Recurrent Neural Networks

Momčilo Vasilijević

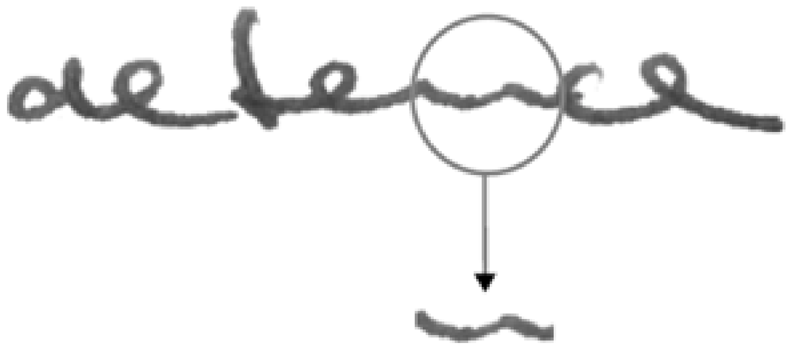
Microsoft | Development Center Serbia

MACHINE LEARNING
PETNICA SUMMER INSTITUTE

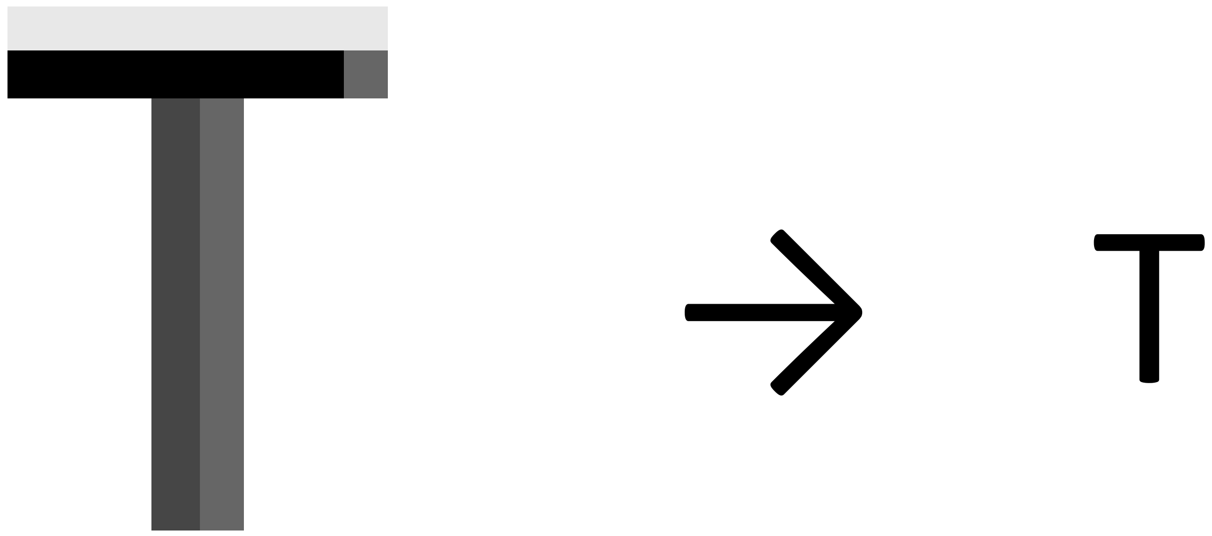
PETNICA Science Center

Introduction

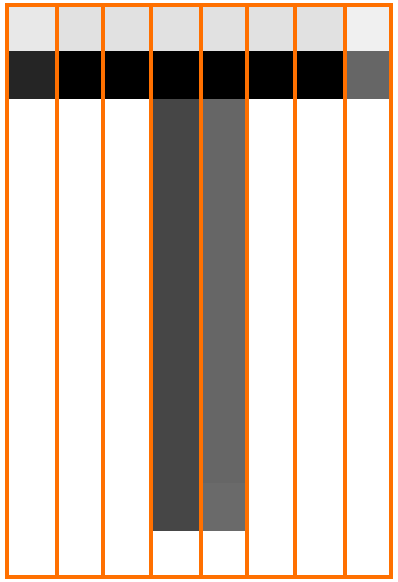
- *Sequence labeling* – process of transcribing data sequence to sequence of discrete labels
- Applications
  - Speech recognition
  - Handwriting recognition
  - Protein secondary structure prediction
- Sequence labeling vs. pattern classification
  - Correlations in input data and output data



