BST 261: Data Science II
Lecture 16
Advanced Topics:
Reinforcement Learning and
Course Review

Heather Mattie
Harvard T.H. Chan School of Public Health
Spring 2 2019

Reinforcement Learning

Reinforcement Learning (RL)

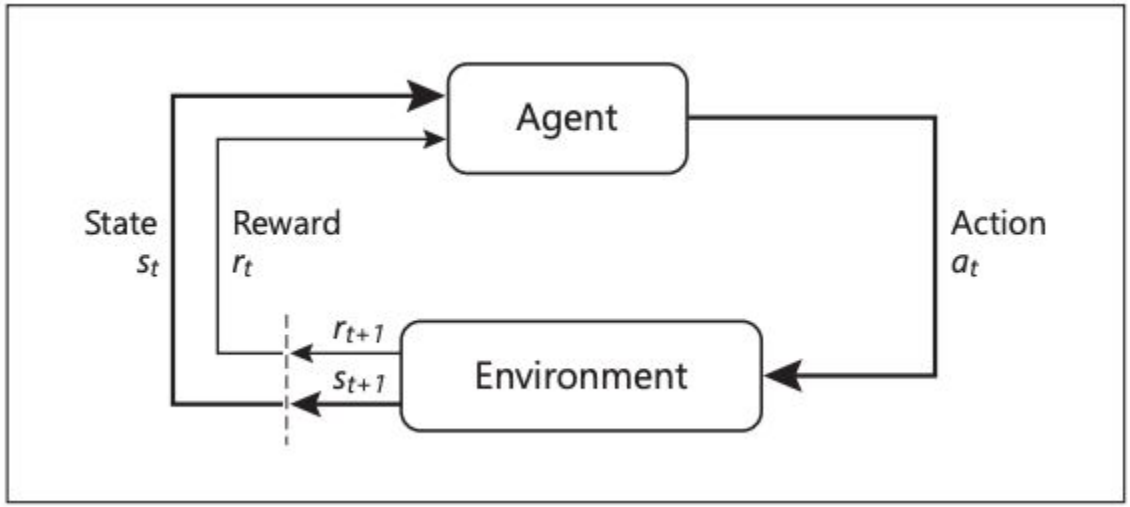
- A subfield of AI that provides tools to optimize **sequences of decisions** for **long-term outcomes**

- Reinforcement learning is learning what to do—how to map situations to actions—so as to maximize a numerical reward signal

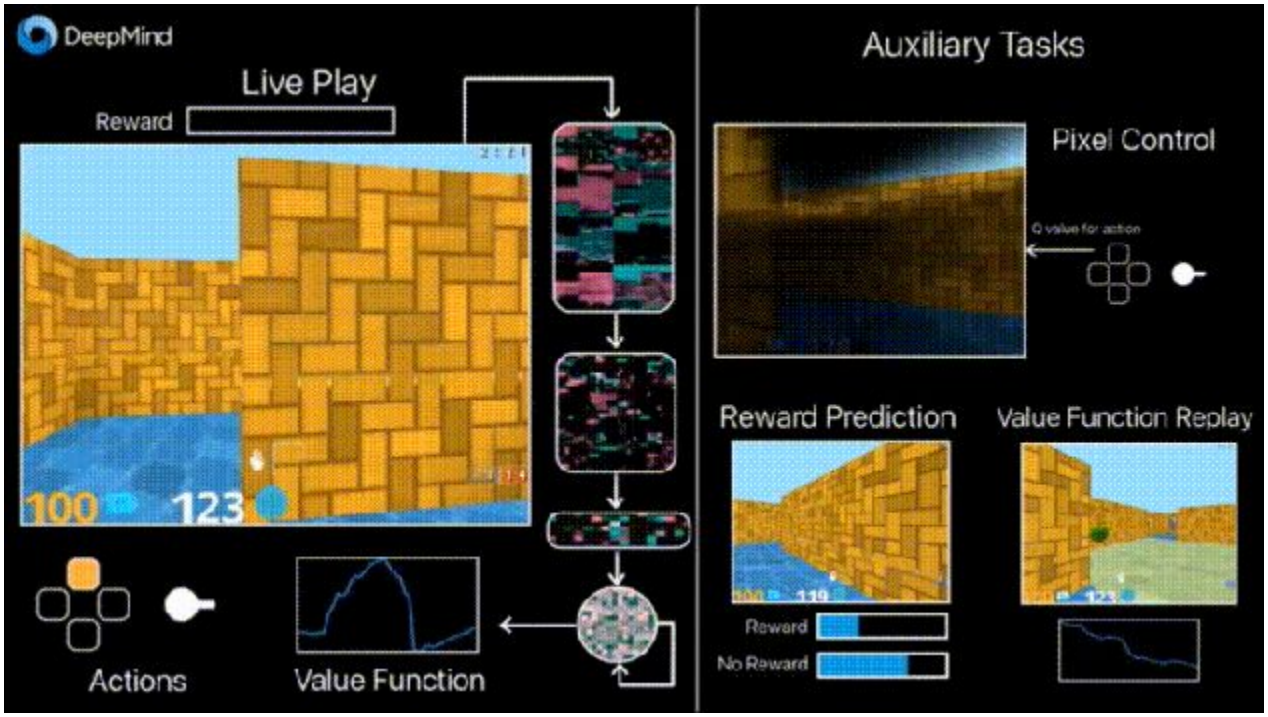
- The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them
  - Lots of interacting with environment
  - Lots of trial and error
  - A decision will affect not only the next action, but actions after that as well

