Generalization in Deep Learning

An overview of Generalization in Deep Learning

Enes Dilber

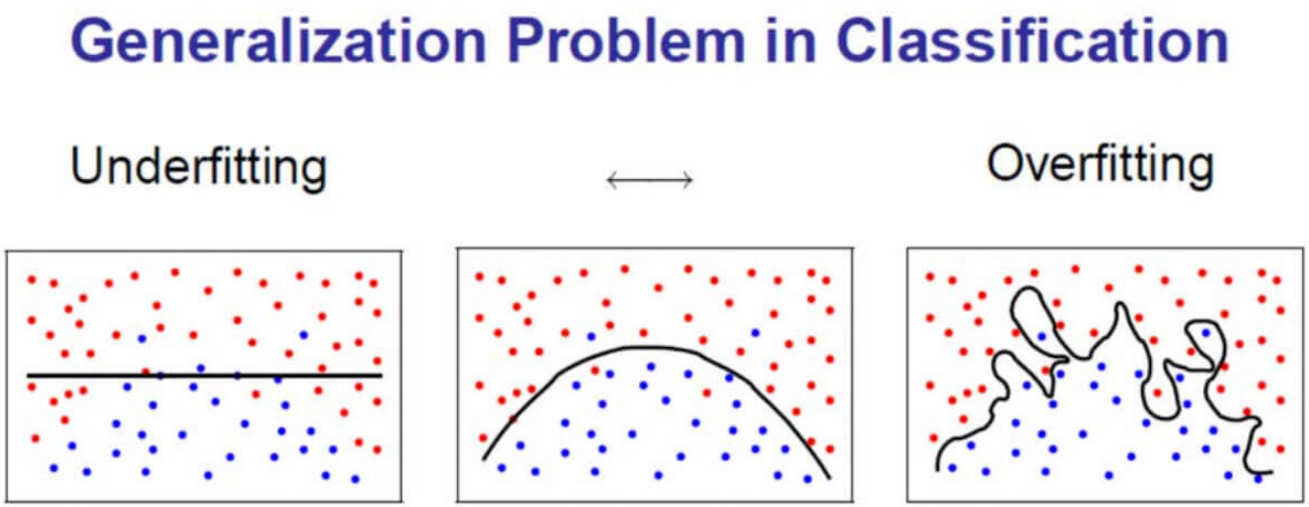
Middle East Technical University

ImageLab Seminars, Spring 17

Outline

Difference between Memorization and Generalization

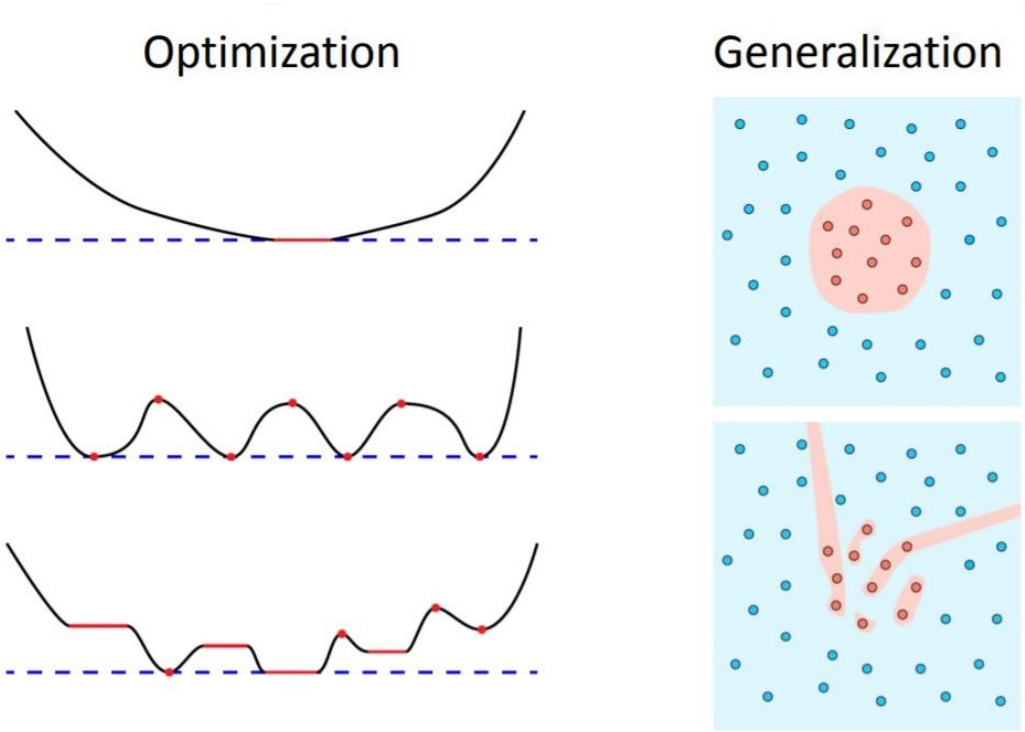
- Memorizing, given facts, is an obvious task in learning. This can be done by storing the input samples explicitly.
- The ability to identify the rules, to generalize, allows the system to make predictions on unknown data.



How well can neural networks generalize?

- Cybenko (1989) proved that a feed-forward network with a multilayer perceptron can approximate continuous functions on compact subsets of IR with using sigmoid activation function.
- Many strategies used in machine learning are explicitly designed to reduce the test error, possibly at the expense of increased training error. These strategies are known collectively as regularization.

