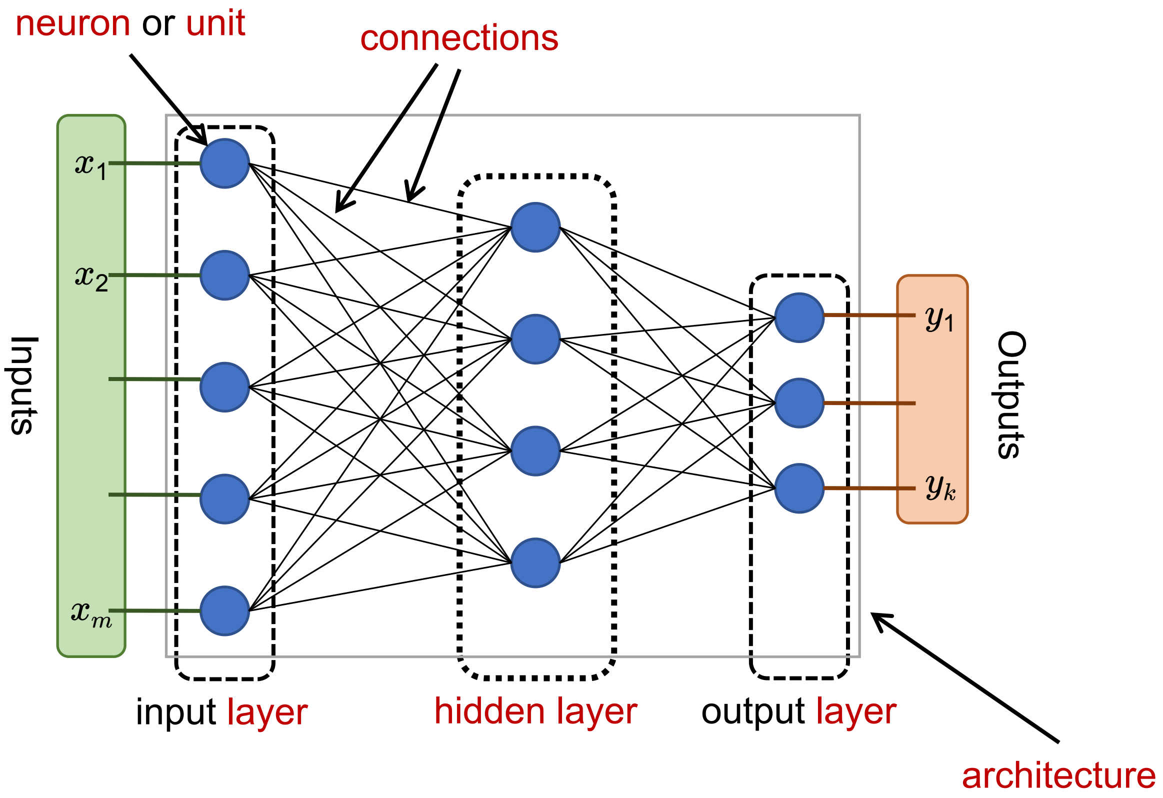
# CAI 4104/6108 – Machine Learning Engineering:

## Recurrent Neural Networks

Prof. Vincent Bindschaedler

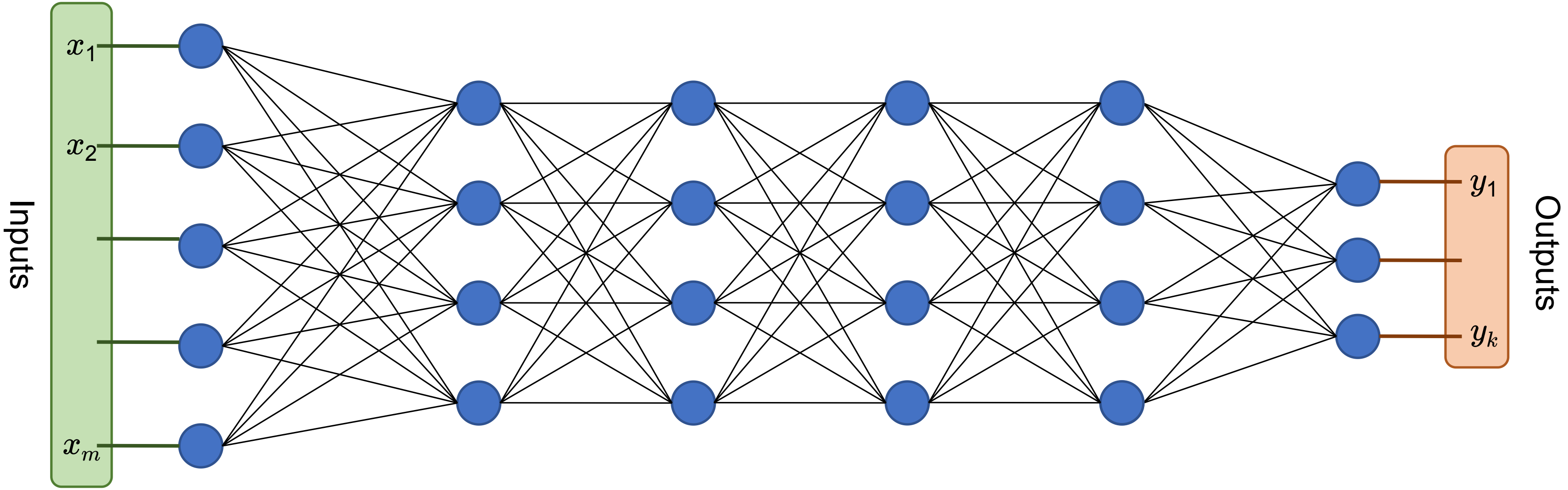
Spring 2024

# Reminder: Neural Network Terminology



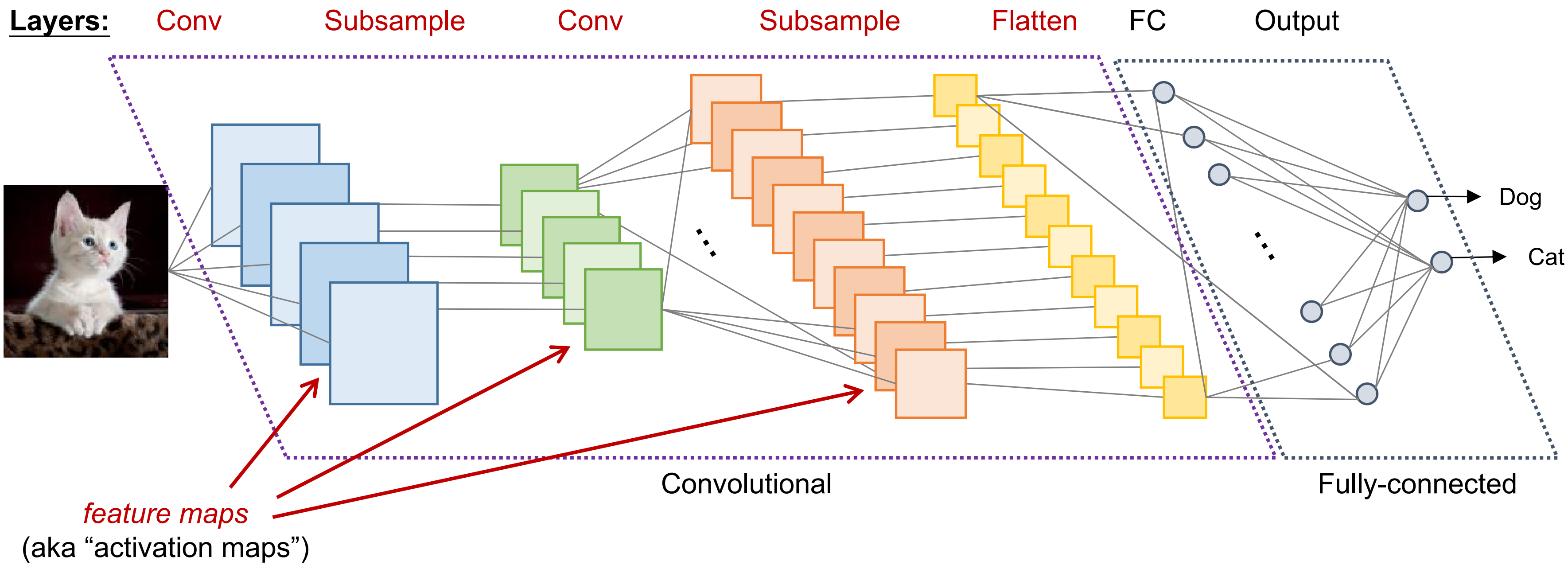
# Reminder: Deep Neural Networks

- What is a deep neural network?
  - Any neural network with two or more hidden layers
  - Nowadays, the best neural networks architectures for many applications & problems are deep
