
Recurrent Neural Network Architectures

— Abhishek Narwekar, Anusri Pampari —

CS 598: Deep Learning and Recognition, Fall 2016

Lecture Outline

1. Introduction
2. Learning Long Term Dependencies
3. Regularization
4. Visualization for RNNs

Section 1: Introduction

Applications of RNNs

A person riding a motorcycle on a dirt road.



Image Captioning [[reference](#)]

... and more!

VIOLA:

Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;

Write like Shakespeare [[reference](#)]

↳ In reply to Thomas Paine



DeepDrumpf @DeepDrumpf · Mar 20

There will be no amnesty. It is going to pass because the people are going to be gone. I'm giving a mandate. #ComeyHearing @Thomas1774Paine

↳ 1

↻ 12

♥ 17

.. and Trump [[reference](#)]

Applications of RNNs

Technically, an RNN models sequences

Time series

Natural Language, Speech

We can even convert non-sequences to sequences, eg: feed an image as a sequence of pixels!

Applications of RNNs

RNN Generated TED Talks

[YouTube Link](#)

RNN Generated Eminem rapper

[RNN Shady](#)

RNN Generated Music

[Music Link](#)

Why RNNs?

Can model sequences having variable length

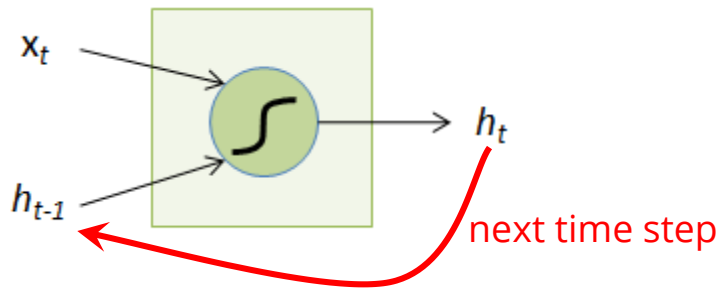
Efficient: Weights shared across time-steps

They work!

SOTA in several speech, NLP tasks

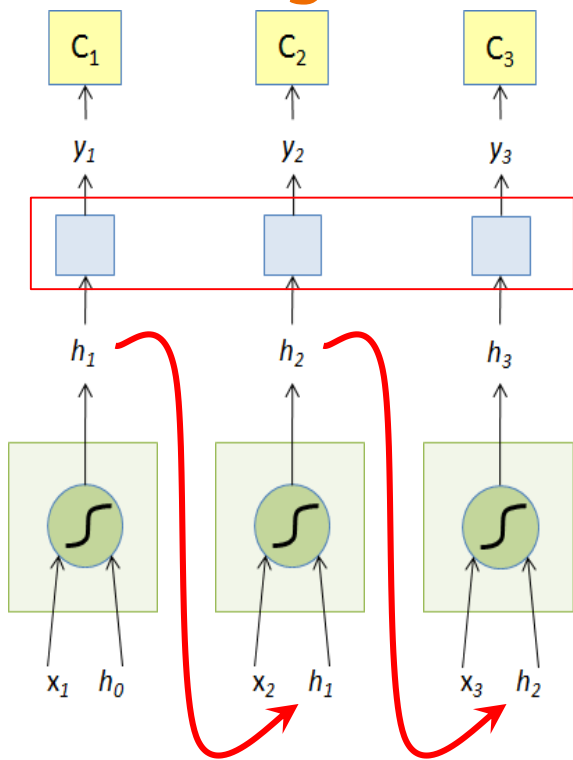
The Recurrent Neuron

- x_t : Input at time t
- h_{t-1} : State at time t-1



$$h_t = f(W_h h_{t-1} + W_x x_t)$$

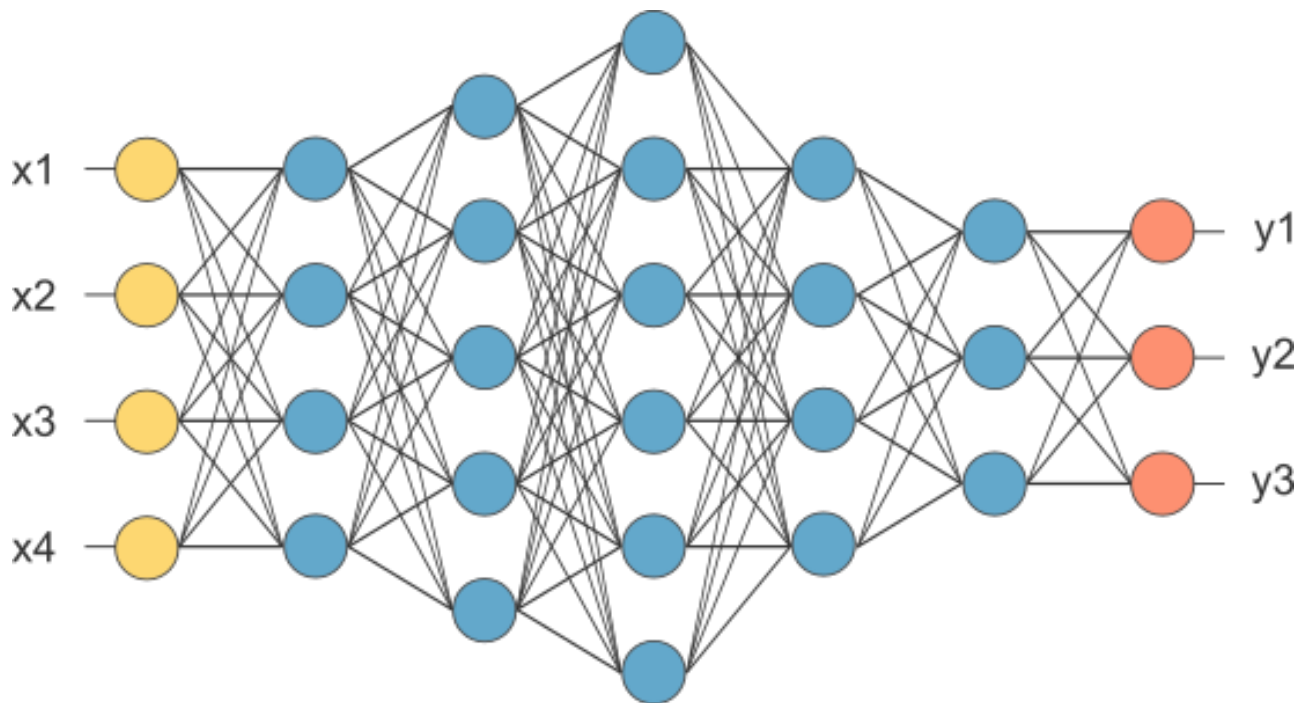
Unfolding an RNN



$$h_t = f(W_h h_{t-1} + W_x x_t)$$

Weights shared over time!

Making Feedforward Neural Networks Deep



Source: http://www.opennn.net/images/deep_neural_network.png

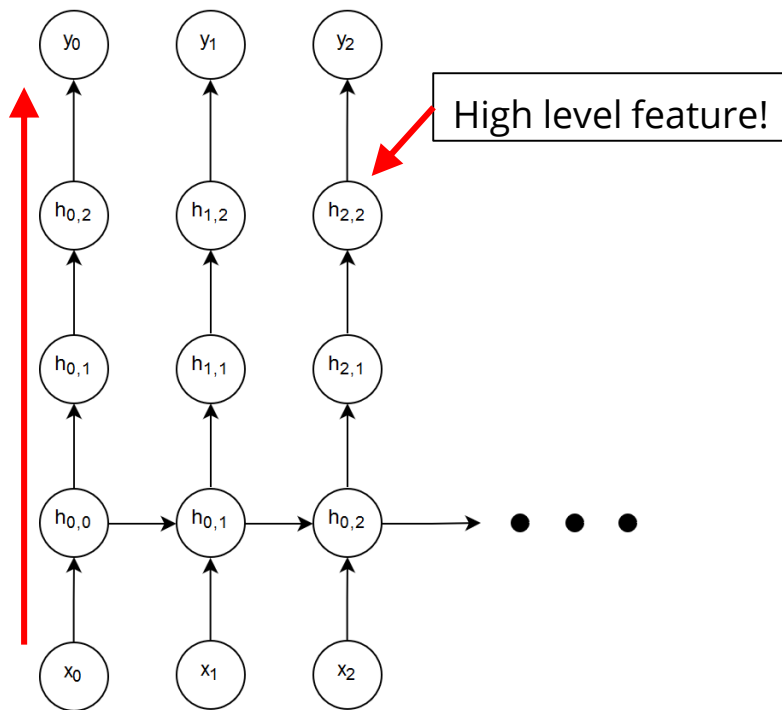
Option 1: Feedforward Depth (d_f)

Notation: $h_{0,1} \Rightarrow$ time step 0, neuron #1

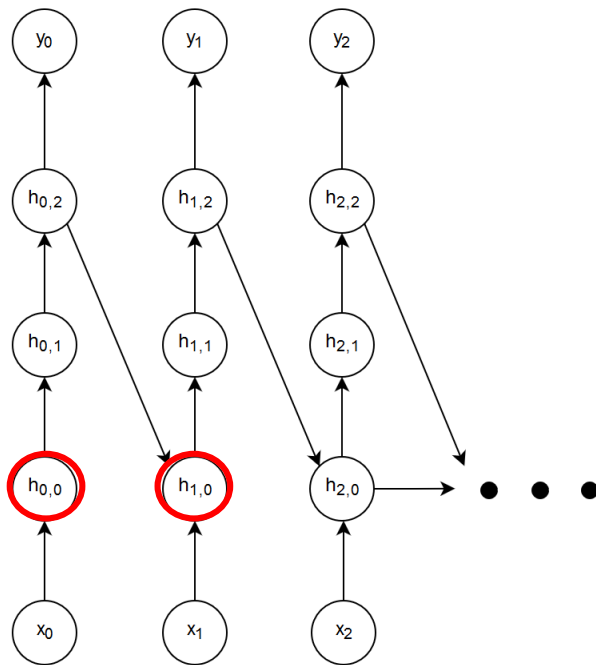
Feedforward depth: longest path

between an input and output at the
same timestep

Feedforward depth = 4



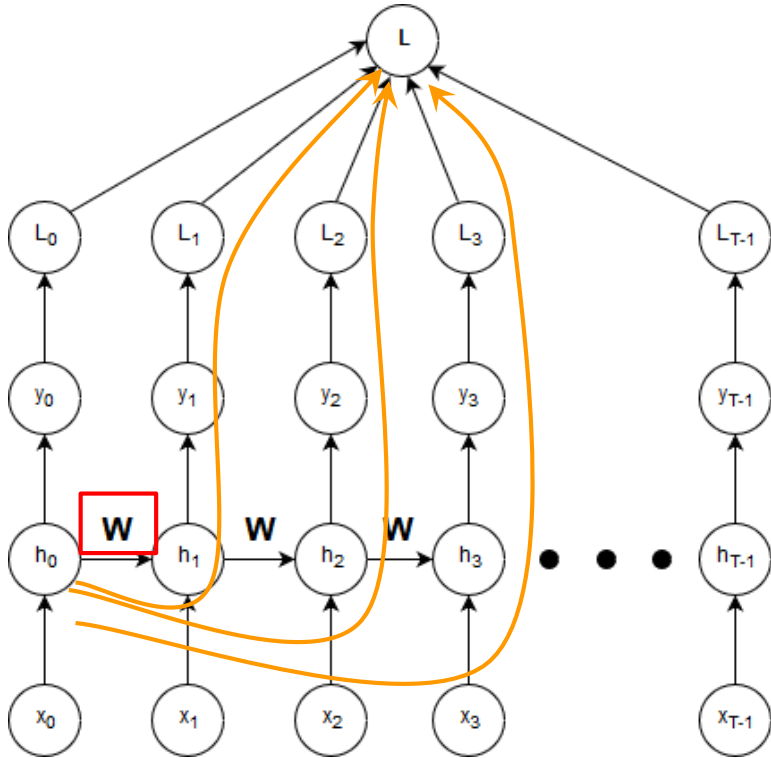
Option 2: Recurrent Depth (d_r)



- Recurrent depth: Longest path between **same hidden state** in **successive timesteps**

Recurrent depth = 3

Backpropagation Through Time (BPTT)



Objective is to update the weight matrix:

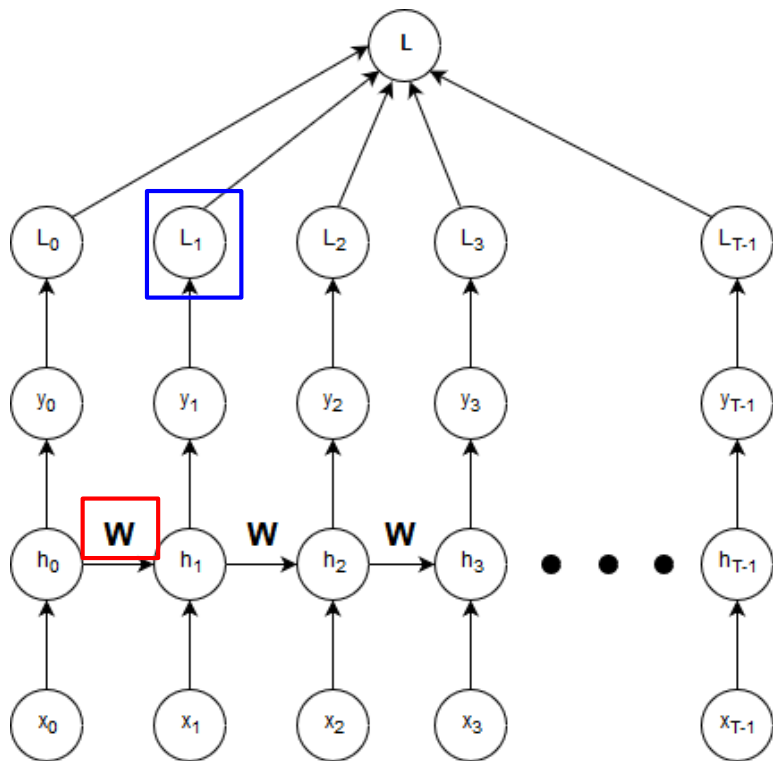
$$\mathbf{W} \rightarrow \mathbf{W} - \alpha \frac{\partial L}{\partial \mathbf{W}}$$

Issue: \mathbf{W} occurs each timestep

Every path from \mathbf{W} to L is one dependency

(note: dropping subscript h from \mathbf{W}_h for brevity)
Find all paths from \mathbf{W} to L !

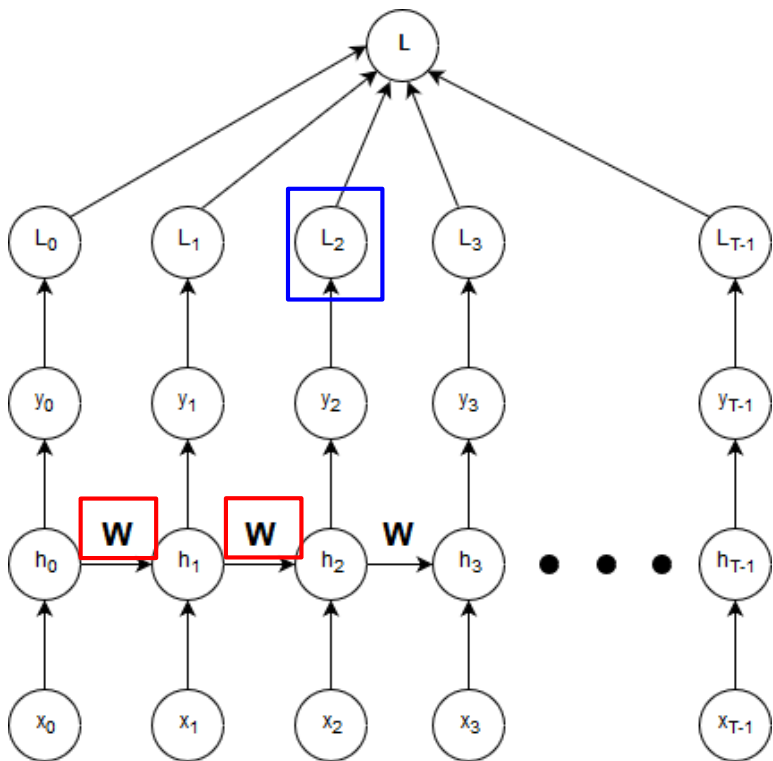
Systematically Finding All Paths



How many paths exist from W to L through L_1 ?

Just 1. Originating at h_0 .

