Asymptotics of MAP Inference in Deep Networks

Parthe Pandit∗, Mojtaba Sahraee∗, Sundeep Rangan†, Alyson K. Fletcher∗

∗ UCLA, Dept. Statistics and ECE, Los Angeles, CA, {parthe,msahraee,akfletcher}@ucla.edu
† NYU, Dept. ECE, Brooklyn, NY, srangan@nyu.edu

Abstract—Deep generative priors are a powerful tool for reconstruction problems with complex data such as images and text. Inverse problems using such models require solving an inference problem of estimating the input and hidden units of the multi-layer network from its output. Maximum a priori (MAP) estimation is a widely-used inference method as it is straightforward to implement, and has been successful in practice. However, rigorous analysis of MAP inference in multi-layer networks is difficult. This work considers a recently-developed method, multi-layer vector approximate message passing (ML-VAMP), to study MAP inference in deep networks. It is shown that the mean squared error of the ML-VAMP estimate can be exactly and rigorously characterized in a certain high-dimensional random limit. The proposed method thus provides a tractable method for MAP inference with exact performance guarantees.

I. INTRODUCTION

We consider inference in an L layer stochastic neural network of the form,

\[
\begin{align*}
\mathbf{z}_{0}^{\ell} &= \mathbf{W}_{\ell} \mathbf{z}_{\ell-1}^{0} + \mathbf{b}_{\ell} + \mathbf{\xi}_{\ell}, \quad \ell = 1, 3, \ldots, L-1 \tag{1a} \\
\mathbf{z}_{1}^{\ell} &= \phi_{\ell}(\mathbf{z}_{\ell-1}^{0}, \mathbf{\xi}_{\ell}), \quad \ell = 2, 4, \ldots, L. \tag{1b}
\end{align*}
\]

where \(\mathbf{z}_{0}^{0}\) is the initial input, \(\mathbf{z}_{\ell}^{0}, \ell = 1, \ldots, L-1\) are the intermediate hidden unit outputs and \(\mathbf{y} = \mathbf{z}_{L}^{1}\) is the output. The number of layers \(L\) is even. The equations (1a) correspond to linear (fully-connected) layers with weights and biases \(\mathbf{W}_{\ell}\) and \(\mathbf{b}_{\ell}\), while (1b) correspond to elementwise activation functions such as sigmoid or ReLU. The signals \(\mathbf{\xi}_{\ell}\) represent noise terms. A block diagram for the network is shown in the top panel of Fig. 1. The inference problem is to estimate the initial and hidden states \(\mathbf{z}_{\ell}^{0}, \ell = 0, \ldots, L-1\) from the final output \(\mathbf{y}\). We assume that network parameters (the weights, biases and activation functions) are all known (i.e. already trained). Hence, this is not the learning problem. The superscript 0 in \(\mathbf{z}_{\ell}^{0}\) indicates that these are the "true" values, to be distinguished from estimates that we will discuss later.

This inference problem arises commonly when deep networks are used as generative priors. Deep neural networks have been extremely successful in providing probabilistic generative models of complex data such as images, audio and text. The models can be trained either via variational autoencoders [1], [2] or generative adversarial networks [3], [4]. In inverse problems, a deep network is used as a generative prior for the data (such as an image) and additional layers are added to model the measurements (such as blurring, occlusion or noise) [5], [6]. Inference can then be used to reconstruct the original image from the measurements.

Many deep network-based reconstruction methods perform maximum a priori (MAP) estimation via minimization of the negative log likelihood [5], [6] or an equivalent regularized least-squares objective [7]. MAP minimization is readily implementable and has worked successfully in practice in problems such as inpainting and compressed sensing. MAP estimation also provides an alternative to a separately learned reconstruction network such as [8]–[10]. However, due to the non-convex nature of the objective function, MAP estimation has been difficult to analyze rigorously. For example, results such as [11] provide only general scaling laws while the guarantees in [12] require that a non-convex projection operation can be performed exactly.

