

# Učení, vývoj, adaptace

Radek Pelánek

# Obsah I

- 1 Neuronové sítě
  - Úvod
  - Model
  - Aplikace
  - Implementace
- 2 Genetické algoritmy
  - Úvodní poznámky
  - Evoluce
  - Genetické algoritmy: model
  - Příklady

# Obsah II

- 3 Klasifikační systémy
  - Popis
  - Příklady
  - Induktivní uvažování
  
- 4 Závěr

# Hlavní zdroj

- **Computation Beauty of Nature.** Gary Flake. The MIT Press, 2000.
- **Hidden Order: How Adaptation Builds Complexity.** J. H. Holland. Addison Wesley, 1995.

# Změny v systémech

Komplexní systémy se mění...

system	rychlost změn
nervový systém	sekundy až hodiny
imunitní systém	hodiny až dny
firma	měsíce až roky
živočišný druh	dny až století
ekosystém	roky až milénia

vztah k horizontu modelu (příklad vlk: reintrodukce, lov ve smečce)

# Smysl modelů adaptace

- 1 pochopení přírody — jak funguje učení, evoluce, ...
- 2 lepší modely komplexních systémů — obohacení modelů s agenty
- 3 řešení náročných algoritmických problémů — modely jako výpočetní mechanismy

# Mozek

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

# Neuronové sítě

- model založený na zjednodušené imitaci fungování mozku
- neurony, spojení
- síť se učí prostřednictvím trénovacích dat
- „znalosti“ jsou v síti uloženy jako váhy spojení mezi neurony

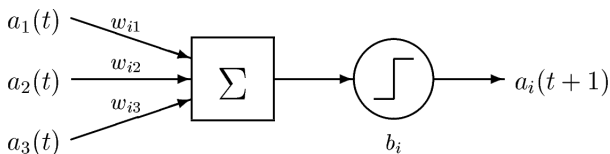




# Srovnání mozku a počítače

	mozek	počítač
počet jednotek	$10^{11}$ neuronů	$10^8$ tranzistorů cca 50 umělých neuronů
počet spojení na jednotku	řádově tisíce	řádově jednotky
počet operací	100/s	$10^9$ /s
rychlost signálu	150 m/s	300 000 km/s

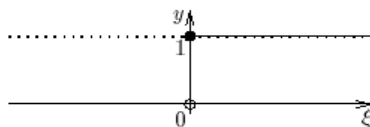
# Model neuronu



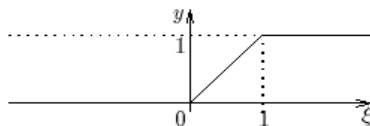
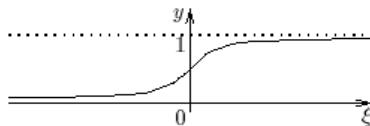
**Figure 18.2** A single McCulloch-Pitts neuron

$$a_i(t+1) = \Phi\left(\sum_{j=1}^n w_{ij} \times a_j(t) - b_i\right)$$

# Model neuronu: aktivační funkce



hard limiter

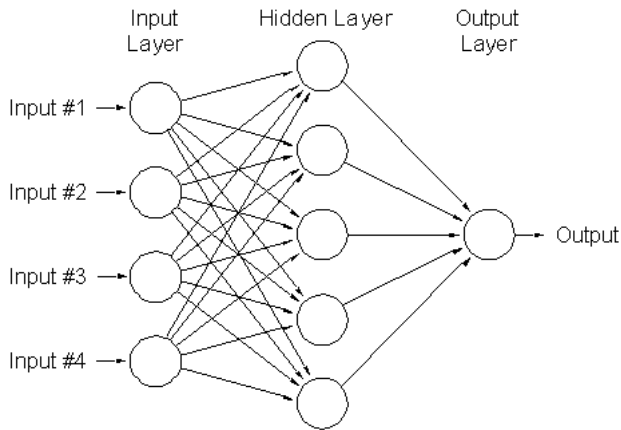
piecewise-linear  
functionstandard (logistic)  
sigmoid

# Umělá neuronová síť

... připomíná reálnou neuronovou síť:

- abstraktní neurony spojené váženými vazbami
- schopná se učit na základě předkládaných dat
- učí se prostřednictvím trénovacích 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 = negativní zpětná vazba

