Beyond Backprop: Online Alternating Minimization with Auxiliary Variables

Anna Choromanska * ECE NYU Tandon
Sadhana Kumaravel * IBM Research
Ronny Luss * IBM Research
Irina Rish* IBM Research
Brian Kingsbury IBM Research
Mattia Rigotti IBM Research
Paolo DiAchille IBM Research
Viatcheslav Gurev IBM Research
Ravi Tejwani IBM Research
Djallel Bouneffouf IBM Research

Abstract

We propose a novel *online alternating minimization (*AltMin*)* algorithm for training deep neural networks, provide theoretical convergence guarantees and demonstrate its advantages on several classification tasks as compared both to standard backpropagation with stochastic gradient descent (backprop-SGD) and to offline alternating minimization. The key difference from backpropagation is an explicit optimization over hidden activations, which eliminates gradient chain computation in backprop, and breaks the weight training problem into independent, local optimization subproblems; this allows to avoid vanishing gradient issues, simplify handling non-differentiable nonlinearities, and perform parallel weight updates across the layers. Moreover, parallel local synaptic weight optimization with explicit activation propagation is a step closer to a more biologically plausible learning model than backpropagation, whose biological implausibility has been frequently criticized. Finally, the online nature of our approach allows to handle very large datasets, as well as continual, lifelong learning, which is our key contribution on top of recently proposed offline alternating minimization schemes (e.g., (Carreira-Perpinan and Wang 2014), (Taylor et al. 2016)).

1 Introduction

The backpropagation algorithm has been the workhorse of neural net learning for several decades, since its introduction in 1970s, and its practical effectiveness is demonstrated by recent successes of deep learning in a wide range of applications. However, it has several drawbacks which continue to motivate research on alternative methods for neural net training.

One well-known problem with backpropagation is *vanishing gradients*: recursive application of the chain rule through multiple layers of compressing nonlinearities causes the magnitudes of gradients to become very small in shallow layers, generating a weak error signal that slows convergence and hampers learning, especially in very deep and/or recurrent networks (Bengio, Simard, and Frasconi 1994; Riedmiller and Braun 1993; Hochreiter and Schmidhuber 1997). Although various techniques were proposed to address this issue, including Long Short-Term Memory (Hochreiter and Schmidhuber 1997), RPROP (Riedmiller and Braun 1993), rectified linear units (ReLU) (Nair and Hinton 2010), the fundamental problem with computing gradients of a deeply nested objective function remains. Furthermore, *backpropagation can only handle differentiable nonlinearities* and it *does not parallelize over the network layers*. Finally, another commonly expressed concern about backpropagation is its *biological implausibility*, including, among other issues, deterministic (rather than stochastic) neurons, precise alternation between the feedforward and backpropagation phases, the requirement that the feedback path have exactly the same connectivity and symmetric weights (transposed) as the corresponding feedforward path, and the fact that the error propagation mechanism does not influence neural activity, unlike known biological feedback mechanisms (Lee et al. 2015; Bartunov et al. 2018).

In order to overcome the vanishing gradient issue and others mentioned above, several approaches were proposed recently, based on the idea of using auxiliary variables associated with hidden unit activations in order to decompose the highly coupled problem of optimizing a nested loss function into multiple, loosely coupled, simpler subproblems. Namely, given a network with L hidden layers, and the output y, the standard formulation involves optimizing the loss function $\mathcal{L}(y, f(\boldsymbol{W}, \boldsymbol{x}_L))$ with respect to the network weights $\boldsymbol{W}$, where $f(\boldsymbol{W}, \boldsymbol{x}_L) = f_{L+1}(\boldsymbol{W}_{L+1}, f_L(\boldsymbol{W}_L, f_{L-1}(\boldsymbol{W}_{L-1}, ...f_1(\boldsymbol{W}_1, \boldsymbol{x})...)$ is a nested function associated with multilayer transformations in a deep network. However, instead of solving the above problem, one can introduce auxiliary variables $\boldsymbol{x}_i, i = 1, ..., L$, and relax constraints ensuring that those variables match the network transformations at each layer. Then the problem of optimizing the loss with respect to both the weights and auxiliary variables decomposes into multiple local subproblems, each involving only activations and weights within the two adjacent layers. Several methods based on auxiliary variables were proposed recently, including the alternating direction method of multipliers (ADMM) (Taylor et al. 2016; Zhang, Chen, and Saligrama 2016) and block coor-

*These authors contributed equally to this work: A.C. - theory, manuscript; S.K. - code, experiments; R.L. - algorithm, code, experiments; I.R. - algorithm, manuscript. Other contributors: B.K. - algorithm, experiments; M.R. - algorithm, code, experiments; V.G - algorithm, code; P.D. - code, experiments; R.T. - code, experiments; D.B. - algorithm.

dinate descent (BCD) methods (Zeng et al. 2018; Zhang and Brand 2017; Zhang and Kleijn 2017; Askari et al. 2018; Carreira-Perpinan and Wang 2014). Empirical evaluation of these demonstrated faster growth of their test accuracy as compared to SGD, due to parallel implementation; moreover, these approaches avoid the vanishing gradient issue and can deal with non-differentiable functions, e.g., binarized neural networks (Courbariaux, Bengio, and David 2015; Hubara et al. 2016). Note that a similar formulation, using Lagrange multipliers, was proposed earlier in (Le Cun 1986; Yann 1987; LeCun et al. 1988), where a constrained formulation involving activations required the output of the previous layer to be equal to the input of the next layer. This gave rise to the approach known as target propagation and its recent extensions (Lee et al. 2015; Bartunov et al. 2018); however, target propagation uses a different approach than iterative BCD and ADMM methods, training instead layer-wise inverses of the forward mappings.

While the BCD and ADMM methods discussed above assume an offline (batch) setting, i.e. the full training dataset being available at each iteration of the training phase, we will focus instead on the *online*, incremental learning approach, performing *alternating minimization (AM)* over the network weights and auxiliary activation variables, which we refer to as *online AM*. One advantage of the online approach is its natural ability to handle very large, practically unlimited, amounts of samples. Furthermore, unlike its offline counterparts, our approach can be applied to multi-task, *continual (lifelong) learning* problems, i.e. online learning in nonstationary environments (though this is not the focus of current paper).

Herein, we introduce two versions of online AM, one using SGD locally for weight updates in each layer (but without the need for chain computation of gradients due to separation into subproblems discussed above), and the second one, based on the online learning approach similar to the online dictionary learning of (Mairal et al. 2009), that relies on accumulation of second-order information which can be viewed as a "memory" of co-activations.

In summary, our contribution is a novel online alternating minimization algorithm for neural net training, utilizing auxiliary activation variables which break the gradient chain rule (unlike classical backpropagation) into independent local optimization subproblems, which is also equipped with a theoretical convergence analysis, often performs similarly or better than its competitors, and possesses additional useful properties lacking in backpropagation, namely: (1) *no vanishing gradients*; (2) handling *non-differentiable nonlinearities* more easily within layer-wise local subproblems; (3) possibility for *parallelization across layers* for weight updates (similar to (Carreira-Perpinan and Wang 2014)); (4) a *more biologically plausible* credit assignment algorithm (local, parallel/distributed, stochastic, activation-based) as compared to backpropagation; (5) *co-activation memory mechanism* – a novel feature arising from our online optimization approach; exploring co-activation memory effects in life-long learning is the direction of our ongoing work, although not the focus of this paper.

This paper is structured as follows. Section 2 introduces the optimization problem for online learning of a deep network with explicit activation variables, and describes the proposed alternating minimization algorithm. Section 3 introduces theoretical results, while Section 4 presents empirical evaluation. Section 5 concludes the paper with a summary and discussion of future work.

2 Alternating Minimization: Breaking Gradient Chains with Auxiliary Variables

We denote as $(\boldsymbol{X}, \boldsymbol{Y}) = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ..., (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ a dataset of n labeled samples, where $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ are the sample and its (vector) label at time t, respectively, such as, for example, one-hot vector encoding $\boldsymbol{y}$ where all entries are zero, except for one entry equal to one. We assume that samples are N-dimensional, $\boldsymbol{x} \in \mathbb{R}^N$, and there are m classes, $\boldsymbol{y} \in \mathbb{R}^m$, Training a fully-connected deep neural network with L hidden layers involves minimizing the (nested) loss function:

$$\min_{\boldsymbol{W}} \sum_{t=1}^{n} \mathcal{L}(\boldsymbol{y}_t, \boldsymbol{a}_t^L, \boldsymbol{W}^{L+1}), \ \ where \ \ \boldsymbol{a}_t^l = \sigma_l(\boldsymbol{c}_t^l),$$
$$\boldsymbol{c}_t^l = \boldsymbol{W}^l \boldsymbol{a}_t^{l-1}, \ l = 1, ..., L, \ and \ \ \boldsymbol{a_t}^0 = \boldsymbol{x}_t, \qquad (1)$$

where $\boldsymbol{W^j}$ denotes the $m_j \times m_{j-1}$ link weight matrix associated with connections from layer $j-1$ to layer j,and where m_{j-1} and m_j are the numbers of nodes in the $j-1$ and jth layers, respectively. We denote by W^{L+1} the $m_L \times m$ weight matrix connecting the last hidden layer L with the output. Here $\boldsymbol{a}^l$ is the *activation* vector of hidden nodes at layer l, obtained by applying a nonlinearity σ, such as the ReLU function used in our experiments, to the linear transformation of the previous-layer activations, which we will refer to as the *codes* $\boldsymbol{c}^l$. We use the multinomial loss as our objective function: $\mathcal{L}(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{W}) = -\log P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{W})$

$$= \ -\sum_{i=1}^{m} \boldsymbol{y}_i (\boldsymbol{w}_i^T \boldsymbol{x}) + \log (\sum_{l=1}^{m} \exp (\boldsymbol{w}_l^T \boldsymbol{x})), \qquad (2)$$

where w_i is the i^{th} column of $\boldsymbol{W}$, y_i is the i^{th} entry of the one-hot vector encoding $\boldsymbol{y}$, and the class likelihood is modeled as $P(y_i = 1|\boldsymbol{x}, \boldsymbol{W}) = \exp (\boldsymbol{w}_i^T \boldsymbol{x}) / \sum_{l=1}^{m} \exp (\boldsymbol{w}_l^T \boldsymbol{x})$.

We use the codes $\boldsymbol{c}^l$ as explicit, auxiliary variables, similar to (Taylor et al. 2016); however, unlike (Taylor et al. 2016) we do not explicitly optimize over $\boldsymbol{a}^l$. Similar to (Carreira-Perpinan and Wang 2014) and (Taylor et al. 2016), we use alternating minimization. However, we develop an online approach, while both previous approaches are formulated in an offline, batch mode that learns from a whole training dataset rather than incrementally. Such approaches have limited scalability to extremely large datasets (even using parallelization across samples as in (Taylor et al. 2016)), and, more importantly, cannot handle online, continual learning scenarios, unlike the standard backpropagation-based stochastic gradient methods.

Offline Alternating Minimization. We first formulate an offline (batch) objective over n input samples, which differs from both (Carreira-Perpinan and Wang 2014) and (Taylor et al. 2016) in several ways: unlike (Taylor et al. 2016) (and equation 3), we only have one set of auxiliary variables instead of two; unlike (Carreira-Perpinan and Wang 2014), the predictive loss is not required to be quadratic, and auxiliary variables are introduced one step earlier, before the nonlinearity. *This is important for utilizing incremental weight updates similar to those used in the online dictionary learning of (Mairal et al. 2009).* The offline/batch objective function now can be written as a function of both the weights $\boldsymbol{W} = \{\boldsymbol{W}^1, ..., \boldsymbol{W}^{L+1}\}$ and the codes $\boldsymbol{C} = \{\boldsymbol{c}_1^1, ..., \boldsymbol{c}_1^L, ...\boldsymbol{c}_n^1, ..., \boldsymbol{c}_n^L, \}$, for all input samples $t = 1, ..., n$:

$$\begin{aligned} f(\boldsymbol{W}, \boldsymbol{C}) = \quad & \textstyle\sum_{t=1}^{n} \mathcal{L}(y_t, \sigma_L(\boldsymbol{c}_t^L), \boldsymbol{W}^{L+1}) \\ & + \mu \textstyle\sum_{t=1}^{n} \sum_{l=1}^{L} ||\boldsymbol{c}_t^l - \boldsymbol{W}^l \sigma_{l-1}(\boldsymbol{c}_t^{l-1})||_2^2 \\ & + \lambda_W ||\boldsymbol{W}^l||_1 + \lambda_C ||\boldsymbol{C}^l||_1. \end{aligned} \tag{3}$$

This objective can be also viewed as the negative log-likelihood of a Bayesian network with Gaussian hidden nodes and multinomial labels. Also, we add sparsity regularizers on both $\boldsymbol{c}$ and $\boldsymbol{W}$, corresponding to Laplace priors, because empirically, proper sparsity levels tend to improve generalization. The parameter μ (Lagrange multiplier) controls the amount of noise, i.e. the variance of the Gaussian distribution with mean $\boldsymbol{W}^l \sigma_{l-1}(\boldsymbol{c}_t^{l-1})$: higher μ corresponds to lower variance. For a fixed μ, alternating minimization can be now performed similar to (Carreira-Perpinan and Wang 2014) and (Taylor et al. 2016), by *iterating between optimizing the codes and optimizing the weights*. In our current experiments, a constant μ was used, though the adaptive scheme of (Carreira-Perpinan and Wang 2014) which allows $\mu \to \infty$ can provide more flexibility and is one of our future research directions.

