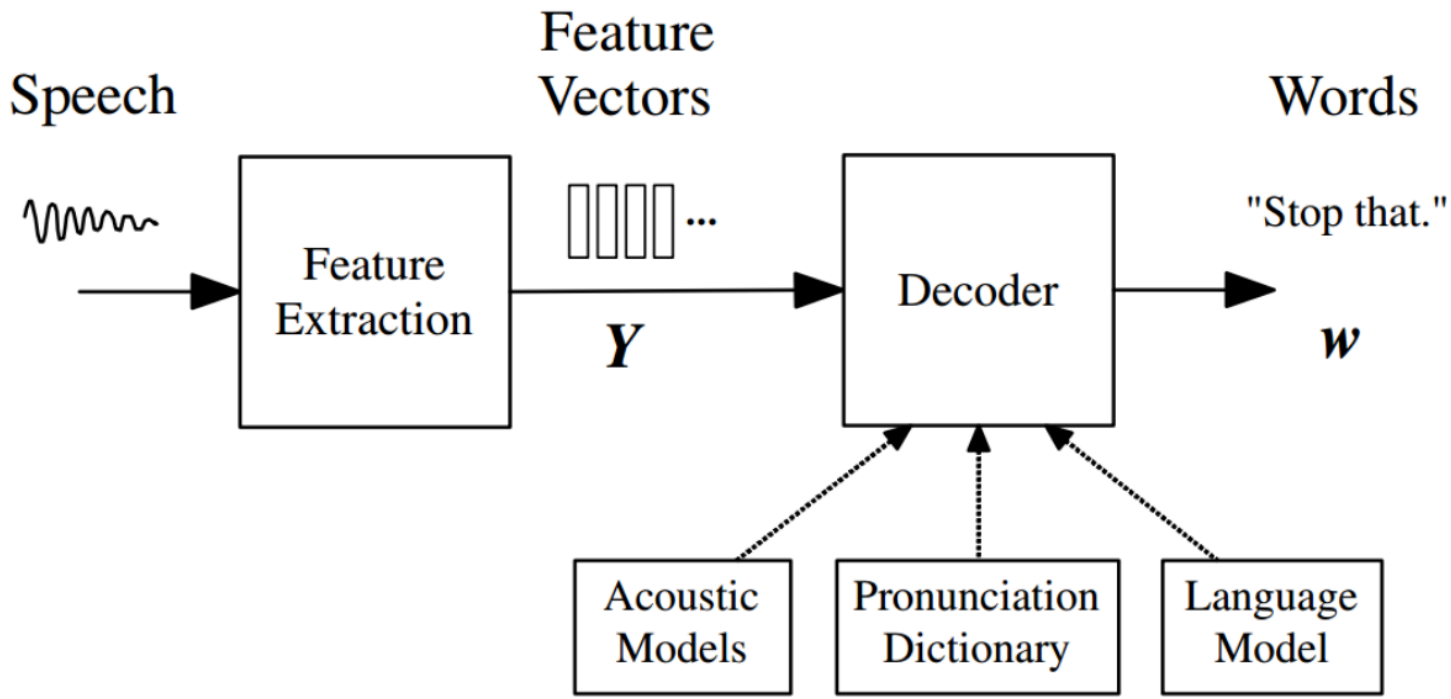
Speaker adaptation of deep acoustic model

