Home Assignment 01

Text Mining Summer Semester 2025

Giulia Luongo 10076102
Mohammad Sukri 10062921
Samer Sakor 10074501

1 Gradient Descent

1. To train a neural network, its parameters $\theta = (\theta_1, \ldots, \theta_p)$ are adjusted to minimize the loss function $J(\theta)$, which measures the difference between predictions and target values.

   The gradient is the vector of partial derivatives:

   $$\nabla_\theta J(\theta) = (\frac{\partial J}{\partial \theta_1}, \ldots, \frac{\partial J}{\partial \theta_p})$$

   It indicates both the direction and magnitude we should change each parameter in order to reduce $J(\theta)$. In particular, the sign of $\frac{\partial J}{\partial \theta_i}$ reveals whether increasing or decreasing θ_i will lower the loss, while its absolute value determines how strongly the adjustment should be.

   Once the gradient is computed, all parameters are updated with the gradient descent rule:

   $$\theta \leftarrow \theta - \alpha \nabla_\theta J(\theta)$$

   where the learning rate α controls the size of each update step.

2. The loss function $J(\theta)$ measures the difference between the network's predictions $\hat{y}$ and the targets y. It is used to compute the gradient and, consequently, update the parameters. Depending on the problem (e.g., classification, regression), different loss functions are preferred. For a regression problem, for example, we commonly use the mean squared error:

   $$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

   If a loss function that is not appropriate for our task is used, we risk missing the true nature of the problem and, as a result, not updating the network's parameters in the most effective way to approximate the underlying function.

   Choosing the wrong loss can result in gradients that are misaligned with the actual goal, causing the network to converge to a solution that appears good under the loss but performs poorly on the real task.

3 Embeddings

1. Embeddings convert complex data like words, images, or documents into numerical vectors that computers can process.

   Embeddings offer several benefits:

   Capturing Meaning and Relationships

   - Word embeddings like "man" and "women" appear close together in vector space
   - The vector difference between "king" and "man" approximates the difference between "queen" and "woman"
   - Helps models understand semantics and context

   Dimensionality Reduction

   - Transforms high-dimensional vectors into low-dimensional vectors
   - Makes computations faster and more efficient

   Enabling Generalization

   - Trained embeddings help models process new inputs
   - Similar items have similar vector representations

   Facilitating Advanced Operations

   - Enables similarity search based on meaning rather than exact matches
   - Powers recommendation systems, search engines, and clustering tasks

   In short, embeddings transform complex data into a format that preserves meaning while being more accessible for machine learning models.

References

1. Neural Network Python | How to make a Neural Network in Python | Python Tutorial | Edureka. https://www.youtube.com/watch?v=9UBqkUJVP4g
2. Neural Networks from Scratch – P.1 Intro and Neuron Code. https://www.youtube.com/watch?v=Wo5dMEP_BbI&list=PLQVvvaa0QuDcjD5BAw2DxE6OF2tius3V3
3. Understanding the training and validation loss curves. https://www.youtube.com/watch?v=p3CcfIjycBA