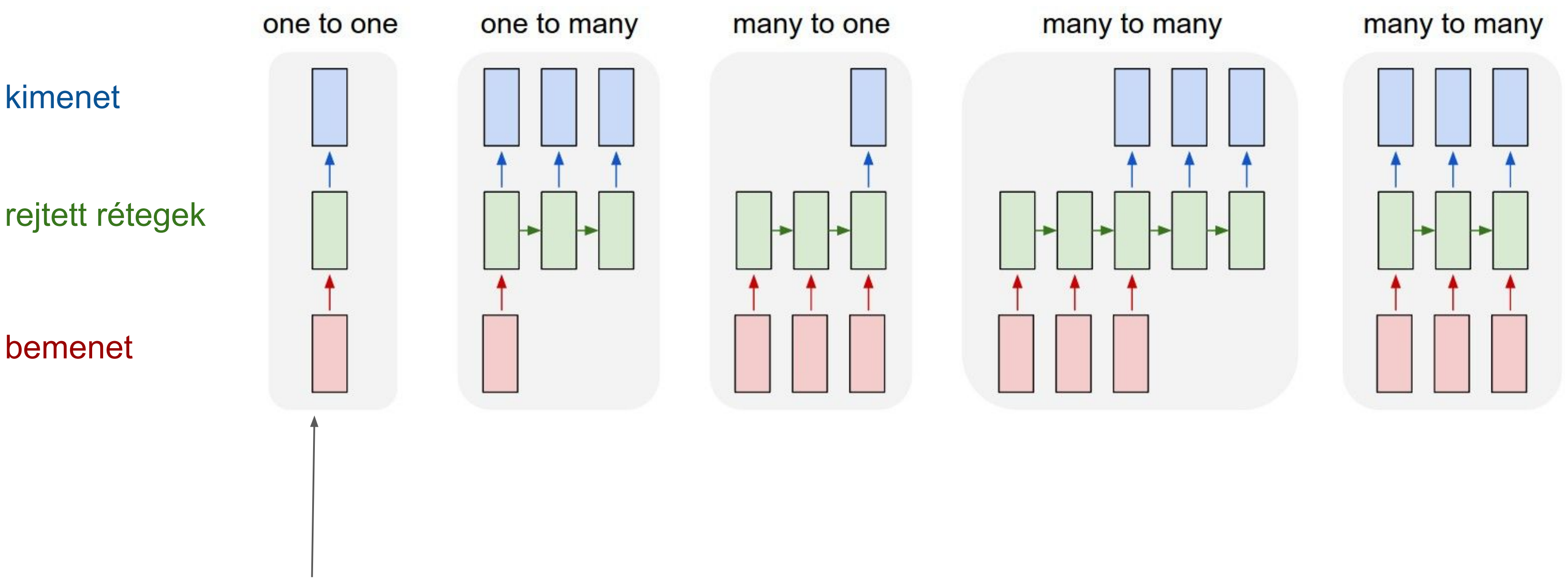
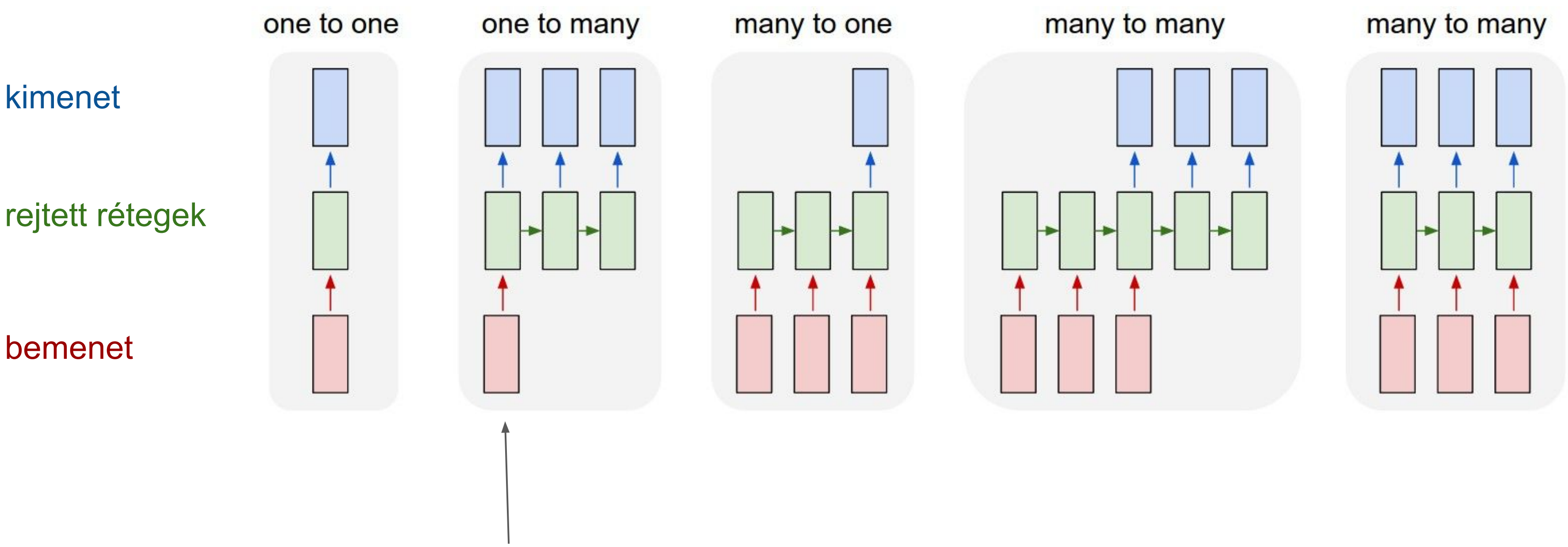
Visszacsatolt (mély) neurális hálózatok

