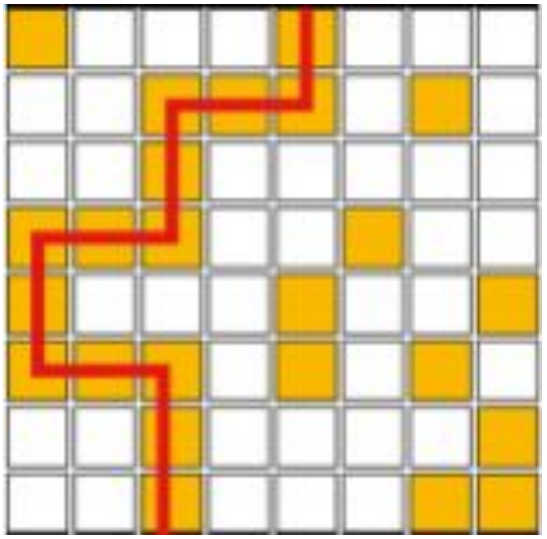
Применение методов глубокого обучения к задаче конкурирующей перколяции

Перколяция - процесс распространения текучего вещества в пористой среде



$$\sum_s n_s(p) \sim |p - p_c|^{2-\alpha}$$

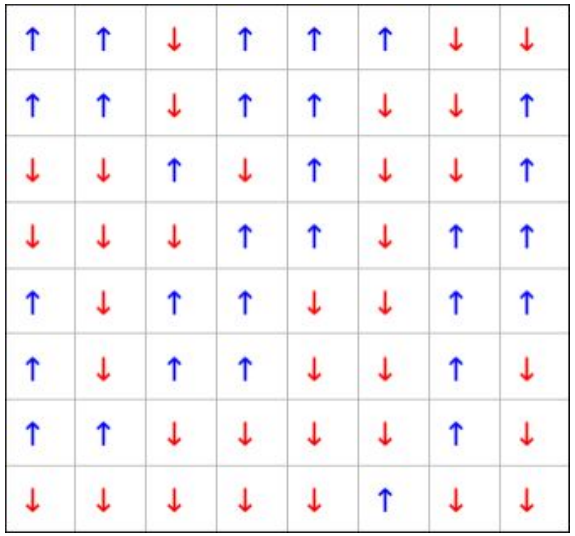
$$P_\infty(p) = \sum_s s * n_s(p) \sim |p - p_c|^{\beta}$$

$$S(p) = \sum_s s^2 * n_s(p) \sim |p - p_c|^{-\gamma}$$

$$C_v \sim |\tau|^{-\alpha}$$

$$\langle \varphi \rangle \sim |\tau|^{\beta}$$

$$\chi \sim |\tau|^{-\gamma}$$

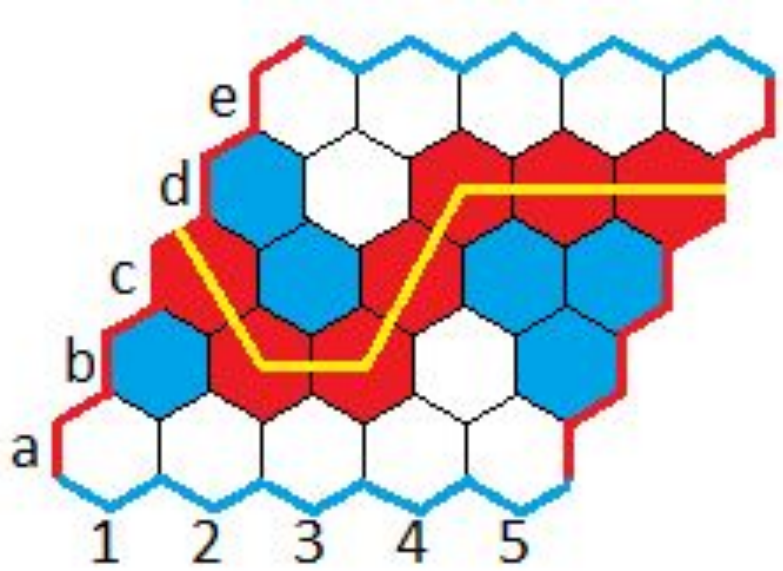


$$\alpha + 2 * \beta + \gamma = 2$$

Фазовый переход - наличие/отсутствие соединяющего кластера

Формулы, описывающие перколяционную систему вблизи фазового перехода, можно сопоставить формулам, описывающим магнитную систему так же вблизи фазового перехода

