



Ruch drogowy a sztuczna inteligencja

Paweł Gora

29.08.2017, LVI Szkoła Matematyki Poglądowej

Problemy



4 mld PLN / rok
7 największych miast



Na świecie rocznie:

- 1 300 000 ofiar śmiertelnych
- ponad 20 milionów rannych

Rodzaje badań

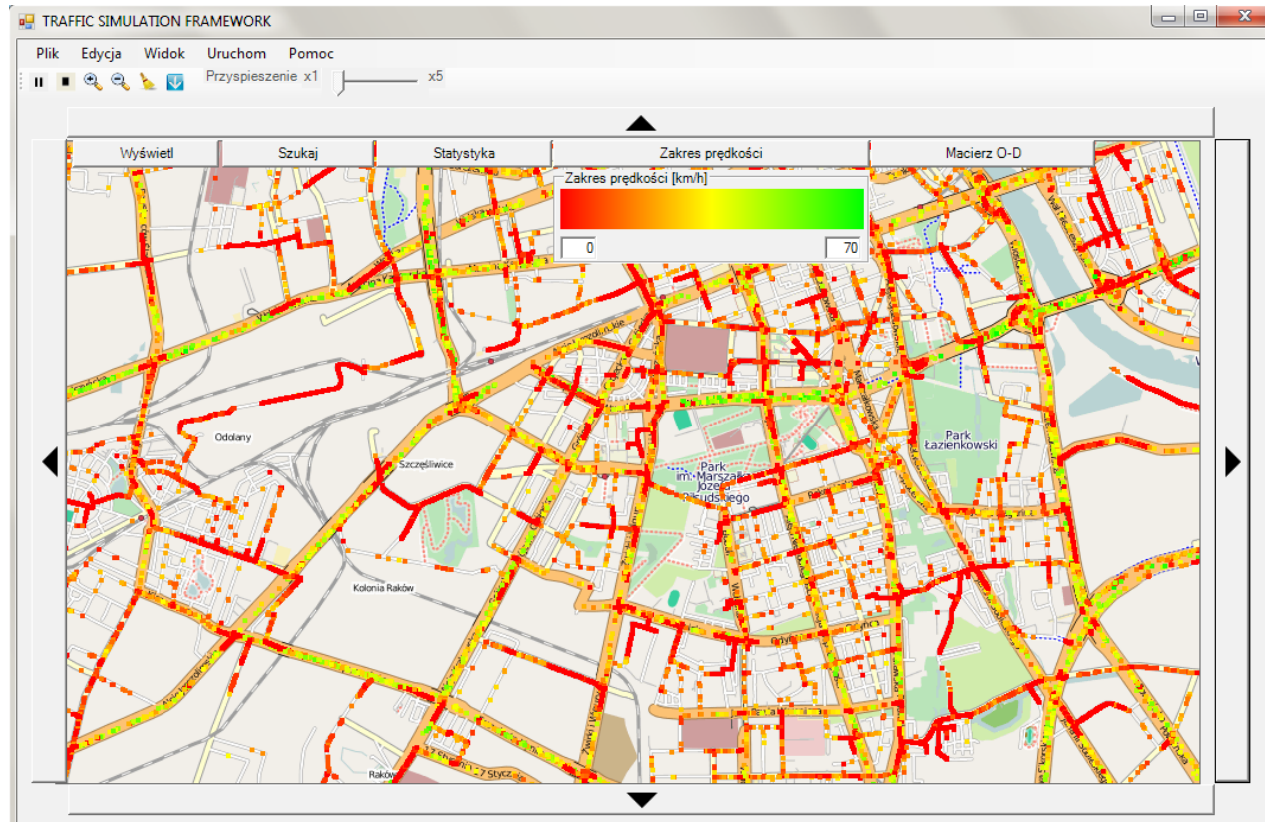
Modelowanie	Analiza
Predykcja	Optymalizacja

Modelowanie

Kalibracja:

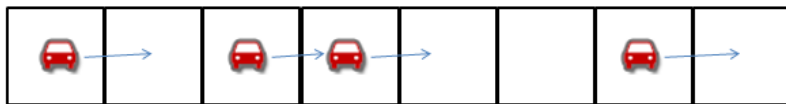
- Klastrowanie (np. k-NN, k najbliższych sąsiadów)
- Algorytmy ewolucyjne
- Sieci neuronowe
- Tabu search
- ...

Model symulacyjny

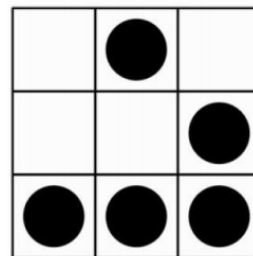


Model Nagela-Schreckenberga

- ▶ 1-wymiarowy model mikroskopowy (probabilistyczny automat komórkowy)
- ▶ Długość komórki ~7.5m
- ▶ Czasy i prędkości są dyskretne
- ▶ Prędkość pojazdu = liczba komórek, o które się przemieszcza (max 5)
- ▶ Każda komórka może być pusta lub zajęta przez max 1 pojazd
- ▶ Pojazdy są nierozróżnialne
- ▶ Stan komórki = null lub prędkość pojazdu



1	<u>null</u>	0	2	<u>null</u>	<u>null</u>	1	null
---	-------------	---	---	-------------	-------------	---	------