Visszacsatolt hálózatok



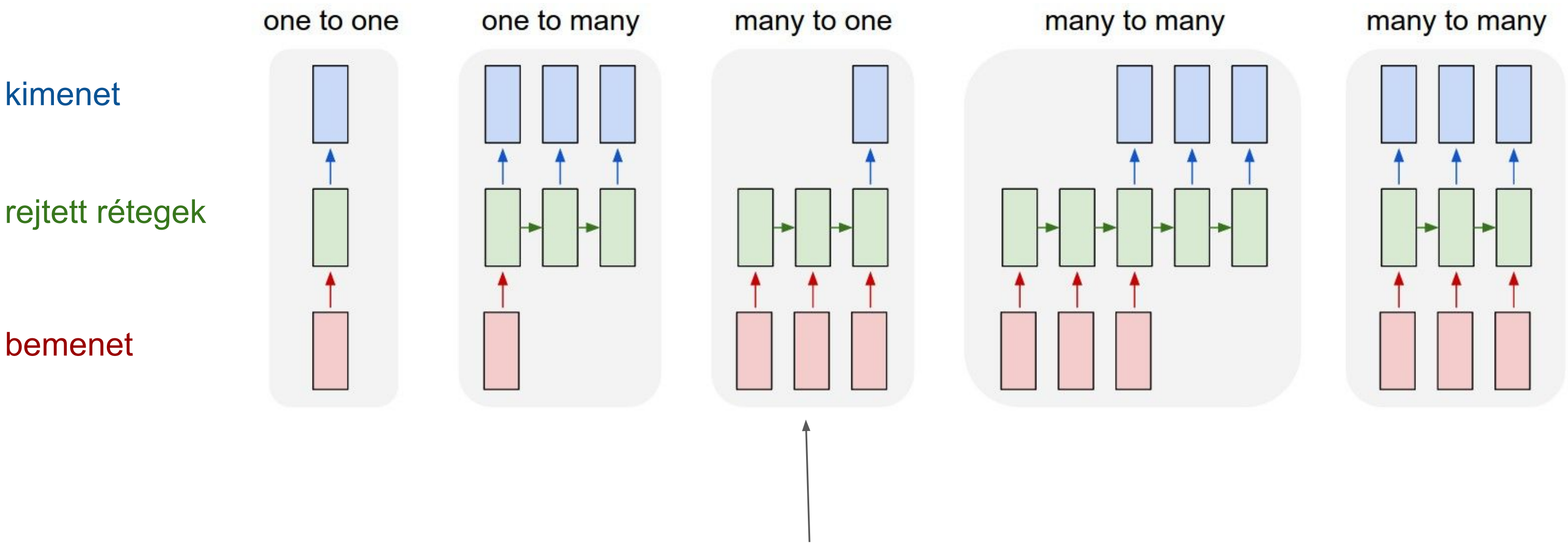
Sima előrecsatolt neurális hálózat

Visszacsatolt hálózatok



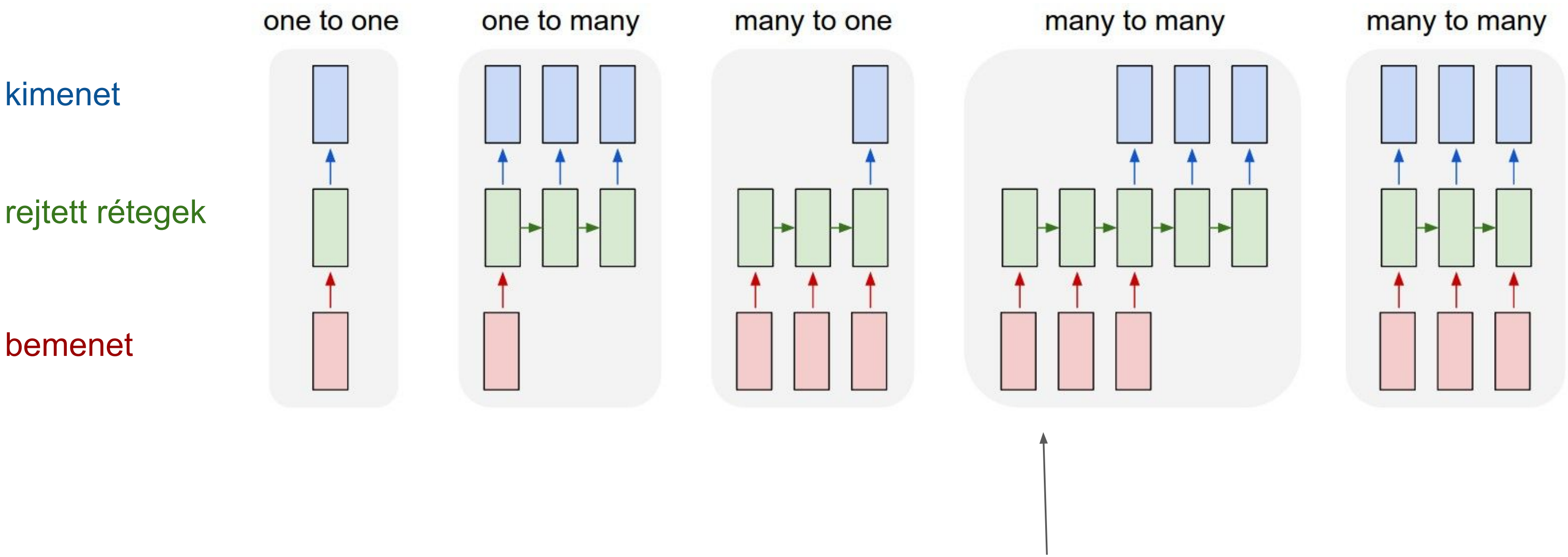
Pl.: kép feliratozás,
képből szavak sorozata

Visszacsatolt hálózatok



Pl.: dokumentum osztályozás,
szavak sorozatából címke

Visszacsatolt hálózatok



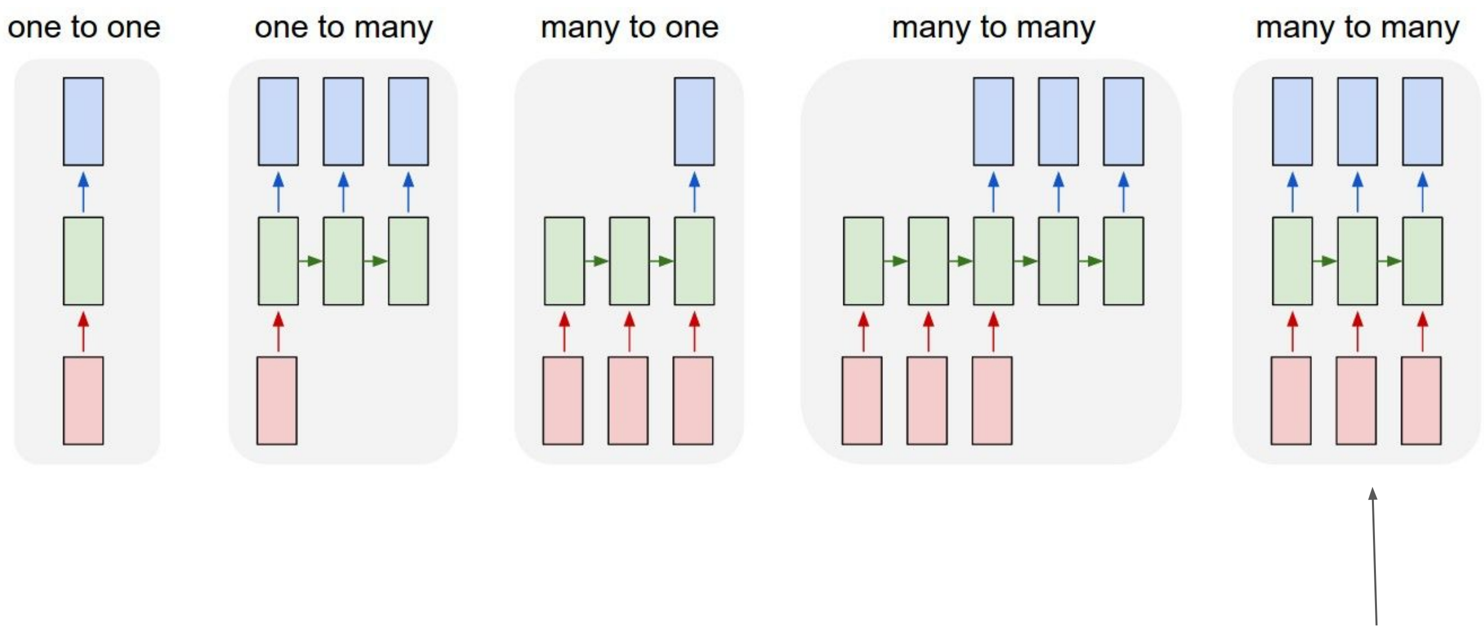
Pl.: gépi fordítás,
Magyar szavak sorozatából, angol szavak sorozata

Visszacsatolt hálózatok

kimenet

rejtett rétegek

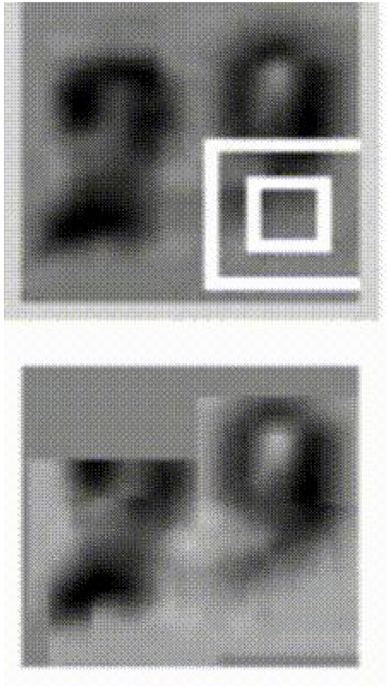
bemenet



Pl.: képkockánkénti video osztályozás,
Képkockák sorozatából, címkék sorozata

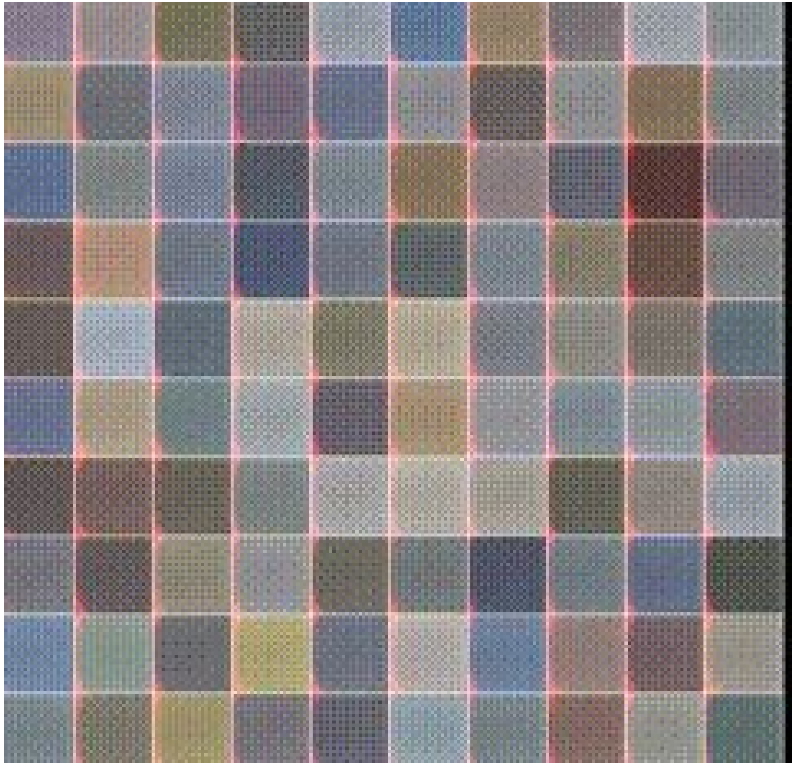
Statikus bemenet sorrendi feldolgozása

Multiple Object Recognition with Visual Attention, Ba et al. ,2014 (DeepMind)

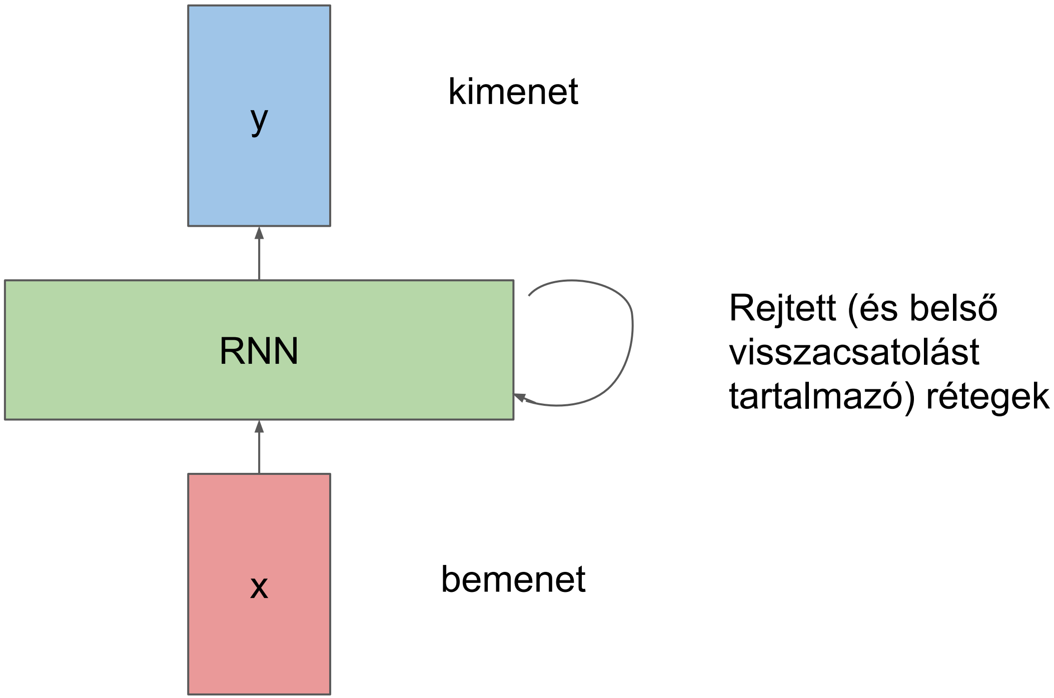


Statikus bemenet sorrendi feldolgozása

DRAW: A Recurrent Neural Network For Image Generation, Gregor et al., 2015
(DeepMind)



