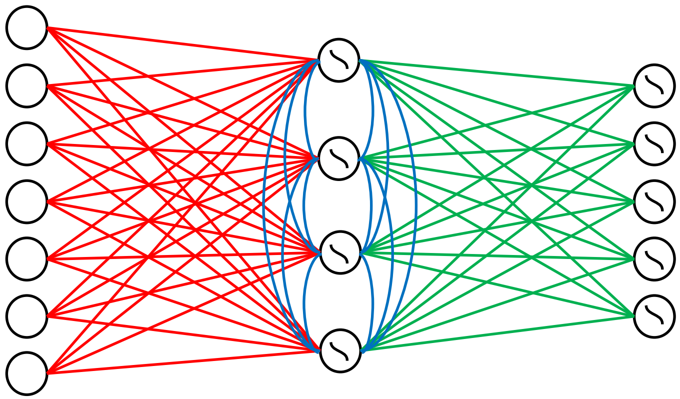
Recurrent Neural Networks

Nikola Milosavljević

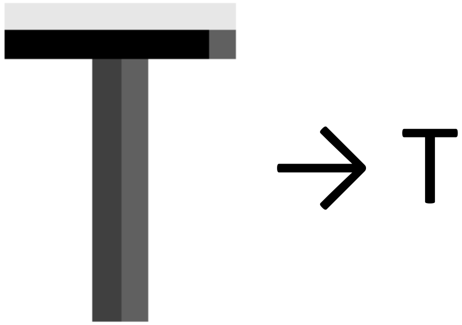
Recurrent neural networks

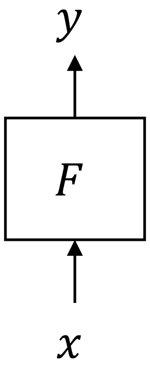
- Suitable for problems where input and/or output is a sequence
- Neural networks with directed cycles (recurrent connections)
  - So far: feed forward networks (directed acyclic computation graphs)



Single item problems

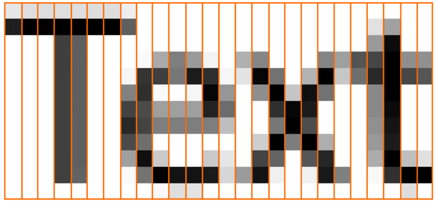
- Solved by feed forward (non-recurrent) neural networks
  - Including convolutional
- Input and output viewed as a single item





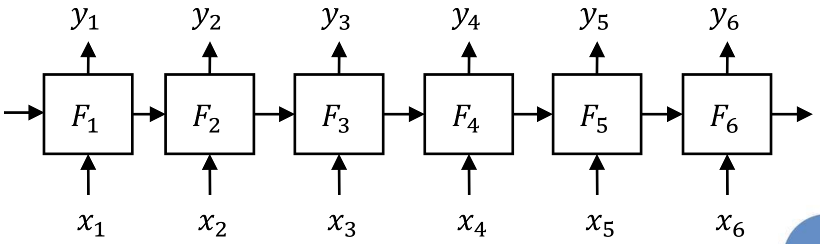
Sequence problems

- Solved by recurrent neural networks
- Input and/or output viewed as one-dimensional sequences (e.g. in space, time…)
  - Processing each item depends on results of processing previous items
  - Input sequences have different lengths



↓

Text

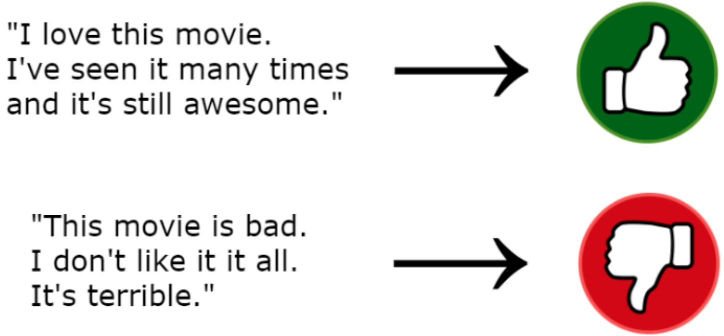


Sequence problems

- Usually involving text, speech, video...
- Sequential input, output, or both?
  - Many-to-one
  - One-to-many
  - Many-to-many

Many to one

- Text sentiment analysis
- Action classification
- Language modeling



One to many

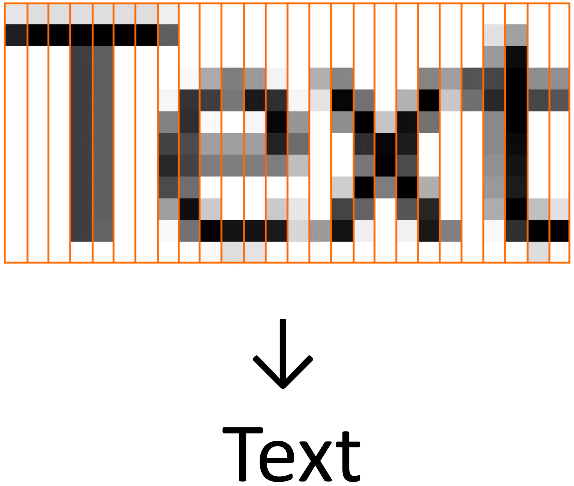
- Image description (captioning)

↓

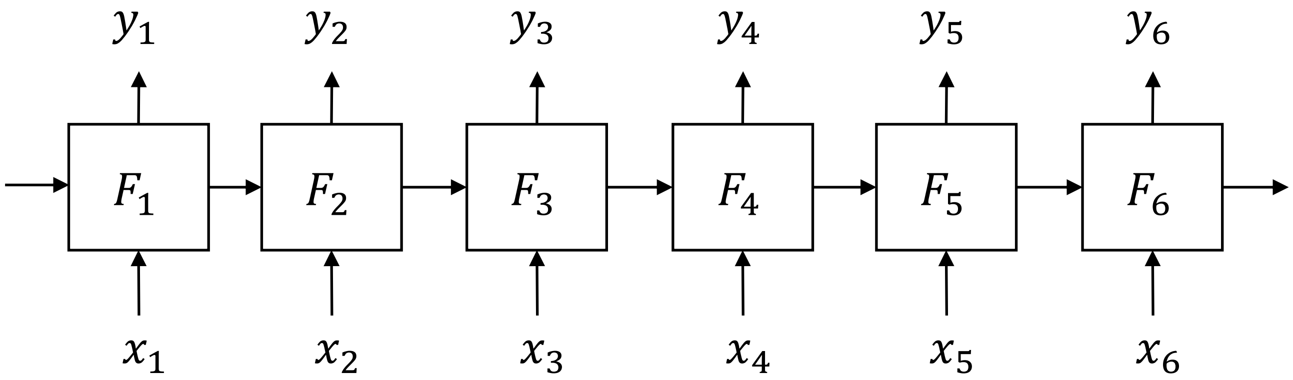
Two dogs are playing in the grass

Many to many

- Optical character recognition (OCR)
- Handwriting recognition
- Machine translation
- Video description
- Speech recognition
- Speech synthesis
- Question answering

