In online learning, tuning the learning rate is essential but tricky. We present LLR, a new algorithm for setting the learning rate that performs as well as the best learning rate in hindsight, and is as fast as the standard exponential weights algorithm.

The exponential weights algorithm:

The Hedge or exponential weights algorithm with learning rate $\eta$:

$$ w_t^k := \frac{e^{-\eta t_k}}{\sum_{k=1}^{K} e^{-\eta t_k}} \text{ where } t_k := \sum_{i=1}^{t} \ell_i. $$

The worst-case safe tuning $\eta = \sqrt{\frac{3 \ln K}{T}}$ results in:

$$ R_T \leq \sqrt{T/2 \ln K}. $$

Prior work on adaptive tuning:

Second-order Bounds [Cesa-Bianchi, Mansour, Stoltz ML’07]:

$$ R_T \leq O\left(\sqrt{V_T \ln K}\right), \quad (1) $$

where $V_T = \sum_{i=1}^{T} v_i$ and $v_i$ is variance of loss $\ell_i$ when $k \sim w_i$.

FlipFlop [De Rooij and us, JMLR’14]:

Satisfies (1) and is never worse than Follow-the-Leader ($\eta = \infty$):

$$ R_T \leq C R_T^\eta. \quad (2) $$

Big question:

Can we compete with all $\eta$?

Intermediate learning rates can be crucial:

- Hedge($\eta$)
- Worst-case bound and $\eta$
- AdaHedge
- FlipFlop
- LLR and $\eta_{ab}$

The Learning the Learning Rate (LLR) algorithm satisfies (1) and (2) and in addition:

$$ R_T \leq \tilde{O}(\ln K \ln \frac{1}{\eta} \eta^\ast) \quad \text{for all } \eta \in [\eta_{ab}, 1]. $$

- Trumps $\sqrt{T}$ additive overhead.
- In above range $\ln \frac{1}{\eta} = O(\ln T)$.
- So $R_T = O(T^a)$ whenever $\min_{\eta} R_T^\eta = O(T^a)$ for some $a > 0$.

LLR algorithm in a nutshell:

- Maintains a finite grid $\eta^1, \ldots, \eta^{\text{max}}, \eta_{ab}$
- Cycles over the grid. For each $\eta^i$:
  - Play the $\eta^i$ Hedge weights
  - Evaluate $\eta^i$ by its mixability gap
  - Until its budget doubled
- Adds next lower grid point on demand

\[ \eta_t \text{ not monotonic in } t \]

Resource consumption: same as Hedge (trick!)

LLR algorithm key ideas:

- Basic idea: budgeted timesharing. Cycle through $\eta$s from an exponentially spaced grid to keep their regrets approximately equal. Then our regret is:

$$ R_T \leq \#\text{gridpoints} \times R_T^{\text{best } \eta}. $$

- Refinement: Use prior distribution on $\eta$s to get

$$ R_T \leq \frac{R_T^{\text{best } \eta}}{\text{prior of best } \eta} $$

independent of the number of grid points.

- Problem: does not quite work, because regret $R_T^{\eta}$ for fixed $\eta$ is not monotonically increasing with time $T$.

- Solution: cumulative mixability gap $\Delta_T^\eta = \sum_{i=1}^{T} \delta_i^\eta$ where

$$ \delta_i^\eta = \text{Hedge loss} - \text{max loss} = w_i^\eta \ell_i - \frac{1}{\eta} \ln \sum_{k} w_i^k e^{-\eta \ell_i} $$

is a good proxy (tightest monotonic lower bound on $R_T^{\eta}$).

