

Biological and biologically-relevant Physics

Brian Ross

February 19, 2008

1 Fluids

1.1 Thermal fluctuations

The force applied to a particle in a thermal bath as a function of time \mathbf{F}_{th} is assumed to be white noise, meaning that each at some level of discretization in time (each component of) the force is completely random and independent of the force at adjacent times. The power spectrum follows from expanding $f(t) = \sum_i a_i \delta(t_i)$ in a Fourier series of $b_j \exp i\omega_j t$.

$$\begin{aligned}\langle |b_j|^2 \rangle &= \sum_i \langle |a_i \langle f(\omega_j) | f(t_i) \rangle|^2 \rangle \\ &\propto \frac{1}{N} \sum_i \langle |a_i|^2 \rangle \\ &= \langle |a|^2 \rangle\end{aligned}$$

Thus white noise is uniform in Fourier space, at least for frequencies below the inverse correlation time.

Another way of looking at this is that the distribution of white noise is smooth in real space. Therefore it is also uniform with the same distribution under any linear transformation whose Jacobian is 1. The Jacobian of a symmetric Fourier transform (using the prefactor $1/\sqrt{2\pi}$) must have modulus 1 since the product of four Fourier transforms is the identity (two are the identity plus space-inversion). Thus the amplitude is the same in Fourier space as in real space.

The equation $\mathbf{F}_{ext} + \mathbf{F}_{th} = m\mathbf{a}$ is called a Langevin equation. In an isotropic and uniform space $\mathbf{F}_{ext} = \mathbf{F}_{ext}(\mathbf{v})$ which equals zero at zero velocity. Expanding \mathbf{F} about $\mathbf{v} = 0$ gives $\mathbf{F}_{ext} \approx -\zeta\mathbf{v}$, where the proportionality is negative to prevent \mathbf{v} from blowing up as $t \rightarrow \infty$. ζ is the viscous friction coefficient and $B = 1/\zeta$ is called the mobility; the time constant for velocity decay is the product mB .

The damping and forcing terms in Langevin's equation are not independent; their balance is constrained so that each component of velocity corresponds on average to $(1/2)k_B T$ of energy. For given $\mathbf{F}(t)$ we can solve the equation of motion via a Laplace transform.

$$\begin{aligned}
\dot{\mathbf{v}} &= -\frac{1}{mB}\mathbf{v} + \mathbf{a}(t) \\
\longrightarrow \mathbf{v}(s) &= \frac{\mathbf{v}(0) + \int_0^\infty \mathbf{a}(t')e^{-st'} dt'}{s + 1/mB} \\
\longrightarrow \mathbf{v}(t) &= \mathbf{v}(0)e^{-t/mB} + \int_0^t \mathbf{a}(t')e^{-(t-t')/mB} dt'
\end{aligned}$$

In the last step above we had to be careful to preserve causality; the inverse-Laplace contour only closes in the left plane for the $t' < t$ part of the integral. After the transients (the term involving $\mathbf{v}(0)$) have died out we have

$$\langle v^2 \rangle \rightarrow \int_0^t \int_0^t \langle \mathbf{a}(t_1)\mathbf{a}(t_2) \rangle e^{-(2t-t_1-t_2)/mB} dt_1 dt_2.$$

Only the combination $t_1 + t_2$ weights the velocity, and the correlation between accelerations should only involve $t_1 - t_2$, so we make a change of variables to $u = (t_1 + t_2)/2$ and $v = t_1 - t_2$ (the Jacobian is 1). Since the fluctuation in $\mathbf{a}(t)$ is fast we can also extend the limits of integration on v to $\pm\infty$; although this takes into account correlations from times *later* than t as well as at $t < 0$, those spurious contributions are a small fraction of the total if the stochastic force's correlation time scale η and the damping time $\tau = 1/mB$ obey $\eta \ll \tau \ll t$.

$$\begin{aligned}
\langle v^2 \rangle &= \left(\int_0^t e^{-(2t-2u)/mB} \int_{\min\{2t, -(2t-2u)\}}^{\min\{2t, 2t-2u\}} \langle \mathbf{a}(u+v/2)\mathbf{a}(u-v/2) \rangle dv du \right) \\
&\approx \left(\int_0^t e^{-(2t-2u)/mB} du \right) \left(\int_{-\infty}^\infty \langle \mathbf{a}(u+v/2)\mathbf{a}(u-v/2) \rangle dv \right)
\end{aligned}$$

The second term is an integral over the correlation function $K(s) = \langle \mathbf{a}(u+0)\mathbf{a}(u+s) \rangle$, which is independent of u since there is no time dependence in the problem. The upper integration limit in the first term can be taken to ∞ since $t \gg \tau$. Finally, by substituting the known value for $\langle v^2 \rangle$ we can relate the mobility B to the correlation function.

$$\begin{aligned}
\langle v^2 \rangle &\approx \frac{1}{2mB} \int_{-\infty}^\infty K(s) ds \\
&= \frac{3k_B T}{m} \\
\longrightarrow \int_{-\infty}^\infty K(s) ds &= 6Bk_B T
\end{aligned}$$

This relates the phenomena of viscosity and thermal fluctuations. Very similar arguments can be applied to many other physical systems (i.e. shot noise in resistors); in its more general form our result is called the fluctuation-dissipation theorem.

1.2 Diffusion [9]

The presence of a stochastic force gives rise to diffusion. The diffusion equations in \mathbf{v} can be derived from the basic fact in probability $p(\mathbf{v}) = \sum_{\mathbf{v}'} p_0(\mathbf{v}') p(\mathbf{v}|\mathbf{v}')$. An initial distribution p_0 can be propagated *forward* in time by rewriting this as

$$p(\mathbf{v}, t + dt) = \int p(\mathbf{v} - \mathbf{w}, t) T(\mathbf{v} - \mathbf{w}, \mathbf{w}) d^3 \mathbf{w}$$

where $T()$ is a transition probability to a state offset by \mathbf{w} . To simplify we expand the right-hand side of this equation in \mathbf{w} . Since T should look like a narrow peak about $\mathbf{w} = 0$, a polynomial series in its second argument will not work well, so instead we expand about \mathbf{v} to a distance \mathbf{w} .

$$p(\mathbf{v}, t + dt) \approx \int d^3 \mathbf{w} \left[p(\mathbf{v}, t) T(\mathbf{v}, \mathbf{w}) - \mathbf{w} \cdot \nabla_{\mathbf{v}} p(\mathbf{v}, t) T(\mathbf{v}, \mathbf{w}) + \frac{\mathbf{w}^2}{2} : \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} p(\mathbf{v}, t) T(\mathbf{v}, \mathbf{w}) \right]$$

Using $\int T(\mathbf{v}, \mathbf{w}) d^3 \mathbf{w} = 1$, we get our final transport equation:

$$\begin{aligned} \frac{dp(\mathbf{v})}{dt} dt &\approx -\nabla_{\mathbf{v}} \cdot \left[p(\mathbf{v}, t) \int \mathbf{w} T(\mathbf{v}, \mathbf{w}) d^3 \mathbf{w} \right] + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} : \left[p(\mathbf{v}, t) \int \mathbf{w}^2 T(\mathbf{v}, \mathbf{w}) d^3 \mathbf{w} \right] \\ &= -\nabla_{\mathbf{v}} \cdot [\overline{\mathbf{w}}(\mathbf{v}) p(\mathbf{v}, t)] + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} : [\overline{\mathbf{w}^2}(\mathbf{v}) p(\mathbf{v}, t)] \end{aligned}$$

This is Kolmogorov's 'forward equation', better known as the Fokker-Planck equation.

If on the other hand p_0 represents a *final* target distribution, we can get another equation by the same method which will propagate the distribution *backwards*; this distribution represents the likelihood of hitting the target at eventual time t_0 .

$$\begin{aligned} p(\mathbf{v}, t - dt) &= \int p(\mathbf{v} + \mathbf{w}, t) T(\mathbf{v}, \mathbf{w}) d^3 \mathbf{w} \\ &\approx \int d^3 \mathbf{w} \left[p(\mathbf{v}, t) T(\mathbf{v}, \mathbf{w}) + T(\mathbf{v}, \mathbf{w}) \mathbf{w} \cdot \nabla_{\mathbf{v}} p(\mathbf{v}, t) + T(\mathbf{v}, \mathbf{w}) \frac{\mathbf{w}^2}{2} : \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} p(\mathbf{v}, t) \right] \\ \rightarrow -\frac{dp(\mathbf{v})}{dt} dt &\approx \overline{\mathbf{w}}(\mathbf{v}) \cdot \nabla_{\mathbf{v}} p(\mathbf{v}, t) + \frac{1}{2} \overline{\mathbf{w}^2}(\mathbf{v}) : \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} p(\mathbf{v}, t). \end{aligned}$$

This is the Kolmogorov 'backward equation'.

