COS 514: Fundamentals of Deep Learning

Fall 2025

Instructor: Prof. Sanjeev Arora
TA: Gon Buzaglo

Assignment 2

Instructions:

- Submission deadline is October 6.
- We recommend start reading all questions as soon as possible, as some are harder than others.
- You may collaborate in groups of up to **3** students.
- If you collaborate on a problem, you must clearly state the names of your collaborators at the beginning of the solution to that problem.
- All group members must declare that they contributed equally to the solutions.
- You must write up your own solutions independently in LaTeX. **Handwritten or scanned solutions will not be accepted.**
- Cite any resources (papers, textbooks, websites) that you use.
- Submit your assignment as a single PDF on gradescope.

Problems

Problem 1: PAC-Bayes with Gaussian Distributions Implies Generalization of Min-Norm Solution

Consider linear classifiers in $\mathbb{R}^k$ parameterized by $w \in \mathbb{R}^k$. Let the prior be $P = \mathcal{N}(0, I_k)$. After training on a sample S of size n you obtain $\hat{w}$, and take the posterior to be $Q = \mathcal{N}(\hat{w}, I_k)$. Assume the loss $\ell(h_w, (x, y)) \in [0, 1]$ for all (x, y) and let $\hat{L}_S(Q)$ and $L_{\mathcal{D}}(Q)$ denote, respectively, the empirical and population risks of the Gibbs classifier that draws $w \sim Q$ and predicts with h_w.

1. (KL divergence calculation, step by step.) We want to compute the KL divergence between

$$Q = \mathcal{N}(\hat{w}, I_k) \quad \text{and} \quad P = \mathcal{N}(0, I_k).$$

Recall the pdf of a k-dimensional Gaussian:

$$p(w) = \frac{1}{(2\pi)^{k/2}(\det \Sigma)^{1/2}} \exp\Big(-\tfrac{1}{2}(w-\mu)^\top \Sigma^{-1}(w-\mu)\Big).$$

(a) **Log-density.** Show that

$$\ln p(w) = -\tfrac{k}{2}\ln(2\pi) - \tfrac{1}{2}\ln\det\Sigma - \tfrac{1}{2}(w-\mu)^\top\Sigma^{-1}(w-\mu).$$

(b) **Form the log-ratio.** For $Q = \mathcal{N}(\hat{w}, I_k)$ and $P = \mathcal{N}(0, I_k)$, the constants and log-determinants cancel, so

$$\ln \frac{q(w)}{p(w)} = \tfrac{1}{2}\Big(\|w\|^2 - \|w - \hat{w}\|^2\Big).$$

(c) **Expand the difference.** Use $\|a-b\|^2 = \|a\|^2 - 2\langle a, b\rangle + \|b\|^2$ to show

$$\ln \frac{q(w)}{p(w)} = \tfrac{1}{2}\big(2\langle w, \hat{w}\rangle - \|\hat{w}\|^2\big) = \langle w, \hat{w}\rangle - \tfrac{1}{2}\|\hat{w}\|^2.$$

(d) **Take expectation under Q.** Recall the identity $\mathbb{E}\langle X, a\rangle = \langle \mathbb{E}X, a\rangle$ for any random vector X and fixed vector a. Use the fact that $w \sim Q = \mathcal{N}(\hat{w}, I_k)$ to conclude

$$D(Q\|P) = \tfrac{1}{2}\,\|\hat{w}\|_2^2.$$

2. (PAC-Bayes bound with explicit target inequality.) Recall the standard PAC-Bayes kl-bound, which states that with probability at least $1-\delta$ over the draw of S,

$$L_{\mathcal{D}}(Q) \;\le\; \hat{L}_S(Q) \;+\; \sqrt{\frac{D(Q\|P) + \ln\!\left(\frac{2\sqrt{n}}{\delta}\right)}{2\,(n-1)}}.$$

Conclude that in our Gaussian setup,

$$L_{\mathcal{D}}(Q) \;\le\; \hat{L}_S(Q) \;+\; \sqrt{\frac{\frac{1}{2}\,\|\hat{w}\|_2^2 + \ln\!\left(\frac{2\sqrt{n}}{\delta}\right)}{2\,(n-1)}}.$$

3. Briefly explain what this bound suggests about why low-ℓ_2-norm solutions (e.g., via L_2-regularization or max-margin bias) are conducive to good generalization.

Problem 2: Typical Generalization in Neural Networks (Guided)

We want to study how a very simple "Guess–and–Check" algorithm can lead to generalization when training neural networks. We will carefully build up the argument in small steps.

Setup.

