Deep Message Passing

Around the middle of the XVII century, a young man called Baruch Spinoza was dealing with a problem tougher than yours: finding the essence of God Himself.

CONTENTS

I. INTRODUCTION

II. LEARNING THROUGH MESSAGE PASSING

TODO ENTIRE SECTION.

In this deep inference problem, we assume that a signal with prior P^{in} is fed to a deep feedforward networks with $L+1$ layers of weights $\boldsymbol{W}^\ell \in \mathbb{R}^{N_{\ell+1}\times N_\ell}, \ell = 0,\dots,L$ and biases $\boldsymbol{b}^\ell \in \mathbb{R}^{N_{\ell+1}}$. The signal is propagated through stochastic neuron layers described by probability distributions P^ℓ conditioned on the preactivations, therefore we have the following Markov chain:

$$\boldsymbol{x}^0 \sim P^{\text{in}} \tag{1}$$

$$\boldsymbol{x}^{\ell+1} \sim P^{\ell+1}\left(\bullet \,|\, \boldsymbol{W}^\ell \boldsymbol{x}^\ell + \boldsymbol{b}^\ell\right) \qquad \ell = 0,\dots,L \tag{2}$$

Only $\boldsymbol{y} = \boldsymbol{x}^{L+1}$ is observed, and the task is to reconstruct the original signal $\boldsymbol{x}^0$. The posterior distribution $p(\boldsymbol{x}^{0:L}) = P(\boldsymbol{x}^{0:L} \,|\, \boldsymbol{x}^{L+1} = \boldsymbol{y})$ reads

$$p(\boldsymbol{x}^{0:L}) \propto \prod_{\ell=0}^{L}\prod_{k=1}^{N_{\ell+1}} P_k^{\ell+1}\left(x_k^{\ell+1} \,\middle|\, \sum_{i=1}^{N_\ell} W_{ki}^\ell x_i^\ell\right) \prod_{k=1}^{N_0} P_k^{\text{in}}(x_k^0), \tag{3}$$

Typical channels are given by deterministic elementwise activation function $f_\ell(z)$ (e.g. $f_\ell(z) = \text{sign}(x)$ or $f_\ell(z) = \text{relu}(z) = \max(0,z)$), combined with Gaussian additive pre-activation noise with variance σ^2. In such cases we have

$$P_k^\ell(x \,|\, z) = \int D\xi\, \delta\left(x - f_\ell(z + \sigma_k^\ell\, \xi)\right)$$

We also call $\alpha_\ell = N_{\ell+1}/N_\ell$ the layer *expansion ratio.*

III. BELIEF PROPAGATION

A. BP updates

The AMP equations have been derived from the BP equation in the Appendix. First we introduce the neuron scalar entropy functions:

$$\varphi_k^0(B,A) = \log \int \mathrm{d}x\; e^{-\frac{1}{2}A^2x^2 + Bx}\, P_k^{\text{in}}(x) \tag{4}$$

$$\varphi_k^\ell(B,A,\omega,V) = \log \int \mathrm{d}x\,\mathrm{d}z\; e^{-\frac{1}{2}A^2x^2+Bx}\, P_k^\ell\,(x|z)\, e^{-\frac{(\omega-z)^2}{2V}} \qquad \ell = 1,\dots,L \tag{5}$$

$$\varphi_k^{L+1}(\omega,V,y) = \log \int \mathrm{d}z\; P_k^{L+1}\,(y|z)\, e^{-\frac{(\omega-z)^2}{2V}} \tag{6}$$

$$\psi_{ki}^\ell(H,G) = \log \int \mathrm{d}w\; e^{-\frac{1}{2}G^2w^2+Hw}\, P_{ki}^\ell(w) \tag{7}$$

For convenience define $\varphi_i^{0,t} = \varphi_i^0\left(B_i^{0,t}, A_i^{0,t}\right)$ and $\varphi_i^{\ell,t} = \varphi_i^\ell\left(B_i^{\ell,t}, A_i^{\ell,t}, \omega_i^{\ell-1,t}, V_i^{\ell-1,t}\right)$ and $\varphi_i^{L+1,t} = \varphi_i^{L+1}\left(\omega_i^{L,t}, V_i^{L,t}, y_i\right)$. Then, we can decompose the BP update rules in a forward and a backward step.

