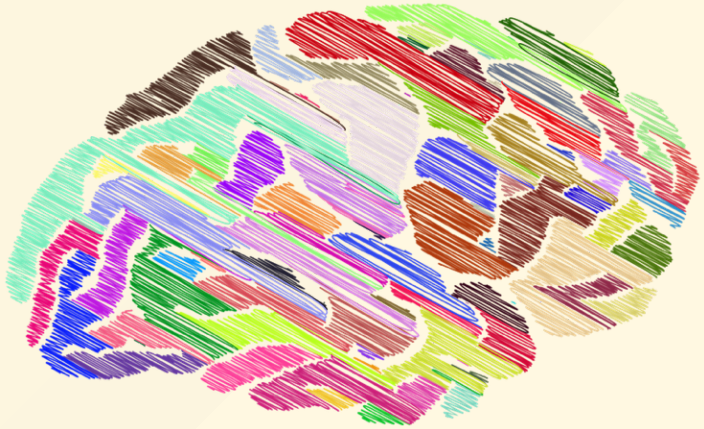
PAC-Bayesian Theory



Jeremy Bernstein
bernstein@caltech.edu

Agenda for today

1. Recap on VC theory
2. Bayesian data modelling
3. PAC-Bayes theorem
4. PAC-Bayes for NNs
5. PAC-Bayesian model comparison

VC theory

We proved that:

$$\mathbb{P}\left[\text{for any } f \in F \;\middle|\; \text{train error} - \text{test error}\right| \geq \varepsilon\Big] \leq \frac{2\,\mathbb{E}_X[N(F,X)]}{\exp\left(m\frac{\varepsilon^2}{2}\right)}$$

In words: the chance that any classifier gets substantially different error on randomly drawn train and test sets (each containing m datapoints) depends on a tradeoff between m and the average number of distinct labellings the function class F can apply to $2m$ datapoints X drawn from the data distribution.

Basic mechanism of VC theory

① If a classifier $f \in F$ gets very different train and test error, then from the point of view of that classifier something very imbalanced and unlikely happened in the data sampling.

② But if there are lots and lots of classifiers, then it can become quite likely that something unlikely happens to one of them.

③ VC theory's advice is thus to limit the number of classifiers, as measured by realisable dichotomies on a finite sample.

Pitfalls of VC theory

- neural nets still generalise even when they are capable of realising all 2^m dichotomies of m points.

- asking for no classifier $f \in F$ to overfit is overkill. We only care about the classifier found by the optimiser.

- the VC bound we derived is blind to the complexity of the training labels —— the complexity measure (RHS) only depends on the inputs.

Agenda for today

1. Recap on VC theory
2. Bayesian data modelling
3. PAC-Bayes theorem
4. PAC-Bayes for NNs
5. PAC-Bayesian model comparison

Bayes' theorem

A Bayesian will tell you that all a data scientist needs is:

$$P(\text{model} \mid \text{data}) = \frac{P(\text{data} \mid \text{model})\, P(\text{model})}{P(\text{data})}$$

(likelihood) (prior)

(posterior)

(evidence)

... but there's a jarring contrast to deep learning — we train a single model, we don't find a posterior distribution over models. Let's ignore this for a moment.

What is the Bayesian evidence?

The denominator of Bayes' thm will be very important to us.

$$\underbrace{P(\text{model}\,|\,\text{data})}_{\text{(posterior)}} = \frac{\overbrace{P(\text{data}\,|\,\text{model})}^{\text{(likelihood)}}\ \overbrace{P(\text{model})}^{\text{(prior)}}}{\underbrace{P(\text{data})}_{\text{(evidence)}}}$$

"evidence" for the model class:

$$P(\text{data}) = \sum_{\text{all models}} P(\text{data}\,|\,\text{model})\ P(\text{model}).$$

It's the probability of randomly sampling a model from the prior and finding that it fits the data.