    - E.g.: AlexNet (2012) has 8 layers, ResNet18 has 18 layers, GPT-2 has 48 layers

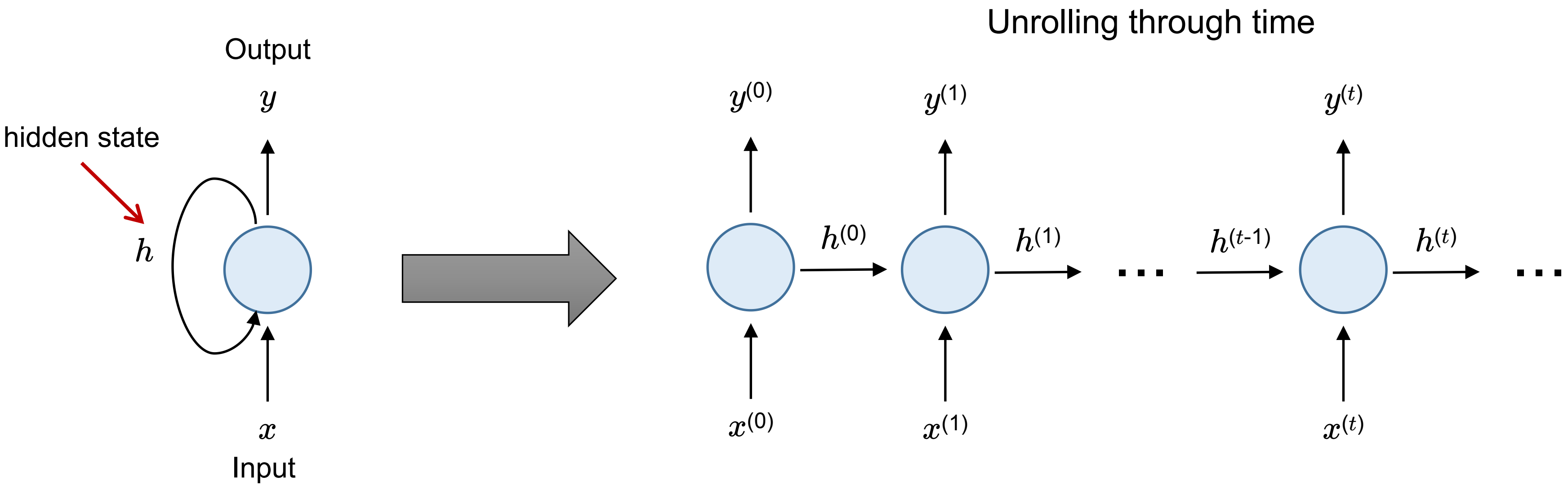


# Reminder: CNNs

- Example & Terminology:



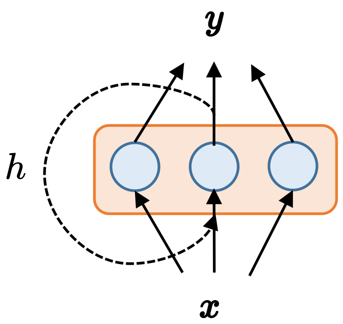
- Recurrent Layers:
  - Made up of recurrent neurons/units which keep state
  - State at time $t$: $h^{(t)}$ is a function $g$ of the previous state $h^{(t\text{-}1)}$ and the current input $x^{(t)}$
    - $h^{(t)} = g(h^{(t\text{-}1)}, x^{(t)})$