Visszacsatolt neurális hálózat

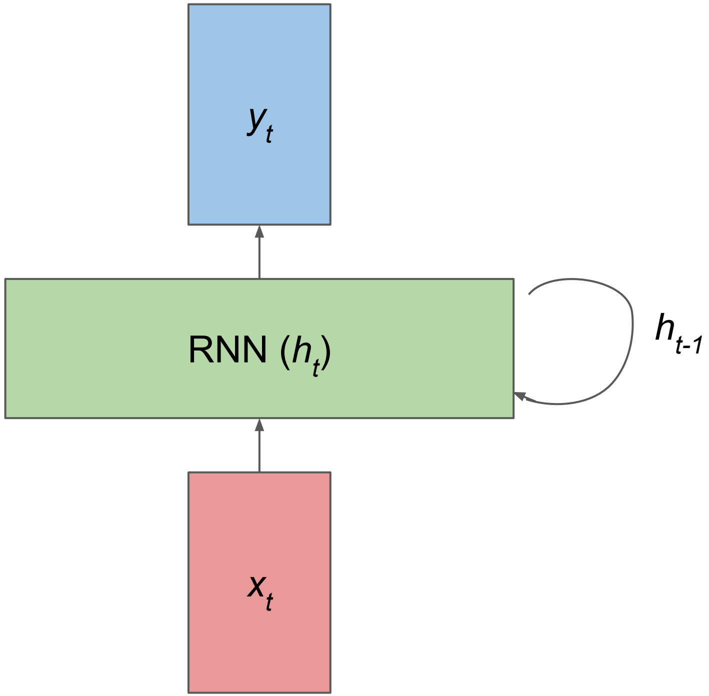


Visszacsatolt neurális hálózat

Bemeneti **x** vektorok sorozatát a következőképp tudjuk feldolgozni:

$$h_t = f_W(h_{t-1}, x_t)$$

Ugyanazt a függvényt, ugyanazzal a W paramétervektorral alkalmazzuk, az **x** sorozat minden $\mathbf{x}_t$ elemére



Visszacsatolt neurális hálózat - klasszikusan

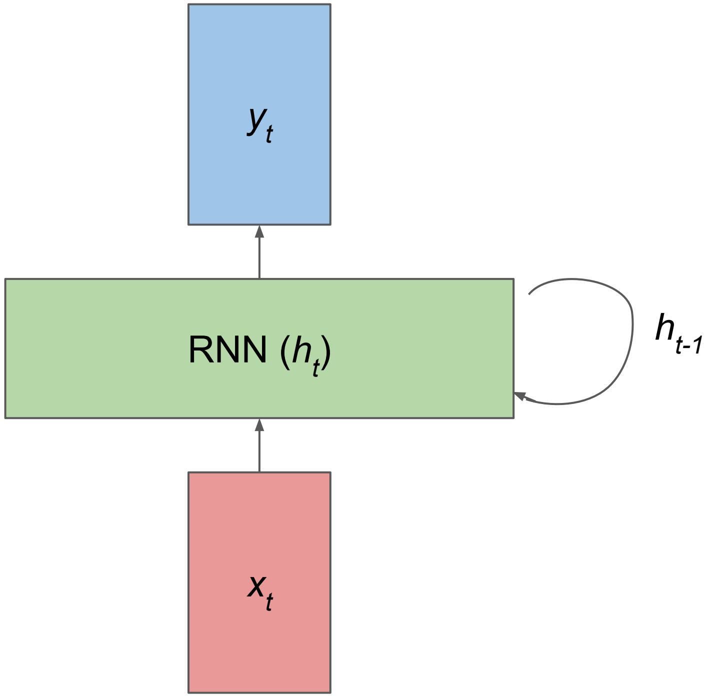
Bemeneti **x** vektorok sorozatát a következőképp tudjuk feldolgozni:

$$h_t = f_W(h_{t-1}, x_t)$$

↓

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$



Visszacsatolt neurális hálózat - példa

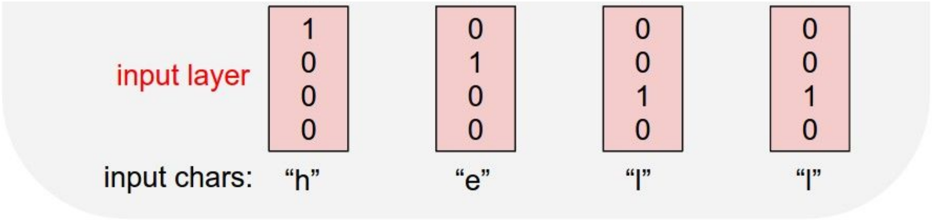
Karakter alapú nyelvi modell (char-rnn)

Lehetséges bemenetek:
[h, e, l, o]

Bemeneti tanító adatsor:
“hello”

Bemenet kódolása:
1-az-n-ből (one-hot)

Kívánt kimenet:
következő karakter



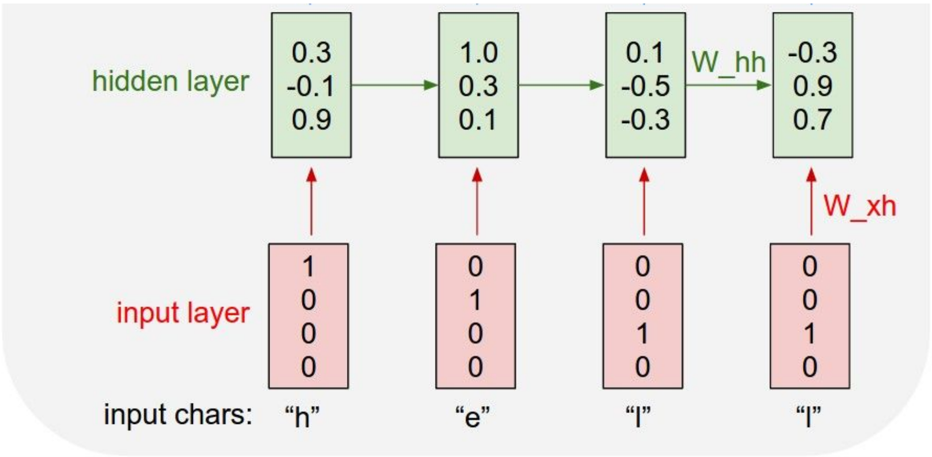
Visszacsatolt neurális hálózat - példa

Karakter alapú nyelvi modell

Lehetséges bemenetek:
[h, e, l, o]

Bemeneti tanító adatsor:
“hello”

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



Visszacsatolt neurális hálózat - példa

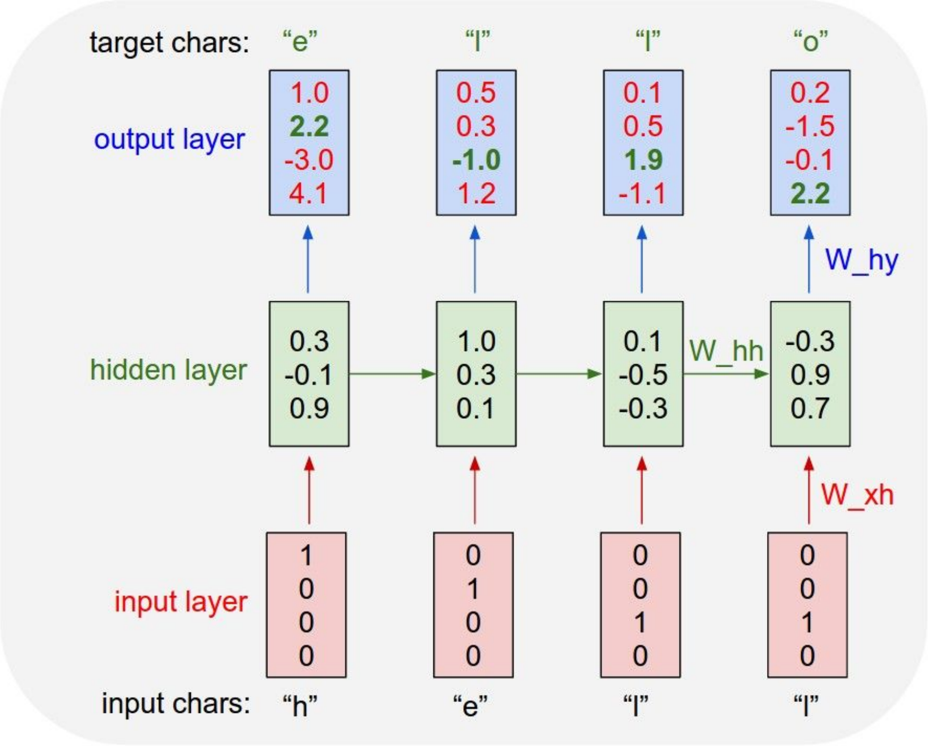
Karakter alapú nyelvi modell

Lehetséges bemenetek:
[h, e, l, o]