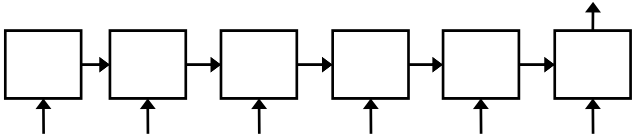


Generic "architecture"

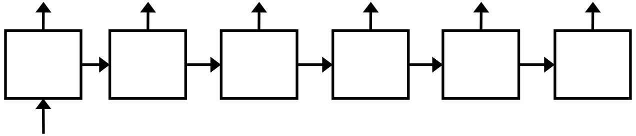


Specializations

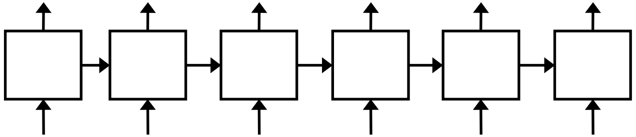
MANY-TO-ONE



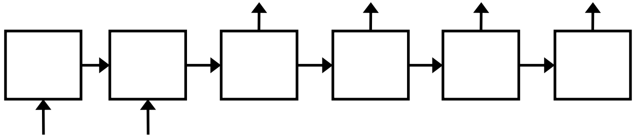
ONE-TO-MANY



MANY-TO-MANY WITH ALIGNMENT

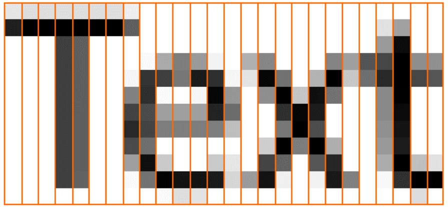


MANY-TO-MANY WITHOUT ALIGNMENT



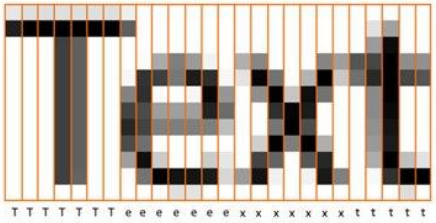
Input/output alignment

- Some problems have it naturally, some don't
  - Example: OCR vs. translation
- Even if it exists, it may not be available for training
  - Example: OCR with and without framewise labels
  - Reasons: expensive labeling, ambiguous labeling



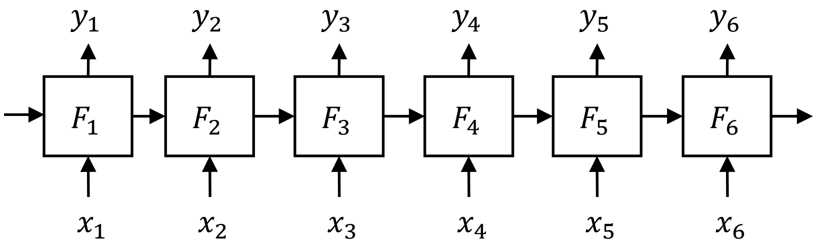
↓

Text



Recurrent computation

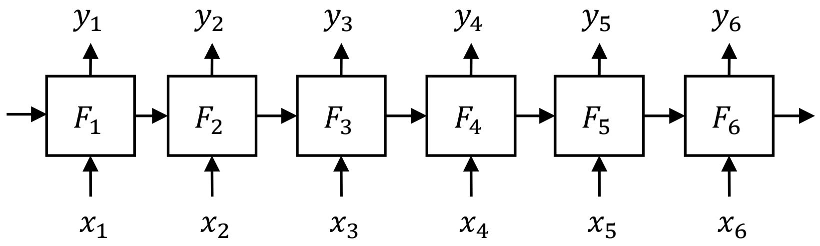
- Recall: generic “architecture”

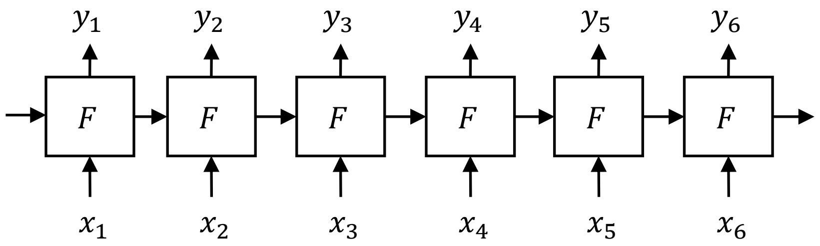


- Recall: input sequences have different lengths
- Network “length” is proportional to length of the sequence
- Trained network can have a fixed set of weights

Recurrent computation

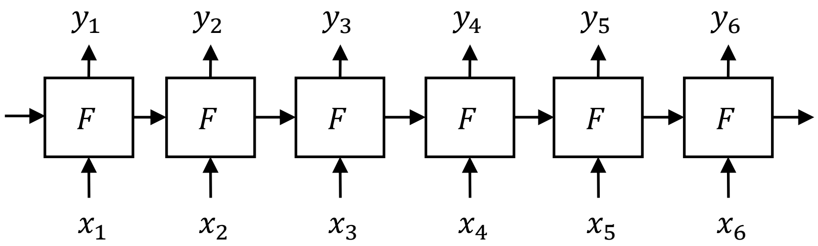
- All computation steps in the sequence share the same weights

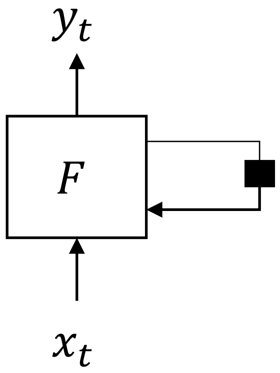




Folded and unfolded view

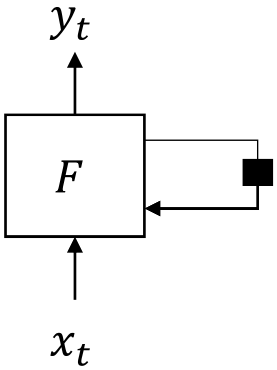
- Can fold network along input sequence
- Folded view has a recurrent connection