Pattern classification

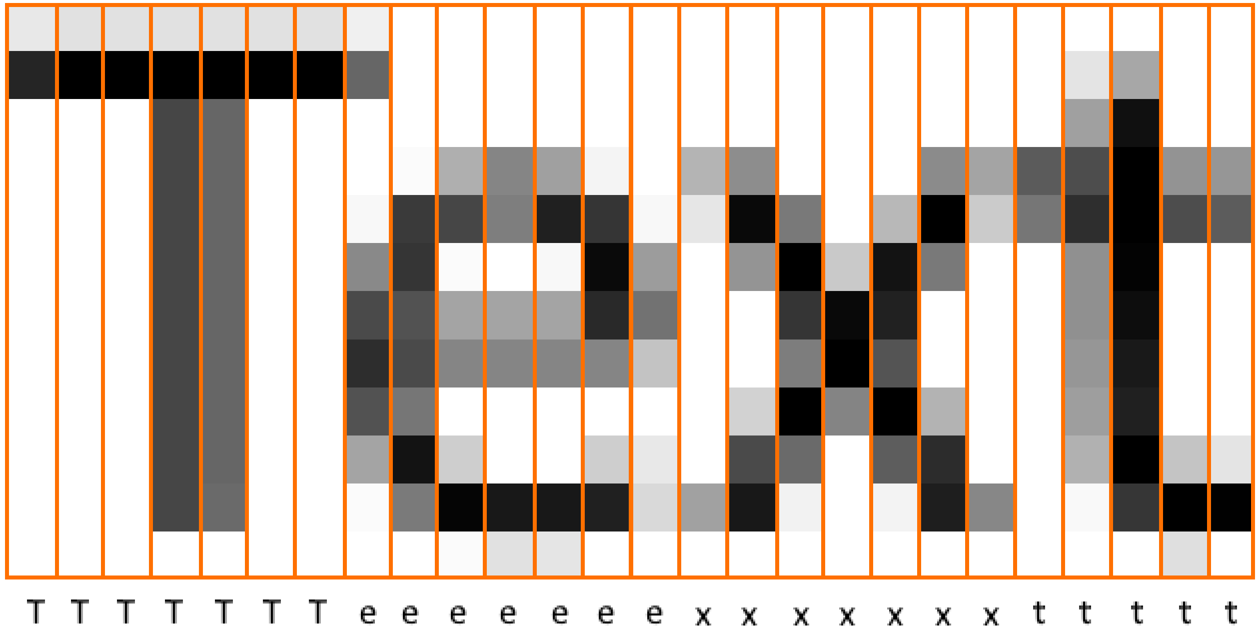


Sequence classification



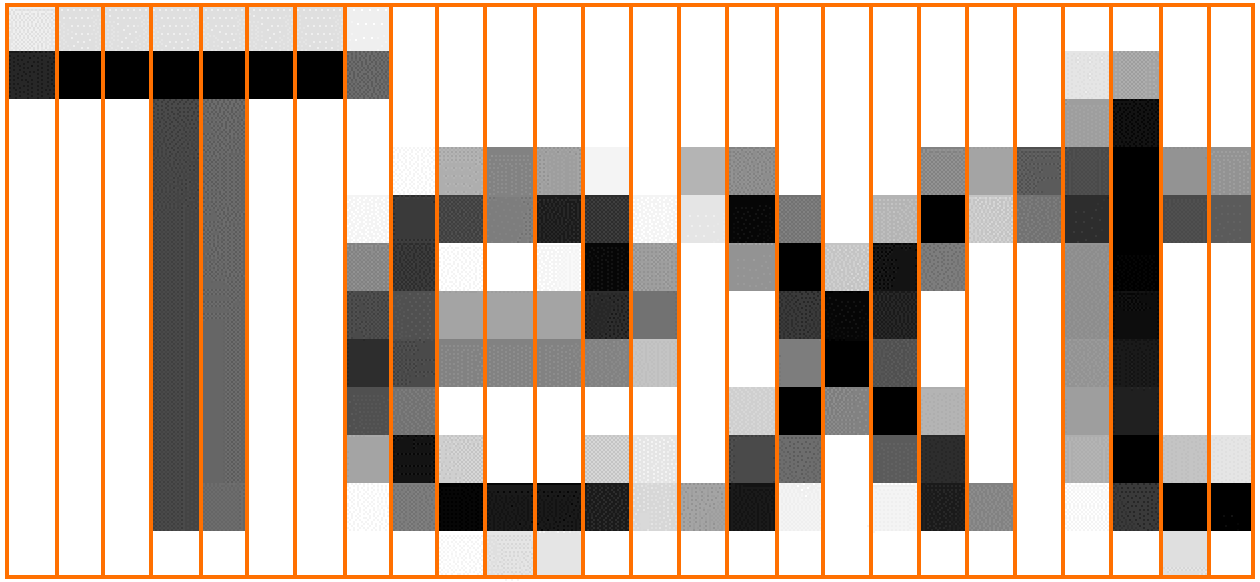
→ T

Segment classification



- Frame-wise labels
- Context
- Time windows

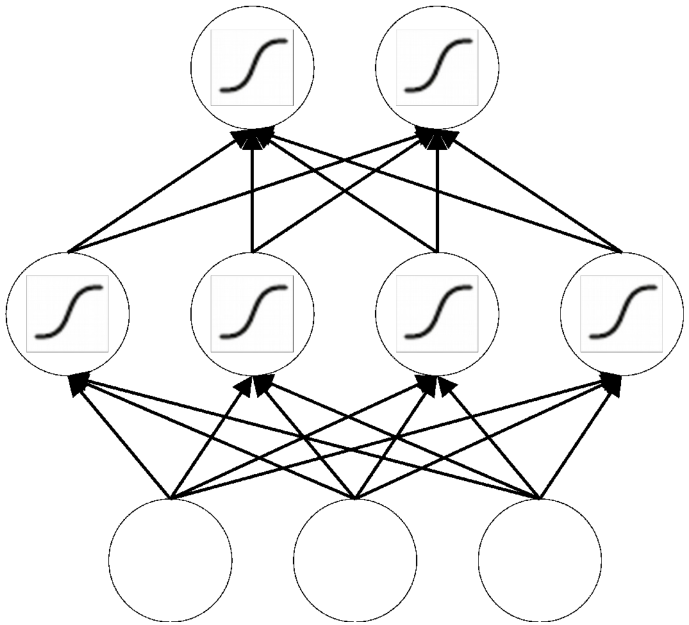
Temporal classification

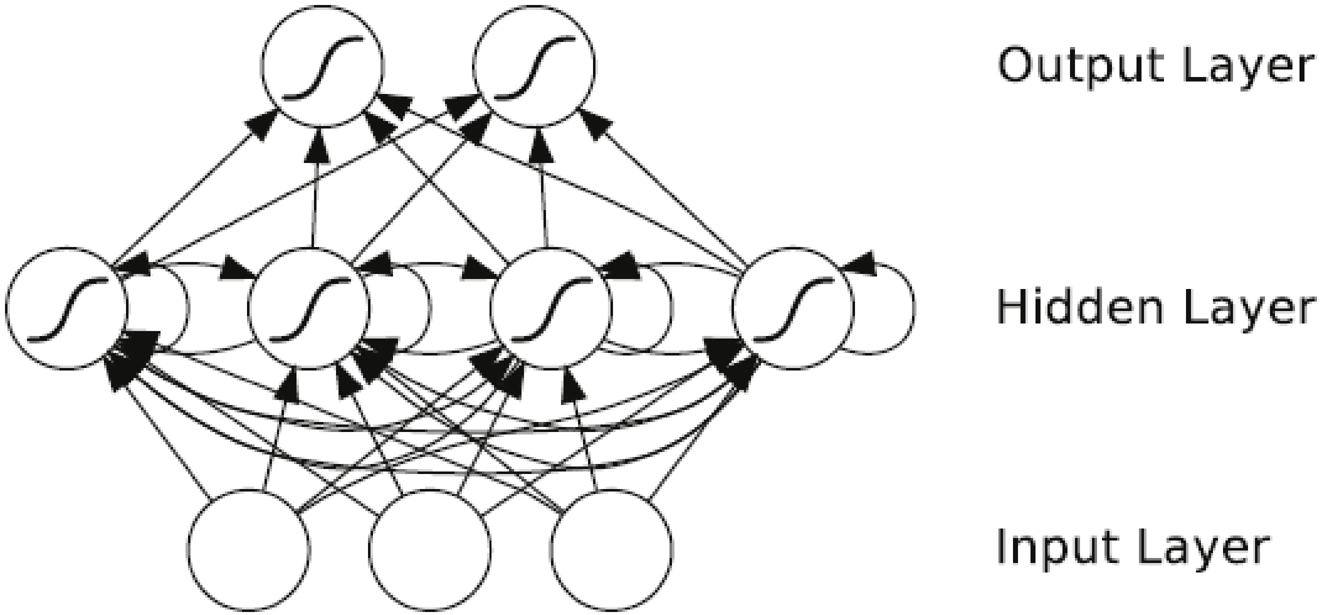


→ Text

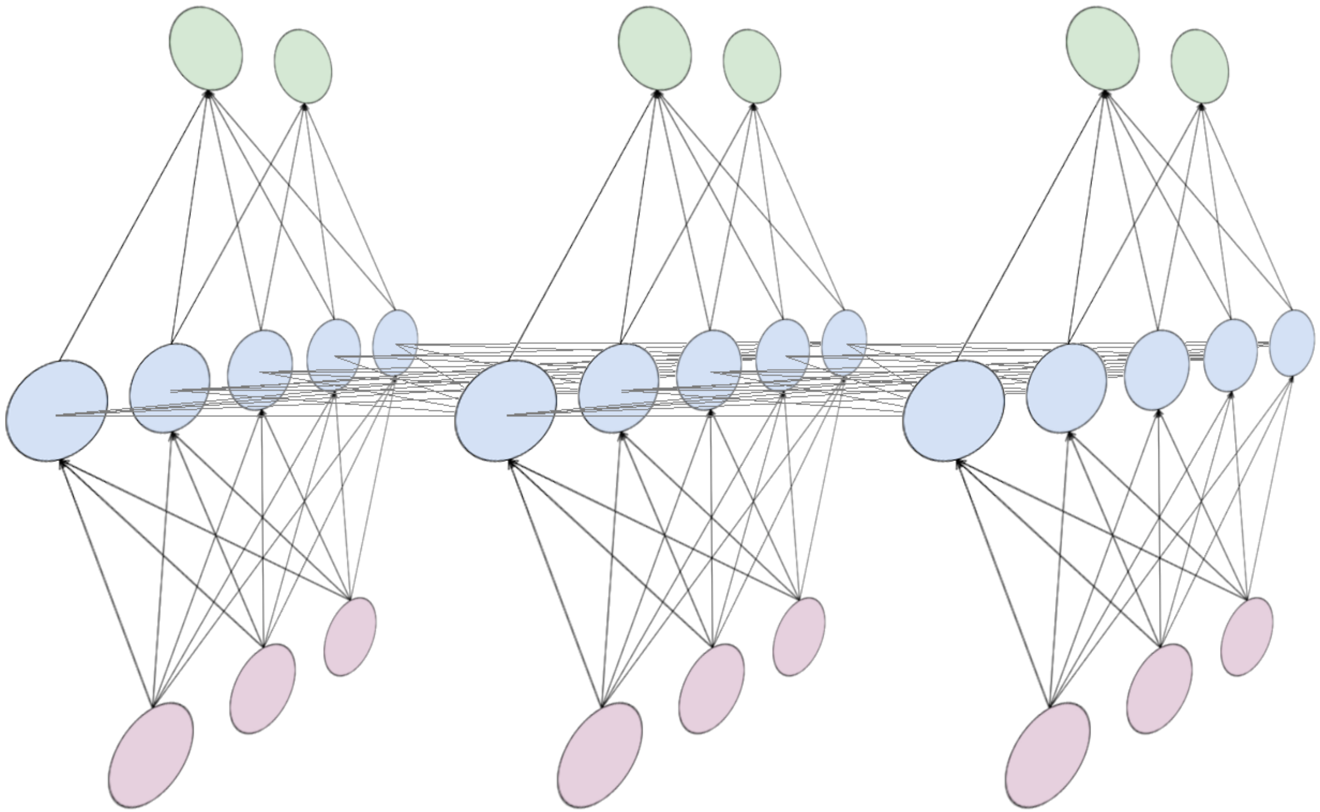
- Unsegmented labels

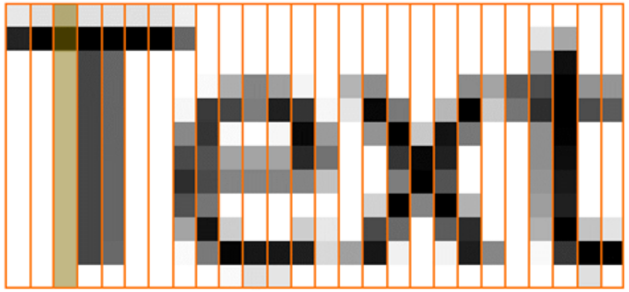
Regular vs recurrent network



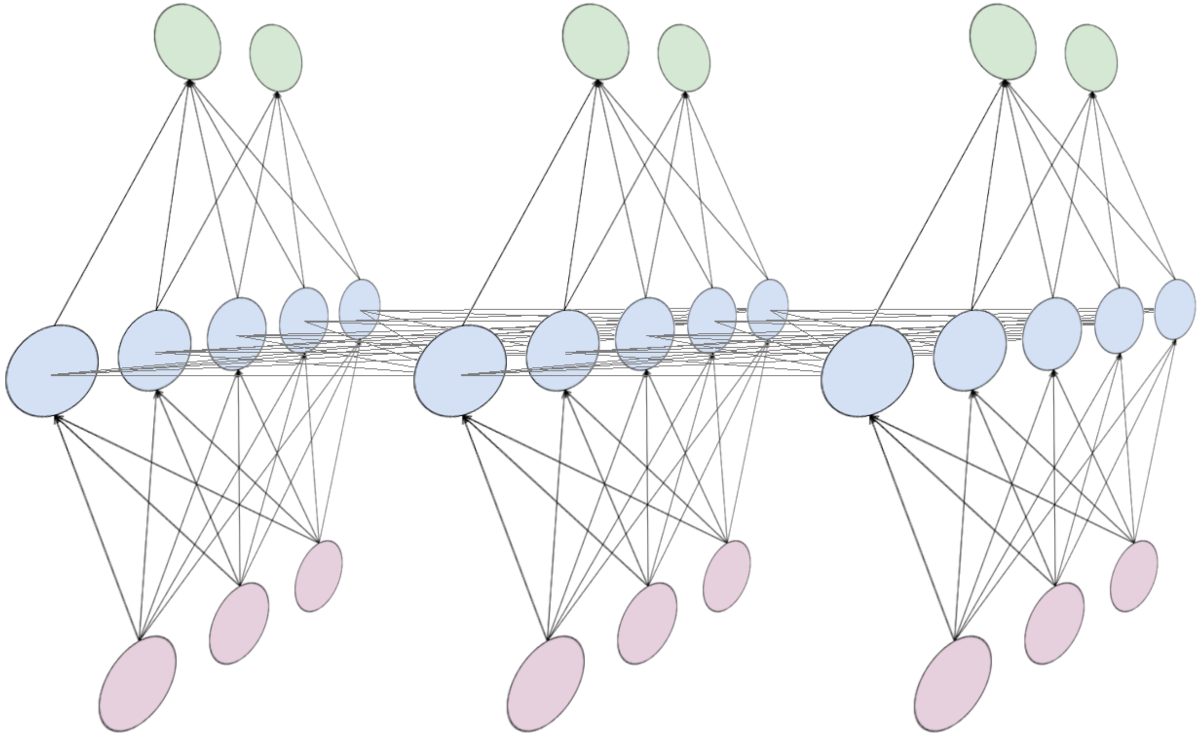


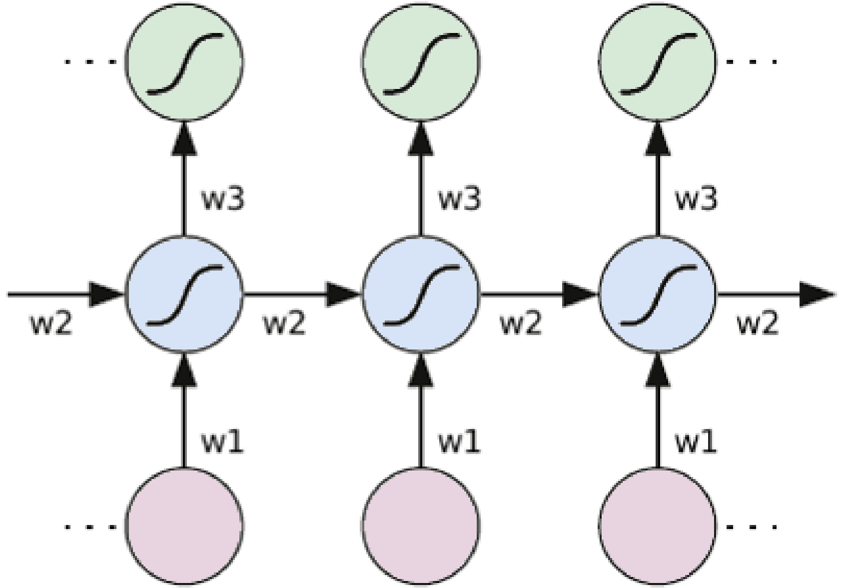
How RNNs work?





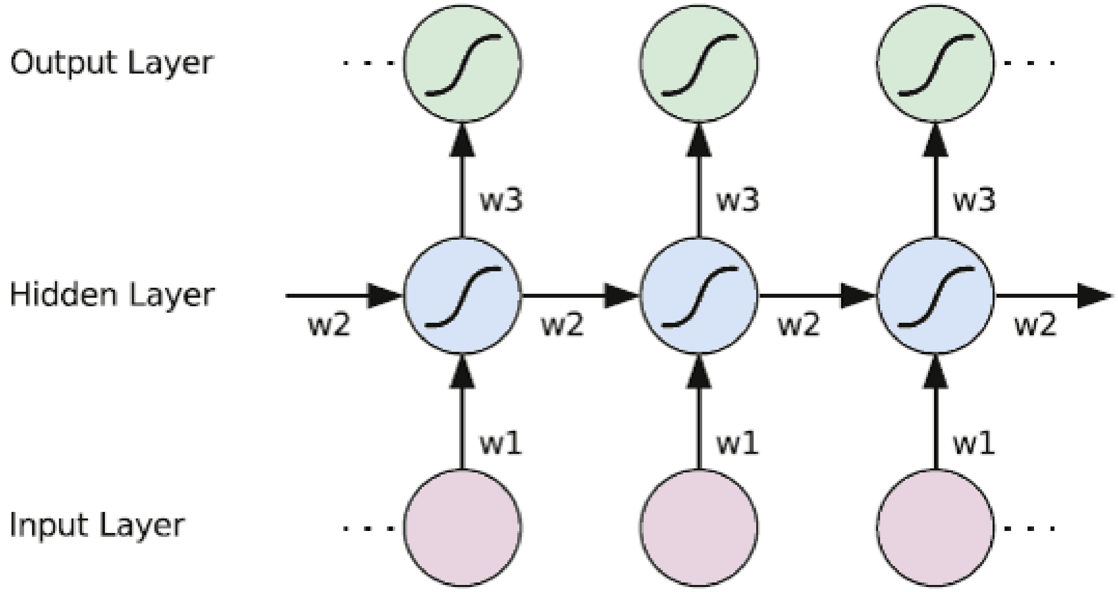
Unfolding RNN





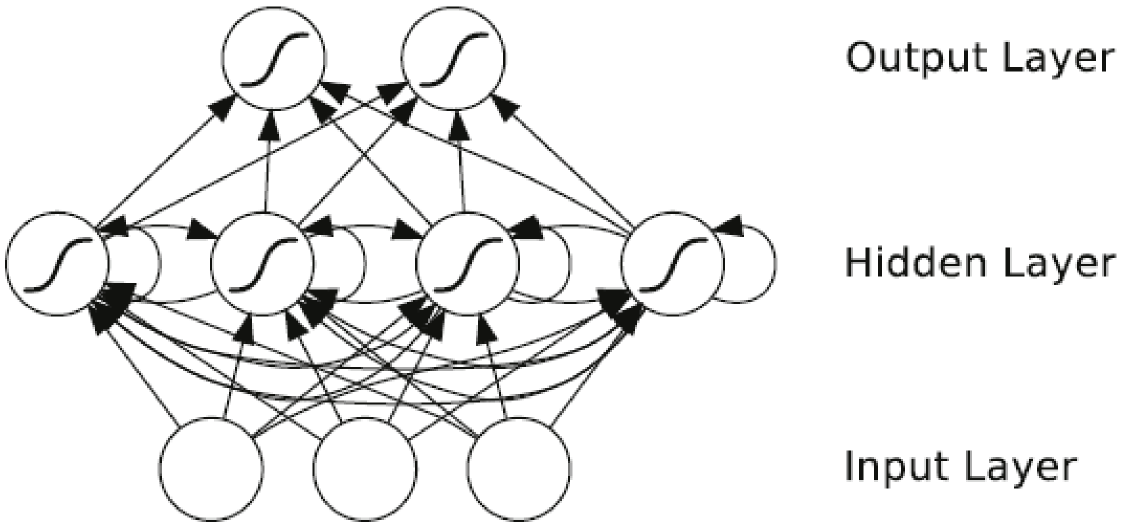
Unfolding RNN

- Unfolding network along input sequence
- No recurrent connections



Recurrent Neural Network (RNN)

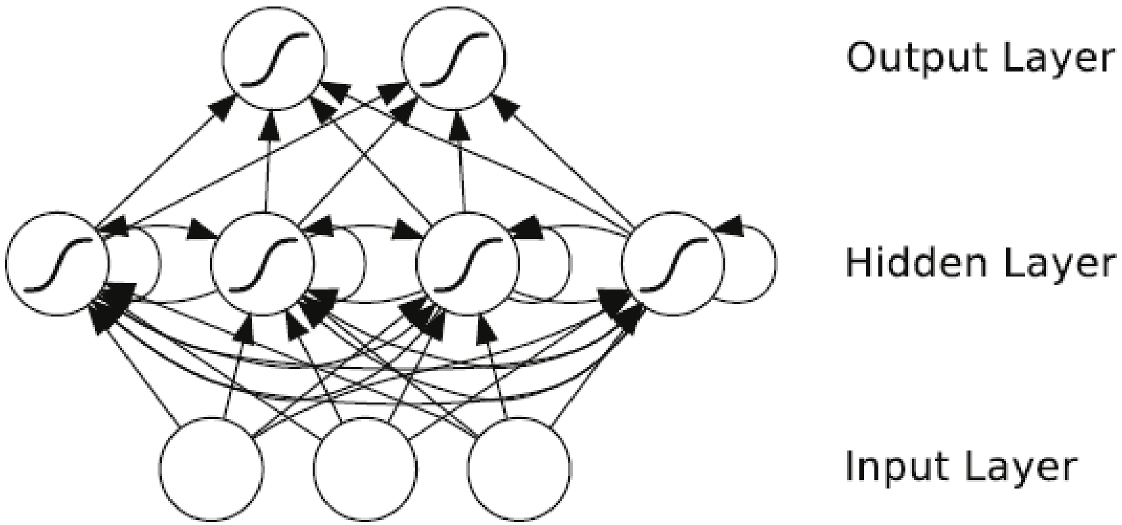
- MLP maps input vector to output vector
- Recurrent connections allow ‘memory’ of previous inputs
- RNN maps entire history of previous inputs to output vector