To better understand MAP-based reconstruction, this work considers inference in deep networks via approximate message passing (AMP). AMP [13] and its variants refer to a powerful class of techniques for inverse problems that are both computationally efficient and admit provable guarantees in certain high-dimensional limits. Recent works [14]–[17] have developed and analyzed variants of AMP for inference in multi-layer networks such as (1). The methods generally consider minimum mean squared error (MMSE) inference and estimation of the posterior density of the hidden units \(\mathbf{z}_{\ell}\) from \(\mathbf{y}\). Similar to other AMP methods, such MMSE-based multi-layer versions of AMP can be rigorously analyzed in cases with with large random transforms. This work specifically considers an extension of the multi-layer vector AMP (ML-VAMP) method proposed in [15]. ML-VAMP is derived from the recently-developed VAMP method of [18]–[20] which is itself based on expectation propagation [21] and expectation consistent approximate inference [22], [23]. Importantly, in the case of large random transforms, it is shown in [15] that the reconstruction error of ML-VAMP with MMSE estimation can be exactly predicted, enabling much sharper results than other analysis techniques. Moreover, under certain testable conditions ML-VAMP can provably asymptotically achieve the Bayes optimal estimate, even for non-convex problems.

However, MAP estimation is often preferable to MMSE inference since MAP can be formulated as an unconstrained optimization and implemented easily via standard deep learning optimizers [5]–[7]. This work thus considers a MAP version of ML-VAMP. We show two key results. First, it is shown that the iterations in MAP ML-VAMP can be regarded as a variant of an ADMM-type minimization [24] of the MAP objective. This result is similar to earlier connections between AMP and ADMM in [25]–[27]. In particular, when MAP ML-VAMP converges, its fixed points are critical points of the MAP objective. Secondly, similar to the MMSE ML-VAMP considered in [15], we can rigorously analyze MAP ML-VAMP...
in a large system limit (LSL) with high-dimensional random transforms $W$. It is shown that, in the LSL, the per iteration mean squared error of the estimates can be exactly characterized by a state evolution (SE). The SE tracks the correlation between the estimates and true values at each layer and are only slightly more complex than the SE updates for the MMSE case. The SE enables an exact characterization of the error of MAP estimation as a function of the network architecture, parameters and noise levels.

For space considerations, all proofs are contained in a full paper [28]. In addition to the proofs, the full paper includes further discussion with prior work, as well as algorithm and simulation details.

II. ML-VAMP FOR MAP INFERENCE

We consider inference in a probabilistic setting where, in (1), $z^0_{\ell}$ and $\xi^e$ are modeled as random vectors with some known densities. Inference can be then performed by MAP estimation,

$$\hat{z} = \arg\min_z J(z, y),$$

where $J(z, y)$ is the negative log posterior,

$$J(z, y) := -\ln p(z_0) - \sum_{\ell=1}^{L-1} \ln p(z_\ell|z_{\ell-1}) - \ln p(y|z_{L-1}),$$

where $p(z_0)$ is the prior on the initial input $z_0^0$ and $\ln p(z_\ell|z_{\ell-1})$ is defined implicitly from the probability distribution on the noise terms $\xi^e$ and the updates in (1).

The ML-VAMP algorithm from [15] for the inference problem is shown in Algorithm 1. For each hidden output $z_\ell$, the algorithm produces two estimates $\hat{z}^+_{\ell}$ and $\hat{z}^-_{\ell}$ indexed by the iteration number $k$. In each iteration, there is a forward pass that produces the estimates $\hat{z}^+_{\ell}$ and a reverse pass that produces the estimates $\hat{z}^-_{\ell}$. The estimates are produced by a set of estimation functions $g^\pm_{\ell}()$ with parameters $\theta^\pm_{\ell}$. The recursions are illustrated in the bottom panel of Fig. 1.

For MAP inference, we propose the following estimation functions $g^\pm_{\ell}()$: For $\ell = 1, \ldots, L - 2$, let $\theta_{\ell} = (\gamma_{\ell-1}, \gamma_{\ell})$, and define the energy function,

$$J_{\ell}(z^-_{\ell-1}, z^+_{\ell}; r^-_{\ell}, r^+_{\ell}, \theta_{\ell}) := -\ln p(z^+_{\ell}|z^-_{\ell-1}) + \frac{\gamma_{\ell-1}}{2}||z^-_{\ell-1} - r^-_{\ell}||^2 + \frac{\gamma_{\ell}}{2}||z^+_{\ell} - r^+_{\ell}||^2. \quad (3)$$

