Generalization & Cross-Validation

Mathematics for Machine Learning

Lecturer: Matthew Wicker

Material Covered

Models: Linear models, basis expansion, logistic regression, neural networks, Prob. densities, Bayesian density estimation

Techniques: Least squares estimation, forward AD, reverse AD, computational graphs, gradient descent, convergence, convexity, Lipschitz continuity, Maximum likelihood, maximum a posteriori, Bayesian inference, LOTUS, change of variables, expectation identities, equating coefficients, joint Gaussian, epistemic/aleatoric uncertainty, concentration inequalities

Settings: Regression, Classification, Density Estimation

This lecture: Further concentration, generalization bounds, regularization, cross-validation

Generalization Bounds

Recall that last lecture we started with Markov's inequality:

$$\mathbb{E}[Z] \geq aP(Z > a)$$

We then extended this from the expectation to the variance to get Chebyshev's inequality and used it to prove the WLLN:

$$P(|\hat{Z} - \mu| \geq \epsilon) \leq \frac{\mathbb{V}[\hat{Z}]}{\epsilon^2} = \frac{\sigma^2}{N\epsilon^2}$$

Generalization Bounds

Recall that last lecture we started with Markov's inequality:

$$\mathbb{E}[Z] \geq aP(Z > a)$$

We then extended this from the expectation to the variance to get Chebyshev's inequality and used it to prove the WLLN:

$$P(|\hat{Z} - \mu| \geq \epsilon) \leq \frac{\mathbb{V}[\hat{Z}]}{\epsilon^2} = \frac{\sigma^2}{N\epsilon^2}$$

Can we do any better than this?

Hoeffding's Inequality

Theorem 2.1. ***(Hoeffding's Inequality):*** *Let* $X_1, X_2, \ldots, X_n$ *be independent random variables such that* $a_i \leq X_i \leq b_i$ *almost surely for all* i*, and let* $\bar{X}_n$ *be their average. Then, for any* $\epsilon > 0$,

$$P\left(\left|\bar{X}_n - \frac{1}{n}\sum_{i=1}^{n} E[X_i]\right| \geq \epsilon\right) \leq 2\exp\left(\frac{-2n\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

prob(| mean - samp. mean | ≥ ε)

$e^{-2} > e^{-10}$

∴ want e^{-big} and i want desired

Hoeffding's Inequality

Theorem 2.1. ***(Hoeffding's Inequality):*** *Let $X_1, X_2, \ldots, X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ almost surely for all i, and let $\bar{X}_n$ be their average. Then, for any $\epsilon > 0$,*

$$P\left(\left|\bar{X}_n - \frac{1}{n}\sum_{i=1}^{n} E[X_i]\right| \geq \epsilon\right) \leq 2\exp\left(\frac{-2n\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

Letting ai and bi be equal to zero and one in the above definition we arrive at a much simpler form of the inequality:

$$P\left(\left|\bar{X}_n - E[X]\right| \geq \epsilon\right) \leq 2\exp\left(-2n\epsilon^2\right).$$

$\frac{1}{N}\sum X_i$

Hoeffding's Inequality

Letting ai and bi be equal in the above definition we arrive at a much simpler form of the inequality:

$$P\left(\left|\bar{X}_n - E[X]\right| \geq \epsilon\right) \leq 2\exp\left(-2n\epsilon^2\right).$$

In this form, it is much easier to see that if we call the left-hand term \delta, then we can solve for the sample complexity of any given statistical guarantee.

find min sample size n s.t. sample mean doesn't deviate from actual mean by more than ϵ

when looking at generalisation: $P\left(\left|\hat{Z}_n - E[Z]\right| > \epsilon\right) < \delta \longrightarrow \delta \leq 2e^{-2n\epsilon^2}$

prob (estimator − expected value > error in estimate) < confidence

(prob (sample − actual) log than)

$\log\left[\frac{\delta}{2}\right] \leq -2n\epsilon^2$

$\frac{1}{2\epsilon^2}\log\left(\frac{\delta}{2}\right) \leq -n \rightarrow \frac{1}{2\epsilon^2}\log\left(\frac{\delta}{2}\right) \leq n$

Hoeffding's Inequality

Letting ai and bi be equal in the above definition we arrive at a much simpler form of the inequality:

$$P\left(\left|\bar{X}_n - E[X]\right| \geq \epsilon\right) \leq 2\exp\left(-2n\epsilon^2\right).$$

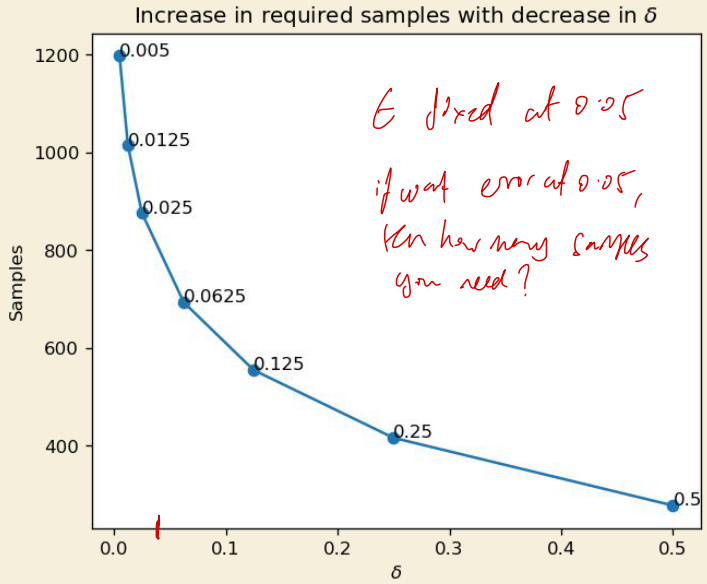
In this form, it is much easier to see that if we call the left-hand term \delta, then we can solve for the sample complexity of any given statistical guarantee.

