Gradient regularization improves performance of discriminative models

Naman Trisoliya (12141150)

1 Abstract

The paper presents the concept of regularizing the gradient norm of a neural network's output with respect to its inputs and the possibility of increasing classification accuracy, especially in the case of vision problems based on little data. It describes gradient regularization as a strong tool and explains that in the future, this technique has already been independently reintroduced several times, stressing its effectiveness in modern-day deep neural networks. The article additionally includes the authors' regularizers, which belong to a class of Jacobian-based regularizers, which the paper provides empirical validation with both real and synthetic data that shows that controlling gradients outside of training points produces generalizing solutions.

2 Key Contributions/Novelty

The main point of the text is the main idea cited to increase classification performance by applying gradient regularization, mainly with a lack of a large number of training datasets. The idea itself is not new but is given a new understanding in the context of using it for modern deep networks. Proposing it as a "regularizer with empirical evidence behind it" also falls into novelty. These regularizers are intended to be part of a more extensive class of Jacobian-based techniques.
The paper offers a design of a novel contribution known as Spectral Regularization . SpectReg approximates a random projection to the Jacobian of the logits and punishes the squared L2 norm of this projection. The goal of adding these techniques to the regular training of neural networks is to contribute to their generalization and robustness, especially for the vision task with tiny training datasets. The novelty in SpectReg is the use of random projections and the punishment of the squared L2 norms, which creates a new regulation for the training data. By projecting a random matrix to the Jacobian, SpectReg gathers the necessary information about the model's response to variations of the input, allowing the model to present smoother gradients, which helps during inference.

3 Experiment Setup

An individual experiment consists of 10 runs of the same setup, each time with a different randomly chosen training set. All reported numbers are the mean of these runs evaluated on 10000 test points, accompanied with the standard deviation in parentheses

3.1 Dataset Setup

For all experiments subset of MNIST dataset was used. For different experiments different dataset setup was required as they varied in the number of samples per class and in total number of datapoints during training of Models. Following are the setup details:

1. Train set size:
    - C2: Small MNIST: 200 samples per class.
    - C4: Total datapoints varying as :[500, 1000, 2000, 3000, 4000, 5000, 10000, 15000, 20000]
    - C6: Small MNIST: 200 samples per class.
2. Test set size: 10000 Data points
3. Batch Size: 50

3.2 Model Setup

The classic LeNet-5 architecture (LeCun et al., 1998), with maxpooling, ReLU activations and dropout after the dense hidden layer was implemented as **baseline model** however some experiments have specific model setup as follows:

1. C2:
   - Baseline: The classic LeNet-5 architecture with maxpooling, ReLU activations but without any dropout layers.
   - Batchnorm: The baseline architecture with batch normalization layers as regularizers.
   - Dropout: The baseline architecture with dropout layers as regularizers.
2. C4: The classic LeNet-5 architecture with maxpooling, ReLU activations and dropout after the dense hidden layer was taken as baseline. Different regularizer technique like DoubleBack, SpectReg, JacReg, CP were compared.
3. C6: The classic LeNet-5 architecture with maxpooling, ReLU activations and dropout after the dense hidden layer was taken as baseline. Different regularizers like SpectReg, JacReg, FrobReg were compared.

3.3 Hyperparameters Setup

1. Optimizer: Adam
2. Learning rate: 0.1 at start , 0.01 at 50% completion and 0.001 at 75% completion
3. Betas: $\beta_1 = 0.9, \beta_2 = 0.999$
4. Weight Decay: 0.0005
5. Dropout Rate: 0.5
6. Training minibatches: Training stopped after 10000 minibatches.
7. Batch size: 50

3.4 Regularizers Setup

1. Double Backpropagation (DoubleBack): Take the original loss term and penalize the squared L2 norm of its gradient.

$$L_{DG}(x, y, \Theta) = L(x, y, \Theta) + \lambda \|(\frac{\partial}{\partial x} L(x, y, \Theta))\|_2^2$$

2. Jacobian Regularizer (JacReg): Penalize the squared Frobenius norm of the Jacobian of the softmax output (probabilities) with respect to the input.

$$L_{JacReg}(x, y, \Theta) = L(x, y, \Theta) + \lambda \|J_f\|_F^2$$

3. Frobenius Regularizer (FrobReg): Penalize the squared Frobenius norm of the Jacobian of the logits with respect to the input.

$$L_{FrobReg}(x, y, \Theta) = L(x, y, \Theta) + \lambda \|J_g\|_F^2$$

4. Spectral Regularization (SpectReg): Apply a random projection to the Jacobian of the logits, and penalize the squared L2 norm of the result.

$$L_{SpectReg}(x, y, \Theta) = L(x, y, \Theta) + \lambda \|P_{rnd}(J_g)\|_2^2$$

- C2: NoGR, SpectReg, DoubleBack. Optimal DoubleBack/SpectReg weights are: Batchnorm 0.001/0.001, Dropout 50/0.01, Baseline 50/0.01.

- C4: DoubleBack, SpectReg, JacReg, CP(Confidence Penalty).

Table 8: Optimal weights for each regularizer and each training set size.

