

# Neuronové sítě v dnešním světě

Tomáš Brázdil

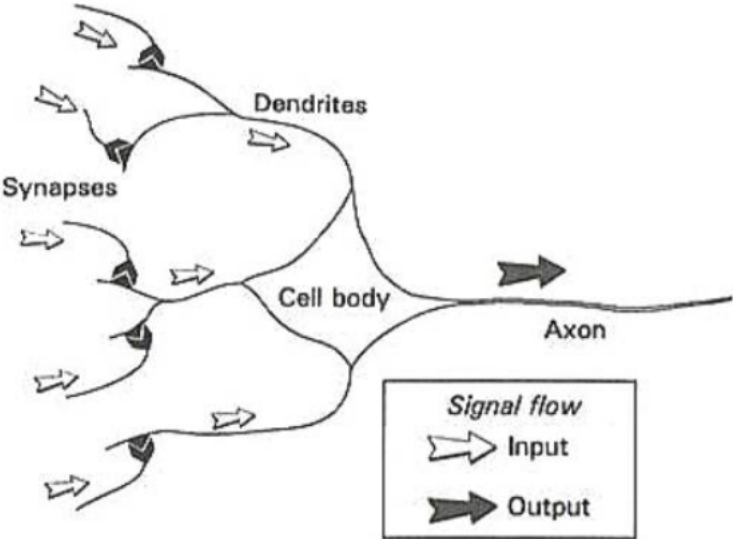
# O čem to bude ...

- ▶ Jemný úvod do neuronových sítí
- ▶ Trocha historie
- ▶ Neuronové sítě a rozpoznávání obrázků

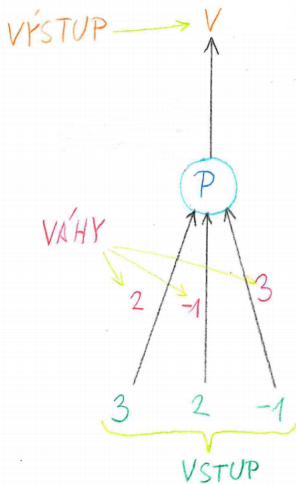
## Vyjasnění pojmu:

