RNNs for Scene Text

Girish Varma
IIIT Hyderabad
http://bit.ly/2u2J1o0

The Scene Text Problem

shakeshack-----

30x100x3
tensor

(26+1)x15
tensor

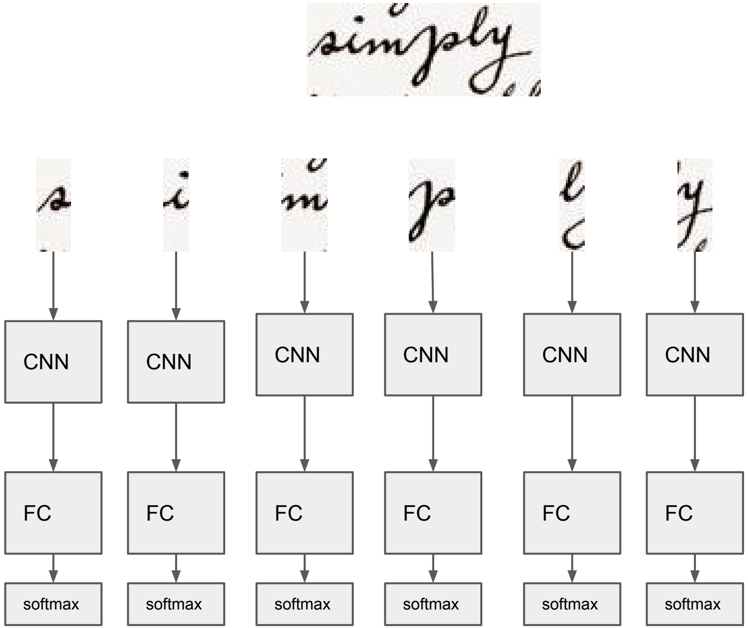
Outputs are supposed to be conditional probabilities : p_1, $p_{2/1}$, $p_{3/1,2}$, $p_{4/1,2,3}$, $p_{5/1,2,3,4}$, ….

Simple Solution

- Segment the characters using known algorithms.
- Use a CNN+FC network to classify each character.
- Train CNN+FC on char images.

The model predicting the i^{th} char, does not know the previously predicted chars or previously seen images.

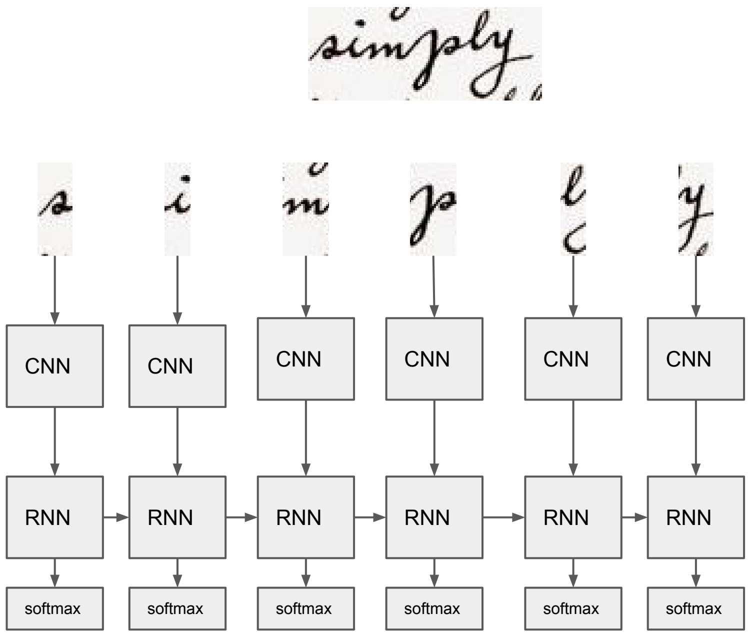
Use a CRF as a post processing step.



