EE239AS, Winter 2018 **Homework #3**
Neural Networks & Deep Learning Prof. J.C. Kao
University of California, Los Angeles; Department of ECE TAs: T. Xing & C. Zheng

Due Monday, 5 Feb 2018, by 11:59pm to Gradescope.
100 points total.

1. (X points) **Backpropagation for autoencoders.** In an autoencoder, we seek to reconstruct the original data after some operation that reduces the data's dimensionality. We may be interested in reducing the data's dimensionality to gain a more compact representation of the data.

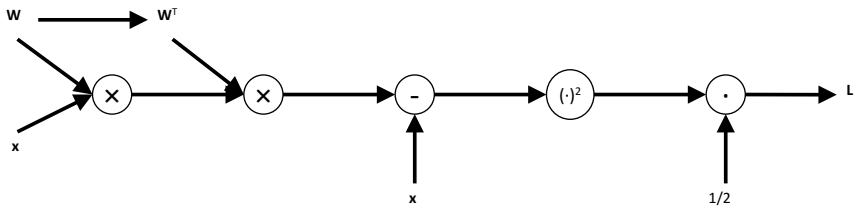
   For example, consider $\mathbf{x} \in \mathbb{R}^n$. Further, consider $\mathbf{W} \in \mathbb{R}^{m \times n}$ where $m < n$. Then $\mathbf{Wx}$ is of lower dimensionality than $\mathbf{x}$. One way to design $\mathbf{W}$ so that it still contains key features of $\mathbf{x}$ is to minimize the following expression

   $$\mathcal{L} = \frac{1}{2} \left\| \mathbf{W}^T \mathbf{W} \mathbf{x} - \mathbf{x} \right\|^2$$

   with respect to $\mathbf{W}$. (To be complete, autoencoders also have a nonlinearity in each layer, i.e., the loss is $\frac{1}{2} \left\| f(\mathbf{W}^T f(\mathbf{Wx})) - x \right\|^2$. However, we'll work with the linear example.)

   (a) (X points) In words, describe why this minimization finds a $\mathbf{W}$ that ought to preserve information about $\mathbf{x}$.
   Solution: With the loss to be minimized, the difference between output $\mathbf{W}^T\mathbf{Wx}$ and input $\mathbf{x}$ will be minimized. So the hidden representation $\mathbf{Wx}$ will preserve the information about $\mathbf{x}$.

   (b) (X points) Draw the computational graph for $\mathcal{L}$.
   Solution:



   (c) (X points) In the computational graph, there should be two paths to $\mathbf{W}$. How do we account for these two paths when calculating $\nabla_{\mathbf{W}} \mathcal{L}$? Your answer should include a mathematical argument.
   Solution: Take a simple example. If the computational graph is composed of two path: $a \to b \to d \to e$ and $a \to c \to d \to e$. In this case, a contributes to e along two paths. Hence, the total derivative of e with respect to as is given by:

   $$\frac{\partial e}{\partial a} = \frac{\partial e}{\partial d} \cdot \frac{\partial d}{\partial b} \cdot \frac{\partial b}{\partial a} + \frac{\partial e}{\partial d} \cdot \frac{\partial d}{\partial c} \cdot \frac{\partial c}{\partial a}$$

(d) (X points) Calculate the gradient: $\nabla_{\mathbf{W}}\mathcal{L}$.
Solution: Let $\mathbf{K} = \mathbf{W}^T\mathbf{W}\mathbf{x} - \mathbf{x} \in \mathbb{R}^n$ and $\mathbf{P} = \mathbf{W}\mathbf{x} \in \mathbb{R}^m$, so

$$\frac{\partial \mathcal{L}}{\partial \mathbf{K}} = \mathbf{K}$$

Backpropagate to $\mathbf{W}^T$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^T} = \mathbf{K}(\mathbf{W}\mathbf{x})^T$$

Backpropagate to $\mathbf{P}$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{P}} = \mathbf{W}\mathbf{K}$$

Backpropagate to $\mathbf{W}$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \mathbf{W}\mathbf{K}\mathbf{x}^T$$

$\nabla_{\mathbf{W}}\mathcal{L}$ = backprop to $\mathbf{W}$ + backprop to $\mathbf{W}^T$

$$\nabla_{\mathbf{W}}\mathcal{L} = \mathbf{W}\mathbf{K}\mathbf{x}^T + \mathbf{W}\mathbf{x}\mathbf{K}^T$$

2. (X points) **Backpropagation for Gaussian-process latent variable model.** An important component of unsupervised learning is visualizing high-dimensional data in low-dimensional spaces. One such nonlinear algorithm to do so is from Lawrence, NIPS 2004, called GP-LVM. GP-LVM optimizes the maximum-likelihood of a probabilistic model. We won't get into the details here, but rather to the bottom line: in this paper, a log-likelihood has to be differentiated with respect to a matrix to derive the optimal parameters.