# Zpětnovazební síť

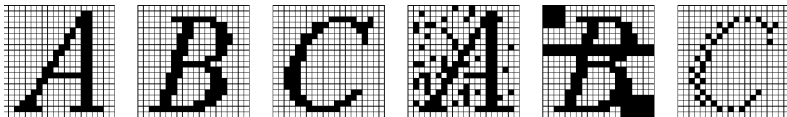
- zpětnovazební síť – obsahuje cykly
- modelování asociativní paměti
  - vězeň, dramatik, prezident
  - kd jinm jmu kop sm do n pad
- Hebbův zákon učení:  
dva neurony aktivovány ve stejnou chvíli  $\Rightarrow$  posílení spojení



# Příklady aplikací

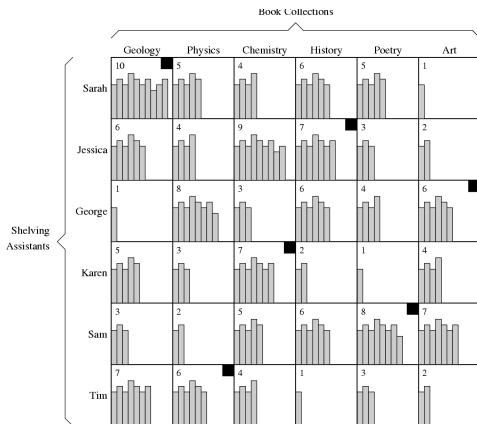
- klasifikace vzorů, např.
  - rozpoznávání jazyka dokumentu podle frekvencí písmen
  - rozpoznání čisté a naředěné malinové hmoty dle infračerveného spektra
- rozpoznávání obrazů, řeči
- aproximace funkcí, predikce řad
- zpracování dat
- hry, rozhodování

