Gatsby Computational Neuroscience Unit
Theoretical Neuroscience

Final examination, theoretical neuroscience
9 May 2022

Part I – short questions

There are four sections with four questions each. Please answer three out of each four, starting the answers for each section on a new page. Don't forget to write your name at the top of each block of answers.

Good luck!

1 Biophysics

1. The lipid bilayer that surrounds a neuron has almost infinite resistance – the only reason current flows is because there are ion-permeable channels. Consider a neuron with only passive channels, and assume its time constant is τ. If you doubled the size of the neuron (assume it's a sphere and double its radius), with the *number* of channels held fixed, what happens to the time constant?

   Solution The time constant of a cell is RC where R is the resistance (across the cell membrane) and C is the capacitance. Capacitance scales linearly with area (see the solution to problem 2). Because the number of channels is fixed, resistance doesn't change. If the radius doubles the area increase by a factor of 4, so the time constant will increase by a factor of 4.

2. To derive the cable equations, we used (see figure)

$$C\,\frac{\partial V(x,t)}{\partial t} = -dx\,\frac{\partial I(x,t)}{\partial x} + I_e(x) - I_m(x) \tag{1a}$$

$$I(x,t) = -\frac{dx}{R}\,\frac{\partial I(x,t)}{\partial x} \tag{1b}$$

   where C is the capacitance of a ring of length dx and R is the axial resistance through a slab of length dx (valid in the limit $dx \to 0$). Use the scaling of capacitance and resistance with geometry to show that the electrotonic length, λ, obeys

$$\lambda^2 \propto \frac{\text{area of dendrite}}{\text{circumference of dendrite}}\,. \tag{2}$$

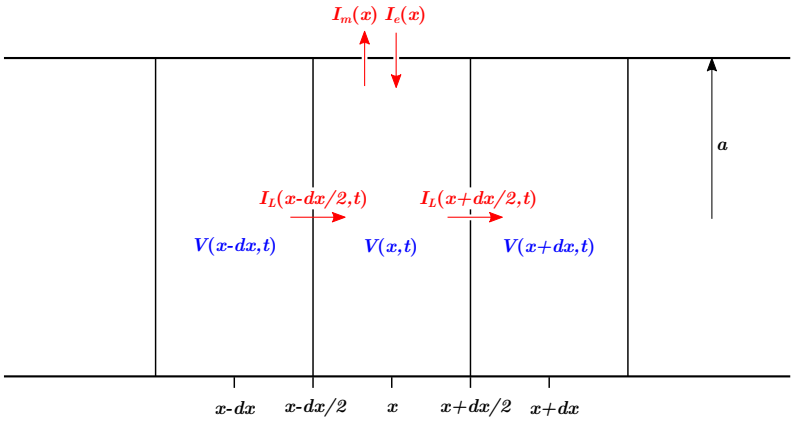

Figure 1: A small section of a dendrite. This was taken from a figure for a cylindrical dendrite, which explains the a on the right hand side next to the arrow, but ignore that, since for this problem the dendrite isn't cylindrical.

Solution Inserting the equation for current into the one for voltage gives us

$$\frac{\partial V(x,t)}{\partial t} = -\frac{dx^2}{RC}\,\frac{\partial^2 V(x,t)}{\partial x^2} + \frac{I_e(x) - I_m(x)}{C}\,. \tag{3}$$

Consequently,

$$\lambda^2 \propto \frac{1}{RC}\,. \tag{4}$$

So we just need to know how R and C scale with geometry.