Forward pass. As the initial condition for the iterations, we set to zero the following quantities: $B_i^{\ell,t=0} = 0$, $A_i^{\ell,t=0} = 0$ and $g_k^{\ell,t=0} = 0$. The following iterations hold at time $t \geq 1$. In the FORWARD pass, starting from $\ell = 0$ and up to $\ell = L$, we have

$$\hat{x}^{\ell,t}_{ia\to k} = \partial_B \varphi^{\ell}_{ia\to k}\left(B^{\ell,t-1}_{ia\to k}, A^{\ell,t-1}_{ia}, \omega^{\ell-1,t}_{ia}, V^{\ell-1,t}_{ia}\right) \tag{8}$$

$$\Delta^{\ell+1,t}_{ia\to k} = \partial^2_B \varphi^{\ell+1,t}_{ia\to k} \tag{9}$$

$$m^{\ell,t}_{ki\to a} = \partial_H \psi^{\ell}_{ki}(H^{t-1}_{ki\to a}, G^{t-1}_{ki}) \tag{10}$$

$$\sigma^{\ell,t}_{ki\to a} = \partial^2_H \psi^{\ell}_{ki}(H^{t-1}_{ki\to a}, G^{t-1}_{ki}) \tag{11}$$

$$V^{\ell,t}_{ka} = \sum_i \left(\left(m^{\ell,t}_{ki\to a}\right)^2 \Delta^{\ell,t}_{ia\to k} + \Sigma^{\ell,t}_{ki\to a}(\hat{x}^{\ell,t}_{ia\to k})^2 + \sigma^{\ell,t}_{ki\to a}\Delta^{\ell,t}_{ia\to k}\right) \tag{12}$$

$$\omega^{\ell,t}_{ka\to i} = \sum_{i'\neq i} m^{\ell,t}_{ki\to a}\hat{x}^{\ell,t}_{ia\to k} \tag{13}$$

Here V^ℓ and ω^ℓ are computed as a function of the previous layer values $V^{\ell-1}$ and $\omega^{\ell-1}$.

Backward pass. In the BACKWARD sweep, starting from $\ell = L$ and down to $\ell = 0$, we have

$$g^{\ell,t}_{ka\to i} = \partial_\omega \varphi^{\ell+1}_{ka\to i}\left(B^{\ell+1,t}_{ka}, A^{\ell+1,t}_{ka}, \omega^{\ell,t}_{ka\to i}, V^{\ell,t}_{ka}\right) \tag{14}$$

$$\Gamma^{\ell,t}_{ka\to i} = -\partial^2_\omega \varphi^{\ell+1,t}_{ka\to i} \tag{15}$$

$$A^{\ell,t}_{ia} = \sum_k \left((m^{\ell,t}_{ki\to a})^2 + \sigma^{\ell,t}_{ki\to a}\right)\Gamma^{\ell,t}_{ka\to i} - \sigma^{\ell,t}_{ki\to a}\left(g^{\ell,t}_{ka\to i}\right)^2 \tag{16}$$

$$B^{\ell,t}_{ia\to k} = \sum_{k'\neq k} m^{\ell}_{k'i\to a} g^{\ell,t}_{k'a\to i} \tag{17}$$

$$G^{\ell,t}_{ki} = \sum_a \left((\hat{x}^{\ell,t}_{ia\to k})^2 + \Delta_{ia\to k}\right)\Gamma^{\ell,t}_{ka\to i} - \Delta_{ia\to k}\left(g^{\ell,t}_{ka\to i}\right)^2 \tag{18}$$

$$H_{ki\to a} = \sum_{a'\neq a} \hat{x}^{\ell,t}_{ia'\to k} g^{\ell,t}_{ka'\to i} \tag{19}$$

Notice that A^ℓ and B^ℓ are computed as as a function of the $A^{\ell+1}, B^{\ell+1}$ of the layer above, with the initial condition given by the output $\boldsymbol{x}^{L+1} = \boldsymbol{y}$ on the top layer.