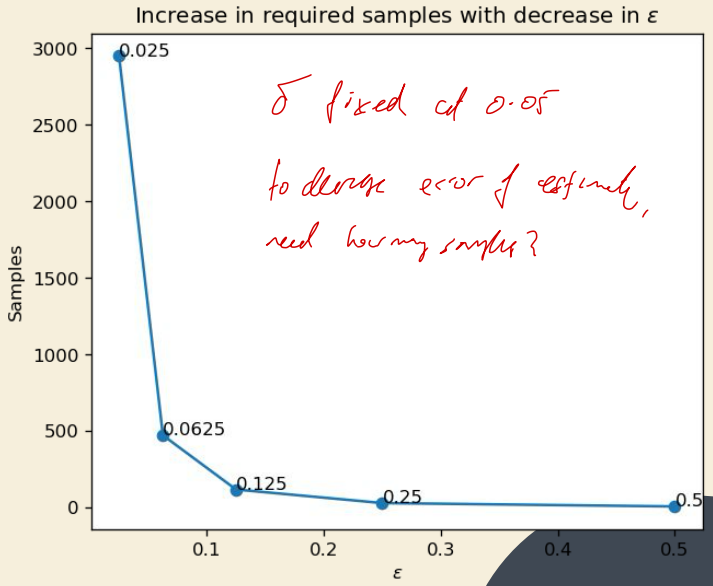
$\delta \leq 2\exp(-2n\epsilon^2)$

$$n \geq \frac{1}{2\epsilon^2}\log\frac{2}{\delta}$$

Hoeffding's Inequality

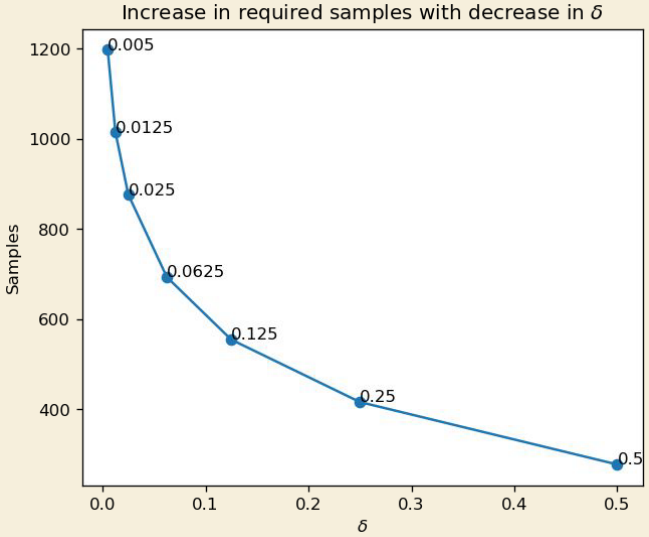
$$n \geq \frac{1}{2\epsilon^2} \log \frac{2}{\delta}$$

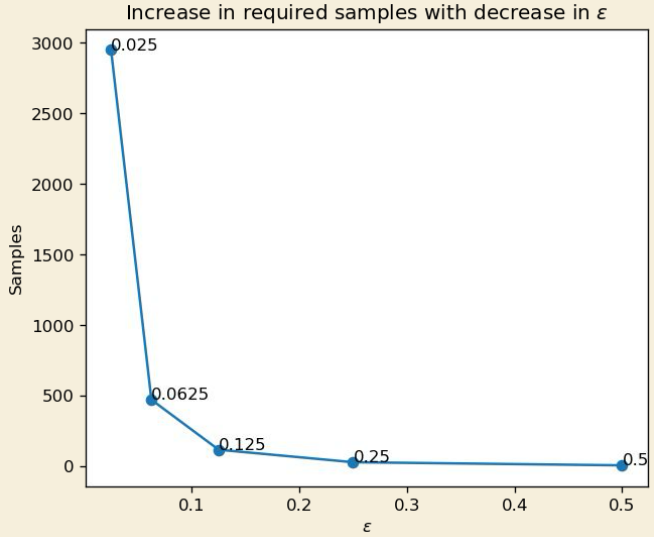




Hoeffding's Inequality

$$n \geq \frac{1}{2\epsilon^2} \log \frac{2}{\delta}$$





Can we do better? Yes!

Idea: Generalization Error Bounds for Algorithms

Can we extend these very useful tools from probability and statistics to further reason about ML specific problems. For example: given two ML model classes can we compare their error analytically rather than empirically?

$$\theta_1 \sim \Theta_1$$

A parameter from polynomial regression

$$\theta_2 \sim \Theta_2$$

A parameter from a neural network

Idea: Generalization Error Bounds for Algorithms

We need to understand what it means to talk about the generalization of an entire class of functions.

$$\theta_1 \sim \Theta_1$$

A parameter from polynomial regression

Considering probability over hypothesis space

We typically talk about these things in the literature in terms of "risk" where:

$$\{Z_1 = \ell(f^\theta, X_1 = x_1, Y_1 = y_1), \ldots, Z_N = \ell(f^\theta, X_N = x_N, Y_N = y_N)\}$$

$$\hat{Z} = \frac{1}{N}\sum_{i=1}^{N} Z_i$$

$$\hat{Z} = R_{\text{emp}}(\theta),\ \mathbb{E}[Z] = R(\theta)$$

empirical risk of θ

real risk of θ in infinite samples

Considering probability over hypothesis space

We typically talk about these things in the literature in terms of "risk" where:

$$\hat{Z} = R_{\text{emp}}(\theta),\ \mathbb{E}[Z] = R(\theta)$$

We would then be interested in the worst case generalization error over our entire "hypothesis" (to us: parameter) space:

$$\mathbb{P}\left[\sup_{\theta \in \Theta} |R(\theta) - R_{\text{emp}}(\theta)| > \epsilon\right]$$

what is largest gap I can have between empirical risk & actual risk

Generalization error bounds

We would then be interested in the worst case generalization error over our entire "hypothesis" (to us: parameter) space:

supremum - value of Θ that maximizes this dist

$$\mathbb{P}\left[\sup_{\theta\in\Theta}|R(\theta)-R_{\text{emp}}(\theta)|>\epsilon\right]=\mathbb{P}\left[\bigcup_{\theta\in\Theta}|R(\theta)-R_{\text{emp}}(\theta)|>\epsilon\right]$$

While proving the above equality is technically beyond the scope of the class, it should be intuitive that: if the probability on one side is high then necessarily the probability of the other side is high and visa versa

Generalization error bounds

We would then be interested in the worst case generalization error over our entire "hypothesis" (to us: parameter) space:

$$\mathbb{P}\left[\bigcup_{\theta\in\Theta} |R(\theta) - R_{\text{emp}}(\theta)| > \epsilon\right]$$

This looks like something we can almost deal with using our bounds but not quite because we are reasoning about the union over events which is not handled by our prior bounds.