**Deep learning** = staré dobré neuronové sítě, ovšem velké, hluboké, podepřené silným hardwarem a všeho schopné.

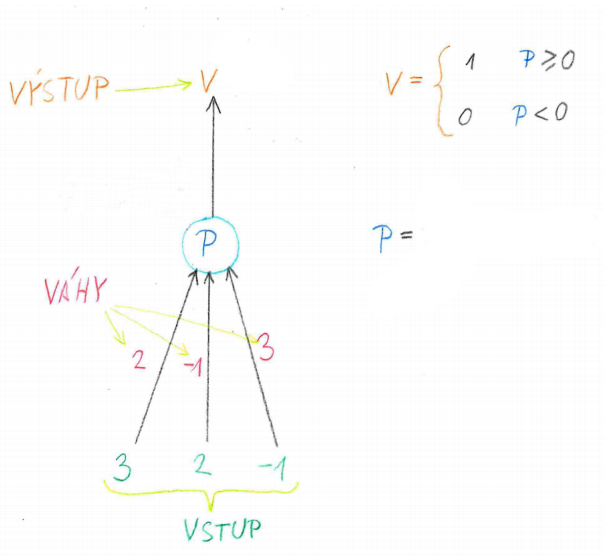
# Biologická inspirace



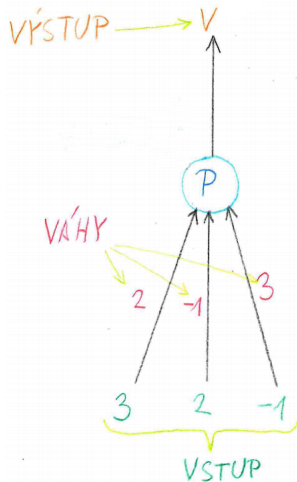
# Umělý neuron



# Umělý neuron



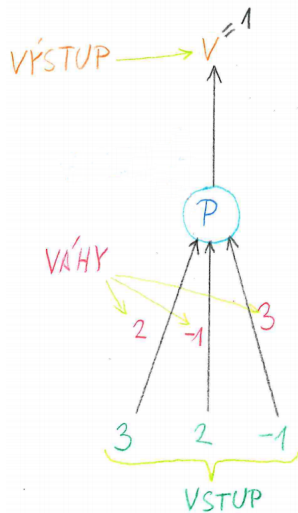
# Umělý neuron



$$V = \begin{cases} 1 & P \geq 0 \\ 0 & P < 0 \end{cases}$$

$$P = 2 \cdot 3 - 1 \cdot 2 + 3 \cdot (-1) \\ = 1$$

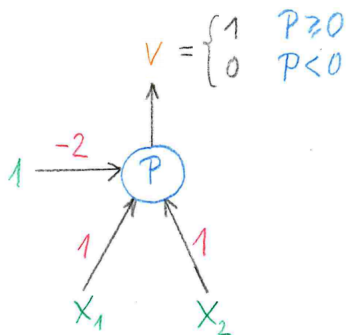
# Umělý neuron



$$V = \begin{cases} 1 & P \geq 0 \\ 0 & P < 0 \end{cases}$$

$$P = 2 \cdot 3 - 1 \cdot 2 + 3 \cdot (-1) \\ = 1$$

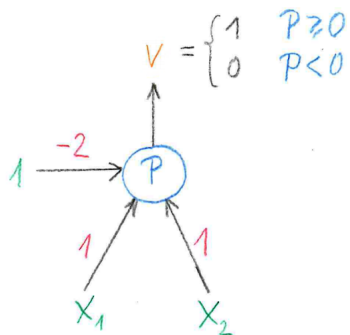
# Logické funkce





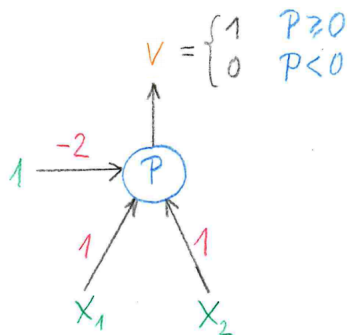


# Logické funkce



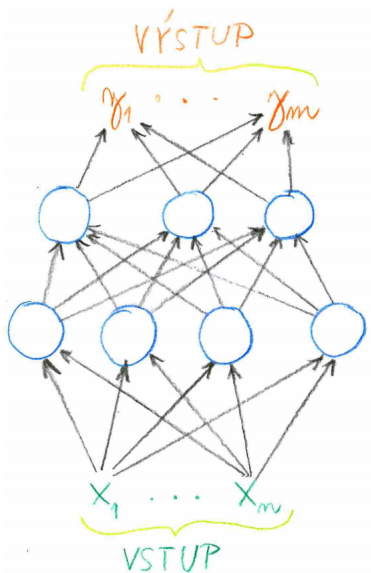
| $x_1$ | $x_2$ | $V$             |
|-------|-------|-----------------|
| 1     | 1     | $\rightarrow$ 1 |
| 1     | 0     | 0               |
| 0     | 1     | 0               |
| 0     | 0     | 0               |

# Logické funkce



| $x_1$ | $x_2$ | $V$ |
|-------|-------|-----|
| 1     | 1     | → 1 |
| 0     | 1     | → 0 |
| 1     | 0     | → 0 |
| 0     | 0     | → 0 |

# Vícevrstvé síť



# Trocha teorie

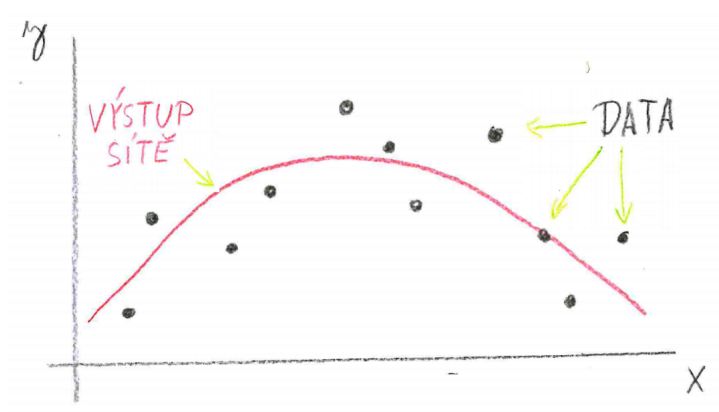
- ▶ Neuronové sítě jsou univerzální výpočetní prostředek
  - ▶ dvouvrstvé sítě zvládají logiku a umí počítat libovolné funkce
  - ▶ sítě jsou univerzálním výpočetním prostředkem  
(jako např. programy v libovolném programovacím jazyce)
- ▶ ... a hlavně se umí **učit!**