# 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.

# 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.

# 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

# 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;
}
```



# Evoluce, genetické algoritmy

*Přírodní výběr je mechanismus pro generování mimořádně velkého stupně nepravděpodobnosti. (Sir Ronald Fisher)*

*Slepice je jenom způsob, jakým vajíčko vyrábí další vajíčko. (Samuel Butler)*

# Historické poznámky

- 1859: Darwin, Origin of species
- 1865: Mendel, dědičnost
- 1953: struktura DNA
- 1976: Dawkins, Selfish gene

# Biologické poznámky

- genetická informace uložena v DNA
- DNA = řetězec “základních bloků”: adenine (A), cytosine (C), guanine (G), thymine (T)
- při reprodukci dochází ke kombinaci (křížení) DNA rodičů
- náhodné mutace

# Základy biologické evoluce

- vyvíjí se populace jako celek (nikoliv jednotlivci)
- diverzita je důležitá
- schopnosti získané během života se nepřenáší  
srovnej: Lamarck, „kulturní evoluce“, simulovaná evoluce
- organismy se adaptují na okolí, které se však může měnit  
(např. i vlivem aktivit organismu)  
koevoluce, Red Queen effect

# Biologický příklad: koevoluce (biologické závody ve zbrojení)

- netopýři mají sonar, kterým hledají můry, sonar je sám o sobě zapeklitě komplikovaný
- můry vyvinuly měkké pokrytí těla, které absorbuje netopýří vysílání
- netopýři přešli na nové frekvence
- můry přišly s novým pokrytím a s “rušičkou” (vlastní signál který interferuje s netopýřím)
- netopýři přišly s novými leteckými manévry a naučily se vypínat sonar (čímž dělají rušení méně efektivním)

# Koevoluce v IT

- viry a antivirová ochrana
- spamy a antispamová ochrana

# Evoluce

- přežití nejsilnějších (survival of the fittest) nebo spíš přežití schopných reprodukce (survival of the reproducers)
- *všichni naši předci žili tak dlouho, že se stihli reprodukovat*
- pozitivní zpětná vazba

# Typy evolučních modelů

- přežití nejsilnějších (bez vývoje) (zváno též ekologické modely)
- přežití nejsilnějších + vývoj (křížení, mutace)
  - genetické algoritmy
  - evoluční programování
  - genetické programování
  - ...



# Interpretace evolučních principů

- „nejsilnější“ mají největší šanci reprodukce
- učení se metodou pokus-omyl (a pamatování si úspěšných)
- nápodoba úspěšných

V žádném případě není potřeba „racionalita“ agentů, uvědomění si, co přesně dělají.

# Genetické algoritmy: základní přístup

- Vyber počáteční populaci  $P$