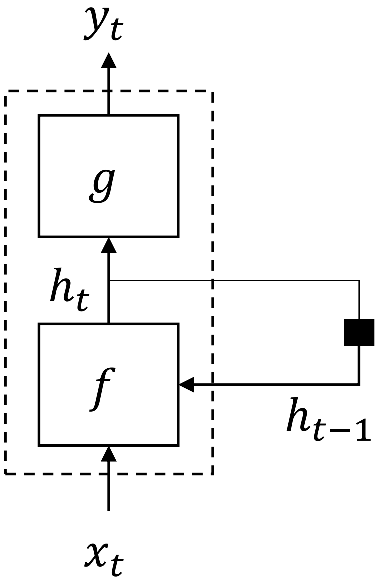




Generic RNN

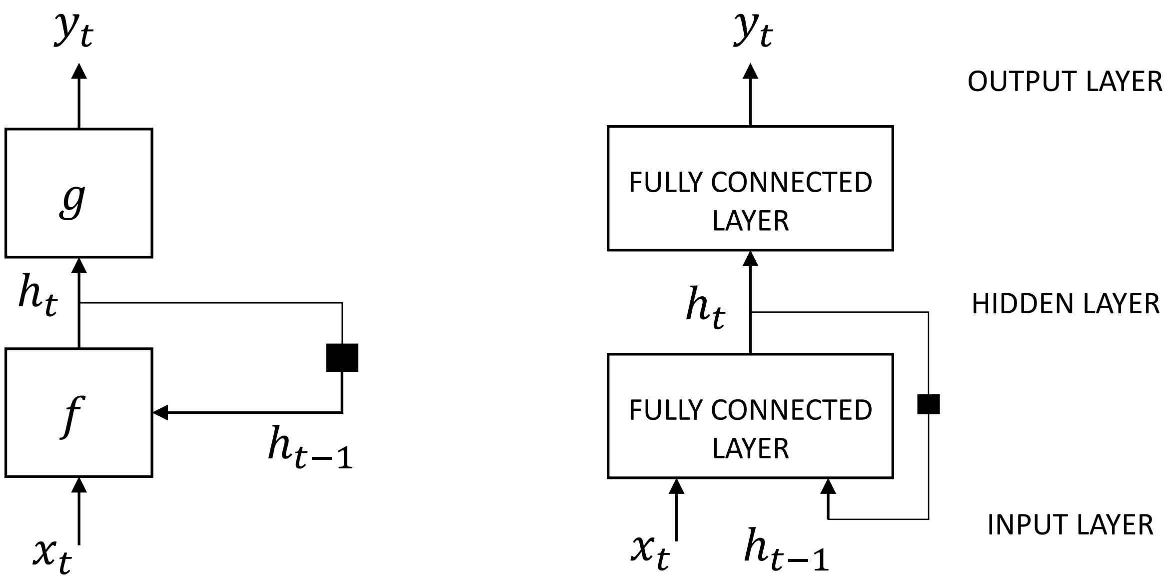
- Recurrent computation implemented using neural networks
- State: subset of activations passed to the next step
  - Fixed-size summary of input seen so far
  - Current state is a function of current input and previous state
  - Output is a function of current state



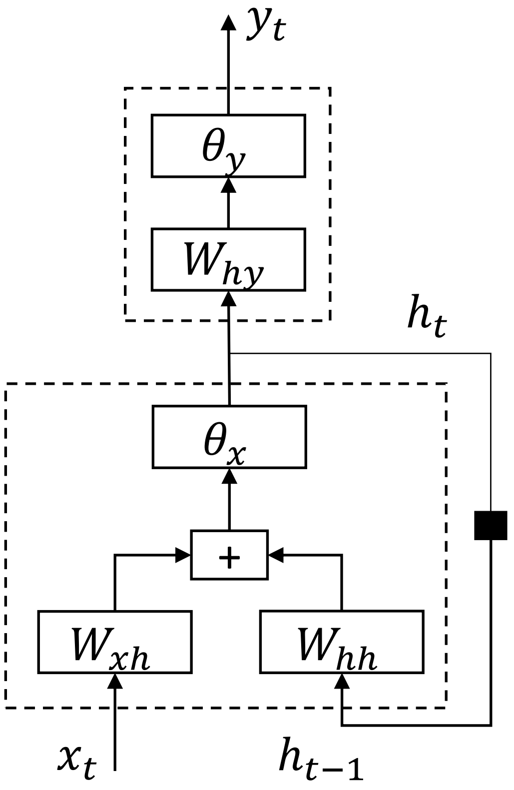


Simple RNN

- Both functions are of type linear + activation



Simple RNN



- W_{xh}, W_{hh}, W_{hy} : linear (matrix mult.)
- θ_x : activation (tanh, sigmoid, ReLU,…)
- θ_y : output activation (identity, softmax)

$$h_t = \theta_x(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1})$$
$$y_t = \theta_y(W_{hy} \cdot h_t)$$

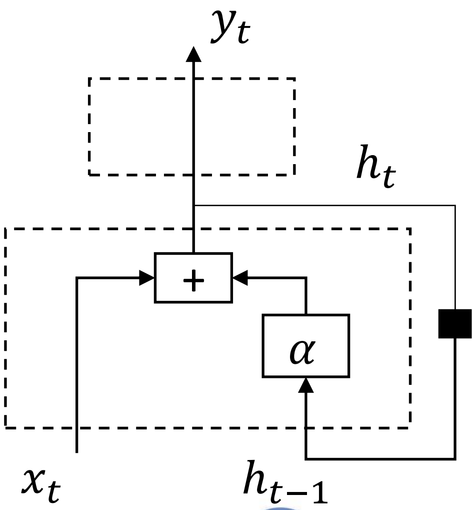
Example: running average

- Given “decay factor” α, compute

$$y_t = x_t + \alpha \cdot x_{t-1} + \alpha^2 \cdot x_{t-2} + \alpha^3 \cdot x_{t-3} + \cdots$$

