



BST 261: Data Science II

Lecture 14

**Reinforcement Learning,
Attention Models**

**Heather Mattie
Harvard T.H. Chan School of Public Health
Spring 2 2021**



Recipe of the Day!

Blueberry Walnut Baked Brie with Drizzled Honey





Reinforcement Learning

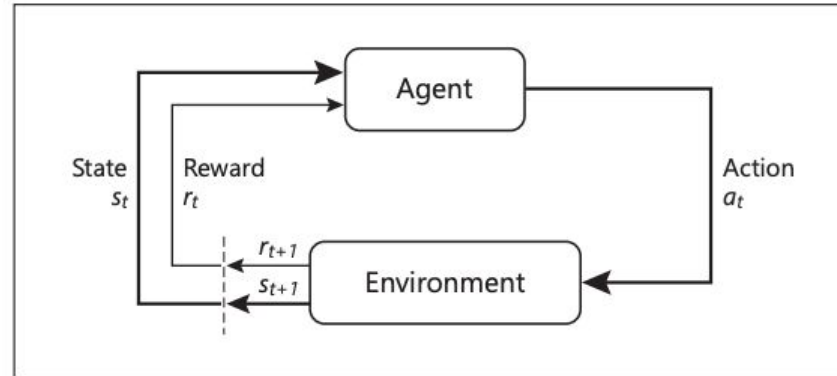
Reinforcement Learning (RL)

- ◎ A subfield of AI that provides tools to optimize **sequences of decisions** for **long-term outcomes**
- ◎ Reinforcement learning is learning what to do—how to map situations to actions—so as to maximize a numerical reward signal
- ◎ The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them
 - Lots of interacting with environment
 - Lots of trial and error
 - A decision will affect not only the next action, but actions after that as well

RL

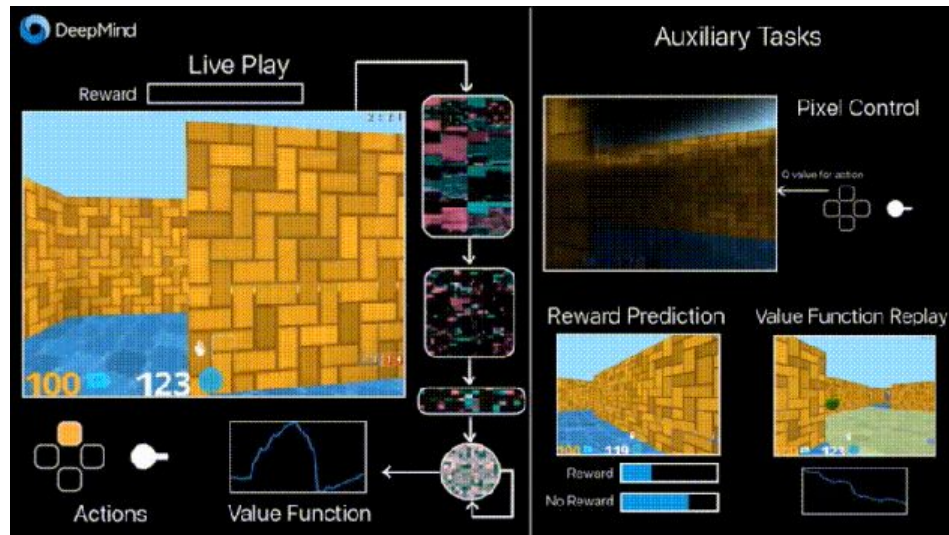
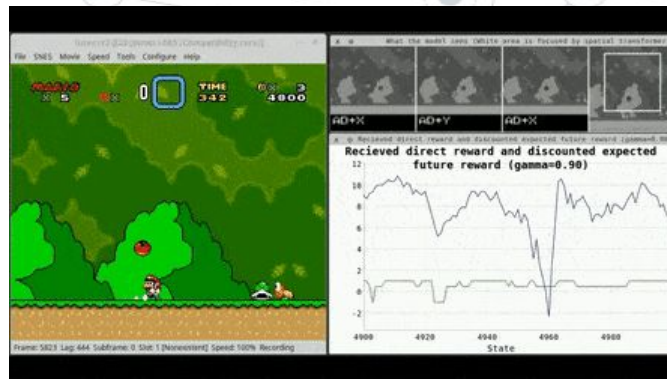
Framework