# Recurrent Layers

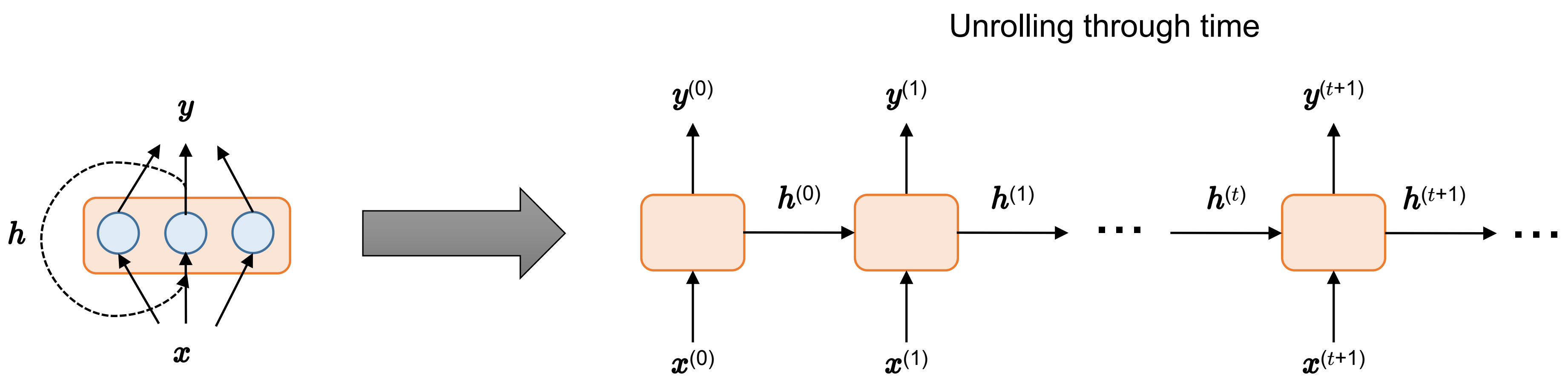
- Recurrent Layers:
  - Weight matrices: $\boldsymbol{W}$ $(m \times k)$ and $\boldsymbol{V}$ $(k \times k)$ Bias vector: $\boldsymbol{b}$ $(k \times 1)$
    - $k$ is the number of units/neurons
  - Activation function: $f$ (e.g., tanh)
  - Hidden state vector: $\boldsymbol{h}^{(t)} = \boldsymbol{V}\ \boldsymbol{y}^{(t\text{-}1)}$ (e.g., we can set $\boldsymbol{h}^{(0)} = 0$)
  - Output vector: $\boldsymbol{y}^{(t)} = f\,(\boldsymbol{W}^T \boldsymbol{x}^{(t)} + \boldsymbol{h}^{(t)} + \boldsymbol{b})$



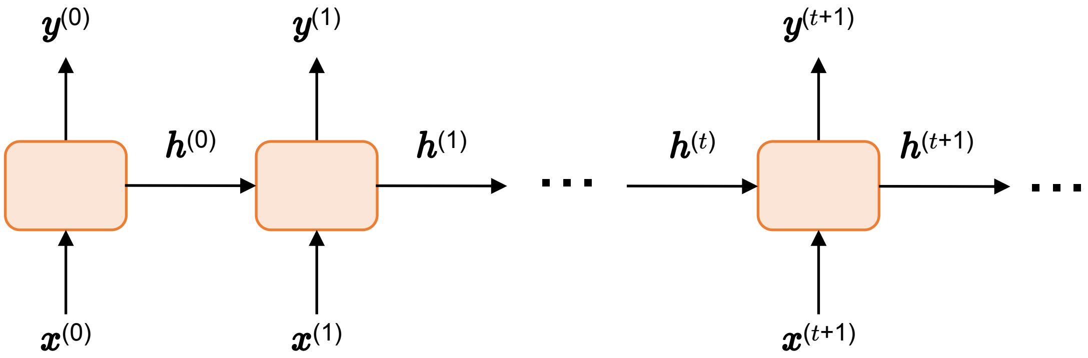
# Processing Sequences

- Architecture & Tasks:
  - Sequence-to-sequence: from an input sequence produce a sequence as output
  - Vector-to-sequence: from a fixed length input produce a sequence as output
  - Sequence-to-vector: from an input sequence produce a fixed length output
  - Encoder-decoder networks: sequence-to-vector followed by vector-to-sequence