Similar diffusion equations can be obtained for other models as well. For example, over longer time scales we can treat the noise as a perturbation directly on some mean $\mathbf{v}(\mathbf{r})$, so the particle will obey

$$\frac{d\mathbf{r}}{dt} = \mathbf{v}(\mathbf{r}) + \mathbf{v}_{th}.$$

Then we get a Fokker-Planck equation in position:

$$\frac{dp(\mathbf{r})}{dt} = -\nabla_{\mathbf{r}} \cdot [\overline{\mathbf{v}}(\mathbf{r}) p(\mathbf{r}, t)] + \nabla_{\mathbf{r}} \nabla_{\mathbf{r}} : \left[\overleftrightarrow{D}(\mathbf{r}) p(\mathbf{r}, t) \right]$$

where D is the diffusion coefficient (usually scalar and a constant in space). A more simple derivation for constant scalar D and $\langle \mathbf{v} \rangle = 0$ is to just combine conservation of probability ($\dot{p} = -\nabla \cdot (p\mathbf{v})$) with Fick's Law ($\mathbf{v} \propto -\nabla p$) to get $\dot{p} = D \nabla^2 p$.

One can connect the diffusion coefficient D with the mobility B by comparing the predicted time-evolution of r^2 from Langevin and Fokker-Planck. In the former case we can obtain this quantity by

taking $\langle \mathbf{r} \cdot \mathbf{f} \rangle$ to each side \mathbf{f} of the Langevin equation. We can also substitute $(1/2)m\dot{r}^2 = (3/2)k_B T$ self-consistently, since we have already chosen γ so that this will be true in steady-state.

$$\begin{aligned} m \langle \mathbf{r} \cdot \ddot{\mathbf{r}} \rangle &= \frac{m}{2} \frac{d^2}{dt^2} \langle r^2 \rangle - m \langle \dot{r}^2 \rangle \\ &= -\frac{1}{B} \langle \mathbf{r} \cdot \dot{\mathbf{r}} \rangle + \langle \mathbf{r} \cdot \mathbf{F}_{th} \rangle \\ &= -\frac{1}{2B} \frac{d}{dt} \langle r^2 \rangle + 0 \\ \longrightarrow \frac{m}{2} \frac{d^2}{dt^2} \langle r^2 \rangle + \frac{1}{2B} \frac{d}{dt} \langle r^2 \rangle &= 3k_B T \end{aligned}$$

The solution to the diffusion-only equation $\dot{p}(\mathbf{r}, t) = D\nabla^2 p(\mathbf{r}, t)$ is

$$p(\mathbf{r}, t) = \left(\frac{1}{4\pi Dt} \right)^{3/2} \exp \left[-\frac{r^2}{4Dt} \right]$$

whose second moment is $\langle r^2(t) \rangle = 6Dt$. By substituting this $\langle r^2 \rangle$ into the modified Langevin equation, we have, for the time scales long enough for Fokker-Planck to be valid, $D = Bk_B T$.

1.2.1 Biological applications of diffusion

Aside from the obvious cases of chemical diffusion inside cells, the diffusion equation describes many biological processes that involve time evolution where the assumptions of the Kolmogorov equations are applicable. One prominent example is in studying the frequency of genetic alleles in a population. Two alleles compete in a constant population through the function $p(n, t)$ where n is the number of individuals with allele A . For diploid organisms there are actually three possibilities: AA , AB and BB , so two independent distributions are involved. Fixation occurs when one allele takes over a population. The likelihood of some gene being fixed from a given initial state is given by the forward Kolmogorov equation over some time T ; the fixation probability of a given allele over a range of initial conditions is given by backward Kolmogorov. The mean fixation time can be found by integrating $t \cdot \partial_t p(\text{fix})$ over all time. Other examples where diffusion equations may be applied include the motion of motor proteins, the growing and shrinking of microtubules, etc. The probability evolution equations are frequently solved by Fourier-transforming, solving for ω and expanding k to second order, in which case by comparison with the Fourier-transformed Fokker-Planck equation the velocity and diffusion coefficient are just the coefficients of $-ik$ and k^2 respectively.

1.3 Fluid flow [6]

The quantities that characterize the state of a fluid at a given point and time may be taken to be the density ρ , pressure p and velocity \mathbf{v} . Thus five differential relations are needed to solve for the complete fluid profile. These five are continuity, a force law, and a thermodynamic relation for ρ and p .

The continuity equation is conservation of mass. The differential form comes from relating the flow to \dot{m} within an infinitesimal volume.

$$\dot{\rho} = -\nabla \cdot (\rho \mathbf{v}).$$

The force law gives us $d\mathbf{v}/dt$ of a fluid particle, but we usually are more interested in $\partial\mathbf{v}/\partial t$ which is measured at a fixed point in space through which many particles enter and leave. The two are related by $d\mathbf{v}/dt = \partial\mathbf{v}/\partial t + (\partial\mathbf{v}/\partial\mathbf{x})(\partial\mathbf{x}/\partial t) = \partial\mathbf{v}/\partial t + \mathbf{v} \cdot \nabla\mathbf{v}$. The forces on an infinitesimal volume give

$$\rho(\partial\mathbf{v}/\partial t + \mathbf{v} \cdot \nabla\mathbf{v}) = \nabla p + \mathbf{f}_{visc} + \mathbf{f}_{ext}.$$

The external force density \mathbf{f}_{ext} is usually just gravity ρg which is usually negligible for microscopic flows. If the viscous force \mathbf{f}_{visc} is zero the force law is called Euler's equation. The most general form of a linear viscous force is [6]

$$\mathbf{f}_{visc} = \eta \nabla^2 \mathbf{v} + \left(\zeta + \frac{1}{3} \eta \right) \nabla \nabla \cdot \mathbf{v}$$

in which case the force law is called the Navier-Stokes equation. The two positive coefficients η and ζ are called coefficients of viscosity. In the case of incompressible flow $\nabla \cdot \mathbf{v} = 0$ takes the place of continuity and the second viscous term is zero; then we just call η the viscosity. In SI units the viscosity of water is $\sim 10^{-3}$.

The usual thermodynamic relation is that the entropy per unit mass s is constant. For flow that is both adiabatic and incompressible we have $de = Tds - pd(1/\rho) = 0$. If the flow is also steady in time we have a conservation law of energy per unit volume along a streamline, called Bernoulli's equation:

$$\frac{1}{2} \rho v^2 + p + u(\mathbf{r}) = \text{const.}$$

Usually $u(\mathbf{r}) = \rho gh$.

Fluids behave very differently depending on whether inertia or viscosity predominates. Loosely speaking, inertia is the momentum density ρv , while the viscous influence of some obstacle depends on η and some characteristic length scale l . The dimensionless combination of these is called the Reynolds number:

$$R = \frac{\rho v l}{\eta}$$

which is, roughly, the ratio of inertial to viscous forces. Two systems that differ only in size can have identical (rather, similar) flow profiles if the Reynolds number is the same; corresponding points in the larger system will be further from the boundary, which is why the characteristic scale contributes inversely to the viscous effect. Phenomena at the scale of a cell occur at small l and therefore low Reynolds number. One famous result is the Stokes drag on a sphere at low Reynolds number: $\zeta = 6\pi\eta$ (where ζ is the viscous friction coefficient).

1.4 Osmosis [8]

Suppose a fluid with two species (e.g. water and a solute) is bounded by a membrane, and that their relative concentrations are different on either side of the membrane. Then there will be an entropic force tending to equilibrate the concentrations. If the membrane is permeable to only one

of the species then there will be a one-way flow into the membrane, and a corresponding force on the membrane in the opposite direction. In the microscopic picture, the force on the membrane results from collisions: solvent molecules pass through with some probability (e.g. by hitting a channel), whereas solute molecules always transfer their momentum to the membrane. (An ideal water channel will only feel force from solute molecules.) Thus there is a higher pressure from the side with the higher concentration of solute. (Incidentally, I think the explanation given by Nelson in 7.2 may be wrong.)

We can take a more abstract perspective by computing the partition function of a solute/solvent system where the N solute molecules are constrained to lie in a volume V .

$$Z = Z_{solvent} Z_{\mathbf{p}_{solute}} \left(\frac{V}{V_0} \right)^N$$

We can extract the pressure from the Helmholtz energy:

$$\begin{aligned} P &= - \left. \frac{\partial}{\partial V} \right|_T - k_B T \ln Z \\ &= \frac{N k_B T}{V} \end{aligned}$$

This is the same as the ideal gas law, because the solvent factors entirely out of the problem, but in this context it is called the van 't Hoff relation.