- A *student network* has weights $w \in \mathbb{R}^k$.
- Inputs are vectors $x \in \mathbb{R}^{d_0}$.
- A training dataset is

$$S = \{x^{(n)}\}_{n=1}^{N}, \quad x^{(n)} \sim P_X \;\text{ i.i.d.}$$

- The labels are generated by a *teacher network* $f_{w^\star}(x)$.
- The student and teacher have the same depth, but the teacher is *narrower* (fewer hidden neurons).

The student has a two-layer form:

$$f_w(x) = \operatorname{sign}\big(w_2^\top [W_1 x]_+\big),$$

where $[\,\cdot\,]_+$ is the ReLU activation (applied coordinate-wise).

Here:

- $W_1 \in \mathbb{R}^{d_1 \times d_0}$ is the first-layer weight matrix that maps the input $x \in \mathbb{R}^{d_0}$ to the hidden layer of size d_1.
- $[W_1 x]_+ \in \mathbb{R}^{d_1}$ is the hidden representation after ReLU.
- $w_2 \in \mathbb{R}^{d_1}$ is the output weight vector combining the hidden activations into a single scalar.
- The function $\operatorname{sign} : \mathbb{R} \to \{-1, +1\}$ outputs $+1$ if its argument is non-negative and -1 otherwise, so the final prediction is binary.

The teacher network is defined in the same form, but it is *narrower*:

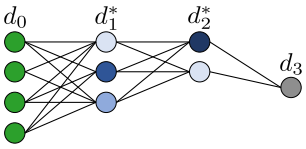
$$f_{w^\star}(x) = \operatorname{sign}\big((w_2^\star)^\top [W_1^\star x]_+\big),$$

where $W_1^\star \in \mathbb{R}^{d_1 \times d_0}$ and $w_2^\star \in \mathbb{R}^{d_1}$. Precisely, only the first $d_1^\star$ hidden neurons are *active*: their incoming and outgoing weights may be nonzero. For the remaining $(d_1 - d_1^\star)$ hidden neurons, both the corresponding rows of $W_1^\star$ and the corresponding entries of $w_2^\star$ are equal to zero. Thus the teacher effectively has width $d_1^\star < d_1$. The teacher's extra neurons are "dead" (all their input and output weights are zero).

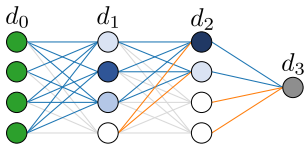
The *Guess–and–Check algorithm*:

1. Sample candidate weights w randomly. Each parameter is chosen independently from q possible values (including 0).
2. Test if this candidate fits all training examples.
3. Stop once we find one that fits perfectly.

We denote the above distribution over weights by P_W.



(a) Teacher network



(b) Student network

Figure 1: **Illustration of teacher and student networks.** The teacher network is narrower, while the student can replicate it by setting certain weights to zero. Blue edges correspond to weights identical to the teacher, while orange edges are forced to zero so that the white neurons have no effect on the output.

Part 1: Probability of guessing the exact teacher.

1. How many parameters does the student have in total? Call this number k.

2. Each parameter is drawn from q possible values. What is the probability of drawing the *exact value* the teacher has for one parameter?

3. Since parameters are drawn independently, what is the probability that all k parameters match the teacher? (Answer should be q^{-k}.)

Part 2: Probability of matching the teacher's function.

1. Do we really need to match *all* the teacher's weights? Or only those corresponding to the *active neurons*?

2. To behave like the teacher, what must happen to the $(d_1 - d_1^\star)$ extra neurons in the student? (Hint: their output weights must be zero.)

3. Put it together: Consider the probability

$$p^\star = P_{w \sim P_W} \left(\forall x \sim P_X \quad f_w(x) = f_{w^\star}(x) \right) .$$

We must correctly guess the input and output weights for each of the $d_1^\star$ active neurons, and guess zero for each of the $(d_1 - d_1^\star)$ inactive neurons. Show that

$$p^\star \;\geq\; q^{-d_0 d_1^\star - d_1}.$$

Part 3: Probability of not fitting after T trials.

1. If the probability of success on one trial is at least $p^\star$, what is the probability of failure on one trial?
2. What is the probability of failing T times in a row? (Answer: $(1-p^\star)^T$.)
3. Take logs of both sides to show:

$$T \;\leq\; \frac{\log \Pr(t > T)}{\log(1 - p^\star)}.$$

Part 4: Generalization bound. We recall a standard inequality:

$$\Pr\left(\epsilon(f) > \frac{\log |F| + \log(1/\delta)}{N}\right) \leq \delta,$$

where $|F|$ is the number of candidate functions.