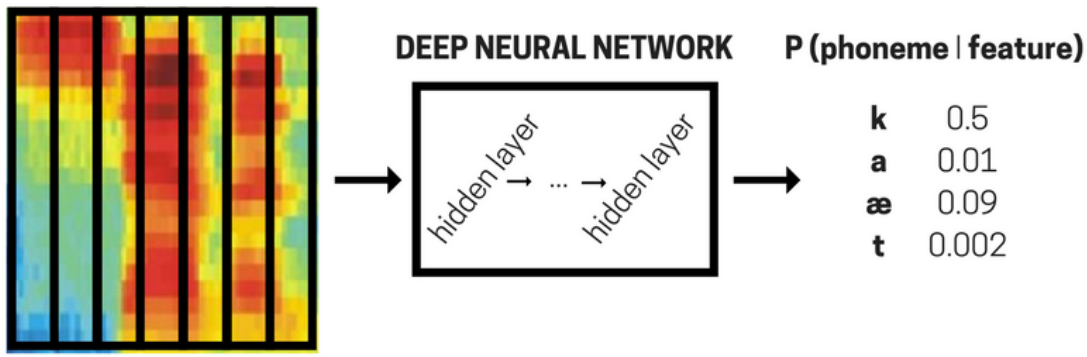
Marek Grochowski

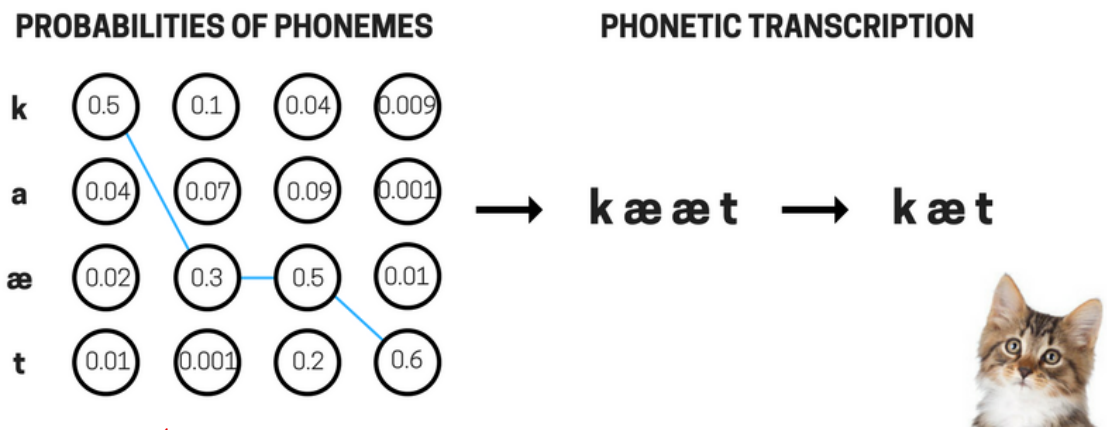
KIS 11.03.2019

Automatic Speech Recognition



Deep acoustic model (AM)





src: https://medium.com/coinmonks/have-you-ever-wondered-how-amazon-echo-siri-or-google-home-work-539abed4092f

Speaker adaptation problem

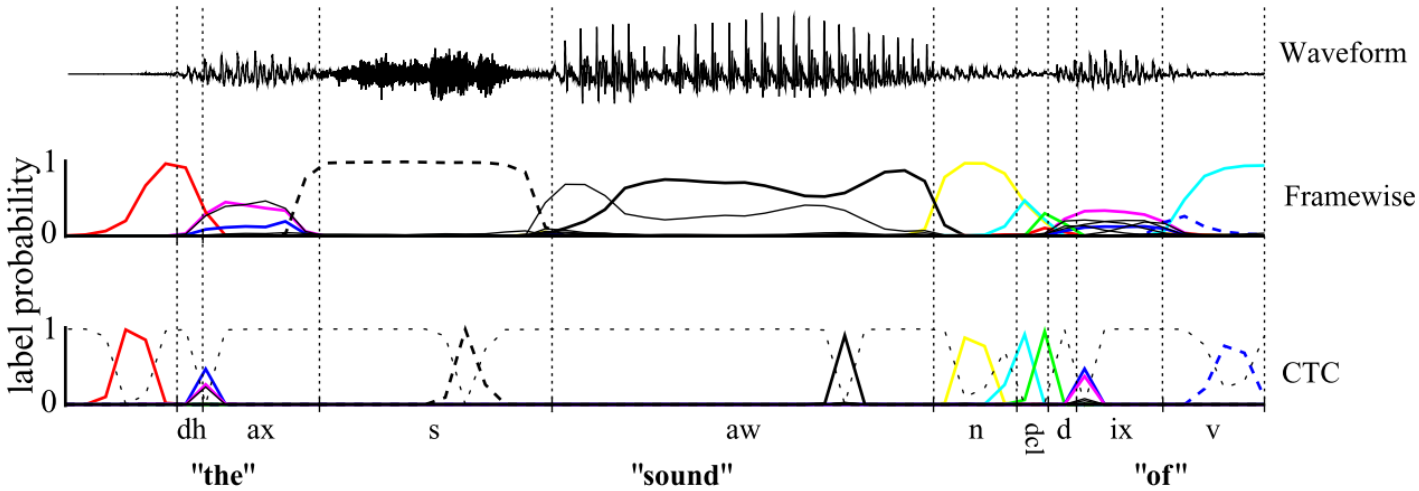
- fine-tune the AM to better match speaker-specific pronunciation
- speaking styles, accents, speech production anatomy
- speech defects: **rhotacism**, lisping, stuttering, ...
- non-speaker variation: channel conditions (telephone, reverberations, etc.)
- tiny data: one or only few sentences available

