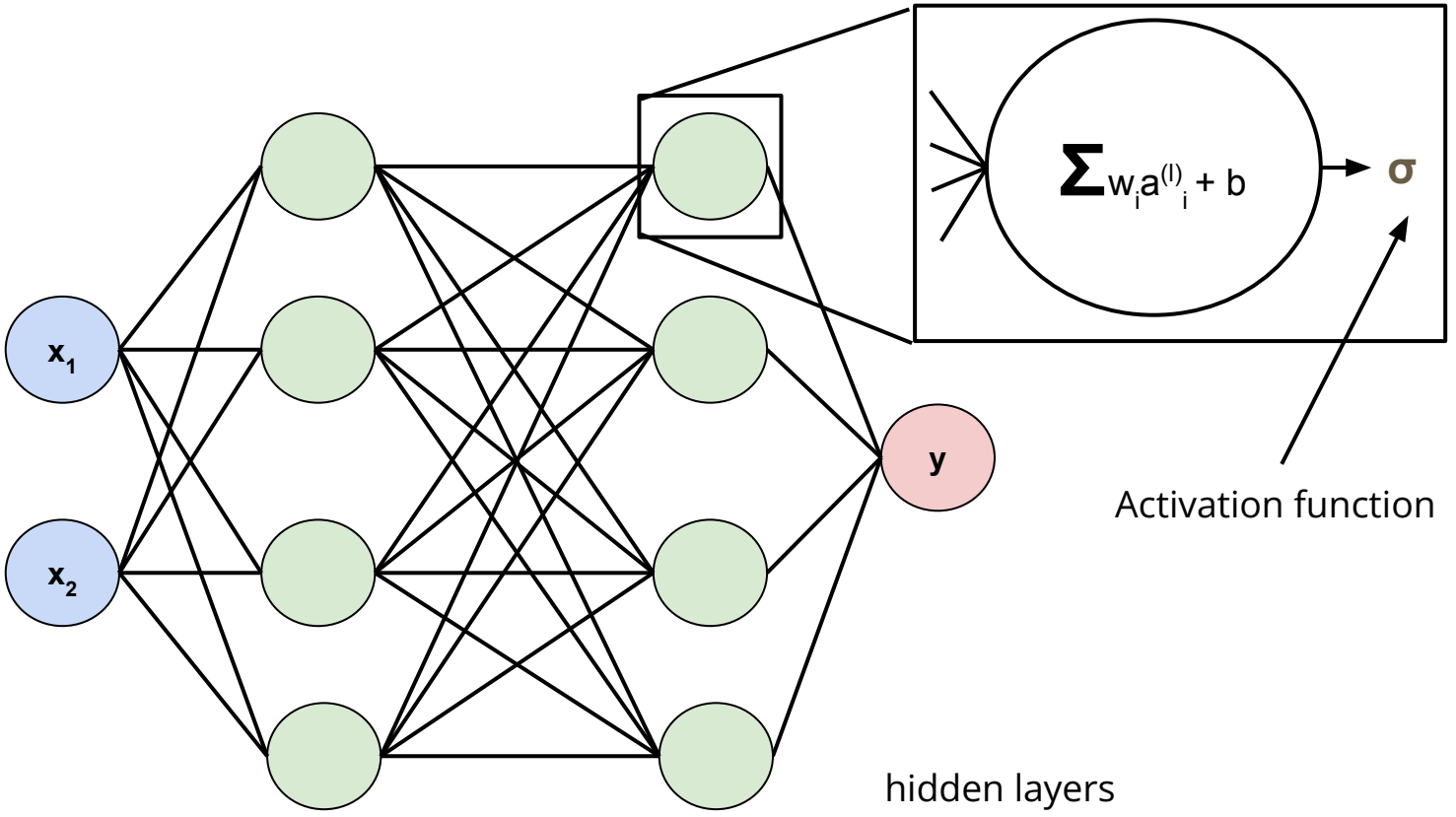
Feedforward Neural Networks

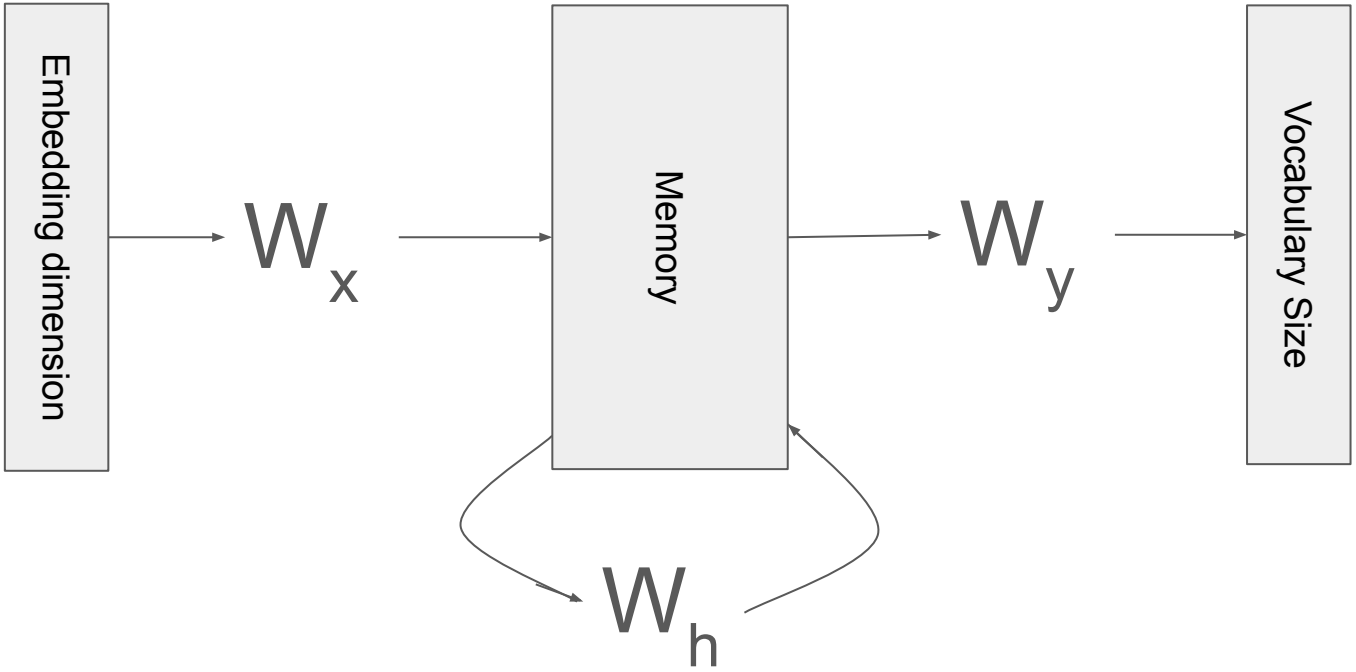
In general:



Feedforward Neural Networks

- Cannot handle variable-length inputs
- No memory of past inputs

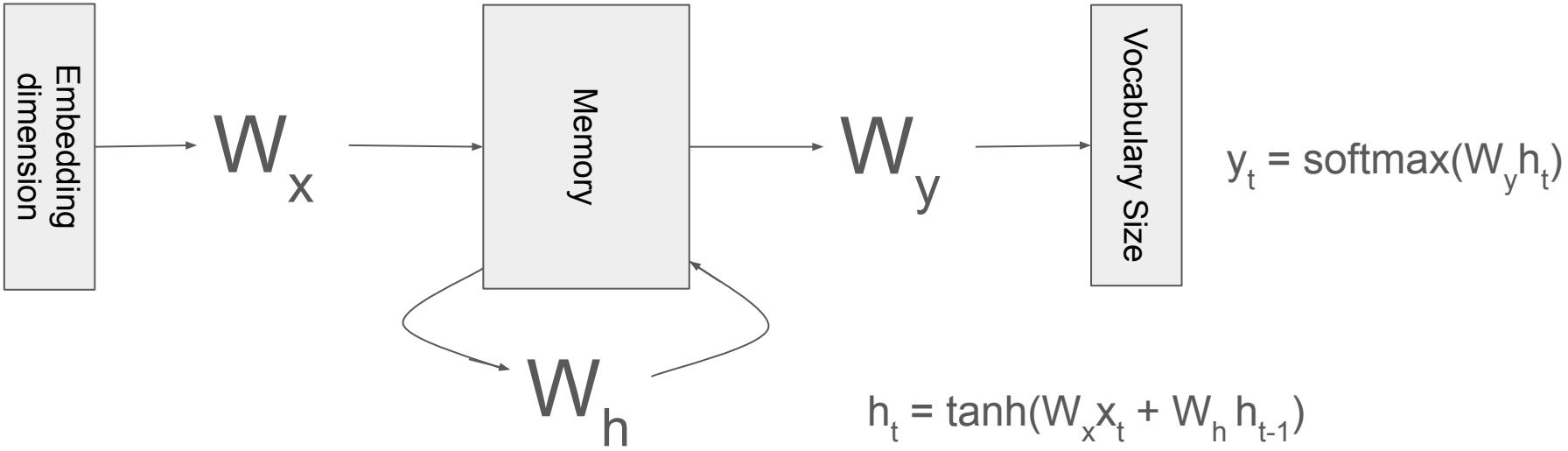
Recurrent Neural Networks



Recurrent Neural Networks

Example: “the cat sat down”.
Vocab = 4 words total. Let’s say embedding dim is 3.

x_1 -> h_1 -> y_1 (which should be x_2 but not as an embedding - as vocab dim probs)
x_2 -> h_2 (depends on h_1) -> y_2 (which should be x_3)



Recurrent Neural Networks

- Struggle with long term memory
- Vanishing gradients
- All hidden state is compressed into one vector (h_t)

LSTM (Long Short Term Memory)

Added a gating mechanism to RNNs for long-term info:

- Forget gate: what to throw away
- Input gate: what to store
- Output gate: what to expose

Basically instead of compressing all past info into a single vector, have h_t be the short-term memory and use the gates to store long term memory using a process to learn what info to keep and expose.

LSTM (Long Short Term Memory)

- Still have to compress long-term info into a single vector...
- Still have to compress short-term info into a single vector...
- Slow to train
- Vanishing gradients

Attention

Key Idea: Use softmax on the sequence of words to learn which words are more important to pay attention to in a given context.

Transformers

Key Idea: We don’t need to process one token at a time. We just need an encoding of the position of a token in a sequence.

Attention + positional encoding = fully parallelized easy access to what matters in the past