- Trial and error search, and delayed reward
- Input: sequences of interactions (called **histories**) between the decision maker and their environment
- At every decision point, the RL algorithm chooses an **action** according to its policy and receives new observations and immediate outcomes (often called **rewards**)



RL

- Has been really popular with games
 - AlphaGo is better than the best Go players in the world



RL in Healthcare

- ◎ Still a recent method being applied in healthcare contexts
- ◎ Examples
 - Optimizing antiretroviral therapy in HIV
 - Tailoring antiepilepsy drugs for seizure control
 - Determining the best approach to managing sepsis
- ◎ Rather than a one-time prediction, RL affects a patient's future health and future treatment options
 - Long-term effects are more difficult to estimate

<https://www.nature.com/articles/s41591-018-0310-5>

<https://towardsdatascience.com/a-review-of-recent-reinforcement-learning-applications-to-healthcare-1f8357600407>

Sepsis Example

- ◎ There is wide variability in the way clinicians make decisions about sepsis management
 - Can RL help with this?
- ◎ **History:** may include a patient's vital signs and laboratory tests
- ◎ **Actions:** all the treatments available to the clinician, including medications and interventions
- ◎ **Rewards:** require clinician input - they should represent the achievement of desirable tasks, such as stabilization of vital signs or survival at the end of the stay
 - Short-term: liberation from mechanical ventilation
 - Long-term: prevention of permanent organ damage
 - <https://arxiv.org/pdf/1711.09602.pdf>