In the MMSE inference problem considered in [15], the estimation functions $g^\pm_{\ell}$ are given by the expectation with respect to the joint density, $p(z^\pm_{\ell-1}, z^\pm_{\ell}) \propto \exp[-J_{\ell}()]$. In this work, we consider the MAP estimation functions given by the mode of this density:

$$\left(\left(g^+_{\ell}(r^-_{\ell}, r^+_{\ell}, \theta_{\ell}), g^-_{\ell}(r^-_{\ell}, r^+_{\ell}, \theta_{\ell})\right) := (\hat{z}^-_{\ell-1}, \hat{z}^+_{\ell}) \right) \quad (4)$$

Algorithm 1 ML-VAMP

Require: Forward estimation functions $g^+_{\ell}()$, $\ell = 0, \ldots, L - 1$ and backward estimation functions $g^-_{\ell}()$, $\ell = 1, \ldots, L$.

1: Initialize $r_0^\ell = 0$
2: for $k = 0, 1, \ldots, N_{it} - 1$ do
3: // Forward Pass
4: $\hat{z}^+_{\ell} = g^+_{\ell}(r^-_{\ell}, \theta^+_{\ell})$
5: $\alpha^+_{\ell} = \frac{\partial g^+_{\ell}(r^-_{\ell}, \theta^+_{\ell})}{\partial r_{\ell}}$
6: $r^+_{\ell} = \frac{(\hat{z}^+_{\ell} - \alpha^+_{\ell} r^-_{\ell})}{(1 - \alpha^+_{\ell})}$
7: for $\ell = 0, \ldots, L - 1$ do
8: $\hat{z}^-_{\ell} = g^-_{\ell}(r^-_{\ell}, \theta^-_{\ell})$
9: $\alpha^-_{\ell} = \frac{\partial g^-_{\ell}(r^-_{\ell}, \theta^-_{\ell})}{\partial r_{\ell}}$
10: $r^-_{\ell} = \frac{(\hat{z}^-_{\ell} - \alpha^-_{\ell} r^-_{\ell})}{(1 - \alpha^-_{\ell})}$
11: end for
12:
13: // Reverse Pass
14: $\hat{z}^-_{L-1} = g^-_{L-1}(r^-_{L-1}, \theta^+_{L-1})$
15: $\alpha^-_{L} = \frac{\partial g^-_{L-1}(r^-_{L-1}, \theta^+_{L-1})}{\partial r^+_{L-1}}$
16: $r^-_{L-1} = \frac{(\hat{z}^-_{L-1} - \alpha^-_{L} r^+_{L-1})}{(1 - \alpha^-_{L})}$
17: for $\ell = L - 2, \ldots, 0$ do
18: $\hat{z}^-_{\ell} = g^-_{\ell+1}(r^-_{\ell+1}, \theta^-_{\ell+1})$
19: $\alpha^-_{\ell} = \frac{\partial g^-_{\ell+1}(r^-_{\ell+1}, \theta^-_{\ell+1})}{\partial r^+_{\ell+1}}$
20: $r^-_{\ell} = \frac{(\hat{z}^-_{\ell} - \alpha^-_{\ell} r^+_{\ell+1})}{(1 - \alpha^-_{\ell})}$
21: end for
22: end for
where
\[
(z_{\ell-1}, z_{\ell}^+) = \arg \min_{z_{\ell-1}, z_{\ell}^+} J_\ell(z_{\ell-1}, z_{\ell}^+; r_{\ell-1}, r_{\ell}, \theta_{\ell}).
\] (5)

Similar equations hold for \( \ell = 0 \) and \( \ell = L - 1 \) by removing the terms for \( \ell = 0 \) and \( L \).

