EE599/699 Deep Learning Fundamentals HW 1

Fall 2025 Test Student

Part 1: Shallow Networks & Expressive Power

Problem 1.1: General Shallow Network Analysis

Consider a general shallow network designed for a complex task with D_i inputs, D hidden units, and D_o outputs.

A.

Parameter Count: State the formula for the total number of parameters (N_{params}) Parameter Count: State the formula for the total number of parameters (N_{params}) in this network. Calculate N_{params} if $D_i = 5$, $D = 100$, and $D_o = 3$. Provide the calculation using both methods of reasoning discussed in the lecture slides (weights/biases and layer-by-layer).

Solution:

The total number of parameters is:

$$
\begin{aligned}
N_{\text{params}} &= (D_i \times D + D) + (D \times D_o + D_o) \\
N_{\text{params}} &= D_i \cdot D + D + D \cdot D_o + D_o
\end{aligned}
$$

For $D_i = 5$, $D = 100$, and $D_o = 3$:

$$
\begin{aligned}
N_{\text{params}} &= 5 \times 100 + 100 + 100 \times 3 + 3 \\
N_{\text{params}} &= 500 + 100 + 300 + 3 \\
N_{\text{params}} &= 903
\end{aligned}
$$

B.

Network Visualization and Structure: For a simplified version of this network where $D_i = 2$, $D = 3$, and $D_o = 1$, state the dimensions of the weight matrices (Ω) and bias vectors (β) used for the hidden layer and the output layer.

Solution:

- Hidden layer: $\Omega_1 \in \mathbb{R}^{3\times 2}$, $\beta_1 \in \mathbb{R}^{3\times 1}$
- Output layer: $\Omega_2 \in \mathbb{R}^{1\times 3}$, $\beta_2 \in \mathbb{R}^{1\times 1}$

C.

Piecewise Linearity: Recall the four-step process (Linear $\rightarrow$ Activation $\rightarrow$ Weight $\rightarrow$ Sum) that creates the output of the shallow network. Why is the output of this ReLU-based shallow network always piecewise linear? How does the number of hidden units (D) relate to the maximum number of linear segments/regions created when the input dimension $D_i = 1$?

Solution:

The output is piecewise linear because:

1. Each hidden unit applies a linear transformation followed by ReLU
2. ReLU creates a piecewise linear function (0 or identity)
3. The output is a weighted sum of these piecewise linear functions
4. A weighted sum of piecewise linear functions is piecewise linear

For $D_i = 1$, the maximum number of linear segments is $D + 1$.

Problem 1.2: Exploring Linear Regions

A.

The maximum number of linear regions created by D hyperplanes (corresponding to D hidden units) in a D_i-dimensional input space ($D > D_i$) is given by Zaslavsky (1975). Write down the mathematical formula for this maximum number of regions based on the lecture material, and then calculate it for $D_i = 2$ and $D = 4$.

Solution:

The Zaslavsky formula is:

$$R(D, D_i) = \sum_{k=0}^{D_i} \binom{D}{k}$$

For $D_i = 2$ and $D = 4$:

$$R(4,2) = \binom{4}{0} + \binom{4}{1} + \binom{4}{2}$$
$$R(4,2) = 1 + 4 + 6$$
$$R(4,2) = 11$$

Part 2: Loss Functions and Training

Problem 2.1: Cross-Entropy Loss

Derive the gradient of the cross-entropy loss function for binary classification.

A.

Write the cross-entropy loss function for a single example.

Solution:

$$\mathcal{L}(y, \hat{y}) = -[y \log(\hat{y}) + (1-y)\log(1-\hat{y})]$$

B.

Compute the derivative with respect to the predicted value $\hat{y}$.

Solution:

$$\frac{\partial \mathcal{L}}{\partial \hat{y}} = -\left[\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right]$$
$$\frac{\partial \mathcal{L}}{\partial \hat{y}} = \frac{\hat{y}-y}{\hat{y}(1-\hat{y})}$$

Problem 2.2: Regularization Techniques

A.

Explain L1 and L2 regularization and their effects on model weights.

Solution:

- **L2 Regularization** (Ridge): Adds $\lambda \sum_i w_i^2$ to the loss. Encourages small weights but rarely exactly zero. Equivalent to Gaussian prior.
- **L1 Regularization** (Lasso): Adds $\lambda \sum_i |w_i|$ to the loss. Encourages sparse solutions with many weights exactly zero. Equivalent to Laplace prior.

B.

When would you prefer L1 over L2 regularization?
Solution:
Prefer L1 when:

1. You want automatic feature selection
2. You believe only a few features are truly relevant
3. Interpretability is important (sparse models are easier to interpret)
4. Storage/computation efficiency is critical (sparse weights can be efficiently stored)

Part 3: Backpropagation

Problem 3.1: Chain Rule Application

Consider a simple network with one hidden layer. Derive the backpropagation equations.

A.

Write the forward pass equations.
Solution:

$$
\begin{aligned}
z^{[1]} &= W^{[1]}x + b^{[1]} \\
a^{[1]} &= \sigma(z^{[1]}) \\
z^{[2]} &= W^{[2]}a^{[1]} + b^{[2]} \\
\hat{y} &= \sigma(z^{[2]})
\end{aligned}
$$

B.

Derive the gradient for the output layer weights.
Solution:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W^{[2]}} &= \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial W^{[2]}} \\
\frac{\partial \mathcal{L}}{\partial W^{[2]}} &= \delta^{[2]} \cdot (a^{[1]})^T
\end{aligned}
$$

where $\delta^{[2]} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \sigma'(z^{[2]})$.

This test demonstrates all three parts with hierarchical numbering: 1.1, 1.2, 2.1, 2.2, 3.1