

# TIME SERIES ANALYSIS TUTORIAL

## EXERCISE 10

LEONARD BERESKA

CENTRAL INSTITUTE  
OF MENTAL HEALTH

JUNE 30, 2021

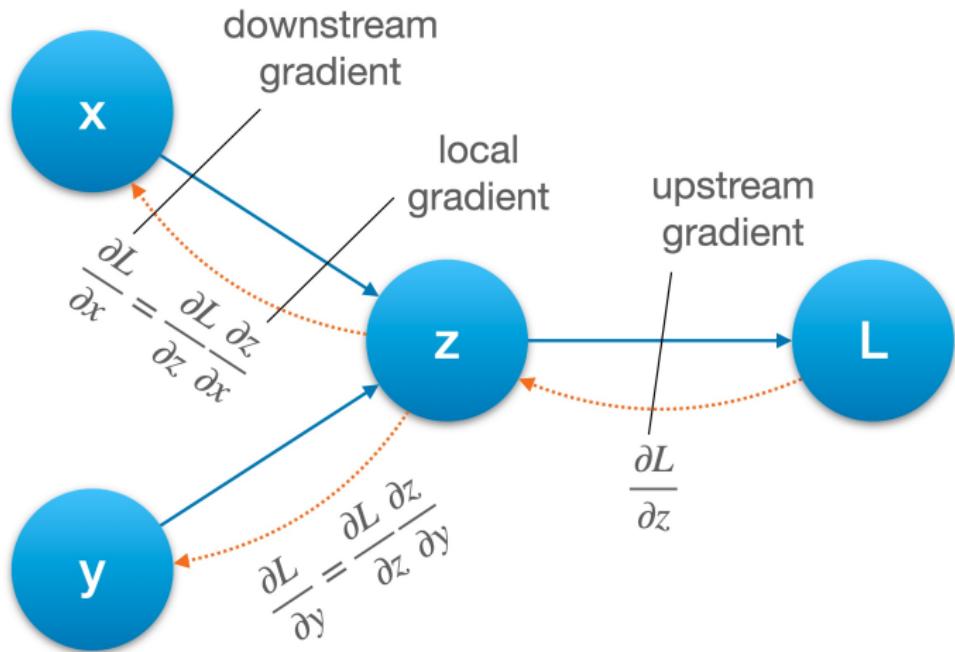


Zentralinstitut für  
Seelische Gesundheit  
Landesstiftung  
des öffentlichen Rechts

UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

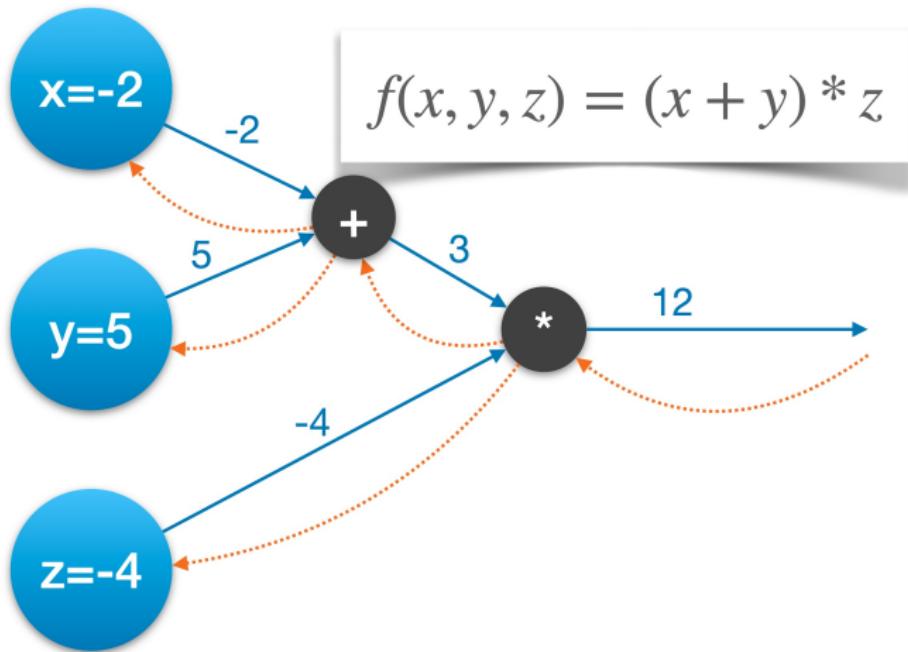
# **VANISHING & EXPLODING GRADIENTS**

# BACKPROPAGATION



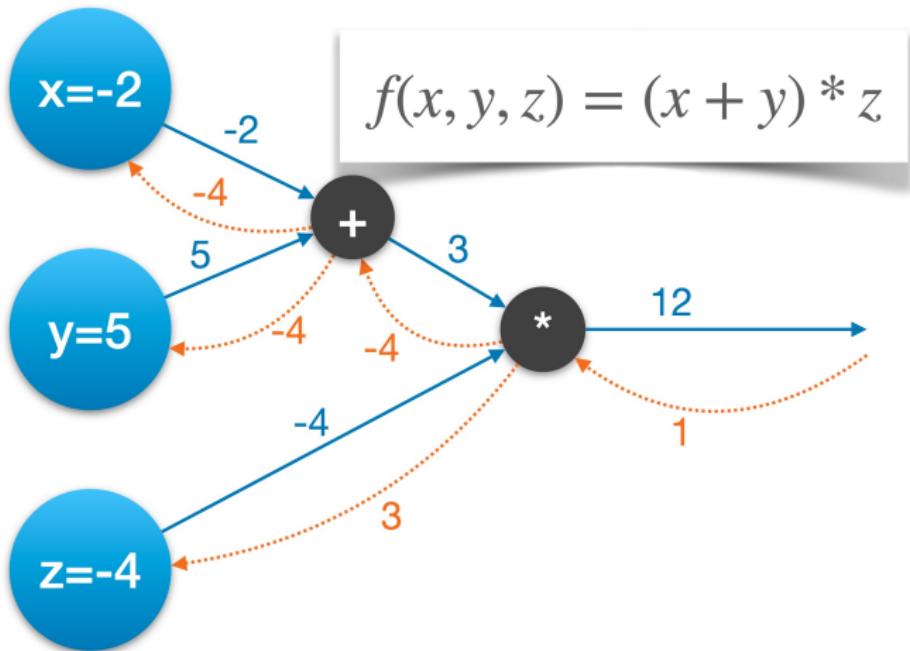
Downstream = Upstream \* Local gradient

# BACKPROPAGATION EXAMPLE



- Multiplication swaps input activations
- Addition passes gradient on

# BACKPROPAGATION EXAMPLE



- Multiplication swaps input activations
- Addition passes gradient on

# VANISHING & EXPLODING GRADIENTS

- Recurrent connections, e.g.  $h_t = \tanh(Wh_{t-1} + Ux_t + b)$ , can be approximated linearly with  $h_t = Wh_{t-1}$ .
- Repeated application of  $W$  leads to exponential growth along the eigenvector direction of maximum absolute eigenvalue:<sup>1</sup>

$$h_t = W^t h_0$$

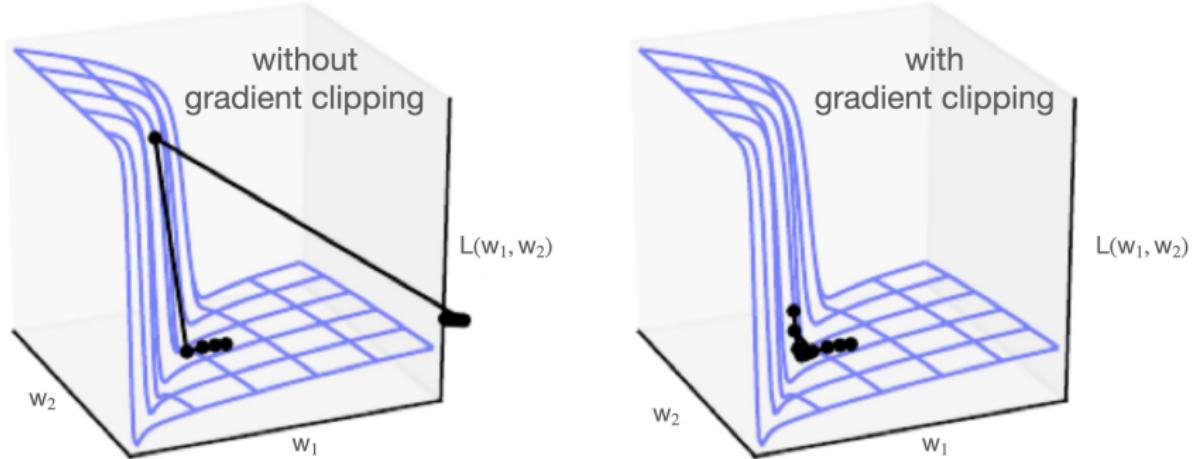
$$h_t = Q\Lambda^t Q^T h_0$$

Hence maximum absolute eigenvalue indicates both stability as the occurrence explosion or vanishing of gradients.

---

<sup>1</sup>If  $W$  admits an eigendecomposition:  $W^t = (Q\Lambda Q^T)^t = Q\Lambda^t Q^T$ .

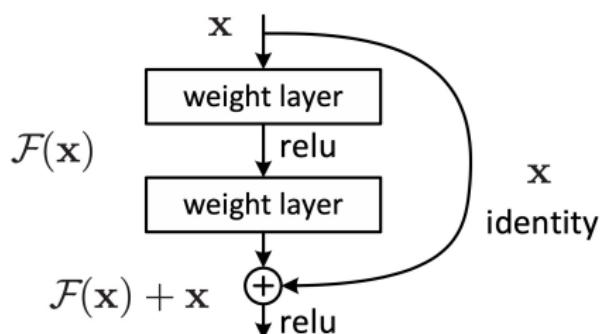
## GRADIENT CLIPPING [2]



- High curvature walls lead to bad gradient updates without clipping

Image from *Deep Learning* [1]

# SKIP CONNECTIONS FOR DEEP NEURAL NETWORKS

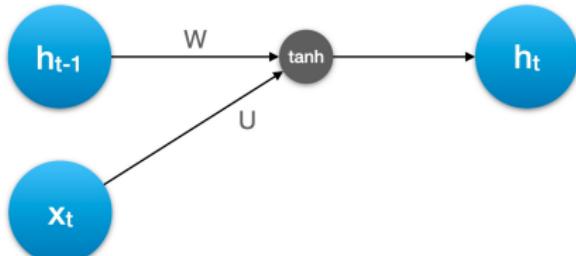


- Residual connections → *ResNet*
- Addition of identity maintains information
- Deep network training much easier

Image from He et al., 2015 [3]

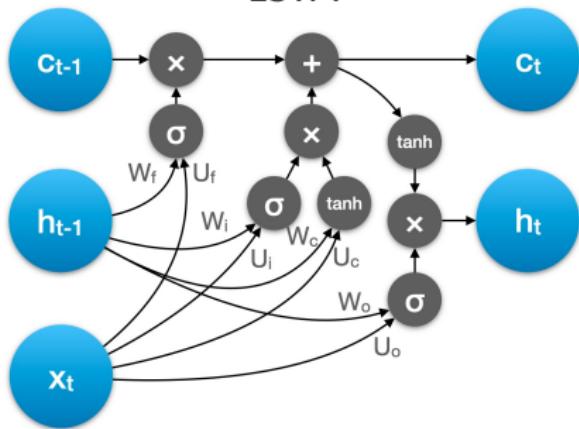
# LONG SHORT TERM MEMORY (LSTM) [4]