1. Why is our hypothesis class not the full class of size q^k, but only the set of functions we actually tried? (Hint: we stop after T trials, so the class has size at most T.)
2. Using Part 3, bound $\log T$ in terms of $p^\star$ and $\eta = \Pr(t > T)$.
3. Approximate $\log(1 - p^\star) \approx -p^\star$. Substitute the lower bound for $p^\star$ from Part 2.
4. Put everything together to prove:

   Theorem. With probability at least $(1 - \eta)(1 - \delta)$,

$$\epsilon \;<\; \frac{(d_0 d_1^\star + d_1)\log q \;+\; \log(1/\delta) \;+\; \log\log(1/\eta)}{N}.$$

Part 5: Compare with the naive bound.

1. If we used the full hypothesis class of the student, how many functions are there? (Answer: $|F| = q^k$ with $k = d_0 d_1 + d_1$.) Why?
2. What generalization bound would we get in that case?
3. Compare with our new bound. When is it better? (Hint: when $d_1^\star < d_1$, i.e. the teacher is narrower.)

Problem 3: Implicit Bias of Logistic Regression on Separable Data

In this problem you will analyze a simple example of implicit bias: logistic regression on linearly separable data. Your goal is to show, step by step, that although the loss does not have a finite minimizer, gradient descent converges in direction to the *maximum-margin separator* (the solution of the Hard SVM problem).

(a) **Understanding implicit bias.** Explain in your own words what it means for an optimization algorithm to have an "implicit bias." Why is this concept important for generalization?

Setup. We are given a dataset $\{(x_n, y_n)\}_{n=1}^N$ with $x_n \in \mathbb{R}^d$ and labels $y_n \in \{-1, 1\}$. A linear classifier without bias predicts $\text{sign}(w^\top x)$. The data are *linearly separable* if there exists $w^\star$ such that $y_n (w^\star)^\top x_n > 0$ for all n. Recall the Hard SVM problem:

$$\min_w \|w\| \quad \text{s.t. } y_n w^\top x_n \geq 1 \;\; \forall n.$$

(b) **Logistic regression loss.** In logistic regression the empirical loss is

$$\mathcal{L}(w) = \sum_{n=1}^N \log\big(1 + \exp(-y_n w^\top x_n)\big).$$

Explain why we may assume $y_n \equiv 1$ w.l.o.g. by reflecting the inputs ($x_n \leftarrow y_n x_n$), and note when this reduction also holds for deeper networks.

(c) **Gradient flow dynamics.** Consider the continuous-time gradient descent (*gradient flow*)

$$\dot{w}(t) \;=\; -\nabla \mathcal{L}(w(t)) \;=\; \sum_{n=1}^N x_n \, \exp\big(-w(t)^\top x_n\big).$$

Show that $\mathcal{L}(w(t))$ is decreasing and $\mathcal{L}(w(t)) \to 0$ as $t \to \infty$. Conclude that $\|w(t)\| \to \infty$. *(Hint: use* $\frac{d}{dt}\mathcal{L}(w(t)) = \nabla \mathcal{L}(w(t))^\top \dot{w}(t) = -\|\dot{w}(t)\|^2 \leq 0$*.)*

(d) **Which points matter?** Examples with large margin $w(t)^\top x_n \gg 1$ have exponentially small contribution to $\dot{w}(t)$. Which examples continue to matter as $\|w(t)\| \to \infty$? Argue that, asymptotically, only the points closest to the current separating hyperplane (the *support vectors*) influence the direction of $\dot{w}(t)$.

(e) **Normalize by the minimum margin.** Define the (time-varying) *minimum margin*

$$\gamma(t) \;=\; \min_n \, w(t)^\top x_n \qquad \text{and} \qquad v(t) \;=\; \frac{w(t)}{\gamma(t)}.$$

(i) Show that $\gamma(t) \to \infty$ and that $v(t)$ is well-defined for large t with $\min_n v(t)^\top x_n = 1$ by construction.

(ii) Using the gradient flow, write $w(t) = w(0) + \sum_{n=1}^N x_n \int_0^t \exp(-w(s)^\top x_n)\, ds$. Divide both sides by $\gamma(t)$ to obtain

$$v(t) \;=\; \frac{w(0)}{\gamma(t)} + \sum_{n=1}^N x_n \, \alpha_n(t), \quad \text{where} \quad \alpha_n(t) \;=\; \frac{1}{\gamma(t)} \int_0^t \exp\big(-w(s)^\top x_n\big)\, ds.$$

Argue that $w(0)/\gamma(t) \to 0$ and hence any limit point of $v(t)$, call it $v^\star$, has the form $v^\star = \sum_{n=1}^N \alpha_n^\star x_n$ with $\alpha_n^\star \geq 0$.