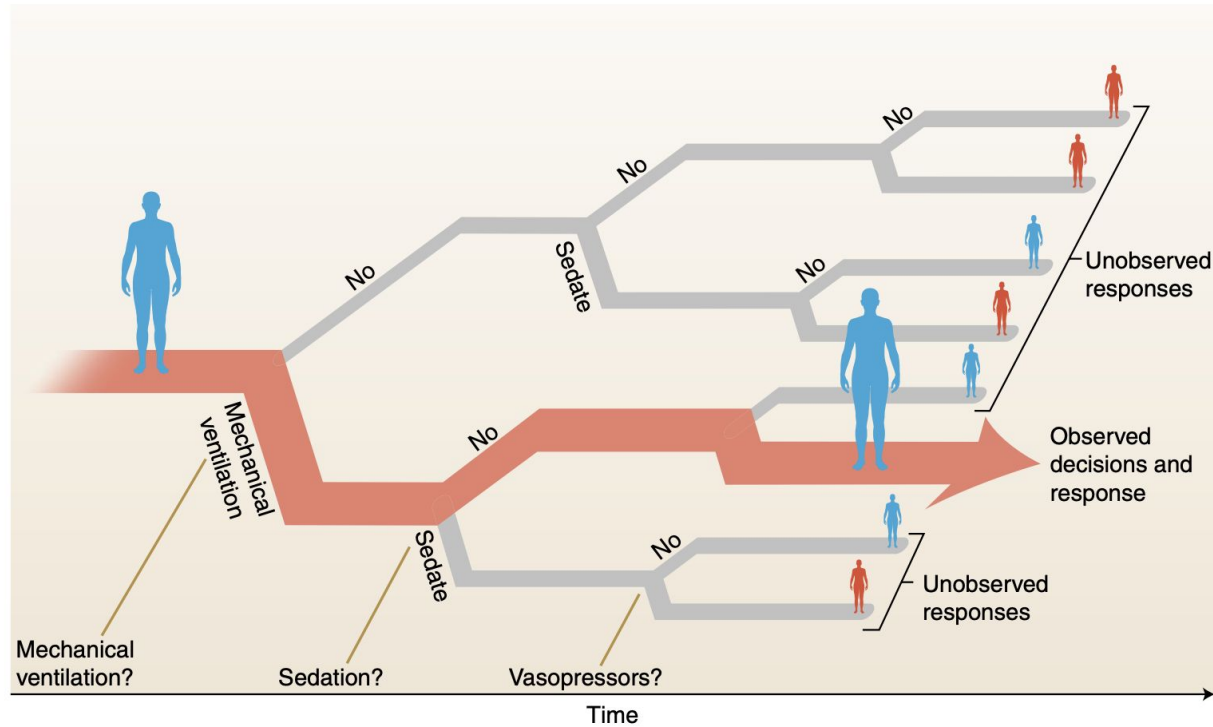


Fig. 1 | Sequential decision-making tasks. To perform sequential decision making, such as for sepsis management, treatment-effect estimation must be solved at a grand scale—every possible combination of interventions could be considered to find an optimal treatment policy. The diagram shows the scale of such a problem with only three distinct decisions. Blue and red people denote positive and negative outcomes, respectively. Credit: Debbie Maizels/Springer Nature

Challenges

- ◎ We only observe one set of actions and rewards for each patient
 - We can't keep trying different combinations of actions to optimize a reward - forced to use previous observational data, called “off-policy” learning
- ◎ We don't observe everything going on in the body
 - We also don't observe the values we do record (blood pressure, etc.) at every time step (dynamic data)
- ◎ It's difficult to find a reward function
 - How do we balance short and long-term rewards?

Need a ton of data, which is difficult to come by

Questions to Consider

- ◎ Is the AI given access to all variables that influence decision making?
 - Typically no because of confounding variables
 - Can lead to confounding in the short term and long term
- ◎ How big is your effective sample size?