To do so, we will use apply the chain rule for multivariate derivatives via backpropagation. The log-likelihood is:

$$\mathcal{L} = -c - \frac{D}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T)$$

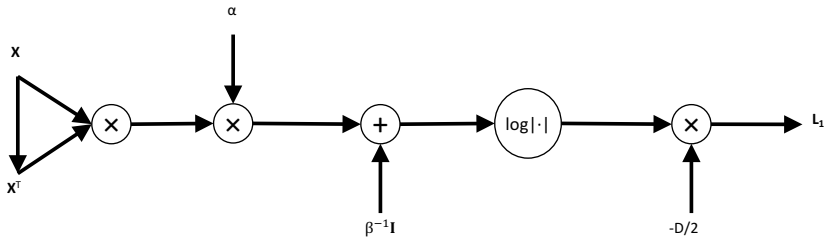
where $\mathbf{K} = \alpha\mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbf{I}$. To solve this, we'll take the derivatives with respect to the two terms with dependencies on $\mathbf{X}$:

$$\begin{aligned}\mathcal{L}_1 &= -\frac{D}{2}\log|\alpha\mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbf{I}| \\ \mathcal{L}_2 &= -\frac{1}{2}\mathrm{tr}\left((\alpha\mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbf{I})^{-1}\mathbf{Y}\mathbf{Y}^T\right)\end{aligned}$$

Hint: Lawrence states the gradients in his paper, so you can check your answer. To receive full credit, you will be required to show all work. You may find following equation useful:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{K}} = -\mathbf{K}^{-1}\frac{\partial \mathcal{L}}{\partial \mathbf{K}^{-1}}\mathbf{K}^{-1}$$

(a) (X points) Draw the computational graph for $\mathcal{L}_1$.
Solution:



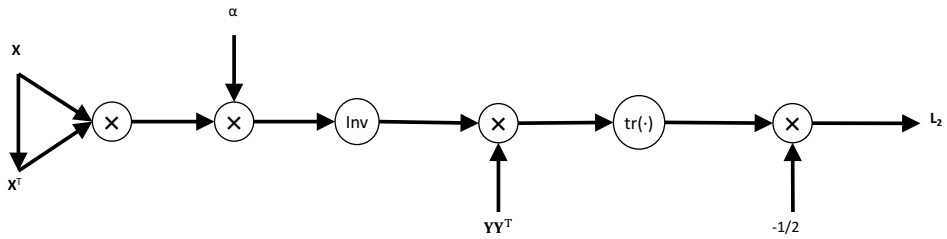
(b) (X points) Compute $\frac{\partial \mathcal{L}_1}{\partial \mathbf{X}}$.
Solution: Calculate the derivative by backpropagation.

$$\frac{\partial \mathcal{L}_1}{\partial \mathbf{K}} = \frac{-D}{2}(\mathbf{K}^T)^{-1}$$

$$\frac{\partial \mathcal{L}_1}{\partial \mathbf{X}\mathbf{X}^T} = \frac{-\alpha D}{2}(\mathbf{K}^T)^{-1}$$

$$\frac{\partial \mathcal{L}_1}{\partial \mathbf{X}} = -\alpha D(\mathbf{K}^T)^{-1}\mathbf{X} = -\alpha D\mathbf{K}^{-1}\mathbf{X}$$

(c) (X points) Draw the computational graph for $\mathcal{L}_2$.
Solution:



(d) (X points) Compute $\frac{\partial \mathcal{L}_2}{\partial \mathbf{X}}$.
Solution: Calculate the derivative by backpropagation.

$$\frac{\partial \mathcal{L}_2}{\partial \mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T} = -\frac{1}{2}\mathbf{I}$$

$$\frac{\partial \mathcal{L}_2}{\partial \mathbf{K}^{-1}} = -\frac{1}{2}\mathbf{Y}\mathbf{Y}^T$$

$$\frac{\partial \mathcal{L}_2}{\partial \mathbf{K}} = \frac{1}{2}\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1}$$

$$\frac{\partial \mathcal{L}_2}{\partial \mathbf{X}} = \alpha\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1}\mathbf{X}$$

(e) (X points) Compute $\frac{\partial \mathcal{L}}{\partial \mathbf{X}}$.
Solution:

$$\frac{\partial \mathcal{L}_2}{\partial \mathbf{X}} = \alpha\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1}\mathbf{X} - \alpha D\mathbf{K}^{-1}\mathbf{X}$$

3. (X points) **2-layer neural network.** Compete the two-layer neural network Jupyter notebook. Print out the entire workbook and relevant code and submit it as a pdf to gradescope.

4. (X points) **General FC neural network.** Compete the FC Net Jupyter notebook. Print out the entire workbook and relevant code and submit it as a pdf to gradescope.