Bemeneti tanító adatsor:
“hello”

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

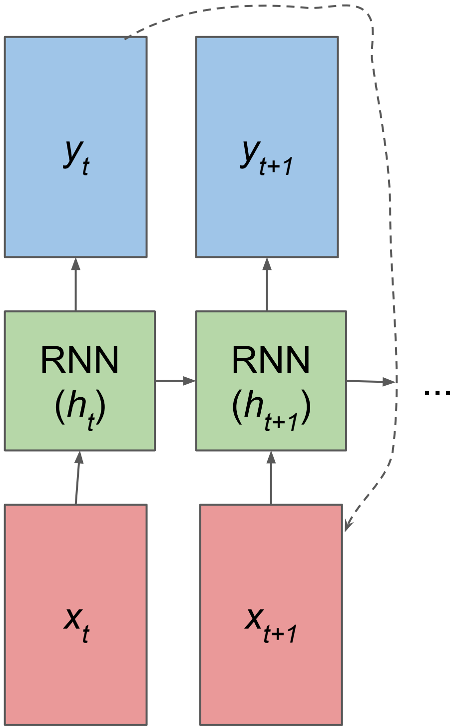
$$y_t = W_{hy}h_t$$



Szöveg generálása visszacsatolt neurális hálózattal

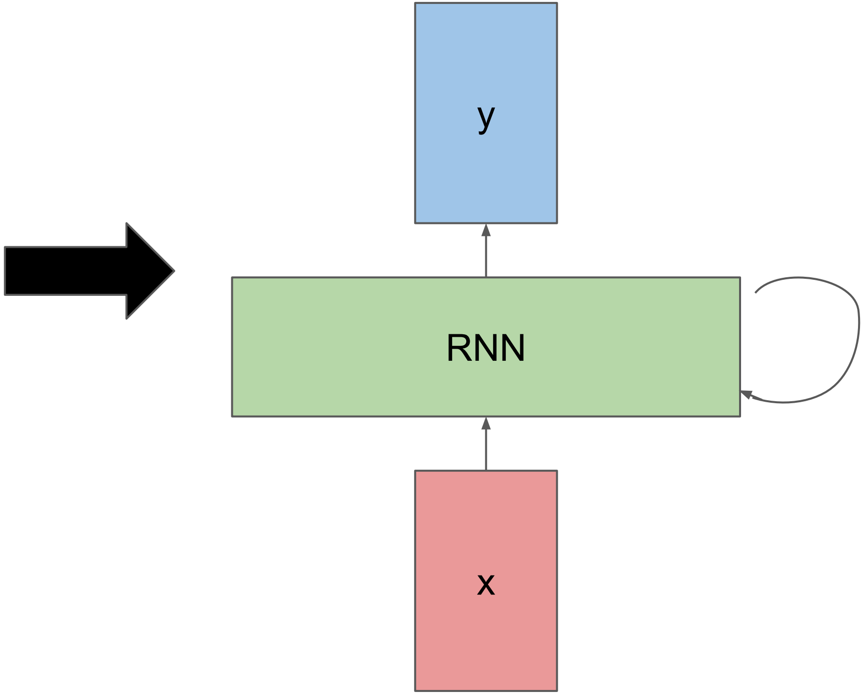
Van egy modellünk, ami egy adott karaktersorozatra megjósolja a következő karakter valószínűségi eloszlását.

Sorsoljunk egy karaktert ebből az eloszlásból, és használjuk fel a sorozat következő elemének.



https://github.com/karpathy/char-rnn
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

A tanítóadatoknak csak a fantázia szab határt



Shakespeare

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

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.

VIOLA:
I'll drink it.
```

Wikipedia

```
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]]
(PJS)[http://www.humah.yahoo.com/guardian.
cfm/7754800786d17551963s89.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.]]
```

Algebrai geometria tankönyv

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 $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

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

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

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ 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

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

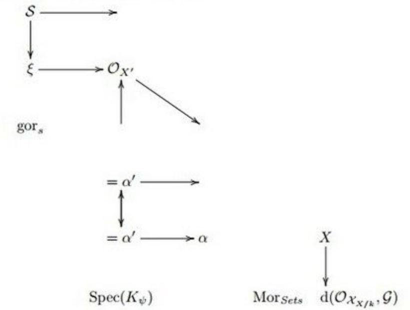
be a morphism of algebraic spaces over S and Y.

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 $f_\star$. 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 $X = \mathrm{Spec}(R)$ 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

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \ \ \text{-}1(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. 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 $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Linux forráskód

```c
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static 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;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
  return segtable;
}
```

```c
/*
 *  Copyright (c) 2006-2010, Intel Mobile Communications.  All rights reserved.
 *
 *   This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License version 2 as published by
 * the Free Software Foundation.
 *
 *        This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *
 *  GNU General Public License for more details.
 *
 *   You should have received a copy of the GNU General Public License
 *    along with this program; if not, write to the Free Software Foundation,
 *  Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */

#include <linux/kexec.h>
#include <linux/errno.h>
#include <linux/io.h>
#include <linux/platform_device.h>
#include <linux/multi.h>
#include <linux/ckevent.h>

#include <asm/io.h>
#include <asm/prom.h>
#include <asm/e820.h>
#include <asm/system_info.h>
#include <asm/setew.h>
#include <asm/pgproto.h>

#define REG_PG    vesa_slot_addr_pack
#define PFM_NOCOMP  AFSR(0, load)
#define STACK_DDR(type)     (func)

#define SWAP_ALLOCATE(nr)     (e)
```

A tanítás közben

```
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
```

```
"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."
```

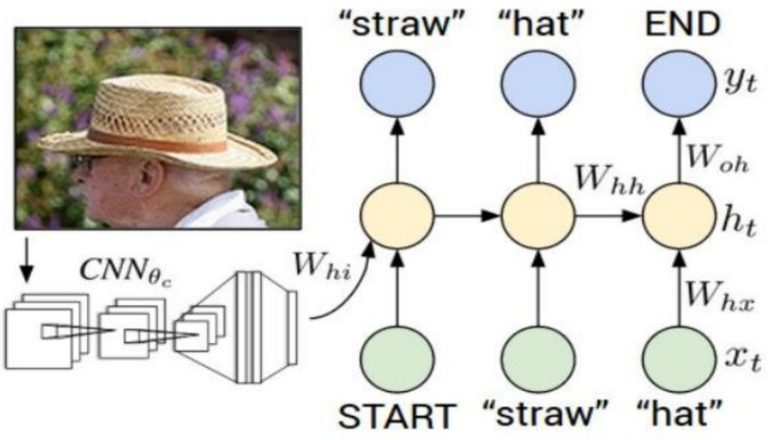
```
we counter. He stutn co des. His stanted out one ofler that concossions and was
to gearang reay Jotrets and with fre colt otf paitt thin wall. Which das stimn
```

```
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.
```

```
"Kite vouch!" he repeated by her
door. "But I would be done and quarts, feeling, then, son is people...."
```

