Underfitting

Underfitting

What is Underfitting ?

Underfitting corresponds to a model that :

- Is too simple and inflexible to learn
- Cannot model (learn) the training data
- Cannot generalize to treat new data

A model that has a poor performance.
This bad performance is detected through the evaluation metrics.

Underfitting

How to reduce/avoid Underfitting ?

Underfitting

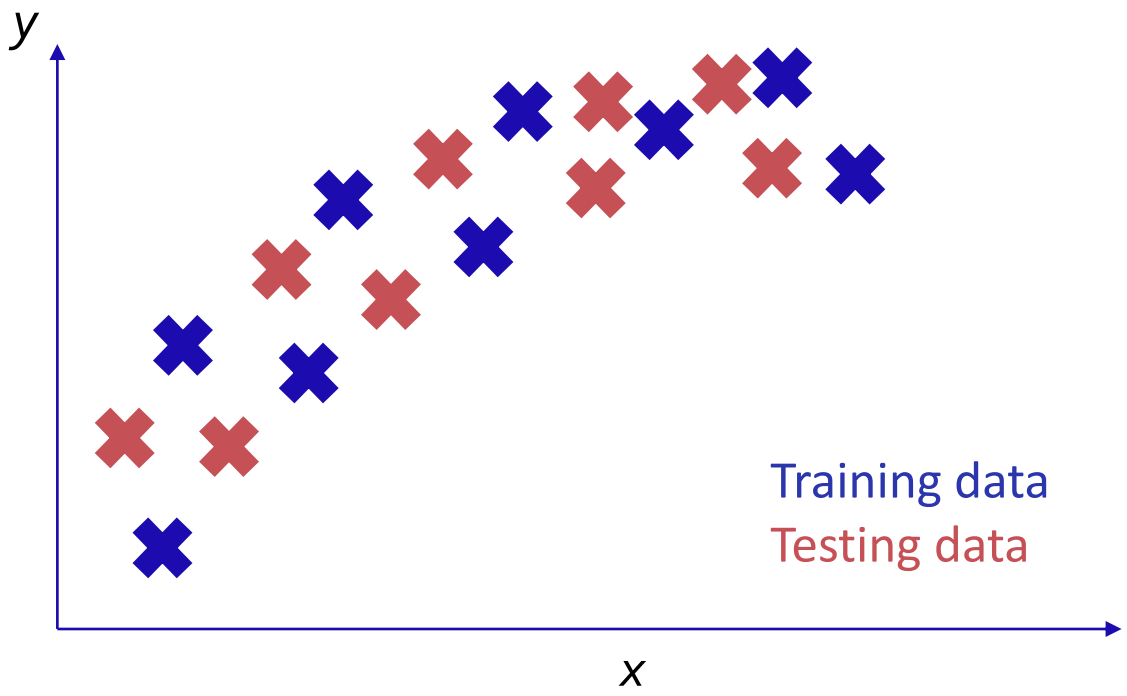
How to reduce/avoid Underfitting ?

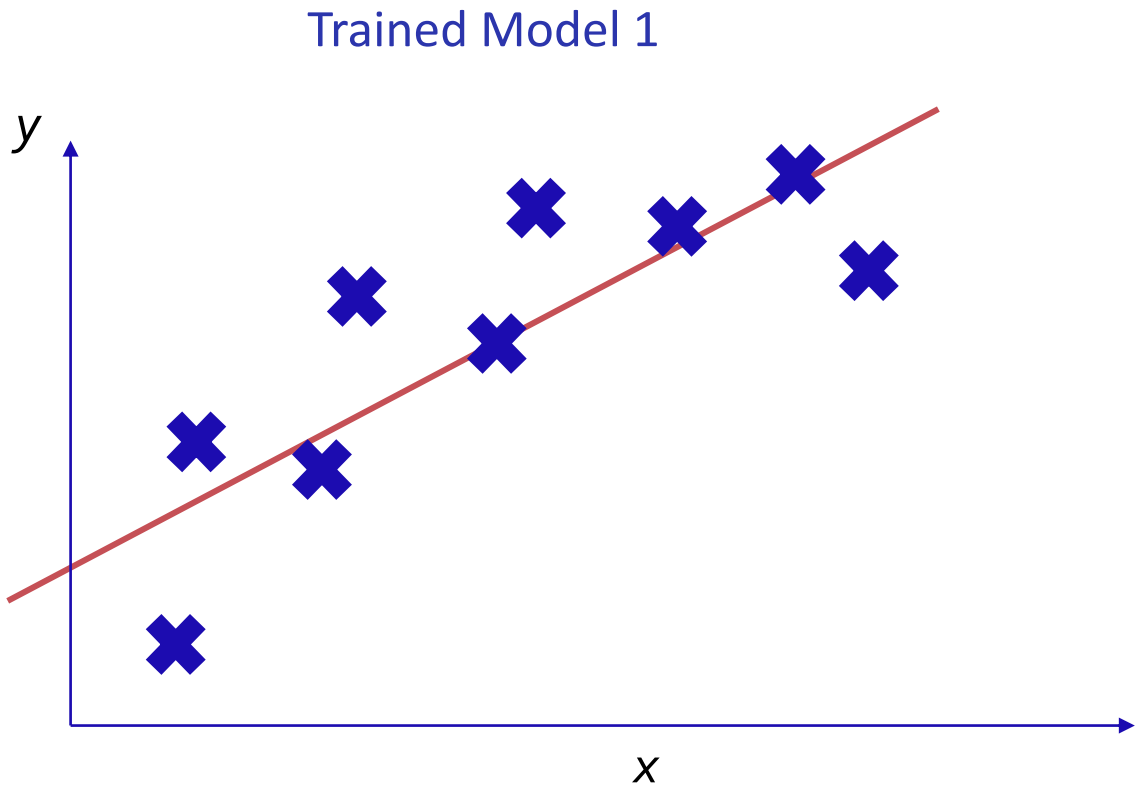
- Increase model complexity
- Increase data features (independent variables)
- Improve data (remove noise, remove redundancy)
- Add more data
- Increase the number of epochs
- Increase training duration