We'll do capacitance first. By definition, $Q = CV$ where V is the voltage across a surface and Q is the charge that accumulated on either side of it. If we, for instance, doubled the area without changing the voltage, Q would double. Thus, capacitance scales with area. For our problem that means

$$C \propto \text{circumference} \times dx\,. \tag{5}$$

Now the resistance. It's related to voltage via $V = IR$. If we doubled the cross section of whatever carries the current while keeping voltage fixed, the current will double. Thus, resistance scales inversely with cross section; for our dendrite, that means

$$R \propto \frac{1}{\text{area}} \,. \tag{6}$$

Combining these with $\lambda \propto 1/R$ yields

$$\lambda^2 \propto \frac{\text{area}}{\text{circumference}} \,. \tag{7}$$

3. A linear integrate-and-fire neuron evolves according to

$$\tau \frac{dV}{dt} = -(V - \mathcal{E}_L) - g(V - \mathcal{E}_K) + V_0 \,. \tag{8}$$

When the neuron reaches $\mathcal{E}_L + V_{th}$ (with V_{th} positive), a spike is emitted and the voltage is reset to $\mathcal{E}_L$. Assume that $\Delta\mathcal{E} \equiv \mathcal{E}_L - \mathcal{E}_K > 0$. Show that the neuron spikes so long as

$$g < \frac{V_0 - V_{th}}{V_{th} + \Delta\mathcal{E}} \,. \tag{9}$$

Explain, qualitatively, why you get this result.

Solution As $t \to \infty$

$$V \to \frac{\mathcal{E}_L + g\mathcal{E}_K + V_0}{1 + g} \,. \tag{10}$$

For the neuron to spike, this must be greater than $\mathcal{E}_L + V_{th}$,

$$\frac{\mathcal{E}_L + g\mathcal{E}_K + V_0}{1 + g} > \mathcal{E}_L + V_{th} \,. \tag{11}$$

Straightforward algebra gives the desired result.

This has a relatively straightforward interpretation. The condition $\mathcal{E}_L > \mathcal{E}_K$ means the reversal potential for the term with g is below rest. Consequently, the larger g is, the more the membrane potential is pulled below rest, and the harder it is to spike. That explains the inverse dependence on ΔE. The other dependencies also makes sense: larger V_0 means the neuron is pulled more strongly to larger values, and larger V_{th} means a larger threshold, making it harder to spike.

4. When a presynaptic neuron fires, eventually there is a voltage change on one of its postsynaptic targets. List all the things that happen between a presynaptic spike and a postsynaptic voltage change.

Solution

- The action potential travels down the axon.
- When it arrives at the presynaptic terminal, the resulting increase in voltage opens voltage-gated calcium channels.
- The influx of calcium initiates a complicated second-messenger cascade that eventually causes vesicles to fuse, releasing neurotransmitter. (At least that's the plan; about half the time, nothing happens.)
- The neurotransmitter travels across the synaptic cleft and binds to the post-synaptic terminal, which opens channels.
- The open channels allow current to flow.
- The current causes a voltage change which, in the typical case, propagates along dendrites, eventually causing a change in membrane potential at the postsynaptic cell. (Sometimes presynaptic neurons synapses directly on the soma, so the dendritic propagation can be skipped.)

2 Networks

1. Consider a non-leaky integrate and fire neuron,

$$\frac{dV_i}{dt} = \sum_j J_{ij} g_j(t) + h_i \,. \tag{1}$$

When the neuron reaches threshold, θ, it emits a spike and is reset to zero. The spikes cause a brief change in $g_j(t)$,

$$g_j(t) = \sum_\mu g_0\left(t - t_j^\mu\right) \tag{2}$$

where t_j^μ is the time of the μ^{th} spike on neuron j and g_0 is a brief pulse that integrates to 1. For example, we might have

$$g_0(t) = \Theta(t)\frac{e^{-t/\tau}}{\tau} \tag{3}$$

where $\Theta(t)$ is the Heaviside step function: it's 1 if $t \geq 0$ and 0 otherwise.

Assume all neurons fire steadily. Show that in the long time limit, the firing rates, denoted ν_i, obey the equation

$$\nu_i = \sum_j \left(\theta\delta_{ij} - J_{ij}\right)^{-1} h_j \tag{4}$$

where δ_{ij} is the Kronecker delta (it's 1 if $i = j$ and 0 otherwise).

This breaks down if one of the eigenvalues of J_{ij} is greater than θ. Explain what goes wrong based on a one neuron example,

$$\frac{dV}{dt} = Jg(t) + h \tag{5}$$

where $J > \theta$ and $h > 0$. What is the behavior of this system of equations?