(standard) RNN



$$h_t = \tanh(Wh_{t-1} + Ux_t + b)$$

LSTM



$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f)$$

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)$$

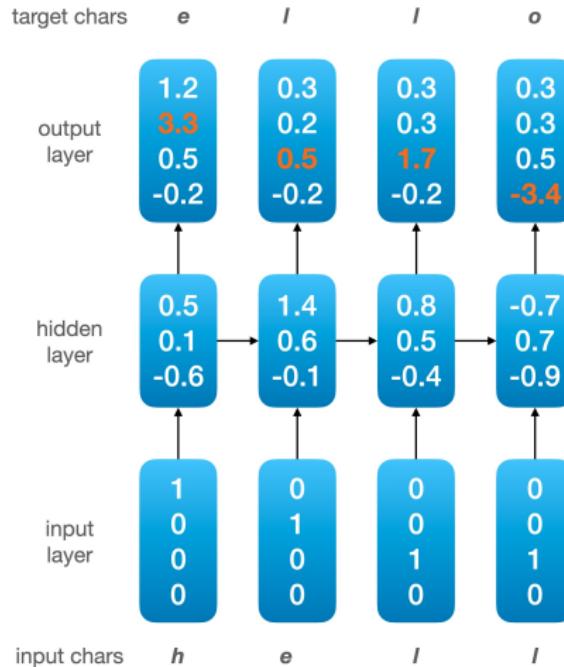
$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o)$$

$$\tilde{c}_t = \tanh(W_c h_{t-1} + U_c x_t + b_c)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$

$$h_t = o_t \circ \tanh(c_t)$$

# EXAMPLE: CHAR-LEVEL RNN



From [http://vision.stanford.edu/teaching/cs231n/slides/2016/winter1516\\_lecture10.pdf](http://vision.stanford.edu/teaching/cs231n/slides/2016/winter1516_lecture10.pdf)

# DATASET: OPEN SOURCE TEXTBOOK

The Stacks Project

home about tags explained tag lookup browse search bibliography recent comments blog add slogans

Browse chapters

Part	Chapter	online	TeX source	view pdf
Preliminaries	1. Introduction	<a href="#">online</a>	<a href="#">tex</a>	<a href="#">pdf &gt;</a>
	2. Conventions	<a href="#">online</a>	<a href="#">tex</a>	<a href="#">pdf &gt;</a>
	3. Set Theory	<a href="#">online</a>	<a href="#">tex</a>	<a href="#">pdf &gt;</a>
	4. Categories	<a href="#">online</a>	<a href="#">tex</a>	<a href="#">pdf &gt;</a>
	5. Topology	<a href="#">online</a>	<a href="#">tex</a>	<a href="#">pdf &gt;</a>
	6. Sheaves on Spaces	<a href="#">online</a>	<a href="#">tex</a>	<a href="#">pdf &gt;</a>
	7. Sites and Sheaves	<a href="#">online</a>	<a href="#">tex</a>	<a href="#">pdf &gt;</a>
	8. Stacks	<a href="#">online</a>	<a href="#">tex</a>	<a href="#">pdf &gt;</a>
	9. Fields	<a href="#">online</a>	<a href="#">tex</a>	<a href="#">pdf &gt;</a>
	10. Commutative Algebra	<a href="#">online</a>	<a href="#">tex</a>	<a href="#">pdf &gt;</a>

Parts

- [Preliminaries](#)
- [Schemes](#)
- [Topics in Scheme Theory](#)
- [Algebraic Spaces](#)
- [Topics in Geometry](#)
- [Deformation Theory](#)
- [Algebraic Stacks](#)
- [Miscellany](#)

Statistics

The Stacks project now consists of

- o 455910 lines of code
- o 14221 tags (56 inactive tags)
- o 2366 sections

From [http://vision.stanford.edu/teaching/cs231n/slides/2016/winter1516\\_lecture10.pdf](http://vision.stanford.edu/teaching/cs231n/slides/2016/winter1516_lecture10.pdf)

# GENERATED LATEX

For  $\bigoplus_{n=1,\dots,m} \mathcal{L}_{m,n} = 0$ , 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  $\text{Sch}_{fppf}$  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  $x, x', x'' \in S'$  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  $X'$ , and  $\mathcal{T}_i$  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  $C$  as a  $\mathcal{F}$ -module. In particular  $\mathcal{F} = U/\mathcal{F}$  we have to show that

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

is a unique morphism of algebraic stacks. Note that

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

and

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

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  $X^{\text{spaces},\text{etale}}$  which gives an open subspace of  $X$  and  $T$  equal to  $S_{\text{Zar}}$ , 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  $X = \lim |X|$  (by the formal open covering  $X$  and a single map  $\underline{\text{Proj}}_X(\mathcal{A}) = \text{Spec}(B)$  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  $U = \lim_i X_i$ .  $\square$

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  $\mathfrak{p}$  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  $\delta_{n+1}$  is a scheme over  $S$ .  $\square$

# VISUALIZING ACTIVATION [5]

## Cell indicating position in line

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.