Neural Networks Optimization Scheme



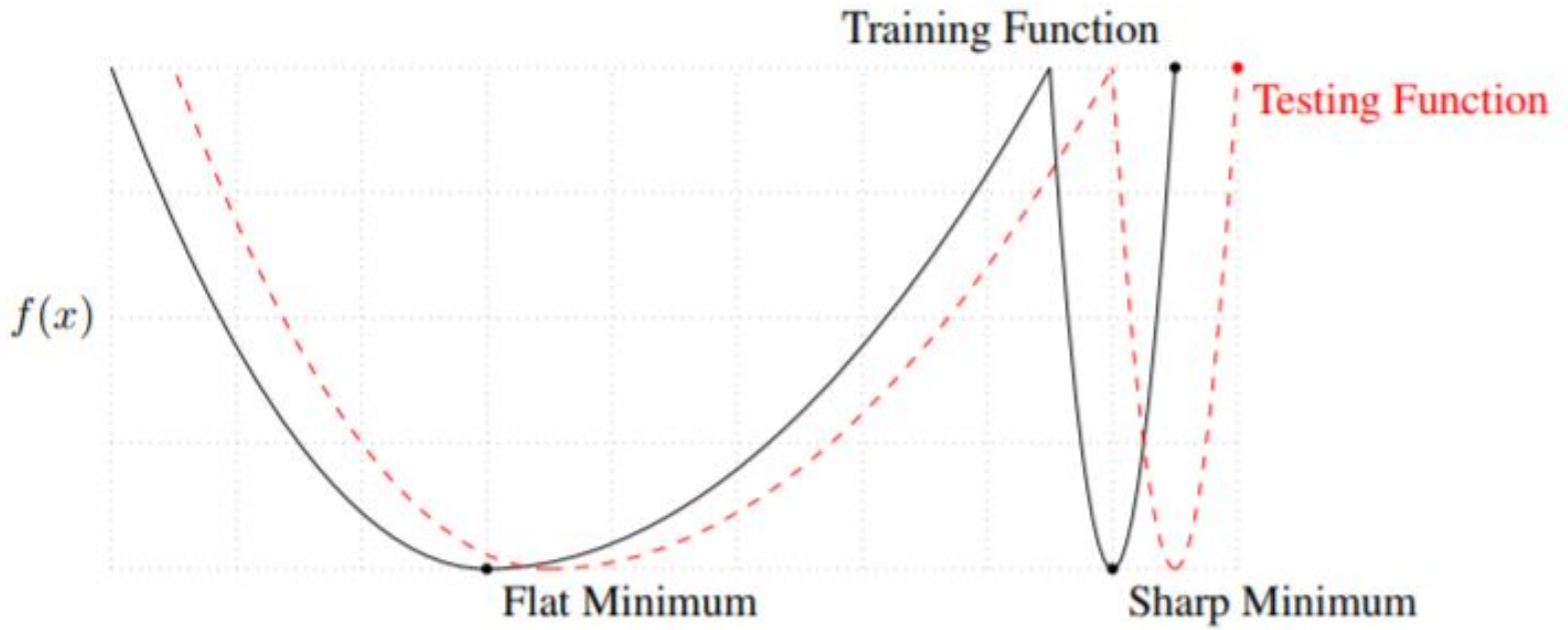
Understanding deep learning requires rethinking generalization - Zhang et al. (2016)

- State-of-the-art convolutional networks for image classification trained with stochastic gradient methods easily fit a random labeling of the training data.
- This phenomenon is qualitatively unaffected by explicit regularization, and occurs even if they replace the true images by completely unstructured random noise.
- Consequently, 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.

On Large-Batch Training For Deep Learning: Generalization Gap And Sharp Minima - Keskar et al. (2016)

- Large-batch minimizers are characterized by a significant number of large positive eigenvalues in $\nabla^2 f(x)$. Large Hessian leads poor generalization.
- They claim that small batches in SGD finds wide minima and that generalizes well.
- To use large batches they suggest warm-starting weights.

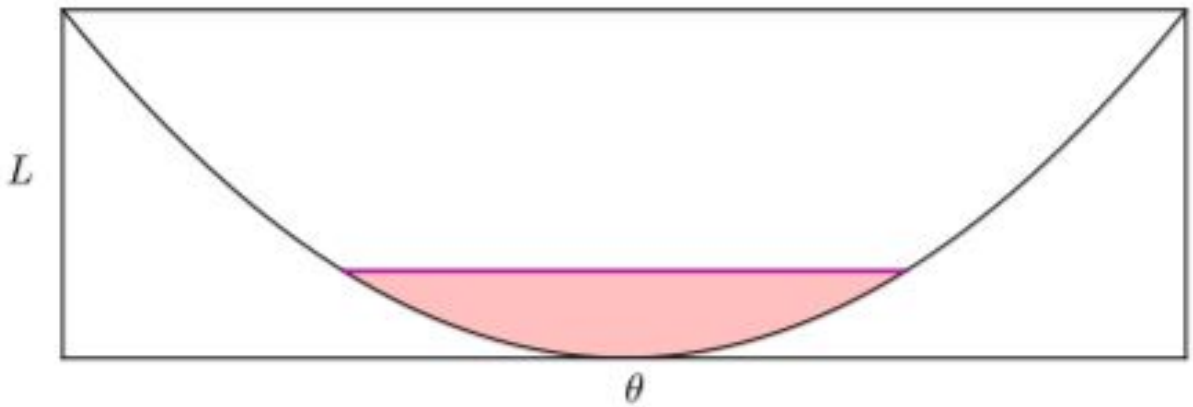
On Large-Batch Training For Deep Learning: Generalization Gap And Sharp Minima - Keskar et al. (2016)



Outline

Definition of flatness/sharpness

- $C(L, \theta, \epsilon)$ as the largest connected set containing θ such that $\forall \theta' \in C(L, \theta, \epsilon)$, $L(\theta') < L(\theta) + \epsilon$. The ϵ - flatness will be defined as the volume of $C(L, \theta, \epsilon)$. It is the purple line.