- Building block of independent interest:

$$ \delta_i^{2\eta} \leq 6K \delta_i^\eta. $$

Fundamental model for learning: Hedge setting:

- $K$ experts
- In round $t = 1, 2, \ldots$
  - Learner plays distribution $w_t = (w_t^1, \ldots, w_t^K)$ on experts
  - Adversary reveals expert losses $\ell_t = (\ell_t^1, \ldots, \ell_t^K) \in \{0,1\}^K$
  - Learner incurs loss $w_t^i \ell_i$

Evaluation criterion is the regret w.r.t. the best expert in hindsight:

$$ R_T = \sum_{t=1}^{T} w_t^i \ell_i - \min_{k} \sum_{t=1}^{T} \ell_t^k $$

LLR algorithm key ideas:

- Basic idea: budgeted timesharing. Cycle through $\eta$s from an exponentially spaced grid to keep their regrets approximately equal. Then our regret is:

$$ R_T \leq \#\text{gridpoints} \times R_T^{\text{best } \eta}. $$

- Refinement: Use prior distribution on $\eta$s to get

$$ R_T \leq \frac{R_T^{\text{best } \eta}}{\text{prior of best } \eta} $$

independent of the number of grid points.

- Problem: does not quite work, because regret $R_T^{\eta}$ for fixed $\eta$ is not monotonically increasing with time $T$.

- Solution: cumulative mixability gap $\Delta_T^\eta = \sum_{i=1}^{T} \delta_i^\eta$ where

$$ \delta_i^\eta = \text{Hedge loss} - \text{max loss} = w_i^\eta \ell_i - \frac{1}{\eta} \ln \sum_{k} w_i^k e^{-\eta \ell_i} $$

is a good proxy (tightest monotonic lower bound on $R_T^{\eta}$).

- Building block of independent interest:

$$ \delta_i^{2\eta} \leq 6K \delta_i^\eta. $$

Fundamental model for learning: Hedge setting:

- $K$ experts
- In round $t = 1, 2, \ldots$
  - Learner plays distribution $w_t = (w_t^1, \ldots, w_t^K)$ on experts
  - Adversary reveals expert losses $\ell_t = (\ell_t^1, \ldots, \ell_t^K) \in \{0,1\}^K$
  - Learner incurs loss $w_t^i \ell_i$

Evaluation criterion is the regret w.r.t. the best expert in hindsight:

$$ R_T = \sum_{t=1}^{T} w_t^i \ell_i - \min_{k} \sum_{t=1}^{T} \ell_t^k $$

Prior work on adaptive tuning:

Second-order Bounds [Cesa-Bianchi, Mansour, Stoltz ML’07]:

$$ R_T \leq O\left(\sqrt{V_T \ln K}\right), \quad (1) $$

where $V_T = \sum_{i=1}^{T} v_i$ and $v_i$ is variance of loss $\ell_i$ when $k \sim w_i$.

FlipFlop [De Rooij and us, JMLR’14]:

Satisfies (1) and is never worse than Follow-the-Leader ($\eta = \infty$):

$$ R_T \leq C R_T^\eta. \quad (2) $$

Big question:

Can we compete with all $\eta$?

Intermediate learning rates can be crucial:

- Hedge($\eta$)
- Worst-case bound and $\eta$
- AdaHedge
- FlipFlop
- LLR and $\eta_{ab}$

The Learning the Learning Rate (LLR) algorithm satisfies (1) and (2) and in addition:

$$ R_T \leq \tilde{O}(\ln K \ln \frac{1}{\eta} \eta^\ast) \quad \text{for all } \eta \in [\eta_{ab}, 1]. $$

- Trumps $\sqrt{T}$ additive overhead.
- In above range $\ln \frac{1}{\eta} = O(\ln T)$.
- So $R_T = O(T^a)$ whenever $\min_{\eta} R_T^\eta = O(T^a)$ for some $a > 0$.

LLR algorithm in a nutshell:

- Maintains a finite grid $\eta^1, \ldots, \eta^{\text{max}}, \eta_{ab}$
- Cycles over the grid. For each $\eta^i$:
  - Play the $\eta^i$ Hedge weights
  - Evaluate $\eta^i$ by its mixability gap
  - Until its budget doubled
- Adds next lower grid point on demand

$$ \eta_t \text{ not monotonic in } t \]

Resource consumption: same as Hedge (trick!)