Evidence for a NN

For a classification problem, a natural likelihood function is

$$P(\text{data} \mid \text{model}) = \begin{cases} 1 & \text{if model fits all training labels} \\ 0 & \text{otherwise.} \end{cases}$$

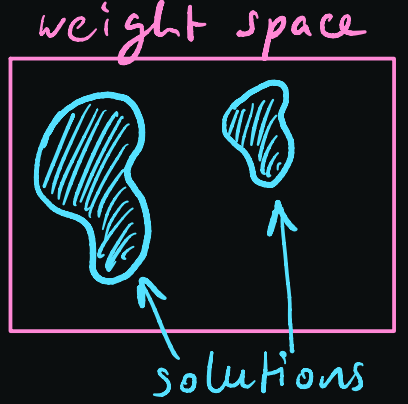
The evidence then simplifies to

$$P(\text{data}) = \sum_{\substack{\text{models that fit} \\ \text{all labels}}} P(\text{model}).$$

total prior probability of models that get 100% training acc.

Think of this as:

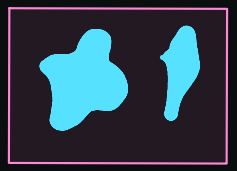
$$\text{evidence} = \frac{\text{volume of solutions}}{\text{volume of weight space}}.$$



Properties of the evidence

The evidence has some nice properties:

① For a fixed NN architecture, we might expect it to to reflect the "hardness" of the dataset.



easy data
⇒ more solutions

hard data
⇒ fewer solutions

② For a fixed dataset, we might expect it to reflect the suitability of an NN architecture.

good architecture
⇒ more solutions

bad architecture
⇒ fewer solutions

Connection to generalisation?

③ The evidence has an information theoretic interpretation:

draws from the prior to find a solution $\approx \frac{1}{\text{evidence}}$

$\Rightarrow$ a <u>random</u> solution can be described by the random seed and $\approx \log \frac{1}{\text{evidence}}$ additional bits.

So it seems like a good candidate for a fairly general purpose complexity measure.

David MacKay wrote in 1995:

"Empirically, the correlation between the evidence and generalization error is often good ... but a theoretical connection between the two is not yet established."

Agenda for today

1. Recap on VC theory
2. Bayesian data modelling
3. PAC-Bayes theorem
4. PAC-Bayes for NNs
5. PAC-Bayesian model comparison

Basic mechanism of PAC-Bayes

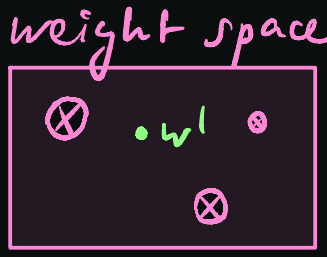
Draw a fresh iid train and test set.

Next, draw a random weight vector w from the weight space.

Q1: would you expect w to have similar train and test error?

A1: Yes!

Next, I look at the training data, and decide to exclude 5% of weight vectors:



You draw a random weight vector w' from the 95% that remain.

Q2: would you expect w' to have similar train and test error?

A2: Yes! Excluding 5% of weight vectors can't change much.

a simplified version.

Statement of the result

PAC-Bayes: the fewer the weight vectors that are excluded by the training data, the better a random solution will generalise.

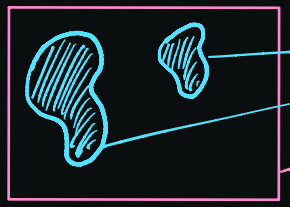
More formally, for a proportion $1-\delta$ of iid training sets,

$$\underset{\text{solutions}}{\text{averaged over}}\left[\text{population error}\right] \leq \frac{\log\left(\frac{1}{\text{evidence}}\right) + \log\left(\frac{2m}{\delta}\right)}{m-1}.$$

where: m = # training points

a solution is a weight vector that gets 100% train acc

$$\text{evidence} = \frac{\text{volume of solutions}}{\text{volume of weight space}}$$



Deriving the result (sketch)

On HW4 we show: $$\mathbb{E}_{w\sim Q}\, h(w) \leq \frac{KL(Q\|P) + \ln \mathbb{E}_{w\sim P} e^{\beta h(w)}}{\beta}.$$

- set $\beta = m-1$ (# train points -1)
- set P to the prior on weight space
- set Q to the prior restricted to the solutions

$$\Rightarrow KL(Q\|P) = \log \frac{1}{P(\text{all solutions})} = \log \frac{1}{\text{evidence}}$$

- set $h(w)$ to the generalisation error of weight vector w
  ↪ for $w \sim Q$, w gets 0% train error $\Rightarrow$ $h(w)$ measures population error.