Train size	DoubleBack	SpectReg	JacReg	CP
500	50	0.03	0.3	0.01
1000	50	0.03	0.03	0.01
2000	50	0.03	1	0.01
3000	20	0.03	1	0.01
4000	20	0.03	1	0.01
5000	20	0.003	1	0.1
10000	5	0.01	1	0.1
15000	2	0.01	1	0.01
20000	2	0.001	0.3	0.03

- C6: SpectReg, JacReg and FrobReg. Optimal weights for SpectReg, JacReg and FrobReg are 0.03, 1 and 0.03, respectively.

4 Experiment Results

I tried to replicate the experiment setup as accurately as possible according to my understanding from the paper. However the results may not match exactly with the paper. The experiments in the paper included sampling random datapoints from the original dataset and no information about the seed is provided. This might be the reason for not being able to observe accurate results.

4.1 Experiment C2

GRADIENT REGULARIZATION COMPARED WITH DROPOUT AND BATCH NORMALIZATION

- **Expected Results:** Both DoubleBack and SpectReg achieve higher accuracy than either Dropout or Batchnorm in itself.

	Dataset	NoGR	SpectReg	DoubleBack
Baseline	small MNIST	96.99 (0.15)	97.59 (0.13)	97.56 (0.24)
Batchnorm	small MNIST	96.89 (0.23)	96.94 (0.27)	96.89 (0.22)
Dropout	small MNIST	97.29 (0.19)	97.65 (0.14)	**97.98** (0.12)

- **Observed Results:**

Table 1: Comparison of Dropout and Batch Normalization versus DoubleBack and SpectReg on small MNIST.

	Dataset	NoGR	SpectReg	DoubleBack
Baseline	small MNIST	96.77 (0.31)	96.61 (0.67)	96.32 (0.72)
Batchnorm	small MNIST	97.13 (0.16)	97.20 (0.14)	97.30 (0.23)
Dropout	small MNIST	91.90 (0.28)	91.65 (0.64)	91.30 (0.35)

DoubleBack (96.61) and SpectReg (96.32) achieve higher accuracy than Dropout (91.90) but achieve lower accuracy than Batchnorm (97.13).

4.2 Experiment C4

THE EFFECT OF TRAINING SET SIZE

- **Expected Results:** DoubleBack performs best for all sizes. SpectReg is better than CP for smaller sizes, but its advantage evaporates with more training data. JacReg is consistently worse than its peers.
- **Observed Results:**

Table 3: Accuracy score for each train size and each regularizer.

Train size	DoubleBack	SpectReg	JacReg	CP
500	94.17 (0.41)	94.06 (0.38)	93.83 (0.41)	94.05 (0.41)
1000	95.54 (0.22)	95.46 (0.34)	95.36 (0.35)	95.58 (0.31)
2000	96.56 (0.18)	96.46 (0.24)	96.60 (0.14)	96.60 (0.14)
3000	97.11 (0.22)	96.97 (0.29)	96.98 (0.20)	97.05 (0.18)
4000	97.30 (0.17)	97.36 (0.19)	97.31 (0.23)	97.33 (0.30)
5000	97.46 (0.13)	97.46 (0.15)	97.57 (0.19)	97.55 (0.15)
10000	98.00 (0.18)	97.94 (0.12)	98.02 (0.16)	97.95 (0.17)
15000	98.10 (0.17)	98.09 (0.10)	98.06 (0.15)	98.07 (0.11)
20000	98.22 (0.17)	98.29 (0.14)	98.23 (0.22)	98.26 (0.15)

  - DoubleBack does not performs best for almost all sizes but yes it achieves very close to best results for all sizes.
  - SpectReg does not perform good in small data sizes but performs good for large data sizes (>4000).
  - For most sizes JacReg performs worse than its peers.

4.3 Experiment C6:

APPROXIMATING THE FROBENIUS NORM OF THE JACOBIAN DOES NOT DECREASE ACCURACY

- **Expected Results:** We also conclude that minimizing a random projection of the Jacobian does not lead to a loss in accuracy, compared with the full calculation of the Jacobian. SpectReg performs best, but the differences are small.

Dataset	SpectReg	JacReg	FrobReg
small MNIST	**97.79%** (0.12)	97.63% (0.15)	97.76% (0.13)

- **Observed Results:**

Table 2: Accuracy score for each train size and each regularizer.

Dataset	SpectReg	JacReg	FrobReg
small MNIST	96.48% (0.19)	96.581% (0.22)	96.62% (0.22)

  - FrobReg performed best but the differences are small.

5 Conclusion

In conclusion, the paper "Regularizing Gradient Norms for Improved Classification Accuracy" introduces Spectral Regularization (SpectReg) as a novel regularization technique within the broader class of Jacobian-based regularizers. SpectReg utilizes random projections and penalizes the squared L2 norm to improve model generalization, especially in scenarios with limited training data.

The key contributions of this work include:

1. Novelty in Regularization: SpectReg adds a unique approach to gradient regularization, enhancing model robustness and interpretability.
2. Unified Framework: The paper presents a unified framework for gradient regularization approaches, including SpectReg, providing a systematic understanding of their impact on classification accuracy.
3. Empirical Validation: Through experiments, the authors demonstrate SpectReg's effectiveness as an unbiased estimator of the Jacobian's squared Frobenius norm.

4. Practical Applicability: SpectReg and related techniques are easily implementable using modern tensor libraries, making them accessible for deep architectures.

The paper suggests future research directions, such as addressing limitations in input space metrics and exploring gradient regularization with hidden layers. Overall, the contributions of this work advance the field of deep learning regularization and model performance.