

Modelování myšlení

Radek Pelánek

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Myšlení, modelování, počítače

(zjednodušené, plynulý přechod)

kognitivní modelování

- cíl: zachytit, jak funguje mysl
- důraz na jednoduchost, věrnost, principy
- není úplně zásadní „výkon“

umělá inteligence

- cíl: reprodukovat inteligentní chování
- důraz na „výkon“
- biologické principy jen jako inspirace

Kognitivní modelování

symbolické modelování

- „high-level“
- uvažování na úrovni pravidel, metafora počítače
- manipulace symbolů

konekcionistické modelování

- „low-level“
- uvažování na úrovni neuronů
- snaha o biologickou věrnost

Porozumění mozku

Kdyby byl mozek tak jednoduchý, že bychom mu mohli rozumět, byli bychom tak jednoduší, že bychom nemohli.
(Lyaal Wattson)

Symbolické modelování

- *computational theory of mind*
- myšlení jako manipulace symbolů
- produkční pravidla: IF X THEN Y
- učení: vytváření nových pravidel (např. spojováním, vyhledáváním)
- kde se vezmou symboly?

Produkční systém

- pravidla typu
IF podmínka THEN akce
- podmínky přichází z prostředí nebo jako následky jiných pravidel
- akce mohou vyvolávat další pravidla nebo měnit prostředí

Příklad:

IF chci čaj THEN uvařit čaj

IF uvařit čaj a nemám horkou vodu THEN dát vodu do konvice

Produkční systémy: kolize

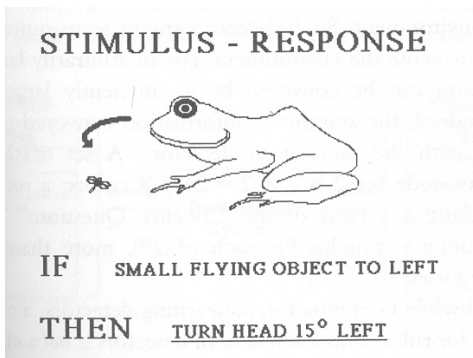
- pravidla mohou být **kolizní** (nedeterministický systém)
- pro danou situaci výběr aplikovatelných, rozhodnutí kolize
- priority, náhoda, ...

Příklad:

IF chci čaj THEN uvařit čaj

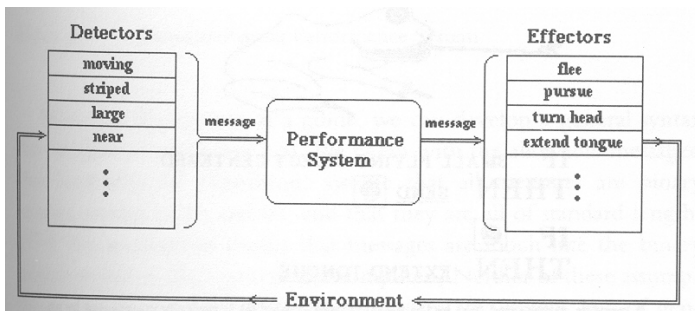
IF chci čaj THEN zajít do hospody a objednat čaj

Příklad žába: základní princip



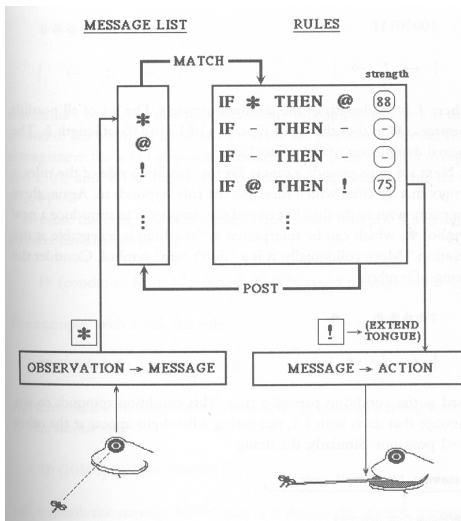
(John Holland, Hidden Order: How Adaptation Builds Complexity)

Příklad žába: kontext

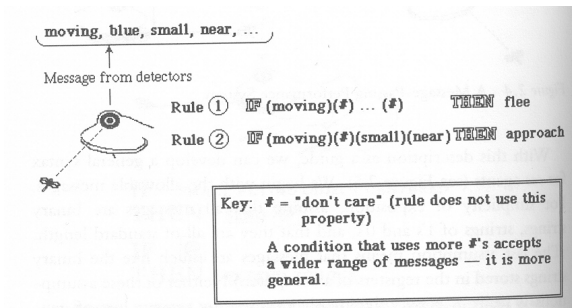


(John Holland, *Hidden Order: How Adaptation Builds Complexity*)

