This appendix is organized as follows:

1. In Section A we prove some technical Lemmas used in our main results.
2. In Section B we prove Theorem 1.
3. In Section C we prove Theorem 2, and also provide a doubling-trick based algorithm that achieves the optimal log factors in its regret bound.
4. In Section D we provide details about our empirical evaluation.

A. Technical Lemmas

We compute a useful Fenchel conjugate below:

Lemma 3. *Let $f(x) = a\exp(bx)$ for $a \geq 0$ and $b \geq 0$. Then $f^\star(y) = \frac{y}{b}\left(\log\left(\frac{y}{ab}\right) - 1\right)$ for all $y \geq 0$.*

Proof. We want to maximize

$$yx - a\exp(bx)$$

as a function of x. Differentiating, we have $y - ab\exp(bx) = 0$, so that $x = \frac{1}{b}\log\left(\frac{y}{ab}\right)$ (where we've used our assumption about non-negativity of all variables). Then we simply substitute this value in to conclude the Lemma. □

Next, we have a useful optimization solution:

Lemma 4. *Suppose A, B, C, D are non-negative constants. Then*

$$\inf_{x\in[0,1/2]} \frac{A}{x}\left[\log\left(\frac{B}{x}\right) - C\right] + Dx \leq 2\max\left[\sqrt{AD\max\left[\log\left(\frac{B\sqrt{D}}{\sqrt{A}}\right) - C, 1\right]}, 2A\max\left[\log\left(\frac{B\sqrt{4A^2 + D}}{\sqrt{A}}\right) - C, 1\right]\right]$$

Proof. We will just guess a value for x:

$$x = \frac{\sqrt{A}}{\sqrt{D}}\sqrt{\max\left[\log\left(\frac{B\sqrt{D}}{\sqrt{A}}\right) - C, 1\right]}$$

Suppose that this quantity is in $[0, 1/2]$ for now. Then we have

$$\log\left(\frac{B}{x}\right) \leq \log\left(\frac{B\sqrt{D}}{\sqrt{A}}\right)$$

so that

$$\begin{aligned}\frac{A}{x}\left[\log\left(\frac{B}{x}\right) - C\right] &\leq \frac{A}{x}\left[\log\left(\frac{B\sqrt{D}}{\sqrt{A}}\right) - C\right] \\ &\leq \sqrt{AD\left(\log\left(\frac{B\sqrt{D}}{\sqrt{A}}\right) - C\right)}\end{aligned}$$

Thus we have:

$$\frac{A}{x}\left[\log\left(\frac{B}{x}\right) - C\right] + Dx \leq 2\sqrt{AD\max\left[\log\left(\frac{B\sqrt{D}}{\sqrt{A}}\right) - C, 1\right]}$$

Now suppose instead that our guess is outside $[0, 1/2]$. Then we must have

$$4A \max\left[\log\left(\frac{B\sqrt{D}}{\sqrt{A}}\right) - C, 1\right] \geq D$$

and also

$$2\sqrt{\max\left[\log\left(\frac{B\sqrt{D}}{\sqrt{A}}\right) - C, 1\right]} \geq \frac{\sqrt{D}}{\sqrt{A}}$$

So now with $x = 1/2$ we obtain:

$$\begin{aligned}
\frac{A}{x}\left[\log\left(\frac{B}{x}\right) - C\right] + Dx &\leq 2A(\log(2B) - C) + 2A\max\left[\log\left(\frac{B\sqrt{D}}{\sqrt{A}}\right) - C, 1\right] \\
&\leq 2A(\log(2B) - C) + 2A\max\left[\log\left(\frac{B\sqrt{D}}{\sqrt{A}}\right) - C, 1\right] \\
&\leq 4A\max\left[\log\left(\frac{B\sqrt{4A^2 + D}}{\sqrt{A}}\right) - C, 1\right]
\end{aligned}$$

□

A.1. Proof of Lemma 2

Proof. Recall that $\text{Wealth}_T(v)$ is the wealth of an algorithm that always uses betting fraction v. So long as $\|v\| \leq 1/2$, we have

$$\text{Wealth}_T(v) \geq \epsilon \exp\left(-v \cdot \sum_{t=1}^{T} g_t - \sum_{t=1}^{T} (v \cdot g_t)^2\right)$$

Setting $X = -\sum_{t=1}^{T} g_t \cdot \frac{\mathring{w}}{\|\mathring{w}\|}$, and $Z = \sum_{t=1}^{T} (g_t \cdot \mathring{w}/\|\mathring{w}\|)^2$ yields:

$$\text{Wealth}_T\left(c\frac{\mathring{w}}{\|\mathring{w}\|}\right) \geq \epsilon \exp\left(cX - c^2 Z\right)$$

By mild abuse of notation, we define the regret of our v-choosing algorithm at $c\frac{\mathring{w}}{\|\mathring{w}\|}$ as $R_T^v(c)$, so that following (9) we can write:

$$\text{Wealth}_T \geq \epsilon \exp\left(cX - c^2 Z - R_T^v(c)\right) = f_c(X) \tag{12}$$

where we defined $f_c(X) = \epsilon \exp\left(cX - c^2 Z - R_T^v(c)\right)$. Now by Lemmas 1 and 3, we obtain:

$$\begin{aligned}
R_T(\mathring{w}) &\leq \epsilon + f_c^\star(\|\mathring{w}\|) \\
&= \epsilon + \frac{\|\mathring{w}\|}{c}\left[\log\left(\frac{\|\mathring{w}\|}{\epsilon c \exp(-c^2 Z - R_T^v(c))}\right) - 1\right] \\
&= \epsilon + \frac{\|\mathring{w}\|}{c}\left[\log\left(\frac{\|\mathring{w}\|}{c\epsilon}\right) + c^2 Z + R_T^v(c) - 1\right] \\
&= \epsilon + \frac{\|\mathring{w}\|}{c}(\log(\|\mathring{w}\|/c\epsilon) - 1) + \|\mathring{w}\| cZ + \frac{\|\mathring{w}\|}{c} R_T^v(c)
\end{aligned} \tag{13}$$

