AdaDelta: Convergence in Direction to the Maximum-Margin Solution

Avi Semler

January 28, 2024

Abstract

(This is a chapter taken from my work-in-progress dissertation on maximum-margin convergence in neural networks.)

Gradient descent optimisation on homogenous neural networks (e.g. with ReLU activation and no bias) is known to converge in direction a KKT point of a corresponding hard-margin SVM optimisation problem. Analogous results have been shown for the RMSProp and Adam (without momentum) optimisers, but not for AdaDelta. This chapter extends existing results to AdaDelta in the case of linear classification.

Contents

1 AdaDelta algorithm

AdaDelta [1] is an algorithm used for optimisation with gradient descent. It uses per-parameter adaptive learning rates to speed up learning whilst being computationally cheap. For this reason, we used the AdaDelta optimiser to generate the dataset for training the diffusion model from the previous section. Although empirically this does result in good reconstructions, it would be desirable to have a corresponding theoretical result that justifies this by showing that AdaDelta converges in direction to the maximum-margin solution.

The algorithm here is modified slightly to include a learning rate $\eta \in \mathbb{R}$, as is done in the PyTorch implementation [2]. The other hyperparameter is a small constant ε - it was initially included to improve numerical conditioning but has since been shown to be critical to the generalisation behaviour [3].

Convergence in direction to the maximum-margin solution has been proven for other adaptive optimisation methods such as RMSProp and Adam (without momentum) [3]. However, there is no corresponding result for AdaDelta.

Algorithm 1 Computing AdaDelta updates

```
Initialise $E[\Delta x^2]_0 = 0, E[g^2]_0 = 0$
for $t = 1, \dots, T$ do
    Let $g_t = \nabla\mathcal{L}(\theta(t))$
    $E[g^2]_t \leftarrow \rho E[g^2]_{t-1} + (1-\rho)g_t^2$
    $\Delta x_t = -\frac{\sqrt{\varepsilon + E[\Delta x^2]_{t-1}}}{\sqrt{\varepsilon + E[g^2]_t}} \odot g_t$
    $E[\Delta x^2]_t \leftarrow \rho E[\Delta x^2]_{t-1} + (1-\rho)\Delta x_t^2$
    Set $x_{t+1} = x_t + -\eta\Delta x$
end for
```

2 Continuous formulation of AdaDelta

In order to be able to analyse the behaviour of AdaDelta by taking derivatives, we consider a continuous formulation derived analogously to the continuous version of RMSProp derived in [3].

The discrete update rule for AdaDelta can be written as

$$\theta(t+1) - \theta(t) = -\eta h(t) \odot \nabla\mathcal{L}(\theta(t)) \tag{1}$$

Here, $\odot$ denotes elementwise product, and $h(t) \in \mathbb{R}^p$ is the conditioner function that scales the size of the per-parameter updates, given by

$$h(t) = \frac{\sqrt{\varepsilon + D(t)}}{\sqrt{\varepsilon + G(t)}} \tag{2}$$

All operations are elementwise, and $D(t), G(t) \in \mathbb{R}^p$ are given by difference equations:

$$\begin{aligned} D(t+1) - D(t) &= (1-\rho)\left(h(t)^2\nabla\mathcal{L}(\theta(t))^2 - D(t)\right) \\ D(0) &= 0 \end{aligned}$$

and

$$\begin{aligned} G(t+1) - G(t) &= (1-\rho)\left(\nabla\mathcal{L}(\theta(t))^2 - G(t)\right) \\ G(0) &= (1-\rho)\left(\nabla\mathcal{L}(\theta(0))^2\right) \end{aligned}$$

This can be seen as a discretisation (using Euler's method) of the initial value problems given by: (I'm a bit unsure about this step - I *think* it's fine to to this even though $h(t)$ is defined in terms of $D(t)$)

$$\begin{aligned} \frac{d}{dt}D(t) &= (1-\rho)\left(h(t)^2\nabla\mathcal{L}(\theta(t))^2 - D(t)\right) \\ D(0) &= 0 \\ \frac{d}{dt}G(t) &= (1-\rho)\left(\nabla\mathcal{L}(\theta(t))^2 - G(t)\right) \\ G(0) &= (1-\rho)\left(\nabla\mathcal{L}(\theta(0))^2\right) \end{aligned}$$

