GENERALIZE NEURAL NETWORK THROUGH SMOOTH HYPOTHESIS FUNCTION

Yupu Yao
University of Electronic Science and Technology of China
Chengdu, China
{yypseek123}@gmail.com

ABSTRACT

The neural network (NN) looks for the hypothesis function in the search space, where the hypothesis function can be conceptualized as a continuous interpolating function. Nevertheless, the issue of overfitting commonly arises in interpolation methods that focus on function values, such as Lagrange interpolation. We construct the geometric intuition to improve extrapolation and induce the Jacobian Regularization (JR) by Hermite interpolation. The concept of Jacobian Regularization resembles the gradient penalty (GP) employed in the Wasserstein GAN, with an experiment verifying the feasibility of our method.

1 INTRODUCTION

The hypothesis function is well-suited for accurately representing the relationship between the input data and the corresponding ground truth. As the hypothesis function predicts each point meticulously, it can be viewed as an interpolating function from the training data. Meanwhile, neural network (NN) attempts to search such function usually by stochastic gradient descent (SGD).

We encounter two predominant challenges in practical training scenarios: underfitting and overfitting Zhang et al. (2021). One may adopt a more sophisticated model to address underfitting or extend the training duration across additional epochs. Overfitting, on the other hand, presents a more nuanced obstacle. While strategies such as early stopping (Caruana et al., 2000; Yao et al., 2007) and data augmentation (Tanner & Wong, 1987; Van Dyk & Meng, 2001; Shorten & Khoshgoftaar, 2019) offer temporary mitigation, they are extrinsic and do not address the inherent complexities of the overfitting phenomenon. Is it feasible to conceptualize the issue of overfitting using a straightforward yet perceptive approach? Although the generalization has been studied through different prespectives (Lugosi & Neu, 2022; Stokes & Baer, 1977; Mitchell, 1982; Guttman & Kalish, 1956), we try to seek to unravel this issue through the lens of the interpolation method.

The NN fits the interpolated part of data and extrapolates other data, showing the ability of NN's generalization. In this paper, we focus on the extrapolation of NN. Hanin (2021) had studied $\updownarrow_2$ norm for parameter could improve NN to extrapolate. We consider the extrapolation by the geometrical intuition inspired by the traditional interpolation approach. Previous studies have investigated the correlation between the interpolation approach and NN (Berrada et al., 2020; Barnard & Wessels, 1992; Montanari & Zhong, 2022). The prominent method for a simple interpolating function is Lagrange interpolation, which constructs most nth degree polynomials through $n + 1$ data points. Nevertheless, the issue of Lagrange interpolation NN, as both methods are susceptible to overfitting. Therefore, it is recommended to utilize Hermite interpolation to avoid such a scenario. This method not only fulfills the requirements of the Lagrange polynomial but also ensures that the derivative of the interpolated polynomial at each point x_i matches the derivative of the original function $f(x_i)$.

In this paper, we rediscover Jacobian Regularization (JR) Sokolić et al. (2017) to generalize induced by Hermite interpolation. The JR-like gradient penalty policy has been discussed in Varga et al. (2017); Ross & Doshi-Velez (2018); Hoffman et al. (2019). However, we start with the property of the hypothesis function to make the smooth interpolating function, which prevents overfitting effectively and induces the JR in an interesting way. Additional analysis is placed in the appendix.

Table 1: Simple FCN performance across penalty coefficients on MINIST and CIFAR10. The Training Loss (↓) and the Test Accuracy (↑) are compared on convergence epochs.

Datasets	λ	Epochs	Training Loss	Accuracy(%)
MINST	0	9	0.0400	87.78
MINST	0.1	9	0.0337	88.11
MINST	0.2	9	**0.0310**	88.15
MINST	0.5	9	0.0365	**88.24**
CIFAR10	0	20	**0.0804**	33.99
CIFAR10	0.1	20	0.0836	34.69
CIFAR10	0.2	20	0.0812	34.12
CIFAR10	0.5	20	0.0819	**35.05**

2 METHODOLOGY

Given the training pair $(\boldsymbol{x}_j, y_j) \in \mathcal{D}, j = 0, 1, ..., n$ and $\boldsymbol{x}_j = (x_0, x_1, ..., x_i, ..., x_m)$. In the following content, we denote x_i directly to imply it as a coordinate component of arbitrary $\boldsymbol{x} \sim p(\boldsymbol{x})$ with corresponding y. Let h be a hypothesis function such that h' is continuous and h'' exists in the hypothesis space $\mathcal{H}$. Then for each x_i, we have $h(x_i) = y_i$. Therefore, the hypothesis is the optimal function we aim for our neural network to explore. However, to require a smooth hypothesis function, which could be considered as the interpolation function of the training pair, the new condition to reduce the search space is $\frac{dh(x_i)}{dx_i} = \frac{dy_i}{dx_i}$. This condition induces a smooth function like the Hermite polynomial, preventing overfitting.