Simple RNN Solution : Learning CRF in the model

- Extract CNN features
- Pass CNN feature sequence through RNN
- Pass RNN output though softmax to get alphabet probabilities
- Loss function: $\sum_i \text{Error}(y_i, y_i^{\text{correct}})$

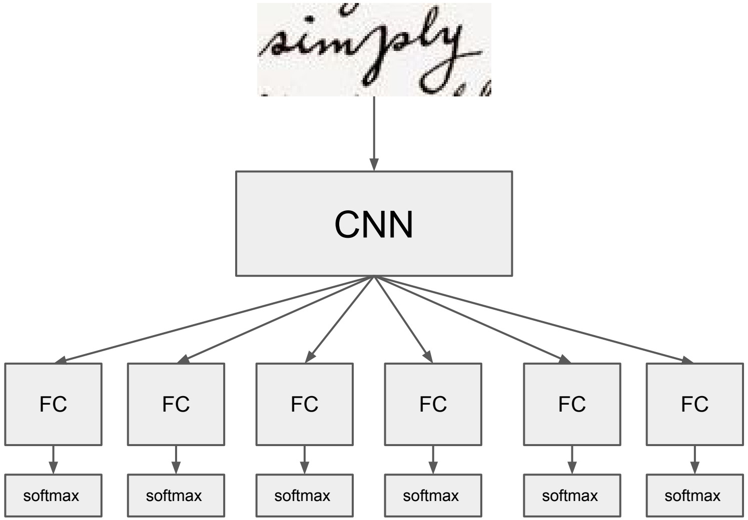
While predicting the i^{th} char, RNN has information about the previously seen char images.



Without Char Segmentation

- CNN for image feature extraction.
- Duplicate CNN features into a sequence of Maxlen.
- Use independent FC layers followed by softmax to get char probabilities.

If Maxlen is large, too many FC layers.

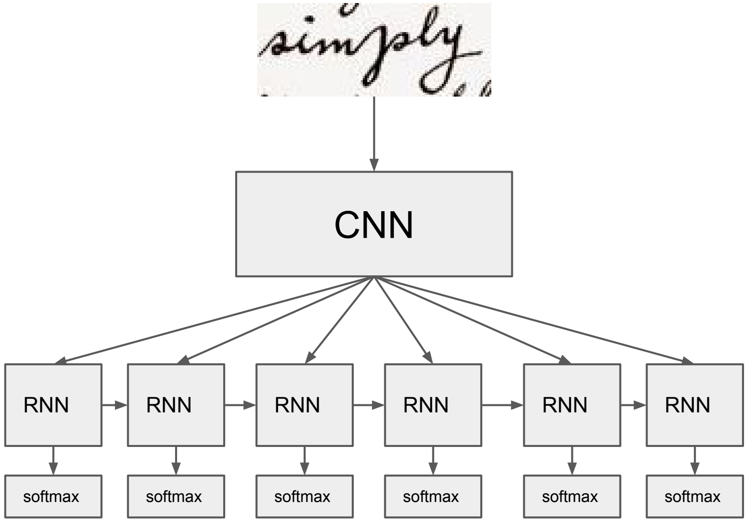


Reading Text in the Wild with Convolutional Neural Networks

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, Andrew Zisserman

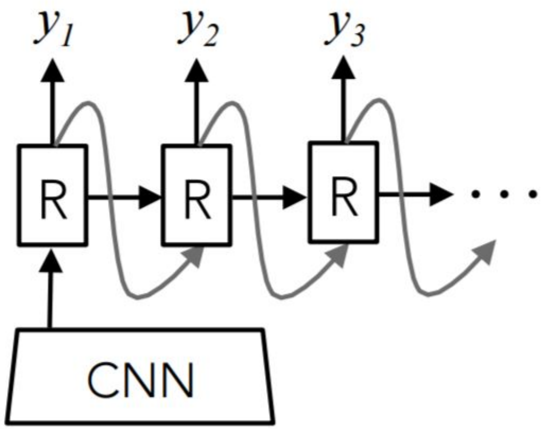
Without Char Segmentation using RNN

- CNN for image feature extraction.
- Duplicate CNN features into a sequence of Maxlen.
- RNN followed by softmax gives probabilities of alphabets of length Maxlen.



Scene text with Char Level Language Modelling

- The extracted image feature is sent to RNN only at the first time step.
- The predicted character y_{t-1} of RNN at time t−1 is fed to the RNN at time t until we obtain an end-of-word (EOW) label.
- Inspired by image captioning and text generation models.