Příklad žába: produkční systém

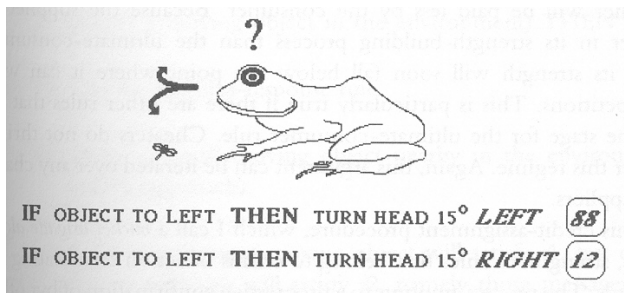


Příklad žába: výběr použitelných pravidel



(John Holland, *Hidden Order: How Adaptation Builds Complexity*)

Příklad žába: řešení kolize



(John Holland, *Hidden Order: How Adaptation Builds Complexity*)

Klasifikační systémy

(Classifier systems)

Konkrétní příklad architektury:

- 1 produkční systém
- 2 ohodnocení pravidel (bucket brigade)
- 3 objevování pravidel pomocí genetického algoritmu

Ohodnocení pravidel: bucket brigade

- pravidla konfliktní
- pravidlo má **kapitál**
- výběr pravidla podle aktuálního kapitálu
- pravidlo **platí** za to, že může být použito
- úspěšná aplikace \Rightarrow zisk **odměny** od prostředí
- část odměny **redistribuována** pravidlům, které umožnili použití pravidla, které vedlo k odměně

Vývoj nových pravidel

- pravidla jsou reprezentována jako binární řetězce
- přímočaré použití genetického algoritmu
- zdatnost = kapitál

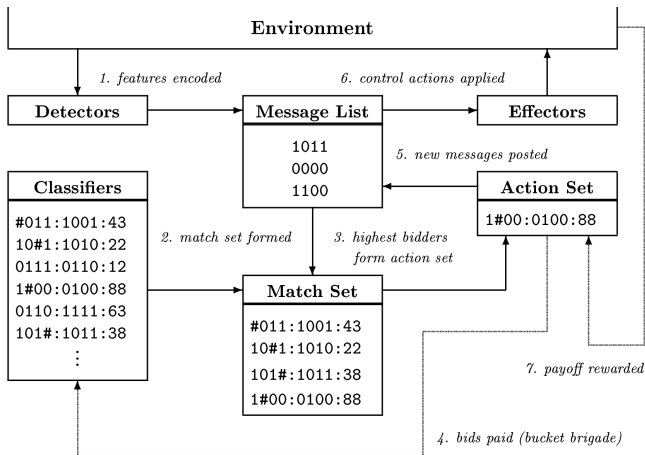


Figure 21.2 A classifier system interacting with its environment

(G. W. Flake, *The Computational Beauty of Nature*)

Příklad Woods

- (podobné jako Robby the Robot u gen. alg.)
- zvíře na čtverečkováném poli
 - '*' zvíře
 - '.' volné pole
 - 'O' skála
 - 'F' jídlo
- zvíře vidí okolních 8 pozic, tj. detektory lze zakódovat pomocí 16 bitového řetězce (. = 00, O = 10, F=11)
- cílem je najít co nejrychleji jídlo

Příklad Woods

```

.....
.00F..00F..00F..00F..00F..00F..00F..00F..00F..00F..
.000..000..000..000..000..000..000..000..000..000..
.000..000..000..000..000..000..000..000..000..000..
.....
.....
.00F..00F..00F..00F..00F..00F..00F..00F..00F..00F..
.000..000..000..000*.000..000..000..000..000..000..
.000..000..000..000..000..000..000..000..000..000..
.....
.....
.00F..00F..00F..00F..00F..00F..00F..00F..00F..00F..
.000..000..000..000..000..000..000..000..000..000..
.000..000..000..000..000..000..000..000..000..000..
.....

