Inference in Deep Networks in High Dimensions

Alyson K. Fletcher,* Sundeep Rangan,† and Philip Schniter‡
*UCLA, akfletcher@ucla.edu †NYU, srangan@nyu.edu, ‡Ohio State, schniter.1@osu.edu,

***Abstract*—Deep generative networks provide a powerful tool for modeling complex data in a wide range of applications. In inverse problems that use these networks as generative priors on data, one must often perform inference of the inputs of the networks from the outputs. Inference is also required for sampling during stochastic training of these generative models. This paper considers inference in a deep stochastic neural network where the parameters (e.g., weights, biases and activation functions) are known and the problem is to estimate the values of the input and hidden units from the output. A novel and computationally tractable inference method called Multi-Layer Vector Approximate Message Passing (ML-VAMP) is presented. Our main contribution shows that the mean-squared error (MSE) of ML-VAMP can be exactly predicted in a certain large system limit. In addition, the MSE achieved by ML-VAMP matches the Bayes optimal value recently postulated by Reeves when certain fixed point equations have unique solutions.**

I. Introduction

Deep neural networks are increasingly used for describing probabilistic generative models of complex data such as images, audio and text. This paper considers the inference problem of estimating the input and hidden units of an (already trained) deep neural network from its output. The problem arises, for example, in image reconstruction where a deep network is used as a generative prior of an image with additional layers added to model the measurements (such as blurring, occlusion or noise) [1], [2]. While optimal inference is generally intractable, there are several methods that have worked well in practice, including MAP estimation via gradient descent [1], [2] and the use of a separate learned deep network, as is done in variational autoencoders [3], [4] and adversarial networks [5]. However, similar to the situation in deep learning in general, there are few analytic tools for understanding how these algorithms perform or how far the estimates are from optimal.

In this work, we address this shortcoming by considering inference based on approximate message passing (AMP) [6]. A recent variant of AMP, called multi-layer AMP, has been proposed for inference in deep networks [7]. That work characterizes the replica prediction for optimality in multi-layer networks and argues that the proposed ML-AMP method can achieve this optimal inference in certain scenarios. Unfortunately, the convergence of ML-AMP in [7] is not rigorously proven. In addition, ML-AMP assumes Gaussian i.i.d. weight matrices $\mathbf{W}_\ell$, and it is well-known that AMP methods often fail to converge when this assumption does not hold [8], [9], [10], [11], [12].

A. K. Fletcher is supported in part by National Science Foundation grants 1254204 and 1564278 as well as the Office of Naval Research grant N00014-15-1-2677. S. Rangan is supported in part by the National Science Foundation under Grants 1302336, 1564142, and 1547332. P. Schniter is supported in part by the National Science Foundation grant CCF-1527162.

In this work, we present a novel AMP method called multi-layer vector AMP (ML-VAMP) that builds on the recent VAMP method of [13] and its extensions to generalized linear models (GLMs) in [14], [15]. The VAMP algorithm of [13] was itself derived from the expectation consistent approximate inference framework of [16], [17], [18] and applies to the special case of a single linear layer. The ML-VAMP algorithm proposed here extends the VAMP method to networks with multiple layers and separable nonlinearities.

We analyze ML-VAMP in a setting where the number of layers is fixed and the weight matrices are orthogonally invariant random matrices with dimensions that grow to infinity. This class of random matrices is much larger than i.i.d. Gaussian ensembles. Importantly, it includes weight matrices with arbitrary condition numbers, which is known to be the main failure mechanism in conventional AMP convergence [8]. Our main theoretical contribution (Theorem 1) shows that the mean squared error (MSE) of ML-VAMP algorithm can be precisely predicted by a simple set of scalar state evolution (SE) equations. In addition, a recent work by Reeves [19] has shown that the fixed point equations for the MSE of ML-VAMP exactly match those of the postulated optimal MSE as predicted by information theoretic techniques. Hence, ML-VAMP may be Bayes optimal when certain fixed point equations have unique solutions. ML-VAMP thus enables computationally tractable inference with rigorous analysis of its performance and testable conditions for optimality. A full version of this paper is available in [20], which includes proofs, simulation details, and further discussion of previous work.

II. ML-VAMP Algorithm

We consider the following $L/2$-layer (for even L) neural-network-based generative stochastic model: A random input $\mathbf{z}_0$ with some density $p(\mathbf{z}_0)$ generates a sequence of vectors, $\mathbf{z}_\ell \in \mathbb{R}^{N_\ell}$, $\ell = 1, \ldots, L$, through operations of the form

$$\mathbf{z}_\ell = \mathbf{W}_\ell \mathbf{z}_{\ell-1} + \mathbf{b}_\ell + \boldsymbol{\xi}_\ell, \quad \boldsymbol{\xi}_\ell \sim \mathcal{N}(\mathbf{0}, \nu_\ell^{-1}\mathbf{I}), \quad \ell = 1, 3, \ldots, L-1 \tag{1a}$$

$$\mathbf{z}_\ell = \boldsymbol{\phi}_\ell(\mathbf{z}_{\ell-1}, \boldsymbol{\xi}_\ell), \quad \boldsymbol{\xi}_\ell \sim p(\boldsymbol{\xi}_\ell), \quad \ell = 2, 4, \ldots, L. \tag{1b}$$