B. Approximate Message Passing

Forward pass. As the initial condition for the iterations, we set to zero the following quantities: $B^{\ell,t=0}_i = 0, A^{\ell,t=0}_i = 0$ and $g^{\ell,t=0}_k = 0$. The following iterations hold at time $t \geq 1$. In the FORWARD pass, starting from $\ell = 0$ and up to $\ell = L$, we have

$$\hat{x}^{\ell,t}_{ia} = \partial_B \varphi^{\ell,t^-}_{ia} \tag{20}$$

$$\Delta^{\ell,t}_{ia} = \partial^2_B \varphi^{\ell,t^-}_{ia} \tag{21}$$

$$m^{\ell,t}_{ki} = \partial_H \psi^{\ell,t^-}_{ki} \tag{22}$$

$$\sigma^{\ell,t}_{ki} = \partial^2_H \psi^{\ell,t^-}_{ki} \tag{23}$$

$$V^{\ell,t}_{ka} = \sum_i \left(\left(m^{\ell,t}_{ki}\right)^2 \Delta^{\ell,t}_{ia} + \sigma^{\ell,t}_{ki}(\hat{x}^{\ell,t}_{ia})^2 + \sigma^{\ell,t}_{ki}\Delta^{\ell,t}_{ia}\right) \tag{24}$$

$$\omega^{\ell,t}_{ka} = \sum_i m^{\ell,t}_{ki}\hat{x}^{\ell,t}_{ia} + TODO: onsagsize(x)er \tag{25}$$

Here V^ℓ and ω^ℓ are computed as a function of the previous layer values $V^{\ell-1}$ and $\omega^{\ell-1}$.

Backward pass. In the BACKWARD sweep, starting from $\ell = L$ and up to $\ell = 0$, we have

$$g^{\ell,t}_{ka} = \partial_\omega \varphi^{\ell+1,t}_{ka} \tag{26}$$
$$\Gamma^{\ell,t}_{ka} = -\partial^2_\omega \varphi^{\ell+1,t}_{ka} \tag{27}$$
$$A^{\ell,t}_{ia} = \sum_k \left((m^{\ell,t}_{ki})^2 + \sigma^{\ell,t}_{ki}\right)\Gamma^{\ell,t}_{ka} - \sigma^{\ell,t}_{ki}\left(g^{\ell,t}_{ka}\right)^2 \tag{28}$$
$$B^{\ell,t}_{ia} = \sum_k m^{\ell}_{ki} g^{\ell,t}_{ka} + TODO : onsager \tag{29}$$
$$G^{\ell,t}_{ki} = \sum_a \left((\hat{x}^{\ell,t}_{ia})^2 + \Delta_{ia}\right)\Gamma^{\ell,t}_{ka} - \Delta_{ia}\left(g^{\ell,t}_{ka}\right)^2 \tag{30}$$
$$H_{ki} = \sum_a \hat{x}^{\ell,t}_{ia} g^{\ell,t}_{ka} + TODO : onsager \tag{31}$$

Notice that A^ℓ and B^ℓ are computed as as a function of the $A^{\ell+1}$, $B^{\ell+1}$ of the layer above, with the initial condition given by the output $\boldsymbol{x}^{L+1} = \boldsymbol{y}$ on the top layer.

IV. NUMERICAL RESULTS

A. Experiments on MNIST

1. *Varying batch-size*

The command to reproduce the experiments in this section is:

```
run_experiment(9; M=Int(6e4), batchsize=batchsize, usecuda=true, gpu_id=0, ρ=1+1e-5, ψ=0.5, lay=lay, epochs=100)
```

with batchsize={1,16,128,1024} and lay={:bp, :bpi, :tap}.

