ELE 538B: ADMM in Neural Networks

Zachary Hervieux-Moore

02/05/18

Overview

Review

Neural Networks:

$$\min_{\theta} \mathcal{L}(f_\theta(x), y)$$

where $\mathcal{L}(\cdot)$ is some loss function and $f_\theta(\cdot)$ is a family of functions parameterized by your neural network, x are the inputs, and y are the outputs.

ADMM: Problems of the form

$$\begin{aligned} &\min f(x) + g(z) \\ &\text{s.t. } Ax + Bz = c \end{aligned}$$

Can be solved using the following iterations

$$x^{k+1} = \arg\min_{x} L_\rho(x, z^k, \lambda^k)$$

$$z^{k+1} = \arg\min_{z} L_\rho(x^{k+1})$$

$$\lambda^{k+1} = \lambda^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$$

Generative Adversarial Networks

Developed in 2014 by Goodfellow et al. Essentially, it is the combination of two different networks with opposing loss functions. The networks simultaneously minimize their loss which turns into the following minimax problem:

$$\min_G \max_D = \mathbb{E}_{x \sim data}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]$$

Where $D(x; \theta_d)$ is the discriminator network. It tries to minimize the error in the labels (real or generated). $G(z; \theta_g)$ is the generative network which takes in a noise vector z and outputs something from your data space.

Case Study 1: Toy Example

To motivate the potential usefulness of ADMM in neural networks, consider the following problem:

$$\min_{\theta} \|f_\theta(x) - y\|_2^2 + \frac{1}{2}\|\theta\|_2^2$$

Turning this into ADMM

$$\min_{\theta_1,\theta_2} \|f_{\theta_1}(x) - y\|_2^2 + \frac{1}{2}\|\theta_2\|_2^2$$

Which yields

$$\theta_1^{k+1} = \arg\min_{\theta_1} \|f_{\theta_1}(x) - y\|_2^2 + \langle \lambda^k, \theta_2^k - \theta_1 \rangle + \frac{\rho}{2}\|\theta_2^k - \theta_1\|_2^2$$

$$\theta_2^{k+1} = \frac{\theta_1^{k+1} - \lambda^k}{1+\rho}$$

$$\lambda^{k+1} = \lambda^k + \rho(\theta_2^{k+1} - \theta_1^{k+1})$$

Case Study 1: Results

While the ADMM has more moving parts, it converges similarly to just doing SGD on the original problem

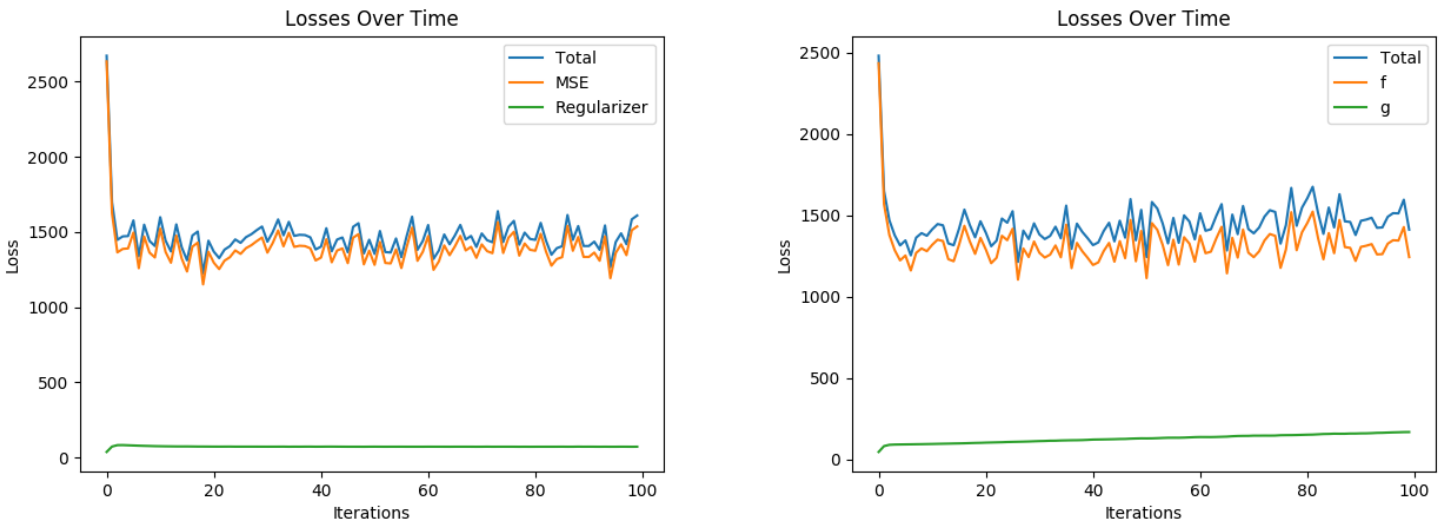

Figure: Objective Value vs. Iterations for normal SGD on the left and ADMM on the right. Notice the slight slope for the loss associated with g on the right.