# VISUALIZING ACTIVATION [5]

Cell indicating depth of expression

```
#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;
}
```

# VISUALIZING ACTIVATION [5]

## Cell active inside comments

```
/* Duplicate LSM field information.  The lsm_rule is opaque, so
 * re-initialized. */
static inline int audit_dupe_lsm_field(struct audit_field *df,
                                       struct audit_field *sf)
{
    int ret = 0;
    char *lsm_str;
    /* our own copy of lsm_str */
    lsm_str = kstrdup(sf->lsm_str, GFP_KERNEL);
    if (unlikely(!lsm_str))
        return -ENOMEM;
    df->lsm_str = lsm_str;
    /* our own (refreshed) copy of lsm_rule */
    ret = security_audit_rule_init(df->type, df->op, df->lsm_str,
                                    (void **)&df->lsm_rule);
    /* Keep currently invalid fields around in case they
     * become valid after a policy reload. */
    if (ret == -EINVAL) {
        pr_warn("audit rule for LSM \\'%s\\' is invalid\n",
               df->lsm_str);
        ret = 0;
    }
    return ret;
}
```

# VISUALIZING ACTIVATION [5]

Cell active 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;
}
```

# VISUALIZING ACTIVATION [5]

Cell activation that is hard to interpret

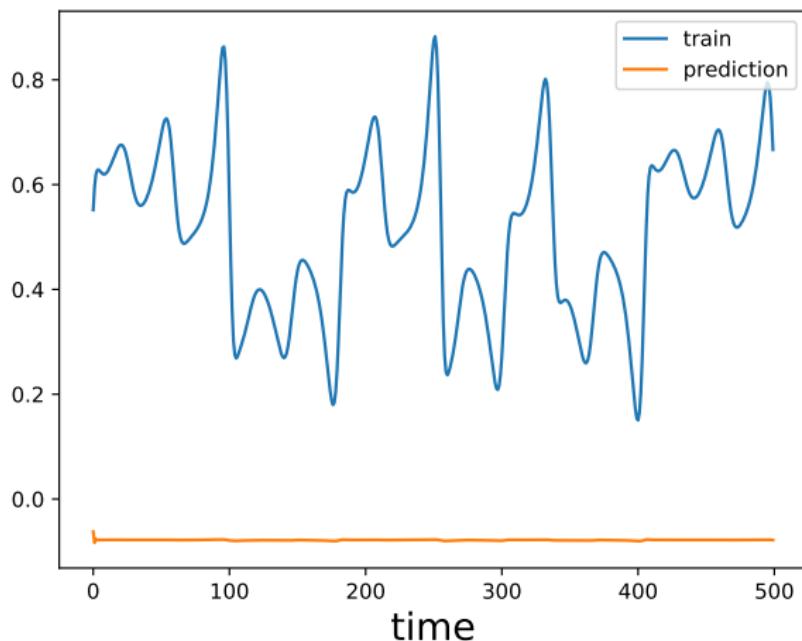
```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
    char *str;
    if (!*bufp || (len == 0) || (len > *remain))
        return ERR_PTR(-EINVAL);
    /* of the currently implemented string fields, PATH_MAX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

Most cells are activating in this manner!