Equation (1a) describes the *linear stages* of the network, which are defined by the weight matrices $\mathbf{W}_\ell$, the bias vectors $\mathbf{b}_\ell$, and the Gaussian noise terms $\boldsymbol{\xi}_\ell$. Equation (1b) describes the *nonlinear stages*, which involve the activation functions $\boldsymbol{\phi}_\ell(\cdot)$

978-1-5386-4780-6/18/$31.00©2018 IEEE

and the possibly non-Gaussian noise terms $\boldsymbol{\xi}_\ell$. We will assume separable $\boldsymbol{\phi}_\ell(\cdot)$ and i.i.d. $\boldsymbol{\xi}_\ell$, i.e.,

$$[\boldsymbol{\phi}_\ell(\mathbf{z}_{\ell-1}, \boldsymbol{\xi}_\ell)]_n = \phi_\ell(z_{\ell-1,n}, \xi_{\ell,n}) \tag{2}$$

$$p(\boldsymbol{\xi}_\ell) = \prod_{n=1}^{N_\ell} p(\xi_{\ell,n}), \tag{3}$$

for scalar-valued functions $\phi_\ell(\cdot)$. This model covers many activation functions commonly used in neural networks, including rectified linear units (ReLUs) and sigmoids. The final output $\mathbf{z}_L$ is observed.

We consider the problem of estimating the hidden network variables $\mathbf{z}_\ell$, $\ell = 0, \ldots, L-1$ from the observed output $\mathbf{y} = \mathbf{z}_L$. Importantly, the activation functions $\phi_\ell(\cdot)$ noise precisions ν_ℓ, weight matrices $\mathbf{W}_\ell$, and bias terms $\mathbf{b}_\ell$ are known (i.e., already trained). Thus, we do not consider the *learning problem*.

The proposed ML-VAMP algorithm for this inference problem is shown in Algorithm 1. It can be derived as an extension of the GEC-SR algorithm [14] proposed for inference in a GLM, which is a special case of our multi-layer problem with $L = 2$ stages (i.e., one layer). The GEC-SR can also be derived from TAP methods [21], [22]. We take a Bayesian approach, where $\mathbf{z}_0$ is i.i.d. with known density $p(\mathbf{z}_0)$. The noise terms $\boldsymbol{\xi}_\ell$ are independent random vectors, so that the sequence $\mathbf{z}_\ell$ in (1) is Markov. As described in the full paper [20], the ML-VAMP algorithm can be derived similar to GEC-SR using a Gaussian approximation of belief propagation on the factor-graph representation of the Markov chain. The quantities $\mathbf{r}_{k\ell}^+$ and $\gamma_{k\ell}^+$ represent the mean and precision (inverse variance) of the Gaussian messages in the forward direction, and $\mathbf{r}_{k\ell}^+$ and $\gamma_{k\ell}^-$ represent the same quantities in the reverse direction.

The terms $\mathbf{r}_{k\ell}^\pm$ and $\gamma_{k\ell}^\pm$ are updated by *estimation functions* $\mathbf{g}_\ell^\pm(\cdot)$ that are defined as follows. For $\ell = 1, \ldots, L-1$, we first define the *belief*

$$b_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1} | \mathbf{r}_{\ell-1}^+, \mathbf{r}_\ell^-, \gamma_{\ell-1}^+, \gamma_\ell^-) \propto \exp\left[-H_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1})\right], \tag{4}$$

which is a probability density, using the energy function

$$H_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1}) := -\ln p(\mathbf{z}_\ell | \mathbf{z}_{\ell-1}) + \frac{\gamma_\ell^-}{2}\|\mathbf{z}_\ell - \mathbf{r}_\ell^-\|^2 + \frac{\gamma_{\ell-1}^+}{2}\|\mathbf{z}_{\ell-1} - \mathbf{r}_{\ell-1}^+\|^2. \tag{5}$$

At each iteration k, the belief (4) represents an estimate of the posterior density $p(\mathbf{z}_{\ell-1}, \mathbf{z}_\ell | \mathbf{y})$. The estimation functions $\mathbf{g}_\ell^\pm(\cdot)$ are defined as the functions that compute the expected value of $\mathbf{z}_{\ell-1}$ and $\mathbf{z}_\ell$ with respect to that belief, i.e.,

$$\mathbf{g}_\ell^+(\mathbf{r}_{\ell-1}^+, \mathbf{r}_\ell^-, \gamma_{\ell-1}^+, \gamma_\ell^-) = \mathbb{E}\left[\mathbf{z}_\ell | \mathbf{r}_{\ell-1}^+, \mathbf{r}_\ell^-, \gamma_{\ell-1}^+, \gamma_\ell^-\right], \tag{6a}$$

$$\mathbf{g}_\ell^-(\mathbf{r}_{\ell-1}^+, \mathbf{r}_\ell^-, \gamma_{\ell-1}^+, \gamma_\ell^-) = \mathbb{E}\left[\mathbf{z}_{\ell-1} | \mathbf{r}_{\ell-1}^+, \mathbf{r}_\ell^-, \gamma_{\ell-1}^+, \gamma_\ell^-\right], \tag{6b}$$

where the expectations are with respect to the density b_ℓ in (4). For the end points $\ell = 0$ and L in the factor graph, we define

$$b_0(\mathbf{z}_0 | \mathbf{r}_0^-, \gamma_0^-), \quad b_L(\mathbf{z}_{L-1} | \mathbf{r}_{L-1}^+, \gamma_{L-1}^+),$$

similar to (4)-(5), but in the case of b_0 we omit the $\mathbf{r}_{\ell-1}^+$ term and replace $p(\mathbf{z}_\ell | \mathbf{z}_{\ell-1})$ by $p(\mathbf{z}_0)$ in (5), and in the case of b_L we omit the $\mathbf{r}_\ell^-$ term in (5).