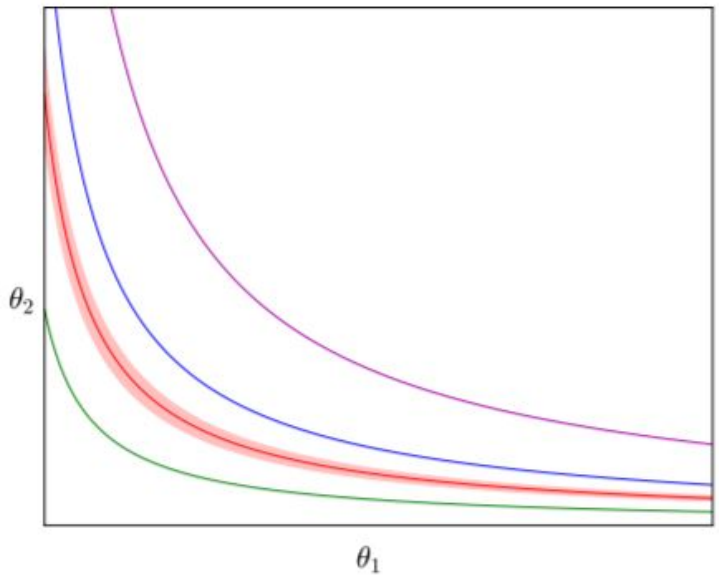
Outline

Deep Rectified Networks

- Most of the results will be on the deep rectified feedforward networks with a linear output layer. Though they can easily be extended to other architectures (e.g. convolutional, etc.).
- $y = \Phi_{rect}(\Phi_{rect} \dots (\Phi_{rect}(x\theta_1) \dots)\theta_{K-1})\theta_K$

Properties of Deep Rectified Networks

- A unit change in the behavior of the model is not constant over parameter space.
- Specifically, each line of a given color corresponds to the parameter assignments (θ_1, θ_2) that result observationally in the same prediction function f_θ.
- Without changing the function value, flatness can be changed by moving along the curve.



Non-negative Homogeneity

- Non-negative Homogeneity:
  Under some conditions, a global (nonconvex) optimization problem with quadratic data is equivalent to a convex minimization problem. (Lasserre et al. 2002)

Definition

A given function ϕ is non-negative homogeneous if

$$\forall (z, a) \in \mathbb{R} \times \mathbb{R}^{+}, \phi(az) = a\phi(z)$$

Non-negative Homogeneity

- For a rectified unit it means:

$$\Phi_{rect}(x(a\theta_1))\theta_2 = \Phi_{rect}(x\theta_1)a\theta_2$$

- For this one (hidden) layer neural network, the parameters $(a\theta_1, \theta_2)$ is observationally equivalent to $(\theta_1, a\theta_2)$. This observational equivalence similarly holds for convolutional layers.

Some Common Non-negative Homogeneous Layers

- Fully Connected + ReLU
- Convolution + ReLU
- Max Pooling
- Linear Layers
- Mean Pooling
- Max Out
- Not Sigmoid

Outline

Model Reparametrization

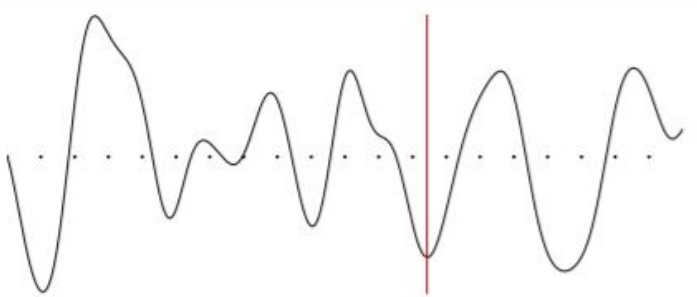
Definition

If we are allowed to change the parametrization of some function f, we can obtain arbitrarily different geometries without affecting how the function evaluates on unseen data. The same holds for reparametrization of the input space.

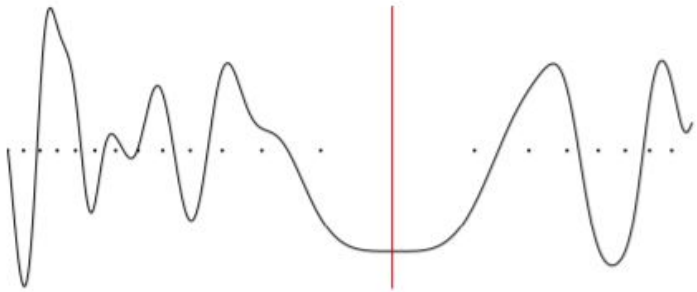
- Flatness of minima to their probable generalization is that the choice of parametrization and its associated geometry are arbitrary.
- Since we are interested in finding a prediction function in a given family of functions, no reparametrization of this family should influence generalization of any of these functions. Given a bijection g onto θ, we can define new transformed parameter $\kappa = g^{-1}(\theta)$. Since θ and κ represent in different space the same prediction function, they should generalize as well.

Exploiting Loss Function

- A one-dimensional example on how much the geometry of the loss function depends on the parameter space chosen.
- They carefully used some parameterization techniques such as weight normalization and batch normalization.



