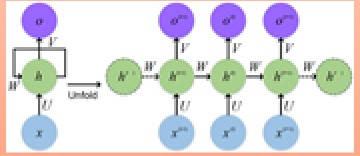
Focus for this lecture

Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

- Introduction to RNN
- RNN

Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

- Introduction to RNN

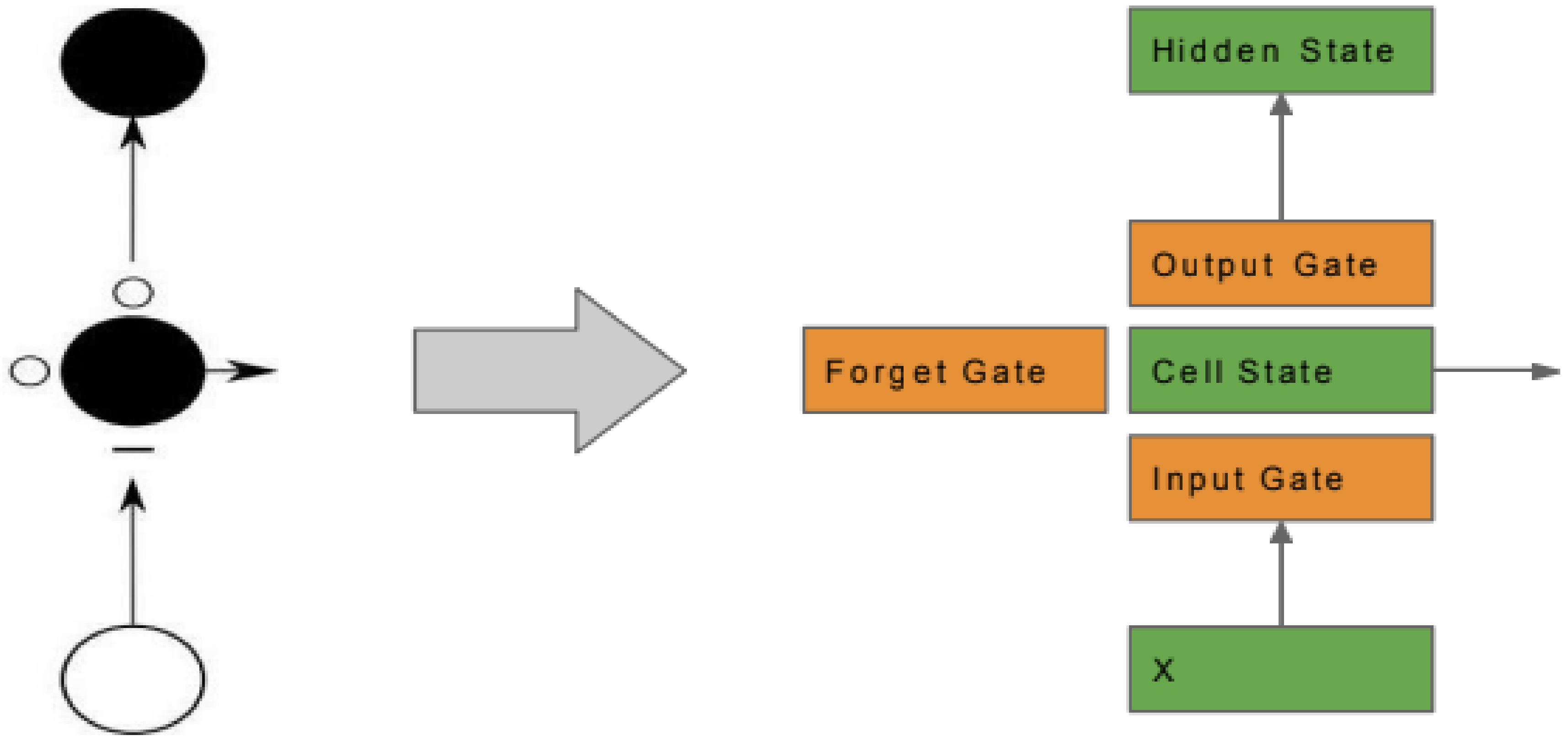


Introduction to RNNs

Why RNNs ?

- Intelligent systems (Networks) need memory.
- Many inputs are sequential in nature.
- Concepts have long term dependencies.
  - Not just one or two steps backwards.
  - Eg. What controls tomato price of tomorrow?
- Popular networks (Eg. CNNs) do not have cycles.

Cell with Memory



Generating poetry with RNNs

Sonnet 116 – Let me not ...

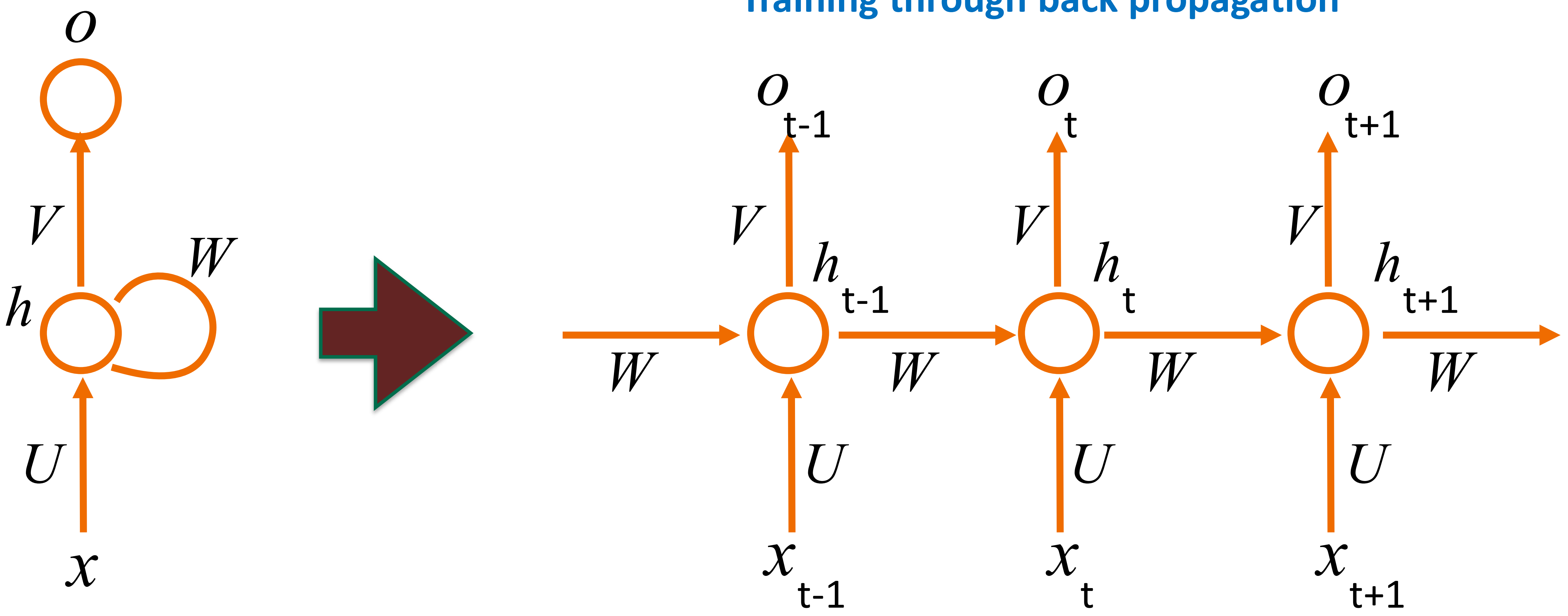
by William Shakespeare

Let me not to the marriage of true minds
        Admit impediments. Love is not love
Which alters when it alteration finds,
        Or bends with the remover to remove:
O no! it is an ever-fixed mark
        That looks on tempests and is never shaken;
It is the star to every wandering bark,
        Whose worth's unknown, although his height be taken.
Love's not Time's fool, though rosy lips and cheeks
        Within his bending sickle's compass come:
Love alters not with his brief hours and weeks,
        But bears it out even to the edge of doom.
If this be error and upon me proved,
        I never writ, nor no man ever loved.