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**)

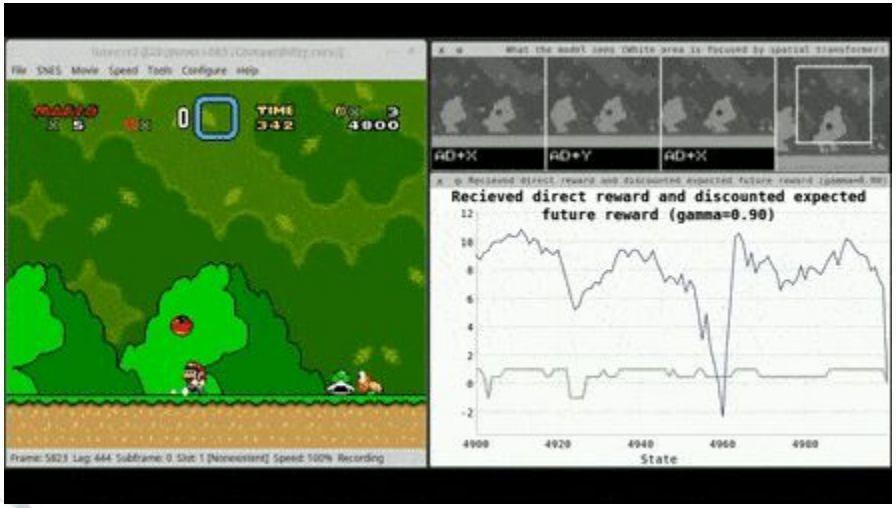


RL


DeepMind
Live Play
Reward
100
123
Actions
Value Function
Auxiliary Tasks
Pixel Control
Q value for action
Reward Prediction
Value Function Replay
Reward
No Reward

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-reinforcment-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