Overfitting

Overfitting

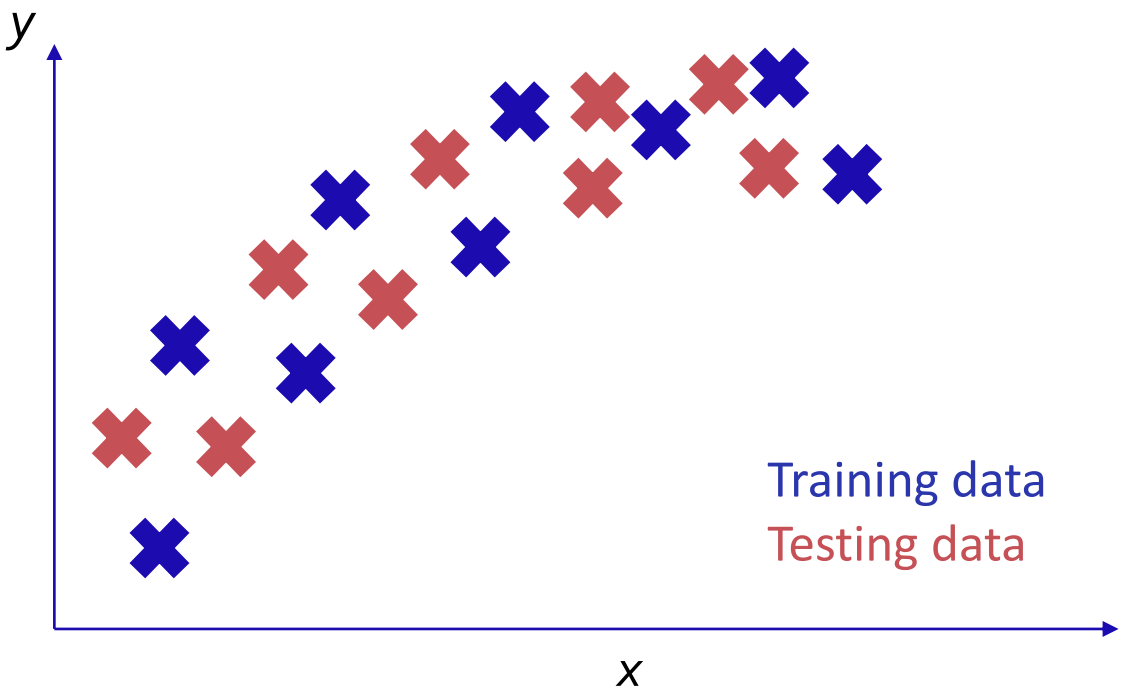
Bias-Variance Tradeoff :



