

# Training Neural Nets with Congestion Control

Nume: Stoian Robert Alin

Grupa: 464

## 1. Articole existente si contextul "Arheologic"

Proiectul isi propune sa testeze o ipoteza inedita: aplicarea algoritmului AIMD (Additive Increase Multiplicative Decrease), fundamental pentru stabilitatea internetului (protocolul TCP), in antrenarea retelelor neuronale.

- **Referinta Arheologica:** Algoritmul AIMD (Chiu & Jain, 1989) garanteaza convergenta si echitatea in retelele de calculatoare prin cresterea liniara a ratei de transmisie si scaderea multiplicativa la congestie
- **Aplicabilitate AI:** Vom adapta acest mecanism pentru a ajusta dinamic **Rata de Invatare (Learning Rate)**. Consideram "cresterea loss-ului" ca fiind echivalentul unei "congestii", declansand o scadere brusca a ratei de invatare pentru a stabiliza antrenarea

## 2. Datele folosite si Analiza Exploratorie (EDA)

Vom utiliza setul de date **CIFAR-10**, un benchmark standard în Computer Vision.

- **Descriere:** Setul contine 60.000 de imagini color (32x32 pixeli), impartite echilibrat in 10 clase (avioane, masini, animale, etc.).
- **Analiza:** Imaginele au o rezolutie scazuta, ceea ce face task-ul dificil pentru modele simple, necesitand straturi convolutionale. Mai jos sunt prezentate esantioane aleatoare din setul de antrenare:



## 3. Ce modele am folosit si necesarul de calcul

Pentru a izola efectul algoritmului AIMD, folosim o arhitectura standard, usor de reprodus:

- Model: O retea neuronală convolutională , Convolutional Neural Network (CNN) cu 3 straturi de convoluție (Conv2D), straturi de MaxPooling pentru reducerea dimensiunii și straturi Dense la final pentru clasificare

- Resurse: Antrenarea necesita accelerare GPU. Folosim mediul Google Colab (T4 GPU), unde un ciclu de antrenare pentru baseline dureaza sub 2 minute, ceea ce permite iterarea rapida a experimentelor

## 4.Ce metode de evaluare si comparatie vei folosi

**Metodologia de evaluare este comparativa. Vom antrena acelasi model CNN in trei scenarii:**

1. **Baseline (Realizat): Learning Rate fix/standard.**
2. **Benchmark: Scheduler modern (ReduceLROnPlateau).**
3. **Experimental: Scheduler-ul propus AIMD.**

**Rezultate Intermediare (Baseline):** Am implementat si rulat cu succes modelul Baseline. Asa cum se observa in graficele de mai jos, modelul converge corect, atingand o acuratete satisfacatoare in primele epoci, fara semne majore de overfitting. Aceasta este linia de baza pe care algoritmul AIMD trebuie sa o depaseasca.