# **SOLUTION EXERCISE 10**

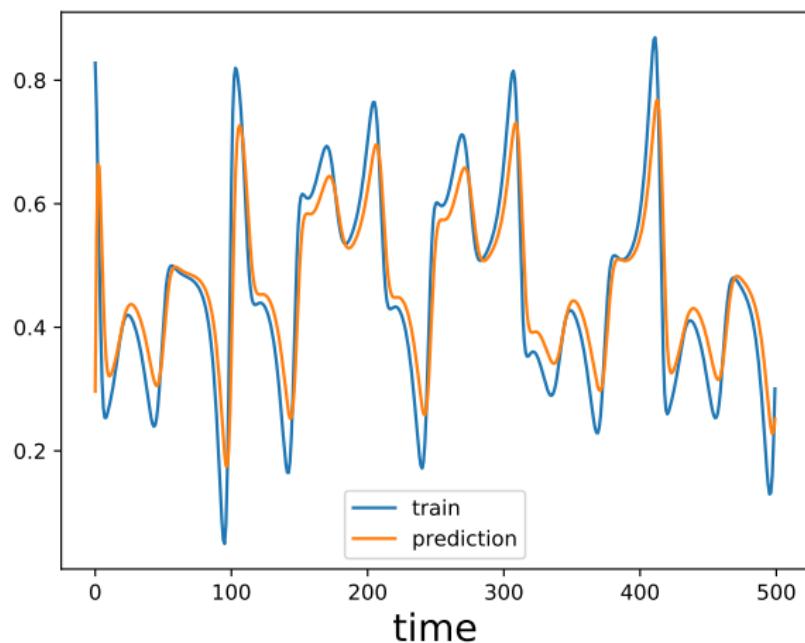
## 1.4 RNN TRAINING

RNN training: epoch 0



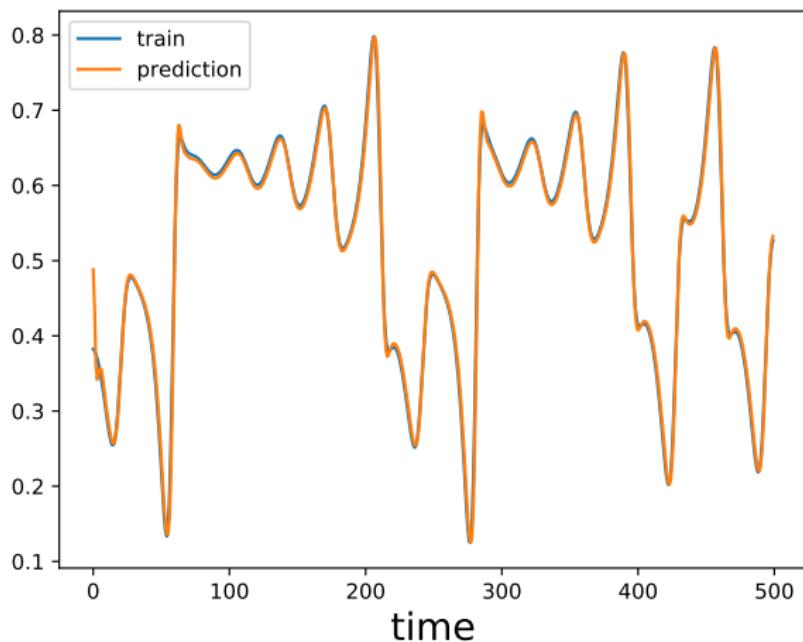
## 1.4 RNN TRAINING

RNN training: epoch 600



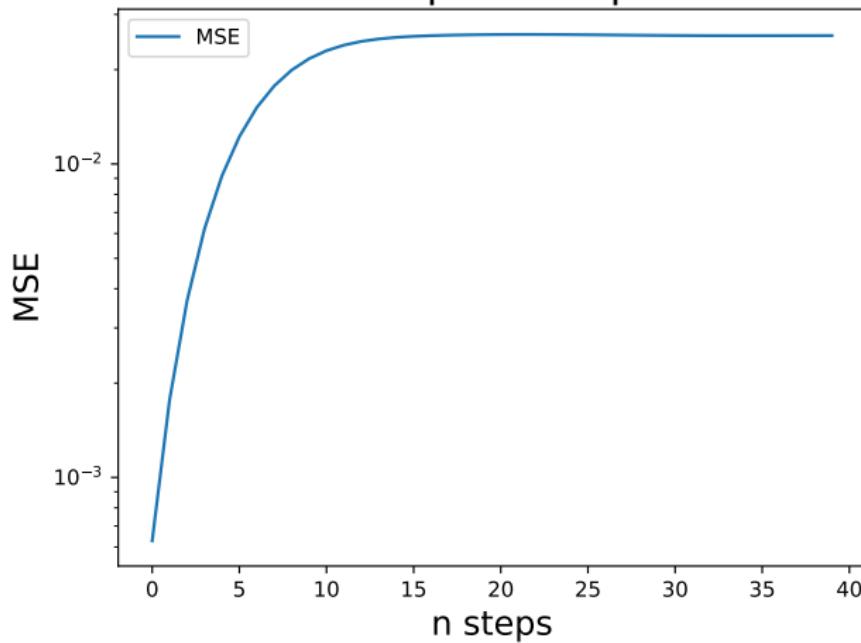
## 1.4 RNN TRAINING

RNN training: epoch 2400

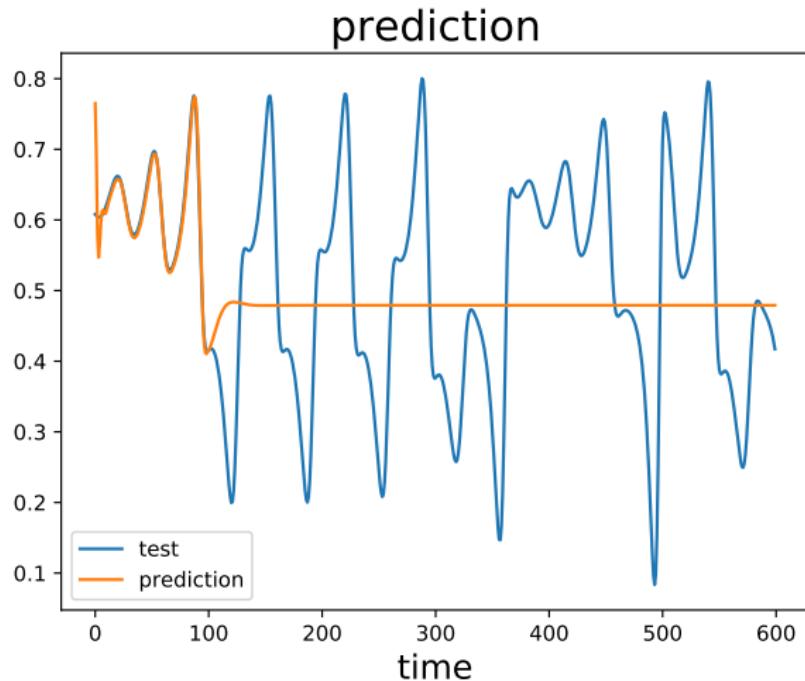


## 1.4 RNN MSE

MSE of n-step ahead prediction

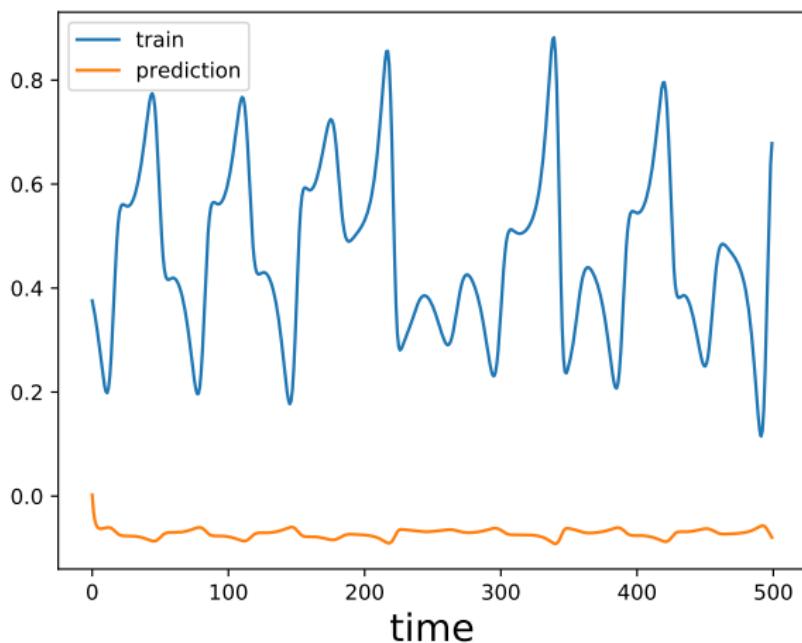


## 1.4 RNN PREDICTION



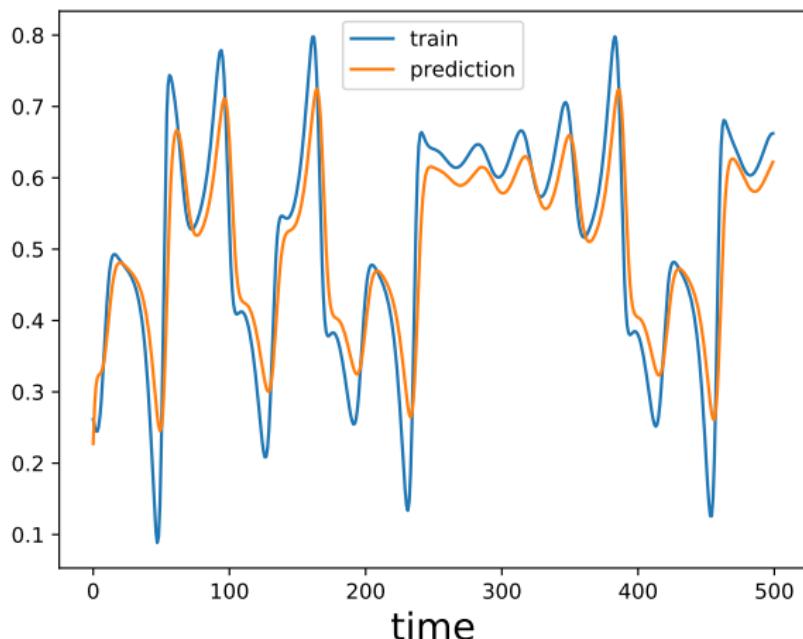
## 1.4 LSTM TRAINING

LSTM training: epoch 0



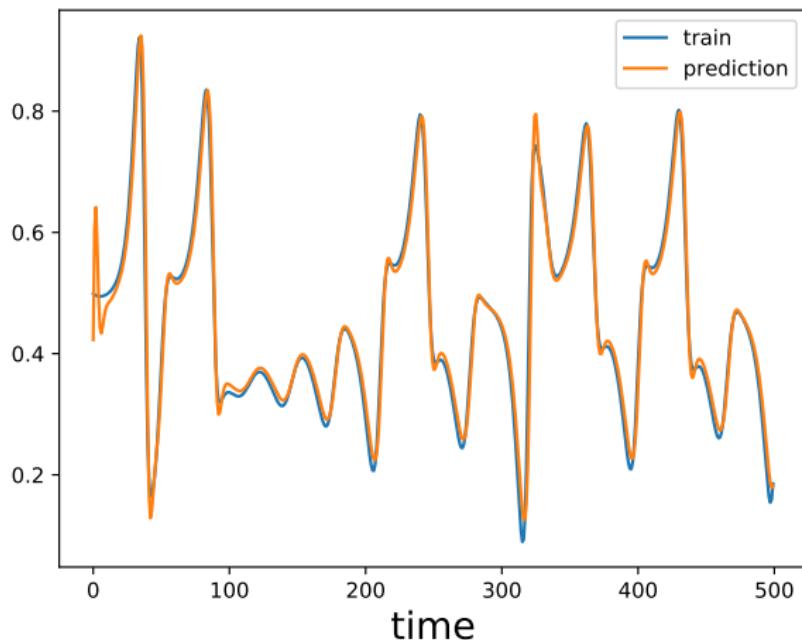
## 1.4 LSTM TRAINING

LSTM training: epoch 600

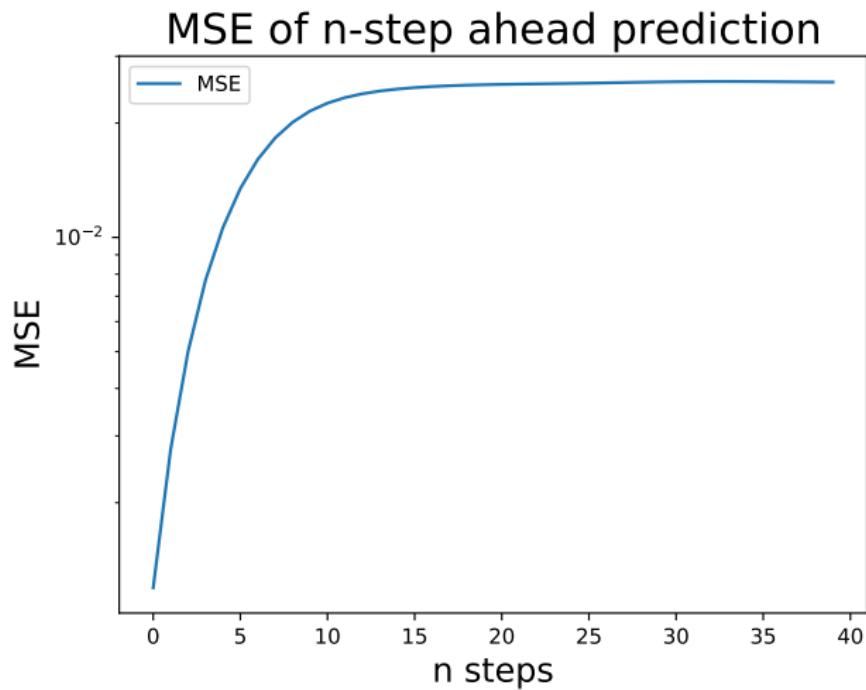


