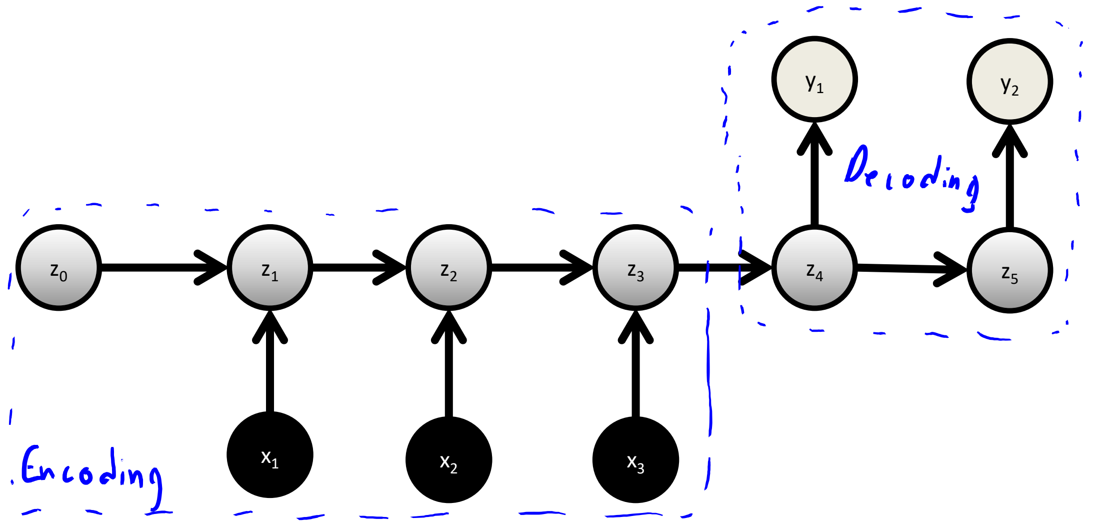
CPSC 540: Machine Learning

Attention

Winter 2020

Previously: Sequence-to-Sequence

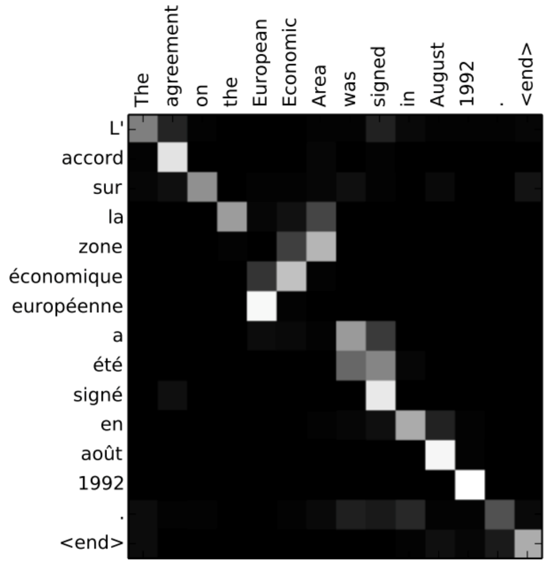
- Sequence-to-sequence:
  - Recurrent neural network for sequences of different lengths.



- Problem:
  - All "encoding" information must be summarized by last state (z_3).
  - Might "forget" earlier parts of sentence.
    - Or middle of sentence if using bi-directional RNN.
  - Might want to "re-focus" on parts of input, depending on decoder state.