```
"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.
```

Kép feliratozás

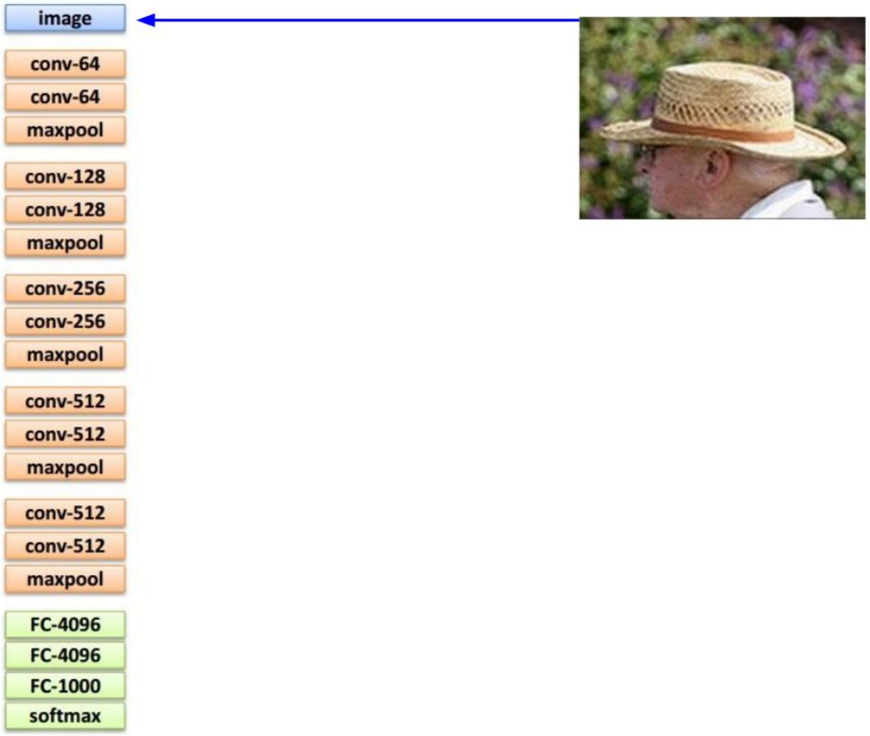


Microsoft COCO
[Tsung-Yi Lin et al. 2014]
mscoco.org

jelenleg:
~120K kép
~5 mondat mindegyikhez

Explain Images with Multimodal Recurrent Neural Networks, Mao et al.
Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
Show and Tell: A Neural Image Caption Generator, Vinyals et al.
Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

Kép feliratozás

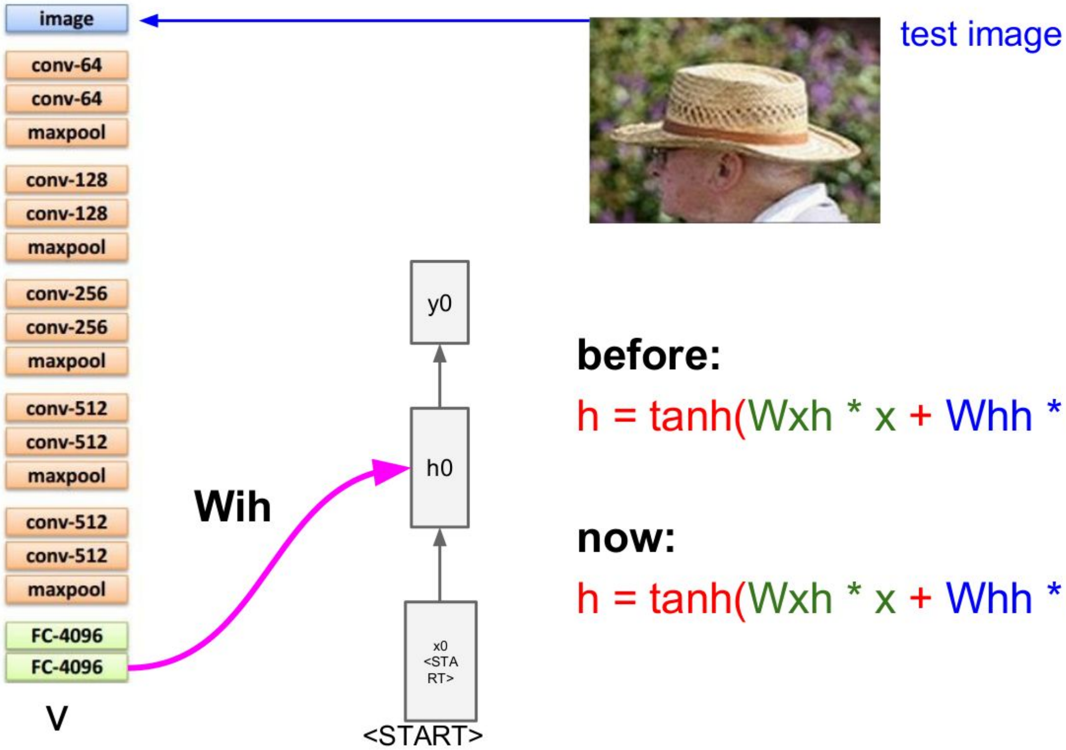


test image

Kép feliratozás

Kép feliratozás

Kép feliratozás



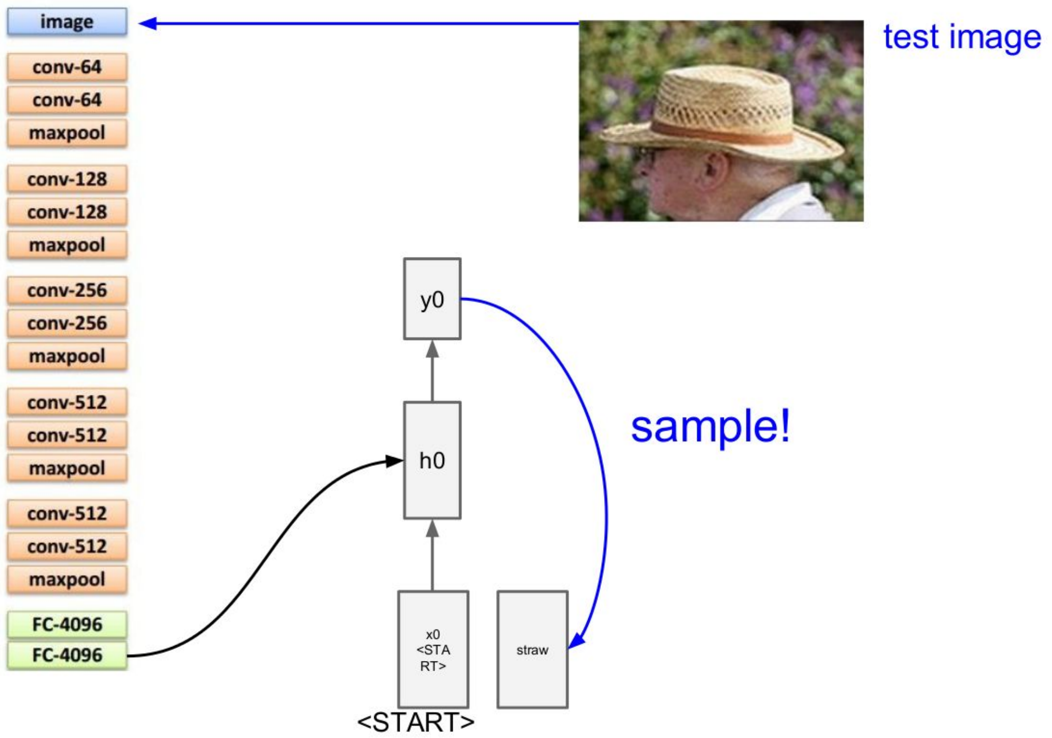
before:

h = tanh(Wxh * x + Whh * h)

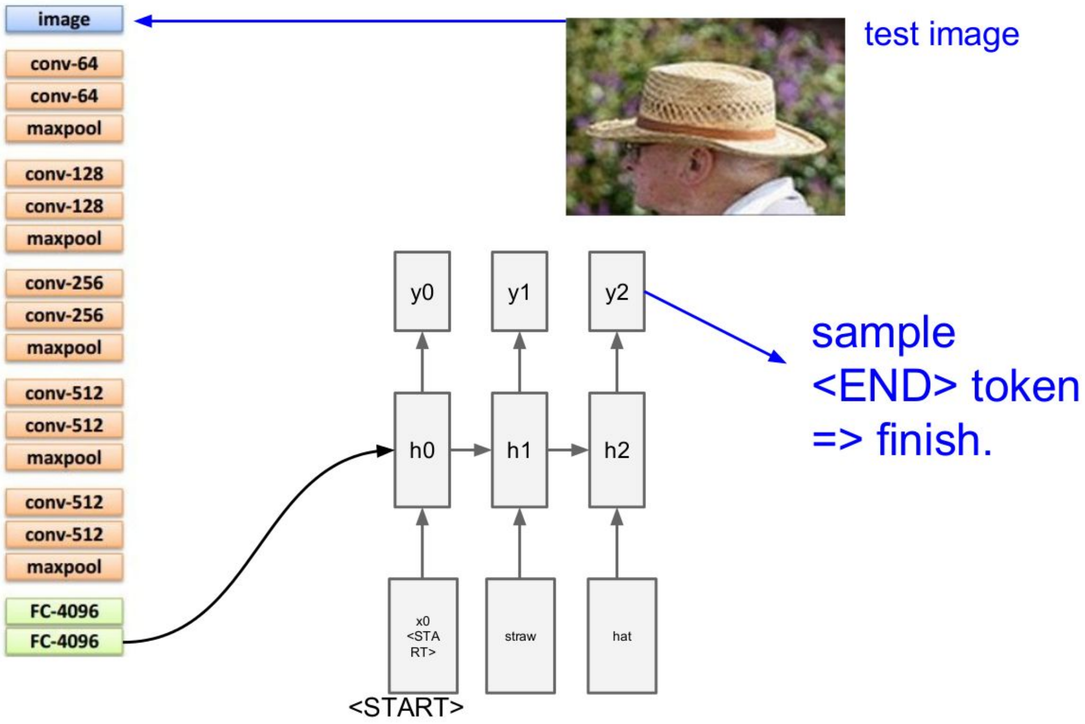
now:

h = tanh(Wxh * x + Whh * h + **Wih * v**)

Kép feliratozás



Kép feliratozás



Kép feliratozás

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

Kép feliratozás

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

"a young boy is holding a baseball bat."

"a cat is sitting on a couch with a remote control."

"a woman holding a teddy bear in front of a mirror."

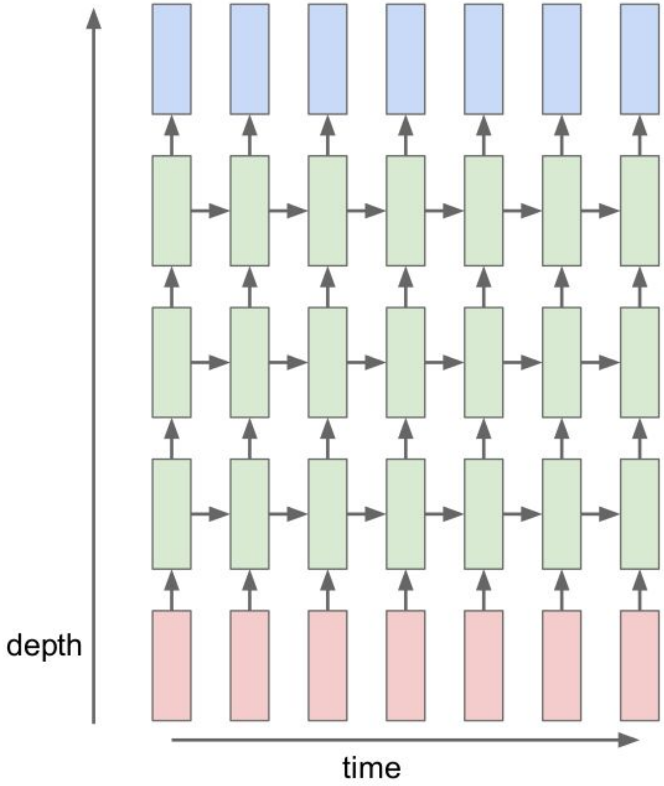
"a horse is standing in the middle of a road."

Komplexitás növelése - több réteg

RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$h \in \mathbb{R}^n$ $\quad W^l \ [n \times 2n]$

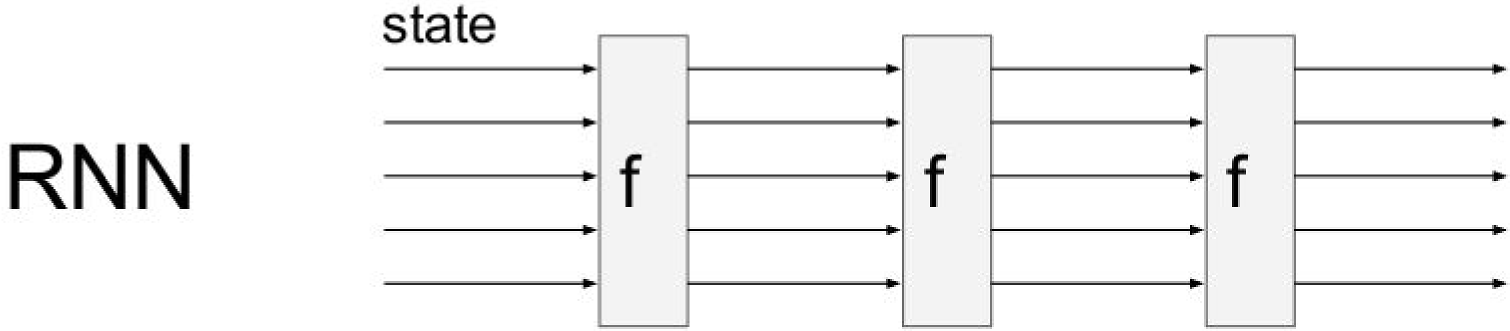


Exploding/vanishing gradients probléma

http://i.imgur.com/vaNahKE.gifv

RNN vs LSTM gradients on the input weight matrix
Error is generated at 128th step and propagated back. No error from other steps.