# Processing Sequences

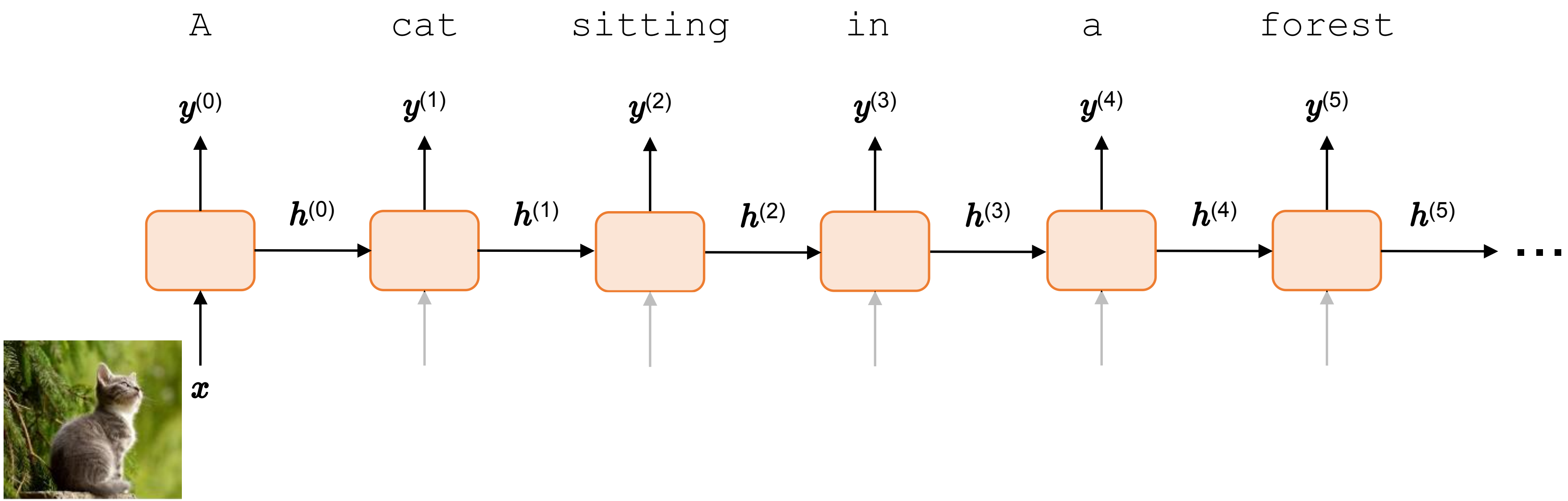
- Architecture & Tasks:
  - Sequence-to-sequence: for each input frame there is a single output frame
  - Example: predicting stock prices
    - Feed the price of a stock over the last $n$ days, network predicts the price on day $n$+1



