Европейски социален фонд
Европейски съюз

Изкуствени невронни мрежи и генетични алгоритми при игрите с открити условия

ас. инж. Тодор Балабанов
ИИКТ-БАН

София - 2013

Благодарности

Настоящата разработка е финансирана от Европейския социален фонд и Република България, Оперативна програма "Развитие на човешките ресурси" 2007-2013 г., Договор № BG051PO001-3.3.06-0048 от 04.10.2012 г.

Съдържание

- Игри с открити условия
- Изкуствени невронни мрежи
- Генетични алгоритми
- Експериментален модел

Игри с открити условия

- Цялата налична информация е достъпна за всички играчи
  - Най-популярните игри
    - Шах-мат
    - Ревърси (Отело)
    - Го
    - Морски шах

Игри със скрити условия

- Частично информацията не е известна на играчите
  - Най-популярни
    - Бридж
    - Покер
    - Сантасе

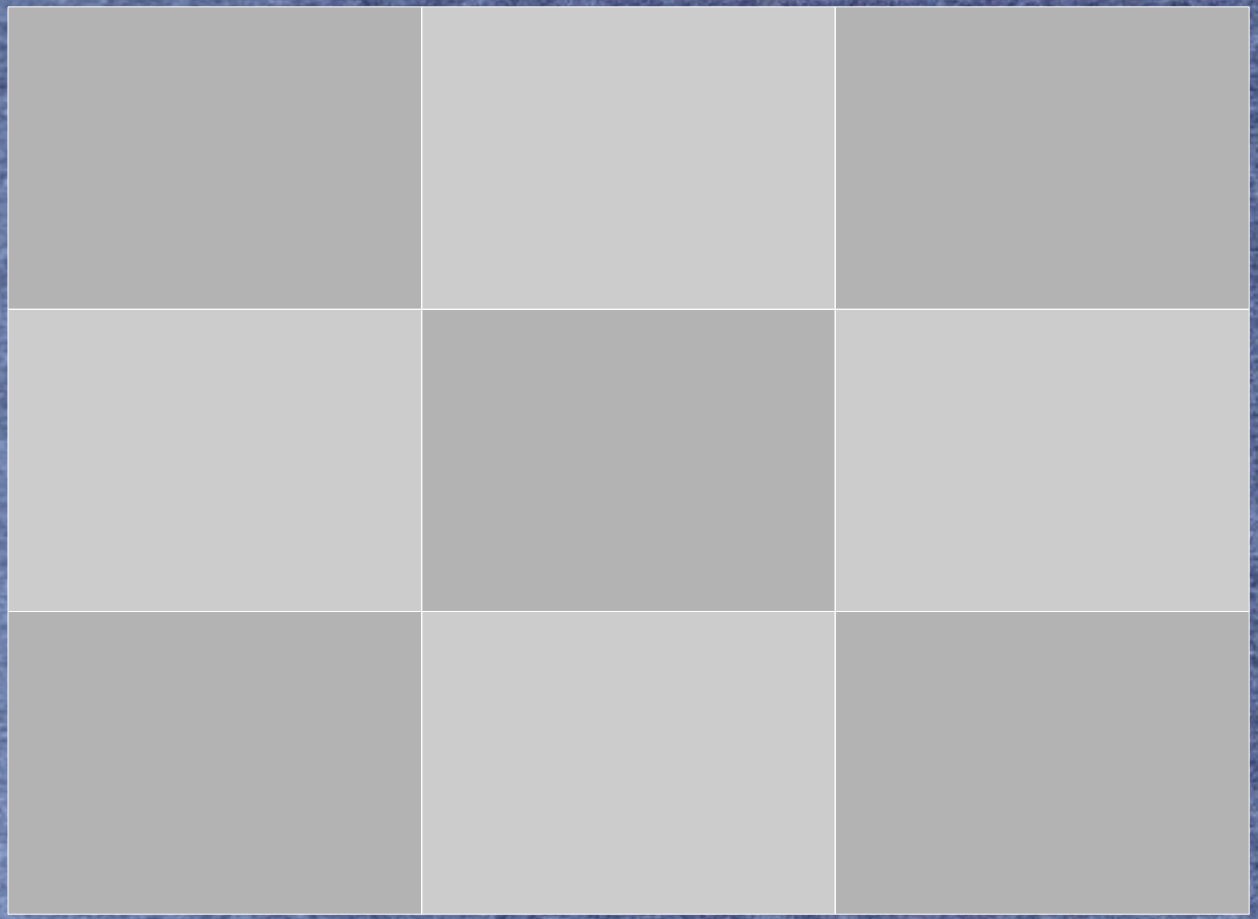
Морски шах

- Играе се на игрално поле 3 x 3 клетки
- Двама играчи
  - Играчите се редуват
  - Единия играч играе с X другия с O
- Целта е да се подредят три последователни символа
  - По някой от трите хоризонтала
  - По някой от трите вертикала
  - По някой от двата диагонала

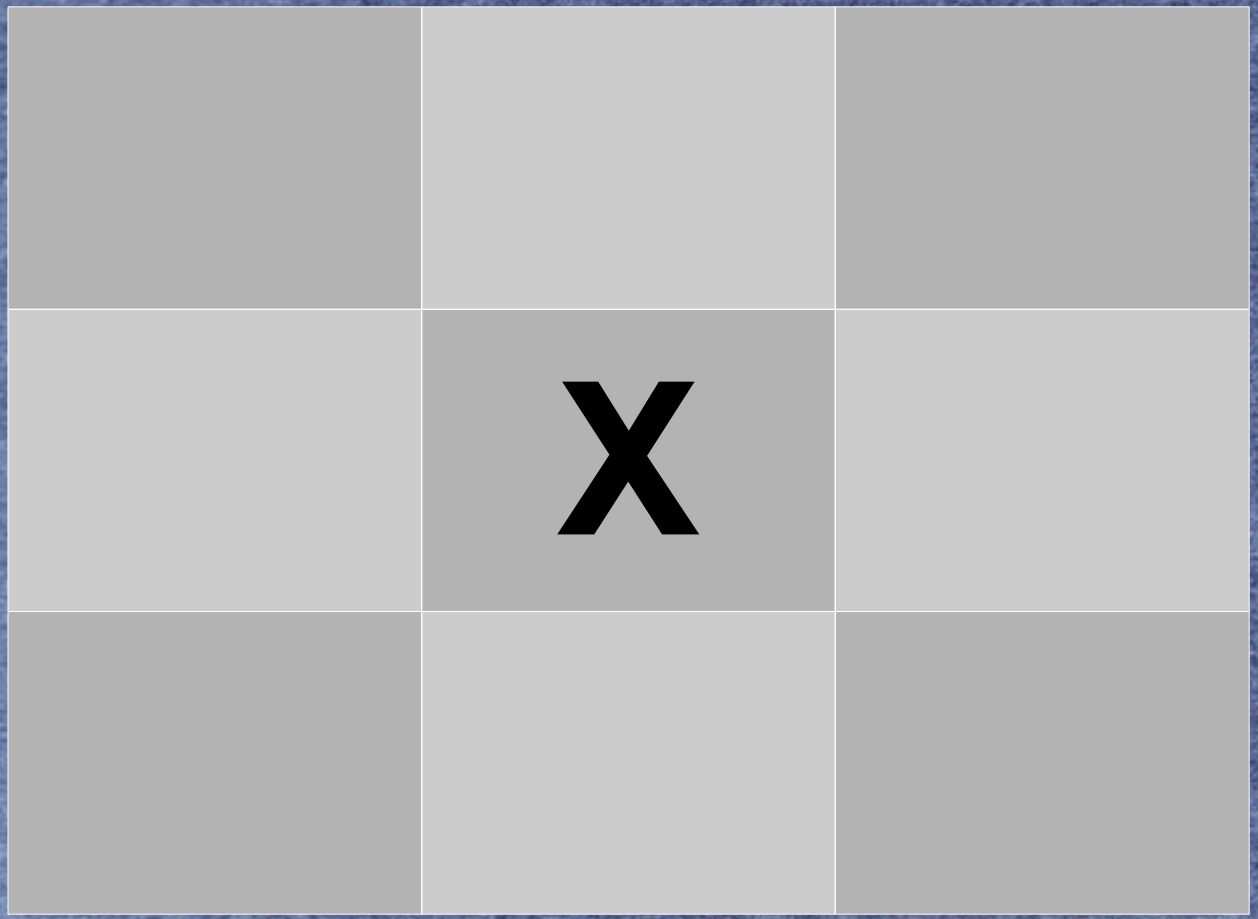
Морски шах - анализ

- Размер на игралното поле - 9 клетки
- Средна продължителност на играта - 9 хода
- Сложност - PSPACE Complete
- Брой Възможни игри - 9! или 362 880 (от които уникални 26 830)
- Единият играч е максимизиращ, а другия минимизиращ

