# ECE367 Implicit Regularization

Qiyao Wei

November 2020

## 1 Background

The word "regularization" should not sound strange to us. We have already studied $l_p$ norm regularizations in least squares, which is a form of explicit regularization. In this report I will be addressing implicit regularization, and featuring some fairly recent works in the machine learning literature.

## 2 Brief Into to Implicit Regularization

See A Primer on Implicit Regularization for more info.

The derivation below can be applied to general problems, but for the sake of simplicity let's assume that we are learning a 2-dimensional weight vector , i.e. $w \in R^2$. Say we run gradient descent updates on a square loss

$$\ell(w) = \frac{1}{2} \|w - w^*\|^2 \tag{1}$$

This is even simpler than a linear regression setting! We are essentially assuming that we know the ground truth weight values, and using gradient descent to get there. We know that the weight updates will be in the negative direction of the gradient

$$\nabla_w = w^* - w \tag{2}$$

Now let's make a small change to our problem setup. Instead of directly optimizing over w, we parameterize w with

$$\begin{aligned} w_1 &= u_1^3 \\ w_2 &= u_2^3 \end{aligned} \tag{3}$$

and optimize over u instead. I will leave this as homework for you, but you will find that gradient updates for w, i.e. $\nabla_{w(u)}$ is multiplied by another weight term compared to $\nabla_w = w^* - w$

What this change does to the gradient is that weight values closer to 0 will change very slowly, and large weight values change very quickly. This example is basically a toy version of the exploding/vanishing gradient problem in deep neural networks.

# 3 Implicit Rank-Minimizing Autoencoder

Paper is Implicit Rank-Minimizing Autoencoder. Also see Understanding implicit regularization in deep learning by analyzing trajectories of gradient descent for more info.

As we have discussed earlier, implicit regularization plays a key role in deep neural networks. This paper essentially utilizes a more advanced form of implicit regularization, and applies it to autoencoders. Don't worry if you don't know what those are. The only change made in this paper (top row) compared to the original autoencoder structure (bottom row) is that they included a bunch of linear matrix multiplications in the middle. As you might have guessed, this theoretically should not change anything, since multiplication of linear matrices can be aggregated into one overall matrix, but empirically results are different due to implicit regularization.

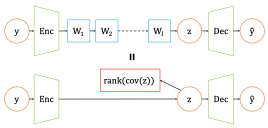


Figure 1: Top row is this paper, bottom row is regular architecture [Jing et al.(2020)Jing, Zbontar, et al.]

The effect of adding those matrices? We can see that we obtain a lower-rank representation. For those familiar with autoencoder jargons, we are effectively learning a more informative latent space representation by decreasing the rank of the covariance matrix of z, the latent space parameters.

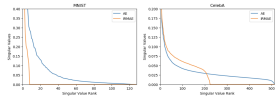

Figure 2: AE is original autoencoder, and IRMAE is this paper [Jing et al.(2020)Jing, Zbontar, et al.]

The underlying reason is actually very similar to the toy example we played with earlier. See Nadav Cohen's blog post about why most singular values go to zero when we do gradient updates over multiplication of linear matrices.

Finally, for those of you familiar with the autoencoder literature, I wanted to share an interpolation experiment in the paper whose results I find pleasantly surprising.

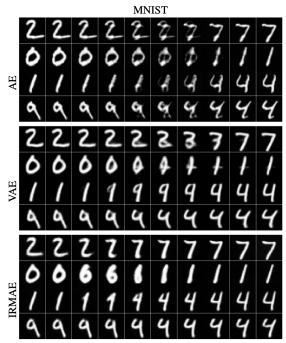


Figure 3: Interpolation results of AE, VAE, and this paper [Jing et al.(2020)Jing, Zbontar, et al.]

# References

[Jing et al.(2020)Jing, Zbontar, et al.] Li Jing, Jure Zbontar, et al. Implicit rank-minimizing autoencoder. *Advances in Neural Information Processing Systems*, 33, 2020.