□

B. Proof of Theorem 1

The following theorem provides a more detailed version of Theorem 1, including all constants:

Theorem 3. *Suppose $\|g_t\|_\star \le 1$ for some norm $\|\cdot\|$ for all t. Further suppose that* INNEROPTIMIZER *has outputs satisfying $\|v_t\| \le 1/2$ and guarantees regret nearly linear in $\|\mathring{v}\|$:*

$$R_T^v(\mathring{v}) = \sum_{t=1}^T z_t \cdot v_t - z_t \cdot \mathring{v} \le \epsilon + \|\mathring{v}\| G_T(\mathring{v}/\|\mathring{v}\|)$$

for some function $G_T(\mathring{v}/\|\mathring{v}\|)$ for any $\mathring{v}$ with $\|\mathring{v}\| \le 1/2$. Then if $-\sum_{t=1}^T g_t \cdot \frac{\mathring{w}}{\|\mathring{w}\|} \ge 2G_T(\mathring{w}/\|\mathring{w}\|)$, RECURSIVEOPTIMIZER *obtains*

$$R_T(\mathring{w}) \le \epsilon + 4\sqrt{\left(4\|\mathring{w}\|^2 + \sum_{t=1}^T (g_t \cdot \mathring{w})^2\right) \max\left[\log\left(\frac{2\sqrt{4\|\mathring{w}\|^2 + \sum_{t=1}^T (g_t \cdot \mathring{w})^2}}{\epsilon}\right) + \epsilon - 1, 1\right]}$$

and otherwise

$$R_T(\mathring{w}) \le \epsilon + 2\|\mathring{w}\| G_T(\mathring{w}/\|\mathring{w}\|)$$

Proof. First, observe that since $\|v_t\| \le 1/2$ for all t, we must have $\text{Wealth}_T \ge 0$ for all t and so

$$\sum_{t=1}^T g_t \cdot w_t \le \epsilon$$

Therefore, if $-\sum_{t=1}^T g_t \cdot \frac{\mathring{w}}{\|\mathring{w}\|} < 2G_T(\mathring{w}/\|\mathring{w}\|)$ we must have

$$\begin{aligned} R_T(\mathring{w}) &= \sum_{t=1}^T g_t \cdot w_t - g_t \cdot \mathring{w} \\ &= \sum_{t=1}^T g_t \cdot w_t - \|\mathring{w}\| \sum_{t=1}^T g_t \cdot \frac{\mathring{w}}{\|\mathring{w}\|} \\ &\le \epsilon + 2\|\mathring{w}\| G(\mathring{w}/\|\mathring{w}\|) \end{aligned}$$

Which proves one case of the Theorem. So now we assume $-\sum_{t=1}^T g_t \cdot \frac{\mathring{w}}{\|\mathring{w}\|} \ge 2G_T(\mathring{w}/\|\mathring{w}\|)$.

Recall the inequality:

$$\log(\text{Wealth}_T) \ge \log(\epsilon) + \sum_{t=1}^T -g_t \cdot \mathring{v} - (g_t \cdot \mathring{v})^2 - R_T^v(\mathring{v})$$

for any $\mathring{v}$ with $\|\mathring{v}\| \le 1/2$. Using our assumption on R_T^v, and setting $\mathring{v} = c\mathring{w}/\|\mathring{w}\|$ for some unspecified $c \in [0, 1/2]$, we have

$$\begin{aligned} \log(\text{Wealth}_T) &\ge \log(\epsilon) + \sum_{t=1}^T -g_t \cdot \mathring{v} - (g_t \cdot \mathring{v})^2 - (\epsilon + \|\mathring{v}\| G(\mathring{v}/\|\mathring{v}\|)) \\ &\ge -\epsilon + \log(\epsilon) + \sum_{t=1}^T -cg_t \cdot \frac{\mathring{w}}{\|\mathring{w}\|} - c^2 Z - cG\left(\frac{\mathring{w}}{\|\mathring{w}\|}\right) \\ &\ge -\epsilon + \log(\epsilon) + \sum_{t=1}^T -\frac{c}{2} g_t \cdot \frac{\mathring{w}}{\|\mathring{w}\|} - c^2 Z \end{aligned}$$

where we have defined $Z = \sum_{t=1}^T \left(g_t \cdot \frac{\mathring{w}}{\|\mathring{w}\|}\right)^2$. Now we define

$$f(X) = \epsilon \exp\left(-\epsilon - c^2 Z + \frac{c}{2} X\right)$$

to obtain

$$\text{Wealth}_T \geq f\left(-\sum_{t=1}^{T} g_t \cdot \frac{\mathring{w}}{\|\mathring{w}\|}\right)$$

Then using Lemmas 1 and 3 we obtain:

$$\begin{aligned} R_T(\mathring{w}) &\leq \epsilon + f^\star(\|\mathring{w}\|) \\ &\leq \epsilon + \frac{2\|\mathring{w}\|}{c}\left[\log\left(\frac{2\|\mathring{w}\|}{c\epsilon}\right) + \epsilon - 1\right] + 2\|\mathring{w}\|cZ \end{aligned}$$

Now we optimize $c \in [0, 1/2]$ using Lemma 4:

$$R_T(\mathring{w}) \leq \epsilon + 4\|\mathring{w}\|\sqrt{(4+Z)\max\left[\log\left(\frac{2\|\mathring{w}\|\sqrt{4+Z}}{\epsilon}\right) + \epsilon - 1, 1\right]}$$

□

C. Proof of Theorem 2

The following theorem provides a more detailed version of Theorem 2, including all constants and logarithmic factors.