Andrej Karpathy

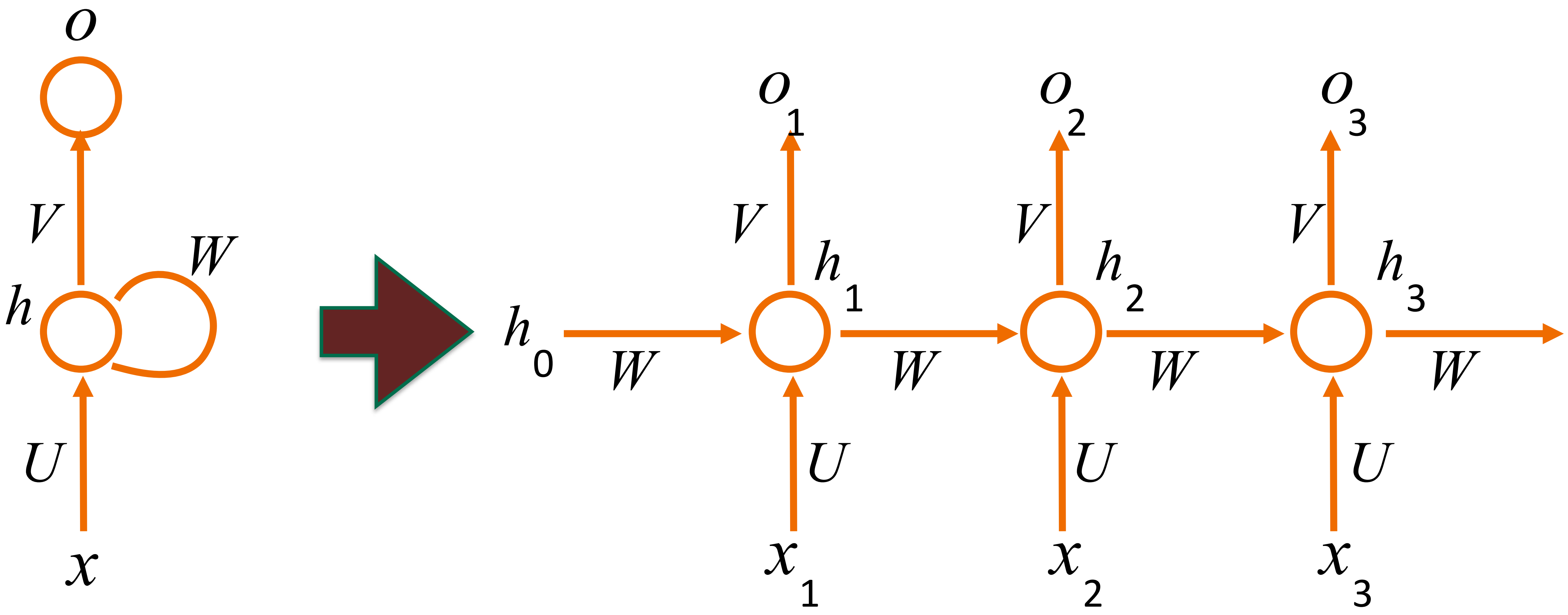
RNN basic architecture

Training through back propagation



$$h_t = f(Ux_t + Wh_{t-1}) \qquad o_t = \mathrm{softmax}(Vh_t)$$

RNN basic architecture

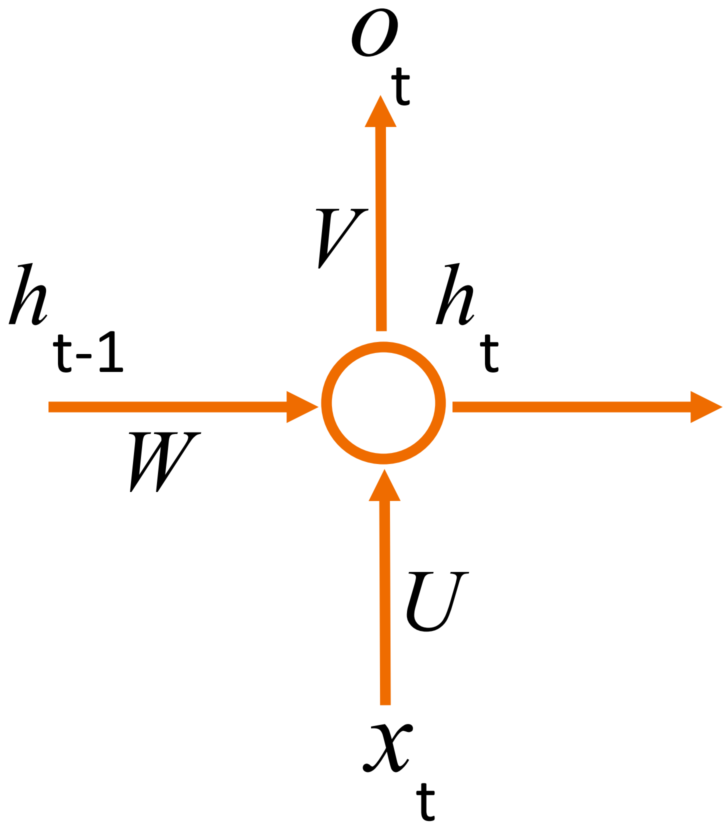


$$h_t = f(Ux_t + Wh_{t-1}) \qquad o_t = \mathrm{softmax}(Vh_t)$$

RNN basic architecture

- x_t - input at time step t
- h_t - hidden state at time step t (memory of the network)
- o_t - output at time step t
- U, V, W are parameters (shared across all layers)

RNN basic architecture



$$h_t = f(Ux_t + Wh_{t-1})$$

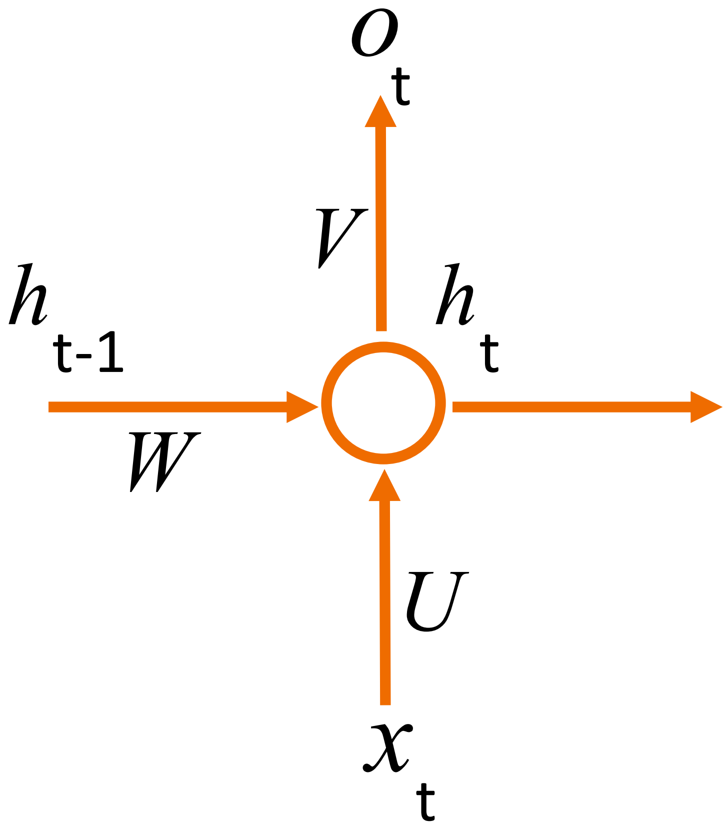
$$U = \begin{pmatrix} 0.5 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}$$

$$W = \begin{pmatrix} 0.4 & 0.5 \\ 0.3 & 0.5 \end{pmatrix} \quad x_t = \begin{pmatrix} 0.4 \\ 0.2 \end{pmatrix} \quad h_{t-1} = \begin{pmatrix} 0.3 \\ 0.8 \end{pmatrix}$$

$$\begin{array}{ccccccc} & U & x_t & & W & h_{t-1} & \\ h_t = \tanh\Bigg(& \begin{pmatrix} 0.5 & 0.4 \\ 0.4 & 0.6 \end{pmatrix} & \begin{pmatrix} 0.4 \\ 0.2 \end{pmatrix} & + & \begin{pmatrix} 0.4 & 0.5 \\ 0.3 & 0.5 \end{pmatrix} & \begin{pmatrix} 0.3 \\ 0.8 \end{pmatrix} & \Bigg) \end{array}$$

$$h_t = \tanh\left(\begin{pmatrix} .28 \\ .28 \end{pmatrix} + \begin{pmatrix} .52 \\ .29 \end{pmatrix}\right) = \tanh\begin{pmatrix} .88 \\ .77 \end{pmatrix} = \begin{pmatrix} .66 \\ .64 \end{pmatrix}$$

