Chemistry 213/313 Test
April 30, 2025

Name: KEY

Write directly on the test. No electronic devices allowed. Points are as indicated below.

Question	Part	Points available	Points achieved
1	(a)	2	
	(b)	5	
	(c)	4	
	(d)	4	
	(e)	3	
2	(a)	6	
	(b)	6	
	(c)	2	
3	(a)	4	
	(b)	5	
	(c)	4	
	(d)	3	
4	(a)	6	
	(b)	3	
	(c)	3	
total		60	

1. In Rogers and Hopfinger, “Application of genetic function approximation to quantitative structure-activity relationships and quantitative structure-property relationships” *Journal of Chemical Information and Computer Sciences* (1994), the authors seek to identify quantitative-structure activity relationships (QSARs) by linear regression with models in which the basis functions are chosen through a genetic algorithm.

   (a) The loss function in that paper is

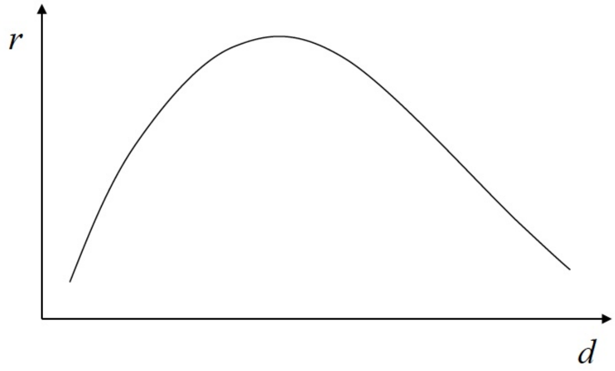
   $$\mathcal{L} = \text{LSE} \bigg/ \left(1 - \frac{c + dp}{M}\right)^2 ,$$

   where LSE is a standard least-squares error, c is the number of basis functions, p is the total number of features contained in all basis functions, M is the number of samples in the training set, and d is a hyperparameter. Briefly explain how this loss can disfavor models with large c or p. (2 points)

   Assuming $M > c + dp$, increasing c or p will decrease the denominator and make the loss function larger for a given value of LSE. By the same token, decreasing c or p will make the loss function smaller.

   (b) Sketch how you expect the cross-validated performance as measured by Pearson’s correlation coefficient between the empirical and predicted values, r, to vary as d increases. Briefly explain the key aspects of your sketch. (5 points)

   At small d, there will be little regularization, so the model will tend to overfit the training data and perform worse on the validation data. At large d, the model will be limited in its capacity (have too few basis functions) and thus will tend to underfit the training and validation data.



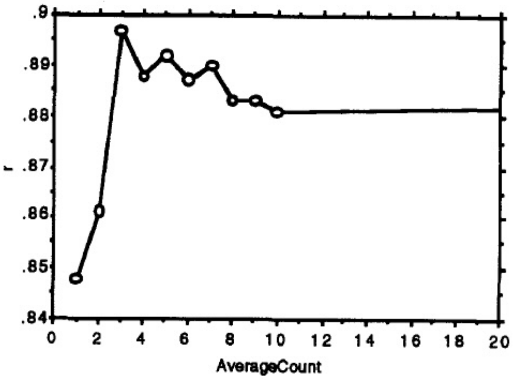
   (c) Identify two aspects of this loss function that make this regularization approach less attractive than L1 and L2 regularization. (4 points)

   Two of

   - The loss function does not reduce to the LSE for $d = 0$.
   - The loss function is zero when LSE is zero regardless of the value of the denominator.
   - c varies in an integer fashion and is not directly affected by d, limiting control of c.
   - The denominator can be zero when $(c + dp)/M = 1$.
   - The denominator is not necessarily a monotonic function of c, d, and p (whether it is in practice depends on their values relative to M).

(d) The genetic algorithm evolves a population of models. Two models can be randomly cut and combined, and a new basis function can be added or modified; the resulting model replaces the one with the largest loss function in the previous population.

