Optimizer Cheat Sheet – Definitions, Formulas & Examples

Batch Gradient Descent

Definition: Uses the entire training dataset to compute gradients and update weights.

Formula: w = w - eta * (1/m) * gradL(w, x)

Examples:

1. If w=0.5, η=0.1, and ∇L(w)=4 (full batch), then w_new = 0.5 - 0.1*4 = 0.1

2. With w=2, η=0.05, ∇L(w)=6 → w_new = 2 - 0.05*6 = 1.7

3. If w=1.0, η=0.2, ∇L(w)=3 → w_new = 1.0 - 0.2*3 = 0.4

Stochastic Gradient Descent (SGD)

Definition: Updates weights using one sample at a time, introducing more noise but faster updates.

Formula: w = w - eta * gradL(w, x)

Examples:

1. w=0.5, η=0.1, ∇L=5 → w_new = 0.5 - 0.1*5 = 0.0

2. w=1.2, η=0.05, ∇L=2 → w_new = 1.2 - 0.05*2 = 1.1

3. w=3.0, η=0.01, ∇L=10 → w_new = 3.0 - 0.01*10 = 2.9

Mini-Batch Gradient Descent

Definition: Uses small random batches of data to update weights; combines stability and speed.

Formula: w = w - eta * (1/k) * gradL(w, x) over k samples

Examples:

1. w=1.0, η=0.1, avg ∇L=4 over mini-batch → w_new = 1.0 - 0.1*4 = 0.6

2. w=2.5, η=0.01, avg ∇L=5 → w_new = 2.5 - 0.01*5 = 2.45

3. w=0.8, η=0.2, avg ∇L=1.5 → w_new = 0.8 - 0.2*1.5 = 0.5

AdaGrad

Definition: Adapts learning rate per parameter using cumulative squared gradients.

Formula: eta = eta / sqrt(G + epsilon)

Examples:

1. η=0.1, G=25 → η_scaled = 0.1 / sqrt(25) = 0.02

2. η=0.1, G=4 → η_scaled = 0.1 / sqrt(4) = 0.05

3. η=0.01, G=1 → η_scaled = 0.01 / sqrt(1) = 0.01

AdaDelta

Definition: Improves AdaGrad by using a moving window of gradient history instead of accumulating all past gradients.

Formula: Deltaw = - RMS(Deltaw) / RMS(g) * g

Examples:

1. Assume RMS(Δw)=1, RMS(g)=2, g=4 → Δw = -1/2 * 4 = -2

2. RMS(Δw)=0.5, RMS(g)=1, g=2 → Δw = -0.5/1 * 2 = -1

3. RMS(Δw)=2, RMS(g)=2, g=1 → Δw = -2/2 * 1 = -1

RMSProp

Definition: Uses exponential moving average of squared gradients to adapt learning rate.

Formula: E[g^2]_t = beta * E[g^2]_(t-1) + (1 - beta) * g^2

Examples:

1. β=0.9, $E[g^2]$=0, g=4 → $E[g^2]$_new = 0.1*16 = 1.6

2. β=0.9, $E[g^2]$=1, g=3 → $E[g^2]$_new = 0.9*1 + 0.1*9 = 0.9 + 0.9 = 1.8

3. β=0.99, $E[g^2]$=2, g=2 → $E[g^2]$_new = 0.99*2 + 0.01*4 = 1.98 + 0.04 = 2.02

Adam

Definition: Combines momentum and RMSProp, using bias-corrected first and second moments.

Formula: m = m / (1 - beta^t), v = v / (1 - beta^t), w = w - eta * m / (sqrt(v) + epsilon)

Examples:

1. m=0.5, β■=0.9, t=1 → m■ = 0.5 / (1 - 0.9) = 5.0

2. v=0.25, β■=0.999, t=1 → v■ = 0.25 / (1 - 0.999) = 250

3. w=1, η=0.01, m■=5, v■=250 → w_new = 1 - 0.01 * 5 / (sqrt(250)) ≈ 0.99

Momentum

Definition: Adds a velocity term to accelerate updates in consistent gradient directions.

Formula: v = * v + eta * gradL, w = w - v

Examples:

1. $v=0.1$, $\gamma=0.9$, $\eta=0.01$, $\nabla L=5$ → v_new = 0.09 + 0.05 = 0.14

2. $v=0.2$, $\gamma=0.8$, $\eta=0.1$, $\nabla L=3$ → v_new = 0.16 + 0.3 = 0.46

3. $v=0$, $\gamma=0.9$, $\eta=0.1$, $\nabla L=2$ → v_new = 0 + 0.2 = 0.2

Nesterov Accelerated Gradient (NAG)

Definition: Improves momentum by computing gradient at the estimated future position.

Formula: v = * v + eta * gradL(w - * v), w = w - v

Examples:

1. $v=0.2$, $\gamma=0.9$, $\eta=0.1$, $\nabla L=3$ → v_new = 0.18 + 0.3 = 0.48

2. $v=0.1$, $\gamma=0.8$, $\eta=0.05$, $\nabla L=4$ → v_new = 0.08 + 0.2 = 0.28

3. $v=0$, $\gamma=0.9$, $\eta=0.1$, $\nabla L=2$ → v_new = 0 + 0.2 = 0.2