Online Alternating Minimization. We assume an *online setting* where the *input samples arrive incrementally, one at a time or in small mini-batches*, and the *previous samples are not stored explicitly*, so that the memory complexity of our approach remains constant w.r.t. the potentially infinite number of samples n, and will only depend on the maximum dimensionality of the input and hidden layers, $\max_l m_l$.

Our approach is summarized in Algorithms 1-3. It takes as an input initial $\boldsymbol{W}$ (e.g., random), and initial memory matrices $\boldsymbol{A_0}$, $\boldsymbol{B_0}$ (described below), typically initialized to all zeros, unless we are in a continual learning setting and would like to remember previous tasks. First, it encodes the input into its representations at each layer (**encodeInput** procedure, Algorithm 2), and makes a prediction based on such encodings. The prediction error is computed, and the backward code updates follow, as shown in the **updateCodes** procedure, where the code vector at layer l is optimized with respect to the only two parts of the global objective that the code variables participate in. Once the codes are updated, we also update the corresponding memory matrices (**updateMemory** procedure), and then proceed with the weight updates (in parallel over the layers) as described below (**updateWeights** procedure in Algorithm 3). We now discuss each step in more detail.

Algorithm 1 Online Alternating Minimization (AM)

Require: $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^N \times S \sim p(\boldsymbol{x}, y)$ (data stream sampled from distribution $p(\boldsymbol{x}, \boldsymbol{y})$; in multi-class classification, $S = \{\boldsymbol{y} \in \mathbb{R}^m | \exists i \; s.t. y_i = 1, \forall j \neq i, y_j = 0\}$ is a one-hot vector; in regression $S = \mathbb{R}$; input dimension N; number of classes m (in classification); number of hidden layers L; dimensions of hidden layers $\{m_l | l = 1, ..., L\}$; initial weights $\boldsymbol{W}_0$; initial memory matrices $\boldsymbol{A_0}$ and $\boldsymbol{B_0}$; $\lambda_C \in \mathbb{R}^+$ (code sparsity); $\lambda_W \in \mathbb{R}^+$ (weight sparsity); $\lambda_C \in \mathbb{R}^+$ (code/activation sparsity);$\mu \in \mathbb{R}^+$ (Lagrange multiplier);$\eta \in \mathbb{R}^+$ (weight update step size).

1: **while** more samples **do**
2: Input $(\boldsymbol{x}_t, y_t)$
3: $\boldsymbol{C} \leftarrow$ **encodeInput**($\boldsymbol{x}_t$, $\boldsymbol{W}_{t-1}$) % forward: compute linear activations at layers $1, ..., L$
4: $\boldsymbol{C} \leftarrow$ **updateCodes**($\boldsymbol{C}$, y_t, $\boldsymbol{W}_{t-1}$, λ_C, μ) % backward: error propagation to update codes
5: $(\boldsymbol{A}_t, \boldsymbol{B}_t) \leftarrow$ **updateMemory**($\boldsymbol{A}_{t-1}$, $\boldsymbol{B}_{t-1}$, $\boldsymbol{C}$)
6: $\boldsymbol{W}_t \leftarrow$ **updateWeights**($\boldsymbol{W}_{t-1}$, $\boldsymbol{A}_t$, $\boldsymbol{B}_t$, $\boldsymbol{x}_t$, y_t, $\boldsymbol{C}$, λ_W, μ, η)
7: **end while**
8: **return** $\boldsymbol{W}_t, \boldsymbol{A}_t, \boldsymbol{B}_t$

Activation propagation: forward and backward passes. Instead of optimizing the function in Eq. 3, which is impossible in the online setting, we will only optimize the codes $\boldsymbol{c}_t^l$ for the current sample $\boldsymbol{x}_t$ at time t, using the weights computed so far. Namely, given the input $\boldsymbol{x}_t$, we compute the last-layer activations $\boldsymbol{a}_t^L = \sigma_L(\boldsymbol{c}_t^L)$ in a forward pass, propagating activations from input to the last layer, and make a prediction about y_t, incurring the loss $\mathcal{L}(y_t, \boldsymbol{a}_t^L, \boldsymbol{W}^{L+1})$. We now propagate this error back to all activations. This is achieved by solving a sequence of optimization problems:

$$\begin{aligned} \boldsymbol{c}^L = \quad & \arg\min_{\boldsymbol{c}} \mathcal{L}(y, \sigma_L(\boldsymbol{c}), \boldsymbol{W}^{L+1}) \\ & + \mu ||\boldsymbol{c} - \boldsymbol{W}^L \sigma_L(\boldsymbol{c}^{L-1})||_2^2 + \lambda_C ||c||_1, \\ \boldsymbol{c}^l = \quad & \arg\min_{\boldsymbol{c}} \mu ||\boldsymbol{c}^{l+1} - \boldsymbol{W}^{l+1} \sigma_l(\boldsymbol{c})||_2^2 \\ & + \mu ||\boldsymbol{c} - \boldsymbol{W}^l \sigma_{l-1}(\boldsymbol{c}^{l-1})||_2^2 + \lambda_C ||c||_1, \end{aligned} \tag{4}$$

for $l = L-1, ..., 1$.

Weights Update Step. Next, assuming the codes are fixed for the current and all previous samples, we optimize the weights using the *surrogate* objective function

$$\hat{f}_t(\boldsymbol{W}) = f(\boldsymbol{W}, \boldsymbol{C}_{hist}),$$

where $\boldsymbol{C}_{hist}$ denotes the set of codes for all previous samples, obtained at all previous iterations, i.e. for previous values of $\boldsymbol{W}$. The difference from the offline setting is that,

Algorithm 2 Activation Propagation (Code Update) Steps

encodeInput($\boldsymbol{x}$, $\boldsymbol{W}$)

$\boldsymbol{c}^0 = \boldsymbol{x}$
for $l = 1$ to L **do**
  $\boldsymbol{c}^l = \boldsymbol{W}^l \sigma_{l-1}(\boldsymbol{c}^{l-1})$
  % $\sigma_0(\boldsymbol{x}) = \boldsymbol{x}, \sigma_l(\boldsymbol{x}) = ReLU(\boldsymbol{x})\ for\ l = 1, ..., L$
end for
return $\boldsymbol{C}$

updateCodes($\boldsymbol{C}, \boldsymbol{y}, \boldsymbol{W}, \lambda_C, \mu$)

$\boldsymbol{c}^L = \arg\min_{\boldsymbol{c}} \mathcal{L}(\boldsymbol{y}, \sigma_L(\boldsymbol{c}), \boldsymbol{W}^{L+1}) + \mu||\boldsymbol{c} - \boldsymbol{W}^L \sigma_{L-1}(\boldsymbol{c}^{L-1})||_2^2 + \lambda_C||c||_1$
$\boldsymbol{c}^0 = \boldsymbol{x}$
for $l = L - 1$ to 1 **do**
  $\boldsymbol{c}^l = \arg\min_{\boldsymbol{c}} \mu||\boldsymbol{c}^{l+1} - \boldsymbol{W}^{l+1}\sigma_l(\boldsymbol{c})||_2^2 + \mu||\boldsymbol{c} - \boldsymbol{W}^l \sigma_{l-1}(\boldsymbol{c}^{l-1})||_2^2 + \lambda_C||c||_1$
end for
return $\boldsymbol{C}$

once the codes are computed, given a set of weights, they will never be recomputed in the future. The surrogate objective decomposes into $L + 1$ independent subproblems, $\hat{f}_t(\boldsymbol{W}) = \sum_{l=1}^{L+1} \hat{f}_t^l(\boldsymbol{W}^l)$, minimizing the corresponding $\hat{f}_t^l(\boldsymbol{W}^l)$ over weights at different layers. *This decomposition allows for parallel optimization of the weights across all layers.* Namely, for the last, predictive layer, we simply solve an online, sparse multinomial regression, for example, using stochastic gradient descent (SGD) in line 1 of the **updateWeights** procedure in Algorithm 3: $\boldsymbol{W}^{L+1} = \arg\min_{\boldsymbol{W}} \hat{f}_t^{L+1}(\boldsymbol{W})$

$$= \arg\min_{\boldsymbol{W}} \sum_{t=1}^{n} \mathcal{L}(y_t, \sigma_L(\boldsymbol{c}_t^L), \boldsymbol{W}) + \lambda_W||\boldsymbol{W}||_1. \quad (5)$$

For layers $l = 1, ..., L$, we have $\boldsymbol{W}^l = \arg\min_{\boldsymbol{W}} \hat{f}_t^l(\boldsymbol{W})$

$$= \arg\min_{\boldsymbol{W}} \mu \sum_{i=1} ||\boldsymbol{c}_i^l - \boldsymbol{W}\sigma_{l-1}(\boldsymbol{c}_i^{l-1})||_2^2 + \lambda_W||\boldsymbol{W}||_1. \quad (6)$$

Co-Activation Memory. Denoting activation in layer l as $\boldsymbol{a}^l = \sigma_l(\boldsymbol{c}^l)$, and following (Mairal et al. 2009), we can rewrite the above objective in Eq. 6 using the following:

$$\sum_{i=1}^{t} ||\boldsymbol{c}_i^l - \boldsymbol{W}\boldsymbol{a}_i^l||_2^2 = Tr(\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{A}_t^l) - 2Tr(\boldsymbol{W}^T\boldsymbol{B}_t^l), \quad (7)$$

where $\boldsymbol{A}_t^l = \sum_{i=1}^t \boldsymbol{a}_i^{l-1}(\boldsymbol{a}_i^{l-1})^T$ and $\boldsymbol{B}_t^l = \sum_{i=1}^t \boldsymbol{c}_i^l(\boldsymbol{a}_i^{l-1})^T$ are the "book-keeping" matrices, which we also refer to as *co-activation memories*, compactly representing the accumulated, over t samples, "strength" of "co-activations" in each layer $l = 1, ..., L$, including the input layer (matrices $\boldsymbol{A_t^l}$) and across consecutive layers (matrices $\boldsymbol{B_t^l}$). At each iteration t, once the new input sample $\boldsymbol{x}_t$ is encoded, the matrices are updated as $\boldsymbol{A_t} \leftarrow \boldsymbol{A_t} + \boldsymbol{a}_t^{l-1}(\boldsymbol{a}_t^{l-1})^T$ and $\boldsymbol{B} \leftarrow \boldsymbol{B_t} + \boldsymbol{c}_t^l(\boldsymbol{a}_t^{l-1})^T$.

It is important to note that, using memory matrices, we are effectively optimizing the weights at iteration t with respect to all previous samples and their previous linear activations at all layers, without the need for an explicit storage of these examples, using only fixed-size memory of $O(M^2)$, where $M = max_{l=0,...,L} m_l$ and m_l is the number of nodes at layer l.

Algorithm 3 Memory and Weight Update Steps

updateMemory($\boldsymbol{A}$, $\boldsymbol{B}, \boldsymbol{C}$)

for $l = 1$ to L **do**
  $\boldsymbol{a} = \sigma_{l-1}(\boldsymbol{c}^{l-1})$
  $\boldsymbol{A}^l \leftarrow \boldsymbol{A}^l + \boldsymbol{a}\boldsymbol{a}^T$
  $\boldsymbol{B}^l \leftarrow \boldsymbol{B}^l + \boldsymbol{c}^l\boldsymbol{a}^T$
end for
return $\boldsymbol{A}, \boldsymbol{B}$

updateWeights($\boldsymbol{W}$, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{x}$, y, $\boldsymbol{C}$, λ_W, μ, η)
% parallelized across layers

$\hat{f}^{L+1}(\boldsymbol{W^{L+1}}) = \mathcal{L}(y, \sigma_L(\boldsymbol{c}^L), \boldsymbol{W^{L+1}}) + \lambda_W||\boldsymbol{W^{L+1}}||_1$
  % Take a proximal step in direction of negative gradient
$\boldsymbol{Z} = \boldsymbol{W^{L+1}} - \eta\nabla_{\boldsymbol{W}}\mathcal{L}(y, \sigma_L(\boldsymbol{c}^L), \boldsymbol{W^{L+1}})$
$\boldsymbol{W}^{L+1} \leftarrow Prox_{\lambda_W||.||_1}(\boldsymbol{Z}) = sgn(\boldsymbol{Z})(|\boldsymbol{Z}| - \lambda_W)_+$
for $l = 1$ to L **do**
  % internal layers: $\hat{f}^l(\boldsymbol{W^L}) \equiv \mu\big(Tr(\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{A}^l) - 2Tr(\boldsymbol{W}^T\boldsymbol{B}^l)\big) + \lambda_W||\boldsymbol{W}||_1$
  $\boldsymbol{W}^l \leftarrow BCD(\boldsymbol{W^l}, \boldsymbol{A^l}, \boldsymbol{B^l}, \frac{\lambda_W}{\mu})$
end for
return $\boldsymbol{W}$

BCD($\boldsymbol{W^l}, \boldsymbol{A^l}, \boldsymbol{B^l}, \lambda$)

$m_l = \#$ of columns in $\boldsymbol{W}^l$
repeat
  for $j = 1$ to m_l **do**
    $\boldsymbol{u}_j \leftarrow \frac{\boldsymbol{b}_j - \sum_{k \neq j} \boldsymbol{w}_k a_{jk}}{a_{jj}}$ % $\boldsymbol{b}_j$ - j-th column of $\boldsymbol{B^l}$, $\boldsymbol{w}_k$ - k-th column of $\boldsymbol{W^l}$, a_{jk} from $\boldsymbol{A^l}$
    $\boldsymbol{w}_j \leftarrow Prox_{\lambda||.||_1}(\boldsymbol{u}_j) = sgn(\boldsymbol{u}_j)(|\boldsymbol{u}_j| - \lambda)_+$
  end for
until convergence
return $\boldsymbol{W^l}$

Weight optimization with Block-Coordinate Descent (BCD). We follow (Mairal et al. 2009) and use *block coordinate descent* to optimize the convex objective in Eq. 7; it iterates over dictionary elements in a fixed sequence, until convergence, optimizing each while keeping the others fixed, except that, in our approach, we add a proximal operator (line 6 of BCD) to enforce l_1-norm regularization.

