AI5031: Machine learning, exercise sheet 3

1 Preparations

Obtain the MNIST data as you did last week if you did not store it. Connect to a LinuxLab machine as shown last week if you are not in LinuxLab.
The following functions and methods may be useful:

- np.dot() or method of an array instance: perform matrix multiplications
- broadcasting can be used to compute outer products
- the *np.newaxis* object can be used in a slicing experession to indicate the insertion of a new one-element axis. Example: *x[:,np.newaxis,:]*. If x had shape (10,11) before, now it has shape (10,1,11). This is especially useful for brocasting!

2 Component-wise data analysis

a) Display the MNIST training data average image! For this, compute the per-pixel mean over all training samples and store it (will be re-used!). Display it using the appropriate matplotlib function!
b) Compute the per-pixel variance over all training samples and display it.
c) Compute the per-pixel standard deviation over all training samples and display it.
d) Slice out the first 500 training samples, and plot a histogram of the values of pixel (13,13) over all 500 samples. Do the same for pixel (0,0). How do you interpret these histograms?
e) Compute the per-sample min and max values and their average over training samples.
f) Compute the per-pixel max values over all training samples and display them as an image. What does this image tell us?

3 Data analysis

a) Slide out MNIST sample 500, flatten it and compute its outer product with itself using broadcasting and *np.newaxis*. Display the result as an image!
b) Do the same thing for the first 1000 samples using broadcasting, then average all outer products over samples. Here, flattening has to be replaced by an appropriate reshape command. The result is the covariance matrix C of shape (784,784). **c)** Compute the eigenvalues of this matrix using the numpy function *eigh*, look up the documentation. Display the first and second eigenvector, reshaped to a 2D image. Careful: the first eigenvector is the last column in the eigenvector array returned by *eigh*.