Algorithm 1 ML-VAMP

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

Initialize $\mathbf{r}_{0\ell}^- = \mathbf{0}$, $\gamma_{0\ell}^- = 0$, $\ell = 0, \ldots, L-1$.
for $k = 0, 1, \ldots, N_{\text{it}} - 1$ **do**
  // Forward Pass
  for $\ell = 0, \ldots, L-1$ **do**
    if $\ell = 0$ **then**
      $\widehat{\mathbf{z}}_{k\ell}^+ = \mathbf{g}_\ell^+(\mathbf{r}_{k\ell}^-, \gamma_{k\ell}^-)$
      $\alpha_{k\ell}^+ = \langle \partial \mathbf{g}_\ell^+(\mathbf{r}_{k\ell}^-, \gamma_{k\ell}^-)/\partial \mathbf{r}_{k\ell}^- \rangle$
    else
      $\widehat{\mathbf{z}}_{k\ell}^+ = \mathbf{g}_\ell^+(\mathbf{r}_{k,\ell-1}^+, \mathbf{r}_{k\ell}^-, \gamma_{k,\ell-1}^+, \gamma_{k\ell}^-)$
      $\alpha_{k\ell}^+ = \langle \partial \mathbf{g}_\ell^+(\mathbf{r}_{k,\ell-1}^+, \mathbf{r}_{k\ell}^-, \gamma_{k\ell-1}^+, \gamma_{k\ell}^-)/\partial \mathbf{r}_{k\ell}^- \rangle$
    end if
    $\gamma_{k\ell}^+ = \eta_{k\ell}^+ - \gamma_{k\ell}^-$, $\eta_{k\ell}^+ = \gamma_{k\ell}^-/\alpha_{k\ell}^+$
    $\mathbf{r}_{k\ell}^+ = (\eta_{k\ell}^+ \widehat{\mathbf{z}}_{k\ell}^+ - \gamma_{k\ell}^- \mathbf{r}_{k\ell}^-)/\gamma_{k\ell}^+$
  end for
  // Reverse Pass
  for $\ell = L-1, \ldots, 0$ **do**
    if $\ell = L-1$ **then**
      $\widehat{\mathbf{z}}_{k\ell}^- = \mathbf{g}_{\ell+1}^-(\mathbf{r}_{k\ell}^+, \gamma_{k\ell}^+,)$
      $\alpha_{k\ell}^- = \langle \partial \mathbf{g}_{\ell+1}^-(\mathbf{r}_{k\ell}^+, \gamma_{k\ell}^+)/\partial \mathbf{r}_{k\ell}^+ \rangle$
    else
      $\widehat{\mathbf{z}}_{k\ell}^- = \mathbf{g}_{\ell+1}^-(\mathbf{r}_{k\ell}^+, \mathbf{r}_{k+1,\ell+1}^-, \gamma_{k\ell}^+, \gamma_{k+1,\ell+1}^-)$
      $\alpha_{k\ell}^- = \langle \partial \mathbf{g}_{\ell+1}^-(\mathbf{r}_{k\ell}^+, \mathbf{r}_{k+1,\ell+1}^-, \gamma_{k\ell}^+, \gamma_{k+1,\ell+1}^-)/\partial \mathbf{r}_{k\ell}^+ \rangle$
    end if
    $\gamma_{k+1,\ell}^- = \eta_{k\ell}^- - \gamma_{k\ell}^+$, $\eta_{k\ell}^- = \gamma_{k\ell}^+/\alpha_{k\ell}^-$
    $\mathbf{r}_{k+1,\ell}^- = (\eta_{k\ell}^- \widehat{\mathbf{z}}_{k\ell}^- - \gamma_{k\ell}^+ \mathbf{r}_{k\ell}^+)/\gamma_{k+1,\ell}^+$
  end for
end for

We also use the following notation. For any vector $\mathbf{u} \in \mathbb{R}^N$, $\langle \mathbf{u} \rangle := (1/N)\sum_{n=1}^N u_n$, which is the empirical average over the components. For a matrix $\mathbf{Q} \in \mathbb{R}^{N \times N}$, we let $\langle \mathbf{Q} \rangle = (1/N)\mathrm{Tr}(\mathbf{Q})$, which is the average of the diagonal components. Finally, $\partial \mathbf{g}_\ell^\pm$ denotes the Jacobian of the estimation function $\mathbf{g}_\ell^\pm : \mathbb{R}^{N\ell} \to \mathbb{R}^{N\ell}$ with respect to its first argument.

As shown in the full paper [20], an appealing feature of ML-VAMP is that the estimation functions $\mathbf{g}_\ell^\pm(\cdot)$ can be easily computed for the network (1). By this, we mean the following.

Nonlinear stages: Consider $\ell \in \{2, 4, \ldots, L\}$, corresponding to a nonlinear stage (1b). For separable activation function $\boldsymbol{\phi}_\ell(\cdot)$ and i.i.d. noise $\boldsymbol{\xi}_\ell$, the estimation functions (6) are themselves separable in that $[\mathbf{g}_\ell^\pm(\mathbf{r}_{\ell-1}, \mathbf{r}_\ell, \gamma_{\ell-1}^+, \gamma_\ell^-)]_n = g_\ell^\pm(r_{\ell-1,n}, r_{\ell,n}, \gamma_{\ell-1}^+, \gamma_\ell^-)$. The corresponding scalar estimation functions $g_\ell^\pm$ can often be evaluated analytically, or if not by two-dimensional numerical integration.