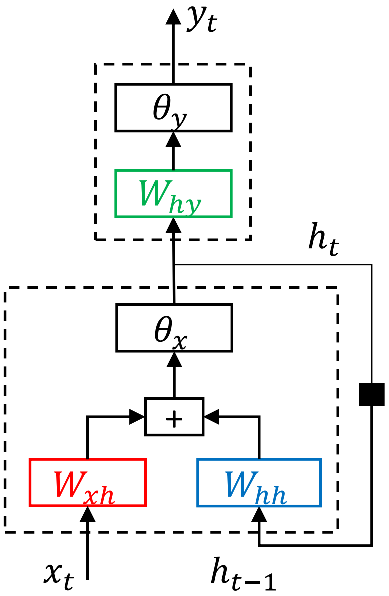
or equivalently

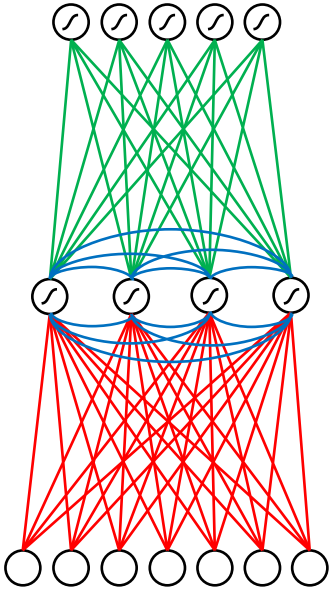
$$y_t = x_t + \alpha \cdot y_{t-1}$$

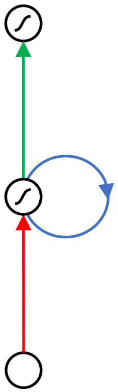


Simplified diagrams

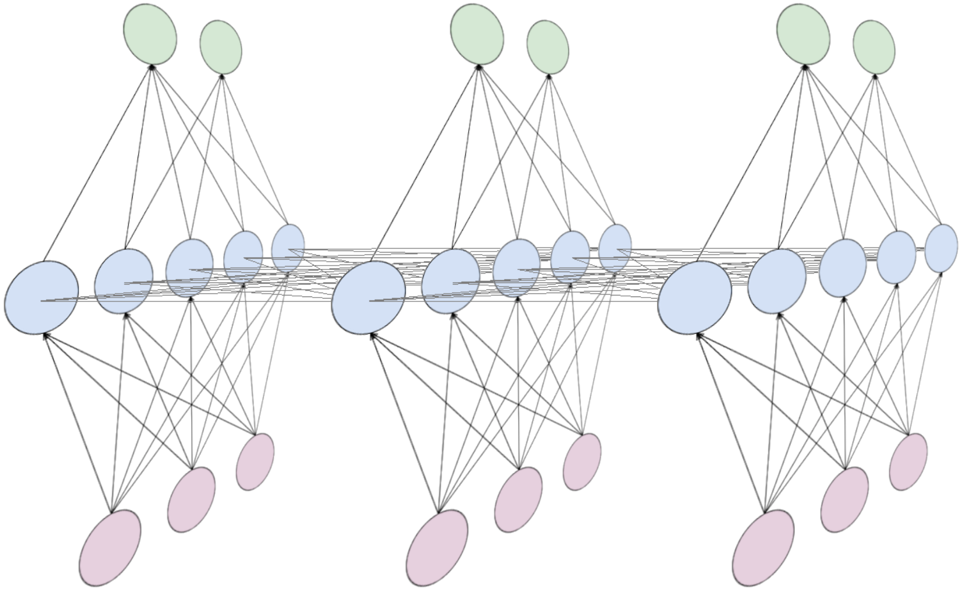
- Drawing neurons as graph nodes, omitting computation blocks

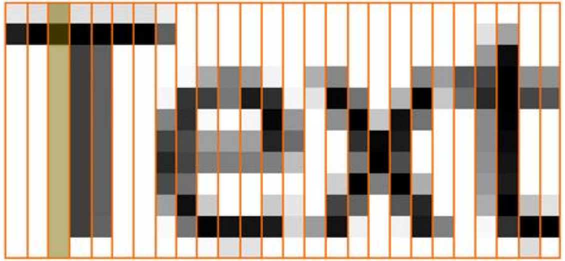






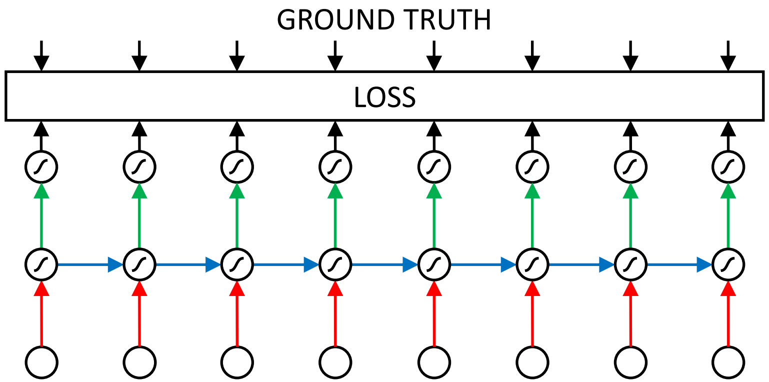
Example: OCR

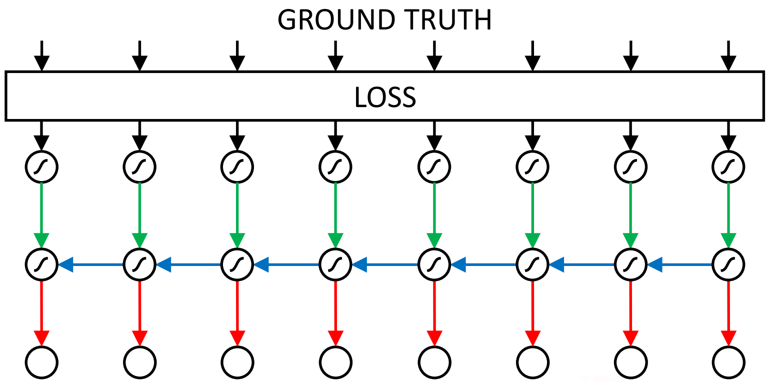




Training

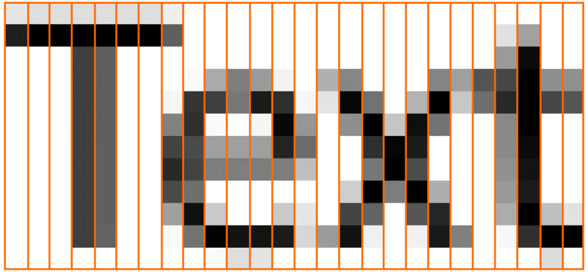
- Backpropagation through time (BPTT)
- The same as standard backpropagation on unfolded RNN