Weight optimization with Stochastic Gradient Descent (SGD). While BCD solves exactly an approximate problem, i.e. optimizes the surrogate function $\hat{f}_t$ approximating the true objective f_t, as described above, another approximation approach can be to use instead an approximate (e.g., stochastic gradient) algorithm for solving the exact optimization problem. We explore this option as well, simply replacing

step 6 in **updateWeights** by a stochastic gradient step similar to the top layer update of $\boldsymbol{W}^{L+1}$ in lines 2 and 3 of the same function. We refer to this version of our approach as *online AM-SGD*, while the BCD-based version discussed earlier is called *online AM-BCD*. However, we are currently not using sparsity regularizers within the SGD-based updates, although incorporating them is easy by switching to a proximal method instead of simple gradient descent.

3 Theoretical analysis

We next provide a theoretical convergence analysis for general alternating minimization (AM) schemes. Under certain assumptions that we will discuss, the proposed AM algorithm(s) falls into the category of approaches that comply with these guarantees, though the theory itself is more general and novel. To the best of our knowledge, we provide the first theoretical convergence guarantees of AM in the stochastic setting.

Setting. Let in general $\hat{f}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$ denote the function to be optimized using AM, where in the i^{th} step of the algorithm, we optimize $\hat{f}$ with respect to $\boldsymbol{\theta}_i$ and keep other arguments fixed. Let K denote total number of arguments. For the theoretical analysis, we consider a smooth approximation to $\hat{f}$ as done in the literature (Schmidt, Fung, and Rosales 2007; Lange et al. 2014).

Let $\{\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_K^*\}$ denote the global optimum of $\hat{f}$ computed on the entire data population. For the sake of the theoretical analysis we assume that the algorithm knows the lower-bound on the radii of convergence $r_1, r_2, \ldots, r_K$ for $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K$.[1] Let $\nabla_{\boldsymbol{\theta}_i} \hat{f}^1$ denote the gradient of $\hat{f}$ computed for a single data sample $(\boldsymbol{x}, \boldsymbol{y})$ or code $\boldsymbol{c}$. In the next section, we refer to $\nabla_{\boldsymbol{\theta}_i} \hat{f}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$ as the gradient of $\hat{f}$ with respect to $\boldsymbol{\theta}_i$ computed for the entire data population, i.e. an infinite number of samples ("oracle gradient"). We assume in the i^{th} step, the AM algorithm performs the update:

$$\boldsymbol{\theta}_i = \Pi_i(\boldsymbol{\theta}_i - \eta^\tau \nabla_{\boldsymbol{\theta}_i} \hat{f}^1(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)), \tag{8}$$

where Π_i denotes the projection onto the Euclidean ball $B_2(\frac{r_i}{2}, \boldsymbol{\theta}_i^0)$ of some given radius $\frac{r_i}{2}$ centered at the initial iterate $\boldsymbol{\theta}_i^0$. Thus, given any initial vector $\boldsymbol{\theta}_i^0$ in the ball of radius $\frac{r_i}{2}$ centered at $\boldsymbol{\theta}_i^*$, we are guaranteed that all iterates remain within an r_i-ball of $\boldsymbol{\theta}_i^*$. This is true for all $i = 1, 2, \ldots, K$.

This scheme is *much more difficult* to prove theoretically and leads to *the worst-case theoretical guarantees* with respect to the original setting from Algorithm 1, i.e. we expect the convergence rate for the original setting to be no worse than the one dictated by the obtained guarantees. This is because we allow only a single stochastic update (i.e. computed on a single data point) with respect to an appropriate argument (when keeping other arguments fixed) in each step of AM, whereas in Algorithm 1 and related schemes in the literature, one may increase the size of the data mini-batch in each AM step (semi-stochastic setting). The convergence rate in the latter case is typically more advantageous (Nesterov 2014). Finally, note that the analysis does not consider running the optimizer more than once before changing the argument of an update, e.g., when obtaining sparse code $\boldsymbol{c}$ for a given data point $(\boldsymbol{x}, \boldsymbol{y})$ and fixed dictionary. We expect this to have a minor influence on the convergence rate as our analysis specifically considers a local convergence regime, where we expect that running the optimizer once produces good enough parameter approximations. Moreover, note that by preventing each AM step to be run multiple times, we analyze more noisy version of parameter updates.

[1] This assumption is potentially easy to eliminate with a more careful choice of the step size in the first iterations.

Statistical guarantees for AM algorithms. The theoretical analysis we provide here is an extension to the AM setting of recent work on statistical guarantees for the EM algorithm (Balakrishnan, Wainwright, and Yu 2017).

We first discuss necessary assumptions that we make. Let $L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K) = -\hat{f}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$ and denote $L_d^*(\boldsymbol{\theta}_d) = L(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_{K-1}^*, \boldsymbol{\theta}_K^*)$. Let $\Omega_1, \Omega_2, \ldots, \Omega_K$ denote non-empty compact convex sets such that for any $i = \{1, 2, \ldots, K\}, \boldsymbol{\theta}_i \in \Omega_i$. The following three assumptions are made on $L_d^*(\boldsymbol{\theta}_d)$ and objective function $L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$.

Assumption 3.1 (Strong concavity)**.** *The function $L_d^*(\boldsymbol{\theta}_d)$ is strongly concave for all pairs $(\boldsymbol{\theta}_{d,1}, \boldsymbol{\theta}_{d,2})$ in the neighborhood of $\boldsymbol{\theta}_d^*$. That is*

$$L_d^*(\boldsymbol{\theta}_{d,1}) - L_d^*(\boldsymbol{\theta}_{d,2}) - \langle \nabla_{\boldsymbol{\theta}_d} L_d^*(\boldsymbol{\theta}_{d,2}), \boldsymbol{\theta}_{d,1} - \boldsymbol{\theta}_{d,2} \rangle \leq -\frac{\lambda_d}{2} \|\boldsymbol{\theta}_{d,1} - \boldsymbol{\theta}_{d,2}\|_2^2,$$

where $\lambda_d > 0$ is the strong concavity modulus.

Assumption 3.2 (Smoothness)**.** *The function $L_d^*(\boldsymbol{\theta}_d)$ is μ_d-smooth for all pairs $(\boldsymbol{\theta}_{d,1}, \boldsymbol{\theta}_{d,2})$. That is*

$$L_d^*(\boldsymbol{\theta}_{d,1}) - L_d^*(\boldsymbol{\theta}_{d,2}) - \langle \nabla_{\boldsymbol{\theta}_d} L_d^*(\boldsymbol{\theta}_{d,2}), \boldsymbol{\theta}_{d,1} - \boldsymbol{\theta}_{d,2} \rangle \geq -\frac{\mu_d}{2} \|\boldsymbol{\theta}_{d,1} - \boldsymbol{\theta}_{d,2}\|_2^2,$$

where $\mu_d > 0$ is the smoothness constant.

Next, we introduce the gradient stability (GS) condition that holds for any d from 1 to k.

Assumption 3.3 (Gradient stability (GS))**.** *We assume $L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \ldots, \boldsymbol{\theta}_K)$ satisfies GS (γ_d) condition, where $\gamma_d \geq 0$, over Euclidean balls $\boldsymbol{\theta}_1 \in B_2(r_1, \boldsymbol{\theta}_1^*), \ldots, \boldsymbol{\theta}_{d-1} \in B_2(r_{d-1}, \boldsymbol{\theta}_{d-1}^*), \boldsymbol{\theta}_{d+1} \in B_2(r_{d+1}, \boldsymbol{\theta}_{d+1}^*), \ldots, \boldsymbol{\theta}_K \in B_2(r_K, \theta_K^*)$ of the form*

$$\|\nabla_{\boldsymbol{\theta}_d} L_d^*(\boldsymbol{\theta}_d) - \nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)\|_2 \leq \gamma_d \sum_{\substack{i=1 \\ i \neq d}}^{K} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_i^*\|_2.$$

Next, we introduce the *population gradient AM operator*, $\mathcal{G}_i(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$, where $i = 1, 2, \ldots, K$, defined as

$$\mathcal{G}_i(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K) \coloneqq \boldsymbol{\theta}_i + \eta \nabla_{\boldsymbol{\theta}_i} \hat{f}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K),$$

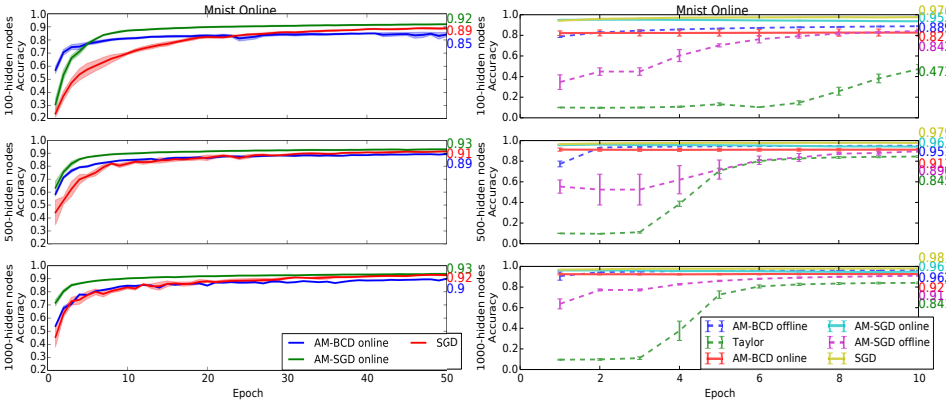

Figure 1: MNIST: online methods, a zoom-in for the first epoch; online AM-BCD, AM-SGD and backprop-SGD over 50 mini-batches of 200 samples each; 2-hidden-layer network architectures with the same number of units in each hidden layer, from 100 (top) to 500 (middle), and 1000 (bottom) hidden units; hyperparameters for each method are optimized via grid search.

Figure 2: MNIST: online vs. offline, 10 epochs. Similar to Figure 1, but showing accuracy on the test set over the course of 10 epochs. Online methods (AM-BCD, AM-SGD and backprop-SGD) are also compared to offline/batch versions of AM-BCD and AM-SGD, trained on the whole dataset as a (mini)batch, repeated 10 times, and to Taylor's offline method.

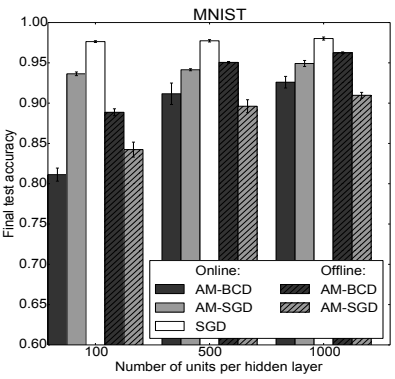

Figure 3: MNIST: the effects of the network width on algorithm's performance. The test set accuracy results are shown for both online and offline methods for hidden layer sizes ranging from 100 to 1000 hidden units.

where η is the step size. We also define the following bound σ on the expected value of the gradient of our objective function (a common assumption made in stochastic gradient descent convergence theorems as well). Define $\sigma = \sqrt{\sum_{d=1}^{K} \sigma_d^2}$ where

$$\sigma_d^2 = \sup\{\mathbb{E}[\|\nabla_{\boldsymbol{\theta}_d} L_1(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)\|_2^2] : \boldsymbol{\theta}_1 \in B_2(r_1, \boldsymbol{\theta}_1^*) \ldots \boldsymbol{\theta}_K \in B_2(r_k, \boldsymbol{\theta}_k^*)\}$$

The following theorem then gives a recursion on the expected error obtained at each iteration of Algorithm 1.

