These are intended mainly as *suggestions* to help seed some ideas, not a menu that you must select from. Some are very ambitious so you are welcome to consider a scaled-down version, or to take the suggestion in a different direction. The point is to explore an idea or question from the course that interests you and to *learn* from the experience.

Dendritic nonlinearities

As we discussed in class, a single neuron is a much more powerful computational device than a Perceptron due to the nonlinear signal integration that occurs within dendritic trees. Demonstrate this by using a biophysically realistic model of a neuron (or a network of such neurons) to perform a pattern classification (or other) task. How many Perceptron models would be needed to obtain similar performance? One advantage of of the Perceptron model is that we can derive a learning rule for changing the weights to perform a task. How would you train a realistic neuron model with dendritic nonlinearities? See papers of Bartlett Mel, and for ideas about learning see his “Clusteron model.”

Visualization of retinal neural images

As you walk around in starlight, the photon catch rate for rods is roughly 1/500 to 1/50 per second per rod. SL figure 8.2 attempts to show what this would look like over a rod array when viewing a baboon, including the dark light. But this isn’t a very good visualization for a number of reasons: 1) it should be a movie rather than a static image, 2) the brain never gets to see the output of individual rods, but rather the retinal ganglion cells which get their input from rod bipolar cells that integrate over pools over 100 or more rods. Create a biophysically realistic animation that shows a scene as it would appear at several different neural stages — on the rod array, on the rod bipolar cells, and in the spiking activity of retinal ganglion cells. And do this at a variety of light levels - starlight, moonlight, dusk - up to about 1 cd/m^2 (which is where rods begin to saturate). For the rod bipolar cells it will be important to include the effect of thresholding synapses (or try with and without to demonstrate its significance), and you can find out more about that from this paper. Such an animation would be of great value in communicating both to the public and neuroscientists just how much of our perception relies upon filling in by the brain.

Silicon retina

Building on the resistive network discussed in class, implement a 2D network as a hexagonal grid and show how it blurs an image. Show the resulting difference image you get from subtracting this from the input image (i.e., the Mahowald/Mead silicon retina). Then making realistic assumptions about capacitance in horizontal cells, simulate the dynamical properties of this system as a function of R and C. From your leaky integrator equation, you should find that $\tau \approx RC/2$ (assuming axial resistance is much less than membrane resistance). When the time constant is small, then the effect of injecting current into any one node will spread quickly to neighboring nodes, and vice-versa when it is large. (Recall that to make a neuron faster, you need to make it larger. Here we can see why: as you increase the diameter D of the cable, R will decrease (more cytoplasm to conduct charge) and C will increase (more membrane). However R decreases proportional to D^2 (area) whereas C increases proportional to D (circumference). Thus, τ falls as 1/D.) Show how a current injected into one (or multiple) node(s) spreads over time for a variety of assumed diameters in the horizontal cell circuit.

Filtering via convolution vs. network recurrence

The silicon retina shows how a simple resistor network can implement a lowpass filter. Essentially the cable is a kind of recurrent neural network that computes the convolution with the kernel $e^{-|r|}$, where r is the radius from the center of the kernel. But what if you wanted to implement convolution with a different kind of kernel, such as the oriented kernels used in convnets? Typically these are computed in an FIR (finite impulse response) fashion, as a weighted sum:$y = Wx$. But note that if we have a recurrent network such as above then we can compute y via $\tau\dot{y} + y = x + My$ where the matrix M represents a set of linkages (conductances) to neighboring units. This has the equilibrium solution $y = (I - M)^{-1}x$. So if we design M appropriately we should be able to implement the desired W. And if we are lucky maybe we can do it just with a few local linkages to neighboring nodes. However one downside here is that such a recurrent network will have a settling time dictated both by τ and the structure of M. So we can also consider a hybrid system, $\tau\dot{y} + y = Wx + My$, which has equilibrium solution $y = (I - M)^{-1}Wx$. This now provides us with a bit more flexibility. So the goal now is to find a combination of M and W that let's us implement a desired kernel but with the fewest linkages to neighbors (sparse M), inputs (sparse W) and minimum settling time. See if you can find such combinations for the kinds of kernels typically used in convnets, such as oriented filters (in the input layer). This strategy can also be generalized to higher layers. The solution is less easy to visualize, but one could still learn M and W. If implemented as a physical analog circuit rather than a digital simulation it could be dramatically more efficient.

Image compression in the retina

As we discussed in class, the spatial differencing operation performed in the retina is a form of signal compression, since it removes correlations in the image data due to structure in the world (the whitening theory of Atick & Redlich). JPEG performs a similar operation but instead by doing a PCA rotation and allocating bits according to variance in each principal component (within an 8x8 pixel image patch). Both can be justified as a decorrelation strategy and they are based on the same pairwise correlation function of natural images ($1/f^2$ power spectrum). Evaluate and compare these different compression schemes in terms of efficiency (compression ratio), and the conditions under which one of these schemes is preferred over the other.