RNN basic architecture



$$h_t = f(Ux_t + Wh_{t-1})$$

$$o_t = softmax(Vh_t)$$

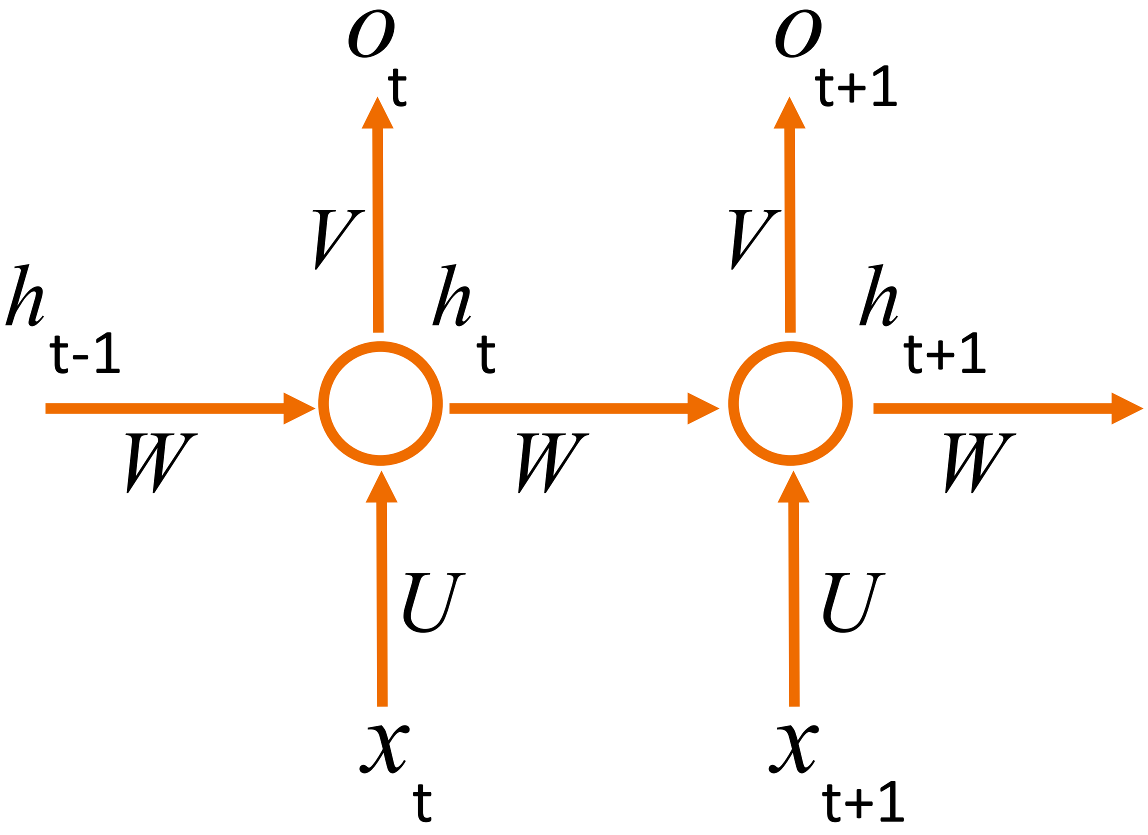
$$U = \begin{pmatrix} 0.5 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}$$

$$W = \begin{pmatrix} 0.4 & 0.5 \\ 0.3 & 0.5 \end{pmatrix} \quad x_t = \begin{pmatrix} 0.4 \\ 0.2 \end{pmatrix} \quad h_{t-1} = \begin{pmatrix} 0.3 \\ 0.8 \end{pmatrix}$$

$$h_t = \begin{pmatrix} .66 \\ .64 \end{pmatrix} \quad V = \begin{pmatrix} 0.4 & 0.7 \\ 0.3 & 0.1 \end{pmatrix}$$

$$O_t = softmax\left(\underbrace{\begin{pmatrix} 0.4 & 0.7 \\ 0.3 & 0.1 \end{pmatrix}}_{V}\underbrace{\begin{pmatrix} .66 \\ .64 \end{pmatrix}}_{h_t}\right) = \begin{pmatrix} .61 \\ .39 \end{pmatrix}$$

RNN basic architecture



$$\boldsymbol{h_{t+1}} = \boldsymbol{f}(\boldsymbol{U}\boldsymbol{x_{t+1}} + \boldsymbol{W}\boldsymbol{h_t})$$

$$\boldsymbol{U} = \begin{pmatrix} 0.5 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}$$

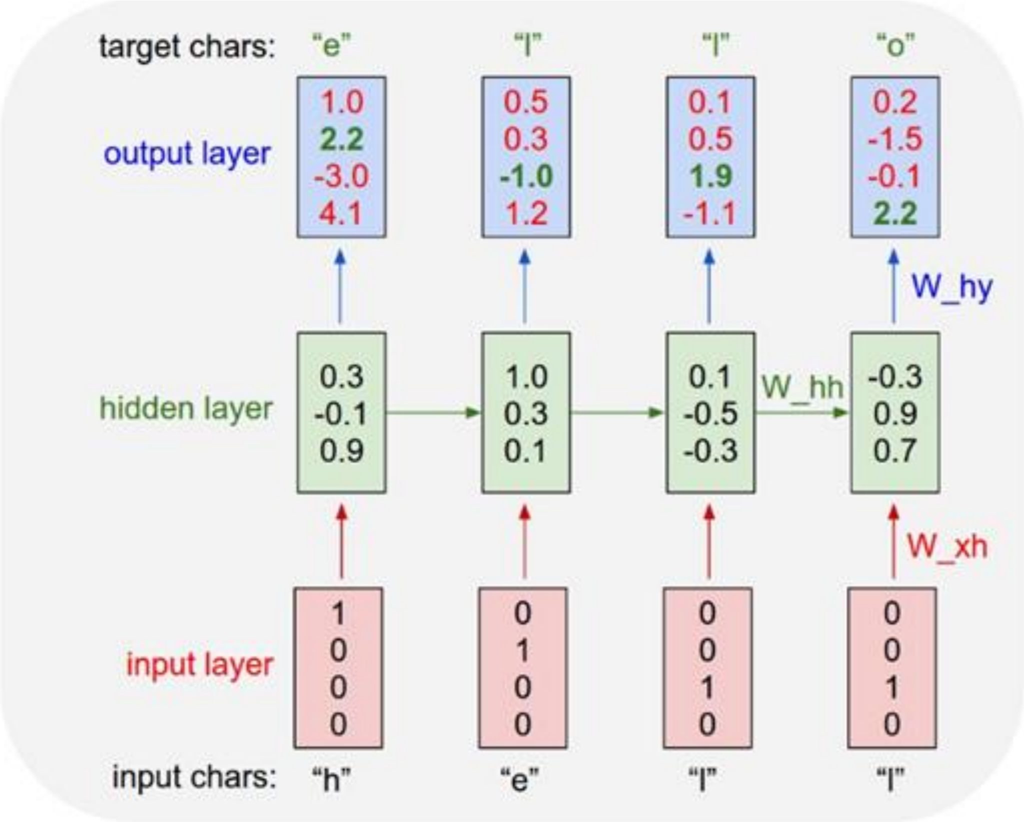
$$\boldsymbol{W} = \begin{pmatrix} 0.4 & 0.5 \\ 0.3 & 0.5 \end{pmatrix} \quad \boldsymbol{x_{t+1}} = \begin{pmatrix} 0.5 \\ 0.4 \end{pmatrix} \quad \boldsymbol{h_t} = \begin{pmatrix} .66 \\ .64 \end{pmatrix}$$

$$\begin{array}{ccccccc} & \boldsymbol{U} & \boldsymbol{x_{t+1}} & & \boldsymbol{W} & \boldsymbol{h_t} & \\ \boldsymbol{h_{t+1}} = \mathbf{tanh}\Bigg(& \begin{pmatrix} 0.5 & 0.4 \\ 0.4 & 0.6 \end{pmatrix} & \begin{pmatrix} 0.5 \\ 0.4 \end{pmatrix} & + & \begin{pmatrix} 0.4 & 0.5 \\ 0.3 & 0.5 \end{pmatrix} & \begin{pmatrix} .66 \\ .64 \end{pmatrix} & \Bigg) \end{array}$$

Character Level Language Modelling

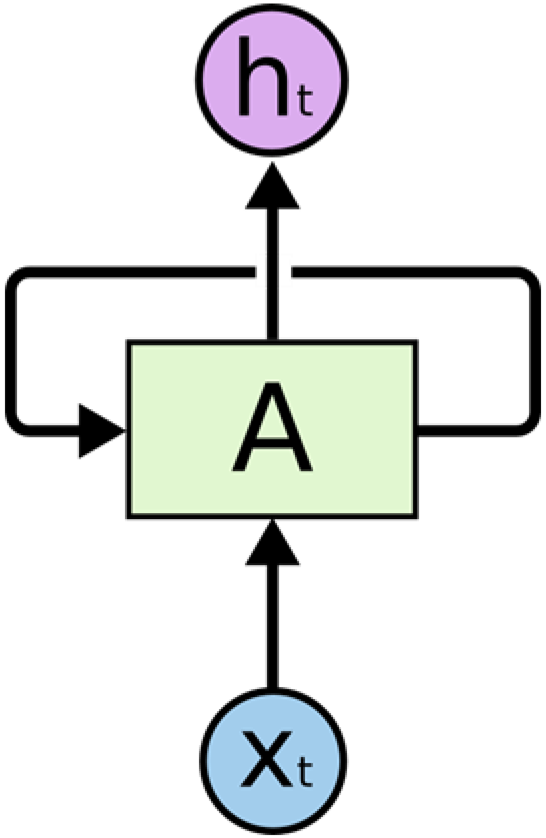
Task:

- Predicting the next character given the current character



It may be today

- RNNs




```
*
* Increment the size file of the new incorrect UI_FILTER
* of the size generatively.
*/
tatic int indicate_policy(void)

 int error;
 if (fd == MARN_EPT) {
   /*
    * The kernel blank will coeld it to userspace.
    */
   if (ss->segment < mem_total)
     unblock_graph_and_set_blocked();
   else
     ret = 1;
   goto bail;
 }
 segaddr = in_SB(in.addr);
 selector = seg / 16;
 setup_works = true;
 for (i = 0; i < blocks; i++) {
   seq = buf[i++];
   bpf = bd->bd.next + i * search;
   if (fd) {
     current = blocked;
```

A. Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/

And Shakespeare

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.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

and Algebraic Geometry!!

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ *(by the formal open covering* X *and a single map* $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ *over* U *compatible with the complex*

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ *is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If* T *is surjective we may assume that* T *is connected with residue fields of* S*. Moreover there exists a closed subspace* $Z \subset X$ *of* X *where* U *in* X' *is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem*

(1) f *is locally of finite type. Since* $S = \mathrm{Spec}(R)$ *and* $Y = \mathrm{Spec}(R)$.