```

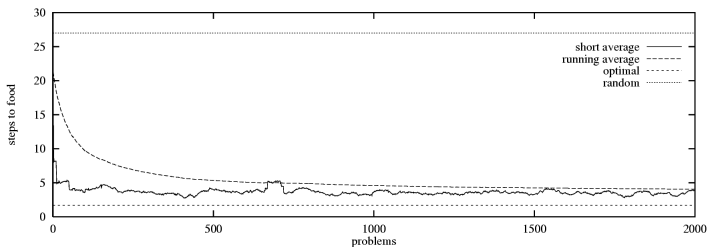
(G. W. Flake, *The Computational Beauty of Nature*)

Příklad Woods

Realizace experimentu (učení):

- náhodně umístit zvíře
- nechat běžat dokud nenajde jídlo
- a pak znova od začátku

Příklad Woods: výsledky



(G. W. Flake, *The Computational Beauty of Nature*)

Příklad Woods: složitější prostředí

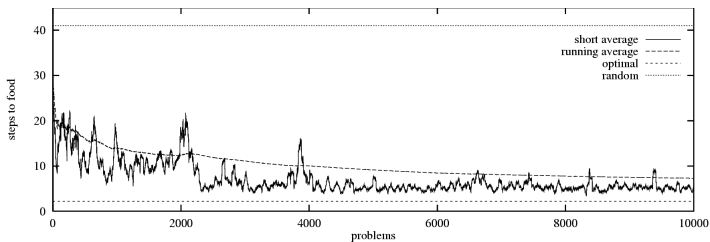
```

.....0......00.....0.....
.OFO.....F.....F.....0.....F.....FO.....
.....0.....00.....F.....
.....0.....0.....F.....0.....
...F.....OFO.....OFO.....F.....00.....F...
...00.....0.....
.....00.....0.....00.....0.....
.OFO.....F.....OF.....F.....OFO.....F.....
.....0.....
...00.....0.....0.....00.....0.....
...F.....F.....0.....FO.....F.....OF...OFO.
.....0.....OF.....
...0.....F.....0.....0.....0.....0...
...F.....F.....F.....FO.....F.....OF...
...0.....00.....0.....0.....
.....0.....0.....
...F.....OFO.....F.....F.....F.....OF.....
...00.....0.....00.....0.....0.....

```

(G. W. Flake, *The Computational Beauty of Nature*)

Příklad Woods: výsledky II

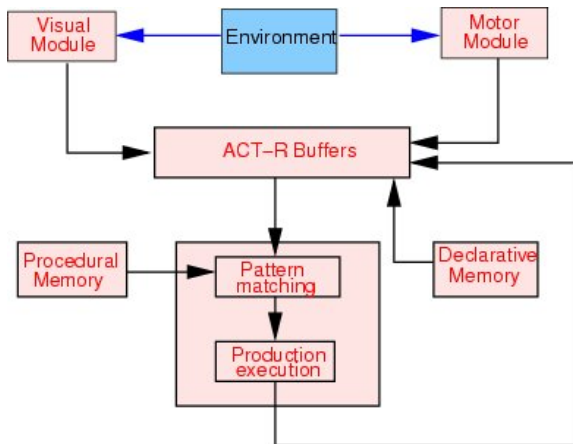


(G. W. Flake, *The Computational Beauty of Nature*)

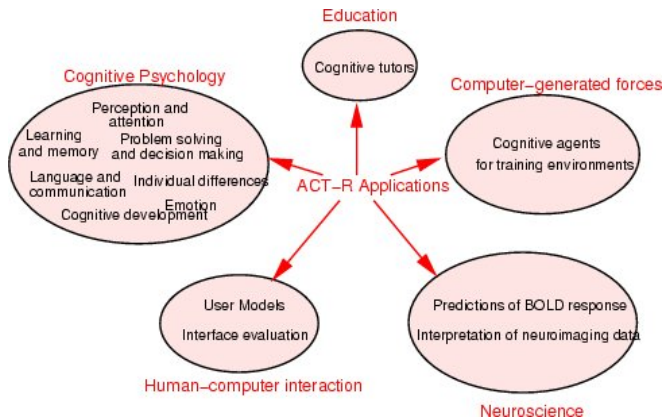
Příklady: modelování induktivního myšlení

- El Farol Bar
- model trhu – *Asset Pricing Under Endogenous Expectations in an Artificial Stock*

ACT-R (Adaptive Control of Thought–Rational)



ACT-R: aplikace



Aplikace ve výuce

- inteligentní výukové systémy
- produkční systém pro určitou oblast (např. aritmetika)
- model „sleduje“ řešení studenta \Rightarrow ohodnocení schopností studenta, identifikace silných/slabých míst
- pravidla pro typické chyby \Rightarrow možnost cílené zpětné vazby

Neuronové sítě

- model založený na zjednodušené imitaci fungování mozku
- neurony, spojení
- **induktivní učení** prostřednictvím trénovacích dat
- „znalosti“ jsou v síti uloženy jako váhy **spojení** mezi neurony
- podrobnosti: PV021 Neuronové sítě