# Učení



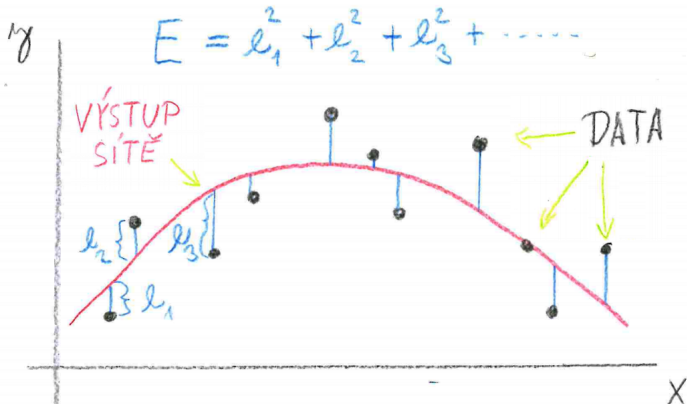
Tréninková množina obsahuje dvojice  $(x, y)$  kde  $x$  je vstup a  $y$  žádaný výstup.

# Učení



Tréninková množina obsahuje dvojice  $(x, y)$  kde  $x$  je vstup a  $y$  žádaný výstup.

# Učení



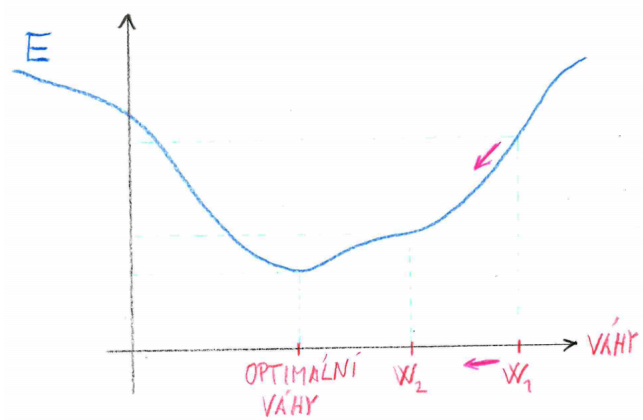
Tréninková množina obsahuje dvojice  $(x, y)$  kde  $x$  je vstup a  $y$  žádaný výstup.

Minimalizujeme chybu  $E$  pomocí změn vah.



# Minimalizace

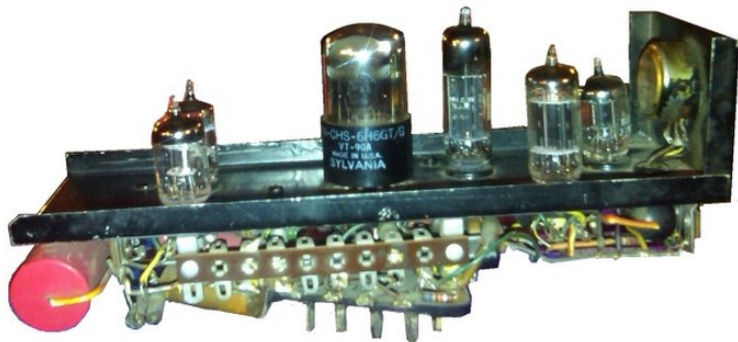
Klasická metoda: Gradientní sestup (+ hromada heuristik).



Prostě utíkáme z kopce v chybové vysočině.

# Trocha historie neuronových sítí

- ▶ 1951: SNARC (Minski a spol.)
  - ▶ první implementace neuronu
  - ▶ krysa hledá cestu ven z bludiště
  - ▶ 40 umělých neuronů (300 elektronek, spousta motorů apod.)