Both can solved using the integrating factor $e^{\int(1-\rho)dt} = e^{(1-\rho)t}$ and then integrating:

$$\frac{d}{dt}\left(e^{(1-\rho)t}D(t)\right) = e^{(1-\rho)t}(1-\rho)\left(h(t)^2\nabla\mathcal{L}(\theta(t))^2\right) \tag{3}$$

$$\implies D(t) = \int_0^t (1-\rho)e^{-(1-\rho)(t-\tau)}h(\tau)^2\nabla\mathcal{L}(\theta(\tau))^2 d\tau + e^{-(1-\rho)t}C_1 \tag{4}$$

$$\frac{d}{dt}\left(e^{(1-\rho)t}G(t)\right) = e^{(1-\rho)t}(1-\rho)\left(\nabla\mathcal{L}(\theta(t))^2\right) \tag{5}$$

$$\implies G(t) = \int_0^t (1-\rho)e^{-(1-\rho)(t-\tau)}\nabla\mathcal{L}(\theta(\tau))^2 d\tau + e^{-(1-\rho)t}C_2 \tag{6}$$

Now we substitute $t = 0$, and use the initial conditions to determine the value of the constants:

$$C_1 = 0, C_2 = (1-\rho)\left(\nabla\mathcal{L}(\theta(0))^2\right)$$

This leads to a definition for a continuous form of AdaDelta, by using the continuous definitions for $D(t)$ and $G(t)$ - which can be seen as being analogous to $E[\Delta x^2]_t$ and $E[g^2]_t$ respectively.

Definition 1 (Continuous AdaDelta flow)**.** *A continuous AdaDelta flow is a function $\theta : [0, \infty) \to \mathbb{R}^p$ satisfying*

$$\dot{\theta}(t)_i = -h(t)_i \frac{\partial}{\partial\theta_i}\mathcal{L}(\theta(t))$$

where $h : [0, \infty) \to \mathbb{R}^p$ is the continuous AdaDelta conditioner function given by (using elementwise operations)

$$h(t) = \frac{\sqrt{\varepsilon + D(t)}}{\sqrt{\varepsilon + G(t)}} = \frac{\sqrt{\varepsilon + \int_0^t (1-\rho)e^{-(1-\rho)(t-\tau)}h(\tau)^2\nabla\mathcal{L}(\theta(\tau))^2 d\tau}}{\sqrt{\varepsilon + \int_0^t (1-\rho)e^{-(1-\rho)(t-\tau)}\nabla\mathcal{L}(\theta(\tau))^2 d\tau + (1-\rho)e^{-(1-\rho)t}\nabla\mathcal{L}(\theta(0))^2}}$$

Note that h occurs on both sides of the definition - this can be interpreted as an integral equation that h must satisfy.

3 Maximum-margin convergence

To prove that the AdaDelta flow converges to the maximum-marign solution, we make assumptions about the scenario.

Assumption 2.

1. *The dataset $\{x_n\}_{n=1}^N$ is linearly separable: there exists $\theta_* \in \mathbb{R}^p$ such that $\theta_*^T x_n > 0$ for all $n = 1, \dots, N$*
2. *We are using θ as a linear classifier: the parameter-function map is given by $\phi(\theta, x) = \theta^T x$.*
3. *$\mathcal{L}$ is the exponential loss function given by $\mathcal{L}(\theta) = \sum_{n=1}^N \exp(-x_n \cdot \theta)$ - I think that this could be relaxed to assuming only that the loss function has exponential tails*

We start with a general theorem providing a sufficient condition for a flow corresponding to an adaptive algorithm to converge in direction to the maxmimum-margin solution. The subsequent lemmas then establish that AdaDelta flows satisfy the conditions of the theorem.

Theorem 3 (A special case of Theorem 2 in Wang et al., 2021 [3]). *Let $\theta : [0, \infty) \to \mathbb{R}^p$ be an adaptive gradient flow of the loss function $\mathcal{L}$ with per-component conditioner function $a(t)$ on the dataset $\{x_n\}_{n=1}^N$ satisfying Assumption 2. That is to say:*

$$\dot{\theta}(t) = -a(t) \odot \nabla \mathcal{L}(\theta(t))$$

for some function $a : [0, \infty) \to \mathbb{R}^p$, the dataset is linearly separable, and $\mathcal{L}$ is the exponential loss using θ as a linear classifier.