In the MMSE inference in [15], the parameters \( \theta_{kl}^\pm \) are selected as,
\[
\theta_{kl}^\pm = (\gamma_{k\ell-1}^+, \gamma_{k\ell}^-), \quad \theta_{kl}^- = (\gamma_{k\ell+1,\ell-1}^+, \gamma_{k\ell}^-).
\] (6)

where the precision levels \( \gamma_{k\ell}^\pm \) are updated by the recursions,
\[
\gamma_{k\ell}^+ = \eta_{k\ell}^+ - \gamma_{k\ell}^-; \quad \eta_{k\ell}^+ = \gamma_{k\ell}^- / \alpha_{k\ell}^-;
\gamma_{k\ell+1,\ell} = \gamma_{k\ell}^- - \gamma_{k\ell}^+; \quad \eta_{k\ell+1,\ell} = \gamma_{k\ell}^- / \alpha_{k\ell^-}.
\] (7)

We can use the same updates for MAP ML-VAMP, although some of our analysis will apply to arbitrary parameterizations.

### III. Fixed Points and Connections to ADMM

Our first results relates MAP ML-VAMP to an ADMM-type minimization of the MAP objective (2). To simplify the presentation, we consider MAP estimation functions (4) with fixed values \( \gamma_{k\ell}^\pm > 0 \). Also, we replace the \( \alpha_{k\ell}^\pm \) updates in Algorithm 1 with fixed values,
\[
\alpha_{k\ell}^+ = \gamma_{k\ell}^- / \eta_{k\ell}, \quad \alpha_{k\ell}^- = \gamma_{k\ell}^+ / \eta_{k\ell}, \quad \text{and} \quad \eta_{k\ell} = \gamma_{k\ell}^- + \gamma_{k\ell}^+.
\] (8)

Now, to apply ADMM [24] to the MAP optimization (2), we use variable splitting where we replace each variable \( z_{\ell} \) with two copies \( z_{\ell}^+ \) and \( z_{\ell}^- \). Then, we define the objective function,
\[
F(z^+, z^-) := -\ln p(z_0^+) - \sum_{\ell=1}^{L-1} \ln p(z_{\ell}^+ | z_{\ell-1}^-) - \ln p(y | z_{L-1}^-),
\] (9)

over the groups of variables \( z^\pm = \{z_{\ell}^\pm \} \). The minimization in (2) is then equivalent to the constrained optimization,
\[
\min_{z^+, z^-} F(z^+, z^-) \text{ s.t. } z_{\ell}^+ = z_{\ell}^- \forall \ell.
\] (10)

Corresponding to this constrained optimization, define the augmented Lagrangian,
\[
L(z^+, z^-, s) = F(z^+, z^-) + \sum_{\ell=0}^{L-1} \eta_{s\ell} (s_{\ell}^+ - z_{\ell}^-) + \frac{\eta_{s\ell}}{2} \|s_{\ell}^+ - z_{\ell}^-\|^2,
\] (11)

where \( s = \{s_{\ell}\} \) is a set of dual parameters and \( \gamma_{k\ell}^\pm > 0 \) are weights and \( \eta_{s\ell} = \gamma_{k\ell}^+ + \gamma_{k\ell}^- \). Now, for \( \ell = 1, \ldots, L - 2 \), define
\[
L_\ell(z_{\ell-1}^+, z_{\ell}^+, z_{\ell}^-, s_{\ell}, s_{\ell-1}) := -\ln p(z_{\ell}^+ | z_{\ell-1}^-, s_{\ell}) + \eta_{s\ell} (s_{\ell}^+ - s_{\ell-1} z_{\ell-1}^-) + \frac{\gamma_{k\ell}^-}{2} \|z_{\ell}^- - z_{\ell-1}^-\|^2 + \frac{\gamma_{k\ell}^+}{2} \|z_{\ell}^+ - z_{\ell-1}^+\|^2,
\] (12)

which represents the terms in the Lagrangian \( L(\cdot) \) in (11) that contain \( z_{\ell-1}^- \) and \( z_{\ell}^- \). Similarly, define \( L_0(\cdot) \) and \( L_{L-1}(\cdot) \) using \( p(z_0^+) \) and \( p(y | z_{L-1}^-) \).