Attention

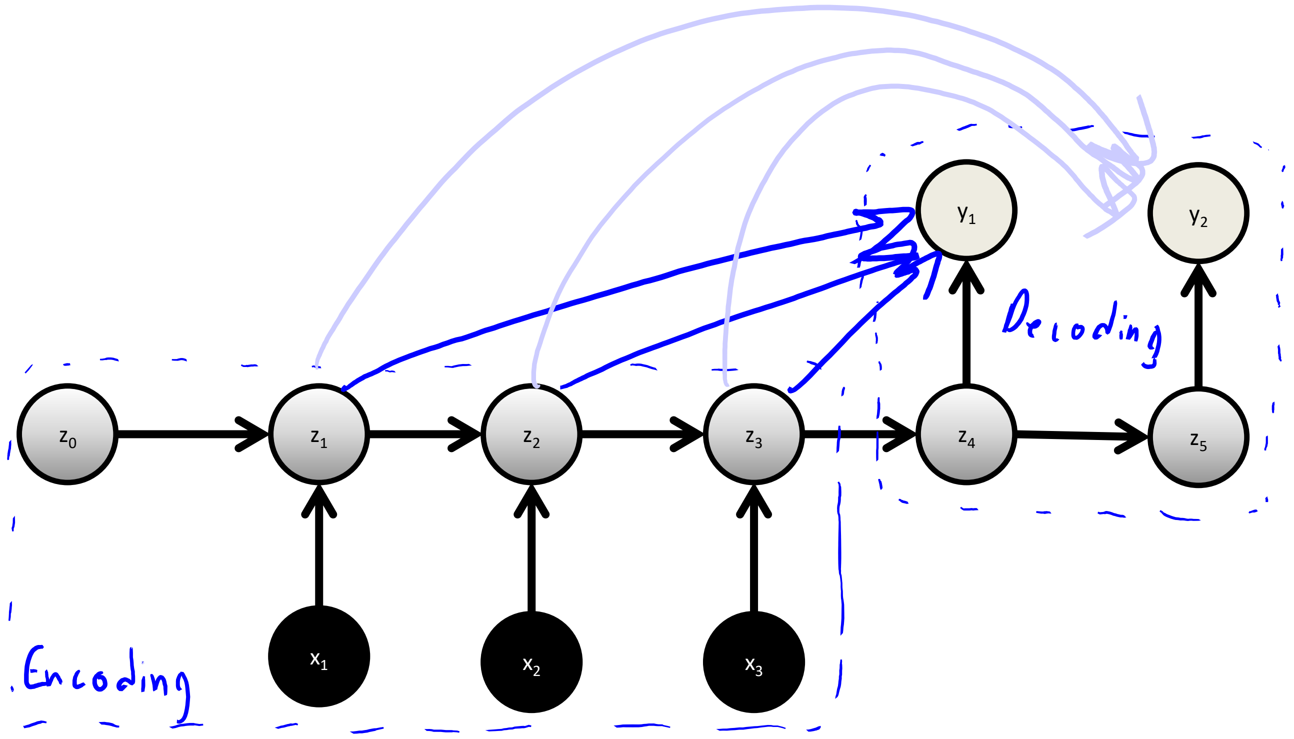
- Many recent systems use "attention" to focus on parts of input.
  - Including "neural machine translation" system of Google Translate.



- Many variations, but usually include the following:
  - Each decoding can use hidden state from each encoding step.
    - Used to re-weight during decoding to emphasize important parts.