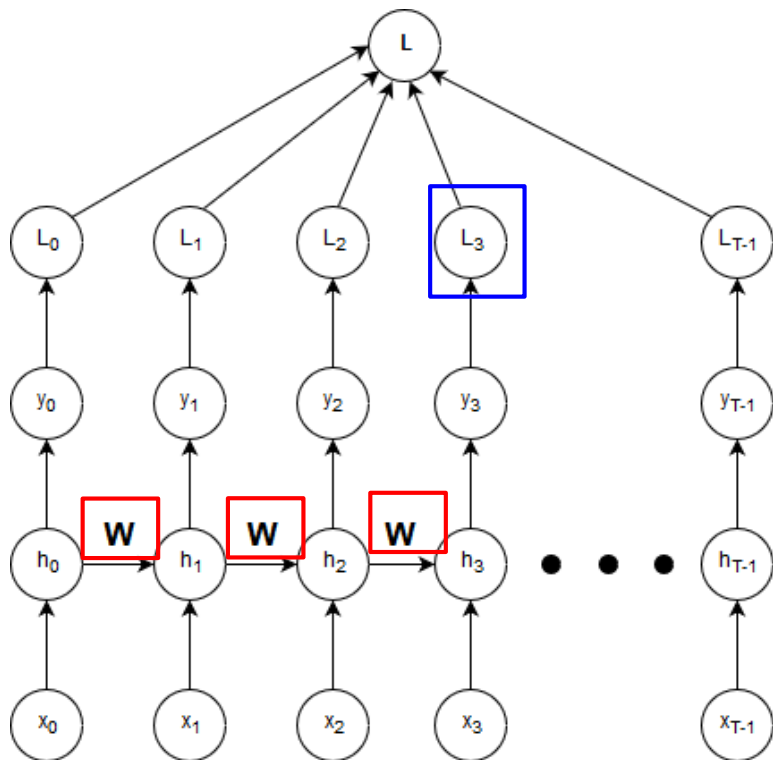
Systematically Finding All Paths



How many paths from W to L through L_2 ?

2. Originating at h_0 and h_1 .

Systematically Finding All Paths



And 3 in this case.

Origin of path = basis for Σ

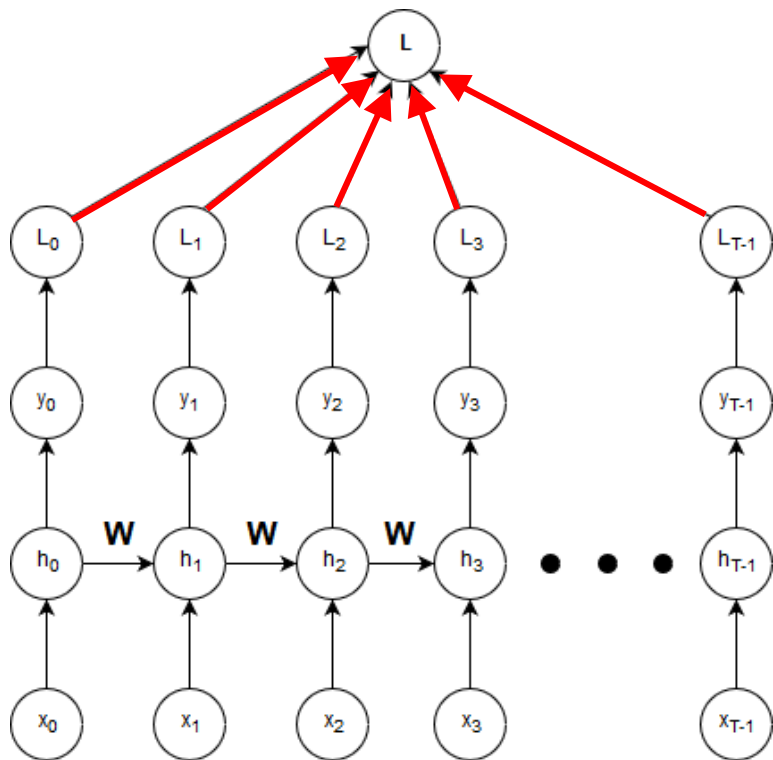
$$\frac{\partial L}{\partial \mathbf{W}}$$

The gradient has two summations:

- 1: Over L_j
- 2: Over h_k

To skip proof, click [here](#).

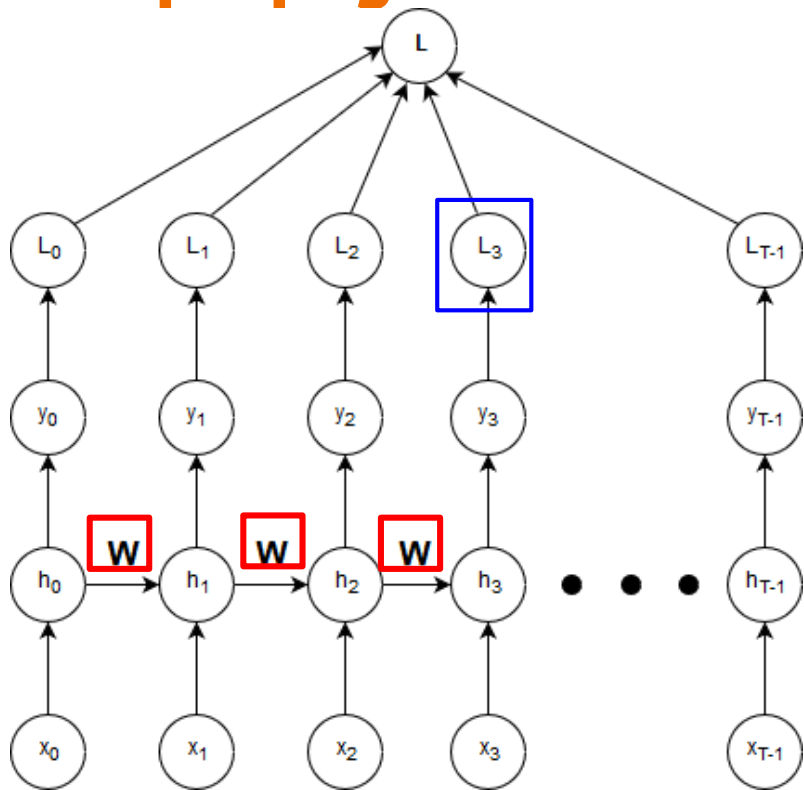
Backpropagation as two summations



First summation over L

$$\frac{\partial L}{\partial \mathbf{W}} = \sum_{j=0}^{T-1} \frac{\partial L_j}{\partial \mathbf{W}}$$

Backpropagation as two summations

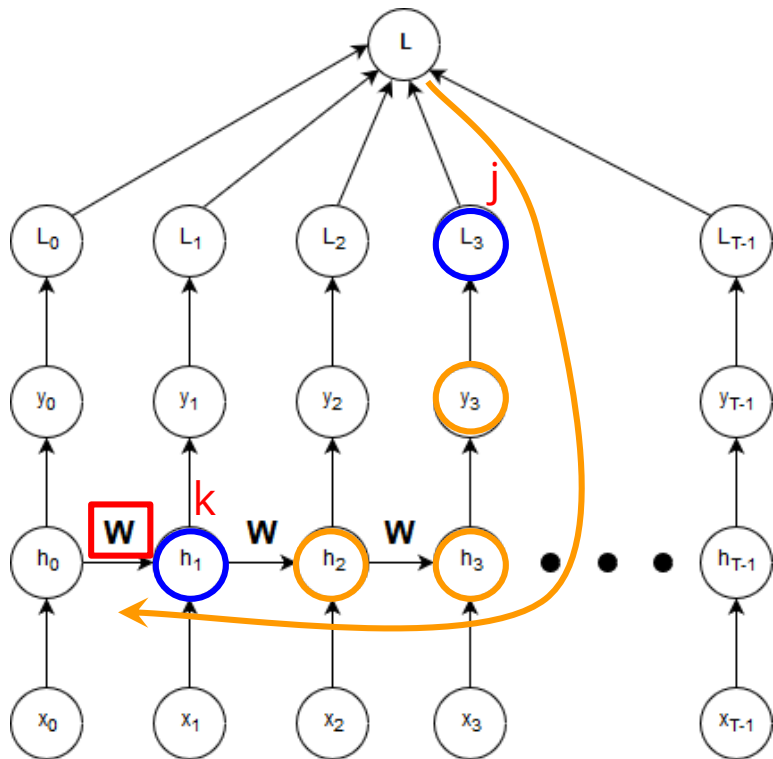


- **Second summation over h :**
Each L_j depends on the weight matrices *before it*

$$\frac{\partial L_j}{\partial \mathbf{W}} = \sum_{k=1}^j \frac{\partial L_j}{\partial h_k} \frac{\partial h_k}{\partial \mathbf{W}}$$

L_j depends on all h_k
before it.

Backpropagation as two summations

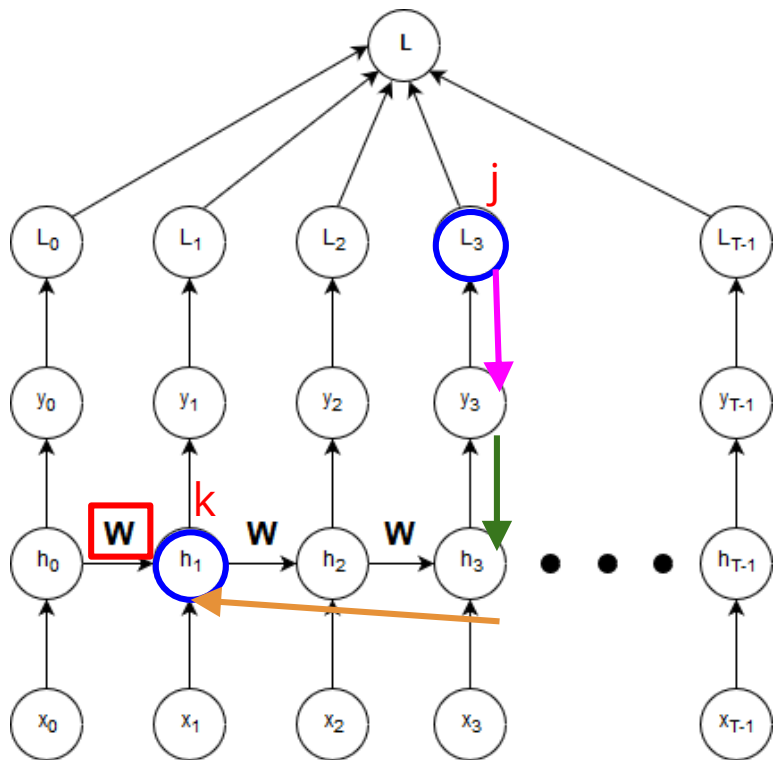


$$\frac{\partial L_j}{\partial W} = \sum_{k=1}^j \frac{\partial L_j}{\partial h_k} \frac{\partial h_k}{\partial W}$$

- No explicit of L_j on h_k
- Use chain rule to fill missing steps

$$\frac{\partial L_j}{\partial W} = \sum_{k=1}^j \frac{\partial L_j}{\partial y_j} \frac{\partial y_j}{\partial h_j} \frac{\partial h_j}{\partial h_k} \frac{\partial h_k}{\partial W}$$

Backpropagation as two summations

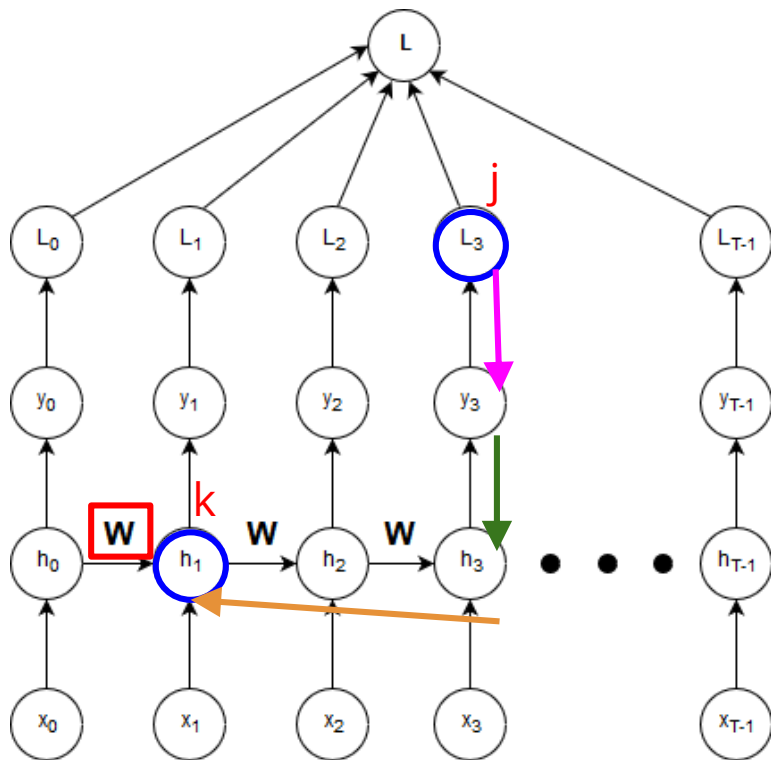


$$\frac{\partial L_j}{\partial W} = \sum_{k=1}^j \frac{\partial L_j}{\partial h_k} \frac{\partial h_k}{\partial W}$$

- No explicit of L_j on h_k
- Use chain rule to fill missing steps

$$\frac{\partial L_j}{\partial W} = \sum_{k=1}^j \frac{\partial L_j}{\partial y_j} \frac{\partial y_j}{\partial h_j} \frac{\partial h_j}{\partial h_k} \frac{\partial h_k}{\partial W}$$

The Jacobian



$$\frac{\partial L_j}{\partial \mathbf{W}} = \sum_{k=1}^j \frac{\partial L_j}{\partial y_j} \frac{\partial y_j}{\partial h_j} \frac{\partial h_j}{\partial h_k} \frac{\partial h_k}{\partial \mathbf{W}}$$

Indirect dependency. One final use of the chain rule gives:

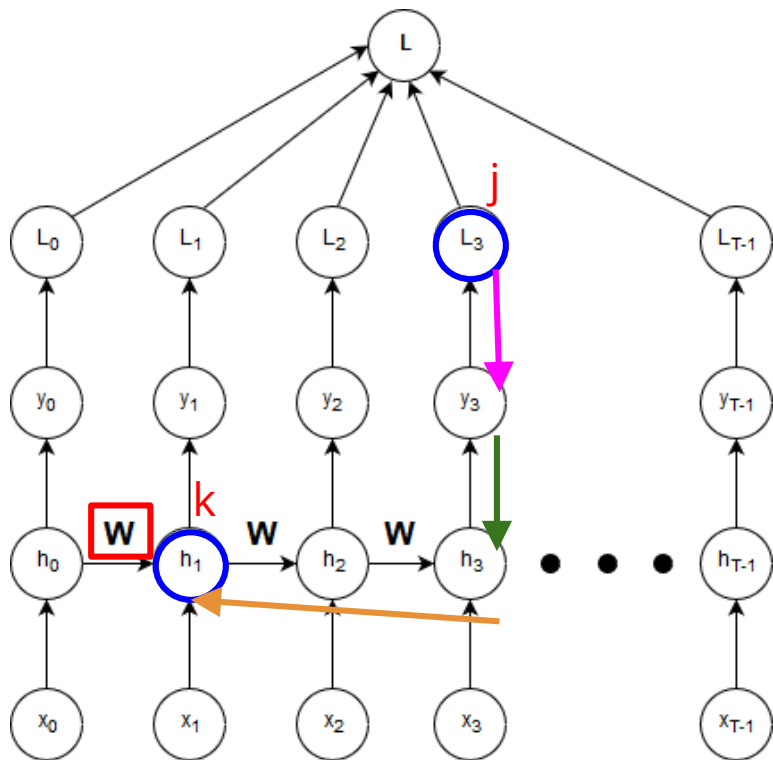
$$\frac{\partial h_j}{\partial h_k} = \prod_{m=k+1}^j \frac{\partial h_m}{\partial h_{m-1}}$$

“The Jacobian”

The Final Backpropagation Equation

$$\frac{\partial L}{\partial \mathbf{W}_h} = \sum_{j=0}^{T-1} \sum_{k=1}^j \frac{\partial L_j}{\partial y_j} \frac{\partial y_j}{\partial h_j} \left(\prod_{m=k+1}^j \frac{\partial h_m}{\partial h_{m-1}} \right) \frac{\partial h_k}{\partial \mathbf{W}_h}$$

Backpropagation as two summations



$$\frac{\partial L}{\partial \mathbf{W}_h} = \sum_{j=0}^{T-1} \sum_{k=1}^j \frac{\partial L_j}{\partial y_j} \frac{\partial y_j}{\partial h_j} \left(\prod_{m=k+1}^j \frac{\partial h_m}{\partial h_{m-1}} \right) \frac{\partial h_k}{\partial \mathbf{W}_h}$$

- Often, to reduce memory requirement, we truncate the network
- Inner summation runs from $j-p$ to j for some $p \implies$ truncated BPTT

Expanding the Jacobian

$$\frac{\partial L}{\partial \mathbf{W}} = \sum_{j=0}^{T-1} \sum_{k=1}^j \frac{\partial L_j}{\partial y_j} \frac{\partial y_j}{\partial h_j} \left(\prod_{m=k+1}^j \frac{\partial h_m}{\partial h_{m-1}} \right) \frac{\partial h_k}{\partial \mathbf{W}}$$

$$h_m = f(\mathbf{W}_h h_{m-1} + \mathbf{W}_x x_m)$$

$$\frac{\partial h_m}{\partial h_{m-1}} = \mathbf{W}_h^T \text{diag}(f'(\mathbf{W}_h h_{m-1} + \mathbf{W}_x x_m))$$

The Issue with the Jacobian

$$\frac{\partial h_j}{\partial h_k} = \prod_{m=k+1}^j \mathbf{W}_h^T \text{diag}(f'(\mathbf{W}_h h_{m-1} + \mathbf{W}_x x_m))$$

Weight Matrix

Derivative of activation function

Repeated matrix multiplications leads to **vanishing and exploding gradients**.