Морски шах – примерна игра



Морски шах – примерна игра



Морски шах – примерна игра

		O
	X	

Морски шах – примерна игра

		O
	X	
X		

Морски шах – примерна игра

		O
	X	
X		O

Морски шах – примерна игра

		O
	X	X
X		O

Морски шах – примерна игра

		O
O	X	X
X		O

Морски шах – примерна игра

	X	O
O	X	X
X		O

Морски шах – примерна игра

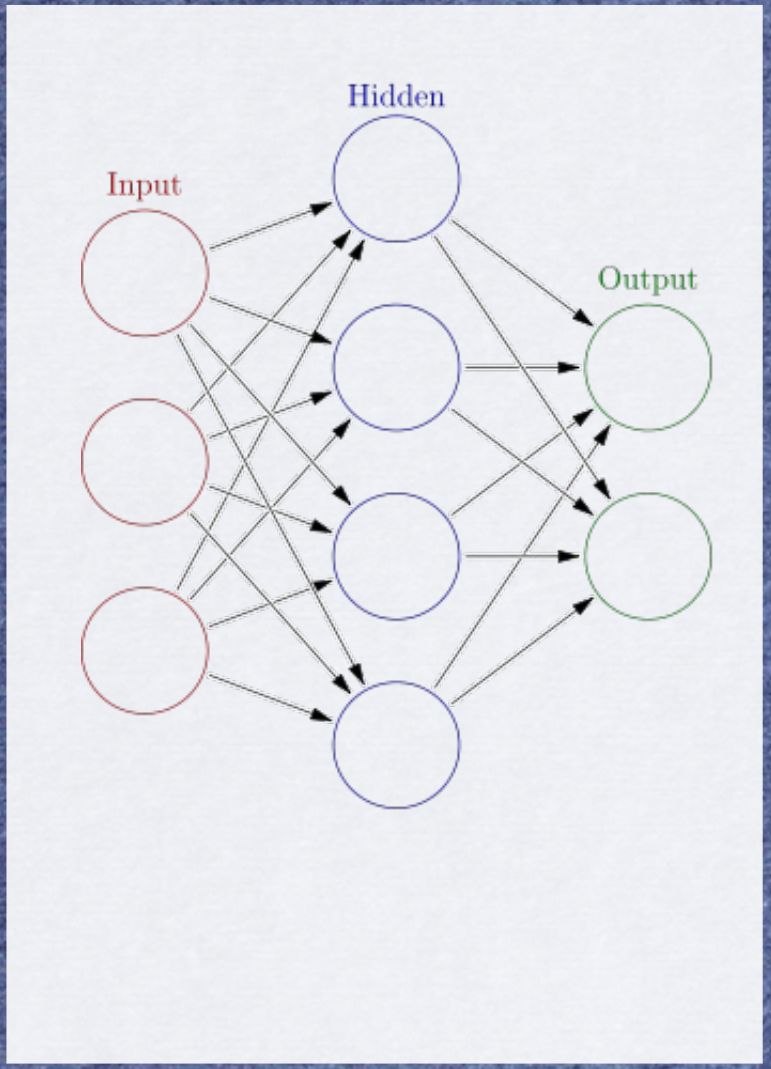
	X	O
O	X	X
X	O	O

Морски шах – примерна игра

X	X	O
O	X	X
X	O	O

Изкуствени невронни мрежи

- Насочен тегловен граф
  - Възли
  - Връзки
- Научаване на шаблони
  - Оценка на игралното табло



Генетични алгоритми

- Евристични популационни алгоритми за оптимизация
- Добра ефективност при оптимизационни проблеми в многомерни пространства
- Взаимстване на идеи от естествената еволюция
  - Кръстосване
  - Мутация
  - Естествен подбор

