Discussion Session Problems 3

01/29/2026

1. The universal approximation theorem states that a feedforward neural network with a single hidden layer can approximate any function over a compact set, given enough neurons. If this is true, why is deep learning so successful in practice? Why do we still need depth instead of just using one very wide layer?

2. In neural networks, what is the role of the activation function and why do we need it?

3. For a softmax applied to the values $(3, 4, 1, 7)$, which of the following could be a possible output?

   A) 0.0171, 0.0465, 0.0023, 0.9341

   B) 0.0011, 0.0085, 0.0003, 0.8362

   C) 0.0023, 0.0171, 0.0465, 0.9341

   D) 0.0211, 0.0785, 0.0103, 0.9362

4. Suppose you are solving three problems: linear regression, logistic regression, and a small neural network. Which one is most likely to benefit from a newly discovered extremely fast large-scale matrix multiplication algorithm? Why?

5. Consider a dataset $D = \{x^{(1)}, \ldots, x^{(100)}\}$ where each $x^{(i)} \in \mathbb{R}^3$. This is a 3-class classification problem. We use a neural network with two hidden layers of sizes 4 and 5, using sigmoid activations in the hidden layers and a softmax output layer.

   a) How would you graphically represent this neural network?

   b) What are the feedforward equations for a single training example $x \in \mathbb{R}^3$?

   c) Describe the relationship between the graphical representation and the feedforward equations. What do the nodes and edges represent?

   d) What is the total number of parameters in this neural network?

6. A wildlife monitoring system uses a Softmax regression model to classify images of animals into $K = 4$ classes:

   $$\{\text{deer}, \text{fox}, \text{bear}, \text{rabbit}\}.$$

   Each image is represented by a feature vector $x \in \mathbb{R}^n$. The model assigns a score

   $$z_k = x^T W_k + b_k$$

   to each class k.

   (a) Explain why the scores $\{z_k\}_{k=1}^K$ cannot be directly interpreted as probabilities.

   (b) Write down the Softmax probability assigned to class k.

   (c) Suppose the true class label is ℓ. Using a one-hot encoding for the label, derive the Softmax cross-entropy cost for this single example and simplify it as much as possible.

7. Suppose we are training a Softmax regression classifier for handwritten digit recognition, where each image belongs to one of $K = 10$ digit classes. Consider a single training example (x, y), where y is one-hot encoded and the true class is ℓ.

   (a) Starting from the simplified cost

   $$J = -z_\ell + \log \sum_{j=1}^{K} e^{z_j},$$

   compute the partial derivative $\frac{\partial J}{\partial z_k}$ for an arbitrary class k.

   (b) Show that your result can be written in terms of the predicted probabilities $\hat{y}_k$ and the true labels y_k.

   (c) Give an intuitive interpretation of the gradient when $k = \ell$ and when $k \neq \ell$.

8. You are training a Softmax regression model on a dataset with m examples, n features, and K classes. Let:

   - $X \in \mathbb{R}^{m \times n}$ be the data matrix,
   - $W \in \mathbb{R}^{n \times K}$ be the weight matrix,
   - $Y \in \mathbb{R}^{m \times K}$ be the matrix of one-hot labels,
   - $\hat{Y} \in \mathbb{R}^{m \times K}$ be the matrix of predicted Softmax probabilities.

   (a) Write the total Softmax cross-entropy cost over all m examples.

   (b) Using your knowledge of the single-example gradient, derive the gradient of the cost with respect to $W_{k,j}$.

   (c) Show how this expression can be written in fully vectorized form and verify that the dimensions are correct.

9. Why is cross-entropy loss typically preferred over mean squared error when training a neural network for classification with a Softmax output layer?

10. In deep neural networks, why can using the sigmoid activation function in many layers lead to slow or difficult training?

11. What is the purpose of bias terms in a neural network layer? What would happen if we removed all bias terms?

12. Why is vectorization important when implementing neural networks and gradient computations?

13. A neural network with nonlinear activation functions can produce nonlinear decision boundaries. Explain why a network with only linear activations cannot do this, no matter how many layers it has.