Abstract

Aim: Understand why certain neural networks generalize better than others.

Why are we studying this?

-Will make NN more interpretable

-More principled and reliable architecture design

Traditional Beliefs: Properties of model or regularization techniques

Disprove traditional Beliefs & deeper understanding of role of various parameters to find what controls generalization

Observations by authors:

1. State of the art CNN for image classification trained with SGD easily fit a random labelling of training data
2. Similar result for explicit regularization
3. Similar result if true images replaced by random noise
4. Simple depth 2 NN have perfect finite sample expressivity as long as number of parameters exceeds number of data points (usually happens)

Introduction

Generalization error: Difference between training and test error

Complexity measures proposed by statistical learning theory for controlling generalization error:

1. VC Dimension
2. Rademacher complexity (similar to VC Dimension)
3. Uniform stability- Sensitivity (how variation of input can influence output of system)

Author later shows that all the 3 measures above are not possible explanations of generalization error.

According to theory, when the number of parameters is large some regularization is required to ensure small generalization error.

Regularization - Implicit & Explicit

1.1 Author's Contributions

1. **Randomization test**
   a. **Random Labels**

      -True labels replaced by random labels ; standard architectures

      -Deep NN easily fit random labels

      -0 training error was found in this case, however test error was bad

      -By randomizing labels alone, generalization error jumps up without changing the model, its size, hyperparameters, optimizer

1.The effective capacity of neural networks is sufficient for memorizing the entire data set.
2. Even optimization on random labels remains easy. In fact, training time increases only by a small constant factor compared with training on the true labels.
3. Randomizing labels is solely a data transformation, leaving all other properties of the learning problem unchanged.

b. Replace true images by completely random pixels (e.g Gaussian Noise)

-Fit data with 0 training error
-Vary amount of randomization
-steady deterioration of the generalization error as we increase the noise level.
-This shows that neural networks are able to capture the remaining signal in the data, while at the same time fit the noisy part using brute-force.

2. **Explicit Regularization**

-Weight decay
-Data Augmentation
-Dropout

-May improve generalization performance but not sufficient or necessary for controlling generalization error.

-In deep learning, explicit regularization is more of a tuning parameter

Finite Sample Expressivity

-simple 2 layer ReLu network with p=2n+d parameters can express any labelling of any size n in d dimensions.
-depth 2 network has large width
-But, a depth k network can also be developed with each layer having only O(n/k) parameters
-Shows that even depth 2 networks of linear size can represent any labeling of training data
-While prior expressivity results focused on what functions neural nets can represent over the **entire domain**, here focus is instead on the expressivity of neural nets with regards to a **finite sample**.

Implicit Regularization

-In NN, models are chosen as an output of running SGD
For linear models, SGD always converges to a solution with small
norm. Hence, the algorithm itself is implicitly regularizing the solution. Indeed, we show on small data sets that even Gaussian kernel methods can generalize well with no regularization. Though this doesn't explain why certain architectures generalize better than other architectures, it does

suggest that more investigation is needed to understand exactly what the properties are inherited by models that were trained using SGD.

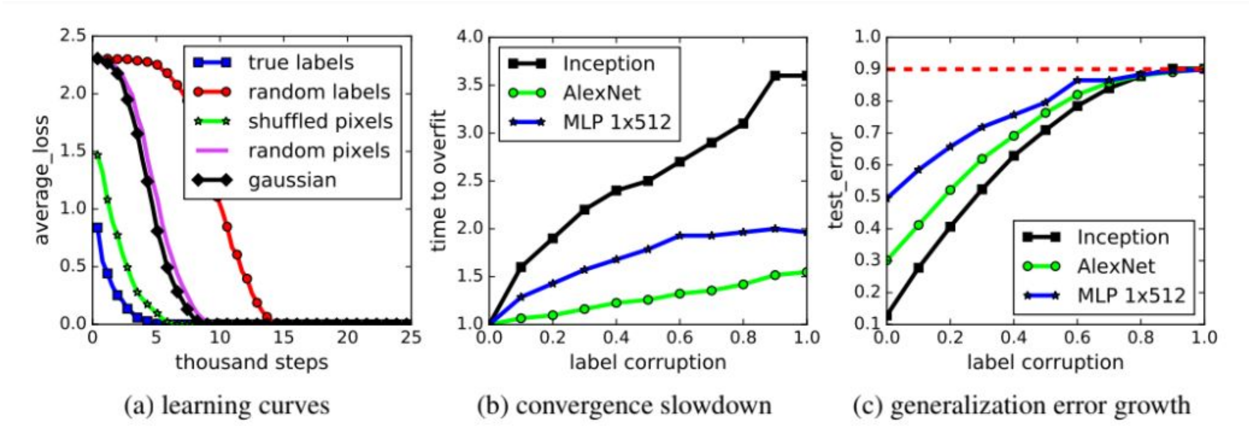
1.2

-Uniform stability of learning algorithm is independent of labeling of training data
-concept is not strong enough to distinguish between the models trained on the true labels (small generalization error) and models trained on random labels (high generalization error).

