

Context and Spatial Interactions between Neurons

There is considerable evidence that low-level vision involves long-range spatial interactions so that human perception of local regions of an image can be strongly influenced by their spatial context. Psychophysicists have discovered many perceptual phenomena demonstrating spatial interactions. For example, local image regions which differ from their neighbors tend to “pop-out” and attract attention while, conversely, similar image features which form spatially smooth structures tend to get “grouped” together to form a coherent percept, see figure (26)(left panel). Image properties such as color tend to spread out, or fill-in regions, until they hit a boundary [55][148] as shown in figure (26)(right panel).

Context and Spatial Interactions between Neurons

In general, there is a tendency for low-level vision to group together similar image features and make breaks at places where the features change significantly. These perceptual phenomena are not surprising from a theoretical perspective since they correspond to low-level visual tasks such as segmentation and the detection of salient features. Segmenting an image into different regions is one of the first stages of object recognition (in the ventral stream) and a pre-cursor to estimating the three-dimensional structure of objects, or surfaces, in order to grasp them or avoid them (dorsal stream). Detection of salient features has many uses including bottom-up attention [68]. It has been suggested that many of these processes are performed in V1 [190] although possibly this involves feedback and interactions between V1 and V2 [154].

Context figures



Figure 26: Left Panel: association fields. The circular alignment of gabor patches (left panel) make it easier to see the circular form in the presence of clutter (right panel). Right Panel: The neon color illusion. A bluish color appears to fill in the white regions between the blue lines creating the appearance of blue transparent disks.

d

Context Electrophysiology

These psychophysical and theoretical studies are supported by single-electrode studies [90], [96] which show that the activities of neurons on monkey area V1 appear to involve spatial interactions with other neurons. When monkeys are shown stimuli like figure 33 their responses over the first 60 msec are similar to those predicted by classic models (e.g., previous sections) but their later activity spreads in from the boundaries, roughly similar to predictions of computational models [183]. There is also a considerable literature the related topic of *non-classical receptive fields* [75].

Neural Network Models

This section discusses neural networks models which address these phenomena. Although the models capture the essence of the phenomena they are simplifications in three respects. Firstly, they use simple models of neurons and it is currently not possible to compare them directly to real neural circuits. Secondly, these models are formulated in terms of lateral, or horizontal connections. Thirdly, the performance of these models on natural images is significantly worse than human's. Although there are more advanced computer vision models, built on similar principles, whose performance starts to approach human vision (unless high level cues are present, which humans can exploit).

Probability Distributions on Graphs

We formulate these models in terms of probability distributions defined over graphs, where the nodes of the graph represent neurons. This differs from some of the standard “neural network” models for these types of phenomena, see [56]. but our approach has several advantages. Firstly, this enables us to use a coherent framework which unifies the models in this section with those we will present in later sections. Secondly, it puts the models in a form where they can be directly related to a class of computer vision models. Thirdly, this probabilistic formulation is of increasing use in models of Artificial Intelligence, Cognitive Science, and in the machine learning and statistical techniques used to analyse experimental neuroscience data. Fourthly, it is possible to derive many of these neural network models as approximations to the probability models.

Probabilistic Models of Neurons

We start by first introducing probabilistic models of neurons and showing how our previous linear filter models can be derived as approximations. Next we introduce neural network models and show their relationship to probability models. Then the following section uses this material to derive some specific models for a range of visual tasks. This section has three interactive demos: (4a) Gibbs sampling. (4b) Mean Field Theory and Neural Models. (4c) Hopfield Networks and Stereopsis.

Single Neurons: Probabilistic Model and Integrate and Fire (I)

In the previous section, we described neurons as linear filters and briefly mentioned thresholds and non-linearities. In this section we provide a more realistic model of a *stochastic neuron* where the neuron has a probability of firing an action potential. We will show how linear filters, thresholds, and non-linearities can be obtained as approximations to this stochastic model. This stochastic model is, in turn, an approximation and we refer to the literature for more realistic models such as assuming that the probability of firing is specified by a Poisson process [139]. For simplicity, we restrict ourselves to the simpler stochastic *integrate-and-fire* model which is easier to analyze and to relate to computational models.