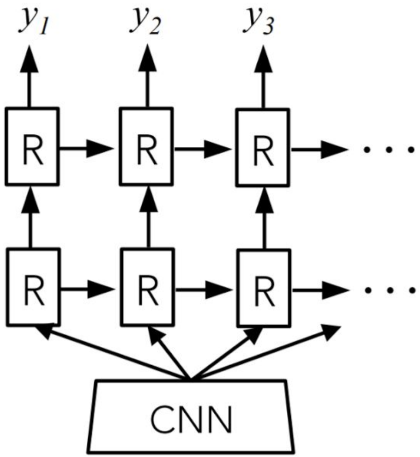
Single Layer, Captioning Style

Base CNN + $\mathbf{RNN_{1c}}$

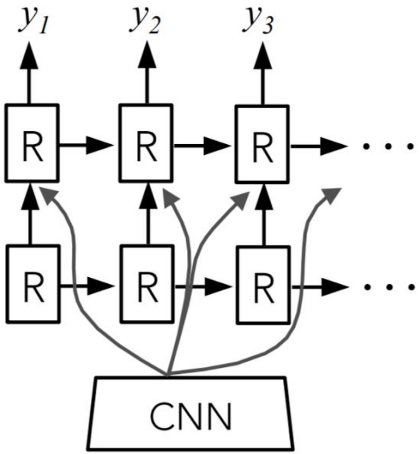
Recursive Recurrent Nets with Attention Modeling for OCR in the Wild