----come back later----

2. Effective capacity of Neural Networks

-Trained NN on randomly labelled data. Learning is impossible. Impossibility should be reflected in training by not converging or slowing down substantially.
-Several properties of training process remain unaffected for multiple standard architectures



(a) learning curves (b) convergence slowdown (c) generalization error growth

Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

For random labels:
a) we do not need to change the learning rate schedule;
b) once the fitting starts,it converges quickly;
c) it converges to (over)fit the training set perfectly. Also note that “random pixels” and “Gaussian” start converging faster than “random labels”.

-random pixels, the inputs are more separated from each other than natural images that originally belong to same category, hence easier to build network for arbitrary labels
belong to the same category, therefore, easier to build a network for arbitrary label assignments.

Random Pixels-Different random permutation applied to each image independently

Potential to note more observations

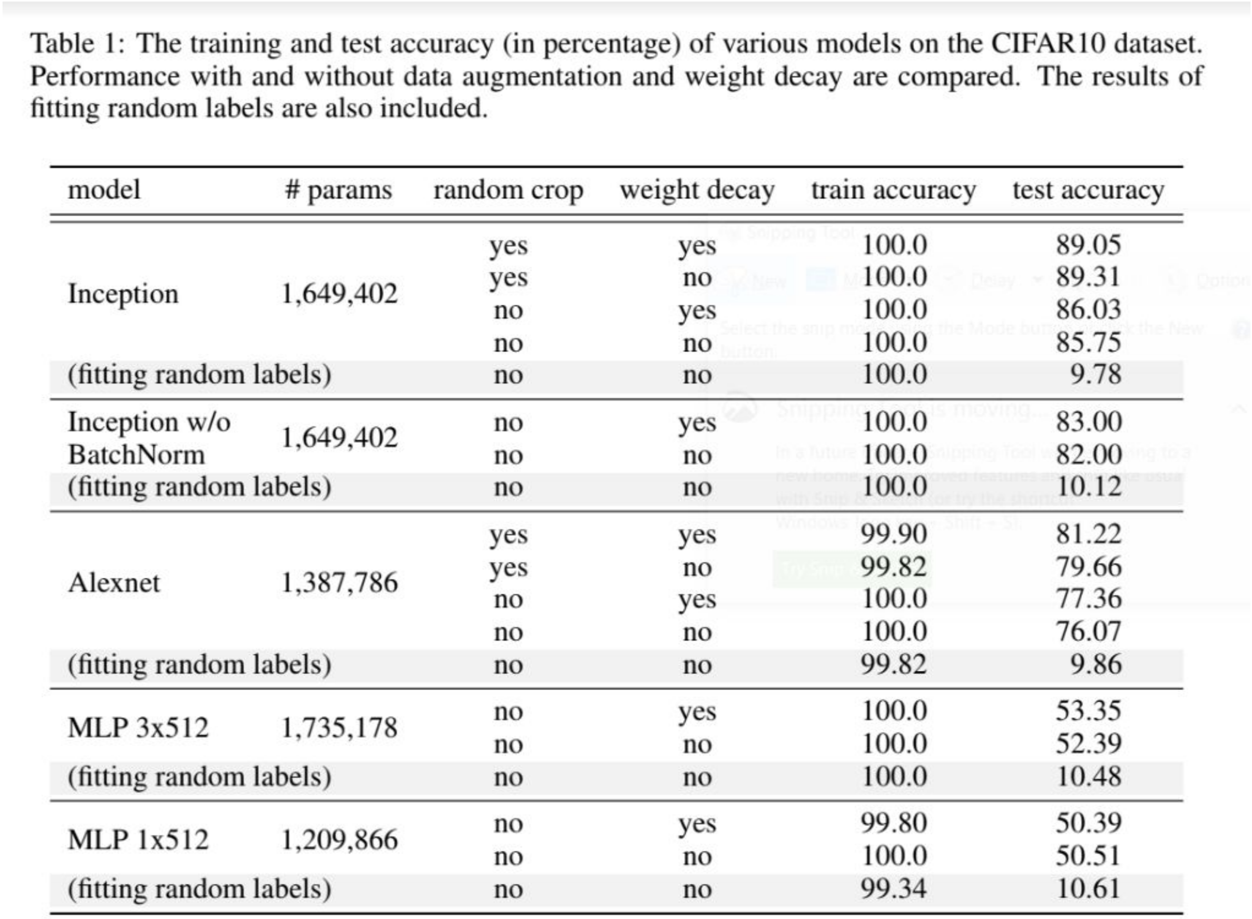


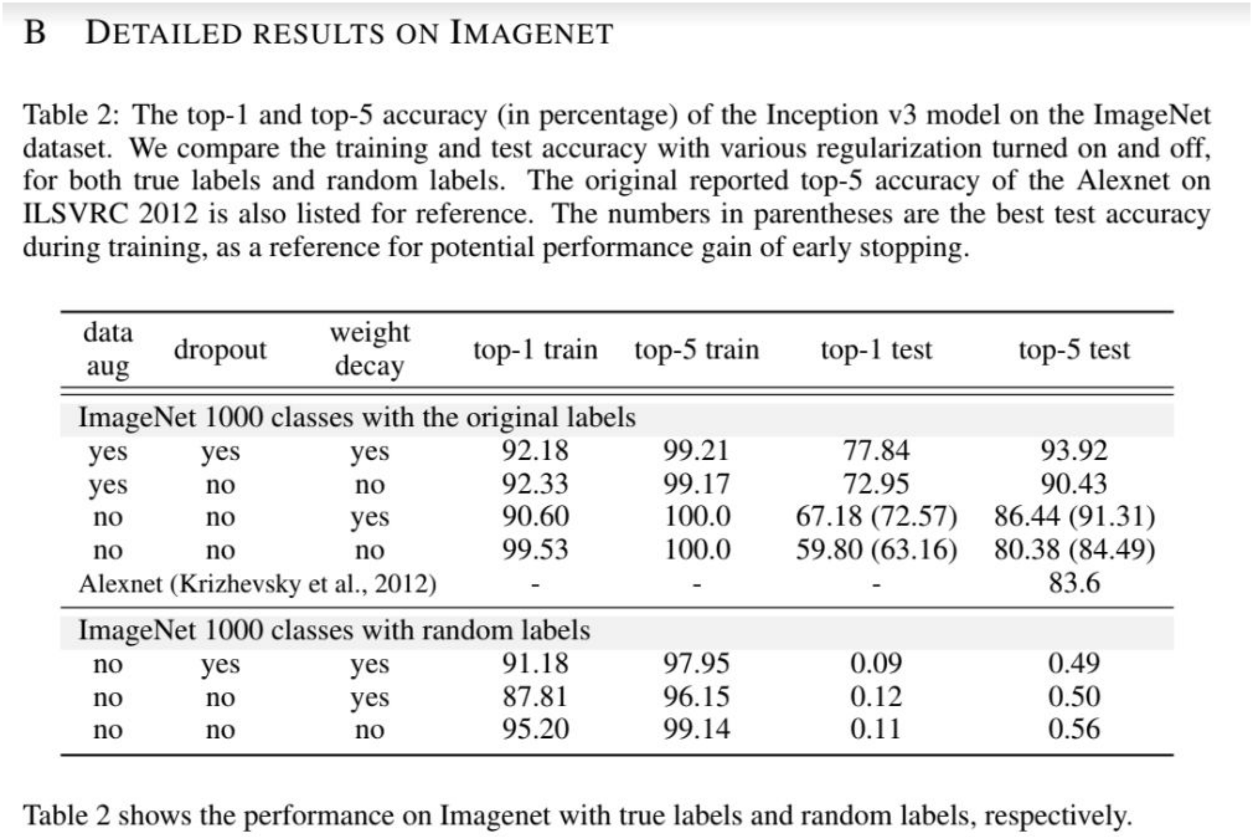
Table 1: The training and test accuracy (in percentage) of various models on the CIFAR10 dataset. Performance with and without data augmentation and weight decay are compared. The results of fitting random labels are also included.