Theorem 4. *Suppose $\|g_t\|_\infty \leq 1$ for all t. Then for all $\|\mathring{w}\|_\infty \leq 1/2$, Algorithm 2 guarantees regret:*

$$\begin{aligned} R_T(\mathring{w}) \leq d\epsilon + 2\sum_{i=1}^{d} |\mathring{w}_i| \max\Bigg[\sqrt{\left[\frac{5}{4\eta} + G_i\left(1+\frac{2}{\eta}\right)\right]\max\left[\log\left(\frac{|\mathring{w}_i|(1+4G_i)^\eta\sqrt{2/\eta + G_i(1+2/\eta)}}{\epsilon}\right) - 1, 1\right]}, \\ 2\max\left[\log\left(\frac{|\mathring{w}_i|(1+4G_i)^\eta\sqrt{4+5/4\eta + G_i(1+2/\eta)}}{\epsilon}\right) - 1, 1\right]\Bigg] \\ \leq d\epsilon + 2\|\mathring{w}\|_\infty \sum_{i=1}^{d} \frac{|\mathring{w}_i|}{\|\mathring{w}\|_\infty} \max\Bigg[\sqrt{\left[\frac{5}{4\eta} + G_i\left(1+\frac{2}{\eta}\right)\right]\max\left[\log\left(\frac{(1+4G_i)^\eta\sqrt{5/4\eta + G_i(1+2/\eta)}}{2\epsilon}\right) - 1, 1\right]}, \\ 2\max\left[\log\left(\frac{(1+4G_i)^\eta\sqrt{4+5/4\eta + G_i(1+2/\eta)}}{2\epsilon}\right) - 1, 1\right]\Bigg] \\ := \epsilon d + \|\mathring{w}\|_\infty G(\mathring{w}/\|\mathring{w}\|_\infty) \end{aligned}$$

Proof. First, observe that Algorithm 2 is running d copies of a 1-dimensional algorithm, one per coordinate. Using the classic diagonal trick, we can write

$$R_T(\mathring{w}) \leq \sum_{t=1}^{T}\langle g_t, w_t - \mathring{w}\rangle = \sum_{i=1}^{d}\sum_{t=1}^{T} g_{t,i}(w_{t,i} - \mathring{w}) = \sum_{i=1}^{d} R_{T,i}(\mathring{w})$$

where $R_{T,i}$ indicates the regret of the ith 1-dimensional optimizer. As a result, we will only analyze each dimension individually and combine all the dimensions at the end. To make notation cleaner during this process, we drop the subscripts i.

Next, we claim that it suffices to examine the regret of the x_ts rather than that of the w_ts. In particular, it holds that:

$$g_t(w_t - \mathring{w}) \leq \tilde{g}_t(x_t - \mathring{w})$$

We show this via case-work. First, if $w_t = x_t$ the claim is immediate because $g_t = \tilde{g}_t$. Suppose $g_t(x_t - w_t) \geq 0$. Then $g_t = \tilde{g}_t$ and $g_t x_t \geq g_t w_t$ so that the claim follows. Finally, suppose $g_t(x_t - w_t) < 0$. Then since $x_t \neq w_t$, we must

have $w_t = \text{clip}(x_t, -1/2, 1/2)$ so that $\text{sign}(x_t) = \text{sign}(x_t - w_t) = \text{sign}(w_t)$ and so $\text{sign}(g_t) = -\text{sign}(w_t)$. Further, since $w_t \in \{-1/2, 1/2\}$ and $\mathring{w} \in [-1/2, 1/2]$, $\text{sign}(w_t - \mathring{w}) = \text{sign}(w_t)$. Therefore $g_t(w_t - \mathring{w}) \leq 0 = \tilde{g}_t(x_t - \mathring{w})$. Therefore we can write:

$$\sum_{t=1}^{T} g_t(w_t - \mathring{w}) \leq \sum_{t=1}^{T} \tilde{g}_t(x_t - \mathring{w})$$

The RHS of the above is the regret of the x_ts with respect to the $\tilde{g}_t$s, so we reduce to analyzing this regret. Eventually the regret bound will be increasing in $|\tilde{g}_t|$, and since $|\tilde{g}_t| \leq |g_t|$, we can seamlessly transition to a regret bound in terms of the g_t.

Finally, observe that the x_ts are generated by a betting algorithm using betting-fractions v_t. Inspection of the formula for v_t reveals that we can write:

$$v_t = \operatorname*{argmin}_{v \in [-1/2, 1/2]} \frac{1}{4\eta} A_t v^2 + \sum_{t=1}^{T} z_t v$$

so that the v_t are actually the outputs of an FTRL algorithm using regularizers $\frac{A_t}{4\eta} v^2$, which are $\frac{A_t}{2\eta}$-strongly convex. That is, the x_ts are actually an instance of RECURSIVEOPTIMIZER.

Thus by Lemma 2 we have

$$R_T(\mathring{w}) \leq \inf_{c \in [0,1/2]} \epsilon + \frac{|\mathring{w}|}{c} \left(\log \left(\frac{|\mathring{w}|}{c\epsilon} \right) - 1 \right) + |\mathring{w}| c Z + \frac{|\mathring{w}|}{c} R_T^v \left(c \frac{\mathring{w}}{|\mathring{w}|} \right)$$

where $Z = \sum_{t=1}^{T} g_t^2$ in this one-dimensional case.