Theorem 3.1. *Given the stochastic AM gradient iterates of Algorithm 1 with decaying step size $\{\eta^t\}_{t=0}^{\infty}$, for any $d = 1, 2, \ldots, K$ and $\gamma < \frac{2\xi}{3(K-1)}$ the error $\boldsymbol{\Delta}_d^{t+1} := \boldsymbol{\theta}_d^{t+1} - \boldsymbol{\theta}_d^*$ at iteration $t+1$ satisfies recursion*

$$\mathbb{E}\left[\sum_{d=1}^{K} \|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] \leq (1 - q^t)\mathbb{E}\left[\sum_{d=1}^{K} \|\boldsymbol{\Delta}_d^{t}\|_2^2\right] + \frac{(\eta^t)^2}{1 - (K-1)\eta^t\gamma}\sigma^2, \quad (9)$$

where $q^t = 1 - \frac{1 - 2\eta^t\xi + 2\eta^t\gamma(K-1)}{1-(K-1)\eta^t\gamma}$.

The recursion in Theorem 3.1 is expanded in the Supplementary Material to prove the final convergence theorem for Algorithm 1 which states the following:

Theorem 3.2. *Given the stochastic AM gradient iterates of Algorithm 1 with decaying step size* $\eta^t = \frac{3/2}{[2\xi - 3\gamma(K-1)](t+2) + \frac{3}{2}(K-1)\gamma}$ *and assuming that* $\gamma < \frac{2\xi}{3(K-1)}$, *the error* $\|\boldsymbol{\Delta}^{t+1}\|_2 := \sqrt{\sum_{d=1}^{K} \|\boldsymbol{\Delta}_d^{t+1}\|_2^2} = \sqrt{\sum_{d=1}^{K} \|\boldsymbol{\theta}_d^{t+1} - \boldsymbol{\theta}_d^*\|_2^2}$ *at iteration* $t+1$ *satisfies*

$$\mathbb{E}\left[\|\boldsymbol{\Delta}^{t+1}\|_2^2\right] \leq \mathbb{E}\left[\sum_{d=1}^{K} \|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}} + \sigma^2 \frac{9}{[2\xi - 3\gamma(K-1)]^2(t+3)} \quad (10)$$

4 Experiments

We evaluated performance of the proposed online alternating minimization algorithms, *AM-BCD* and *AM-SGD* on several datasets, comparing our methods versus standard backpropagation with SGD (*SGD*), the offline alternating minimization method of Taylor et al (Taylor et al. 2016), as well as our own offline alternating minimization versions, or *offline-AM-BCD* and *offline-AM-SGD*. In offline versions, the whole task (dataset) is treated as one batch, so that the algorithm performs one iteration over the activation optimization followed by one iterations of weight updates and then moves on to the next batch, which in the offline case is the same dataset, i.e. the next epoch. In all experiments, all online algorithms use a fixed mini-batch size of 200 samples (instead of single sample). We used grid search to optimize hyperparameters such as initial learning rate (which was adaptive in all SGD-involving methods, using Adam), as well as the sparsity weights; overall, we observed that denser networks performed better in our relatively small architectures, so in most of the reported results, sparsity weights were zero.

We first experimented with the standard MNIST dataset, consisting of 28×28 gray-scale images of hand-drawn digits, with 50K samples, and a test set of 10K samples. We evaluated three different 2-hidden-layer architectures, with equal

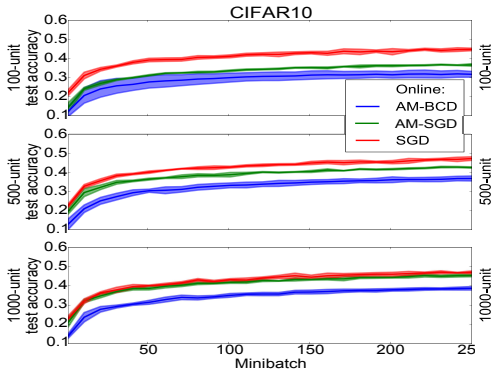

Figure 4: CIFAR10: online methods, 1st epoch; similar experiment (methods, architectures) to Figure 1, but for 250 minibatches of 200 samples each (the entire CIFAR10 training set).

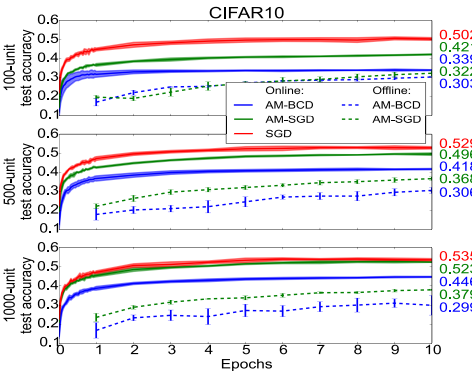

Figure 5: CIFAR10: online vs. offline, 10 epochs. Similar experiments to Figure 2.

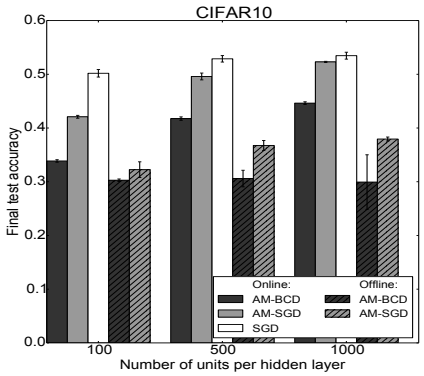

Figure 6: CIFAR10: the effects of the network width on algorithm's performance, similar to Figure 3.

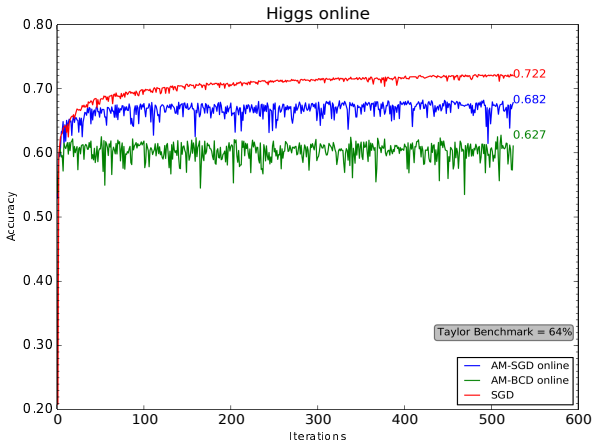

Figure 7: HIGGS: online methods, one epoch.

hidden layer sizes of 100, 500 and 1000. Figure 1 zoomed-in on the performance of the online *AM-BCD*, *AM-SGD* and (*SGD*) over 50 minibatches of size 200 each. We observe that, on all three architectures, both online AM methods, AM-BCD and AM-SGD (and especially the latter) considerably outperform SGD at early stages, although later on SGD catches up with them; note that this is only the first 50 x 200 = 10,000 samples, so the final accuracy is still low for MNIST. Next, in Figure 2, we plot the test error after training on 10 epochs over the whole dataset. The top performers are SGD and AM-SGD, achieving, depending on the architecture, between 0.95 and 0.98 accuracy[2]. We also see that all algorithms outperform Taylor's approach, sometimes by far, and even in best case architecture for Taylor (1000×1000, bottom plot), it only achieves 0.84 accuracy, while online AM-SGD and AM-BCD achieve 0.96 and 0.92 respectively. We also note that the offline AM-BCD actually outperforms quite noticeably the offline AM-SGD, while the order is reversed for their online counterparts. Finally, Figure 3 demonstrates that, while the width of the network does not seem to have much effect on the performance of SGD and online AM-SGD, the performance of AM-BCD (offline and online) clearly improves with increasing width; this is also the case for offline AM-SGD.

The CIFAR-10 dataset is a subset of the 80 Million Tiny Images Torralba et al. (2008), and contains 10 balanced classes. It provides a training set with 50000 samples and a test set of 10000 samples. Each sample is a color image with 32×32 RGB pixels. In Figures 4-6, we show the results of our experiments on CIFAR-10, using similar methodology to MNIST experiments above[3]. Namely, we start with comparing all online methods in Figure 4 (one epoch over the whole datset); unlike MNIST, SGD dominates here, although for wider architectures, online AM-SGD is practically indistinguishable (note that the results are averaged over several initializations which explains error bars, or a wider line, around the plots). In Figure 5, similarly to Figure 2, we compare all online methods to the two offline AM versions, over 10 epochs over the whole datset, and can clearly see that online AM significantly outperforms its offline counterparts. Figure 6 shows the effect of architecture width: increasing seems to benefit most algorithms we considered.

Finally, in Figure 7 we present our experiments on a very large datatset, HIGGS, comprising 11M datapoints of 28 features each, with each datapoint labeled as either a signal process producing a Higgs boson or a background process which does not. We use a single-hidden layer network with ReLU activations and 300 hidden nodes, as suggested in (Baldi, Sadowski, and Whiteson 2014), in order to compare our results with the Taylor's offline alternating minimization (Taylor et al. 2016) on exactly same architecture

[2]note that we are using fully connected, not convolutional, networks, and thus are not achieving state-of-art results on MNIST; the point is to compare different algorithms on identical architectures.

[3]Note, again, simple fully connected architectures are not expected to reach convnet state-of-art results here; we comparing state-of-art Adam SGD vs AM but on the same simple architecture.

and data. We ran SGD, online AM-SGD and online AM-BCD on 10.5M training samples, and achieved, respectively, 0.72, 0.68 and 0.63 test accuracy, outperforming Taylor's method achieving 0.64 accuracy. Note that the attractive part of online methods is their ability to scale (and, apparently, perform accurately) on arbitrarily large datasets, unlike offline/batch methods such as Taylor's, which required massive parallelization across data (not layers) to be able to process this datset. Note also that both SGD and online AM-SGD tend to asymptote much earlier than they reach the end of the dataset, outperforming Taylor's method on a much smaller subset of data. (We also ran offline AM-BCD offline over 2M points, or about 19% of the data, achieving 60% in 10 epochs.)

In summary, proposed online AM methods are competitive with online backprop via SGD (Adam), noticeably outperforming SGD on some datasets early on (faster learners), though SGD catches up later; moreover, they clearly outperform both our offline versions and offline alternating minimization algorithm such as (Taylor et al. 2016).

References

[Askari et al. 2018] Askari, A.; Negiar, G.; Sambharya, R.; and El Ghaoui, L. 2018. Lifted neural networks. arXiv:1805.01532 [cs.LG].

[Balakrishnan, Wainwright, and Yu 2017] Balakrishnan, S.; Wainwright, M. J.; and Yu, B. 2017. Statistical guarantees for the em algorithm: From population to sample-based analysis. *Ann. Statist.* 45(1):77–120.

[Baldi, Sadowski, and Whiteson 2014] Baldi, P.; Sadowski, P.; and Whiteson, D. 2014. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications* (5).

[Bartunov et al. 2018] Bartunov, S.; Santoro, A.; Richards, B. A.; Hinton, G. E.; and Lillicrap, T. 2018. Assessing the scalability of biologically-motivated deep learning algorithms and architectures.

[Bengio, Simard, and Frasconi 1994] Bengio, Y.; Simard, P.; and Frasconi, P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5(2):157–166.

[Carreira-Perpinan and Wang 2014] Carreira-Perpinan, M., and Wang, W. 2014. Distributed optimization of deeply nested systems. In *Artificial Intelligence and Statistics*, 10–19.

[Courbariaux, Bengio, and David 2015] Courbariaux, M.; Bengio, Y.; and David, J.-P. 2015. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, 3123–3131.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

[Hubara et al. 2016] Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; and Bengio, Y. 2016. Binarized neural networks. In *Advances in neural information processing systems*, 4107–4115.

[Lange et al. 2014] Lange, M.; Zühlke, D.; Holz, O.; and Villmann, T. 2014. Applications of lp-norms and their smooth approximations for gradient based learning vector quantization. In *ESANN*.

[Le Cun 1986] Le Cun, Y. 1986. Learning process in an asymmetric threshold network. In *Disordered systems and biological organization*. Springer. 233–240.

[LeCun et al. 1988] LeCun, Y.; Touresky, D.; Hinton, G.; and Sejnowski, T. 1988. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann.

[Lee et al. 2015] Lee, D.-H.; Zhang, S.; Fischer, A.; and Bengio, Y. 2015. Difference target propagation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 498–515. Springer.

[Mairal et al. 2009] Mairal, J.; Bach, F.; Ponce, J.; and Sapiro, G. 2009. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*.

[Nair and Hinton 2010] Nair, V., and Hinton, G. E. 2010. Rectified linear units improve Restricted Boltzmann Machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

[Nesterov 2014] Nesterov, Y. 2014. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition.

[Riedmiller and Braun 1993] Riedmiller, M., and Braun, H. 1993. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Neural Networks, 1993., IEEE International Conference on*, 586–591. IEEE.

[Schmidt, Fung, and Rosales 2007] Schmidt, M.; Fung, G.; and Rosales, R. 2007. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In Kok, J. N.; Koronacki, J.; Mantaras, R. L. d.; Matwin, S.; Mladenič, D.; and Skowron, A., eds., *ECML*.

[Taylor et al. 2016] Taylor, G.; Burmeister, R.; Xu, Z.; Singh, B.; Patel, A.; and Goldstein, T. 2016. Training neural networks without gradients: A scalable admm approach. In *International conference on machine learning*, 2722–2731.

[Yann 1987] Yann, L. 1987. *Modèles connexionnistes de l'apprentissage*. Ph.D. Dissertation, PhD thesis, These de Doctorat, Universite Paris 6.

[Zeng et al. 2018] Zeng, J.; Lau, T. T.-K.; Lin, S.; and Yao, Y. 2018. Block coordinate descent for deep learning: Unified convergence guarantees. *arXiv preprint arXiv:1803.00225*.