 - Most approaches for evaluating RL policies from observational data weigh each patient's history on the basis of whether the clinician decisions match the decisions of the policy proposed by the RL algorithm
 - The reliability (variance) of the treatment-quality estimate depends on the number of patient histories for which the proposed and observed treatment policies agree—a quantity known as the effective sample size
 - The possibilities for mismatch between the actual decision and the proposed decision grow with the number of decisions in the patient's history, and thus RL evaluation is especially prone to having small effective sample sizes

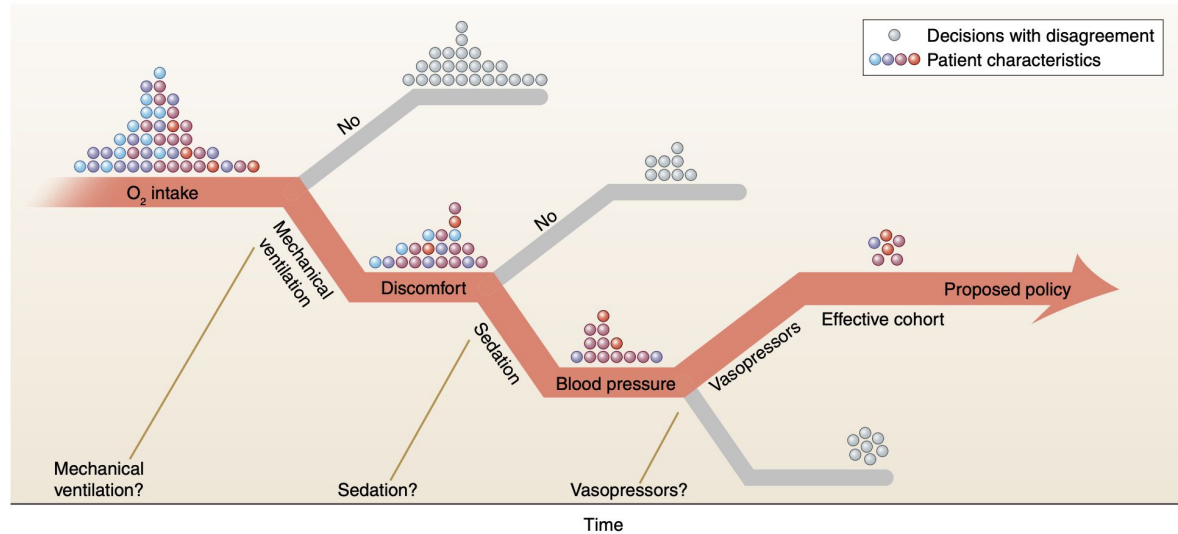


Fig. 2 | Effective sample size in off-policy evaluation. Each dot represents a single patient at each stage of treatment, and its color indicates the patient's characteristics. The more decisions that are performed in sequence, the likelier it is that a new policy disagrees with the one that was learned from. Gray decision points indicate disagreement. Use of only samples for which the old policy agrees with the new results in a small effective sample size and a biased cohort, as illustrated by the difference in color distribution in the original and final cohort. Credit: Debbie Maizels/Springer Nature

