

# Обратное распространение ошибки

Практика  
Корлякова М.О.  
2019

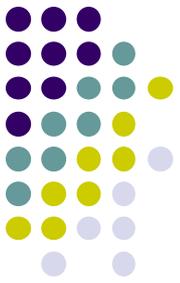




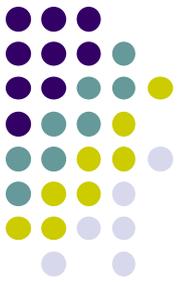
# Нейронная сеть

- Число слоев/число нейронов по слоям
- Обратные связи
- Активационные функции
- Число входов
- Число выходов
- Алгоритм настройки весов(обучения)
- Тип задачи, решаемой сетью

# Режимы функционирования нейросетей



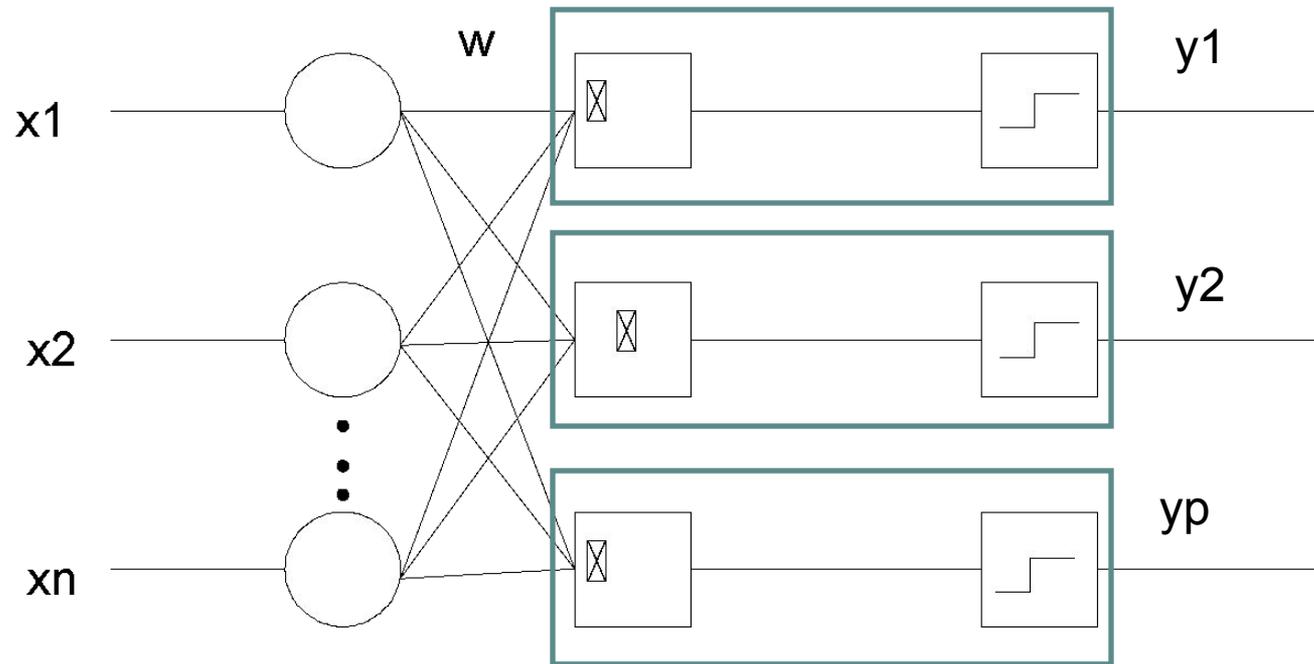
- Обучение
- Работа



# Виды обучения

- С учителем
  - $T = \{(X_i, d_i)\}, i = 1, N$
  - $X_i$  - входной вектор
  - $d_i$  - выходной вектор
- Без учителя
  - $T = \{X_i\}, i = 1, N$
  - $X_i$  - входной вектор

# Однослойная сеть



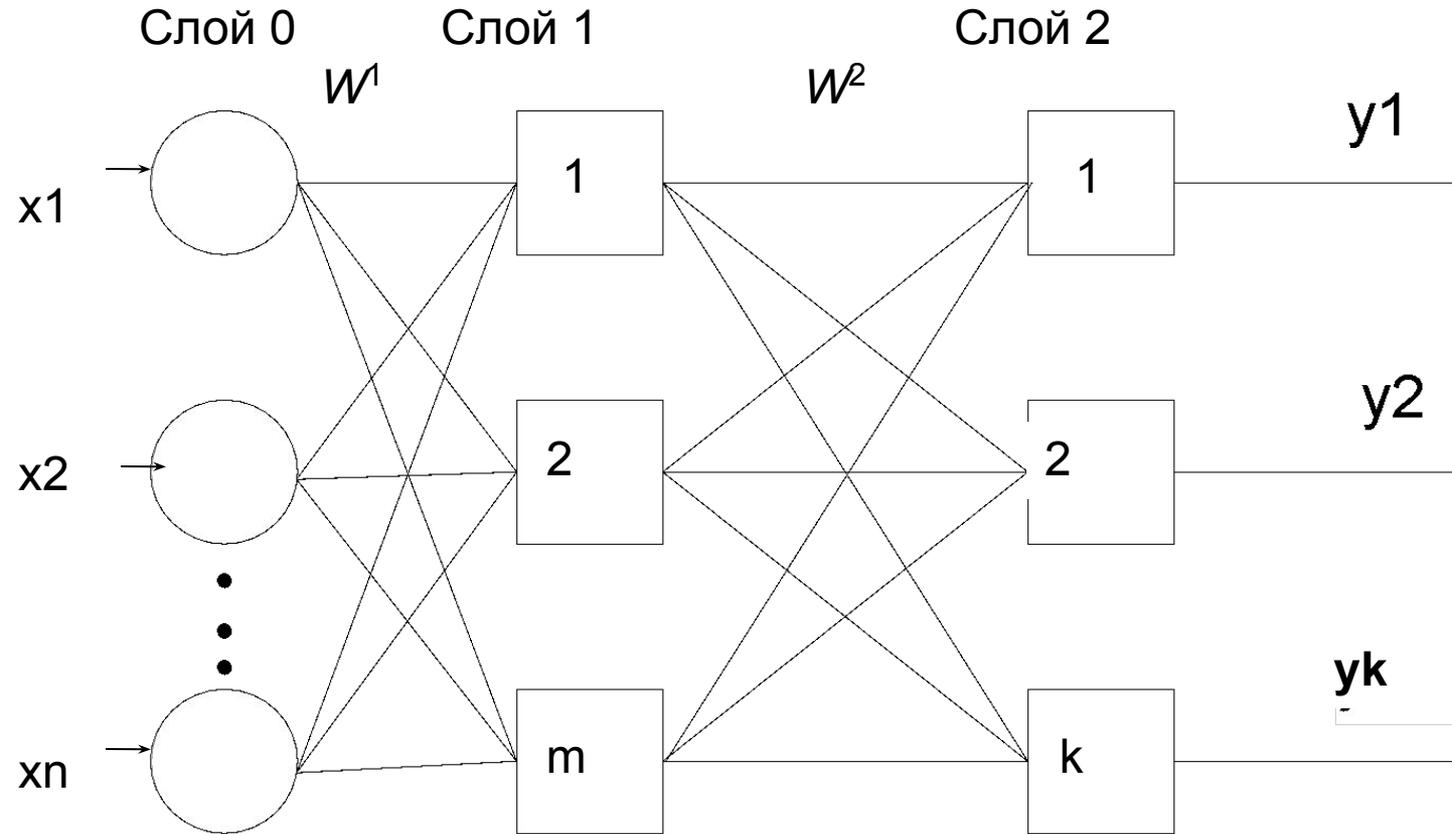
персептрон

Слой 1

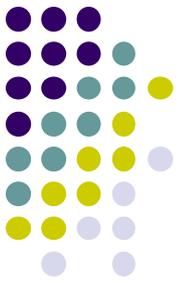
$$y_i = f_i \left( \sum_{l=0, n} x_l w_l \right)$$