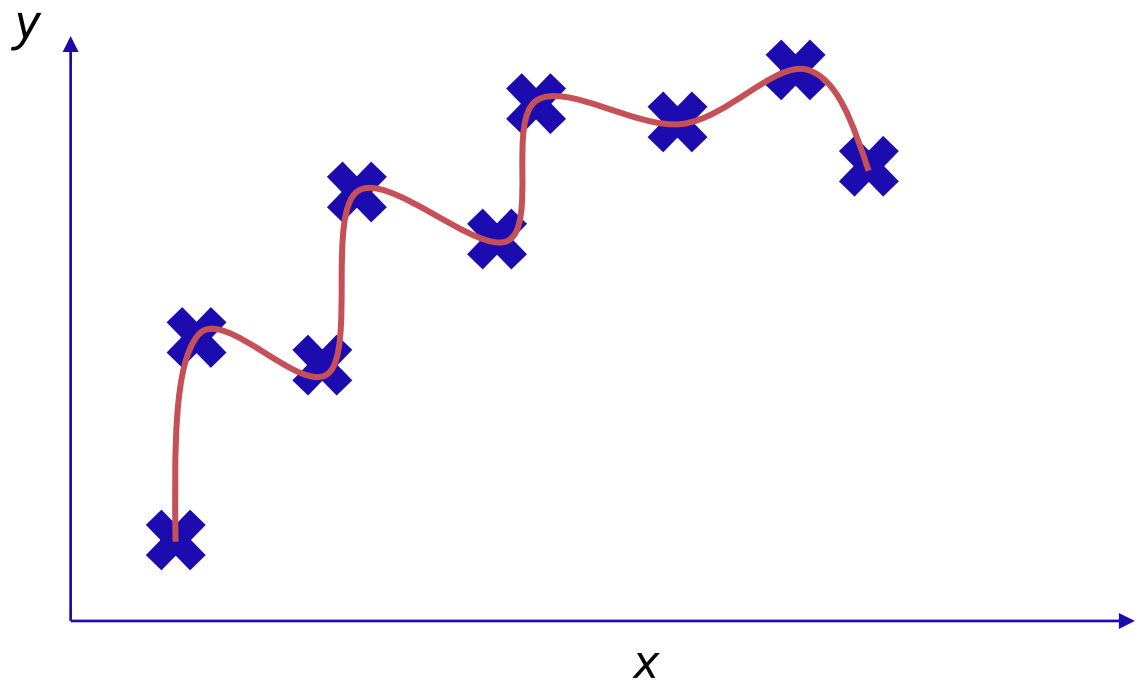


Overfitting

Bias-Variance Tradeoff :



