





# Zajęcia

- Laboratoria: ćwiczenia, zadania programistyczne, mini-projekty
  - Zaliczenie: ważona suma ocen za zadania
- Wykłady z elementami interaktywnymi/ćwiczeniowymi.  
Ocena końcowa:
  - 80% Ocena z kolokwium
  - 20% Ocena z laboratoriów
  - +1% Obecność na wykładzie

# Spotkania

- 1 4.03,
- 2 11.03,
- 3 18.03,  
[Wielkanoc],
- 4 1.04,
- 5 8.04
- 6 15.04,
- 7 22.04,
- 8 29.04
- 9 6.05,
- 10 13.05,
- 11 20.05,
- 12 27.05 [26.05 Boże Ciało],
- 13 3.06,
- 14 10.06,
- 15 17.06 (kolokwium)

# Inteligencja Obliczeniowa vs. Sztuczna Inteligencja I

- Ang. Computational Intelligence (CI)
- Różne opinie na ten temat:
  - ① CI=AI: study of the design of **intelligent agents**. Intelligent agent is a system that acts intelligently: what it does is appropriate for its circumstances and its **goal**, it is **flexible** to changing environments and changing goals, it **learns** from experience, and it makes appropriate choices given perceptual limitations and finite computation.
  - ② AI: not natural, **engineered**; strongly oriented to **symbolic representations**, and reasoning in **top-down manner**  
CI: **non-symbolic, bottom-up** fashion
  - ③ CI: only **low-level numerical data**; does not knowledge in the AI sense.
  - ④ CI: (...) adaptation is arguably the most appropriate term for what computationally intelligent systems do. *Computation intelligence and **adaptation** are synonymous*

# Inteligencja Obliczeniowa vs. Sztuczna Inteligencja II

- ⑤ Central focus of AI: **emulating human behavior** by extracting rules and knowledge from **human experts**. Majority of AI has nothing with learning (ale  $ML \in AI$ ).
  - ⑥ CI: to adapt solutions to new problems and do not rely on **explicit human knowledge**
  - ⑦ CI=**soft computing**, Operations Research=**hard computing**
  - ⑧ **AI**: The science of creating a non-human intelligence with machines or computers  
**CI**: Combined fields of: Neural Computation, Evolutionary Computation, Fuzzy Computation, DNA Computing, Quantum Computing
- 
- CI: Birth as a field: IEEE World Congress on CI in 1994, Orlando, Florida.
  - Metody CI:
    - ① Neural Computation and Neural Networks
    - ② Evolutionary Computation and Genetic Algorithms



# Tematy

- 1 Wstęp [1]
- 2 Inteligentne agenty [1]
- 3 Niepewność w AI [2, 3]
  - Modelowanie niepewności [2]
  - Sieci bayesowskie, wnioskowanie probabilistyczne [3]
- 4 Estymacja stanu [5, 6]
  - Lokalizacja i filtrowanie w warunkach niepewności [5]
  - Filtr Kalmana i Ukryty Model Markov'a [6]
- 5 Planowanie w warunkach niepewności [7]
  - Problem Decyzyjny Markov'a (MDP i POMDP) [7]
- 6 Uczenie ze wzmocnieniem [8, 9, 10, 11]
  - Pasywne i Aktywne [8]
  - Aproksymacja i Generalizacja [9]
  - Studia przypadków [10]
  - LSPI, CB-AMPI, Sarsa, apprenticeship learning, inverse RL [11]
- 7 Programowanie genetyczne [Krzysztof Krawiec, 12, 13]