Single Neurons: Probabilistic Model and Integrate and Fire (II)

In the integrate-and-fire model a neuron i receives input I_j at each dendrite j . These inputs are weighted by the synaptic strengths w_{ij} and sent along the dendrites to the soma. At the soma, these weighted inputs are summed linearly to yield summed linearly to yield $\sum_j w_{ij} I_j$. The probability of firing $s_i = 1$, or not firing $s_i = 0$, is given by:

$$P(s_i | \vec{I}) = \frac{\exp\{s_i(\sum_j w_{ij} I_j - T_i)\}}{1 + \exp\{\sum_j w_{ij} I_j - T_i\}}, \quad (22)$$

where T_i is a threshold.

Relations to the Stochastic Model (I)

To relate this stochastic model to our earlier linear models, we calculate the probability that the neuron fires. This is given by a sigmoid function:

In particular, the probability of firing ($s_i = 1$) is given by a sigmoid function:

$$\sum_{s_i=0}^1 s_i P(s_i | \vec{I}) = \frac{1}{1 + \exp\{\sum_j w_{ij} I_j - T_i\}} = \sigma(\sum_j w_{ij} I_j - T_i). \quad (23)$$

Observe that this is also the *expected firing rate* $\sum_{s_i=0,1} s_i P(s_i | \vec{I})$ because

$$\sum_{s_i=0,1} s_i P(s_i | \vec{I}) = P(s_i = 1 | \vec{I}) = \sigma(\sum_j w_{ij} I_j - T_i). \quad (24)$$

Relations to the Stochastic Model (II)

By computing the expected firing rate we obtain a deterministic approximation to a stochastic neuron. This is a sigmoid function of a linear weighted sum of the input (minus a threshold). The sigmoid function is approximately linear for small inputs, saturates at value 1 for large positive inputs, and suppresses large negative inputs to 0. Hence there is a linear regime where the probability of firing is $\sum_j w_{ij} I_j - T_i$. This enables us to recover the linear models used in the previous section as an approximation.

Next we modify the model so that it deals with non-linear image features. This allows us to relate it to the types of computational models described in the previous section and will enable us to construct richer models of this type that can deal with spatial context.

Enhance the Model to allow complex input

Consider detecting if there is an edge at pixel x . Formulate the problem as Bayes estimation where we have conditional distributions $P(f(I(x))|s)$ and priors $P(s)$ for $s \in \{0, 1\}$. The posterior distribution $P(s|f(I(x)))$ can be expressed in form:

$$P(s|f(I(x))) = \frac{1}{Z} \exp\left\{s \left(\log \frac{P(f(I(x))|s=1)}{P(f(I(x))|s=0)} + \log \frac{P(s=1)}{P(s=0)} \right)\right\},$$

where Z is a normalization constant (chosen so that $\sum_{s=0}^1 P(s|f(I(x))) = 1$). This shows that the posterior distribution for the presence of an edge can be expressed in the same form. The only difference is that the input is a nonlinear function of the image instead of the image itself. This claim can be justified by expressing $P(f(I(x))|s) = \{P(f(I(x))|s=1)\}^s \{P(f(I(x))|s=0)\}^{1-s}$, $P(s) = \{P(s=1)\}^s \{P(s=0)\}^{1-s}$, substituting these into the posterior $P(s|f(I(x))) = P(f(I(x))|s)P(s)/P(f(I(x)))$.

Probability Models with Context.

Now apply the model to foreground/background classification and modify it to include spatial context. Intuitively neighboring pixels in the image are likely to be either all background or all foreground. This is a form of prior knowledge which can be learnt by analyzing natural images.