<u>Critical step</u>: since P does not depend on the train set, then we would expect $\ln \mathbb{E}_{w\sim P} e^{\beta h(w)}$ to be small. In fact, we can show that for a fraction $(1-\delta)$ of iid train sets, $\ln \mathbb{E}_{w\sim P} e^{\beta h(w)} \leq \ln \frac{2m}{\delta}$.

Proving this requires doing some statistics. See references on website.

Agenda for today

1. Recap on VC theory
2. Bayesian data modelling
3. PAC-Bayes theorem
4. PAC-Bayes for NNs
5. PAC-Bayesian model comparison

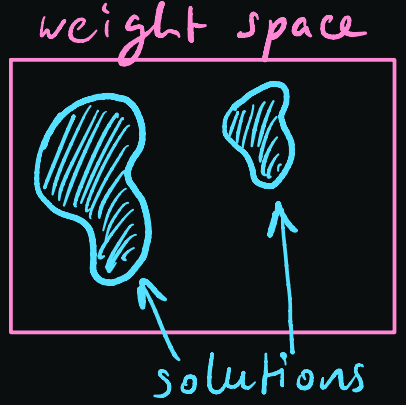
Evidence for a neural net

PAC – Bayes: for a fraction $(1-\delta)$ of iid train sets,

$$\underset{\text{over solutions}}{\text{averaged}}\left[\text{population error}\right] \leq \frac{\log\left(\frac{1}{\text{evidence}}\right) + \log\left(\frac{2m}{\delta}\right)}{m-1}.$$

Recall:

$$\text{evidence} = \frac{\text{volume of solutions}}{\text{volume of weight space}}.$$



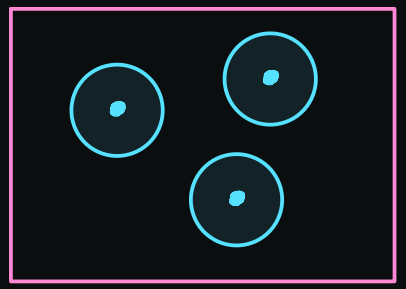
How can we compute the evidence for a neural net?

Attempt #1: sharpness/flatness

IDEA: if a solution is "flatter" then the evidence will be larger.

Think of "flatness" as the amount we can perturb a solution w before we induce a misclassification.

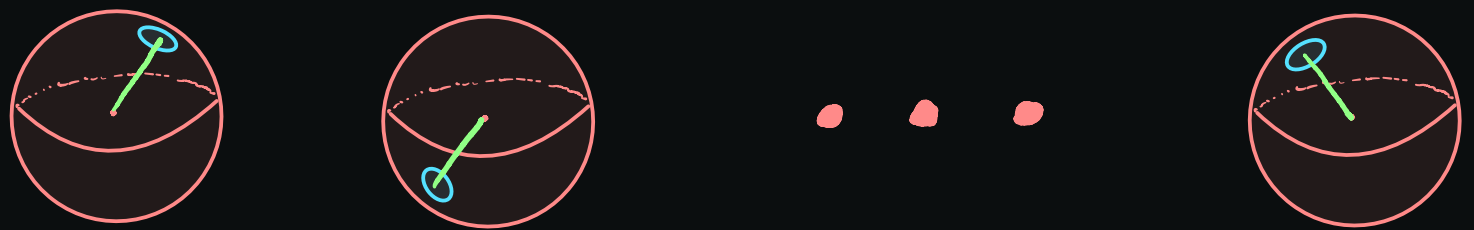
Let's model the solution space as being comprised of K such non-intersecting solutions:



Attempt #1: sharpness/flatness

To work this idea through, let's consider a neural net composed of n neurons with fan-in d, each with a weight vector w satisfying $\sum_{i=1}^{d} w_i = 0$, $\sum_{i=1}^{d} w_i^2 = 1$.