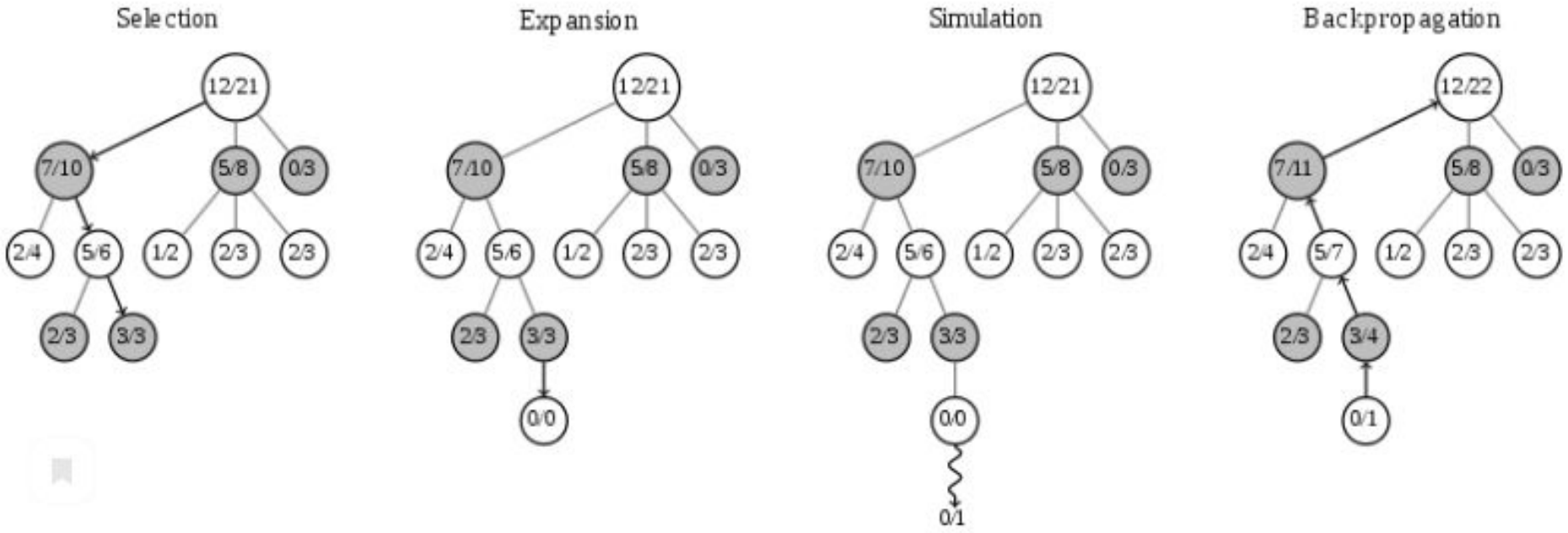
Понятие о конкурирующей перколяции вводится на основе пошаговой настольной игры Hex:



1. Игроки ходят по очереди и занимают одно из свободных полей на доске
2. Побеждает игрок, первый построивший соединяющий кластер между сторонами своего цвета

В Hex невозможна ничья. Методом от противного легко доказать, что стартующий первым игрок всегда имеет выигрышную стратегию.