Linear stages: Consider $\ell \in \{1, 3, \ldots, L-1\}$. Since the transformation in (1a) is linear and the noise is Gaussian, the belief (4) is also Gaussian. Therefore, the expectation in (6) and covariance can be computed in closed form. For the purpose of analysis, we compute the estimate using an SVD. Specifically,

suppose that the weight matrix $\mathbf{W}_\ell$ has the SVD

$$\mathbf{W}_\ell = \mathbf{V}_\ell \mathbf{\Sigma}_\ell \mathbf{V}_{\ell-1}, \quad \mathbf{\Sigma}_\ell = \begin{bmatrix} \mathrm{Diag}(\mathbf{s}_\ell) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{N_\ell \times N_{\ell-1}}, \tag{7}$$

where $\mathbf{V}_\ell$ and $\mathbf{V}_{\ell-1}$ are orthogonal matrices, the vector $\mathbf{s}_\ell = (s_{\ell 1}, \ldots, s_{\ell R_\ell})$ contains singular values, and $\mathrm{rank}(\mathbf{W}_\ell) \le R_\ell$. Also, let $\bar{\mathbf{b}}_\ell := \mathbf{V}_\ell^{\mathsf{T}} \mathbf{b}_\ell$ and $\bar{\boldsymbol{\xi}}_\ell := \mathbf{V}_\ell^{\mathsf{T}} \boldsymbol{\xi}_\ell$ so that

$$\mathbf{b}_\ell = \mathbf{V}_\ell \bar{\mathbf{b}}_\ell, \quad \boldsymbol{\xi}_\ell = \mathbf{V}_\ell \bar{\boldsymbol{\xi}}_\ell, \tag{8}$$

Then, as shown in the full paper [20], the estimation functions (6) reduce to matrix-vector multiplications with $\mathbf{V}_\ell$ and $\mathbf{V}_\ell^{\mathsf{T}}$ and scalar inversions.

III. State Evolution Analysis of ML-VAMP

A. Large System Limit Model

Our main contribution is to rigorously analyze ML-VAMP in a certain large system limit (LSL). The LSL analysis is widely-used in studying AMP algorithms and their variants [23], [13]. For this, we consider a sequence of problems indexed by N. The number of stages L is fixed and the dimensions $N_\ell = N_\ell(N)$ and ranks $R_\ell = R_\ell(N)$ in each stage are deterministic functions of N. We assume that $\lim_{N\to\infty} N_\ell/N$ and $\lim_{N\to\infty} R_\ell/N$ converge to non-zero constants, so that the dimensions grow linearly with N. We follow the framework of Bayati and Montanari [23], which models various sequences as deterministic, but with components converging empirically to a distribution. See [20] for a review of this framework. Specifically, let us denote the "true" realization of $\mathbf{z}_\ell$ using the superscripted variable $\mathbf{z}_\ell^0$. Then we assume that the signal realization $\mathbf{z}_\ell^0 \in \mathbb{R}^{N_0}$ for $\ell = 0$, and the noise realizations $\boldsymbol{\xi}_\ell$ in the nonlinear stages $\ell = 2, 4, \ldots, L$, all converge empirically to random variables Z^0 and Ξ_ℓ, i.e.,

$$\lim_{N\to\infty} \left\{ z_{0,n}^0 \right\} \overset{PL(2)}{=} Z_0^0, \quad \lim_{N\to\infty} \left\{ \xi_{\ell,n} \right\} \overset{PL(2)}{=} \Xi_\ell, \quad \ell = 2m, \tag{9}$$

For the linear stages $\ell = 1, 3, \ldots, L$–1, let $\bar{\mathbf{s}}_\ell$ be a version of the singular-value vector $\mathbf{s}_\ell$ zero-padded to length N_ℓ. We assume that $\bar{\mathbf{s}}_\ell$, the transformed bias $\bar{\mathbf{b}}_\ell = \mathbf{V}_\ell^{\mathsf{T}} \mathbf{b}_\ell$, and the transformed noise realization $\bar{\boldsymbol{\xi}}_\ell = \mathbf{V}_\ell^{\mathsf{T}} \boldsymbol{\xi}_\ell$ all converge empirically as

$$\lim_{N\to\infty} \left\{ \bar{s}_{\ell,n}, \bar{b}_{\ell,n}, \bar{\xi}_{\ell,n} \right\} \overset{PL(2)}{=} (\bar{S}_\ell, \bar{B}_\ell, \bar{\Xi}_\ell), \quad \ell = 2m+1, \tag{10}$$

to independent random variables $\bar{S}_\ell$, $\bar{B}_\ell$, and $\bar{\Xi}_\ell$, with $\bar{\Xi}_\ell \sim \mathcal{N}(0, \nu_\ell^{-1})$, where ν_ℓ is the noise precision. We assume that $\bar{S}_\ell \ge 0$ and $\bar{S}_\ell \le S_{\max}$ for some upper bound $S_{\max}$.