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\ldots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. ☐

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{\mathcal{X},\ldots,0}$.

Lemma 0.2. *Let* X *be a locally Noetherian scheme over* S*,* $E = \mathcal{F}_{X/S}$*. Set* $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$*. Since* $\mathcal{I}^n \subset \mathcal{I}^n$ *are nonzero over* $i_0 \leq \mathfrak{p}$ *is a subset of* $\mathcal{J}_{n,0} \circ \overline{A}_2$ *works.*

Lemma 0.3. *In Situation ??. Hence we may assume* $\mathfrak{q}' = 0$.

Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (??). On the other hand, by Lemma ?? we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F-algebra where δ_{n+1} is a scheme over S. ☐

NSE talent sprint | IIIT Hyderabad

at first:

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

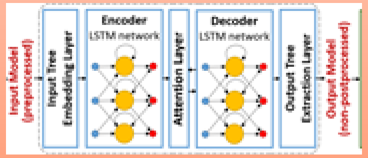
Summary

- RNNs are powerful networks
  - With feedback
  - With memory
  - Hard to "fully" understand.
- CNNs are very useful for a class of tasks
  - Both in Image and Text.
- Shall revisit them again

Questions?

Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

✓ Introduction to RNN
- RNN



Why RNN's?

remember the history

Machine learning with sequential data

- Turn an input sequence into an output sequence that lives in a different domain or similar domain
  - e.g. sequence of sound pressures into a sequence of words
  - e.g. sequence of images into sequence of words
  - e.g. language translation

Machine learning with sequential data

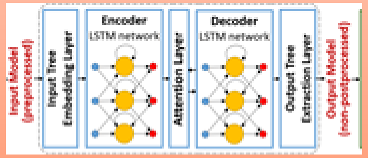
- Learn a model to predict the next term in the input sequence
  - e.g. predict next word given a set of words (we use this on everyday basis)
  - e.g. stock market time series data
- Predicting the next term in a sequence is supervised or unsupervised learning?
- Uses methods designed for supervised learning but doesn’t require a separate teaching signal

RNN has been a natural choice

- In many NLP tasks
- In time series data (stock prices, web logs, weather etc.)
- Fill in the blanks or predict what happens next in time
- Sequence to sequence (translation, speech recognition etc.)
- Sometimes for classifying a sequence (sentiment analysis)

Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

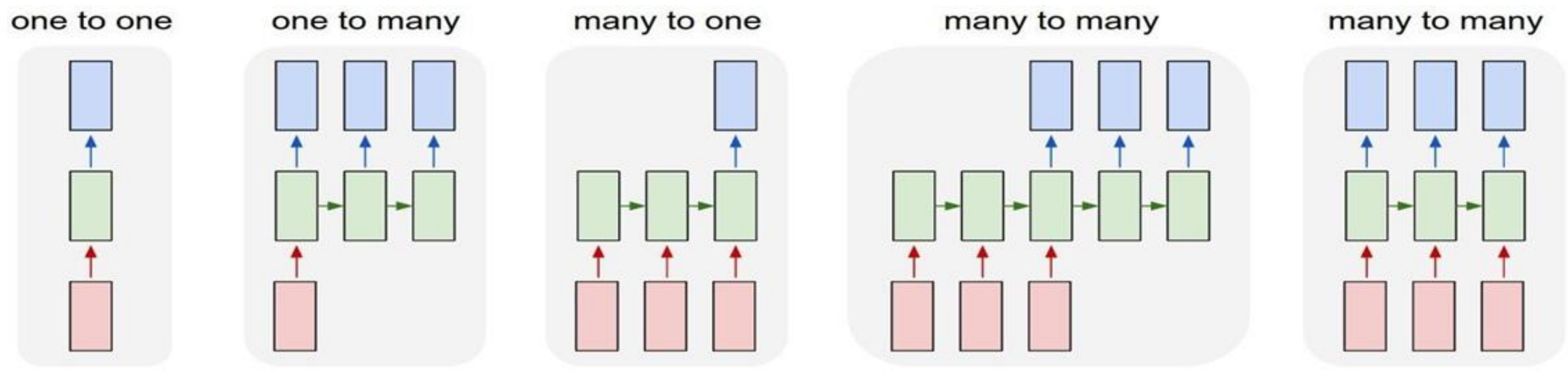
- ✓ Introduction to RNN
- RNN



Applications and Use cases

remember the history

Recurrent Networks offer a lot of flexibility:



vanilla neural networks

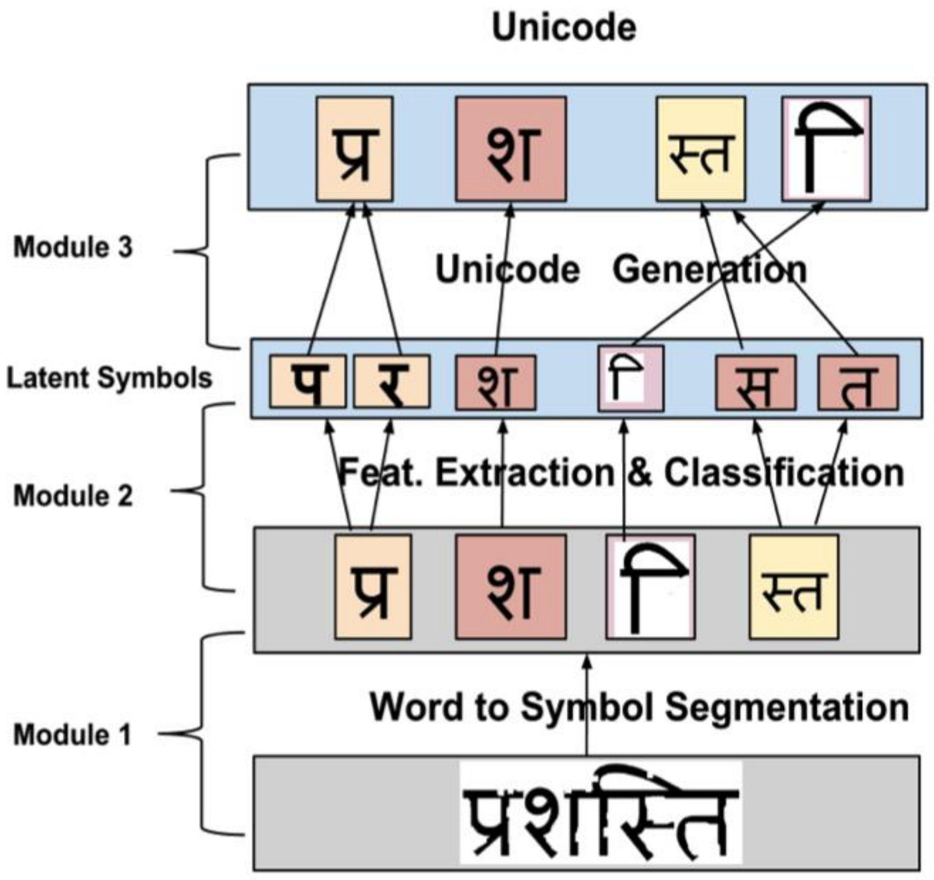
e.g. **image captioning**
image -> sequence of words

