hannah-small-hw2

Question 1. Fourier Basis and Sparse Coding (15 points)

1.

Fourier transform

$$I(\omega) = \frac{1}{2\pi} \int \hat{I}(x) e^{-i\omega \cdot x} dx$$

inverse fourier transform

$$\hat{I}(x) = \frac{1}{2\pi} \int I(\omega) e^{i\omega \cdot x} d\omega$$

Representing an image with

$$I(x) = \sum_{i=1}^{N} \alpha_i b_i(x)$$

where b_i are the basis functions and the alpha are the coefficients of the bases. When you have a orthonormal set of basis functions, you can estimate the coefficients alpha by

$$\alpha_i = \sum_x b_i(x) \cdot I(x)$$

2. Sparsity is a principle for modeling the brain that was motivated by the observation that only a small number of neurons are active. Specifically for vision, cells in V1 fire sparsely, suggesting that they are tuned for specific stimuli and may be reducing the amount of firing for metabolic reasons. Sparse coding involves an over-complete set of basis functions. An image is represented by Fourier analysis and unit impulses, which provides an over-complete set of basis functions. The Fourier analysis basis functions more easily represent the smoothly varying regions and the impulse basis functions more easily represent the jagged intensity regions. The sparsity penalty

selects a small number of basis functions that are activated for each image, creating an efficient representation.

The larger λ is, the larger the degree of sparsity. This term weights the sum of the magnitude of the coefficients. Since we are minimizing this (along with the error of the approximation of the image), the larger the λ is, the smaller the sum of the magnitude of the coefficients will have to be.

$$f(x;a) = (x{-}a)^2 + \lambda|x|$$

$$\hat{x} = argminf(x;a)$$

Show that, for some values of a, the answer is sparse

First, we find the minimum of the function (differentiating and setting to zero)

$$\frac{df_+}{da} = 2(x-a) + \lambda \qquad \frac{df_-}{da} = 2(x-a) - \lambda$$

$$0 = 2(x-a) + \lambda \qquad 0 = 2(x-a) - \lambda$$

$$-\lambda/2 + a = x \qquad \lambda/2 + a = x$$

then, we plug in x = 0 to find when a is sparse

$$-\lambda/2 + a = 0 \qquad \lambda/2 + a = 0$$

the answer is sparse when:

$$a = \lambda/2, -\lambda/2$$

Question 2. Hebbian Learning and Binocular Stereo (15 points)

1.

Hebbian learning at its core is not a kind of supervised learning because it does not use labeled data during training. It uses properties of the neurons (neurons that fire together, wire together) to learn. However, Hebbian can be applied to both supervised and unsupervised contexts by updating the loss function to account for labels.

The purpose of the term that decreases the value of all weights by the amount proportional to their strength is to prevent strengths from going to infinity. The larger the weight, the larger the decrease in the output. If there is no term, the weights just grow forever and learning will not end.

2.

Stereokinetic effect:

This is an example of inferring depth from the motion. Priors of small movements observed during motion help to form the depth percept. A Gestalt general principle that minimizes speed differences can explain why we perceive the depth instead of a series of rotating circles.

Ames Windows illusion:

The Ames window illusion shows how perception is shaped by priors on the world. When we see the trapezoid shape, we interpret it as a rectangle that is extending into space, like viewing a fence in a field. We interpret it with perspective distortion. When the largest side of the fence is in the back in the illusion, we still interpret it as closest to us, since that's what we normally see when we are looking at a fence extending away from us. During the turning, when that interpretation is no longer sufficiently supported, our vision 'flips' so that it can be. This results in the ballerina seemingly moving through the fence.

3.

It is not possible for a single cell to detect disparity by itself when $\rho l - \rho r = \omega D$. The response is formulated as:

$$r_1 = 2\rho cos(\theta - \frac{\rho_l + \rho_r}{2} - \frac{\omega D}{2})cos(\frac{\rho_1 - \rho_r}{2} - \omega\frac{D}{2})$$

So, when ρl − ρr = ωD, the second cosine will be 1, its largest possible value. However, the response also depends on theta, which is an image property (phase). The cell would not be able to distinguish if the response is due to the disparity or the phase, so it cannot detect the disparity by itself. Disparity could be determined from the response of a population of quadrature cells tuned to different frequencies.