- Opakuj dle potřeby:
  - Vytvoř novou prázdnou populaci  $P'$
  - Opakuj dokud  $P'$  není plná:
    - Vyber dva jedince z populace  $P$  v závislosti na kritériu zdatnosti (fitness)
    - Volitelně: křížení a nahrazení potomky
    - Volitelně: mutace
    - Přidej do populace  $P'$
  - $P := P'$

# Reprezentace

Jedinci v populaci reprezentovány jako řetězce (většinou binární).

Občas vyžadujeme, aby řetězec splňoval určitou vlastnost (např. permutace čísel).

# Výběr

- výběr založen na zdatnosti (fitness), pokud nemáme absolutní měřítko, necháme jedince spolu „soutěžit“
- deterministický výběr nejlepších vede k:
  - ztráta diverzity
  - uváznutí na lokálním minimu
- proto: náhodnostní výběr s přihlédnutím k zdatnosti



# Mutace

- náhodná změna v řetězci, opět mohou být potřeba speciální úpravy
- většinou nevede ke zlepšení
- může pomoci překonat lokální optima
- používáno, ale s malou pravděpodobností

# Velikost populace

- velikost populace  $\times$  počet generací  $>$  100 000
- velikost populace  $\gg$  počet genů

# Hledání řetězce

Time	Average Fitness	Best Fitness	Best String
0	0.035314	0.200000	"pjrmrubynrksxiidwctxfodkodjjzfunpk "
1	0.070000	0.257143	"pjrmrubynrksxiidnybvsqcq piisyexdt"
⋮	⋮	⋮	⋮
25	0.708686	0.771429	"qurmous gresn idnasvsweqt prifuseky"
26	0.724286	0.800000	"qurmous green idnasvsweqt prifuseky"
⋮	⋮	⋮	⋮
36	0.806514	0.914286	"uurious green idnas sweqt profusely"
37	0.820857	0.914286	"qurmous green ideas sweqt profusely"
⋮	⋮	⋮	⋮
41	0.895943	0.942857	"uurious green idnas sweat profusely"
42	0.908457	0.971429	"qurious green ideas sweat profusely"
⋮	⋮	⋮	⋮
45	0.927714	0.971429	"qurious green ideas sweat profusely"
46	0.936800	1.000000	"furious green ideas sweat profusely"

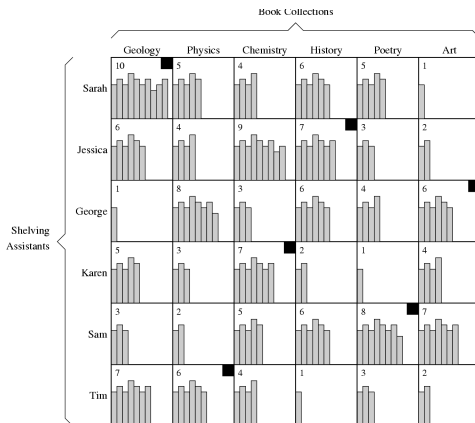
**Table 20.2** Evolving the text string "furious green ideas sweat profusely" from an initially random pool of strings: Fitness scores shown are for the raw fitness.



# Demo příklady

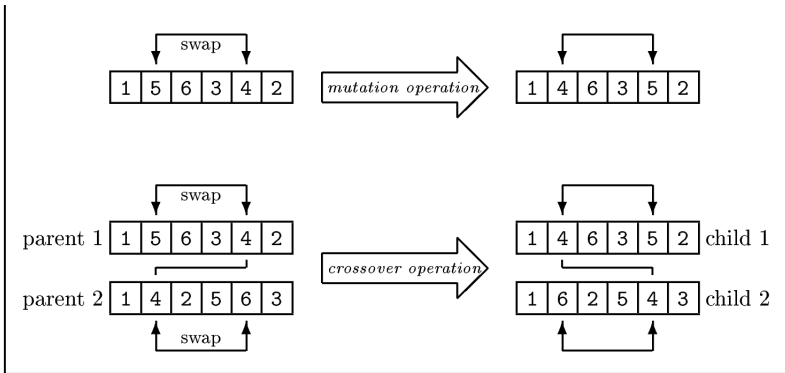
mnoho demo příkladů (Java applety) na Internetu — stačí hledat “Genetic algorithm”

# Optimalizační problém



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

# Optimalizační problém



**Figure 20.3** The crossover and mutation operations applied to candidate solutions of a combinatorial optimization problem

# Dilema vězně

- strategie které zohledňují posledních  $k$  kol můžeme jednoduše reprezentovat řetězcem; fitness = průměrný bodový zisk