[Zhang and Brand 2017] Zhang, Z., and Brand, M. 2017. Convergent block coordinate descent for training Tikhonov regularized deep neural networks. In *Advances in Neural Information Processing Systems*, 1719–1728.

[Zhang and Kleijn 2017] Zhang, G., and Kleijn, W. B. 2017. Training deep neural networks via optimization over graphs. arXiv:1702.03380 [cs.LG].

[Zhang, Chen, and Saligrama 2016] Zhang, Z.; Chen, Y.; and Saligrama, V. 2016. Efficient training of very deep neural networks for supervised hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1487–1495.

Supplemental Material

A Proofs

Proof of Theorem 3.2 relies on Theorem 3.1, which in turn relies on Theorem A.1 and Lemma A.1, both of which are stated below. Proofs of the lemma and theorems follow in the subsequent subsections.

The next result is a standard result from convex optimization (Theorem 2.1.14 in (Nesterov 2014)) and is used in the proof of Theorem A.1 below.

Lemma A.1. *For any $d = 1, 2, \ldots, K$, the gradient operator $\mathcal{G}_d(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_{K-1}^*, \boldsymbol{\theta}_K^*)$ under Assumption 3.1 (strong concavity) and Assumption 3.2 (smoothness) with constant step size choice $0 < \eta \leq \frac{2}{\mu_d + \lambda_d}$ is contractive, i.e.*

$$\|\mathcal{G}_d(\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_K^*) - \boldsymbol{\theta}_d^*\|_2 \leq \left(1 - \frac{2\eta\mu_d\lambda_d}{\mu_d + \lambda_d}\right) \|\boldsymbol{\theta}_d - \boldsymbol{\theta}_d^*\|_2 \tag{11}$$

for all $\boldsymbol{\theta}_d \in B_2(r_d, \boldsymbol{\theta}_d^*)$.

The next theorem also holds for any d from 1 to K. Let $r_1, \ldots, r_{d-1}, r_{d+1}, \ldots, r_K > 0$ and $\boldsymbol{\theta}_1 \in B_2(r_1, \boldsymbol{\theta}_1^*), \ldots, \boldsymbol{\theta}_{d-1} \in B_2(r_{d-1}, \boldsymbol{\theta}_{d-1}^*), \boldsymbol{\theta}_{d+1} \in B_2(r_{d+1}, \boldsymbol{\theta}_{d+1}^*), \ldots, \boldsymbol{\theta}_K \in B_2(r_k, \boldsymbol{\theta}_K^*)$.

Theorem A.1. *For some radius $r_d > 0$ and a triplet $(\gamma_d, \lambda_d, \mu_d)$ such that $0 \leq \gamma_d < \lambda_d \leq \mu_d$, suppose that the function $L(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_{K-1}^*, \boldsymbol{\theta}_K^*)$ is λ_d-strongly concave (Assumption 3.1) and μ_d-smooth (Assumption 3.2), and that the GS (γ_d) condition of Assumption 3.3 holds. Then the population gradient AM operator $\mathcal{G}_d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$ with step η such that $0 < \eta \leq \min_{i=1,2,\ldots,K} \frac{2}{\mu_i + \lambda_i}$ is contractive over a ball $B_2(r_d, \boldsymbol{\theta}_d^*)$, i.e.*

$$\|\mathcal{G}_d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K) - \boldsymbol{\theta}_d^*\|_2 \leq (1 - \xi\eta)\|\boldsymbol{\theta}_d - \boldsymbol{\theta}_d^*\|_2 + \eta\gamma \sum_{\substack{i=1 \\ i \neq d}}^{K} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_i^*\|_2 \tag{12}$$

where $\gamma \coloneqq \max_{i=1,2,\ldots,K} \gamma_i$, and $\xi \coloneqq \min_{i=1,2,\ldots,K} \frac{2\mu_i\lambda_i}{\mu_i + \lambda_i}$.

A.1 Proof of Theorem A.1

$\|\mathcal{G}_d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K) - \boldsymbol{\theta}_d^*\|_2 = \|\boldsymbol{\theta}_d + \eta\nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \ldots, \boldsymbol{\theta}_K) - \boldsymbol{\theta}_d^*\|_2$
by the triangle inequality we further get
$\leq \|\boldsymbol{\theta}_d + \eta\nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_K^*) - \boldsymbol{\theta}_d^*\|_2$
$+\eta\|\nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d, \ldots, \boldsymbol{\theta}_K)$
$-\nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_K^*)\|_2$
by the contractivity of T from Equation 11 from Lemma A.1 and GS condition

$$\leq \left(1 - \frac{2\eta\mu_d\lambda_d}{\mu_d + \lambda_d}\right) \|\theta_d - \theta^*\|_2 + \eta\gamma_d \sum_{\substack{i=1 \\ i \neq d}}^{K} \|\theta_i - \theta_i^*\|_2.$$

A.2 Proof of Theorem 3.1

Let $\boldsymbol{\theta}_d^{t+1} = \Pi_d(\tilde{\boldsymbol{\theta}}_d^{t+1})$, where $\tilde{\boldsymbol{\theta}}_d^{t+1} \coloneqq \boldsymbol{\theta}_d^t + \eta^t \nabla_{\boldsymbol{\theta}_d} L^1(\boldsymbol{\theta}_1^{t+1}, \boldsymbol{\theta}_2^{t+1}, \ldots, \boldsymbol{\theta}_{d-1}^{t+1}, \boldsymbol{\theta}_d^t, \boldsymbol{\theta}_{d+1}^t, \ldots, \boldsymbol{\theta}_K^t)$, where $\nabla_{\boldsymbol{\theta}_d} L^1$ is the gradient computed with respect to a single data sample, is the update vector prior to the projection onto a ball $B_2(\frac{r_d}{2}, \boldsymbol{\theta}_d^0)$. Let $\boldsymbol{\Delta}_d^{t+1} \coloneqq \boldsymbol{\theta}_d^{t+1} - \boldsymbol{\theta}_d^*$ and $\tilde{\boldsymbol{\Delta}}_d^{t+1} \coloneqq \tilde{\boldsymbol{\theta}}_d^{t+1} - \boldsymbol{\theta}_d^*$. Thus

$$\begin{aligned} \|\boldsymbol{\Delta}_d^{t+1}\|_2^2 - \|\boldsymbol{\Delta}_d^t\|_2^2 &\leq \|\tilde{\boldsymbol{\Delta}}_d^{t+1}\|_2^2 - \|\boldsymbol{\Delta}_d^t\|_2^2 \\ &= \|\tilde{\boldsymbol{\theta}}_d^{t+1} - \boldsymbol{\theta}_d^*\| - \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\| \\ &= \left\langle \tilde{\boldsymbol{\theta}}_d^{t+1} - \boldsymbol{\theta}_d^t, \tilde{\boldsymbol{\theta}}_d^{t+1} + \boldsymbol{\theta}_d^t - 2\boldsymbol{\theta}_d^* \right\rangle. \end{aligned}$$

Let $\hat{\boldsymbol{W}}_d^t \coloneqq \nabla_{\boldsymbol{\theta}_d} L^1(\boldsymbol{\theta}_1^{t+1}, \boldsymbol{\theta}_2^{t+1}, \ldots, \boldsymbol{\theta}_{d-1}^{t+1}, \boldsymbol{\theta}_d^t, \boldsymbol{\theta}_{d+1}^t, \ldots, \boldsymbol{\theta}_K^t)$. Then we have that $\tilde{\boldsymbol{\theta}}_d^{t+1} - \boldsymbol{\theta}_d^t = \eta^t \hat{\boldsymbol{W}}_d^t$. We combine it with Equation 13 and obtain:

$$\begin{aligned} & \|\boldsymbol{\Delta}_d^{t+1}\|_2^2 - \|\boldsymbol{\Delta}_d^t\|_2^2 \\ \leq & \left\langle \eta^t \hat{\boldsymbol{W}}_d^t, \eta^t \hat{\boldsymbol{W}}_d^t + 2(\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*) \right\rangle \\ = & (\eta^t)^2 (\hat{\boldsymbol{W}}_d^t)^\top \hat{\boldsymbol{W}}_d^t + 2\eta^t (\hat{\boldsymbol{W}}_d^t)^\top (\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*) \\ = & (\eta^t)^2 \|\hat{\boldsymbol{W}}_d^t\|_2^2 + 2\eta^t \left\langle \hat{\boldsymbol{W}}_d^t, \boldsymbol{\Delta}_d^t \right\rangle. \end{aligned}$$

Let $\boldsymbol{W}_d^t \coloneqq \nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1^{t+1}, \boldsymbol{\theta}_2^{t+1}, \ldots, \boldsymbol{\theta}_{d-1}^{t+1}, \boldsymbol{\theta}_d^t, \boldsymbol{\theta}_{d+1}^t, \ldots, \boldsymbol{\theta}_K^t)$. Recall that $\mathbb{E}[\hat{\boldsymbol{W}}_d^t] = \boldsymbol{W}_d^t$. By the properties of martingales, i.e. iterated expectations and tower property:

$$\mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] \leq \mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + (\eta^t)^2 \mathbb{E}[\|\hat{\boldsymbol{W}}_d^t\|_2^2] + 2\eta^t \mathbb{E}[\langle \boldsymbol{W}_d^t, \boldsymbol{\Delta}_d^t \rangle] \tag{13}$$

Let $\boldsymbol{W}_d^* \coloneqq \nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_K^*)$. By self-consistency, i.e. $\boldsymbol{\theta}_d^* = \arg\max_{\boldsymbol{\theta}_d \in \Omega_d} L(\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_K^*)$ and convexity of Ω_d we have that

$$\left\langle \boldsymbol{W}_d^*, \boldsymbol{\Delta}_d^t \right\rangle = \left\langle \nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_K^*), \boldsymbol{\Delta}_d^t \right\rangle \leq 0.$$

Combining this with Equation 13 we have

$$\mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] \leq \mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + (\eta^t)^2 \mathbb{E}[\|\hat{\boldsymbol{W}}_d^t\|_2^2] + 2\eta^t \mathbb{E}[\langle \boldsymbol{W}_d^t - \boldsymbol{W}_d^*, \boldsymbol{\Delta}_d^t \rangle].$$

Define $\mathcal{G}_d^t \coloneqq \boldsymbol{\theta}_d^t + \eta^t \boldsymbol{W}_d^t$ and $\mathcal{G}_d^{t*} \coloneqq \boldsymbol{\theta}_d^* + \eta^t \boldsymbol{W}_d^*$. Thus

$$\begin{aligned} & \eta^t \left\langle \boldsymbol{W}_d^t - \boldsymbol{W}_d^*, \boldsymbol{\Delta}_d^t \right\rangle \\ = & \left\langle \mathcal{G}_d^t - \mathcal{G}_d^{t*} - (\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*), \boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^* \right\rangle \\ = & \left\langle \mathcal{G}_d^t - \mathcal{G}_d^{t*}, \boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^* \right\rangle - \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\|_2^2 \end{aligned}$$

by the fact that $\mathcal{G}_d^{t*} = \boldsymbol{\theta}_d^* + \eta^t \boldsymbol{W}_d^* = \boldsymbol{\theta}_d^*$ (since $\boldsymbol{W}_d^* = 0$):

$$= \left\langle \mathcal{G}_d^t - \boldsymbol{\theta}_d^*, \boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^* \right\rangle - \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\|_2^2$$

by the contractivity of $\mathcal{G}^t$ from Theorem A.1:

$$\begin{aligned} \leq & \left\{ (1 - \eta^t \xi) \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\| + \eta^t \gamma \left(\sum_{i=1}^{d-1} \|\boldsymbol{\theta}_i^{t+1} - \boldsymbol{\theta}_i^*\|_2 + \sum_{i=d+1}^{K} \|\boldsymbol{\theta}_i^t - \boldsymbol{\theta}_i^*\|_2 \right) \right\} \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\|_2 - \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\|_2^2 \\ \leq & \left\{ (1 - \eta^t \xi) \|\boldsymbol{\Delta}_d^t\|_2 + \eta^t \gamma \left(\sum_{i=1}^{d-1} \|\boldsymbol{\Delta}_i^{t+1}\|_2 + \sum_{i=d+1}^{K} \|\boldsymbol{\Delta}_i^t\|_2 \right) \right\} \cdot \|\boldsymbol{\Delta}_d^t\|_2 - \|\boldsymbol{\Delta}_d^t\|_2^2 \end{aligned}$$