Monte Carlo Tree Search - алгоритм принятия решений, часто используемый в играх в качестве основы искусственного интеллекта



Принятие решения осуществляется на основе нескольких сотен итераций поиска. Практически выгодно в качестве итоговой выбирать ветку (ноду) с максимальным количеством посещений.

$$v_i + C \times \sqrt{\frac{\ln N}{n_i}}$$ - формула, определяющая движение по нодам на этапе Selection

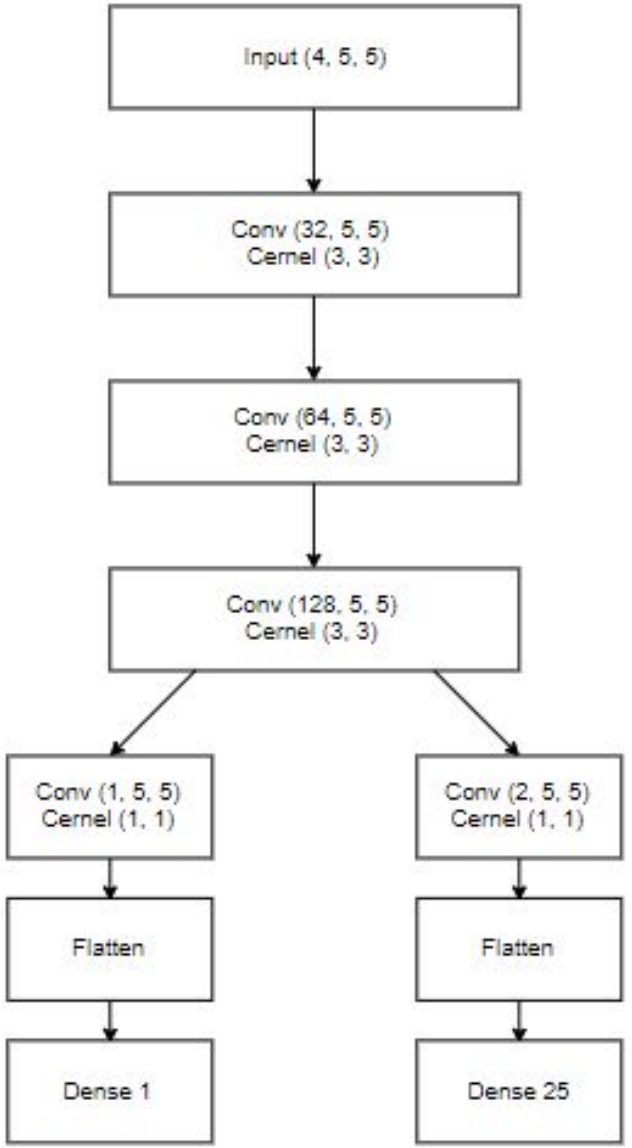
Обучение нейронной сети происходит согласно следующему алгоритму:

1. Накопление обучающей выборки, с импользованием алгоритма MCTS во время игры
2. Тренировка сети на небольшом наборе из обучающей выборки
3. По прошествию определенного числа итераций прогресс нейросети отслеживается путем проведения нескольких партий с классическим MCTS

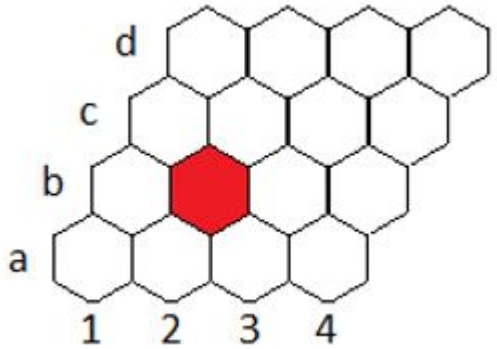
Шаги 1-3 выполняются вплоть до насыщения функции ошибки:

$$l = (z - v)^2 - \boldsymbol{\pi}^{\mathrm{T}} \log \boldsymbol{p} + c\|\theta\|^2$$