Solution

The reset part of the problem makes things difficult, but we can get rid of it by simply not resetting. Instead, the spiking rule is that whenever the membrane potential exceeds $n\theta$ a spike is emitted, but only once. (The "but only once" part means that if the membrane potential exceeds $n\theta$, and then later goes below it, when it exceeds $n\theta$ again it doesn't emit another spike.)

With this convention, we can now integrate both sides of Eq. (1) from 0 to T. In the large T limit the voltage on neuron i will have increased by $n_i\theta$ where n_i is the number of spikes emitted by neuron i, and the integral of $g_j(t)$ simply counts spikes. We thus have

$$n_i\theta = \sum_j J_{ij} n_j + h_i T \,. \tag{6}$$

Dividing both sides by T gives us Eq. (4).

To see what goes wrong in Eq. (5) when $J > \theta$, note that when the neuron spikes, the right hand side increases rapidly to J, which is greater than the threshold, θ. Thus, another spike will occur within the rise time of $g_j(t)$. This will lead to runaway excitation, and the firing rate will increase rapidly.

2. Consider a classical Hopfield network, whose update rule is

$$S_i(t+1) = \text{sign}\left[\frac{1}{N}\sum_{j=1}^{N}\left(\sum_{\nu=1}^{P}\xi_i^\nu\xi_j^\nu\right)S_j(t)\right] \tag{7}$$

where the ξ_i^ν are pulled *iid* from a random binary vectors,

$$\xi_i^\nu = \begin{cases} +1 & \text{probability } 1/2 \\ -1 & \text{probability } 1/2\,. \end{cases} \tag{8}$$

At time $t = 0$, the network is initialized according to

$$S_i(0) = \begin{cases} +\xi_i^\mu & \text{probability } 1-q \\ -\xi_i^\mu & \text{probability } q\,. \end{cases} \tag{9}$$

In other words, $S_i(0)$ is set to ξ_i^μ, but with an error rate of q.

What is the probability that $S_i(t) = \xi_i^\mu$? Your answer will depend on N and P as well as q.

Solution

Separating the terms with $\nu = \mu$ and $\nu \neq \mu$, Eq. (7) can be written

$$S_i(1) = \text{sign}\left[\xi_i^\mu \frac{1}{N}\sum_j \xi_j^\mu S_j(0) + \sum_{\nu\neq\mu}\xi_i^\nu\xi_j^\nu\frac{1}{N}\sum_j S_i(0)\right]\,. \tag{10}$$

The first sum on j is

$$\frac{1}{N}\sum_j \xi_j^\mu S_j(0) = (1-q)\times(+1) + q\times(-1) = 1-2q\,. \tag{11}$$

It's variance is given by

$$\text{Var}\left[\frac{1}{N}\sum_j \xi_j^\mu S_j(0)\right] = \frac{1}{N^2}\sum_j\left\langle\left(\xi_j^\mu S_j(0) - (1-2q)\right)^2\right\rangle = \frac{1}{N}\left(1-(1-2q)^2\right) = \frac{4q(1-q)}{N}\,. \tag{12}$$

For the sum over ν and j, we note that the product $\xi_i^\nu\xi_j^\nu S_0(0)$ is uncorrelated across i, j and ν, and zero mean. Thus, the sum is Gaussian (with respect to index, i) with variance $N(P-1)$. Taking into account the division by N, we thus have

$$\sum_{\nu\neq\mu}\xi_i^\nu\xi_j^\nu\frac{1}{N}\sum_j S_i(0) \sim \left(\frac{P-1}{N}\right)^{1/2}\eta_i \tag{13}$$

where η_i is a zero mean, unit variance Gaussian random variable.