(a) Loss function with default parametrization



(b) Loss function with reparametrization

(c) Loss function with another reparametrization

Discussion

- In earlier researches, it has been observed empirically that minima found by standard deep learning algorithms that generalize well tend to be flatter than found minima that did not generalize.
- The whole geometry of the error surface with respect to the parameters can be changed arbitrarily under different parametrizations.
- Minima concept can not be divorced from the particular parametrization of the model or input space.

Outline

Expected Risk

Definition

$R[f]$: Expected Risk
$\hat{R}_m[f]$: Empirical Risk

$$R[f] = E_{x,y\ p(x,y)}[loss(f(x), y)]$$

$$\hat{R}_m[f] = \frac{1}{m}\sum_{i=1}^{m} loss(f(x_i), y_i)$$

- The aim in machine learning is minimization of expected risk by minimizing the computable empirical risk. The goal of generalization theory is questioning *how this approach is a sensible one.*

Generalization Gap

Definition

Generalization Gap $\triangleq R[f_A(S_m)] - \hat{R}[f_A(S_m)]$ where;
$f_A(S_m) : X \to Y$ be a model learned by algorithm A with a training dataset S_m

- Generalization Gap is not computable. This is the difference between error on the training set and error on the underlying joint probability distribution.
- The aim of many problems in statistical learning theory is to bound or characterize the generalization error in probability.

Apparent Paradox

- Zhang et al. (2017a) empirically showed that several deep model classes can memorize random labels, while having the ability to produce zero training error and small test errors for particular natural datasets. This phenomena is even true for all ML algorithms.
- For any model class F whose model complexity is large enough to memorize any dataset and which includes f_ϵ possibly at an arbitrarily sharp minimum, there exist learning algorithms A such that the generalization gap of $f_{A(S_m)}$ is at most ϵ.
- There exist arbitrarily unstable and arbitrarily non-robust algorithms A such that the generalization gap of $f_{A(S_m)}$ is at most ϵ.
- The current generalization theory states that how we get f and the set from which we choose f are all that matters.

Generalization Puzzle

- He suggest that the methods that we use for generalization such as flat minima, small complexity and robustness imply small generalization gap, however small generalization gap does not imply flat minima, small complexity and robustness.
- p: model is stable, robust and has flat minima.
  q: model generalize well
  p implies q does not imply *q implies p*.
- In space of $f \in F$ there are such learning mechanisms that are unstable and non-robust but generalize well.

Practical Role of Generalization Theory

1. Provide guarantees on expected risk.
2. Guarantee generalization gap
    1. To be small for a given fixed S_m, and/or
    2. to approach zero with a fixed model class as m increases.
3. Provide theoretical insights to guide the search over model classes.

Outline

Training-Validation Paradigm

- Search over model classes by changing architecture to obtain low validation loss.
- *We can generalize well because we can obtain a good model via model search with validation errors.* In deep learning community there is a cumulative traditional knowledge for generalization.
- Provide theoretical insights to guide the search over model classes.

Upper bound for Expected Risk

Proposition

For any $\delta > 0$. This bound holds with probability $1 - \delta$

$$R[f] \leq \hat{R}_{val}[f] + \frac{2C\ ln(\frac{|F_{val}|}{\delta})}{3m_{val}} + \sqrt{\frac{2\gamma^2\ ln(\frac{|F_{val}|}{\delta})}{m_{val}}}$$

- $|F_{val}|$ is the number of times we use the validation dataset in our decision making to choose a final model.
- Consider *MNIST* example with $m_{val} = 10000$ and $\delta = 0.1$. In a worst-case scenerio($C = 1,\ \gamma^2 = 1,\ |F_{val} = 1,000,000,000|$): $R[f] \leq \hat{R}_{val}[f] + 6.94\%$
- In a realistic scenario it became $R[f] \leq \hat{R}_{val}[f] + 0.49\%$

Outline

Model Description

Definition

Here is form of k-th output unit. It is structure of a Directed Acyclic Graph with an activation. H is the number of hidden layers.

$$h_k^{(H+1)} = \sum_{path} \bar{x}_{path} \bar{\sigma}_{path}(x, w_\sigma) \bar{w}_{path,k}$$

$$= [\bar{x} \circ \bar{\sigma}(x.w_\sigma)]^T \bar{w}_k$$

Method

Proposition

$$loss = original\ loss + \frac{\lambda}{\bar{m}} \hat{E}_{S_m,\xi} \left[\max_k \sum_{i=1}^{m} \xi h_k^{(H+1)}(x_i) \right]$$

- ξ: *Rademacher variable* ($\sim$ *Uniform*$(-1, 1)$)
- $\bar{m}$: Batch Size of SGD
- $\sum_{i=1}^{m} \xi h_k^{(H+1)}(x_i)$: a family of regulizers

Directly Approximately Regularizing Complexity (DARC1)

Definition

$$loss = original\ loss + \frac{\lambda}{\bar{m}} \left(\max_k \sum_{i=1}^{\bar{m}} |h_k^{(H+1)}(x_i)| \right)$$

- $h_k^{(H+1)}(x_i)$ is already computed in the original loss.
- With this family of regularizers they claim that they could outperformed the state-of-the-art models on CIFAR-10 and MNIST.

Outline

A simple Bayesian Model

- Let X be training inputs, Y training labels and M be classification model with a single parameter ω.