At the beginning of training. Weights sampled from Normal Distribution in (-0.1, 0.1).

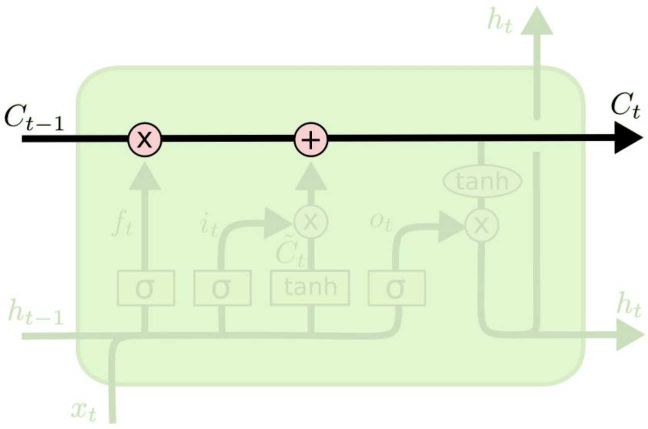


Exploding/vanishing gradients probléma

- A súlymátrixtól függően a hosszú időbeli kiterítés során a gradiens vagy exponenciálisan csökken, vagy exponenciálisan nő
- Növekedés esetén: gradient clipping
- Csökkenés esetén: nincs igazán jó egyszerű megoldás -> új módszerre van szükség: LSTM

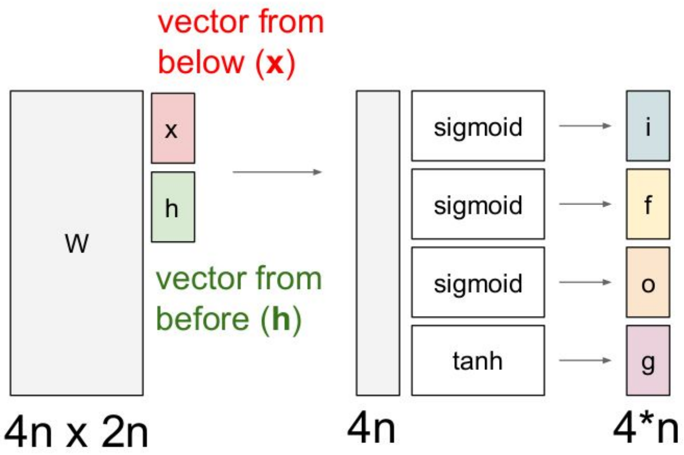
Exploding/vanishing gradients probléma

- A súlymátrixtól függően a hosszú időbeli kiterítés során a gradiens vagy exponenciálisan csökken, vagy exponenciálisan nő
- Növekedés esetén: gradient clipping
- Csökkenés esetén: nincs igazán jó egyszerű megoldás -> új módszerre van szükség: LSTM



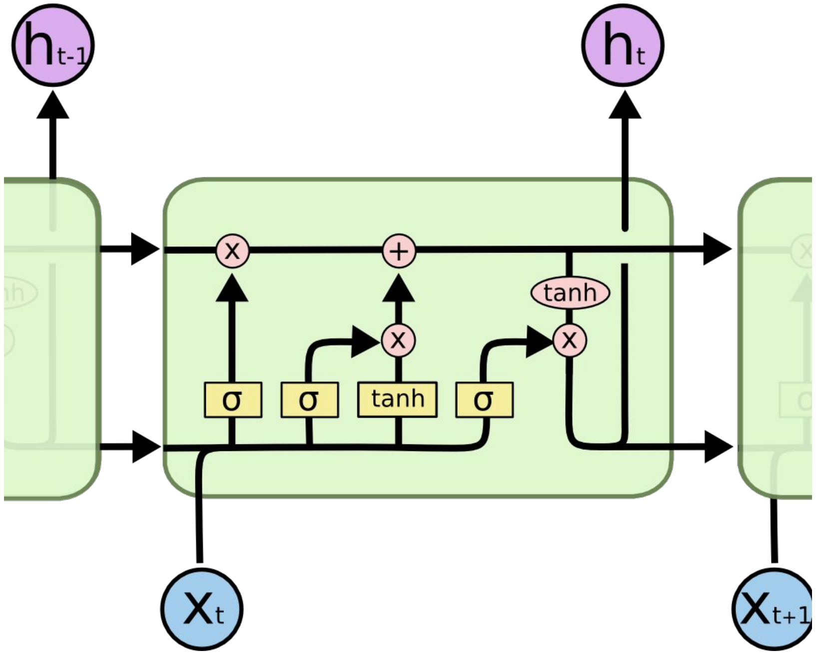
LSTM - Long Short-Term Memory

[Hochreiter, Schmidhuber:"Long short-term memory", 1997]



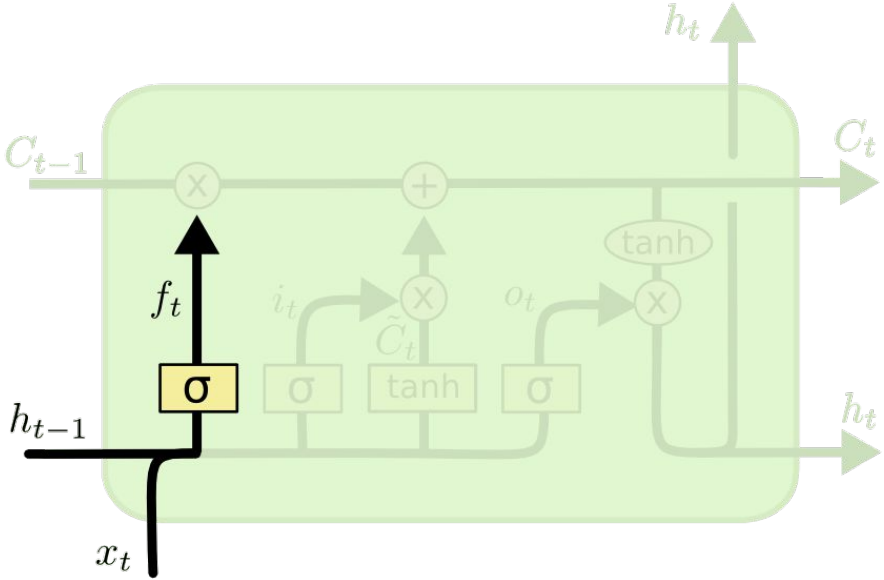
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$



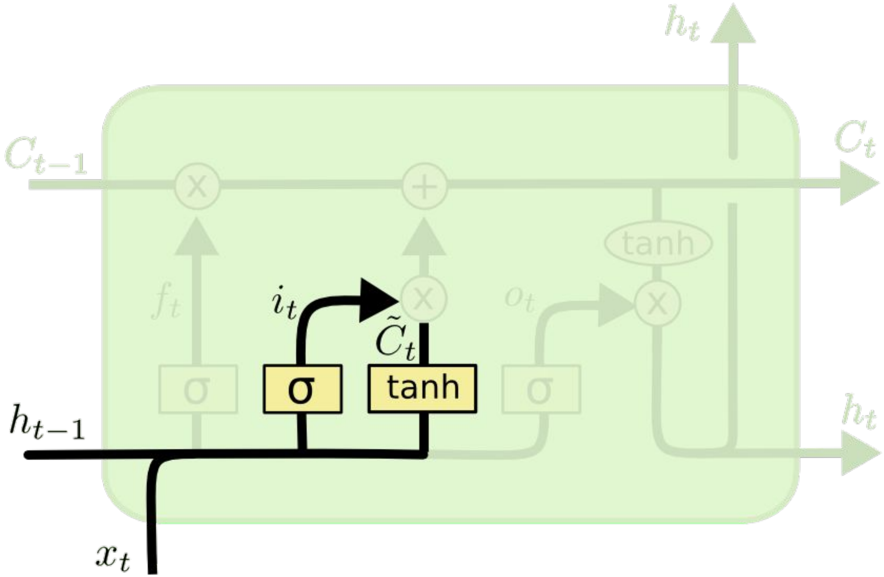
További infó: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM - Long Short-Term Memory



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

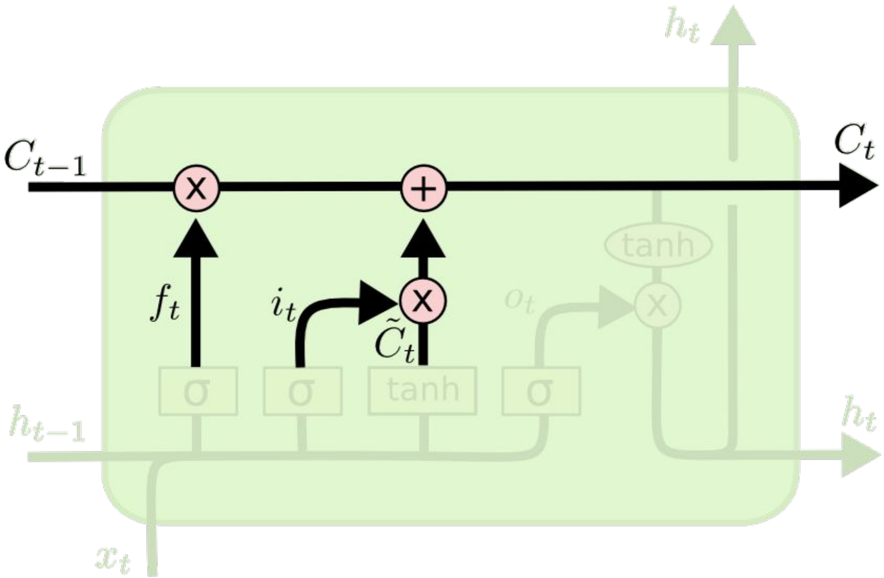
LSTM - Long Short-Term Memory



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

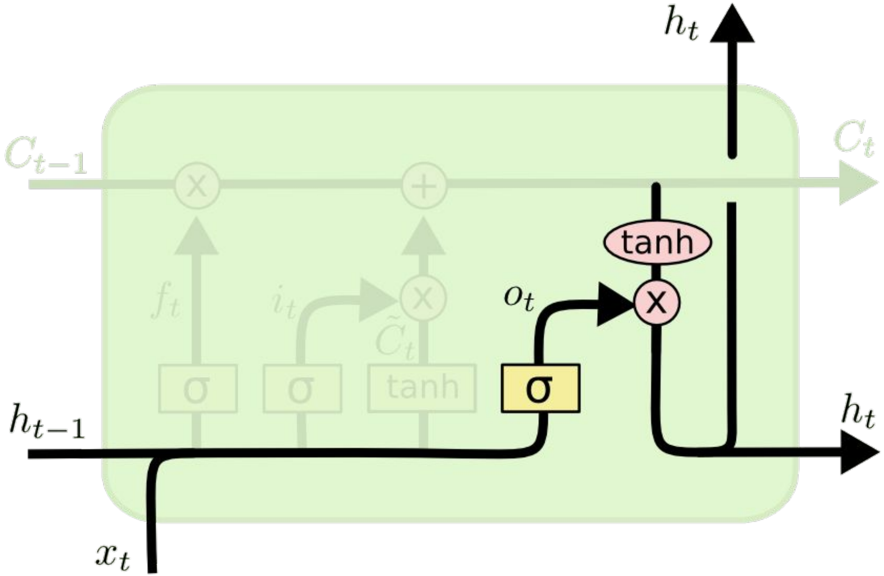
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

LSTM - Long Short-Term Memory



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

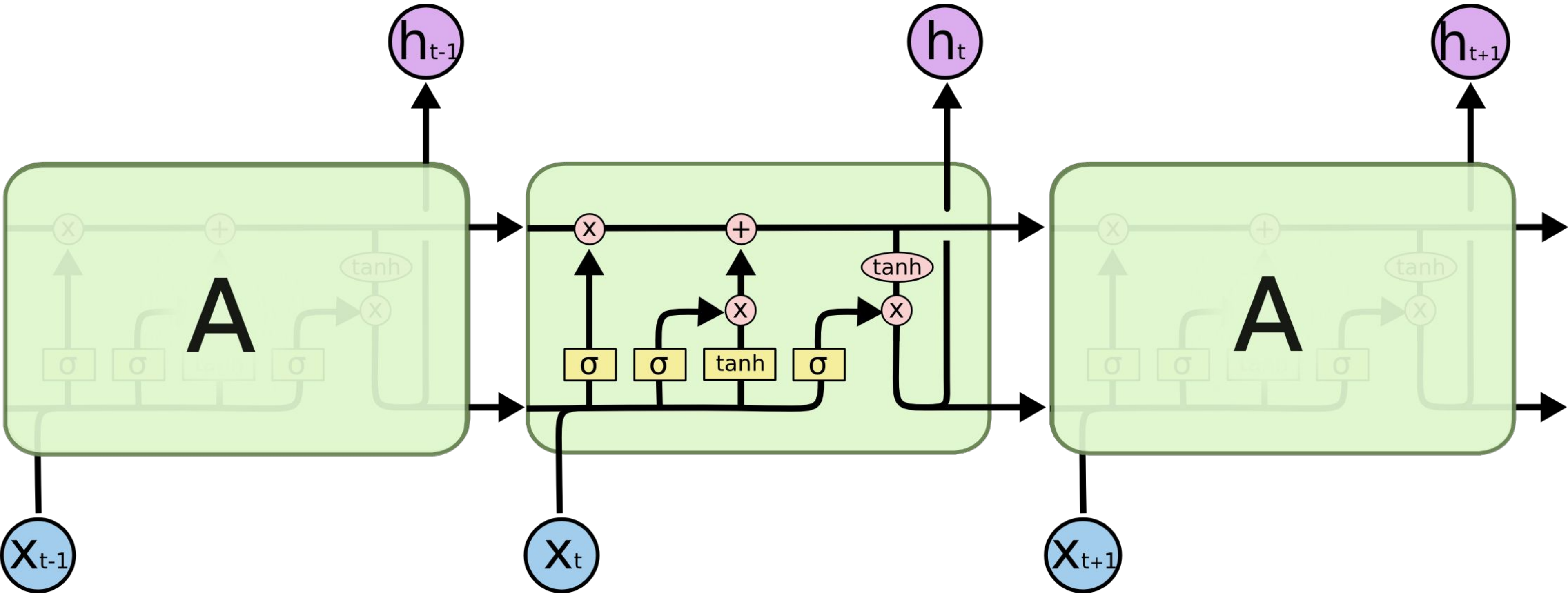
LSTM - Long Short-Term Memory