# Многослойная сеть (2 слоя)



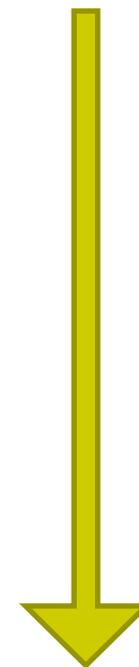
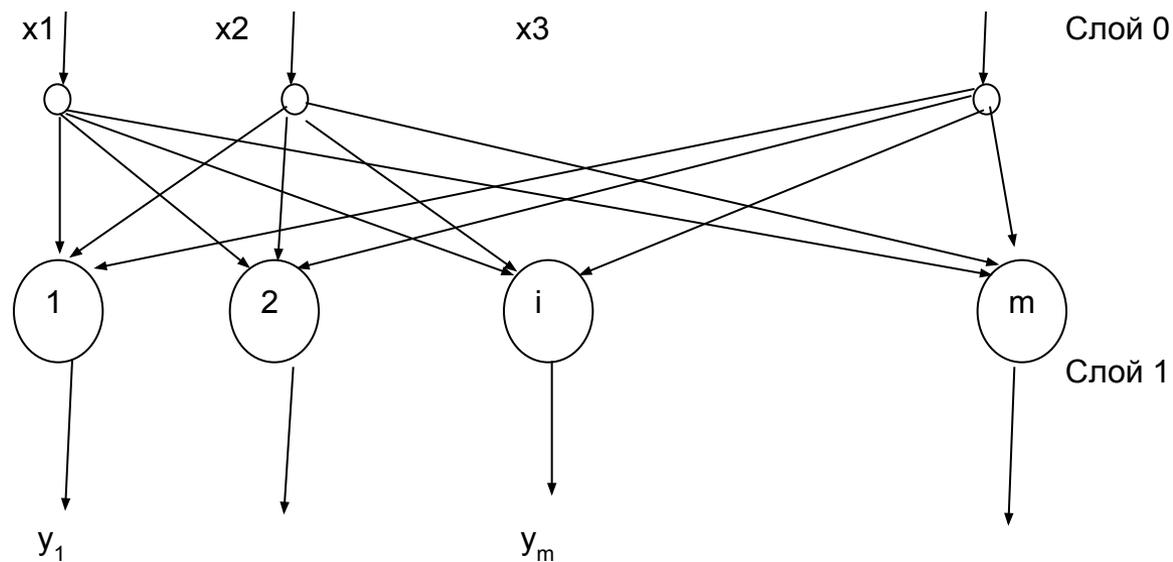
$$y_i = f_i^2 \left( \sum_{j=0..m} f_j^1 \left( \sum_{l=0..n} x_l w_l^1 \right) w_j^2 \right)$$



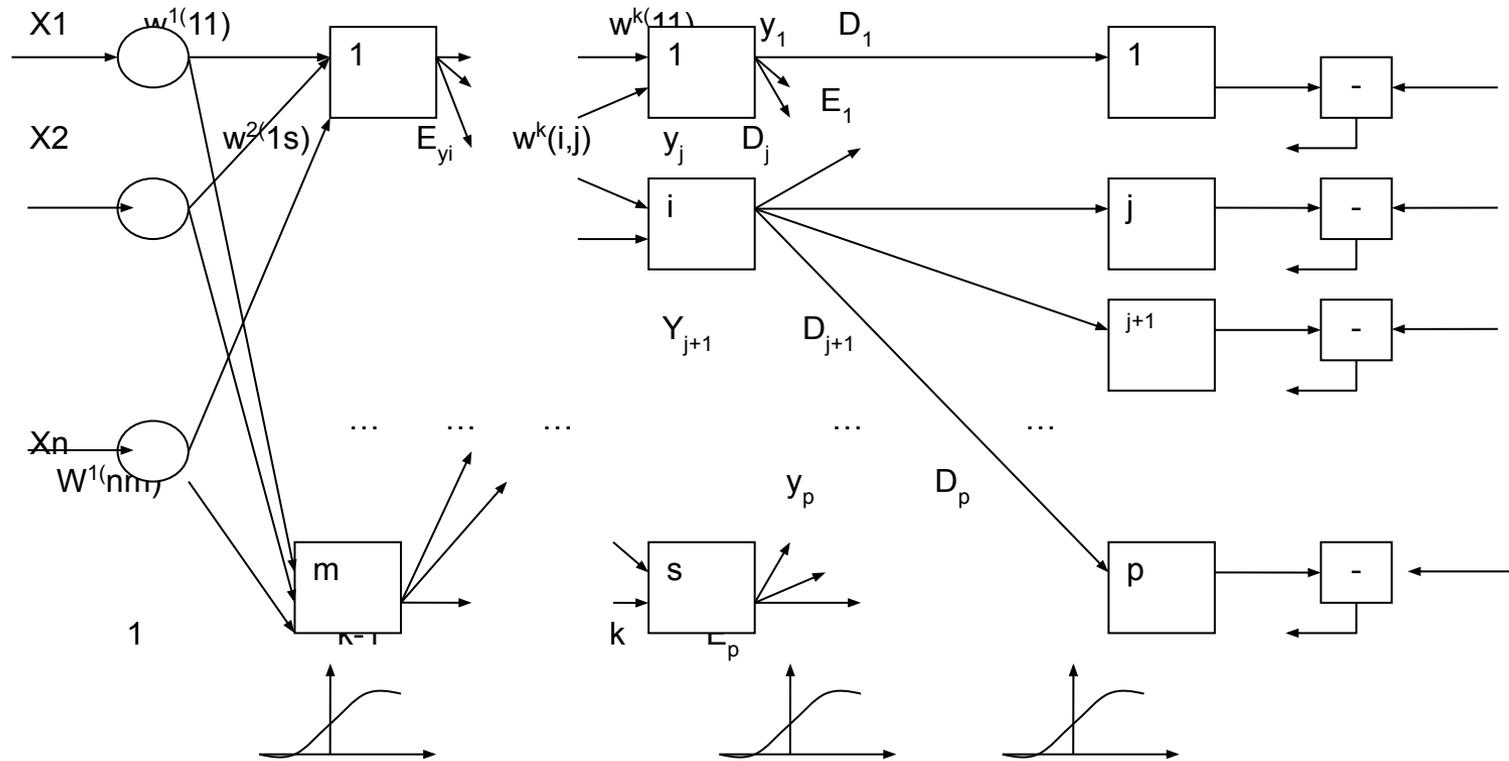
# Слой нейросети

- Группа нейронов сети:
  - Общие функции (задачи)
  - Общие внешние связи
  - Общие входы
  
- Зачем выделять слой?