Real membranes are of course never fully permeable; there is a proportionality constant P which is called the permeability of the membrane, defined by $j_i = P_i \Delta \rho_i$ for species i . The permeability has units of velocity, and can be thought of as the speed at which the interior concentration crosses the membrane if it were to do so uniformly. Cell membranes have permeabilities of $\sim 50 \mu\text{m/s}$ to water. In the more general case in which the pressure and concentration jump across the membrane one has to assume a relationship between both variables and the flows of both solute u and solvent v , which in the mild limit is linear:

$$\begin{bmatrix} j_u \\ j_v \end{bmatrix} = -\mathbf{P} \begin{bmatrix} \Delta p \\ \Delta c \end{bmatrix}.$$

P_{11} is called the filtration coefficient. Osmotic flow is contained in P_{12} .

1.5 Electrostatics in fluids [8]

In water all electrostatic phenomena are modified by the dielectric constant of water $\epsilon \approx 80\epsilon_0$. The Bjerrum length l_B is a characteristic length scale over which electrostatic forces of bare elementary charges ($e \sim 5 \times 10^{-10}$ in Gaussian units) are comparable to thermal forces.

$$\begin{aligned} \frac{e^2}{\epsilon l_B} &\sim k_B T \\ \longrightarrow l_B &= \frac{e^2}{\epsilon k_B T} \end{aligned}$$

In equilibrium, the distribution of charged species in an electrostatic potential is given by a Boltzmann distribution.

$$n(\mathbf{r}) = n_0 e^{-\beta q e V}$$

Amazingly, this equation is given its own name: the Nernst relation. (Across biological membranes V is usually a few tens of millivolts.) The Nernst equation predicts that bare charges in solution will accumulate a cloud of counterions, given by a consistent solution of the Poisson and Boltzmann equations, which will screen that charge from objects outside the cloud.

$$\begin{aligned} \nabla^2 V &= -\frac{1}{\epsilon} q e n(\mathbf{r}) \\ &= -\frac{q e n_0}{\epsilon} e^{\beta q e V} \end{aligned}$$

In electrophoresis, a charged object is pulled through a viscous medium by an electric field. The electric force qE balances the drag v/B , so its speed is $\mathbf{v} = Bq e \mathbf{E}$. For a collection of n charged objects, the current is $\mathbf{j} = \rho Bq e \mathbf{E}$.

2 Mechanics

2.1 Semiflexible polymers [3]

The simplest discrete (segmented) models of polymers differ in the distributions of relative segment orientations, parametrized by θ and ϕ where $\mathbf{u}_1 \cdot \mathbf{u}_2 = \cos \theta$. In the random flight model both θ and ϕ are completely random for each bond such that any orientation Ω is equally likely. In the freely-rotating chain model, each bond has the same given θ but ϕ is random. In a Gaussian chain all Ω are equally probable, but unlike the first two models the length of each segment is not fixed, but rather drawn from a Gaussian distribution. The continuous polymer models include a continuous Gaussian chain (essentially one segment of a discrete Gaussian chain), and the wormlike chain model of an inextensible bendable rod, with bending energy $\int \frac{\kappa}{2} \dot{\mathbf{u}}^2 ds$, embedded in a thermal environment.

Each of the above models assumes an isotropic cross-section, so the configuration of a polymer is entirely represented by $\mathbf{u}(s)$ where s represents distance along the polymer. Because the only intrinsic parameter of a segment is its tangent vector, the system is one-dimensional and correlations decay exponentially. The correlation function of such a (discrete) polymer $\langle \mathbf{u}_a \cdot \mathbf{u}_b \rangle$ is $\prod \langle \mathbf{u}_i | \mathbf{u}_{i+1} \rangle = \langle \cos \theta \rangle^{|b-a|}$. This exponential decay can be rewritten as $\langle \mathbf{u}(s) \cdot \mathbf{u}(s+L) \rangle = e^{-L/l_p}$ (which generalizes to continuous chains) where l_p , the persistence length, is the fundamental geometrical parameter of the polymer. The number of persistence lengths characterizes the polymer by the ratio $\langle R^2 \rangle / L$, which varies from L for a perfectly stiff polymer at zero persistence lengths to $2l_p$ for a flexible polymer at many persistence lengths.

To connect the mechanical stiffness of the wormlike chain with its geometrical persistence length, we compute $\langle \theta \rangle$ for a single link of a discretized chain, which for a short enough segment is a small angle.

$$\begin{aligned}
\langle \cos \theta \rangle &\approx \frac{\int d\theta (1 - \theta^2/2) \sin \theta \exp[-\Delta s \times \beta \kappa \theta^2 / 2\Delta s^2]}{\int d\theta \sin \theta \exp[-\Delta s \times \beta \kappa \theta^2 / 2\Delta s^2]} \\
&\approx 1 + \frac{1}{2} \frac{2\Delta s}{\beta} \frac{d}{d\kappa} \ln \left[\int_0^\infty d\theta \theta \exp[-\beta \kappa \theta^2 / 2\Delta s] \right] \\
&= 1 - \frac{\Delta s}{\beta \kappa}
\end{aligned}$$

Letting $L = N\Delta s$, we have $\langle \cos \theta \rangle^N = e^{-L/\beta \kappa}$, so $l_p = \beta \kappa$ for the wormlike chain.

In thermal equilibrium, if we neglect long-range interactions (i.e. excluded volume) then a polymer samples each bond angle independently, so we can ignore the surrounding segments and consider just the end-to-end statistics of the segment: consisting of a relative position \mathbf{R} and relative orientation Ω . The probabilities of various end-to-end positions and orientations can be found by summing the probabilities of all conformations with these restrictions. Alternatively, one can evolve a diffusion equation in L (analogous to time). Evidently the polymer distribution $G(\mathbf{R}, \Omega; L)$ convects in \mathbf{R} while diffusing in Ω , so the diffusion equation is of the form

$$\frac{\partial G}{\partial L} = -\mathbf{u} \cdot \nabla_{\mathbf{R}} G + D \nabla_{\mathbf{u}}^2 G.$$

Because $G(L < 0) = 0$ we have a discontinuity: a unit change in $\int G() d^3 \mathbf{R}$ occurs at $L = 0$ where the probability is concentrated at the origin. Thus a more general evolution equation for G that properly connects negative and positive L is

$$(\partial_L + \mathbf{u} \cdot \nabla_{\mathbf{R}} - D \nabla_{\mathbf{u}}^2) G = \delta(L) \delta(\mathbf{R}).$$

which explains why G is called a Green's function.

Ignore \mathbf{R} and consider only the \mathbf{u} -only diffusion equation. The probability density is

$$G(\mathbf{u}_f, L; \mathbf{u}_0) = \frac{1}{Z} \int_{\mathbf{u}_0}^{\mathbf{u}_f} D[\mathbf{u}] \exp \left[\int_0^L -\frac{l_p}{2} \dot{\mathbf{u}}^2 ds \right]$$

Feynman's path-integral prescription [2] can be used to relate the diffusion coefficient of the above differential equation to the persistence length in the partition function integral. Feynman defines a path integral to be the limit of many discrete integrals over equally-spaced segments of the path, where a normalizing factor $1/A$ is paired with each integral so that as the limit is taken the answer does not depend on the fineness of discretization. To obtain a differential equation in length we examine the change in the path as we add a single extra integral; since we want an equation in the Green's function we work entirely in terms of G at different values of \mathbf{u} and L , which we will approximate using derivatives from $G(\mathbf{u}, L)$.

$$\begin{aligned}
G(\mathbf{u}, L + dL) &= \frac{1}{A} \int d^3 \mathbf{u}_t \exp \left[-\frac{l_p}{2} \frac{(\mathbf{u} - \mathbf{u}_t)^2}{dL^2} dL \right] G(\mathbf{u}_t, L) \\
&= \frac{1}{A} \int d^3 \delta \mathbf{u} \exp \left[-\frac{l_p}{2} \frac{\delta \mathbf{u}^2}{dL} \right] G(\mathbf{u} - \delta \mathbf{u}, L)
\end{aligned}$$

The exponential fluctuates wildly beyond $\delta \mathbf{u}^2 \gg dL$ and cancels, so we can expand the terms outside the fluctuating exponential to order dL and $\delta \mathbf{u}^2$. (If we were to expand the exponential, however, the integrals would not remain convergent). Then we get rid of the antisymmetric terms and calculate the remaining integrals.