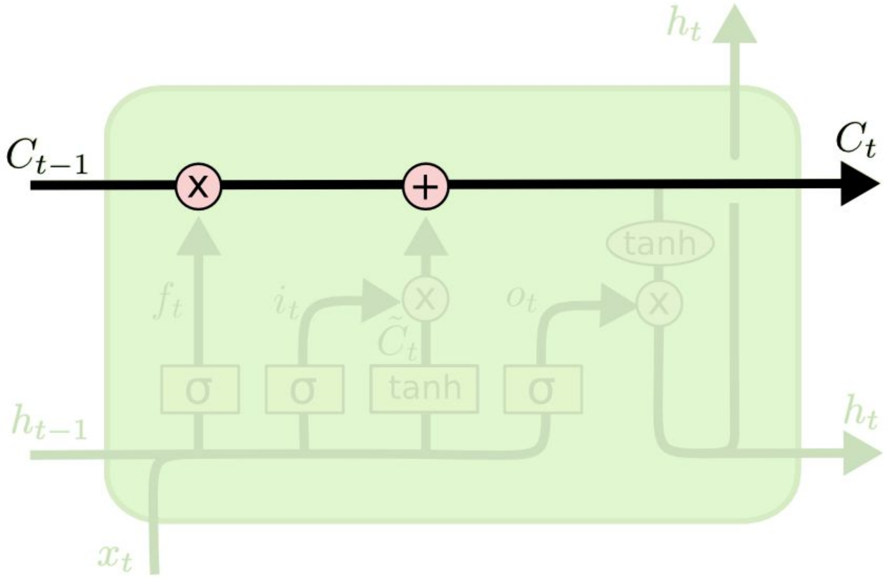
$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

LSTM - Long Short-Term Memory



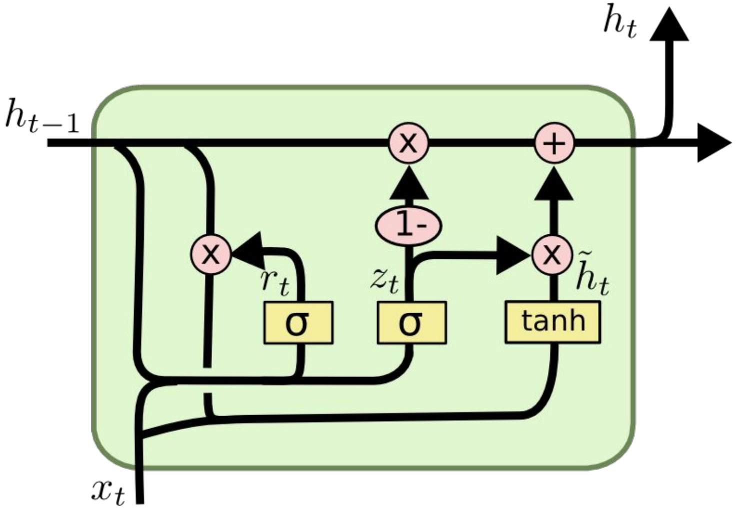
LSTM - Long Short-Term Memory



A ResNet-hez hasonlóan, a visszaterjesztett gradienst csak additívan móosítjuk (felejtő kaput leszámítva)

GRU - Gated Recurrent Unit

[Learning phrase representations using rnn encoder-decoder for statistical machine translation, Cho et al. 2014]



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

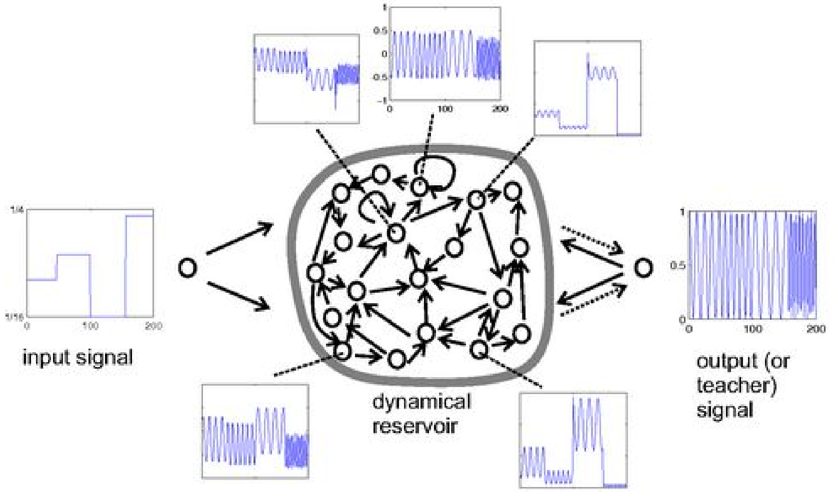
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

- Hasonlóan jó, mint az LSTM
- Nincs külön cella állapot, csak a kimenet

ESN - Echo state network

[Herbert Jaeger "Echo State Network" 2007]

- 1. Random RNN
    - Szivárgó (leaky) integrátor
    - minden mindennel összekötve
    - Kimeneti neuronok
- 2. Harvest reservoir states
    - Tanítóadatokkal szimuláljuk a hálózatot
    - A kimeneti idősort tároljuk el
- 3. Compute output weights
    - Lineáris regresszióval
    - Az eltárolt kimeneti idősor és a kívánt kimenetek között
- (Extreme learning machine (ELM))



ESN - Echo state network

[Herbert Jaeger “Echo State Network” 2007]

Rendszeregyenletek:

$\mathbf{x}(n)$ is the N-dimensional reservoir state

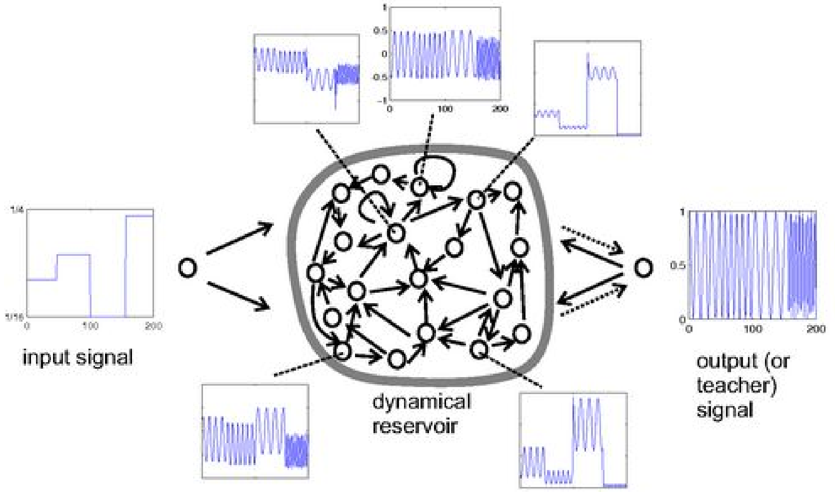
$\mathbf{u}(n)$ is the Kdimensional input signal

$\mathbf{y}(n)$ is the L-dimensional output signal

$$\mathbf{z}(n) = [\mathbf{x}(n); \mathbf{u}(n)]$$

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}^{fb}\mathbf{y}(n))$$