(iii) Show that only support vectors can have $\alpha_n^\star > 0$. *(Hint: if x_n is always strictly above the minimum margin by a fixed gap, its exponential weight becomes negligible relative to $\gamma(t)$.)*

(iv) Prove that every limit point $v^\star$ satisfies the *margin equalities* ${v^\star}^\top x_n = 1$ for all support vectors x_n, and ${v^\star}^\top x_m \geq 1$ for all remaining examples. Conclude that $v^\star$ is a feasible solution to the Hard SVM constraints with unit minimum margin.

(f) **Identifying the limit direction.** In part (e) you showed that any limit point $v^\star$ of the normalized weights

$$v(t) = \frac{w(t)}{\gamma(t)} \qquad \text{with } \gamma(t) = \min_n w(t)^\top x_n$$

satisfies:

- $v^\star$ is a combination of the support vectors,

- $v^{\star\top} x_n = 1$ for every support vector x_n, and
- $v^{\star\top} x_m \geq 1$ for all other points.

Now use only geometry to argue that $v^\star$ must be the *maximum-margin separator*:

(g) **Identifying the limit direction (very guided).** From part (e) we know that any limit point $v^\star$ satisfies:

- $v^{\star\top} x_n = 1$ for every support vector x_n,
- $v^{\star\top} x_m \geq 1$ for all other points.

Let S denote the set of indices of the support vectors.

(i) **Feasible set.** Write down the feasible set of all weight vectors as

$$\mathcal{F} = \{\, w \in \mathbb{R}^d : w^\top x_n \geq 1 \;\; \forall n \,\}.$$

Explain in words why this is an intersection of halfspaces, and note that the support vectors are exactly those with $w^\top x_i = 1$. This defines the set

$$\mathcal{A} = \{\, w : w^\top x_i = 1 \;\; \forall i \in S \,\}.$$

Explain why this is an intersection of affine spaces.

(ii) **Projection idea.** State that among all points in $\mathcal{A}$, the one with smallest Euclidean norm is the point in $\mathcal{A}$ *closest to the origin*, and denote it $w_{\min}$.

(iii) **Why unique?** Explain in one sentence why this projection is unique (in any affine subspace there is only one point closest to the origin).

(iv) **Connect to $v^\star$.** Recall that $v^\star \in \mathcal{A}$ and is a combination of the support vectors. *Instruction:* argue that since the projection point also lies in the span of the support vectors, and the minimum-norm point is unique, it must be that

$$v^\star = w_{\min}.$$

(v) **Conclude about directions.** Now we want to go from the convergence

$$\frac{w(t)}{\gamma(t)} \longrightarrow v^\star$$

to a statement about the unit vector $w(t)/\|w(t)\|$.

(a) Write the identity

$$\frac{w(t)}{\|w(t)\|} = \frac{\gamma(t)}{\|w(t)\|} \cdot \frac{w(t)}{\gamma(t)}.$$

Here the second factor has a limit $v^\star$ from part (e).

(b) Argue that the scalar $\gamma(t)/\|w(t)\|$ also has a limit. *Hint:* take norms of both sides in the convergence $\frac{w(t)}{\gamma(t)} \to v^\star$ to deduce that $\|w(t)\|/\gamma(t) \to \|v^\star\|$, so its reciprocal tends to $1/\|v^\star\|$.

(c) Combine the two limits:

$$\frac{w(t)}{\|w(t)\|} \longrightarrow \frac{1}{\|v^\star\|}\, v^\star = \frac{v^\star}{\|v^\star\|}.$$

(d) Finally, recall that in the previous step we argued $v^\star$ must equal $w_{\min_{\text{norm}}}$, the unique minimum-norm feasible vector (the orthogonal projection of the origin onto the support-vector affine subspace). Conclude that the direction of $w(t)$ converges to $w_{\min}/\|w_{\min}\|$, which is exactly the hard-SVM (maximum-margin) separator.

Takeaway. Although logistic regression has no finite minimizer on separable data, gradient descent implicitly biases the solution toward the maximum-margin separator, a predictor known to generalize well.

Problem 4: Influence Functions

At first sight, computing influence functions appears difficult due to the inverse Hessian computation, which naively has cubic complexity in the number of parameters. In this question, we will see how to use the matrix power series for fast but approximate computation of the form $H^{-1}v$. The key idea is a simple identify in the following question.