Not-Very-Practial Attention

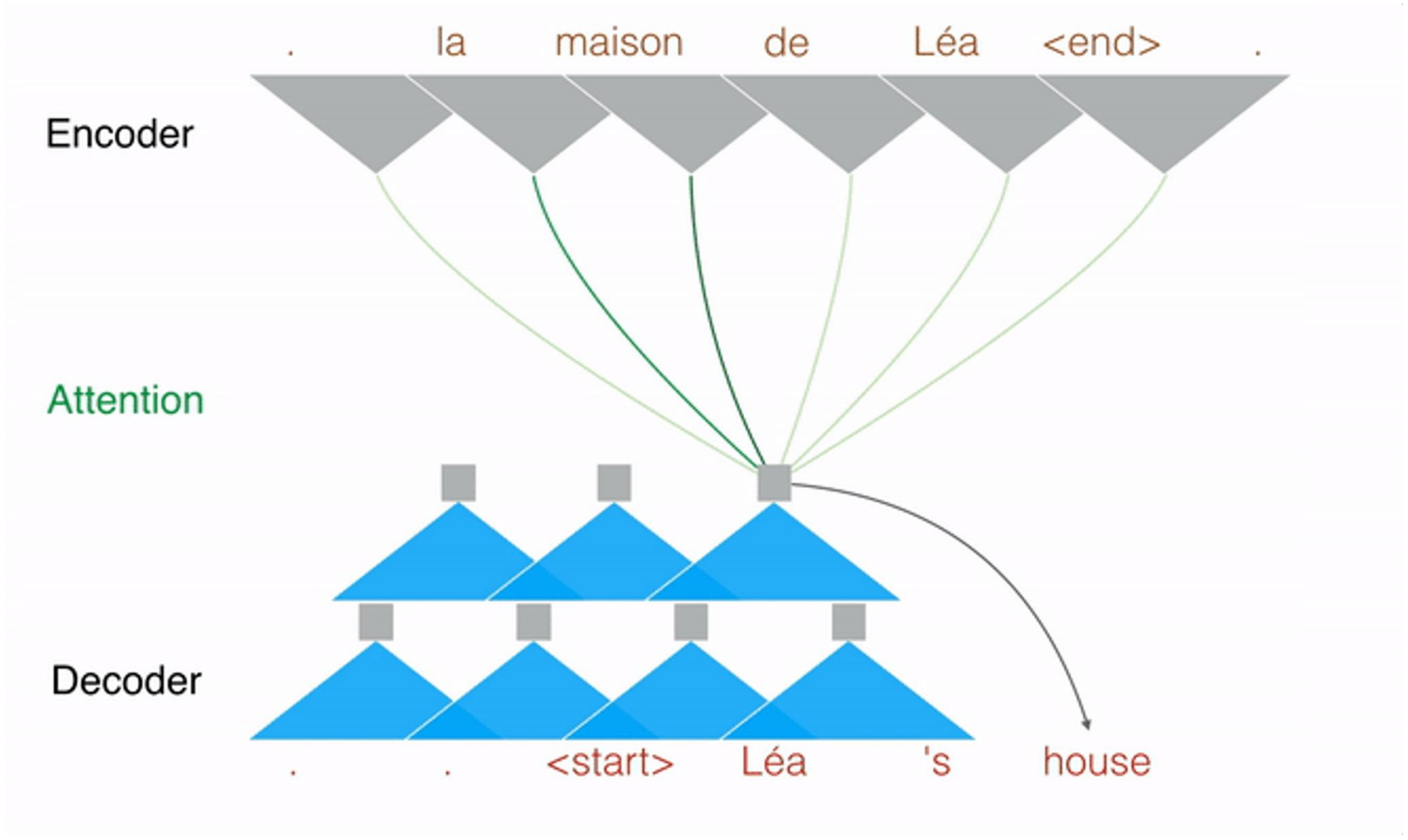
- A naïve “attention” method (no one uses this, but idea is similar):
  - At each decoding step, use decoder state (as usual) and all encoder states.



  - Another variation on the “residual connection” or “denseNet” trick.
  - But this variant is not practical since number of decoding weights depends on input size.
    - Practical variations try to summarize encoder information through a “context vector”.

Attention

- Attention for language translation:



https://code.facebook.com/posts/1978007565818999/a-novel-approach-to-neural-machine-translation/

More-Practical Attention

- A common way to generate the context vector:
  - Take current decoder state.
  - Compute inner product with each encoder state.
    - Gives a scalar for each encoding "time".
  - Pass these scalars through the softmax function.
    - Gives a probability for each time (can be shown in pairwise tables).
  - Multiply each encoder state by probability, add them up.
    - Gives fixed-length "context vector".

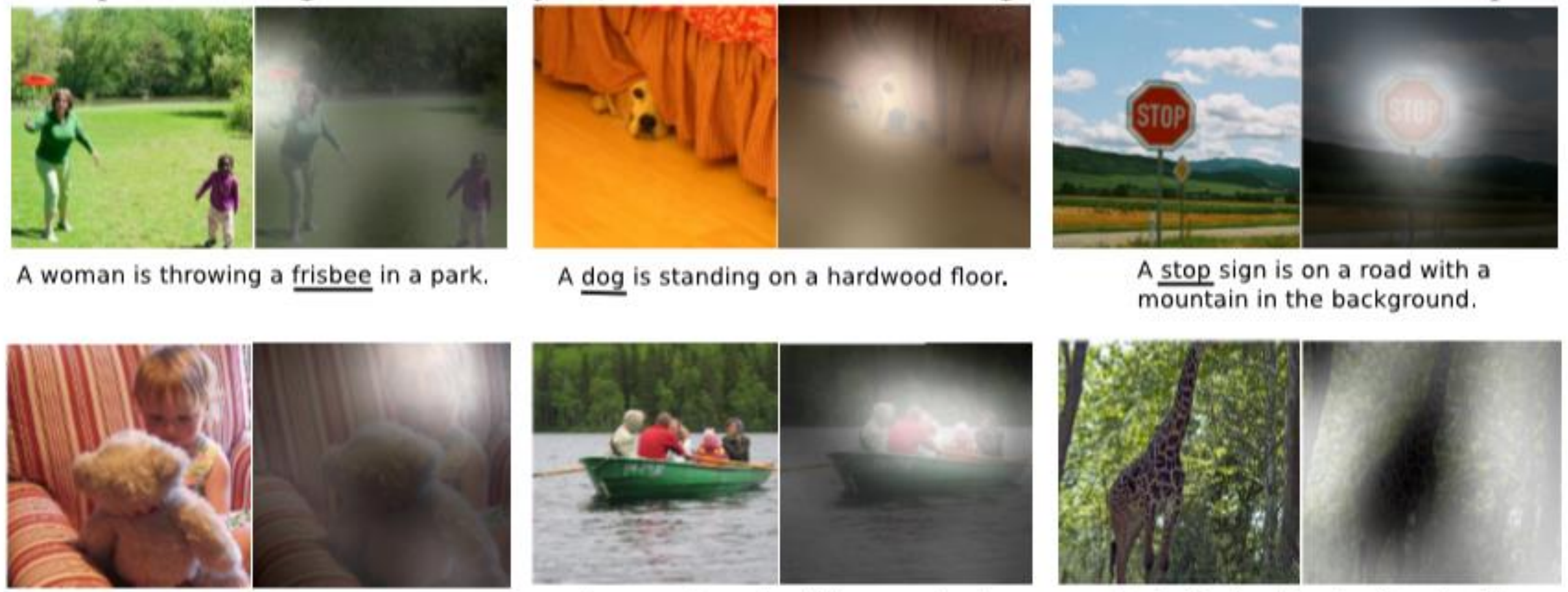
- Context vector is usually appended to decoder's current state.
  - Output could be generated directly from this, or passed through a neural net.
  - Common variation is "multi-headed attention": can get scores from different aspects.
    - Semantics, grammar, tense, and so on.