Definition

$$P(\omega|Y,X;M) = \frac{P(Y|\omega,X;M)P(\omega;M)}{P(Y|X;M)}$$
$$\propto \underbrace{P(Y|\omega,X;M)}_{\text{Likelihood}} \underbrace{P(\omega;M)}_{\text{Prior}}$$

Posterior

- The Likelihood, $P(Y|\omega, X; M) = \prod_i P(y_i|\omega, x_i; M) = e^{-H(\omega;M)}$
- Cross Entropy for one hot labels $= H(\omega; M) = -\sum_i ln(P(y_i|\omega, x_i, ; M))$
- Prior is a Gaussian, $P(\omega; M) = \sqrt{\lambda/2\pi}e^{-\lambda\omega^2/2}$
  Where, λ is regularization coefficient.
- Posterior will be, $P(\omega|Y, X; M) \propto \sqrt{\lambda/2\pi}e^{-C(\omega;M)}$
  Where, $C(\omega; M) = H(\omega; M) + \lambda\omega^2/2$
  $C(\omega; M)$ denotes L2 regularized cross-entropy.
- ω_0 is the value that maximizes the density.

Prediction of New Input

$$P(y_t|x_t, X, Y; M) = \int P(y_t|\omega, x_t; M)P(\omega|Y, X; M)d\omega$$

- These integrals are dominated by the region near ω_0, and since $P(y_t|\omega, x_t; M)$ is smooth, it could be approximated by $P(y_t|x_t; M) \approx P(y_t|\omega_0, x_t; M)$.

Modal Comparison

$$\frac{P(M1|Y,X)}{P(M2|Y,X)} = \frac{P(Y|X;M1)P(M1)}{P(Y|X;M2)P(M2)}$$

- To eliminate subjectivity, Prior ratio could be considered as 1.
- To Calculate likelihood ratio we need to find normalizing constant at the posterior finding for ω.

$$P(Y|X;M) = \int P(Y|\omega,X;M)P(\omega;M)d\omega = \sqrt{\frac{\lambda}{2\pi}} \int e^{-C(\omega;M)} d\omega$$

- $C(\omega;M) \approx C(\omega_0) + C''(\omega_0)(\omega-\omega_0)^2/2$

Modal Comparison

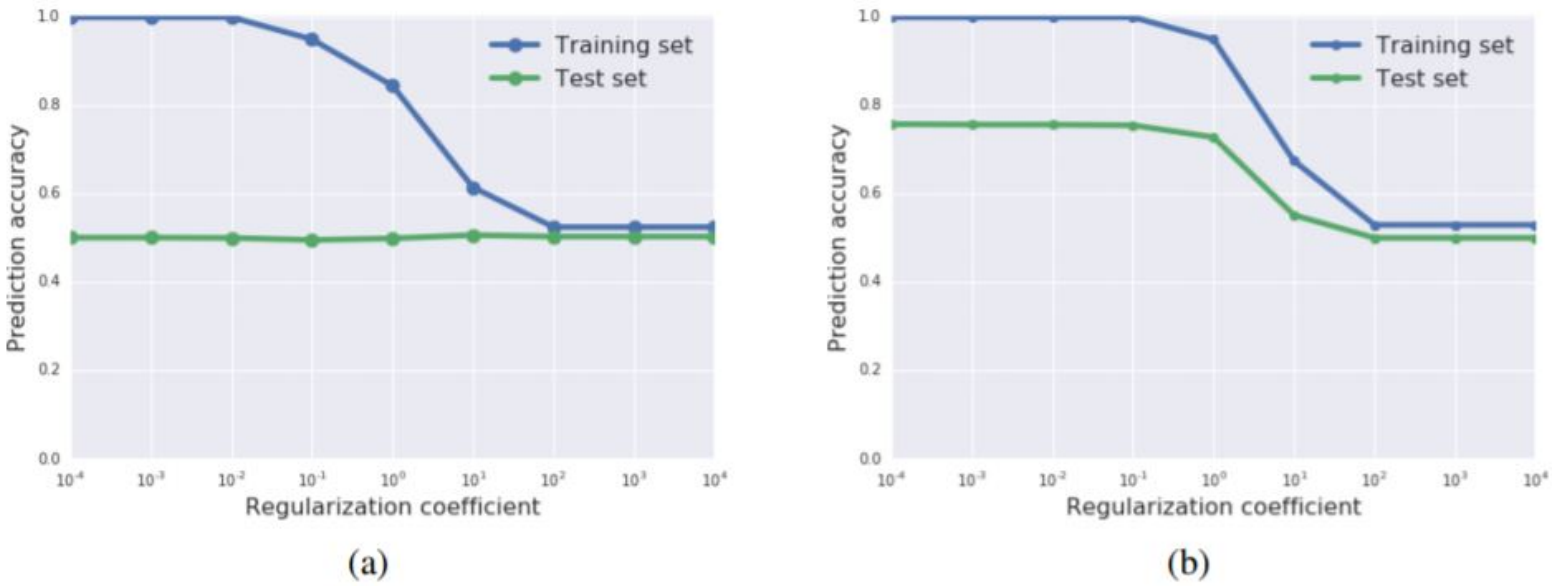
$$\frac{P(Y|X;M)}{P(Y|X;NULL)} = e^{-E(\omega_0)}$$

- *NULL* model assumes labels are entirely random. No parameters, only controlled by likelihood.
- $E(\omega_0) = C(\omega_0) + (1/2)\sum_i ln(\lambda_i/\lambda) - Nln(n)$.
  Log evidence ratio in favor of the *NULL* model
- The Bayesian evidence supports the intuition that broad minima generalize better than sharp minima, but unlike the curvature it does not depend on the model parameterization.
- Deriving Bayesian Evidence in Deep Neural Networks is nearly impossible.