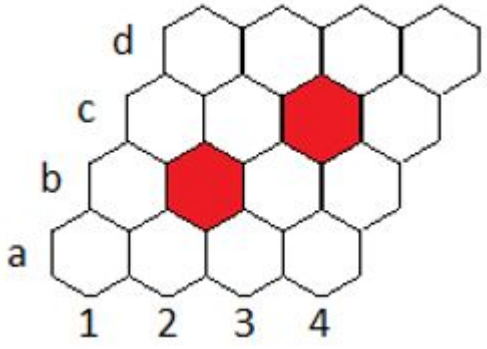
Где z - результат, с которым закончилась партия, π - распределение дочерних к корневой нод по ходам, v и p – value и policy, предсказанные нейронной сетью. C - некоторая постоянная.



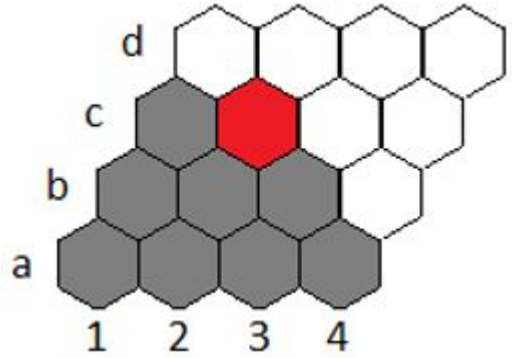
Основные позиции в Hex - список некоторых теоретических позиций, на которые мы будем ссылаться при анализе стратегий, генерируемых сетью