We assume that the matrices $\mathbf{V}_\ell$ are Haar distributed (i.e., uniform on the set of $N_\ell \times N_\ell$ orthogonal matrices) as well as independent of one another. For any linear stage ℓ, the weight matrix $\mathbf{W}_\ell$, bias $\mathbf{b}_\ell$, and noise $\boldsymbol{\xi}_\ell$ are then generated from (7) and (8). Finally, the true $\mathbf{z}_\ell^0$ are generated from the recursions,

$$\mathbf{z}_\ell^0 = \mathbf{W}_\ell \mathbf{z}_{\ell-1}^0 + \mathbf{b}_\ell + \boldsymbol{\xi}_\ell, \quad \ell = 1, 3, \ldots, L-1 \tag{11a}$$

$$\mathbf{z}_\ell^0 = \boldsymbol{\phi}_\ell(\mathbf{z}_{\ell-1}^0, \boldsymbol{\xi}_\ell), \quad \ell = 2, 4, \ldots, L. \tag{11b}$$

Algorithm 2 ML-VAMP State Evolution

Require: Random variables Z_0^0, Ξ_ℓ, $\bar{B}_\ell$, $\bar{S}_\ell$, $\bar{\Xi}_\ell$.

Initialize $\overline{\gamma}_{0\ell}^- = 0$
$Q_0^0 = Z_0^0, \quad P_0 \sim \mathcal{N}(0, \tau_0^0), \quad \tau_0^0 = \mathbb{E}(Q_0^0)^2$
for $\ell = 1, 2, \ldots, L-1$ **do**
  if ℓ is odd **then**
    $Q_\ell^0 = \bar{S}_\ell P_{\ell-1}^0 + \bar{B}_\ell + \bar{\Xi}_\ell$
  else
    $Q_\ell^0 = \phi_\ell(P_{\ell-1}^0, \Xi_\ell)$
  end if
  $P_\ell^0 = \mathcal{N}(0, \tau_\ell^0), \quad \tau_\ell^0 = \mathbb{E}(Q_\ell^0)^2$
end for

for $k = 0, 1, \ldots$ **do**
  // Forward Pass
  $\overline{\eta}_{k0}^+ = 1/\mathcal{E}_0^+(\overline{\gamma}_{k0}^-)$
  $\overline{\gamma}_{k0}^+ = \overline{\eta}_{k0}^+ - \overline{\gamma}_{k0}^-, \quad \overline{\alpha}_{k0}^+ = \overline{\gamma}_{k0}^+ / \overline{\eta}_{k0}^+$
  for $\ell = 1, \ldots, L-1$ **do**
    $\overline{\eta}_{k\ell}^+ = 1/\mathcal{E}_\ell^+(\overline{\gamma}_{k,\ell-1}^+, \overline{\gamma}_{k\ell}^-, \tau_{\ell-1}^0)$
    $\overline{\gamma}_{k\ell}^+ = \overline{\eta}_{k\ell}^+ - \overline{\gamma}_{k\ell}^-, \quad \overline{\alpha}_{k\ell}^+ = \overline{\gamma}_{k\ell}^+ / \overline{\eta}_{k\ell}^+$
  end for

  // Reverse Pass
  $\overline{\eta}_{k,L-1}^- = 1/\mathcal{E}_L^-(\overline{\gamma}_{k,L-1}^+)$
  $\overline{\gamma}_{k,L-1}^- = \overline{\eta}_{k,L-1}^+ - \overline{\gamma}_{k,L-1}^+, \quad \overline{\alpha}_{k,L-1}^- = \overline{\gamma}_{k,L-1}^- / \overline{\eta}_{k,L-1}^+$
  for $\ell = L-1, \ldots, 0$ **do**
    $\overline{\eta}_{k,\ell-1}^- = 1/\mathcal{E}_\ell^-(\overline{\gamma}_{k,\ell-1}^+, \overline{\gamma}_{k\ell}^-, \tau_{\ell-1}^0)$
    $\overline{\gamma}_{k,\ell-1}^- = \overline{\eta}_{k,\ell-1}^- - \overline{\gamma}_{k,\ell-1}^+, \quad \overline{\alpha}_{k,\ell-1}^- = \overline{\gamma}_{k,\ell-1}^+ / \overline{\eta}_{k,\ell-1}^-$
  end for
end for

B. State Evolution Equations

Define the quantities

$$\begin{aligned} &\mathbf{q}_\ell^0 := \mathbf{z}_\ell^0, \quad \mathbf{p}_\ell^0 := \mathbf{V}_\ell \mathbf{q}_\ell^0 = \mathbf{V}_\ell \mathbf{z}_\ell^0 \quad \ell = 0, 2, \ldots, L \\ &\mathbf{q}_\ell^0 := \mathbf{V}_\ell^{\mathsf{T}} \mathbf{z}_\ell^0, \quad \mathbf{p}_\ell^0 := \mathbf{z}_\ell^0 = \mathbf{V}_\ell \mathbf{q}_\ell^0, \quad \ell = 1, 3, \ldots, L-1, \end{aligned} \tag{12}$$

which represent the true vectors $\mathbf{z}_\ell^0$ and their transforms. Similarly, define the ML-VAMP estimates

$$\widehat{\mathbf{q}}_{k\ell}^\pm := \widehat{\mathbf{z}}_{k\ell}^\pm, \quad \widehat{\mathbf{p}}_{k\ell}^\pm := \mathbf{V}_\ell \widehat{\mathbf{z}}_{k\ell}^\pm \quad \ell = 0, 2, \ldots, L \tag{13a}$$

$$\widehat{\mathbf{q}}_{k\ell}^\pm := \mathbf{V}_\ell^{\mathsf{T}} \widehat{\mathbf{z}}_{k\ell}^\pm, \quad \widehat{\mathbf{p}}_{k\ell}^\pm := \widehat{\mathbf{z}}_{k\ell}^\pm \quad \ell = 1, 3, \ldots, L-1. \tag{13b}$$