#### Theorem 1.
Consider the outputs of the ML-VAMP (Algorithm 1) with MAP estimation functions (4) for fixed \( \gamma_{k\ell}^\pm > 0 \). Suppose lines 9 and 19 are replaced with fixed values \( \alpha_{k\ell}^\pm = \alpha_{k\ell}^\in (0, 1) \) from (8). Let,
\[
s_{\ell}^- := \alpha_{k\ell}^+(z_{\ell-1}^- - r_{\ell-1}^+), \quad s_{\ell}^+ := \alpha_{k\ell}^-(r_{\ell+1}^+ - z_{\ell}^+). \] (13)

Then, the forward pass iterations satisfy,
\[
- z_{\ell}^+ = \arg \min_{(z_{\ell-1}^-, z_{\ell}^+)} L_\ell(z_{\ell-1}^-, z_{\ell}^+, z_{\ell}^+; z_{\ell}^-; s_{\ell}, s_{\ell-1}, s_{\ell+1,\ell}).
\] (14a)

whereas the backward pass iterations satisfy,
\[
- z_{\ell,\ell-1} \coloneqq = \arg \min_{(z_{\ell-1}^-, z_{\ell}^+)} L_\ell(z_{\ell-1}^-, z_{\ell}^+, z_{\ell}^+; z_{\ell}^-; s_{\ell}, s_{\ell-1}, s_{\ell+1,\ell}).
\] (15a)

for \( \ell = 0, \ldots, L - 1 \). Further, any fixed point of Algorithm 1 corresponds to a critical point of the Lagrangian (11).

As shown in the above result, the fixed \( (\alpha_{k\ell}^\in) \) version of ML-VAMP is an ADMM-type algorithm for solving the optimization problem (10). The full ML-VAMP algorithm adaptively updates \( (\alpha_{k\ell}^\in) \) to the take into account information regarding the curvature of the objective in (4). Note that in (14a) and (15a), we compute the joint minima over \( (z_{\ell-1}^+, z_{\ell}^+) \), but only use one of them at a time.

### IV. Analysis in the Large System Limit

As mentioned in the Introduction, the paper [15] provides an analysis of ML-VAMP with MMSE estimation functions in a certain large system limit (LSL). We extend this analysis to general estimators, including the MAP estimators (4). The LSL analysis has the same basic assumptions as [15]. Details are in the full paper and can be summarized as follows. We consider a sequence of problems indexed by \( N \). For each \( N \), and \( \ell = 1, 3, \ldots, L - 1 \), suppose that the weight matrix \( W_{\ell} \) has the SVD
\[
W_{\ell} = V_{\ell} \Sigma_{\ell} V_{\ell-1}^T, \quad \Sigma_{\ell} = \left[ \begin{array}{cc} \text{Diag}(s_{\ell}) & 0 \\ 0 & 0 \end{array} \right] \in \mathbb{R}^{N_{\ell} \times N_{\ell-1}},
\] (16)

where \( V_{\ell} \) and \( V_{\ell-1} \) are orthogonal matrices, the vector \( s_{\ell} = (s_{\ell1,1}, s_{\ell1,R_{\ell}}) \) contains singular values, and \( \text{rank}(W_{\ell}) \leq R_{\ell} \). Also, let \( b_{\ell} := V_{\ell}^T b \) and \( \xi_{\ell} := V_{\ell}^T \xi \). The number of layers \( L \) is fixed and the dimensions \( N_{\ell} = N_{\ell1}(N) \) and ranks \( R_{\ell} = R_{\ell}(N) \) in each layer are deterministic functions of \( N \). We assume that \( \lim_{N \to \infty} N_{\ell1}(N) \) and \( \lim_{N \to \infty} R_{\ell}(N) \) converge to non-zero constants, so that the dimensions grow linearly with \( N \).

For the estimation functions in the linear layers \( \ell = 1, 3, \ldots, L - 1 \), we assume that they are the MAP estimation functions (4), but the parameters \( \gamma_{k\ell}^+ \) and \( \gamma_{k\ell}^- \) can be chosen arbitrarily. Since the conditional density \( p(z_{\ell} | z_{\ell-1}) \) is given by the linear update (1a), the MAP estimation function (4) is
identical to the MMSE function and is given by a solution to a least squares problem. For the nonlinear layers, \( \ell = 0, 2, \ldots, L \), the estimation functions \( g_\ell(\cdot) \) can be arbitrary as long as they operate elementwise and are Lipschitz continuous. For simplicity, we will assume that for all the estimation functions, the parameters \( \theta_{\ell, k} \) are deterministic and fixed. However, data dependent parameters can also be considered as in [29].