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75
(fitting random labels)		no	no	100.0	9.78
Inception w/o BatchNorm	1,649,402	no	yes	100.0	83.00
		no	no	100.0	82.00
(fitting random labels)		no	no	100.0	10.12
Alexnet	1,387,786	yes	yes	99.90	81.22
		yes	no	99.82	79.66
		no	yes	100.0	77.36
		no	no	100.0	76.07
(fitting random labels)		no	no	99.82	9.86
MLP 3x512	1,735,178	no	yes	100.0	53.35
		no	no	100.0	52.39
(fitting random labels)		no	no	100.0	10.48
MLP 1x512	1,209,866	no	yes	99.80	50.39
		no	no	100.0	50.51
(fitting random labels)		no	no	99.34	10.61

See darkened rows in table. On CIFAR 10 dataset, AlexNet and MLPs converge to 0 loss for the training set.

-Behavior of neural network training with a varying level of label corruptions from 0 (no corruption) to 1 (complete random labels) on the CIFAR10 dataset. The networks fit the corrupted training set perfectly for all the cases.
-Figure 1b shows the slowdown of the convergence time with increasing level of label noises.
-Figure 1c depicts the test errors after convergence. Since the training errors are always zero, the test errors are the same as generalization errors. As the noise level approaches 1, the generalization errors converge to 90% —the performance of random guessing on CIFAR10.

CIFAR 10 dataset- 50,000 training images and 10 classes



B Detailed results on Imagenet

Table 2: The top-1 and top-5 accuracy (in percentage) of the Inception v3 model on the ImageNet dataset. We compare the training and test accuracy with various regularization turned on and off, for both true labels and random labels. The original reported top-5 accuracy of the Alexnet on ILSVRC 2012 is also listed for reference. The numbers in parentheses are the best test accuracy during training, as a reference for potential performance gain of early stopping.

data aug	dropout	weight decay	top-1 train	top-5 train	top-1 test	top-5 test
ImageNet 1000 classes with the original labels						
yes	yes	yes	92.18	99.21	77.84	93.92
yes	no	no	92.33	99.17	72.95	90.43
no	no	yes	90.60	100.0	67.18 (72.57)	86.44 (91.31)
no	no	no	99.53	100.0	59.80 (63.16)	80.38 (84.49)
Alexnet (Krizhevsky et al., 2012)			-	-	-	83.6
ImageNet 1000 classes with random labels						
no	yes	yes	91.18	97.95	0.09	0.49
no	no	yes	87.81	96.15	0.12	0.50
no	no	no	95.20	99.14	0.11	0.56

Table 2 shows the performance on Imagenet with true labels and random labels, respectively.

Without explicit regularization, 95.2% top1 accuracy is observed. This is quite good considering million random labels from 1000 categories
Reaches 90% accuracy even with explicit regularizers turned on

2.2 Challenging initial beliefs about role of VC Dimension, Rademacher Complexity and Uniform Stability in Generalization

-VC Dimension, Rademacher Complexity - not clear

-Uniform stability-Uniform stability of an algorithm A measures how sensitive the algorithm is to the replacement of a single example. However, it is solely a property of the algorithm, which does not take into account specifics of the data or the distribution of the labels.

The weakest stability measure is directly equivalent to bounding generalization error and does take the data into account. However, it has been difficult to utilize this weaker stability notion effectively.