Our goal is to describe the mean squared error of these estimates in the LSL. To this end, similar to those in VAMP [13], we introduce the concept of *error functions*. Let $\ell = 2, 4, \ldots, L-2$ be the index of a nonlinear stage and suppose that we are given parameters $\gamma_{\ell-1}^+$, γ_ℓ^-, and $\tau_{\ell-1}^0$. Define a set of random variables $(R_{\ell-1}^+, Z_{\ell-1}^0, Z_\ell^0, R_\ell^-)$ by the Markov chain

$$R_{\ell-1}^+ \sim \mathcal{N}(0, \tau_{\ell-1}^0 - 1/\gamma_{\ell-1}^+), \quad Z_{\ell-1}^0 \sim \mathcal{N}\left(R_{\ell-1}^+, \frac{1}{\gamma_{\ell-1}^-} \right),$$
$$Z_\ell^0 = \phi_\ell(Z_{\ell-1}^0, \Xi_\ell), \quad R_\ell^- \sim Z_\ell^0 + \mathcal{N}(0, 1/\gamma_\ell^-).$$

Define the error functions

$$\begin{aligned}\mathcal{E}_\ell^+(\gamma_{\ell-1}^+,\gamma_\ell^-,\tau_{\ell-1}^0) &:= \mathrm{var}(Z_\ell^0|R_{\ell-1}^+,R_\ell^-),\\ \mathcal{E}_\ell^-(\gamma_{\ell-1}^+,\gamma_\ell^-,\tau_{\ell-1}^0) &:= \mathrm{var}(Z_{\ell-1}^0|R_{\ell-1}^+,R_\ell^-),\end{aligned} \tag{14}$$

which represent the error variances in estimating the inputs and outputs. For $\ell = 0$, we can define $\mathcal{E}_0^+(\gamma_0^-)$ by dropping the terms associated with $R_{\ell-1}^+$ and $Z_{\ell-1}^0$. For $\ell = L$, we define $\mathcal{E}_{L-1}^-(\gamma_{L-1}^+,\tau_{L-1}^0)$ by dropping the terms associated with R_ℓ^-. Next, let $\ell = 1,3,\ldots,L-1$ be the index of a linear stage, and consider a Markov chain,

$$\begin{aligned}&\bar{R}_{\ell-1}^+ \sim \mathcal{N}(0,\tau_{\ell-1}^0 - 1/\gamma_{\ell-1}^+), \quad P_{\ell-1}^0 \sim \mathcal{N}(\bar{R}_{\ell-1}^+, 1/\gamma_{\ell-1}^-),\\ &Q_\ell^0 = \bar{S}P_{\ell-1}^0 + \bar{B} + \bar{\Xi}_\ell, \quad \bar{R}_\ell^- \sim Q_\ell^0 + \mathcal{N}(0,1/\gamma_\ell^-),\end{aligned} \tag{15}$$

which represents the inputs and outputs of a scalar linear channel with parameters $\bar{S}$, $\bar{B}$ and $\bar{\Xi}_\ell$ given from variables (10). Define

$$\begin{aligned}\mathcal{E}_\ell^+(\gamma_{\ell-1}^+,\gamma_\ell^-,\tau_{\ell-1}^0) &:= \mathrm{var}(Q_\ell^0|\bar{R}_{\ell-1}^+,\bar{R}_\ell^-,\bar{S}_\ell,\bar{B}_\ell),\\ \mathcal{E}_\ell^-(\gamma_{\ell-1}^+,\gamma_\ell^-,\tau_{\ell-1}^0) &:= \mathrm{var}(P_{\ell-1}^0|\bar{R}_{\ell-1}^+,\bar{R}_\ell^-,\bar{S}_\ell,\bar{B}_\ell),\end{aligned} \tag{16}$$

Under these definitions, the SE equations for ML-VAMP are given in Algorithm 2, which defines a sequence of random variables and constants.

Theorem 1. *Consider the outputs of the ML-VAMP algorithm, Algorithm 1, and the corresponding outputs of the SE equations in Algorithm 2. In addition to the assumptions in Section III-A, assume:*

(i) The constants $\overline{\alpha}_{k\ell}^\pm \in (0,1)$ for all k and ℓ.
(ii) The activation functions $\phi_\ell(z_{\ell-1},\xi_\ell)$ in (2) are pseudo-Lipschitz continuous of order two.
(iii) The components of the estimation functions $\mathbf{g}_\ell^\pm(\cdot)$ are uniformly Lipschitz continuous (see [20] for more details).