For the NN $f_{\boldsymbol{\theta}}$, which search or approach for such a hypothesis function can be converted into an optimization problem with distance measure $\|\cdot\|$,

$$\min_{\boldsymbol{\theta}} \frac{1}{mn} \sum_{j=0}^{n} \sum_{i=0}^{m} \|f_{\boldsymbol{\theta}}(x_i) - h(x_i)\|^2 + \lambda \| \frac{df_{\boldsymbol{\theta}}(x_i)}{dx_i} - \frac{dh(x_i)}{dx_i} \|^2, \tag{1}$$

where λ is a penalty coefficient. The new problem of our optimization objective in Eq. 1 is, what is exactly for the $\frac{dh(x_i)}{dx_i}$? Reminding of our training pair is discrete data, the punctured open ball $\overset{\circ}{B}(x_i, \delta)$ with $\delta > 0$ satisfy such a property we expect for our hypothesis function:

$$h(x) = h(x_i),\ x \in \overset{\circ}{B}(x_i, \delta). \tag{2}$$

With little perturbation, we hope the output of the hypothesis function remains at its original value. Then $\frac{dh(x_i)}{dx_i} < \epsilon$ with $\forall \epsilon > 0$ for each x_i, and naturally, h satisfies Lipschitz consistency. According to this result and recall $h(\boldsymbol{x}) = y$, the optimization objective in Eq. 1 can be converted into

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})} \|f_{\boldsymbol{\theta}}(\boldsymbol{x}) - y\|^2 + \lambda \|\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2. \tag{3}$$

The interesting point is that the result in Eq. 3 is really similar to Gradient Penalty (GP)

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})} \|f_{\boldsymbol{\theta}}(\boldsymbol{x}) - y\|^2 + \lambda \left(\|\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\boldsymbol{x})\| - 1\right)^2, \tag{4}$$

proposed by Gulrajani et al. (2017). The GP is the relaxed constraint of JR, which only requires the norm of a gradient smaller than one.

3 EXPERIMENT

In this section, we set up the experiment to validate our result. We take the MINST and CIFAR10 as our dataset with superficial Full Connected Net (FCN). More specifically, we set the batch size to 64. With the SGD set of the learning rate of 0.01, we research the result of different λ, while the $\lambda = 0$ implies without the JR. To compare the overfitting situation, we compare the accuracy of classification tasks on the test datasets with similar training losses. The results shown in Tab. 1 imply that with JR, the NN performs better on test sets and even reduces the training loss in some situations, which indicates it can somehow prevent overfitting.

4 Conclusion

Our work starts from the previous interpolating method, especially for Hermite polynomial, deducing Jacobian Regularization to a smooth hypothesis function in search space. We analyze the error and the feasibility of gradient descent with experiment validation. As our approach predicted, the overfitting result is reduced when the proper hyperparameter λ is given. Ultimately, we point out the interesting relationship between JR and GP, where our term is a more strict constraint than GP.

URM Statement

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

References

Randall Balestriero, Jerome Pesenti, and Yann LeCun. Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485*, 2021.

Etienne Barnard and LFA Wessels. Extrapolation and interpolation in neural network classifiers. *IEEE Control Systems Magazine*, 12(5):50–53, 1992.

Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Training neural networks for and by interpolation. In *International Conference on Machine Learning (ICML)*, pp. 799–809. PMLR, 2020.

Rich Caruana, Steve Lawrence, and C Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. *Advances in Neural Information Processing Systems*, 13, 2000.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30, 2017.

Norman Guttman and Harry I Kalish. Discriminability and stimulus generalization. *Journal of Experimental Psychology*, 51(1):79, 1956.

Boris Hanin. Ridgeless interpolation with shallow relu networks in $1d$ is nearest neighbor curvature extrapolation and provably generalizes on lipschitz functions. *arXiv preprint arXiv:2109.12960*, 2021.

Judy Hoffman, Daniel A Roberts, and Sho Yaida. Robust learning with jacobian regularization. 2019.

Gábor Lugosi and Gergely Neu. Generalization bounds via convex analysis. In *Conference on Learning Theory (COLT)*, pp. 3524–3546. PMLR, 2022.

Tom M Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, 1982.

Andrea Montanari and Yiqiao Zhong. The interpolation phase transition in neural networks: Memorization and generalization under lazy training. *The Annals of Statistics*, 50(5):2816–2847, 2022.

Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.

Trevor F Stokes and Donald M Baer. An implicit technology of generalization 1. *Journal of Applied Behavior Analysis*, 10(2):349–367, 1977.

Martin A Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540, 1987.

David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

Dániel Varga, Adrián Csiszárik, and Zsolt Zombori. Gradient regularization improves accuracy of discriminative models. *arXiv preprint arXiv:1712.09936*, 2017.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26:289–315, 2007.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, pp. 928–936, 2003.