# Сеть прямого распространения

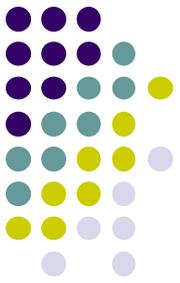


# Многослойная сеть прямого распространения (к слоев)

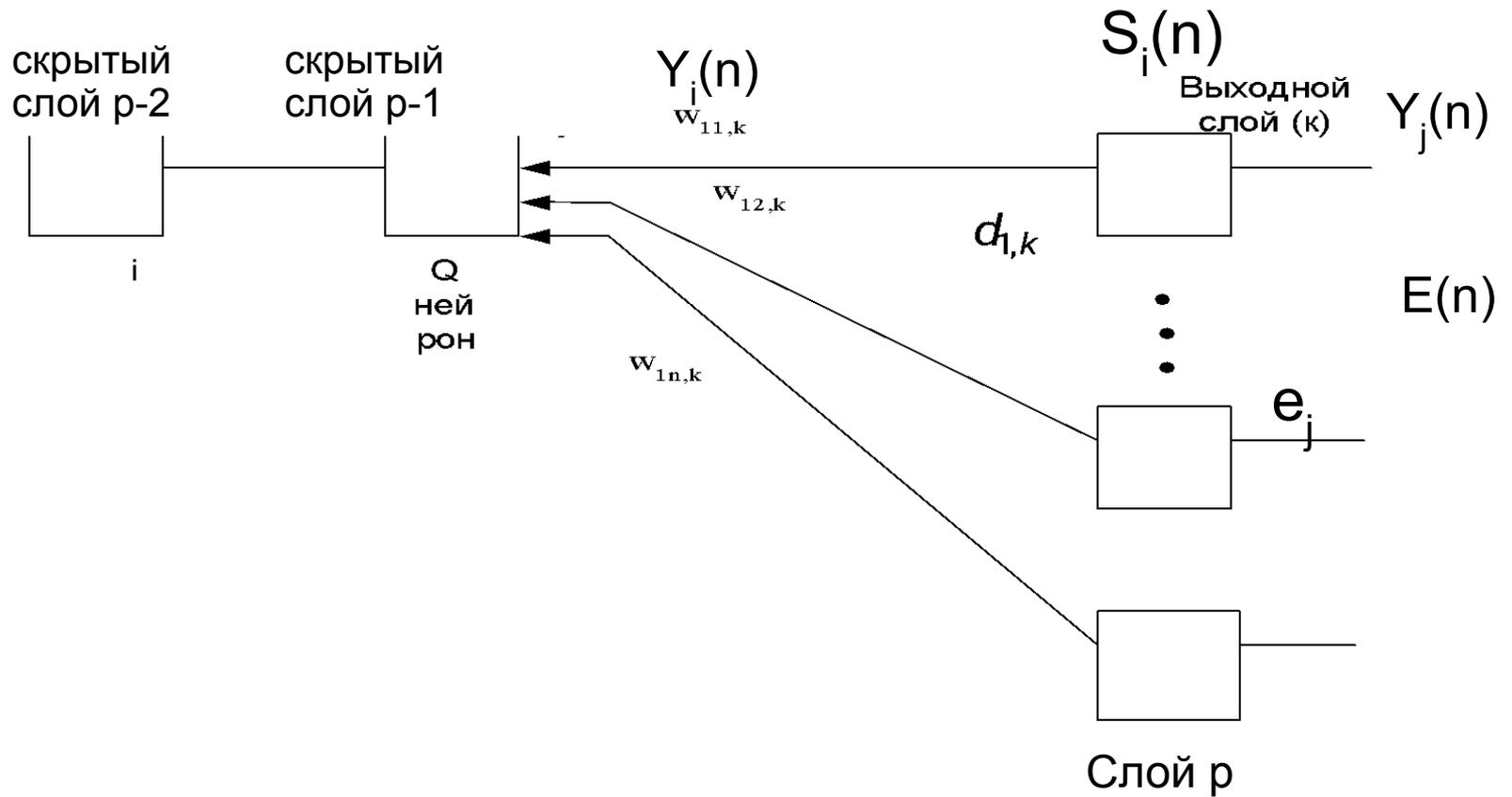


$$y_i = f_i \left( \sum_{j=0}^m f_j \left( \sum_{l=0}^n x_l w_l \right) w_j \right)$$

# Обратное распространение ошибки



# Фрагмент сети

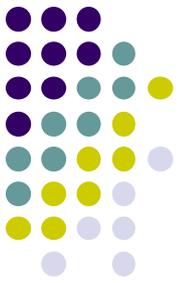




# Обучение по ошибке

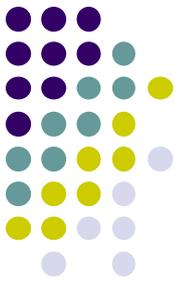
- Глобальная задача – сложная.
- $E(n) = \sum e_j(n)^2$
- $e_j(n) = D_j - Y_j(n)$
- Зависит от всех настраиваемых параметров.

# Функции пакета Neural NetWork ToolBox MatLab



- `net = newff([0 8],[2 1],{'tansig' 'purelin'},'trainlm');`
- `net=train(net,p,t);`
- `y2 = sim(net,p)`
- `lab3_prim.m`

# Алгоритмы



- TRAINCGB
- TRAINBFG
- TRAINOSS
- TRAINGD
- TRAINRP
- TRAINLM

# Neural Network object architecture



numInputs: 1

numLayers: 2

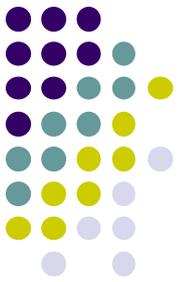
biasConnect: [1; 1]

inputConnect: [1; 0]

layerConnect: [0 0; 1 0]

outputConnect: [0 1]

targetConnect: [0 1]



numOutputs: 1 (read-only)

numTargets: 1 (read-only)

numInputDelays: 0 (read-only)

numLayerDelays: 0 (read-only)



# subobject structures:

inputs: {1x1 cell} of inputs

layers: {2x1 cell} of layers

outputs: {1x2 cell} containing 1 output

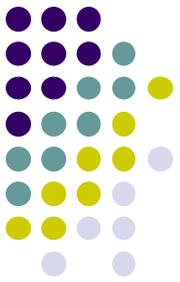
targets: {1x2 cell} containing 1 target

biases: {2x1 cell} containing 2 biases

inputWeights: {2x1 cell} containing 1 input weight

layerWeights: {2x2 cell} containing 1 layer weight

# functions:



- adaptFcn: 'train'
- initFcn: 'initlay'
- performFcn: 'mse'
- trainFcn: 'trainlm'

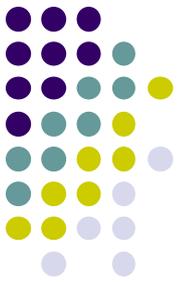
# trainParam:

.epochs,

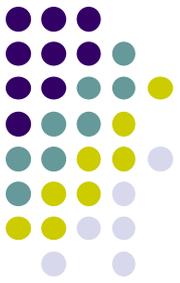
.goal,

.show,

.time



# weight and bias values:



- IW: {2x1 cell} containing 1 input weight matrix
- LW: {2x2 cell} containing 1 layer weight matrix
- b: {2x1 cell} containing 2 bias vectors



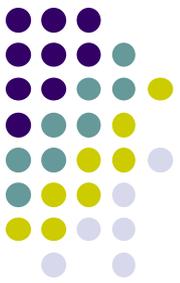
# Ошибка обучения

- $(P, T)$  – обучающая выборка
- $P$  – входные вектора
- $T$  – выходные вектора
- $net$  – нейронная сеть
- $Y = \text{sim}(net, P)$  – выходы сети
- Средняя Абсолютная ошибка обучения
- $E_{\text{обучения}} = \text{mean}(\text{abs}(Y - T))$



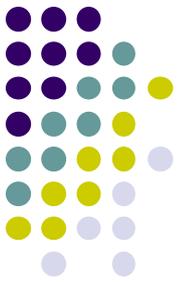
# Ошибка обучения

- $(P, T)$  – обучающая выборка
- $P$  – входные вектора
- $T$  – выходные вектора  $\{-1, 1\}$
- $N$  – число обучающих пар
- $net$  – нейронная сеть
- $Y = \text{sim}(net, P)$  – выходы сети
- Ошибка классификации обучения
- $E_{\text{обучения\_cls}} = \text{sum}((\text{sign}(Y) \neq T)) / N$



# Ошибка обобщения (теста)

- $(P_{\text{тест}}, T_{\text{тест}})$  – тест выборка
- $P_{\text{тест}}$  – входные вектора
- $T_{\text{тест}}$  – выходные вектора
- $net$  – нейронная сеть
- $Y_{\text{тест}} = \text{sim}(net, P_{\text{тест}})$  – выходы сети
- Средняя Абсолютная ошибка обучения
- $E_{\text{тест}} = \text{mean}(\text{abs}(Y_{\text{тест}} - T_{\text{тест}}))$



# Ошибка обобщения

- $(P_{\text{тест}}, T_{\text{тест}})$  – тест выборка
- $P_{\text{тест}}$  – входные вектора
- $T_{\text{тест}}$  – выходные вектора  $\{-1, 1\}$
- $N$  – число тестовых пар
- $net$  – нейронная сеть
- $Y_{\text{тест}} = \text{sim}(net, P_{\text{тест}})$  – выходы сети
- Ошибка классификации обучения
- $E_{\text{тест\_cls}} = \text{sum}(\text{abs}(\text{sign}(Y_{\text{тест}}) - T_{\text{тест}})) / N$