## 1.4 LSTM TRAINING

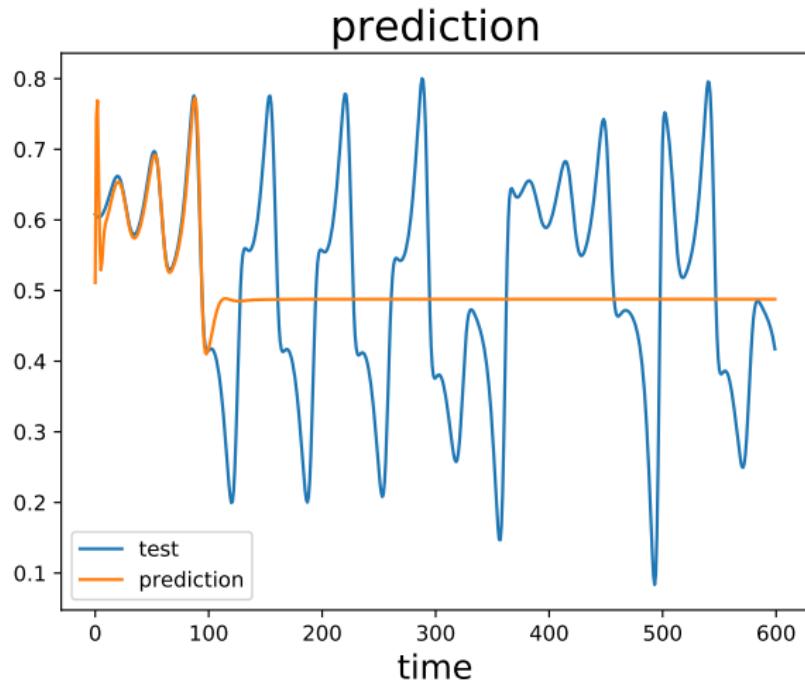
LSTM training: epoch 2400



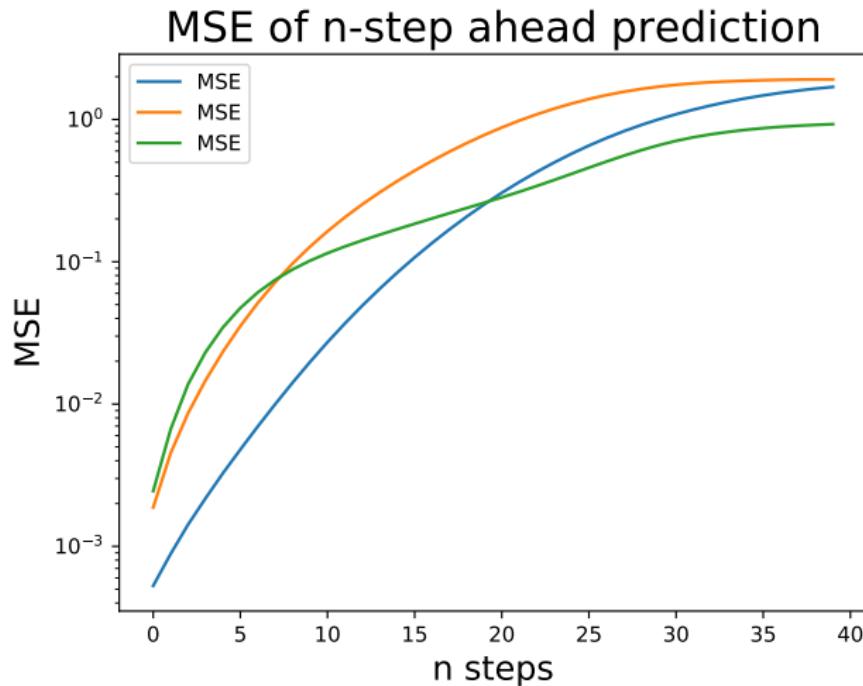
## 1.4 LSTM MSE



## 1.4 LSTM PREDICTION

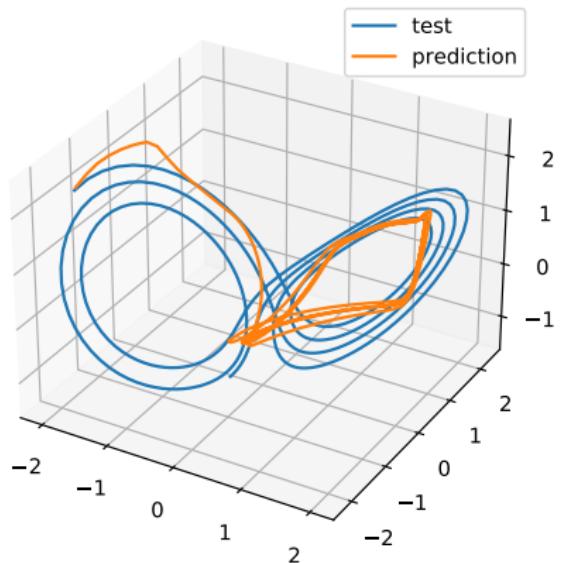


## 1.4 RNN3D MSE

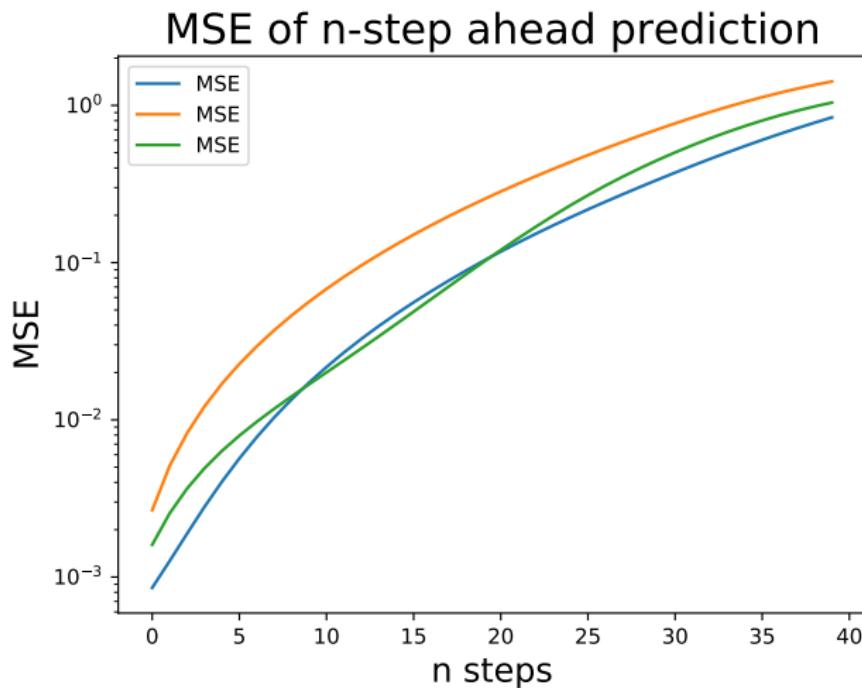


## 1.4 RNN3D PREDICTION

RNN3D prediction

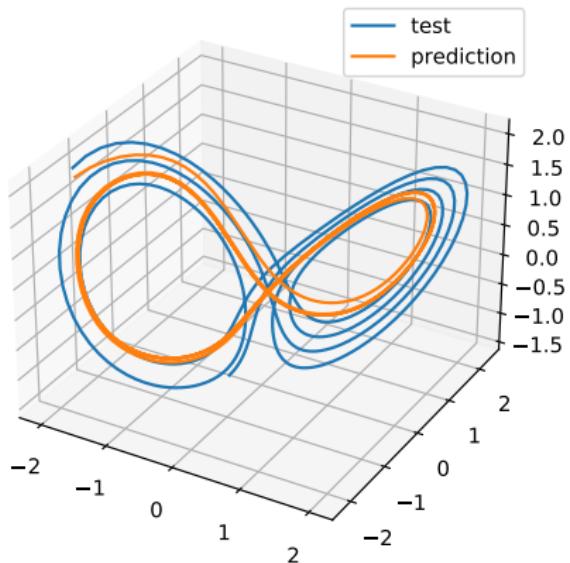


## 1.4 LSTM3D MSE



## 1.4 LSTM3D PREDICTION

LSTM3D prediction



# LECTURE REVIEW

# EXTENDED KALMAN FILTER

Assume:

$$z_1 \sim \mathcal{N}(\mu_0, \Sigma_0)$$

$$z_t = F_\theta(z_{t-1}) + \epsilon_t, \epsilon_t \sim \mathcal{N}(0, \Sigma)$$

$$x = Bz_t + \eta_t, \eta_t \sim \mathcal{N}(0, \Gamma)$$