Base AM

- training data: 679K utterances from various sources: 500h voicelab recordings, telephone calls, ninateka, sejm, ...
- trained with CTC loss for 25 epochs + noise + reverberation
- 91 inputs: 13 MFCC features + 3 context size
- 42 outputs
  (phonemes + special labels (e.g. `spn`, `nsn`) + blank)
- 6 layers with 512 LSTM nodes
- Adam optimisation, dropout ($p = 0.8$), gradient normalisation, lr decay (0.9 per epoch), ...

Data	PER [%]	WER [%]
932	10.6	12.4
932 ($bs = 0.1$)	11.0	10.3

CTC vs. CE training



- CE: frame-wise training $(\mathbf{x}_i, \mathbf{y}_i)$, phonemes are aligned to input frames
- CTC: net predicts the sequence of phonemes as spikes separated by 'blank' (additional label), no force alignment required, more costly because of searching and decoding process

Input data

'DonaldTusk.wav'	00:03:57.23
'HannaGronkiewiczWaltz.wav'	00:05:31.77
'JanMariaRokita.wav'	00:01:11.09
'KazimieraSzczuka.wav'	00:04:22.83
'MarcinKydrynski.wav'	00:01:45.95
'NinaTerentiew.wav'	00:01:19.92
'PascalBrodnicki.wav'	00:00:28.52

Rhotacizm vs. base AM errors

Subject	$\frac{N_{err('r')}}{N}$	$\frac{N_{err('r')}}{N_{'r'}}$		substitutions
932	0.8	2.6	22/859	<eps>:14 rz:1 j:1 l:1 n:1 p:1 t:1 spn:1 y:1
DT	2.8	7.5	4/53	ł:1 j:1 l:1 <eps>:1
HGW	**8.9**	**68.0**	70/103	<eps>:35 **l:14** t:5 o:3 e:2 dzi:1 j:1 a:1 rz:1 sp:1 p:1 d:1 w:1 m:1 ł:1 cz:1
JMR	9.6	64.7	22/34	<eps>:15 o:2 t:2 e:1 ł:1 d:1
KS	3.4	**62.3**	38/61	<eps>:28 h:2 w:2 j:2 t:1 sz:1 a:1 rz:1
MK	4.8	27.0	10/37	w:4 <eps>:4 j:1 l:1
NT	7.3	50.0	11/22	<eps>:4 ł:3 z:1 w:1 l:1 o:1

Input data

Data	# train	# dev
hgw	23	11
ks	20	11
rot	63	55

- 20-25s. sentence length (wav + transcriptions)
- input: 13 MFCC coefficients
- output: phoneme sentences $\rightarrow$ alignment
- rhotacism data:
  train = { HGW train, KS train, NT, JMR}
  dev = { HGW dev, KS dev, DT, MK}

Output layer training with CTC

- Fine-tune output layer of base AM
  all remaining network weights are frozen
- Stacked output layer (SL)
  add new output layer initialized with identity matrix
  previous output layer become last hidden layer with linear activation (logits)
  train weights in new output layer
  trainable parameters: $42^2 \approx 1.7K \quad \ll \quad 512 * 42 \approx 21.5K$

CTC training results

Model	WER (%)			
	hgw	ks	rot	932
base AM	54.0	69.2	43.1	12.4
CTC	51.0	63.0	41.9	11.8-11.9
CTC SL	51.0	64.0	43.1	11.3-12.1

Output scaling

- scale CE error back signal depending on class label p

$$J(\mathbf{x}) = -\sum_p r_p y_p \log f_p(\mathbf{x})$$

- disable `'blank'`

$$r_p = \begin{cases} 0 & \texttt{if} \quad p = \texttt{'blank'} \\ 1 & \texttt{otherwise} \end{cases}$$

- `only 'r'`

$$r_p = \begin{cases} 1 & \texttt{if} \quad p = \texttt{'r'} \\ 0 & \texttt{if} \quad p \neq \texttt{'r'} \end{cases}$$