We follow the analysis methodology in [30], and assume that the signal realization \( z_{k, \ell}^{0, \ell} \in \mathbb{R}^{N_{\ell}} \) for \( \ell = 0 \), and the noise realizations \( \xi_{\ell} \) in the nonlinear stages \( \ell = 2, 4, \ldots, L \), all converge empirically to random variables \( Z^0 \) and \( \Xi_{\ell} \). For the linear stages \( \ell = 1, 3, \ldots, L - 1 \), let \( \bar{s}_{\ell} \) be the zero-padded singular value vector, and we assume that \( (\bar{s}_{\ell}, \mathbf{b}_{\ell}, \xi_{\ell}) \) also converge empirically.

Now define the quantities

\[
\begin{align*}
q^0_k &:= z^0_k, & p^0_k &:= V_k z^0_k = V_k z^0_0, & \ell = 0, 2, \ldots, L, \\
q^0_k &:= V_k^T z^0_k, & p^0_k &:= \hat{v}_k = V_k q^0_k, & \ell = 1, 3, \ldots, L - 1,
\end{align*}
\]

which represent the true vectors \( z^0_k \) and their transforms. For \( \ell = 0, 2, \ldots, L - 2 \), we next define the vectors:

\[
\begin{align*}
\hat{q}^0_{k, \ell} &:= q^0_k, & q^\pm_{k, \ell} &:= r_{k, \ell} - z^0_k, \\
\hat{p}^0_{k, \ell+1} &:= z^0_k, & p^\pm_{k, \ell+1} &:= r_{k, \ell+1} - z^0_k, \\
\hat{q}^0_{k, \ell+1} &:= V_k^T p^\pm_{k, \ell+1}, & q^\pm_{k, \ell+1} &:= V_k^T p^\pm_{k, \ell+1}, \\
\hat{p}^0_{k, \ell} &:= V_k q^\pm_{k, \ell}.
\end{align*}
\]

The vectors \( \hat{q}^0_{k, \ell} \) and \( \hat{p}^0_{k, \ell} \) represent the estimates of \( q^0_k \) and \( p^0_k \). Also, the vectors \( q^\pm_{k, \ell} \) and \( p^\pm_{k, \ell} \) are the differences \( r_{k, \ell} - z^0_k \) or their transforms. These represent errors on the inputs \( r_{k, \ell} \) to the estimation functions \( g^0_{\ell}(\cdot) \).

**Theorem 2.** Under the above assumptions, for any fixed iteration \( k \) and \( \ell = 1, 2, \ldots, L-1 \), the components of \( p^0_{k, \ell+1}, q^0_{k, \ell+1}, \hat{q}^0_{k, \ell+1} \) are Gaussian random variables with

\[
\begin{align*}
\text{Cov}(P^0_{\ell, k, \ell+1}, D^+_{\ell, k, \ell+1}) &= K^+_{k, \ell+1}, & \mathbb{E}(Q^0_{\ell, k, \ell+1}) &= \tau^0_{k, \ell+1}, \\
\mathbb{E}(P^0_{\ell, k, \ell+1} Q^0_{\ell, k, \ell+1}) &= 0, & \mathbb{E}(D^+_{\ell, k, \ell+1} Q^0_{\ell, k, \ell+1}) &= 0,
\end{align*}
\]

for parameters \( K^+_{k, \ell+1} \) and \( \tau^0_{k, \ell+1} \). The identical result holds for \( \ell = 0 \) with the variables \( p^0_{k, \ell}, q^0_{k, \ell} \) and \( D^+_{k, \ell+1} \) removed.

Also, a similar result holds for the variables \( p^0_{k, \ell+1}, q^0_{k, \ell+1}, \hat{q}^0_{k, \ell+1} \).

The full paper [28] provides a precise and simple description of the limiting random variables on the right hand side of (19). In addition, the parameters \( K^+_{k, \ell+1} \) and \( \tau^0_{k, \ell+1} \) can be computed by deterministic recursive formulas, thus representing a state evolution (SE) for the MAP ML-VAMP system. In the case of MMSE estimation functions, the SE equations reduce to those of [29].