- např. pro  $k = 1$  řetězec 5 znaků:
  - 1 tah v prvním kole
  - 2 co dělat když minule: oba spolupracovali
  - 3 co dělat když minule: já spolupracoval on zradil
  - 4 co dělat když minule: já zradil on spolupracoval
  - 5 co dělat když minule: oba zradili
- např. TFT je "SSZSZ"

# Dilema vězně

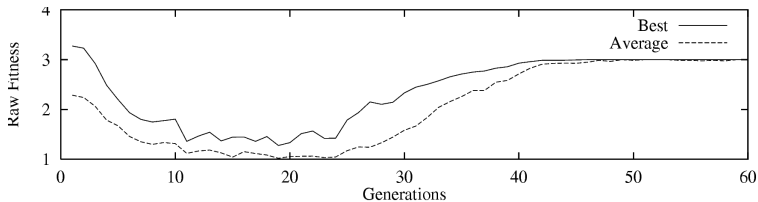


Figure 20.4 Average and best raw fitness scores for the IPD-playing GA

# Evoluční dilema vězně: Axelrodova studie

- otázka: jak moc byly výsledky turnajů ovlivněny tím, že lidé očekávali určité složení odeslaných strategií
- strategie uvažující poslední 3 tahy, začíná z náhodných
- strategie, které se vyvinou připomínají charakteristiky TFT — tj. dominance principů, na kterých je TFT založena není způsobena lidskými očekáváním, kulturními hodnotami, ...

# Další aplikace genetických algoritmů

- zejména optimalizační problémy: TSP, rozvrhy, protein folding
- učení a plánování: robotika, hry
- návrh: hardware, materiály

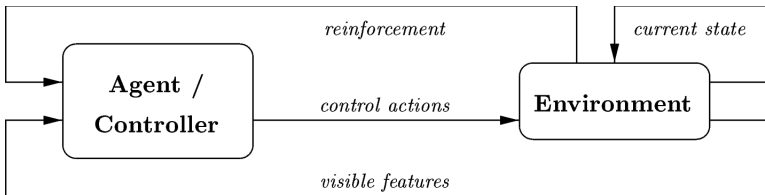
# Klasifikační systémy

(Classifier systems)

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

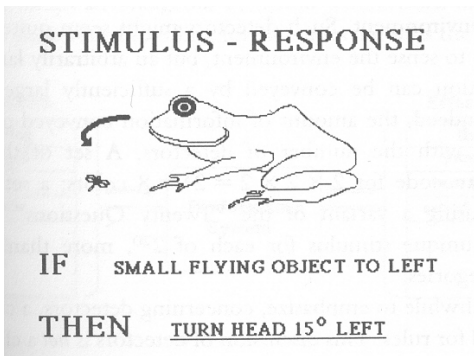


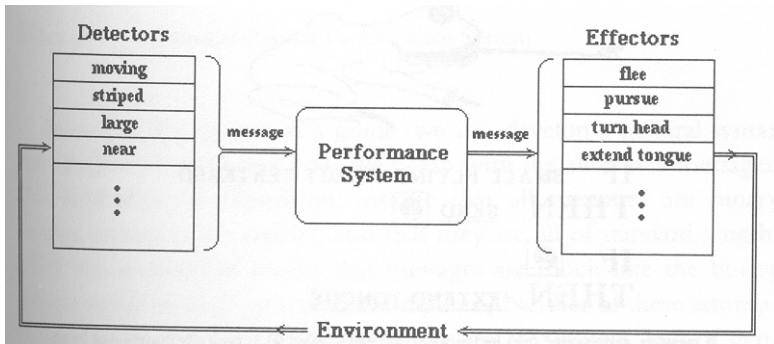
# Prostředí



**Figure 21.1** Adaptation through environmental reinforcement

# Příklad žába

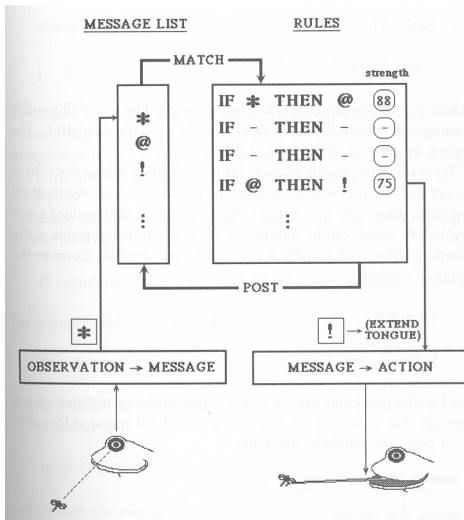




# Produkční systém

- pravidla typu  
if podmínka then akce
- podmínky i akce reprezentovány jako binární řetězce
- podmínky přichází z prostředí nebo jako následky jiných pravidel
- akce mohou vyvolávat další pravidla nebo měnit prostředí
- dovoleno použití divokých karet (#) v podmínkách

# Příklad žába II







# Bucket brigade

- pravidla mohou být konfliktní, soutěží spolu o to, které bude aplikováno
- každé pravidlo má „kapitál“ s jehož pomocí soutěží o to, aby bylo aplikováno (a platí za to)
- úspěšná aplikace vede k zisku odměny od prostředí
- část odměny je redistribuována pravidlům, které umožnili použití pravidla, které vedlo k odměně





# Classifier systems: vývoj nových pravidel

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

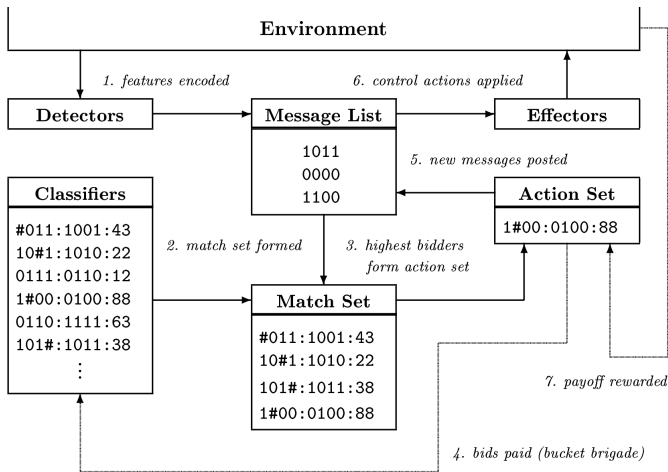


Figure 21.2 A classifier system interacting with its environment

# Příklad Woods

- zvíře na čtverečkovaném poli
  - '\*' zvíře
  - '.' volné pole
  - 'O' skála
  - 'F' jídlo
- zvíře vidí okolních 8 pozic, tj. detektory lze zakódovat pomocí 16 bitové řetězce (. = 00, O = 10, F=11)
- cílem je najít co nejrychleji jídlo

# Příklad Woods