In the plot to the right, the predictions of the highest-scoring models are averaged and the cross-validated Pearson's correlation coefficient between the empirical and predicted values, r, is computed. Explain why the performance first increases and then decreases as the number of models averaged increases. (4 points)



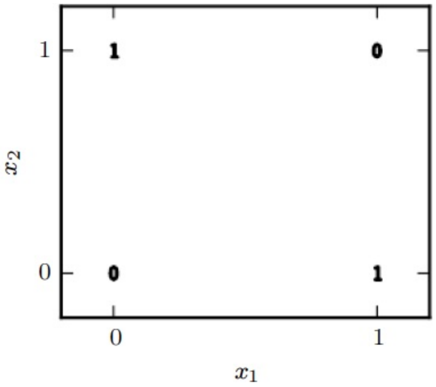
Initially, averaging the predictions will tend to reduce the variance of predictions, similarly to other ensemble methods. This will tend to improve the predictions for the validation data. However, because models are added in an ordered fashion based on their performance, the models added later will be worse and tend to decrease the average.

(e) Explain why the gain from averaging is relatively modest in this case. (3 points)

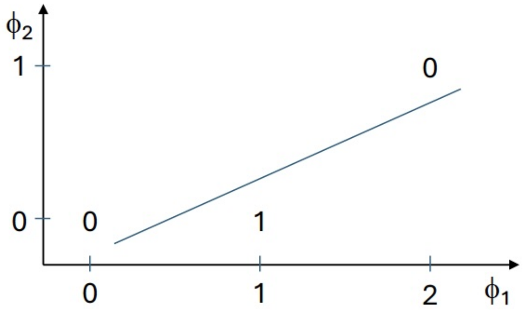
Owing to the way models are generated by the genetic algorithm, they will tend to be strongly correlated (e.g., have overlapping features), so the reduction in variance due to averaging will tend to be small.

2. The four digits on the plot below represent two classes (0 and 1).

   (a) Identify basis functions that can enable perfect classification and plot the four digits and a possible linear decision boundary in the space of basis functions. Take care to indicate the scales of the axes. (6 points)



There are many possible answers. One is $\phi_1 = x_1 + x_2$ and $\phi_2 = \text{ReLU}(x_1 + x_2 - 1)$.



   (b) Binary cross entropy is commonly used as a loss function for classification. Provide a mathematical justification for this loss function, making clear the assumptions underlying it. (6 points)

Based on an assumption of Bayesian likelihood, one can model $p(C_1 \mid \phi) = \sigma(w^\top \phi)$, where C_1 is one class, σ is the logistic sigmoid function, w is a vector of weights, and ϕ is a vector of basis functions. That is we assume that the probability of the class is a sigmoid function that takes as its argument a linear combination of the basis functions. Then, we model the likelihood as a Bernoulli distribution:

$$p(y \mid w) = \prod_{n=1}^{N} [p(C_1 \mid \phi)]^y [1 - p(C_1 \mid \phi)]^{1-y}.$$

Because the logarithm is monotonic, we can maximize the likelihood by minimizing

$$-\log p(y \mid w) = -\sum_{n=1}^{N} [y \log \sigma(w^\top \phi) + (1 - y) \log(1 - \sigma(w^\top \phi))]$$

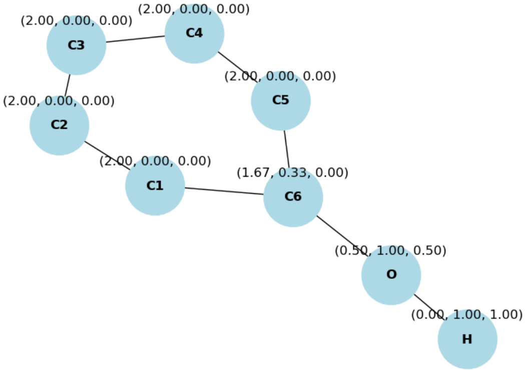
   (c) Should the binary cross entropy ever be negative? Why or why not? (2 points)

No. The class labels are nonnegative, while both the logarithms should be negative since their arguments are models of probabilities, which are between zero and one. The overall expression should be nonnegative owing to the minus sign in front.

3. (a) A well-known issue for neural networks is that they can suffer from vanishing or exploding gradients in which the gradients become very small or very large, respectively. Explain why this issue becomes more significant as networks become deeper. (4 points)

The gradients are computed by backward propagation of error (backprop), which accumulates the error as a product over layers (each layer contributes a factor). When there are many layers, there is a greater likelihood of multiplying many small or large numbers.