Questions to Consider

- ◎ Will the AI behave prospectively as intended?
 - Errors in problem formulation or data processing can lead to poor decisions
 - Simplistic reward functions may neglect long-term effects for meaningless gains: for example, rewarding only blood pressure targets may result in an AI that causes long-term harm by excessive dosing of vasopressors
 - Errors in data recording or preprocessing may introduce errors in the reward signal, misleading the RL algorithm
 - The learned policy may not work well at a different hospital or even in the same hospital a year later if treatment standards shift

RL in Medicine

- ◎ RL in medicine seems promising, but very difficult
- ◎ May help guide clinicians in treatment decisions based on all of a patient's history and not their immediate symptoms/responses to treatment
- ◎ Really nice [review paper of RL in medicine](#)

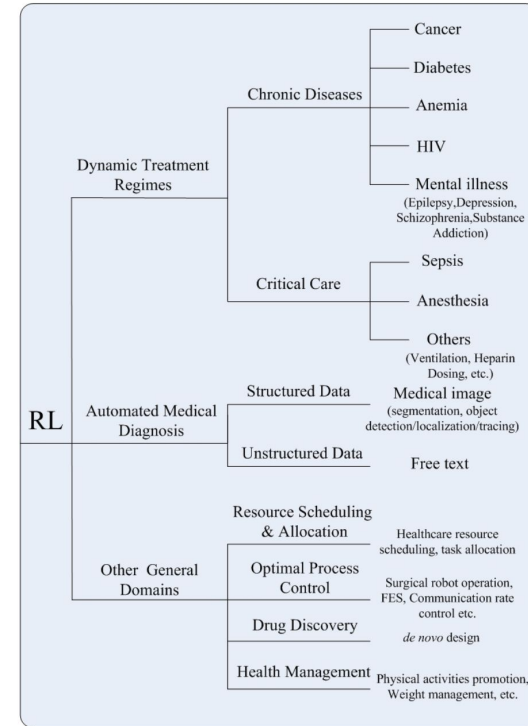
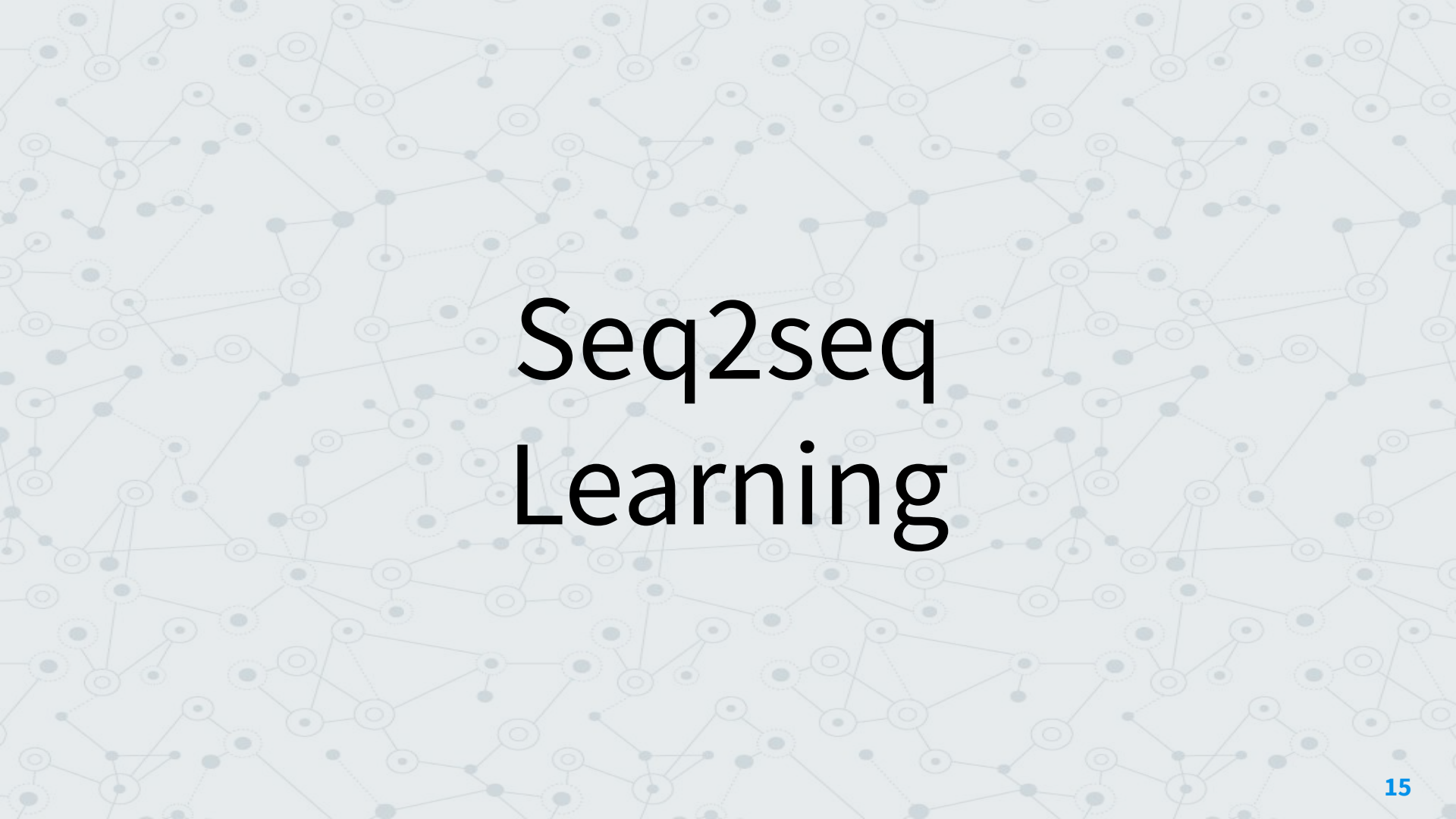