Putting all this together, and using the central limit theorem (which tells us that large sums are Gaussian), we arrive at

$$S_i(1) = \text{sign}\left[(1-2q)\xi_i^\mu + \sigma\zeta_i\right] \tag{14}$$

where ζ_i is a zero mean, unit variance Gaussian random variable and

$$\sigma^2 \equiv \frac{P-1+4q(1-q)}{N}\,. \tag{15}$$

Thus, the probability that $S_i(1) = \xi_i^\mu$ is the probability that $\sigma\zeta_i > -(1-2q)$. More formally,

$$P\big(S_i(1) = \xi_i^\mu\big) = H\left(\frac{1-2q}{\sigma}\right) \tag{16}$$

where H is the cumulative normal function.

3. Assume that x (which we should think of as a firing rate of a population of neurons relative to baseline) obeys the equation

$$\frac{dx}{dt} = \beta\tanh(x) - x\,. \tag{17}$$

Assume $\beta > 1$, so this has an unstable fixed point at $x = 0$ and two stable ones at the nonzero solutions to the equation $\beta\tanh(x) = x$. Show that you can stabilize the unstable fixed point by adding a dynamical variable,

$$\frac{dx}{dt} = \beta\tanh(x-y) - x \tag{18a}$$

$$\frac{dy}{dt} = \alpha x - \lambda y\,. \tag{18b}$$

What are the conditions on α on λ that will ensure that the fixed point at $x = y = 0$ is stable? Under those conditions, do the fixed points of the original equation still exist?

Solution The equations for x and y clearly have a fixed point at $x = y = 0$. Linearizing around that fixed point gives us

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \beta - 1 & -\beta \\ \alpha & -\lambda \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}, \tag{19a}$$

valid so long as both x and y are near zero. Both eigenvalues are negative if and only if the trace is negative and the determinant is positive, which yield the inequalities

$$\lambda + 1 > \beta \tag{20a}$$

$$\alpha\beta > \lambda(\beta - 1)\,. \tag{20b}$$

The most illuminating way to determine what happened to the fixed points of the original equation is to draw nullclines. But the easiest way is to take an algebraic approach. Setting the right hand sides of Eqs. (18a) and (18b) to zero yields

$$x = \beta \tanh\left(\left(1 - \frac{\alpha}{\lambda}\right)x\right)\,. \tag{21}$$

Equation (20b) tells us that $\alpha/\lambda > (\beta - 1)/\beta$, which tells us that

$$1 - \frac{\alpha}{\lambda} < 1 - \frac{\beta - 1}{\beta} = \frac{1}{\beta}\,. \tag{22}$$

Thus, the slope of the right hand side of Eq. (21) must be less than 1, so it has only one solution. This means the two stable fixed points that existed in the original equation disappear, leaving us with only one stable one.

4. A recent paper (arXiv:2109.03879, 2021) proposed the following update rule,

$$\frac{dx_i}{dt} = \sigma\left(\beta\sum_k W_{ik}x_k\right)\left(\sum_j J_{ij}\phi(x_j) - x_i\right) \tag{23}$$

where σ is the standard sigmoidal function, $\sigma(x) = e^x/(1 + e^x)$. In the large β limit, these equations have an interesting property: all units for which the argument of the sigmoid is negative are "frozen" – they do not change with time (or at least they change exponentially slowly). It turns out that, for a large range of parameters, the other units go to a fixed point. What's interesting about this is that there is a whole manifold along which the frozen units stay frozen. Thus, the network admits a high-dimensional attractor with no fine tuning – something that is, in general, very hard to do.

Why does this model have nothing to do with the brain?

Solution Under this scheme, the time derivative vanishes for the frozen units. But for real neurons the dynamical variable is the membrane potential, and if its time derivative vanished the neuron would stop spiking.

3 Coding

1. Consider spikes that arrive at a synapse according to a homogeneous Poisson process with rate λ. Suppose the post-synaptic current in response to a single transmitted spike follows a timecourse given by a fixed function $h(\tau)$, where τ is the time since the spike.

   (a) Assuming the synapse is perfectly reliable (i.e. each pre-synaptic spike causes a fixed amount of transmitter release), what is the variance of the integrated PSC at some time t?