We specify neurons by spatial position \vec{x} instead of index i . As above, we have distributions $P(f(I(\vec{x}))|s)$ for the features $f(I(\vec{x}))$ at position \vec{x} conditioned on whether this is part of the foreground object $s(\vec{x}) = 1$, or not $s(\vec{x}) = 0$. We use the notation \vec{S} to be the set of the states of all neurons $\{s(\vec{x})\}$. We also specify a prior distribution:

$$P(\vec{S}) = \frac{1}{Z} \exp\{-\gamma \sum_{\vec{x}} \sum_{\vec{y} \in N(\vec{x})} \{s(\vec{x}) - s(\vec{y})\}^2\},$$

where γ is a constant. This prior uses a neighborhood $N(\vec{x})$ which specifies those spatial positions which directly interact with \vec{x} in the model. In graphical terms, the positions \vec{x} are the nodes \mathcal{V} of a graph \mathcal{G} and the edges \mathcal{E} specify which nodes are connected. This is illustrated in figure (27)(far left panel).

Markov Structure (I)

Formally, the edges of the graph define the *Markov structure* of the probability distribution $P(\vec{S})$. It can be shown that the conditional distribution of the state $s(\vec{x})$ at one position depends *only* on the states of positions in its neighborhood $N(\vec{x})$. This is the *Markov condition*:

$$P(s(\vec{x})|\vec{S}/s(\vec{x})) = P(s(\vec{x})|\{s(\vec{y}) : \vec{y} \in N(\vec{x})\}),$$

where $\vec{S}/s(\vec{x})$ denotes all states in \vec{S} except $s(\vec{x})$. In real vision applications this type of prior, including the size of the neighborhoods, can be estimated from the statistics of natural images.

Markov Structure (II)

Next, we define a probability model for the observed image features at positions \vec{x} in the image. We use the same models as before, at each position \vec{x} :

$$P(f(I(\vec{x}))|s) = \{P(f(I(\vec{x}))|s = 1)\}^s \{P(f(I(\vec{x}))|s = 0)\}^{1-s}.$$

We combine these, using independence assumptions, to get a distribution:

$$P(f(\vec{I})|\vec{S}) = \prod_{\vec{x}} P(f(I(\vec{x}))|s) = \frac{1}{Z_I} \exp\left\{\sum_{\vec{x}} s(\vec{x}) \left(\log \frac{P(f(I(\vec{x}))|s = 1)}{P(f(I(\vec{x}))|s = 0)}\right)\right\},$$

where Z_I is a normalization term (which can be calculated directly).

Posterior Distribution (I)

These distributions $P(f(\vec{I})|\vec{S})$ and $P(\vec{S})$ can be combined to get the posterior distribution $P(\vec{S}|f(\vec{I}))$ which is of form:

$$P(\vec{S}|f(\vec{I})) = \frac{1}{Z_p} \exp\{-E(\vec{S})\},$$

where

$$E(\vec{S}) = - \sum_{\vec{x}} s(\vec{x}) \log \frac{P(f(I(\vec{x}))|s=1)}{P(f(I(\vec{x}))|s=0)} + \sum_{\vec{x}} \sum_{\vec{y} \in N(\vec{x})} \gamma \{s(\vec{x}) - s(\vec{y})\}^2.$$

The first term of $E(\vec{S})$ gives the local cues for foreground or background (the log-likelihood ratios of the features) while the second term adds the local context. This context encourages neighboring positions to be either all foreground or all background. Note that this method of specifying a distribution $P(\vec{S})$ in terms of a function $E(\vec{S})$ will keep re-occurring throughout this section.

Posterior Distribution (II)

This model specifies the posterior distribution for foreground-background classification using spatial context and, as we will show, similar methods can be applied to other visual tasks. But there remains the issue of how to estimate the most probable states, i.e. computing the Bayes estimator.

$$\hat{\vec{S}} = \arg \max P(\vec{S} | f(\vec{I})).$$

In the next two sections we will discuss neurally plausible algorithms which can do this. There are two types: (i) stochastic models which are natural extensions of the probabilistic neural models discussed earlier and, which the Statistics literature, are called *Gibbs samplers* [102], and (ii) neural network models which are based on simplified biophysics of neurons but which can also, in certain cases, be related to *mean field approximations* to the stochastic models.