- `strong 'r'`

$$r_p = \begin{cases} +1 & \texttt{if} \quad p = \texttt{'r'} \\ 0 & \texttt{if} \quad p = \texttt{'blank'} \\ 0.1 & \texttt{otherwise} \end{cases}$$

Results

Model	WER (%)			
	hgw	ks	rot	932
base AM	54.0	69.2	43.1	12.3
CE	**40.5**	**61.6**	**36.7**	**10.6-13.1**
CE SL	**40.0**	**55.9**	**36.5**	**12.4-15.2**
CE only 'r'	52.5	68.7	42.9	11.7-13.2
CE SL only 'r'	100.0	?	?	?
CE strong 'r'	**43.5**	**61.1**	**36.6**	**11.9-13.9**
CE SL strong 'r'	49.0	67.7	54.5	24.2-27.7
CTC	51.0	63.0	41.9	11.8-11.9
CTC SL	51.0	63.9	41.5	11.3-11.8

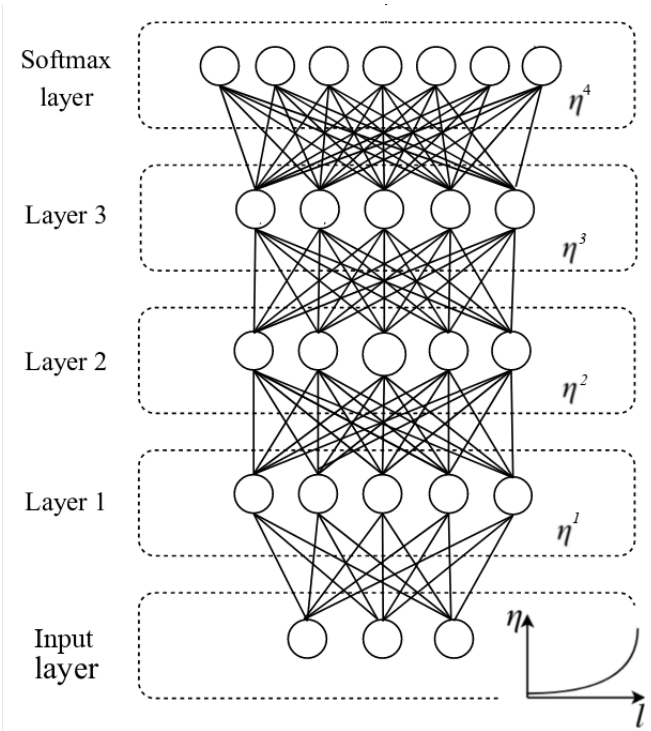
Layer-wise learning rate

- discriminative fine-tuning for LM (Howard, 2018)
- Instead of using the same learning rate for all layers of the model - tune each layer with different learning rates
- the SDG update for parameters Θ^k in k-th layer

$$\Theta^k = \Theta^k - \eta^k \cdot \nabla_{\Theta^k} J(\Theta)$$

- learning rate decreases with depth

$$\eta^k = \alpha \eta^{k+1}, \qquad \alpha \in [0, 1]$$



Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification

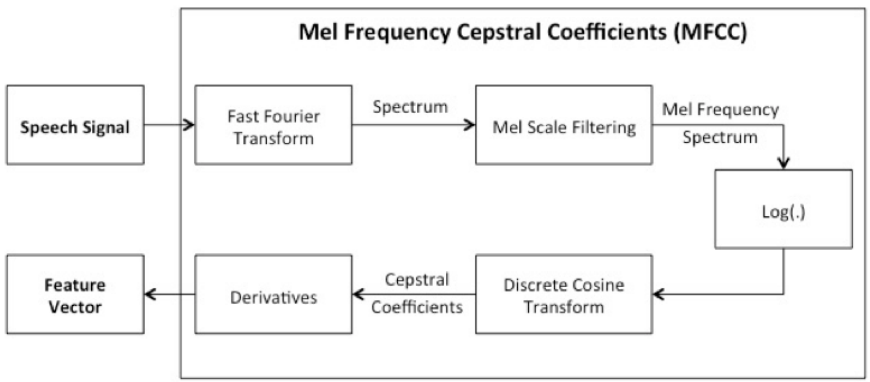
Results

- CTC LWLR best results with $\alpha = 0.6$
- all presented results obtained using improved LM (up to 10% better WER on base AM)