Geometrically, a solution weight vector is a point on the Cartesian product of n $(d-2)$-spheres:



Define "flatness" to be the max angle α we can arbitrarily rotate every neuron through without causing a misclassification.

Attempt #1: sharpness/flatness

Letting the prior be uniform over the n hyperspheres, we get:

$$\text{evidence} = \frac{\text{volume of solutions}}{\text{volume of weight space}} = K \times \left[\frac{\text{area of } \bigcirc}{\text{area of } \bigcirc}\right]^n$$

Geometry result: the proportion of a $(d-2)$ sphere's surface area taken up by a cap of angle α satisfies:

$$\frac{\text{area of } \bigcirc}{\text{area of } \bigcirc} \geq \frac{1}{2} \sin^{(d-2)} \frac{\alpha}{2}.$$

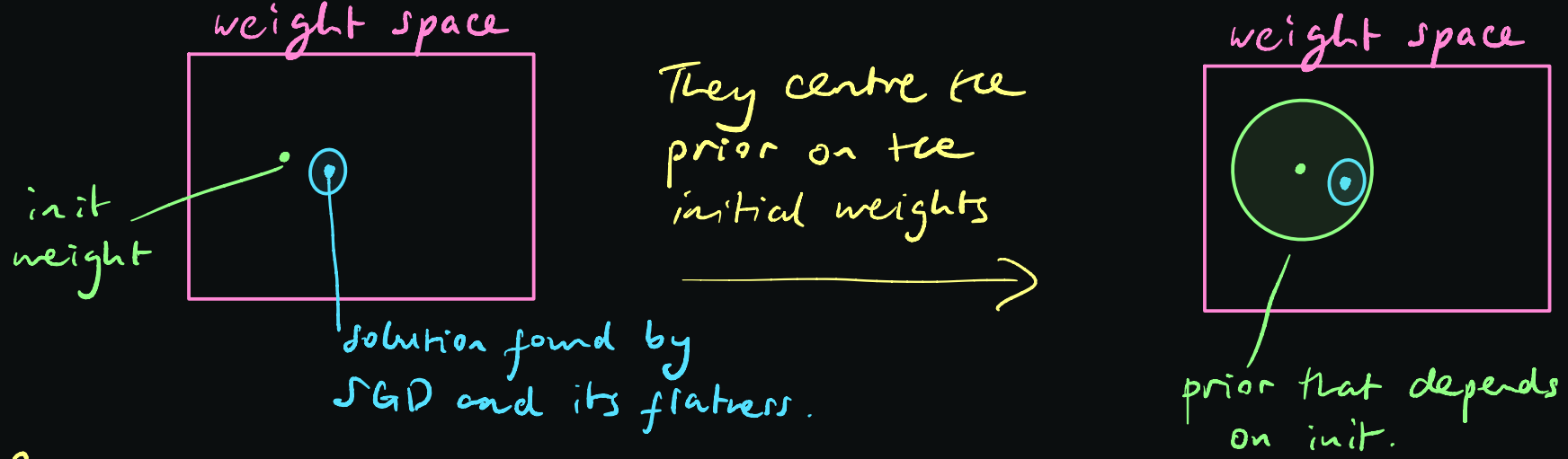
$$\Rightarrow \boxed{\text{evidence} \geq \frac{K}{2^n} \sin^{n(d-2)} \frac{\alpha}{2}}$$

This depends on # degrees of freedom $n(d-2)$, but also on the number of solutions K.

Attempt #2: breaking symmetries

Problem with the previous approach: we don't know K.

Dziugaite & Roy overcome this by using a different prior:



Problem: they don't know how large the prior should be.

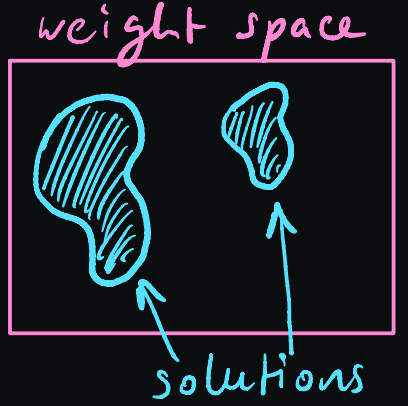
Solution: they derive PAC-Bayes bounds for many priors of varying size, and choose the best one in hindsight.