$$\mathbf{y}(n) = g(\mathbf{W}^{out}\mathbf{z}(n))$$



ESN - Echo state network

[Herbert Jaeger “Echo State Network” 2007]

Rendszeregyenletek:

$\mathbf{x}(n)$ is the N-dimensional reservoir state

$\mathbf{u}(n)$ is the K dimensional input signal

$\mathbf{y}(n)$ is the L-dimensional output signal

Bemeneti sorozatból rendszerállapot sorozat:

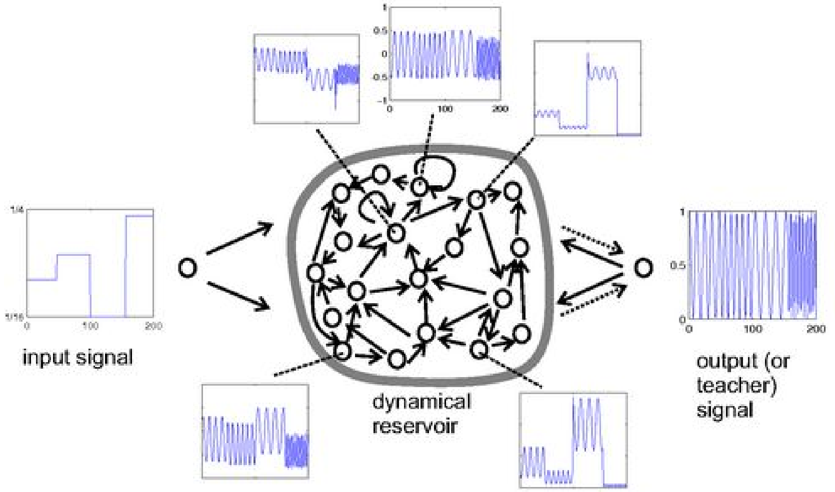
$$\mathbf{u}(1), \ldots, \mathbf{u}(n_{max})$$

$$\downarrow$$

$$\mathbf{z}(1), \ldots, \mathbf{z}(n_{max})$$

$$\downarrow$$

$\mathbf{S}$ of size $n_{max} \times (N + K)$



ESN - Echo state network

[Herbert Jaeger “Echo State Network” 2007]

Tanítás:

$\mathbf{R} = \mathbf{S}'\mathbf{S}$ be the correlation matrix of the extended reservoir states
$\mathbf{P} = \mathbf{S}'\mathbf{D}$ be the cross-correlation matrix of the states vs. the desired outputs.

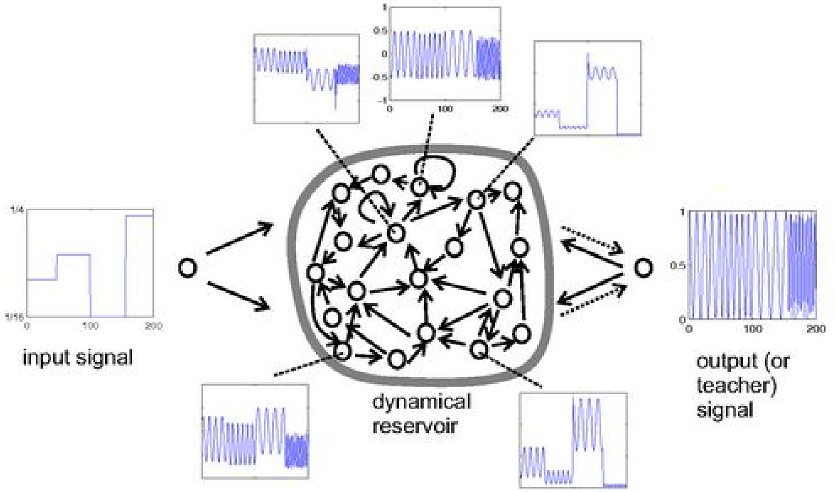
$$\mathbf{W}^{out} = (\mathbf{R}^{-1}\mathbf{P})'$$ Wiener-Hopf

vagy

$$\mathbf{W}^{out} = (\mathbf{S}^{\dagger}\mathbf{D})'$$ pszeudo-inverz

vagy

$$\mathbf{W}^{out} = (\mathbf{R} + \alpha^2\mathbf{I})^{-1}\mathbf{P}$$ Ridge-regression (Tikhonov regularizáció)



Hivatkozások

A prezentáció a következő helyekről vett tartalmakkal készült:

- CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University
  - http://cs231n.stanford.edu/syllabus.html
- Understanding LSTM Networks
  - http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- The Unreasonable Effectiveness of Recurrent Neural Networks
  - http://karpathy.github.io/2015/05/21/rnn-effectiveness/