Union Bound

The union bound is a crude way of dealing with probability bounds over unions of events:

$$P\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{i=1}^{n} P(A_i)$$

Union Bound

The union bound is a crude way of dealing with probability bounds over unions of events:

$$P\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{i=1}^{n} P(A_i)$$

eg. find range which runs how 345 as factors:
find factors of 3, 5,
take away factors of 15

Consider inclusion-exclusion principle, but dropping the correcting subtraction:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Equality no longer holds, but this clears up where the bound comes from

Reasoning about multiple hypotheses

Applying the union bound to the probability we had before, now we have an summation over terms that we know how to deal with:

$$\mathbb{P}\left[\bigcup_{\theta\in\Theta} |R(\theta) - R_{\text{emp}}(\theta)| > \epsilon\right] \leq \sum_{\theta\in\Theta} \left[|R(\theta) - R_{\text{emp}}(\theta)| > \epsilon\right]$$

Sum over all θ in Θ

no. of items summed × bounding bound used

Union bound

$$P\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{i=1}^{n} P(A_i)$$

Deriving a first generalization bound

$$\mathbb{P}\left[\bigcup_{\theta\in\Theta} |R(\theta) - R_{\text{emp}}(\theta)| > \epsilon\right] \leq \sum_{\theta\in\Theta} |R(\theta) - R_{\text{emp}}(\theta)| > \epsilon$$

Plugging in Hoeffding's bound:

$$P\left(\left|\bar{X}_n - E[X]\right| \geq \epsilon\right) \leq 2\exp\left(-2n\epsilon^2\right)$$

sup term = U term

$$\mathbb{P}\left(\sup_{\theta\in\Theta} |R(\theta) - R_{\text{emp}}(\theta)| > \epsilon\right) \leq 2|\Theta|\exp\left(-2n\epsilon^2\right)$$

Size of param space (cardinality)

Before inspecting this too carefully, let's do a few algebraic manipulations

Deriving a first generalization bound

$$\mathbb{P}\left(\sup_{\theta\in\Theta}|R(\theta)-R_{\text{emp}}(\theta)|>\epsilon\right)\leq \underbrace{2|\Theta|\exp\left(-2n\epsilon^2\right)}_{\delta}$$

$$P\left(\left|\bar{X}_n-E[X]\right|\geq\epsilon\right)\leq 2\exp\left(-2n\epsilon^2\right).$$

We have that with probability 1-delta:

$$|R(\theta)-R_{\text{emp}}(\theta)|\leq\epsilon\Rightarrow R(\theta)\leq R_{\text{emp}}(\theta)+\epsilon$$

Deriving a first generalization bound

$$\mathbb{P}\left(\sup_{\theta\in\Theta}|R(h)-R_{\text{emp}}(\theta)|>\epsilon\right)\leq \underbrace{2|\Theta|\exp\left(-2n\epsilon^2\right)}_{\delta}$$

$$P\left(\left|\bar{X}_n-E[X]\right|\geq\epsilon\right)\leq 2\exp\left(-2n\epsilon^2\right).$$

$\delta \leq 2|\Theta|\exp(2n\epsilon^2)$

Using the same isolation algebra as we did for Hoeffding's inequality (now isolating for epsilon) we have our first generalization error:

$$R(\theta)\leq R_{\text{emp}}(\theta)+\sqrt{\frac{\log(|\Theta|)+\log(2/\delta)}{2n}}$$

Interpreting the generalization bound

$$R(\theta) \leq R_{\text{emp}}(\theta) + \sqrt{\frac{\log(|\Theta|) + \log(2/\delta)}{2n}}$$

Now that we have this in a similar form to our prior inequality, let's again do some interpretation:

- Increase in parameter size leads to increase in bound
- Increase in dataset leads to decrease in bound

Interpreting the generalization bound

$$R(\theta) \leq R_{\text{emp}}(\theta) + \sqrt{\frac{\log(|\Theta|) + \log(2/\delta)}{2n}}$$

Elephant in the bound: for our studied models, the parameter space is infinite.

Though this bound can be refined, it does connect nicely back to no free lunch

Additionally: reducing the support of our model space might improve generalization (intuition).

Limiting the expressiveness of our functions: regularization

Regularization is the general term we use for limiting the expressiveness of our function class in machine learning and is typically used to encourage good generalization performance.

Explicit Regularization

$$\mathcal{L}_{\text{reg.}}(\theta) := \mathcal{L}(\theta) + \alpha C(\theta)$$

Explicit regularization is characterized by the addition of terms to our loss function that encode constraints over the parameters or outputs of our ML model.

- Weight decay (next slides)
- Robustness constraints
- Fairness constraints

Explicit Regularization

$$\mathcal{L}_{\text{reg.}}(\theta) := \mathcal{L}(\theta) + \alpha C(\theta)$$

Explicit regularization is characterized by the addition of terms to our loss function that encode constraints over the parameters or outputs of our ML model. Weight decay:

$$\mathcal{L}_{\text{ridge}}(\theta) := \mathcal{L}(\theta) + \alpha ||\theta||_2$$

- try to restrict magnitude of params

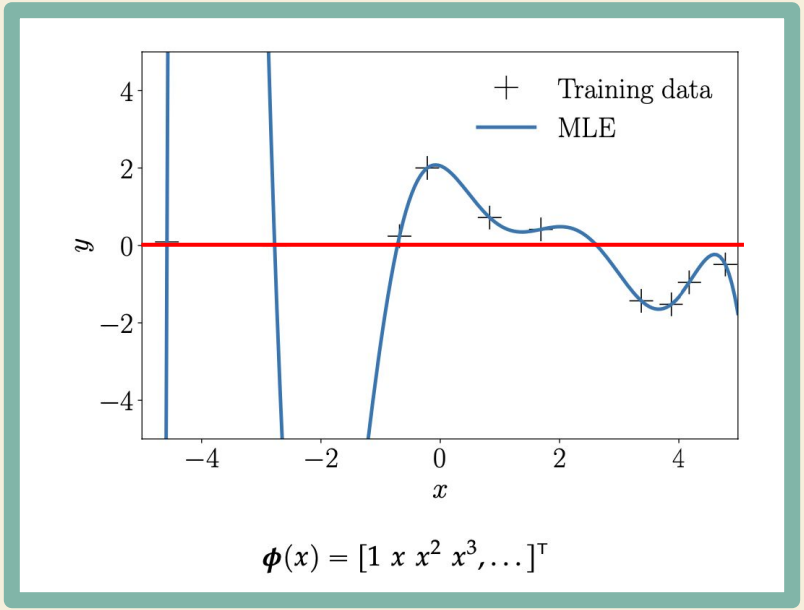
Explicit Regularization

$$\mathcal{L}_{\text{ridge}}(\theta) := \mathcal{L}(\theta) + \alpha||\theta||_2$$