Next we tackle R_T^v. To do this, we invoke the FTRL analysis of (McMahan, 2017) to claim:

$$\begin{aligned} R_T^v(x) &\leq \frac{A_T}{4\eta} x^2 + \sum_{t=1}^{T} \frac{z_t^2 \eta}{A_{t-1}^2} \\ &\leq \frac{A_T}{4\eta} x^2 + \eta \sum_{t=1}^{T} \frac{z_t^2}{5 + \sum_{i=1}^{t-1} z_i^2} \end{aligned}$$

Now observe that each z_t satisfies $|z_t| \leq 2|g_t| \leq 2$ so that

$$\begin{aligned} \sum_{t=1}^{T} \frac{z_t^2}{5 + \sum_{i=1}^{t-1} z_i^2} &\leq \sum_{t=1}^{T} \frac{z_t^2}{1 + \sum_{i=1}^{t} z_i^2} \\ &\leq \log \left(1 + \sum_{t=1}^{T} z_t^2 \right) \\ &\leq \log \left(1 + 4 \sum_{t=1}^{T} g_t^2 \right) \end{aligned}$$

Therefore we have

$$\begin{aligned} R_T^v(x) &\leq \frac{5 + 4\sum_{t=1}^{T} g_t^2}{4\eta} x^2 + \eta \log \left(1 + 4 \sum_{t=1}^{T} g_t^2 \right) \\ &= \frac{5 + 4Z}{4\eta} x^2 + \log \left((1 + 4Z)^{\eta} \right) \end{aligned}$$

Plugging back into the result from Lemma 2 we obtain:

$$R_T(\mathring{w}) \leq \inf_{c \in [0,1/2]} \epsilon + \frac{|\mathring{w}|}{c} \left(\log \left(\frac{|\mathring{w}|(1 + 4Z)^{\eta}}{c\epsilon} \right) - 1 \right) + |\mathring{w}| c (5/4\eta + Z(1 + 2/\eta))$$

Then using Lemma 4 we get:

$$R_T(\mathring{w}) \leq \epsilon + 2|\mathring{w}| \max\left[\sqrt{\left[\frac{5}{4\eta} + Z\left(1+\frac{2}{\eta}\right)\right]\max\left[\log\left(\frac{|\mathring{w}|(1+4Z)^\eta\sqrt{2/\eta + Z(1+2/\eta)}}{\epsilon}\right) - 1, 1\right]}, \right.$$
$$\left. 2\max\left[\log\left(\frac{|\mathring{w}|(1+4Z)^\eta\sqrt{4+5/4\eta + Z(1+2/\eta)}}{\epsilon}\right) - 1, 1\right]\right]$$

Now we simply combine each of the d dimensional regret bounds and observe that in a one-dimension, $Z = \sum_{t=1}^T g_t, i^2 = G_i$ to obtain:

$$R_T(\mathring{w}) \leq d\epsilon + 2\sum_{i=1}^d |\mathring{w}_i| \max\left[\sqrt{\left[\frac{5}{4\eta} + G_i\left(1+\frac{2}{\eta}\right)\right]\max\left[\log\left(\frac{|\mathring{w}_i|(1+4G_i)^\eta\sqrt{5/4\eta + G_i(1+2/\eta)}}{\epsilon}\right) - 1, 1\right]}, \right.$$
$$\left. 2\max\left[\log\left(\frac{|\mathring{w}_i|(1+4G_i)^\eta\sqrt{4+5/4\eta + G_i(1+2/\eta)}}{\epsilon}\right) - 1, 1\right]\right]$$

□

C.1. Optimal Logarithmic Factors

The previous analysis obtains logarithmic factors of the form $\log(|w|Z^{1/2+\eta}/\epsilon)$ for any given $\eta > 0$. For $|w| > \epsilon$, this is the same up to constant factors as the optimal bound $\log(|w|\sqrt{Z}/\epsilon)$. However, for small w this is not so. In the small-$\mathring{w}$ case, our bound is already an improvement on the previous exponent (Cutkosky & Orabona, 2018), which has an exponent of 4.5 instead of $1/2+\eta$, but here we sketch how to remove η completely using the classic doubling trick. We present the idea in one dimensional unconstrained problems only: conversion to constrained or high dimensional problems may be accomplished via per-coordinate updates as in Theorem 4, or via the dimension-free reduction in (Cutkosky & Orabona, 2018). The idea is essentially the same as Algorithm 2, but instead of using a varying A_t, we use a *fixed* A and set $\eta = 1$. We restart the algorithm with a doubled value for A whenever we observe $2Z = 2\sum g_t^2 > A$. Let us analyze this scheme during one epoch of fixed A-value. Following identical analysis as in Theorem 4, we observe that

$$\begin{aligned} R_T^v(x) &\leq \frac{A}{2}x^2 + \frac{1}{2}\sum_{t=1}^T \frac{z_t^2}{A} \\ &\leq \frac{A}{2}x^2 + \frac{1}{2}\sum_{t=1}^T \frac{4g_t^2}{A} \\ &\leq \sum_{t=1}^T g_t^2 x^2 + 1 = Zx^2 + 1 \end{aligned}$$

Then applying Theorem 2 we have

$$\begin{aligned} R_T^k(\mathring{w}) &\leq \inf_{c\in[0,1/2]} \epsilon + \frac{|\mathring{w}|}{c}\left(\log\left(\frac{|\mathring{w}|}{c\epsilon}\right) - 1\right) + 2|\mathring{w}|cZ_k + \frac{|\mathring{w}|}{c} \\ &= \epsilon + \frac{|\mathring{w}|}{c}\log\left(\frac{|\mathring{w}|}{c\epsilon}\right) + 2|\mathring{w}|cZ_k \end{aligned}$$

where R_T^k indicates regret in the kth epoch and Z_k is the value of Z in the kth epoch. Optimizing c, we obtain:

$$R_T(\mathring{w}) \leq \epsilon + 2|\mathring{w}| \max\left[\sqrt{2Z_k \max\left[\log\left(\frac{|\mathring{w}|\sqrt{2Z_k}}{\epsilon}\right), 1\right]}, \; 2\max\left[\log\left(\frac{|\mathring{w}|\sqrt{4+2Z_k}}{\epsilon}\right), 1\right]\right]$$

Let Z be the true value of Z (i.e. $Z = \sum_{t=1}^{T} g_t^2$ across all epochs, in contrast to a Z_k). Then we have

$$R_T^k(\mathring{w}) \leq \epsilon + 2|\mathring{w}| \max\left[\sqrt{2Z_k \max\left[\log\left(\frac{|\mathring{w}|\sqrt{2Z}}{\epsilon}\right), 1\right]},\ 2\max\left[\log\left(\frac{|\mathring{w}|\sqrt{4+2Z}}{\epsilon}\right), 1\right]\right]$$