Graphical Model figures

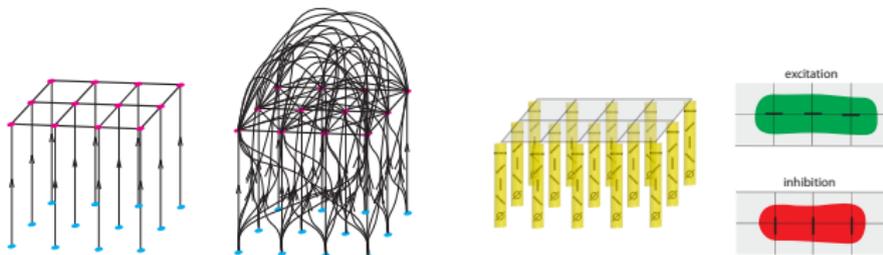


Figure 27: Far Left Panel: The graphical structure of the Markov model with nearest neighbor connections. Left Panel: a fully connected graphical model. Right Panel: A hyper-column structure where neurons within each column are tuned to different orientations and inhibit each other. Far Right Panel: Edges have excitation (green) along the direction of the edge and inhibition (red) perpendicular to the edge.

Probabilistic models of groups of neurons. (I)

In this section we introduce a more general probability distribution. It is also specified by a model defined over a graph where the nodes correspond to neurons and the edges to connections between them. But we will not make any Markov restrictions on the edges and so this model can be fully connected, see figure (27)(left panel).

More specifically, we have set of M neurons with states $\vec{S} = (s_1, \dots, s_M)$ and with input $\vec{I} = (I_1, \dots, I_N)$. We specify a *Gibbs probability distribution* over the set of activity of all neurons $\vec{S} = (s_1, \dots, s_n)$ as follows. First we define an energy function:

$$E(\vec{S}, \vec{I}; \vec{W}, \vec{\theta}) = - \sum_{ij} W_{ij} s_i I_j - (1/2) \sum_{kl} \theta_{kl} s_k s_l.$$

Probabilistic models of groups of neurons. (II)

This energy contains two types of terms: (i) those of form $s_i I_j$ which give the interactions between the states of the neurons \vec{S} and the input \vec{I} and (ii) those which specify interactions between the neurons. This energy is used to specify a *Gibbs distribution*:

$$P(\vec{S}, \vec{I}) = \frac{1}{Z} \exp\{-E(\vec{S}, \vec{I}; \vec{W}, \vec{\theta})\}. \quad (25)$$

Here Z is a normalization constant chosen to ensure that $\sum_{\vec{S}} P(\vec{S}|\vec{I}) = 1$. We note that Gibbs distribution originally arose in statistical physics where they specify the probability distribution of a physical system in thermal equilibrium. Here the physical energy of the system is E and the distribution can be derived using the maximum entropy principle.

Probabilistic models of groups of neurons. (III)

The weights $\{w_{ij}\}, \{\theta_{kl}\}$ specify the strength of the interactions between the neuron and the inputs, and between the neurons and each other. In particular, the *interaction term* $\sum_{kl} \theta_{kl} s_k s_l$ specifies the interactions between the neurons. If this term was not present, then the distribution simplifies and it can be expressed as a product of independent distributions:

$$P(\vec{S}|\vec{I}) = \frac{1}{Z} \exp\left\{\sum_{ij} w_{ij} s_i I_j\right\} = \prod_{i=1}^n P(s_i|\vec{I}). \quad (26)$$

Hence in this special case the neurons act independently and are driven purely by the input (i.e. there is no context). As a technical point, in this case the normalization factor can be computed directly as $Z = \prod_i Z_i$, where $Z_i = \sum_{s_i=0}^1 \exp\{\sum_j w_{ij} s_i I_j\}$.