$$\begin{aligned} G(\mathbf{u}, L) + \frac{\partial G(\mathbf{u}, L)}{\partial L} dL &\approx \frac{1}{A} \int d^3 \delta \mathbf{u} \exp \left[-\frac{l_p}{2dL} \delta \mathbf{u}^2 \right] \left[G(\mathbf{u}, L) - \delta \mathbf{u} \cdot \nabla_{\mathbf{u}} G(\mathbf{u}, L) + \frac{1}{2} \delta u_i \delta u_j \nabla_{u_i} \nabla_{u_j} G(\mathbf{u}, L) \right] \\ &= \frac{1}{A} \int d^3 \delta \mathbf{u} \exp \left[-\frac{l_p}{2dL} \delta \mathbf{u}^2 \right] \left(G(\mathbf{u}, L) + \frac{1}{2} \delta u_i^2 \nabla_{u_i}^2 G(\mathbf{u}, L) \right) \\ &= \frac{1}{A} \left(\frac{2\pi dL}{l_p} \right)^{3/2} G(\mathbf{u}, L) - \frac{1}{2A} \left(\frac{2\pi dL}{l_p} \right)^{2/2} \partial_{l_p/2dL} \left(\frac{2\pi dL}{l_p} \right)^{1/2} \nabla_{\mathbf{u}}^2 G(\mathbf{u}, L) \end{aligned}$$

To be convergent for small ds the normalizing factor must be $A = (2\pi dL/l_p)^{3/2}$; then the finite terms cancel and we are left with

$$\frac{\partial G(\mathbf{u}, L)}{\partial L} = \frac{1}{2l_p} \nabla_{\mathbf{u}}^2 G(\mathbf{u}, L).$$

Thus $D = 1/2l_p$.

In the Gaussian limit the squared radius of gyration of a wormlike chain is

$$\begin{aligned} \langle R^2 \rangle &= \int \int \langle \mathbf{u}_1 \cdot \mathbf{u}_2 \rangle dr_1 dr_2 \\ &= L \int_{-\infty}^{\infty} e^{-|s|/l_p} ds \\ &= 2Ll_p = La_K. \end{aligned}$$

Here we introduced the ‘Kuhn length’ $a_K \equiv 2l_p$. Compare this to a long random flight chain with N segments each of length a , which diffuses in each direction by an amount $\langle \delta x_i^2 \rangle = Na^2/3$, so that $\langle R_{rf}^2 \rangle = Na^2 = La$. The analogous discrete quantity to a_K is therefore the segment length a ; this gives the justification for treating the ‘Kuhn length’ as a sort of mean segment length of a continuous chain. The Kuhn length is *twice* the persistence length because the polymer extends on average a distance l_p in both directions of a known tangent.

Flory gave a mean-field estimate of the partition function of a polymer with self-interactions and excluded volume, under the assumption that the polymer occupies a sphere of volume $V = (4/3)\pi R^3$. Each of the N links is given some entropic allowance g minus the volume fraction occupied by the other links; the other corrections are the Boltzmann factor for the self-interaction energy $-E(N, R)$, and the confinement of the polymer within R which imposes a penalty $p(r \leq R)$.

$$\begin{aligned} Z &\approx \left(\prod_{i=1}^N g(1 - (i-1)v_1/V) \right) p(r \leq R) e^{\beta E} \\ p(r \leq R) &\approx \int_0^R \left(\frac{3}{2\pi La_K} \right)^{3/2} \exp \left[-\frac{3r^2}{2La_K} \right] \times 4\pi r^2 dr \end{aligned}$$

(Kardar’s notes omit the integral over $p(r \leq R)$, and while my formula only accounts for the two ends of the polymer, his seems to be definitely wrong since it has the wrong limits (max at $R = 0$, zero at $R \rightarrow \infty$.) The interaction energy can be estimated by the potential of a uniformly-charged sphere of charge density N/V . The only detail is that the force is attractive, so all charges are opposite from all others; by playing the thought experiment of bringing charges into the volume one at a time, one can see that this just flips the sign of E . The point at which the entropic cost of volume exclusion balances the gain from the interaction is called the theta point.

2.1.1 Helical polymers

Most polymers do not have perfect cylindrical symmetry, and as a result they will in general tend to both twist and bend at every point (ignoring thermal fluctuations). Label the direction of bending perpendicular to the tangent vector $\hat{\mathbf{t}}$ as the normal vector $\hat{\mathbf{n}}$. The angular ‘velocities’ of these two rotations add to form some rotation vector which stays fixed as the polymer evolves in length. Both the tangent and normal vectors rotate around this rotation axis and the polymer thus forms a helix. The displacement of the polymer over many turns must be in the direction of the rotation vector, which gives the only fixed direction in the system. Coarse-graining over the helical repeats allows one to approximate the helix as a straight-line polymer without either bend or twist, with different elastic moduli from the real polymer, and whose bending moduli are the same in all directions. This helical averaging justifies using a wormlike chain to model, for example, DNA.

2.2 Base-paired nucleotides [5]

2.2.1 Double-stranded melting

DNA melting can be described by the Poland-Scheraga model, in which double-stranded DNA is interrupted in the interior by bubbles of single-stranded DNA. The double-stranded DNA is stiff; its free energy comes from a binding energy of $-E_b$ per nucleotide. The free energy of the single-stranded bubbles is entropic, since unpaired nucleotide strands are very flexible. Each successive unpaired nucleotide contributes some constant g to the free energy, but the entire bubble has the constraint that it must form a closed loop; imagining the two separated strands as a single looped polymer this would impose an entropic penalty $p(\mathbf{r} = 0)$. This penalty is assumed to be proportional to l^{-c} where l is the number of base pairs; if excluded volume interactions are neglected then $c = 3/2$. Putting all together, the partition function for a piece of DNA with N bubbles is

$$Z = \sum_{\{l_i\}} \left(\prod_{i=1}^N Z_{ss}(l_i^s) Z_{ds}(l_i^d) \right) Z_{ss}(l_{N+1}^s) \delta(\sum l_i - L)$$

$$Z_{ds}(l) = e^{\beta E_b l} \quad Z_{ss}(l) = K \frac{g^l}{l^c}.$$

The constraint that the l_i s must add up to L is hard to deal with directly. Instead we replace the delta-function with a factor of $e^{\beta \mu \sum l_i}$ where the (negative) μ is tuned so that $\langle \sum l_i \rangle = L$. This is the ‘grand canonical ensemble’.

$$\begin{aligned}\Gamma_N &= \sum_{\{l_i\}} \left(\prod_{i=1}^N Z_{ss}(l_i^s) Z_{ds}(l_i^d) \right) Z_{ss}(l_{N+1}^s) e^{\beta\mu \sum l_i} \\ &= (\Gamma_{ss}(\mu) \Gamma_{ds}(\mu))^N \Gamma_{ss}(\mu)\end{aligned}$$

$$\Gamma_{ds}(l) = \sum_{l=1}^{\infty} e^{\beta(E_b + \mu)l} = \frac{e^{\beta(E_b + \mu)}}{1 - e^{\beta(E_b + \mu)}} \quad \Gamma_{ss}(l) = K \sum_{l=1}^{\infty} \frac{(g + e^{\beta\mu})^l}{l^c} = f_c^+(g + e^{\beta\mu})$$

The final task is to sum over the possible numbers N of bubbles.

$$\begin{aligned}\Gamma &= \sum_{i=1}^{\infty} \Gamma_i \\ &= \frac{1}{1 - \Gamma_{ss}(\mu) \Gamma_{ds}(\mu)} \Gamma_{ss}(\mu)\end{aligned}$$

From this we see that the series stops converging at $\Gamma_{ss}(\mu) \Gamma_{ds}(\mu) = 1$, which implies that the number of bubbles diverges. Conveniently, the total length $\sum l_i$ and the number of bound nucleotides can be found by taking appropriate derivatives (with respect to $\beta\mu$ and βE_b respectively) of $\log \Gamma$.

2.2.2 Single-stranded self-hybridization

A partition function can also be derived for single-stranded nucleotides with arbitrary base-pairings provided that they are not pseudoknots: meaning that if nucleotides i and j are bound, then all nucleotides between i and j can only be bound to other nucleotides within this same stretch. The model assumes a constant binding energy $-E_b$ and does not incorporate the entropy of exploration of space by flexible nucleotide strands.