The importance of this limiting model is that we can compute several important performance metrics of the ML-VAMP system. For example, for a non-linear layer, \( \ell = 0, 2, \ldots, L \), it is shown in the full paper [28] that the asymptotic mean-squared error (MSE) is given by,

\[
\lim_{N \to \infty} \frac{1}{N} \|Z^0_k - \hat{Z}^0_{k, \ell}\|^2 = \mathbb{E}(Q^0_{k, \ell} - \hat{Q}^0_{k, \ell})^2,
\]

where the expectation can be computed from the model from the random variables in (19). In this way, we see that MAP ML-VAMP provides a computationally tractable method for computing critical points of the MAP objective with precise predictions on its performance.

V. NUMERICAL SIMULATIONS

To validate the MAP ML-VAMP algorithm and the LSL analysis, we simulate the method in a random synthetic network similar to [29]. Details are given in Appendix ?? specifically, we consider a network with \( N_y = 20 \) inputs and two hidden stages with 100 and 500 units with ReLU activations. The number of outputs is \( N_y \) is varied. In the final layer, AWGN noise is added at an SNR of 20 dB. The weight matrices have Gaussian i.i.d. components and the biases \( b_\ell \) are selected so that the ReLU outputs are non-zero, on average, for 40% of the samples. For each value of \( N_y \), we generate 40 random instances of the network and compute (a) the MAP estimate using the Adam optimizer [31] in Tensorflow; (b) the estimate from MAP ML-VAMP; and (c) the MSE for MAP ML-VAMP predicted by the state evolution. Fig. 2 shows the median and normalized MSE, \( 10 \log_{10}(\|Z^0_k - \hat{Z}^0_{k, \ell}\|^2/\|Z^0_k\|^2) \) for the input variable \( (\ell = 0) \) for the three methods. We see that for \( N_y \geq 100 \), the actual performance of MAP ML-VAMP matches the SE closely as well as the performance of MAP estimation via a generic solver. For \( N_y < 100 \), the match is still close, but there is a small discrepancy, likely due to the relatively small size of the problem. Also, for small \( N_y \), MAP ML-VAMP appears to achieve a slightly better performance than the Adam optimizer. Since both are optimizing the same objective, the difference is likely due to the ML-VAMP finding better local minima.

To demonstrate that MAP ML-VAMP can also work on a simple non-random dataset, Fig. 3 shows samples of reconstructions results for inpainting for MNIST digits. A VAE [2] is used to train a generative model. The MAP ML-VAMP reconstruction obtains similar results as MAP inference using the Adam optimizer, although sometimes different local minima are found. The main benefit is that MAP ML-VAMP can be rigorously analyzed. Details are in the full paper [28].

**CONCLUSIONS**

ML-VAMP with MAP estimation provides a computationally tractable method for performing the MAP inference with performance that can be rigorously and precisely characterized in a certain large system limit. The approach thus offers a new and potentially powerful approach for understanding and improving deep network-based inference.
Fig. 2: Normalized MSE for a random multi-layer network for (a) MAP inference computed by Adam optimizer; (b) MAP inference from ML-VAMP; (c) State evolution prediction.

Fig. 3: MNIST inpainting where the rows 10-20 of the 28 × 28 digits are erased.

ACKNOWLEDGEMENTS

AKF was supported by NSF grants 1254204 and 1738286, and ONR N00014-15-1-2677. SR was supported by NSF Grants 1116589, 1302336, and 1547332, NIST award 70NANB17H166, SRC, and NYU WIRELESS.