e.g. **sentiment classification**
sequence of words -> sentiment

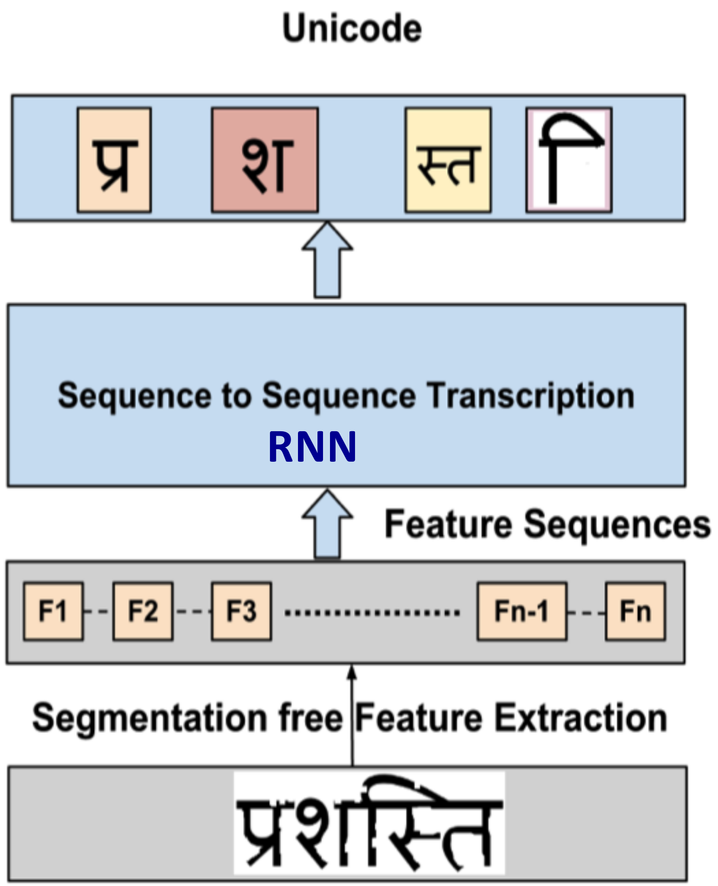
e.g. **machine translation**

e.g. **video classification on frame level**

OCR as Translation: RNN-OCR



Traditional OCR

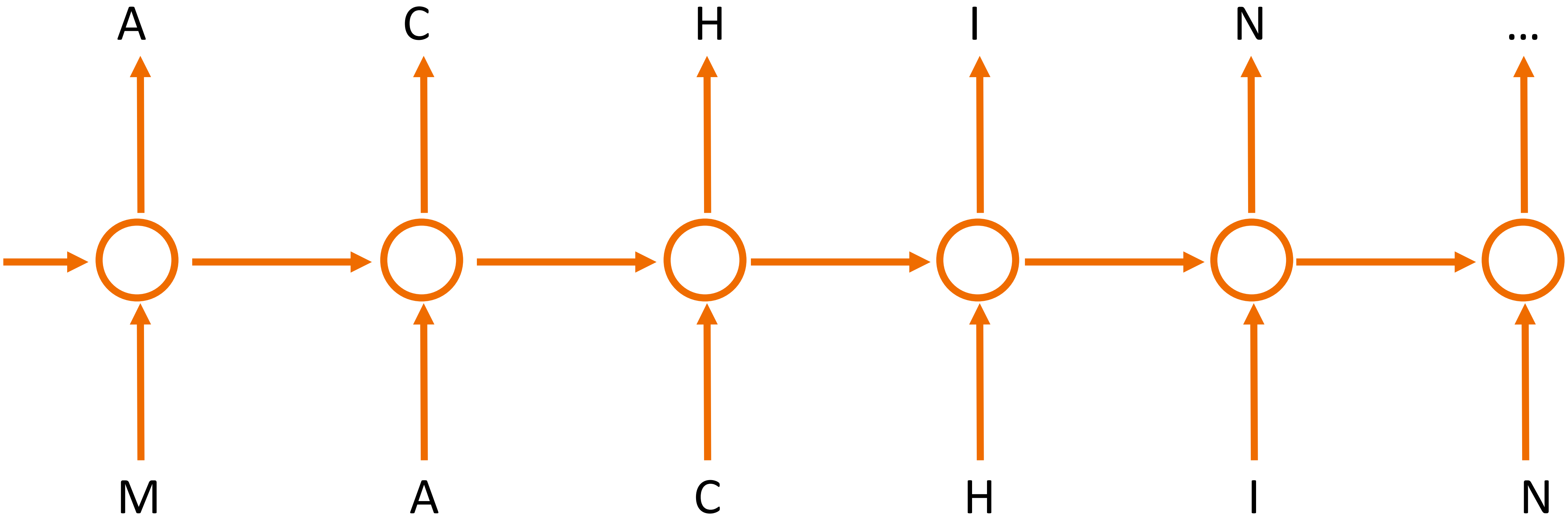


RNN OCR

Overcome: Segmentation, Unicode Reordering,

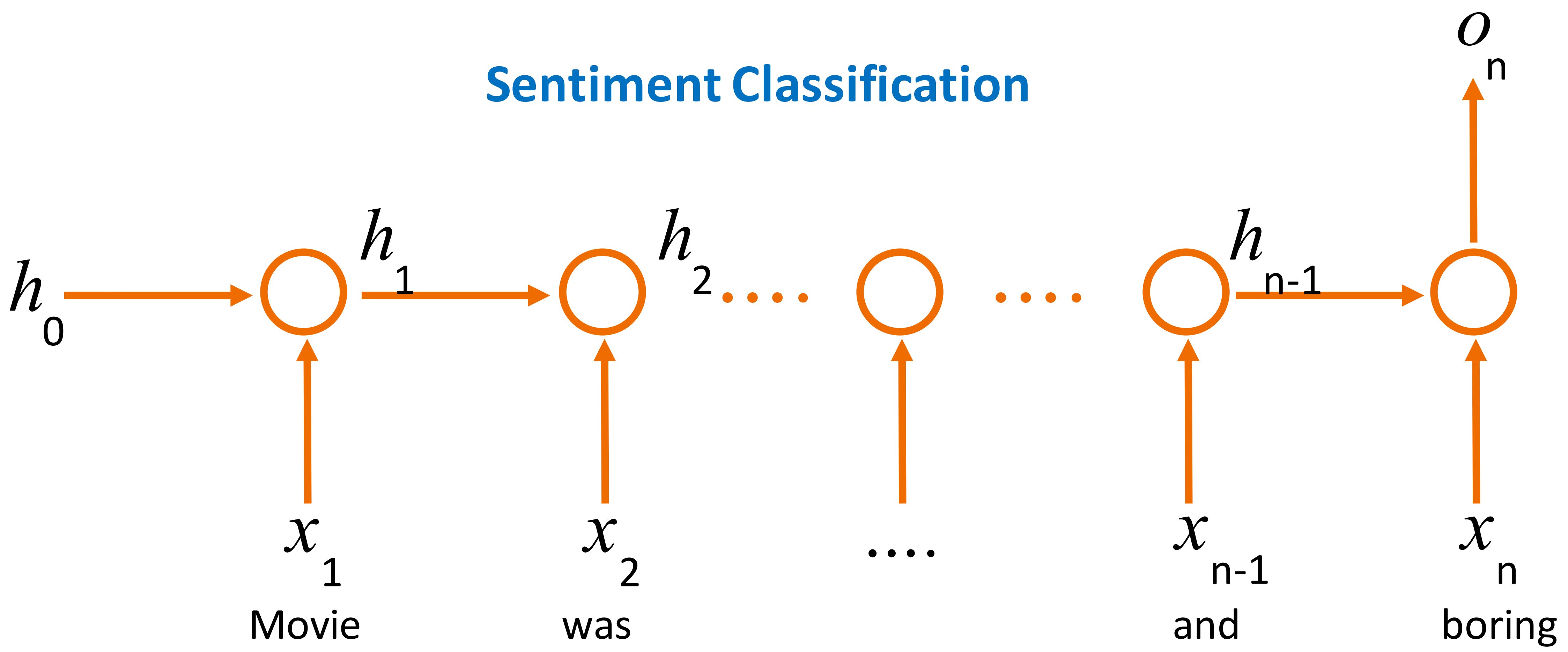
Sankaran and Jawahar, Devanagari Text Recognition: A Transcription Based Formulation ICDAR 2013