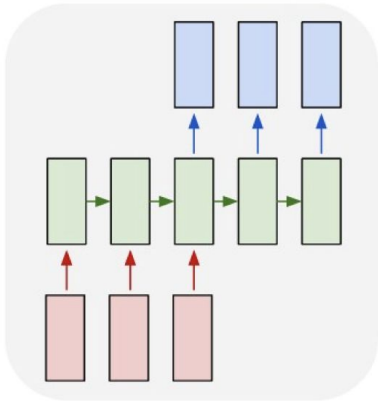
Fig. 2. The outline of application domains of RL in healthcare.

The background of the slide is a light gray network diagram. It consists of numerous small circular nodes, some of which are highlighted with a darker blue or gray fill. These nodes are interconnected by a web of thin, light gray lines, creating a complex, interconnected pattern that resembles a neural network or a data graph.

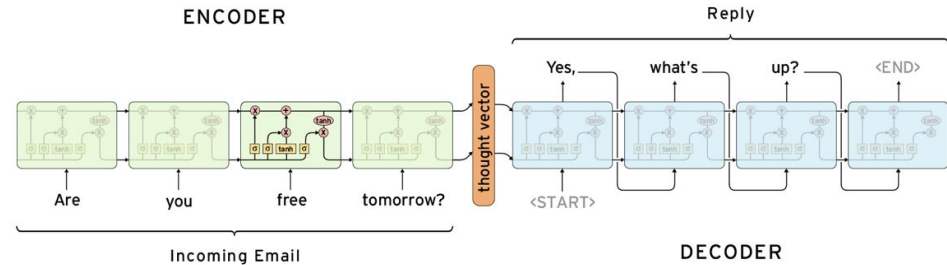
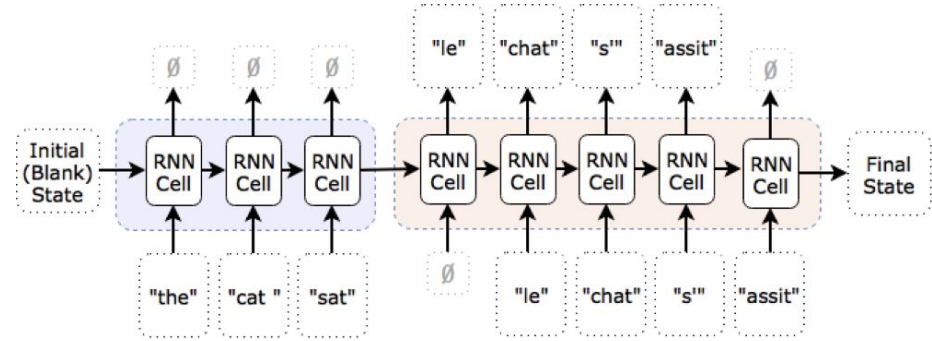
Seq2seq Learning

Recall: Many to Many RNNs

many to many



Ex:
Translation,
automated
response



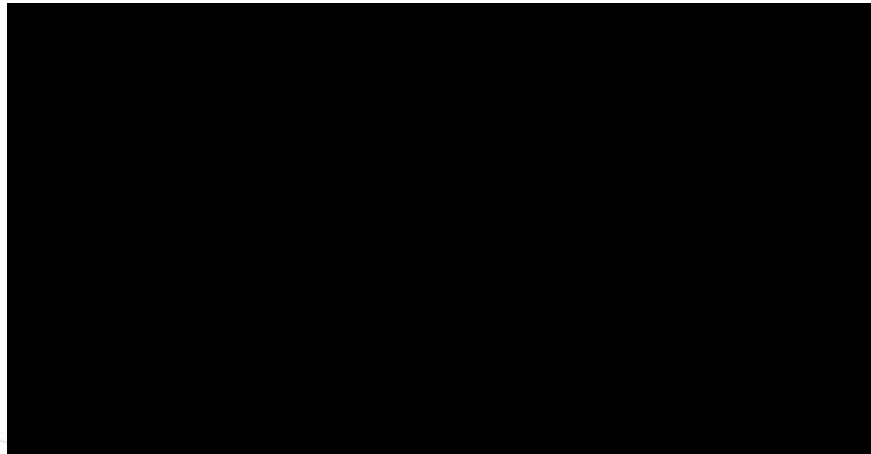
Seq2seq Models

- ◎ Sequence-to-sequence models have been successful for tasks like machine translation, text summarization and image captioning
- ◎ Both the input and output are sequences
 - Words, letters, features of an image, etc.
- ◎ We'll focus on machine translation
 - Input will be a sequence of words and the output will also be a sequence of words

[Video from amazing post about attention and seq2seq models](#)