Trained Model 2

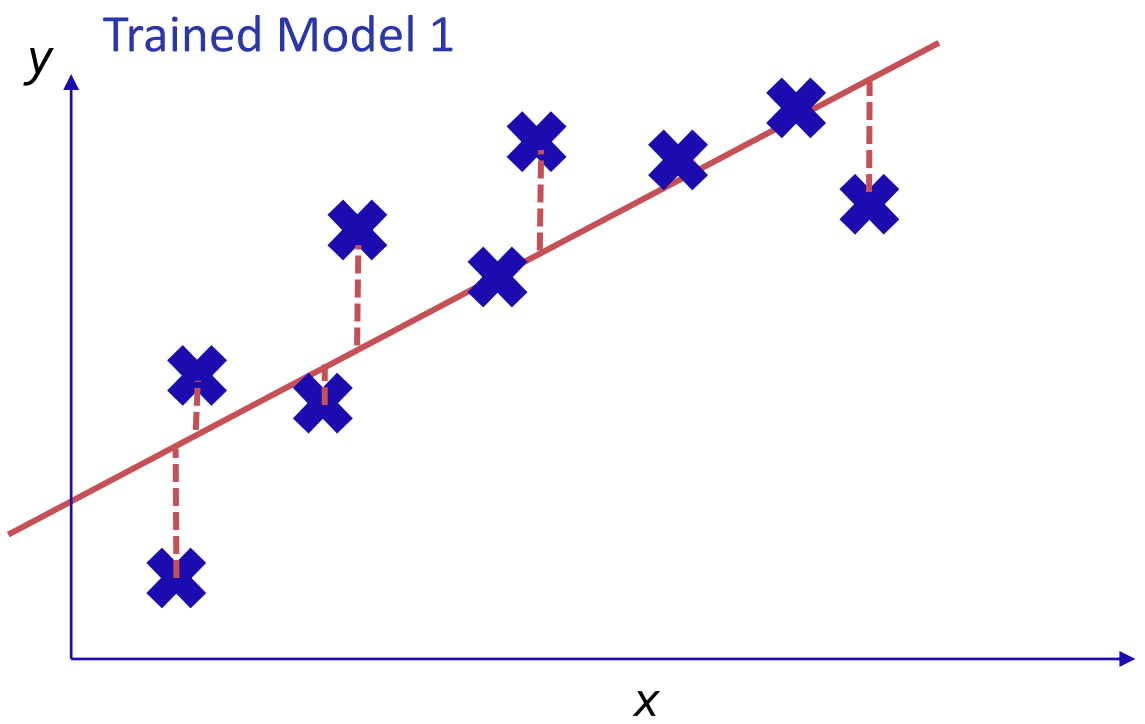


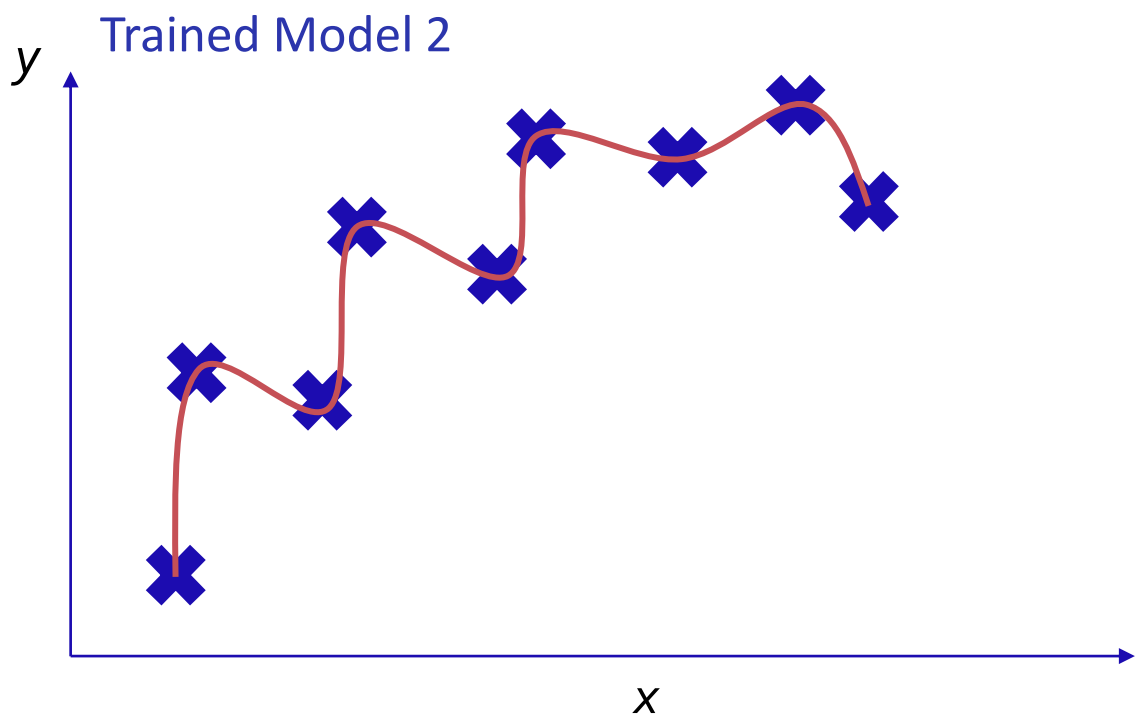
Overfitting

Bias-Variance Tradeoff :

We compare both trained models using RMSE (Root-Mean-Square Error) :