   (b) Reconcile your result with the properties of the Poisson counting distribution by considering

   $$h(\tau) = \begin{cases} 1 & 0 \le \tau < T \\ 0 & \text{otherwise} \end{cases}$$

   (c) Suppose instead that transmission fails (i.e. no transmitter is released) on average for half the incoming spikes, independently of their relative timing. How would this affect the variance?

   Solution The PSC at t is given by $C(t) = \int d\tau\, h(\tau)s(t-\tau)$. Using the Poisson properties

   $$\langle s(t)dt \rangle = \lambda dt$$
   $$\langle s(t_1)s(t_2)dt_1 dt_2 \rangle = (\lambda^2 + \delta(t_1 - t_2)\lambda)dt_1 dt_2$$

   we have

   $$\langle C(t) \rangle = \left\langle \int d\tau\, h(\tau)s(t-\tau) \right\rangle = \int d\tau\, h(\tau)\lambda = H\lambda$$
   $$\langle C(t)^2 \rangle = \left\langle \int d\tau_1\, h(\tau_1)s(t-\tau_1) \int d\tau_2\, h(\tau_2)s(t-\tau_2) \right\rangle = \int d\tau_1\, d\tau_2\, h(\tau_1)h(\tau_2)(\lambda^2 + \delta(\tau_1 - \tau_2)\lambda)$$
   $$= H^2\lambda^2 + \int d\tau\, h(\tau)^2\lambda$$

   so

   $$\mathbb{V}[C(t)] = \lambda \int d\tau\, h(\tau)^2$$

   The boxcar h is just counting spikes in an interval of length T. We have $\int d\tau\, h(\tau)^2 = T$ so the result above gives a variance of λT as expected.

   With failures as described the rate just halves. So $\mathbb{V}[C(t)]$ is halved too.

2. You read a paper describing the following experimental results: Similar-sized population recordings with overlapping receptive fields were made in both LGN and V1 while an animal viewed a natural movie. The experimenters applied a set of Gabor-like filters to the movie images at the common RF location to obtain an estimate of instantaneous orientation energy in the movie. Using a decoding approach, they computed the mutual information rate between this time-series and the LGN and V1 populations, finding that the measured information in V1 is higher. *Prima facie*, this would seem to contradict the data-processing inequality. Suggest at least two possible explanations. You may consider:

   - known properties of the two areas
   - limitations of the decoder
   - the natural movie context

   or take a different approach.

   Solution Lots of options here.

3. Recall that STA provides an unbiased estimate of a single dimension of stimulus selectivity if the experimental stimulus distribution is radially symmetric, but otherwise arbitrary. For STC to return an unbiased subspace estimate (in general) the distribution must be Gaussian.

(a) Explain why.

(b) A colleague suggests taking a (non-Gaussian) radially symmetric distribution, rescaling the stimulus lengths to Gaussianise the distribution (i.e., so that the lengths will follow a Rayleigh distribution), and performing STC with the transformed vectors. Will this procedure identify the correct subspace (in the sense of unbiasedness)?

Solution

(a) We require the variance along any axis orthogonal to the coding subspace to be same in the background and spike-triggered ensembles regardless of the form of the neural non-linearity. This requires $K_\perp \cdot \mathbf{s}$ to be uncorrelated with $f(K \cdot \mathbf{s})$ for any f and coding space K with nullspace $K_\perp$. That in turn requires $\mathbf{s}$ projected onto any pair of orthogonal axes to be independent. The only distribution that is radially symmetric and has this property is Gaussian.

(b) Yes. It might appear paradoxical that stretching vectors can create independence, but it is equivalent to importance sampling.

4. Recall that the Fisher information carried by firing modelled as Gaussian with stimulus-dependent mean $\mu(s)$ and covariance $\Sigma(s)$ is given by

$$F(s) = \frac{1}{2}\mathsf{Tr}\left[\Sigma'\Sigma^{-1}\Sigma'\Sigma^{-1}\right] + \mu'^{\mathsf{T}}\Sigma^{-1}\mu'$$

where the primes denote derivatives with respect to s.