If the following conditions hold:

- $\lim_{t\to\infty} a(t) = (1, \ldots, 1) \in \mathbb{R}^p$
- $\frac{d}{dt}\log(a(t))$ *is Lebesgue integrable*
- $\lim_{t\to\infty} \mathcal{L}(\theta(t)) = 0$*; i.e. the parameter achieves a global minimum of the loss in the limit*

Then θ lies in the direction of a KKT point of the following optimisation problem:

$$\min \frac{1}{2}|w|^2$$

$$\textit{subject to } x_n \cdot w \geq 1 \textit{ for all } n \in \{1, \ldots, N\}$$

The roadmap to establishing that these conditions hold is:

- Prove that the derivative of the AdaDelta conditioner function $h(t)$ points towards 1 and $h(t)$ is therefore bounded
- Bound the gradients of the loss and hence show that the loss tends to zero
- Use the bound on the gradients to show that $h(t)$ tends to $(1, \ldots, 1)$

Lemma 4 (Derivative of AdaDelta conditioner eventually points towards 1 in every component and is bounded). *For $i = 1, \ldots, p$, there exists $T_i > 0$ such that for all $t \geq T_i$:*

- *If $h(t)_i > 1$, then $h'(t)_i < 0$*
- *If $h(t)_i < 1$, then $h'(t)_i > 0$*
- *If $h(t)_i = 1$, then $h'(t)_i = 0$*

This also means that $h(t)_i$ is bounded, being contained in either $[h(0)_i, 1]$ or $[1, h(0)_i]$.

Proof. We use the abbrevation $g(t) := \nabla\mathcal{L}(\theta(t))$ for the gradient. In order to simplify the notation, this proof drops the subscripts, e.g. referring to $h(t)_i$ as simply $h(t)$.

$$\frac{d}{dt}h(t)^2 = 2h(t)h'(t) = \frac{d}{dt}\left(\frac{\varepsilon + D(t)}{\varepsilon + G(t)}\right)$$

$$= \frac{d}{dt}\left(\frac{\varepsilon + \int_0^t (1-\rho)e^{-(1-\rho)(t-\tau)}h(\tau)^2 g(\tau)^2 d\tau}{\varepsilon + \int_0^t (1-\rho)e^{-(1-\rho)(t-\tau)}g(\tau)^2 d\tau + (1-\rho)e^{-(1-\rho)t}g(0)^2}\right)$$

$$= \frac{(1-\rho)g(t)^2h(t)^2 - (1-\rho)D(t)}{\varepsilon + G(t)} - \frac{\Big(\varepsilon + D(t)\Big)\Big((1-\rho)g(t)^2 - (1-\rho)G(t)\Big)}{(\varepsilon + G(t))^2}$$

by applying the quotient rule for differentiation. Expanding the product on the second numerator:

$$\begin{aligned}=&\frac{(1-\rho)g(t)^2h(t)^2-(1-\rho)D(t)}{\varepsilon+G(t)}\\ &-\frac{\varepsilon(1-\rho)g(t)^2-\varepsilon(1-\rho)G(t)+(1-\rho)g(t)^2D(t)-(1-\rho)G(t)D(t)}{(\varepsilon+G(t))^2}\end{aligned}$$

By rearranging the definition of $h(t)$, we know that $D(t)=\varepsilon h(t)^2+G(t)h(t)^2-\varepsilon$. Substitute this for the first occurence of $D(t)$ in the second numerator:

$$\begin{aligned}=&\frac{(1-\rho)g(t)^2h(t)^2-(1-\rho)D(t)}{\varepsilon+G(t)}\\ &-\frac{\varepsilon(1-\rho)g(t)^2-\varepsilon(1-\rho)G(t)+(1-\rho)g(t)^2(\varepsilon h(t)^2+G(t)h(t)^2-\varepsilon)-(1-\rho)G(t)D(t)}{(\varepsilon+G(t))^2}\end{aligned}$$

Four terms on the numerators now cancel:

$$\begin{aligned}=&\frac{-(1-\rho)D(t)}{\varepsilon+G(t)}-\frac{-\varepsilon(1-\rho)G(t)-(1-\rho)G(t)D(t)}{(\varepsilon+G(t))^2}\\ =&\frac{-(1-\rho)D(t)}{\varepsilon+G(t)}-\frac{-(1-\rho)G(t)(D(t)+\varepsilon)}{(\varepsilon+G(t))^2}\end{aligned}$$

