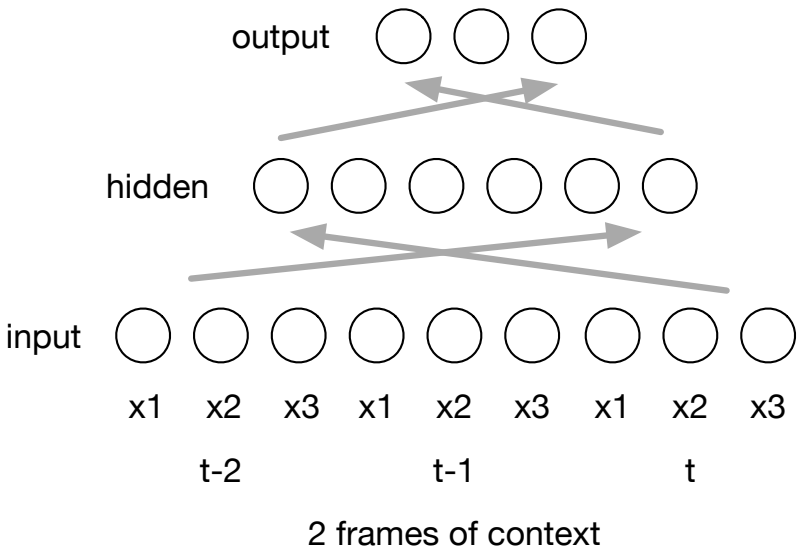
Recurrent Network Acoustic Models

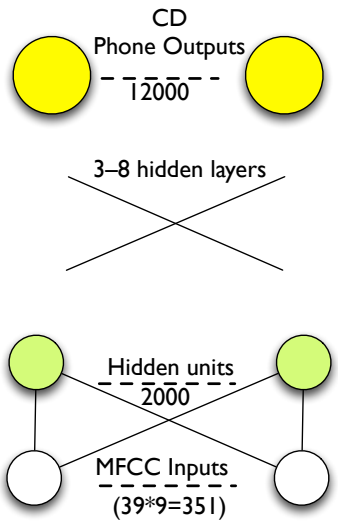
Steve Renals

Automatic Speech Recognition – ASR Lecture 13
29 February 2016

Sequential Data

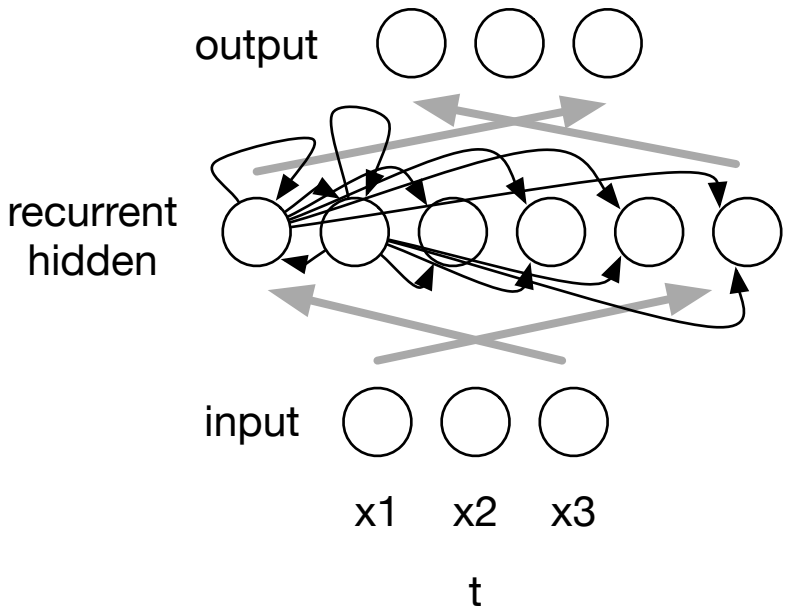


- Modelling sequential data with time dependences between feature vectors
- Can model fixed context with a feed-forward network with previous time input vectors added to the network input (in signal processing this is called FIR – **finite** input response)



DNN for acoustic modelling

Sequential Data



Don't use an input context window – context is learned by the recurrent hidden (state) units

- Modelling sequential data with time dependences between feature vectors
- Can model fixed context with a feed-forward network with previous time input vectors added to the network input (in signal processing this is called FIR – **finite** impulse response)
- Model sequential inputs using *recurrent* connections to learn a *time-dependent state* (in signal processing this is called IIR – **infinite** impulse response)

Recurrent networks

Can think of recurrent networks in terms of the dynamics of the recurrent hidden state

- Settle to a fixed point – stable representation for a sequence (e.g. machine translation)
- Regular oscillation ("limit cycle") – learn some kind of repetition
- Chaotic dynamics (non-repetitive) – theoretically interesting ("computation at the edge of chaos")

