HOMEWORK PROBLEMS 12, ANLY 561, FALL 2018
DUE 12/05/18

Readings: §9.1, 9.2, 9.3 of **Goodfellow, Bengio, and Courville**. Chapter 11 of **Géron**.

Exercises:

1. Consider the example feedforward neural network and block backtracking code in Lecture 4 Part II. This code creates a loss function and computes the gradient of this loss function for training a three layer neural network having 30 nodes in the input layer, 20 logistic units in a single hidden layer, and softmax activations for a two dimensional vector at the output layer. Modify this code to create a loss function and its gradient for a **four** layer feedforward neural network, where there are now two hidden layers each with 20 logistic units.

    (a) Using the first 400 examples from the Wisconsin Breast Cancer dataset, run 100 steps of gradient descent with block backtracking to train your four layer neural network. Use the `random_matrix` function to randomly initialize your weight variables, and use the random seed `1234` to keep the behavior of your program deterministic. Keep all other variables (e.g. α and β) fixed, and report the final test accuracy after running gradient descent 100 times.

    (b) List three ways that you could make this implementation more efficient (that is, make it use less memory or less time).

2. Convolutional neural networks employ convolution of stacks of images that output a single image. For example, if we convolve the stack of images

$$\left(\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 1\end{pmatrix}, \begin{pmatrix}1 & 1 & 1\\ 1 & 0 & 1\\ 1 & 1 & 1\end{pmatrix}\right)$$

with the *filter*

$$\mathcal{H} = \left(\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}\right),$$

we get

$$\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 1\end{pmatrix} * \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} + \begin{pmatrix}1 & 1 & 1\\ 1 & 0 & 1\\ 1 & 1 & 1\end{pmatrix} * \begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix} = \begin{pmatrix}2 & 0\\ 0 & 2\end{pmatrix} + \begin{pmatrix}3 & 3\\ 3 & 3\end{pmatrix} = \begin{pmatrix}5 & 3\\ 3 & 5\end{pmatrix}$$

    (a) Express convolution of a 3 by 3 matrix X with the matrix

$$\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

    as contraction of X with a 2 by 2 by 3 by 3 tensor. In particular, explicitly write down this 4th order tensor and indicate the indices along which contraction should occur. *Hint*: This should be simple to write down if you choose the right slices for the 2 by 2 by 3 by 3 tensor.

    (b) Express the convolution of a 2 by 3 by 3 tensor $\mathcal{X}$ with the above 2 by 2 by 2 $\mathcal{H}$ as contraction with a 2 by 2 by 2 by 3 by 3 tensor. In particular, explicitly write down this 5th order tensor and indicate the indices along which contraction occurs.

3. Provide a one page outline for your project's white paper.

4. **This is an optional problem worth +20 points:** Why is it a bad idea to initialize $C = \mathbf{0}$ and $V = \mathbf{0}$ when performing gradient descent or alternating minimization of $\|X - CV\|^2$ in the low-rank approximation program?

5. **This is an optional problem worth +20 points:**

   (a) Derive update rules for non-negative matrix factorization when only certain entries of X are known. That is, determine updates for the non-negative matrix completion problem.

   (b) Code up the iterative method from part (a) and run it for 1000 iterations on the incomplete MovieLens dataset to produce a rank $k = 100$ approximation that imputes the entries of X. Initialize C and V randomly using the same two lines used the non-negative matrix factorization example shown in lecture.

   (c) Report the norm-square error of the approximation found in part (b), and use `matplotlib.pyplot.imshow` to display CV, C, and V. Use the 5th and 95th percentiles of the values of these matrices when plotting to exclude outliers that may distort the color scale, as is done in the Lecture 5 notes.