How? Let's take a slight detour.

Eigenvalues and Stability

Consider identity activation function

If Recurrent Matrix \mathbf{W}_h is diagonalizable:

$$\mathbf{W}_h = \mathbf{Q}^{-1} * \mathbf{\Lambda} * \mathbf{Q}$$

Computing powers of \mathbf{W}_h is simple:

$$\mathbf{W}_h^n = \mathbf{Q}^{-1} * \mathbf{\Lambda}^n * \mathbf{Q}$$

\mathbf{Q} matrix composed of eigenvectors of \mathbf{W}_h

$\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues placed on the diagonals

Eigenvalues and stability

$$\Lambda = \begin{bmatrix} -0.6180 & 0 \\ 0 & 1.6180 \end{bmatrix} \longrightarrow \Lambda^{10} = \begin{bmatrix} 0.0081 & 0 \\ 0 & 122.9919 \end{bmatrix}$$

Vanishing gradients

Exploding gradients

$$W_h^n = Q^{-1} * \Lambda^n * Q$$

All Eigenvalues < 1

Eigenvalues > 1

2. Learning Long Term Dependencies

Outline

Vanishing/Exploding Gradients in RNN

```
graph TD; A[Vanishing/Exploding Gradients in RNN] --> B[Weight Initialization Methods]; A --> C[Constant Error Carousel]; A --> D[Hessian Free Optimization]; A --> E[Echo State Networks]; B --> F["• Identity-RNN<br>• np-RNN"]; C --> G["• LSTM<br>• GRU"];
```

Weight Initialization Methods

- Identity-RNN
- np-RNN

Constant Error Carousel

- LSTM
- GRU

Hessian Free Optimization

Echo State Networks

Outline

Vanishing/Exploding Gradients in RNN

```
graph TD; A[Vanishing/Exploding Gradients in RNN] --> B[Weight Initialization Methods]; A --> C[Constant Error Carousel]; A --> D[Hessian Free Optimization]; A --> E[Echo State Networks]; B --> F["• Identity-RNN<br>• np-RNN"]; C --> G["• LSTM<br>• GRU"];
```

Weight Initialization Methods

- Identity-RNN
- np-RNN

Constant Error Carousel

- LSTM
- GRU

Hessian Free Optimization

Echo State Networks

Weight Initialization Methods

$$\frac{\partial h_j}{\partial h_k} = \prod_{m=k+1}^j \mathbf{W}_h^T \text{diag}(f'(\mathbf{W}_h h_{m-1} + \mathbf{W}_x x_m))$$

Activation function : ReLU

$$\frac{\partial h_j}{\partial h_k} = (\mathbf{W}_h^T)^n = Q^{-1} * \Lambda^n * Q$$

Weight Initialization Methods

Random W_h initialization of RNN has no constraint on eigenvalues

⇒ vanishing or exploding gradients in the initial epoch

Weight Initialization Methods

Careful initialization of W_h with suitable eigenvalues

⇒ allows the RNN to learn in the initial epochs

⇒ hence can generalize well for further iterations

Weight Initialization Trick #1: IRNN

- W_h initialized to Identity
- Activation function: ReLU

Weight Initialization Trick #2: np-RNN

- W_h positive definite (+ve real eigenvalues)
- At least one eigenvalue is 1, others all less than equal to one
- Activation function: ReLU

Geoffrey et al, "Improving Performance of Recurrent Neural Network with ReLU nonlinearity"

np-RNN vs IRNN

Sequence Classification Task

RNN Type	Accuracy Test	Parameter Complexity Compared to RNN	Sensitivity to parameters
IRNN	67 %	x1	high
np-RNN	75.2 %	x1	low
LSTM	78.5 %	x4	low

Summary

- np-RNNs work as well as LSTMs utilizing 4 times less parameters than a LSTM

Outline

Vanishing/Exploding Gradients in RNN

```
graph TD; A[Vanishing/Exploding Gradients in RNN] --> B[Weight Initialization Methods]; A --> C[Constant Error Carousel]; A --> D[Hessian Free Optimization]; A --> E[Echo State Networks]; B --> F["• Identity-RNN<br>• np-RNN"]; C --> G["• LSTM<br>• GRU"];
```

Weight
Initialization
Methods

- Identity-RNN
- np-RNN

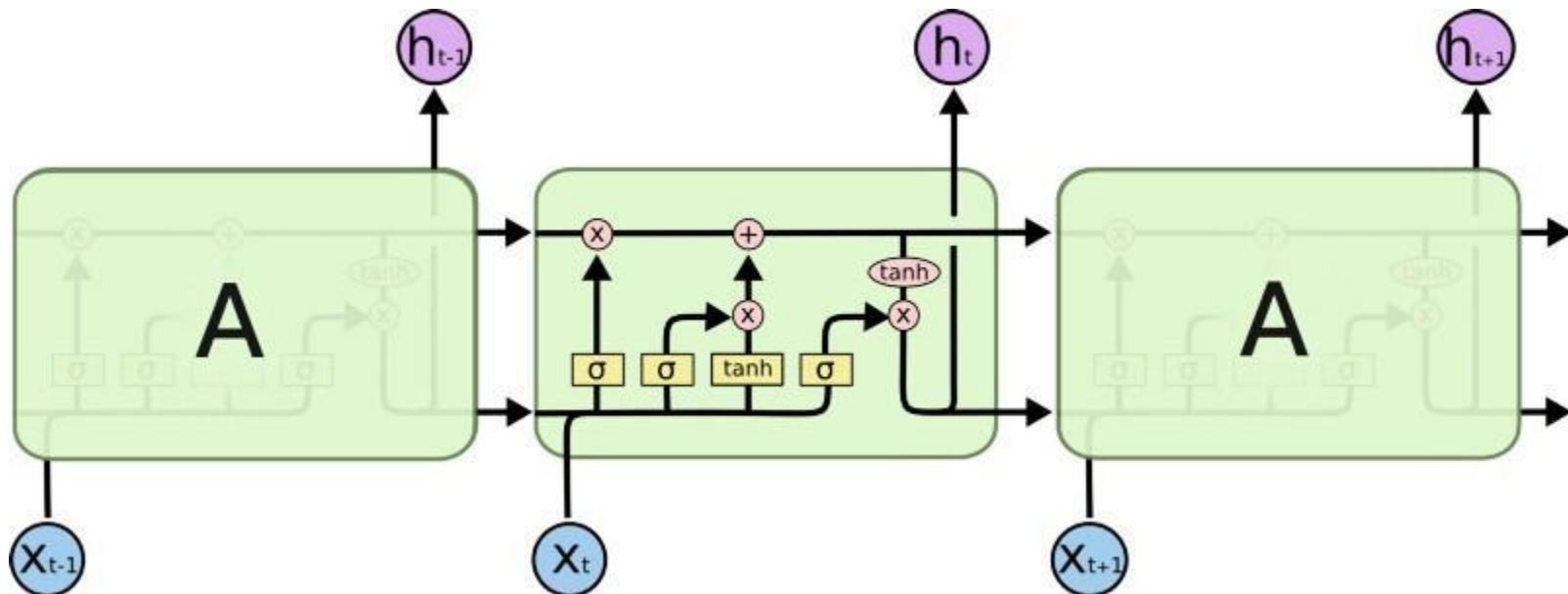
Constant Error
Carousel

- LSTM
- GRU

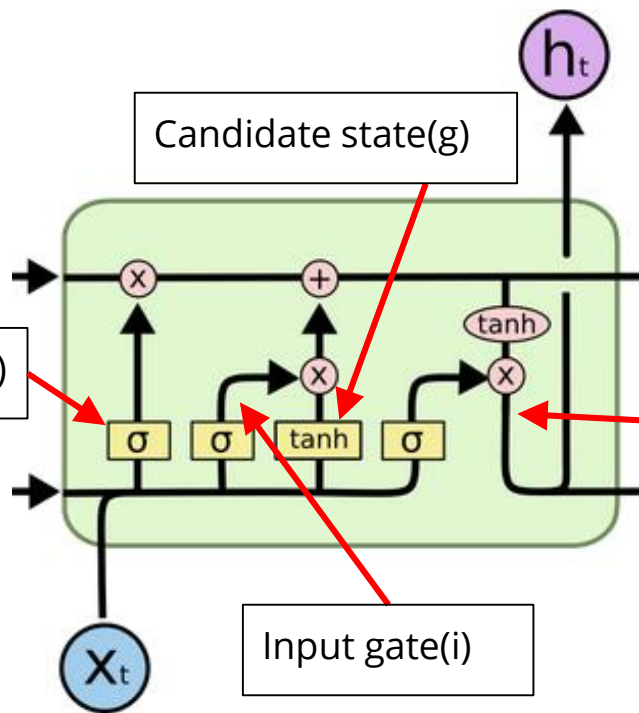
Hessian Free
Optimization

Echo State
Networks

The LSTM Network



The LSTM Cell



- $\sigma()$: sigmoid non-linearity
- \otimes : element-wise multiplication

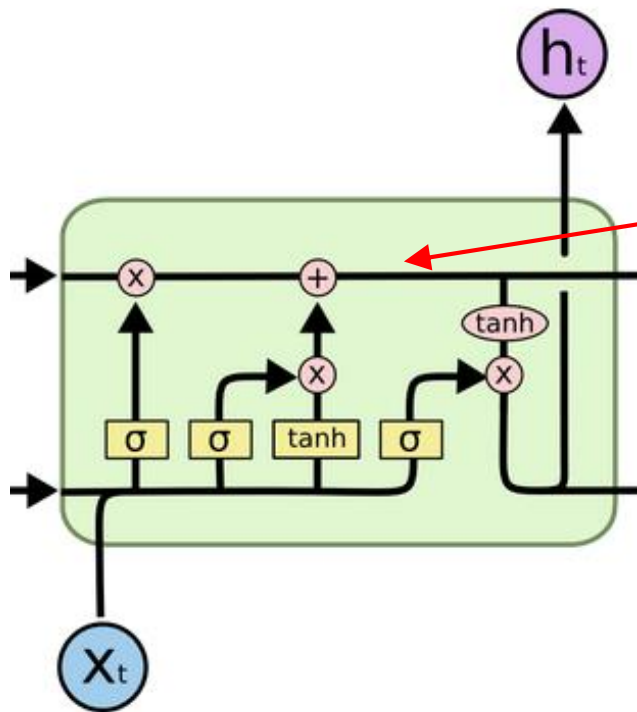
Forget gate(f)

Candidate state(g)

Output gate(g)

Input gate(i)

The LSTM Cell



$$\begin{pmatrix} \tilde{\mathbf{f}}_t \\ \tilde{\mathbf{i}}_t \\ \tilde{\mathbf{o}}_t \\ \tilde{\mathbf{g}}_t \end{pmatrix} = \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}$$

$$\mathbf{c}_t = \sigma(\tilde{\mathbf{f}}_t) \odot \mathbf{c}_{t-1} + \sigma(\tilde{\mathbf{i}}_t) \odot \tanh(\tilde{\mathbf{g}}_t)$$

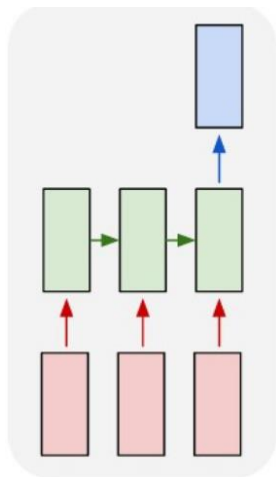
$$\mathbf{h}_t = \sigma(\tilde{\mathbf{o}}_t) \odot \tanh(\mathbf{c}_t),$$

Forget old state

Remember new state

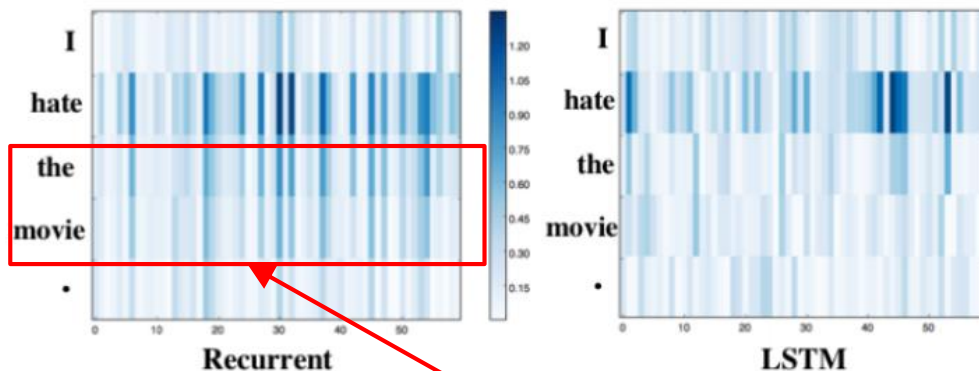
Long Term Dependencies with LSTM

Sentiment Analysis



Many-one network

Saliency Heatmap



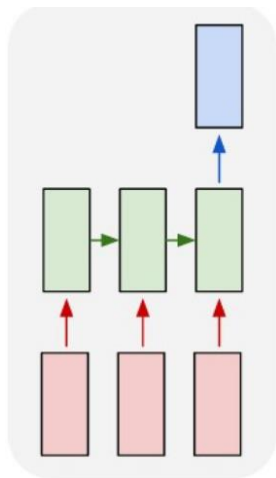
Recent words more salient

LSTM captures long term dependencies

"Jiwei Li et al, "Visualizing and Understanding Neural Models in NLP"

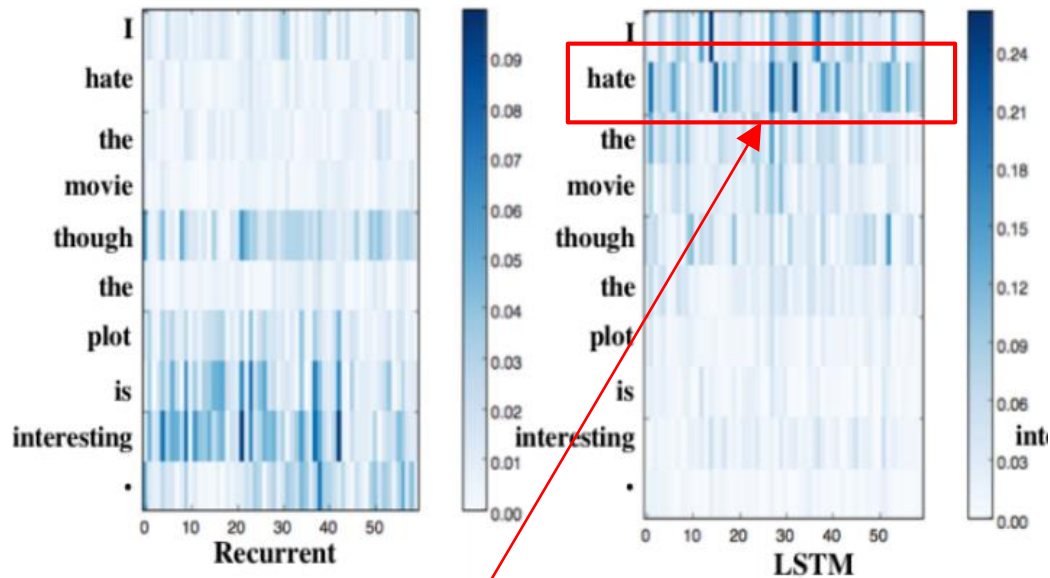
Long Term Dependencies with LSTM

Sentiment Analysis



Many-one network

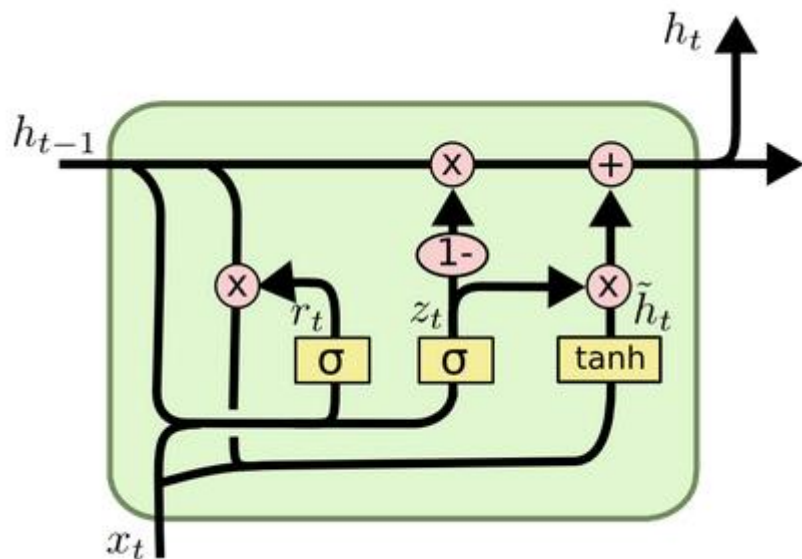
Saliency Heatmap



LSTM captures long term dependencies

“Jiwei Li et al, “Visualizing and Understanding Neural Models in NLP”