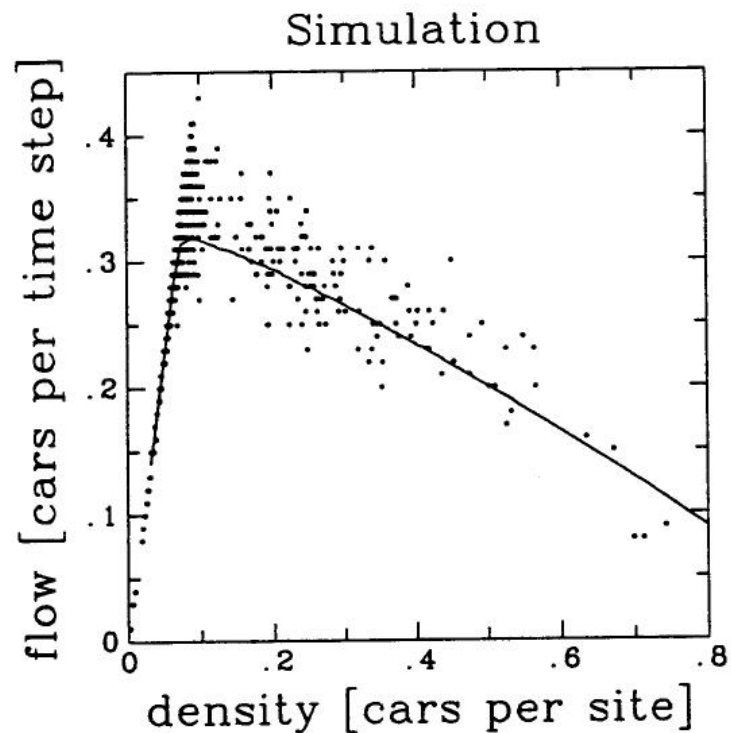
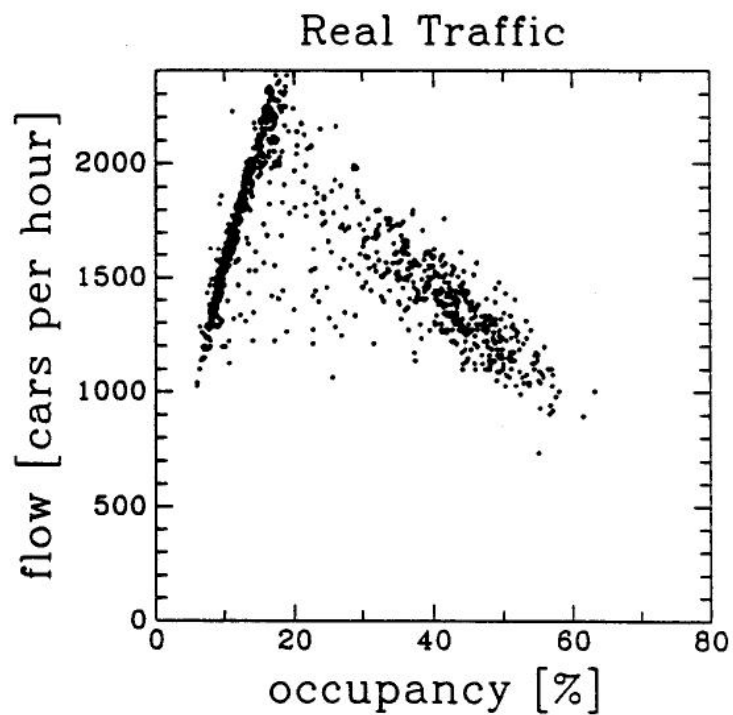


Model Nagela-Schreckenberga

Reguły przejścia:

1. $V_i := \min\{V_i + 1, V_{MAX}\}$ (przyspieszenie)
2. $V_i := \min\{V_i, d_i\}$ (bezpieczeństwo)
(d_i - dystans do najbliższego pojazdu przed i)
3. $V_i := \max\{0, V_i - 1\}$ z prawdopodobieństwem p (randomizacja) (kluczowa dla spontanicznego formowania się korków)
4. Pojazd i przemieszcza się do przodu o V_i komórek (przemieszczenie)

Model Nagela-Schreckenberga



Uogólnienie - sieć drogowa w mieście

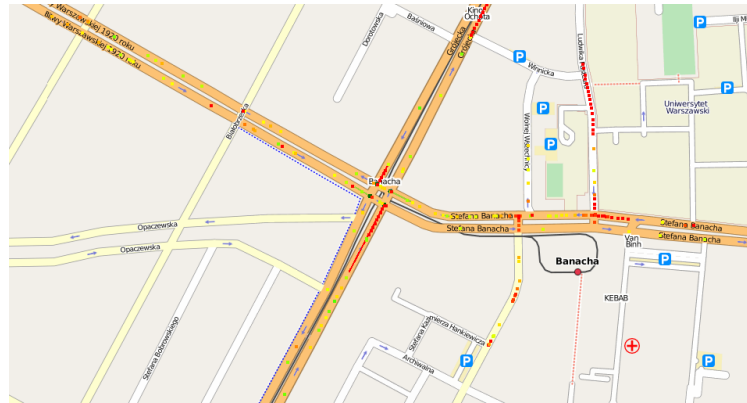
- ▶ Sieć drogowa - graf skierowany
- ▶ Sygnalizacje świetlne w niektórych wierzchołkach
- ▶ Krawędzie podzielone na taśmy, podzielone na komórki
- ▶ Ruch po krawędziach zgodnie z regułami Na-Sch z pewnymi modyfikacjami
- ▶ Redukcja prędkości przed skrzyżowaniem / zakrętem
- ▶ Różne profile kierowców (np. max prędkość)
- ▶ Różne typy dróg (wiele pasów, różne limity prędkości)
- ▶ Profile rozkładów punktów startowych i końcowych