Stochastic Dynamics (I)

Now we specify stochastic dynamics on this model. These dynamics have two purposes. Firstly, to describe the activities of sets of neurons interacting with each other. Secondly, to give algorithms for estimating properties such as the most probable configurations of the states \vec{S} , which can be used for visual tasks and for making decisions.

To specify stochastic dynamics, we generalize the stochastic neural model, see equation (22), to deal with a set of neurons. A neuron received input \vec{S} from other neurons in addition to direct input from the stimulus \vec{I} . Consider only the activity of this neuron, fixing the states of all the others. Then the neurons will have total input of $\sum_j w_{ij} I_j$ plus input $\sum_k \theta_{ik} s_k$ from the other neurons.

Stochastic Dynamics (II)

Then, extending equation (22), the probability that the cell i fires is:

$$P(s_i|\vec{I}, \vec{S}_{/i}) = \frac{1}{Z_i} \exp\{s_i(\sum_j w_{ij} I_j + \sum_{k \neq i} \theta_{ik} s_k)\}. \quad (27)$$

where the notation $\vec{S}_{/i}$ means the states $\{s_j : j \neq i\}$ of all the neurons except the neuron we are considering. The term Z_i is defined so that the distribution is normalized, so it is given by $Z_i = 1 + \exp\{\sum_j w_{ij} I_j + \sum_{k \neq i} \theta_{ik} s_k\}$. This gives the following dynamics for a group of neurons. At each time, a neuron is selected at random and fires with a probability specified by equation (27). This model assumes that no neurons ever fire at the same time and ignores the time for a spike fired from one neuron to reach other neurons. This is illustrated in interactive demo (4a).

Relations to Gibbs Distribution?

How does this stochastic dynamics relate to the Gibbs distribution specified above? From the statistical perspective, this is an example of *Markov Chain Monte Carlo* (MCMC) sampling [102]. MCMC refers to a class of algorithms which explore the state space of \vec{S} stochastically so that it will gradually move to configurations which have high probability $P(\vec{S}|\vec{I})$. More precisely, MCMC algorithms are guaranteed to give samples from the Gibbs distribution — $\vec{S}_1, \dots, \vec{S}_M \sim P(\vec{S}|\vec{I})$. The stochastic update rule in equation (27) is a special type of MCMC algorithm which is known as a *Gibbs sampler*, because it samples from the conditional distribution $P(s_i|\vec{I}, \vec{S}_{/i})$. These samples enable us to estimate the most probable state of the system $\vec{S} = \arg \max P(\vec{S}|\vec{I})$, hence they can estimate the MAP estimator of \vec{S} and make optimal decisions for visual tasks.

Learning and Boltzmann Machines

To apply these models to visual tasks, we need to specify the weights. One strategy is purely data driven and consists of learning the weights from training examples, this is the *Boltzmann Machine* [2] which is out of scope for this chapter. Another strategy is to specify distributions for specific visual tasks, and we will give examples in the next few sections.

Dynamical System Models of Neurons (I)

There is an alternative way to model sets of neurons using *dynamical systems* based on simplified models of their biophysics [139],[26]. Pioneering work on this topic was done by Wilson and Cowan [180], Grossberg and Mingolla [53, 56], Hopfield and Tank [63], Abbott and Kepler [1], and others. There is no space to cover the richness of these models and, in any case, our chapter concentrates on the probabilistic formulation. But we will discuss an important subclass of dynamical models [63] which, as we will show, have very close relations to the probabilistic approach.

Dynamical System Models of Neurons (II)

Following Hopfield and Tank, these dynamical systems are described as follows. A neuron is described by two (related) variables: (i) a continuous valued variable $u_i \in \{-\infty, \infty\}$, and (ii) a continuous variable $q_i \in \{0, 1\}$. Roughly speaking, u_i represents the input to the cell body (soma), due to the direct input and the input from other neurons, and q_i describe the probability that the cell will fire an action potential. These variables are related by the equations $u_i = \log(q_i/(1 - q_i))$ or, equivalently, by $q_i = \sigma(u_i)$ (where $\sigma(\cdot)$ is the sigmoid function).