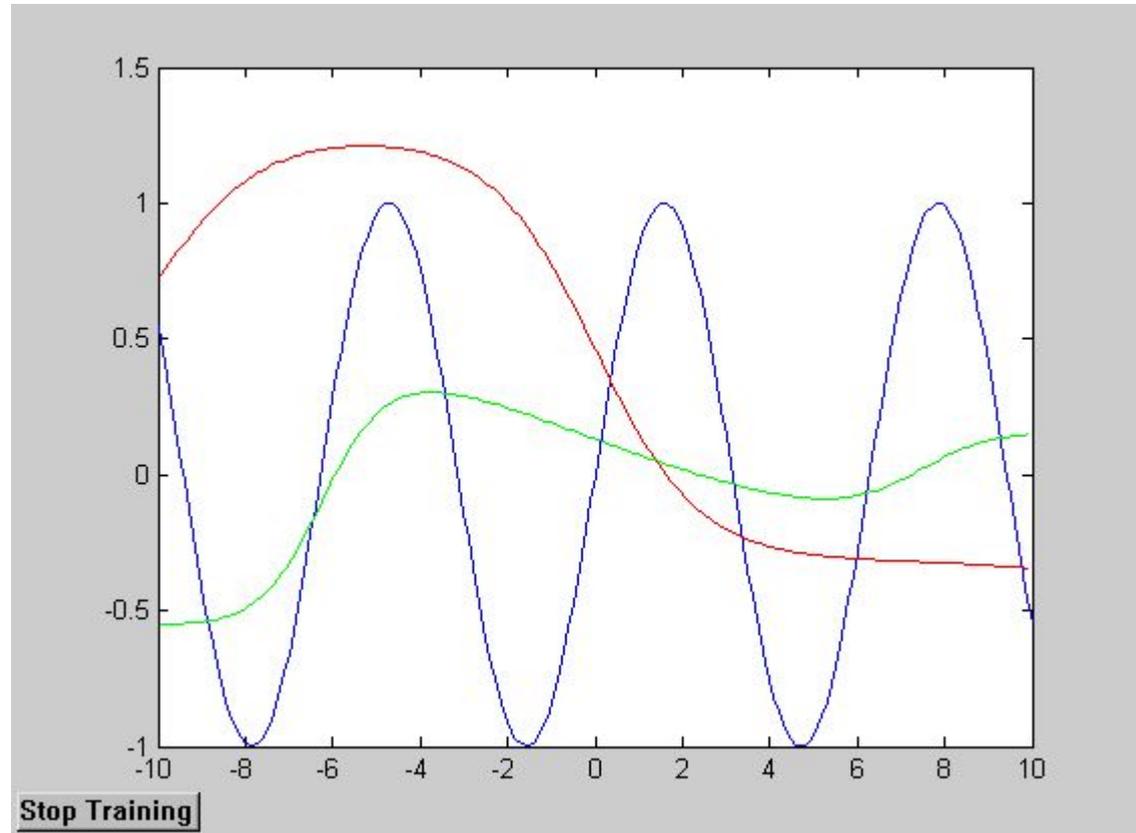
# Обучение сети. lab3\_prim\_2.m

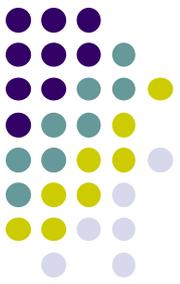
- 3 – нелинейных нейрона, градиентный алгоритм