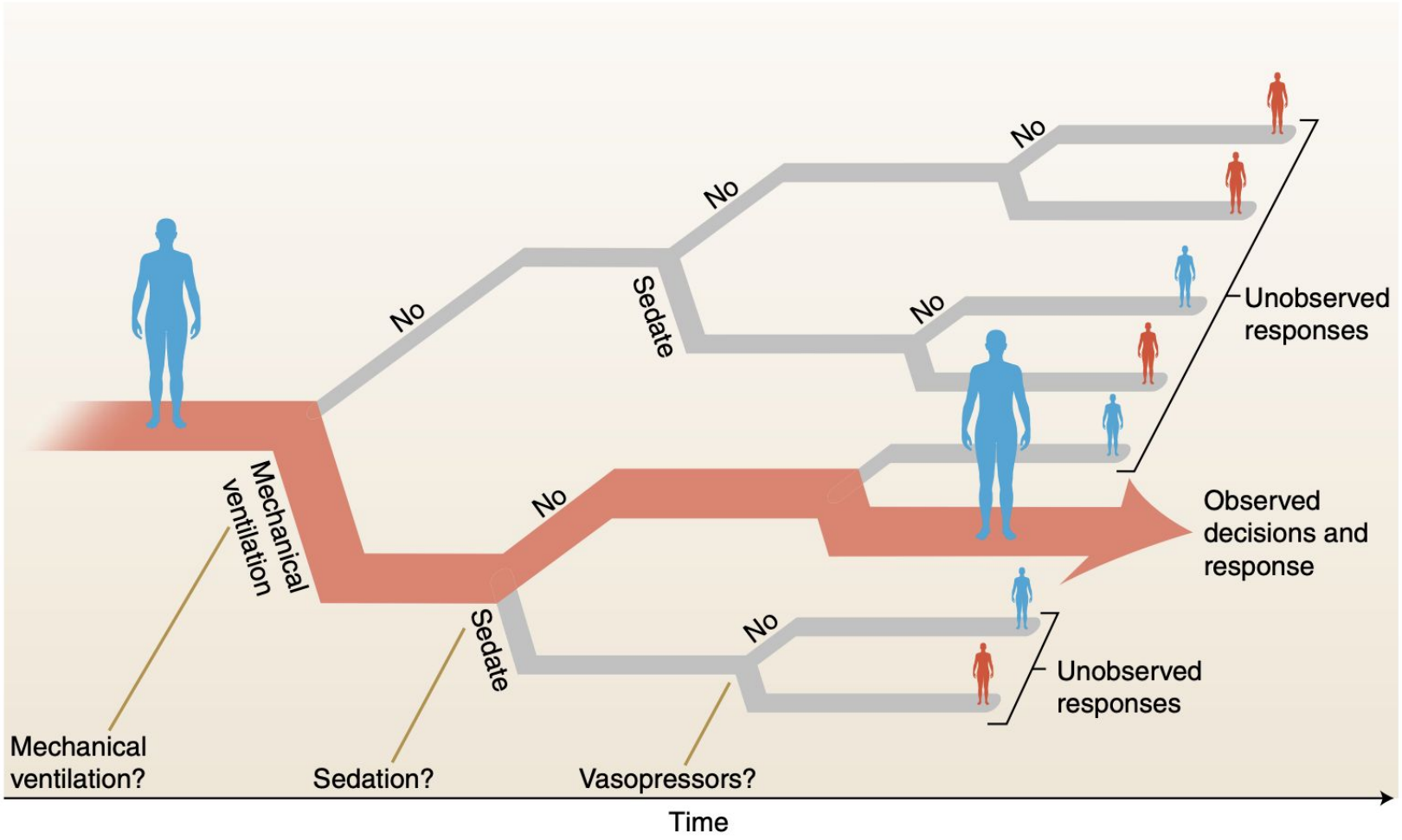


Fig. 1 | Sequential decision-making tasks. To perform sequential decision making, such as for sepsis management, treatment-effect estimation must be solved at a grand scale—every possible combination of interventions could be considered to find an optimal treatment policy. The diagram shows the scale 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?
  - 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