Analiza

- Np. klasyfikacja stanów ruchu drogowego



The Observer



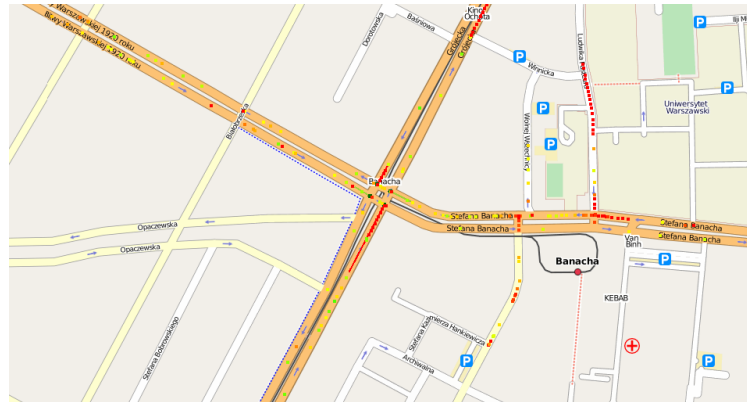
Korek

Analiza

- Np. klasyfikacja stanów ruchu drogowego



The Observer



Korek



	A	B	C	D	E	F	G	H	
1	Time	ICD	NodeBegin	NodeEnd	Distance	Velocity	Latitude	Longitude	
2	5	180577	31930544	31930533	0.045	49.99	52.147598	49761317	21.0296434316288
3	5	180619	32333890	32215288	0.045	48.95	52.205567	1688495	21.0804926821586
4	5	180649	31156950	31156949	0.045	49.91	52.128385	3682484	21.1966988249598
5	5	180738	216205665	252338939	0.045	49.82	52.3584825	496801	21.1342807446003
6	5	180855	216205665	252338939	0.035	50.52	35867232	47158	21.1343484847423
7	5	180956	131642280	131642282	0.044	46.83	52.2555962	395403	21.1502231597833
8	5	181050	216205665	252338939	0.035	48.89	52.358574	4554871	21.1343433237868
9	5	181343	131642280	131642282	0.033	14.82	52.255635	30330191	21.156349446218
10	5	181517	36307573	36307574	0.007	0.52	263407	879532	21.0139406106401
11	5	181538	132460751	27282383	0.034	50.52	1442173895	647	20.9088220106712
12	5	181601	216205665	252338939	0.029	0.52	35862308	42905	21.1343883141395
13	4	181603	13674742	13674279	0.035	48.85	52.1191888	45162	21.11511596888792

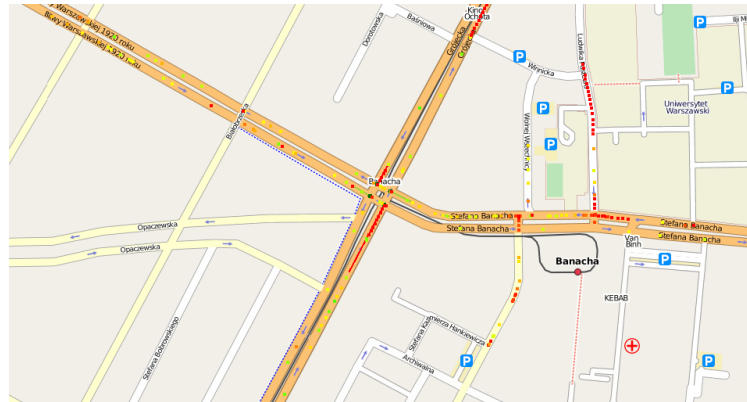


Analiza

- Np. klasyfikacja stanów ruchu drogowego



The Observer



Korek



	A	B	C	D	E	F	G	H	
1	Time	ICD	NodeBegin	NodeEnd	Distance	Velocity	Latitude	Longitude	
2	5	180577	31930544	31930533	0.045	48.99	52.147569	49763117	21.0296434316288
3	5	180619	32333890	32215288	0.045	48.95	52.205567	1688495	21.0804926821568
4	5	180649	31156950	31156949	0.045	49.91	52.128385	3682484	21.1966988249598
5	5	180738	216205665	252338939	0.045	49.82	52.358492	25496801	21.1342807446003
6	5	180855	216205665	252338939	0.035	50.52	35867232	47158	21.1343484847423
7	5	180956	131642280	131642282	0.044	46.83	52.255592	395403	21.1502231597833
8	5	181050	216205665	252338939	0.035	48.89	52.358574	45548719	21.134343323788
9	5	181343	131642280	131642282	0.033	14.82	52.255635	30330191	21.156349446218
10	5	181517	36307573	36307574	0.007	0.52	283407	879532	21.0139406106401
11	5	181538	132460751	27282383	0.034	50.52	144217	3895647	20.9088220106712
12	5	181601	216205665	252338939	0.029	0.52	358623	30842905	21.1343883141395
13	5	181603	1367474	1367475	0.035	48.85	52.119188	45142	21.1151159688874



Predykcja

- Krótkoterminowa (np. 10 minut) / długoterminowa (np. następny dzień)
- Natężenia ruchu / czasy przejazdu
- Warunki typowe / warunki nietypowe (np. wypadek, remont, impreza masowa, zła pogoda)

Jeżeli są sytuacje typowe, dobrym podejściem zazwyczaj są rekurencyjne sieci neuronowe (mogą w nich występować cykle), np. LSTM (Long short term memory).

Predykcja „what-if”

- **Co będzie, jeśli** zmienimy ustawienie konfiguracji świetlnej ?
- Mamy ustalone punkty początkowe i końcowe, trasy i reguły ruchu
- Zmieniają się jedynie ustawienia świateł
- Chcemy policzyć np. łączne czasy czekania pojazdów

Ustawienie sygnalizacji świetlnej

- Sygnalizacja = obiekt o atrybutach:
 - Długość fazy światła **zielonego** (np. 58 sekund)
 - Długość fazy światła **czerrwonego** (np. 62 sekundy)
 - Offset = przesunięcie w fazie (liczba ze zbioru {0, 1, 2, ..., 119})