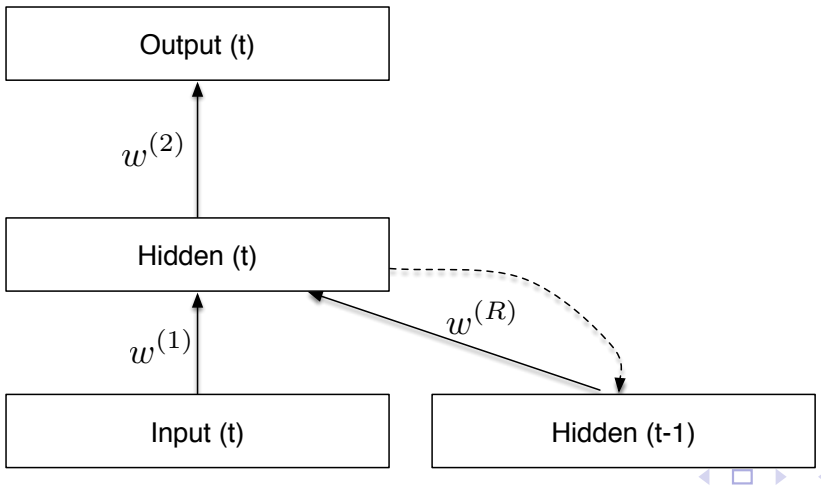
Useful behaviours of recurrent networks:

- Recurrent state as memory – remember things for (potentially) an infinite time
- Recurrent state as information compression – compress a sequence into a state representation

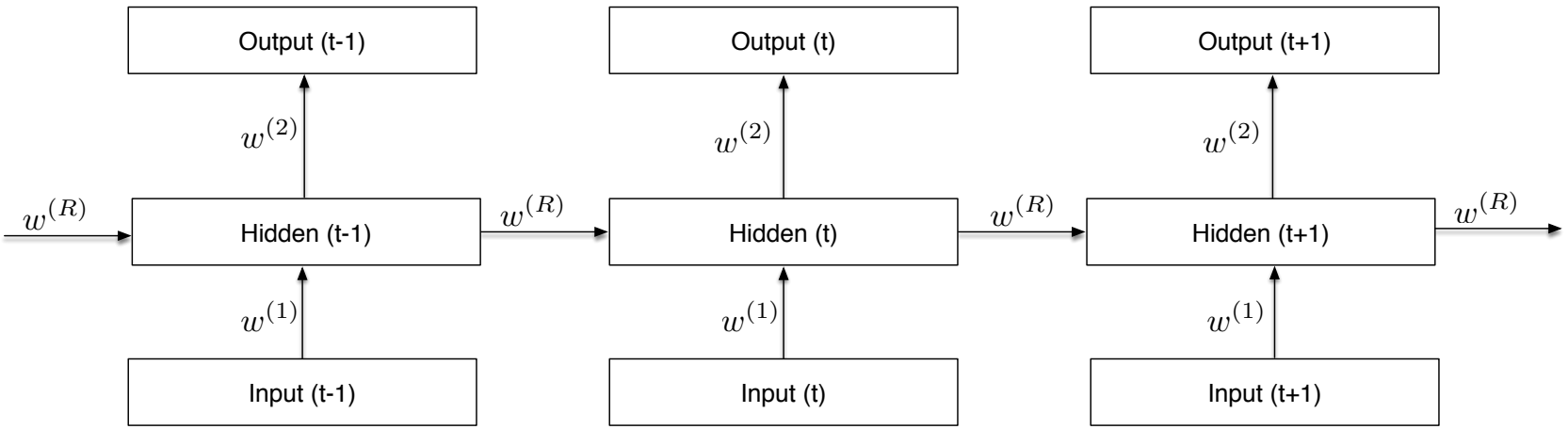
Simplest recurrent network

$$y_k(t) = \text{softmax}\left(\sum_{r=0}^{H} w_{kr}^{(2)} h_r(t)\right)$$

$$h_j(t) = \text{sigmoid}\left(\sum_{s=0}^{d} w_{js}^{(1)} x_s(t) + \underbrace{\sum_{r=0}^{H} w_{jr}^{(R)} h_r(t-1)}_{\text{Recurrent part}}\right)$$



Recurrent network unfolded in time



- An RNN for a sequence of T inputs can be viewed as a deep T-layer network with shared weights
- We can train an RNN by doing backprop through this unfolded network, making sure we share the weights
- Weight sharing
  - if two weights are constrained to be equal ($w_1 = w_2$) then they will stay equal if the weight changes are equal ($\partial E/\partial w_1 = \partial E/\partial w_2$)
  - achieve this by updating with ($\partial E/\partial w_1 + \partial E/\partial w_2$) (cf Conv Nets)

