Online Linear Learning Tricks with Vowpal Wabbit

Contents

- Batch vs Online Learning
- Learning Rate Tricks
  - Invariant
  - Adaptive
  - Normalized

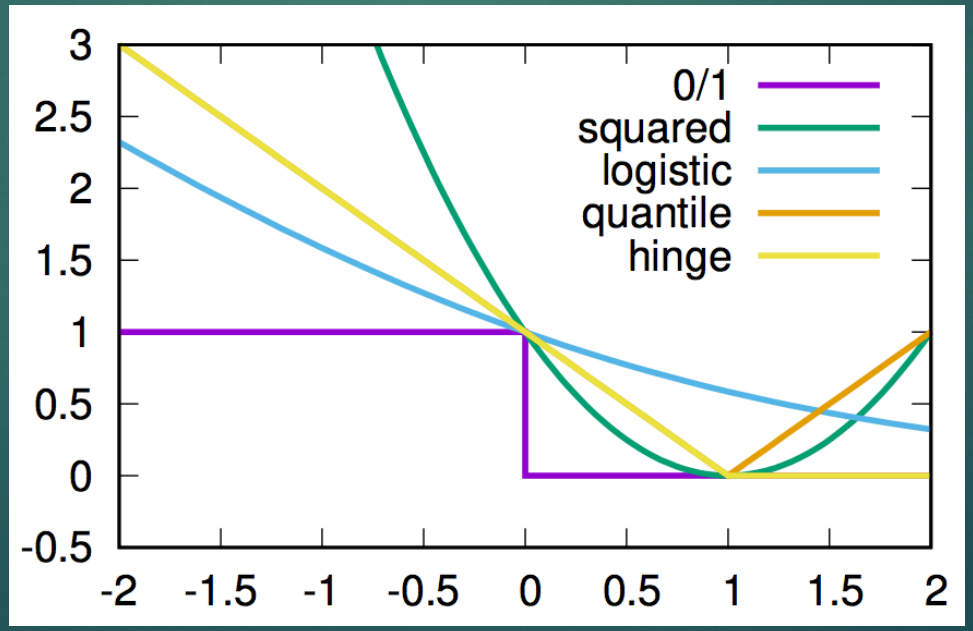
Batch vs Online Learning

- Batch Learning
  - Input: {x, y} where x $\in R^n$ and y $\in$R
  - Learn w $\in R^n$ s.t. error is minimized

- Online Learning <- **Vowpal Wabbit specializes in this!**
  - for i..n
  - repeat:
    - Input: $\{x_i, y_i\}$
    - Update w s.t. $\hat{y}_w$ is closer to y

Batch vs Online Learning

- Define a loss function $L(\widehat{y_w}, y)$
- Update via SGD $w_i = w_i - \eta \frac{dL(\widehat{y_w}(x), y_i)}{dw_i}$



Learning Rate Tricks

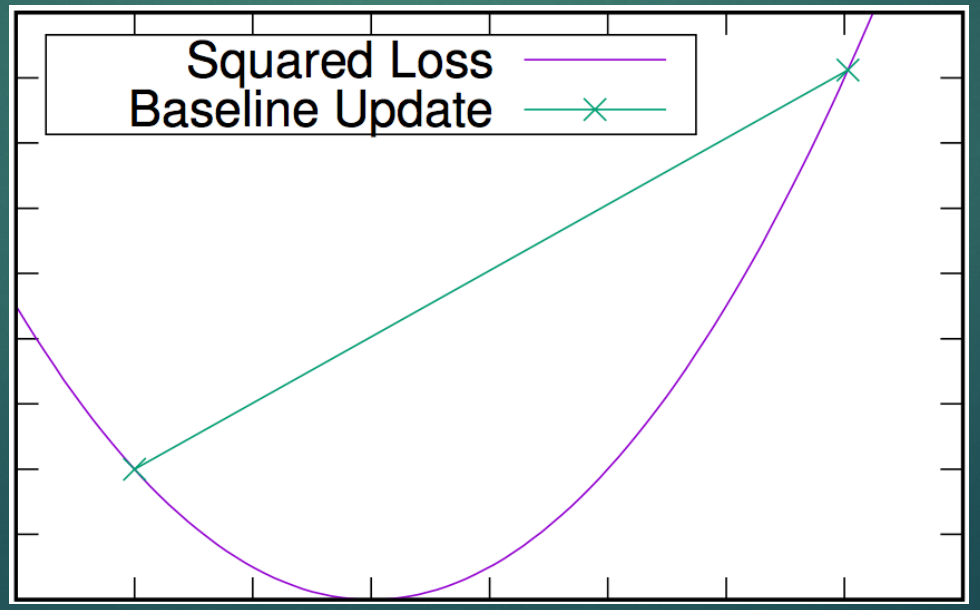
- Vowpal Wabbit's learning rate function does a few things by default
- They are parameters that help the learner convergence
  - --invariant, use safe / importance aware updates
  - --adaptive, use adaptive, individual learning rates
  - --normalized, use per feature normalized updates

