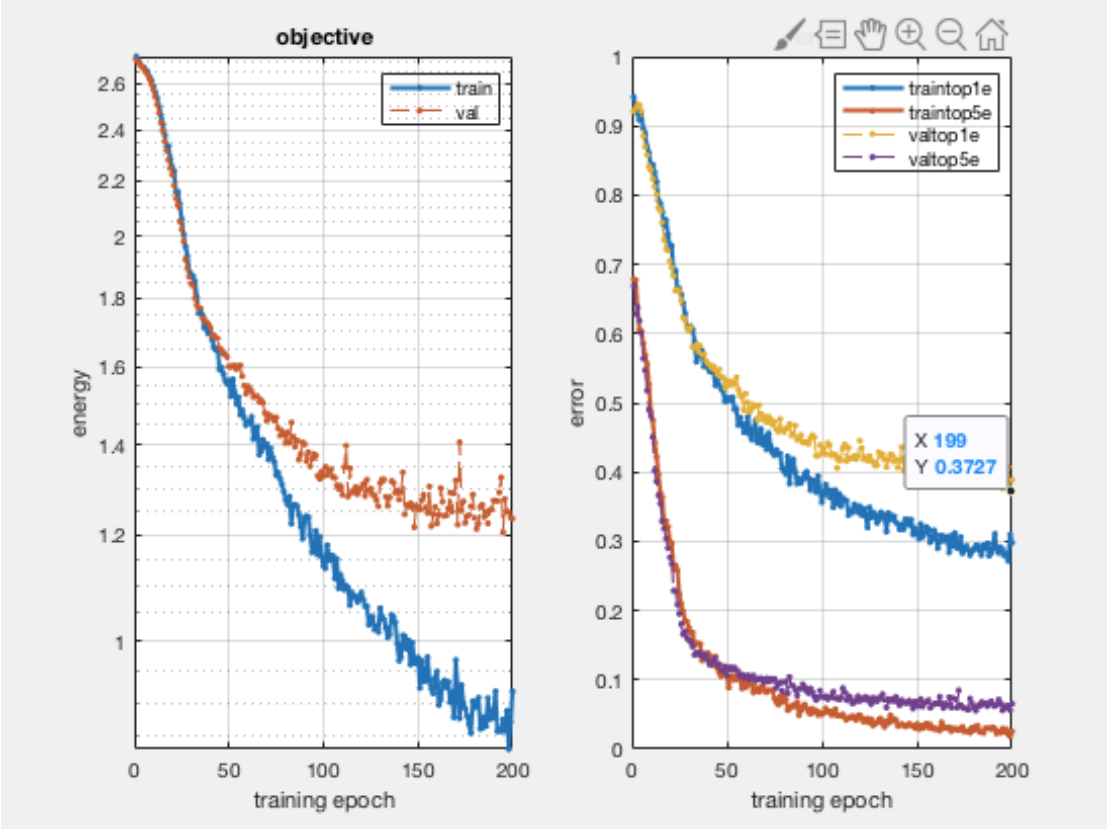
CSE152

Homework4

PID: A15677346

Name: ShihHan Chan

4.2



Lowest validation error (validation top1 error) is 0.372667

4.3

1. Explain why subtracting the mean from our images improves the performance of our neural network.

It's a kind of normalization that makes the network performance better. And we do this mainly because we want different features to have similar range to to make sure the gradient will not go out of control in back propagation.

Secondly, deep neural network shares many parameters, if you don't scale your input that resulted in a similar-ranged value, sharing parameters may not be so effective since some of the parameter weight is much bigger than others.

2. Describe the role that pooling plays in the convolutional neural network. How does pooling change the number of parameters in the network, and what effects does this have?

The role that the pooling plays in convolutional neural network is a non-trainable layer which takes groups of input and apply functions to each of these groups independently. It can help extract the feature we want, like max-pooling extracts the biggest value, average-pooling extracts the average value.

And the pooling layer can reduce the dimensionality of the data (reduce the number of parameters)

And the biggest effect of the pooling layer is “invariance”. In many cases, we want our model to be invariant to small perturbations of the data since they are not so important. And pooling layer can help us.

3. What issues might arise in using sigmoids as activation functions? How does the rectified linear unit function address these issues? Can you describe problems that may arise with using ReLU?

Sigmoid function has “vanishing gradient” problem. It means that the output value of the sigmoid function will be very close to 0 or 1 after many multiplications between output from previous layer and matrix of weights (sigmoid function takes very big or very small value as input will output value that is very close to 0 or 1). If the output become very close to 0 or 1, the derivative of sigmoid will become close to 0, and all the gradient of parameters will become very close to 0 during back propagation since the chain rule. And it causes local gradient to vanish and stop learning.
Since the gradient of ReLU is either 0 or 1, it never saturates, and the gradient cannot vanish. The gradient is transferred through the whole network during back propagation. So it solves the problem of sigmoid function and it’s useful in deep network.
Since ReLU has 0 gradient on the left-hand side, it will cause a problem called “dead neurons”. It means that some dead neurons always output same value 0 for negative input. It means that there is no way to discriminate between the input and the parameter will not update during the back propagation. We can use leaky ReLU to give a small positive gradient for negative input to solve this. Furthermore, ReLU has

unbounded positive response, so it may cause faster convergence or overstep