Forward pass

- Almost the same as MLP, except inputs come from the hidden layer as well
  - $a_h^t = \sum_{i=1}^{I} w_{ih} x_i^t + \sum_{h'=1}^{H} w_{h'h} b_{h'}^{t-1}$
  - $b_h^t = \theta_h(a_h^t)$
- a_h^t - input to hidden unit h at time t
- b_h^t - output of hidden unit h at time t

Almost the same as MLP, except inputs come from the hidden layer as well
- input to hidden unit h at time t



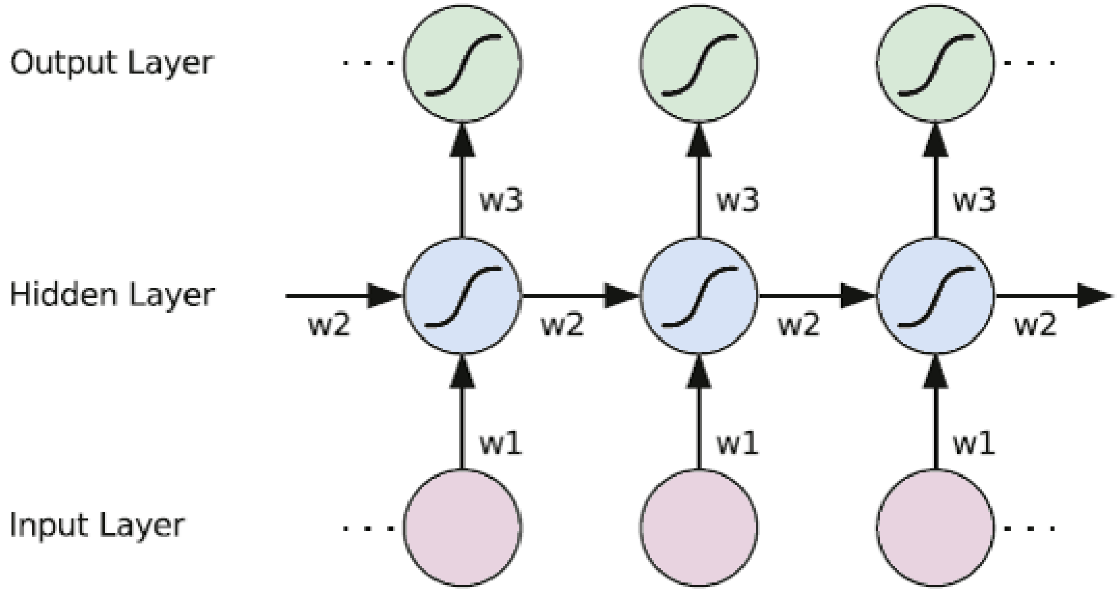
Backward pass

- Backpropagation through time (BPTT)

Backpropagation through time (BPTT)

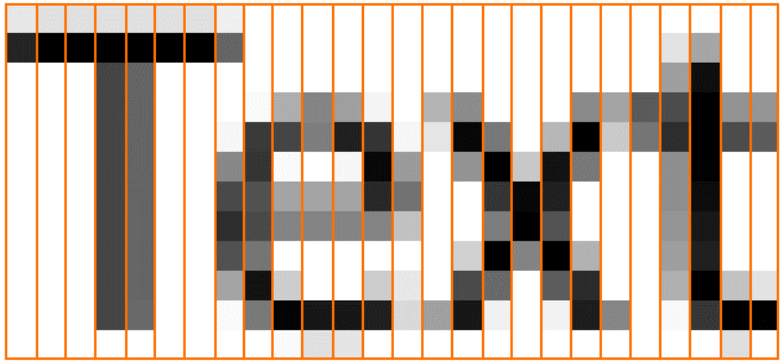
- $\delta_h^t = \theta'(a_h^t)\left(\sum_{k=1}^{K} w_{hk}\delta_k^t + \boldsymbol{\sum_{h'=1}^{H} w_{hh'}\delta_{h'}^{t+1}}\right)$

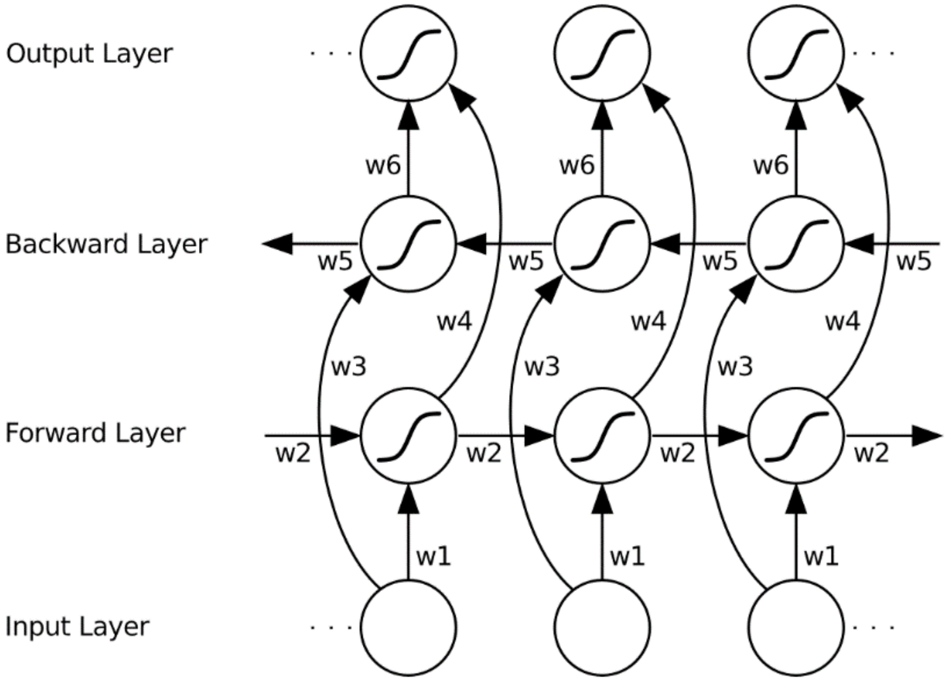
- $\frac{\partial L}{\partial w_{ij}} = \sum_{t=1}^{T} \frac{\partial L}{\partial a_j^t}\frac{\partial a_j^t}{\partial w_{ij}} = \sum_{t=1}^{T} \delta_j^t b_i^t$



Bidirectional network

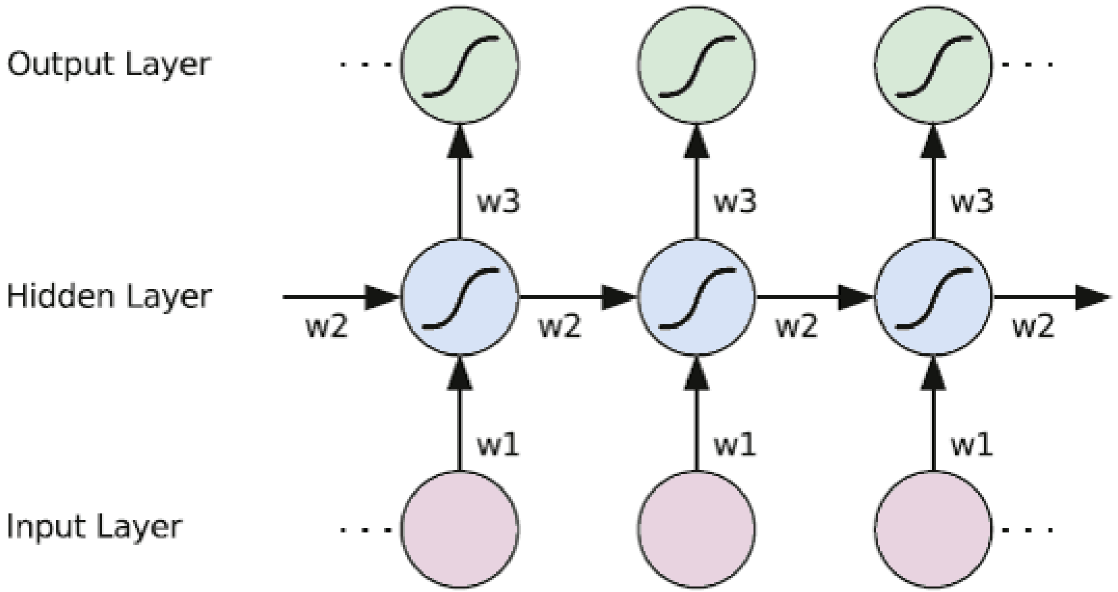
- Context from past and context from future
- In handwriting it is useful to know letters before and letters coming after

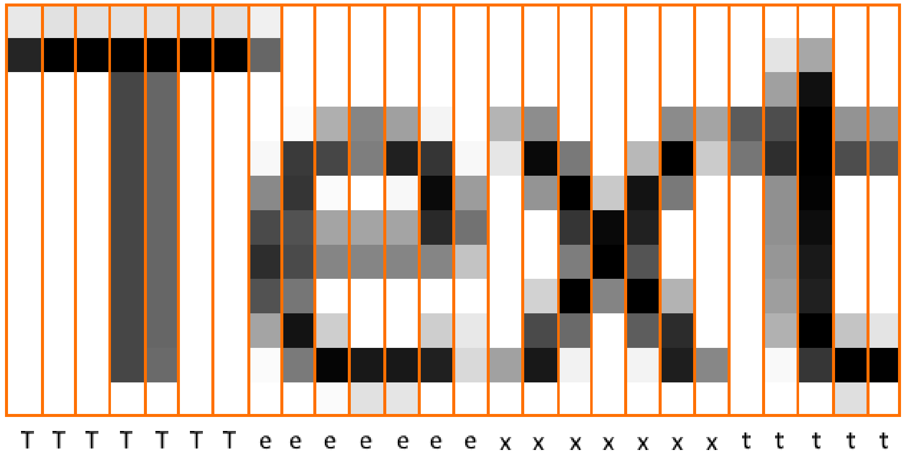




Framewise labels

- Labeling each frame is expensive
- In some cases (e.g. speech recognition), you don't know where one label finishes and where the other starts
- Connectionist temporal classification (CTC)



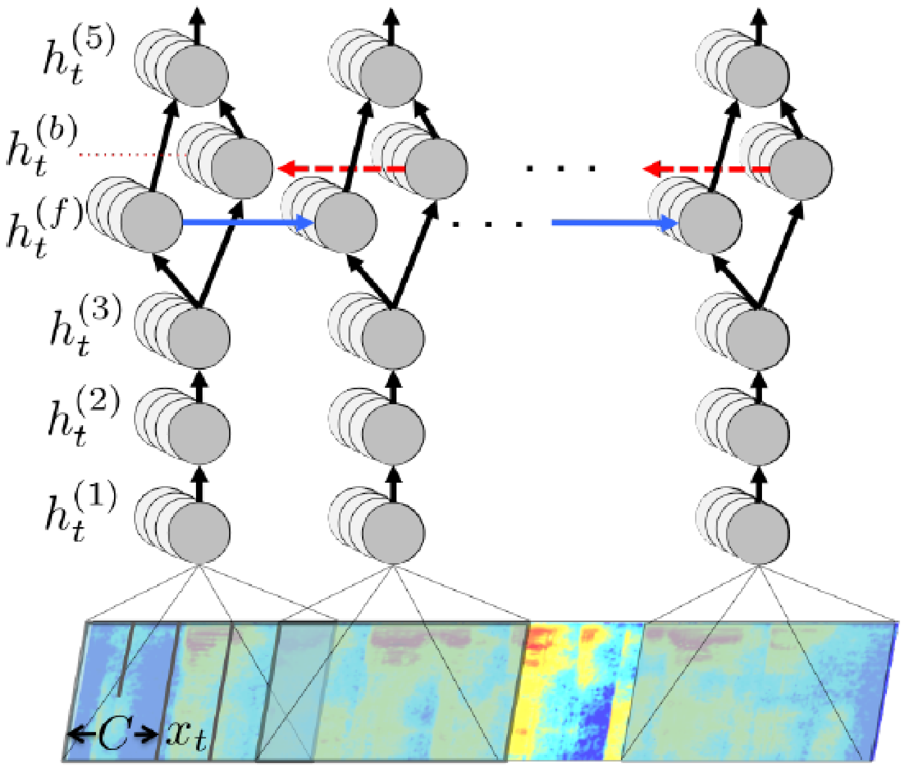


Speech recognition

- [“Deep speech”]