Combining this result with Equation 14 gives

$$\begin{aligned} \mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] \leq & \mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + (\eta^t)^2 \mathbb{E}[\|\hat{\boldsymbol{W}}_d^t\|_2^2] + 2\mathbb{E}\left[\left\{ (1 - \eta^t \xi) \|\boldsymbol{\Delta}_d^t\|_2 + \eta^t \gamma \left(\sum_{i=1}^{d-1} \|\boldsymbol{\Delta}_i^{t+1}\|_2 + \sum_{i=d+1}^{K} \|\boldsymbol{\Delta}_i^t\|_2 \right) \right\} \right. \\ & \left. \cdot \|\boldsymbol{\Delta}_d^t\|_2 - \|\boldsymbol{\Delta}_d^t\|_2^2 \right] \\ \leq & \mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + (\eta^t)^2 \sigma_d^2 + 2\mathbb{E}\left[\left\{ (1 - \eta^t \xi) \|\boldsymbol{\Delta}_d^t\|_2 + \eta^t \gamma \left(\sum_{i=1}^{d-1} \|\boldsymbol{\Delta}_i^{t+1}\|_2 + \sum_{i=d+1}^{K} \|\boldsymbol{\Delta}_i^t\|_2 \right) \right\} \right. \\ & \left. \cdot \|\boldsymbol{\Delta}_d^t\|_2 - \|\boldsymbol{\Delta}_d^t\|_2^2 \right], \quad \text{where} \end{aligned}$$

$$\sigma_d^2 = \sup_{\substack{\boldsymbol{\theta}_1 \in B_2(r_1, \boldsymbol{\theta}_1^*) \\ \ldots \\ \boldsymbol{\theta}_K \in B_2(r_K, \boldsymbol{\theta}_K^*)}} \mathbb{E}[\|\nabla_{\boldsymbol{\theta}_d} L^1(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)\|_2^2].$$

After re-arranging the terms we obtain

$$\mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] \leq (\eta^t)^2\sigma_d^2 + (1-2\eta^t\xi)\mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + 2\eta^t\gamma\mathbb{E}\left[\left(\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2 + \sum_{i=d+1}^{K}\|\boldsymbol{\Delta}_i^t\|_2\right)\|\boldsymbol{\Delta}_d^t\|_2\right]$$

apply $2ab \leq a^2 + b^2$

and define $\mathbb{1}(x) = 1$ for $x > 0$ and $\mathbb{1}(x) = 0$ otherwise:

$$\begin{aligned}
&\leq (\eta^t)^2\sigma_d^2 + (1-2\eta^t\xi)\mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\left(\|\boldsymbol{\Delta}_i^{t+1}\|_2^2 + \mathbb{1}(d-1)\|\boldsymbol{\Delta}_d^t\|_2^2\right)\right] \\
&+ \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\left(\|\boldsymbol{\Delta}_i^t\|_2^2 + \mathbb{1}(K-d)\|\boldsymbol{\Delta}_d^t\|_2^2\right)\right] \\
&= (\eta^t)^2\sigma_d^2 + \mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2]\cdot\left[1 - 2\eta^t\xi + \eta^t\gamma\left(\sum_{i=1}^{d-1}\mathbb{1}(d-1) + \sum_{i=d+1}^{K}\mathbb{1}(K-d)\right)\right] \\
&+ \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2^2\right] + \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^t\|_2^2\right]
\end{aligned}$$

We obtained

$$\mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] \leq (\eta^t)^2\sigma_d^2 + [1 - 2\eta^t\xi + \eta^t\gamma(K-1)]\mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2^2\right] + \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^t\|_2^2\right]$$

we next re-group the terms as follows

$$\mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] - \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2^2\right] \leq [1 - 2\eta^t\xi + \eta^t\gamma(K-1)]\mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^t\|_2^2\right] + (\eta^t)^2\sigma_d^2$$

and then sum over d from 1 to K

$$\begin{aligned}
&\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] - \eta^t\gamma\mathbb{E}\left[\sum_{d=1}^{K}\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2^2\right] \\
\leq\ & [1 - 2\eta^t\xi + \eta^t\gamma(K-1)]\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^t\|_2^2\right] + \eta^t\gamma\mathbb{E}\left[\sum_{d=1}^{K}\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^t\|_2^2\right] + (\eta^t)^2\sum_{d=1}^{K}\sigma_d^2
\end{aligned}$$

Let $\sigma = \sqrt{\sum_{d=1}^{K}\sigma_d^2}$. Also, note that

$$\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] - \eta^t\gamma(K-1)\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] \leq \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] - \eta^t\gamma\mathbb{E}\left[\sum_{d=1}^{K}\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2^2\right]$$

and

$$\begin{aligned}
&[1 - 2\eta^t\xi + \eta^t\gamma(K-1)]\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^t\|_2^2\right] + \eta^t\gamma\mathbb{E}\left[\sum_{d=1}^{K}\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^t\|_2^2\right] + (\eta^t)^2\sigma^2 \\
&\leq [1 - 2\eta^t\xi + \eta^t\gamma(K-1)]\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^t\|_2^2\right] + \eta^t\gamma(K-1)\mathbb{E}\left[\sum_{d=1}^{K}\|\Delta_d^t\|_2^2\right] + (\eta^t)^2\sigma^2
\end{aligned}$$

Combining these two facts with our previous results yields:

$$\begin{aligned}
&[1-(K-1)\eta^t\gamma]\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t+1}\|_2^2\right] \\
&\leq [1-2\eta^t\xi+\eta^t\gamma(K-1)]\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+\eta^t\gamma(K-1)\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+(\eta^t)^2\sigma^2 \\
&= [1-2\eta^t\xi+2\eta^t\gamma(K-1)]\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+(\eta^t)^2\sigma^2
\end{aligned}$$

Thus:

$$\begin{aligned}
\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t+1}\|_2^2\right] \leq \frac{1-2\eta^t\xi+2\eta^t\gamma(K-1)}{1-(k-1)\eta^t\gamma}\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right] \\
+\frac{(\eta^t)^2}{1-(K-1)\eta^t\gamma}\sigma^2.
\end{aligned}$$

Since $\gamma < \frac{2\xi}{3(K-1)}$, $\frac{1-2\eta^t\xi+2\eta^t\gamma(K-1)}{1-(K-1)\eta^t\gamma} < 1$.

A.3 Proof of Theorem 3.2

To obtain the final theorem we need to expand the recursion from Theorem 3.1. We obtained

$$\begin{aligned}
&\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t+1}\|_2^2\right] \\
&\leq \frac{1-2\eta^t[\xi-\gamma(K-1)]}{1-(K-1)\eta^t\gamma}\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+\frac{(\eta^t)^2}{1-(K-1)\eta^t\gamma}\sigma^2 \\
&= \left(1-\frac{\eta^t[2\xi-3\gamma(K-1)]}{1-(K-1)\eta^t\gamma}\right)\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+\frac{(\eta^t)^2}{1-(K-1)\eta^t\gamma}\sigma^2
\end{aligned}$$

Recall that we defined q^t in Theorem 3.1 as

$$q^t = 1-\frac{1-2\eta^t\xi+2\eta^t\gamma(K-1)}{1-(K-1)\eta^t\gamma} = \frac{\eta^t[2\xi-3\gamma(K-1)]}{1-(K-1)\eta^t\gamma}$$

and denote

$$\beta^t = \frac{(\eta^t)^2}{1-(K-1)\eta^t\gamma}.$$

Thus we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t+1}\|_2^2\right] \leq (1-q^t)\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+\beta^t\sigma^2 \\
\leq\ & (1-q^t)\left\{(1-q^{t-1})\mathbb{E}\left[\sum_{d=1}^{K}\|\Delta_d^{t-1}\|_2^2\right]+\beta^{t-1}\sigma^2\right\}+\beta^t\sigma^2 \\
=\ & (1-q^t)(1-q^{t-1})\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t-1}\|_2^2\right]+(1-q^t)\beta^{t-1}\sigma^2+\beta^t\sigma^2 \\
\leq\ & (1-q^t)(1-q^{t-1})\left\{(1-q^{t-2})\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t-2}\|_2^2\right]+\beta^{t-2}\sigma^2\right\}+(1-q^t)\beta^{t-1}\sigma^2+\beta^t\sigma^2 \\
=\ & (1-q^t)(1-q^{t-1})(1-q^{t-2})\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t-2}\|_2^2\right] \\
& +(1-q^t)(1-q^{t-1})\beta^{t-2}\sigma^2+(1-q^t)\beta^{t-1}\sigma^2+\beta^t\sigma^2
\end{aligned}$$

We end-up with the following

$$\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] \leq \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=0}^{t}(1-q^i)+\sigma^2\sum_{i=0}^{t-1}\beta^i\prod_{j=i+1}^{t}(1-q^j)+\beta^t\sigma^2.$$

Set $q^t=\frac{\frac{3}{2}}{t+2}$ and

$$\begin{aligned}\eta^t &= \frac{q^t}{2\xi-3\gamma(K-1)+q^t(K-1)\gamma}\\ &= \frac{\frac{3}{2}}{[2\xi-3\gamma(K-1)](t+2)+\frac{3}{2}(K-1)\gamma}.\end{aligned}$$

Denote $A=2\xi-3\gamma(K-1)$ and $B=\frac{3}{2}(K-1)\gamma$. Thus

$$\eta^t=\frac{\frac{3}{2}}{A(t+2)+B}$$

and

$$\beta^t=\frac{(\eta^t)^2}{1-\frac{2}{3}B\eta^t}=\frac{\frac{9}{4}}{A(t+2)[A(t+2)+B]}.$$

$$\begin{aligned}&\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right]\\ \leq\ &\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=0}^{t}\left(1-\frac{\frac{3}{2}}{i+2}\right)+\sigma^2\sum_{i=0}^{t-1}\frac{\frac{9}{4}}{A(i+2)[A(i+2)+B]}\prod_{j=i+1}^{t}\left(1-\frac{\frac{3}{2}}{j+2}\right)\\ &+\sigma^2\frac{\frac{9}{4}}{A(t+2)[A(t+2)+B]}\\ =\ &\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=2}^{t+2}\left(1-\frac{\frac{3}{2}}{i}\right)+\sigma^2\sum_{i=2}^{t+1}\frac{\frac{9}{4}}{Ai[Ai+B]}\prod_{j=i+1}^{t+2}\left(1-\frac{\frac{3}{2}}{j}\right)+\sigma^2\frac{\frac{9}{4}}{A(t+2)[A(t+2)+B]}\end{aligned}$$

Since $A>0$ and $B>0$ thus

$$\begin{aligned}&\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right]\\ \leq\ &\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=2}^{t+2}\left(1-\frac{\frac{3}{2}}{i}\right)+\sigma^2\sum_{i=2}^{t+1}\frac{\frac{9}{4}}{Ai[Ai+B]}\prod_{j=i+1}^{t+2}\left(1-\frac{\frac{3}{2}}{j}\right)+\sigma^2\frac{\frac{9}{4}}{A(t+2)[A(t+2)+B]}\\ \leq\ &\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=2}^{t+2}\left(1-\frac{\frac{3}{2}}{i}\right)+\sigma^2\sum_{i=2}^{t+1}\frac{\frac{9}{4}}{(Ai)^2}\prod_{j=i+1}^{t+2}\left(1-\frac{\frac{3}{2}}{j}\right)+\sigma^2\frac{\frac{9}{4}}{[A(t+2)]^2}\end{aligned}$$

We can next use the fact that for any $a\in(1,2)$:

$$\prod_{i=\tau+1}^{t+2}\left(1-\frac{a}{i}\right)\leq\left(\frac{\tau+1}{t+3}\right)^a.$$

The bound then becomes

$$\begin{aligned}
& \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] \\
\leq\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=2}^{t+2}\left(1-\frac{\frac{3}{2}}{i}\right)+\sigma^2\sum_{i=2}^{t+1}\frac{\frac{9}{4}}{(Ai)^2}\prod_{j=i+1}^{t+2}\left(1-\frac{\frac{3}{2}}{j}\right)+\sigma^2\frac{\frac{9}{4}}{[A(t+2)]^2} \\
\leq\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\sum_{i=2}^{t+1}\frac{\frac{9}{4}}{(Ai)^2}\left(\frac{i+1}{t+3}\right)^{\frac{3}{2}}+\sigma^2\frac{\frac{9}{4}}{[A(t+2)]^2} \\
=\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\sum_{i=2}^{t+2}\frac{\frac{9}{4}}{(Ai)^2}\left(\frac{i+1}{t+3}\right)^{\frac{3}{2}}
\end{aligned}$$

Note that $(i+1)^{\frac{3}{2}} \leq 2i$ for $i = 2, 3, \ldots$, thus

$$\begin{aligned}
& \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] \\
\leq\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\frac{\frac{9}{4}}{A^2(t+3)^{\frac{3}{2}}}\sum_{i=2}^{t+2}\frac{(i+1)^{\frac{3}{2}}}{i^2} \\
\leq\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\frac{\frac{9}{2}}{A^2(t+3)^{\frac{3}{2}}}\sum_{i=2}^{t+2}\frac{1}{i^{\frac{1}{2}}} \\
& \text{finally note that } \sum_{i=2}^{t+2}\frac{1}{i^{\frac{1}{2}}} \leq \int_1^{t+2}\frac{1}{x^{\frac{1}{2}}}dx \leq 2(t+3)^{\frac{1}{2}}. \text{ Thus} \\
\leq\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\frac{9}{A^2(t+3)} \\
& \text{substituting } A = 2\xi - 3\gamma(K-1) \text{ gives} \\
=\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\frac{9}{[2\xi-3\gamma(K-1)]^2(t+3)}
\end{aligned}$$

This leads us to the final theorem.

Supplemental Material

A Proofs

Proof of Theorem **??** relies on Theorem **??**, which in turn relies on Theorem A.1 and Lemma A.1, both of which are stated below. Proofs of the lemma and theorems follow in the subsequent subsections.