Then, for any fixed iteration k and index ℓ,

$$\lim_{N\to\infty}(\gamma_{k\ell}^\pm,\alpha_{k\ell}^\pm,\eta_{k\ell}^\pm) = (\overline{\gamma}_{k\ell}^\pm,\overline{\alpha}_{k\ell}^\pm,\overline{\eta}_{k\ell}^\pm) \tag{17}$$

almost surely, where the quantities on the right hand side are from the SE equations, Algorithm 2. In addition, the components of the transformed true vectors $\mathbf{p}_\ell^0$ and $\mathbf{q}_\ell^0$ and their estimates $\widehat{\mathbf{p}}_{k\ell}^\pm$ and $\widehat{\mathbf{q}}_{k\ell}^\pm$ converge empirically as

$$\lim_{N\to\infty}\left\{(p_{\ell,n}^0,q_{\ell,n}^0,\widehat{p}_{k\ell,n}^\pm,\widehat{q}_{k\ell,n}^\pm)\right\} \overset{PL(2)}{=} (P_\ell^0,Q_\ell^0,\widehat{P}_{k\ell}^\pm,\widehat{Q}_{k\ell}^\pm), \tag{18}$$

where the random-variable limits have moments

$$\mathbb{E}(\widehat{P}_{k\ell}^\pm - P_\ell^0)^2 = \mathbb{E}(\widehat{Q}_{k\ell}^\pm - Q_\ell^0)^2 = \frac{1}{\overline{\eta}_{k\ell}^\pm}. \tag{19}$$

Theorem 1 shows that the components of the true signals $\mathbf{p}_\ell^0$ and $\mathbf{q}_\ell^0$ and the corresponding ML-VAMP estimates $\widehat{\mathbf{p}}_{k\ell}^\pm$ and $\widehat{\mathbf{q}}_{k\ell}^\pm$ converge empirically to random variables $(P_\ell^0,Q_\ell^0,\widehat{P}_{k\ell}^\pm,\widehat{Q}_{k\ell}^\pm)$. The full paper [20] provides a complete description of the joint distribution of these variables and thus gives an exact characterization of the asymptotic behavior of the true signal and their estimates. In particular, the asymptotic MSE of the ML-VAMP can be exactly computed from (19).

Importantly, this asymptotic MSE can be information theoretically optimal. Specifically, following a pre-print of this paper [20], Reeves [19] has postulated the optimal MSE for inference in deep networks using information theoretic methods. It is shown there that the fixed points of the SE of this work satisfy the same fixed point equations for the postulated optimal MSE. Hence, when the fixed points are unique, ML-VAMP achieves the postulated information-theoretically optimal MSE.

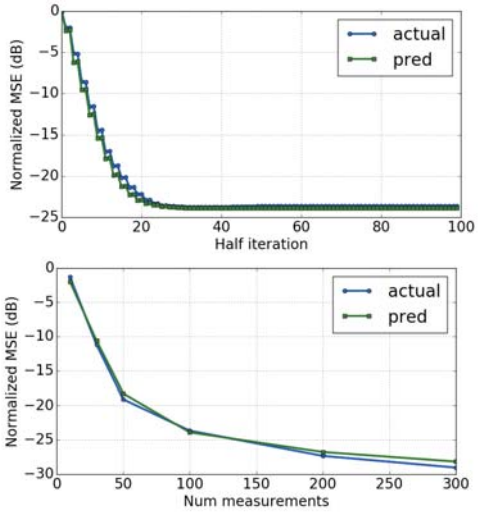

Fig. 1. Simulation with a randomly generated neural network. Top: Normalized mean squared error (NMSE) for ML-VAMP and the predicted MSE as a function of the iteration with $M = 100$ measurements. Bottom: Final NMSE for ML-VAMP and the predicted MSE as a function of the number of measurements, M.

IV. Numerical Experiments

Synthetic random network: To illustrate the SE analysis, we first consider a randomly generated neural network that follows the theoretical model of the paper. (Details are in [20].) Briefly, the network accepts $N_0 = 20$ dimensional unit-variance Gaussian noise $\mathbf{z}_0$, and has three hidden layers, of dimension 100, 500 and 784, respectively. (Similar dimensions will be used for the MNIST experiment below). The observed output is a compressed linear measurement $\mathbf{y} = \mathbf{A}\mathbf{z}_5 + \mathbf{w}$, where $\mathbf{z}_5$ is the vector from the final hidden layer, the matrix $\mathbf{A}$ is $M \times 784$, and $\mathbf{w}$ is Gaussian noise, scaled to achieve a sigal-to-noise ratio of 30 dB. The number of measurements M is varied from 100 to 600. To follow the theory, the weight matrices are drawn from the i.i.d. Gaussian ensemble and the observation matrix $\mathbf{A}$ is drawn from the orthogonally invariant matrix ensemble with singular values spaced logarithmically to give condition number $\kappa = 10$. This model cannot be treated by the ML-AMP algorithm in [7].

The left panel of Fig. 1 shows the normalized mean squared error (NMSE) for the estimation of the inputs to the networks $\mathbf{z}_0$ as a function of the iteration number for a fixed number of measurements $M = 300$. Also plotted is the state evolution (SE) prediction. Plotted values are the average of 1000 random realizations. We see that the SE predicts the ML-VAMP behavior remarkably well, within approximately 1 dB. The right

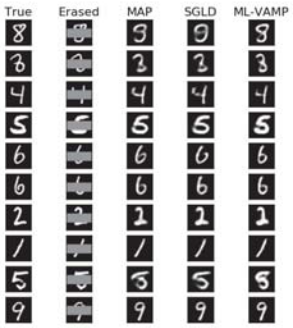

Fig. 2. Inpainting of handwritten digits using MAP estimation, stochastic gradient Langevin dynamics (SGLD) and ML-VAMP.

panel shows the NMSE after 50 iterations (100 half-iterations) for various values of M. We again see an excellent agreement between the actual values and the SE predictions.