Gated Recurrent Unit



- Replace forget (f) and input (i) gates with an update gate (z)
- Introduce a reset gate (r) that modifies h_{t-1}
- Eliminate internal memory c_t

Comparing GRU and LSTM

- Both **GRU** and **LSTM** better than **RNN with tanh** on music and speech modeling
- GRU performs comparably to LSTM
- No clear consensus between GRU and LSTM

Source: Empirical evaluation of GRUs on sequence modeling, 2014

3. Regularization in RNNs

Outline

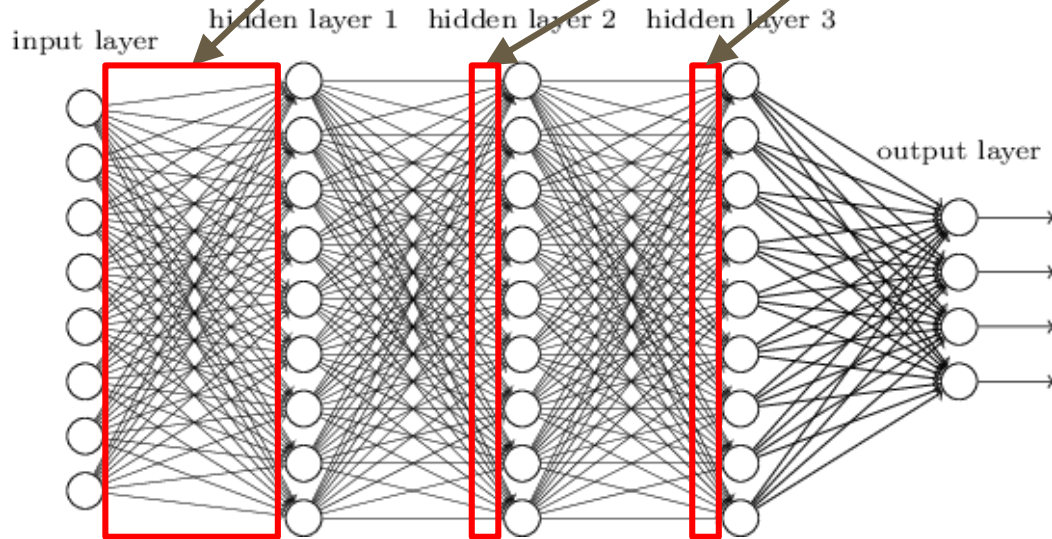
Batch Normalization

Dropout

Recurrent Batch Normalization

Internal Covariate Shift

If these weights are updated... the distributions change in layers above!



The model needs to learn parameters **while adapting to the changing input distribution**
⇒ slower model convergence!

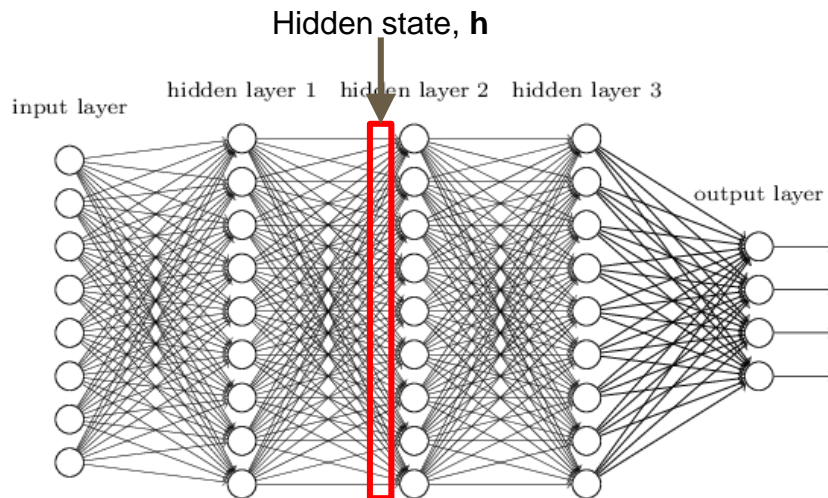
Source: <https://i.stack.imgur.com/1bCQl.png>

Solution: Batch Normalization

Batch Normalization Equation:

$$\text{BN}(\mathbf{h}; \gamma, \beta) = \beta + \gamma \odot \frac{\mathbf{h} - \hat{\mathbb{E}}[\mathbf{h}]}{\sqrt{\widehat{\text{Var}}[\mathbf{h}] + \epsilon}}$$

Bias, Std Dev: To be learned



Extension of BN to RNNs: Trivial?

- RNNs **deepest along temporal dimension**
- Must be careful: repeated scaling could cause **exploding** gradients

The method that's effective

$$\begin{pmatrix} \tilde{\mathbf{f}}_t \\ \tilde{\mathbf{i}}_t \\ \tilde{\mathbf{o}}_t \\ \tilde{\mathbf{g}}_t \end{pmatrix} = \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}$$

$$\mathbf{c}_t = \sigma(\tilde{\mathbf{f}}_t) \odot \mathbf{c}_{t-1} + \sigma(\tilde{\mathbf{i}}_t) \odot \tanh(\tilde{\mathbf{g}}_t)$$

$$\mathbf{h}_t = \sigma(\tilde{\mathbf{o}}_t) \odot \tanh(\mathbf{c}_t),$$

Original LSTM Equations

$$\begin{pmatrix} \tilde{\mathbf{f}}_t \\ \tilde{\mathbf{i}}_t \\ \tilde{\mathbf{o}}_t \\ \tilde{\mathbf{g}}_t \end{pmatrix} =$$

$$\text{BN}(\mathbf{W}_h \mathbf{h}_{t-1}; \gamma_h, \beta_h) + \text{BN}(\mathbf{W}_x \mathbf{x}_t; \gamma_x, \beta_x) + \mathbf{b}$$

$$\mathbf{c}_t = \sigma(\tilde{\mathbf{f}}_t) \odot \mathbf{c}_{t-1} + \sigma(\tilde{\mathbf{i}}_t) \odot \tanh(\tilde{\mathbf{g}}_t)$$

$$\mathbf{h}_t = \sigma(\tilde{\mathbf{o}}_t) \odot \tanh(\text{BN}(\mathbf{c}_t; \gamma_c, \beta_c))$$

Batch Normalized LSTM

Observations

- x, h_{t-1} normalized **separately**

- c_t **not normalized**

(doing so may disrupt gradient flow) **How?**

- New state (h_t) normalized

$$\begin{pmatrix} \tilde{\mathbf{f}}_t \\ \tilde{\mathbf{i}}_t \\ \tilde{\mathbf{o}}_t \\ \tilde{\mathbf{g}}_t \end{pmatrix} = \text{BN}(\mathbf{W}_h \mathbf{h}_{t-1}; \gamma_h, \beta_h) + \text{BN}(\mathbf{W}_x \mathbf{x}_t; \gamma_x, \beta_x) + \mathbf{b}$$
$$\mathbf{c}_t = \sigma(\tilde{\mathbf{f}}_t) \odot \mathbf{c}_{t-1} + \sigma(\tilde{\mathbf{i}}_t) \odot \tanh(\tilde{\mathbf{g}}_t)$$
$$\mathbf{h}_t = \sigma(\tilde{\mathbf{o}}_t) \odot \tanh(\text{BN}(\mathbf{c}_t; \gamma_c, \beta_c))$$

Additional Guidelines

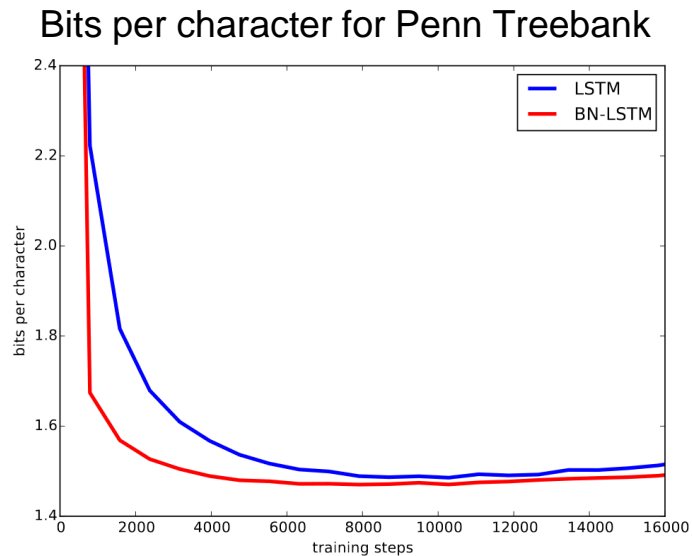
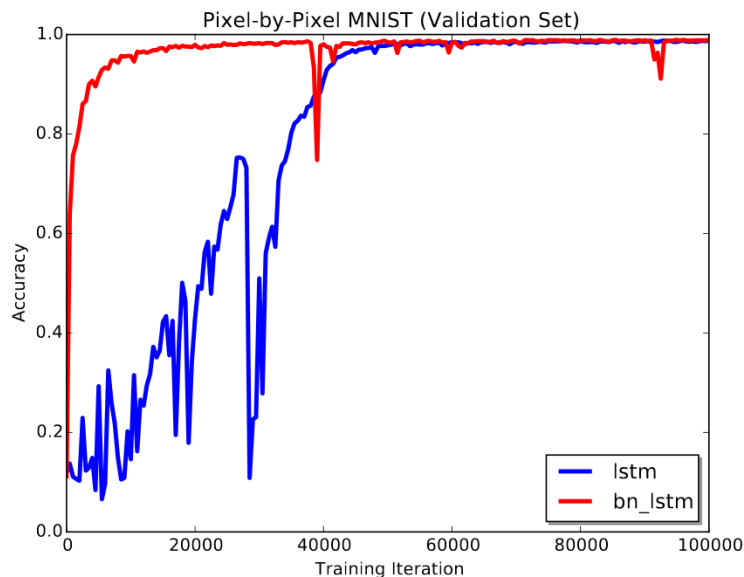
$$\text{BN}(\mathbf{W}_h \mathbf{h}_{t-1}; \gamma_h, \beta_h)$$

- Initialize β to $\mathbf{0}$, γ to a small value such as ~ 0.1 . Else vanishing gradients (think of the tanh plot!)
- Learn statistics for each time step **independently** till some time step \mathbf{T} . Beyond \mathbf{T} , use statistics for \mathbf{T}

Results

A: **Faster convergence** due to Batch Norm

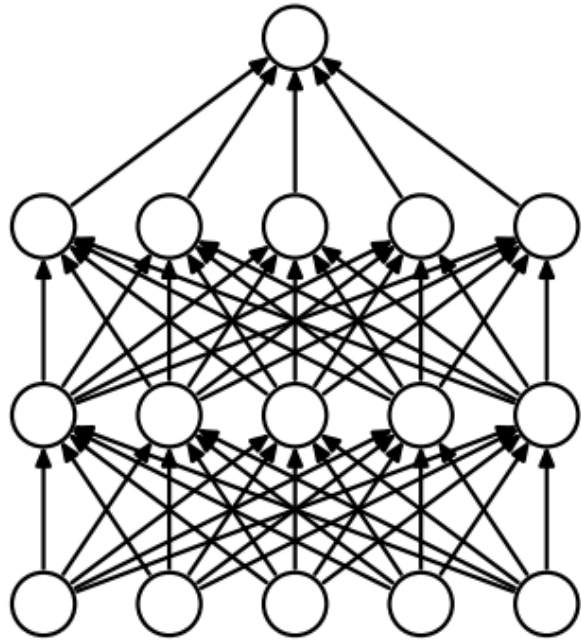
B: Performance **as good** as (if not better than) unnormalized LSTM



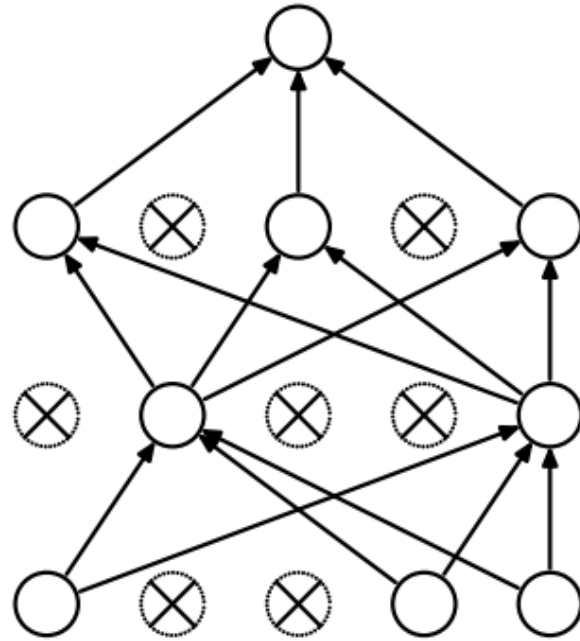
Cooijmans, Tim, et al. "Recurrent batch normalization."(2016).

Dropout In RNN

Recap: Dropout In Neural Networks

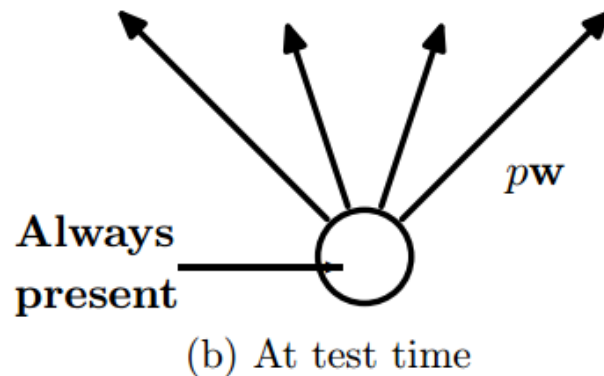
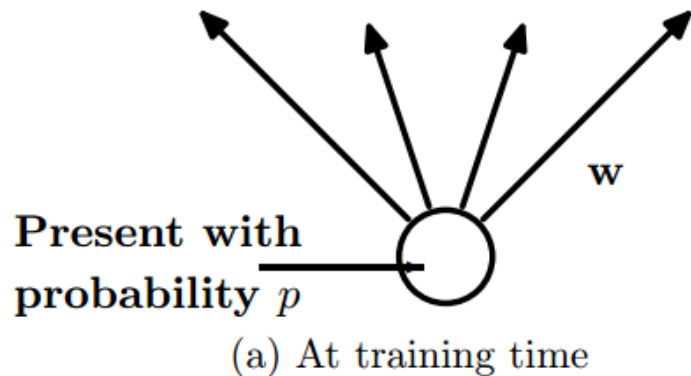


(a) Standard Neural Net



(b) After applying dropout.

Recap: Dropout In Neural Networks



Dropout

To prevent over confident models

High Level Intuition: Ensemble of thinned networks sampled through dropout

Interested in a theoretical proof ?