- Remember that this is being trained end-to-end.

Attention

- Attention for image captioning:

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

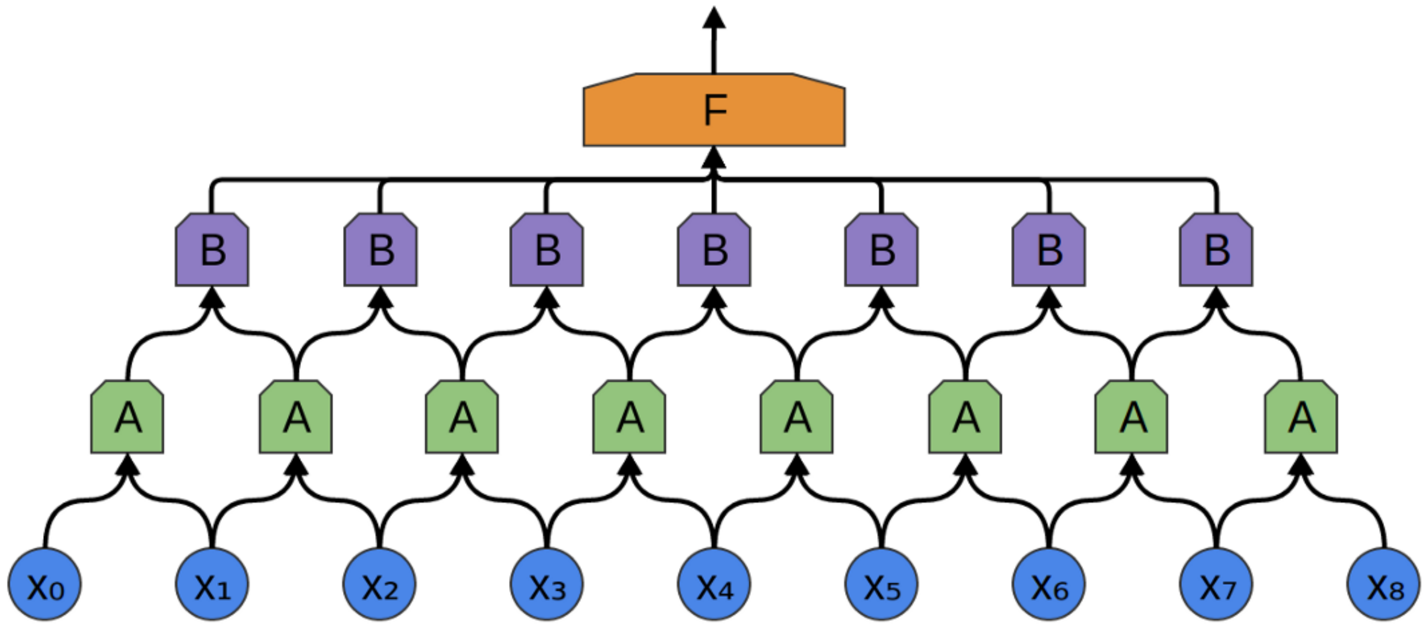
A giraffe standing in a forest with trees in the background.

https://arxiv.org/pdf/1502.03044.pdf

Convolutions for Sequences?