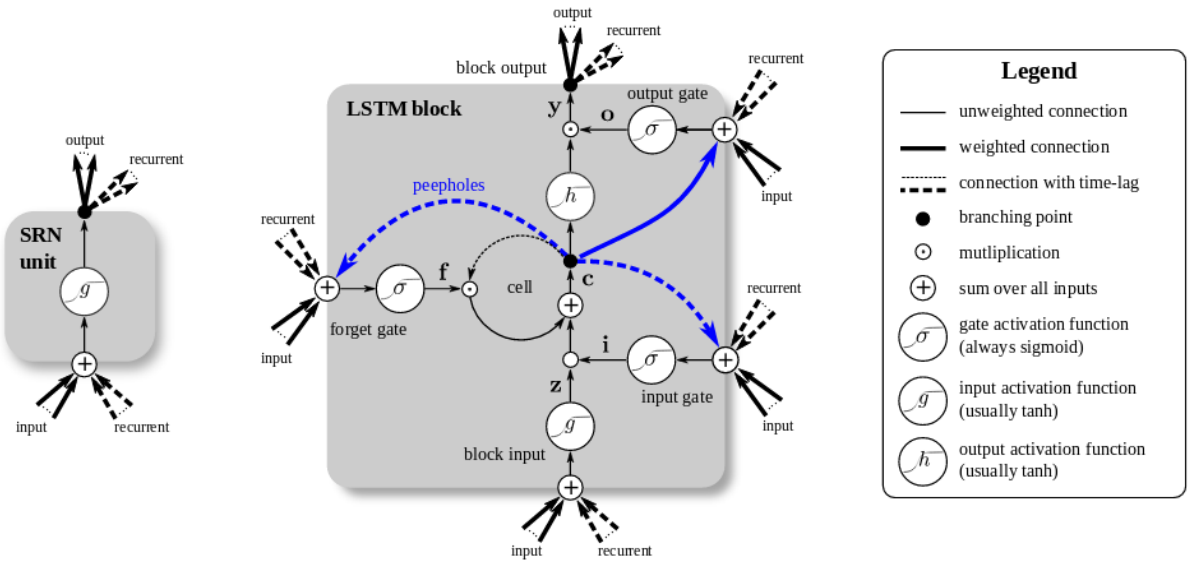
Model	WER (%)		
	hgw	ks	rot
base AM	46.0	63.5	34.7
CE output adj. best	36.5	52.6	33.7
CTC LWLR $\alpha = 0.6$	29.5	39.8	24.9

Mel-frequency cepstral coefficients (MFCC)

- Input: 16kHz sampled audio (wav)
- 25ms window (shift by 10 ms)
- Multiply by windowing function e.g. Hamming
- FFT
- Take log energy in each frequency bin
- Discrete cosine transform (DCT) $\rightarrow$ "cepstrum"
- Keep the first 13 coefficients of the cepstrum (input vector)