Experiment Setup

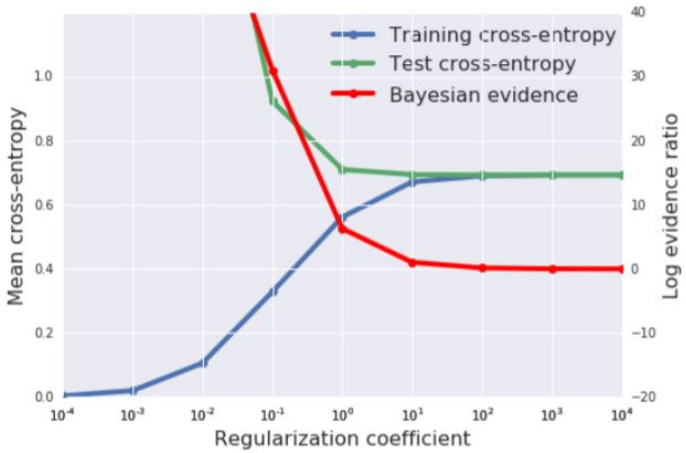
- They compare their intuition with observations of Zhang et al. (2016)
- For input, they generated 200 with size 200 random samples from Gaussian distribution each belongs to one of two classes. Test set contains 10000 examples. The loss is L2-regularized sigmoid cross-entropy.
- They crerated two tasks:
  1. The labels of both the training and test sets are random
  2. The label $y = 1$ if $\sum_i x_i > 1$, $y = 0$ otherwise

Effect of Regularization

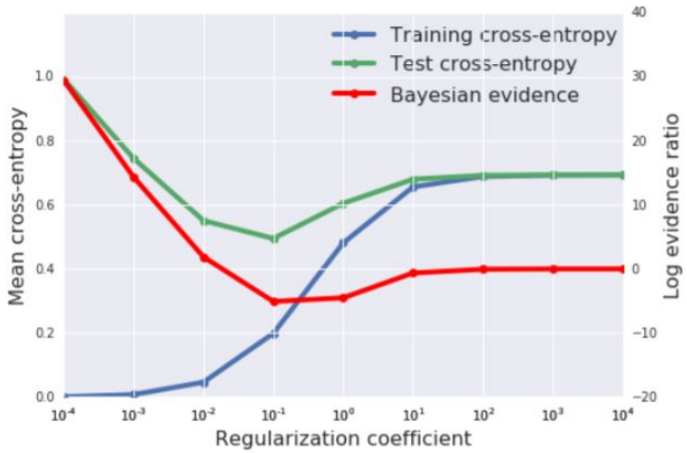


- Exactly as observed by Zhang et al., weakly regularized logistic regression generalizes well on informative labels but memorizes random labels.

Bayesian Evidence



(a)



(b)

- Bayesian evidence has successfully explained the generalization of their logistic regression.

Outline

Noise in training Deep Learning

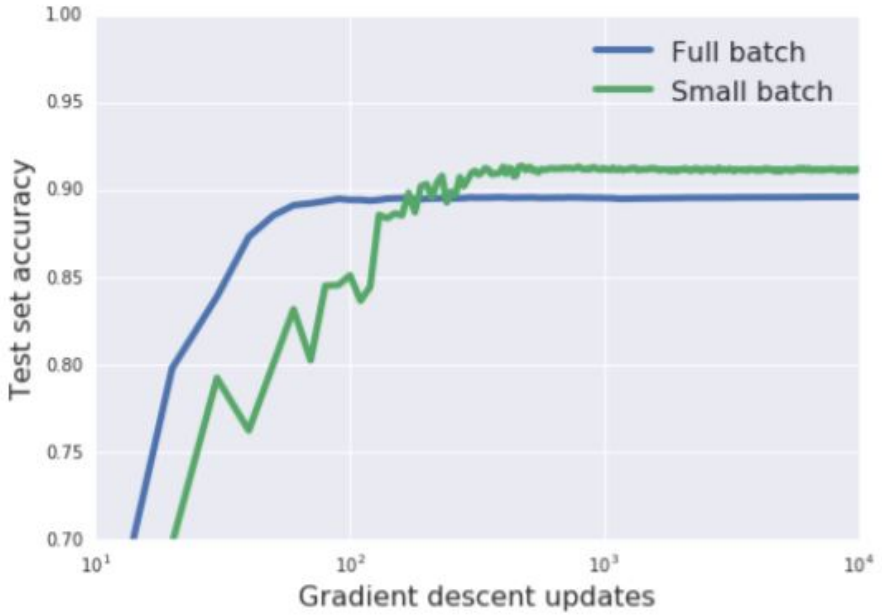
- Generalization is strongly correlated with the Bayesian evidence, which is a weighted combination of the depth of a minimum (the cost function) and its breadth (the Occam factor).
- Small batch training introduces noise to the gradients, and this noise drives the SGD away from sharp minima.

Experiment Setup

- MNIST (randomly selected 1000 images, to compare full-batch vs small batch)
- SGD with learning rate $= 1$ and momentum $= 0.9$ coefficient.
- Purpose is to explore the simplest possible model which shows a generalization gap between small and large batch training.