α

Equal to zero

Limit tending to infinity



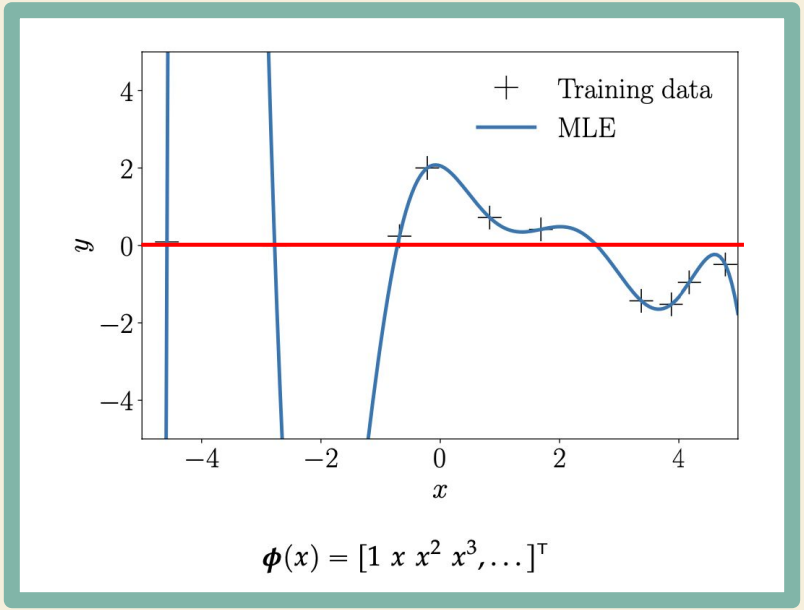
Explicit Regularization

$$\mathcal{L}_{\text{ridge}}(\theta) := \mathcal{L}(\theta) + \alpha||\theta||_2$$

α

Equal to zero

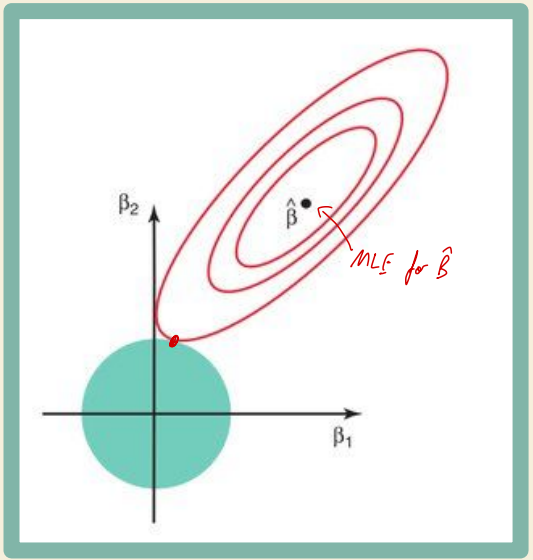
Limit tending to infinity



We have seen this kind of behavior before in BLR. Exercise: compute OLS for the above and show when/if it compares to MLE/MAP for linear regression

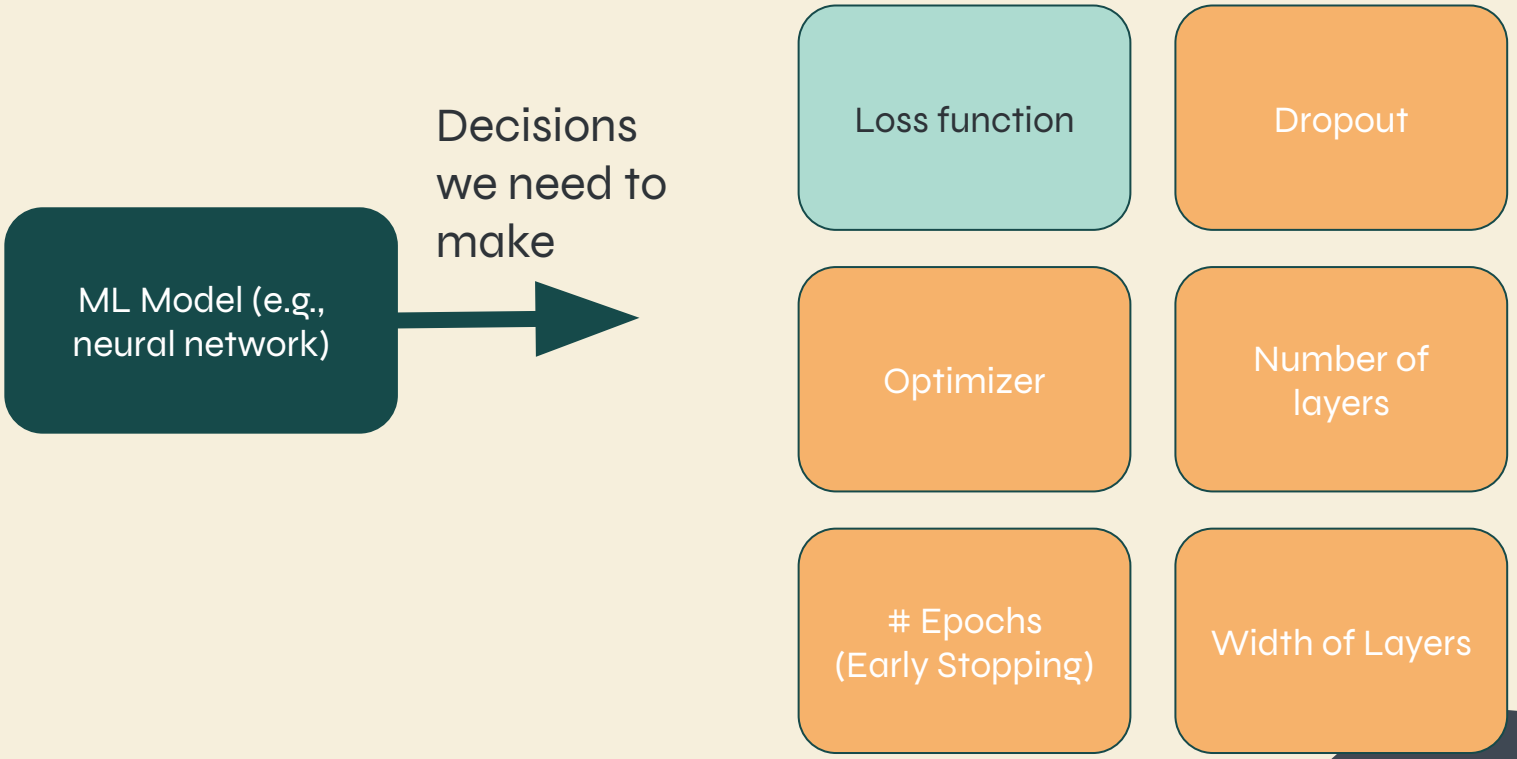
Explicit Regularization

$$\mathcal{L}_{\text{ridge}}(\theta) := \mathcal{L}(\theta) + \alpha||\theta||_2$$