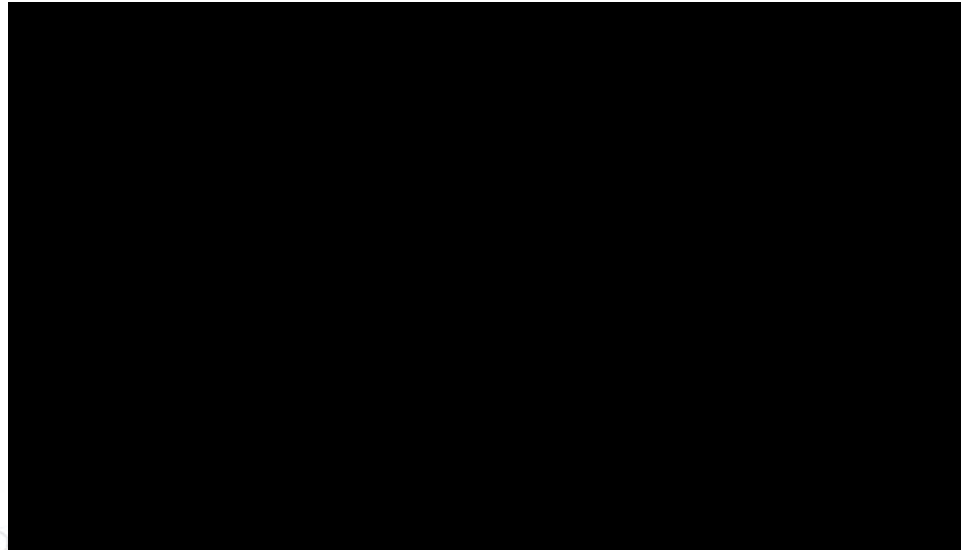
Seq2seq Models

- ◎ Model consists of an **encoder** (RNN) and a **decoder** (RNN)
 - Encoder processes each item in the input sequence and compiles information into a vector called the **context**
 - The encoder sends the context to the decoder and the decoder produces the output sequence one item at a time



Seq2seq Models

- © The context is really a hidden state that is passed to the decoder once



Seq2seq Models

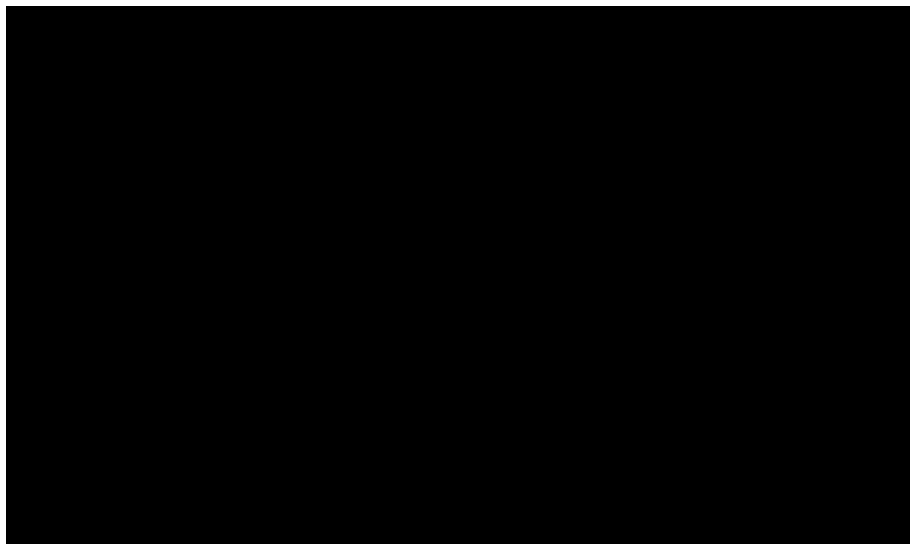
- ◎ Major drawback of seq2seq models
 - The context vector acts as a bottleneck
 - These models struggle with very long sequences
- ◎ In [2014](#) and [2015](#), “**Attention**” was proposed as a solution
 - Attention allows the model to focus on the relevant parts of the input sequence, allowing for longer sequences to be used

The background of the slide is a light gray network diagram. It consists of numerous small circular nodes, some of which are solid gray and others are hollow with a gray outline. These nodes are interconnected by a web of thin, light gray lines, creating a complex, interconnected pattern that fills the entire background.