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

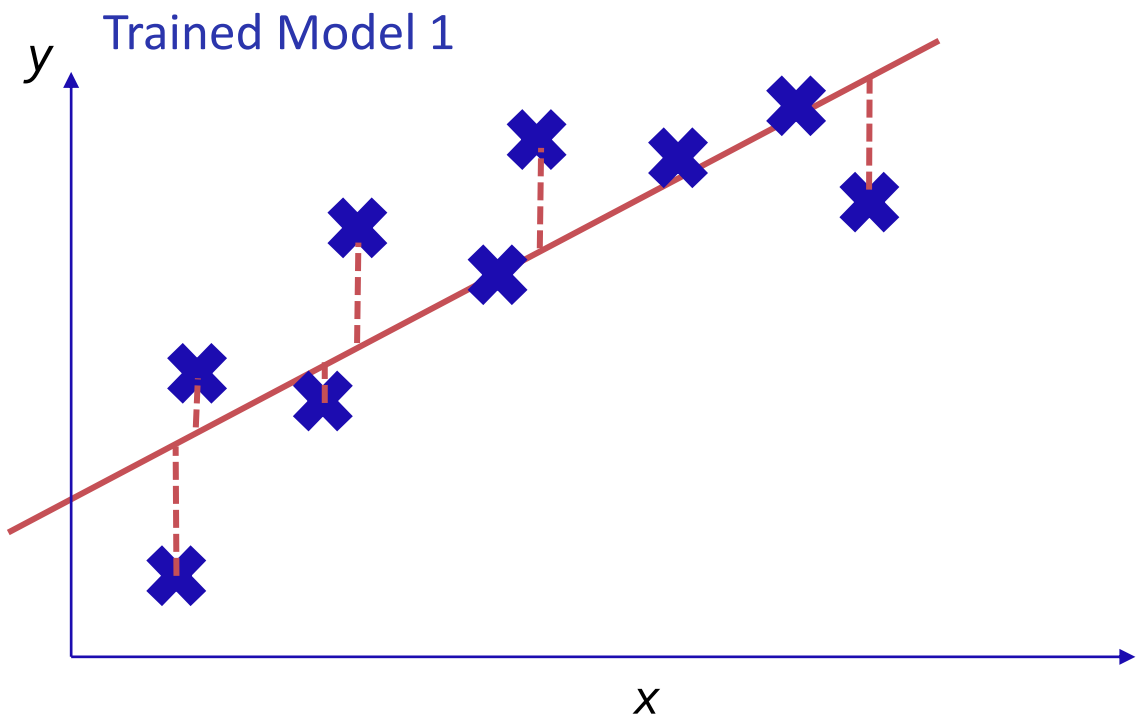


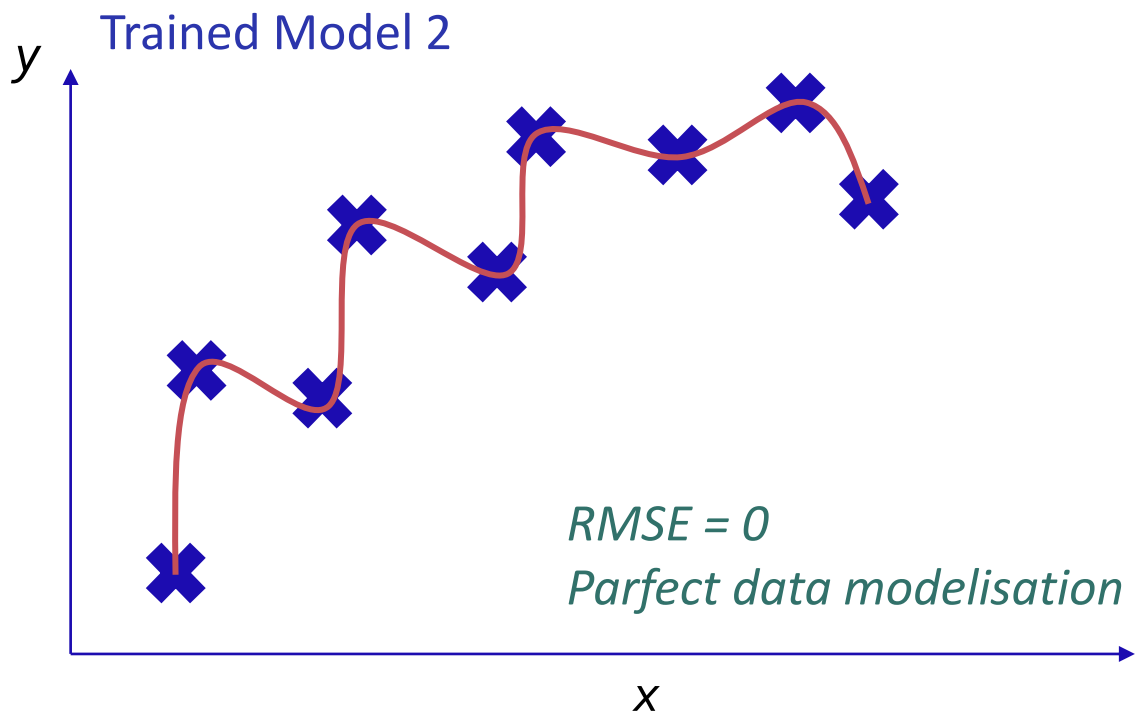


Overfitting

Bias-Variance Tradeoff :

We compare both trained models using RMSE :