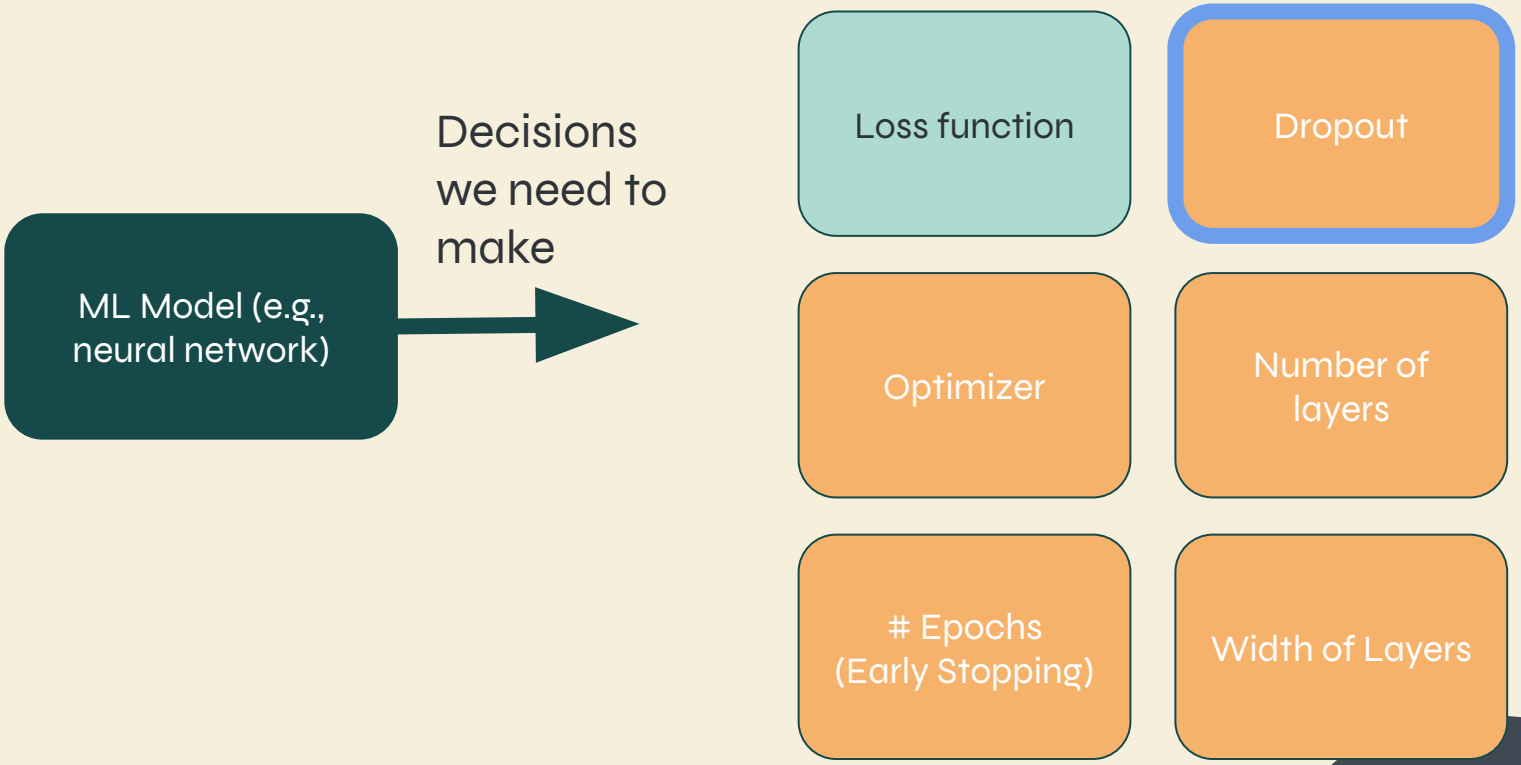


For the in between zero and infinity alpha case, one useful thing to think about is where the constraint set and unconstrained loss intersect.