The last nucleotide in an $n + 1$ -length sequence can either be bound to one of the others or not, and the partition function can be broken into a sum of partition functions of the remaining n nucleotides for these various cases.

$$Z_{n+1} = Z_n + e^{\beta E_b} \sum_{i=1}^n Z_{i-1} Z_{n-i}$$

The stopping criterion is provided by $Z_0 = 1$. Again, by introducing a chemical potential via a Laplace-transform, we are able to perform the summation:

$$\begin{aligned}\sum_{n=0}^{\infty} e^{\beta\mu n} Z_{n+1} &= e^{-\beta\mu} \sum_{n=0}^{\infty} e^{\beta\mu n} Z_n = e^{-\beta\mu} Z(\mu) \\ &= Z(\mu) + e^{\beta E_b} e^{\beta\mu} \sum_{n=0}^{\infty} \sum_{i=1}^n \left(e^{\beta\mu(i-1)} Z_{i-1} \right) \left(e^{\beta\mu(n-i)} Z_{n-i} \right) \\ &= Z(\mu) + e^{\beta E_b} e^{\beta\mu} \left(\sum_{i=1}^n e^{\beta\mu i} Z_i \right) \left(\sum_{j=1}^n e^{\beta\mu j} Z_j \right) \\ &= Z(\mu) + e^{\beta E_b} e^{\beta\mu} Z(\mu)^2\end{aligned}$$

which can of course be explicitly solved for $Z(\mu)$.

2.3 Membranes [5]

The energy of a fluid membrane with no resistance to in-plane flow is determined by its geometry. The Hamiltonian must be an area integral of some energy density formed from the various scalar quantities that one can define at a given point on the membrane. One scalar is just the identity, which when integrated gives us the total area. The coefficient leading this quantity is the surface tension. The next-lowest-order scalars in ∂_{x_i} are the curvature (2nd order). We could choose these scalars to be the two principal radii of curvature R_1 and R_2 , but it is more convenient instead to work with the more symmetric mean curvature $H \equiv 1/R_1 + 1/R_2$ and Gaussian curvature $K \equiv 1/R_1 R_2$.

There are two reasons for working with H and K rather than R_1 and R_2 . 1) The Gauss-Bonnet theorem gives the integral over a closed membrane of the Gaussian curvature to be $2\pi\chi$ where χ is the ‘Euler characteristic’, an integer determined by the topology of the surface (i.e. bubble vs. doughnut). Continuous deformations of the membrane cannot change this quantity so it does not give rise to a ‘force’ $-\nabla_q E$ in any coordinate q . 2) A small imbalance in the areas of the upper and lower bilayers manifests itself as spontaneous *mean* curvature. For a surface curved only in x the area difference between the two bilayers separated by t is

$$\begin{aligned}\Delta A(R_x, 0) &= d_x d_y - d_y d_x \left(\frac{R_x - t}{R_x} \right) \\ &= d_x d_y \frac{t}{R_x}\end{aligned}$$

so

$$\begin{aligned}\Delta A(R_x, R_y) &= R_x \partial_{R_x} \Delta A + R_y \partial_{R_y} \Delta A \\ &= d_x d_y t \left(\frac{1}{R_x} + \frac{1}{R_y} \right) \\ &\propto d_x d_y H.\end{aligned}$$

Thus any infinitesimal excess in area directly biases the mean curvature.

Another way of justifying H and K is that they are the two invariants (trace and determinant respectively) of the curvature tensor to lowest order, which in terms of the ‘height’ h above the mean surface is $c_{ij} = \partial_{ij} h + O(3)$.

The final Hamiltonian, to second order, is:

$$E = \int dA \left[\gamma + \frac{1}{2} \kappa (H - H_0)^2 + \bar{\kappa} K \right]$$

where the first term is usually small for fluid membranes, $H_0 = 0$ for equal bilayers, and the last term can be neglected except when topology changes. γ is the surface tension, and κ and $\bar{\kappa}$ are the bending and Gaussian rigidities.

In the special limit $\gamma \rightarrow 0$ and $H_0 = 0$, and if we ignore the constant Gaussian curvature term, we have in the small-amplitude fluctuation limit

$$E \approx \frac{\kappa}{2} \int dA (\nabla^2 h)^2.$$

The energy of an eigenmode with amplitude B and wavevector \mathbf{q} is

$$E_q = \frac{1}{4} \kappa A B^2 q^4$$

where A is the area of the membrane.

The surface tension of a spherical membrane is related to the excess pressure inside the membrane by Laplace's formula. At equilibrium $dE = \gamma dA - PdV = 0$, so

$$\begin{aligned} 4\pi R^2 \delta R \cdot P &= 8\pi R \delta R \cdot \gamma \\ \longrightarrow \gamma &= PR/2. \end{aligned}$$

3 Chemical reaction networks

3.1 Rates and detailed balance

The Arrhenius equation is an empirical law that predicts the rate of a reaction from the activation energy E_a of the reaction.

$$r \propto e^{-\beta \Delta E_a}$$

The proportionality factor is a function of the concentrations of the reactants.

In equilibrium there can be no net flow between any two chemical species: the forward rate equals the backward rate for all reactions. This is true because a) any flow would necessitate a cycle in the graph of all chemical species whose edges are the allowed reactions, which would b) violate the second law of thermodynamics. This principle is called detailed balance.

The justification for (a) is that equilibrium implies that the concentrations of all chemical species are constant in time. Therefore if there is a net flow from A *in* to B , then there must be some flow *out* of B to C . This in turn implies some flow out of C , etc. and we keep applying this logic until we return to a species we have already seen. Thus a net flow along any reaction pathway implies that we can follow some positive net flow around a closed loop in the graph.

The second law of thermodynamics requires that any reaction move downhill in the Gibbs energy G . Thus along our cycle we have $G_1 > G_2 > \dots > G_1$ which is impossible.

3.2 Michaelis-Menten kinetics [11]

The situation described by the Michaelis-Menten equation is the conversion of a substrate (S) into a product (P) by enzyme E.



We make the following assumptions: 1) binding and unbinding is much faster than substrate conversion, so $k_1, k_{-1} \gg k_2$; 2) as a consequence of (1) there is an intermediate time scale during which binding/unbinding is in equilibrium and ensures that $d[ES]/dt = 0$, while 3) the total amount of unreacted substrate is still essentially its initial value S_0 . Substituting (3) into condition (2), and introducing the total enzyme concentration E_0 , gives:

$$\begin{aligned}\frac{d[\text{ES}]}{dt} &= k_1[\text{E}][\text{S}] - (k_{-1} + k_2)[\text{ES}] \approx 0 \\ \rightarrow [\text{ES}] &\approx \frac{k_1 E_0 S_0}{k_{-1} + k_2 + k_1 S_0}\end{aligned}$$

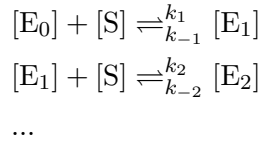
This allows us to find the rate, during quasi-steady state, at which substrate is converted into product.

$$\begin{aligned}\frac{d[\text{P}]}{dt} &= k_2[\text{ES}] \\ &= \frac{r_{max} S_0}{S_0 + K_m}\end{aligned}$$

where the maximum turnover rate that the enzyme could theoretically allow if completely bound to substrate is $r_{max} = k_2 E_0$, and the Michaelis-Menten constant, which accounts for the slowing of the reaction due to the finite bound fraction of enzyme, is $K_m = (k_{-1} + k_2)/k_1$.

3.3 Binding to multiple substrates [11]

A full description of an enzyme that has N binding sites for substrates involves 2^N species in the chemical reaction equations. If the binding sites are identical, and if we ignore the fact that the different realizations of an enzyme bound to s substrates may have different properties, then we can describe the enzyme using $N + 1$ species: $[\text{E}_0]$ for the completely unbound form, $[\text{E}_1]$ for the concentration that is bound to one substrate, etc.



The rate constants must reflect the fact that each $[\text{E}_i]$ is really a group of distinct species; if the ‘original’ rate constants are k' , then $k_1 = Nk'_1$ since there are N binding sites to choose from, $k_{-2} = 2k'_{-2}$, etc. Introduce the association constants $K_i = k_i/k_{-i}$. In equilibrium the different states of the enzyme are related by

$$\begin{aligned}[\text{E}_i] &= K_i[\text{E}_{i-1}][\text{S}] \\ &= \left(\prod_{j=1}^i K_j \right) [\text{S}]^i [\text{E}_0]\end{aligned}$$

where the second step is obtained recursively from the first. This leads to Adair’s equation for the mean occupied fraction Y (called the ‘saturation function’) of the enzyme.

$$\begin{aligned}Y &= \frac{1}{N} \frac{\sum_{i=0}^N i \times [\text{E}_i]}{\sum_{i=0}^N [\text{E}_i]} \\ &= \frac{1}{N} \frac{\sum_{i=0}^N i \times \left(\prod_{j=1}^i K_j \right) [\text{S}]^i}{\sum_{j=0}^N \left(\prod_{j=1}^i K_j \right) [\text{S}]^i}\end{aligned}$$

If the binding sites are truly independent then the association constants are related to one another in a straightforward way; however, many enzymes show *cooperativity* which changes the affinity of the enzyme for further binding of the substrate following an initial binding. Positive cooperativity means that the affinity for the substrate increases with increasing occupancy. There are two tests for mean cooperativity over K_i using the saturation function (which give different results). The first test compares Y with a \tilde{Y} that assumes zero cooperativity; $Y > \tilde{Y}$ indicates positive cooperativity. The second test defines positive cooperativity as occurring when there is some zero-crossing in the second derivative of $Y([S])$, implying that Y is sigmoidal.