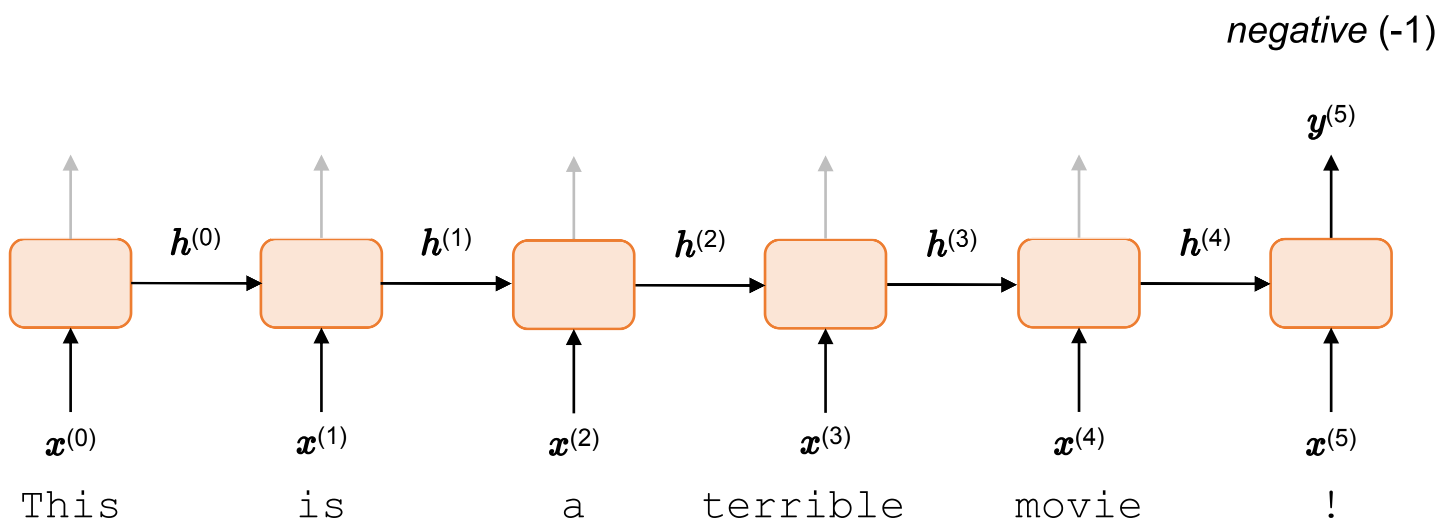
# Processing Sequences

- Architecture & Tasks:
  - Vector-to-sequence: there is a vector of inputs, the model produces a sequences as output
  - Example: Image captioning
    - Feed the image to the model. Model predicts one word/character of the caption at a time



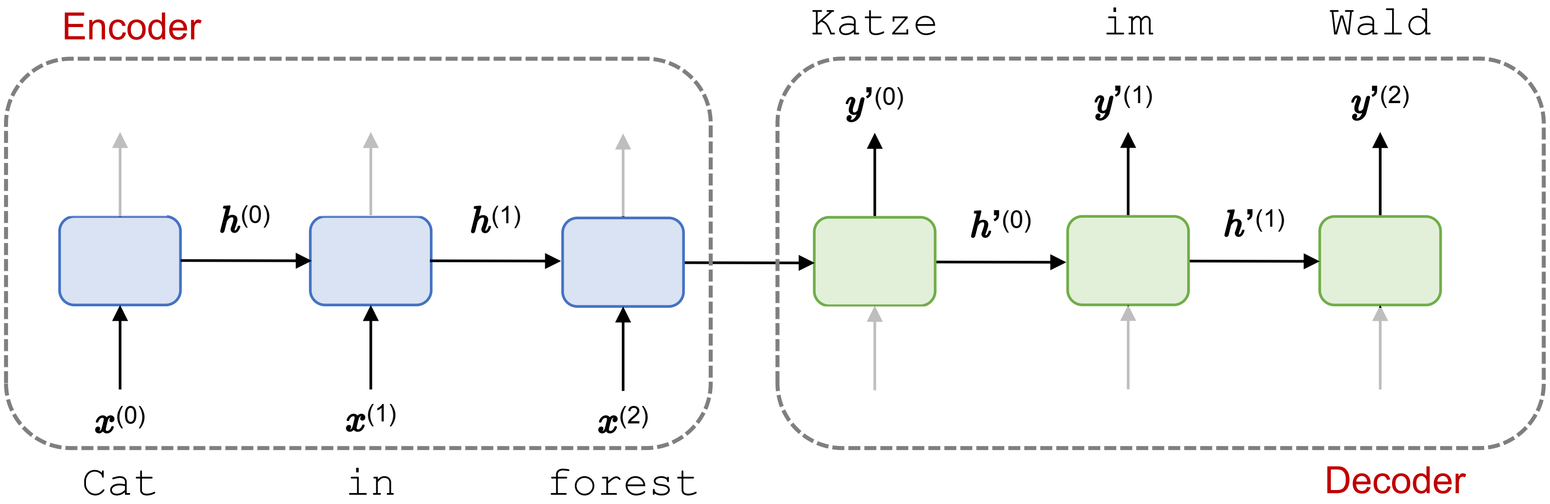
# Processing Sequences

- Architecture & Tasks:
  - Sequence-to-vector: input is a sequence, the model produces a vector (fixed length) as output
  - Example: Sentiment analysis
    - Given the text of a movie review the model outputs “positive” (+1) or “negative” (-1)