Implicit Regularization

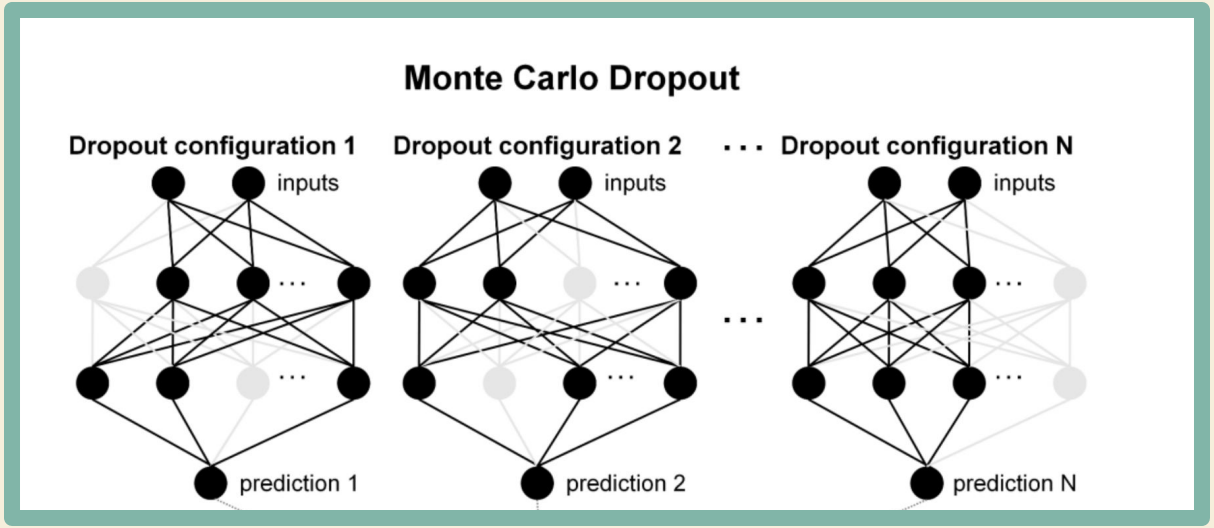


Example case: MC Dropout



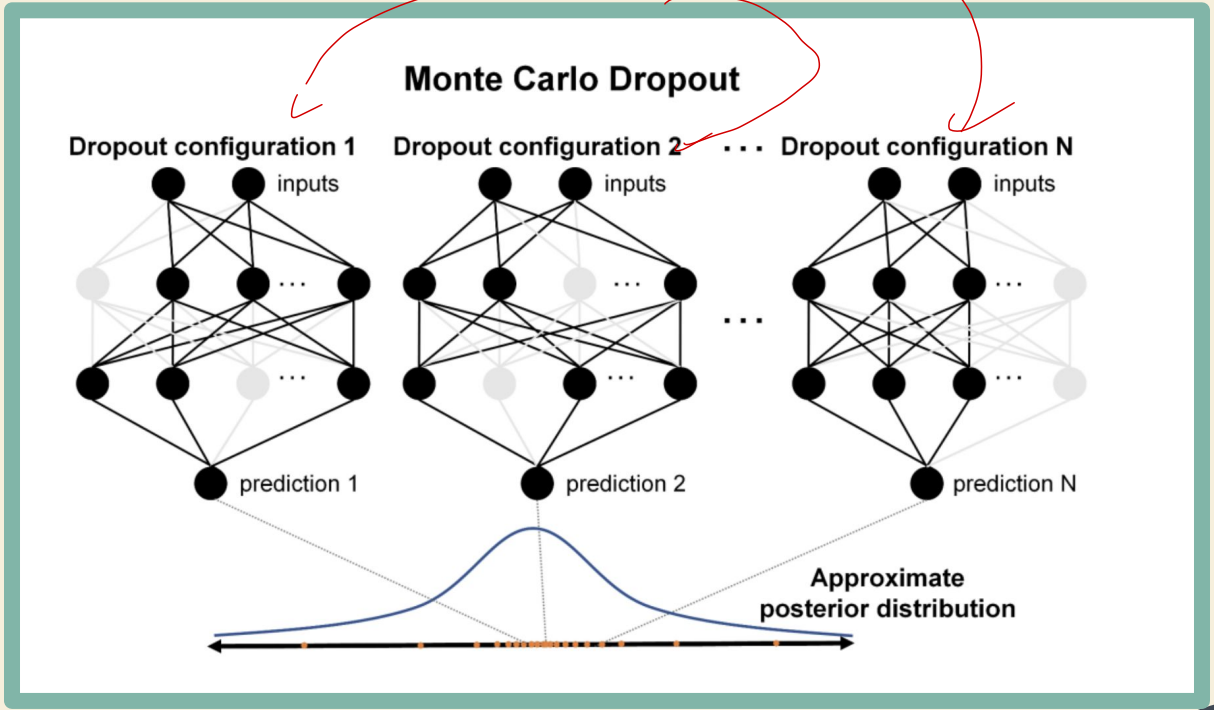
Example case: MC Dropout

not terminable



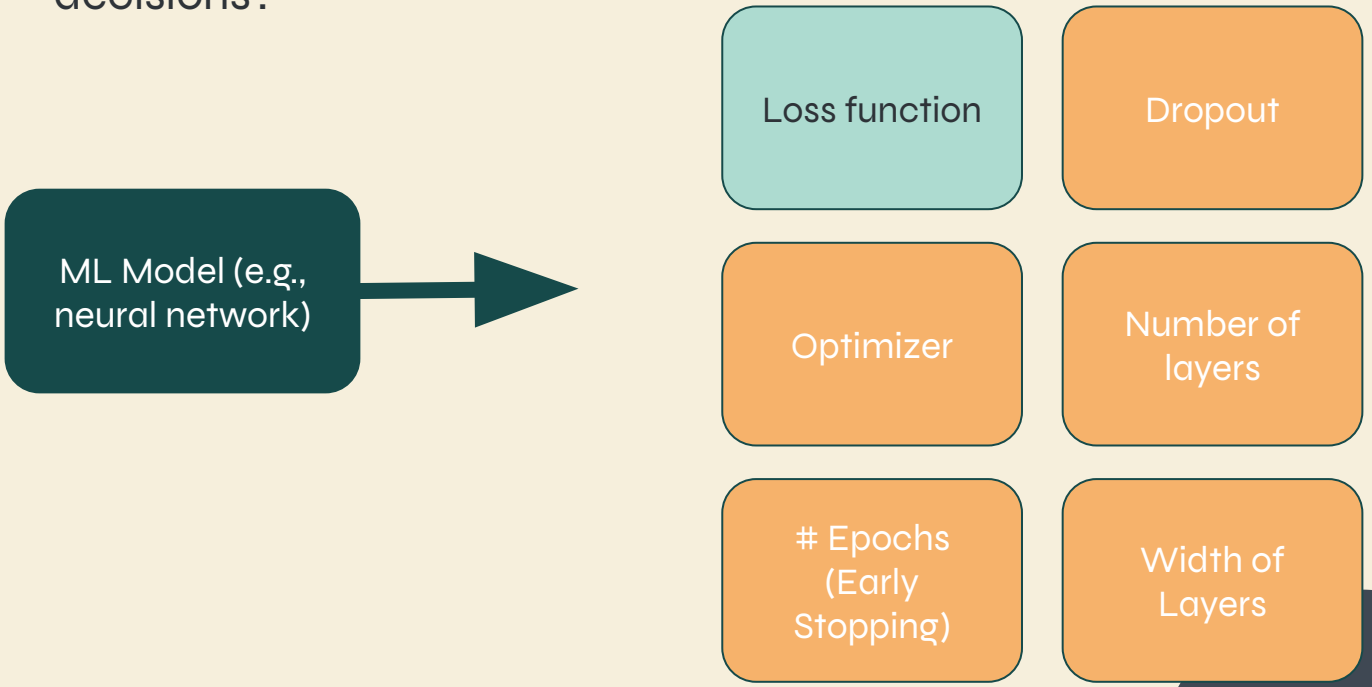
prevent relying on a particular neuron

Example case: MC Dropout

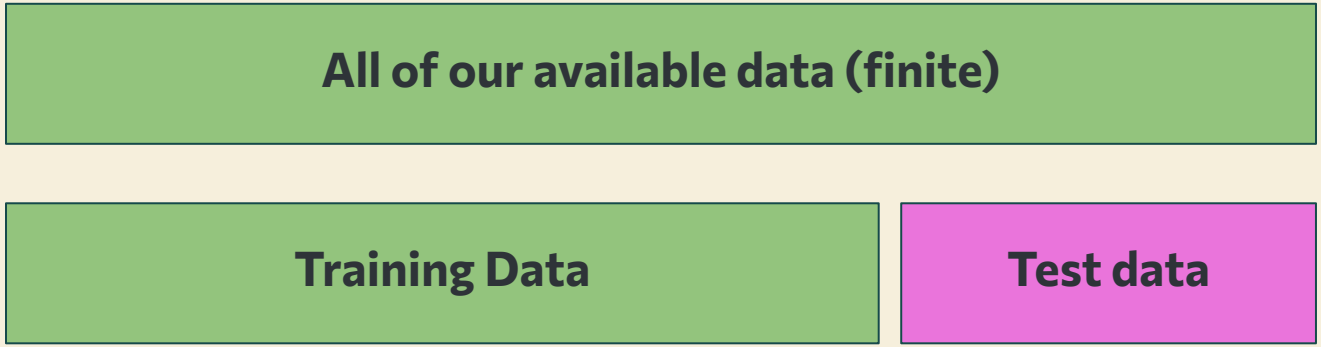


Hyper-parameters

We have established that there are a lot of decisions to make, but how can we compare all of our potential decisions?



Why not use the test set?



Consider training thousands of models each with a different hyper-parameter setting. We then just use the test set to pick the best one. What is wrong with this?

Why not use the test set?

$$\mathbb{E}_{p(\mathbf{x},y)}\left[\ell\left(y^{(i)}, f^{\theta}\left(\mathbf{x}^{(i)}\right)\right)\right]$$

We wanted an estimate of how our model would would perform on *unseen* data. But if we incorporate the test data into our optimization loop (e.g., optimizing hyper-parameters) then we have biased our generalization estimate.

Validation set