Обучение на ИНМ с ГА

- Тегловните коефициенти на ИНМ се явяват хромозоми в ГА
- На входа на ИНМ се подава игралната дъска
  - Празните позиции се заместват с числена 0
  - Позициите на максимизиращия играч с +1
  - Позициите на минимизиращия играч с -1
- В изхода на ИНМ
  - Оценка на всяка свободна позиция в интервала от -1 до +1

Цикъл на обучение

- Популацията се състои от група ИНМ индивиди
- Оценката на всеки индивид (жизнеспособността му в ГА) се определя в турнир
  - Всеки играе срещу всеки друг
    - 3 точки за победа, 1 точка за равенство, 0 за загуба
- Обучението протича без учител (тоест без обучаващи образци)
- След оценката
  - Селекция, кръстосване и мутация

Модел на данните

```
struct ANN {
        static const int INPUT_SIZE = 9+1;
        static const int HIDDEN_SIZE = 8+1;
        static const int OUTPUT_SIZE = 9+1;
        static const int INPUT_BIAS_INDEX = INPUT_SIZE - 1;
        static const int HIDDEN_BIAS_INDEX = HIDDEN_SIZE - 1;
        static const int OUTPUT_BIAS_INDEX = OUTPUT_SIZE - 1;
        double input[ INPUT_SIZE ];
        double wih[ INPUT_SIZE*HIDDEN_SIZE ];
        double hidden[ HIDDEN_SIZE ];
        double who[ HIDDEN_SIZE*OUTPUT_SIZE ];
        double output[ OUTPUT_SIZE ];
        long score;
};
```

Представяне в пеметта

```
int board[3][3] = {
        {0, 0, 0},
        {0, 0, 0},
        {0, 0, 0}
};

double strength[3][3] = {
        {0.0, 0.0, 0.0},
        {0.0, 0.0, 0.0},
        {0.0, 0.0, 0.0}
};

ANN population[ POPULATION_SIZE ];

ANN *white = NULL;
ANN *black = NULL;

ANN *first = NULL;
ANN *second = NULL;
ANN *child = NULL;
```

Инициализация на популацията

```
for(int p=0; p<POPULATION_SIZE; p++) {
        for(int i=0; i<(ANN::INPUT_SIZE*ANN::HIDDEN_SIZE); i++) {
                population[p].wih[i] = (double)2.0*EPSILON*(double)rand()/
(double)RAND_MAX - EPSILON;
        }


        for(int i=0; i<(ANN::HIDDEN_SIZE*ANN::OUTPUT_SIZE); i++) {
                population[p].who[i] = (double)2.0*EPSILON*(double)rand()/
(double)RAND_MAX - EPSILON;
        }
}
```

Кръстосване

```
for (int i = 0; i < (ANN::INPUT_SIZE*ANN::HIDDEN_SIZE); i++) {
        child->wih[i] = first->wih[i];
}


for (int i = 0; i < (ANN::HIDDEN_SIZE*ANN::OUTPUT_SIZE); i++) {
        child->who[i] = second->who[i];
}
```

Мутация

```
const double WEIGHT = 0.0000001 * (double)(rand()/10000);

ANN *a = &population[ rand()%POPULATION_SIZE ];
ANN *b = &population[ rand()%POPULATION_SIZE ];

for (int i = 0; i < (ANN::INPUT_SIZE*ANN::HIDDEN_SIZE); i++) {
      child->wih[i] -= WEIGHT*(a->wih[i]-b->wih[i]);
}

for (int i = 0; i < (ANN::HIDDEN_SIZE*ANN::OUTPUT_SIZE); i++) {
      child->who[i] -= WEIGHT*(a->who[i]-b->who[i]);
}
```

Цели при обучението

- ИНМ да постига винаги победа или поне равенство
  - Първоначално, срещу играч играещ безразборно
  - На второ ниво, играч играещ по предварително избрана стратегия
  - На трето ниво, играч човек

Параметри за изследване

- Размер на скрития слой в ИНМ
- Различни операции при кръстосване
- Различни операции при мутация
- Различни начини за селекция
- Размер на популацията
- Възможности за подобряване на сходимостта при изчисления в разпределена среда
  - Алгоритми за разпределяне на популацията

Благодаря за вниманието!