# Trocha historie neuropočítačů

- ▶ 1957: Mark I Perceptron (Rosenblatt a spol.) - první úspěšná neuronová síť pro rozpoznávání obrázků



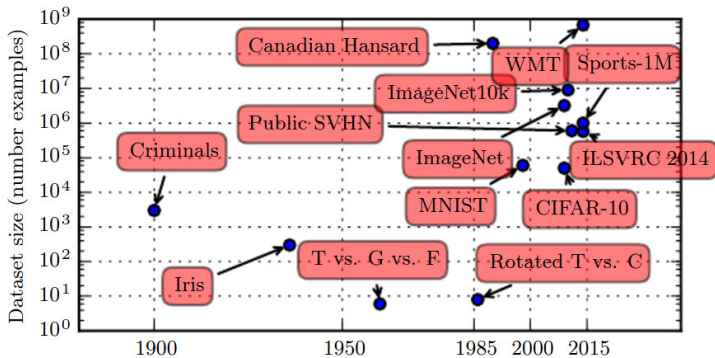
- ▶ jednovrstvá síť
- ▶ obraz snímán  $20 \times 20$  fotovodiči
- ▶ intenzita bodů byla vstupem, síť rozpoznala znak
- ▶ váhy implementovány pomocí potenciometrů (nastavovaných motory)

# Trocha historie

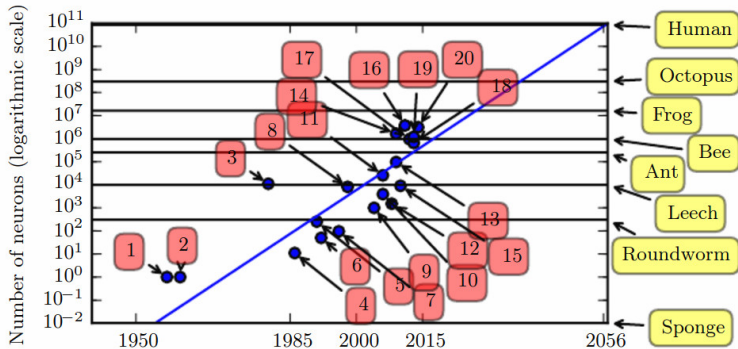
- ▶ **1967-82:** Převážně mrtvo po zveřejnění práce Miského a Paperta (oficiálně vydána roku 1969 pod názvem *Perceptrons: an introduction to computational geometry*)
- ▶ **1983-konec devadesátých let:** Rozmach neuronových sítí
- ▶ **konec devadesátých let-cca 2005:** NS zatlačeny do pozadí jinými modely (support vector machines (SVM))
- ▶ **2006-nyní:** Renaissance neuronových sítí
  - ▶ Hluboké sítě (mnoho vrstev)
  - ▶ Převážně softwarové implementace
  - ▶ Hardware pro počítačové hry – grafické karty (GPU)
  - ▶ Úžasné aplikace (obraz, zvuk, text, řízení, hry, ...)

# Data?

Rostou ...



Rostou ...




2. Rosenblatt's Perceptron
4. Early back-propagation network (Rumelhart et al., 1986b)
8. Image recognition: LeNet-5 (LeCun et al., 1998b)
10. Dimensionality reduction: Deep belief network (Hinton et al., 2006)  
... here the third "wave" of neural networks started
15. Digit recognition: GPU-accelerated multilayer perceptron (Ciresan et al., 2010)
18. Image recognition (AlexNet): Multi-GPU convolutional network (Krizhevsky et al., 2012)
20. Image recognition: GoogLeNet (Szegedy et al., 2014a)

# Hardware

Zmenšuje se ...

2012

GOOGLE DATACENTER



1,000 CPU Servers  
2,000 CPUs • 16,000 cores

**600 kWatts**  
**\$5,000,000**



2014

STANFORD AI LAB



3 GPU-Accelerated Servers  
12 GPUs • 18,432 cores

**4 kWatts**  
**\$33,000**

# Software

- ▶ Jazyk **Python**
- ▶ Specializované knihovny: Tensorflow (Google), CNTK (Microsoft) a další

```
model = Sequential()  
model.add(Dense(32, activation='relu', input_dim=100))  
model.add(Dense(10, activation='softmax'))  
model.compile(optimizer='rmsprop',  
              loss='categorical_crossentropy',  
              metrics=['accuracy'])
```

```
# Train the model, iterating on the data in batches of 32 samples  
model.fit(data, one_hot_labels, epochs=10, batch_size=32)
```



# Applikace

- ▶ Rozpoznávání obrazu
- ▶ Strojový překlad
- ▶ Řízení a strategické rozhodování



# ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)

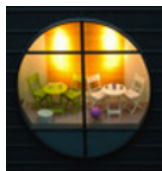
Soutěž v klasifikaci obrázků nad podmnožinou databáze ImageNet.