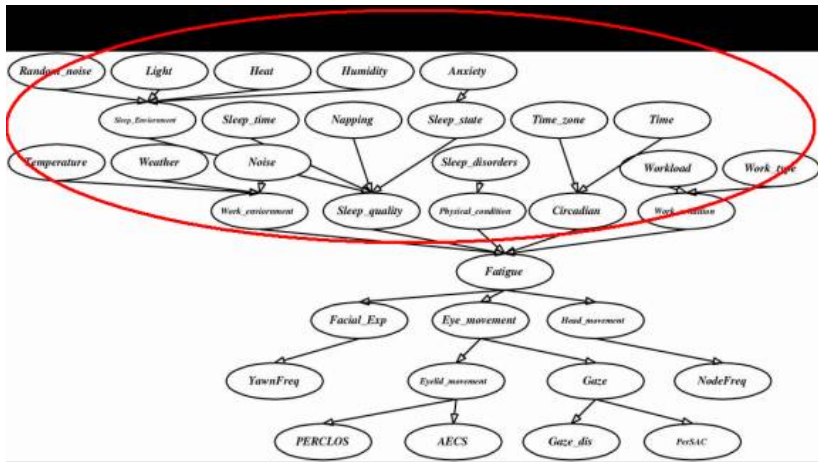






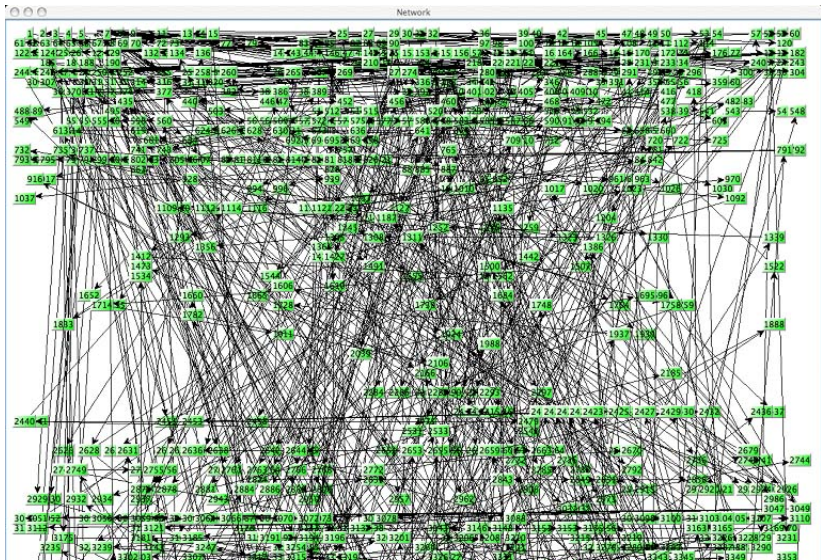
# Sieci baysowskie

- Zastosowania: modelowanie wiedzy (ekspertkiej lub nauczonej z danych)
- Modelowanie zmęczenia

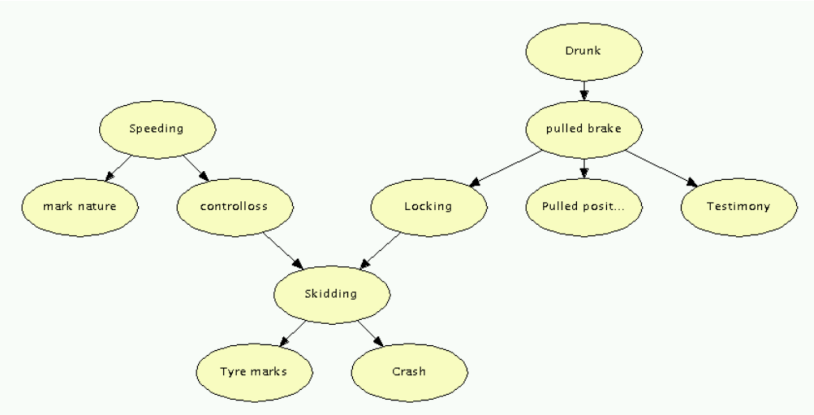


# Przykład: system rekomendacyjny

- Systemy rekomendacyjne (np. sklepy internetowe)