Back-propagation through time (BPTT)

- We can train a network by unfolding and *back-propagating through time*, summing the derivatives for each weight as we go through the sequence
- More efficiently, run as a recurrent network
  - cache the unit outputs at each timestep
  - cache the output errors at each timestep
  - then backprop from the final timestep to zero, computing the derivatives at each step
  - compute the weight updates by summing the derivatives across time
- Expensive – backprop for a 1,000 item sequence equivalent to a 1,000-layer feed-forward network
- Truncated BPTT – backprop through just a few time steps (e.g. 20)

Vanishing and exploding gradients

- BPTT involves taking the product of many gradients (as in a very deep network) – this can lead to vanishing (component gradients less than 1) or exploding (greater than 1) gradients
- This can prevent effective training
- Modified optimisation algorithms
  - RMSProp (normalise the gradient for each weight by average of it magnitude, learning rate for each weight)
  - Hessian-free – an approximation to second-order approaches which use curvature information
- Modified hidden unit transfer functions
  - Long short term memory (LSTM)
    - Linear self-recurrence for each hidden unit (long-term memory)
    - Gates - dynamic weights which are a function of the inputs
  - ReLUs

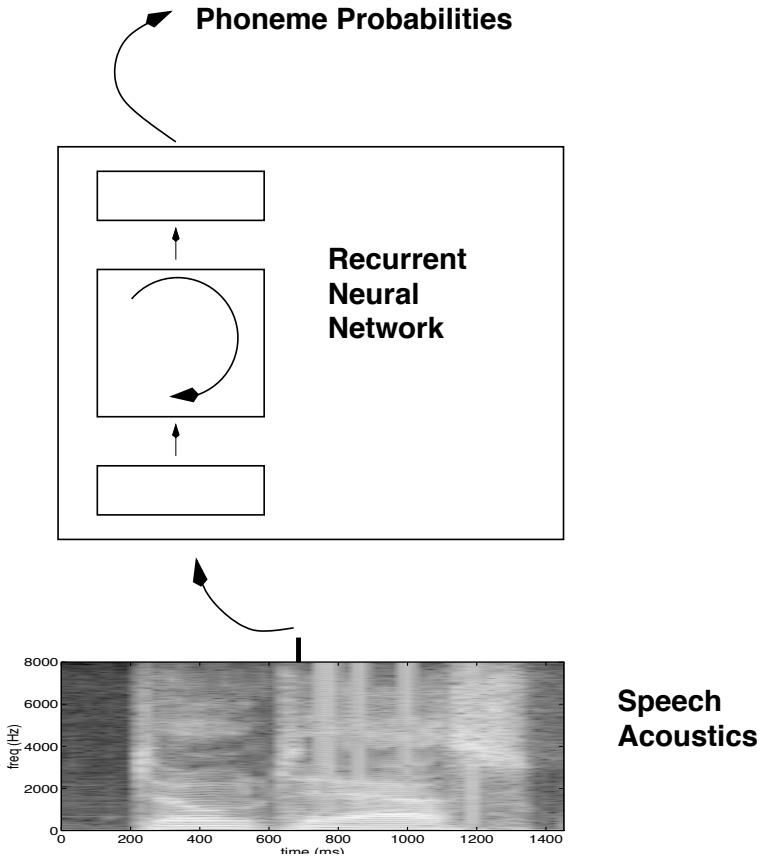
Recurrent networks in speech recognition

- 1990s – Hybrid RNN/HMM speech recognition (Robinson et al)
- 2009 onwards – RNN language models (Mikolov – new state of the art; next week's lecture)
- 2013 onwards – RNN/LSTM models at state of the art for acoustic modelling (Graves, Sak et al)
- 2015 onwards – RNN sequence modelling also replaces the HMM – "HMM-free" ASR (next week also)

Recurrent networks in speech recognition

- **1990s – Hybrid RNN/HMM speech recognition (Robinson et al)**
- 2009 onwards – RNN language models (Mikolov – new state of the art; next week's lecture)
- **2013 onwards – RNN/LSTM models at state of the art for acoustic modelling (Graves, Sak et al)**
- 2015 onwards – RNN sequence modelling also replaces the HMM – "HMM-free" ASR (next week also)

Speech recognition with recurrent networks (1990s)