A Error Analysis

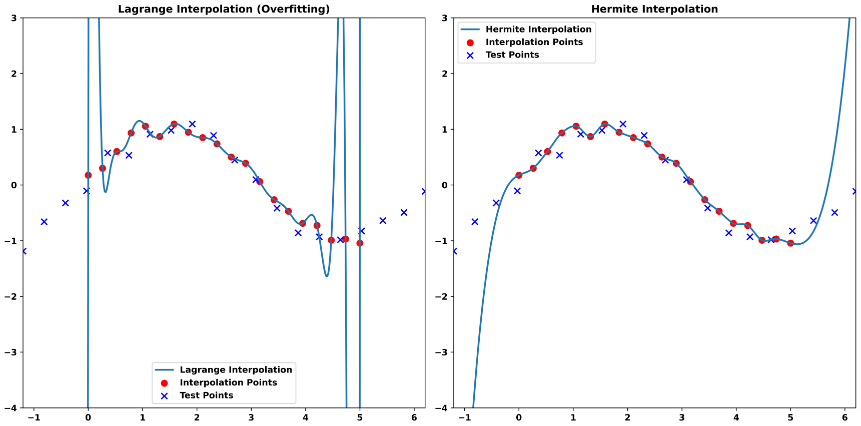

Figure 1: A straightforward yet easily understandable example demonstrating the smooth constraint. While Lagrange interpolation and Hermite interpolation may yield comparable results in the convex hull, the oscillation of the interpolation function can pose challenges when attempting to extrapolate outside the convex hull. This phenomenon can be attributed to overfitting. The generalizable hypothesis should resemble Hermite interpolation, as it is a good fit for points within a convex hull and can extrapolate.

In this appendix, we give the error analysis of HR. We study the error through the error at each point. If $\hat{\boldsymbol{x}} \in \overset{\circ}{B} (\boldsymbol{x}_0, \delta)$, the Taylor polynomial $P(\hat{\boldsymbol{x}})$ of NN is

$$f_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}) = f_{\boldsymbol{\theta}}(\boldsymbol{x}_0) + \left[\nabla_{\hat{\boldsymbol{x}}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_0)\right]^T (\hat{\boldsymbol{x}} - \boldsymbol{x}_0) + \mathcal{R}(\hat{\boldsymbol{x}}|\boldsymbol{x}_0), \tag{5}$$

where

$$\mathcal{R}(\hat{\boldsymbol{x}}|\boldsymbol{x}_0) = \frac{1}{2!} \left[\hat{\boldsymbol{x}} - \boldsymbol{x}_0\right]^T H\left(\boldsymbol{c}\right) \left[\hat{\boldsymbol{x}} - \boldsymbol{x}_0\right]. \tag{6}$$

$\boldsymbol{c}$ is a point on the line segment induced by $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}_0$. $H\left(\boldsymbol{c}\right)$ is the Hessian matrix on $\boldsymbol{c}$. Obviously, the polynomial $P(\hat{\boldsymbol{x}})$ satisfy

$$P(\boldsymbol{x}_0) = y,\ \nabla_{\boldsymbol{x}_0} P(\boldsymbol{x}_0) = \nabla_{\boldsymbol{x}_0} y, \tag{7}$$

thus $P(\hat{\boldsymbol{x}})$ is the polynomial we want to meet the Lipschitz condition. Consequently, considering the whole training set, we give the error term of the hypothesis function:

$$r = \mathbb{E}_{\boldsymbol{x}_j \sim p(\boldsymbol{x})} \sup_{\boldsymbol{c}} \mathcal{R}(\hat{\boldsymbol{x}}|\boldsymbol{x}_j). \tag{8}$$

B Explanation of Generalization

The necessity for a well-defined hypothesis function arises due to several reasons. Revisiting the interpolation, we desired a generalization model, as depicted in Fig. 1. It can be observed that the two interpolation methods exhibit comparable performance when applied to the test points falling within the convex hull. However, although the extrapolation test points, as discussed in Balestriero et al. (2021), exhibit the phenomenon of test data always being extradited in high dimensions, Lagrange interpolation cannot be performed confidently. Overfitting can be likened to Lagrange interpolation, which exhibits good performance when applied within the convex hull but performs poorly when extrapolating outside the convex hull. A method that imposes gradient constraints, such as Hermite interpolation, can be employed to mitigate overfitting.

On the other hand, when the gradient restriction is applied, Hermite interpolation can be extended to the extraction points. The ideal hypothesis function's desired characteristic is to fit the points within the convex hull accurately but also exhibit reliable performance in extrapolation. Therefore, the NN that is trained to approximate such a function has the potential to exhibit better generalization capabilities.

C Convergence Analysis

The optimization object in Eq. 3, induces the JR term

$$\|\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2. \tag{9}$$

Notice that JR is differentiable, and if $f_{\boldsymbol{\theta}}$ is a convex function, the gradient descent can guarantee the problem converges into a global minimum as a result of Zinkevich (2003). However, while $f_{\boldsymbol{\theta}}$ is usually a complex NN, the SGD can only ensure a local minimum with the parameter update by Eq. 3:

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} - \eta \left(\nabla_{\boldsymbol{\theta}} \left\| f_{\boldsymbol{\theta}}(\boldsymbol{x}) - y \right\|^2 + \lambda \nabla_{\boldsymbol{\theta}} \left\| \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\boldsymbol{x}) \right\|^2 \right),\ \boldsymbol{x} \sim p(\boldsymbol{x}), \tag{10}$$

where η is the learning rate set as 0.01 in Sec. 3.