1. If A is any positive definite matrix with full rank and maximum eigenvalue less than 1, then show that

$$A^{-1} = \sum_{i=0}^{\infty} (I - A)^i.$$

Hint: Note that A is diagonalizable. How do eigenvectors and eigenvalues of A^i relate to those of A?

2. If S_r denotes the truncation of the series to its first r terms, then show that $\lambda_{max}(A^{-1} - S_r) \leq \lambda_{min}(A)^{-1}(1 - \lambda_{min}(A))^{r+1}$. Compute a value of r for which the r-term approximation to the Hessian-vector product is within $(1 + \epsilon)$ multiplicative factor of the correct value.

Problem 5: Shapley Values

The *Shapley value* is a way to distribute credit among N players in a cooperative game with utility function $U(\cdot)$. Formally, the Shapley value of player i is

$$s_i = \mathbb{E}_\pi[U(\pi_{\leq i}) - U(\pi_{<i})],$$

where π is a random permutation of $\{1, \ldots, N\}$ and $\pi_{<i}$ are the players before i in π.

The following are natural axioms for any credit-attribution scheme:

1. **Efficiency:** $\sum_i s_i = U([N])$.
2. **Symmetry:** If $U(S \cup \{i\}) = U(S \cup \{j\})$ for all S not containing i, j, then $s_i = s_j$.
3. **Linearity:** For utilities U_1, U_2, the values for $U_1 + U_2$ equal the sum of the values for U_1 and U_2.
4. **Null Player:** If $U(S \cup \{i\}) = U(S)$ for all S not containing i, then $s_i = 0$.

By linearity of expectation in the definition of Shapley value, all four axioms are satisfied.

1. In the first part of the exercise we will use a toy model to make the non-linearity issue of neural networks concrete and easy to analyze. The goal is to see how the Shapley axioms produce a mathematically "fair" but potentially misleading explanation for a model with strong feature interactions.

   Consider a simple 2x2 binary image with four pixels: Top-Left (P_{TL}), Top-Right (P_{TR}), Bottom-Left (P_{BL}), and Bottom-Right (P_{BR}).

   A simple "corner detection" model is defined as follows. It outputs a high score (logit) if and only if three specific pixels are "on" (value 1): P_{TL}, P_{TR}, and P_{BL}.

   The utility function $U(S)$ for a set of pixels S is:

$$U(S) = \begin{cases} 1 & \text{if } \{P_{TL}, P_{TR}, P_{BL}\} \subseteq S, \\ 0 & \text{otherwise.} \end{cases}$$

   The image we want to explain is $\{P_{TL} = 1, P_{TR} = 1, P_{BL} = 1, P_{BR} = 0\}$. For this image, the total utility is $U(\{P_{TL}, P_{TR}, P_{BL}\}) = 1$. We assume the baseline (empty set) utility $U(\emptyset) = 0$.

   (a) Calculate the Shapley values for all four pixels: s_{TL}, s_{TR}, s_{BL}, and s_{BR}.

   (b) Analyze the result:

   - How does the model distribute the credit for the prediction?
   - Does the Null Player axiom hold for the P_{BR} pixel?
   - Do the Symmetry and Efficiency axioms hold? How do they help you find the solution?
   - Discuss whether the resulting Shapley values provide a "true" explanation of how this AND-gate-like model works. What insight do the values provide, and what do they obscure about the model's logic?

2. Computing Shapley values exactly is NP-hard, since it requires evaluating $U(S)$ for all 2^N subsets. A simple approximation method is to sample $O(R^2 N \log N/\epsilon^2)$ random permutations and estimate the expectation in the definition

$$s_i = \mathbb{E}_\pi[U(\pi_{\leq i}) - U(\pi_{<i})] \,.$$

With high probability this approximates the vector of Shapley values within ℓ_2 error at most ϵ.

A more efficient approach is as follows:

(a) Approximate s_1 within error $\epsilon/(2\sqrt{N})$ using the naive method above.

(b) Use the following fact to estimate the differences $s_1 - s_j$ for all j, within error $\epsilon/2\sqrt{N}$:

$$s_i - s_j \;=\; \frac{1}{N-1} \sum_{S \subseteq [N] \setminus \{i,j\}} \frac{U(S \cup \{i\}) - U(S \cup \{j\})}{\binom{N-2}{|S|}}.$$

Design and analyze step the above approach in detail. The key insight is that a single randomly chosen subset S can be used to estimate the marginal contribution of a player i versus all other players $j \notin S \cup i$ simultaneously, leading to significant computational savings