Consider the 'Poisson-like' case where $\Sigma(s) = \mathsf{diag}[\mu(s)]$.

(a) Show that the second term in the Gaussian expression (the "linear" Fisher information) equals the Poisson Fisher information in this case.

Consider the simple case of two neurons, with $\mu_1(s) = \rho\cos(s)$ and $\mu_2(s) = \rho\sin(s)$.

(b) By considering the differences in the optimal decoders for the Poisson and Poisson-like Gaussian case, suggest how the "extra" information in the first "trace" term might arise.

4 Learning

1. Oja's rule applies to a linear neuron receiving N dimensional inputs with output $v = w^T u$ where v is scalar, $w \in R^N$ is a vector of weights, and $u \in R^N$ is the input vector. Oja's rule always converges to a weight vector of a particular norm. Suppose instead we want a learning rule that either learns or not depending on the magnitude of the input correlations. Consider the rule

$$\tau \frac{d}{dt} w = vu - \alpha v^2 w - \beta w$$

where α and β are non-negative parameters.

Suppose we train this rule on a dataset with input correlations $\langle uu^T \rangle = C$, where the correlation matrix C has maximal eigenvalue λ. What does the norm of the weights converge to as a function of α, β and λ?

Solution On average, we have the update

$$\tau \langle \frac{d}{dt} w \rangle = \langle uu^T \rangle w - \alpha \langle v^2 \rangle w - \beta w \tag{1}$$

$$= Cw - (\alpha v^2 + \beta) w \tag{2}$$

from which we can see that w will either align to the principle eigenvector e_1 of C or decay to zero. We can therefore write w in steady state as ae_1 for an unknown scale a.

The average time derivative of the square of the norm is

$$\tau \langle \frac{d}{dt} ||w||_2^2 \rangle = \langle 2w^T (vu - \alpha v^2 w - \beta w) \rangle \tag{3}$$

$$= 2 \left[\langle v^2 \rangle - (\alpha \langle v^2 \rangle + \beta) ||w||_2^2 \right]. \tag{4}$$

Using the fact that the weight vector eventually aligns to the principle eigenvector, $w = ae_1$, we have

$$\langle v^2 \rangle = w^T \langle uu^T \rangle w \tag{5}$$

$$= w^T C w \tag{6}$$

$$= a^2 e_1^T C e_1 \tag{7}$$

$$= a^2 \lambda \tag{8}$$

where λ is the maximum eigenvalue of C. Returning to Eqn. (4), in steady state we have

$$0 = 2 \left[a^2 \lambda - (\alpha a^2 \lambda + \beta) a^2 \right] \tag{9}$$

$$a = \sqrt{\frac{\lambda - \beta}{\alpha \lambda}}. \tag{10}$$

For $\beta > \lambda$, the time derivative is less than zero and the weight vector converges to zero. Hence the steady state norm is

$$\lim_{t \to \infty} ||w||_2 = \begin{cases} \sqrt{\frac{\lambda - \beta}{\alpha \lambda}} & \text{for } \beta < \lambda \\ 0 & \text{for } \beta \geq \lambda \end{cases}. \tag{11}$$

2. Consider a neural network with an input layer of dimension N, a hidden layer of dimension M and a scalar output. Each hidden unit of the neural network computes its activity as

$$h_i = \frac{1}{\sqrt{M}} g(w_i \cdot x) \quad \text{for } i = 1, \cdots, M, \tag{12}$$

where $w_i \in R^N$ is its weight vector, $x \in R^N$ is the input vector, the activation function $g(x)$ is a step function

$$g(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \tag{13}$$

and the scaling by $1/\sqrt{M}$ is a normalization to simplify analysis of $M \to \infty$.

If we assume that the weights from the input to hidden layer are selected randomly from a unit normal Gaussian distribution, then argue that in the limit of $M \to \infty$, the inner product of the hidden (feature) layer for inputs $x \in R^N$ and $z \in R^N$ has the form:

$$K(x,z) = \frac{1}{(2\pi)^{N/2}} \int du e^{-\frac{1}{2}\|u\|_2^2} g(u \cdot x) g(u \cdot z) \tag{14}$$

where the integration variable u is a vector in R^N.