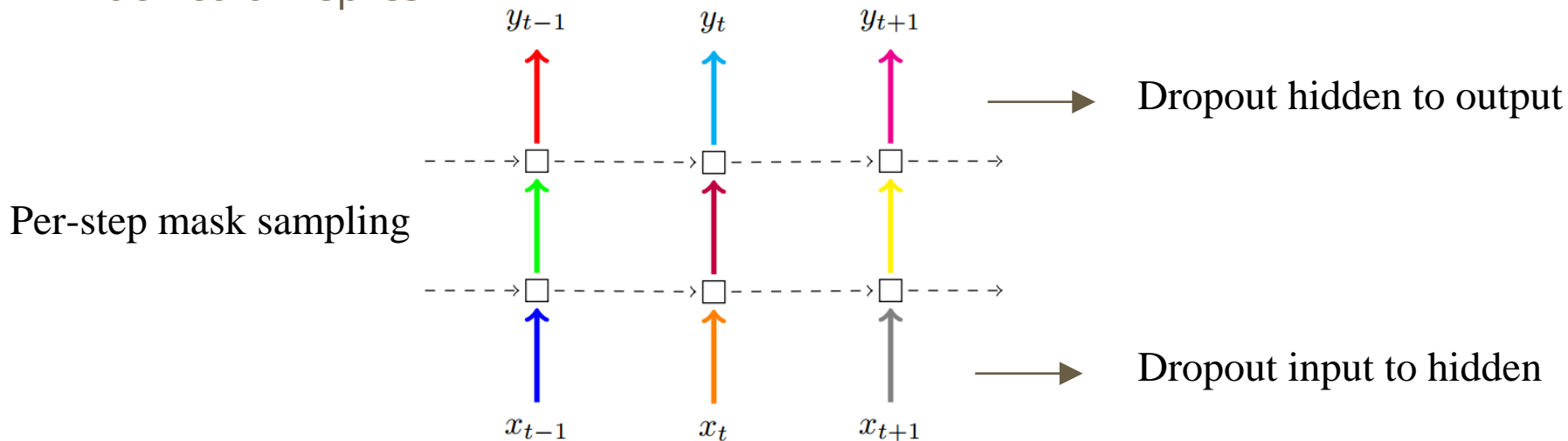
A Probabilistic Theory of Deep Learning, [Ankit B. Patel](#), Tan Nguyen, [Richard G. Baraniuk](#)

[Skip Proof Slides](#)

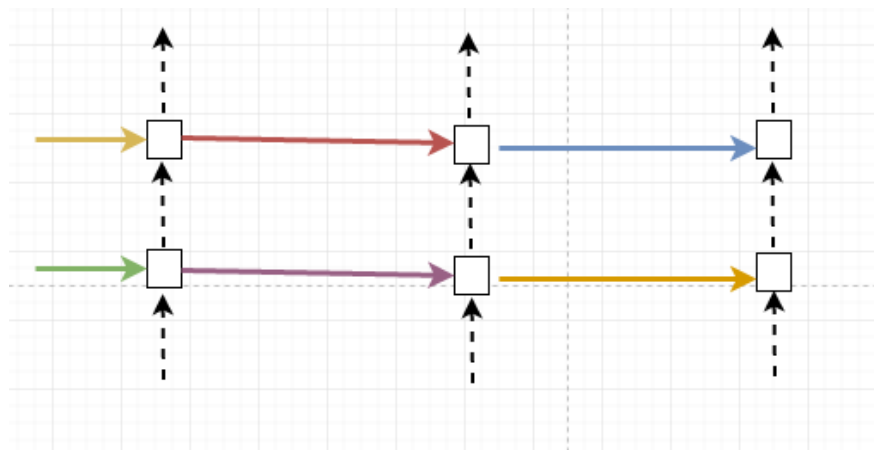
RNN Feedforward Dropout

Beneficial to use it once in correct spot rather than put it everywhere

Each color repres



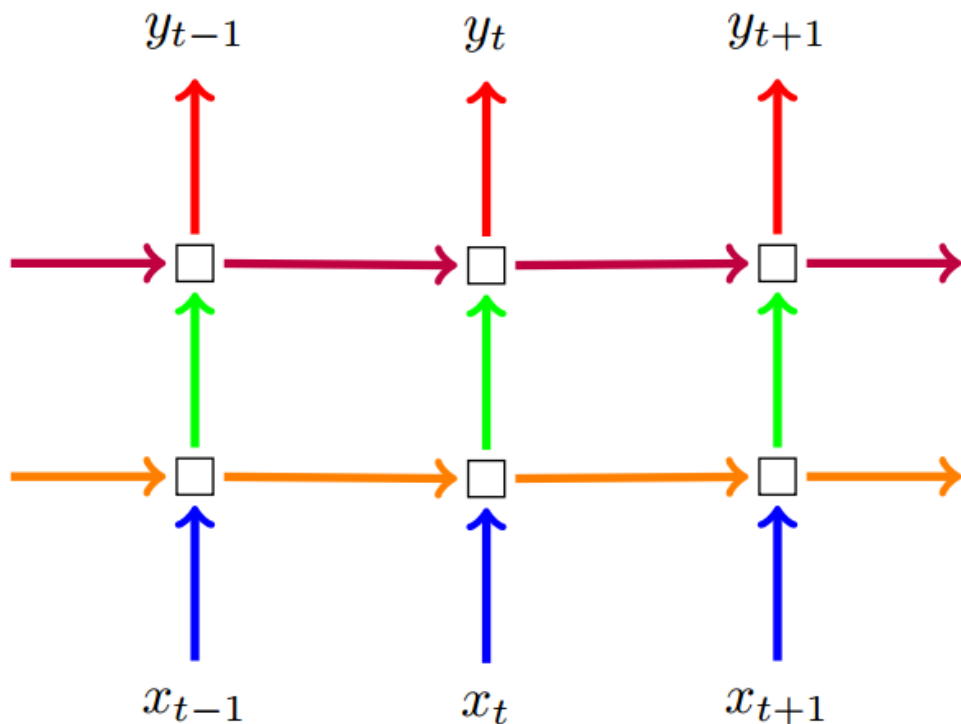
RNN Recurrent Dropout



MEMORY LOSS !

Only tends to retain short term dependencies

RNN Recurrent+Feedforward Dropout



Per-sequence mask
sampling

Drop the time dependency
of an entire feature

Dropout in LSTMs

Dropout on cell state (c_t)

Inefficient

Dropout on cell state update
($\tanh(\tilde{g}_t)$ or (h_{t-1})

Optimal

$$\begin{pmatrix} \tilde{\mathbf{f}}_t \\ \tilde{\mathbf{i}}_t \\ \tilde{\mathbf{o}}_t \\ \tilde{\mathbf{g}}_t \end{pmatrix} = \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}$$
$$\mathbf{c}_t = \sigma(\tilde{\mathbf{f}}_t) \odot \mathbf{c}_{t-1} + \sigma(\tilde{\mathbf{i}}_t) \odot \tanh(\tilde{\mathbf{g}}_t)$$
$$\mathbf{h}_t = \sigma(\tilde{\mathbf{o}}_t) \odot \tanh(\mathbf{c}_t),$$

[Skip to Visualization](#)

Some Results: Language Modelling Task

Model	Perplexity Scores
Original	125.2
Forward Dropout + Drop ($\tanh(g_t)$)	87 (-37)
Forward Dropout + Drop (h_{t-1})	88.4 (-36)
Forward Dropout	89.5 (-35)
Forward Dropout + Drop (c_t)	99.9 (-25)

Lower perplexity score is better !


Section 4: Visualizing and Understanding Recurrent Networks

Visualization outline

Observe evolution of features during training

Visualize output predictions

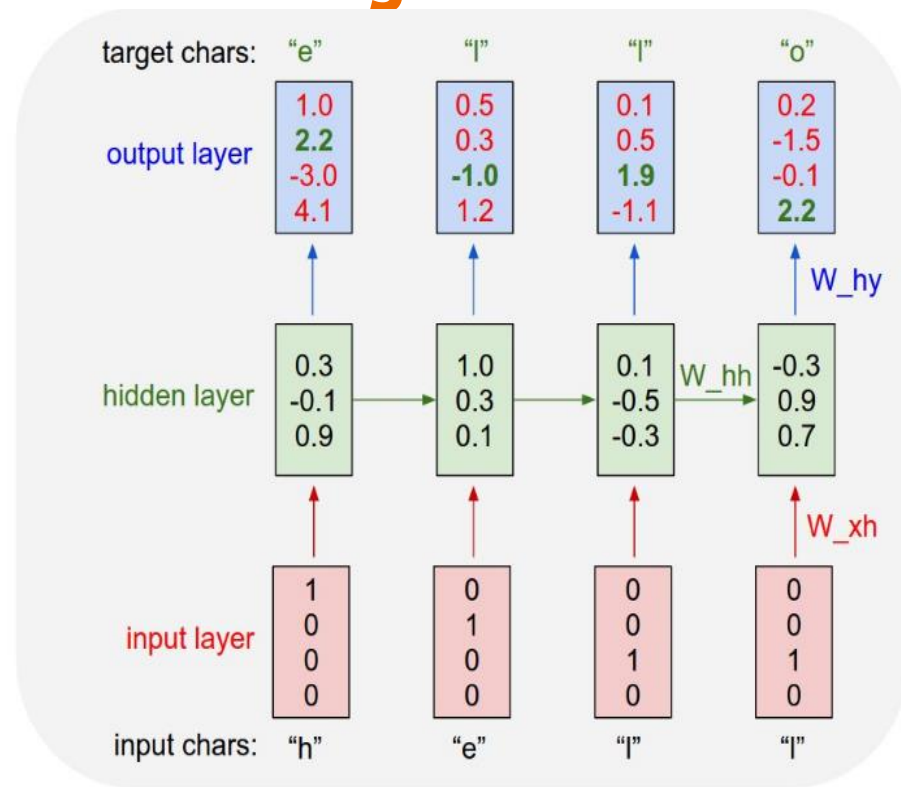
Visualize neuron activations



Character Level Language
Modelling task

Character Level Language Modelling

Task: Predicting the next character given the current character



Andrej Karpathy, Blog on "Unreasonable Effectiveness of Recurrent Neural Networks"

Generated Text:

Naturalism and decision for the majority of Arab countries' capitalide was grounded by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]] associated with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal in the [[Protestant Immineners]], which could be said to be directly in Cantonese Communication, which followed a ceremony and set inspired prison, training. The emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known in western [[Scotland]], near Italy to the conquest of India with the conflict. Copyright was the succession of independence in the slop of Syrian influence that was a famous German movement based on a more popular servicious, non-doctrinal and sexual power post. Many governments recognize the military housing of the [[Civil Liberalization and Infantry Resolution 265 National Party in Hungary]], that is sympathetic to be to the [[Punjab Resolution]]
(P.S)[<http://www.humah.yahoo.com/guardian.cfm/7754800786d17551963589.htm> Official economics Adjoint for the Nazism, Montgomery was swear to advance to the resources for those Socialism's rule, was starting to signing a major tripad of aid exile.]]

- Remembers to close a bracket
- Capitalize nouns
- 404 Page Not Found! :P The LSTM hallucinates it.

**100 th
iteration**

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

↓
train more

**300 th
iteration**

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

↓
train more

**700 th
iteration**

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 offer.

↓
train more

**2000 th
iteration**

"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.

Visualizing Predictions and Neuron “firings”

t	t	p	:	/	/	w	w	.	y	n	e	t	n	e	w	s	.	c	o	m	/]	E	n	g	l	i	s	h	-	l	a	n	g	u	a	g	e	w	e	b	s	i	t	e		
t	p	:	/	/	w	w	.	b	a	c	a	h	e	t	s	.	c	o	m	/			-	x	g	l	i	s	h	l	i	n	g	u	a	g	e	s	a	i	r	s	i	t	e		
	d	:	x	n	e	.	w	a	e	a	.	.	a	w	a	t	o	a	.		s	&	n	t	i	a	c	a	-	s	a	r	d	e	e	l	h	o	a	n	t	b	i	s			
m	w	-	2		p	i	i	s	o	e	s	s	i	s	.	/	e	r	n	.	c]	(d	c	e	e	n	e	p	e	s	a	a	i	k	i	i	e	e	l	e	d	h	,		
d	r	.	<	:	a	h	b	-	n	p	t	w	t	.	x	i	g	h	/	m	a)	T	v	d	r	y	z	i	c	o	u	e	d	l	s	u	:	t	h	a	-	o	o			
s	t	p	,	t	c	o	a	2	d	r	u	l	w	o	c	l	e	n	s	r]	p	.	l	l	v	a	o	d	,	,	e	y	t	c	-	n	d	m	-	o	i	b	u	v	s]



Likely prediction



Not a likely prediction



Excited neuron in url



Not excited neuron outside url

Features RNN Captures in Common Language ?

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



Cell Sensitive to Position in Line

- Can be interpreted as tracking the line length

The sole importance of the crossing of the Berezina lies in the fact that it plainly and indubitably proved the fallacy of all the plans for cutting off the enemy's retreat and the soundness of the only possible line of action--the one Kutuzov and the general mass of the army demanded--namely, simply to follow the enemy up. The French crowd fled at a continually increasing speed and all its energy was directed to reaching its goal. It fled like a wounded animal and it was impossible to block its path. This was shown not so much by the arrangements it made for crossing as by what took place at the bridges. When the bridges broke down, unarmed soldiers, people from Moscow and women with children who were with the French transport, all--carried on by vis inertiae--pressed forward into boats and into the ice-covered water and did not, surrender.

Cell That Turns On Inside Quotes

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

Features RNN Captures in C Language?

```
for (i = 0; i < 16; i++) {  
    if (k & (1 << 1))  
        pipe = (in_use & UMXTHREAD_UNCCA) +  
                ((count & 0x00000000ffffffff8) & 0x000000f) << 8;  
    if (count == 0)  
        sub(pid, ppc_md.kexec_handle, 0x20000000);  
    pipe_set_bytes(i, 0);  
}  
  
/* Free our user pages pointer to place camera if all dash */
```

Cell That Activates Inside IF Statements

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
                           siginfo_t *info)
{
    int sig = next_signal(pending, mask);
    if (sig) {
        if (current->notifier) {
            if (sigismember(current->notifier_mask, sig)) {
                if (!(current->notifier)(current->notifier_data)) {
                    clear_thread_flag(TIF_SIGPENDING);
                    return 0;
                }
            }
        }
        collect_signal(sig, pending, info);
    }
    return sig;
}
```

Cell That Is Sensitive To Indentation

- Can be interpreted as tracking indentation of code.
- Increasing strength as indentation increases