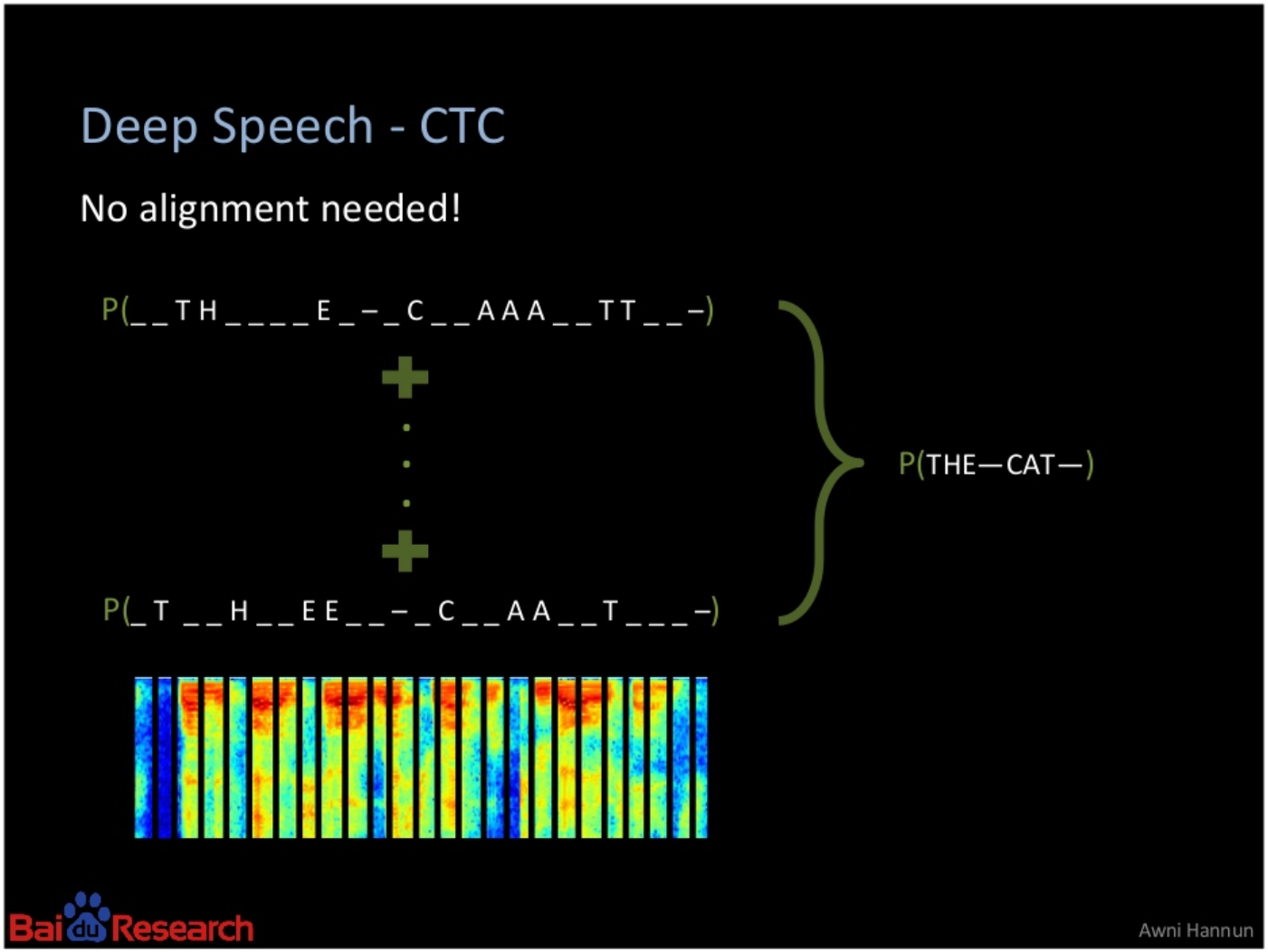
Long Short-Term Memory



$$\mathbf{z}^t = g(\mathbf{W}_z \mathbf{X}^t + \mathbf{R}_z \mathbf{y}^{t-1} + \mathbf{b}_z)$$
$$\mathbf{i}^t = \sigma(\mathbf{W}_i \mathbf{X}^t + \mathbf{R}_i \mathbf{y}^{t-1} + \mathbf{p}_i \odot \mathbf{c}^{t-1} + \mathbf{b}_i)$$
$$\mathbf{f}^t = \sigma(\mathbf{W}_f \mathbf{X}^t + \mathbf{R}_f \mathbf{y}^{t-1} + \mathbf{p}_f \odot \mathbf{c}^{t-1} + \mathbf{b}_f)$$
$$\mathbf{c}^t = \mathbf{i}^t \odot \mathbf{z}^t + \mathbf{f}^t \odot \mathbf{c}^{t-1}$$
$$\mathbf{o}^t = \sigma(\mathbf{W}_o \mathbf{X}^t + \mathbf{R}_o \mathbf{y}^{t-1} + \mathbf{p}_o \odot \mathbf{c}^t + \mathbf{b}_o)$$
$$\mathbf{y}^t = \mathbf{o}^t \odot h(\mathbf{c}^t)$$

Sepp Hochreiter, Jürgen Schmidhuber, Łong short-term memory", 1997

Connectionist Temporal Classification (CTC)



Connectionist Temporal Classification (CTC)

- loss function

$$L = -\sum \log p(\mathbf{y}|\mathbf{x})$$

- where

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{y}' \in \mathcal{M}^{-}1(\mathbf{y})} P(\mathbf{y}'|\mathbf{x})$$

where $\mathcal{M}$ map labels sequence of length T by removing all blanks and repetitions, e.g.
$\mathcal{M}(a - ab-) = \mathcal{M}(-aa - abb) = aab$

Graves, A., Fernández, S., Gomez, F., Schmidhuber, J., Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. 2006.