Neuron

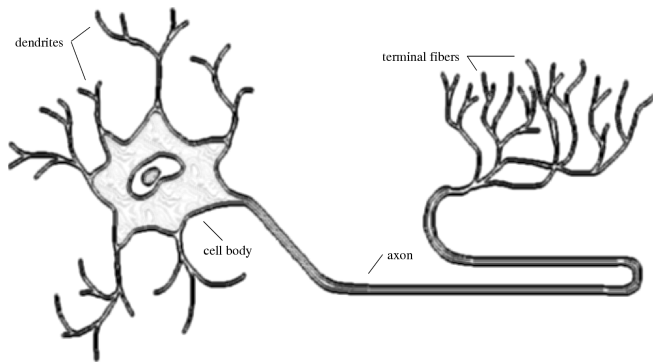
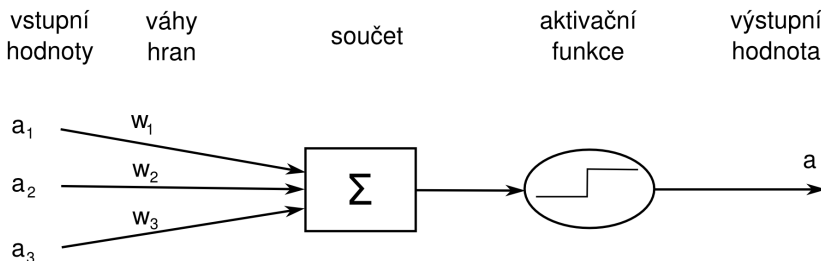


Figure 18.1 A “typical” neuron with major components identified

(G. W. Flake, *The Computational Beauty of Nature*)

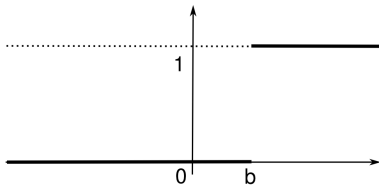
Model neuronu



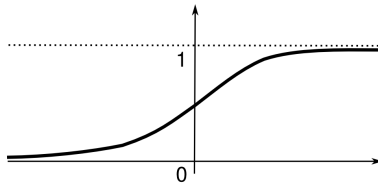
$$a(t + 1) = \Phi\left(\sum_{i=1}^n w_i \times a_i(t) - b\right)$$

Model neuronu: aktivační funkce

diskrétní aktivační funkce



spojitá aktivační funkce

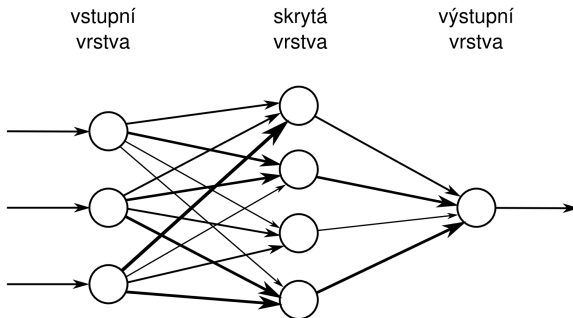


Umělá neuronová síť

... připomíná reálnou neuronovou síť:

- neurony spojené váženými vazbami
- schopná se učit na základě předkládaných dat
- „znalosti“ uloženy jako váhy spojení mezi neurony

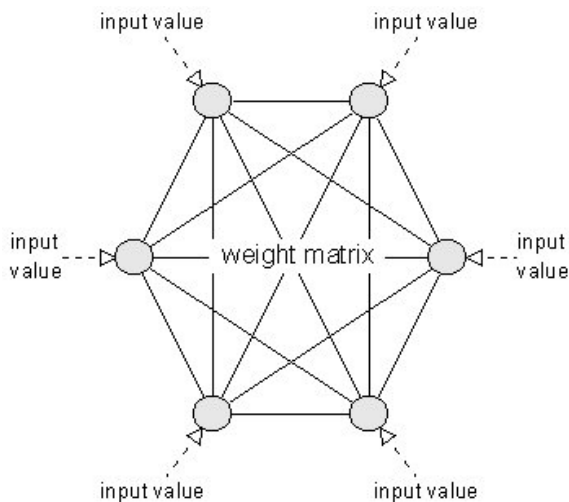
Příklad architektury — dopředná síť



Dopředná síť: učení

- předkládáme sadu příkladů: vstup + žádaný výstup
- upravujeme váhy spojení mezi neurony (např. backpropagation algoritmus)
- učení = korigující negativní zpětná vazba