```
#ifdef CONFIG_AUDITSYSCALL
static inline int audit_match_class_bits(int class, u32 *mask)
{
    int i;
    if (classes[class]) {
        for (i = 0; i < AUDIT_BITMASK_SIZE; i++)
            if (mask[i] & classes[class][i])
                return 0;
    }
    return 1;
}
```

Non-Interpretable Cells

- Only 5% of the cells show such interesting properties
- Large portion of the cells are not interpretable by themselves

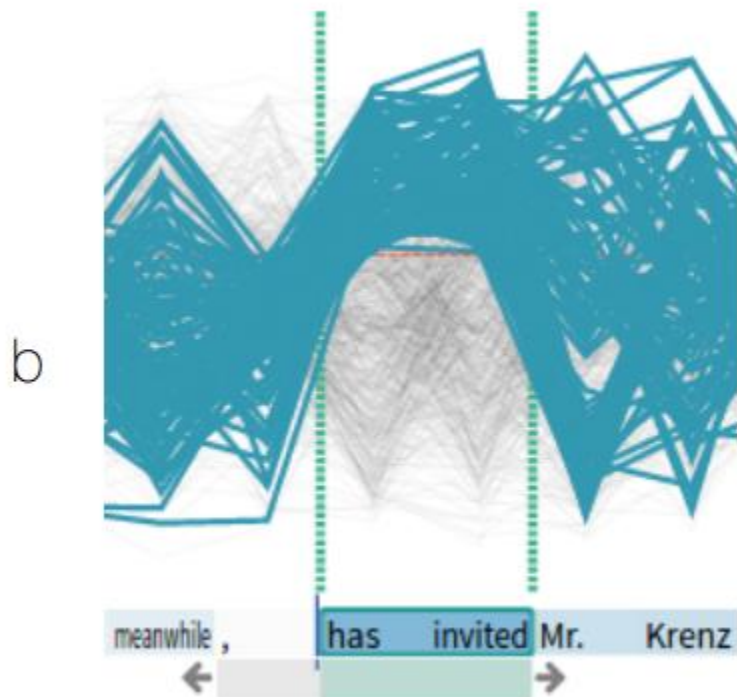
```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
    char *str;
    if (!*bufp || (len == 0) || (len > *remain))
        return ERR_PTR(-EINVAL);
    /* Of the currently implemented string fields, PATH_MAX
     * defines the longest valid length.
     */
}
```

Visualizing Hidden State Dynamics

Observe changes in hidden state representation overtime

Tool : LSTMVis

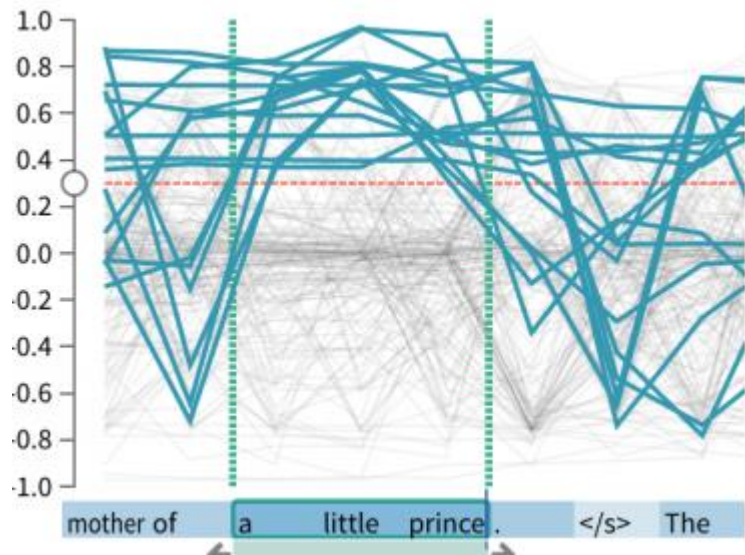
Visualizing Hidden State Dynamics



meanwhile ,	has	invited	Mr.	Krenz	to
however ,	has	n't	been	bad	enough
) ,	has	used	his	position	to
all ,	has	mainly	been	for	the
said ,	has	n't	yet	determined	what
Commission ,	has	promised	Poland	and	Hungary
however ,	have	n't	stopped	asset-backed	securities
central bank	has	allowed	a	key	interest
Inc. ,	has	failed	to	make	about
Life ,	has	agreed	not	to	make
Sen regime	has	sent	thousands	of	<unk>
resigned ,	has	helped	renew	calls	for
) ,	has	only	one	produced	picture

Hendrick et al, "Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks"

Visualizing Hidden State Dynamics



mother	of	a	little	prince	.	</s>
mother	of	a	little	prince	.	</s>
wife	in	a	little	hut	,	which
presence	of	a	little	old	woman	.
lived	in	a	little	cottage	with	her
her	in	a	great	nobleman	;	and
,	in	a	white	coat	and	a
not	in	a	good	temper	,	`
hare	in	a	fishing	net	and	fastened

Hendrick et al, "Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks"

Key Takeaways

- **Deeper RNNs** are more expressive
 - Feedforward depth
 - Recurrent depth
- **Long term dependencies** are a major problem in RNNs. Solutions:
 - Intelligent weight initialization
 - LSTMs / GRUs
- **Regularization** helps
 - Batch Norm: faster convergence
 - Dropout: better generalization
- **Visualization** helps
 - Analyze finer details of features produced by RNNs

References

Survey Papers

Lipton, Zachary C., John Berkowitz, and Charles Elkan. [A critical review of recurrent neural networks for sequence learning](#), arXiv preprint arXiv:1506.00019 (2015).

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. [Chapter 10: Sequence Modeling: Recurrent and Recursive Nets](#). MIT Press, 2016.

Training

Semeniuta, Stanislau, Aliaksei Severyn, and Erhardt Barth. [Recurrent dropout without memory loss](#). arXiv preprint arXiv:1603.05118 (2016).

Arjovsky, Martin, Amar Shah, and Yoshua Bengio. [Unitary evolution recurrent neural networks](#). arXiv preprint arXiv:1511.06464 (2015).

Le, Quoc V., Navdeep Jaitly, and Geoffrey E. Hinton. [A simple way to initialize recurrent networks of rectified linear units](#). arXiv preprint arXiv:1504.00941 (2015).

Cooijmans, Tim, et al. [Recurrent batch normalization](#). arXiv preprint arXiv:1603.09025 (2016).

References (contd)

Architectural Complexity Measures

Zhang, Saizheng, et al. [Architectural Complexity Measures of Recurrent Neural Networks](#). Advances in Neural Information Processing Systems. 2016.

Pascanu, Razvan, et al. [How to construct deep recurrent neural networks](#). arXiv preprint arXiv:1312.6026 (2013).

RNN Variants

Zilly, Julian Georg, et al. [Recurrent highway networks](#). arXiv preprint arXiv:1607.03474 (2016)

Chung, Junyoung, Sungjin Ahn, and Yoshua Bengio. [Hierarchical multiscale recurrent neural networks](#), arXiv preprint arXiv:1609.01704 (2016).

Visualization

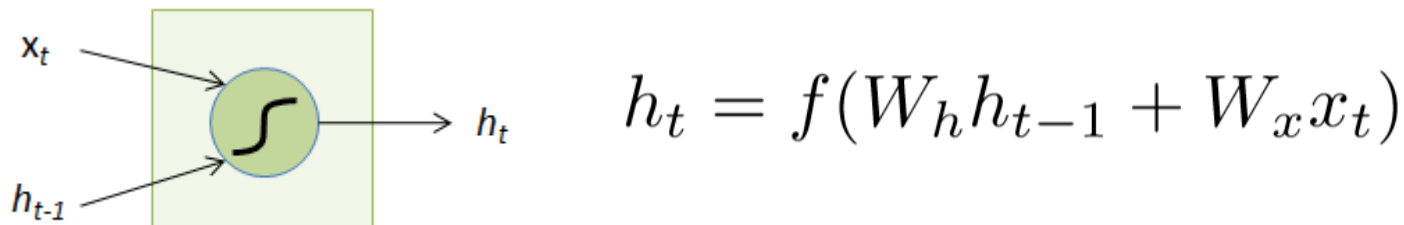
Karpathy, Andrej, Justin Johnson, and Li Fei-Fei. [Visualizing and understanding recurrent networks](#). arXiv preprint arXiv:1506.02078 (2015).

Hendrik Strobelt, Sebastian Gehrmann, Bernd Huber, Hanspeter Pfister, Alexander M. Rush. [LSTMVis: Visual Analysis for RNN](#), arXiv preprint arXiv:1606.07461 (2016).

Appendix

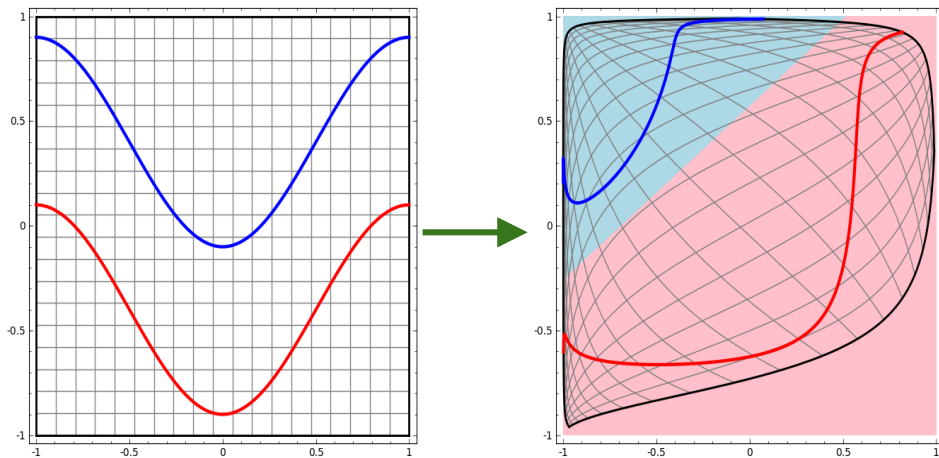
Why go deep?

Another Perspective of the RNN



- **Affine transformation + element-wise non-linearity**
- It is equivalent to **one fully connected layer** feedforward NN
- **Shallow** transformation

Visualizing Shallow Transformations



The Fully Connected Layer does 2 things:

1: Stretch / Rotate (affine)

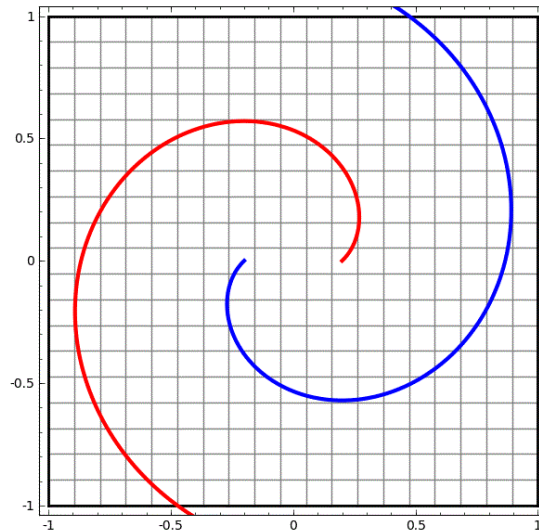
2: Distort (non-linearity)

Linear separability is achieved!

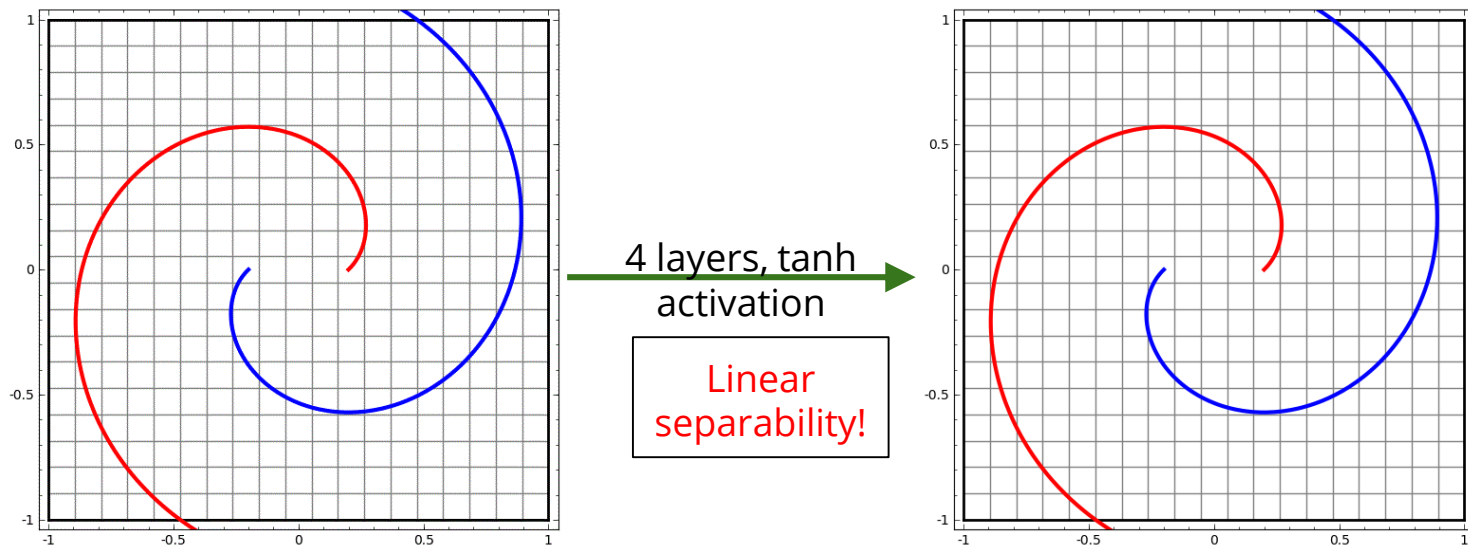
Shallow isn't always enough

Linear Separability **may not be achieved** for more complex datasets using **just one layer**
⇒ NN isn't expressive enough!

Need more layers.



Visualizing Deep Transformations

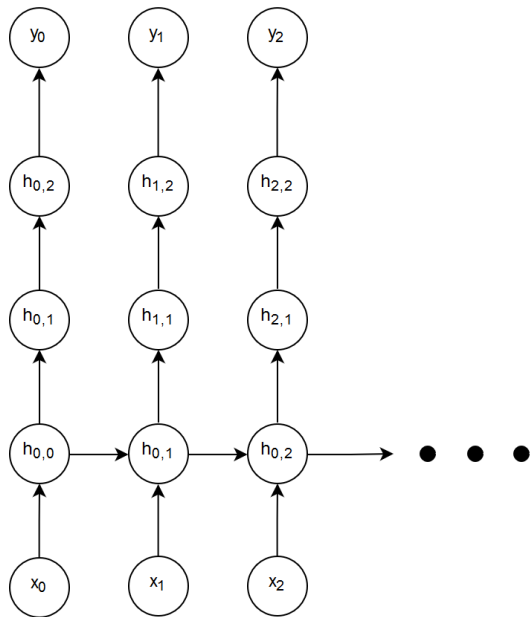


Deeper networks utilize high level features \Rightarrow more expressive!

Can you tell apart the effect of each layer?

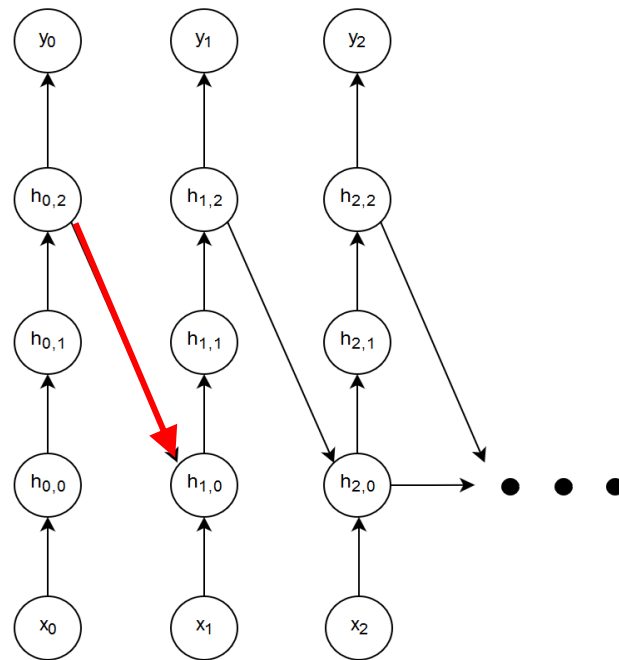
Source: <http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>

Which is more expressive?



Recurrent depth = 1
Feedforward depth = 4

Higher level features
passed on \Rightarrow win!



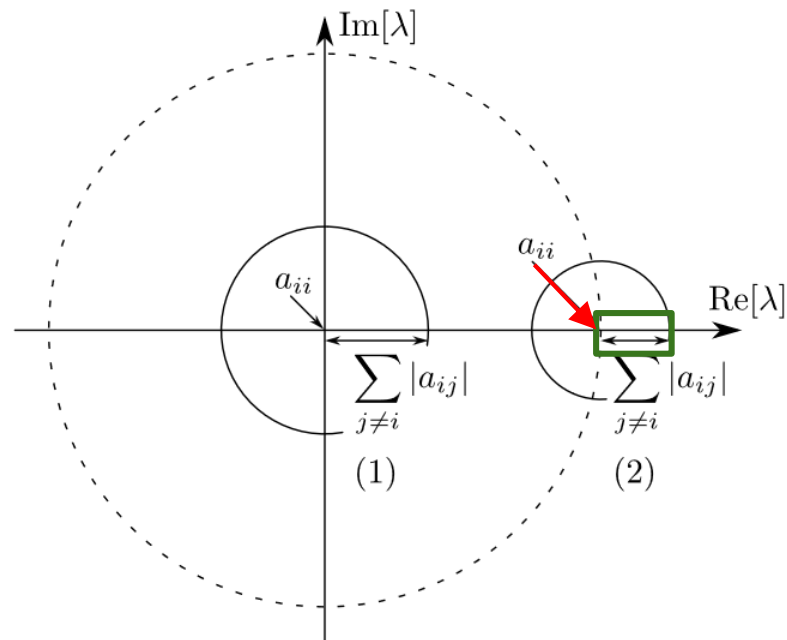
Recurrent depth = 3
Feedforward depth = 4

Gershgorin Circle Theorem (GCT)

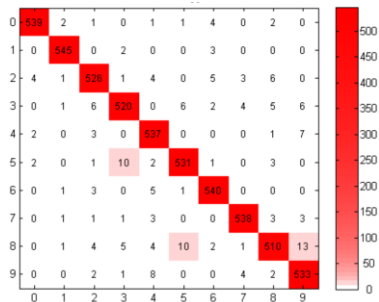
Gershgorin Circle Theorem (GCT)

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdot & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & a_{23} & & & & \cdot \\ a_{31} & a_{32} & a_{33} & & & & \cdot \\ \cdot & & & \cdot & & & \cdot \\ \cdot & & & & \cdot & & \cdot \\ \cdot & & & & & \cdot & \cdot \\ a_{n1} & \cdot & \cdot & \cdot & \cdot & \cdot & a_{nn} \end{bmatrix}$$

For any square matrix: The set of all eigenvalues is the **union of circles** whose **centers are a_{ii}** and the **radii are $\sum_{j \neq i} |a_{ij}|$**



Implications of GCT



Nearly diagonal matrix

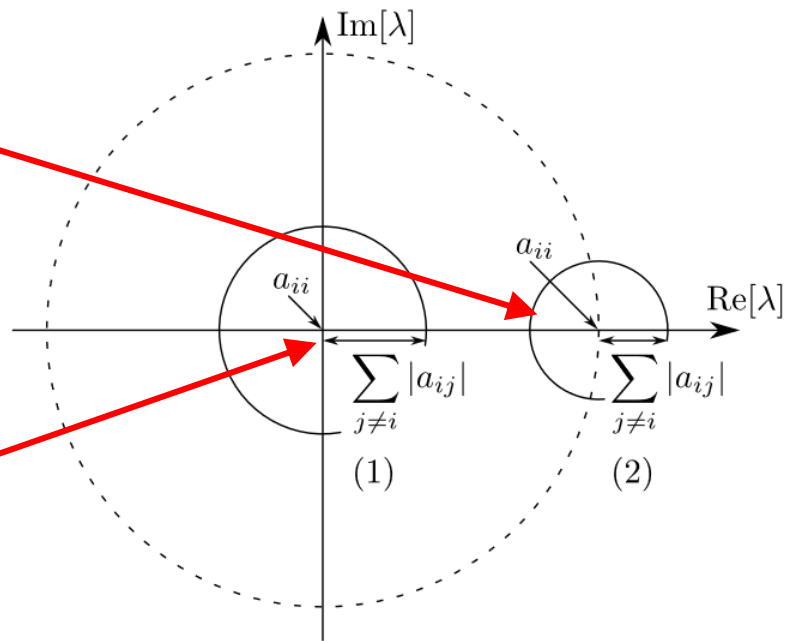
Source: https://de.mathworks.com/products/demos/machine-learning/handwriting_recognition/handwriting_recognition.html



Diffused matrix
(strong off-diagonal terms), mean of all terms = 0

Source: <https://i.stack.imgur.com/9inAk.png>

Zilly, Julian Georg, et al. "Recurrent highway networks." *arXiv preprint arXiv:1607.03474* (2016).



More Weight Initialization Methods

Weight Initialization Trick #2: np-RNN

- W_h positive semi-definite (+ve real eigenvalues)
- At least one eigenvalue is 1, others all less than equal to one
- Activation Function: ReLU

$$A = \frac{1}{N} \langle R^T, R \rangle$$

$$e = \max(\lambda(A + I))$$

$$W_{hh} = \frac{I + A}{e}$$

- R: standard normal matrix, values drawn from a Gaussian distribution with mean zero and unit variance
- N: size of R
- \langle, \rangle dot product
- e: Maximum eigenvalue of (A+I)

Weight Initialization Trick #3: Unitary Matrix

Unitary Matrix: $W_h W_h^* = \mathbf{I}$ (note: weight matrix is now complex!)

(W_h^* is the complex conjugate matrix of W_h)

All eigenvalues of W_h have absolute value 1

Challenge: Keeping a Matrix Unitary over time

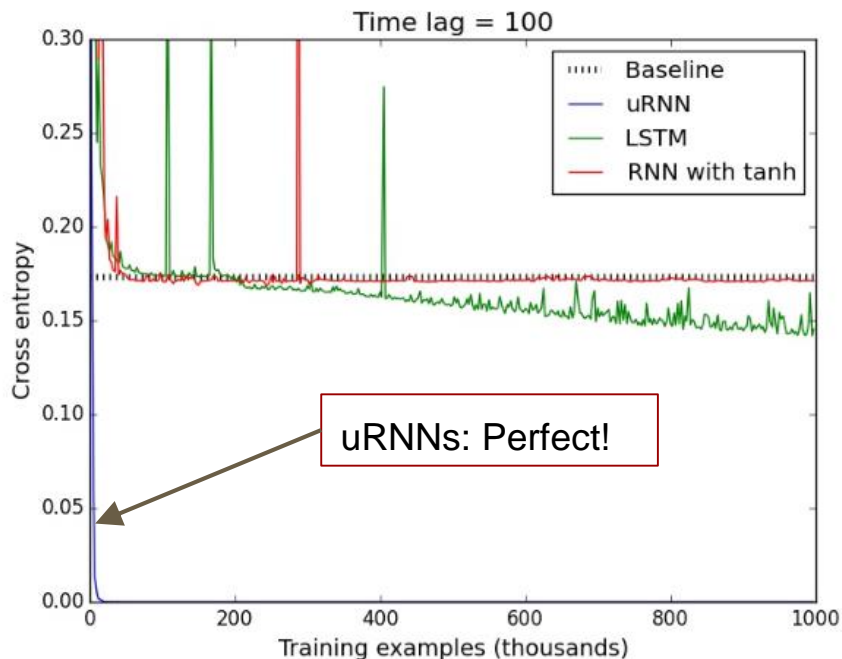
Efficient Solution: Parametrize the matrix

$$W = D_3 R_2 \mathcal{F}^{-1} D_2 \Pi R_1 \mathcal{F} D_1.$$

Rank 1 Matrices derived from vectors

- Storage and updates: **$O(n)$** : efficient!

Results for the Copying Memory Problem



- Input:

0 $a_1 a_2 \dots a_{10}$ $a_{10} 0 0 0 0 0 \dots$
10 symbols T zeros

- Output: $a_1 \dots a_{10}$
- **Challenge:** Remembering symbols over an arbitrarily large time gap

Cross entropy for the copying memory problem

Arjovsky, Martin, Amar Shah, and Yoshua Bengio. "Unitary evolution recurrent neural networks." (2015).

Summary

Model	I-RNN	np-RNN	Unitary-RNN
Activation Function	ReLu	ReLu	ReLu
Initialization	Identity Matrix	Positive Semi-definite (normalized eigenvalues)	Unitary Matrix
Performance compared to LSTM	Less than or equal	Equal	Greater
Benchmark Tasks	Action Recognition, Addition, MNIST	Action Recognition, Addition MNIST	Copying Problem, Adding Problem
Sensitivity to hyper-parameters	High	Low	Low

Dropout

Model Moon (2015)

$$\mathbf{c}_t = d(\mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * \mathbf{g}_t)$$

Moon (2015)

Able to learn long term dependencies, not capable of exploiting them during test phase

Test time equations for GRU,

$$\mathbf{h}_t = (\mathbf{h}_{t-1} + \mathbf{g}_t)p$$

$$\mathbf{h}_t = ((\mathbf{h}_{t-2} + \mathbf{g}_{t-1})p + \mathbf{g}_t)p$$

$$\mathbf{h}_t = p^{t+1}\mathbf{h}_0 + \sum_{i=0}^t p^{t-i+1}\mathbf{g}_i$$

- P is the probability to not drop a neuron
- For large t, hidden state contribution is close to zero during test