Then summing over all epochs, we obtain

$$R_T(\mathring{w}) \leq O\left(\epsilon\log(Z) + |\mathring{w}| \max\left[\sqrt{Z \max\left[\log\left(\frac{|\mathring{w}|\sqrt{2Z}}{\epsilon}\right), 1\right]},\ \log(Z)\max\left[\log\left(\frac{|\mathring{w}|\sqrt{4+2Z}}{\epsilon}\right), 1\right]\right]\right)$$

D. Experimental Details

In this Section we describe our experiments in detail. All of our neural network experiments were conducted using the Tensor2Tensor library (Vaswani et al., 2018). We evaluated RECURSIVEOPTIMIZER on several datasets included in the library, including MNIST and CIFAR-10 image classification, LM1B language modeling with 32k, and IMDB sentiment analysis tasks. On CIFAR-10, we used a ResNet model (He et al., 2016) (ResNet-32), on MNIST we used a simple two layer fully connected network as well as logistic regression, and for the remaining tasks we used the Transformer model (Vaswani et al., 2017).

We used $\epsilon = 1.0$ in RECURSIVEOPTIMIZER. For our baseline optimizers Adam and Adagrad, we used default parameters provided by Tensor2Tensor for each dataset when available. Often these were not available for Adagrad, in which case we manually tuned the learning rate on a small exponentially spaced grid. Experiments with larger models or data sets, i.e. CIFAR-10 and LM1B, ran on single NVIDIA P100 GPU, the rest on single NVIDIA K1200 GPU.

D.1. Choice of Inner Optimizer

Our analysis uses a Follow-the-Regularized-Leader algorithm in the inner optimizer DIAGOPTIMIZER to choose the inner-most betting fraction v_t. However, according to Theorem 1, we may use *any* optimizer with a sufficiently good regret guarantee as the inner optimizer. Since our initial submission, (Kempka et al., 2019) proposed the SCINOL algorithm that obtains regret similar to DIAGOPTIMIZER (albeit with somewhat worse logarithmic factors). However, we found that using SCINOL resulted in much better performance on the Transformer model tasks, so we used it as the inner optimizer in all experiments. We conjecture that our algorithm is inheriting some of the scale-invariance properties of SCINOL, which allows it to be more robust. We stress that this is still theoretically sound - the only change will be a small increase in the logarithmic factors.

D.2. Momentum Analog

In our experiments we found that augmenting RECURSIVEOPTIMIZER with the "momentum"-like offsets for parameter-free online learning proposed by (Cutkosky & Boahen, 2017b; Cutkosky & Orabona, 2018) improved the empirical performance on CIFAR-10, so all of our results show two curves for RECURSIVEOPTIMIZER, both with and without momentum (except for the synthetic experiments, in which we did not use momentum). In brief, this consists of replacing each iterate w_t with $w_t + \overline{w}_t$ where

$$\overline{w}_t = \sum_{t'=1}^{t} \frac{\|g_{t'}\|_\star^2 w_{t'}}{\sum_{t'=1}^{t} \|g_{t'}\|_\star^2}$$

D.3. Dealing with unknown bound on g_t

Our theory requires $\|g_t\|_\star \leq 1$ where $\|\cdot\|$ is the ∞ norm. Although we may replace 1 with any known bound $g_{\max}$, it is not possible to simply ignore this requirement in implementing the algorithm: doing so may cause wealth to become negative, which will completely destabilize the algorithm since it will be implicitly differentiating the logarithm of a negative number. However, we do not wish to have to provide this bound to the algorithm, so we adopt a simple heuristic. We maintain $g_{\max}$, the maximum value of $\|g_t\|_1$ we have observed so far during the course of the optimization. Then instead of providing g_t to RECURSIVEOPTIMIZER, we provide $g_t/g_{\max}$. Ideally, $g_{\max}$ will only increase during the very beginning of

the optimization, after which we will simply be rescaling the gradients by a constant factor. Since our regret bounds are nearly scale-free, this should hopefully have negligible effect on the performance. Note that it is actually impossible to design an algorithm that maintains regret nearly linear in $\|\mathring{w}\|$ while also being adaptive to the unknown final value of $g_{\max}$ (Cutkosky & Boahen, 2017a).

D.4. Initial Betting Fraction

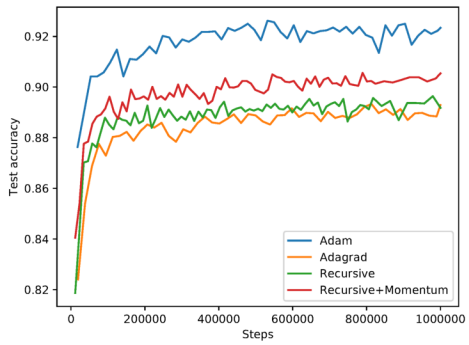
Prior results on coin-betting in deep learning (Orabona & Tommasi, 2017) suggest that a valuable heuristic is to keep the initial betting fraction smaller than some moderate constant. This has the effect of preventing the initial step taken by the algorithm from being too large. We choose to apply this heuristic to the betting fraction of the inner-optimizer - *not* the betting fraction of the outer optimizer. Note that SCINOL is also a coin-betting algorithm, so it still makes sense to apply the heuristic in this manner. We clip the inner betting fraction of dimension i to be always at most $\eta = 0.1$ until $\sum v_{t,i}^2 \geq 1$ where v_t is the gradient passed to the inner betting fraction. This trick has no theoretical basis, but seems to provide significant improvement in the deep learning experiments.