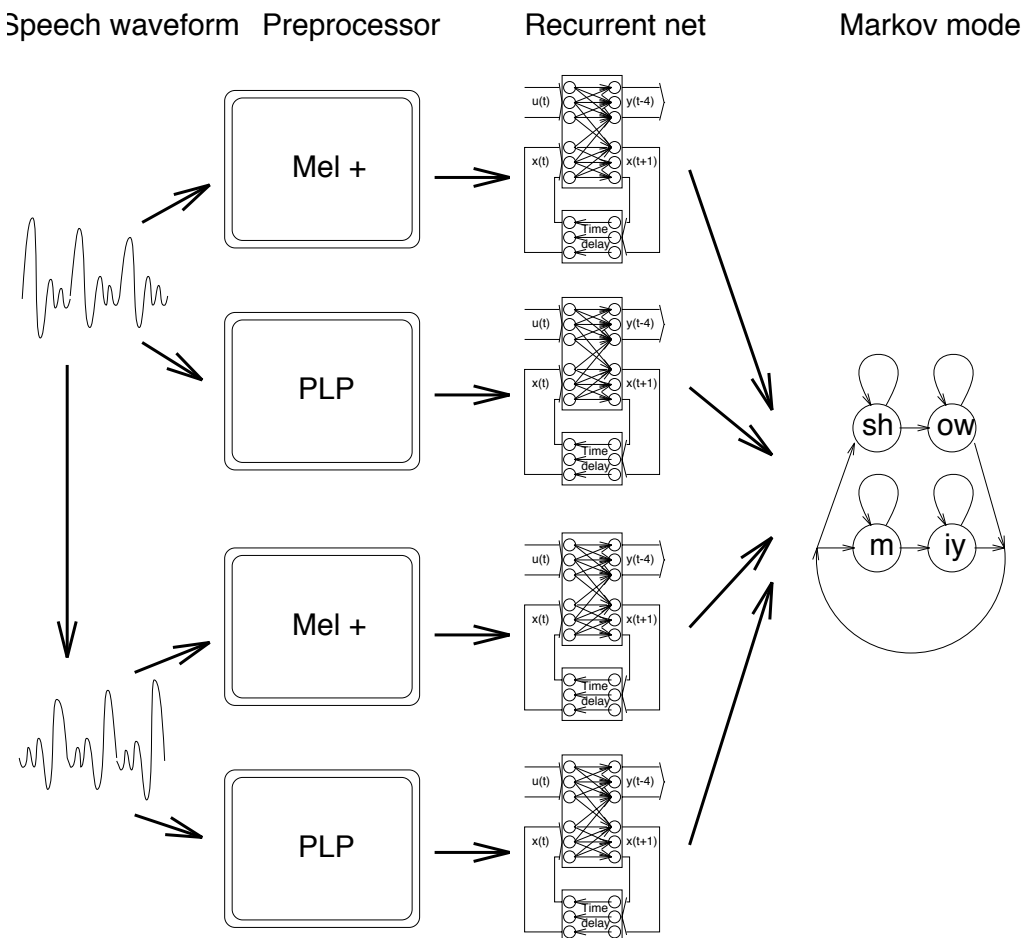


Robinson et al (1996)

1990s RNN

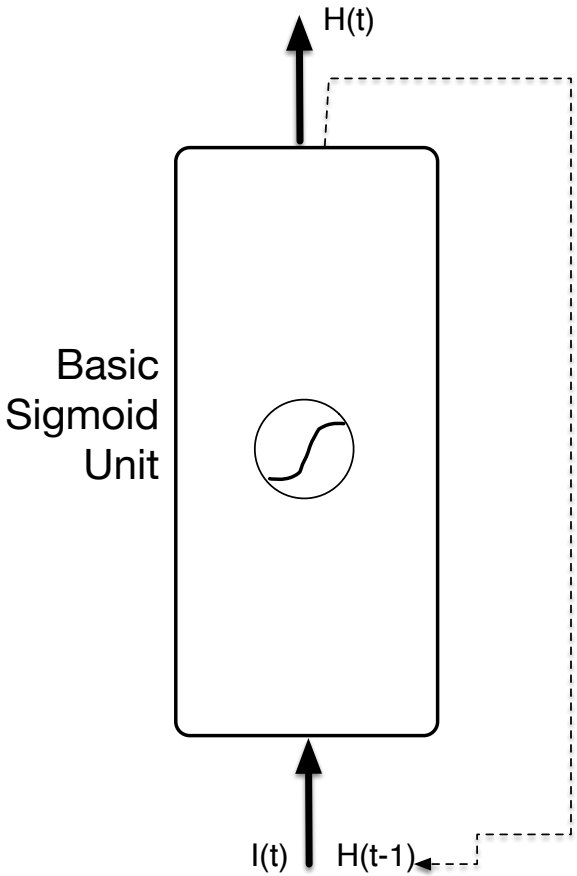
- Features
  - MEL+: Filter-bank outputs + voicing parameters (23 features/frame)
  - PLP cepstral coefficients (13 features)
  - Coefficients normalized to zero mean and unit variance
- 256 hidden (state) units
- 79 context-independent phone classes (outputs)
- Output training target delayed by 5 frames
- Use RNN scaled likelihoods in hybrid RNN/HMM
- About 100k trainable parameters
- Trained using stochastic BPTT (using method similar to Rprop/RMSprop) on WSJ0 (3M training examples)
- In 1994 training took five days on a specially designed parallel computer (the RAP)