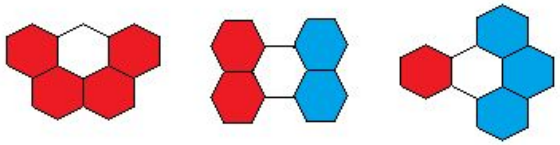
Вилка



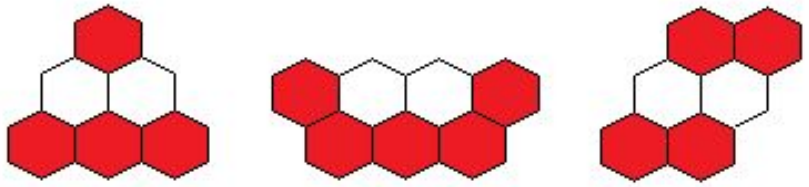
Ромб



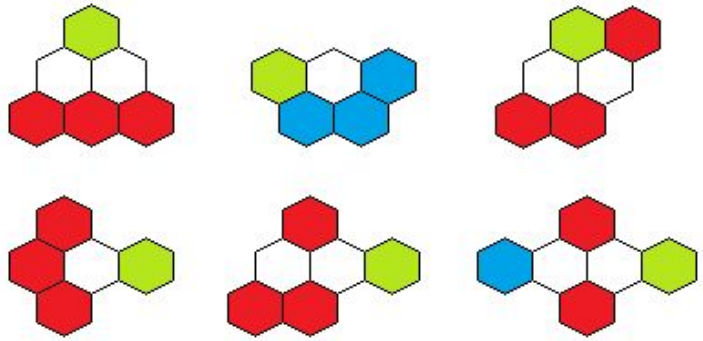
Трапеция Типцова



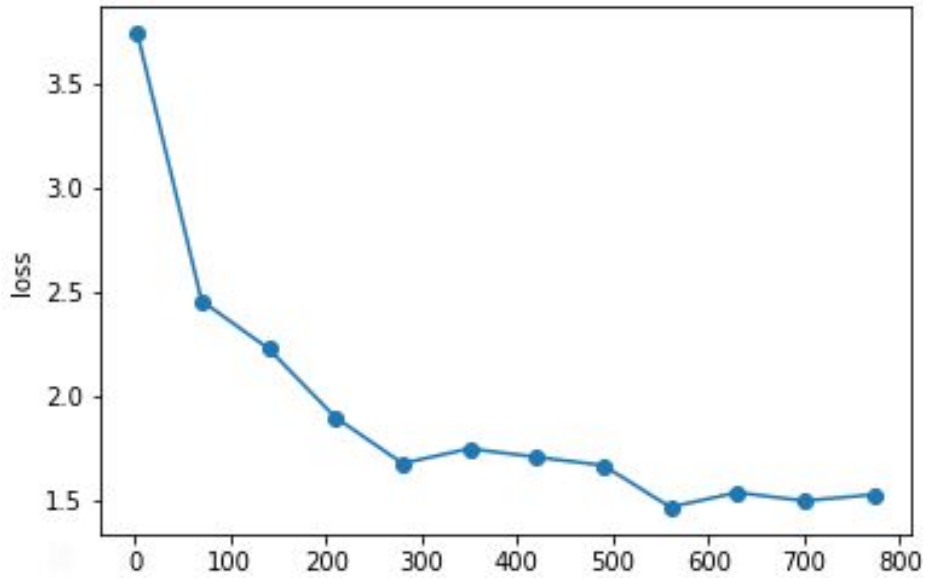
Мертвые ячейки