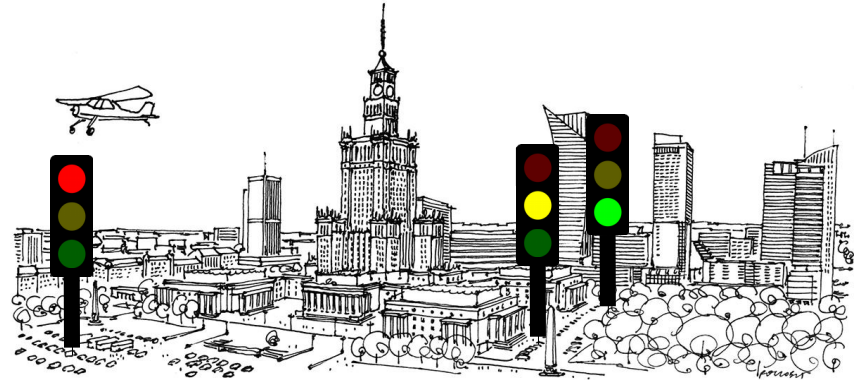
Ustawienie sygnalizacji świetlnej

- Sygnalizacja = obiekt o atrybutach:
 - Długość fazy światła **zielonego** (np. 58 sekund)
 - Długość fazy światła **czerrwonego** (np. 62 sekundy)
 - Offset = przesunięcie w fazie (liczba ze zbioru $\{0, 1, 2, \dots, 119\}$)
- W Warszawie mamy ok. 800 skrzyżowań ze światłami
 - Wystarczy ustawić po 1 sygnalizacji na każdym z nich (pozostałe powinny być zsynchronizowane, aby zapewnić bezpieczeństwo)
 - To daje 120^{800} możliwych ustawień offsetów
 - **Chcielibyśmy znaleźć najlepsze ustawienie**

Ciekawe porównanie



Liczba możliwych rozgrywek w grze w Go: 10^{700}

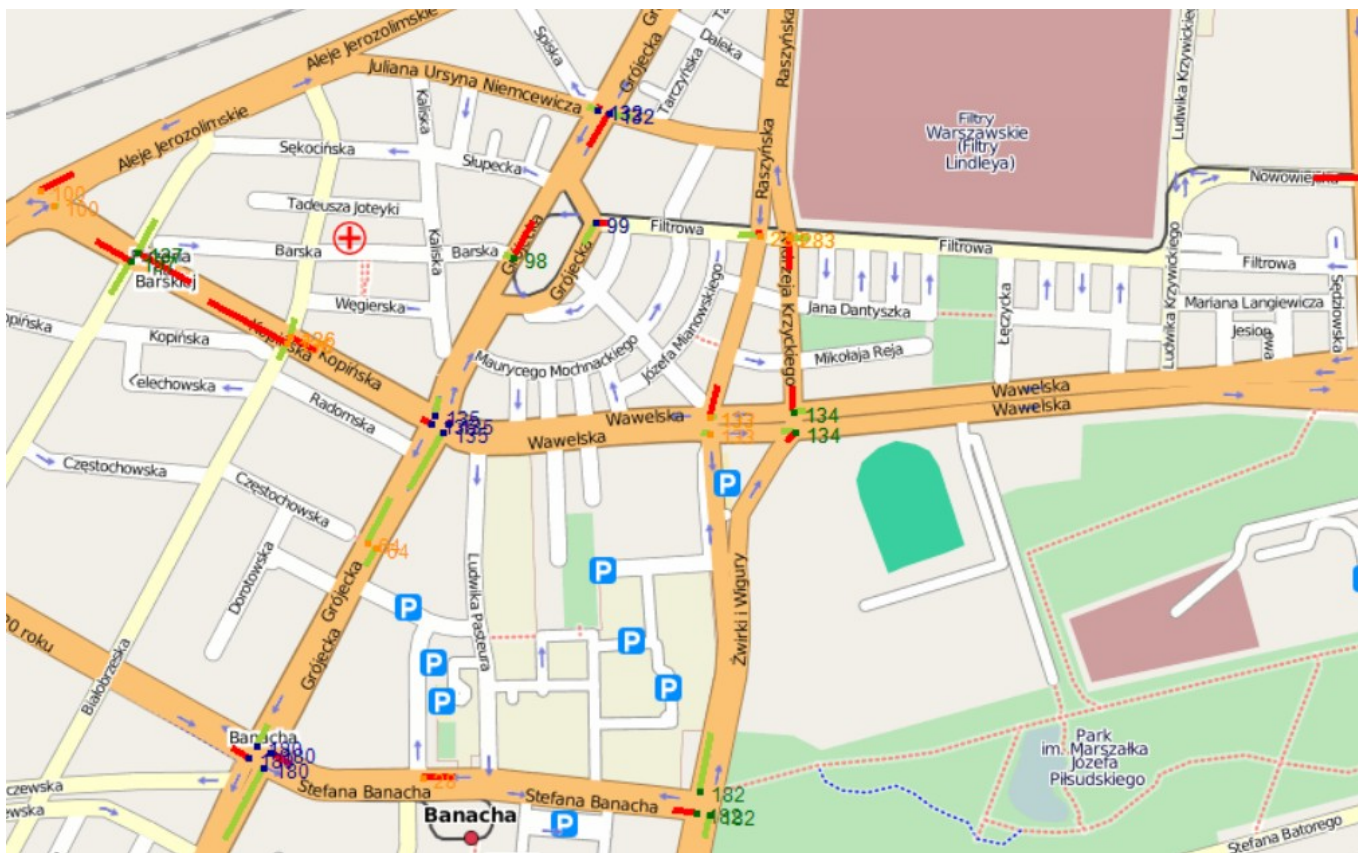


Liczba możliwych ustawień offsetów: 120^{800}

Ewaluacja ustawień świateł

- Dla każdego ustawienia świateł możemy obliczyć różne parametry ruchu poprzez symulację komputerową (np. łączny czas czekania)
- 1 symulacja na standardowym komputerze trwa kilka minut
- Może da się w jakiś sposób przybliżyć wynik?