If only the fully-bound and fully-unbound states exist in significant concentrations in equilibrium (as can happen in the case of high cooperativity), then the saturation function is approximately

$$Y \approx \frac{K_T[S]^N}{1 + K_T[S]^N}$$

where $K_T = \prod_{j=1}^N K_j$. In this case it is easy to experimentally estimate N from Y by varying $[S]$. The ‘Hill number’ n_H is defined as

$$n_H = \frac{\partial}{\partial \ln[S]} \ln \frac{Y}{1 - Y}.$$

In the limit of negligible intermediate states the Hill number converges to the number of substrate binding sites on the enzyme.

4 Neural networks

4.1 Modeling neurons

One tractable and widely-used model of neurons is the integrate-and-fire model [1], in which the state of a neuron consists of a voltage difference across its membrane and a conductance for each synapse on its dendrites. Inputs to the neuron increase the voltage (for excitatory inputs) or decrease the voltage (inhibitory inputs); if the voltage reaches some firing threshold V_{th} the neuron ‘fires’ an action potential and its voltage is reset to some lower value V_{reset} . Action potentials are treated as discrete events that raise the synaptic conductances of downstream neurons. In between action potentials the membrane voltages decays towards some resting potential V_0 and the synaptic conductances decay towards zero.

The integrate-and-fire neuron may be represented schematically as a capacitor and a resistor in parallel, connected to voltage sources (upstream neurons) by switches (synapses) whose conductance is a series of delta functions in time. The evolution equations for the voltage and conductances are

$$\begin{aligned} C \frac{dV}{dt} &= -\frac{V - V_0}{R} + \sum_s (V_s - V) g_s(t) \\ \frac{dg_s}{dt} &= -\frac{g_s}{\tau_s} + \frac{\bar{\alpha}_s}{\tau_s} \sum_{\beta} \delta(t - t_{\beta}) \end{aligned}$$

where s indexes the synapses and β indexes the spiking events at a given synapse. The integrated contribution to g_s from a single action potential is $\bar{\alpha}_s$. For excitatory synapses $V_s = V_E > V_{th}$ which

drives the voltage up; for inhibitory synapses $V_s = V_I < V_{th}$ which helps prevent the neuron from reaching threshold. Given sufficiently strong excitation a neuron will eventually reach threshold and fire an action potential, at which point its voltage will clamp immediately at $V_{reset} < V_{th}$ for a refractory period τ_r before resuming time evolution.

One makes a number of simplifying assumptions to arrive at the standard neural network equations [10]. All synapses from presynaptic neuron j fire with the same time series, and are each assumed to apply the same synaptic voltage V_{ij} and decay with the same constant τ_s . Introduce the normalized conductance $x_i = g_s/\alpha_s$ which has the same value for all synapses from neuron i , and which essentially counts the sum of the input current over presynaptic neurons rather than synapses:

$$\begin{aligned}\frac{dV_i}{dt} &= -\frac{V_i - V_{0i}}{\tau_n} + \frac{1}{C} \sum_j (V_{ij} - V_i) \alpha_{ij} x_j(t) \\ \frac{dx_i}{dt} &= -\frac{x_i}{\tau_s} + \frac{1}{\tau_s} \sum_\beta \delta(t - t_{i,\beta})\end{aligned}$$

where $\tau_n = RC$ and $\alpha_{ij} = \sum_{s_{ij}} \bar{\alpha}_{s_{ij}}$.

Secondly, we assume that the decay time of a synapse is significantly longer than the time between incoming action potentials, so that the synaptic conductances are nearly constant over the time it takes to generate an action potential. Then we can solve the first equation for $V_i(t)$:

$$V_i(t) = V_{ss} + (V_{reset} - V_{ss}) e^{-t/t_0}$$

where $t_0 = \left(1/\tau_n + \sum_j \alpha_{ij} x_j/C\right)^{-1}$ and $V_{ss} = t_0 \left(V_{0i}/\tau_n + \sum_j V_{ij} \alpha_{ij} x_j/C\right)$. This can be inverted to obtain a firing rate $\nu = 1/\Delta t = f(\sum_j \alpha_{ij} x_j, \sum_j V_{ij} \alpha_{ij} x_j)$ which we further approximate as $f(\sum_j V_{ij} \alpha_{ij} x_j)$ by neglecting the explicit dependence on t_0 . In the limit of $\tau_r \rightarrow 0$ and $\Delta t \ll t_0$ (hard spiking) the firing rate is approximately linear in the stimulus. Finally, we replace $\sum_\beta \delta(t - t_\beta)$ with its mean ν_i to obtain a single equation.

$$\tau \frac{dx_i}{dt} + x_i = f \left(\sum_j V_{ij} \alpha_{ij} x_j \right)$$

The dependence of f on the free parameters α_{ij} and V_0 depend (after treating the explicit t_0 as a constant parameter) in the following combination:

$$\begin{aligned}V_{th} - V_{ss} &= \frac{V_{0i}/RC + \sum_{j,s} V_{ij} \alpha_{ij} x_j/C}{1/RC + \sum_j \alpha_{ij} x_j/C} - V_{th} \\ &\propto \frac{1}{R} (V_{0i} - V_{th}) + \sum_j \alpha_{ij} x_j (V_{ij} - V_{th}) \\ &\equiv b_i + \sum_j W_{ij} x_j\end{aligned}$$

which gives the canonical neural network evolution equations

$$\tau \dot{x}_i + x_i = f \left(b_i + \sum_j W_{ij} x_j \right).$$

4.2 Backpropagation [4]

Consider a multi-layer feed-forward network: the input to neurons in a given layer only come from the outputs of the previous layer. Without loss of generality we write the equations as

$$\begin{aligned} \tau \dot{x}_i + x_i &= f(w_{ij} x_j) \\ \tau \dot{x}_j + x_j &= f(w_{jk} x_k) \\ &\dots \end{aligned}$$

where a constant bias has been replaced by a weight to some imaginary neuron whose output is always 1. Commonly-used activation functions f are the logistic function $1/(1+e^{-u})$ (whose output is on the interval $[0, 1]$) and $\tanh(u)$ ($[-1, 1]$). i indexes the last layer, j the next-to-last, etc. For simplicity the ‘Einstein summation convention’ of summing repeated indices of linear equations is being used. The complication is the term-wise nonlinearity $f(u_i)$: when we take derivatives we pick up an extra factor of $f'[u_i]$ that a linear equation would not have. Thus we put u_i in square brackets to emphasize that it does *not* count as a repeated index (though if we do sum we sum over it as well).

A feed-forward network can be solved without iterating, so we drop the \dot{x}_i terms. The output is given by:

$$x_i = f(w_{ij}[x_j = f(w_{jk}[x_k = f(\dots)])])$$

The task is to minimizing some error function $E(x)$ by adjusting the weights, which we will do by gradient descent. One could compute $(\partial E/\partial x_i)(\partial x_i/\partial w_{mn})$ from scratch for each weight (which involves multiple summations), but since $\partial E/\partial x_i$ depends partly on the effect of x_i on downstream neurons whose error sensitivities have already been computed, we can reuse that part of the calculation. This strategy is called backpropagation: using it we compute the error sensitivities to inputs of the the final layers first, then propagate these back to earlier layers.

Unfortunately, by expanding the partial derivatives we see that we cannot directly backpropagate the full error gradient dE/dw_{ij} , because this involves a term x_j whereas the gradient in higher layers use the derivative of this term. Instead we propagate backwards the error gradient in $u_i = w_{ij}x_j$ (so $x_i = f(u_i)$), which contains all terms except x_j . The sensitivity is thus defined as $s_i = dE/du_i$. We can calculate s_i in terms of the sensitivities in subsequent layers, plus the error associated with x_i which is a direct function of u_i .

$$\begin{aligned} s_j &= \frac{\partial E}{\partial u_j} + \frac{\partial E}{\partial u_i} \frac{\partial u_i}{\partial u_j} \\ &= \frac{\partial E}{\partial x_j} \frac{\partial x_j}{\partial u_j} + \frac{\partial E}{\partial u_i} \frac{\partial u_i}{\partial u_j} \\ &= \left(\frac{\partial E}{\partial x_j} + s_i w_{ij} \right) f'[u_j] \end{aligned}$$

Then the gradient in the weights is given by

$$\frac{dE}{dw_{ij}} = s_i x_j.$$

Backpropagation can also be applied to non-layered networks that have an error as a function of time. Align all nodes of the network along one vector x_j . The trick is to think of each successive time point of the network as a new layer. Then we apply the backpropagation rule as usual, but at the end sum over all the corresponding nodes of the same layer, since the same weight matrix w_{ij} applies at each layer and the total error gradient is the sum of the gradients at each time point. For static multi-layer networks the error function is usually only a function of the last layer of nodes x_i , but for backprop-in-time networks the error is often a function of time, in which case it is applied over many layers.