Lets design some Recurrent networks

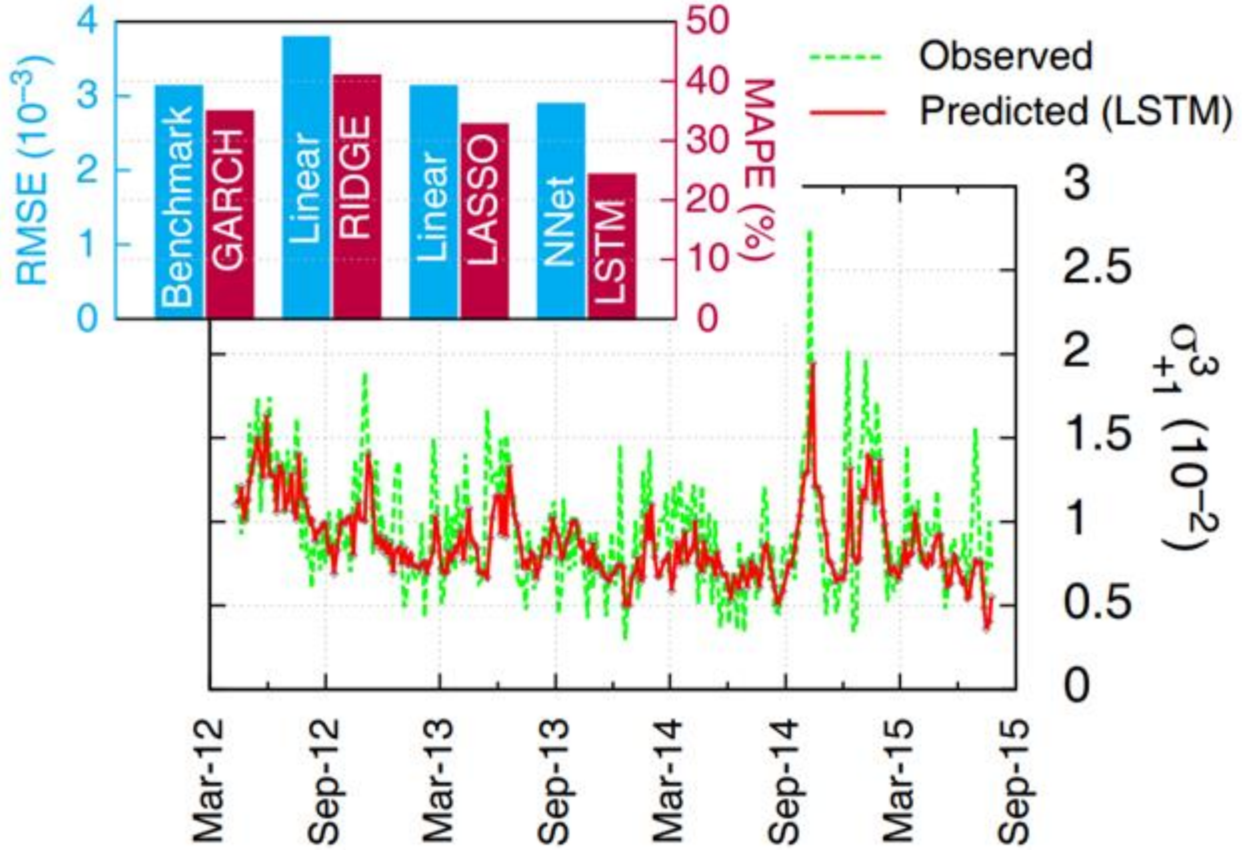


Lets design some Recurrent networks

Sentiment Classification

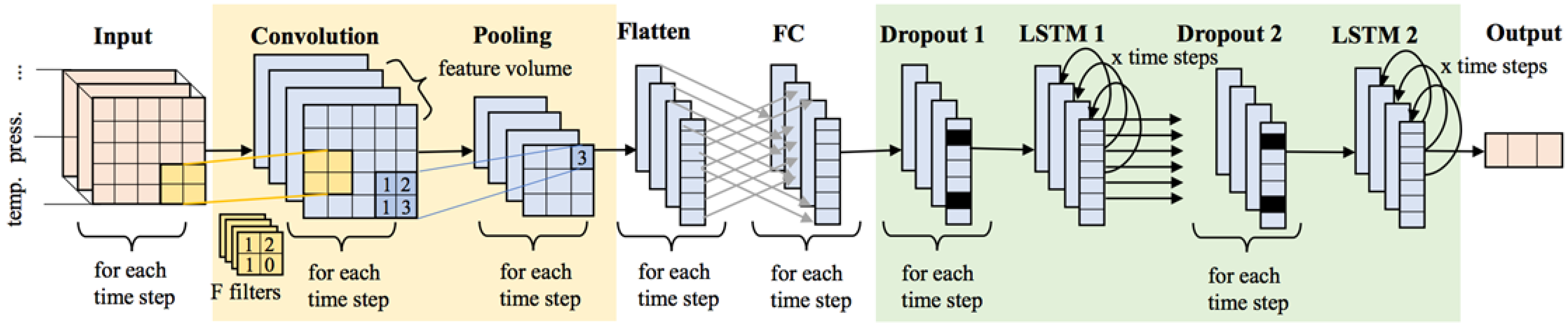


Lets design some Recurrent networks



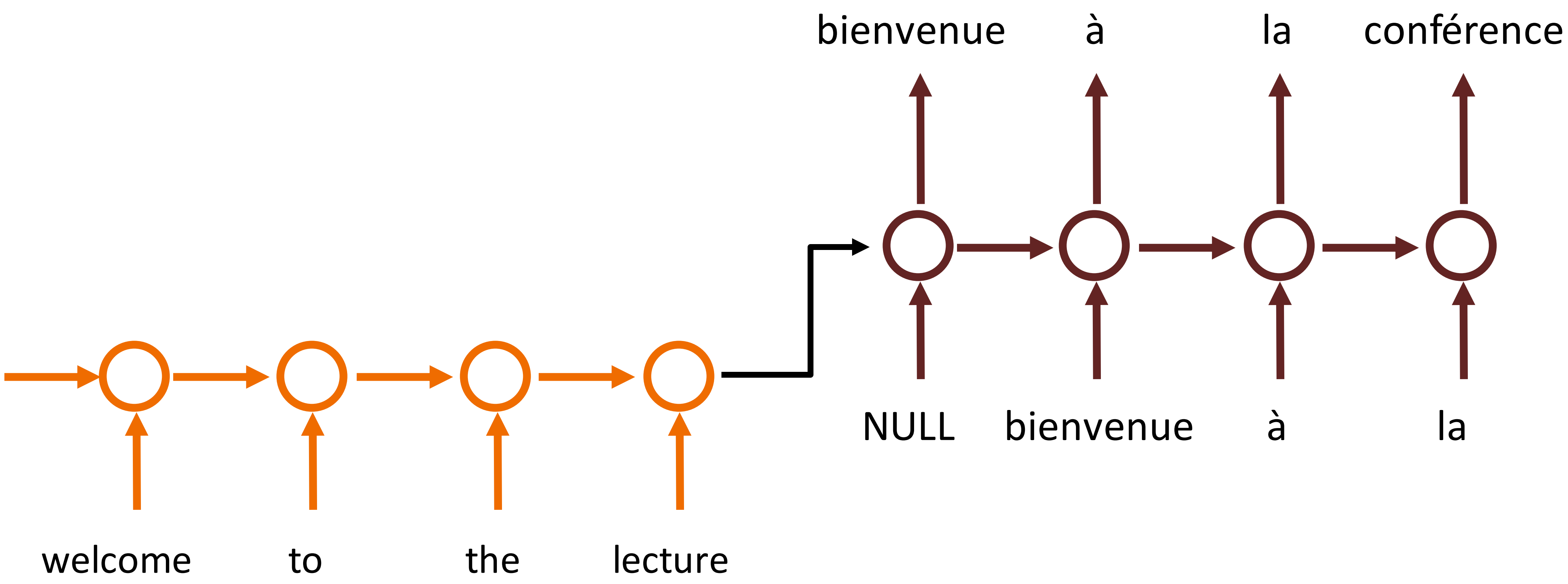
Xiong et al. Deep Learning Stock Volatility with Google Domestic Trends. arXiv 2016

Weather Prediction



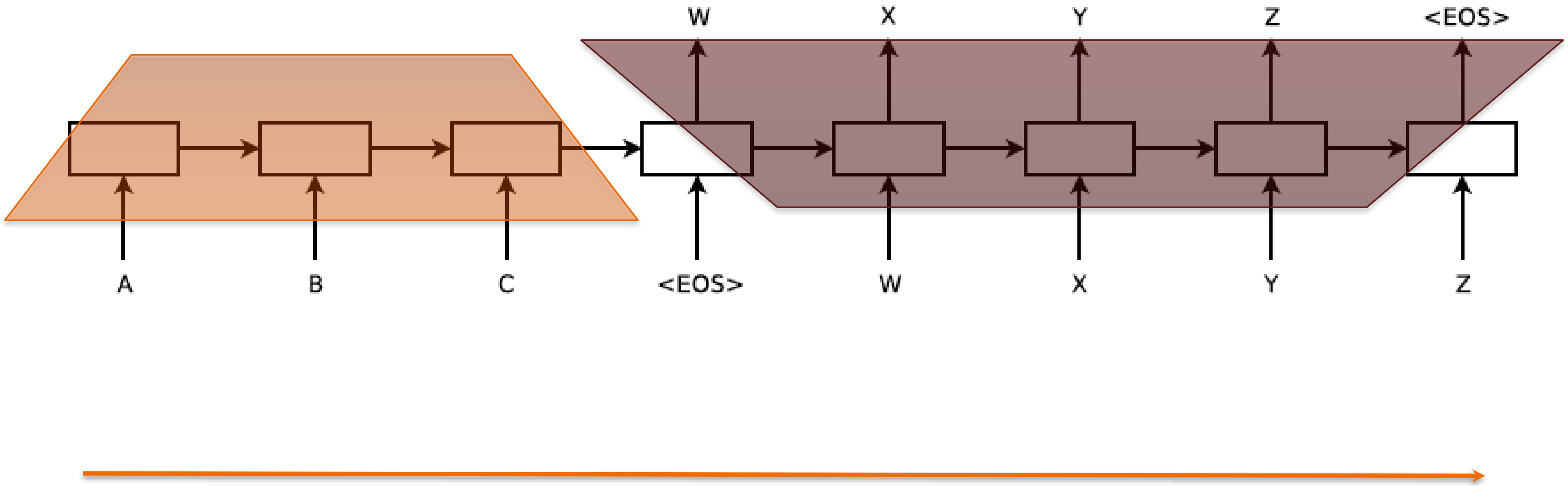
Roesch et al. Computer Graphics Forum, 2017