```

.....
.OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF.
.OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO.
.OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO.
.....
.....
.OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF.
.OOO..OOO..OOO..OOO*.OOO..OOO..OOO..OOO..OOO..OOO..OOO.
.OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO.
.....
.....
.OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF..OOF.
.OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO.
.OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO..OOO.
.....

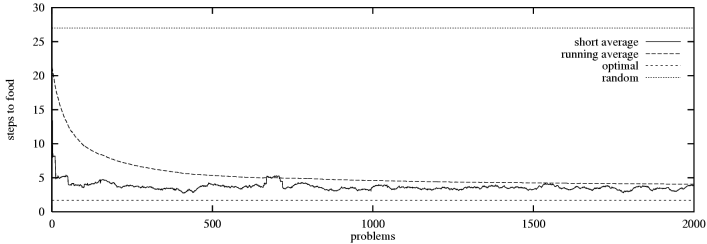
```

# Příklad Woods

Realizace experimentu (učení):

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

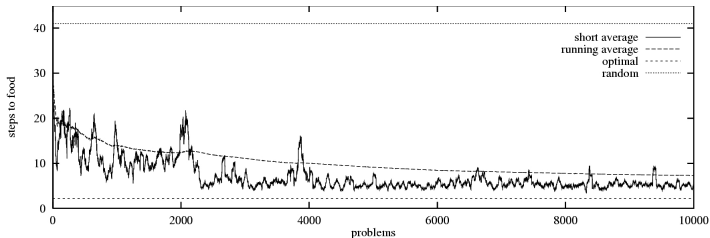
# Příklad Woods: výsledky







# Příklad Woods: výsledky II



# Deduktivní a induktivní uvažování

- **deduktivní** uvažování
  - z daných předpokladů se dle zákonů logiky odvozují logicky platné závěry (konkretizace)
  - modelování není velký problém (deduktivní uvažování lze vyjádřit formálně), hojně používáno např. v ekonomii
  - ovšem neodpovídá realitě - lidé používají deduktivní uvažování jen v omezené míře
- **induktivní** uvažování
  - zevšeobecňování, odvozování obecných zákonů z konkrétních příkladů, odhadování vývoje, ...
  - často a úspěšně používáno lidmi
  - jak modelovat?

# El Faron Bar

- (El Faron je Bar v Santa Fe, kde hrajou ve čtvrtek večer irskou hudbu)
- v okolí baru žije 100 lidí
- každý by rád zašel do baru, ale když je tam moc narváno, tak to za nic nestojí:
  - pokud je v baru méně jak 60 lidí, tak je lepší být v baru než doma
  - pokud je v baru více jak 60 lidí, tak je lepší být v doma než v baru
- všichni se rozhodují zaráz, žádná domluva

# Induktivní model rozhodování

Každý agent má několik **hypotéz** pro předpovídání počtu návštěvníků. Předpokládejme, že dosavadní sekvence byla:

44, 78, 56, 15, 23, 67, 84, 34, 45, 76, 40, 56, 22, 35

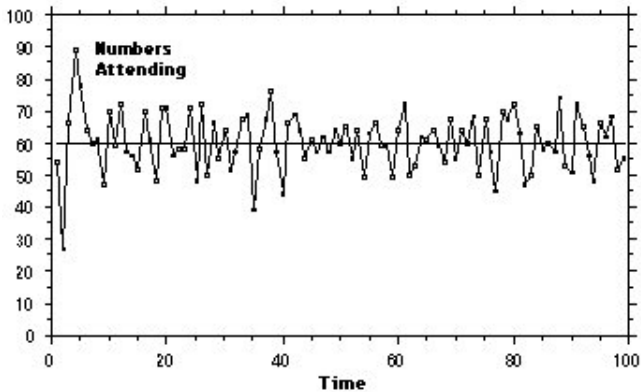
Příklady hypotéz:

- stejně jako minulý týden [35]
- zaokrouhlený průměr za minulé 4 týdny [49]
- stejně jako před dvěma týdny [22]
- zrcadlový obraz okolo 50 z minulého týdne [65]
- 42 [42]

# Induktivní model rozhodování

- pro každou hypotézu si agent pamatuje její **úspěšnost**
- vždy se rozhoduje podle aktuální nejúspěšnější hypotézy (případně průměruje několik nejúspěšnějších)
- hypotézy vygenerovány „ručně“, každému agentu náhodně přiřazeno několik z nich

# Simulace



# Výsledky a rozšíření

- počet návštěvníků osciluje okolo hodnoty 60
- induktivně uvažující agenti tedy řeší problém poměrně dobře
- možná rozšíření: tvorba nových hypotéz (genetický algoritmus), komunikace mezi agenty, ...

# Modelování a simulace: souvislosti

- 1 Biologicky inspirované *modely* (mozek, evoluce) jejichž *simulací* dostáváme nové metody řešení problémů (např. optimalizačních).
- 2 Možnost rozšíření ABM modelů o učení, adaptaci, vývoj, tj. větší věrohodnost, širší záběr ABM modelů.