Combined RNN system

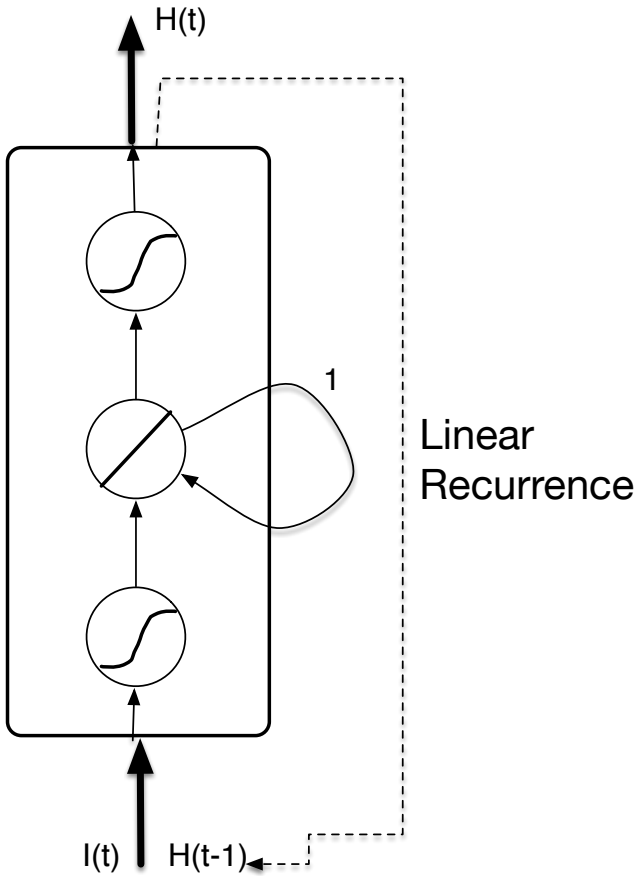


Individual systems: WER of 14–15% on WSJ "spoke 6 data"
Interpolated in log domain: WER=11%
(Best context dependent GMM/HMM system in 1995: WER=8%)

