Chemistry 313 Exam Revisions

Connor Blake

May 12, 2025

Questions

In writing this, I realized that most of my answers were basically exactly what the given key says, so I am confused why I lost points at all, particularly on 3a, 3b, 4b

1

a

+1 Write assumption that for it to make sense, $M > c + dp$
+1 As c, p increase, they will make the loss function larger for a given LSE and vice versa

b

+1 A plot should show that for low and high d, r is low but there is a maximum in r for intermediate levels of d
+1 explanation that the graph shows a **local maximum** in d at intermediate levels of regularization
+1 $d \neq 0$ so that some regularization **prevents overfitting**
+1 high d **penalizes the number of learned basis functions** and will lead to poor performance
+1 high d will **underfit** the model

c

Two of (+2 each)

- The loss function does not reduce to the LSE for $d = 0$f.
- 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

+2 the high model count decrease in r is explained by the aggregate models at the end being worse since they are overwhelmed by bad models dragging down the average
+2 the initial increase in r for low count is explained by a decrease in variance of the prediction by having an ensemble prediction

e

+1 the genetic algorithm will generate models that are increasingly similar to each other as time goes on +2 the model errors are correlated with each other, so the gain is not as high as would be predicted theoretically with an ensemble

2

a

+1 an input-output table of the XOR function
+3 basis functions properly evaluate the XOR function
+2 the graph properly shows the transformed inputs correctly identifying the outputs

$$\phi_1 = x_1 + x_2$$
$$\phi_2 = \text{ReLU}(x_1 + x_2 - 1)$$

x_1	x_2	$\text{XOR}(x_1, x_2)$	$\phi_1 = x_1 + x_2$	$\phi_2 = \text{ReLU}(x_1 + x_2 - 1)$
0	0	0	0	0
0	1	1	1	0
1	0	1	1	0
1	1	0	2	1

b

+1 Cross entropy assumes Bayesian likelihood +1 The probability is expressed in the form $p(C_1|\phi) = \sigma(w_i\phi_i)$ with a sigmoid of a linear combination of weights +2 the likelihood is modeled as Bernoulli distribution:

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

+2 Minimizing cross entropy is the same as maximizing the log likelihood:

$$-log(p|w) = -\sum_{n=1}^{N}[y\log(\sigma(w_i\phi_i) + (1-y)\log(1-\sigma(w_i\phi_i))]$$

c

+2 the product of prefactor (-), "probability" $\in [0,1]$, log of "probability" $\in [0,1]$ (-) must be positive

3

a

+1 the gradients arise during backpropagation
+2 repeated multiplication by chain rule may result in very large or small numbers
+1 deeper networks can make larger or smaller numbers than shallower networks

b

$$h_{ik}^{l+1} = h_{ik}^{l} + \frac{1}{|N_i|}\sum_{j\in N_i} h_{jk}^{l}$$

(This answer actually doesn't make any sense by the way - everything would just get smoothed out since there are no trainable weights or nonlinear activations...) i is the vertex index, k is the feature index, N_i is the set of all neighbor node indices of node i

A BETTER SOLUTION (would be my original)

$$h_{ij}^{k+1} = \phi(\frac{1}{d_i}e_{il}h_{lm}w_{mj})$$

where i, l index the vertices, j, m index the features, e is the adjacency matrix/graph, w is the weight matrix
+3 for the formula showing linear updates (or wrapped by nonlinear) on each of the nearest neighbors only (via adjacency matrix or explicit sum on neighbors)
+2 all index labels accurately describe the features/nodes/indexing sets

c

+2 only nearest neighbors are communicated with in a given step l
+2 long-range interactions are difficult to propagate because the long distance messages get washed out because of averaging on nearest neighbors

d

$$\mathcal{O}(nkd^2)$$

where n is the number of vertices, k is the average degree, d is the number of features per node
+1 it is linear in the number of vertices due to an update on every step of each vertex
+1 it is linear in the average degree due to an average on each of the nearest neighbors on each vertex on each step
+1 it is quadratic in the feature number because each of the features is both updated and contracted over in forming the updates to h_{ik} (not shown in "given" solution to b, but my original solution shows this)

4

a

+2 standard convolution is not a reflection-equivariant operation because reflection is a highly "nonlocal" operation and convolution is not guaranteed to be symmetric
+2 equivariance means that the reflection is preserved, and this is not a property of an asymmetric operation like convolution
+2 equivariance to reflection is not a desirable feature because amino acid sequences have a natural direction from N to C terminations

b

+2 the equivariant network is more data efficient than the invariant one
+1 Allegro is very close using only a single frame while DeePMD is not close after 10 frames

c

+1 understanding of what is meant by invariant that it $f(Tx + g) = f(x)$
+1 derivation showing that mean squared distance is invariant under rigid transformations
+1 derivation showing the whole function is invariance if r_{ij} is transformation invariant