A neural network estimates disparity by 1) defining a set of disparity cells that are tuned to disparities and each of these have a vote (summed from quadrature pair inputs), and 2) having a winner-take-all network to compute the disparity with the biggest vote.

Question 3. Decision Theory (28 points)

1.

In Bayes Decision Theory, priors are probability of a hypothesis P(y). It is the existing data before observing data. The likelihood function is the probability of the data given y, P(x|y). The loss function is the cost of making a certain decision alpha(x) if the real decision should be y.

Risk is:

$$R(\alpha) = \sum_{x,y} P(x,y)L(\alpha(x),y)$$

Bayes Rule is:

$$\hat{\alpha} = argmin_{\alpha} R(\alpha)$$

and Bayes Risk is:

$$min_{\alpha} R(\alpha) = R(\hat{\alpha})$$

The maximum likelihood estimation occurs when the prior is a uniform distribution (P(y)=constant):

$$\alpha(x) = argmax P(x|y)$$

The maximum a posteriori estimation occurs when the loss function penalizes all errors by the same amount and is:

$$\alpha(x) = argmax(P(y|x))$$

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

2.

A false positive is the case where the output was positive but it should be negative.

A false negative is the case where the output was negative but it should be positive.

T = decision threshold

$$FP = \int_{T}^{\infty} P(x|y=-1)dx$$

$$= 1 - \int_{-\infty}^{T} P(x|y=-1)dx$$

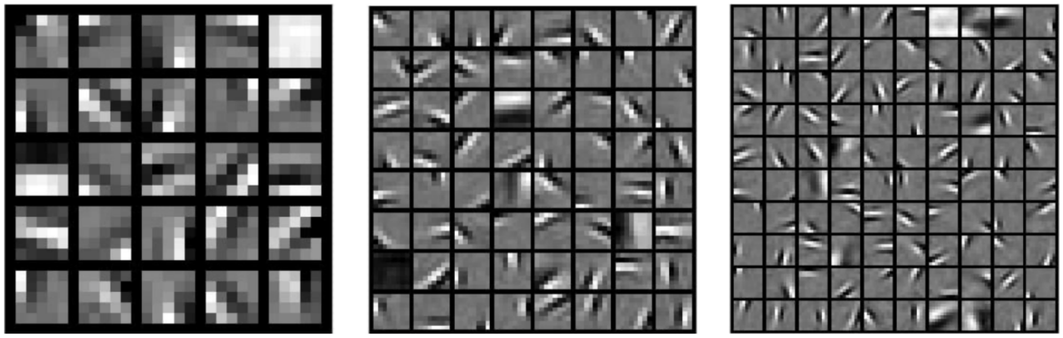
$$= 1 - \frac{1}{2}[1 + erf(\frac{T - uT}{\sigma\sqrt{2}})]$$

$$FN = \int_{-\infty}^{T} P(x|y=1)dx$$

$$= \frac{1}{2}[1 + erf(\frac{T - uD}{\sigma\sqrt{2}})]$$

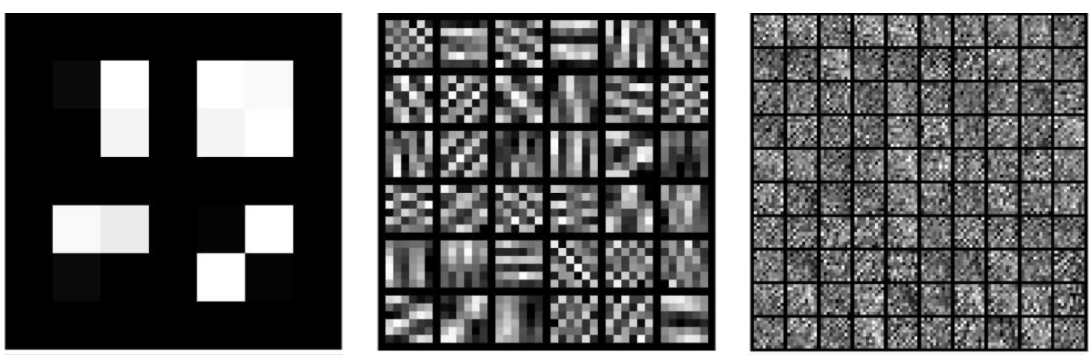
Question 4. Experimental Section: Sparse Coding (20 points)