Lets design some Recurrent networks



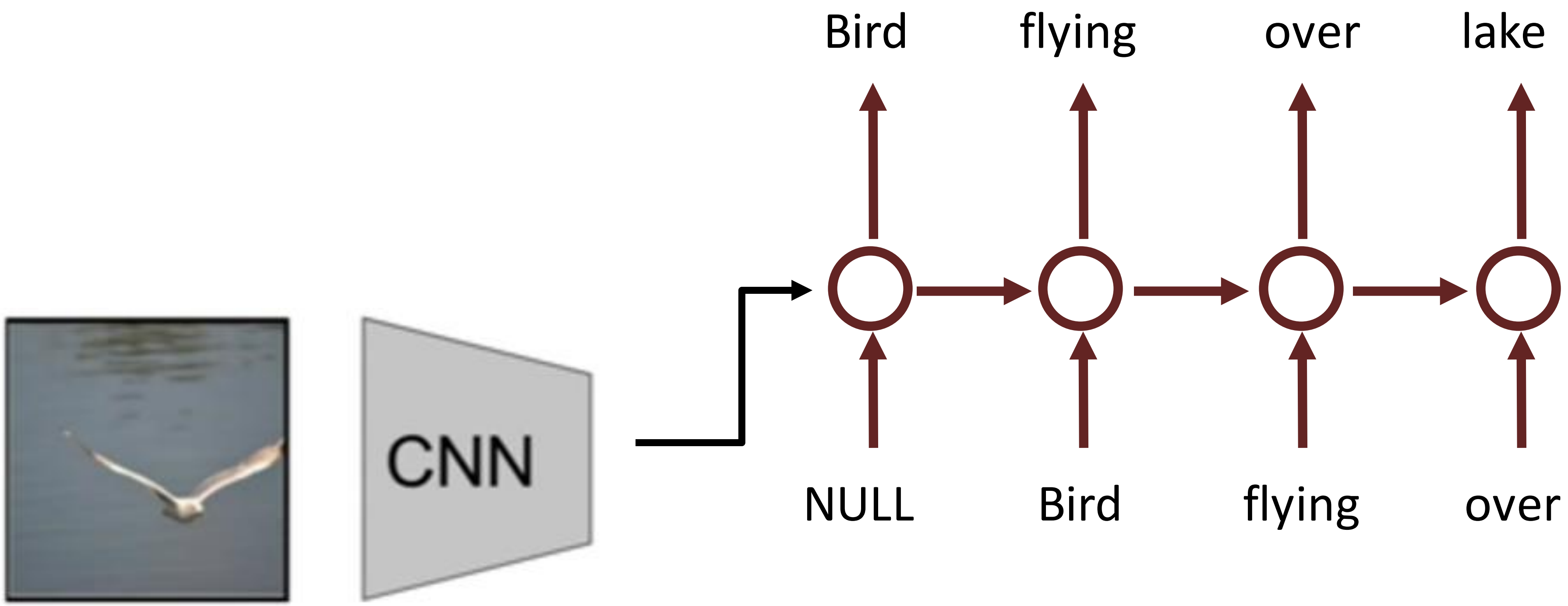
Neural Machine Translation

- Model



Sutskever et al. 2014

Lets design some Recurrent networks



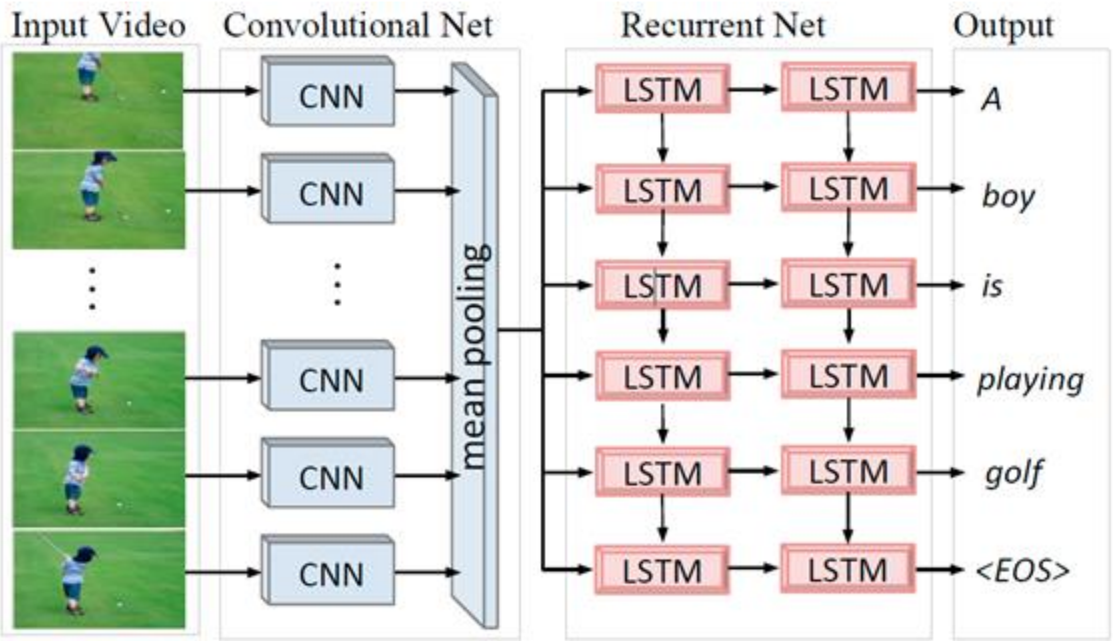
Machine learning with sequential data

- Input is an image and output is a sequence of words

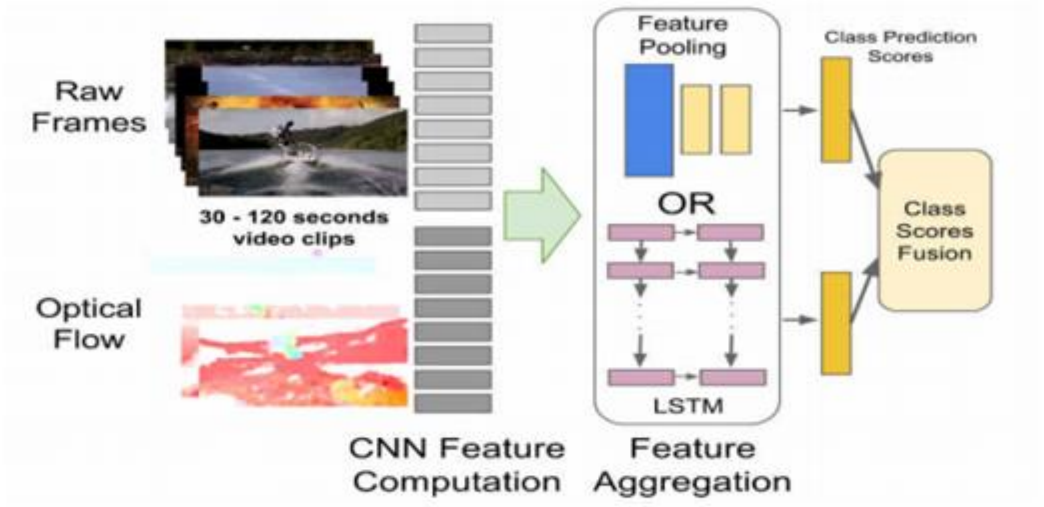
A horse carrying a large load of hay and two people sitting on it

Hybrid Architectures

Video Captioning



Video Classification

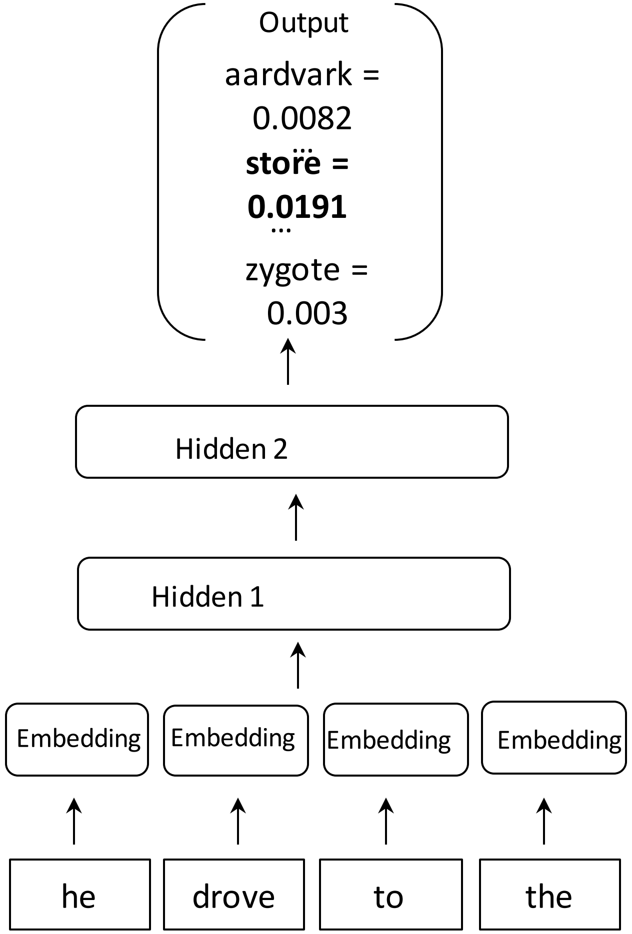


“Shannon Game”

