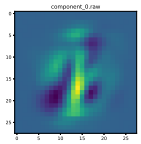
1 Question 1: Linear Layer

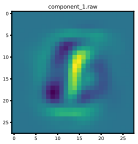
Done as instructed; tests passed.

2 Question 2: Optimization Algorithm

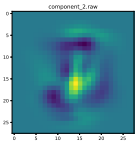
Done as instructed including the L2 penalization.
Finished with: Training set MSE:13.520405, Test set MSE:13.381790. The 10 principal components of the MNIST data set obtained by using a single linear hidden layer are shown in figure 1. These principal components clearly show the main 'activity' in the center of the figures where all digits have 'active' regions.



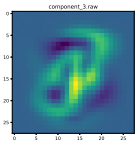
(a) Component 0



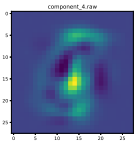
(b) Component 1



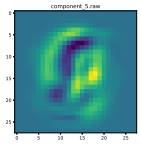
(c) Component 2



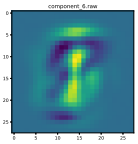
(d) Component 3



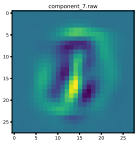
(e) Component 4



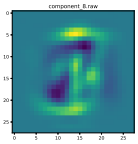
(f) Component 5



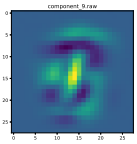
(g) Component 6



(h) Component 7



(i) Component 8

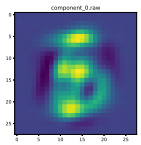


(j) Component 9

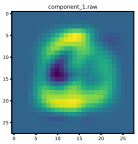
Figure 1: 10 Principal components of MNIST data set obtained from single linear hidden layer.

3 Question 3: Non-linearity

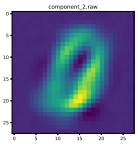
Done as instructed; tests passed.
Finished with: Training set MSE:13.538446, Test set MSE:13.395468. The 10 principal components of the MNIST data set obtained by using three hidden layers with tanh activation function are shown in figure 2. Now certain elements of specific digits are clearly identifiable among the principal components, which is clearly owed to the much larger network (2 additional hidden layers with 100 neurons each) as well as the non-linear activation function.



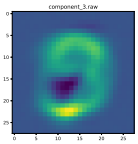
(a) Component 0



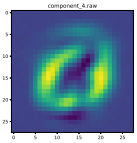
(b) Component 1



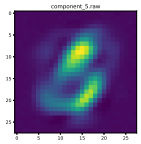
(c) Component 2



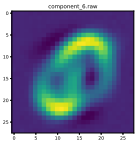
(d) Component 3



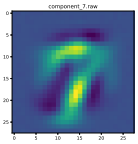
(e) Component 4



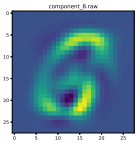
(f) Component 5



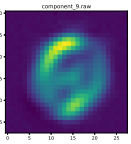
(g) Component 6



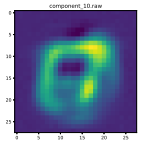
(h) Component 7



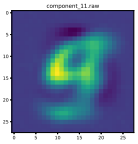
(i) Component 8



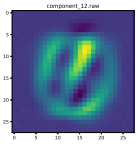
(j) Component 9



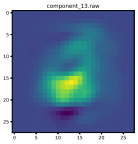
(k) -Component 0



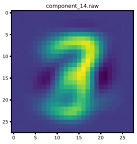
(l) -Component 1



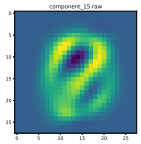
(m) -Component 2



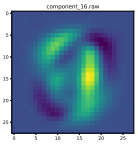
(n) -Component 3



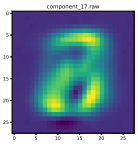
(o) -Component 4



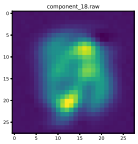
(p) -Component 5



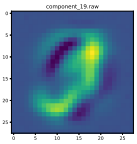
(q) -Component 6



(r) -Component 7



(s) -Component 8



(t) -Component 9

Figure 2: 10 Principal components of MNIST data set obtained from three hidden layers with tanh activation function.

4 Question 4: Parallelization

Most parallelization loops can be done in a straightforward way based on `#pragma omp parallel for`. A few loops however required special care:

- In `LinearLayer::bckward`, the loops aren't perfectly nested and therefore can't be collapsed. Thus, only the inner loop was OMP'ed.
- In `main_linear.cpp` starting on line 88, the operation on `std::vector<int> sample_ids` is not thread-safe. This was solved by declaring a private variable and using a `critical` section for `pop_back`.
- In `main_linear.cpp` starting on lines 97 and 128, updating `epoch_mse` (or `test_mse`, respectively) leads to a data race. This was solved with a `reduction(+: __ _mse)` statement.

I tested my implementation for correctness with different numbers of threads (especially the edge cases of one single and the maximum number of threads). While there were no errors (passed all test; achieved same training/test errors), the parallelization did not lead to any noticeable speedups. I imagine this is due to the comparably small batch sizes and both small images and overall small size of the neural network.