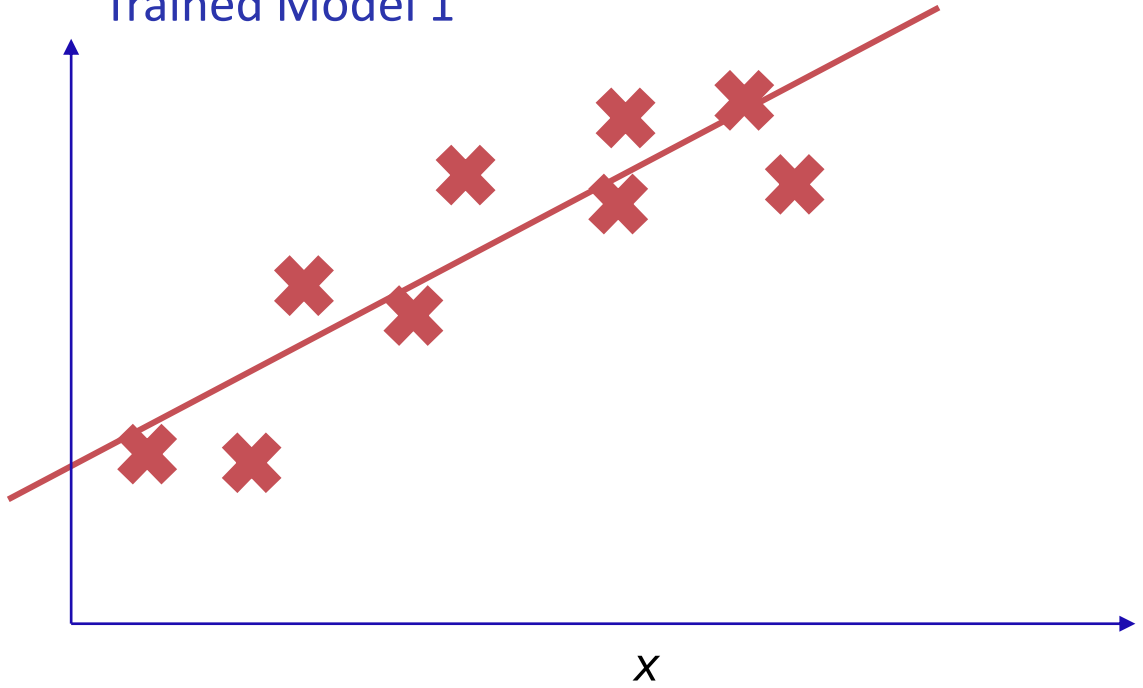


Overfitting

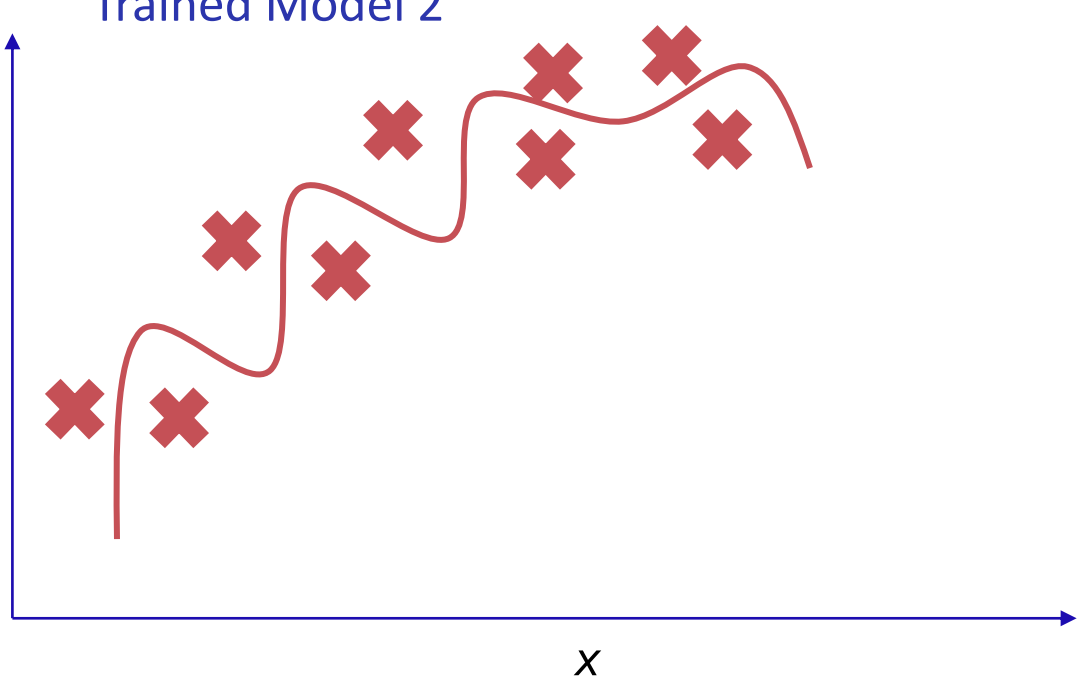
Bias-Variance Tradeoff : We compare both trained models using the test data:

RMSE 1 < RMSE 2 ➔ Model 1 has a better performance with new data !