Оккупированные регионы



Шаблонные ячейки

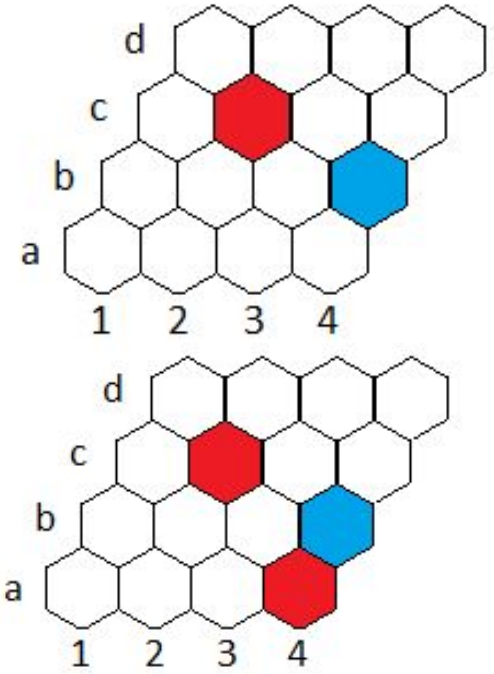


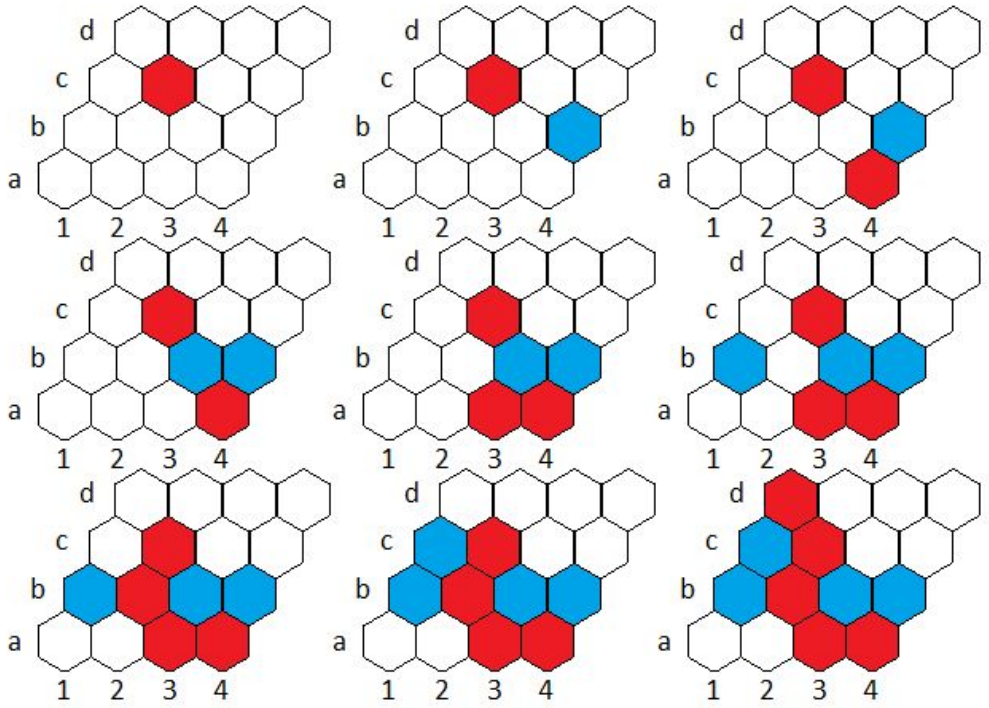
Результат обучения нейронной сети на примере размерности игрового поля Hex 4x4:

На графике слева изображена зависимость функции ошибки от числа итераций обучения

```
[ 0.1275,  0.015 ,  0.03  ,  0.0025]
[ 0.045 ,  0.    ,  0.15  ,  0.045 ]
[ 0.0275,  0.095 ,  0.0375,  0.32  ]
[ 0.005 ,  0.025 ,  0.0275,  0.0475]
```

```
[ 0.    ,  0.    ,  0.0075,  0.    ]
[ 0.    ,  0.    ,  0.    ,  0.    ]
[ 0.    ,  0.    ,  0.    ,  0.    ]
[ 0.    ,  0.    ,  0.    ,  0.9925]
```



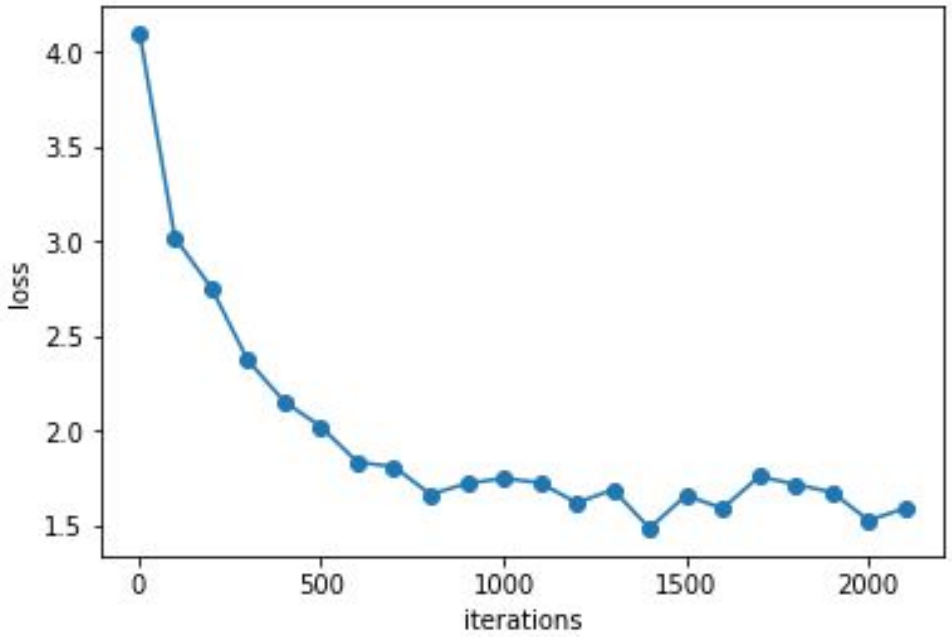


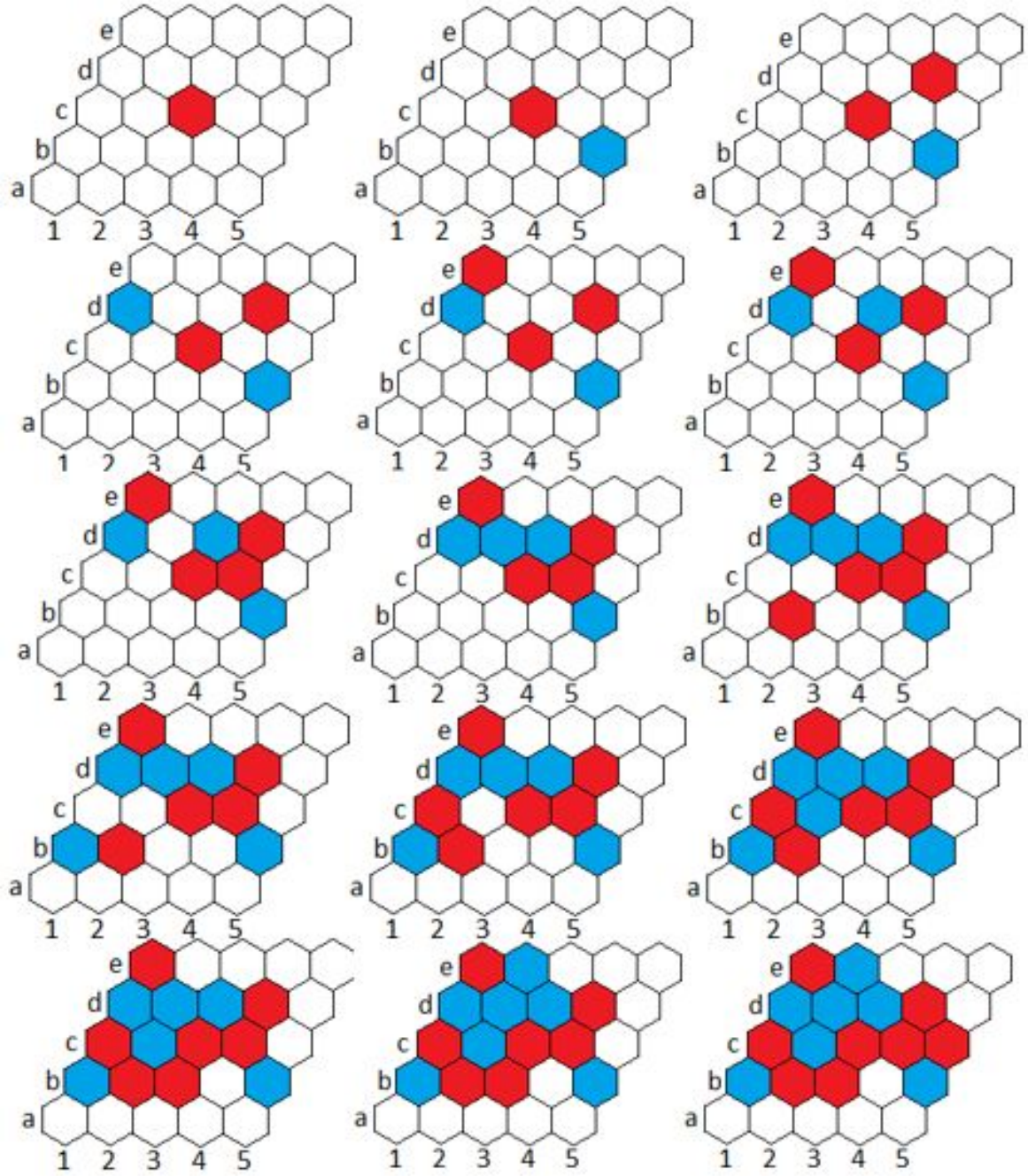
Результат обучения нейронной сети на примере размерности игрового поля Hex 5x5:

```
[ 0.,  0.,  0.,  0.,  0.]
[ 0.,  0.,  0.,  0.,  0.]
[ 0.,  0.,  1.,  0.,  0.]
[ 0.,  0.,  0.,  0.,  0.]
[ 0.,  0.,  0.,  0.,  0.]
```

- распределение нод по количеству посещений, нормированное на 1 перед началом парии

Зависимость значения функции ошибки от числа итераций обучения нейронной сети:





Материал про решетку Hex 6x6

Спасибо за внимание!