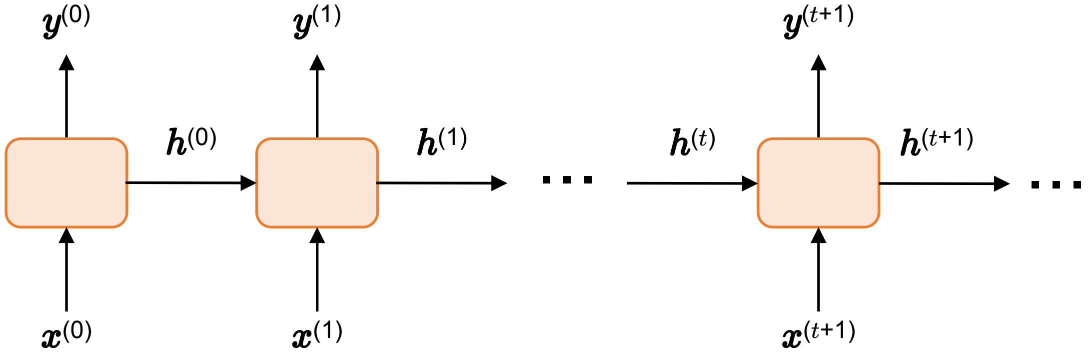
# Processing Sequences

- Architecture & Tasks:
  - Encoder-decoder networks: sequence-to-vector followed by vector-to-sequence
  - Example: Language translation
    - Translate a sentence from one language to another



# Training Recurrent Neural Networks

- How does training work?
  - Backpropagation through time
  - Note:
    - Loss is typically averaged over the entire output
    - The weights are shared across time
    - Training is slow



# Unstable Gradients & Short-Term Memory

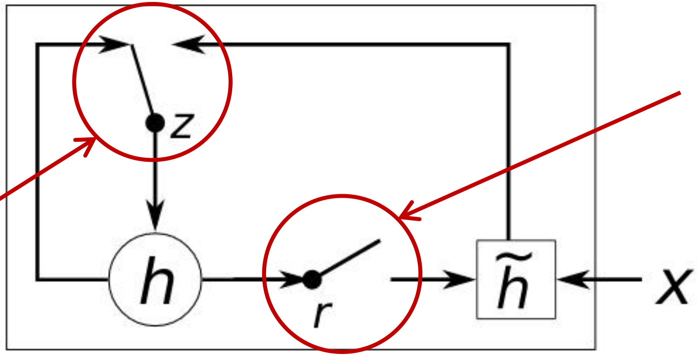
- Unstable gradients problem
  - Activation functions that do not saturate (e.g., ReLU) can make things worse
    - Typically, we use activation functions such as tanh or sigmoid
  - We cannot use batch normalization across time steps
    - But we can use gradient clipping

- Short-term memory problem
  - RNNs cannot remember long-term dependencies well
    - Intuition: information is lost at each time step
  - Mitigation
    - Use a different type of cell (e.g., LSTM, GRU, etc.)

# Gates & Recurrent Units

- Types of cells
  - Simple/traditional RNN cell
  - Long Short-Term Memory (LSTM)
  - Gated Recurrent Unit (GRU)
    - Cho et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.” In EMNLP, 2014.

update gate $z$ — allows information from previous hidden state to carry over to current hidden state




Reset gate $r$ — decides if previous hidden state is ignored (reset to current input)

# Next Time

- Wednesday (3/27): Lecture

- Upcoming:
  - Homework 4 out soon
  - Project Proposals due 3/27