Generalization Gap between Small Batch vs Full Batch

- Small Batch = 30 and Full Batch = 1000



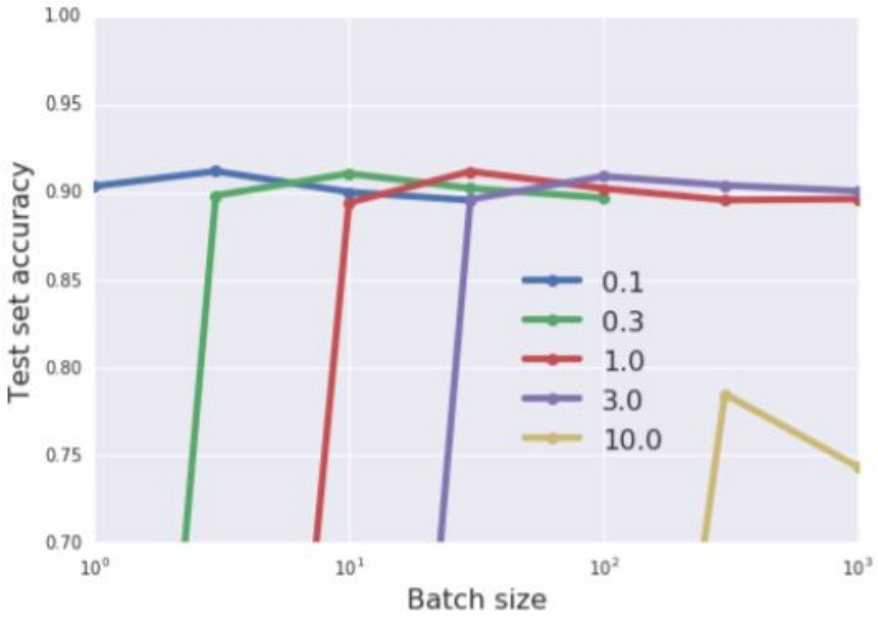
(a)

(b)

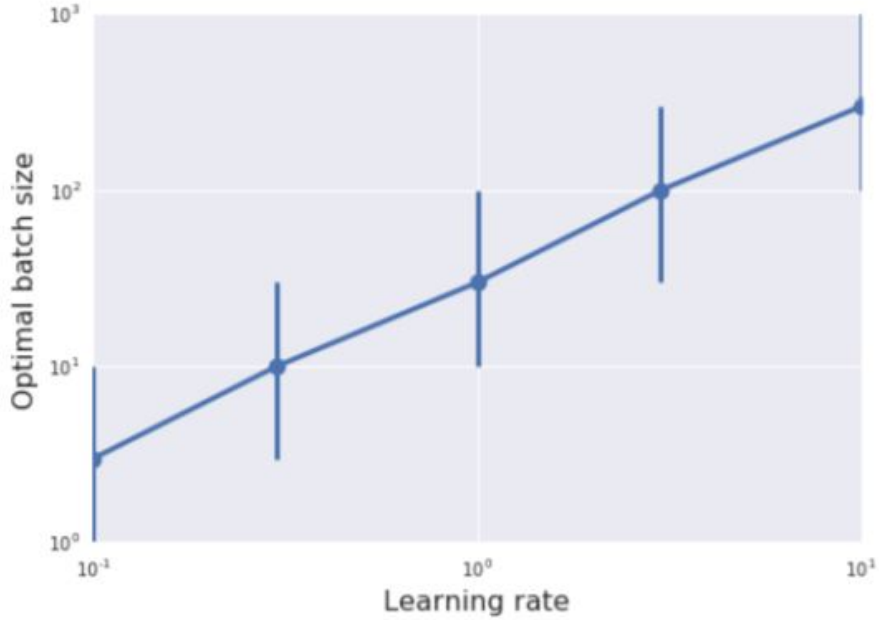
Stochastic Differential Equation of SGD

- Random fluctuation $g = \epsilon\left(\frac{N}{B} - 1\right) \approx \epsilon\frac{N}{B}$
- An optimal batch size emerges when the underlying scale of random fluctuations is also optimal.
- $B_{opt} \propto \epsilon$
- $B_{opt} \propto N$

Batch Size and Learning Rate

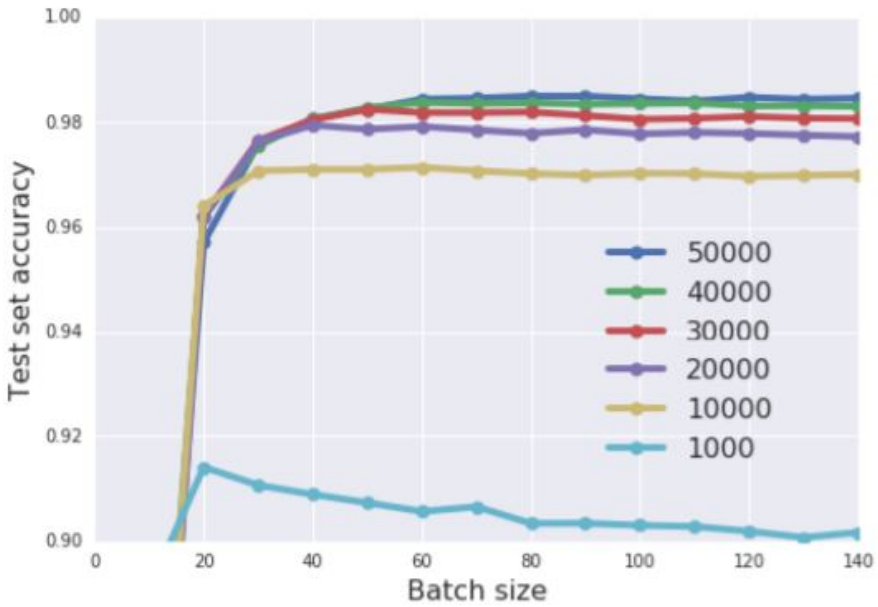


(a)