Case Study 2: GANs

Using the minimax formulation of GANs from before, we get the following problem for the discriminator

$$\min_{\theta_d} \log D(x;\theta_d) + \log(1 - D(G(z);\theta_d))$$

Turning this into ADMM

$$\min_{\theta_{d_1},\theta_{d_2}} \log D(x;\theta_{d_1}) + \log(1 - D(G(z);\theta_{d_2}))$$

Which yields

$$\theta_{d_1}^{k+1} = \arg\min_{\theta_{d_1}} \log D(x;\theta_{d_1}) + \langle \lambda^k, \theta_{d_2}^k - \theta_{d_1} \rangle + \frac{\rho}{2}\|\theta_{d_2}^k - \theta_{d_1}\|_2^2$$

$$\theta_{d_2}^{k+1} = \arg\min_{\theta_{d_2}} \log(1 - D(G(z);\theta_{d_2})) + \langle \lambda^k, \theta_{d_2} - \theta_{d_1}^{k+1} \rangle + \frac{\rho}{2}\|\theta_{d_2} - \theta_{d_1}^{k+1}\|_2^2$$

$$\lambda^{k+1} = \lambda^k + \rho(\theta_{d_2}^{k+1} - \theta_{d_1}^{k+1})$$

Case Study 2: Results

While the ADMM implementation was actually faster per itertaion, below is the generative data from both models:

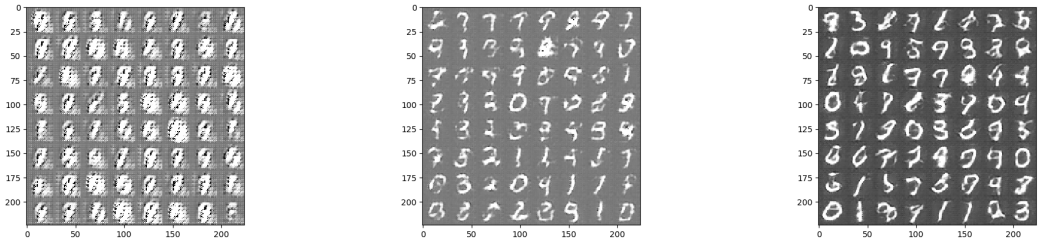

Figure: Generative examples from the non ADMM model at the 50, 150, and 250 epochs.

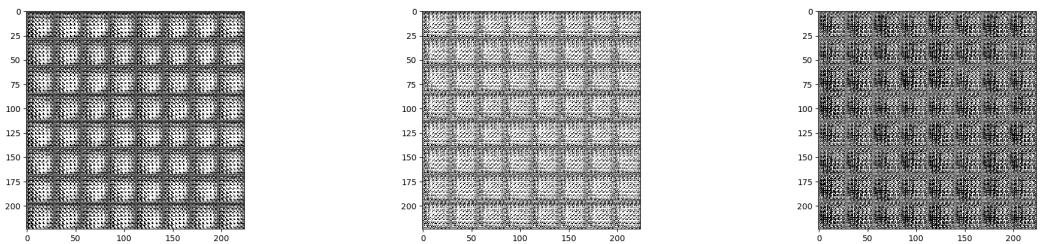

Figure: Generative examples from the ADMM model at the 50, 150, and 250 epochs.

Problems and Possible Extensions

The ADMM model was extremely sensitive to values of ρ and implementation details.

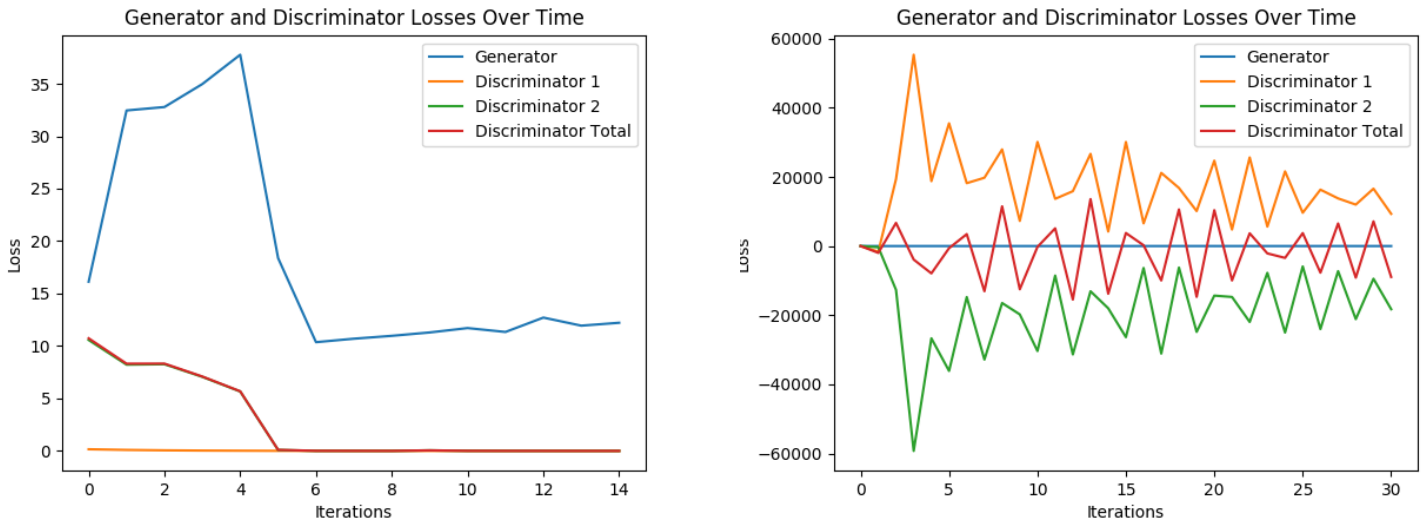

Figure: Two experiments with the same ρ value but with different implementations of the loss functions. The one on the right has floating point errors.

Possible Extensions

Possible extensions:

- Implementing some form of restart on the ADMM or only performing a few iterations of ADMM between normal SGD
- Modifying the objective function so that the two discriminators still have their objectives tied
- Enhancing the objective with many generators and discriminators and doing ADMM across those instead