Deep Learning

Lecture 9: Generalisation theory

Chris G. Willcocks

Durham University

Lecture overview

1. **Generalisation theory**

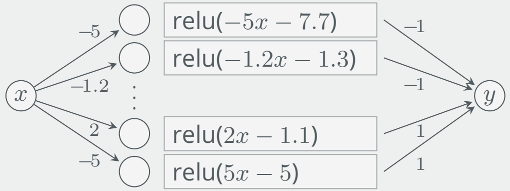
- universal approximation theorem
- fixed width and arbitrary depth?
- empirical risk minimisation
- noise, training set size, and regularisation
- generalisation of unsupervised models
- a cycle that amplifies biases
- no free lunch theorem and Occam's razor
- bias-variance tradeoff on nasty densities
- increasing capacity through double descent

Generalisation theory universal approximation theorem

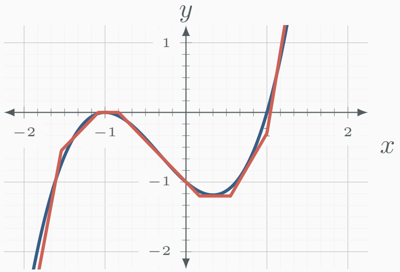
Theorem: universal function approximator

Arbitrary width

A network with a single hidden layer, containing a finite number of neurons, can approximate any continuous function under mild assumptions.



original function $f(x) = x^3 + x^2 - x - 1$



$$f(x) = -r(-5x - 7.7) - r(-1.2x - 1.3) - r(1.2x + 1) + r(1.2x - 0.2) + r(2x - 1.1) + r(5x - 5)$$

Generalisation theory universal approximation theorem

Theorem: universal function approximator

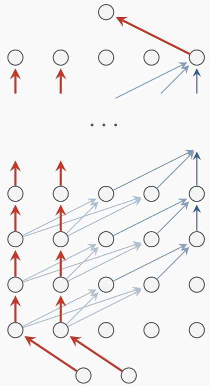
Arbitrary depth (fixed width)

Does the theorem still hold for fixed width and arbitrary depth? **Yes!**

For a network of n inputs and m outputs, [1] show universal approximaton holds true for:

- width $n + m + 2$ for almost any activation function
- width $n + m + 1$ for most activation functions

Short YouTube visual proof ▶



Recap learning the data distribution

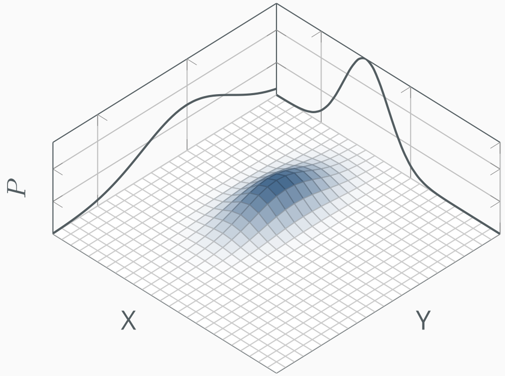
Learning the data distribution

So what is it we want exactly?

- $P(Y|X)$ discriminative model (classification)
- $P(X|Y)$ conditional generative model
- $P(X,Y)$ generative model

We want to learn the probability density function of our data (natures distribution)

The data distribution $P(X,Y)$



Generalisation theory empirical risk minimisation

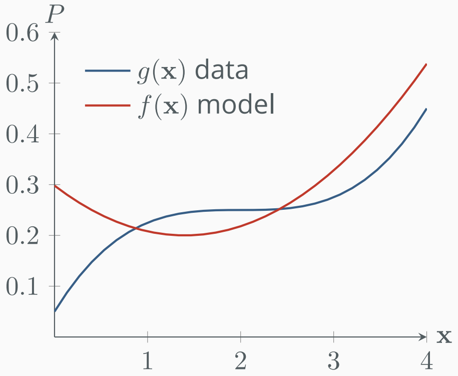
Finding a model

Lets illustrate this in 1D, but try to imagine it in ND. Given the target probability distribution of our data

$$g(\mathbf{x}) = P(X, Y)$$

we want to find (design) a model $f(\mathbf{x}; \theta)$ with parameters θ and optimise θ such that

$$f(\mathbf{x}; \theta) = g(\mathbf{x})$$



Generalisation theory empirical risk minimisation

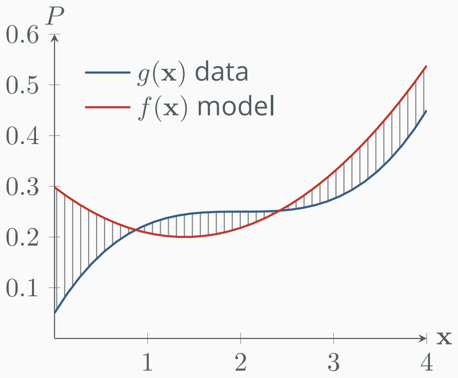
Optimising the model

The total error is therefore the entire area. Modifying the parameters θ will cause the area to change, where we want to find

$$\hat{\theta} = \arg\min_{\theta} \int \mathcal{L}(f(\mathbf{x};\theta), g(\mathbf{x}))\mathrm{d}\mathbf{x}.$$

where $\mathcal{L}$ is a 'loss function', e.g. a 0-1 loss function $\mathcal{L}(\hat{x}, x) = \mathbb{I}(\hat{x} \neq x)$ or a mean squared error loss.