The technique that is often used is to *further* split our data into training, testing, and validation data.

We can then train on our training set, and optimize our hyper-parameters over the validation set while still getting our generalization error from un-biased concentration inequalities

Validation set

Potential issues?

Validation set

Potential issues?

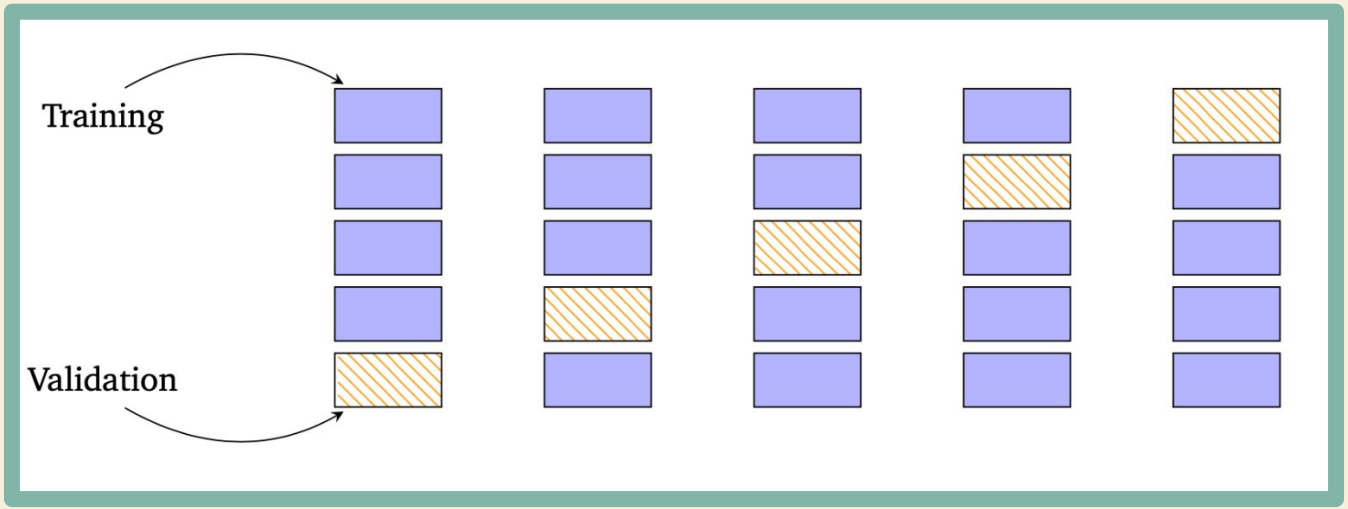
- Validation set is potentially very small so might be a high-variance estimator of our hyper-parameter performance
- Limiting our training data even further we know will have negative effects on our generalization

Cross-validation

The key idea behind cross-validation is to split our training data into K mutually exclusive subsets each of which will be used as the validation in one of K seperate algorithm runs

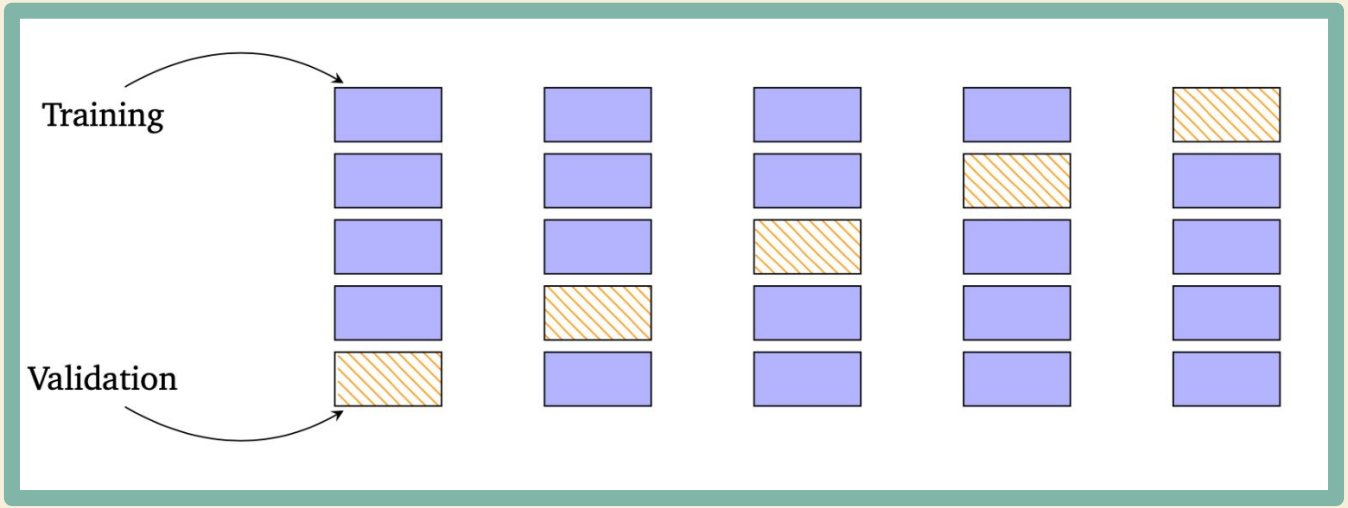
(K fold) Cross-validation

The key idea behind cross-validation is to split our training data into K mutually exclusive subsets each of which will be used as the validation in one of K seperate algorithm runs