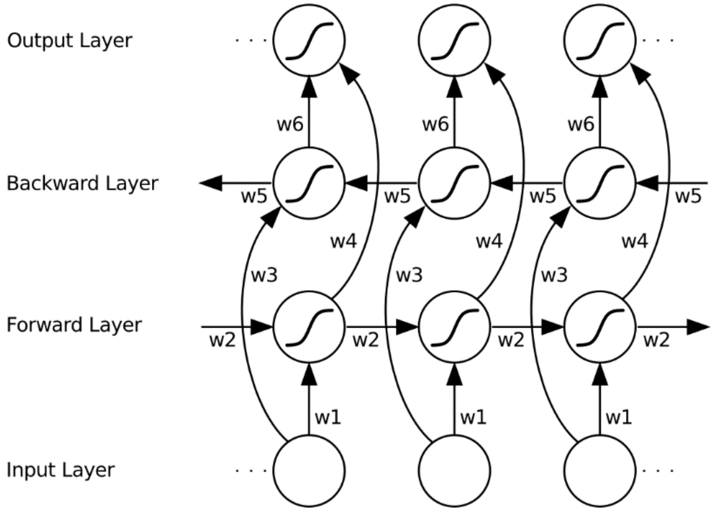




Bidirectional network

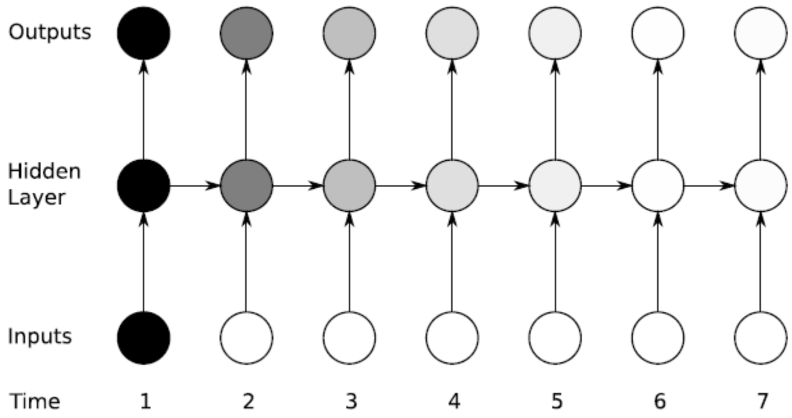
- Context from past and context from future
- In handwriting it is useful to know letters before and letters coming after





Long range dependency issue

- Influence of an input item decays over time
- New inputs overwrite activations of the hidden layer
- Vanishing and exploding gradient



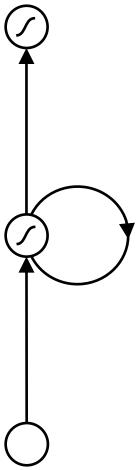
Strategies for long range dependencies

- Exploding gradient is addressed by clipping gradients
- Vanishing gradients:
  - Adding long-range connections (bigger time delays)
  - Removing some short-range connections
  - Hardcoding weights of recurrent connections to 1
  - Removing activation functions from recurrent connections
- Better solution: gating

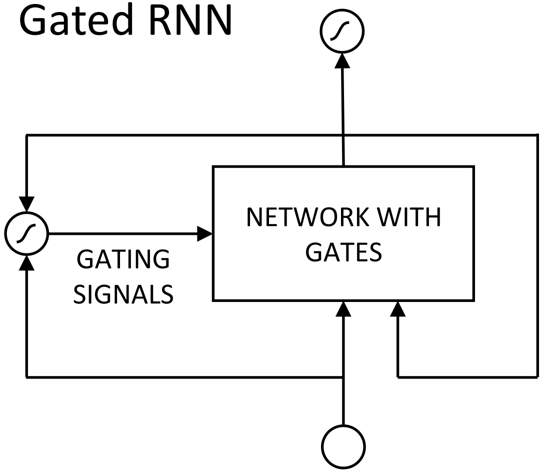
Gated RNNs

- Introduce learnable “gates” on hidden units that control flow of information based on previous state and current input

Basic RNN

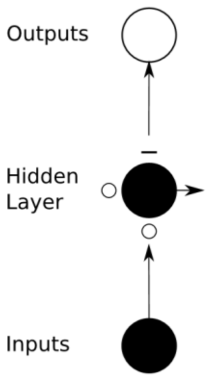


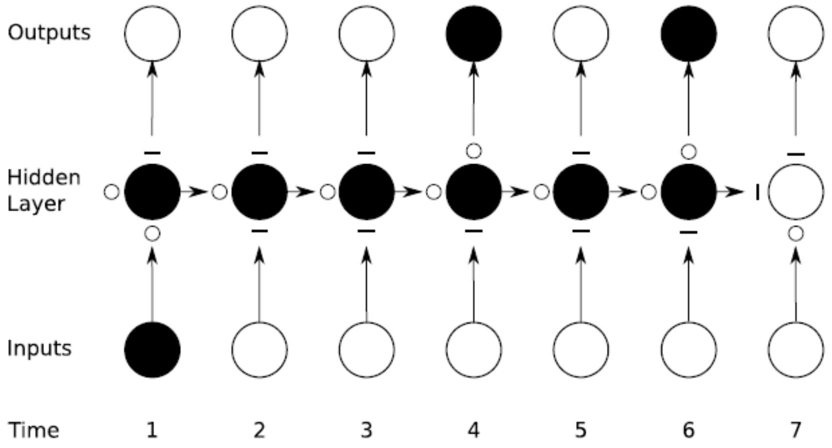
Gated RNN



Gated RNNs: toy example

- Information is preserved if input-to-hidden gate is closed and hidden-to-hidden gate is open

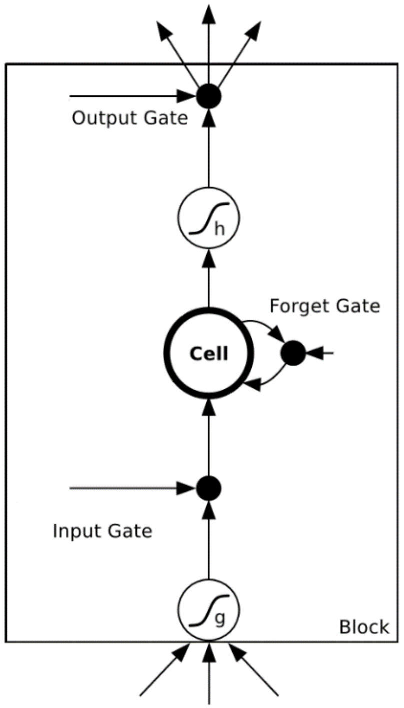




o: open (weight 1)
-: closed (weight 0)