Příklad architektury — zpětnovazebná síť



Zpětnovazební síť

- zpětnovazební síť – obsahuje cykly
- modelování asociativní paměti
 - vězeň, dramatik, prezident
 - kd jinm jmu kop sm do n pad
- Hebbův zákon učení:
dva neurony aktivovány ve stejnou chvíli \Rightarrow posílení spojení

Rozpoznávání znaků

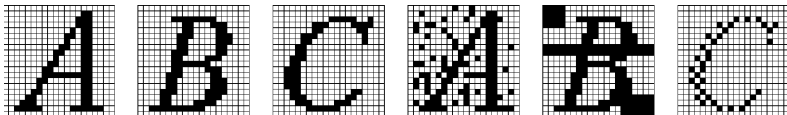


Figure 18.3 Bitmapped images of letters from the alphabet: The first three are clean version that are used as patterns to be stored. The last three are used as seed images that the associative memory must use to recall one of the first three.

(G. W. Flake, The Computational Beauty of Nature)

Řešení optimalizačního problému

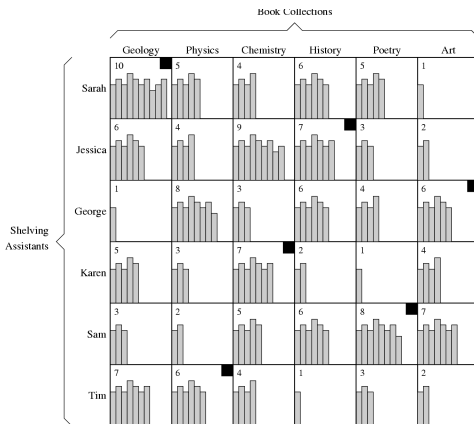


Figure 18.6 The task assignment problem: Black squares in the entries denotes the optimal assignment with a total shelving rate of 44.

Řešení optimalizačního problému

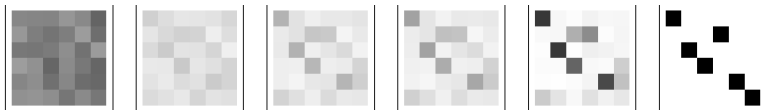


Figure 18.7 Computing a neural solution to the task assignment problem: This particular solution yields a total rate of 42, which is just less than the optimal solution.

(G. W. Flake, *The Computational Beauty of Nature*)

Vývoj sdíleného slovníku

- model **vytváření sdíleného slovníku** ve skupině agentů
- agenti si vytváří slovník pro objekty, pojmenování je reprezentováno pomocí neuronové sítě
- interakce = agenti si vymění svoje názory na název objektu, trochu se přeučí
- díky interakci vzniká sdílený slovník

Příklady aplikací

- klasifikace vzorů, např.
 - rozpoznávání jazyka dokumentu podle frekvencí písmen
 - rozpoznání čisté a naředěné malinové hmoty dle infračerveného spektra
- rozpoznávání obrazů, řeči
- aproximace funkcí, predikce řad
- zpracování dat
- hry, rozhodování

Příklad implementace: FANN

- Fast Artificial Neural Network Library (FANN)
<http://leenissen.dk/fann/>
- implementuje klasický model neuronové sítě včetně učících algoritmů
- implementováno v C, rozhraní pro další jazyky
- volně dostupné
- příklad: určení jazyka na základě frekvencí písmen

Použití FANN: učení

```
#include "fann.h"

int main() {
    struct fann *ann = fann_create(1, 0.7, 3,
                                    26, 13, 3);
    fann_train_on_file(ann, "frequencies.data",
                      200, 10, 0.0001);
    fann_save(ann, "language_classify.net");
    fann_destroy(ann);
    return 0;
}
```

Použití FANN: klasifikace pomocí sítě

```
int main(int argc, char* argv[]) {  
    struct fann *ann = fann_create_from_file(  
        "language_classify.net");  
    float frequencies[26];  
    generate_frequencies(argv[1], frequencies);  
    float *output = fann_run(ann, frequencies);  
    cout << "English: " << output[0] << endl  
    << "French : " << output[1] << endl  
    << "Polish : " << output[2] << endl;  
    return 0;  
}
```

Shrnutí

- účel modelování myšlení:
 - pochopení, jak myšlení funguje
 - zlepšení modelů (s agenty)
 - aplikace pro řešení „inženýrských“ problémů
- typy modelů:
 - symbolické
 - konekcionistické
 - hybridní (neprobíráno)