REFERENCES

[1] D. J. Rezende, S. Mohamed, and D. Wierstra, “Stochastic backpropagation and approximate inference in deep generative models,” in Proc. ICML, 2014, pp. 1278–1286.
[2] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” arXiv preprint arXiv:1312.6114, 2013.
[3] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” arXiv preprint arXiv:1511.06434, 2015.
[4] R. Salakhutdinov, “Learning deep generative models,” Annual Review of Statistics and Its Application, vol. 2, pp. 361–385, 2015.
[5] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with perceptual and contextual losses,” arXiv:1607.07539, 2016.
[6] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, “Compressed sensing using generative models,” Proc. ICML, 2017.
[7] J. R. Chang, C.-L. Li, B. Poczos, and B. V. Kumar, “One network to solve them all—solving linear inverse problems using deep projection models,” in 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017, pp. 5889–5898.
[8] A. Mousavi, A. B. Patel, and R. G. Baraniuk, “A deep learning approach to structured signal recovery,” in Proc. IEEE Allerton Conference, 2015, pp. 1336–1343.
[9] C. Metzler, A. Mousavi, and R. Baraniuk, “Learned D-amp: Principled neural network based compressive image recovery,” in Proc. NIPS, 2017, pp. 1772–1783.
[10] M. Borgerding, P. Schniter, and S. Rangan, “AMP-inspired deep networks for sparse linear inverse problems,” IEEE Transactions on Signal Processing, vol. 65, no. 16, pp. 4293–4308, 2017.
[11] P. Hand and V. Vorontinski, “Global guarantees for enforcing deep generative priors by empirical risk,” arXiv preprint arXiv:1705.07576, 2017.
[12] V. Shah and C. Hegde, “Solving linear inverse problems using gan priors: An algorithm with provable guarantees,” arXiv preprint arXiv:1802.08406, 2018.
[13] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” PNAS, vol. 106, no. 45, pp. 18914–18919, Nov. 2009.
[14] A. Manoel, F. Krzakala, M. Mézard, and L. Zdeborová, “Multi-layer generalized linear estimation,” arXiv:1701.06981, 2017.
[15] A. K. Fletcher, S. Rangan, and P. Schniter, “Inference in deep networks in high dimensions,” Proc. IEEE ISIT, 2018.
[16] M. Gabrié, A. Manoel, C. Luneau, J. Barbier, N. Macris, F. Krzakala, and L. Zdeborová, “Entropy and mutual information in models of deep neural networks,” in Proc. NIPS, 2018.
[17] G. Reeves, “Additivity of information in multilayer networks via additive gaussian noise transforms,” arXiv preprint arXiv:1710.04580, 2017.
[18] S. Rangan, P. Schniter, and A. K. Fletcher, “Vector approximate message passing,” in Proc. IEEE ISIT, 2017, pp. 1588–1592.
[19] J. Ma and L. Ping, “Orthogonal AMP,” IEEE Access, vol. 5, pp. 2020–2033, 2017.
[20] K. Takeuchi, “Rigorous dynamics of expectation-propagation-based signal recovery from unitarily invariant measurements,” in Proc. IEEE ISIT, 2017, pp. 501–505.
[21] T. P. Minka, “Expectation propagation for approximate bayesian inference,” in Proc. Uncertainty in artificial intelligence, 2001, pp. 362–369.
[22] M. Opper and O. Winther, “Expectation consistent approximate inference,” Journal of Machine Learning Research, vol. 6, no. Dec, pp. 2177–2204, 2005.
[23] A. K. Fletcher, S. Rangan, and P. Schniter, “Inference in deep networks using generative models,” in Proc. ICML, 2015.
[24] B. Cakmak, O. Winther, and B. H. Fleury, “S-AMP: Approximate message passing for general matrix ensembles,” in Proc. IEEE ITW, 2014.
[25] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein et al., “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Foundations and Trends® in Machine learning, vol. 3, no. 1, pp. 1–122, 2011.
[26] S. Rangan, P. Schniter, E. Riegler, A. K. Fletcher, and V. Cevher, “Fixed points of generalized approximate message passing with arbitrary matrices,” IEEE Trans. Info. Theory, vol. 62, no. 12, pp. 7464–7474, 2016.
[27] S. Rangan, A. K. Fletcher, P. Schniter, and U. S. Kamilov, “Inference for generализed linear models via alternating directions and Bethe free energy minimization,” IEEE Trans. Info. Theory, vol. 63, no. 1, pp. 676–697, 2017.
[28] A. Manoel, F. Krzakala, G. Varoquaux, B. Thirion, and L. Zdeborová, “Approximate message-passing for convex optimization with non-separable penalties,” arXiv preprint arXiv:1809.06304, 2018.
[29] P. Pandit, M. Sahraee, S. Rangan, and A. K. Fletcher, “Asymptotics of map inference in deep networks,” arXiv preprint arXiv:1903.01293, 2019.
[30] M. Bayati and A. Montanari, “The dynamics of message passing on dense graphs, with applications to compressed sensing,” arXiv preprint arXiv:1905.10930, 2019.
[31] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.