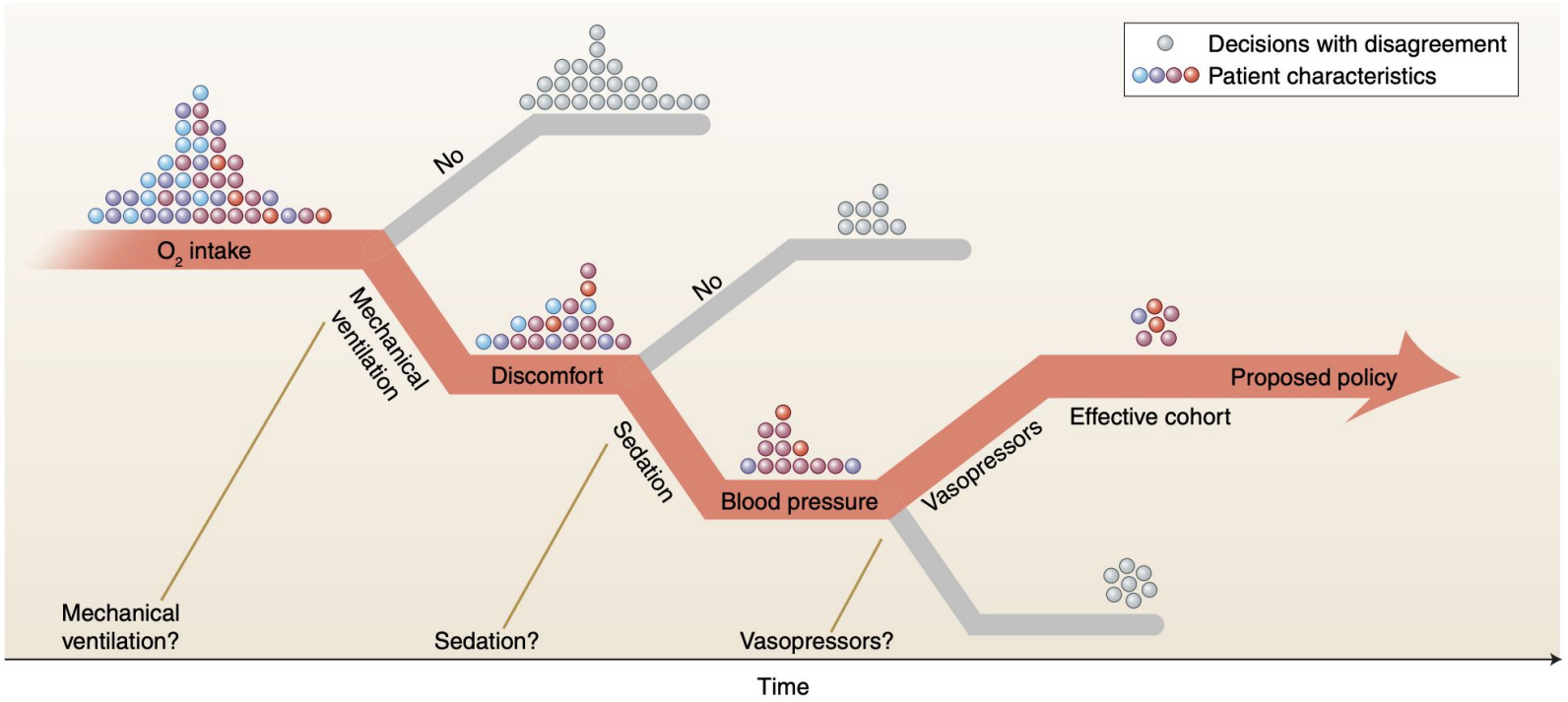


Fig. 2 | Effective sample size in off-policy evaluation. Each dot represents a single patient at each stage of treatment, and its color indicates the patient's characteristics. The more decisions that are performed in sequence, the likelier it is that a new policy disagrees with the one that was learned 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 difficult
- May help guide clinicians in treatment decisions based on all of a patient's history and not their immediate symptoms/responses to treatment

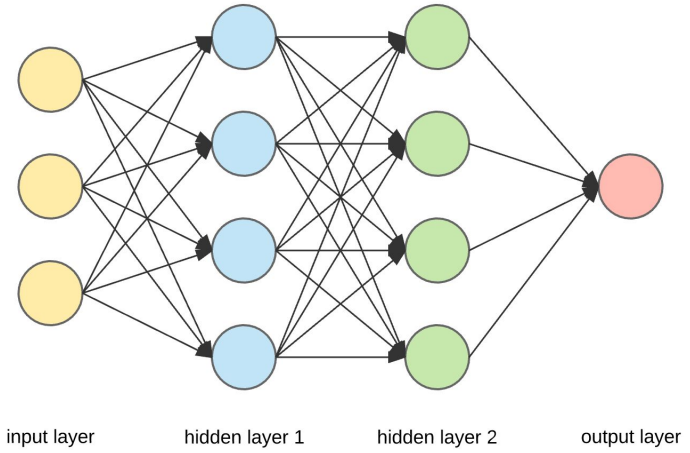
Course Review

What have we learned?