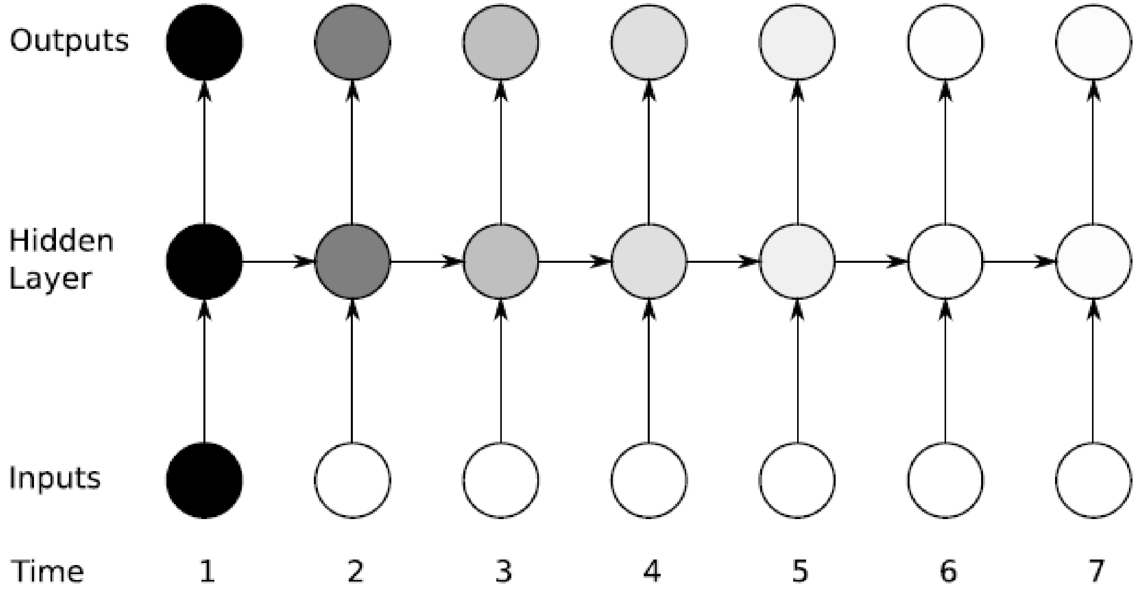
- Bidirectional RNN
- Feature window
- Unaligned data – positions of outputs are unknown
- Connectionist temporal classification (CTC)

- Set of novel data synthesis techniques
- Large amount of training data



RNN problem

- Sensitivity decays over time
- New inputs overwrite activations of the hidden layer
- Darker the shade, greater the sensitivity



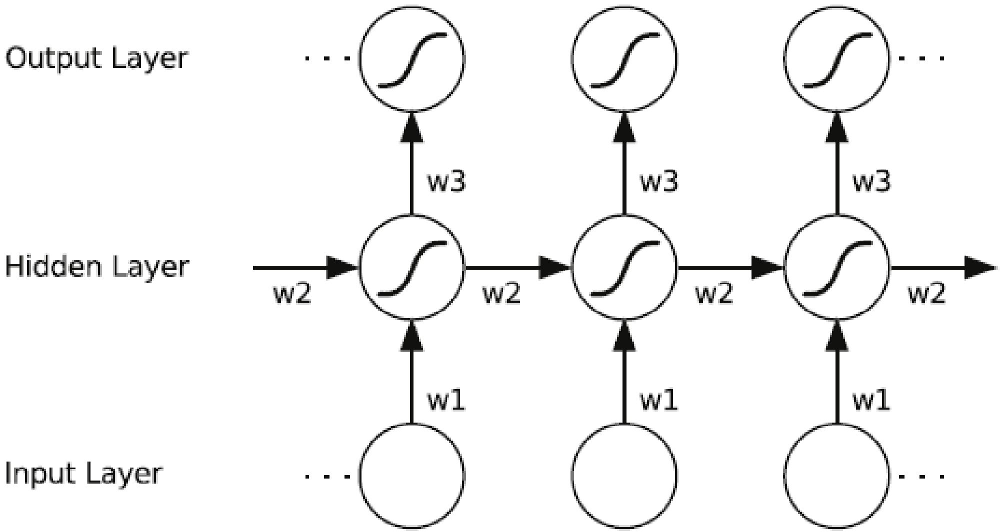
Training RNN

- Very difficult to train
- Limited range of context to access
- Vanishing and exploding gradient
- Influence of error from timestamp *T+N*

$$\delta_h^t[\delta^{t+N}] = \theta'(a_h^{t+1})w_2 \cdot \ldots \cdot \theta'(a_h^{t+N})w_2\delta_h^{t+N}$$
$$= \delta_h^{t+N} w_2^N \prod_{i=1}^{N} \theta'(a_h^{t+1})$$

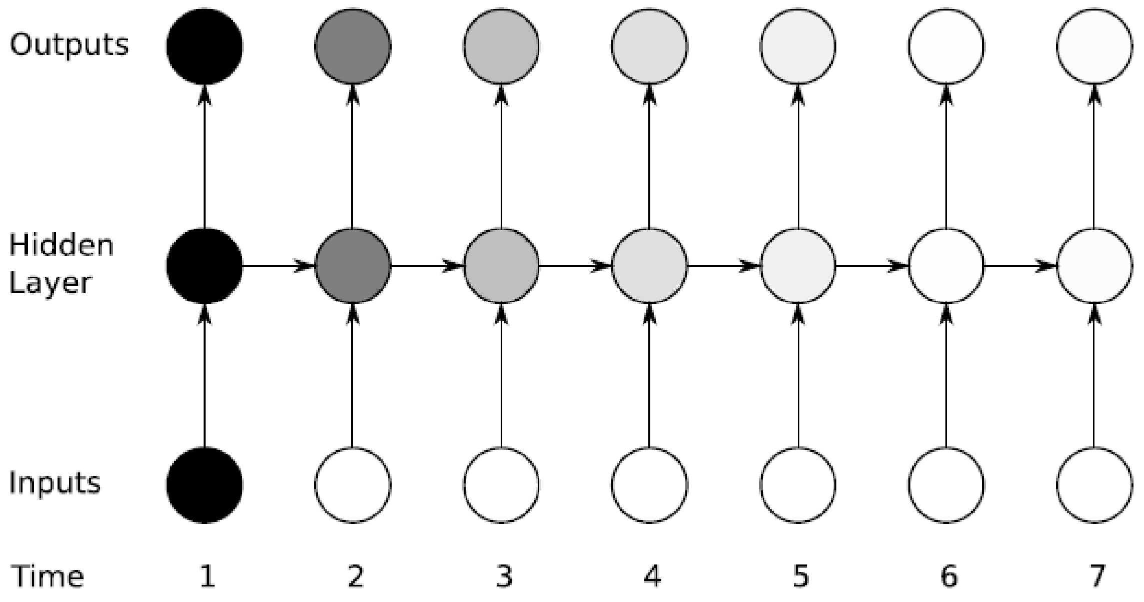
- LSTM – long short term memory

- Very difficult to train
- Limited range of context to access
- Vanishing and exploding gradient
- Influence of error from timestamp *T+N*
- LSTM – long short term memory



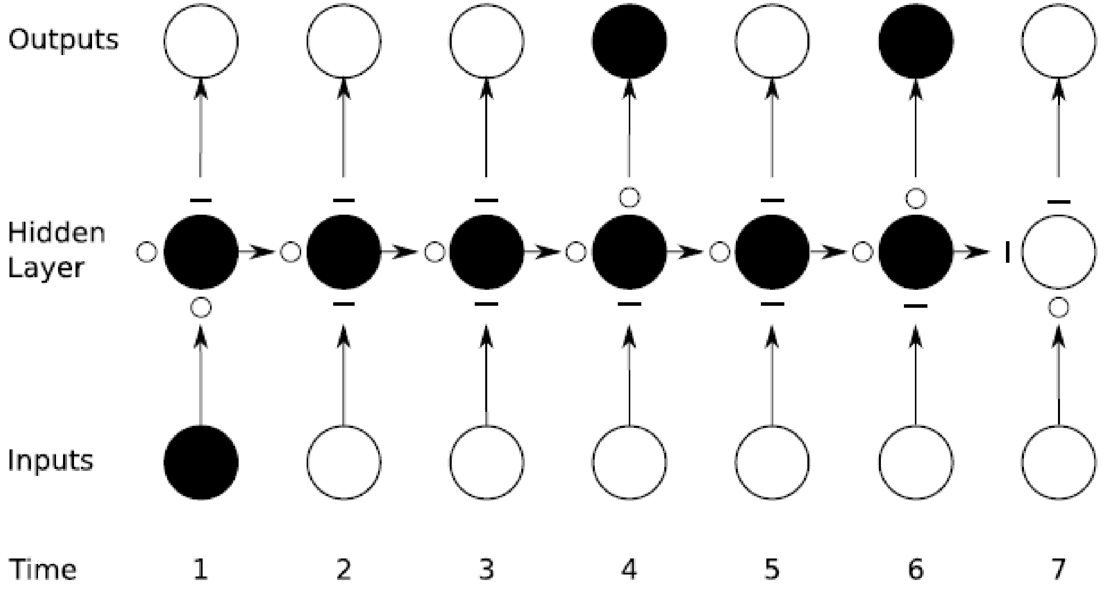
RNN problem

- Sensitivity decays over time
- New inputs overwrite activations of the hidden layer
- Darker the shade, greater the sensitivity

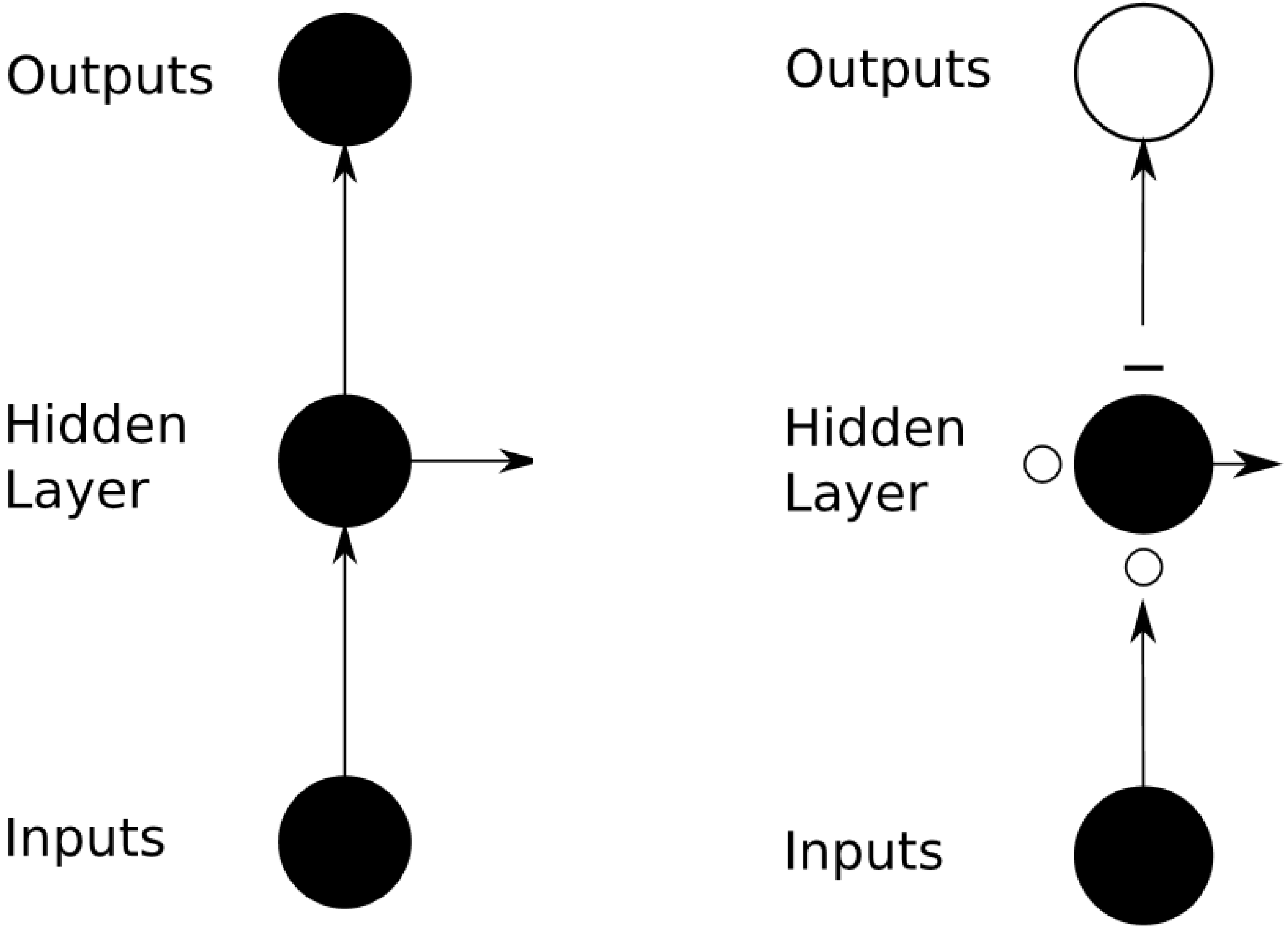


Gates

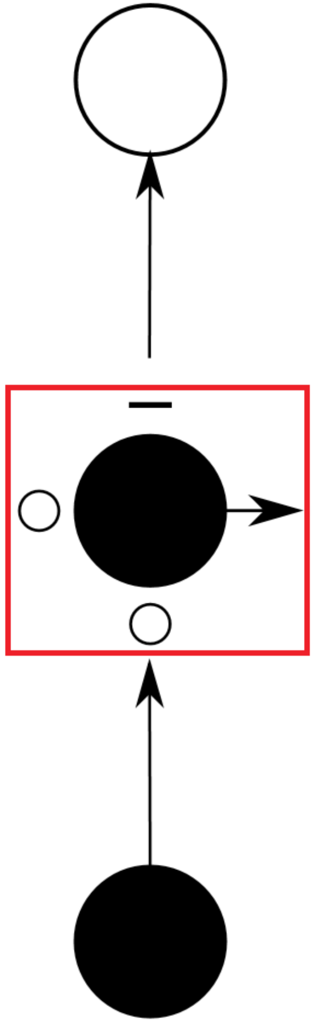
- Information is preserved as long as input is closed and forget gate is opened
  - 'o' – gate is open
  - '-' – gate is closed

