**CSC2547: Topics in Statistical Learning Theory** Winter 2019

# Project Report: Investigating Generalization in Neural Networks

*Hitarth Choubisa* *Collaborators:*

## 1. Overview:

Despite the fact that Neural Networks (NNs) often have parameters much more in number than the number of datapoints, we find that these NNs are able to perform well on the testing dataset - this is called generalization. It is a question of wide interest to figure out what makes some models generalize better than the others. In this report, I wish to explore generalization in NNs, which method is likely to improve the ability to generalize, learning capacity of NNs and if the ability of generalization is specific to NNs alone.

## 2. Generalization & methods:

Learning the patterns is what gives NNs the ability to generalize on test dataset. However, a recent study [3] questions this understanding of NNs. They took candidate architectures and trained them both on the true data and a copy of the data in which true labels were replaced by random labels. In second case, there is no longer any relationship between the inputs and the labels. However contrary to the expectation, NNs were able to easily fit the random labels as well; this implies that effective capacity of NNs is sufficient for memorizing the entire dataset. Even training time increases only by a constant factor in case of randomized labels as compared to original dataset. This makes us wonder if we can quantify expressive power of a neural network of which the proof is in [3]:

**Theorem 1.** *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.*

Traditional complexity measures, in particular Rademacher complexity doesn't seem to give an insightful bound on generalization errors either. Rademacher complexity is defined as:

$$\mathcal{R}_n(\mathcal{H}) = \mathbb{E}_\sigma[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h(x_i)]$$

where $\mathcal{H}$ is the hypothesis class on a dataset $\{x_1, ..., x_n\}$ and $\{\sigma_1, .., \sigma_n\} \in \{+1/-1\}$ are iid uniform random variables. Since the randomization tests indicate that many NNs fit the training set with random labels, we expect $\mathcal{R}_n(\mathcal{H}) \approx 1$. This is, of course, a trivial upper bound on the Rademacher complexity and doesn't lead to useful generalization bounds.
Regularization is the standard tool to mitigate overfitting in NNs. Idea is that using regularization, we can confine the hypothesis space to a manageable complexity; therby reducing the Rademacher Complexity and obtaining better bounds on generalization errors. A few common methods of explicit regularization include dropout, weights decay and data augmentation. Experiments conducted in [3] indicate that despite the presence of the above-mentioned regularizers, both Inception & MLPs still fit the random training dataset. A separate experiment to compare results of the networks on true data by turning the regularizers ON and OFF suggested that the results don't vary too much.

The observations clearly indicate that we need a better tool for understanding generalization and a tool that is likely to enable the users to ensure generalization. Thus, comes the concept of Bayesian Evidence! Specifically, I will explore the Bayesian Approach (inspiration from [2]) which will enable us to explain these observations, in simple cases enable us to choose a model that generalizes well and will also explore how SGD tend to find minimas which generalize well.

## 3. Bayesian Approach

In this section, we will try to explain the observation of generalization and memorization using *Bayesian evidence*. The Bayesian way of comparing two models $M_1$ and $M_2$ is to observe the following ratio,

$$\frac{P(M_1|y,x)}{P(M_2|y,x)} = \frac{P(y|x;M_1)\cdot P(M_1)}{P(y|x;M_2)\cdot P(M_2)}$$

The second factor on the right is the prior ratio which we usually set to 1 to avoid unnecessary subjectivity.

The first factor is of crucial importance and is called *evidence ratio* which controls how much the training data changes our prior beliefs,

$$P(y|x;M) = \int dw P(y|w,x;M)P(w;M) = \sqrt{\frac{\lambda}{2\pi}} \int e^{-C(w;M)} dw$$

where w are the model parameters and

$$P(y|w,x;M) \cdot P(w;M) = \prod_i P(y_i|w,x_i;M) \cdot \sqrt{\frac{\lambda}{2\pi}} e^{-\lambda w^2/2} = e^{-H(w;M)} \cdot \sqrt{\frac{\lambda}{2\pi}} e^{-\lambda w^2/2} \propto \sqrt{\frac{\lambda}{2\pi}} e^{-C(w;M)}$$

where $C(w;M) = H(w;M) + \lambda w^2/2 = -\sum_i ln(P(y_i|w,x_i;M)) + \lambda w^2/2$.
We call $C(w;M)$ as cost function or a L2 regularized cross entropy. Also, since evidence is calculated by integrating over the model parameters, it is invariant to model parameterization.

Till now we have considered model with a single parameter. But in realistic scenario, we have mulitple parameters and in that case, we can use Taylor expansion to get the following: Let $w_0$ be the point of minima of the cost function. Using Taylor expansion of the cost function $C(w;M)$ around $w_0$, we have,

$$P(y|x;M) \approx \lambda^{p/2} e^{-C(w_0)} / |\nabla\nabla C(w)|_{w_0}^{1/2}$$

where $|\nabla\nabla C(w)|$ is the determinant of the Hessian and p denotes the number of model parameters. The term $\lambda^{p/2}/|\nabla\nabla C(w)|_{w_0}^{1/2}$ is also called *Occam factor*. Since the determinant of the Hessian is simply product of its eigenvalues($\lambda_i$'s), the above expression simplified to,

$$P(y|x;M) \approx exp\left\{ -C(w_0) - \frac{1}{2} \sum_{i=1}^{p} ln(\lambda_i/\lambda) \right\}$$