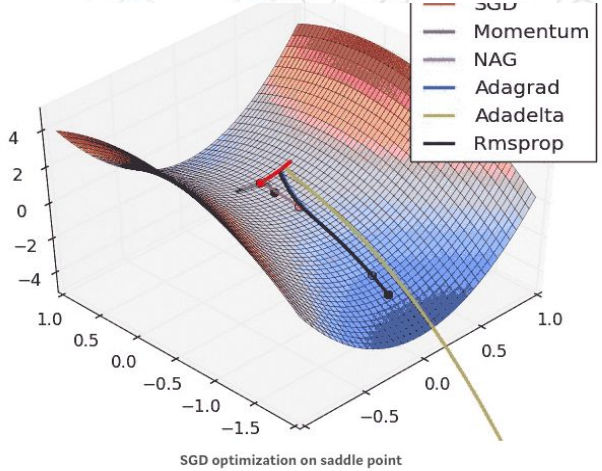
Quite a bit!

- Multilayer perceptrons → CNNs → RNNs → Advanced architectures

- Network architecture
  - Hidden units
  - Layers
  - Activation function
  - Loss functions
  - Optimization algorithms
  - Batch size

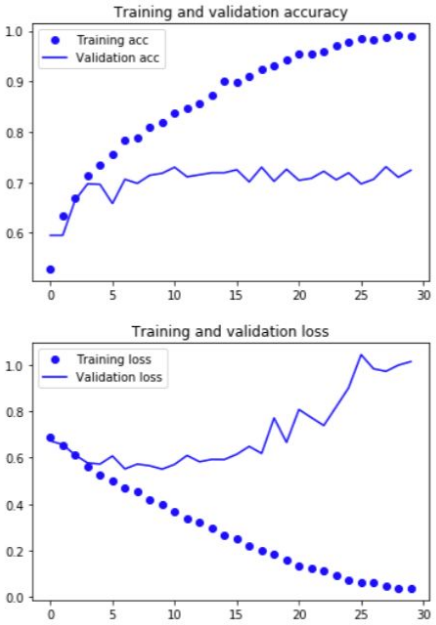


What have we learned?

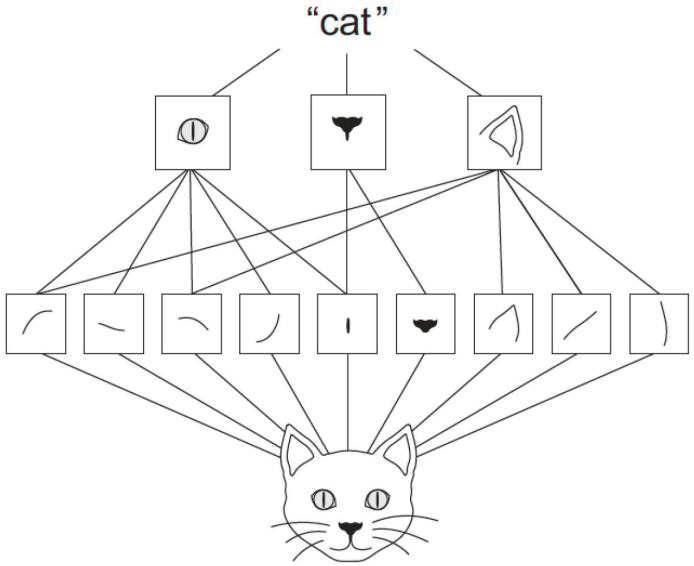


- How networks learn
  - Gradient descent/ascent
  - Backpropagation
  - Forward pass and backward pass
  - Visualizing filters
  - Dense layer vs convolution layer vs RNN/LSTM/GRU layer