Без  
Обучения

Целевое  
состояние

Обучение





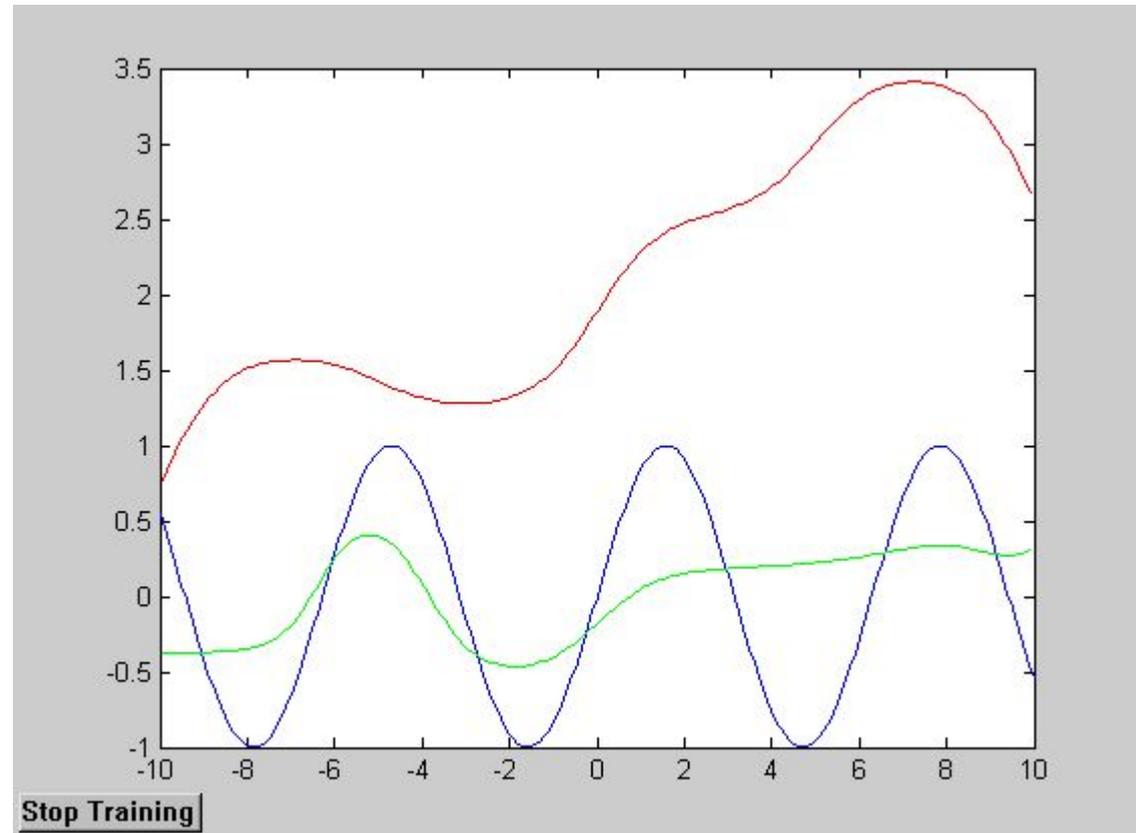
# Обучение сети

- 8 – нелинейных нейронов, градиентный алгоритм

Без  
Обучения

Целевое  
состояние

Обучение





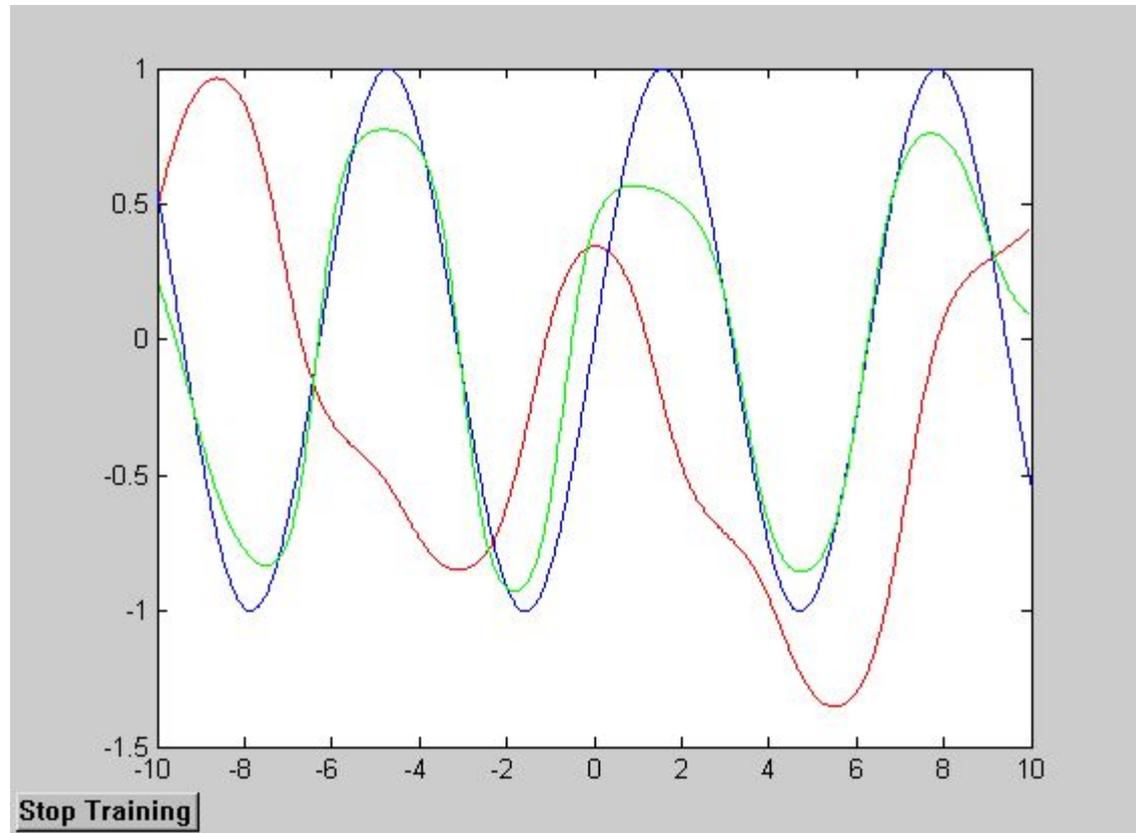
# Обучение сети

- 8 – нелинейных нейронов, градиентный алгоритм

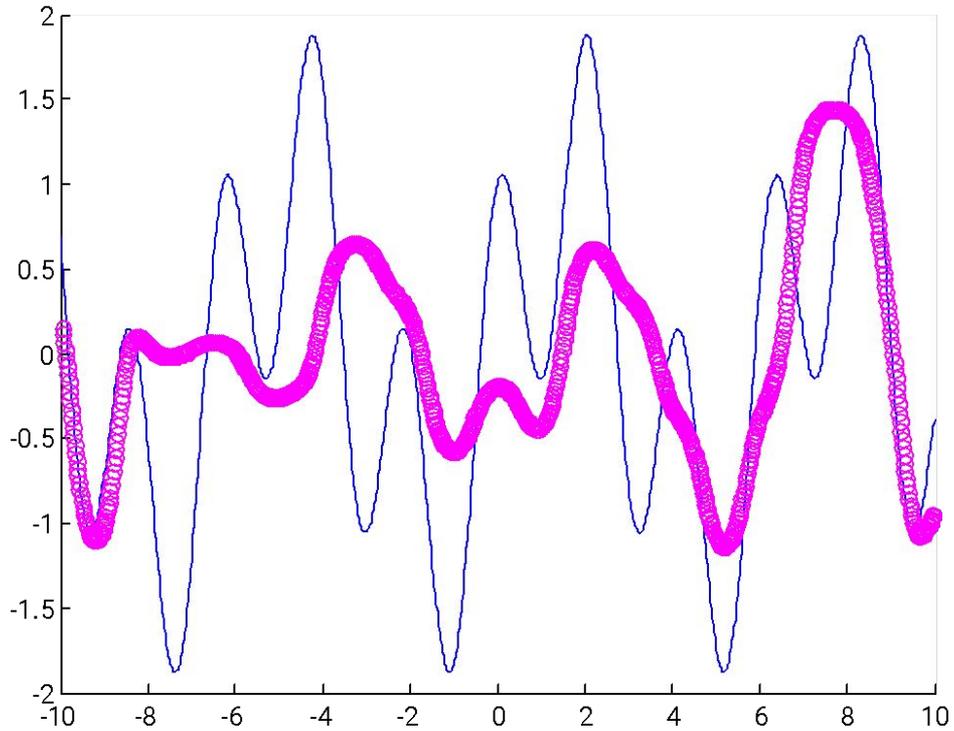
