

نام و نام خانوادگی: امیرحسین شانقی

شماره دانشجویی: 810899056

نام استاد: سایه میرزایی

عنوان: گزارش تمرین کامپیوتری سوم

بهار 1403

## مقدمه

این گزارش پیاده‌سازی یک شبکه عصبی برای دسته‌بندی ارقام دست‌نویس از مجموعه داده MNIST را تشریح می‌کند. این شبکه عصبی به منظور شناسایی ارقام از 0 تا 9 از طریق یادگیری از یک مجموعه بزرگ از نمونه‌های برچسب‌دار طراحی شده است. معماری شبکه شامل یک لایه ورودی، یک لایه پنهان و یک لایه خروجی است. این گزارش به تشریح انتخاب‌های طراحی، فرضیات، فرمول‌ها و نتایج حاصل از آموزش و ارزیابی مدل می‌پردازد. هدف اصلی دستیابی به دقت بالا در مجموعه تست است.

## بیان مسئله

هدف انجام دسته‌بندی چندکلاسه با استفاده از یک شبکه عصبی با مشخصات زیر است:

1. لایه ورودی با 784 نرون (یکی برای هر پیکسل در تصاویر ورودی  $28 \times 28$ )
2. لایه پنهان با 100 نرون با استفاده از تابع سیگموئید.
3. لایه خروجی با 10 نرون (یکی برای هر کلاس) با استفاده از تابع سافت‌مکس.
4. استفاده از مجموعه داده MNIST با 60000 نمونه آموزشی و 10000 نمونه تست.
5. استفاده از تابع خطای کراس‌انتروپی برای محاسبه خطا.

## دیتاست

مجموعه داده MNIST یک معیار استاندارد در یادگیری ماشین است که شامل 70000 تصویر خاکستری از ارقام دست‌نویس است. این مجموعه به 60000 نمونه آموزشی و 10000 نمونه تست تقسیم می‌شود. هر تصویر  $28 \times 28$  پیکسل دارد که به یک وکتور 784 بُعدی تبدیل می‌شود. مجموعه داده با نرمال‌سازی مقادیر پیکسل‌ها به محدوده  $[0, 1]$  پیش‌پردازش شده است. برچسب‌ها نیز به منظور تسهیل دسته‌بندی چندکلاسه به صورت وان-هات انکد شده‌اند.

## معماری شبکه

معماری شبکه عصبی به صورت زیر تعریف شده است:

- لایه ورودی 784: نرون که به مقادیر پیکسل‌های تصاویر ورودی اختصاص دارد.
- لایه پنهان 100: نرون با تابع سیگموید.
- لایه خروجی 10: نرون با تابع سافت‌مکس که نمایانگر 10 کلاس ارقام (0-9) است.

## فرضیات

- مقادیر پیکسل‌های تصاویر به محدوده  $[0, 1]$  نرمال‌سازی شده‌اند.
- پارامترهای شبکه با استفاده از روش مقداردهی اولیه Xavier برای اطمینان از مقیاس مناسب وزن‌ها مقداردهی اولیه شده‌اند.
- نرخ یادگیری 1.6 تنظیم شده است.
- مدل با استفاده از گرادیان نزولی کامل دسته‌ای آموزش داده می‌شود.

## فرمول ها و پیاده سازی

### 1. مقداردهی اولیه وزن ها

وزن ها با استفاده از روش مقداردهی اولیه Xavier مقداردهی می شوند :

$$W \sim N(0, 1/n)$$

که در آن  $n$  تعداد نرون ها در لایه قبلی است.

### 2-توابع فعالیت

Sigmoid Activation:

$$\sigma(z) = 1 / (1 + e^{-z})$$

Softmax Activation:

$$\text{softmax}(z_i) = e^{z_i} / \sum_j e^{z_j}$$

### Forward Propagation-3

-لایه مخفی:

$$Z1 = X * W1 + b1$$

$$A1 = \sigma(Z1)$$

-لایه خروجی:

$$Z2 = A1 * W2 + b2$$

$$A2 = \text{softmax}(Z2)$$

### 4-تابع خطا

-تابع خطای کراس انتروپی:

$$L(Y, A2) = -\frac{1}{m} \sum_{i=1}^m \sum_{j=1}^k Y_{ij} \log(A2_{ij})$$

که در آن  $m$  تعداد نمونه ها و  $k$  تعداد کلاس ها است.

### Backpropagation-5

-گرادیان های لایه خروجی:

$$dZ2 = A2 - Y$$

$$dW2 = \frac{1}{m} \cdot A1^T \cdot dZ2$$

$$db2 = \frac{1}{m} \sum dZ2$$

-گرادیان های لایه مخفی:

$$\begin{aligned}dA1 &= dZ2 \cdot W2^T \\dZ1 &= dA1 \cdot \sigma'(Z1) \\dW1 &= \frac{1}{m} X^T \cdot dZ1 \\db1 &= \frac{1}{m} \sum dZ1\end{aligned}$$

6-بروزرسانی پارامترها

-گرادیان دیسنت:

$$\begin{aligned}W1 &= W1 - \eta dW1 \\b1 &= b1 - \eta db1 \\W2 &= W2 - \eta dW2 \\b2 &= b2 - \eta db2\end{aligned}$$

که در آن  $\eta$  نرخ یادگیری است.

فرآیند آموزش

مدل در طول 50 دوره (epoch) با نرخ یادگیری 0.1 آموزش داده می شود. حلقه آموزش شامل مراحل زیر است:

(1) **Forward Propagation**: محاسبه فعالیت های لایه های پنهان و خروجی.

(2) محاسبه خطا: محاسبه خطای کراس انتروپی برای اندازه گیری خطا بین پیش بینی ها و برچسب های واقعی.

(3) **Backpropagation**: محاسبه گرادیان های خطا نسبت به هر وزن و بایاس.

(4) به روزرسانی پارامترها: به روزرسانی وزن ها و بایاس ها با استفاده از گرادیان نزولی.

## نتایج

پس از آموزش شبکه عصبی به مدت 100 دوره و با نرخ یادگیری 1.6، مدل به دقت 90.22٪ در مجموعه تست دست می‌یابد. این یک بهبود قابل توجه نسبت به دقت اولیه است و نشان می‌دهد که تنظیمات جدید منجر به عملکرد بهتری شده است. برای مقایسه، یک پیاده‌سازی شبکه عصبی با استفاده از TensorFlow با همان معماری به دقت 97.28٪ دست می‌یابد.

```
Epoch 1/100, Loss: 2.4504393709811434
Epoch 8/100, Loss: 1.7719853101746554
Epoch 15/100, Loss: 1.1776729908183021
Epoch 22/100, Loss: 0.925153318650677
Epoch 29/100, Loss: 0.7375225789630429
Epoch 36/100, Loss: 0.6481489155949324
Epoch 43/100, Loss: 0.5371172326222754
Epoch 50/100, Loss: 0.5206745466747266
Epoch 57/100, Loss: 0.4560617990472848
Epoch 64/100, Loss: 0.41128787881966994
Epoch 71/100, Loss: 0.3863139514802078
Epoch 78/100, Loss: 0.3698284780939989
Epoch 85/100, Loss: 0.357207156428861
Epoch 92/100, Loss: 0.34702322451289874
Epoch 99/100, Loss: 0.33847161524275926
Manual Implementation Accuracy: 90.22%
```