On- and Off-coding in retina

The Karklin & Simoncelli model demonstrates the conditions under which On- and Off-cells emerge as an optimal coding strategy for retinal images. Their method relies upon a fairly complicated optimization that involves estimating mutual information. See if you can derive a simpler method for obtaining the same results utilizing an auto-encoder model, where the objective is to minimize reconstruction error under a constraint on firing rate (assuming noise in the channel). Going even further, how would you extend this to color, or time?

Foveated imaging

Simulate the foveated sampling lattice in the retina, using the formula for sample spacing as a function of eccentricity described in Van Essen & Anderson's chapter. How should the data acquired from such an imaging array be visualized? One possibility is foveated rendering, where the image is reconstructed according to the resolution it is sampled at within each local region. Ideally, such an image should appear identical to the original when you fixate the center. Alternatively, consider how the image appears when RGC axons project into cortex (via LGN), which results in a log-polar mapping.

Auditory coding

Simulate an auditory filter bank, following the specifications in Lyon's book, in response to a variety of natural sounds. Using the types of LIF models described in Eliasmith & Anderson chapter 4, how well can you reconstruct the auditory waveform from such a representation? (Related: Logan's theorem states that you can reconstruct a one-octave band-limited signal purely from its zero crossings.) Would this constitute an effective audio encoding or compression strategy for a neuromorphic chip? Alternatively, see if you can derive an auditory filter bank representation from the Karklin/Simoncelli approach of maximizing mutual information subject to a firing-rate constraint.

Neurobiological implementation of LCA

The Locally Competitive Algorithm (LCA) is a dynamical system considered to be a "neurobiologically plausible" model of how sparse coding could be implemented in a cortical circuit. But it still lacks many important aspects of an actual neural circuit: 1) Dale's law rule restricts any given neuron to be excitatory or inhibitory, thus the lateral inhibitory connections would need to be implemented by a separate set of inhibitory neurons, yet there are relatively few of these neurons in cortex compared to pyramidal cells (see Zhu & Rozell). 2) Neurons communicate via spike trains, not graded analog signals (see Shapero et al and Zylberberg et al.). 3) The input signal is dynamic, not static, thus one must consider how quickly such a circuit could track the actual dynamics of input from the retina. See if you can develop a more biophysically realistic sparse coding model taking into account these (or other) considerations.

Extreme sparse coding

As you increase the overcompleteness ratio of a sparse code, the learned dictionary shows a greater diversity of features and the representation becomes increasingly sparse (see this paper). If you keep going, it will become so sparse that an image patch could conceivably be represented by a single active unit, in which case the elements of the dictionary should resemble a collection of all the possible patches extracted from natural images. In the engineering literature this is called "vector quantization," and in neuroscience, "grandmother cells." What degree of overcompleteness would be required for an 8x8 pixel image to be faithfully encoded this way (with perceptually good SNR)?

Denoising via sparse coding.

Simoncelli & Adelson describe how one can exploit sparsity in natural image statistics in order to denoise images via a coring function on wavelet filter outputs. Implement this model either as described, or see if you can recast it in terms of a generative model such as the sparse coding model.

Dynamical sparse coding

The sparse coding model we discussed in class was only applied to static images. What about dynamic inputs such as time-varying images or sound? One possibility is a sparse convolution model, which has been applied to both images (paper) and sound (paper). What about using a recurrent neural network to model dynamics? For example consider a model of the form $\mathbf{I}(t) = \mathbf{\Phi}\,\mathbf{a}(t) + \mathbf{n}(t)$, where the dynamics on the $\mathbf{a}(t)$ are modeled as $\mathbf{a}(t+1) = \mathbf{M}\,\mathbf{a}(t) + \mathbf{u}(t)$. Can you learn a dynamics matrix $\mathbf{M}$ and infer the inputs $\mathbf{u}(t)$ for such a model? (essentially this amounts to a sparse coding extension of a Kalman filter model)

Topographic ICA/sparse coding

The standard sparse coding model assumes statistical independence among coefficients a_i. But when one examines the joint responses of certain sets of coefficients (corresponding to neighboring basis functions), this assumption is clearly violated. Topographic ICA was designed to take these into account. How would you extend this model to sparse coding? See this paper by Pierre Garrigues for one example. Another related approach is group sparse coding, where coefficients are grouped together in the sparse cost function to capture their co-activation (see this paper by Paiton et al.). Do these models yield improved denoising or performance improvements on other inference tasks?

Magic TV

Suppose you woke up one day to find someone rewired your optic nerve (or you have been implanted with a prosthetic retina). The signals from retina to brain are intact, but the wires are all mixed up projecting to the wrong places. Since neighboring pixels in natural images are correlated, it should be possible to learn a remapping that "descrambles" the image by exploiting these correlations. See if you can train a Kohonen-style self-organizing network to learn the proper topographic mapping of an image based on the statistics of natural images. (Kohonen dubbed a simpler version of this problem the 'Magic TV'.)