Long short-term memory (LSTM)

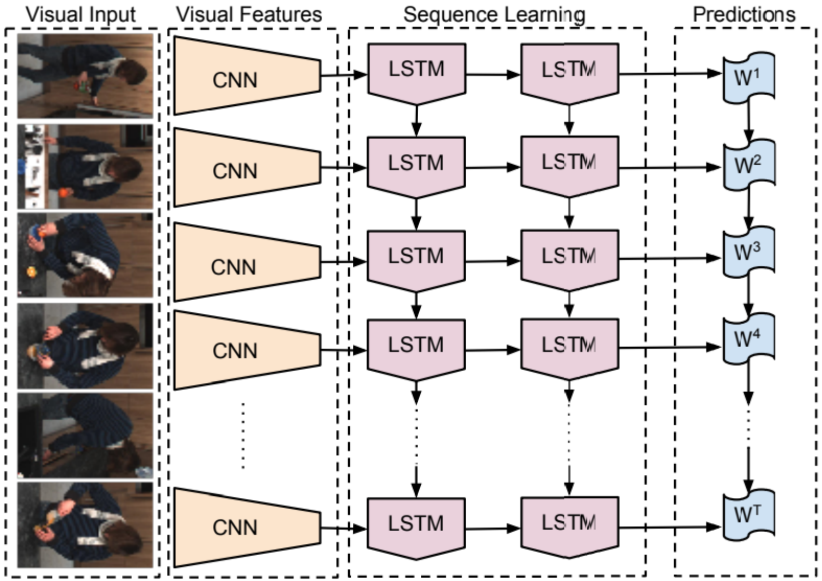
- Network with gates is a memory cell
  - Has internal recurrent connection
- Input, output, and forget gates
  - Act as read, write, and reset signals



Hochreiter and Schmidhuber Long Short-Term Memory (1997)

Extensions

- Combine with CNNs
  - Use image features as RNN inputs
- Deep RNNs



Donahue *et al*, Long-term Recurrent Convolutional Networks for Visual Recognition and Description (2015)

Character level language model

- Given a sequence of characters, predict next character
- Train on a large corpus of text
  - Shakespeare
  - Wikipedia
  - Math text (Latex)
  - C code

```
PANDARUS:

Alas, I think he shall be come approached and the day

When little srain would be attain'd into being never fed,

And who is but a chain and subjects of his death,

I should not sleep.


Second Senator:

They are away this miseries, produced upon my soul,

Breaking and strongly should be buried, when I perish

The earth and thoughts of many states..
```

Karpathy, [The Unreasonable Effectiveness of Recurrent Neural Networks]() (2015)

Character level language model

- Training time: target the next character in each step
- Test time: sample from output distribution and feed back as input in the next step

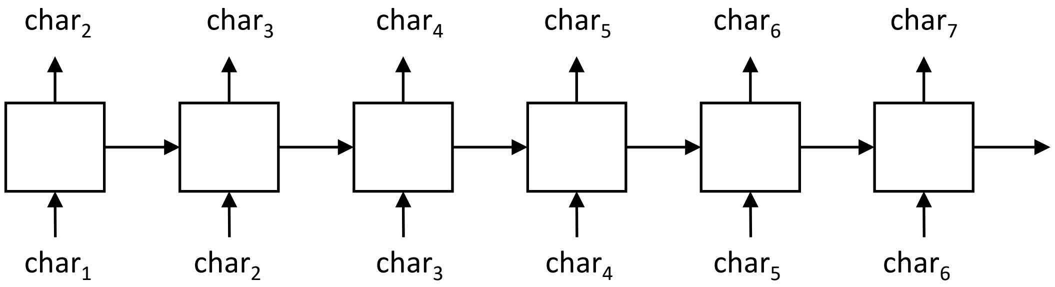


Image captioning

- Describe input image by a word sequence of arbitrary length

↓

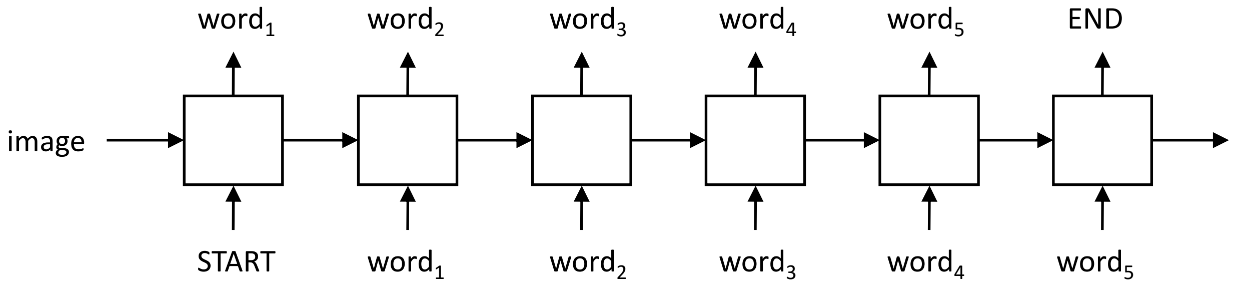
Two dogs are playing in the grass

Karpathy and Fei-Fei, Deep Visual-Semantic Alignments for Generating Image Descriptions (2015)

Donahue *et al*, Long-term Recurrent Convolutional Networks for Visual Recognition and Description (2015)

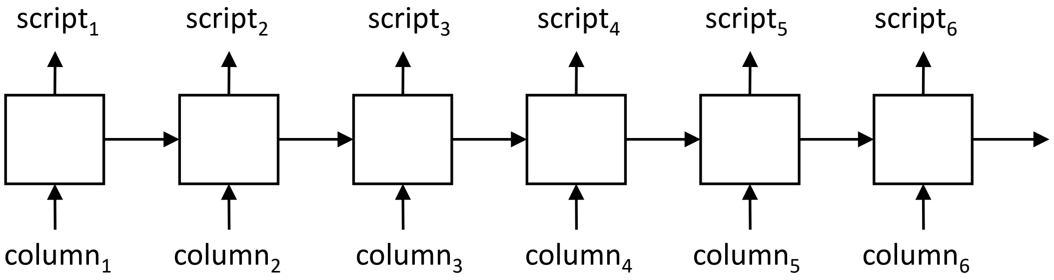
Image captioning

- Image is represented by CNN features
- Test time: sampling from output distribution until END is sampled
- Image can be passed as additional input at every step