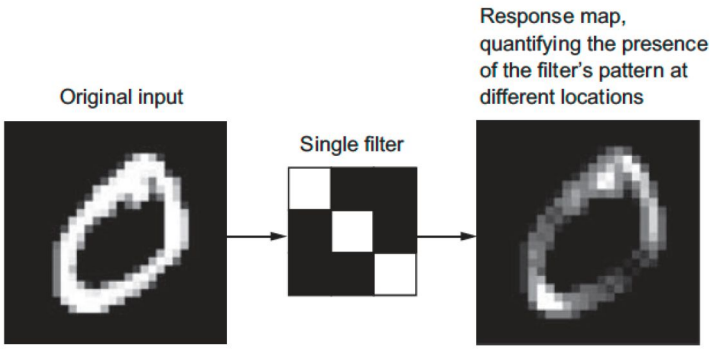
- Model performance
  - Underfitting vs overfitting
  - Regularization techniques
    - Dropout, L2 and L1 norms, network size
  - Bias/variance tradeoff



What have we learned?



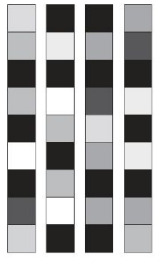
- CNNs
  - Padding
  - Pooling
  - Strides
  - Filters
  - Translation invariance
  - Hierarchical learning
  - Lower layer representations vs higher layer representation
  - Data format (3D vs 4D tensors)
  - Object detection and localization
  - Face recognition
  - 1D CNN for sequential data
  - Landmark detection
  - Data augmentation
  - Neural style transfer



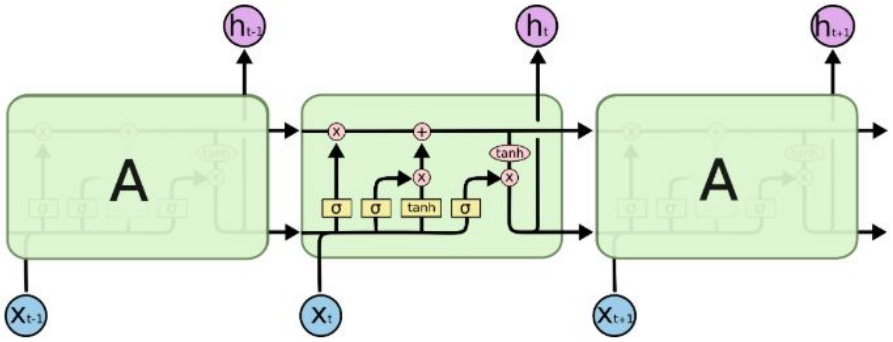
What have we learned?

- RNNs
  - Different types of sequential data
  - How RNNs preserve order in this type of data
  - SimpleRNN vs LSTM vs GRU layers
  - Tokens and tokenization
  - One-hot encoding and hashing
  - Word embeddings
  - Word2Vec and Glove
  - Time series data
  - Recurrent dropout
  - Text generation
  - Bidirectional recurrent layers

One-hot word vectors:
- Sparse
- High-dimensional
- Hardcoded



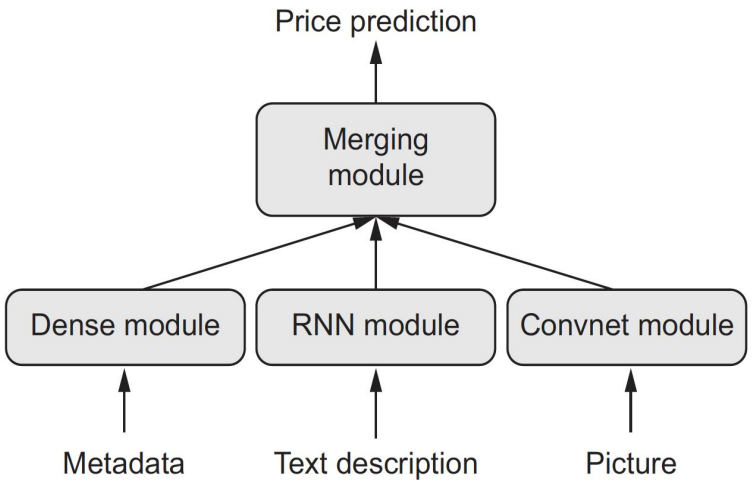
Word embeddings:
- Dense
- Lower-dimensional
- Learned from data