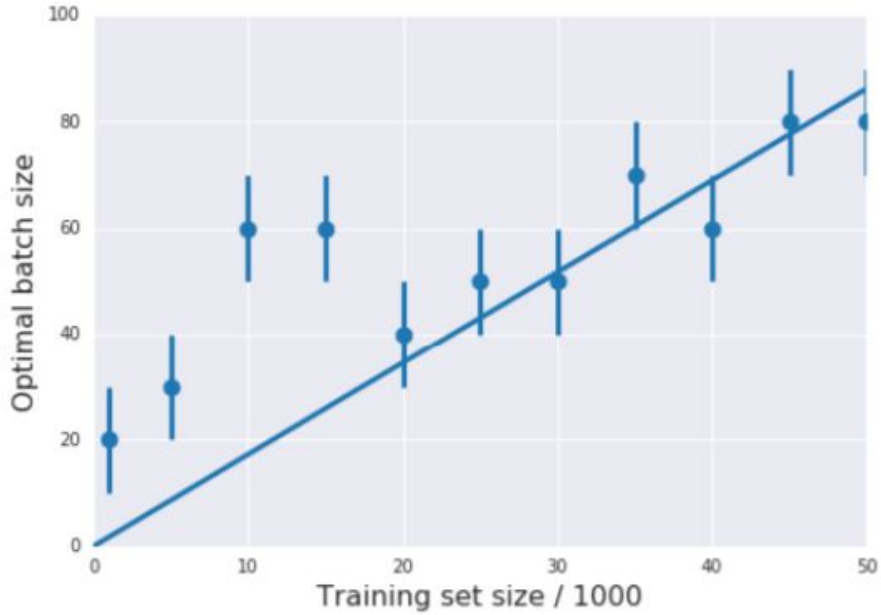


(b)

Batch Size and Training Set Size



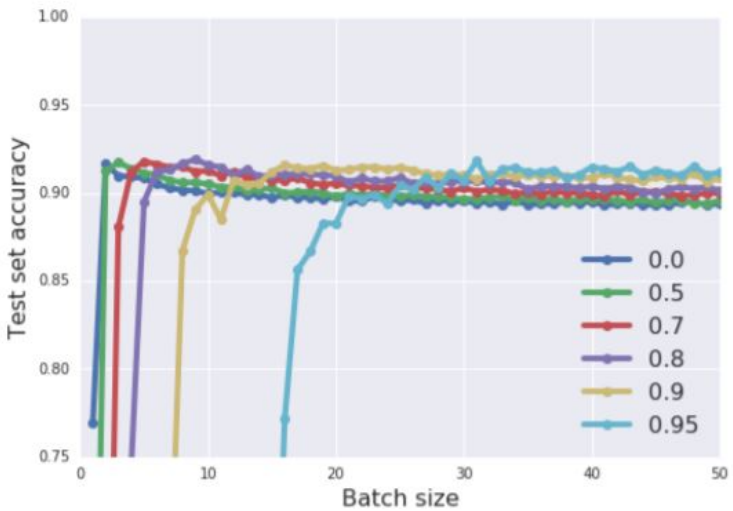
(a)



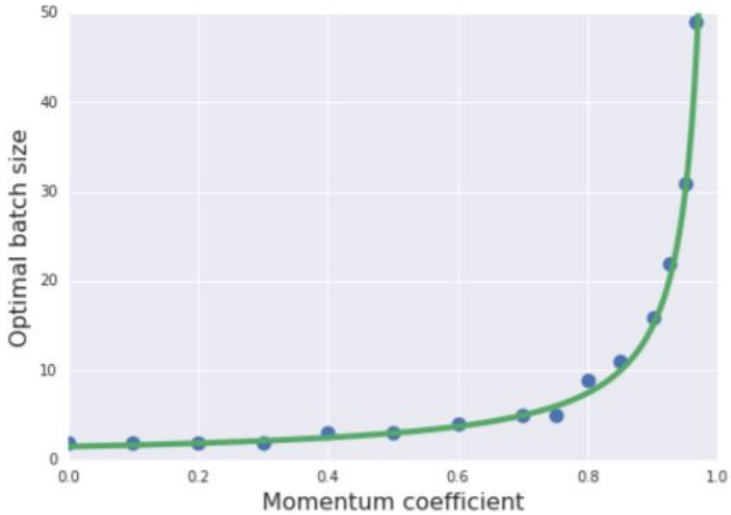
(b)

Batch Size and Momentum Coefficient

- $g \approx \frac{\epsilon N}{B(1-m)}$
- $B_{opt} \propto \frac{1}{(1-m)}$
- Constant $\epsilon = 1$ learning rate, until 10000 gradient updates



(a)



(b)

How to Tune Batch Size

1. Set the learning rate to 0.1 and the momentum coefficient to 0.9. Run experiments at a range of batch sizes on a logarithmic scale, and identify the optimal batch size which maximizes the validation set accuracy.
2. Repeatedly increase the batch size by a factor of 3, while scaling the learning rate $\epsilon \propto B$, until the validation set accuracy starts to fall. Then repeatedly increase the batch size by a factor of 3, while scaling the momentum coefficient $(1 - m) \propto 1/B$, until either the validation set accuracy falls or the batch size reaches the limits of your hardware.
3. Having identified the final learning rate and momentum parameter, retune the batch size on a linear scale in the local neighborhood of the current batch size.

Discussion

- Like deep neural networks, linear models which generalize well on informative labels can memorize random labels of the same inputs. They are explained by evaluating the Bayesian evidence.
- They show that there is an optimum batch size which maximizes the test set accuracy.
- $B_{opt} \propto \frac{\epsilon N}{B(1-m)}$

References

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. (2017). Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*.

Kawaguchi, K., Kaelbling, L. P., and Bengio, Y. (2017). Generalization in deep learning. *arXiv preprint arXiv:1710.05468*.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

Smith, S. L. and Le, Q. V. (2018). A bayesian perspective on generalization and stochastic gradient descent. In *International Conference on Learning Representations*.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.