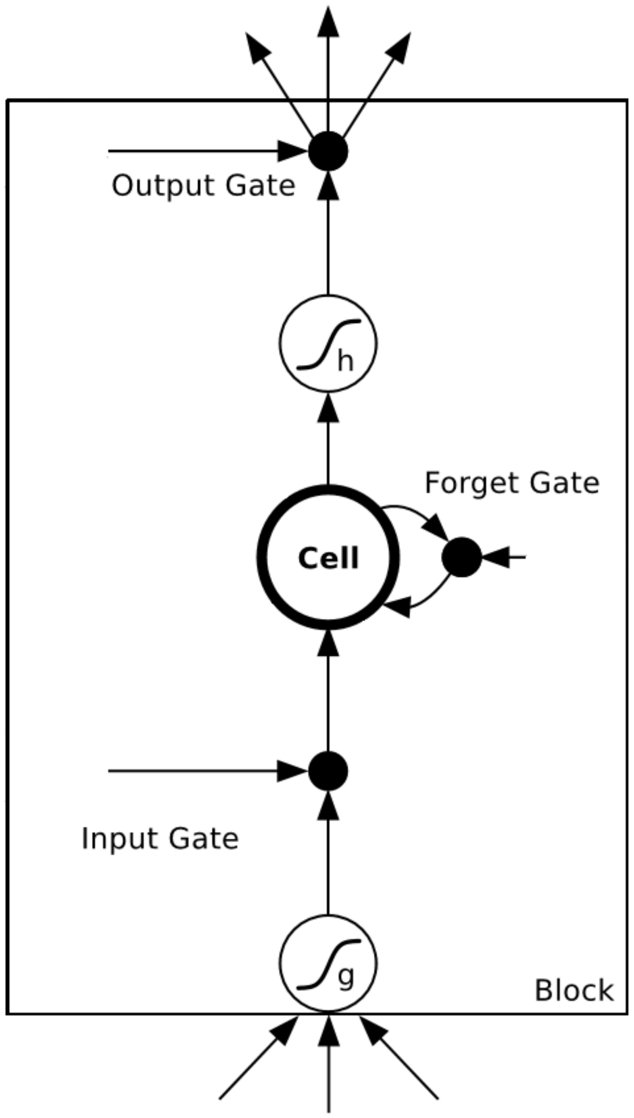


Hidden units



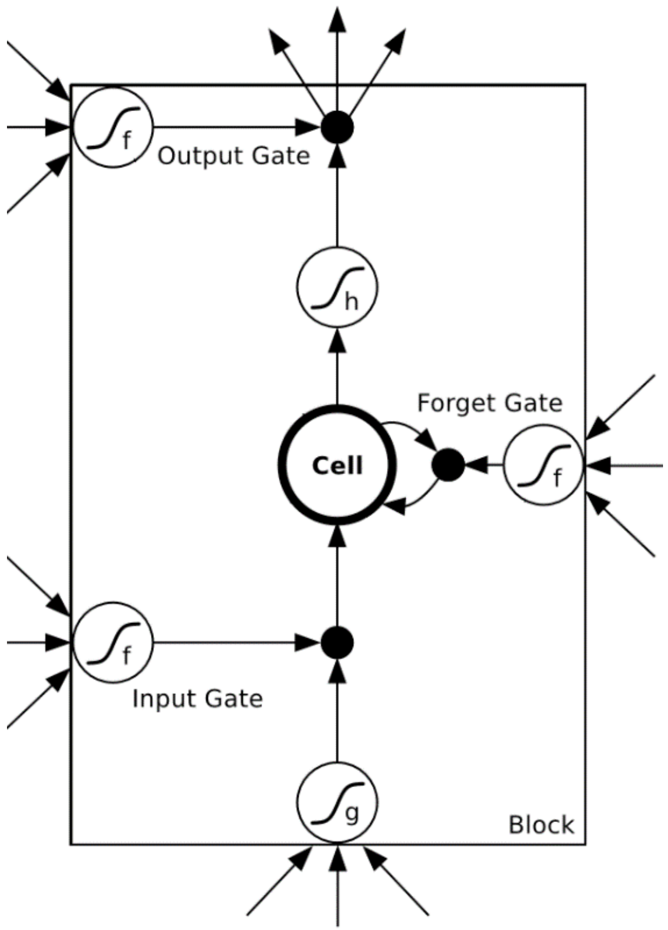
LTSM cell





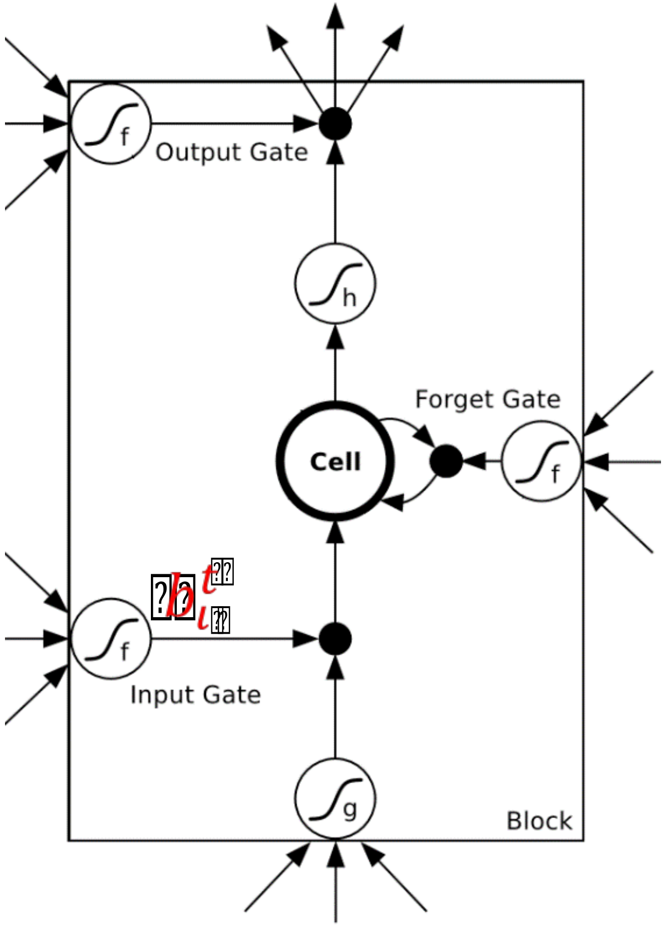
LSTM – forward pass

- Input gate ι
- Input gate
  - $a_\iota^t = \sum_{i=1}^{I} w_{i\iota} x_i^t + \sum_{h=1}^{H} w_{h\iota} b_h^{t-1}$
  - $b_\iota^t = f(a_\iota^t)$
- Forget gate ϕ
- Forget gate
  - $a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{h=1}^{H} w_{h\phi} b_h^{t-1}$
  - $b_\phi^t = f(a_\phi^t)$
- Cell c
- Cell
  - $a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1}$
  - $s_c^t = b_\iota^t g(a_c^t) + b_\phi^t s_c^{t-1}$



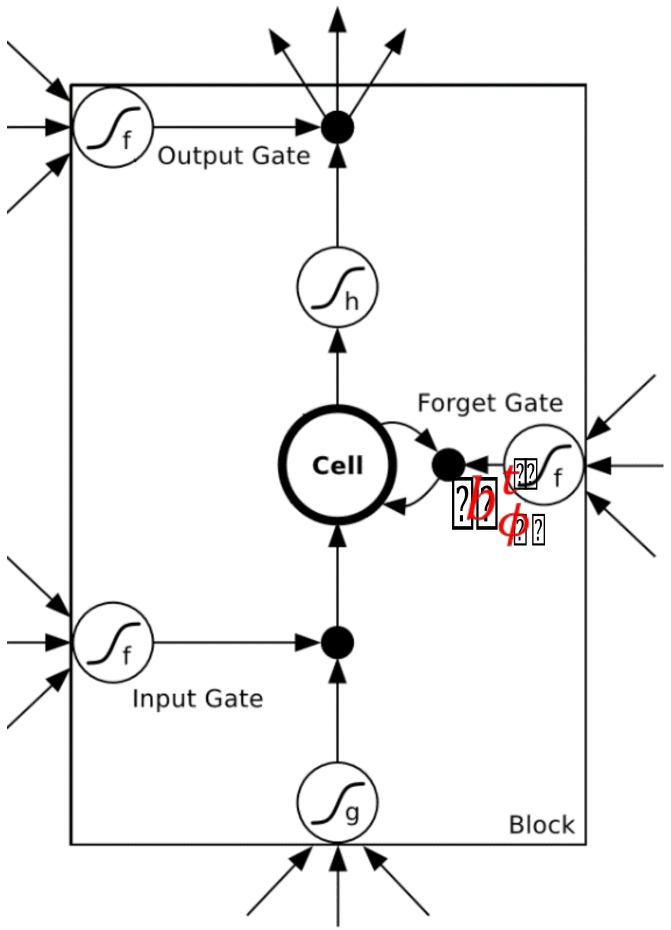
LSTM – forward pass

- Input gate ι
  - $a_\iota^t = \sum_{i=1}^{I} w_{i\iota} x_i^t + \sum_{h=1}^{H} w_{h\iota} b_h^{t-1}$
  - $b_\iota^t = f(a_\iota^t)$
- Forget gate ϕ
  - $a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{h=1}^{H} w_{h\phi} b_h^{t-1}$
  - $b_\phi^t = f(a_\phi^t)$
- Cell c
  - $a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1}$
  - $s_c^t = b_\iota^t g(a_c^t) + b_\phi^t s_c^{t-1}$



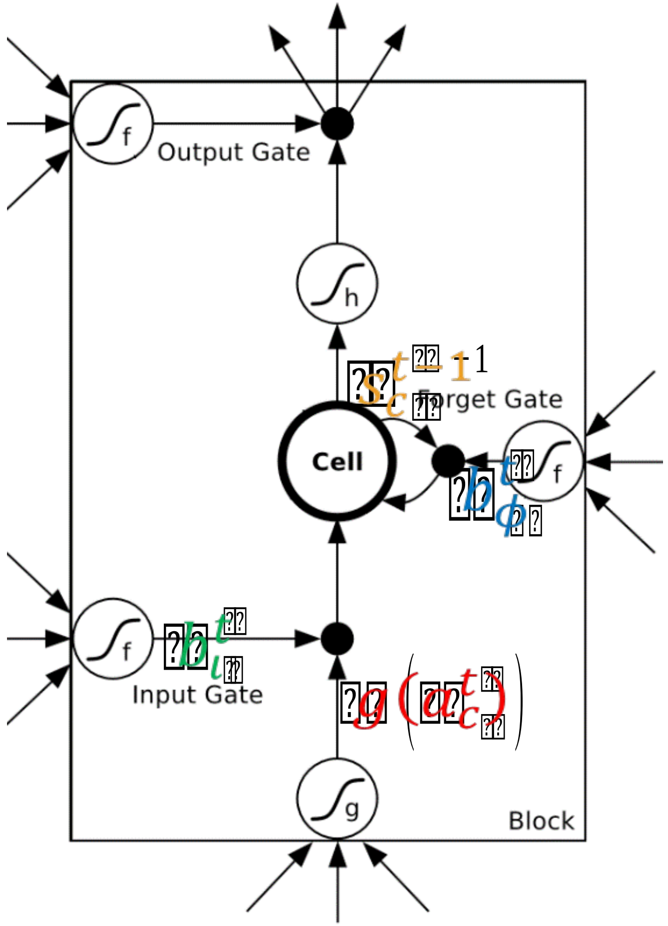
LSTM – forward pass

- Input gate ι
- Input gate
  - $a_\iota^t = \sum_{i=1}^{I} w_{i\iota} x_i^t + \sum_{h=1}^{H} w_{h\iota} b_h^{t-1}$
  - $b_\iota^t = f(a_\iota^t)$
- Forget gate ϕ
- Forget gate
  - $a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{h=1}^{H} w_{h\phi} b_h^{t-1}$
  - $b_\phi^t = f(a_\phi^t)$
- Cell c
- Cell
  - $a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1}$
  - $s_c^t = b_\iota^t g(a_c^t) + b_\phi^t s_c^{t-1}$



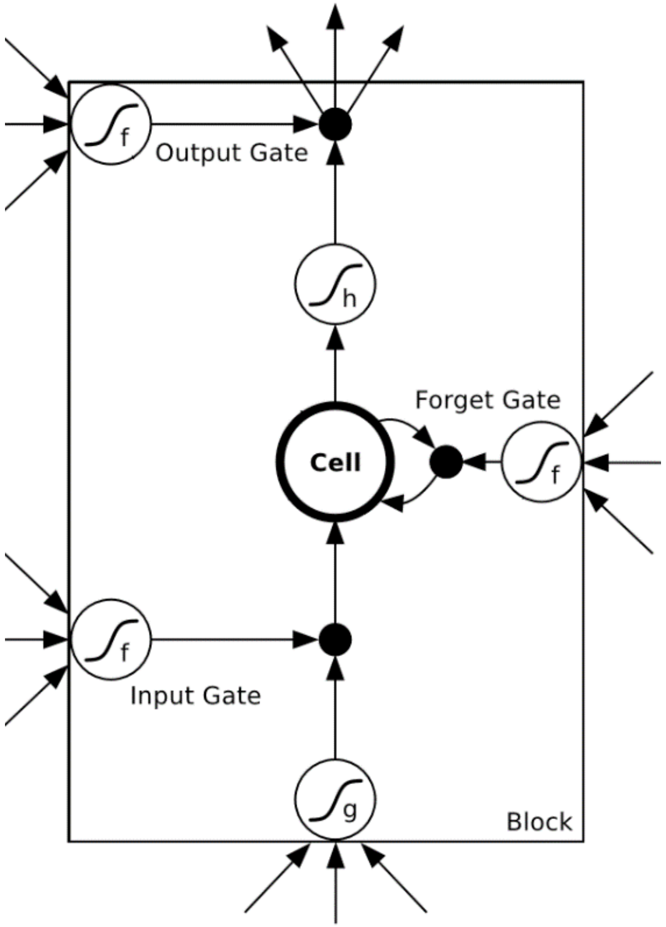
LSTM – forward pass

- Input gate ι
- Input gate
  - $a_\iota^t = \sum_{i=1}^{I} w_{i\iota} x_i^t + \sum_{h=1}^{H} w_{h\iota} b_h^{t-1}$
  - $b_\iota^t = f(a_\iota^t)$
- Forget gate ϕ
- Forget gate
  - $a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{h=1}^{H} w_{h\phi} b_h^{t-1}$
  - $b_\phi^t = f(a_\phi^t)$
- Cell c
- Cell
  - $a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1}$
  - $s_c^t = b_\iota^t g(a_c^t) + b_\phi^t s_c^{t-1}$