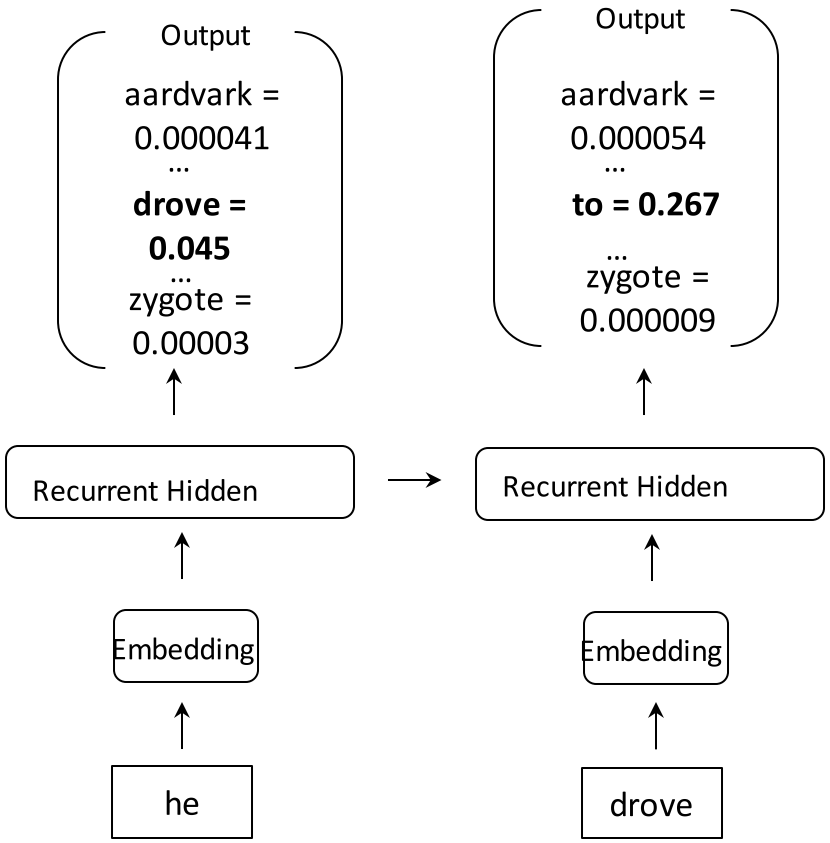
- Claude E. Shannon. “Prediction and Entropy of Printed English”, Bell System Technical Journal 30:50-64. 1951.
- Predict the next word, given (n-1) previous words
- Determine probability of different sequences by examining training corpus

Neural Network Language Models (NNLMs)

Feed-forward NNLM



Recurrent NNLM

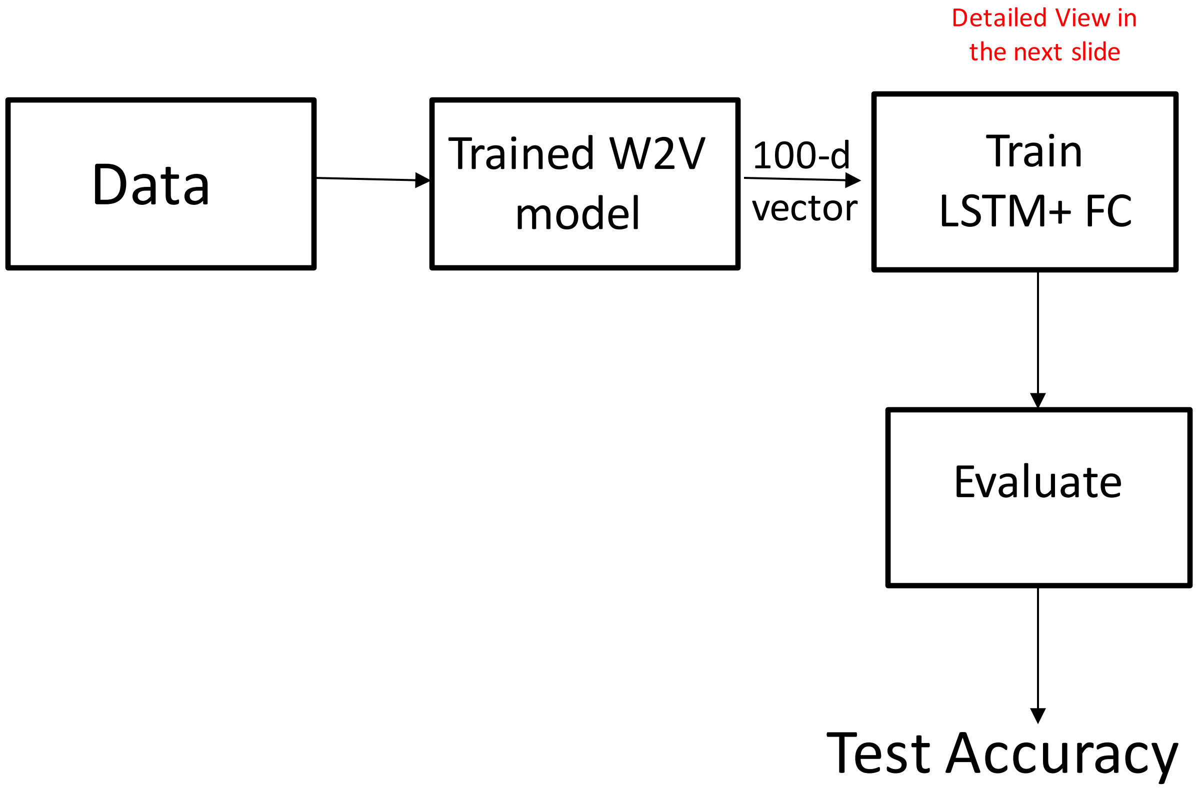


Casestudy

Sentence Level Author Identification

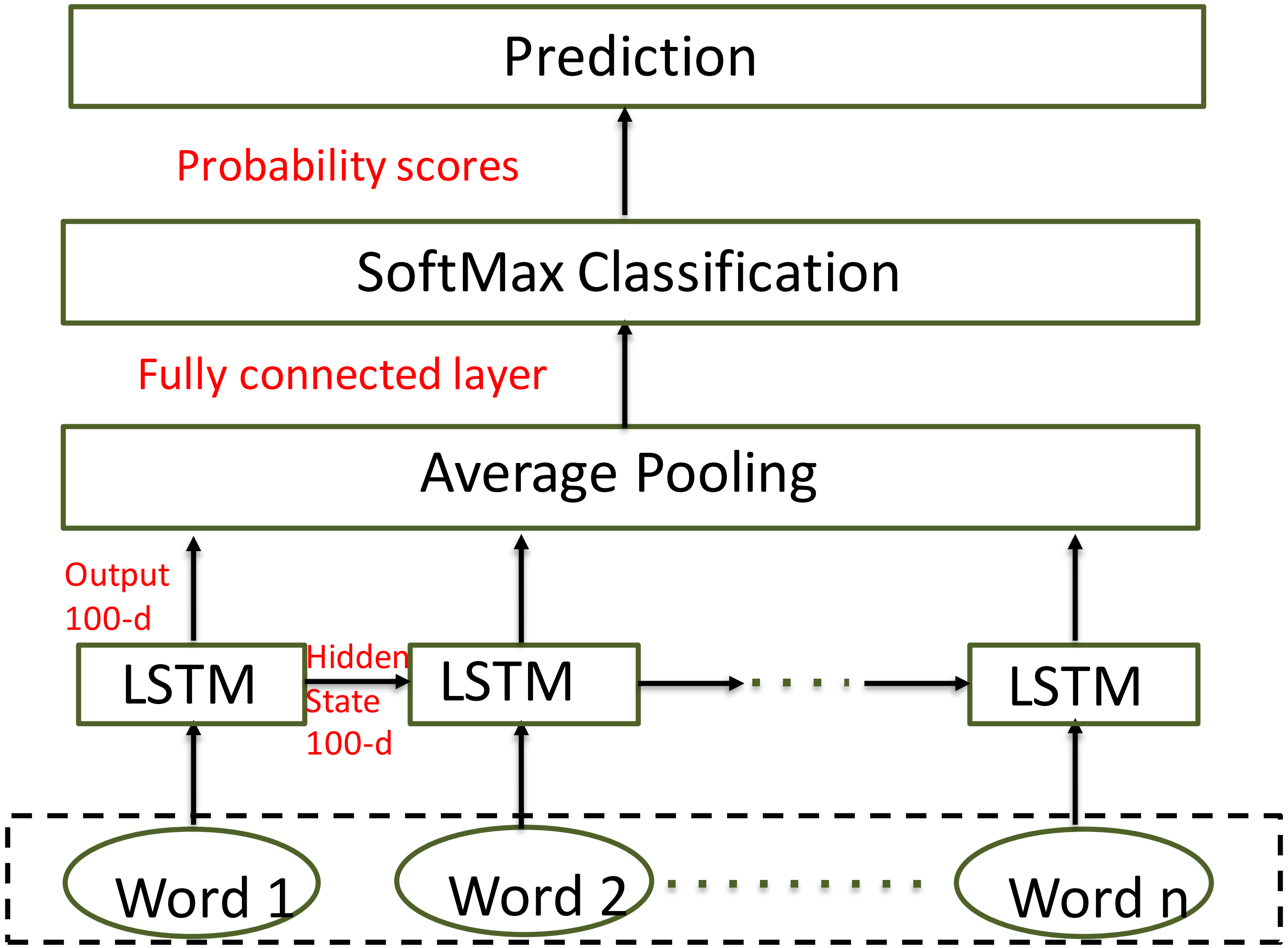
Sentence Level Author Identification

- Sentences from same genre are taken from the works of
  - Edgar Allan Poe
  - Howard Phillips Lovecraft
  - Mary Wollstonecraft Shelley
- Train Word2vec for the given text corpus
- Train a GRU -> FC for classifying the sentences
- Test accuracy



Code Available

Detailed view



Sentence

Each word represented as 100d vector

Challenges

Longer dependencies required

The clouds are in the _____

I grew up in France. In a small beautiful town… I speak fluent ______

Learning

- Vanishing Gradient Problem
- Training in Recurrent Neural Networks

Summary

Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

- ✓ Introduction to RNN
- ✓ RNN

Questions?