode an image, use RNN to decode LaTeX.

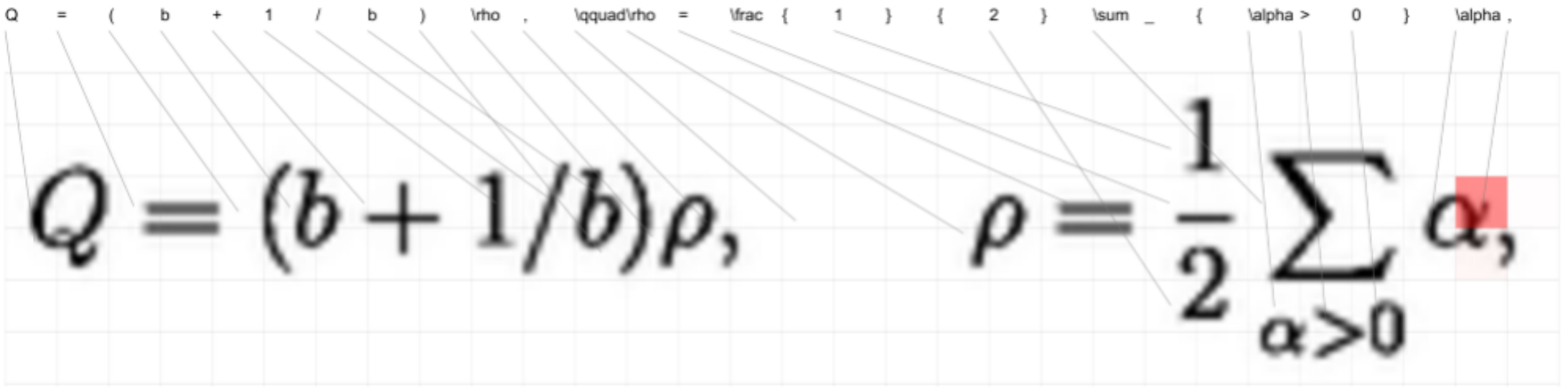
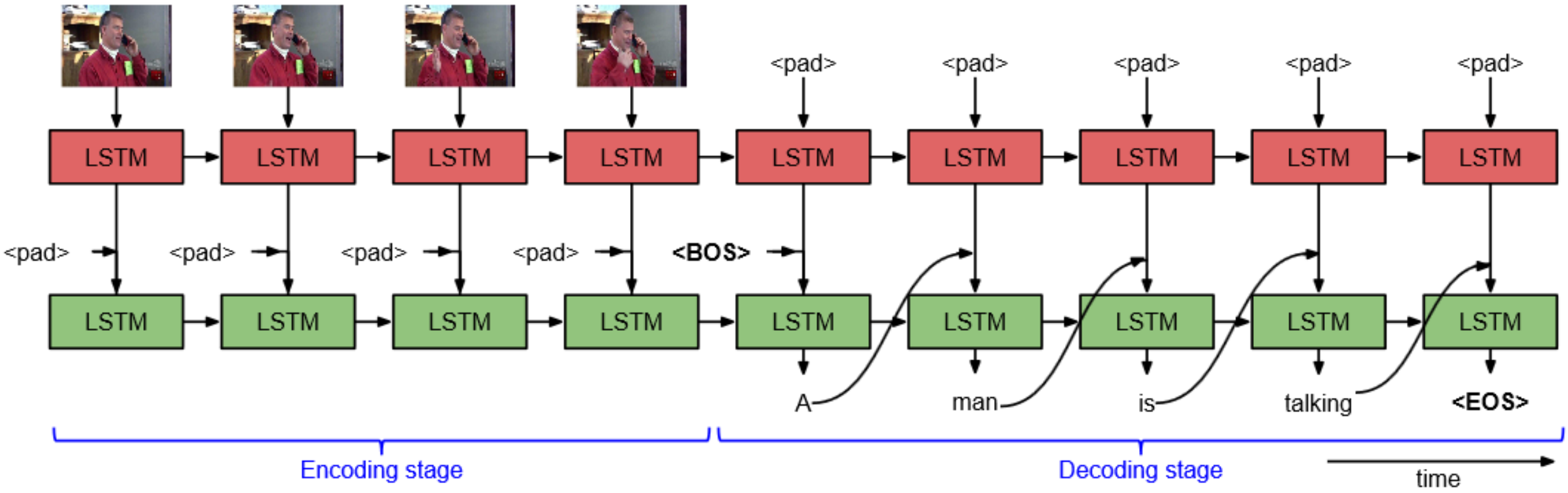


Figure 1: Example of the model generating mathematical markup. The model generates one LaTeX symbol y at a time based on the input image $\mathbf{x}$. The gray lines highlight $H' \times V'$ grid features after the CNN $\mathbf{V}$ and RNN Encoder $\tilde{\mathbf{V}}$. The dotted lines indicate the center of mass of α for each word (only non-structural words are shown). Red cells indicate the relative attention for the last token. See `http://lstm.seas.harvard.edu/latex/` for a complete interactive version of this visualization over the test set.

- Unlike generic image captioning, there is a “correct” label.