LSTM – forward pass

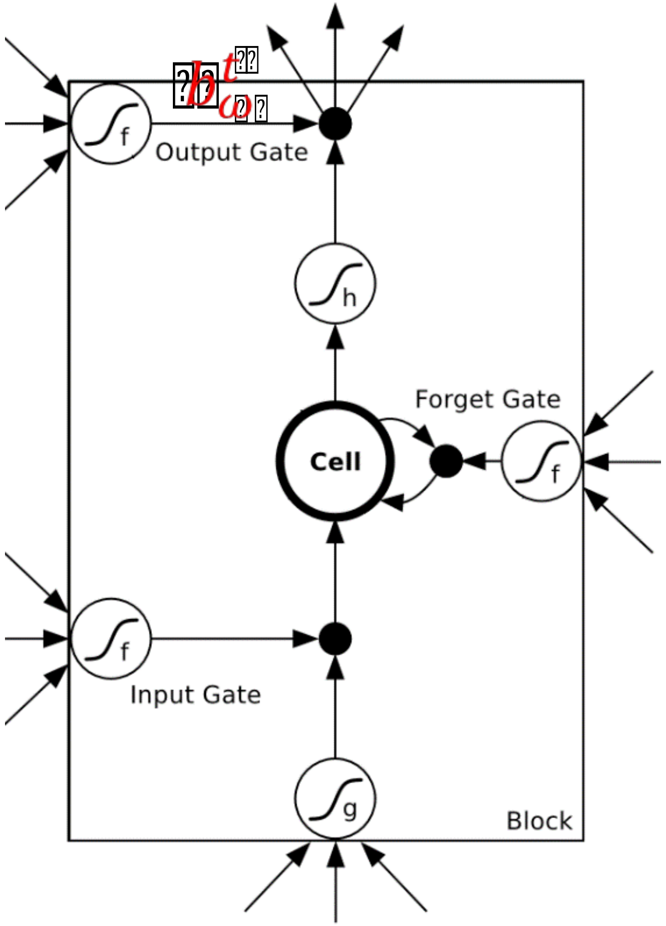
- Output gate ω
  - $a_{\omega}^{t} = \sum_{i=1}^{I} w_{i\omega} x_{i}^{t} + \sum_{h=1}^{H} w_{h\omega} b_{h}^{t-1}$
  - $b_{\omega}^{t} = f(a_{\omega}^{t})$
- Cell output b_c
  - $b_{c}^{t} = b_{\omega}^{t} \cdot h(s_{c}^{t})$
- Output gate
- Cell output



LSTM – forward pass

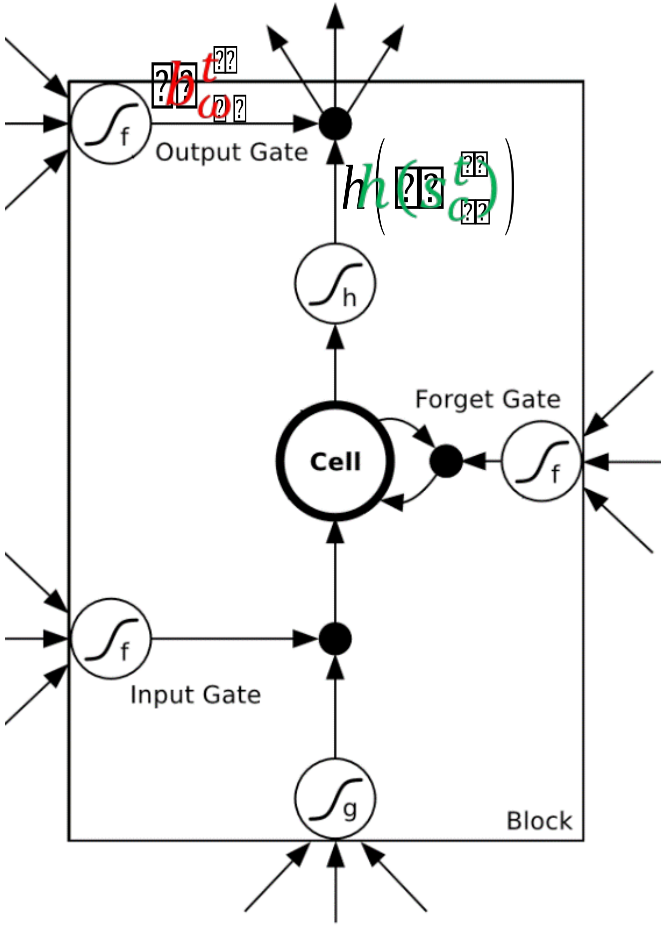
- Output gate ω
  - $a^t_\omega = \sum_{i=1}^{I} w_{i\omega} x_i^t + \sum_{h=1}^{H} w_{h\omega} b_h^{t-1}$
  - $b^t_\omega = f(a^t_\omega)$
- Cell output b_c
  - $b_c^t = b_\omega^t \cdot h(s_c^t)$

- Output gate
- Cell output



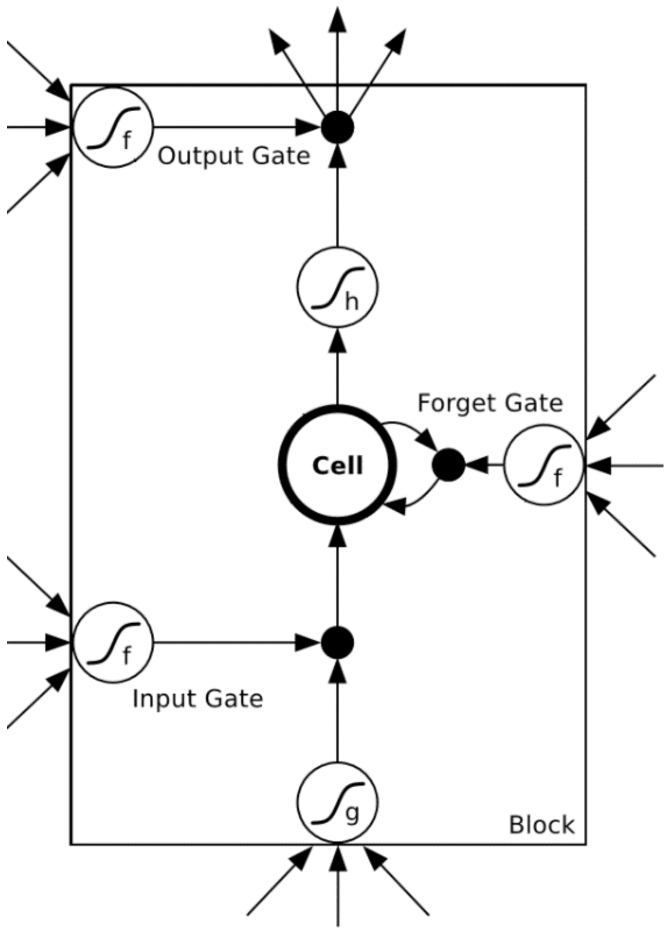
LSTM – forward pass

- Output gate ω
- Output gate
  - $a_\omega^t = \sum_{i=1}^{I} w_{i\omega} x_i^t + \sum_{h=1}^{H} w_{h\omega} b_h^{t-1}$
  - $b_\omega^t = f(a_\omega^t)$
- Cell output b_c
- Cell output
  - $b_c^t = b_\omega^t \cdot h(s_c^t)$



LSTM – architecture

- RNN where neuron units replaced with memory blocks
- Blocks consist of memory cells
- Gates
    - Read, write, reset signals

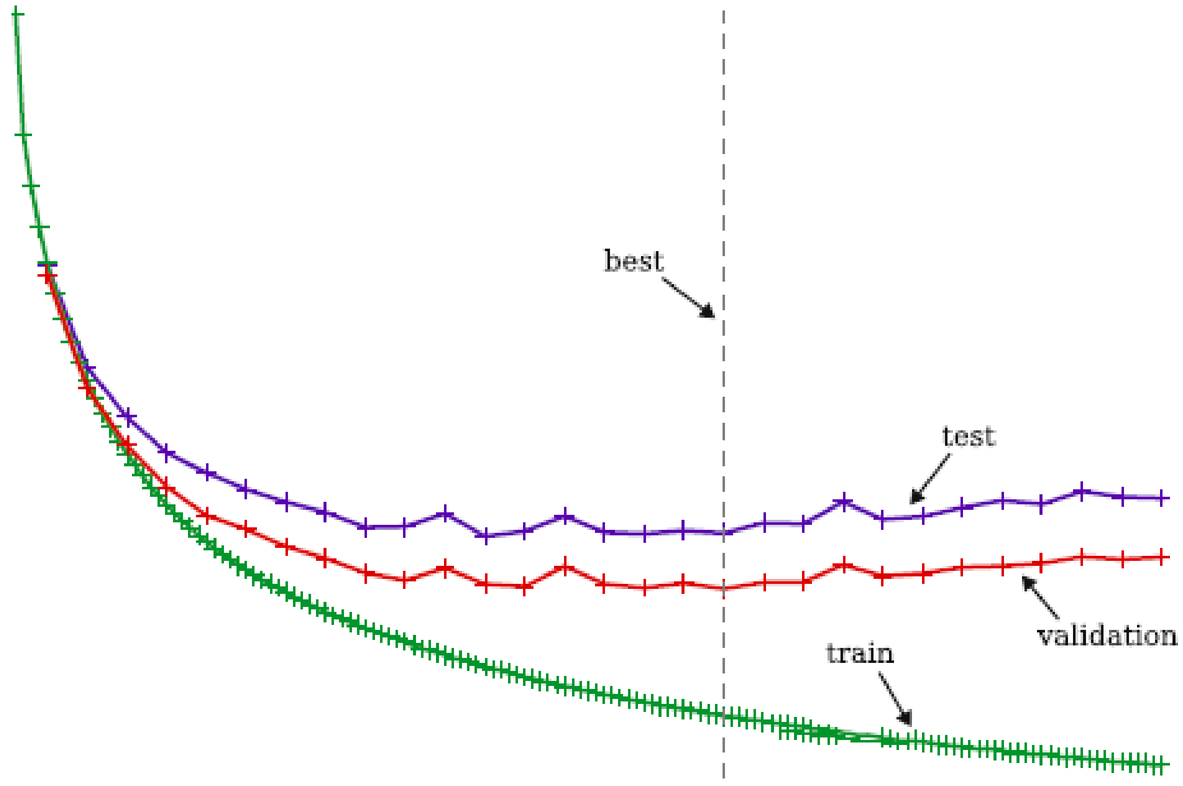


LSTM – training

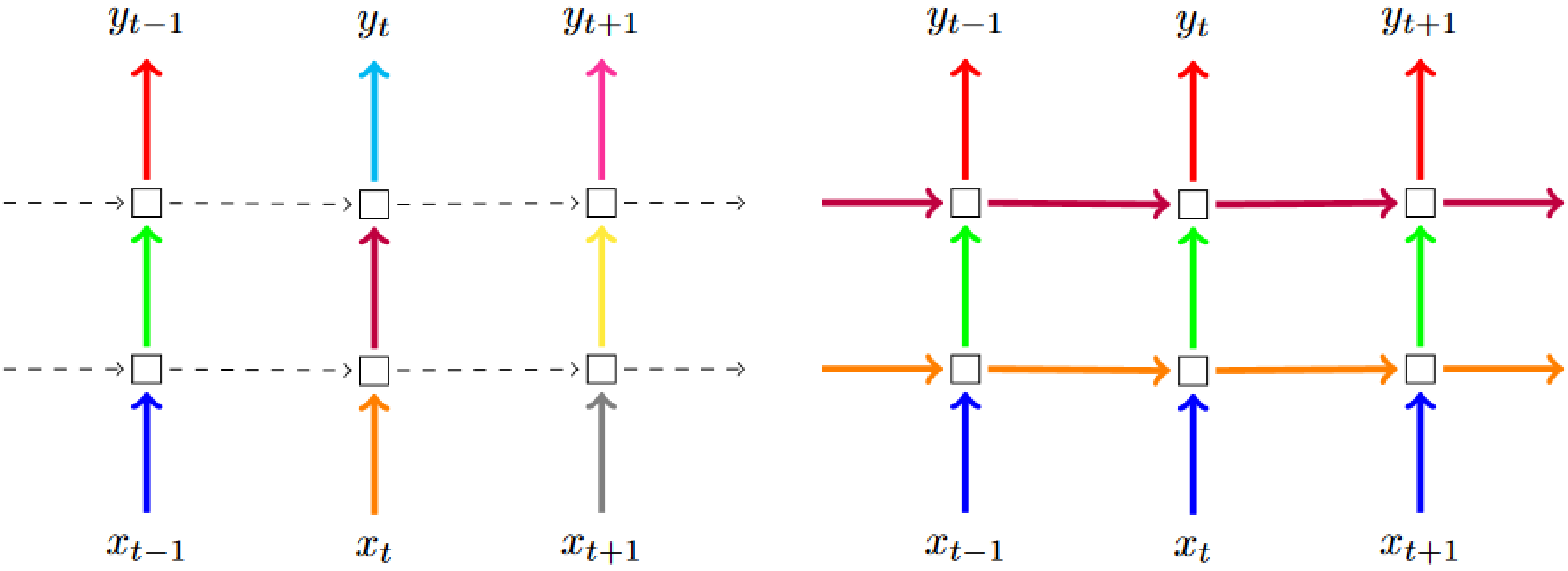
- No vanishing gradient problem
- Exploding gradient is addressed by **clipping gradient**
- Gradient
  - $\Delta\omega^n = -\alpha \frac{\partial \mathrm{L}}{\partial \omega^n}$
- Gradient with **momentum** m helps escaping local minima
  - $\Delta\omega^n = -\alpha \frac{\partial \mathrm{L}}{\partial \omega^n} + m\,\Delta\omega^{n-1}$