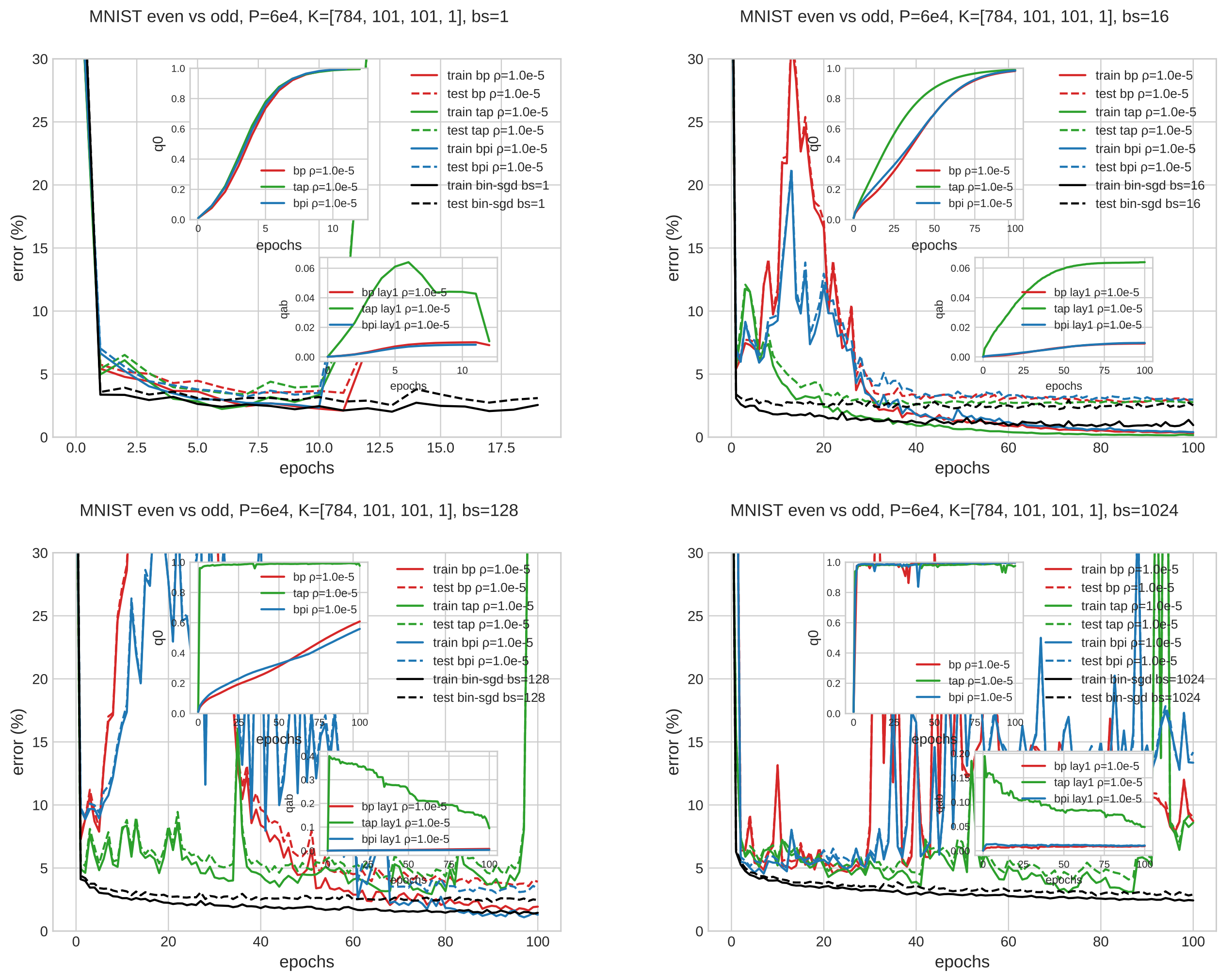

Figure 1. Comparison of BP, TAP, BPI, SGD varying the batchsize (upper left: bs=1; upper right: bs=16, lower left: bs=128; lower right=1024). The parameter $\rho - 1$ is fixed in all experiments to 10^{-5}.