D.5. Empirical Results

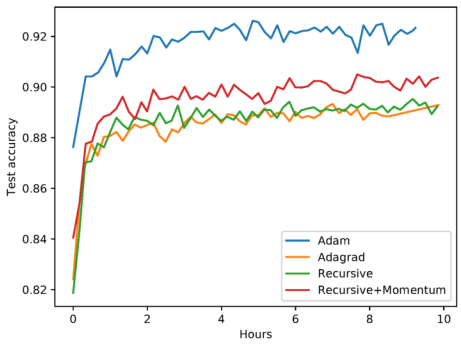
Now we plot our performance on the benchmarks. We record performance on train and test set, both in terms of number of iterations as well as wall clock time. Generally from eight possible combinations of train, test-top-1 accuracy, loss-steps, time curves we show train loss and test accuracy both by steps and time, other combinations are indistinguishably similar and omitted for brevity. For LM1B and Penn Tree Bank language models we include the log perplexity metric as well.

With regard to efficiency observe that the right hand side plots of Figures 3 and 8 whose x-axis is wall clock time are rather similar to left hand side plots based on number of iterations. For a more accurate view, Figure 9 shows that RECURSIVEOPTIMIZER is somewhat slower than both Adam and Adagrad. It is evident that the algorithm requires more computation, although only by a constant factor. We made essentially no effort to optimize our code. We expect that with more careful implementation these numbers can be improved. Secondly, Adam (more specifically LazyAdam used by Tensor2Tensor framework) and Adagrad optimizers handle sparse and dense gradients differently. Our current implementation treats sparse gradients as if they were dense ignoring their sparsity which is detrimental for large vocabulary embeddings.

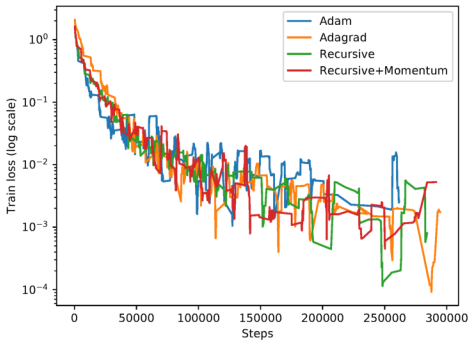
Observe that on the convex logistic regression task, all optimizers converge to the same minimum of train loss, as theory predicts. On the non-convex neural network tasks, RECURSIVEOPTIMIZER seems to be marginally better than the baselines on the Transformer task, but slightly worse than Adam on CIFAR-10. Interestingly, the momentum heuristic was helpful on CIFAR-10, but seemed detrimental on the Transformer tasks. We suspect that RECURSIVEOPTIMIZER is held back on these non-convex tasks by the somewhat global nature of our update. Because our iterates are $v_t\text{Wealth}_{t-1}$, it is easily feasible for the iterate to change quite dramatically in a single round as wealth becomes larger. In contrast, proximal methods such as Adam or Adagrad enforce some natural stability in their iterates. In future, we plan to develop a version of our techniques that also enforces some natural stability, which may be more able to realize gains in the non-convex setting.