- Should we really be going through a sequence sequentially?
  - What if stuff in the middle is really important, and changes meaning?

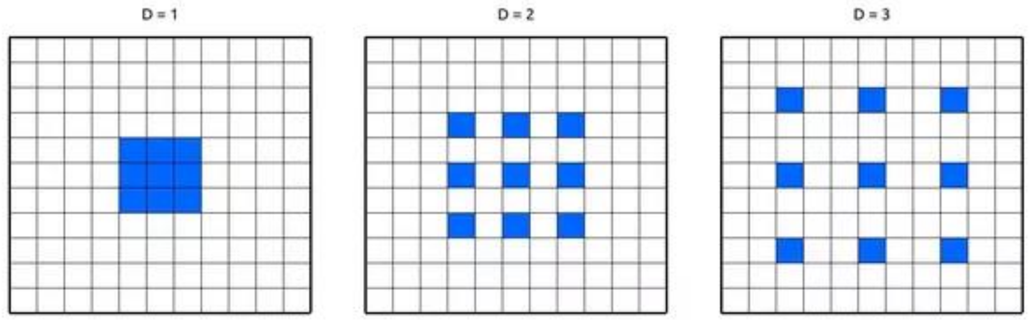
- Recent works have explored using convolutions for sequences.

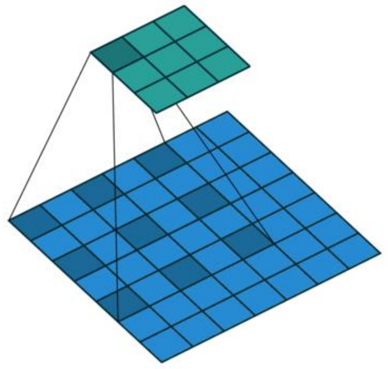


https://medium.com/@TalPerry/convolutional-methods-for-text-d5260fd5675f

Digression: Dilated Convolutions (“a trous”)

- Best CNN systems have gradually reduced convolutions sizes.
  - Many modern architectures use 3x3 convolutions, far fewer parameters!
- Sequences of convolutions take into account larger neighbourhood.
  - 3x3 convolution followed by another gives a 5x5 neighbourhood.
  - But need many layers to cover a large area.
- Alternative recent strategy is dilated convolutions (“a trous”).