In order to simplify this expression further, we rewrite $G(t)(D(t)+\varepsilon)$ using rearrangements of the equation $h(t)^2=\frac{\varepsilon+D(t)}{\varepsilon+G(t)}$.

$$G(t)(D(t)+\varepsilon)=\left(\frac{\varepsilon+D(t)}{h(t)^2}-\varepsilon\right)(\varepsilon h(t)^2+h(t)^2G(t)-\varepsilon+\varepsilon)=\quad(\varepsilon+D(t)-\varepsilon h(t)^2)(\varepsilon+G(t))$$

Now substitute this back in to the expression for $\frac{d}{dt}h(t)^2$:

$$\frac{-(1-\rho)D(t)}{\varepsilon+G(t)}-\frac{-(1-\rho)(\varepsilon+D(t)-\varepsilon h(t)^2)(\varepsilon+G(t))}{(\varepsilon+G(t))^2}$$

The denominator of the second fraction partially cancels, leading to:

$$\begin{aligned}&\frac{-(1-\rho)D(t)+(1-\rho)(\varepsilon+D(t)-\varepsilon h(t)^2)}{\varepsilon+G(t)}\\ =&\frac{\varepsilon(1-\rho)(1-h(t)^2)}{\varepsilon+G(t)}\end{aligned}$$

From here we can see that the desired result holds: since $h(t)>0$, $h'(t)$ has the same sign as $\frac{d}{dt}(h(t)^2)=2h(t)h'(t)>0$. And for $h(t)>1$ this sign is negative, for $h(t)<1$ this sign is positive, and the expression is zero for $h(t)=1$. □

This brings us one step closer to showing that $h(t)_i$ tends to 1. It is now useful to bound $G(t)_i$, so that we can ensure that the gradient does not shrink too fast (if it did shrink too fast then $h(t)$ would not tend to 1, even if the derivative points in the right direction).

Lemma 5 (Loss and gradient of loss tend to zero; analogous to Lemma 2 in [4])**.** *For an AdaDelta flow satisfying Assumption 2 we have that* $\lim_{t\to\infty}\mathcal{L}(\theta(t))=0$ *and* $\lim_{t\to\infty}g(t)=0$ *(using the abbrevation* $g(t):=\nabla\mathcal{L}(\theta(t))$ *as above).*

Proof. We will start with the limit of the gradient.

Assume, for a contradiction, that for some $j \in \{1, \ldots, p\}$

$$\int_0^\infty g(t)_j^2 dt = \infty \tag{7}$$

Now we will show that it would follow from this assumption that $\mathcal{L}(\theta(0)) - \mathcal{L}(\theta(t))$ tends to infinity as $t \to \infty$, which would be a contradiction as this quantity clearly never exceeds $\mathcal{L}(\theta(0))$.

$$\begin{aligned}
\mathcal{L}(\theta(0)) - \mathcal{L}(\theta(t)) &= -\int_0^t \frac{d\mathcal{L}(\theta(s))}{ds} ds \\
&= -\int_0^t \nabla\mathcal{L}(\theta(s)) \cdot \dot{\theta}(s) ds \\
&= -\int_0^t \nabla\mathcal{L}(\theta(s)) \cdot \left(-h(s) \odot \nabla\mathcal{L}(\theta(s))\right) ds \\
&= \int_0^t g(s) \cdot (h(s) \odot g(s)) ds \\
&= \sum_{i=1}^p \int_0^t h_i(s) g(s)_i^2 ds
\end{aligned}$$

Next, by the previous lemma we know that $h(t)_i \geq \min\{1, h(0)_i\}$. So:

$$\mathcal{L}(\theta(0)) - \mathcal{L}(\theta(t)) \geq \sum_{i=1}^p \min\{1, h(0)_i\} \int_0^t g(s)_i^2 ds$$

Each summand is positive, so if any one of the summands tends to infinity, then so does the whole sum. But by our assumption, the j^{th} summand does tend to infinity! This implies that $\mathcal{L}(\theta(0)) - \mathcal{L}(\theta(t)) \to \infty$ too. This brings us to a contradiction as $\mathcal{L}(\theta(0)) - \mathcal{L}(\theta(t)) \leq \mathcal{L}(\theta(0))$. So instead:

$$\int_0^\infty g(t)_j^2 dt < \infty \tag{8}$$

implying that $g(t)^2$ tends to 0.

