2025

CS 4407
Data Mining & Machine Learning

LEARNING JOURNAL UNIT 7

SANA UR REHMAN

INSTRUCTOR: NIRMAL ADHIKARI

WEEK 7 LEARNING JOURNAL: ARTIFICIAL NEURAL NETWORKS – PART 2

This week in Data Mining and Machine Learning, I focused on the core components of modern neural networks: the **Multi-Layer Perceptron (MLP)** and the **Backpropagation algorithm**. My learning activities included reviewing the fundamental differences between the **Perceptron** and the **MLP**, completing a discussion post comparing their training methodologies, and finalizing a programming assignment that involved tuning and analyzing an MLP.

WHAT I DID AND MY REACTIONS

I dedicated the week to understanding how a network with **hidden layers** transcends the limitations of a single-layer perceptron. I first wrote a discussion post contrasting the perceptron's **mistake-driven, discrete updates** with the MLP's **gradient-driven, continuous optimization** via backpropagation. The perceptron converges finitely for **linearly separable** data, but the MLP can solve **non-linear problems** like XOR using differentiable activation functions and loss minimization (Rosenblatt, 1958).

The programming assignment provided practical application. I analyzed the design iterations of a neural network built to map seven-segment display patterns to their ASCII equivalents. This involved evaluating three network topologies: a baseline perceptron, a compact 7-3-7 network, and the final **7-5-7 architecture**. To achieve the final, low-error model, I had to **tune hyperparameters** like the learning rate and momentum, test **different initializations**, and use techniques like **early stopping** and **pattern shuffling** to minimize the required training steps.

My primary reaction was one of appreciation for the elegance of **backpropagation**, which efficiently uses the chain rule to distribute the error signal across all layers. However, this was immediately tempered by realizing the **complexity of non-convex optimization**. Unlike the guaranteed convergence of the perceptron for separable data, backpropagation only guarantees finding a local minimum, not necessarily a global optimum (Rumelhart et al., 1986).

LEARNING AND REALIZATIONS

The most important thing I learned was the direct trade-off between **model capacity** and **training complexity**. The MLP's **nonlinear representation power** allows it to approximate a wide class of functions (the Universal Approximation Theorem), but this power comes at the cost of **richer hyperparameter needs** and the absence of **global convergence guarantees** (Goodfellow et al., 2016).

What caused me to wonder was the critical role of the **hidden layer size** in the programming assignment. The 7-5-7 network converged faster to a low error than the smaller 7-3-7 network, suggesting that the extra capacity eased the optimization surface for that specific task. It surprised me that more capacity, up to a point, could *speed up* convergence rather than slow it down, underscoring that network design is an experimental process.

The most challenging aspect was managing the **hyperparameters** in the programming exercise. Finding the right balance for the **learning rate** and **momentum** felt like a constant battle against oscillation and slow descent, emphasizing that effective training relies heavily on empirical testing and careful tuning, often requiring significant computational resources.

I am gaining practical knowledge in **neural network design** and recognizing myself as a learner who benefits immensely from **practical application** (the programming assignment) paired with **theoretical comparison** (the discussion post). I can apply this knowledge directly in my own experience when deciding between simple linear models and complex deep learning solutions, focusing the choice on the data's inherent complexity and the available computational budget (Goodfellow et al., 2016).

The one important thing I am thinking about is the practical necessity of **regularization** techniques, which were mentioned in the context of deep networks having high capacity and a tendency to overfit. Moving forward, I need to delve deeper into techniques like weight decay and dropout to manage the high variance that comes with the power of MLPs.

References

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Wordcount: 573