LSTMs for Video Captioning



LSTMs for Video Captioning

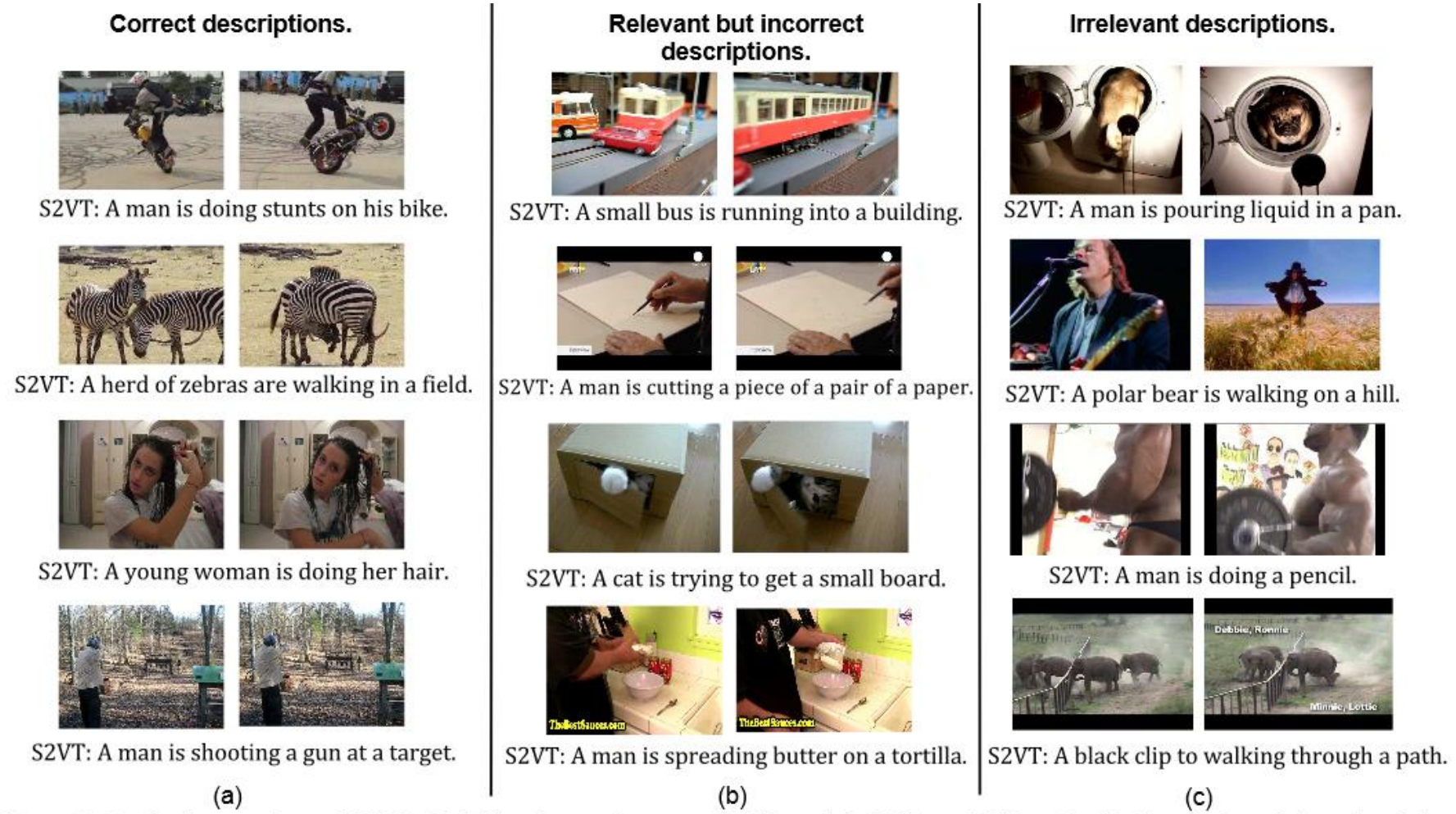


Figure 3. Qualitative results on MSVD YouTube dataset from our S2VT model (RGB on VGG net). (a) Correct descriptions involving different objects and actions for several videos. (b) Relevant but incorrect descriptions. (c) Descriptions that are irrelevant to the event in the video.

http://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Venugopalan_Sequence_to_Sequence_ICCV_2015_paper.pdf

Video Captioning Application: Lip Reading

- Unlike generic video captioning, there is a “correct” label.

https://www.youtube.com/watch?v=5aogzAUPilE

RNNs/CNNs for Poetry

- Generating poetry:

```
And still I saw the Brooklyn stairs
With the shit, the ground, the golden haze
Of the frozen woods where the boat stood.
When I thought of shame and silence,
I was a broken skull;
I was the word which I called it,
And I saw the black sea still,
So long and dreary and true;
The way a square shook out my ground,
And the black things were worth a power,
To find the world in a world of reason,
And I saw how the mind saw me.
```

- Image-to-poetry:

```
A man is sitting on the edge of the waters.
I should see him begin to stand at the throat of the graveyard
and my love is like a stairway in his left arm and a piece of the stairs,
and there is a girl in the doorway and she and I am a good time.
I want to see her the best thing with the footprints in the woods
and the candle shifts back to the shrine and the last late sun
the sky and the candle and the noise of the snow.
```

```
Dropout 0.25, Loss 1.1465, 1:16:1, Railroad
```

A train traveling over a bridge over a river to the end of the street and the sea is a strange street with a cold sun on the street where the sun stands and the sun is still and the sun is still and the sun is gone. The sun is all around me. I am the same as the sun on the street with a strange contract.

A train traveling over a bridge over a river to the graveyard and the barn was a strange street of straw halls and the sun was always sinking in the sun.

I was the one who was still in the street when he was standing in the sun and the sun was still alive.

He was a big smile and I was a child who was a stranger.

- Movie script:
  - https://www.youtube.

https://medium.com/artists-and-machine-intelligence/adventures-in-narrated-reality-6516ff395ba3

Summary

- Standard RNNs lead to exponential forgetting of information.
- Long short term memory:
  - The trick that made RNNs start working.
  - Gating functions which update “memory cells” for long-range interactions.
- Multi-modal learning:
  - Encoder and decoder may work with different types of data.
  - For example, CNN as encoder and RNN as decoder for image-to-text.

- Next time: generating music and dance moves.