Idea: linearize non-linear functions

$$F_\theta(z_{t-1}) \approx F_\theta(\mu_{t-1}) + F'_\theta(\mu_{t-1})(z_{t-1} - \mu_{t-1})$$

# EKF UPDATE

$$\mu_t = F(\mu_{t-1} + K_t[x_t - G(F(\mu_{t-1}))])$$

$$V_t = L_{t-1} - K_t \nabla_{t-1} L_{t-1}$$

$$L_{t-1} = J_{t-1} V_{t-1} J_{t-1}^T + \Sigma$$

$$K_t = L_{t-1} \nabla_{t-1}^T (\nabla_{t-1} L_{t-1} \nabla_{t-1}^T + \Gamma)^{-1}$$

$$\nabla_{ij,t-1} := \frac{\partial G_i}{\partial F(\mu_{j,t-1})}, J_{ij,t-1} := \frac{\partial F_i}{\partial \mu_{j,t-1}}$$

## REFERENCES |

-  IAN GOODFELLOW, YOSHUA BENGIO, AND AARON COURVILLE.  
**DEEP LEARNING.**  
MIT Press, 2016.  
<http://www.deeplearningbook.org>.
-  RAZVAN PASCANU, TOMAS MIKOLOV, AND YOSHUA BENGIO.  
**ON THE DIFFICULTY OF TRAINING RECURRENT NEURAL NETWORKS.**  
*arXiv:1211.5063 [cs]*, February 2013.  
*arXiv: 1211.5063*.
-  KAIMING HE, XIANGYU ZHANG, SHAOQING REN, AND JIAN SUN.  
**DEEP RESIDUAL LEARNING FOR IMAGE RECOGNITION.**  
*arXiv:1512.03385 [cs]*, December 2015.  
*arXiv: 1512.03385*.
-  S. HOCHREITER AND J. SCHMIDHUBER.  
**LONG SHORT-TERM MEMORY.**  
*Neural Computation*, 9:1735–1780, 1997.

## REFERENCES II

- 
- ANDREJ KARPATHY, JUSTIN JOHNSON, AND LI FEI-FEI.  
**VISUALIZING AND UNDERSTANDING RECURRENT NETWORKS.**  
*arXiv:1506.02078 [cs]*, November 2015.  
arXiv: 1506.02078.