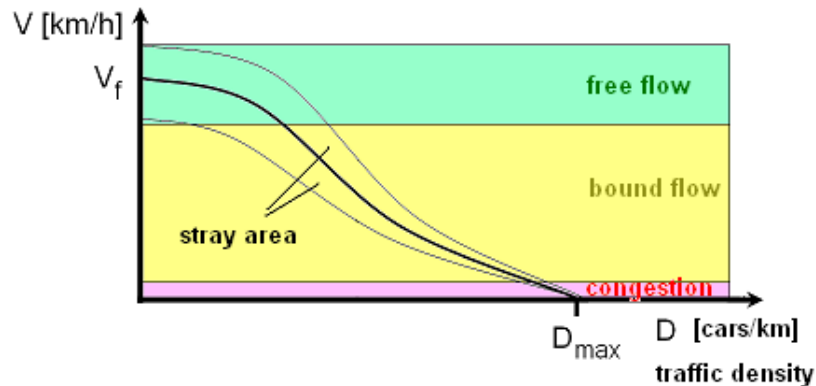
Ewaluacja ustawień świateł



Stara Ochota
15 skrzyżowań ze światłami

Przybliżanie wyników symulacji

- Model analityczny: szacowanie czasów przejazdu na podstawie długości odcinków i liczby pojazdów
- Relacja gęstość - prędkość

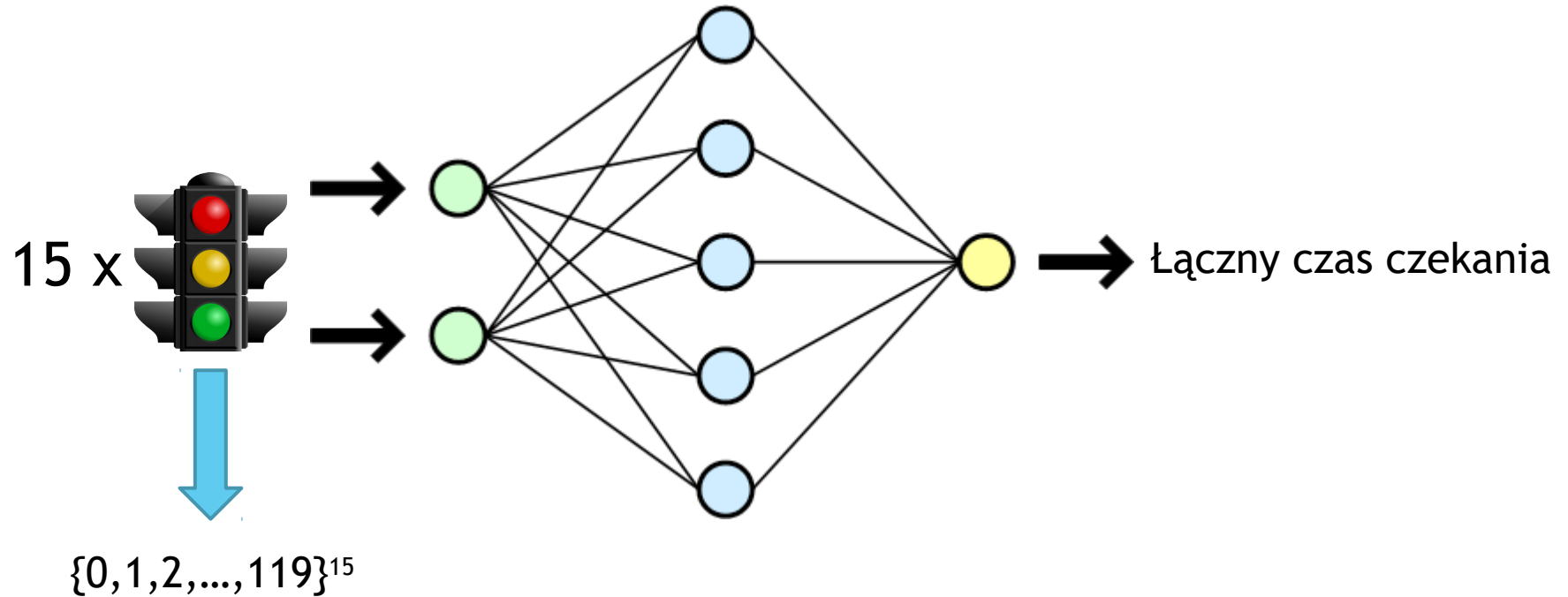


- Efekt: czas obliczeń: ~2 sekundy, dokładność: ~20%
- Wniosek: przybliżenie jest „za proste”

Przybliżanie wyników symulacji

- Podejście machine learning:
 - Mamy funkcję $F : U \rightarrow N$ (U - ustawienia świateł, $|U| = 120^{15}$,
 N - łączne czasy czekania (liczby naturalne))
 - Generujemy pewien zbiór treningowy (np. 10K-100K symulacji)
 - Trenujemy na nim alg. machine learning, aby przybliżyły wyniki symulacji

Przybliżanie wyników symulacji



Przybliżanie wyników symulacji

Eksperymenty:

- ▶ Wygenerowałem ponad 100 000 wyników symulacji (lab MIMUW, klaster HPC w Rzeszowie, Azure), największy czas czekania: 62911s, najkrótszy: 35831s
- ▶ Trenowanie sieci:
 - ▶ Powstał program **TensorTraffic** (korzysta z biblioteki TensorFlow)
 - ▶ Funkcja kosztu: RMSE
 - ▶ Adam optimizer
 - ▶ Batch: 100
 - ▶ Grid search: 12 konfiguracji (learning rate: {0.1, 0.01, 0.001}, sieci: [100, 100], [100, 100, 100], [200], [200, 200])
- ▶ Średni błąd przybliżeń: 1.56%, max błąd: 9.18%
- ▶ Najlepsza sieć: [100, 100, 100], lr = 0.01

Przybliżanie wyników symulacji

Kontynuacja badań - dużo więcej parametrów:

- ▶ Sieci: [100, 100, 100], [100, 200, 100], [200, 300, 200], [300, 400, 300], [100, 150, 200, 150, 100], [50, 100, 150, 200, 150, 100, 50]
- ▶ Learning rate: 0.1, 0.01, 0.001, 0.0001
- ▶ Dropout: 0.05, 0.1, 0.15, 0.2
- ▶ Zbiór treningowy: 81920, 30720, 10240
- ▶ „Mechanizm zatrzymania”

Przybliżanie wyników symulacji

Kontynuacja badań - dużo więcej parametrów:

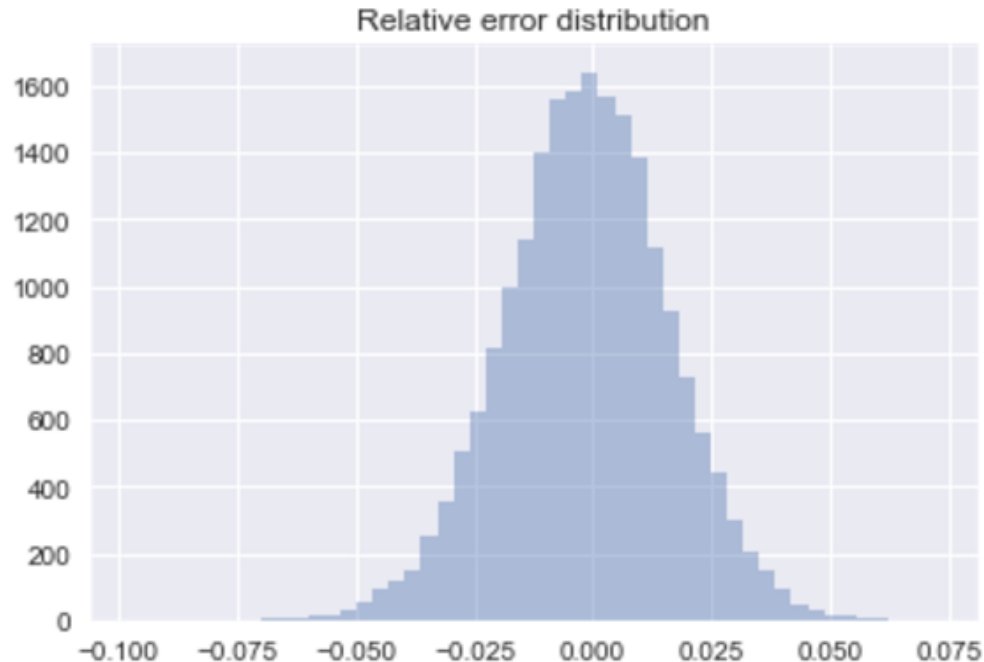
▶ Wnioski:

- Najlepszy model: [100, 200, 100], lr= 0.01, dropout=0.05, zbiór treningowy: 10240
- Średni błąd: 1.2%, błąd maksymalny: 6.8%
- Czas inferencji: ~ 0.2 sekundy

Przybliżanie wyników symulacji

Kontynuacja badań - dużo więcej parametrów:

- ▶ Rozkład błędu na zbiorze testowym (14634 przypadki):



Uzasadnienie teoretyczne

Twierdzenie o uniwersalnej aproksymacji:

Każda funkcja ciągła na podziorze zwartym \mathbb{R}^n może być przybliżana z dowolną dokładnością siecią neuronową z 1 warstwą ukrytą.

(Twierdzenie nie mówi jak duża musi być warstwa ukryta, ile czasu trzeba trenować sieć, jak duży musi być zbiór treningowy itp., ale sugeruje, że sieci neuronowe są dobrymi aproksymatorami funkcji ciągłych)

U - zbiór możliwych ustawień świateł (dyskretny)

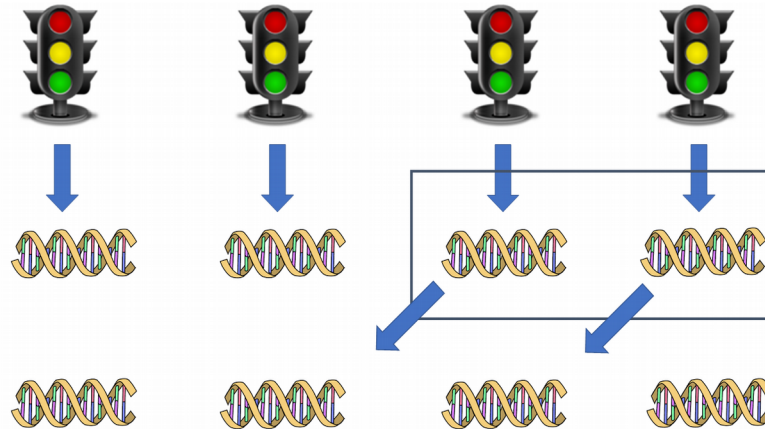
N - zbiór liczb naturalnych

F: U → N

Funkcję F można uciągnąć, aby dało się zastosować twierdzenie. Ale to jeszcze nie oznacza, że da się efektywnie wytrenować sieci.