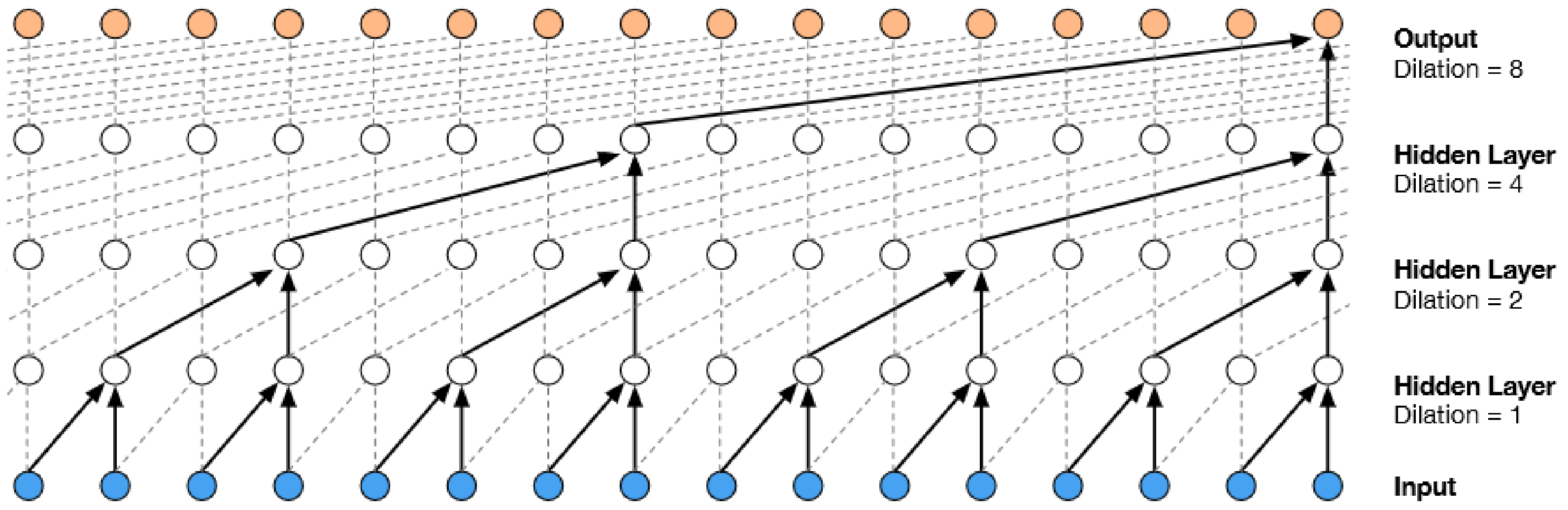


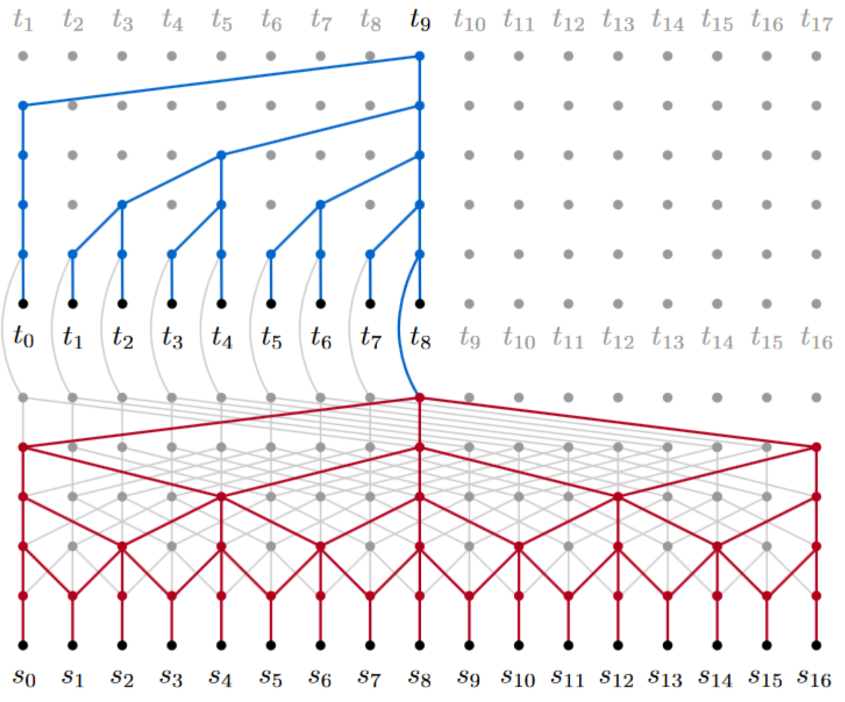
- Not the same as “stride” in a CNN:
  - Doing a 3x3 convolution at all locations, but using pixels that are not adjacent.
  - During upsampling, you can use interpolation to fill the holes.

https://medium.com/@TalPerry/convolutional-methods-for-text-d5260fd5675f
https://github.com/vdumoulin/conv_arithmetic

Dilated Convolutions ("a trous")

- Modeling music and language and with dilated convolutions:





t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11} t_{12} t_{13} t_{14} t_{15} t_{16} t_{17}
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11} t_{12} t_{13} t_{14} t_{15} t_{16}
s_0 s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10} s_{11} s_{12} s_{13} s_{14} s_{15} s_{16}

Figure 1. The architecture of the ByteNet. The target decoder (blue) is stacked on top of the source encoder (red). The decoder generates the variable-length target sequence using dynamic unfolding.