Early stopping

- Prevent overfitting training data
- Stopping after error fails to decrease for certain number of epochs

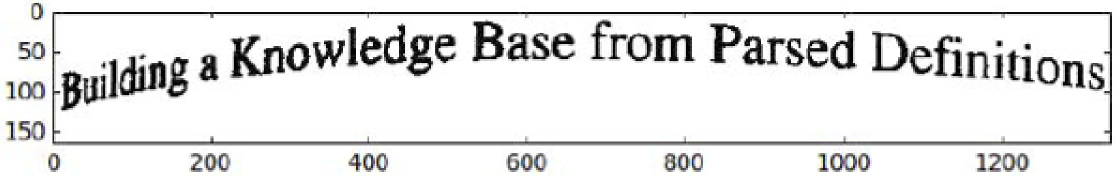


Regularization

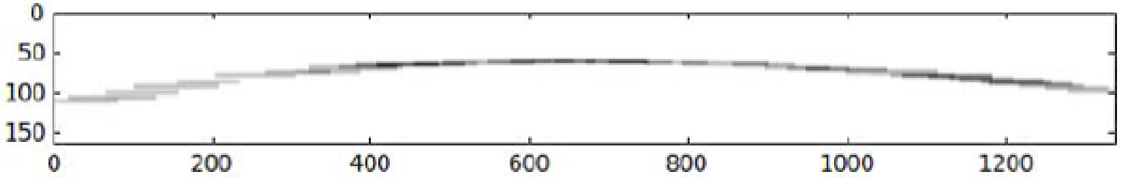


OCR

- Segmented approach
  - Extracting character candidates
  - Individual character classification
  - Search through list of guesses
  - Tesseract
- Unsegmented approach
  - Text line normalization
  - No language model
  - OCRopus



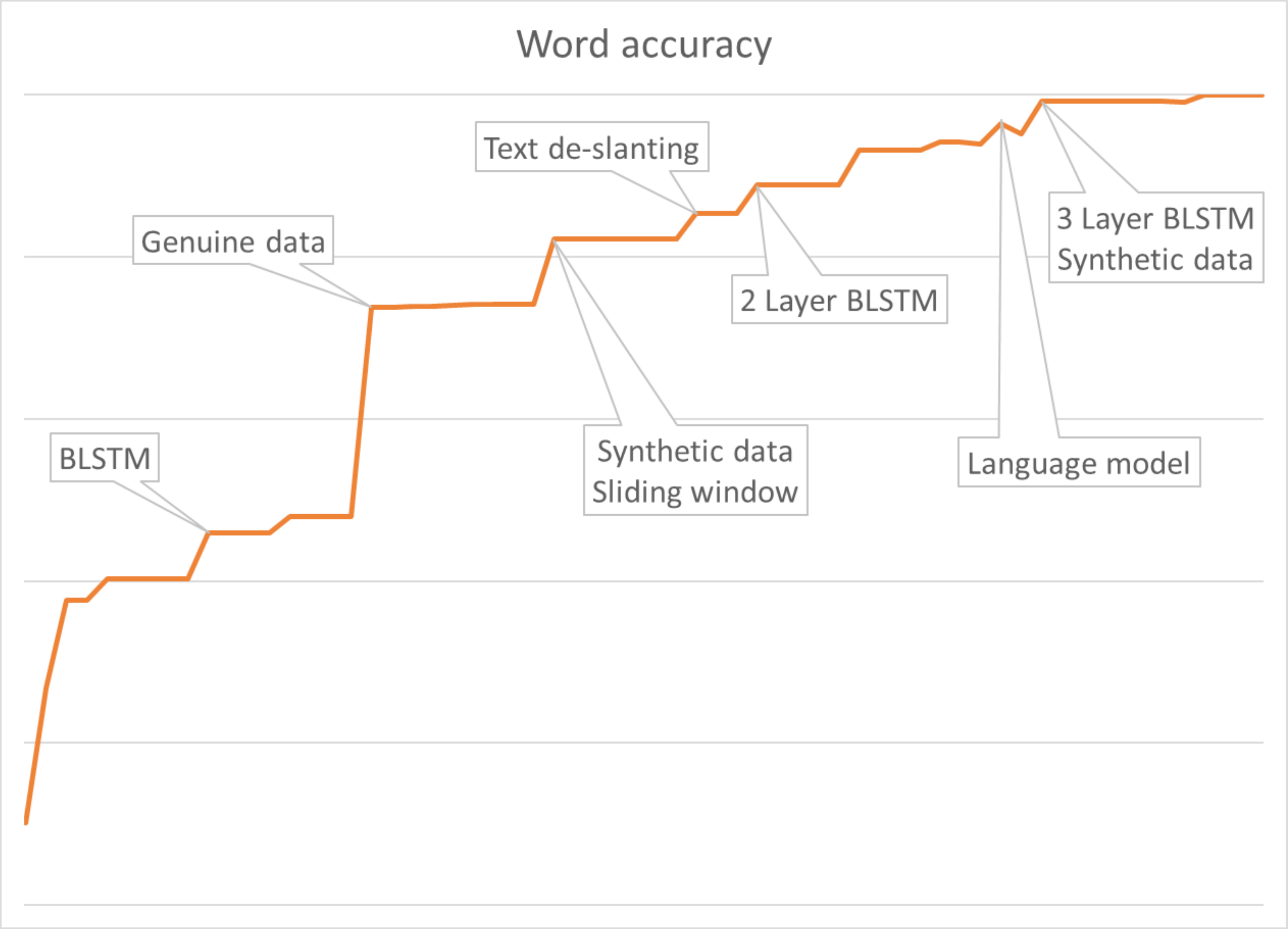
original text-line image



baseline map of the text-line

OCR - training

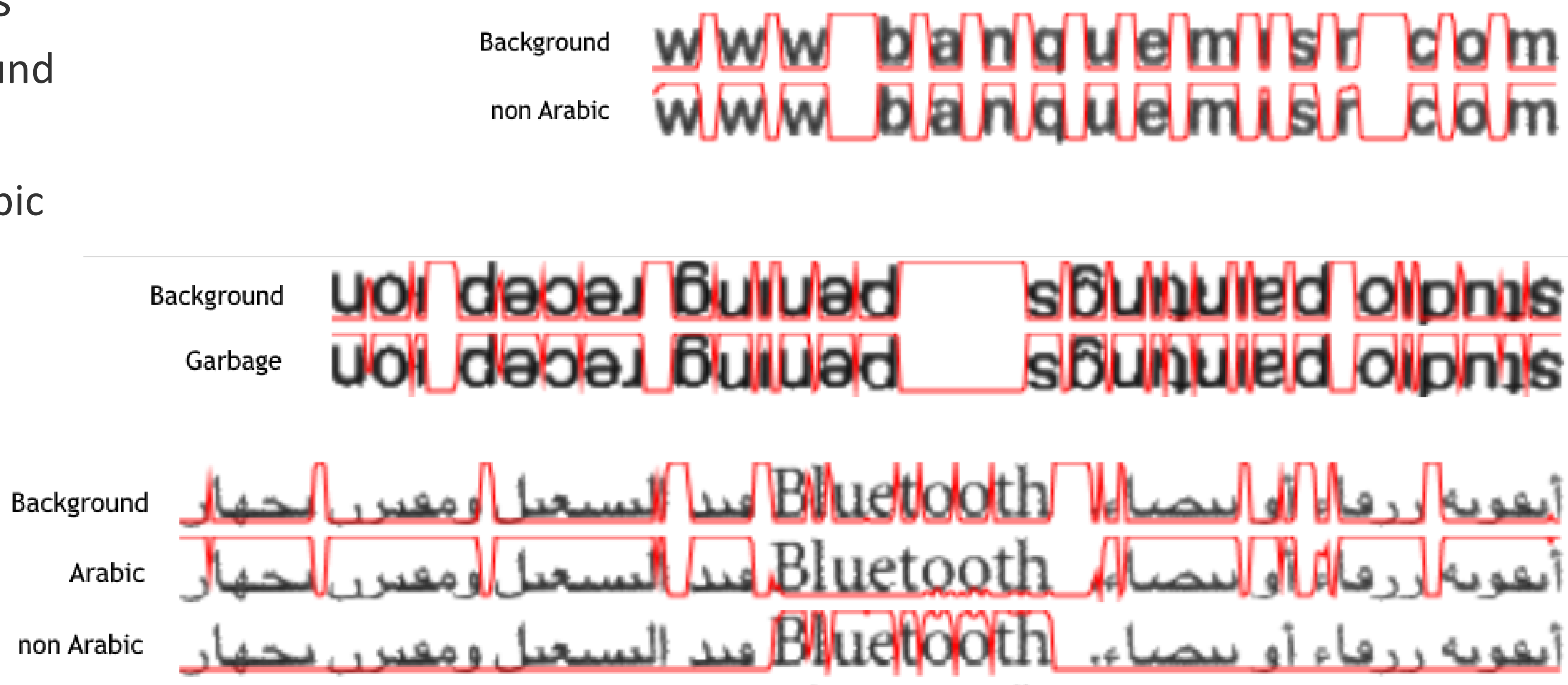
Arabic OCR



Word accuracy
Text de-slanting
Genuine data
2 Layer BLSTM
3 Layer BLSTM
Synthetic data
BLSTM
Synthetic data
Sliding window
Language model

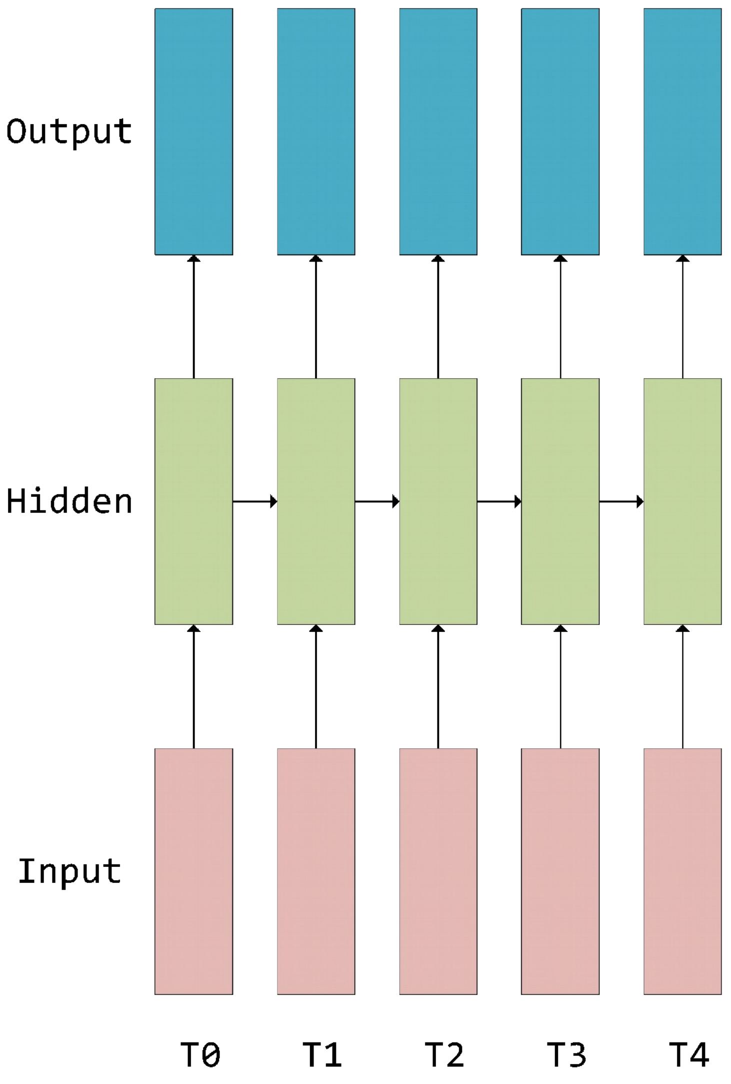
Arabic script detection

- Target signals
  - Background
  - Arabic
  - Non-Arabic
  - Garbage



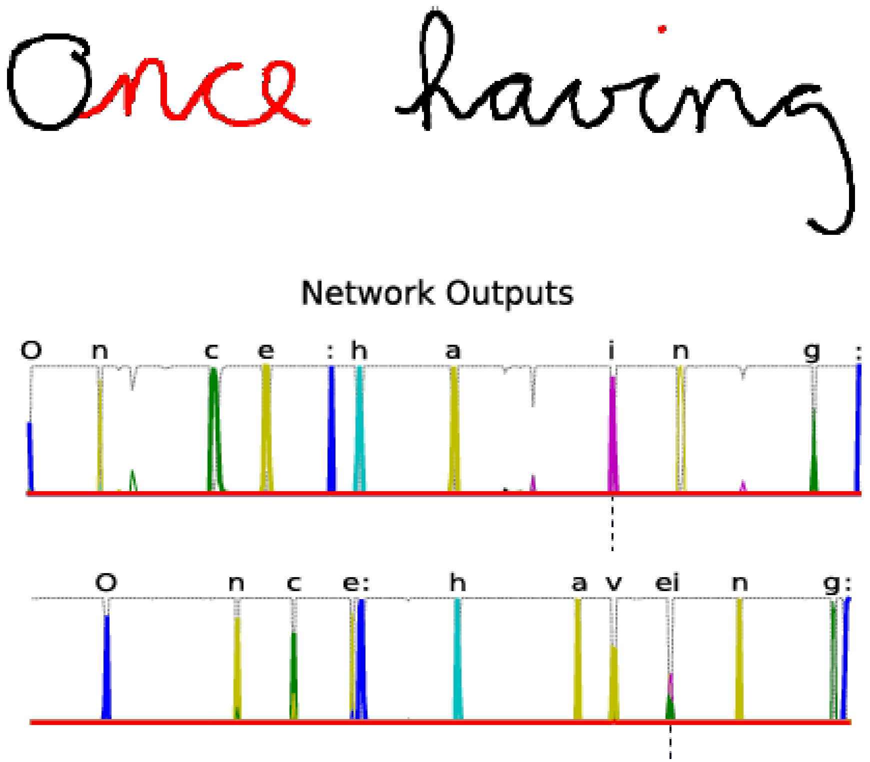
Arabic script detection

- Target signals
  - Background
  - Arabic
  - Non-Arabic
  - Garbage
- Decode output signal