Chen-Yu Lee, Simon Osindero

Complex Models



Deeper RNNs

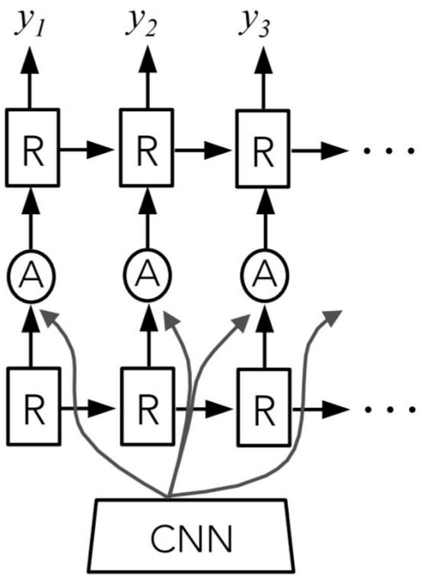


Char Level
Modelling

See : https://arxiv.org/abs/1603.03101

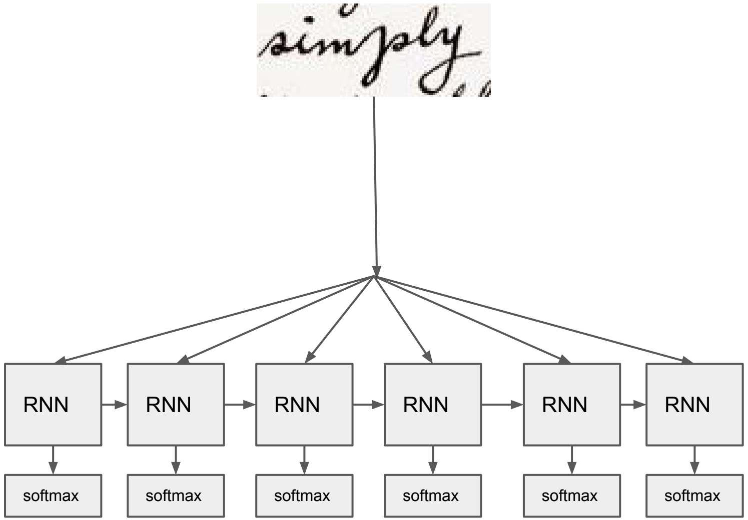
With Attention Modeling

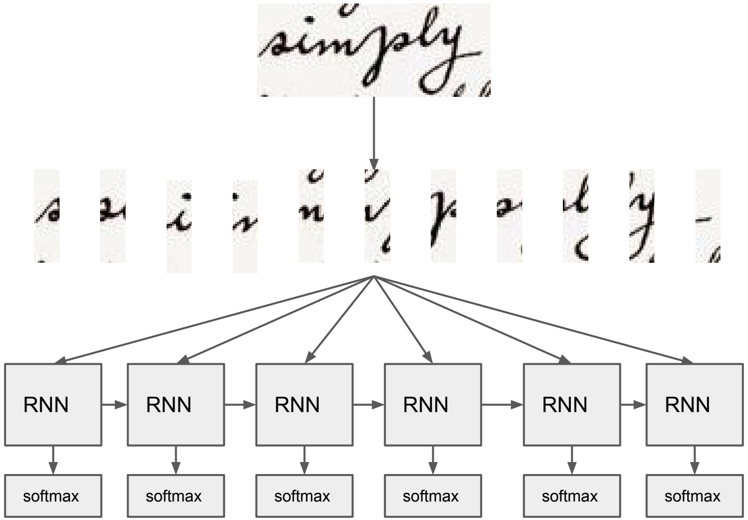
- Attention model allows the focus of the patches corresponding to the ith character while predicting it.
- First level RNN learns char level lang. model.
- Compute a context vector
  - $T_t = f_{attension}(I, s_t) = \tanh(\phi(I) + \chi(s_t))$
  - $\alpha = \mathrm{softmax}(T_t)$
  - $C_t = \alpha \circ I$
- Feed in C_t to second level RNN.
- ϕ, χ are neural networks



See : https://arxiv.org/abs/1603.03101

Why can't we split image into windows?

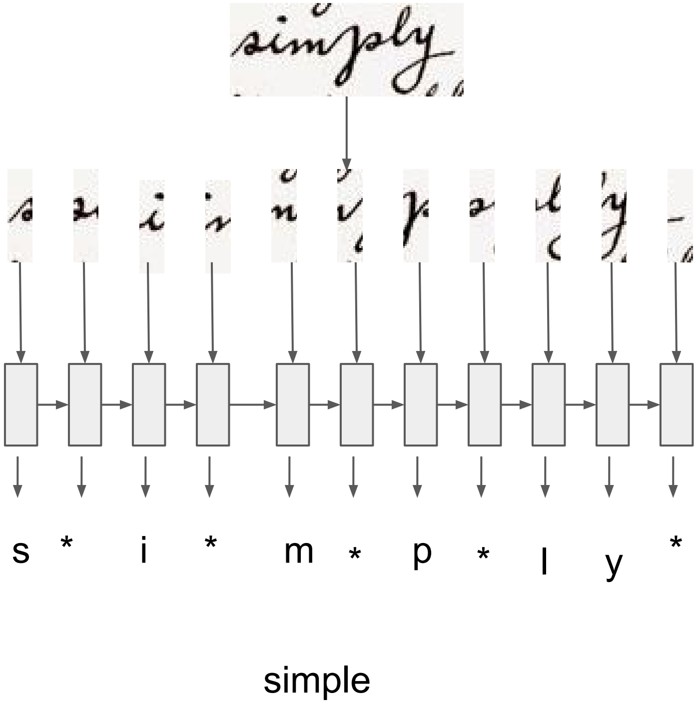




Input sequence and output sequence are not aligned!

Connectionist Temporal Classification (CTC)

- Removes need for char segmentation.
- Used with there is a mismatch between the input sequence and output sequence.
- Introduce an extra character blank character in to the alphabet (*).
- Decode s**i***mp*l*e, *s*i*m**p*l*e**, *sim**p*l*e** as simple.
- How do you write the loss function?



[Connectionist temporal classification: labelling unsegmented ...](dl.acm.org/citation.cfm?id=1143891)
dl.acm.org/citation.cfm?id=1143891
by A Graves - 2006