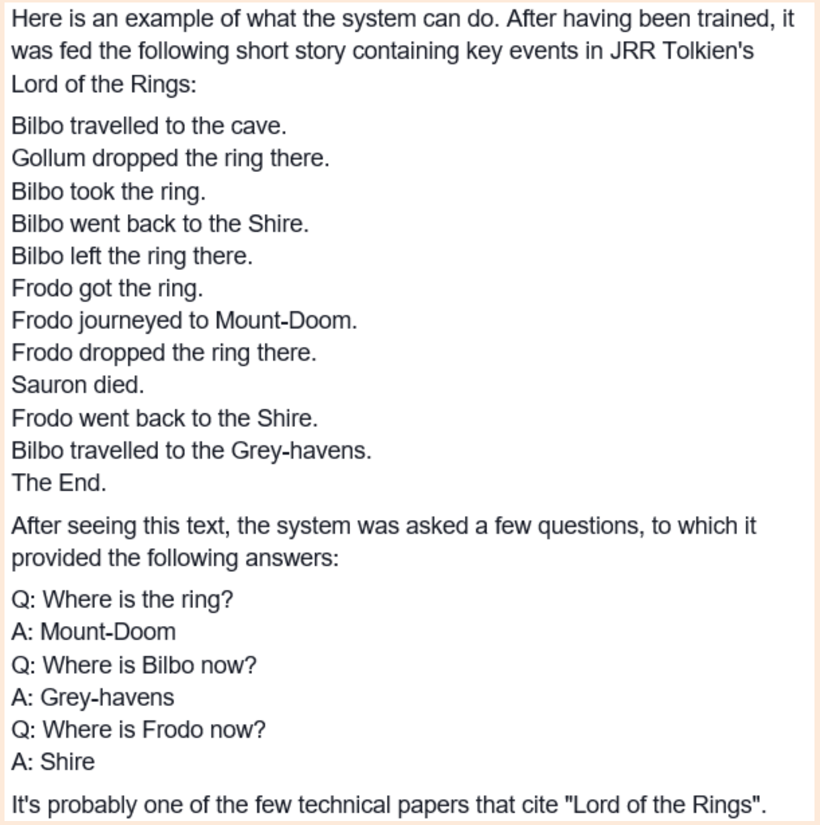
https://arxiv.org/pdf/1610.10099.pdf
https://arxiv.org/pdf/1609.03499.pdf

RNNs/CNNs for Music and Dance

- Music generation:
  - https://www.youtube.com/watch?v=RaO4HpM07hE
- Text to speech and music waveform generation:
  - https://deepmind.com/blog/wavenet-generative-model-raw-audio
- Dance choreography:
  - http://theluluartgroup.com/work/generative-choreography-using-deep-learning
- Music composition:
  - https://www.facebook.com/yann.lecun/videos/10154941390687143

Neural Turing/Programmers

- Many interesting recent variations on memory/attention.
  - A good place to start is here: https://distill.pub/2016/augmented-rnns



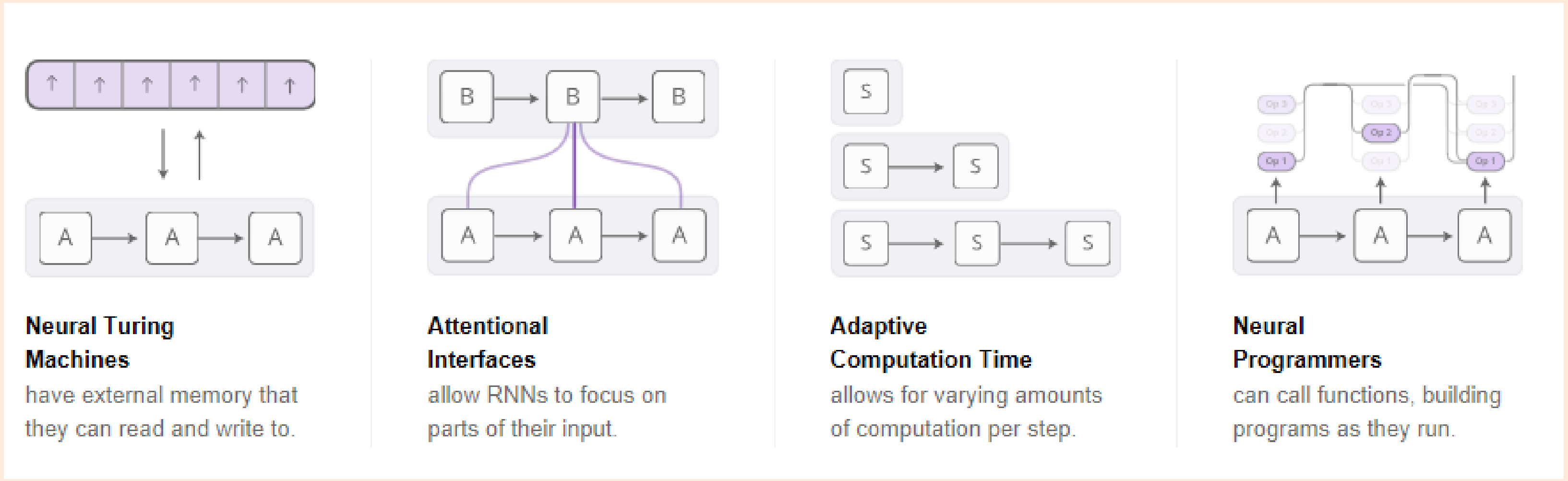
Here is an example of what the system can do. After having been trained, it was fed the following short story containing key events in JRR Tolkien's Lord of the Rings:

Bilbo travelled to the cave.
Gollum dropped the ring there.
Bilbo took the ring.
Bilbo went back to the Shire.
Bilbo left the ring there.
Frodo got the ring.
Frodo journeyed to Mount-Doom.
Frodo dropped the ring there.
Sauron died.
Frodo went back to the Shire.
Bilbo travelled to the Grey-havens.
The End.

After seeing this text, the system was asked a few questions, to which it provided the following answers:

Q: Where is the ring?
A: Mount-Doom
Q: Where is Bilbo now?
A: Grey-havens
Q: Where is Frodo now?
A: Shire

It's probably one of the few technical papers that cite "Lord of the Rings".


A
B
S
Op 1
Op 2
Op 3
Neural Turing Machines have external memory that they can read and write to.
Attentional Interfaces allow RNNs to focus on parts of their input.
Adaptive Computation Time allows for varying amounts of computation per step.
Neural Programmers can call functions, building programs as they run.

https://www.facebook.com/FBAIResearch/posts/362517620591864

(pause)

Remaining Topics

- Major topics we didn’t cover in 340 or 540:
  - Optimization methods (will be covered sometime June, watch the MLRG webpage).
  - Online learning (data coming in over time).
  - Active learning (semi-supervised where you choose examples to label).
  - Causality (distinguishing cause from effect.).
  - Learning theory (VC dimension).
  - Probabilistic context-free grammars (recursive version of Markov chains).
  - Relational models (“object oriented” graphical models).
  - Sub-modularity (discrete version of convexity).
  - Spectral methods (consistent HMM parameter estimation).

- The biggest topic we didn’t cover is probably reinforcement learning:
  - Read Sutton ad Barto’s “Introduction to Reinforcement Learning”.
  - You can also take EECE 592 or Michiel van de Panne’s graphics course.

A Word of Caution

- ML world is really exciting right now, but proceed with caution:
  - ML should still be combined with rigorous testing, sanity checking, and considering misuse cases.
  - “Microsoft deletes ‘teen girl’ AI after it became a Hitler-loving sex robot within 24 hours”:
    - https://www.telegraph.co.uk/technology/2016/03/24/microsofts-teen-girl-ai-turns-into-a-hitler-loving-sex-robot-wit
  - “Amazon AI Designed to Choose Phone Cases Terribly Malfunctions, Fills Store with 31,000+ Hilarious Products:
    - https://www.boredpanda.com/funny-amazon-ai-designed-phone-cases-fail
  - “Uber video shows the kind of crash self-driving cars are made to avoid”:
    - https://www.wired.com/story/uber-self-driving-crash-video-arizona/
  - “One pixel attack for fooling deep neural networks”:
    - https://arxiv.org/abs/1710.08864
  - “Failures of Gradient-Based Deep Learning”:
    - https://arxiv.org/abs/1703.07950
  - “Meaningless Comparisons Lead to False Optimism in Medical Machine Learning”:
    - http://www.arxiv.org/abs/1707.06289
    - https://lukeoakdenrayner.wordpress.com
  - It’s important to get a sense of what can and can’t be done (now and in near-future).
    - Many industry people have unrealistic expectations.

What’s Next?

- “Calling Bullshit in the Age of Big Data”:
  - https://www.youtube.com/playlist?list=PLPnZfvKID1Sje5jWxt-4CSZD7bUI4gSPS
  - There is a lot of bullshit in the machine learning world right now.
    - E.g., cherry-picking of examples in papers and overfitting to test sets.
  - You should try to start recognizing obvious non-sense,
    and not accidently produce non-sense yourself!
- I’m putting material from all my courses (“90 Lectures on Machine Learning”) here:
  - https://www.cs.ubc.ca/~schmidtm/Courses/LecturesOnML
  - (I’ll try to keep this up to date and exhaustive.)
- Our Machine Learning Reading Group:
  - http://www.cs.ubc.ca/labs/lci/mlrg
- Thank you for your patience (this course is not easy to organize),
  and good luck with the next steps!