V roce 2012:

- ▶ Tréninková množina: 1,200,000 obrázků, 1000 kategorií
- ▶ Testovací: 150,000 obrázků

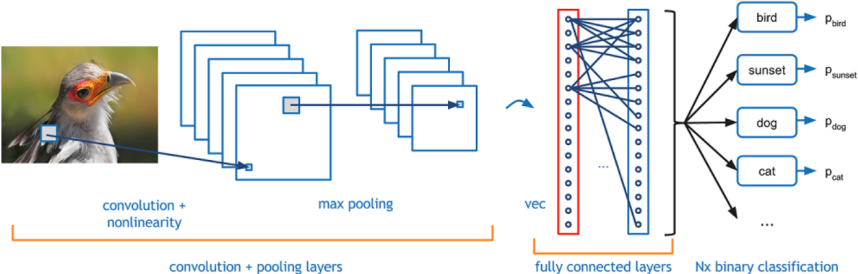
Mnoho obrázků obsahuje několik objektů → síť seřadí jednotlivé kategorie podle pravděpodobnosti výskytu daného objektu:

- ▶ Stůl - 0.1
- ▶ Židle - 0.08
- ▶ Pomeranč - 0.055
- ▶ ...



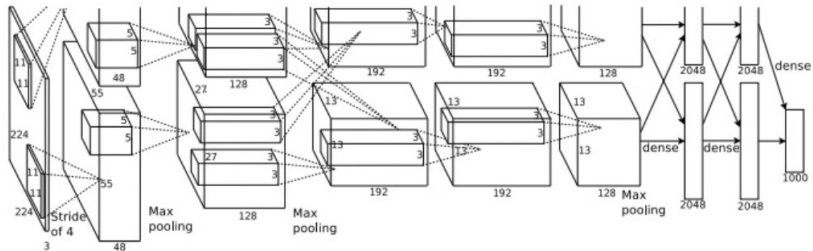
Uspěje, pokud je správný objekt mezi 5 nejpravděpodobnějšími.

# Konvoluční síť



# ILSVRC 2012 – AlexNet

ImageNet classification with deep convolutional neural networks, by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton (2012).



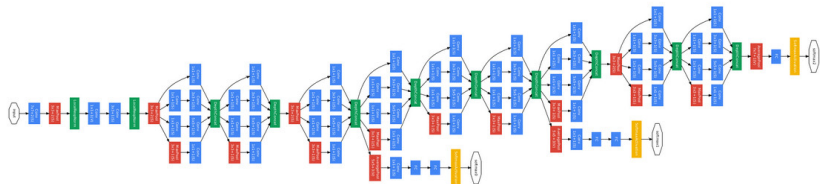
Trénováno na dvou GPU (NVIDIA GeForce GTX 580)

Výsledky:

- ▶ Chyba **15.3%** v top-5 kategoriích!  
(druhý nejlepší algoritmus: 26.2%)

# ILSVRC 2014 – GoogLeNet

hluboká konvoluční síť, 22 vrstev



Výsledky:

- ▶ Chyba: **6.5%** v top-5 kategoriích

Je to nadlidský výkon?

# ILSVRC 2015 – ResNet

- ▶ 152 vrstev, složitá architektura
- ▶ 8 GPU
- ▶ Chyba: **3.57% accuracy** v top-5

Je tohle nadlidské? :-)



## 2016 (2017)

Vítěz spojil několik modelů z předešlých let do „hlasovacího“ systému, chyba se dostala pod 3% a soutěž již není příliš zajímavá.

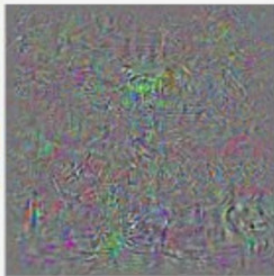


# Limity hlubokých sítí

Správně!

modifikace

Špatně!!



... je možné konstruovat celé prostory takových příkladů.

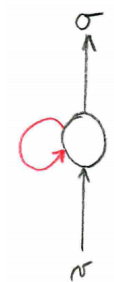
# Práce s textem

- ▶ Strojový překlad
- ▶ Generování textu
- ▶ ...