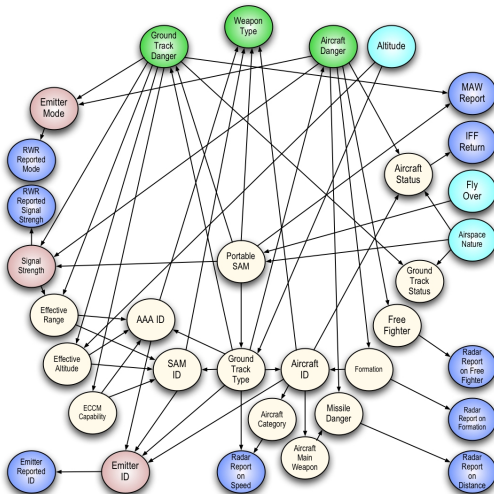
# Przykład: Powód wypadku samochodowego





# Przykład wojskowy

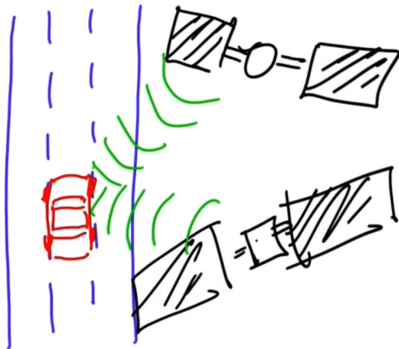
- Wsparcie dla decyzji pilota samolotu bojowego (Wise Pilot)







# Estymacja stanu (lokalizacja)

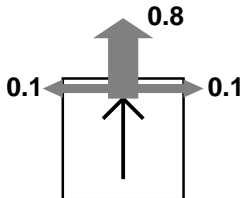
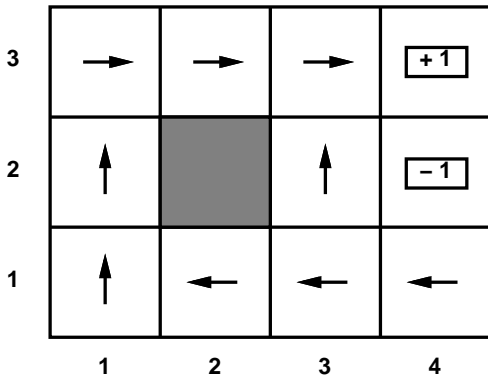


- Filtr histogramowy, Kalmana, cząsteczkowy (wnioskowanie probabilistyczne)

# Filtr cząsteczkowy

- Ukryty model Markov'a (prosta dynamiczna sieć bayesowska)
- <http://www.youtube.com/watch?v=ABzzFMzFE3Y>
- <http://robots.stanford.edu/movies/sca80a0.avi>

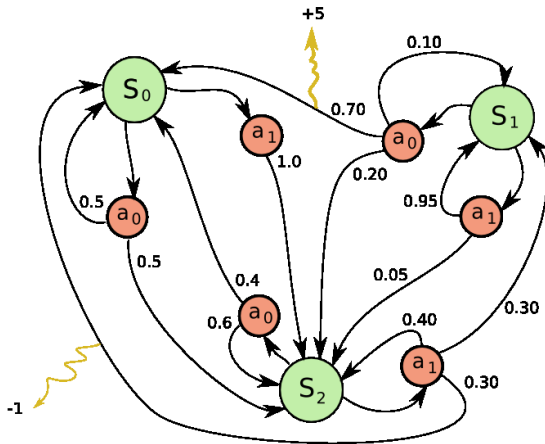
# Sekwencyjne problemy decyzyjne



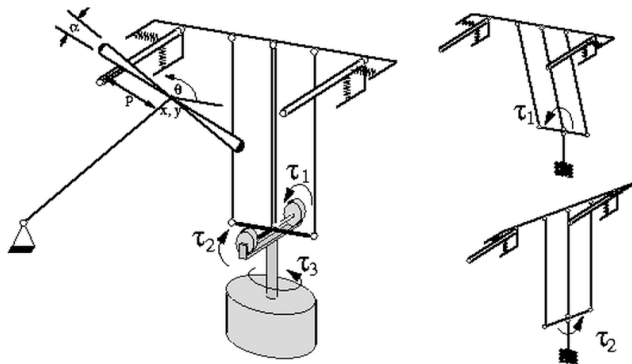


# Uczenie ze wzmocnieniem

- Jeśli prawd. albo nagrody są nieznane



# Przykład: Devil-sticking robot



- Devil Stick (pol. diabelski kijek)
- 6-wymiarowa przestrzeń stanów, tylko 40 prób uczących (lepiej niż człowiek)