Model of horizontal connections

The neurons in layer 2/3 of cortex are richly interconnected via a network of recurrent, excitatory horizontal connections. One theory for these connections in area V1 is that they facilitate the encoding of contours or other extended structures in natural images. Building on the theoretical approach of attractor neural networks such as Hopfield nets, and combining with a feature representation such as sparse coding, can you construct a model capable of enhancing contours in natural images? (see this paper for some recent ideas along these lines; also this paper by Pierre Garrigues awhile back.)

Bump circuit stability

Kechen Zhang's bump attractor network (paper) shows how a network of neurons could maintain a stable representation of heading, and shift the bump in response to transient inputs from vestibular or other modalities. The model requires a specific pattern of connections between neurons. But it is implausible that biology could implement such connection strengths exactly. Experiment with this model to investigate its sensitivity to noise in either the weights or neural activities (e.g., consider a spiking version of this network). How could you make a more robust model?

MRF Boltzmann machines

Let's say you wanted to create a very large image texture whose statistics on a local scale were consistent with oriented lines but otherwise globally random. An efficient way to approach this is to embed a Restricted Boltzmann Machine (RBM) as part of a Markov random field (MRF), where each RBM only looks at a local region of the image, and they overlap and tile the entire image as a whole (see e.g. this classic paper by Geman & Geman and another by Roth & Black using an energy-based model that is basically an analog version of RBM). What happens when you use an RBM trained on small image patches containing oriented lines to produce a large image texture this way? What if you train the RBM MRF on other examples of line drawings, or images or text?

Langevin sparse coding

As we discussed in class, the sparse coding model can be formulated as a probabilistic model with a sparse prior over the latent variables of a linear generative model. In this case, the problem of inference is one of taking samples from the posterior distribution, rather than simply finding the explanation with highest probability (or the MAP (maximum-a-posteriori) estimate). In class we discussed sampling in Boltzmann machines via Gibbs sampling. But how to sample from a joint distribution over continuous variables? For this, one can use Langevin dynamics, which basically amounts to doing noisy gradient descent on the energy (negative log of posterior), as described in this recent paper. An important advantage of this framework is that it now allows you to learn other parameters of the prior such as horizontal connections (see above), but this has not yet been explored. Try implementing Langevin sparse coding and experimenting with new directions to take this.

Hierarchical RBM model

Hinton & Salakutinov described a hierarchical network composed by stacking Restricted Boltzmann machines (RBMs), also known as a 'deep belief network' (paper). However inference in their model is purely feedforward with no feedback from higher levels to lower levels. Another version of this model by Honglak Lee (paper) showed how top-down inference in such a model can fill in missing information in the input. See if you can implement and elaborate upon this idea to apply it to a denoising task. See also this recent paper from Hinton's group.

A model of the cerebellum

The cerebellum has a crystalline, matrix-like architecture that suggests that it could serve as a memory for high-dimensional patterns. This idea is commonly known as the Marr-Albus-Kanerva model, for three different theorists who have conceived highly related models. Kanerva's model is known Sparse Distributed Memory and is described here. Implement this model, and apply it to a sensorimotor (or other) task that might be solved in the cerebellum.

Dollar of Mexico

Pentti Kanerva's paper on hyperdimensional computing describes a simple example of analogical reasoning implemented with HD vectors: answering the question, "what is the dollar of Mexico?" Implement and elaborate on this idea - how would you apply it to a broader range of analogical reasoning problems?

Semantic word vectors via random indexing

Random indexing is a simple method for computing high-dimensional vectors for tokens (letters, words, images) so that similar tokens are assigned similar vectors (see poster). The standard methods for computing word embeddings in the neural nets literature such as word2vec or Glove are substantially more computationally intensive and require multiple passes of gradient descent for learning. Compare the types of vectors learned from random indexing to those learned from neural nets, for example in terms of their ability to express relationships between word meanings in terms of linear operations. How could you improve upon the performance of random indexing to make it competitive but still keeping the method simple?

Coupled oscillator Ising model

Tianshi Wang has shown how you can implement the Ising model (Boltzmann machine with visible units only) as a coupled-oscillator network, and that it can be highly effective for finding the ground state (energy minimum) of the network (paper). Implement this model and try applying it to solve an NP hard optimization problem.

Learning in sensorimotor loops

A goal of research in both robotics and neuroscience is to understand the principles of adaptive behavior in embodied systems. However, currently there are few theories for guiding work in this area. One approach advanced by O'regan and colleagues is based on learning "sensorimotor contingencies" (ref1, ref2). Another by Ralf Der and colleagues is based on minimizing both predictive and 'postdictive' error (ref). Both propose simple algorithmic examples that you can implement in computer simulation, or you may want to try implementing on a simple robotic platform.