Model Barth (2016)

$$\mathbf{c}_t = \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * d(\mathbf{g}_t)$$

Barth (2016)

Drop differences that are added to the network, not the actual values

Allows to use per-step dropout

Test time equation after recursion,

$$\mathbf{h}_t = p\mathbf{h}_0 + \sum_{i=0}^t p \mathbf{g}_i = p\mathbf{h}_0 + p \sum_{i=0}^t \mathbf{g}_i$$

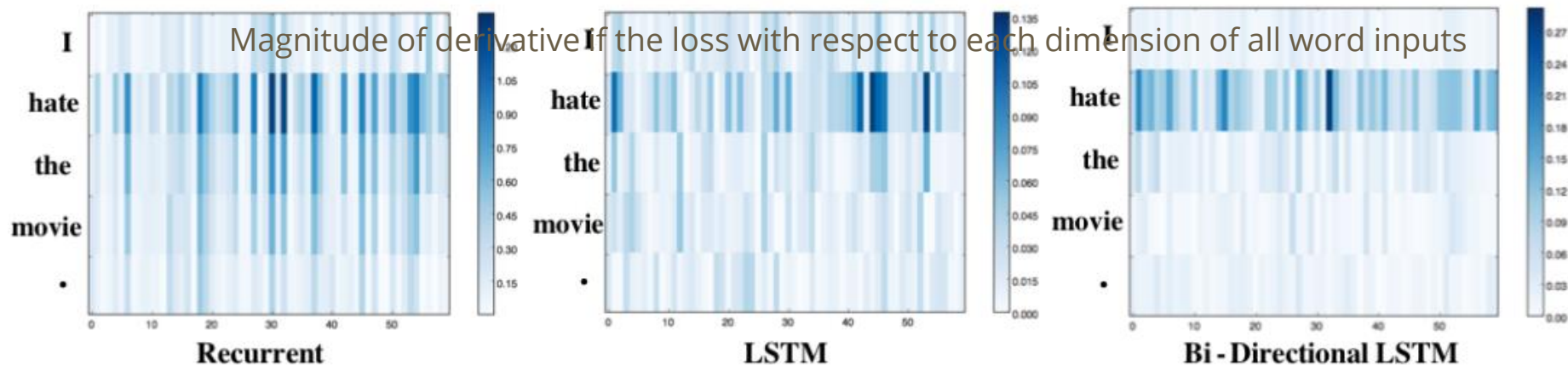
- P is the probability to not drop a neuron
- For large t, hidden state contribution is retained as at train time

Visualization

Visualize gradients: Saliency maps

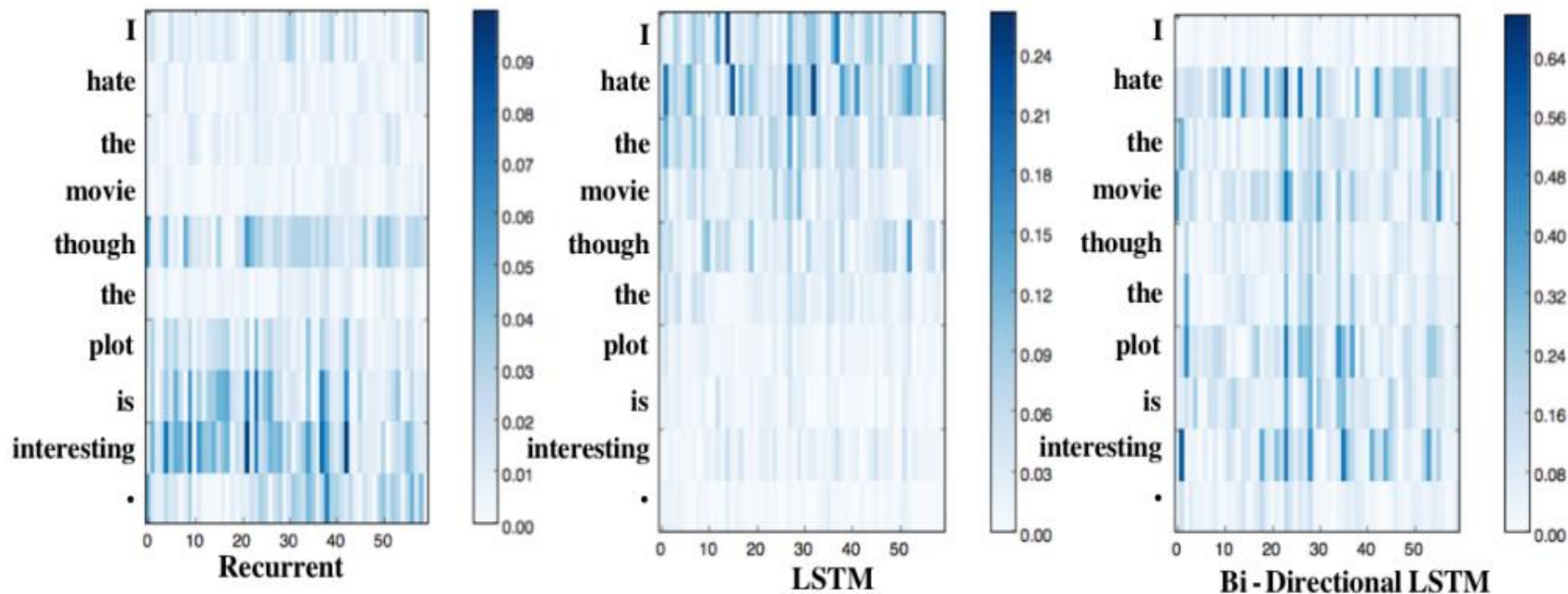
Categorize phrase/sentence into (v.positive, positive, neutral, negative, v.negative)

How much each unit contributes to the decision ?



“Jiwei LI et al, “Visualizing and Understanding Neural Models in NLP”

Visualize gradients: Saliency maps



"Jiwei LI et al, "Visualizing and Understanding Neural Models in NLP"

Error Analysis

N-gram Errors

Dynamic n-long memory Errors

Rare word Errors

Word model Errors

Punctuation Errors

Boost Errors

“Karpathy et al, Visualizing and Understanding Recurrent Networks”

	50K -> 500K parameter model
Reduced Total Errors	44K (184K-140K)
N-gram Error	81% (36K/44K)
Dynamic n-long memory Errors	1.7% (0.75K/44k)
Rare words Error	1,7% (0.75K/44K)
Word model Error	1.7% (0.75K/44k)
Punctuation Error	1,7% (0.75K/44K)
Boost Error	11.36% (5K/44K)

Error Analysis: Conclusions

- N-gram Errors
 - Scale model
- Dynamic n-long memory
 - Memory Networks
- Rare words
 - Increase training size
- Word Level Predictions/ Punctuations:
 - Hierarchical context models
 - Stacked Models
 - GF RNN, CW RNN

“Karpathy et al, Visualizing and Understanding Recurrent Networks”

Recurrent Highway Networks

Understanding Long Term Dependencies from Jacobian

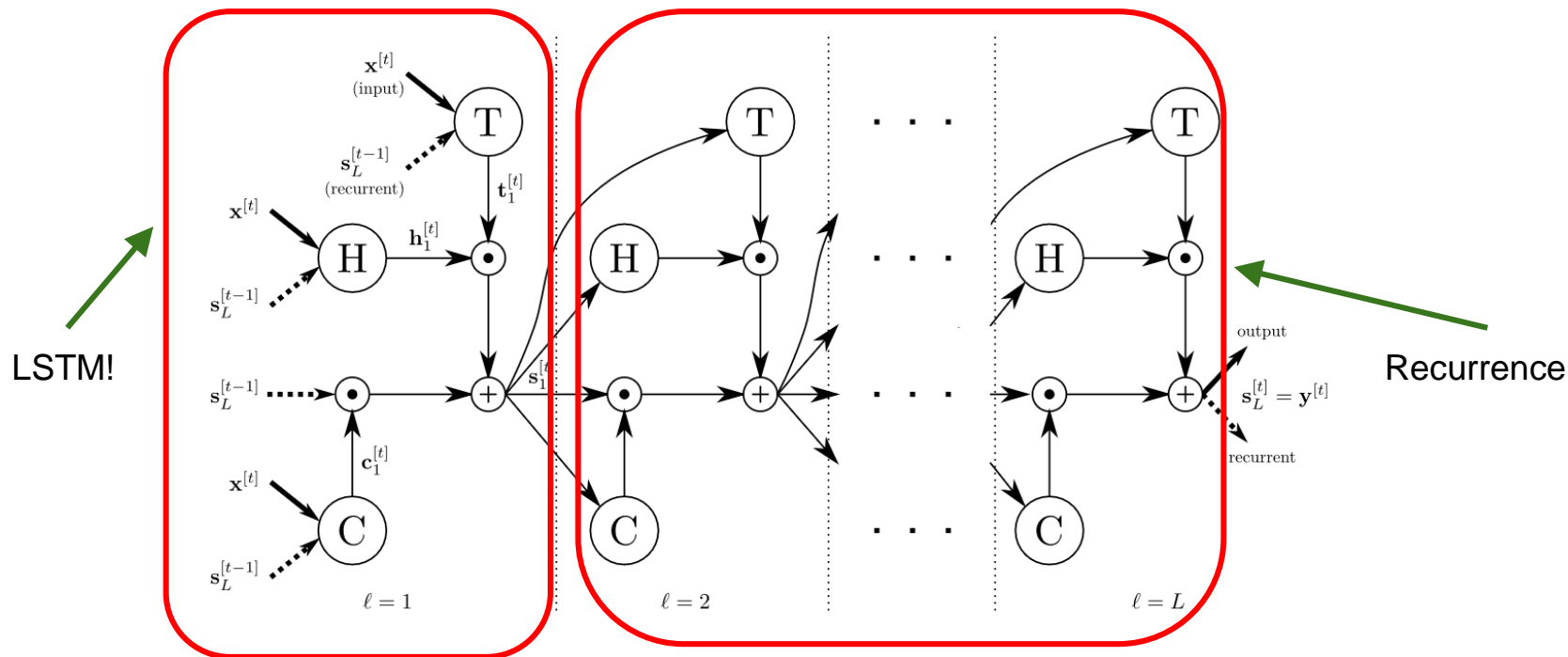
Learning long term dependencies is a challenge because:

If the Jacobian has a spectral radius (absolute largest eigenvalue) < 1 , the network faces **vanishing gradients**. Here it happens if $\gamma \sigma_{max} < 1$

Hence, ReLU's are an attractive option! They have $\sigma_{max} = 1$ (given at least one positive element)

If the Jacobian has a spectral radius > 1 , the network faces **exploding gradients**

Recurrent Highway Networks (RHN)



RHN Equations

RHN:

Recurrent depth

Feedforward depth (not shown)

$$\begin{aligned} \mathbf{h} &= H(x, \mathbf{W}_H) \\ \mathbf{t} &= T(x, \mathbf{W}_T) \\ \mathbf{c} &= C(x, \mathbf{W}_C) \end{aligned}$$

Input transform: $\mathbf{y} = \mathbf{h} \cdot \mathbf{t} + \mathbf{x} \cdot \mathbf{c}$

Note: h is transformed input, y is state

T, C: Transform, Carr $\mathbf{s}_\ell^{[t]} = \mathbf{h}_\ell^{[t]} \cdot \mathbf{t}_\ell^{[t]} + \mathbf{s}_{\ell-1}^{[t]} \cdot \mathbf{c}_\ell^{[t]}$

$$\mathbf{h}_\ell^{[t]} = \tanh(\mathbf{W}_{H\ell} \mathbf{x}^{[t]} \mathbb{I}_{\{\ell=1\}} + \mathbf{R}_{H\ell} \mathbf{s}_{\ell-1}^{[t]} + \mathbf{b}_{H\ell}),$$

RHN Output:

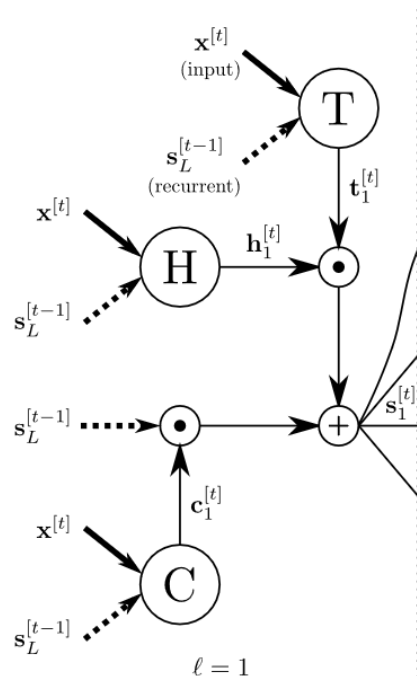
$$\mathbf{t}_\ell^{[t]} = \sigma(\mathbf{W}_{T\ell} \mathbf{x}^{[t]} \mathbb{I}_{\{\ell=1\}} + \mathbf{R}_{T\ell} \mathbf{s}_{\ell-1}^{[t]} + \mathbf{b}_{T\ell}),$$

$$\mathbf{c}_\ell^{[t]} = \sigma(\mathbf{W}_{C\ell} \mathbf{x}^{[t]} \mathbb{I}_{\{\ell=1\}} + \mathbf{R}_{C\ell} \mathbf{s}_{\ell-1}^{[t]} + \mathbf{b}_{C\ell}),$$

State update equation for RHN with recurrence depth L:

Indicator function

Recurrence layer Index



Gradient Equations in RHN

$$\mathbf{y}^{[t]} = \mathbf{h}^{[t]} \cdot \mathbf{t}^{[t]} + \mathbf{y}^{[t-1]} \cdot \mathbf{c}^{[t]}$$

For an RHN with **recurrence depth 1**, RHN Output is:

Jacobian is simple: $\mathbf{A} = \partial \mathbf{y}^{[t]} / \partial \mathbf{y}^{[t-1]}$ $\mathbf{A} = \text{diag}(\mathbf{c}^{[t]}) + \mathbf{H}' \text{diag}(\mathbf{t}^{[t]}) + \mathbf{C}' \text{diag}(\mathbf{y}^{[t-1]}) + \mathbf{T}' \text{diag}(\mathbf{h}^{[t]})$

$$\mathbf{H}' = \mathbf{R}_H^\top \text{diag}[\text{tanh}'(\mathbf{R}_H \mathbf{y}^{[t-1]})],$$

But then $\mathbf{T}' = \mathbf{R}_T^\top \text{diag}[\sigma'(\mathbf{R}_T \mathbf{y}^{[t-1]})],$

where $\mathbf{C}' = \mathbf{R}_C^\top \text{diag}[\sigma'(\mathbf{R}_C \mathbf{y}^{[t-1]})],$

$$\mathbf{c}_i^{[t]} + \mathbf{H}'_{ii} \mathbf{t}_i^{[t]} + \mathbf{C}'_{ii} \mathbf{y}_i^{[t-1]} + \mathbf{T}'_{ii} \mathbf{h}_i^{[t]}$$

$$\sum_{j=1, j \neq i}^n |\mathbf{H}'_{ij} \mathbf{t}_j^{[t]} + \mathbf{C}'_{ij} \mathbf{y}_j^{[t-1]} + \mathbf{T}'_{ij} \mathbf{h}_j^{[t]}|$$

The eigenvalues lie within these circles

Using the above and GCT, the centers of the circles are:

The radii are:

Analysis

$$\mathbf{c}_i^{[t]} + \mathbf{H}'_{ii} \mathbf{t}_i^{[t]} + \mathbf{C}'_{ii} \mathbf{y}_i^{[t-1]} + \mathbf{T}'_{ii} \mathbf{h}_i^{[t]}$$

Centers:

$$\sum_{j=1, j \neq i}^n |\mathbf{H}'_{ij} \mathbf{t}_i^{[t]} + \mathbf{C}'_{ij} \mathbf{y}_i^{[t-1]} + \mathbf{T}'_{ij} \mathbf{h}_i^{[t]}|$$

, radii.

If we wish to completely remember the previous state: $\mathbf{c} = \mathbf{1}$, $\mathbf{t} = \mathbf{0}$

$$\text{Saturation} \Rightarrow \mathbf{T}' = \mathbf{C}' = \mathbf{0}_{n \times n}$$

Thus, centers (λ) are $\mathbf{1}$, radii are $\mathbf{0}$

If we wish to completely forget the previous state: $\mathbf{c} = \mathbf{0}$, $\mathbf{t} = \mathbf{1}$

Eigenvalues are those of \mathbf{H}'

Possible to span the spectrum between these two cases by adjusting the Jacobian \mathbf{A}

(*) Increasing depth improves expressivity

Results

Model	Size	Best Val.	Test
RNN-LDA + KN-5 + cache (Mikolov & Zweig, 2012)	9 M	-	92.0
Conv.+Highway+LSTM+dropout (Kim et al., 2015)	19 M	-	78.9
LSTM+dropout (Zaremba et al., 2014)	66 M	82.2	78.4
Variational LSTM (Gal, 2015)	66 M	77.3	75.0
Variational LSTM + WT (Press & Wolf, 2016)	51 M	75.8	73.2
Pointer Sentinel-LSTM (Merity et al., 2016)	21 M	72.4	70.9
Variational LSTM + WT + augmented loss (Inan et al., 2016)	51 M	71.1	68.5
Variational RHN	32 M	71.2	68.5
Neural Architecture Search with base 8 (Zoph & Le, 2016)	32 M	-	67.9
Variational RHN + WT	23 M	67.9	65.4
Neural Architecture Search with base 8 + WT (Zoph & Le, 2016)	25 M	-	64.0
Neural Architecture Search with base 8 + WT (Zoph & Le, 2016)	54 M	-	62.4

BPC on Penn Treebank

Model	BPC	Size
Grid-LSTM (Kalchbrenner et al., 2015)	1.47	17 M
MI-LSTM (Wu et al., 2016)	1.44	17 M
mLSTM (Krause et al., 2016)	1.42	21 M
LN HyperNetworks (Ha et al., 2016)	1.34	27 M
LN HM-LSTM (Chung et al., 2016)	1.32	35 M
RHN - Rec. depth 5	1.31	23 M
RHN - Rec. depth 10	1.30	21 M
Large RHN - Rec. depth 10	1.27	46 M

BPC on enwiki8 (Hutter Prize)

Model	BPC	Size
MI-LSTM (Wu et al., 2016)	1.44	17 M
mLSTM (Krause et al., 2016)	1.40	10 M
BN LSTM (Cooijmans et al., 2016)	1.36	16 M
HM-LSTM (Chung et al., 2016)	1.32	35 M
LN HM-LSTM (Chung et al., 2016)	1.29	35 M
RHN - Rec. depth 10	1.29	20 M
Large RHN - Rec. depth 10	1.27	45 M

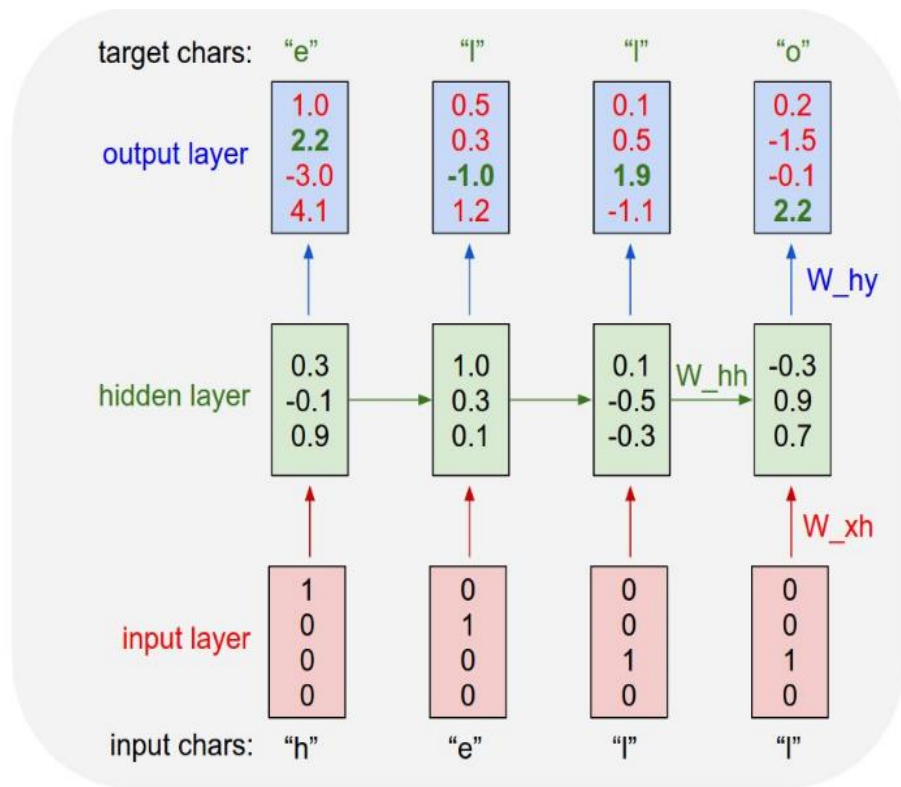
BPC on text8 (Hutter Prize)

LSTMs for Language Models

LSTMs are Very Effective!

Application: Language Model

Task: Predicting the next character given the current character



Train Input: Wikipedia Data

Hutter Prize 100 MB Dataset of raw wikipedia, 96 MB for training

Trained overnight on a LSTM

Generated Text:

Naturalism and decision for the majority of Arab countries' capitalide was grounded by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]] associated with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal in the [[Protestant Immineners]], which could be said to be directly in Cantonese Communication, which followed a ceremony and set inspired prison, training. The emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known in western [[Scotland]], near Italy to the conquest of India with the conflict. Copyright was the succession of independence in the slop of Syrian influence that was a famous German movement based on a more popular servicious, non-doctrinal and sexual power post. Many governments recognize the military housing of the [[Civil Liberalization and Infantry Resolution 265 National Party in Hungary]], that is sympathetic to be to the [[Punjab Resolution]]
(P.S)[<http://www.humah.yahoo.com/guardian.cfm/7754800786d17551963589.htm> Official economics Adjoint for the Nazism, Montgomery was swear to advance to the resources for those Socialism's rule, was starting to signing a major tripad of aid exile.]]

- Remembers to close a bracket
- Capitalize nouns
- 404 Page Not Found! :P The LSTM hallucinates it.

Train Input:

16MB of Latex source of algebraic stacks/geometry

Trained on Multi-Layer LSTM

Test Output

Generated Latex files “almost” compile, the authors had to fix some issues manually

We will look at some of these errors

Generated Latex Source Code

