

Q

# Neurol Network”

## Descent algorithm with constant rate

### Algorithm

Require  $f\colon \mathbf{R}^n \rightarrow \mathbf{R}, x_0 \in \mathbf{R}^n$  ( initial point)

- $\eta = 0.1$
- $k \leftarrow 0$
- continue = true
- **While** continue
  - $d_k = -\nabla f(x_k)$
  - $x_{k+1} \leftarrow x_k + \eta d_k$
  - $k \leftarrow k + 1$
  - continue = stop\_function( $\nabla f_k, x_k, x_{k+1}, f_k, f_{k+1}, \text{AbsTol}, \text{RelTol}, \varepsilon$ )
- **EndWhile**

### Application : Simple Linear Regression

► Code

```
A = [5.0 0.0; 0.0 45.0]
b = [-10.0, -45.0]
n= 5
X*x0 = [10.90797688063037, 20.0, 22.5, 25.0, 34.09202311936963]
x0 = [22.5, 2.5]
f(x0) = 440.5
xsol = [18.4, -0.20000000000000018]
xsol = [15.119999999999997, 1.9600000000000004]
xsol = [12.495999999999997, 0.2319999999999954]
xsol = [10.396799999999997, 1.6144000000000003]
xsol = [8.717439999999998, 0.508479999999998]
xsol = [7.373951999999998, 1.3932160000000002]
xsol = [6.299161599999998, 0.6854271999999997]
```

## With the Flux Package

Find the same results with **Flux** (see <https://fluxml.ai/Flux.jl/stable/>)

► Code

```
x_train = Float32[-4.6368093 -1.0 0.0 1.0 4.6368093]
y_train = Float32[-2.6368093 1.0 2.0 3.0 6.6368093]
model.weight = Float32[2.5;;]
model.bias = Float32[22.5]
model(x_train) = Float32[10.907976 20.0 22.5 25.0 34.092026]
loss(model, x_train, y_train) = 440.5
Float32[183.46121 361.0 420.25 484.0 753.7889]
data = Tuple{Matrix{Float32}, Matrix{Float32}}{([-4.6368093 -1.0 0.0 1.0 4.6368093], [-2.6368093 1.0 2.0 3.0 6.6368093])}
n= 5
data = (Float32[-4.6368093 -1.0 0.0 1.0 4.6368093], Float32[-2.6368093 1.0 2.0 3.0 6.6368093])
Flux.setup(rule, model) = (weight = Leaf(Descent(0.1), nothing), bias = Leaf(Descent(0.1), nothing), σ = (dLdm = (weight = Float32[27.000004;;], bias = Float32[41.0], σ = nothing)
model.weight = Float32[-0.20000052;;]
model.bias = Float32[18.4]
model.weight = Float32[1.9600008;;]
model.bias = Float32[15.12]
model.weight = Float32[0.23199975;;]
model.bias = Float32[12.496]
model.weight = Float32[1.6144003;;]
model.bias = Float32[10.3968]
model.weight = Float32[0.50847995;;]
model.bias = Float32[8.717441]
model.weight = Float32[1.3932161;;]
model.bias = Float32[7.3739524]
model.weight = Float32[0.685427;;]
model.bias = Float32[6.299162]
Xsol = [22.5 2.5; 18.4 -0.20000000000000018; 15.119999999999997 1.9600000000000004; 12.495999999999997 0.2319999999999954; 10.396799999999997 1.6144000000000003; 8.717439999999998 0.508479999999998; 7.373951999999998 1.3932160000000002; 6.299161599999998 0.6854271999999997]
XsolLNN = [22.5 2.5; 18.399999618530273 -0.20000052452087402; 15.119999885559082 1.96000075340271; 12.496000000000000 0.2319999999999954; 10.396799999999997 1.6144000000000003; 8.717439999999998 0.508479999999998; 7.373951999999998 1.3932160000000002; 6.299161599999998 0.6854271999999997]
Xsol - XsolLNN
```

```
8×2 Matrix{Float64}:
 0.0      0.0
 3.8147e-7  5.24521e-7
 1.14441e-7 -7.53403e-7
-2.89917e-7  2.45094e-7
-4.11987e-8 -2.67601e-7
-6.05164e-7  4.71878e-8
-3.88763e-7 -1.33118e-7
-3.11011e-7  1.89941e-7
```