The next result is a standard result from convex optimization (Theorem 2.1.14 in (**?**)) and is used in the proof of Theorem A.1 below.

Lemma A.1. *For any $d = 1, 2, \ldots, K$, the gradient operator $\mathcal{G}_d(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_{K-1}^*, \boldsymbol{\theta}_K^*)$ under Assumption **??** (strong concavity) and Assumption **??** (smoothness) with constant step size choice $0 < \eta \leq \frac{2}{\mu_d + \lambda_d}$ is contractive, i.e.*

$$\|\mathcal{G}_d(\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_K^*) - \boldsymbol{\theta}_d^*\|_2 \leq \left(1 - \frac{2\eta\mu_d\lambda_d}{\mu_d + \lambda_d}\right) \|\boldsymbol{\theta}_d - \boldsymbol{\theta}_d^*\|_2 \tag{1}$$

for all $\boldsymbol{\theta}_d \in B_2(r_d, \boldsymbol{\theta}_d^*)$.

The next theorem also holds for any d from 1 to K. Let $r_1, \ldots, r_{d-1}, r_{d+1}, \ldots, r_K > 0$ and $\boldsymbol{\theta}_1 \in B_2(r_1, \boldsymbol{\theta}_1^*), \ldots, \boldsymbol{\theta}_{d-1} \in B_2(r_{d-1}, \boldsymbol{\theta}_{d-1}^*), \boldsymbol{\theta}_{d+1} \in B_2(r_{d+1}, \boldsymbol{\theta}_{d+1}^*), \ldots, \boldsymbol{\theta}_K \in B_2(r_k, \boldsymbol{\theta}_K^*)$.

Theorem A.1. *For some radius $r_d > 0$ and a triplet $(\gamma_d, \lambda_d, \mu_d)$ such that $0 \leq \gamma_d < \lambda_d \leq \mu_d$, suppose that the function $L(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_{K-1}^*, \boldsymbol{\theta}_K^*)$ is λ_d-strongly concave (Assumption **??**) and μ_d-smooth (Assumption **??**), and that the GS (γ_d) condition of Assumption **??** holds. Then the population gradient AM operator $\mathcal{G}_d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$ with step η such that $0 < \eta \leq \min_{i=1,2,\ldots,K} \frac{2}{\mu_i + \lambda_i}$ is contractive over a ball $B_2(r_d, \boldsymbol{\theta}_d^*)$, i.e.*

$$\|\mathcal{G}_d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K) - \boldsymbol{\theta}_d^*\|_2 \leq (1 - \xi\eta)\|\boldsymbol{\theta}_d - \boldsymbol{\theta}_d^*\|_2 + \eta\gamma \sum_{\substack{i=1 \\ i \neq d}}^{K} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_i^*\|_2 \tag{2}$$

where $\gamma \coloneqq \max_{i=1,2,\ldots,K} \gamma_i$, and $\xi \coloneqq \min_{i=1,2,\ldots,K} \frac{2\mu_i\lambda_i}{\mu_i + \lambda_i}$.

A.1 Proof of Theorem A.1

$$\begin{aligned}
&\|\mathcal{G}_d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K) - \boldsymbol{\theta}_d^*\|_2 = \|\boldsymbol{\theta}_d + \eta\nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \ldots, \boldsymbol{\theta}_K) - \boldsymbol{\theta}_d^*\|_2 \\
&\text{by the triangle inequality we further get} \\
&\leq \|\boldsymbol{\theta}_d + \eta\nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_K^*) - \boldsymbol{\theta}_d^*\|_2 \\
&+\eta\|\nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d, \ldots, \boldsymbol{\theta}_K) \\
&-\nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_K^*)\|_2 \\
&\text{by the contractivity of } T \text{ from Equation 1 from Lemma A.1 and GS condition} \\
&\leq \left(1 - \frac{2\eta\mu_d\lambda_d}{\mu_d + \lambda_d}\right) \|\theta_d - \theta^*\|_2 + \eta\gamma_d \sum_{\substack{i=1 \\ i \neq d}}^{K} \|\theta_i - \theta_i^*\|_2.
\end{aligned}$$

A.2 Proof of Theorem ??

Let $\boldsymbol{\theta}_d^{t+1} = \Pi_d(\tilde{\boldsymbol{\theta}}_d^{t+1})$, where $\tilde{\boldsymbol{\theta}}_d^{t+1} \coloneqq \boldsymbol{\theta}_d^t + \eta^t \nabla_{\boldsymbol{\theta}_d} L^1(\boldsymbol{\theta}_1^{t+1}, \boldsymbol{\theta}_2^{t+1}, \ldots, \boldsymbol{\theta}_{d-1}^{t+1}, \boldsymbol{\theta}_d^t, \boldsymbol{\theta}_{d+1}^t, \ldots, \boldsymbol{\theta}_K^t)$, where $\nabla_{\boldsymbol{\theta}_d} L^1$ is the gradient computed with respect to a single data sample, is the update vector prior to the projection onto a ball $B_2(\frac{r_d}{2}, \boldsymbol{\theta}_d^0)$. Let $\boldsymbol{\Delta}_d^{t+1} \coloneqq \boldsymbol{\theta}_d^{t+1} - \boldsymbol{\theta}_d^*$ and $\tilde{\boldsymbol{\Delta}}_d^{t+1} \coloneqq \tilde{\boldsymbol{\theta}}_d^{t+1} - \boldsymbol{\theta}_d^*$. Thus

$$\begin{aligned}
\|\boldsymbol{\Delta}_d^{t+1}\|_2^2 - \|\boldsymbol{\Delta}_d^t\|_2^2 &\leq \|\tilde{\boldsymbol{\Delta}}_d^{t+1}\|_2^2 - \|\boldsymbol{\Delta}_d^t\|_2^2 \\
&= \|\tilde{\boldsymbol{\theta}}_d^{t+1} - \boldsymbol{\theta}_d^*\| - \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\| \\
&= \left\langle \tilde{\boldsymbol{\theta}}_d^{t+1} - \boldsymbol{\theta}_d^t, \tilde{\boldsymbol{\theta}}_d^{t+1} + \boldsymbol{\theta}_d^t - 2\boldsymbol{\theta}_d^* \right\rangle.
\end{aligned}$$

Let $\hat{\boldsymbol{W}}_d^t \coloneqq \nabla_{\boldsymbol{\theta}_d} L^1(\boldsymbol{\theta}_1^{t+1}, \boldsymbol{\theta}_2^{t+1}, \ldots, \boldsymbol{\theta}_{d-1}^{t+1}, \boldsymbol{\theta}_d^t, \boldsymbol{\theta}_{d+1}^t, \ldots, \boldsymbol{\theta}_K^t)$. Then we have that $\tilde{\boldsymbol{\theta}}_d^{t+1} - \boldsymbol{\theta}_d^t = \eta^t \hat{\boldsymbol{W}}_d^t$. We combine it with Equation 3 and obtain:

$$\begin{aligned} & \|\boldsymbol{\Delta}_d^{t+1}\|_2^2 - \|\boldsymbol{\Delta}_d^t\|_2^2 \\ \leq & \left\langle \eta^t \hat{\boldsymbol{W}}_d^t, \eta^t \hat{\boldsymbol{W}}_d^t + 2(\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*) \right\rangle \\ = & (\eta^t)^2 (\hat{\boldsymbol{W}}_d^t)^\top \hat{\boldsymbol{W}}_d^t + 2\eta^t (\hat{\boldsymbol{W}}_d^t)^\top (\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*) \\ = & (\eta^t)^2 \|\hat{\boldsymbol{W}}_d^t\|_2^2 + 2\eta^t \left\langle \hat{\boldsymbol{W}}_d^t, \boldsymbol{\Delta}_d^t \right\rangle. \end{aligned}$$

Let $\boldsymbol{W}_d^t \coloneqq \nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1^{t+1}, \boldsymbol{\theta}_2^{t+1}, \ldots, \boldsymbol{\theta}_{d-1}^{t+1}, \boldsymbol{\theta}_d^t, \boldsymbol{\theta}_{d+1}^t, \ldots, \boldsymbol{\theta}_K^t)$. Recall that $\mathbb{E}[\hat{\boldsymbol{W}}_d^t] = \boldsymbol{W}_d^t$. By the properties of martingales, i.e. iterated expectations and tower property:

$$\mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] \quad \leq \quad \mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + (\eta^t)^2 \mathbb{E}[\|\hat{\boldsymbol{W}}_d^t\|_2^2] + 2\eta^t \mathbb{E}[\langle \boldsymbol{W}_d^t, \boldsymbol{\Delta}_d^t \rangle] \tag{3}$$

Let $\boldsymbol{W}_d^* \coloneqq \nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_K^*)$. By self-consistency, i.e. $\boldsymbol{\theta}_d^* = \arg\max_{\boldsymbol{\theta}_d \in \Omega_d} L(\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{d-1}^*, \boldsymbol{\theta}_d, \boldsymbol{\theta}_{d+1}^*, \ldots, \boldsymbol{\theta}_K^*)$ and convexity of Ω_d we have that

$$\left\langle \boldsymbol{W}_d^*, \boldsymbol{\Delta}_d^t \right\rangle = \left\langle \nabla_{\boldsymbol{\theta}_d} L(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_K^*), \boldsymbol{\Delta}_d^t \right\rangle \leq 0.$$

Combining this with Equation 3 we have

$$\mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] \quad \leq \quad \mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + (\eta^t)^2 \mathbb{E}[\|\hat{\boldsymbol{W}}_d^t\|_2^2] + 2\eta^t \mathbb{E}[\langle \boldsymbol{W}_d^t - \boldsymbol{W}_d^*, \boldsymbol{\Delta}_d^t \rangle].$$

Define $\mathcal{G}_d^t \coloneqq \boldsymbol{\theta}_d^t + \eta^t \boldsymbol{W}_d^t$ and $\mathcal{G}_d^{t*} \coloneqq \boldsymbol{\theta}_d^* + \eta^t \boldsymbol{W}_d^*$. Thus

$$\begin{aligned} & \eta^t \left\langle \boldsymbol{W}_d^t - \boldsymbol{W}_d^*, \boldsymbol{\Delta}_d^t \right\rangle \\ = & \left\langle \mathcal{G}_d^t - \mathcal{G}_d^{t*} - (\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*), \boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^* \right\rangle \\ = & \left\langle \mathcal{G}_d^t - \mathcal{G}_d^{t*}, \boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^* \right\rangle - \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\|_2^2 \end{aligned}$$

by the fact that $\mathcal{G}_d^{t*} = \boldsymbol{\theta}_d^* + \eta^t \boldsymbol{W}_d^* = \boldsymbol{\theta}_d^*$ (since $\boldsymbol{W}_d^* = 0$):

$$= \left\langle \mathcal{G}_d^t - \boldsymbol{\theta}_d^*, \boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^* \right\rangle - \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\|_2^2$$

by the contractivity of $\mathcal{G}^t$ from Theorem A.1:

$$\begin{aligned} \leq & \left\{ (1 - \eta^t \xi) \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\| + \eta^t \gamma \left(\sum_{i=1}^{d-1} \|\boldsymbol{\theta}_i^{t+1} - \boldsymbol{\theta}_i^*\|_2 \ + \sum_{i=d+1}^{K} \|\boldsymbol{\theta}_i^t - \boldsymbol{\theta}_i^*\|_2 \right) \right\} \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\|_2 - \|\boldsymbol{\theta}_d^t - \boldsymbol{\theta}_d^*\|_2^2 \\ \leq & \left\{ (1 - \eta^t \xi) \|\boldsymbol{\Delta}_d^t\|_2 + \eta^t \gamma \left(\sum_{i=1}^{d-1} \|\boldsymbol{\Delta}_i^{t+1}\|_2 + \sum_{i=d+1}^{K} \|\boldsymbol{\Delta}_i^t\|_2 \right) \right\} \cdot \|\boldsymbol{\Delta}_d^t\|_2 - \|\boldsymbol{\Delta}_d^t\|_2^2 \end{aligned}$$

Combining this result with Equation 4 gives

$$\begin{aligned} \mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] \quad \leq & \quad \mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + (\eta^t)^2 \mathbb{E}[\|\hat{\boldsymbol{W}}_d^t\|_2^2] + 2\mathbb{E}\left[\left\{ (1 - \eta^t \xi) \|\boldsymbol{\Delta}_d^t\|_2 + \eta^t \gamma \left(\sum_{i=1}^{d-1} \|\boldsymbol{\Delta}_i^{t+1}\|_2 + \sum_{i=d+1}^{K} \|\boldsymbol{\Delta}_i^t\|_2 \right) \right\} \right. \\ & \left. \cdot \|\boldsymbol{\Delta}_d^t\|_2 - \|\boldsymbol{\Delta}_d^t\|_2^2 \right] \\ \leq & \quad \mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + (\eta^t)^2 \sigma_d^2 + 2\mathbb{E}\left[\left\{ (1 - \eta^t \xi) \|\boldsymbol{\Delta}_d^t\|_2 + \eta^t \gamma \left(\sum_{i=1}^{d-1} \|\boldsymbol{\Delta}_i^{t+1}\|_2 + \sum_{i=d+1}^{K} \|\boldsymbol{\Delta}_i^t\|_2 \right) \right\} \right. \\ & \left. \cdot \|\boldsymbol{\Delta}_d^t\|_2 - \|\boldsymbol{\Delta}_d^t\|_2^2 \right], \quad \text{where} \end{aligned}$$

$$\sigma_d^2 = \sup_{\substack{\boldsymbol{\theta}_1 \in B_2(r_1, \boldsymbol{\theta}_1^*) \\ \ldots \\ \boldsymbol{\theta}_K \in B_2(r_K, \boldsymbol{\theta}_K^*)}} \mathbb{E}[\|\nabla_{\boldsymbol{\theta}_d} L^1(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)\|_2^2].$$

