## Обучение

Первым делом для более эффективного обучения была выполнена нормализация данных. В качестве метода обучения выбран traingda. Сеть была обучена на тренировочных данных и сохранена.

Исходный код этапа обучения:

load x; load s;

load x1; load s1;

[xn,meanx,stdx,sn,means,stds] = prestd(x,s);

x1n = trastd(x1, meanx, stdx);

mn = min(xn);

mx = max(xn);

net = newff([mn mx],[10 10 1], {'tansig' 'tansig' 'purelin'}, 'traingda');

net.trainParam.epochs = 10;

net = train(net,xn,sn);

Yn = sim(net,x1n);

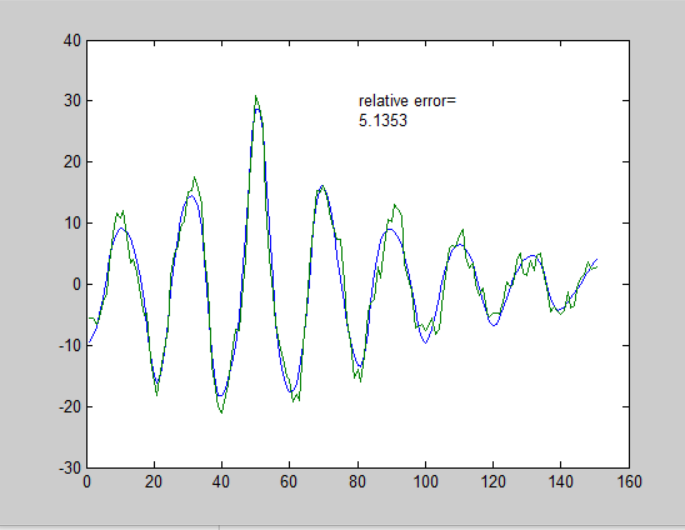
Y = poststd(Yn, means, stds);

labelsx = 1:length(x1);

plot(labelsx, Y, labelsx, s1)

save mynetff mynetff

## Тестирование

На следующем этапе сеть была загружена и протестирована на тестовых данных, результат представлен на рисунке:

Исходный код этапа тестирования:

load x; load s;

load x1; load s1;

[xn,meanx,stdx,sn,means,stds] = prestd(x,s);

x1n = trastd(x1, meanx, stdx);

load mynetff

Yn = sim(net,x1n);

Y = poststd(Yn, means, stds);

labelsx = 1:length(x1);

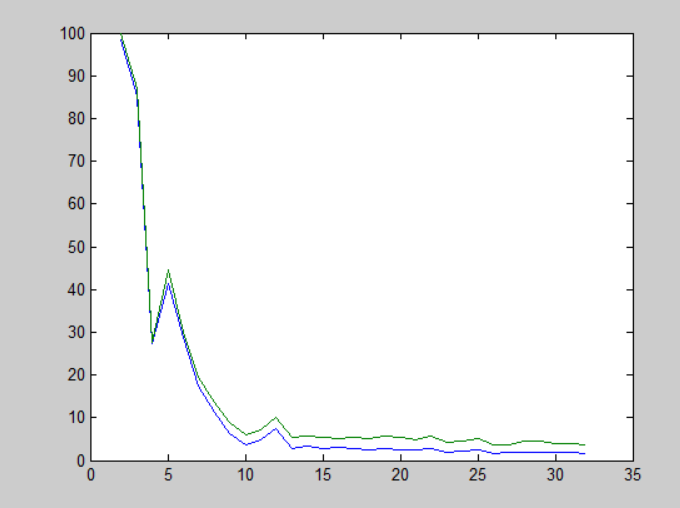
plot(labelsx, Y, labelsx, s1)

delta = mean((Y-s1).^2);

gtext({'relative error=',num2str(delta)});

## Получение зависимости

Завершающим этапом работы стало получение зависимости между погрешностью обучения и тестирования от размера MLP (числа нейронов) для числа MLP от 4 до 64. Была выбрана архитектура с двумя скрытыми слоями размера l, где l от 2 до 32. Количество эпох обучения для каждой сети: 2000. График зависимости ошибок на обучающих и тестовых данных представлен на графике ниже (синий цвет – тренировочные данные, зелёный цвет – тестовые):



По вертикали: размер ошибки,  
По горизонтали: l = размер каждого из 2 скрытых слоёв.

Размер MLP равен l+l.

Исходный код этапа получения зависимости:

load x; load s;

load x1; load s1;

[xn,meanx,stdx,sn,means,stds] = prestd(x,s);

x1n = trastd(x1, meanx, stdx);

mn = min(xn);

mx = max(xn);

MAX\_N = 31;

layers = 1:MAX\_N;

delta1 = 1:MAX\_N;

delta=1:MAX\_N;

for i=1:MAX\_N

l = 10+i-9;

layers(i) = l;

net = newff([mn mx],[l l 1], {'tansig' 'tansig' 'purelin'}, 'traingda');

net.trainParam.epochs = 2000;

net = train(net,xn,sn);

Y1n = sim(net,x1n);

Y1 = poststd(Y1n, means, stds);

Yn = sim(net, xn);

Y = poststd(Yn, means, stds);

delta(i) = mean((Y-s).^2);

delta1(i) = mean((Y1-s1).^2);

end

plot(layers, delta, layers, delta1)

На графике видно, что серьезного переобучения с ростом количества нейронов не происходит, но при l > 10 ошибка почти не уменьшается. Кроме того, именно при l = 10 ошибка на тестовых данных начинает отклоняться от тренировочных, что говорит о каком-то переобучении. По результатам эксперимента можно сделать вывод, что оптимальное количество нейронов в MLP для данной задачи: 20.