```
\begin{proof}
```

```
We may assume that  $\mathcal{I}$  is an abelian sheaf on  $\mathcal{C}$ .
```

```
\item Given a morphism  $\Delta : \mathcal{F} \rightarrow \mathcal{I}$ 
```

```
is an injective and let  $\mathfrak{q}$  be an abelian sheaf on  $X$ .
```

```
Let  $\mathcal{F}$  be a fibered complex. Let  $\mathcal{F}$  be a category.
```

```
\begin{enumerate}
```

```
\item \hyperref[setain-construction-phantom]{Lemma}
```

```
\label{lemma-characterize-quasi-finite}
```

```
Let  $\mathcal{F}$  be an abelian quasi-coherent sheaf on  $\mathcal{C}$ .
```

```
Let  $\mathcal{F}$  be a coherent  $\mathcal{O}_X$ -module. Then
```

```
 $\mathcal{F}$  is an abelian catenary over  $\mathcal{C}$ .
```

```
\item The following are equivalent
```

```
\begin{enumerate}
```

```
\item  $\mathcal{F}$  is an  $\mathcal{O}_X$ -module.
```

```
\end{lemma}
```

- Begins with a proof but ends with a lemma
- Begins enumerate but does not end it
- Likely because of the long term dependency.
- Can be reduced with larger/better models

Compiled Latex Files: Hallucinated Algebra

For $\bigoplus_{n=1, \dots, m}$ where $\mathcal{L}_{m, \bullet} = 0$, hence we can find a closed subset \mathcal{H} in \mathcal{H} and any sets \mathcal{F} on X , U is a closed immersion of S , then $U \rightarrow T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \text{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \rightarrow V$. Consider the maps M along the set of points Sch_{fppf} and $U \rightarrow U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ?? . Hence we obtain a scheme S and any open subset $W \subset U$ in $\text{Sh}(G)$ such that $\text{Spec}(R') \rightarrow S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S . We claim that $\mathcal{O}_{X, x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X, x'} \rightarrow \mathcal{O}_{X', x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\text{GL}_{S'}(x'/S'')$ and we win. \square

To prove study we see that $\mathcal{F}|_U$ is a covering of \mathcal{X}' , and T_i is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and \mathcal{F}_p exists and let \mathcal{F}_i be a presheaf of \mathcal{O}_X -modules on \mathcal{C} as a \mathcal{F} -module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to know that

$$\tilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\text{Spec}(k)} \mathcal{O}_{S, s} - i_X^{-1} \mathcal{F}$$

is a unique morphism of algebraic stacks. Note that

$$\text{Arrows} = (\text{Sch}/S)_{fppf}^{\text{opp}} \cdot (\text{Sch}/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \rightarrow (U, \text{Spec}(A))$$

is an open subset of X . Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S .

Proof. See discussion of sheaves of sets. \square

The result for prove any open covering follows from the less of Example ?? . It may replace S by $X_{\text{spaces}, \text{étale}}$ which gives an open subspace of X and T equal to S_{Zar} , see Descent, Lemma ?? . Namely, by Lemma ?? we see that R is geometrically regular over S .

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

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

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

When in this case of to show that $\mathcal{Q} \rightarrow \mathcal{C}_{Z/X}$ 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 $S = \text{Spec}(R)$ and $Y = \text{Spec}(R)$.

Proof. This is form all sheaves of sheaves on X . But given a scheme U and a surjective étale morphism $U \rightarrow X$. Let $U \cap U = \coprod_{i=1, \dots, n} U_i$ be the scheme X over S at the schemes $X_i \rightarrow X$ and $U = \text{lim}_i X_i$. \square

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

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 $i_0 \leq p$ is a subset of $\mathcal{J}_{n,0} \circ \mathcal{A}_2$ works.

Lemma 0.3. In Situation ?? . Hence we may assume $q' = 0$.

Proof. We will use the property we see that p is the next 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 \square

- Generates Lemmas and their proofs
- Equations with correct latex structure
- No, they dont mean anything yet!

Compiled Latex Files: Hallucinated Algebra

Nice try on the diagrams!

Proof. Omitted.



Lemma 0.1. Let \mathcal{C} be a set of the construction.

Let \mathcal{C} be a gerber covering. Let \mathcal{F} be a quasi-coherent sheaves of \mathcal{O} -modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

Proof. This is an algebraic space with the composition of sheaves \mathcal{F} on $X_{\text{étale}}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{\text{morph}_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where \mathcal{G} defines an isomorphism $\mathcal{F} \rightarrow \mathcal{F}$ of \mathcal{O} -modules.

Lemma 0.2. This is an integer \mathcal{Z} is injective.

Proof. See Spaces, Lemma ??.

Lemma 0.3. Let \mathcal{S} be a scheme. Let X be a scheme and X is an affine open covering. Let $U \subset X$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \rightarrow Y' \rightarrow Y \rightarrow Y \rightarrow Y' \times_X Y \rightarrow X.$$

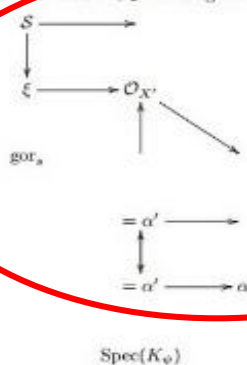
be a morphism of algebraic spaces over S and Y .

Proof. Let X be a nonzero scheme of X . Let X be an algebraic space. Let \mathcal{F} be a quasi-coherent sheaf of \mathcal{O}_X -modules. The following are equivalent

- (1) \mathcal{F} is an algebraic space over S .
- (2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type.

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then \mathcal{G} is a finite type and assume S is a flat and \mathcal{F} and \mathcal{G} is a finite type f_* . This is of finite type diagrams, and

- the composition of \mathcal{G} is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

Proof. We have see that $X = \text{Spec}(R)$ and \mathcal{F} is a finite type representable by algebraic space. The property \mathcal{F} is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U .

Proof. This is clear that \mathcal{G} is a finite presentation, see Lemmas ??.
A reduced above we conclude that U is an open covering of \mathcal{C} . The functor \mathcal{F} is a "field"

$$\mathcal{O}_{X,x} \rightarrow \mathcal{F}_x \rightarrow \mathcal{O}_{X_{\text{étale}}} \rightarrow \mathcal{O}_{X'}^{-1} \mathcal{O}_{X'}(\mathcal{O}_{X'}^{\mathcal{G}})$$

is an isomorphism of covering of $\mathcal{O}_{X'}$. If \mathcal{F} is the unique element of \mathcal{F} such that X is an isomorphism.

The property \mathcal{F} is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme \mathcal{O}_X -algebra with \mathcal{F} are opens of finite type over S .

If \mathcal{F} is a scheme theoretic image points.

If \mathcal{F} is a finite direct sum $\mathcal{O}_{X'}$ is a closed immersion, see Lemma ??.
This is a sequence of \mathcal{F} is a similar morphism.