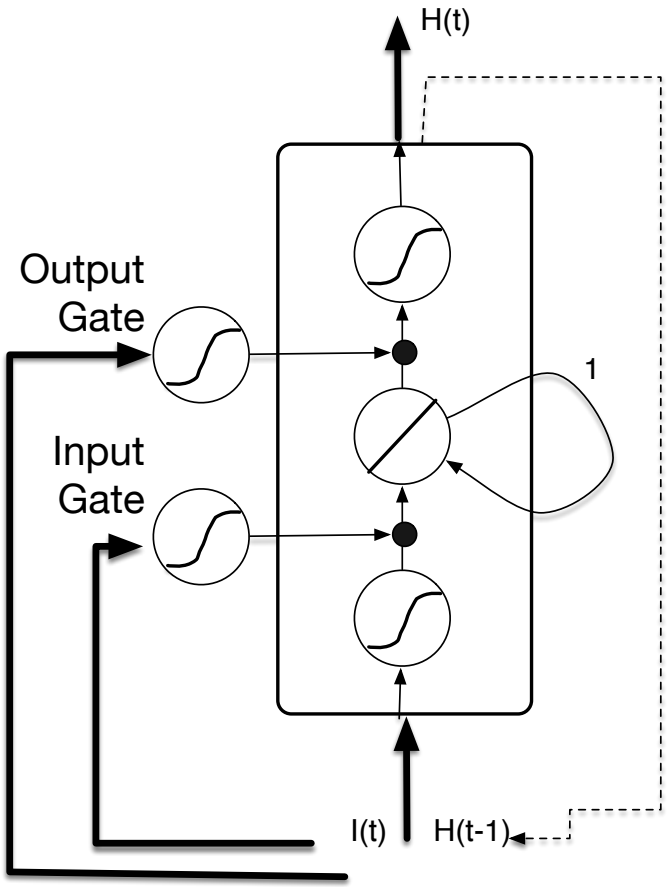
LSTM v1



LSTM v1

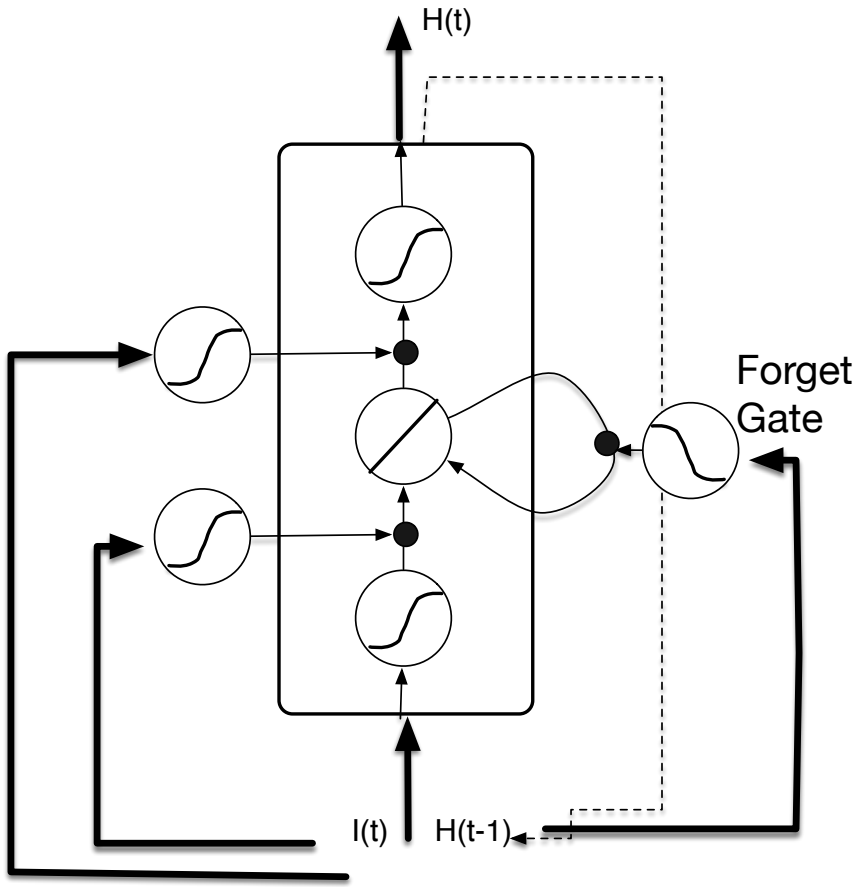


LSTM v1



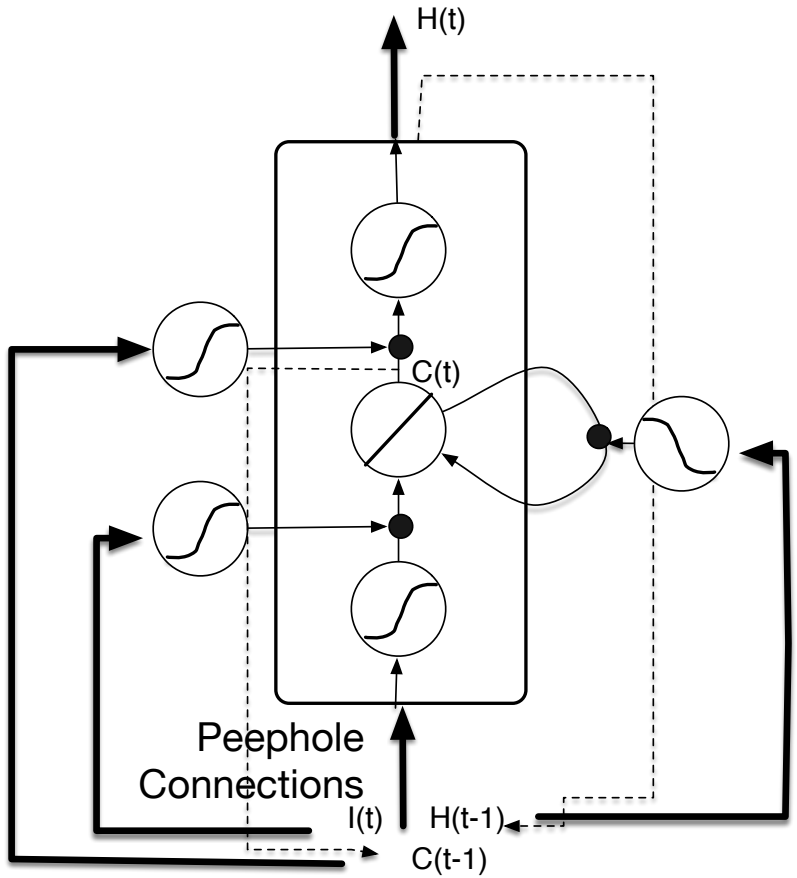
S Hochreiter and J Schmidhuber (1997). "Long Short-Term Memory", *Neural Computation*, 9:1735–1780.

LSTM v2



FA Gers et al (2000). "Learning to Forget: Continual Prediction with LSTM", *Neural Computation*, 12:2451–2471.