# Autonomous Helicopter via Reinforcement Learning [Ng, et al.]

- <http://www.youtube.com/watch?v=kN6ifrqwIMY>
- <http://www.youtube.com/watch?v=VCdxqn0fcnE>



# TD-Gammon [Tesauro]

- Tryktrak (Backgammon)
- Uczenie poprzez grę z samym sobą (300tys-1.5mln rozgrywek)
- TD-Gammon gra na poziomie mistrzowskim => Rozwój uczenia ze wzmocnieniem
- Metoda różnic czasowych (Temporal Difference (TD))
- Sieć neuronowa

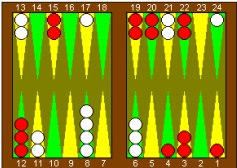


Figure 3. A complex situation where TD-Gammon's positional judgment is apparently superior to traditional expert thinking. White is to play 4-4. The obvious human play is 8-4\*, 8-4, 11-7, 11-7. (The asterisk denotes that an opponent checker has been hit.) However, TD-Gammon's choice is the surprising 8-4\*, 8-4, 21-17, 21-17! TD-Gammon's analysis of the two plays is given in Table 3.

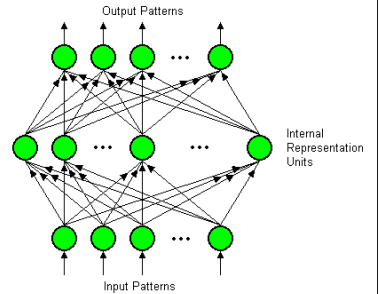
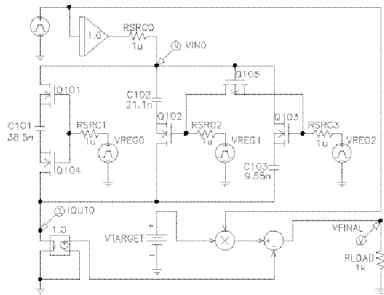
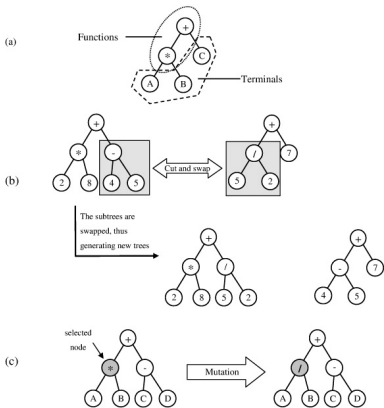


Figure 1. An illustration of the multilayer perceptron architecture used in TD-Gammon's neural network. This architecture is also used in the popular backpropagation learning procedure. Figure reproduced from [9].

# Programowanie genetyczne

## ● Ewolucja układów bramek logicznych [Koza]



# Naprawianie błędów w kodzie

- Automatyczne naprawianie błędów w programach w C [Weimer et al.]

```
1 void zunebug(int days) {
2   int year = 1980;
3   while (days > 365) {
4     if (isLeapYear(year)){
5       if (days > 366) {
6         days -= 366;
7         year += 1;
8       }
9     }
10    }
11  }
12  else {
13    days -= 365;
14    year += 1;
15  }
16 }
17 printf("current year is %d\n", year);
18 }
```

```
1 void zunebug_repair(int days) {
2   int year = 1980;
3   while (days > 365) {
4     if (isLeapYear(year)){
5       if (days > 366) {
6         // days -= 366; // repair deletes
7         year += 1;
8       }
9     }
10    }
11  }
12  else {
13    days -= 366; // repair inserts
14  }
15  }
16 }
17 printf("current year is %d\n", year);
18 }
```