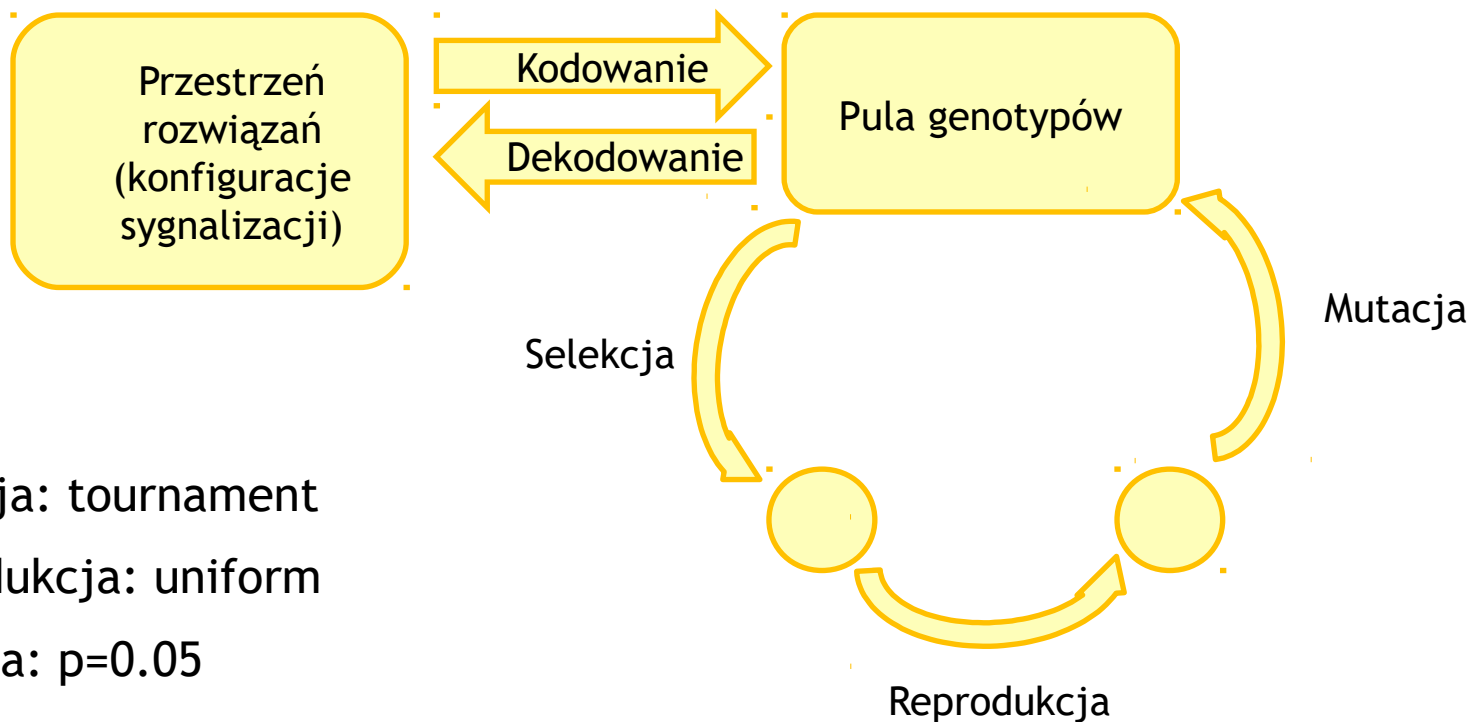
Optymalizacja ruchu

- Możemy już szybko ewaluować czasy czekania pojazdów (jako funkcja konfiguracji offsetów świateł), więc możemy ich sprawdzić sporo.
- Alg. genetyczny znajdujący coraz lepsze ustawienia świateł dla zadanej sytuacji



Optymalizacja ruchu

- Algorytm genetyczny



Selekcja: tournament

Reprodukcja: uniform

Mutacja: $p=0.05$

Optymalizacja ruchu

Algorytm genetyczny

- Dla populacji 100-elementowej po 30 iteracjach udaje się znajdować genotyp lepszy o ok. 20-30% od najlepszego genotypu w początkowej populacji
- Czas: ok. 800 sekund (< 30 sekund z GPU)

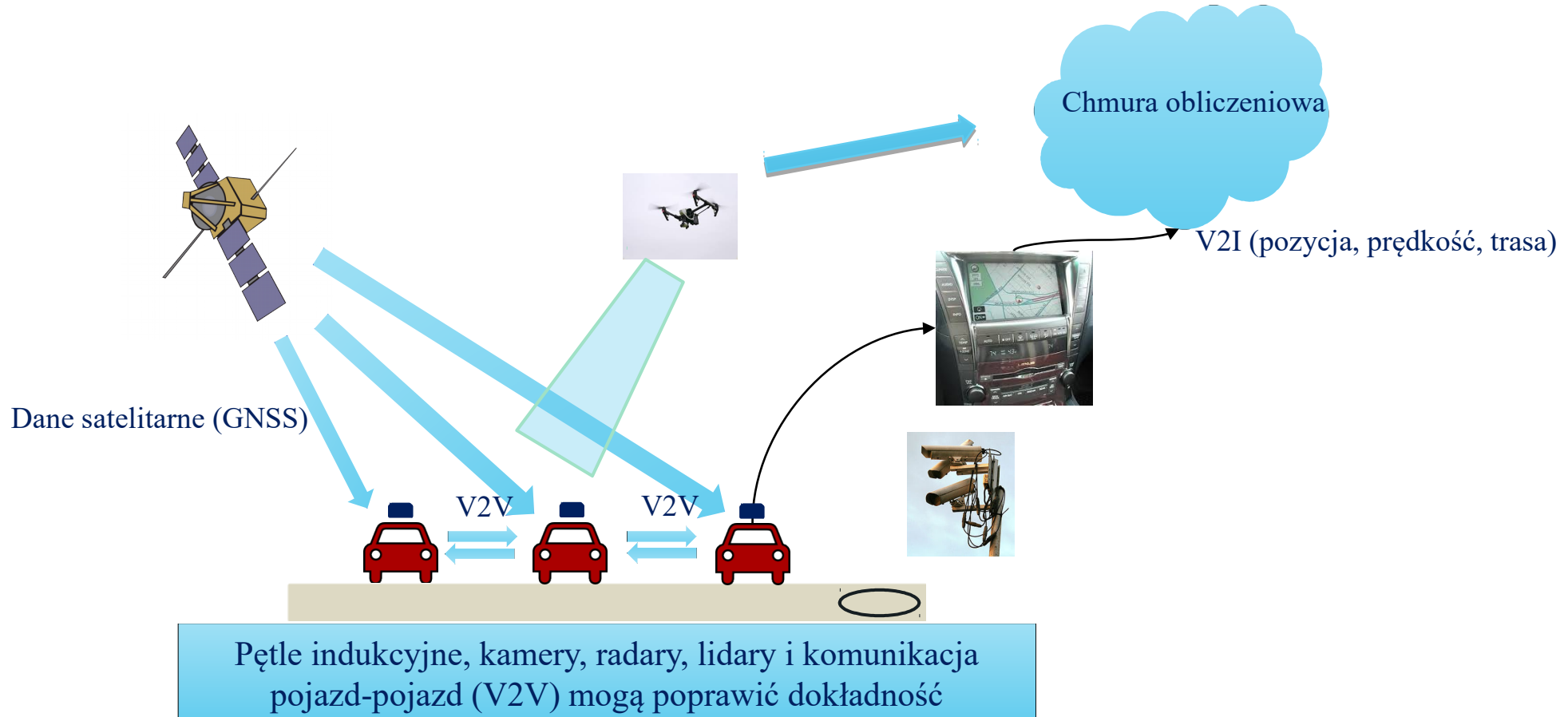
Porównanie:

- Dla (losowo wybranych) 10 mln ustawień najlepsze z nich okazało się gorsze o ok. 2% (czas obliczeń: ~10 minut z GPU)

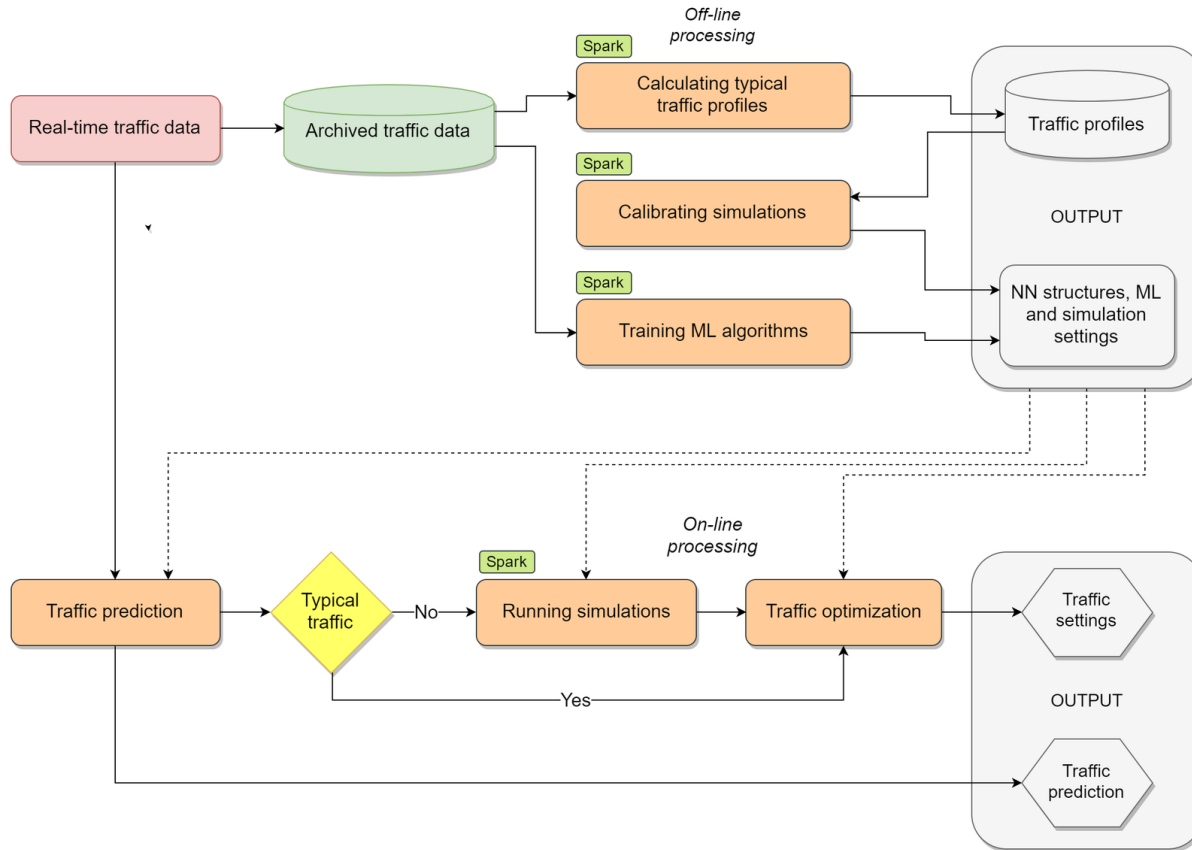
Adaptacyjna optymalizacja ruchu drogowego

- ▶ Trenujemy sieci neuronowe i znajdujemy dobre ustawienia „offline”, dla sytuacji typowych
- ▶ W czasie rzeczywistym: wykrywamy stany ruchu/ robimy predykcję
- ▶ Jeśli wykryjemy sytuację nietypową, niepożądaną (np. wypadek / zamknięta droga):
 - ▶ Dotrenowujemy sieć neuronową (transfer learning)
 - ▶ Znajdujemy lepsze ustawienie dla danej sytuacji

System zarządzania ruchem – zbieranie danych



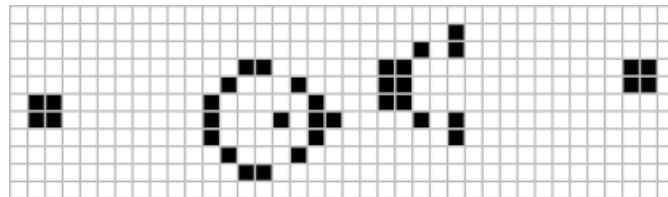
System zarządzania ruchem



„MELODIC” H2020

Przybliżanie symulacji ruchu drogowego – dlaczego to działa?

- Nie przybliżamy rzeczywistego ruchu tylko pewien jego model oparty na automatach komórkowych
- Być może to automaty komórkowe da się łatwo przybliżyć sieciami neuronowymi?
- Pomysł: spróbujemy przybliżyć inne automaty komórkowe (np. „Gra w życie” Conwaya, „Reguła 110”, automaty modelujące rozwój nowotworu)



Przybliżanie innych automatów komórkowych

- Reguła 110 i „Gra w życie” to automaty Turing-zupełne
- Czy możemy przybliżać ich ewolucję? Co z tego wynika?
- Co jeśli moglibyśmy przybliżać rozwój nowotworu?

Analogia:

- ruch drogowy → rozwój nowotworu
- konfiguracje świateł → terminy podawania radioterapii

Dziękuję za uwagę!

Paweł Gora

p.gora@mimuw.edu.pl

<http://www.mimuw.edu.pl/~pawelg>

„Logic can get you from A to B, imagination will take you everywhere”

A. Einstein

„Sky is not the limit”