The dynamics of the neuron is given by:

$$\frac{du_i}{dt} = -u_i + \sum_j w_{ij} l_j + \sum_k \theta_{ik} q_k. \quad (28)$$

Here, as before, $\sum_j w_{ij} l_j + \sum_k \theta_{ik} q_k$ represent the direct input and the input from the other neurons.

Dynamical System Models of Neurons (III)

It can be shown, next slides, that this dynamic system continually decreases a function $F(\vec{q})$, so that $(dF)/dt \leq 0$. The function F acts as a *Lyapunov function* for the system in the sense that it decreases monotonically as time t increases and is bounded below. The existence of a Lyapunov function for the dynamics guarantees that the system converges to a state which minimizes $F(\vec{q})$ (note that $F(\vec{q})$ will typically have many minimum, and the system may converge to any one of them). This dynamical system is illustrated in interactive demo (4b).

Relations between probabilistic models and dynamical system models (I)

Perhaps surprisingly, there is a very close relationship between the dynamic systems in equation (28) and the stochastic update in equation (22). More specifically, the dynamic system is a *mean field approximation* to the stochastic dynamics. *Mean field theory* (MFT) was developed by physicists as a way to approximate stochastic systems.

To explain this relationship we first define *the mean field free energy* $F(\vec{q})$:

$$F(\vec{q}) = - \sum_{ij} W_{ij} I_j q_i - (1/2) \sum_{ij} \theta_{ij} q_i q_j + \sum_i \{q_i \log q_i + (1 - q_i) \log(1 - q_i)\}. \quad (29)$$

Next we specify dynamics by performing steepest descent on the free energy (multiplies by a positive factor):

$$\frac{dq_i}{dt} = -q_i(1 - q_i) \frac{\partial F(\vec{q})}{\partial q_i}. \quad (30)$$

Relations between probabilistic models and dynamical system models (II)

Interestingly these are identical to the dynamical system in equation (28). This can be seen by introducing a new variable $u_i = \log q_i / (1 - q_i)$, which implies that $q_i = \sigma(u_i)$. Note that $\partial F / \partial q_i = -\sum_j W_{ij} I_j - \sum_j \theta_{ij} q_j + \log q_i / (1 - q_i)$, $u_i = \log q_i / (1 - q_i)$, and $dq_i / q_i (1 - q_i) = du_i$.

Equation (30) implies that the dynamical system decreases the free energy $F(\vec{q})$ monotonically with time t . This is because

$dF/dt = -\sum_i (\partial F / \partial q_i) (\partial q_i / \partial t) = -\sum_i q_i (1 - q_i) (\partial F / \partial q_i)^2$. Hence $F(\vec{q})$ is a Lyapunov function for equations (28,30) and so the dynamics converges to a fixed point.

Relations between probabilistic models and dynamical system models (III)

This shows that there is a close connection between the neural dynamical system and minimizing the mean field free energy. In turn, the mean field free energy is related to deterministic approximations to stochastic update methods like Gibbs sampling [4] [60]. This connection is technically advanced and is not needed to understand the rest of this chapter. Briefly, the mean field free energy $F(\vec{q})$ is the *Kullback-Leibler divergence* $F(Q) = \sum_{\vec{S}} Q(\vec{S}) \log \frac{Q(\vec{S})}{P(\vec{S}|\vec{I})}$ between the distribution $P(\vec{S}|\vec{I})$ and a factorized distribution $Q(\vec{S}) = \prod_i q_i^{S_i} (1 - q_i)^{1-S_i}$ (plus an additive constant). Hence the dynamical system seeks to find the factorized distribution $\hat{Q}(\vec{S})$ which best approximates $P(\vec{S}|\vec{I})$ by minimizing the Kullback-Leibler divergence. In this approximation the response q_i is an approximation to the expected response $\sum_{S_1} S_1 P(\vec{S}|\vec{I})$. The connections between mean field theory and neural models was described in [182]). For technical discussions about mean field theory and Gibbs sampling see [184].