

Dokumentacija za paralelnu implementaciju neuronske mreže (Multilayer Perceptron) za klasifikaciju podataka

Aleksandar Vig

1. Uvod

Ovaj projekat implementira **Multilayer Perceptron (MLP)** za klasifikacione zadatke koristeći sekvencijalne i paralelne tehnike programiranja. Implementacija podržava paralelizaciju pomoću **MPI** i meri poboljšanje performansi prilikom rada na više procesa. Dataset koji se koristi za treniranje i testiranje je poznati **MNIST dataset**.

2. Ulaz u Program

- **Broj skrivenih slojeva i njihove dimenzije:** Definiše se u kodu kao niz celih brojeva.
- **Dataset:** MNIST dataset, prethodno procesiran u grayscale formatu i normalizovan.
- **Broj procesa ili niti:** Specificira se tokom izvršavanja za paralelno treniranje.

3. Funkcionalnosti Programa

- Implementira **paralelno** treniranje neuronske mreže pomoću podeljenih podataka na više procesa.
- Koristi **ReLU aktivacionu funkciju** za skrivene slojeve i **Softmax aktivacionu funkciju** za izlazni sloj.
- Koristi **gradient descent** kao algoritam optimizacije.
- Sinhronizuje težine između procesa nakon svake epohe pomoću **MPI_Allreduce**.
- Izračunava i upoređuje vreme treniranja za sekvencijalne i paralelne implementacije.

4. Ključne Komponente Implementacije

Sekvencijalna Implementacija

1. Ceo dataset se učitava i obrađuje na jednom procesu.
2. Implementirani su forward i backward prolazi za treniranje.
3. Vreme treniranja se meri za jednu ili više epoha.

Paralelna Implementacija

1. Dataset se ravnomerno deli između MPI procesa.
2. Svaki proces obavlja forward i backward prolaze na svom delu podataka.

3. Gradijenti i težine se sinhronizuju između svih procesa pomoću MPI.
4. Ukupno vreme treniranja za jednu ili više epoha se meri.

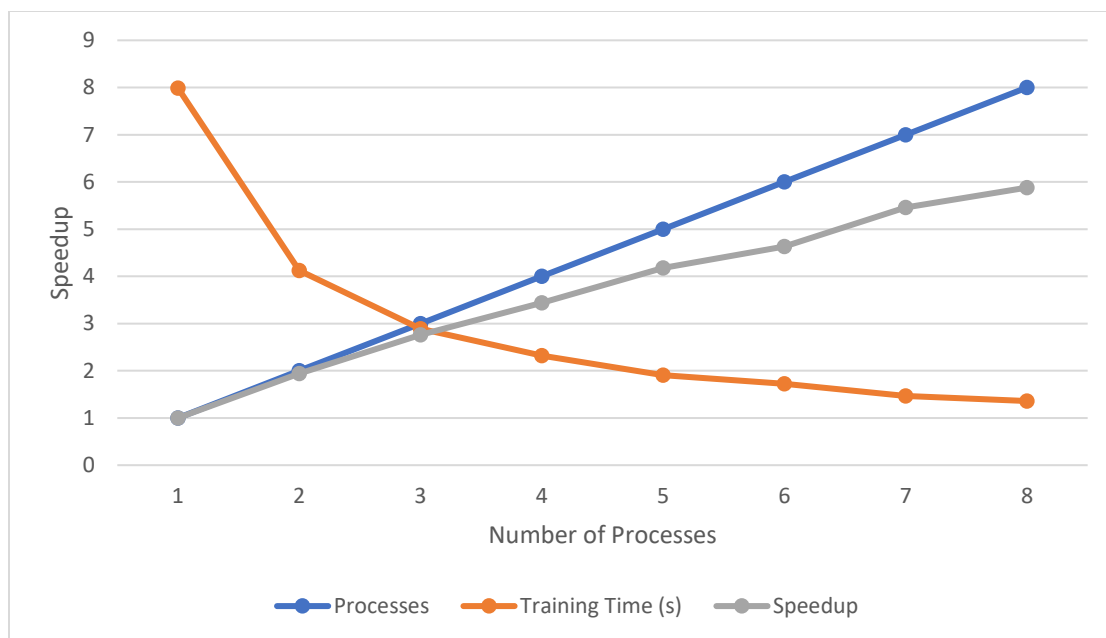
5. Analiza rezultata

- **Konfiguracija Neuronske Mreže:**
 - Ulazni Sloj: 784 neurona (28x28 piksela za MNIST slike).
 - Skriveni Slojevi: Dva sloja sa 128 i 64 neurona.
 - Izlazni Sloj: 10 neurona (10 klasa za klasifikaciju).
- **Veličina Dataseta:** 10.000 uzoraka za treniranje.
- **Broj Epoha:** 1.
- **Learning Rate:** 0.01.

Implementacija	Vreme treniranja [s]	Ubrzanje
Sekvencijalni Program	7.988	1.00
Paralelni Program (2 procesa)	4.124	1.94
Paralelni Program (3 procesa)	2.893	2.76
Paralelni Program (4 procesa)	2.319	3.44
Paralelni Program (5 procesa)	1.910	4.18
Paralelni Program (6 procesa)	1.726	4.63
Paralelni Program (7 procesa)	1.464	5.46
Paralelni Program (8 procesa)	1.359	5.88

Vreme treniranja i ubrzanje

Kako se broj procesa povećava, vreme treniranja značajno opada. Međutim, ubrzanje dostiže opadajuće povraćaje (**diminishing returns**) zbog overhead-a komunikacije između procesa.



Grafički prikaz ubranja i broja procesa