Learning Rate Tricks - Invariant

- Often in classification problems, one choice is more expensive than the other
  - For instance, in spam classification, it's a lot more expensive to classify a non-spam email as spam than to classify a spam email as non-spam
- We want to give an importance weight, I, to the classes
- How can we update the naïve SGD equation?
  - $w_i = w_i - \eta \frac{dL(\widehat{y_w}(x), y_i)}{dw_i}$

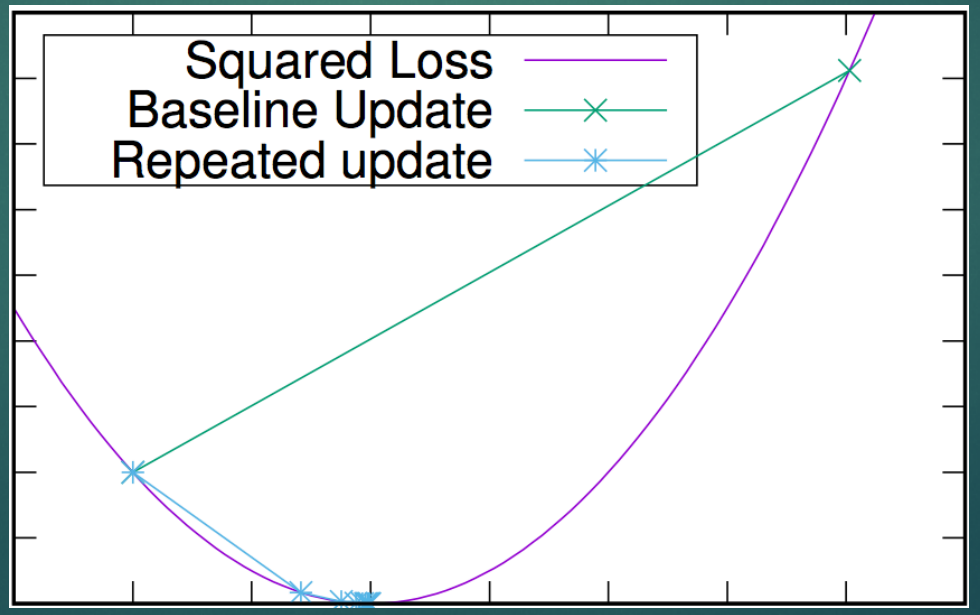
Learning Rate Tricks - Invariant

- Baseline – Let's scale the gradient by I
  - $w_i = w_i - \eta\, I \frac{dL(\widehat{y_w}(x), y_i)}{dw_i}$
- Works in batch GD, but not online!