(b) The graph to the right shows the output from the first message-passing layer of a graph neural network that takes as its initial features $\mathbf{h}_i^{(0)}$ a one-hot encoding of the atom types. The example shown is phenol with implicit aromatic hydrogen atoms. Write the node update function, making clear the meaning of all symbols. (5 points)



$$\mathbf{h}_i^{(\ell+1)} = \mathbf{h}_i^{(\ell)} + \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{h}_j^{(\ell)}$$

where $\mathbf{h}_i^{(\ell)}$ is the feature vector for node i at layer ℓ, $\mathcal{N}$ is the list of neighbors of node i, and $|\mathcal{N}_i|$ is its size (number of neighbors).

(c) An issue with using message-passing graph neural networks to predict molecular properties from chemical structures is that it can be hard to capture long-range dependencies. Explain why this is the case. (4 points)

Each message passing layer typically transfers information between neighbors. To communicate information across the graph can require many layers. Because each layer mixes the information from multiple nodes, the feature vectors can tend to become homogeneous when there are many layers. This phenomenon is known as oversmoothing.

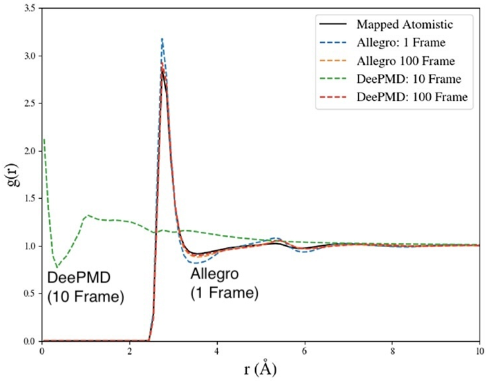
(d) How do you expect the complexity (and in turn computational cost for training and inference) of a graph neural network to scale with the number of nodes in the graph, the average node degree, and the feature vector dimension? (3 points)

Complexity scales as $O(nkd^2)$, where n is the number of nodes, k is the average degree (akin to a the filter size in a convolutional neural network), and d is the size of the feature vectors (in the example above, the message passing operation is linear in d, but in general any feature component can be mixed with any other, giving rise to the d^2 scaling).

4. (a) A researcher develops a convolutional neural network to analyze protein sequence data. Will a standard architecture exhibit equivariance to reflections? Why or why not? Would that be a desirable property of the architecture in this case? (6 points)

In the absence of constraints to the contrary, the filter learned in a convolutional neural network will not be symmetric, so that reflecting the data will result in different output from the filter. Equivariance of reflections is not desirable for analyzing protein sequences since they are directional (run between N- and C-termini).

(b) In Loose *et al.* "Coarse-graining with equivariant neural networks: A path toward accurate and data-efficient models." *Journal of Physical Chemistry B* (2023), the authors compare an invariant neural network architecture (DeePMD) with an equivariant one (Allegro) with respect to learning a coarse-grained force field from atomistic molecular dynamics data. A figure from the paper is reproduced to the right (curves that deviate visibly from others are labeled). What conclusion can one draw from the figure? (3 points)



Less data are required to train the equivariant neural network than the invariant one. Allegro with 1 frame is already close to the atomistic reference, while DeePMD with 10 frames is still quite far.

(c) Equation 4 of the above paper is

$$s(r_{ij}) = \begin{cases} \frac{1}{r_{ij}}, & r_{ij} < r_{c1} \\ \frac{1}{r_{ij}} \left\{ \frac{1}{2} \cos\left[\pi \frac{(r_{ij} - r_{c1})}{(r_{c2} - r_{c1})} \right] + \frac{1}{2} \right\}, & r_{c1} \leq r_{ij} \leq r_{c2} \\ 0, & r_{ij} > r_{c2} \end{cases}$$

where r_{ij} is the distance between atoms i and j and r_{c1} and r_{c2} are cutoff distances. Is this function invariant or equivariant? Briefly justify your answer. (3 points)

The function is invariant because it depends only on distances between atoms, which will not be sensitive to translations, rotations, or reflections of the system.