3. Role of Regularization

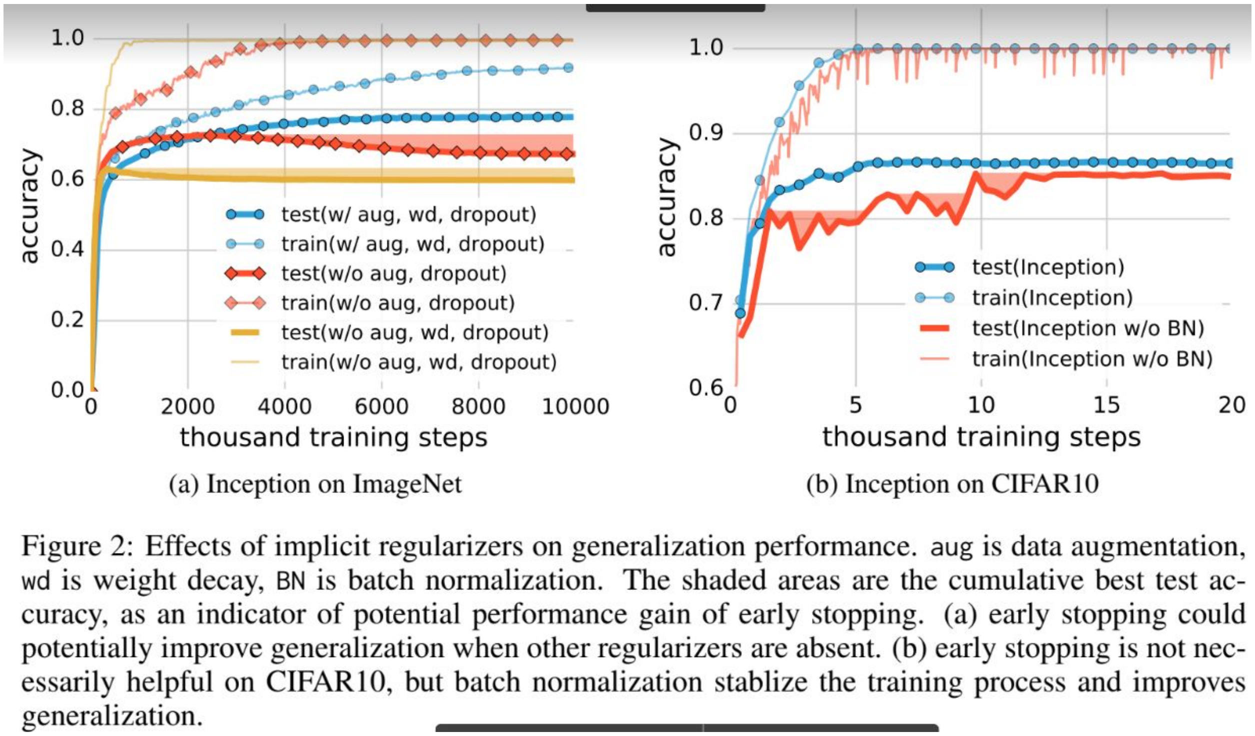
From table 2 , we can see that Inception v3 fits randomly labelled data well with dropout and weight decay(91%)
Augmenting data is more powerful than weight decay and dropout

-18% drop in top 1 accuracy is observed when no regularization is used
-AlexNet vs Inception

Potential to note more observations

3.1 Implicit Regularization

-early stopping can improve performance
-batch normalization improves generalization performance but impact on generalization is 3-4%.
-



(a) Inception on ImageNet

(b) Inception on CIFAR10

Figure 2: Effects of implicit regularizers on generalization performance. aug is data augmentation, wd is weight decay, BN is batch normalization. The shaded areas are the cumulative best test accuracy, as an indicator of potential performance gain of early stopping. (a) early stopping could potentially improve generalization when other regularizers are absent. (b) early stopping is not necessarily helpful on CIFAR10, but batch normalization stablize the training process and improves generalization.

Results of explicit and implicit regularization show that regularizers are unlikely to be a fundamental reason for generalization as the networks continue to perform well without regularization.

4. Finite sample expressivity

The authors emphasize that expressive power of neural networks on a finite sample of size n instead of on the entire domain.

There exists a two-layer neural network with ReLU activations and 2n+d weights that can represent any function on a sample of size n in d dimensions.
As soon as the number of parameters p of a networks is greater than n, even simple two-layer neural networks can represent any function of the input sample.

5. Study generalization in linear models to study if parallel insights can help understand neural networks better

Inspected which solution SGD converges to.
It was thus found that we can perfectly fit any set of labels by forming the gram matrix and solving linear system K(alpha)=y

MNIST- without preprocessing, test error of 1.2% obtained by simply solving equation
Gabor wavelet transform and then solving equation, error drops
Regularization does not improve performance
CIFAR 10- Applying a Gaussian kernel on pixels and using no regularization achieves 46% test error. By preprocessing with a random convolutional neural net with 32,000 random filters, this test error drops to 17% error.
Adding l2 regularization further reduces this number to 15% error. Note that this is without any data augmentation.

Kernel solution is equivalent to minimum l2 norm solution of Xw=y. SGD often converges to minimum norm solution..
Unfortunately, this notion of minimum norm is not predictive of generalization performance.

6.
Effective capacity of several successful neural network architectures is large enough to shatter the training data
These models are in principle rich enough to memorize the training data.
This situation poses a conceptual challenge to statistical learning theory as traditional measures of model complexity struggle to explain the generalization ability of large artificial neural networks.
We argue that we have yet to discover a precise formal measure under which these enormous models are simple. Another insight resulting from our experiments is that optimization continues to be empirically easy even if the resulting model does not generalize. This shows that the reasons for why optimization is empirically easy must be different from the true cause of generalization.