1. Sparse coding of natural images.



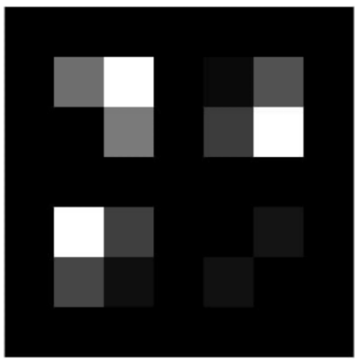
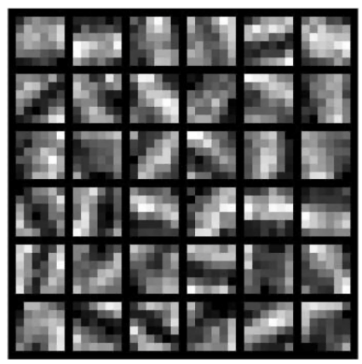
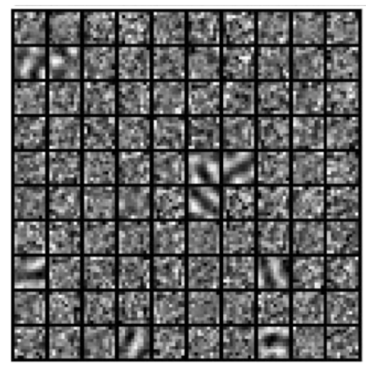
This code finds basis functions that describe the input images in a sparse manner. For each image, the image is broken down into random subimages and the coefficients are calculated using the current basis functions (initialized randomly), using a conjugate gradient routine. The residual error of the input image and the reconstructed image is used to update the basis functions, which are then normalized.

2. Sparse coding of high-frequency sinusoid images.



The larger M is, the more spatial localizations, orientations, and scales there are in the bases, allowing the detection of more varied features. This is very evident moving from M=4 to M=36, where you an increase in the different orientations and sizes of filters. When M=100, it looks like more of the bases look like one another, suggesting more over-completeness. This makes sense because the more bases you have, the more the algorithm would allow overcompleteness if it minimizes the reconstruction error while effectively describes structures in the image.

3. Sparse coding of low-frequency sinusoid images.

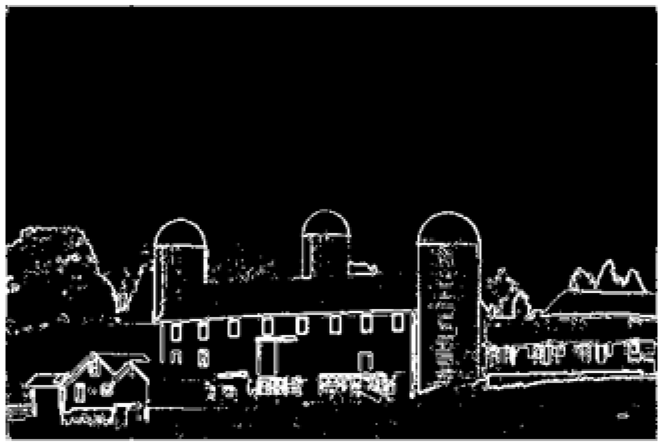




Again, the larger M is, the more spatial localizations, orientations, and scales there are. Compared to the high-frequency bases, these are a bit 'blurrier'. It seems that the bases resulting from low-frequency sinusoids are more sensitive to larger spatial scales (there are more bases that have larger spatial scales), which makes sense since low-freq sinusoids are 'wider' that high-frequency sinusoids, therefore to minimize the reconstruction error, the sparsity algorithm allows more bases to be tuned to larger spatial structures. This would be the opposite in bases created for high-frequency sinusoids — in that case, more bases would be devoted to small spatial structures. Compared to the bases created from natural images , there is less spatial localization here, which makes sense because sinusoids do not have good spatial localization.