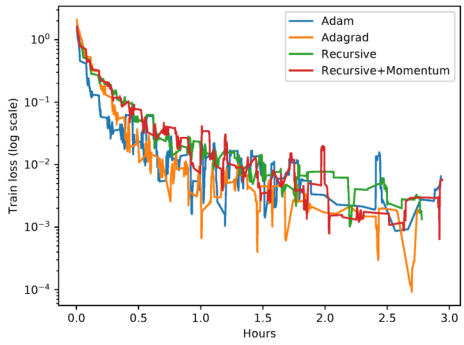
(a) Test accuracy vs steps



(b) Test accuracy vs time

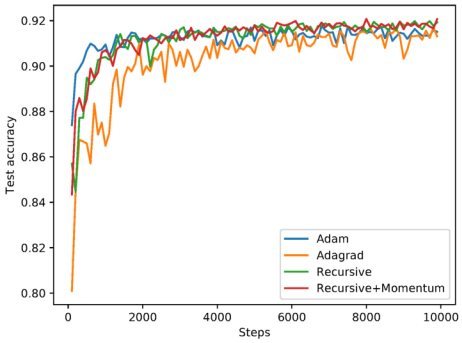


(c) Train loss vs steps

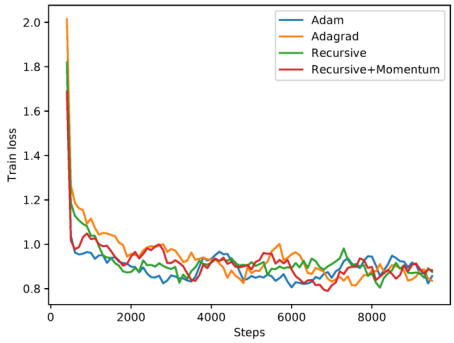


(d) Train loss vs time

Figure 3. CIFAR-10 with ResNet-32

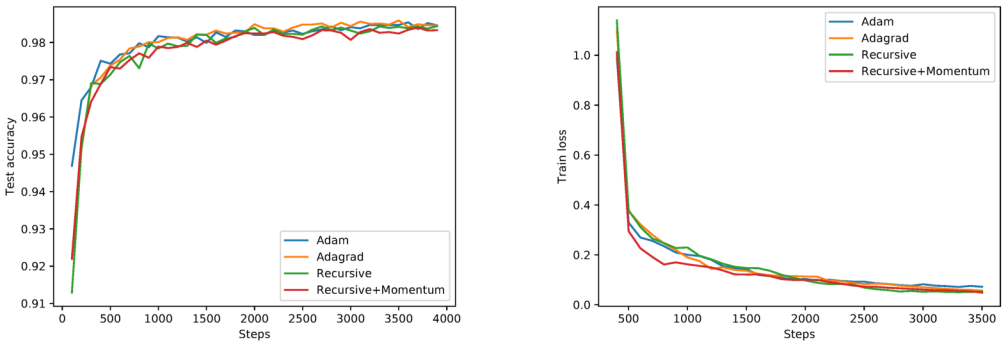


(a) Test accuracy vs steps



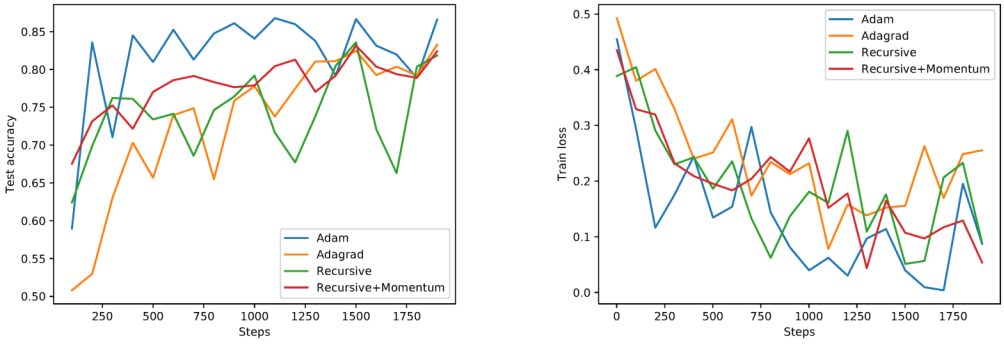
(b) Train loss vs steps

Figure 4. MNIST with logistic regression



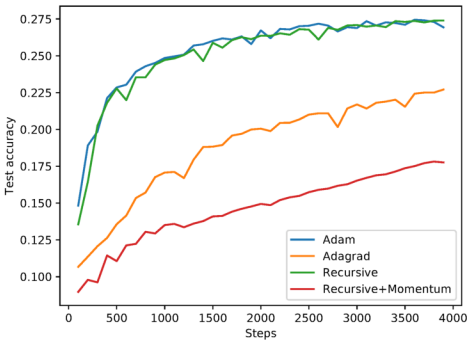
(a) Test accuracy vs steps (b) Train loss vs steps

Figure 5. MNIST with two layer fully connected network

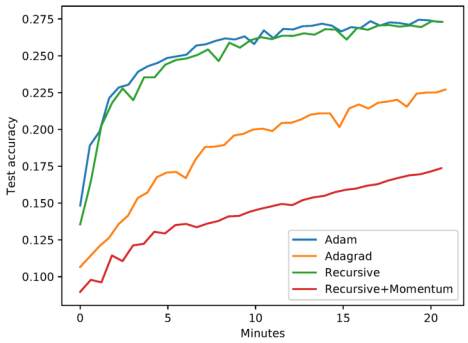


(a) Test accuracy vs steps (b) Train loss vs steps

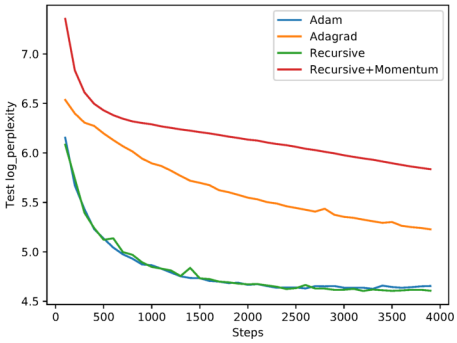
Figure 6. IMDB sentiment classification with Transformer



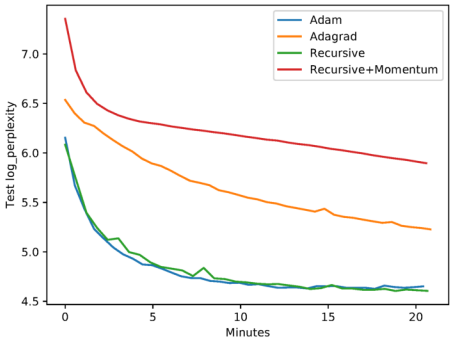
(a) Test accuracy vs steps



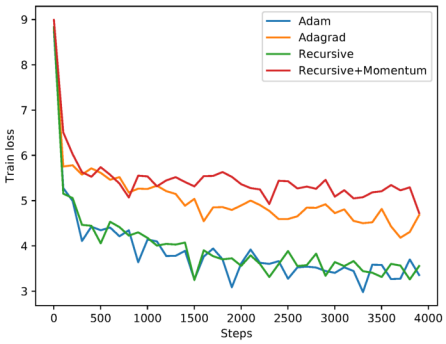
(b) Test accuracy vs time



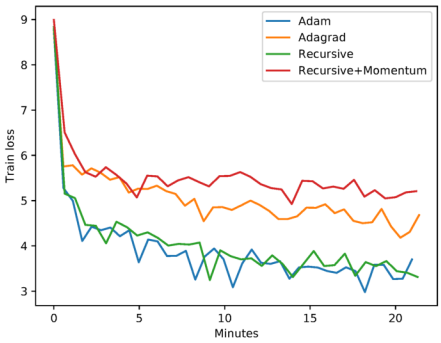
(c) Test log perplexity vs steps



(d) Test log perplexity vs time

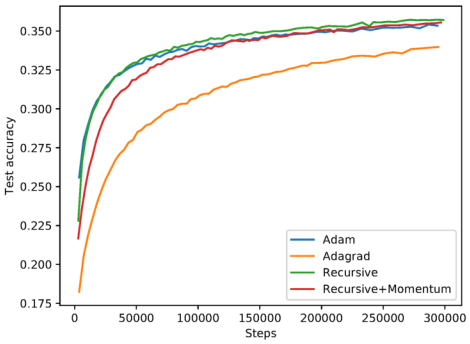


(e) Train loss vs steps

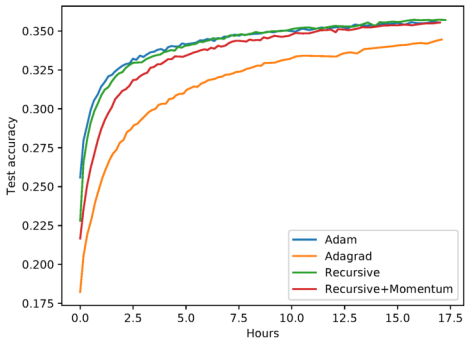


(f) Train loss vs time

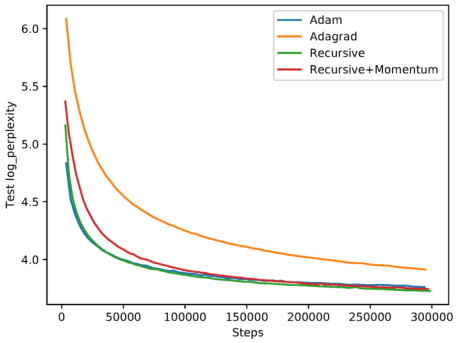
Figure 7. Penn Tree Bank with Transformer