Trained Model 1

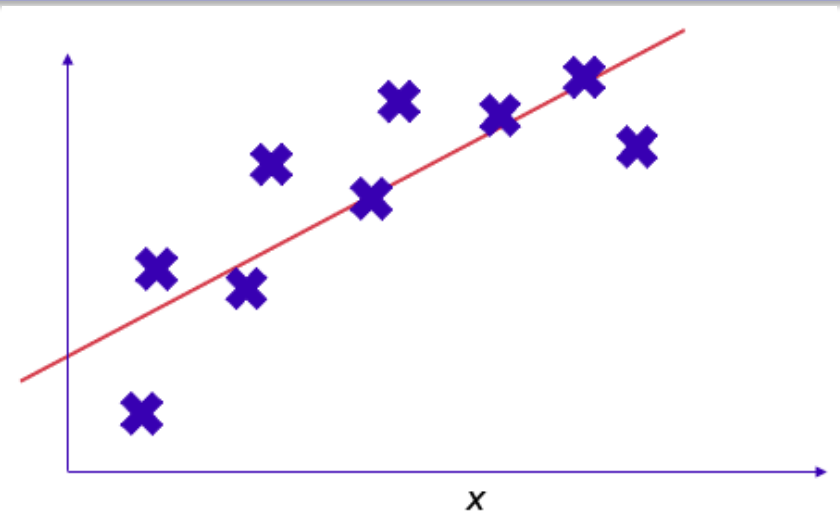


Trained Model 2

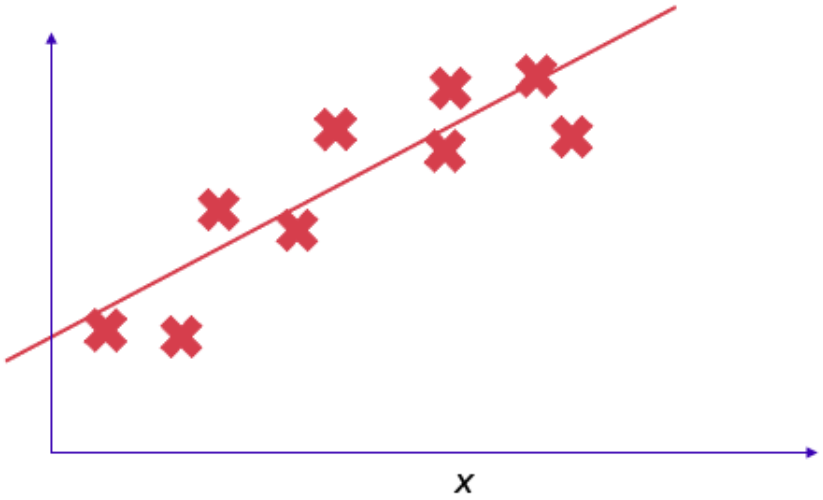


Overfitting

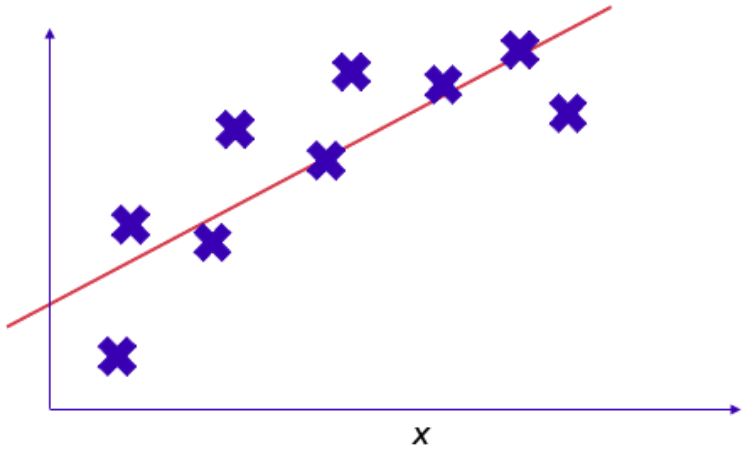
The incapability of a model
to reproduce data exactly is called **bias**



The different ‘fits’ of a trained model to
different datasets is called the **variance**