Question 5. Experimental Section: Edge Detection (14 points)

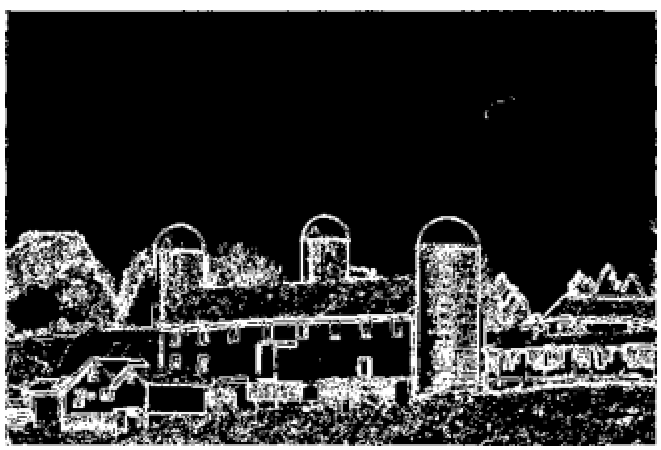
1. Load another image (e.g. the butterfly or building), apply this edge detection algorithm, find a good threshold and display your result (6 points)

   threshold = 1.3

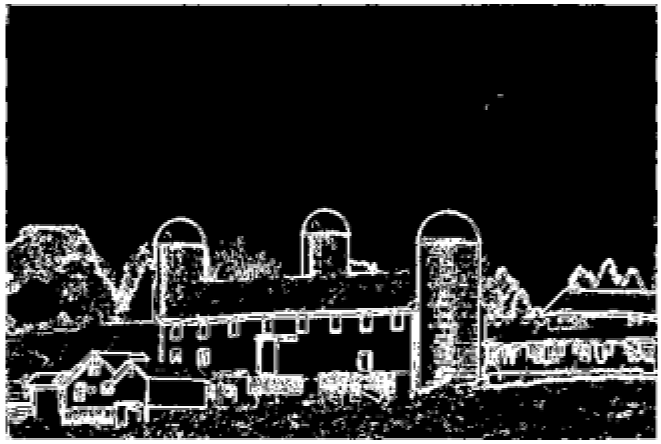


2. Use $dG*Idx$ instead of $dIdx$ for edge detection where G is a Gaussian. Show results for a couple of different variances `sigma`. (8 points)

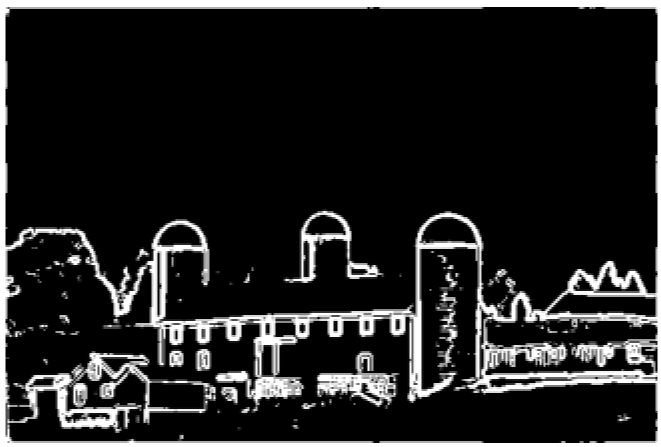
   a. sigma = 0.1



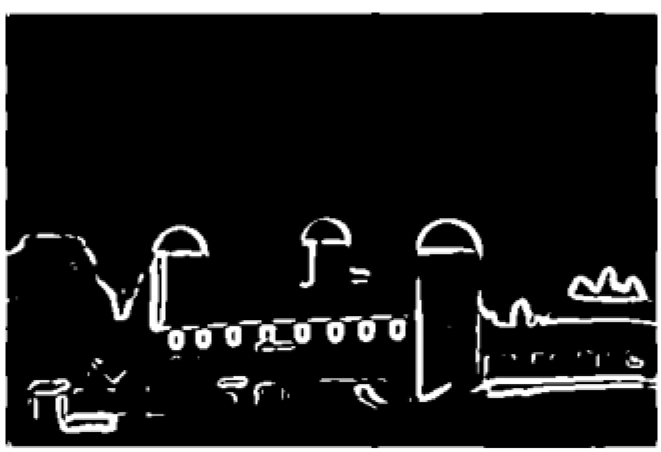
sigma = 0.5



sigma = 1



sigma = 2



sigma = 5