Attention

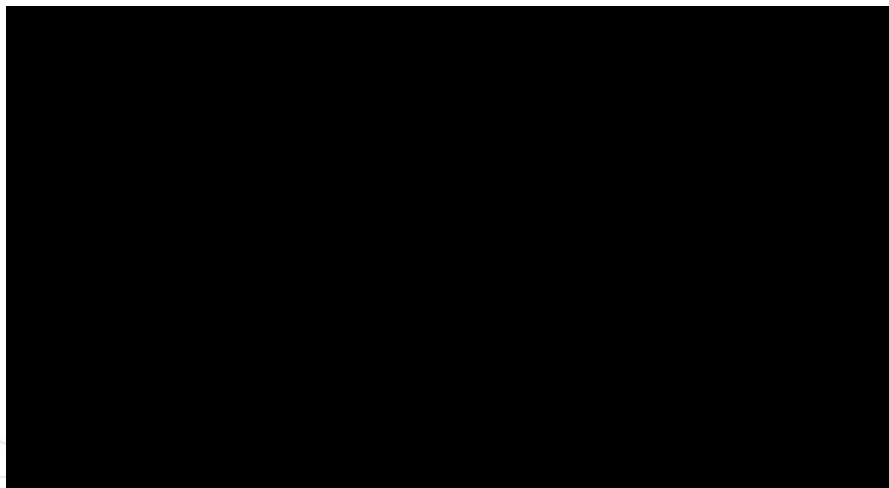
Attention

- ⊙ Rather than passing 1 context vector (the last hidden state) to the decoder, the encoder passes **all** of the hidden states to the decoder
- ⊙ This means that for each output that the decoder makes, it **has access to the entire input sequence and can selectively pick out specific elements from that sequence to produce the output.**



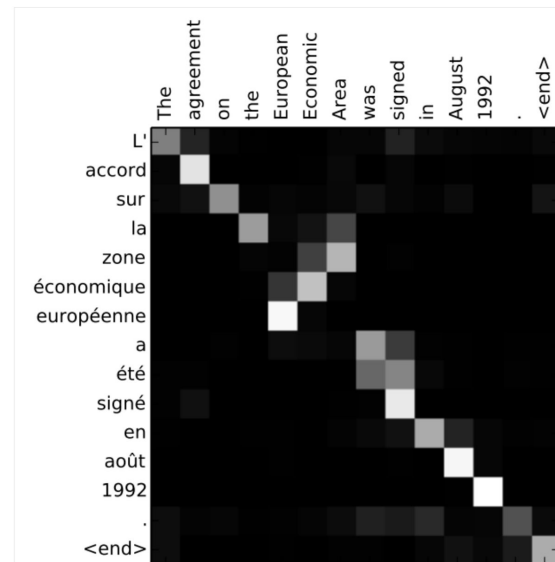
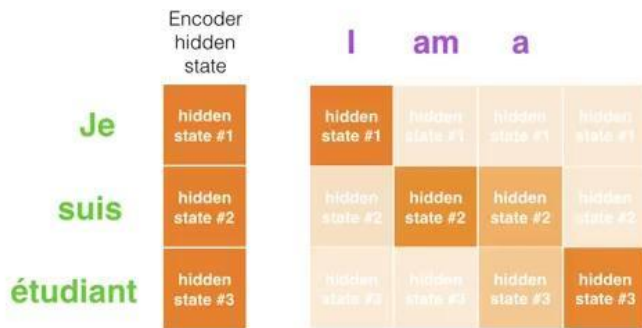
Attention

- ◎ An attention decoder does an extra step before producing its output
 - Looks at each of the hidden states from the encoder
 - Scores each of the hidden states
 - Multiplies each hidden state by its softmax score
 - ◎ Hidden states with **higher scores** will be **amplified**
 - ◎ Hidden states with **smaller scores** will be **dampened** and ignored
 - Sums the weighted vectors into a context vector for that time step



Attention

- Scoring is done by the decoder at each time step
 - For each output word, scoring maps important / relevant words from the input sequence - higher weight means more relevance
 - This helps with the accuracy of the output prediction



You can see how the model paid attention correctly when outputting "European Economic Area". In French, the order of these words is reversed ("européenne économique zone") as compared to English. Every other word in the sentence is in similar order.

Attention Process

- ◎ Send input sequence through encoder
 - Keep track of hidden state at each time point
- ◎ Send all hidden states to decoder along with the final hidden state from the encoder
- ◎ Use the hidden states from the encoder to calculate the context vector for this time step and concatenate it with the final hidden state from the encoder
- ◎ Pass the vector through a feedforward neural network (dense network)
- ◎ The output of the dense network is the output word for this time step
- ◎ Repeat for the next time steps

Attention Process

Neural Machine Translation SEQUENCE TO SEQUENCE MODEL WITH ATTENTION