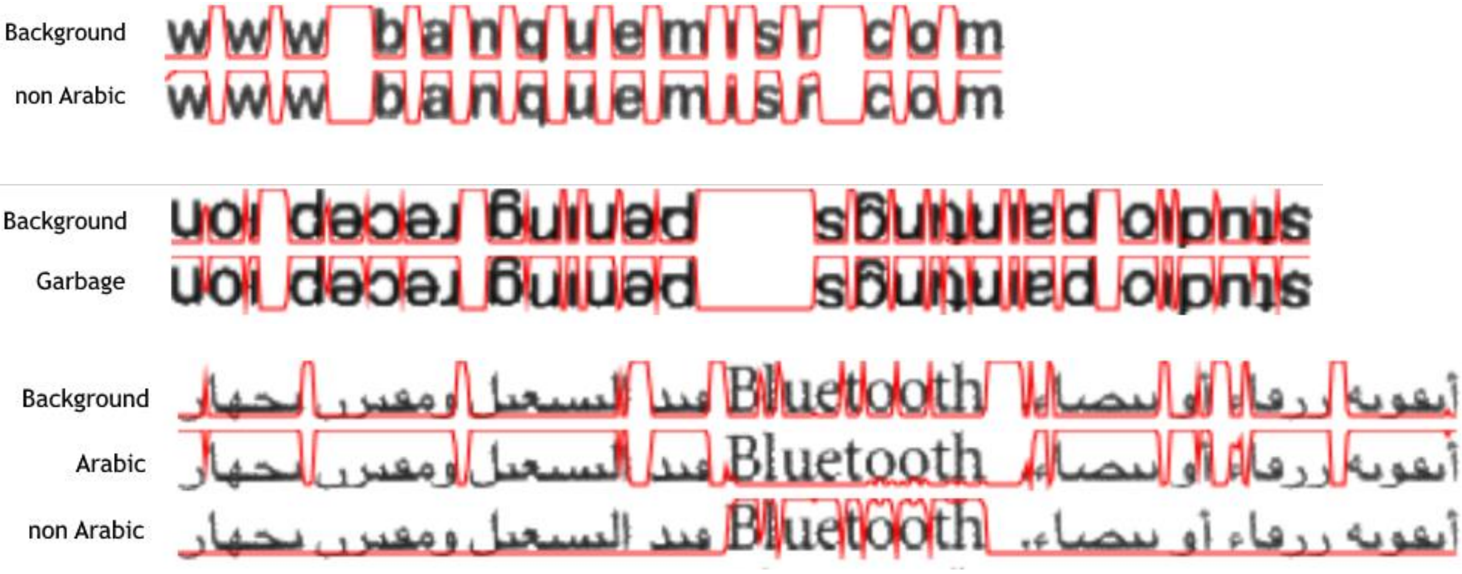
Arabic script detection

- Given an image of a line of printed text, classify each column as
  - Background
  - Non-Arabic
  - Arabic
  - Garbage



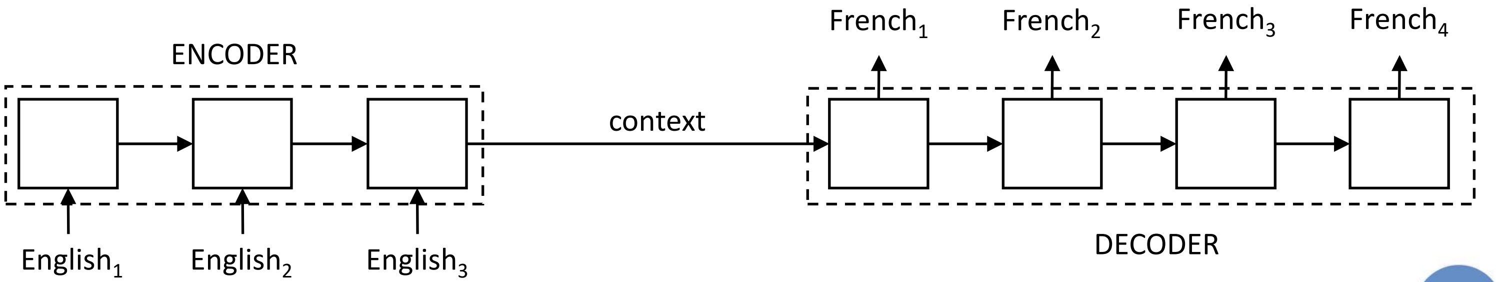
- 1-1 input/output alignment

Arabic script detection



Machine translation

- No input/output alignment
- Encoder-decoder (sequence-to-sequence) architecture
  - Encoder RNN consumes input, computes fixed-sized context (last state)
  - Decoder RNN generated output from context



Sutskever *et al*, Sequence to Sequence Learning with Neural Networks (2014)

Cho *et al*, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation (2014)

Machine translation

- Context can be passed to decoder as input in every step
- Encoder and decoder may or may not share weights
  - Language specific encoder/decoder
- Reversing order of source words
  - Create more short-range connections

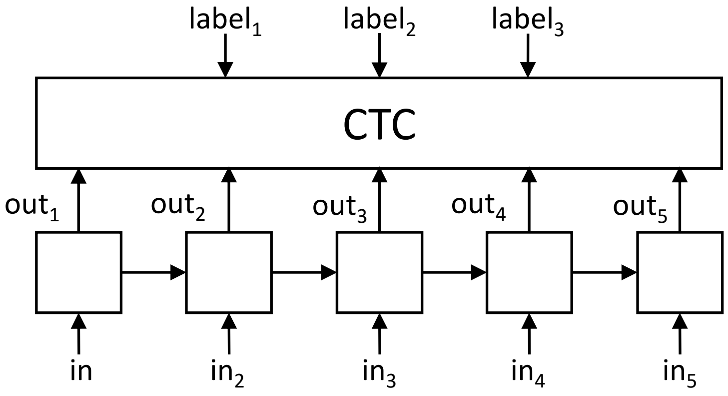
Line OCR

- Recognize text from an image of one line of text
- No alignment, due to lack of labeling
  - No explicit word/char segmentation