CTC Loss

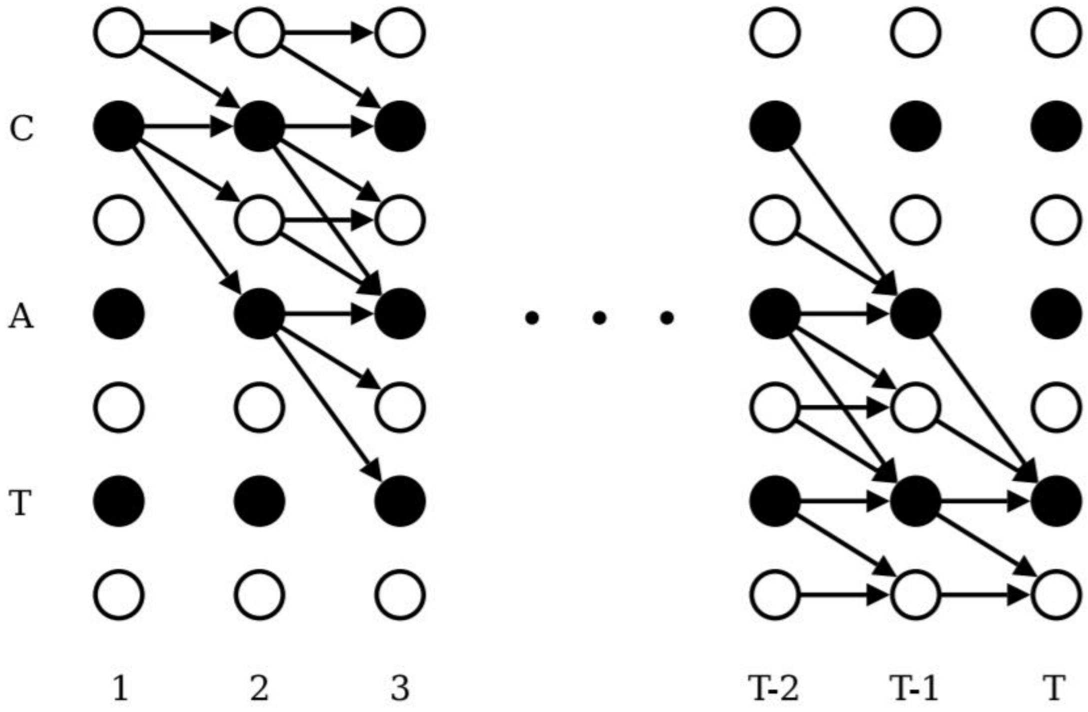
- Let B be the decoding function. i.e.

  B(s**i***mp*l*e) = B(*s*i*m**p*l*e**) = B(*sim**p*l*e**) = simple

- $p(\text{simple}) = \sum_{\text{w that decodes to simple}} p(w)$
- Loss = 1 - p(simple)
- But how to compute the summation with exponential number of terms?

Connectionist temporal classification: labelling unsegmented ...
dl.acm.org/citation.cfm?id=1143891
by A Graves - 2006

CTC Loss



CTC loss with Dynamic Programming

For some sequence $\mathbf{q}$ of length r, denote by $\mathbf{q}_{1:p}$ and $\mathbf{q}_{r-p:r}$ its first and last p symbols respectively. Then for a labelling $\mathbf{l}$, define the forward variable $\alpha_t(s)$ to be the total probability of $\mathbf{l}_{1:s}$ at time t. i.e.

$$\alpha_t(s) \stackrel{\text{def}}{=} \sum_{\substack{\pi \in N^T: \\ \mathcal{B}(\pi_{1:t}) = \mathbf{l}_{1:s}}} \prod_{t'=1}^{t} y^{t'}_{\pi_{t'}}. \qquad (5)$$

As we will see, $\alpha_t(s)$ can be calculated recursively from $\alpha_{t-1}(s)$ and $\alpha_{t-1}(s-1)$.

CTC loss with Dynamic Programming

y_k^t: output at time t for symbol k
l: label, l': label with blanks
Initialization:

$$\alpha_1(1) = y_b^1$$
$$\alpha_1(2) = y_{l_1}^1$$
$$\alpha_1(s) = 0, \forall s > 2$$

Recurrence relation:

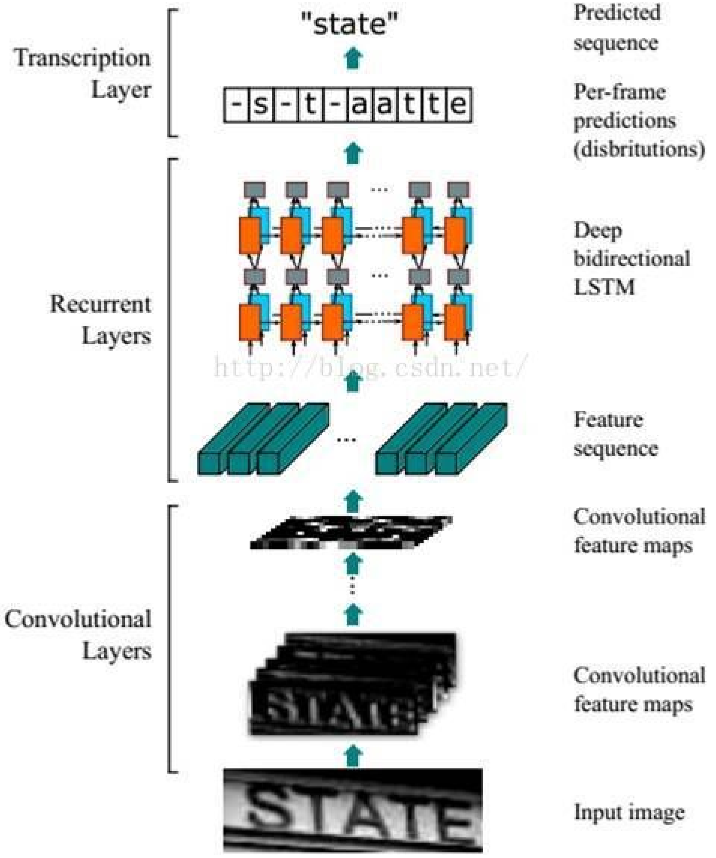
$$\alpha_t(s) = \begin{cases} \bar{\alpha}_t(s) y_{l'_s}^t & \text{if } l'_s = b \text{ or } l'_{s-2} = l'_s \\ (\bar{\alpha}_t(s) + \alpha_{t-1}(s-2)) y_{l'_s}^t & \\ \qquad \text{otherwise} & \end{cases}$$

$$\bar{\alpha}_t(s) = \alpha_{t-1}(s) + \alpha_{t-1}(s-1)$$

Finally, we have:

$$p(l|x) = \alpha_T(|l'|) + \alpha_T(|l'| - 1)$$

A CNN+RNN+CTC Model



An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition

Baoguang Shi, Xiang Bai, Cong Yao

CRNN Model Accuracy

	IIIT5k			SVT		IC03				IC13
	50	**1k**	**None**	**50**	**None**	**50**	**Full**	**50k**	**None**	**None**
ABBYY [34]	24.3	-	-	35.0	-	56.0	55.0	-	-	-
Wang *et al.* [34]	-	-	-	57.0	-	76.0	62.0	-	-	-
Mishra *et al.* [28]	64.1	57.5	-	73.2	-	81.8	67.8	-	-	-
Wang *et al.* [35]	-	-	-	70.0	-	90.0	84.0	-	-	-
Goel *et al.* [13]	-	-	-	77.3	-	89.7	-	-	-	-
Bissacco *et al.* [8]	-	-	-	90.4	78.0	-	-	-	-	87.6
Alsharif and Pineau [6]	-	-	-	74.3	-	93.1	88.6	85.1	-	-
Almazán *et al.* [5]	91.2	82.1	-	89.2	-	-	-	-	-	-
Yao *et al.* [36]	80.2	69.3	-	75.9	-	88.5	80.3	-	-	-
Rodrguez-Serrano *et al.* [30]	76.1	57.4	-	70.0	-	-	-	-	-	-
Jaderberg *et al.* [23]	-	-	-	86.1	-	96.2	91.5	-	-	-
Su and Lu [33]	-	-	-	83.0	-	92.0	82.0	-	-	-
Gordo [14]	93.3	86.6	-	91.8	-	-	-	-	-	-
Jaderberg *et al.* [22]	97.1	92.7	-	95.4	80.7*	**98.7**	**98.6**	93.3	**93.1***	**90.8***
Jaderberg *et al.* [21]	95.5	89.6	-	93.2	71.7	97.8	97.0	93.4	89.6	81.8
CRNN	**97.6**	**94.4**	**78.2**	**96.4**	**80.8**	**98.7**	97.6	**95.5**	89.4	86.7

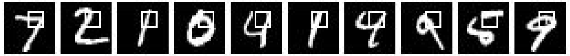
Which one should i use?