LSTM v3



LSTM equations

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{fx}x_t + W_{mf}m_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) \quad (4)$$

$$m_t = o_t \odot h(c_t) \quad (5)$$

$$y_t = W_{ym}m_t + b_y \quad (6)$$

Google LSTM experiments (2014–date)

- Google voice search data – 1900h training set
- Input features: 40-d log mel filterbank energies.
- No input context, single frame 40-d frame presented each timestep
- LSTM networks
  - 400–6,000 LSTM cells per layer
  - 1–7 layers
  - 13M–37M trainable parameters
  - 14,727 context-dependent states
- Delay output targets by 5 frames
- Use RNN to estimated scaled likelihoods
- Train using BPTT

LSTM acoustic modelling architectures

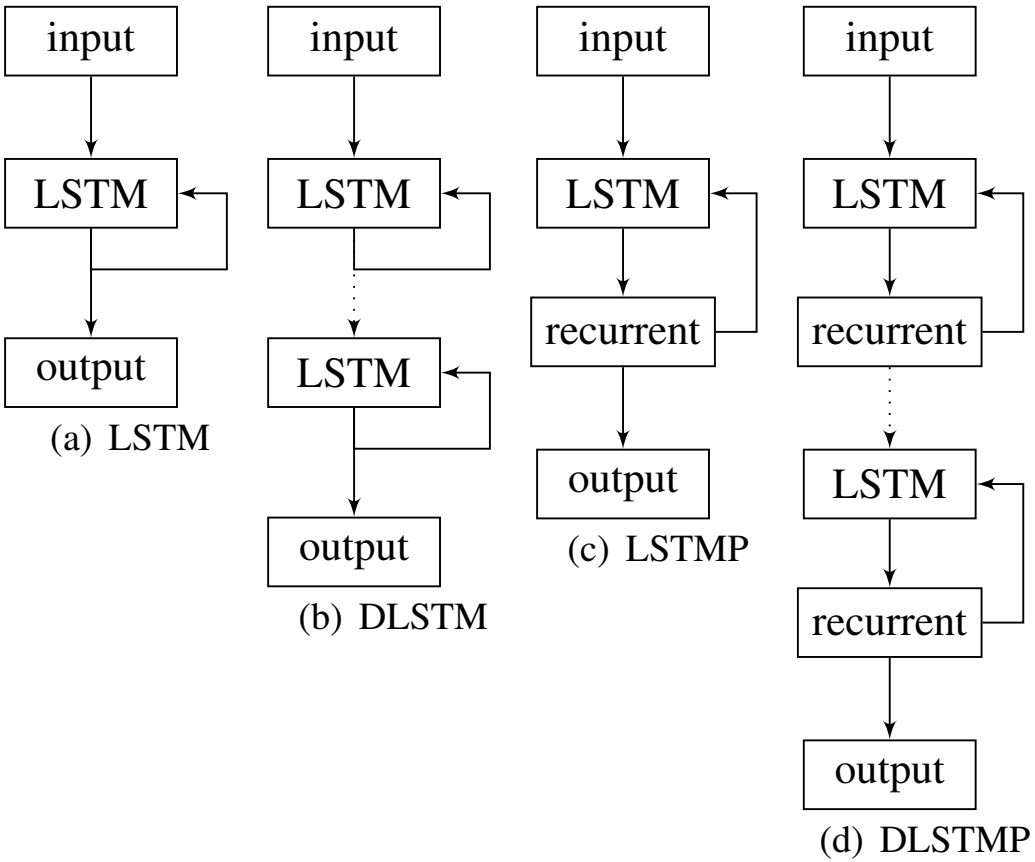


Figure 2: LSTM RNN architectures.

(Sak, 2014)

LSTM with projection layer (LSTMP)