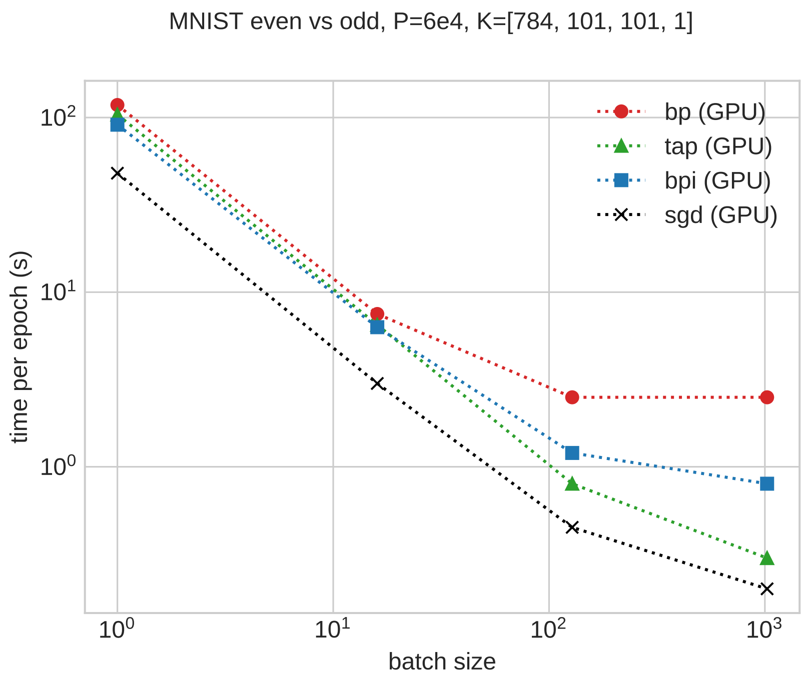

Figure 2. Algorithms time scaling with the batchsize. The reported time refers to one epoch for each algorithm.

2. *Varying ρ*

3. *Varying maxiters*

4. *Varying r*

5. *Varying damping*

6. *Varying dataset size*

7. *Varying initial weights*

B. Experiments on RFM

Here we present results with the Random Features Model, concerning in particular the permutation symmetry. In order to investigate the role of the permutation symmetry we present results on the fully connected committee machine (1 hidden layer network learning only the first weight layer, with the weights of the second layer fixed to all ones).

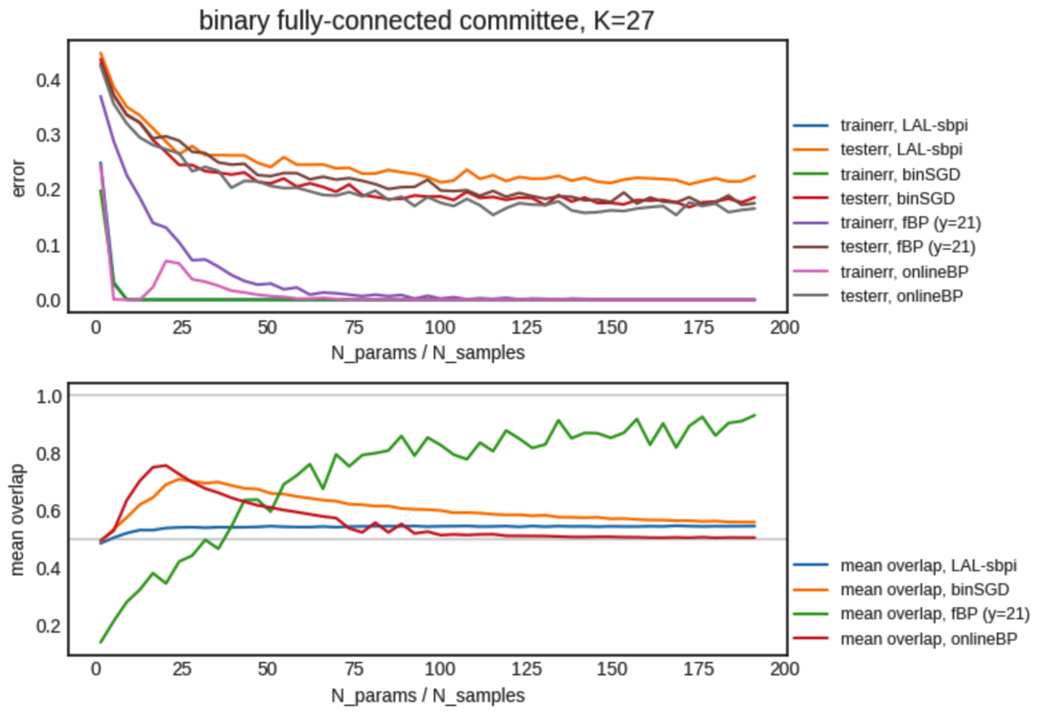

Figure 3. Overlap varying N in the RFM, fully connected committee machine with various algorithms.