We can even use backpropagation to optimize the fixed points of recurrent networks. At steady state, again $x_i = f(w_{ij}x_j)$, but now the fixed-point x_i moves in some complicated way in response to a change in w_{pq} , so we have to solve for dx_i/dw_{pq} self-consistently.

$$\begin{aligned} \frac{dx_i}{dw_{pq}} &= f'[u_i] \left(x_q \delta_{ip} + w_{ij} \frac{dx_j}{dw_{pq}} \right) \\ \longrightarrow [\delta_{ij} - f'[u_i]w_{ij}] \frac{dx_j}{dw_{pq}} &= f'[u_i] \delta_{ip} x_q \end{aligned}$$

The gradient could be computed by solving for the unknown vector dx_i/dw_{pq} with a matrix inversion, and contracting it against the error vector dE/dx_i . Alternatively, we could avoid the inversion and instead use our matrix equation to iterate the unknowns to a fixed point, just as we did for the forward pass of the network.

$$\frac{dx_i}{dw_{pq}} \rightarrow f'[u_i]w_{ij} \frac{dx_j}{dw_{pq}} + f'[u_i] \delta_{ip} x_q$$

Unfortunately, this involves iterating over all p and q . It is actually possible to compute the contraction with just a single iteration, however. The trick is to recognize that, if we have the following two relations

$$\begin{aligned} R_{ij} a_j &= a'_i \\ b_i R_{ij} &= b'_j \end{aligned}$$

then by inverting R_{ij} we see that

$$a'_i b_i = a_j b'_j.$$

The heuristic explanation is that the linear transformation R_{ij} both rotates and scales; left and right-multiplied vectors are rotated backwards from each other but their rescaling is unaffected, so we can either rotate one vector forwards or the other backwards and the dot product works out the same. The correspondence to our circumstance is $a_j = dx_j/dw_{pq}$, $b'_j = dE/dx_j$. So instead we

can compute the sum by iteratively solving for b_i —which is a function of only *one* variable, not three—and then computing the contraction $S = a'_i b_i$.

$$b_i [\delta_{ij} - f'[u_i]w_{ij}] = \partial_{x_j} E$$

$$\text{solve} \longrightarrow \tau \dot{b}_j + b_j = f'[u_i]w_{ij}b_i + \partial_{x_j} E$$

$$S = b_i f'[u_i] \delta_{ip} x_q$$

$$= b_q f'[u_q] x_q$$

The parameter b is a sort of sensitivity to x : $b_i = \partial E / \partial x_i$ rather than $\partial E / \partial u_i$. This result is the *same* backpropagation algorithm we encountered in layered networks, which as we see generalizes to steady states of recurrent networks. The training involves the backwards weight matrix $f'[\mathbf{u}] \cdot \mathbf{w}^T$, a so-called ‘network transposition’ of the feed-forward weight matrix \mathbf{w} .

4.3 Hebbian networks [4]

The canonical biological learning rule is the Hebbian rule, which states roughly that w_{ij} is strengthened (increased) when x_i and x_j are correlated in time. This rule was originally proposed based on experimental evidence, but variants on it can actually be derived from very simple models of neural networks that are rooted in statistical physics.

First we consider very primitive Boolean neurons with the activation function $f(x) = \text{sgn}(x)$. We also make the crucial postulate that the weight matrix is symmetric, so $w_{ij} = w_{ji}$ and $\partial_{x_i} w_{ij} x_i x_j = 2w_{ij} x_j$ for $i \neq j$. This type of network is useful for pattern association; the steady states can be thought of as memories. For example, to explicitly encode the memory p_i we could initialize the network as $w_{ij} = p_i p_j$, so $x_i = p_i$ is then a stable state of the network. We can explicitly encode multiple memories by $w_{ij} = \sum_n p_i^{(n)} p_j^{(n)}$; the state $x_i = p_i$ may be a steady state if the spurious signals from the other patterns are small enough (N well-separated patterns give a contribution to each u_i on the order $1/\sqrt{N}$).

If we don’t know the memories to be stored a-priori, we can use an online training rule to store them in the Boolean network. A pattern to be memorized is imposed upon the network by forcing $x_i = p_i$, and the weights are adjusted to store the pattern permanently. By comparison with the explicit weight initialization, the weight update should then be proportional to $x_i x_j$, which is just the Hebbian rule.

Now if we think of the x_i as spins and the w_{ij} as the couplings between them (the biases w_{i0} correspond to some applied field), then we have an Ising model with all its usual properties. In particular, we can define an energy function $H = -(1/2)w_{ij}x_i x_j$; because the neuron update rule aligns each x_i with its mean input, the energy function never increases. Formally, because the weight matrix is symmetric $-\partial_i H = w_{ij} x_j$, which is the trajectory of the network under the standard update rule: $x_i \rightarrow w_{ij} x_j$. The energy therefore decreases throughout the trajectory and the network converges to a steady state.

Boolean neurons are equivalent to spins at zero temperature. The discontinuous activation function can be awkward to deal with, but we can smooth it by raising the temperature so that the spins fluctuate with a Boltzmann distribution. For a spin in the field $u = w_j x_j + b$ as before, the average fraction of time spent in the ‘up’ state, and its mean spin in the ‘up’ direction, are respectively:

$$t_{up} = \frac{e^{\beta u}}{e^{-\beta u} + e^{\beta u}} = \frac{1}{1 + e^{2\beta u}}$$

$$\langle S \rangle = \frac{e^{\beta u} - e^{-\beta u}}{e^{-\beta u} + e^{\beta u}}.$$

which are just the logistic and tanh functions. These two common activation functions thus correspond to the mean spin $f(u) = \langle u \rangle$, if the allowed spin states are $\{0, 1\}$ or $\{-1, 1\}$ respectively.

In a thermal environment we expect the *free* energy to be minimized [7]. If the allowed spins are $\{0, 1\}$, then the probability of an up-spin is x_i and the free energy of the network can be written

$$F = H - TS$$

$$= -\frac{1}{2}w_{ij}x_i x_j + T \sum_i (x_i \ln x_i + (1 - x_i) \ln(1 - x_i))$$

This corresponds to the logistic function, as we can see by solving $\nabla_i F = 0$ for $x_i = f(w_{ij}x_j)$. But by expressing F in a form that is independent of the type of activation function (i.e. TS is a function only of $f()$), we can obtain a quantity that always decreases for *any* arbitrary $f()$ as signals propagate through a neural network.

$$F = -\frac{1}{2}w_{ij}x_i x_j + \sum_i \int^{x_i} f^{-1}(s) ds$$

We can check explicitly that this more general expression always decreases as the network evolves.

$$\frac{dF}{dt} = \frac{\partial F}{\partial x_i} \frac{\partial x_i}{\partial t}$$

$$= (-w_{ij}x_j + f^{-1}(x_i)) \dot{x}_i$$

Remember that before steady-state is reached $x_i \neq f(u_i)$. The sign of $\partial F/\partial x_i$ is negative if the input is greater than that needed to drive x_i , in which case x_i should be increasing; likewise if the first term is positive then \dot{x}_i should be negative. Thus $dF/dt < 0$ until steady state is reached.

A technique called contrastive Hebbian learning can be applied to these networks which use arbitrary activation functions. The network is run to a steady state in both a clamped and a free mode; some subset of neurons are fixed at some desired value in the clamped mode that are allowed to relax to their $f(u_i)$ in the free mode. The strategy is to minimize not some direct error measure of the x_i between the two steady-states, but rather their free-energy gap $\Delta F = F^{(c)} - F^{(f)}$; this indirectly forces the network to learn the clamped state. The gradient of the free energy gap with respect to the weights includes both a direct term, as well as the indirect effect of changing the steady state x_k :

$$\frac{d\Delta F}{dw_{ij}} = \frac{\partial \Delta F}{\partial x_k} \frac{\partial x_k}{\partial w_{ij}} + \frac{\partial \Delta F}{\partial w_{ij}}$$

Only the x_k that stay unclamped should be counted in the sum. Crucially, steady state minimizes the free energy with respect to the free x_k , so $\partial F/\partial x_k = 0$ and therefore $dF/dw_{ij} = -\frac{1}{2}x_i x_j$. Thus the overall free energy difference is minimized in the direction of

$$-\frac{d\Delta F}{dw_{ij}} \propto x_i^{(c)} x_j^{(c)} - x_i^{(f)} x_j^{(f)}$$

where $x^{(c)}$ is measured at the clamped steady state, and $x^{(f)}$ is taken from the free steady state.