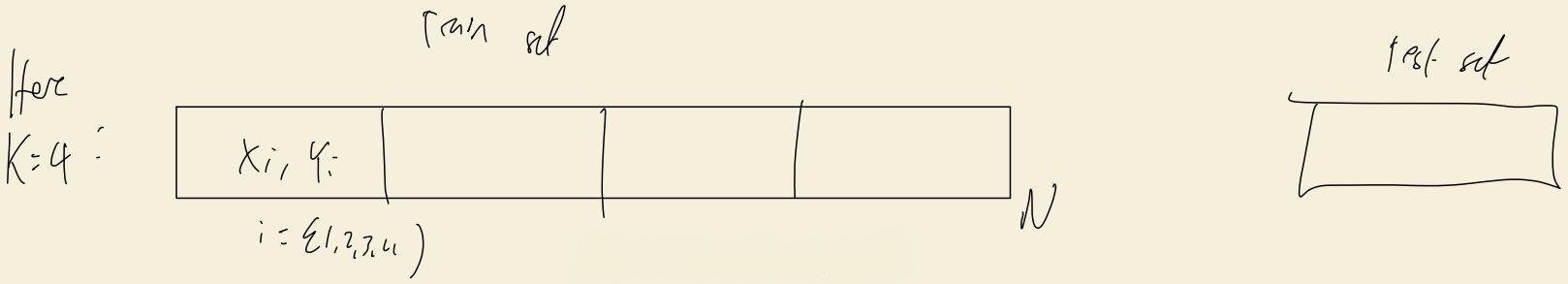


LOOCV: Cross-validation

What size should we make the validation set? One extreme choice is leave-one-out cross-validation (LOOCV): each validation set is 1 data-point large.



Writing out the full algorithm

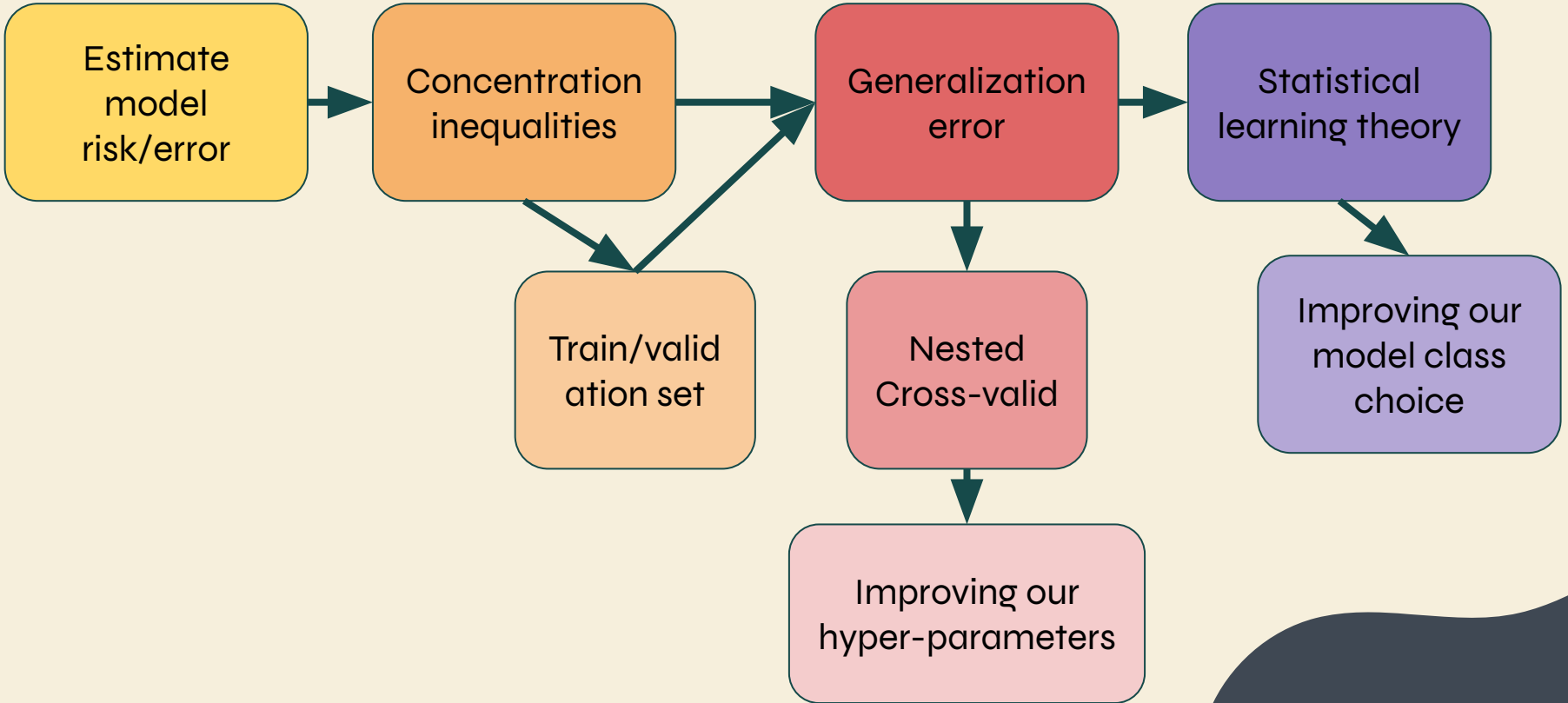


if Train on $x_{2\ldots4}, y_{2\ldots4}$, validate on x_1, y_1

V_{Err_1} = validation error 1

total validation error : $V_{err} = \frac{1}{4} \sum_{i=1}^{4} V_{err_i}$

Zooming out on model validation



Next lecture:
Bias-variance trade-off