Jak dostat do sítě sekvence různé délky?

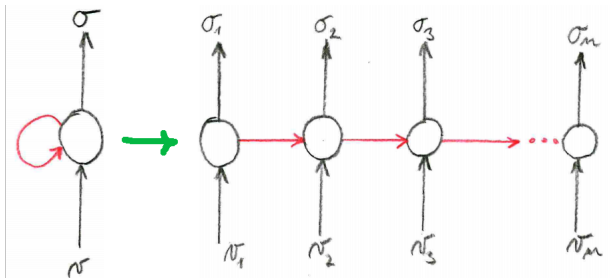
**Rekurentní sítě!**

# Rekurentní síť



- ▶  $v$  je vstup
- ▶  $o$  je výstup
- ▶ červenou smyčkou síť předává svůj výstup na svůj vstup

# Rekurentní síť



LSTM = trochu složitější verze tétož

# LSTM – Shakespeare

- ▶ Generování Shakespearových her znak po znaku!
- ▶ Tréninková množina: Všechna díla seřazená za sebe (4.4MB).
- ▶ 3-vrstvy LSTM, 512 neuronů v každé z nich.

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;  
When I was heaven of presence and our fleets,  
We spare with hours, but cut thy council I am great,  
Murdered and by thy master's ready there  
My power to give thee but so much as hell:  
Some service in the noble bondman here,  
Would show him to her wine.

KING LEAR: O, if you were a feeble sight, the courtesy of your law,  
Your sight and several breath, will wear the gods  
With his heads, and my hands are wonder'd at the deeds,  
So drop upon your lordship's head, and your opinion  
Shall be against your honour.

## Fun with LSTM – Wikipedia

- ▶ Hutter Prize 100MB dataset dat z encyklopedie Wikipedia (96MB)

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>

## Náhodná halucinace korektního xml:

```
<page>
  <title>Antichrist</title>
  <id>865</id>
  <revision>
    <id>15900676</id>
    <timestamp>2002-08-03T18:14:12Z</timestamp>
    <contributor>
      <username>Paris</username>
      <id>23</id>
    </contributor>
    <minor />
    <comment>Automated conversion</comment>
    <text xml:space="preserve">
      #REDIRECT [[Christianity]]</text>
  </revision>
</page>
```

# LaTeX

- ▶ Trénováno na učebnici algebraické geometrie.
- ▶ LaTeX zdroj (16MB).
- ▶ Výsledek bylo téměř možné přeložit.



*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  $S$  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*

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

*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

# Zdrojový kód Linuxu

- ▶ Trénováno na všech zdrojových a hlavičkových souborech Linuxu seřazených do jednoho velkého souboru (474MB of C kódu).
- ▶ 3-vrstvá LSTM, zhruba 10 milionů vah.

```

/*
 * 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;
}

```

```
/*
 * 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>
```

# Evolution of Shakespeare

100 iter.:

tyntd-iafhatawiaoihrdemot 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

300 iter.:

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

500 iter.:

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

700 iter.:

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.

1200 iter.:

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

2000 iter.:

"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.

# Závěr

- ▶ Staré sítě nový hardware (GPU)
- ▶ Potřebují obrovské množství dat
- ▶ Aplikace: zpracování obrazu, strojový překlad (a další)

## Budoucnost?

- ▶ „Univerzální“ AI
- ▶ Nutné překonávat omezení
- ▶ Lépe porozumět, co to vlastně dělá
- ▶ Větší a rychlejší!

## Srovnání s klasickou architekturou počítačů

|                     | Neuronové sítě                                       | „Klasické“ počítače                                  |
|---------------------|--|--|
| Data                | implicitně ve váhách                                 | explicitně   |
| Výpočet             | přírodně paralelní                                   | sekvenční (obvykle),<br>lokalizovaný                 |
| Odolnost            | odolné vůči nepřesnosti<br>vstupu a poškození        | změna jednoho bitu<br>může znamenat krach<br>výpočtu |
| Přesnost<br>výpočtu | nepřesný, síť si „vybaví“<br>podobný tréninkový vzor | přesný   |
| Progra-<br>mování   | učí se ze vzorového<br>chování                       | je nutné precizně<br>programovat                     |