Без  
Обучения

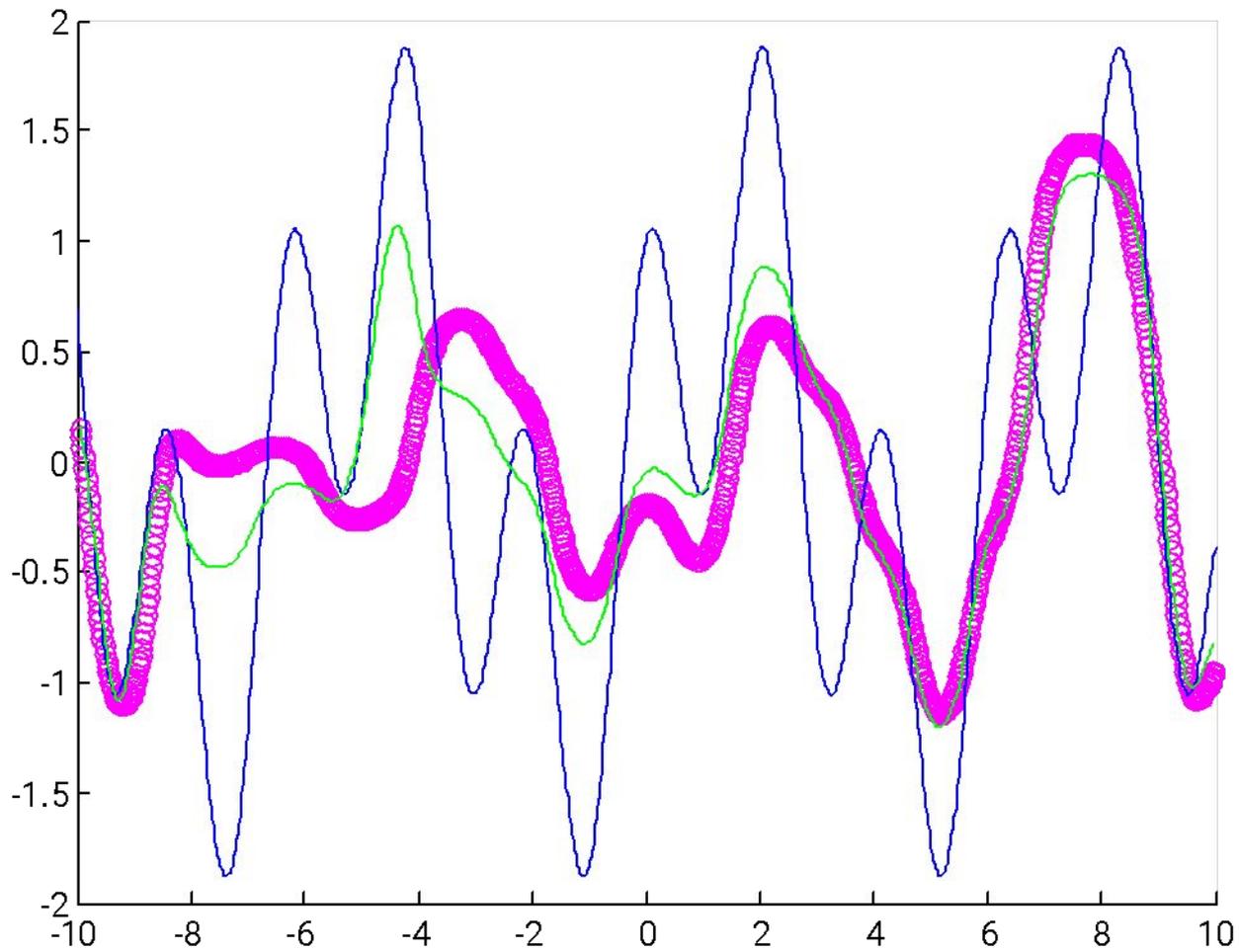
Целевое  
состояние

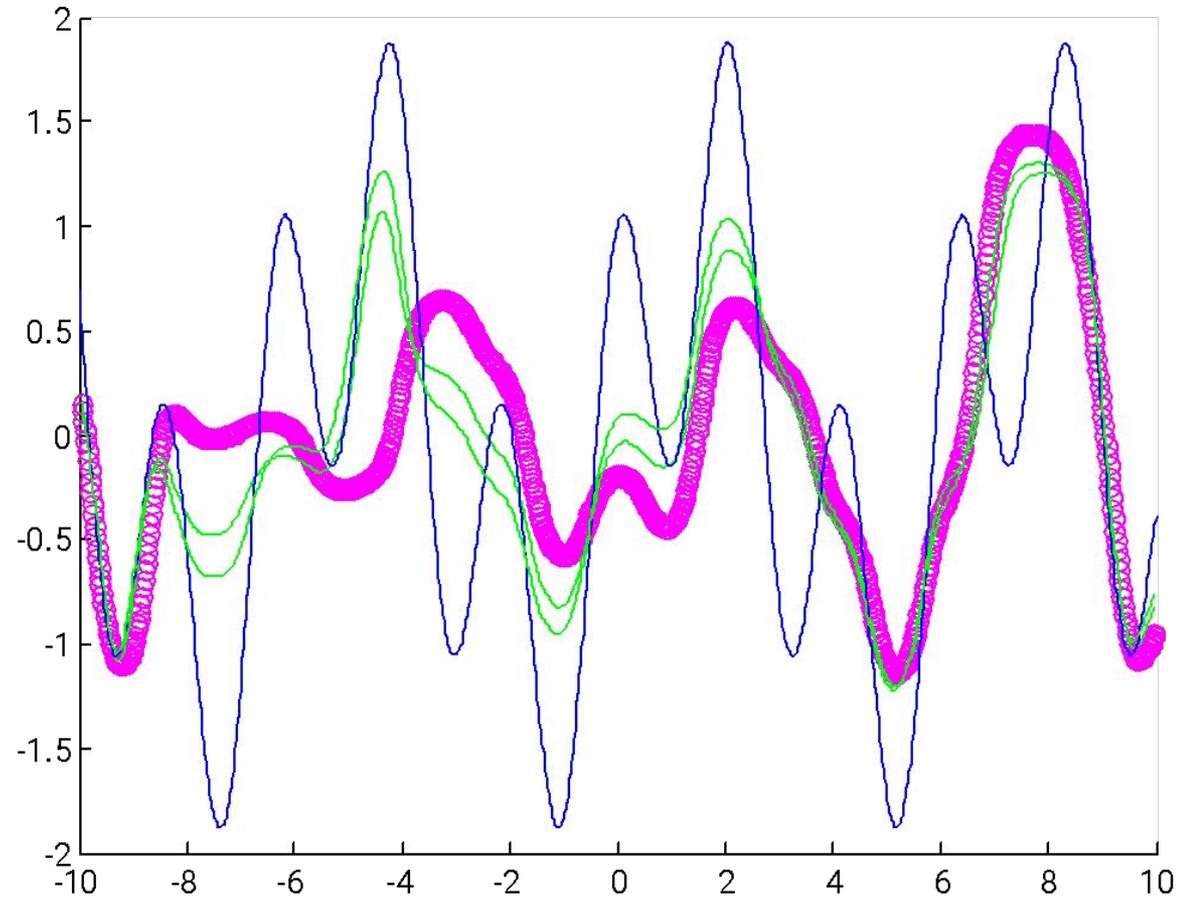
Обучение

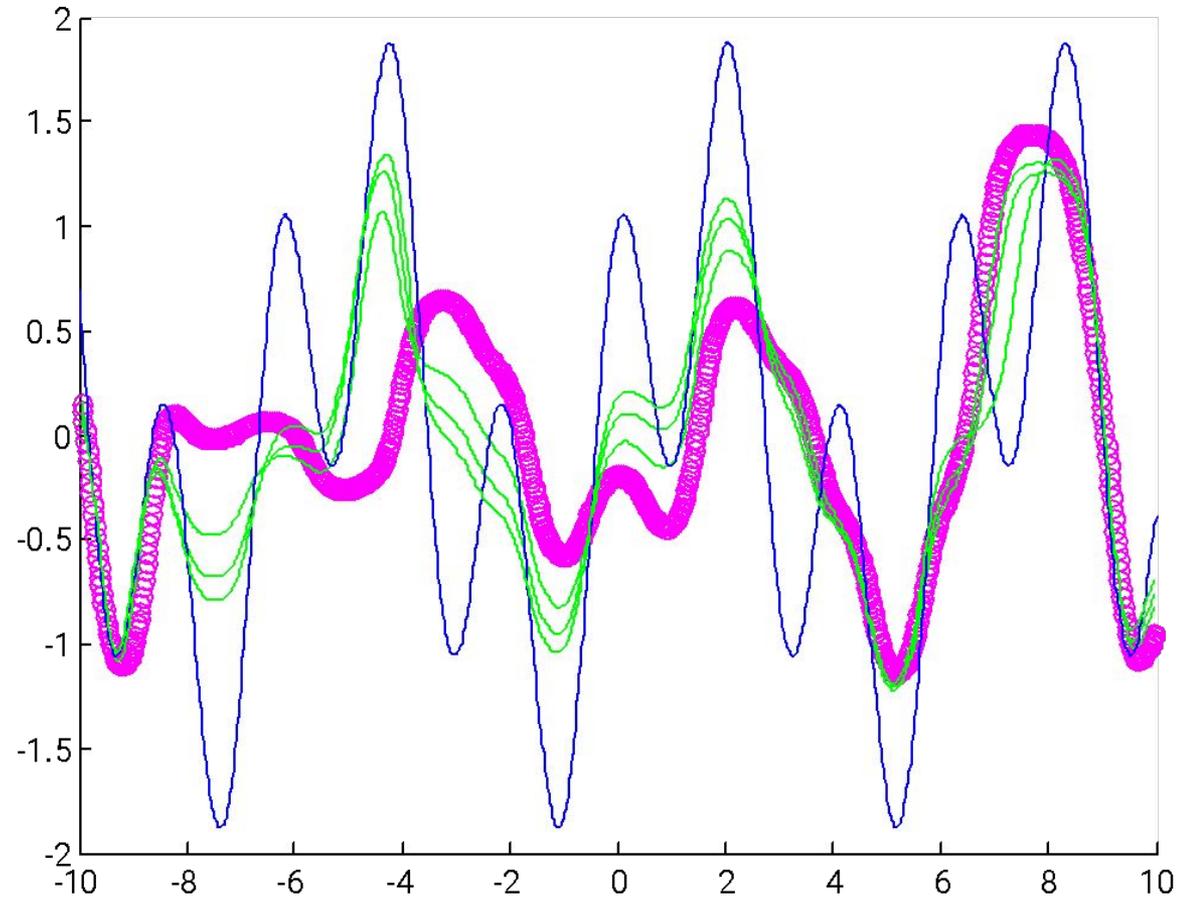


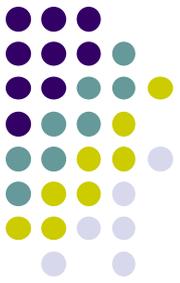
# Добавим обучения net\_train2.m











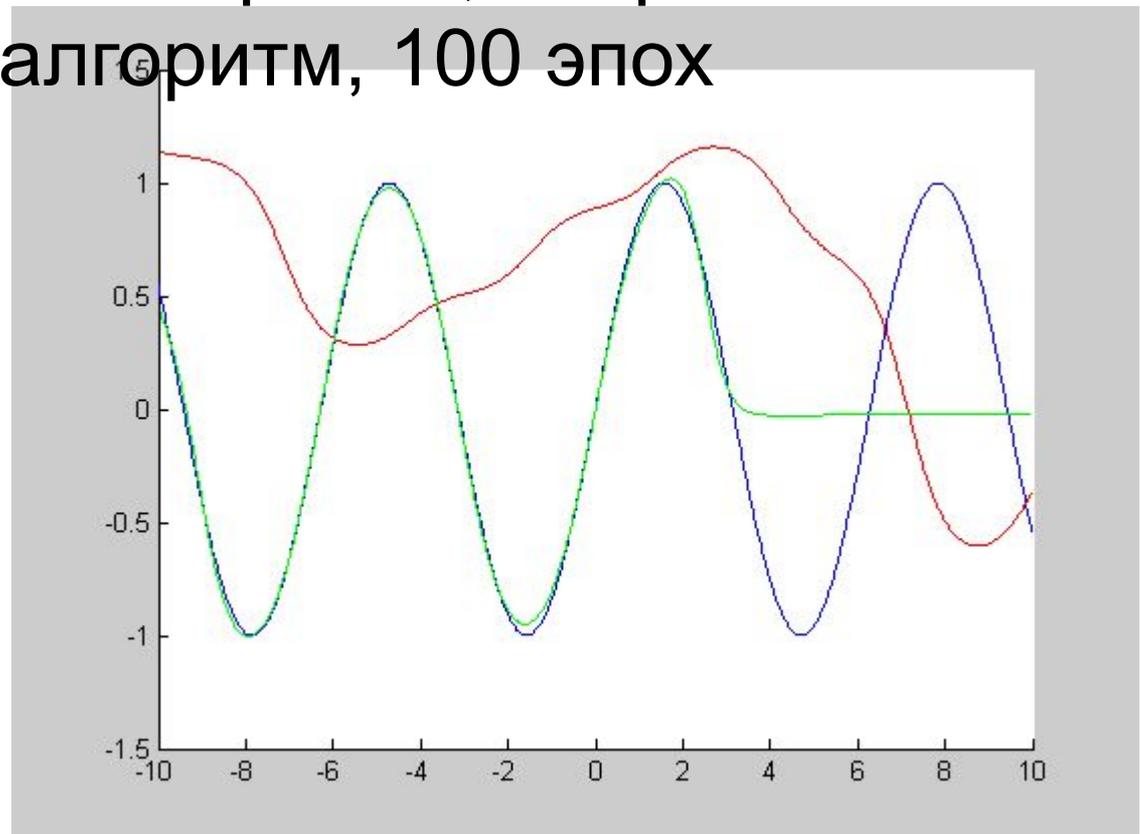