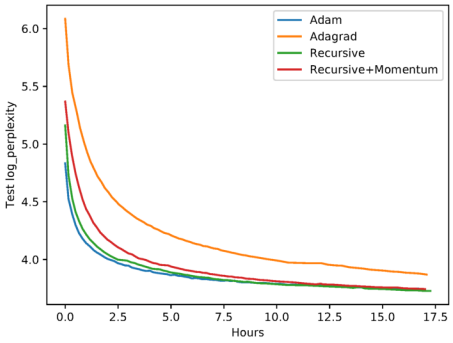
(a) Test accuracy vs steps



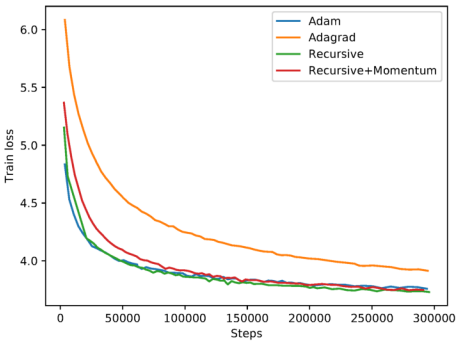
(b) Test accuracy vs time



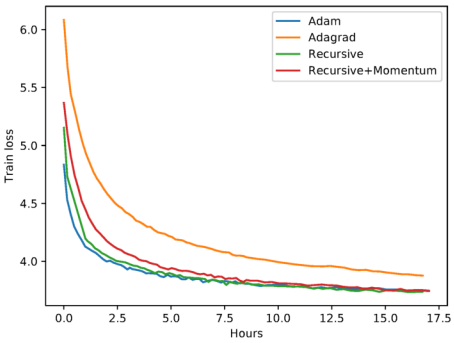
(c) Test log perplexity vs steps



(d) Test log perplexity vs time

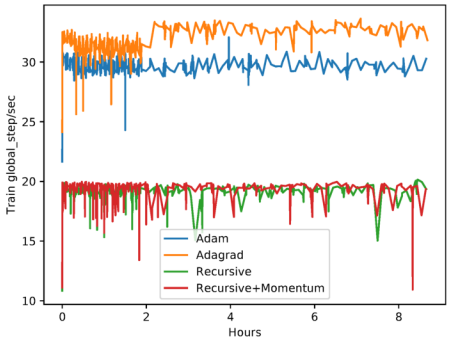


(e) Train loss vs steps

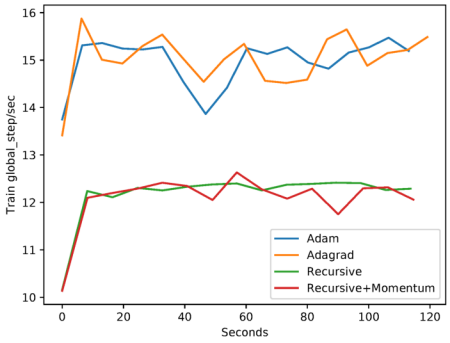


(f) Train loss vs time

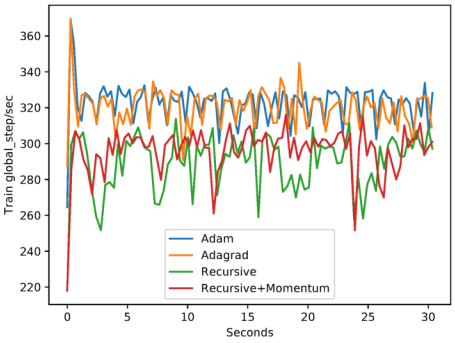
Figure 8. LM1B with Transformer



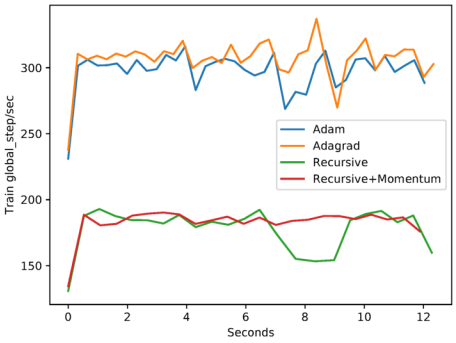
(a) CIFAR-10 with ResNe-32t



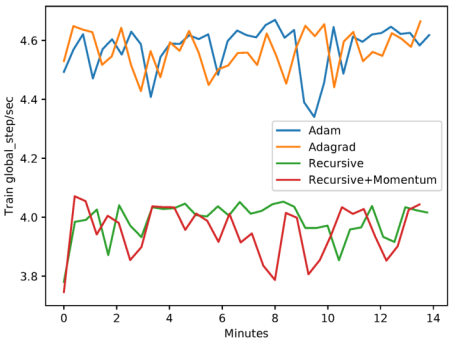
(b) IMBD with Transformer



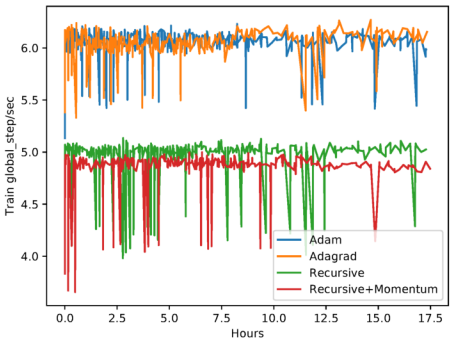
(c) MNIST with logistic regression



(d) MNIST with fully connected network



(e) Penn Tree Bank with Transformer



(f) LM1B with Transformer

Figure 9. Number of iterations per second