Learning Rate Tricks - Invariant

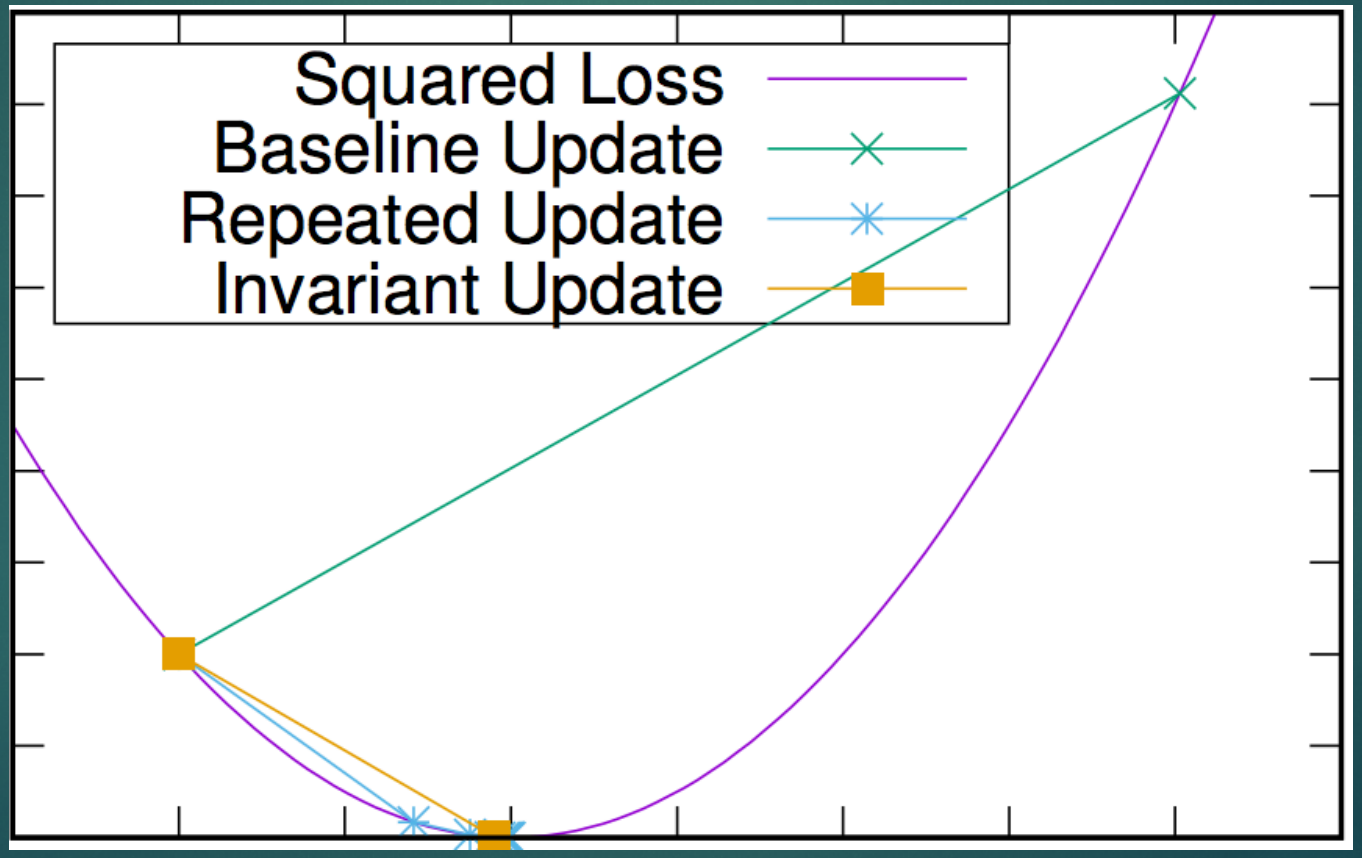
- Better solution – Let's just update I times
  - $w_i = w_i - \eta \frac{dL(\widehat{y_w}(x), y_i)}{dw_i}$, I times
- Converges a bit better



Learning Rate Tricks - Invariant

- Invariant solution – Learn to derive the appropriate step size so that we can update in one step
  - $w_i = w_i - S(\eta\ I)\frac{dL(\widehat{y_w}(x), y_i)}{dw_i}$
  - $s(h+1) = s(h) + \eta\ \frac{\partial l}{\partial p}$, where $p = (w_t - s(h)x)^T x$
  - See convergence proof in *Online Importance Weight Aware Updates, Karampatziakis et al, 2010*

Learning Rate Tricks - Invariant



Learning Rate Tricks - Adaptive

- In SGD, we need the learning rate η to decay to converge
  - Naïve solution is to decay η parametrically, $\eta_t = \frac{1}{t^{0.5}}$ or $\frac{1}{t}$
- Adopts a partial derivative update strategy
  - $w_{it} = w_{it} - \eta \frac{g_{it}}{\sqrt{\sum_{t'}^{t} g_{it}^2}}$
  - Where g_{it} is a partial derivative
- Common features stabilize quickly; Rare features have bigger updates
- Very similar to RMSProp that's currently popular in deep learning

Learning Rate Tricks - Normalized

- The partial derivative $g_{it} = 2(\widehat{y_w}(x) - y)x_i$ can be rewritten
  - $w_i = w_i - Cx_i$
- If a feature i has twice the magnitude of feature j, w_i needs to have half the mangitude of w_j
- In batch GD, we can simply standardize the features via
  - $x_i = \frac{x_i - \mu}{\sigma}$
- But not in Online GD…

Learning Rate Tricks - Normalized

- As per *Normalized Online Learning, Ross et al, UAI 2013.*, the scale free update function
  - $w_i = w_i - \eta \sqrt{\frac{t}{N}} \frac{1}{s^2} \frac{\partial\, l(\hat{y},y)}{\partial w_i}$
  - Where $s_i = \max_t x_{it}$
  - And $N = N + \sum_i^t \frac{x_i^2}{s_i^2}$

References

- Vowpal Wabbit Command Line Arguments, https://github.com/JohnLangford/vowpal_wabbit/wiki/Command-line-arguments
- John Langford, Machine Learning the Future, http://hunch.net/~mltf/online_linear.pdf