The solution is a function f that has the capacity to exactly represent $g(\mathbf{x})$



Generalisation theory empirical risk minimisation

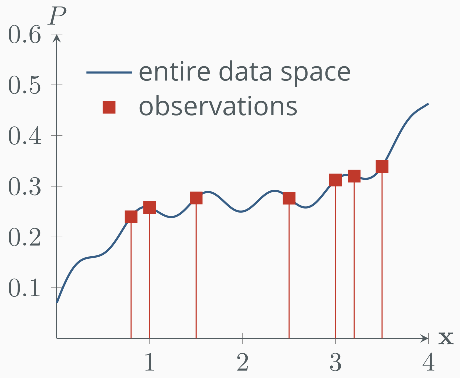
The generalisation problem

However there's a big problem:

In practice, we can't observe all of $g(\mathbf{x})$

This means:

1. We don't know how smooth the function is between the observations
2. Noise can be difficult to interpret
3. Optimisation is highly sensitive to the sampling process



Generalisation theory empirical risk minimisation

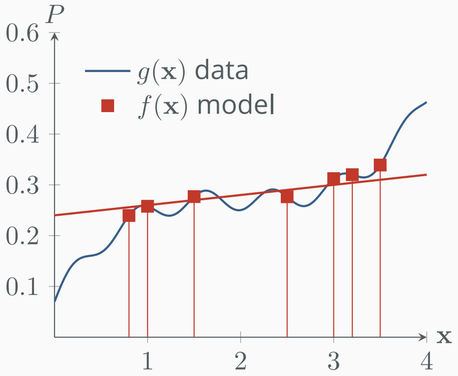
Definition: empirical risk

The expected error (or risk) is the average error over the entire space, which we can't compute:

$$\mathbb{E}[\mathcal{L}(f(\mathbf{x};\theta), g(\mathbf{x}))] = \int \mathcal{L}(f(\mathbf{x};\theta), g(\mathbf{x}))\mathrm{d}\mathbf{x}.$$

Therefore we minimise the **empirical estimate** of the risk as an average over the samples:

$$\mathbb{E}[\mathcal{L}(f(\mathbf{x};\theta), g(\mathbf{x}))] \approx \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x};\theta), g(\mathbf{x})).$$

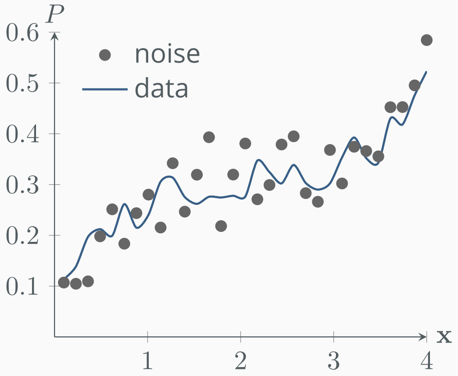


Generalisation theory empirical risk minimisation

Noise and regularisation

Given that the shape of the distribution outside of the observations is unknown, it is easy to overfit to noise, especially when you have limited data.

We can often regularise our function to be invariant to this.

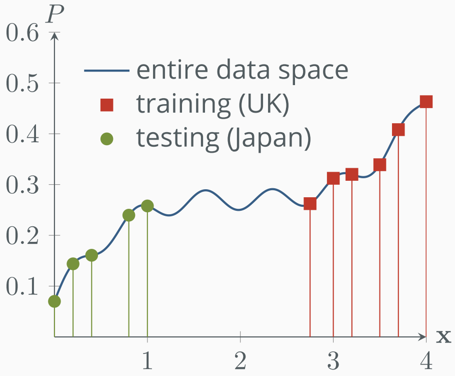


Generalisation theory empirical risk minimisation

Out-of-distribution data

Usually the training dataset collection process draws samples from the data space in a way that is not **independent and identically distributed** (abbreviated i.i.d.) to the expected testing (operational) conditions of the model.

Sampling data in a way that is representative of the task/testing/operational distribution is extraordinarily difficult to do properly. It is often a worthwhile investment though!



Generalisation theory generalisation of unsupervised models

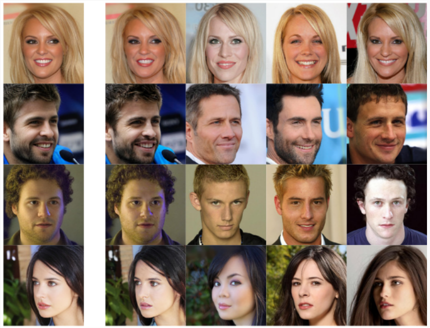
Similarity measures

How can we evaluate the generalisation of unsupervised generative models?