# Обучение сети

- 8 – нелинейных нейронов, сопряженно-градиентный алгоритм, 100 эпох

Без  
Обучения

Обучение

Целевое  
состояние





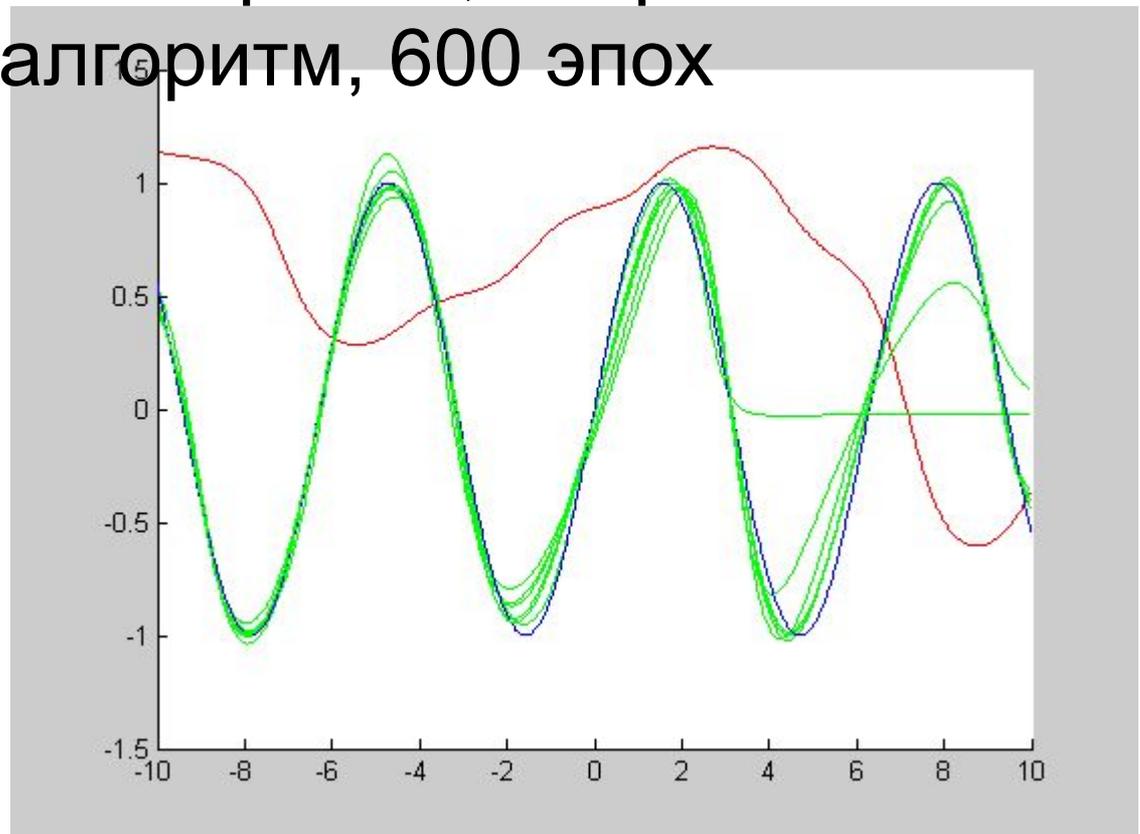
# Обучение сети

- 8 – нелинейных нейронов, сопряженно-градиентный алгоритм, 600 эпох

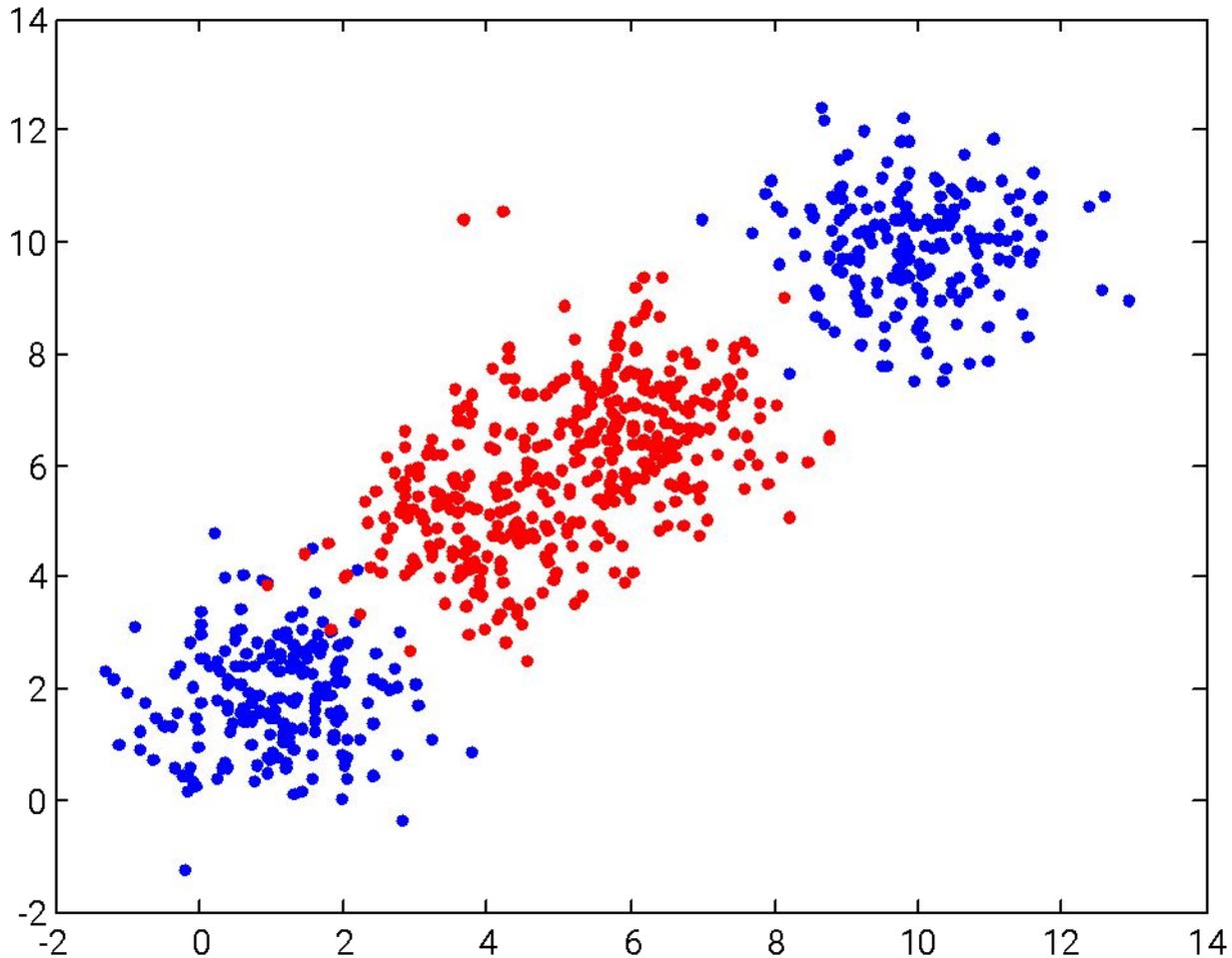
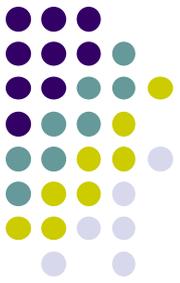
Без  
Обучения