The repeating module in an LSTM contains four interacting layers.

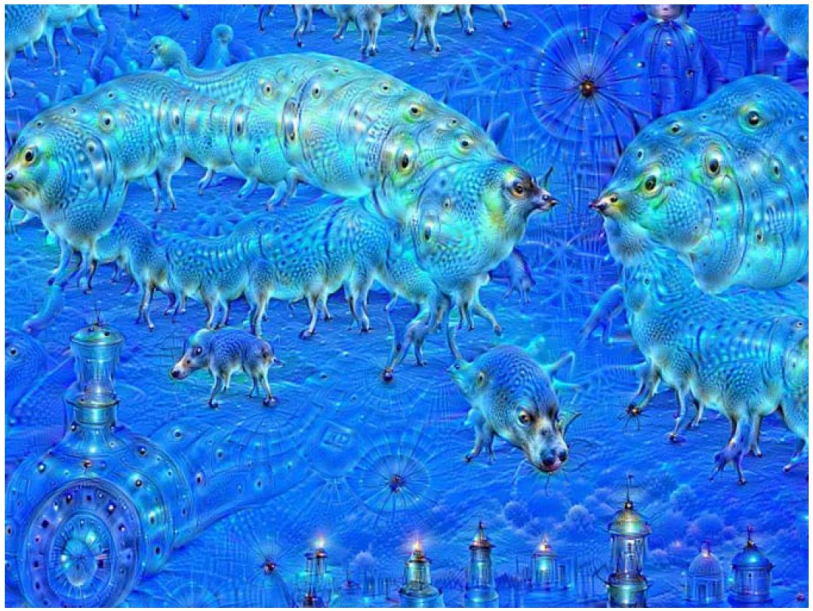
What have we learned?

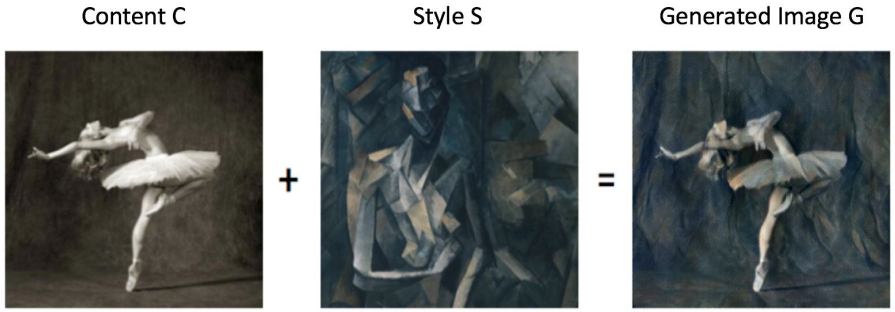
- Advanced network architectures
  - One-to-many (multi-output/multi-head models)
  - Many-to-many
  - Many-to-one (multi-modal models)
  - Directed acyclic graphs
- Advanced architecture patterns
  - Batch normalization
  - Hyperparameter optimization
  - Model ensembling
- Implementation in Keras
  - Tensor (data) manipulation
  - Sequential model
  - MLP, CNN, RNN
  - Using GCP
  - Functional API



What have we learned?

- Advanced topics
  - Variational autoencoders (VAEs)
  - Generative adversarial networks (GANs)
  - Reinforcement learning (RL)
  - DeepDream
  - Neural style transfer
  - Text generation





Picasso Dancer