This result now allows us to prove that the loss itself tends to zero too.

$$\theta_* \cdot g(t) = -\sum_{n=1}^N e^{-\theta(t)\cdot x_n} \theta_* \cdot x_n$$

where θ_* is a vector that linearly seperates the dataset, i.e. $\theta_* \cdot x_i > 0$ for all i. Hence this is a sum of positive terms, and the only way that it could possibly tend to 0 is if $e^{-\theta(t)\cdot x_n} \to 0$ for all n. But:

$$\mathcal{L}(t) = \sum_{n=1}^N e^{-\theta(t)\cdot x_n}$$

so it tends to 0 too.

□

Corollary 6 (Bound on decaying integral of gradient squared)**.** *For all $i = 1, \ldots, p$, we have that $\int_0^t (1-\rho) e^{-(1-\rho)(t-\tau)} g(\tau)_i^2 d\tau$ is bounded.*

Proof. We have that $g(t)_i$ is bounded for all i; i.e. there exists M_i such that $g(t)_i \leq M_i \forall t$. Then:

$$\int_0^t (1-\rho)e^{-(1-\rho)(t-\tau)} g(\tau)_i^2 d\tau \leq M_i^2 \int_0^t (1-\rho)e^{-(1-\rho)(t-\tau)} d\tau = M_i^2 \left(1 - (e^{(\rho-1)t})\right)$$

□

Corollary 7 (AdaDelta conditioner function tends to 1)**.**

$$\lim_{t\to\infty} h(t)_i = 1$$

for $i = 1, \dots, p$

Proof. As before, we will focus on each $h(t)_i$ seperately, referring to them as $h(t)$.

The two previous lemmas provide a bound on $\frac{d}{dt}h^2(t)$:

$$\frac{\varepsilon(1-\rho)(\frac{1}{h(t)} - h(t))}{\varepsilon + M_i^2 + (1-\rho)g(0)^2} \leq h'(t) \leq \frac{\varepsilon(1-\rho)(\frac{1}{h(t)} - h(t))}{\varepsilon}$$

This is a differential inequality that allows us to bound $h(t)$.

In the case of $h(0) > 1$:

$$\sqrt{1 + \exp\left(\frac{2(\rho-1)t + c_1}{\varepsilon + (1-\rho)g(0)^2 + M_i^2}\right)} \leq h(t) \leq \sqrt{1 + e^{c_2 + \frac{1}{2}(\rho-1)(t-1)}}$$

In the case of $h(0) < 1$, these bounds are the other way around.

We observe that $h(t)$ is sandwiched between two functions that tend to 1 (as $1-\rho < 0$ so the coefficients of t are negative), so each component tends to 1 too.

Hence $\lim_{t\to\infty} h(t) = (1, \dots, 1) \in \mathbb{R}^p$. □

Corollary 8. *AdaDelta flows converge to the maximum-margin solution*

Proof. The above lemmas establish the sufficient conditions given in Theorem 3, besides for Lebesgue integrability of $\frac{d}{dt}\log(h(t))$.

For this final condition (Lebesgue integrability), we consider cases. If $h(0) > 1$, then h is decreasing and $\frac{d}{dt}\log(h(t))$ is always negative. So:

$$\int_0^\infty \left|\frac{d\log(h(t))}{dt}\right| dt = \int_0^\infty -\frac{d\log(h(t))}{dt} dt = \log(h(0)) < \infty$$

If instead $h(0) < 1$, $h(t)$ is increasing and:

$$\int_0^\infty \left|\frac{d\log(h(t))}{dt}\right| dt = \int_0^\infty \frac{d\log(h(t))}{dt} dt = -\log(h(0)) < \infty$$

In either case, this is Lebesgue integrable. □

References

[1] Matthew D. Zeiler. *ADADELTA: An Adaptive Learning Rate Method*. 2012. arXiv: `1212.5701 [cs.LG]`.

[2] *Adadelta - PyTorch 2.1 documentation*. URL: `https://pytorch.org/docs/stable/generated/torch.optim.Adadelta.html`.

[3] Bohan Wang et al. “The Implicit Bias for Adaptive Optimization Algorithms on Homogeneous Neural Networks”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 10849–10858. URL: `https://proceedings.mlr.press/v139/wang21q.html`.

[4] Daniel Soudry, Elad Hoffer, and Nathan Srebro. “The Implicit Bias of Gradient Descent on Separable Data”. In: *International Conference on Learning Representations*. 2018. URL: `https://openreview.net/forum?id=r1q7n9gAb`.