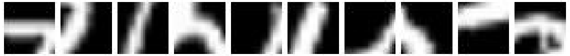
Depends on

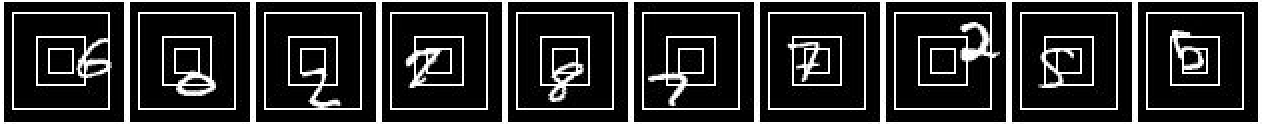
1. The number of different font styles.
2. Complexity of the language.
    a. Indic languages more complex than latin.
    b. Urdu/Arabic can be very complex.

Hard Attention Modelling

How to predict the window containing the next char?



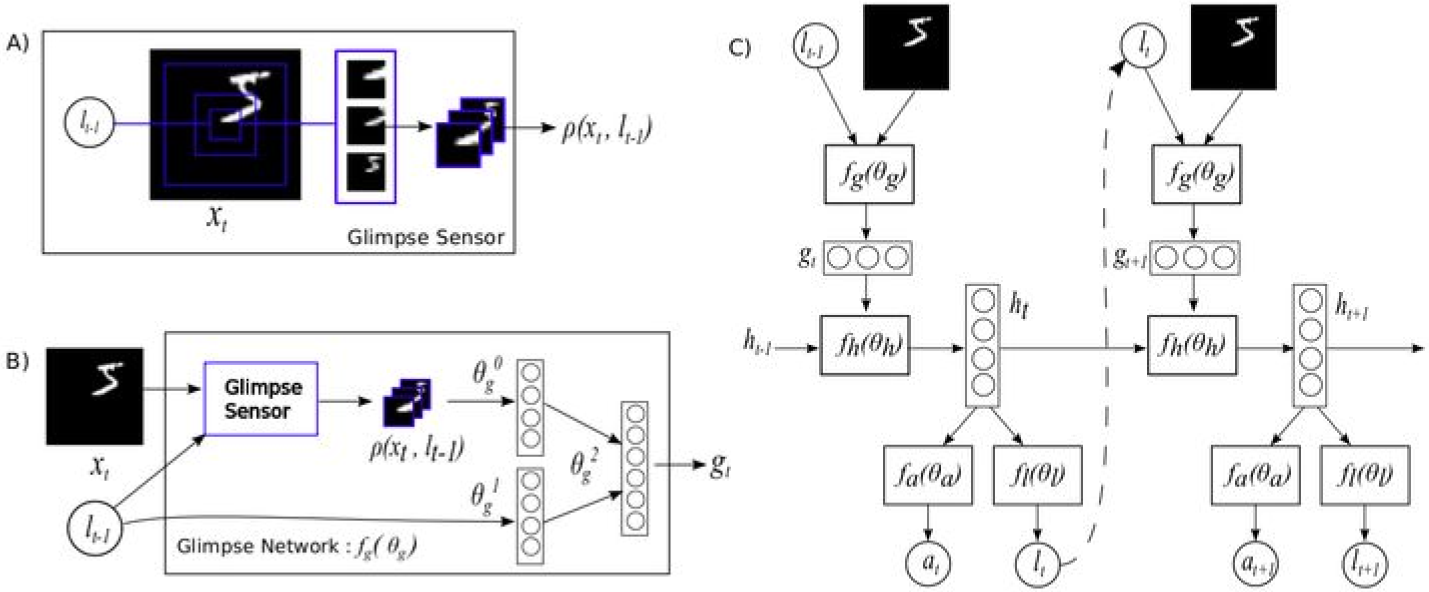




Recurrent Models of Visual Attention

Volodymyr Mnih, Nicolas Heess, Alex Graves, Koray Kavukcuoglu

Hard Attention Modelling



http://torch.ch/blog/2015/09/21/rmva.html