5 Miscellaneous

5.1 Extreme value distributions [5]

Often when a large number N of samples is taken from a distribution $p(x)$ we would like to know what the ‘worst case’, or largest, sample is likely to be. This is the extreme value problem, and the limiting distribution takes one of three forms depending on $p(x)$. Let P_1 be the integrated distribution $P_1(x) = \int_x^\infty p(x')dx'$, and let $P_N(x)$ be the cumulative extreme value distribution. Then for N samples

$$\begin{aligned} P_N(x) &= (1 - P_1(x))^N \\ &\approx e^{-NP_1(x)}. \end{aligned}$$

The extreme value distribution peaks strongly about the most likely value x^* , found by

$$\begin{aligned} P_N''(x)|_{x^*} &= 0 \\ \longrightarrow NP_1'^2(x^*) &= P_1''(x^*) \end{aligned}$$

The form of P_N depends on the behavior of $P_1(x)$ at large x . This is the Fréchet distribution of P_N results when $P_1(x) \rightarrow cx^{-p}$. Substituting the derivatives of P_1 gives

$$\begin{aligned} Nc^2p^2x^{*-2p-2} &= cp(p+1)x^{*-p-2} \\ \longrightarrow x^* &= \left(\frac{Ncp}{p+1}\right)^{1/p}. \end{aligned}$$

For a stationary distribution $c = k/N$. The extremal CDF and PDF can then be found by substituting directly into $P_N = e^{-NP_1}$.

$$\begin{aligned} P_N &= e^{-kx^{-p}} \\ p_N &= kpx^{-p-1}e^{-kx^{-p}} \end{aligned}$$

The Gumbel distribution results when $P_1 \rightarrow ce^{-(x/a)^p}$. Substituting into the $NP_1' = P_1''$ gives:

$$N \left(-\frac{cp}{a} \left(\frac{x^*}{a}\right)^{p-1} e^{-\left(\frac{x^*}{a}\right)^p} \right)^2 = c \left(\frac{p^2}{a^2} \left(\frac{x^*}{a}\right)^{2p-2} - \frac{p(p-1)}{a^2} \left(\frac{x^*}{a}\right)^{p-2} \right) e^{-\left(\frac{x^*}{a}\right)^p}.$$

Consistent with the Gumbel approximation (see below), we can drop the lowest-order term in x^*/a , which allows this equation to be solved for x^* .

$$Nce^{-\left(\frac{x^*}{a}\right)^p} = 1$$

$$\longrightarrow x^* = a(\ln Nc)^{1/p}$$

For large x the leading terms in the expansion of \bar{P}_1 come from repeated differentiations of the exponential. This is true for any $p > 0$, since differentiating the exponent increases the order of the leading polynomial in x , while differentiating the polynomial decreases the order by 1. Therefore in the limit the asymptotic form of P_N reduces to a power series:

$$\bar{P}_1 \approx c \left[1 + \left(-p \left(\frac{x^*}{a} \right)^{p-1} \right) (x - x^*) + \frac{1}{2} \left(-p \left(\frac{x^*}{a} \right)^{p-1} \right)^2 (x - x^*)^2 \right] e^{-(x^*/a)^p}$$

$$= ce^{-(x^*/a)^p} e^{-p(x^*/a)^{p-1}(x-x^*)}.$$

For normalization $ce^{-(x^*/a)^p}$ must approach $1/N$, so

$$P_N(x) \approx e^{-e^{-p(x^*/a)^{p-1}(x-x^*)}}.$$

which is a cumulative Gumbel distribution with $\lambda = p(x^*/a)^{p-1}$ and $x_0 = x^*$.

The Weibull distribution results when $P_1(x) \rightarrow c(a-x)^p$. Substituting the derivatives of P_1 into $N\bar{P}'_1(x^*)^2 = \bar{P}''_1(x^*)$ gives

$$Nc^2p^2(a-x^*)^{2p-2} = cp(p-1)(a-x^*)^{p-2}$$

$$\longrightarrow x^* = a - \left(\frac{p-1}{Ncp} \right)^{1/p}.$$

Again, the distribution is stationary so $c = k/N$. For $x < a$ the extremal CDF and PDF are:

$$P_N = e^{-k(a-x)^p}$$

$$p_N = kp(a-x)^{p-1} e^{-k(a-x)^p}.$$

5.2 Networks [5]

There are many different kinds of biological networks: chemical reaction networks, neural networks, protein interaction networks, gene regulatory networks, etc. It is often useful to treat networks in the abstract as graphs of connected nodes, and there are some generic network properties that can be interesting to all these cases even though the underlying systems are very different.

We first consider random graphs where $N \rightarrow \infty$ vertices (nodes) are connected at random by edges so that there are on average $\langle k \rangle$ edges per vertex. One fundamental parameter is the diameter of the network: the average shortest distance between two vertices along edges assuming that there is a path between them. We can estimate the diameter by noting that, until we have explored the graph, the number of nodes we reach in l steps is $\langle k \rangle^l$. Thus the diameter is $d = l_{max} \sim (\ln N)/(\ln \langle k \rangle)$.

At infinite N it is possible for a single connected cluster (but not two!) to develop that contains some finite proportion P of the nodes. The probability of two given nodes being connected is $\langle k \rangle / N$, so the probability of an arbitrary node *not* being connected to the cluster is $p = (1 - \langle k \rangle / N)^{NP} \rightarrow e^{-NP}$. Consistency requires that $p = 1 - P$; equating these two expressions allows us to solve for $P(\langle k \rangle)$. The main result is that large clusters only develop when $\langle k \rangle > 1$.

A second type of network, called a scale-free network, is formed by sequentially adding nodes connecting each with m edges to existing nodes. The existing N nodes are chosen randomly with a probability proportional to their connectivity; thus this process tends to enrich highly-connected nodes. We can write a master equation for this process in the long-time limit so that we can ignore initial conditions, noting that the probability of connecting to a k -linked node is $N_k / (2mN)$ (if there are k edges per node on average there are only $Nk/2$ total edges since each edge connects two nodes).

$$\begin{aligned} \langle N_k \rangle (t + 1) &= \langle N_k \rangle - \frac{m}{2mkN} \langle N_k \rangle + \frac{m}{2m(k-1)N} \langle N_{k-1} \rangle + \delta_{mk} \\ \rightarrow (N + 1)p_k &= Np_k - \frac{kp_k}{2} + \frac{(k-1)p_{k-1}}{2} + \delta_{mk} \\ (2 + k)p_k &= (k-1)p_{k-1} + 2\delta_{mk} \end{aligned}$$

To go to the second step we used $t \rightarrow N$ and assumed a steady state of p . At long times all the initial nodes have connectivity higher than m , so the distribution is zero until k , then $p_m = 2/(2 + m)$, and afterwards evolves according to

$$\begin{aligned} \ln p_k - \ln p_{k-1} &= \ln(k-1) - \ln(k+2) \\ \rightarrow \partial_k \ln p_k &\approx -\frac{1}{K} - \frac{2}{K} \\ p_k &\sim k^{-3} \end{aligned}$$

independently of m .

References

- [1] A. N. Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biol Cybern*, 95:1–19, 2006.
- [2] R. P. Feynman and A. R. Hibbs. *Quantum Mechanics and Path Integrals*. McGraw-Hill, Inc, 1965.
- [3] K. F. Freed. *Renormalization Group Theory of Macromolecules*. John Wiley and Sons, Inc, 1987.
- [4] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company, 1991.
- [5] M. Kardar, 2007. 8.592 Statistical Physics in Biology lecture notes.
- [6] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics*. Pergamon Press Ltd, 2 edition, 1987.

- [7] J. R. Movellan. Contrastive hebbian learning in interactive networks. *Neural Computation*, 1990 (submitted).
- [8] P. Nelson. *Biological Physics*. Philip C. Nelson, 2004.
- [9] R. K. Pathria. *Statistical Mechanics*. Butterworth-Heinemann, 2 edition, 1996.
- [10] S. Seung, 2002. 9.641 Intro. to Neural Networks lecture notes.
- [11] A. van Oudenaarden, 2006. 8.591 Systems Biology lecture notes.