Compute the integral above, thereby determining an equivalent kernel function to an infinite width neural network with random Gaussian weights.

Solution

For M units, the inner product between hidden representations is

$$\frac{1}{M} \sum_{i=1}^{M} g(w_i \cdot x) g(w_i \cdot z). \tag{15}$$

In the limit as $M \to \infty$, due to independence of the w_i, this sample average converges to the exact average $\langle g(u \cdot x) g(u \cdot z) \rangle$ where u is drawn from the same distribution as any w_i, that is, a multivariate unit normal distribution. The expression for this average is the desired result

$$\frac{1}{(2\pi)^{N/2}} \int du e^{-\frac{1}{2}\|u\|_2^2} g(u \cdot x) g(u \cdot z).$$

To compute this integral, note that the term $g(u \cdot x) g(u \cdot z)$ is one only if $u \cdot x > 0$ and $u \cdot z > 0$ and zero otherwise. These conditions will be met if u lies within ninety degrees of both x and z. Consider projecting u onto the plane spanned by x and z. Because the scale of this projection does not matter, we can consider only its angle in this plane. The desired average is thus the probability that this angle lies within ninety degrees of both x and z. By drawing this situation geometrically we see that the projection must lie in a sector of angle $\pi - \theta$ where θ is the angle between x and z. By radial symmetry of the Gaussian distribution, the angle of the projection of u is uniform on $[0, 2\pi]$. Hence the probability that the projection lies in this sector is $\frac{\pi - \theta}{2\pi}$. Finally, using the fact that $\theta = \cos^{-1}\left(\frac{x \cdot z}{||x||||z||}\right)$, we have

$$K(x,z) = \frac{1}{2} - \frac{1}{2\pi} \cos^{-1}\left(\frac{x \cdot z}{||x||||z||}\right).$$

3. A deep linear network has weight matrices $W_2 \in R^{N_o \times N_h}$ and $W_1 \in R^{N_h \times N_i}$, where N_o, N_h, and N_i are the dimensions of the output, hidden, and input layers respectively. The network computes its output for a given input as $y = W_2 W_1 x$.

   The neural tangent kernel $K(x, x')$ between inputs x and x' is given by

$$K(x,x') = \sum_{i=1}^{N_o} \frac{\partial y_i(x)}{\partial \theta} \cdot \frac{\partial y_i(x')}{\partial \theta}$$

   where $\theta \in R^{NoN_h + N_h N_i}$ is all parameters (i.e. W_1 and W_2) flattened into a vector, and $y_i(x)$ denotes the i^{th} component of the output vector in response to input x.

   Suppose we have a dataset of P examples, $x^\mu \in R^{N_i}, i = 1, \cdots, P$. For convenience we can concatenate all examples into the matrix $X \in R^{N_i \times P}$. We can compute the kernel matrix of examples $\Theta \in R^{P \times P}$ where $\Theta_{ij} = K(x^i, x^j)$. Find an expression for the neural tangent kernel matrix Θ in terms of W_1, W_2 and X.

   Solution

   Let $w_i^{2^T}$ denote the i^{th} row of W_2. Then

$$\frac{\partial y_i(x)}{\partial W_1} = \frac{\partial}{\partial W_1} w_i^{2^T} W_1 x \tag{16}$$

$$= w_i^2 x^T \tag{17}$$

$$\frac{\partial y_i(x)}{\partial w_i^{2^T}} = \frac{\partial}{\partial w_i^{2^T}} w_i^{2^T} W_1 x \tag{18}$$

$$= x^T W_1^T \tag{19}$$

and the partial derivative with respect to other rows of W_2 is zero. Eqn. (19) yields a vector and we can straightfowardly compute its dot product for two inputs. Eqn. (17) however is a matrix. We can compute the dot product between its flattened form using the trace. In particular,

$$\begin{aligned}
\frac{\partial y_i(x)}{\partial \theta} \cdot \frac{\partial y_i(x')}{\partial \theta} &= \text{Tr}\left(w_i^2 x^T x' w_i^{2^T}\right) + x^T W_1^T W_1 x' && (20)\\
&= \text{Tr}\left(x^T x' w_i^{2^T} w_i^2\right) + x^T W_1^T W_1 x' && (21)\\
&= ||w_i^{2^T}||_2^2 x^T x' + x^T W_1^T W_1 x'. && (22)
\end{aligned}$$