MNIST inpainting: To demonstrate the feasibility of ML-VAMP on a real dataset, we performed inpainting on the MNIST dataset, as in [1], [2], [24]. The MNIST dataset consists of $28 \times 28 = 784$ pixel images of hand-written digits as shown in the first column of Fig. 2. Following [4], a generative model for these digits was trained using a variational autoencoder (VAE), so that each image $\mathbf{x}$ is modeled as the output of an L-stage neural network. In this experiment, we used a network with 20 input units, 400 hidden units, and 784 output units, corresponding to the dimension of the images. (Details about the network and its training are given in [20].) For each image $\mathbf{x}$, we then created an occluded image, $\mathbf{y}$, by removing the rows 10–20 of the original image, as shown in the second column of Fig. 2. Combining the generative layers with the occlusion layer creates a deep network model for the occluded image $\mathbf{y}$. ML-VAMP was then used to recover the original image $\mathbf{x}$ from the occluded image $\mathbf{y}$.

Fig. 2 shows a typical reconstructions from i) ML-VAMP, ii) MAP estimation via numerical optimization of the posterior density as performed in [1], [2], and iii) estimation of the posterior mean $\mathbb{E}(\mathbf{x}|\mathbf{y})$ via Stochastic Gradient Langevin Dynamics (SGLD) [25]. (See [20] for details.) We see that, visually, the ML-VAMP, MAP, and SGLD estimates are similar. However, the ML-VAMP algorithm was significantly faster than its competitors: ML-VAMP used only 20 iterations, while MAP used 500 iterations, and SGLD used 10000. Thus, this experiment suggests that, in addition to providing theoretical guarantees, ML-VAMP may be a computationally efficient approach to reconstruction. Of course, further experimentation on a variety of data sets is still needed to evaluate its practical applicability.

V. Conclusions

We have presented a principled and computationally tractable method for inference in deep networks whose performance can be rigorously characterized in certain high-dimensional random settings. Importantly, the asymptotic MSE of ML-VAMP satisfies a fixed point equation that is identical to that of the optimal MSE postulated by Reeves [19].

References

[1] 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.
[2] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," *arXiv:1703.03208*, 2017.
[3] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. ICML*, 2014, pp. 1278–1286.
[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv:1312.6114*, 2013.
[5] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, "Adversarially learned inference," *arXiv:1606.00704*, 2016.
[6] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.
[7] A. Manoel, F. Krzakala, M. Mézard, and L. Zdeborová, "Multi-layer generalized linear estimation," *arXiv:1701.06981*, 2017.
[8] S. Rangan, P. Schniter, and A. K. Fletcher, "On the convergence of approximate message passing with arbitrary matrices," in *Proc. IEEE ISIT*, Jul. 2014, pp. 236–240.
[9] F. Caltagirone, L. Zdeborová, and F. Krzakala, "On convergence of approximate message passing," in *Proc. IEEE ISIT*, Jul. 2014, pp. 1812–1816.
[10] J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborová, "Adaptive damping and mean removal for the generalized approximate message passing algorithm," in *Proc. IEEE ICASSP*, 2015, pp. 2021–2025.
[11] A. Manoel, F. Krzakala, E. W. Tramel, and L. Zdeborová, "Swept approximate message passing for sparse estimation," in *Proc. ICML*, 2015, pp. 1123–1132.
[12] S. Rangan, A. K. Fletcher, P. Schniter, and U. S. Kamilov, "Inference for generalized linear models via alternating directions and Bethe free energy minimization," *IEEE Trans. Inform. Theory*, vol. 63, no. 1, pp. 676–697, 2017.
[13] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," *arXiv:1610.03082*, 2016.
[14] H. He, C.-K. Wen, and S. Jin, "Generalized expectation consistent signal recovery for nonlinear measurements," *arXiv:1701.04301*, 2017.
[15] P. Schniter, S. Rangan, and A. K. Fletcher, "Vector approximate message passing for the generalized linear model," in *Asilomar Conf. Sig., Sys., Comput.*, 2016, pp. 1525–1529.
[16] M. Opper and O. Winther, "Expectation consistent free energies for approximate inference," in *Proc. NIPS*, 2004, pp. 1001–1008.
[17] ——, "Expectation consistent approximate inference," *J. Mach. Learning Res.*, vol. 1, pp. 2177–2204, 2005.
[18] A. K. Fletcher, M. Sahraee-Ardakan, S. Rangan, and P. Schniter, "Expectation consistent approximate inference: Generalizations and convergence," in *Proc. IEEE ISIT*, 2016, pp. 190–194.
[19] G. Reeves, "Additivity of information in multilayer networks via additive gaussian noise transforms," *arXiv preprint arXiv:1710.04580*, 2017.
[20] A. K. Fletcher, S. Rangan, and P. Schniter, "Inference in deep networks in high dimensions," *arXiv preprint arXiv:1706.06549*, 2017.
[21] Y. Kabashima, "Inference from correlated patterns: a unified theory for perceptron learning and linear vector channels," in *Journal of Physics: Conference Series*, vol. 95, no. 1, 2008.
[22] T. Shinzato and Y. Kabashima, "Perceptron capacity revisited: classification ability for correlated patterns," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, no. 32, p. 324013, 2008.
[23] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.
[24] E. W. Tramel, A. Manoel, F. Caltagirone, M. Gabrié, and F. Krzakala, "Inferring sparsity: Compressed sensing using generalized restricted boltzmann machines," in *Proc. ITW*, 2016, pp. 265–269.
[25] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proc. ICML*, 2011, pp. 681–688.