Обучение

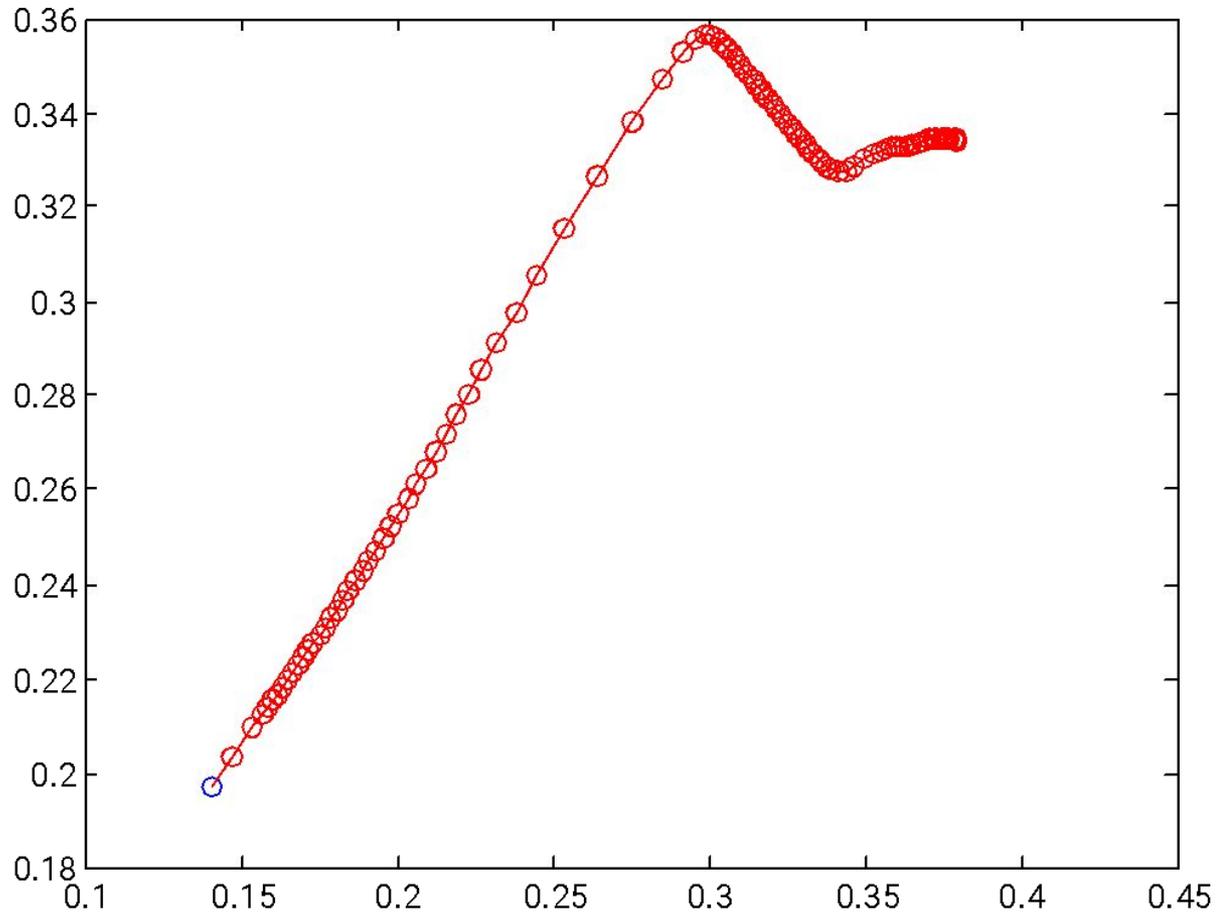
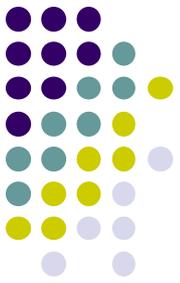
Целевое  
состояние



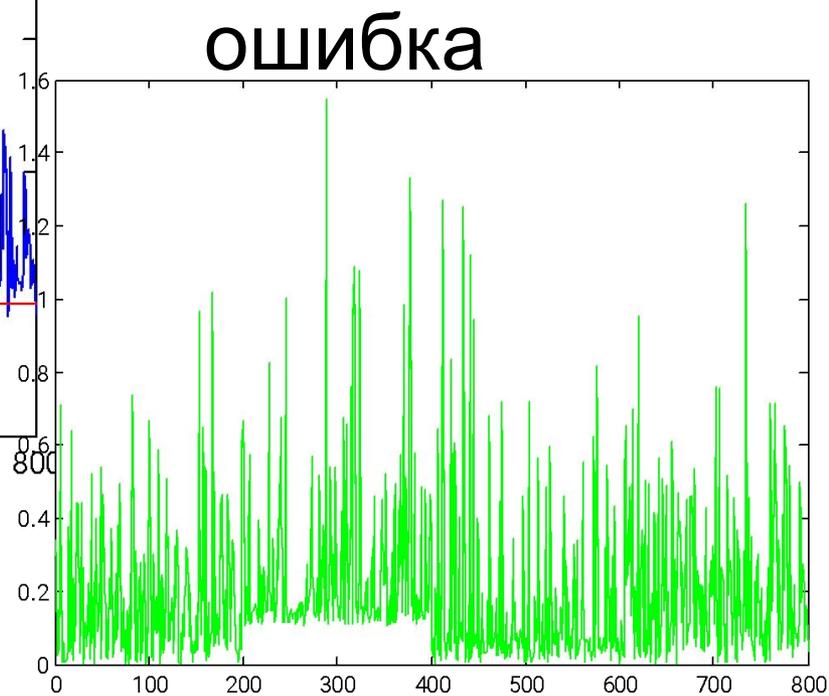
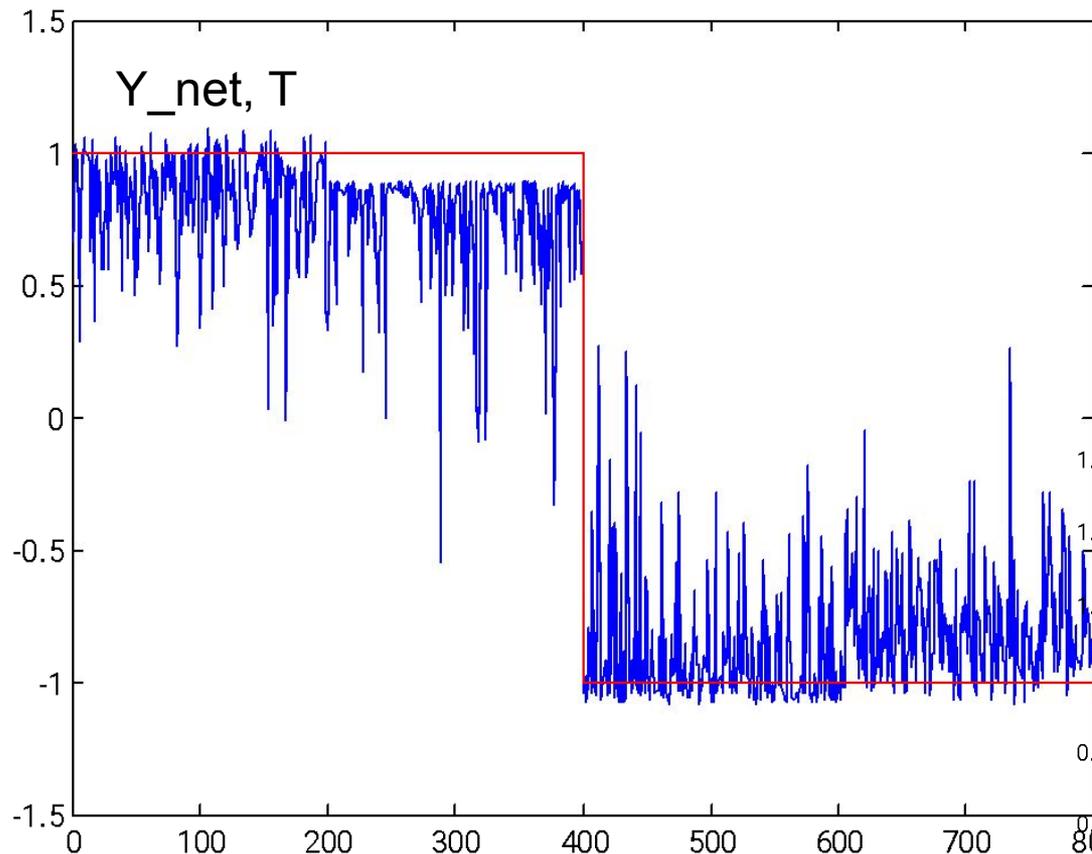
# Классификатор. newffLm.m



# Траектория весов

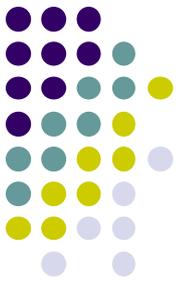


# результаты



# Проблемы

- Локальные минимумы
- Переобучение
- Паралич сети
- Устойчивость/пластичность



# Вопрос

- Приведите общую схему обучения многослойной сети с обратным распространением ошибки.