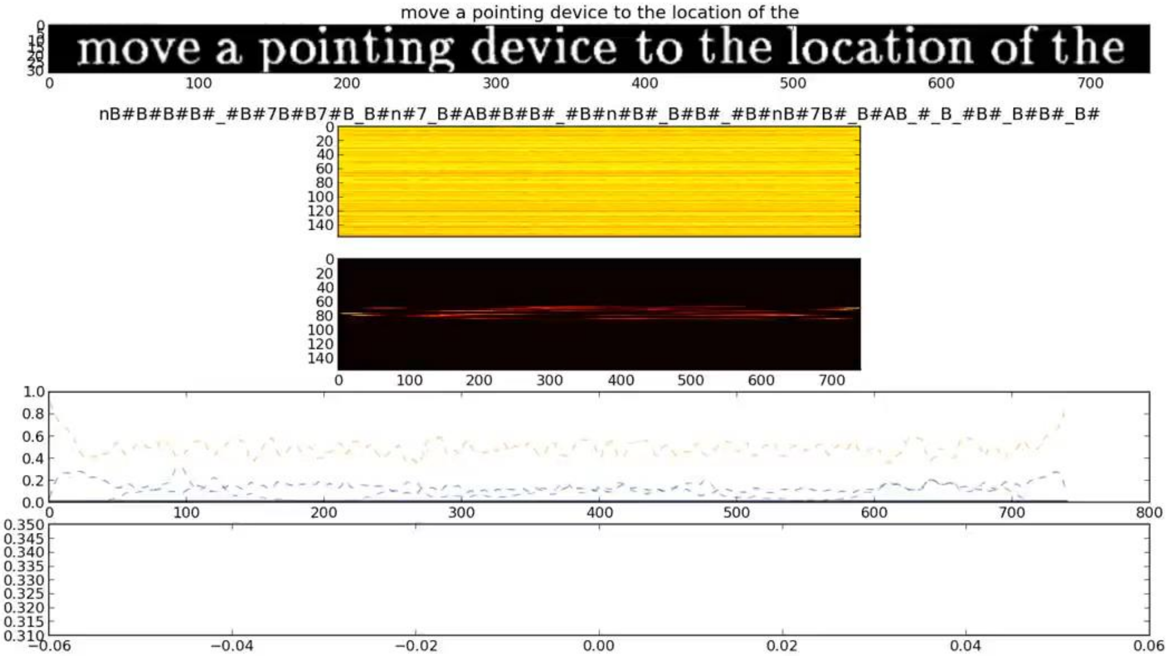
Ul-Hasan *et al*, [High-Performance OCR for Printed English and Fraktur using LSTM Networks]

Line OCR

- Network makes framewise predictions, repeating labels
- At runtime: decode by removing repetitions
- At training time: connectionist temporal classification (CTC)
  - Loss layer that takes the decoding process into account



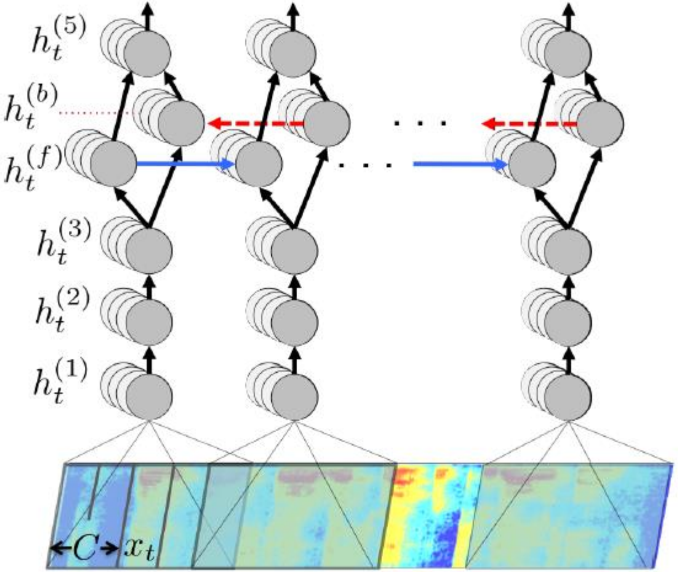
Graves *et al*, Graves et al. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks (2006)



https://www.youtube.com/watch?v=czG5Jk9iC7c

Speech recognition

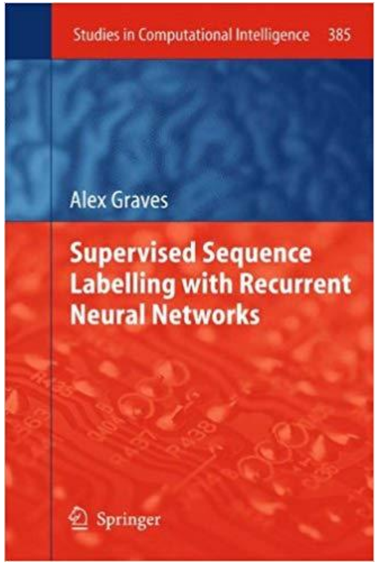
- No input/output alignment
  - Uses CTC
- Input is a spectrogram
- Deep RNN with heterogeneous layers
  - Layers 1, 2, 3, 5 are non-recurrent
  - Layer 4 is bidirectional
- Computational efficiency
  - Recurrent layers are harder to parallelize
  - Does not use LSTM

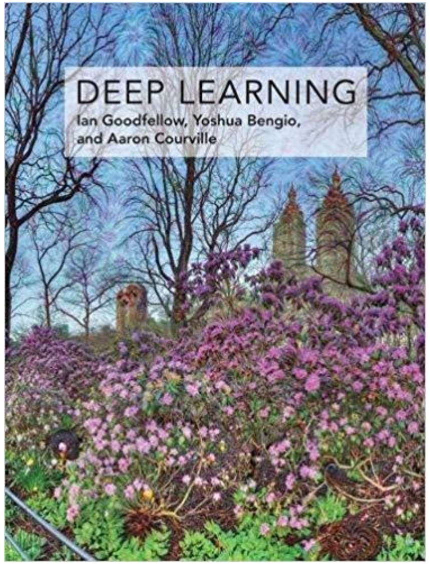


Hannun *et al*, [Deep Speech: Scaling up end-to-end speech recognition]() (2014)

Literature

- [Stanford CS231N] (lecture 10) lecture notes/videos
- Supervised sequence labeling with Recurrent Neural Networks
  Alex Graves
- Deep Learning Book (chapter 10)
  Ian Goodfellow, Yoshua Bengio, Aaron Courville





Thanks

- Questions?