We compare it against a model which assumes the labels are assigned randomly; evidence for such a model is independent of w and is give by, $P(y|x;NULL) = (1/n)^N = e^{-Nln(n)}$ where n denotes the number of model classes and N the number of training labels. The evidence ratio then becomes,

$$\frac{P(y|x;M)}{P(y|x;NULL)} = e^{-E(w_0)}$$

where $E(w_0) = C(w_0) + (1/2)\sum_i ln(\lambda_i/\lambda) - Nln(n)$, where $E(w_0)$ is also called log-evidence ratio in favor of NULL model. We would prefer model M only if $E(w_0) < 0$. The evidence, as derived above having a curvature factor in it, also supports the intuition that broad minima generalize better than sharp minima since the curvature term (determinant of Hessian) will be smaller for broad minima.

[2] conducted a series of experiments to show how evidence ratio indeed indicates the level of generalization. Since, the neural networks' parameters are very large in number, it becomes close to impossible to track its Hessian matrix. So, instead they targeted Logistic Regression for verifying the relationship between generalization and *Bayesian evidence* for a couple of reasons: first, Logistic regression behaves similarly to neural networks in terms of memorizing and learning behavior; secondly, the number of parameters in logistic regression is tractable. Their experiment results in Figure 1 show a clear correlation between Test cross-entropy and log evidence ratio. Having a low value of $E(w_0)$ (in case of informative labels) correlates with low values of test error (better generalization) and a large positive value of evidence correlates with poor generalization (in case of random labels). In both the cases, however, the training error is low indicating that in case (a), we have memorizing and in case (b), we achieve learning. Thus, the Bayesian evidence approach described here explains the observations made by [3] as to when the model generalizes and when it memorizes.

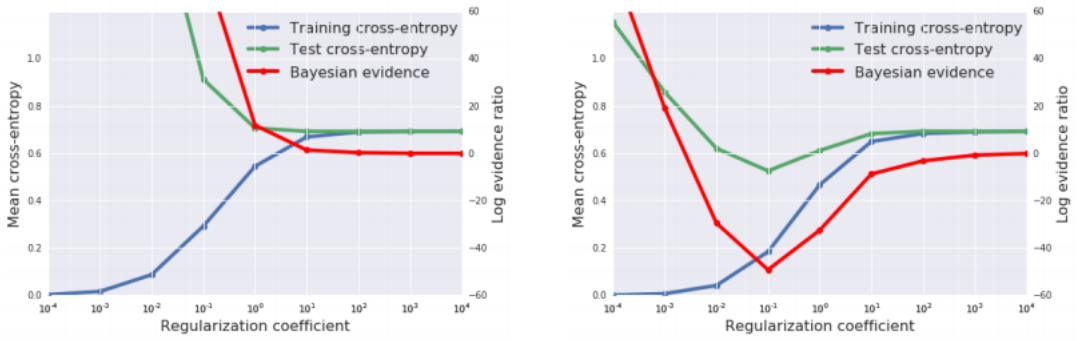


Figure 1: Figure (a) represent the behavior in case of randomized labels. Figure (b) represents the behavior in case of informative labels

# 4. Noise Level and Evidence

Let's suppose that the log term of the eigenvalues in the *evidence* is dominated by the maximum eigenvalue of the Hessian. We would expect models to not generalize well in case of noisy inputs. Thus, the two factors - amount of noise and the maximum eigenvalue of Hessian should be correlated. Thus, this little analysis will not only enable us to verify the claims but also gives us a way to identify the noise levels in the input. It has been indeed been shown by [1] that maximum eigenvalue of the Hessian matrix increases as noise levels increase. Thus, using *Bayesian evidence* as a tool, we can predict about the generalization nature of the model.

# 5. Stochastic Gradient Descent(SGD) and Bayesian Evidence

The "Occam factor" $(\lambda^{p/2}/|\nabla\nabla C(w)|_{w_0}^{1/2})$ in the Bayesian evidence penalizes sharp minima and hence, model with sharp minima is unlikely to generalize well. Mini-batch noise in SGD drives SGD to find a broader minima resulting in a better evidence. [2] modelled the SGD via stochastic differential equation which was then used to obtain SGD noise scale(g) as, $g \approx \frac{\epsilon N}{B}$ where N is the number of data-samples, B is the batch size and $\epsilon$ is the learning rate. Since the gradient drives the SGD towards deep minima, while noise drives SGD towards broad minima, we expect the test set performance to show peak at an optimal batch size, which balances these competing contributions to the evidence. Thus, the above rule allows us to increase the learning rate, by varying the batch size and at the same time, keeping the generalization error the same. The fact that $B_{opt} \propto N$ (fixing the noise level which is optimal) also enables to us put the machine learning models in production; progressively increase the batch size as the training set size increases.

# 6. Conclusion

In this exploratory report, I have tried to demystify the generalization behavior of learning models. Using Bayesian evidence as our tool, we can easily figure out when is our model learning and when it is memorizing. We conducted some experiements to show how we can potentially use evidence as a measure of level of noise as well. Towards the end, we saw how SGD tends to find broader minimas which generalize well.

# References

[1] David Krueger, Nicolas Ballas, and Stainslaw Jastrzebski et al. Deep nets don't learn via memorization. 2017.

[2] Samuel L. Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. 2018.

[3] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. 2017.