Deep pre-trained network activations work surprisingly well (e.g. FID, LPIPS [2] for perceptual similarity between images):

```
import lpips
distance_fn = lpips.LPIPS(net="alex")
```

Example code usage



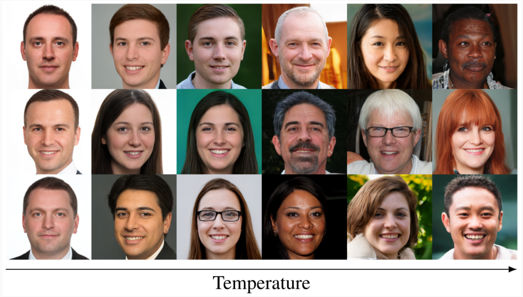
Model samples (leftmost column) and nearest neighbours in training set (LPIPS distance [2]) increasing to right. From [3]

Generalisation theory a cycle that amplifies biases

Bias amplification

Judging models solely based on pre-trained measures such as FID can amplify the inherited biases.

FID (currently how the majority of image-based generative model papers are ranked) can favour less diversity (with focus shifted towards the distribution mode).



sampling away from the mode of the data distribution

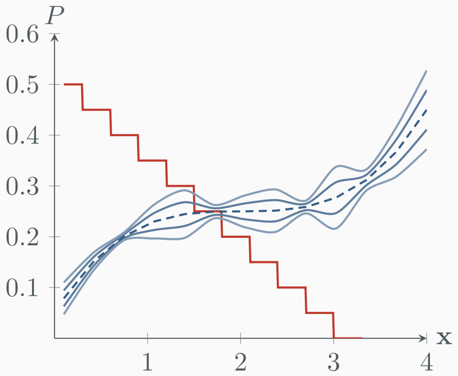
Generalisation theory no free lunch theorem and Occam’s razor

Definition: no free lunch theorem

We can use methods such as cross validation to empirically choose the best method for our particular problem. However, there is no universally best model — this is sometimes called the no free lunch theorem [4].

Definition: Occam’s razor

‘Prefer the simplest hypothesis $\mathcal{H}$ that fits the data.’ In the case of deep learning, this implies the smoothest function that fits the data.



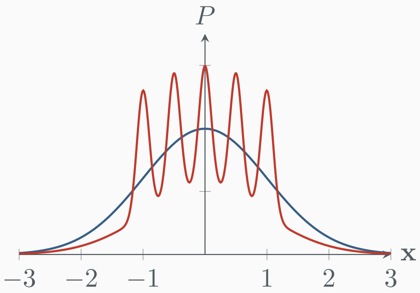
Generalisation theory bias-variance tradeoff and nasty densities

Example: problematic densities

In a more complex setting, we may have a density such as the ‘Bart Simpson’ density, as Nando de Freitas likes to call it

$$p(x) = \frac{1}{2}\phi(x; 0, 1) + \frac{1}{10}\sum_{j=0}^{4}\phi(x; (j/2) - 1, 1/10)$$

where ϕ is the normal density with mean μ and standard deviation σ. This density cannot be sufficiently estimated with a normal distribution, as the result is over-smoothed (blue).

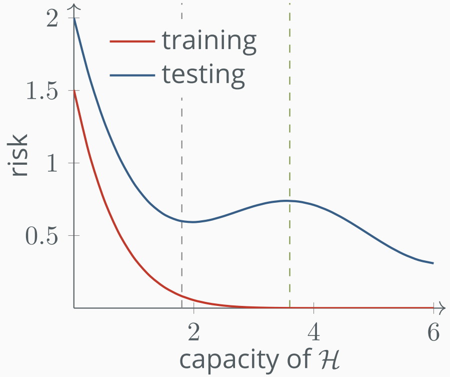


Generalisation theory increasing capacity to double descent

Definition: double descent

Traditionally, we know that increasing the parameters lowers the bias (fitting), but the variance (test risk) will eventually reach a 'sweet spot' (first dashed line) and start to increase again.

The full story has a double descent curve [5], as higher capacity functions past the interpolation threshold (second dashed line) lead again to smoother fitting (Occam's razor).



Take Away Points

Summary

In summary:

- deep learning isn't magic
- there's hundreds of formal results [6]
- how the data was sampled is important
- how the model is sampled is important
- the model capacity is important
- the architecture itself is often less important
- can we use data $P(X|Y)$ to focus on parts of the distribution at inference time?

There's several theorems that haven't been covered today. Topics have been chosen based on relevance in practice.

References I

[1] Patrick Kidger and Terry Lyons. “Universal approximation with deep narrow networks”. In: Conference on Learning Theory. 2020, pp. 2306–2327.

[2] Richard Zhang et al. “The unreasonable effectiveness of deep features as a perceptual metric”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, pp. 586–595.

[3] Patrick Esser, Robin Rombach, and Bjorn Ommer. “Taming transformers for high-resolution image synthesis”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 12873–12883.

[4] Kevin P Murphy. Machine learning: a probabilistic perspective. MIT press, 2012.

[5] Mikhail Belkin et al. “Reconciling modern machine-learning practice and the classical bias–variance trade-off”. In: Proceedings of the National Academy of Sciences 116.32 (2019), pp. 15849–15854.

References II

[6] Ovidiu Calin. Deep learning architectures: a mathematical approach. Springer, 2020.