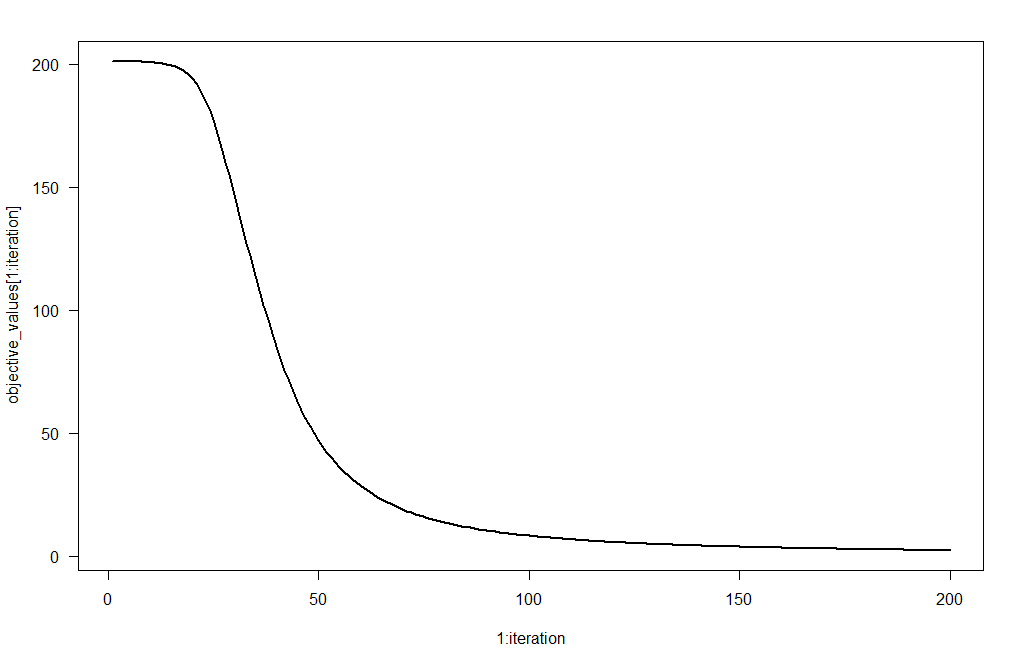
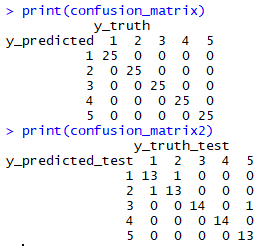
**Introduction to Machine Learning HW3**

1. I first read the csv files and converted the labels to numeric format, i.e. A, B, C, D, E to 1, 2, 3, 4, 5, respectively.
2. I introduced given epsilon, eta, max\_iteration and D, H, N parameters
3. I converted label matrices for train and test data into proper format by separating each true values for labels in their according column in ones and zeros (y\_truth 🡪 y\_truth\_new)
4. I defined the safelog, sigmoid, and softmax functions that will be needed later.
5. I initialized W and V parameter sets. Different than 2-class discrimination, V should have been a matrix instead of a vector this time, with a dimension of (H+1)xK. The dimension of W stays the same ((D+1)xK).
6. I ran the functions once to find initial values of Z and y\_predicted. Then, I iterated the algorithm that minimizes the objective value which is to minimize total error. I introduced V\_temp and W\_temp that are actually the gradients for V and W. They have the values according to their functions that are discussed in the book.
7. I plotted the objective values versus iterations and printed confusion matrix of tranining data. Lasly, I used trained parameters to predict the test data and printed its confusion matrix, too.

*I shared the results next page in case the .r file does not work properly.*

**

> objective\_values

[1] 201.186648 201.160556 201.139791 201.116938 201.089819 201.056184

[7] 201.013267 200.957512 200.884247 200.787260 200.658242 200.486054

[13] 200.255778 199.947511 199.534877 198.983287 198.248090 197.272918

[19] 195.988857 194.315416 192.164606 189.449434 186.097219 182.066076

[25] 177.359953 172.035626 166.196529 159.973855 153.502009 146.898157

[31] 140.252276 133.627847 127.068899 120.608417 114.274906 108.095986

[37] 102.099270 96.311480 90.756796 85.455283 80.421833 75.665755

[43] 71.190898 66.996171 63.076253 59.422402 56.023265 52.865633

[49] 49.935124 47.216761 44.695456 42.356379 40.185244 38.168496

[55] 36.293442 34.548307 32.922258 31.405388 29.988675 28.663934

[61] 27.423748 26.261404 25.170828 24.146516 23.183478 22.277179

[67] 21.423492 20.618651 19.859212 19.142020 18.464177 17.823021

[73] 17.216095 16.641134 16.096045 15.578892 15.087883 14.621357

[79] 14.177774 13.755704 13.353820 12.970889 12.605765 12.257381

[85] 11.924746 11.606936 11.303091 11.012408 10.734140 10.467590

[91] 10.212106 9.967081 9.731945 9.506169 9.289253 9.080734

[97] 8.880175 8.687168 8.501330 8.322300 8.149742 7.983339

[103] 7.822792 7.667821 7.518162 7.373566 7.233799 7.098641

[109] 6.967883 6.841328 6.718790 6.600094 6.485074 6.373573

[115] 6.265441 6.160538 6.058731 5.959892 5.863902 5.770646

[121] 5.680017 5.591911 5.506230 5.422881 5.341776 5.262831

[127] 5.185964 5.111100 5.038165 4.967089 4.897807 4.830255

[133] 4.764371 4.700099 4.637382 4.576169 4.516407 4.458049

[139] 4.401048 4.345360 4.290941 4.237752 4.185753 4.134906

[145] 4.085175 4.036526 3.988925 3.942340 3.896741 3.852098

[151] 3.808382 3.765567 3.723625 3.682533 3.642264 3.602797

[157] 3.564107 3.526174 3.488976 3.452493 3.416706 3.381595

[163] 3.347143 3.313331 3.280144 3.247563 3.215574 3.184162

[169] 3.153311 3.123006 3.093236 3.063985 3.035240 3.006991

[175] 2.979223 2.951926 2.925088 2.898697 2.872744 2.847218

[181] 2.822109 2.797407 2.773102 2.749186 2.725650 2.702484

[187] 2.679681 2.657233 2.635131 2.613368 2.591936 2.570828

[193] 2.550038 2.529557 2.509381 2.489501 2.469912 2.450608

[199] 2.431583 2.412831 2.394346