Online handwriting recognition

- Data consists of stroke intervals (periods when the pen was pressed against the board) and sequences of x, y coordinates and time
- Raw features: *[x, y, t]*
- Preprocessed features
  - Reducing variance (slant, skew, character width)
  - Online features – position, speed, curvature
  - Offline features – sliding window



System	Input	LM	WER
HMM	preprocessed	✓	35.5%
CTC	raw	✗	$30.1 \pm 0.5\%$
CTC	preprocessed	✗	$26.0 \pm 0.3\%$
CTC	raw	✓	$22.8 \pm 0.2\%$
CTC	preprocessed	✓	$20.4 \pm 0.3\%$

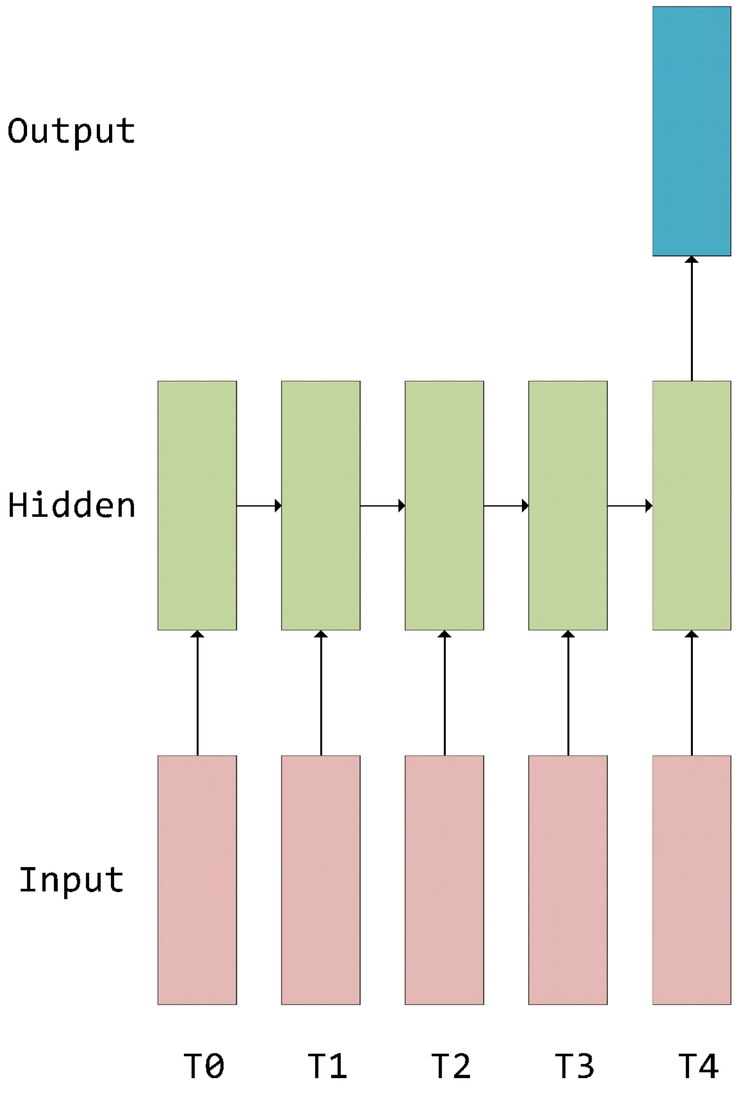
Language modeling

- Character level language model

- Shakespeare

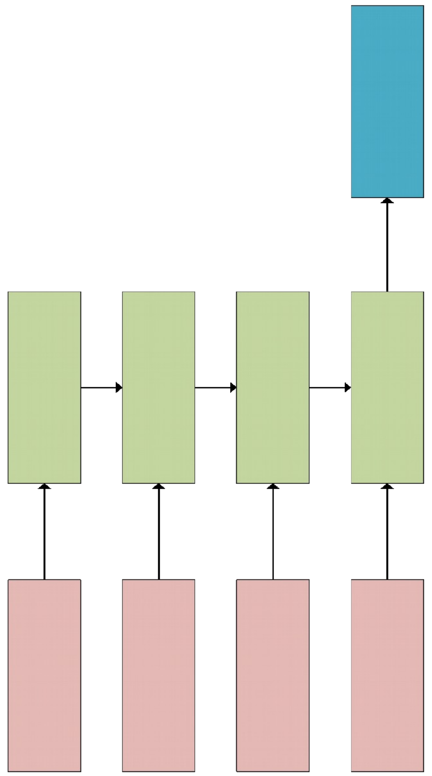
  PANDARUS:
  Alas, I think he shall be come approached and the day
  When little srain would be attain'd into being never fed,
  And who is but a chain and subjects of his death,
  I should not sleep.

  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..



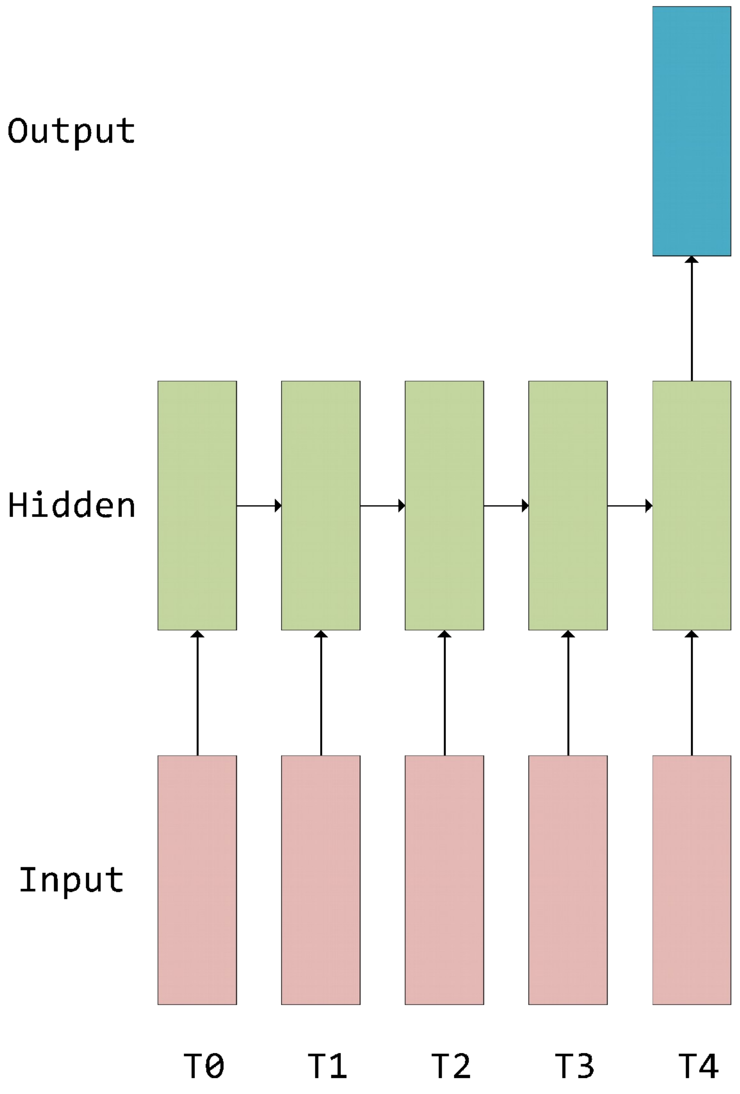
Machine translation

- [Translate English to French]
- Encoder decoder architecture
- Data set
  - 12M sentences
  - 348M French words
  - 304M English words
- Training took 10 days on 8 GPUs

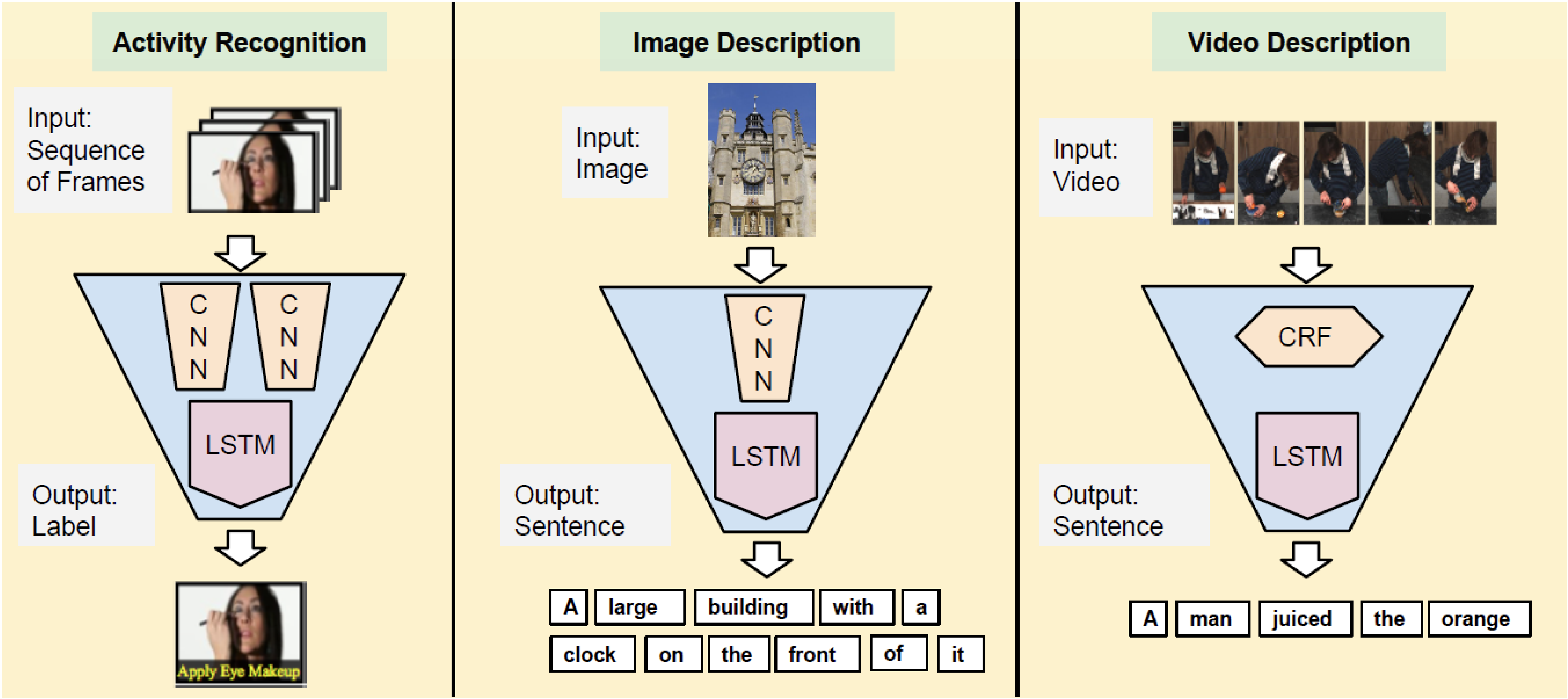


General purpose encoder

- [Learning general purpose distributed sentence representations via large scale multi-task learning]
- Multitask learning for sentence representations
- Encoder is bidirectional GRU
- Encoder is shared
- Each task has it’s own decoder/classifier
- Transfer learning



Visual Recognition and Description



Literature and useful links

- Supervised Sequence Labeling with Recurrent Neural Networks, Alex Graves
- Neural networks for Machine Learning, Geoffrey Hinton, www.coursera.org
- A Theoretically Grounded Application of Dropout in Recurrent Neural Networks
- High-Performance OCR for Printed English and Fraktur using LSTM Networks
- LSTM tutorial
- OCRopus line recognizer
- Deep Speech: Scaling up end-to-end speech recognition
- Unconstrained handwriting recognition using recurrent neural networks
- Best Practices for Convolutional Neural Networks Applied to Visual Document Analysi
s
- Visual recognition and description

Q/A

Thanks!