Now summing over outputs we have

$$\sum_{i=1}^{N_o} \frac{\partial y_i(x)}{\partial \theta} \cdot \frac{\partial y_i(x')}{\partial \theta} = ||W_2||_F^2 x^T x' + N_o x^T W_1^T W_1 x'. \quad (23)$$

Finally, we can express the full NTK matrix as

$$\Theta = ||W_2||_F^2 X^T X + N_o X^T W_1^T W_1 X.$$

4. The perceptron computes its output as $\hat{y} = \text{sgn}(w \cdot x)$ where $w \in R^N$ is a weight vector and $x \in R^N$ is an N-dimensional input vector. The perceptron learning algorithm tries to find a weight vector such that $y^\mu = \hat{y}^\mu$ for every example in a dataset $\{x^\mu, y^\mu\}, \mu = 1, \cdots, P$ where P is the size of the dataset. Cover's theorem tells us that, for random inputs and outputs that are ± 1 with probability $1/2$ and large N, the capacity of the perceptron is $P = 2N$. Suppose instead we have a dataset where each input element

$$x_i^\mu = \begin{cases} 1 & \text{with probability } p \\ -1 & \text{w.p. } 1-p \end{cases} \quad \text{for } i = 1, \cdots, N, \mu = 1, \cdots, P,$$

and each output

$$y^\mu = \begin{cases} 1 & \text{w.p. } p \\ -1 & \text{w.p. } 1-p \end{cases}$$

for $\mu = 1, \cdots, P$, where $0 < p < 1/2$ is a small probability (that is, the patterns are sparse). For the following questions you can simply state your answers without showing work if you like.

(a) Would you expect a perceptron trained with the perceptron learning algorithm to obtain a capacity larger or smaller than $2N$?

(b) Would you expect the algorithm to find a zero-error solution if one exists?

(c) Suppose we now amend the perceptron learning algorithm to require that patterns be correct by a margin $\delta > 0$, that is, we require that each pattern's margin $\delta^\mu = y^\mu w \cdot x^\mu / ||w||_2 \geq \delta$. Would you expect the capacity to increase or decrease?

(d) Suppose after training an adversary can manipulate each input vector x^μ. They choose a perturbation ξ^μ to apply to each input to obtain $\tilde{x}^\mu = x^\mu + \xi^\mu$ where $||\xi^\mu||_2 < \epsilon$ for all μ where ϵ is a parameter controlling the extent to which they can manipulate an example, and try to flip the perceptron's output to be incorrect. If we know ϵ before we train the perceptron, how should we pick the margin δ to fend off this adversary?

Solution

(a) The capacity increases due to the correlations in the dataset.

(b) The perceptron learning algorithm is guaranteed to converge to a solution if one exists, regardless of the structure of the dataset.

(c) Requiring a margin will decrease the capacity because the constraints are more stringent.

(d) On the adversarial example we have the margin

$$\begin{aligned}
\tilde{\delta}^\mu &= y^\mu w \cdot \tilde{x}^\mu / ||w||_2 && (24)\\
&= y^\mu w \cdot (x^\mu + \xi^\mu) / ||w||_2 && (25)\\
&= y^\mu w \cdot x^\mu / ||w||_2 + y^\mu w \cdot \xi^\mu / ||w||_2 && (26)\\
&> \delta^\mu - \epsilon && (27)\\
&> \delta - \epsilon && (28)
\end{aligned}$$

so we should choose $\delta > \epsilon$ to ensure a positive margin on all examples.