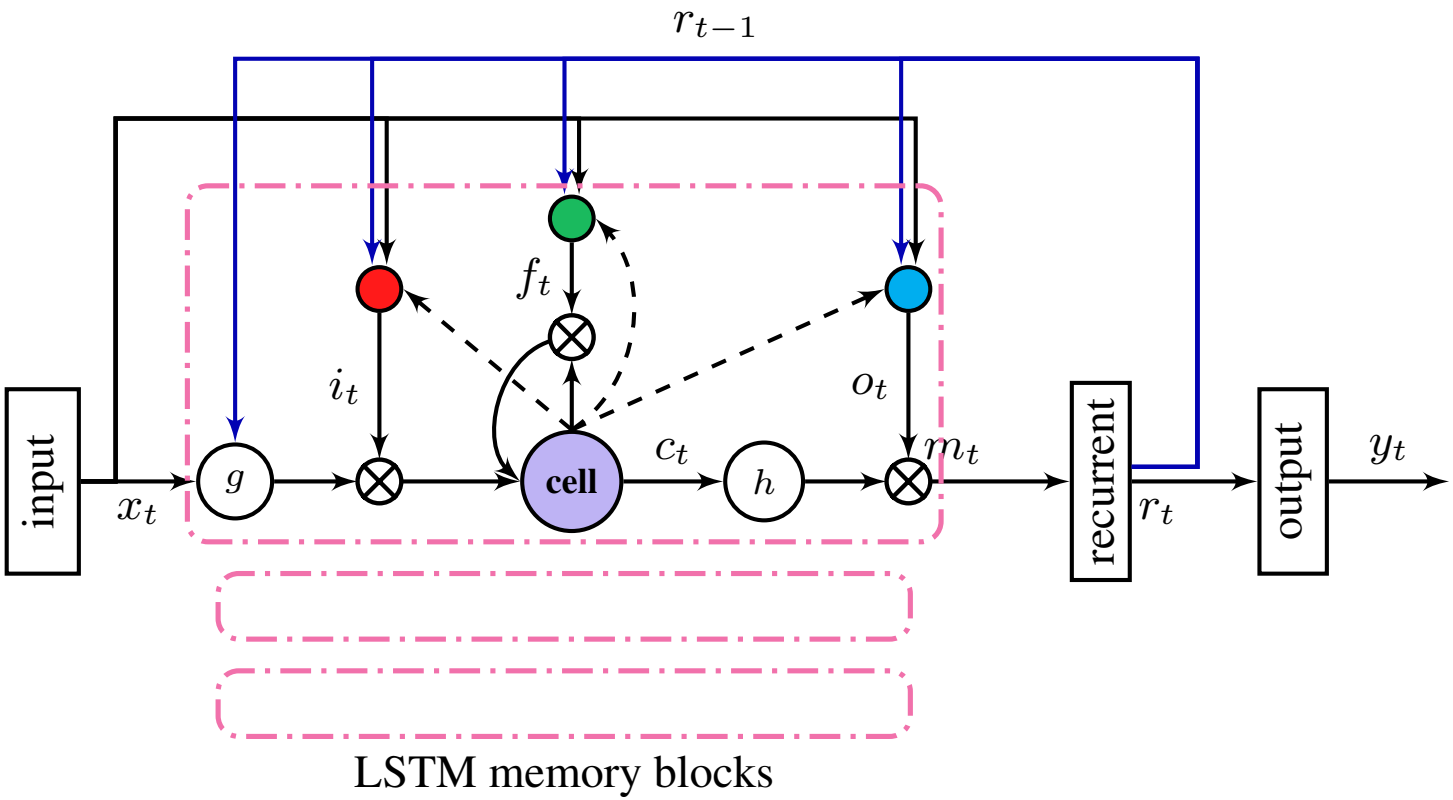


Figure 1: LSTMP RNN architecture. A single memory block is shown for clarity.

(Sak, 2014)

LSTM ASR Results (Google Voice Search task)

Table 1: *Experiments with LSTM and LSTMP RNN architectures showing test set WERs and frame accuracies on development and training sets. L indicates the number of layers, for shallow (1L) and deep (2,4,5,7L) networks. C indicates the number of memory cells, P the number of recurrent projection units, and N the total number of parameters.*

C	P	Depth	N	Dev (%)	Train (%)	WER (%)
840	-	5L	37M	67.7	70.7	10.9
440	-	5L	13M	67.6	70.1	10.8
600	-	2L	13M	66.4	68.5	11.3
385	-	7L	13M	66.2	68.5	11.2
750	-	1L	13M	63.3	65.5	12.4
6000	800	1L	36M	67.3	74.9	11.8
2048	512	2L	22M	68.8	72.0	10.8
1024	512	3L	20M	69.3	72.5	10.7
1024	512	2L	15M	69.0	74.0	10.7
800	512	2L	13M	69.0	72.7	10.7
2048	512	1L	13M	67.3	71.8	11.3

(Sak, 2014)

Reading

S Hochreiter and J Schmidhuber (1997). "Long Short-Term Memory", *Neural Computation*, 9:1735–1780.
http://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735#.VtN93Mehb5k

FA Gers et al (2000). "Learning to Forget: Continual Prediction with LSTM", *Neural Computation*, 12:2451–2471.
http://www.mitpressjournals.org/doi/abs/10.1162/089976600300015015#.VtN9ncehb5k

T Robinson et al (1996). "The use of recurrent networks in continuous speech recognition", in *Automatic Speech and Speaker Recognition Advanced Topics*, Lee et al (eds), Kluwer, 233–258.
http://www.cstr.ed.ac.uk/downloads/publications/1996/rnn4csr96.pdf

H Sak et al (2014), "LSTM recurrent neural network architectures for large scale acoustic modeling", Interspeech-2014.
http://research.google.com/pubs/archive/43905.pdf