

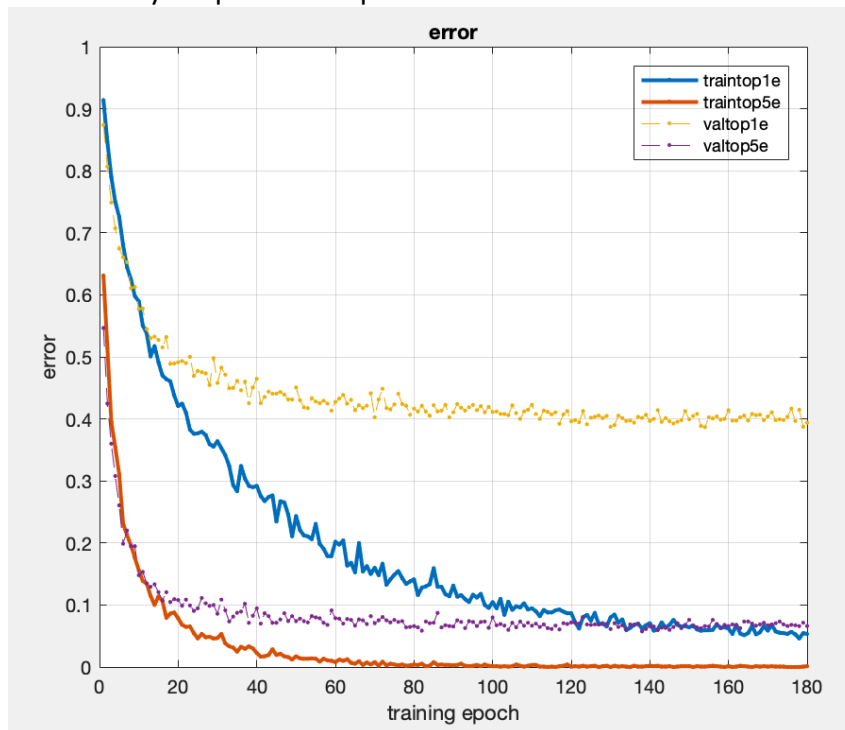
## Homework 4

### 4.3 Writing questions

1. We do this so that there are no scaling factors affect different images (thereby the word normalization).
2. Every time we do a conv layer, the number of data will seem to increase drastically because conv layer tend to have very deep dimensions. Therefore, we use the pooling layer to reduce the amount of data we get so that the calculation will be faster with just a little less accuracy.
3. The sigmoid function, since it is asymptotic, its value does not scale with the large input: at very large input, the derivative of sigmoid tends to 0, makes the backpropagation progress slower.

The ReLU function, on the other hand, has a point, the zero-point, that is not differentiable. And if some input falls onto that point, the outcome is unpredictable.

Below is my output for deeper neural network:



Lowest validation error is 0.386667

ans =

struct with fields:

layers: {1×9 cell}

Below is my configuration for the neural network:

```
net.layers{end+1} = struct('type', 'conv', ...
    'weights', {{f*randn(9,9,1,10, 'single')}, zeros(1, 10, 'single')}} , ...
    'stride', 1, ...
    'pad', 0, ...
    'name', 'conv1') ;

net.layers{end+1} = struct('type', 'pool', ...
    'method', 'max', ...
    'pool', [7 7], ...
    'stride', 7, ...
    'pad', 0) ;

net.layers{end+1} = struct('type', 'relu') ;

% PROBLEM 3, 4: YOUR CODE HERE -----
% feel free to add or modify code anywhere in this file;
% this is just a suggested place to start
% net.layers{end+1} = struct('type', 'dropout', 'rate', 0.5);

net.layers{end+1} = struct('type', 'conv', ...
    'weights', {{f*randn(5,5,10,30, 'single')}, zeros(1, 30, 'single')}} , ...
    'stride', 1, ...
    'pad', 2, ...
    'name', 'conv2') ;

|
net.layers{end+1} = struct('type', 'pool', ...
    'method', 'max', ...
    'pool', [3 3], ...
    'stride', 1, ...
    'pad', 1) ;

net.layers{end+1} = struct('type', 'relu') ;

net.layers{end+1} = struct('type', 'dropout', 'rate', 0.55);

net.layers{end+1} = struct('type', 'conv', ...
    'weights', {{f*randn(8,8,30,15, 'single')}, zeros(1, 15, 'single')}} , ...
    'stride', 1, ...
    'pad', 0, ...
    'name', 'fc1') ;
```

With learning rate and epoch configured by `logspace(-3.5, -4.5, 180)`