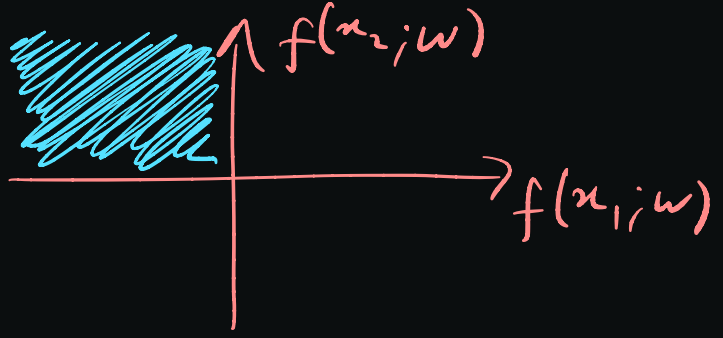
This leads to non-vacuous generalisation bounds for neural nets, although choosing over many priors pays a cost in terms of failure probability.

Attempt #3: infinite width

The structure of the solution manifold in weight space is complicated.

But in function space, it is simple!





For binary classification, it is an orthant.

So for ∞-ly wide NN binary classifiers, by the NNGP correspondence,

$$\text{evidence} = \mathbb{P}_{z \sim \mathcal{N}(0,\Sigma)}\left[\text{sign}(z) = y\right]$$

Σ: NNGP covariance.

y: vector of training labels

Agenda for today

1. Recap on VC theory
2. Bayesian data modelling
3. PAC-Bayes theorem
4. PAC-Bayes for NNs
5. PAC-Bayesian model comparison

Comparing two architectures

What is any of this good for?

Suppose we want to know which of two NN architectures is better on a certain task, a Bayesian would tell us to compare the evidence for each architecture.

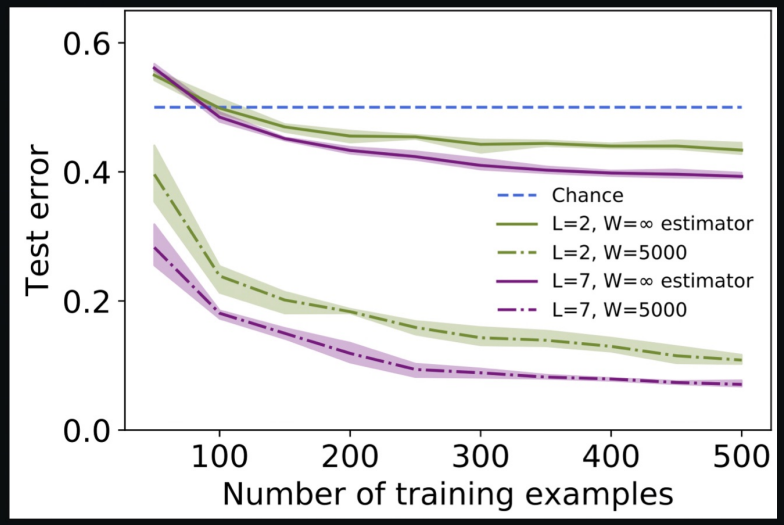
good architecture
⇒ more solutions
⇒ more evidence

bad architecture
⇒ fewer solutions
⇒ less evidence

PAC-Bayes goes further — it says we'd expect random solutions from the architecture with more evidence to generalise better.

An example

Want to choose between a 7-layer MLP and a 2-layer MLP for classifying MNIST digits as even or odd.



comparing the PAC-Bayes upper bound estimated for infinitely wide networks versus the empirical test error at finite width as a fn. of training set size.

Seems to be a viable tool for model comparison!

Summary

- PAC-Bayes relates the expected generalisation error of a random solution to the volume of weight space excluded by the training data.
- The complexity measure is $\log \frac{1}{\text{Bayesian evidence}}$.
- Yields generalisation bounds that are non-vacuous, and can be used to perform model comparison or compare the hardness of different datasets.

Next lecture

Project ideas.