After re-arranging the terms we obtain

$$\mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] \leq (\eta^t)^2\sigma_d^2 + (1-2\eta^t\xi)\mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + 2\eta^t\gamma\mathbb{E}\left[\left(\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2 + \sum_{i=d+1}^{K}\|\boldsymbol{\Delta}_i^t\|_2\right)\|\boldsymbol{\Delta}_d^t\|_2\right]$$

apply $2ab \leq a^2 + b^2$

and define $\mathbb{1}(x) = 1$ for $x > 0$ and $\mathbb{1}(x) = 0$ otherwise:

$$\begin{aligned}
&\leq (\eta^t)^2\sigma_d^2 + (1-2\eta^t\xi)\mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\left(\|\boldsymbol{\Delta}_i^{t+1}\|_2^2 + \mathbb{1}(d-1)\|\boldsymbol{\Delta}_d^t\|_2^2\right)\right] \\
&+ \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\left(\|\boldsymbol{\Delta}_i^t\|_2^2 + \mathbb{1}(K-d)\|\boldsymbol{\Delta}_d^t\|_2^2\right)\right] \\
&= (\eta^t)^2\sigma_d^2 + \mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2]\cdot\left[1 - 2\eta^t\xi + \eta^t\gamma\left(\sum_{i=1}^{d-1}\mathbb{1}(d-1) + \sum_{i=d+1}^{K}\mathbb{1}(K-d)\right)\right] \\
&+ \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2^2\right] + \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^t\|_2^2\right]
\end{aligned}$$

We obtained

$$\mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] \leq (\eta^t)^2\sigma_d^2 + [1 - 2\eta^t\xi + \eta^t\gamma(K-1)]\mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2^2\right] + \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^t\|_2^2\right]$$

we next re-group the terms as follows

$$\mathbb{E}[\|\boldsymbol{\Delta}_d^{t+1}\|_2^2] - \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2^2\right] \leq [1 - 2\eta^t\xi + \eta^t\gamma(K-1)]\mathbb{E}[\|\boldsymbol{\Delta}_d^t\|_2^2] + \eta^t\gamma\mathbb{E}\left[\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^t\|_2^2\right] + (\eta^t)^2\sigma_d^2$$

and then sum over d from 1 to K

$$\begin{aligned}
&\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] - \eta^t\gamma\mathbb{E}\left[\sum_{d=1}^{K}\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2^2\right] \\
\leq\ & [1 - 2\eta^t\xi + \eta^t\gamma(K-1)]\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^t\|_2^2\right] + \eta^t\gamma\mathbb{E}\left[\sum_{d=1}^{K}\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^t\|_2^2\right] + (\eta^t)^2\sum_{d=1}^{K}\sigma_d^2
\end{aligned}$$

Let $\sigma = \sqrt{\sum_{d=1}^{K}\sigma_d^2}$. Also, note that

$$\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] - \eta^t\gamma(K-1)\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] \leq \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] - \eta^t\gamma\mathbb{E}\left[\sum_{d=1}^{K}\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^{t+1}\|_2^2\right]$$

and

$$\begin{aligned}
&[1 - 2\eta^t\xi + \eta^t\gamma(K-1)]\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^t\|_2^2\right] + \eta^t\gamma\mathbb{E}\left[\sum_{d=1}^{K}\sum_{i=1}^{d-1}\|\boldsymbol{\Delta}_i^t\|_2^2\right] + (\eta^t)^2\sigma^2 \\
&\leq [1 - 2\eta^t\xi + \eta^t\gamma(K-1)]\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^t\|_2^2\right] + \eta^t\gamma(K-1)\mathbb{E}\left[\sum_{d=1}^{K}\|\Delta_d^t\|_2^2\right] + (\eta^t)^2\sigma^2
\end{aligned}$$

Combining these two facts with our previous results yields:

$$\begin{aligned}
&[1-(K-1)\eta^t\gamma]\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t+1}\|_2^2\right] \\
&\leq [1-2\eta^t\xi+\eta^t\gamma(K-1)]\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+\eta^t\gamma(K-1)\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+(\eta^t)^2\sigma^2 \\
&= [1-2\eta^t\xi+2\eta^t\gamma(K-1)]\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+(\eta^t)^2\sigma^2
\end{aligned}$$

Thus:

$$\begin{aligned}
\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t+1}\|_2^2\right] \leq \frac{1-2\eta^t\xi+2\eta^t\gamma(K-1)}{1-(k-1)\eta^t\gamma}\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right] \\
+\frac{(\eta^t)^2}{1-(K-1)\eta^t\gamma}\sigma^2.
\end{aligned}$$

Since $\gamma < \frac{2\xi}{3(K-1)}$, $\frac{1-2\eta^t\xi+2\eta^t\gamma(K-1)}{1-(K-1)\eta^t\gamma} < 1$.

A.3 Proof of Theorem ??

To obtain the final theorem we need to expand the recursion from Theorem **??**. We obtained

$$\begin{aligned}
&\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t+1}\|_2^2\right] \\
&\leq \frac{1-2\eta^t[\xi-\gamma(K-1)]}{1-(K-1)\eta^t\gamma}\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+\frac{(\eta^t)^2}{1-(K-1)\eta^t\gamma}\sigma^2 \\
&= \left(1-\frac{\eta^t[2\xi-3\gamma(K-1)]}{1-(K-1)\eta^t\gamma}\right)\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+\frac{(\eta^t)^2}{1-(K-1)\eta^t\gamma}\sigma^2
\end{aligned}$$

Recall that we defined q^t in Theorem **??** as

$$q^t = 1-\frac{1-2\eta^t\xi+2\eta^t\gamma(K-1)}{1-(K-1)\eta^t\gamma} = \frac{\eta^t[2\xi-3\gamma(K-1)]}{1-(K-1)\eta^t\gamma}$$

and denote

$$\beta^t = \frac{(\eta^t)^2}{1-(K-1)\eta^t\gamma}.$$

Thus we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t+1}\|_2^2\right] \leq (1-q^t)\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t}\|_2^2\right]+\beta^t\sigma^2 \\
\leq\ & (1-q^t)\left\{(1-q^{t-1})\mathbb{E}\left[\sum_{d=1}^{K}\|\Delta_d^{t-1}\|_2^2\right]+\beta^{t-1}\sigma^2\right\}+\beta^t\sigma^2 \\
=\ & (1-q^t)(1-q^{t-1})\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t-1}\|_2^2\right]+(1-q^t)\beta^{t-1}\sigma^2+\beta^t\sigma^2 \\
\leq\ & (1-q^t)(1-q^{t-1})\left\{(1-q^{t-2})\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t-2}\|_2^2\right]+\beta^{t-2}\sigma^2\right\}+(1-q^t)\beta^{t-1}\sigma^2+\beta^t\sigma^2 \\
=\ & (1-q^t)(1-q^{t-1})(1-q^{t-2})\mathbb{E}\left[\sum_{d=1}^{K}\|\mathbf{\Delta}_d^{t-2}\|_2^2\right] \\
& +(1-q^t)(1-q^{t-1})\beta^{t-2}\sigma^2+(1-q^t)\beta^{t-1}\sigma^2+\beta^t\sigma^2
\end{aligned}$$

We end-up with the following

$$\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] \leq \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=0}^{t}(1-q^i)+\sigma^2\sum_{i=0}^{t-1}\beta^i\prod_{j=i+1}^{t}(1-q^j)+\beta^t\sigma^2.$$

Set $q^t=\frac{\frac{3}{2}}{t+2}$ and

$$\begin{aligned}\eta^t &= \frac{q^t}{2\xi-3\gamma(K-1)+q^t(K-1)\gamma}\\ &= \frac{\frac{3}{2}}{[2\xi-3\gamma(K-1)](t+2)+\frac{3}{2}(K-1)\gamma}.\end{aligned}$$

Denote $A=2\xi-3\gamma(K-1)$ and $B=\frac{3}{2}(K-1)\gamma$. Thus

$$\eta^t=\frac{\frac{3}{2}}{A(t+2)+B}$$

and

$$\beta^t=\frac{(\eta^t)^2}{1-\frac{2}{3}B\eta^t}=\frac{\frac{9}{4}}{A(t+2)[A(t+2)+B]}.$$

$$\begin{aligned}&\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right]\\ \leq\ &\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=0}^{t}\left(1-\frac{\frac{3}{2}}{i+2}\right)+\sigma^2\sum_{i=0}^{t-1}\frac{\frac{9}{4}}{A(i+2)[A(i+2)+B]}\prod_{j=i+1}^{t}\left(1-\frac{\frac{3}{2}}{j+2}\right)\\ &+\sigma^2\frac{\frac{9}{4}}{A(t+2)[A(t+2)+B]}\\ =\ &\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=2}^{t+2}\left(1-\frac{\frac{3}{2}}{i}\right)+\sigma^2\sum_{i=2}^{t+1}\frac{\frac{9}{4}}{Ai[Ai+B]}\prod_{j=i+1}^{t+2}\left(1-\frac{\frac{3}{2}}{j}\right)+\sigma^2\frac{\frac{9}{4}}{A(t+2)[A(t+2)+B]}\end{aligned}$$

Since $A>0$ and $B>0$ thus

$$\begin{aligned}&\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right]\\ \leq\ &\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=2}^{t+2}\left(1-\frac{\frac{3}{2}}{i}\right)+\sigma^2\sum_{i=2}^{t+1}\frac{\frac{9}{4}}{Ai[Ai+B]}\prod_{j=i+1}^{t+2}\left(1-\frac{\frac{3}{2}}{j}\right)+\sigma^2\frac{\frac{9}{4}}{A(t+2)[A(t+2)+B]}\\ \leq\ &\mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=2}^{t+2}\left(1-\frac{\frac{3}{2}}{i}\right)+\sigma^2\sum_{i=2}^{t+1}\frac{\frac{9}{4}}{(Ai)^2}\prod_{j=i+1}^{t+2}\left(1-\frac{\frac{3}{2}}{j}\right)+\sigma^2\frac{\frac{9}{4}}{[A(t+2)]^2}\end{aligned}$$

We can next use the fact that for any $a\in(1,2)$:

$$\prod_{i=\tau+1}^{t+2}\left(1-\frac{a}{i}\right)\leq\left(\frac{\tau+1}{t+3}\right)^a.$$

The bound then becomes

$$\begin{aligned}
& \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] \\
\leq\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\prod_{i=2}^{t+2}\left(1-\frac{\frac{3}{2}}{i}\right)+\sigma^2\sum_{i=2}^{t+1}\frac{\frac{9}{4}}{(Ai)^2}\prod_{j=i+1}^{t+2}\left(1-\frac{\frac{3}{2}}{j}\right)+\sigma^2\frac{\frac{9}{4}}{[A(t+2)]^2} \\
\leq\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\sum_{i=2}^{t+1}\frac{\frac{9}{4}}{(Ai)^2}\left(\frac{i+1}{t+3}\right)^{\frac{3}{2}}+\sigma^2\frac{\frac{9}{4}}{[A(t+2)]^2} \\
=\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\sum_{i=2}^{t+2}\frac{\frac{9}{4}}{(Ai)^2}\left(\frac{i+1}{t+3}\right)^{\frac{3}{2}}
\end{aligned}$$

Note that $(i+1)^{\frac{3}{2}} \leq 2i$ for $i = 2, 3, \ldots$, thus

$$\begin{aligned}
& \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{t+1}\|_2^2\right] \\
\leq\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\frac{\frac{9}{4}}{A^2(t+3)^{\frac{3}{2}}}\sum_{i=2}^{t+2}\frac{(i+1)^{\frac{3}{2}}}{i^2} \\
\leq\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\frac{\frac{9}{2}}{A^2(t+3)^{\frac{3}{2}}}\sum_{i=2}^{t+2}\frac{1}{i^{\frac{1}{2}}} \\
& \text{finally note that } \sum_{i=2}^{t+2}\frac{1}{i^{\frac{1}{2}}} \leq \int_1^{t+2}\frac{1}{x^{\frac{1}{2}}}dx \leq 2(t+3)^{\frac{1}{2}}. \text{ Thus} \\
\leq\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\frac{9}{A^2(t+3)} \\
& \text{substituting } A = 2\xi - 3\gamma(K-1) \text{ gives} \\
=\ & \mathbb{E}\left[\sum_{d=1}^{K}\|\boldsymbol{\Delta}_d^{0}\|_2^2\right]\left(\frac{2}{t+3}\right)^{\frac{3}{2}}+\sigma^2\frac{9}{[2\xi-3\gamma(K-1)]^2(t+3)}
\end{aligned}$$

This leads us to the final theorem.