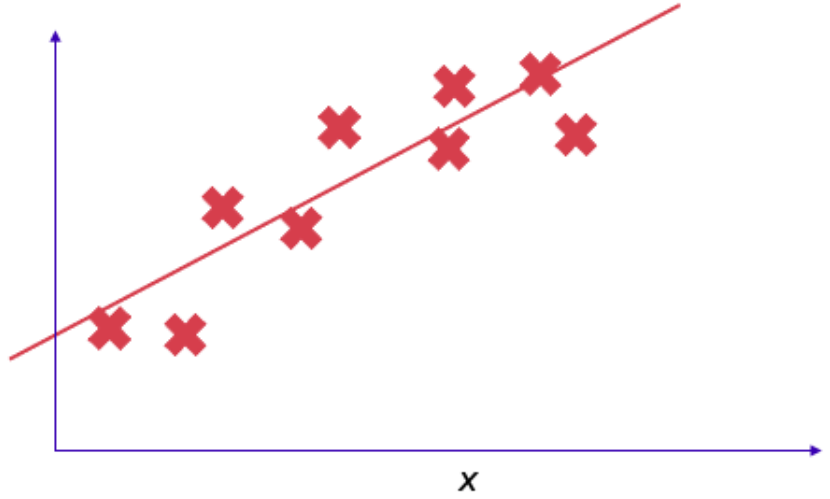


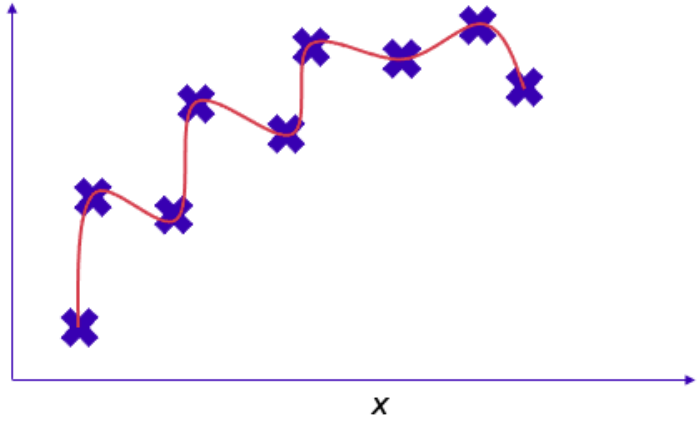
Overfitting



Relatively high bias
Relatively low variance

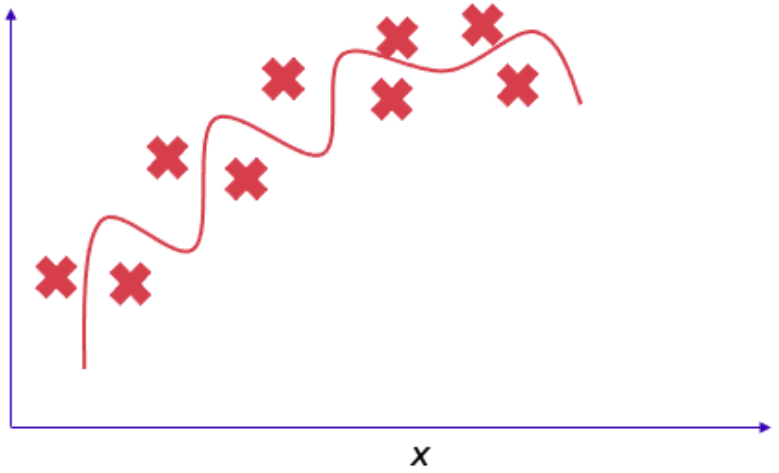
➔ Predictible model





Very low bias
Very high variance

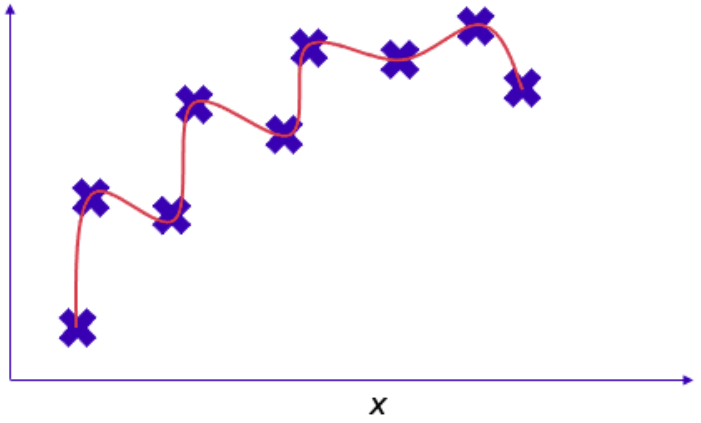
➔ Unpredictible model

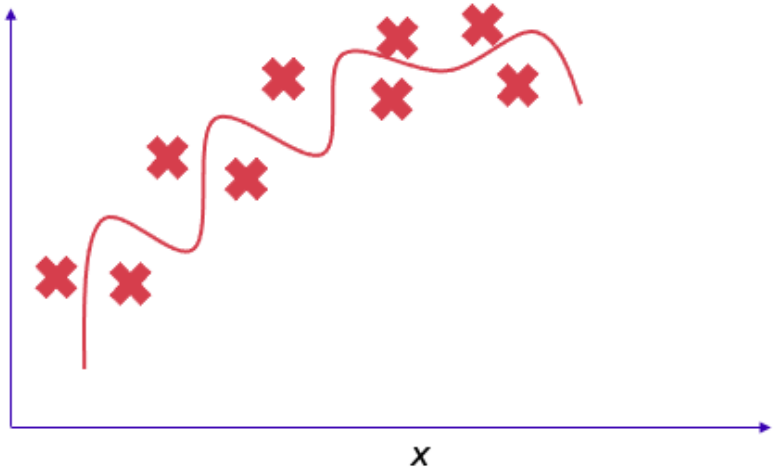


Overfitting

Overfitting :

The ideal model has both the bias and the variance low





Since this model has a perfect fit to the training data and a bad performance with new data

➔ The model is overfit

Overfitting

How to reduce/avoid Overfitting ?

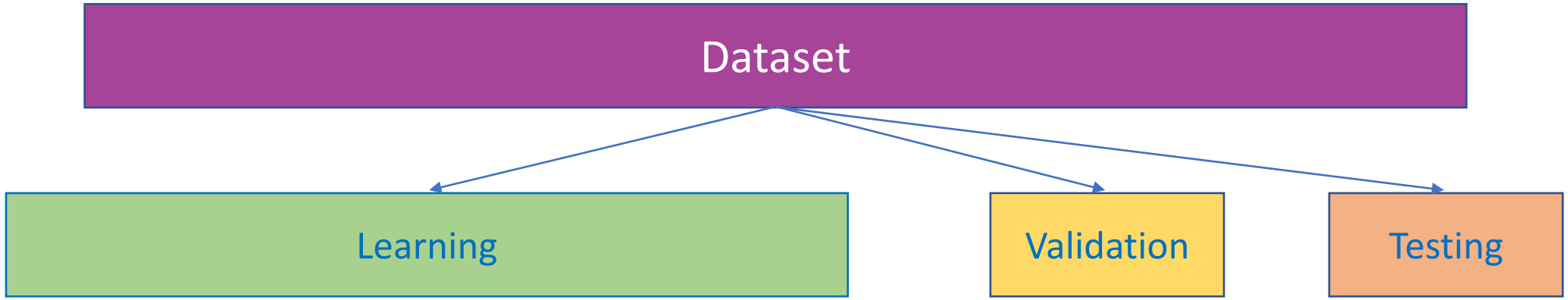
Overfitting

How to reduce/avoid Overfitting ?

- Cross-validation to tune hyperparameters
- Use more data for training
- Feature selection (removing irrelevant features)
- Early stopping
- Regularisation

Learning, Validation & Testing

Learning, Validation & Testing



Due to overfitting risk, the learning loss is not a good indication of a well-trained model

==> We need validation and training datasets

Note : These three datasets are separate and contain no repeated data

Learning, Validation & Testing

Learning (Training) dataset : The data used to train the model (adjust the weights)

- During each epoch, our model will be trained over the entire learning dataset
- Labeled (loss & accuracy in each epoch can be provided)

Validation dataset : The data used to validate our model during training

- After weights are adjusted, the model is validated using this dataset
- Does not contribute in the weight adjustement (learning)
- Helps us choose the best hyperparametres for our model
- Helps us detect overfitting
- Labeled (loss & accuracy in each epoch can be provided)

Learning, Validation & Testing

Test dataset : The data used to test the model after it has been trained and validated

- Should not be labeled (the same condition as in using our deployed model in real life)

The ultimate goal of machine learning and deep learning is to build models that are able to generalize and perform well on new unseen data.

Learning, Validation & Testing

Hyperparameters :

- Variables set before the model's training

- Help in model selection

- Include parameters that :

  - specify the network structure (number of hidden layers)

  - determine how the network is trained (number of epochs, batch size, learning rate)

Learning, Validation & Testing

Définitions :

Batch size :

The size of the training set

Hidden layers :

Layers between the input and the output

layers in the neural network

Epoch :

A pass over the entire training database

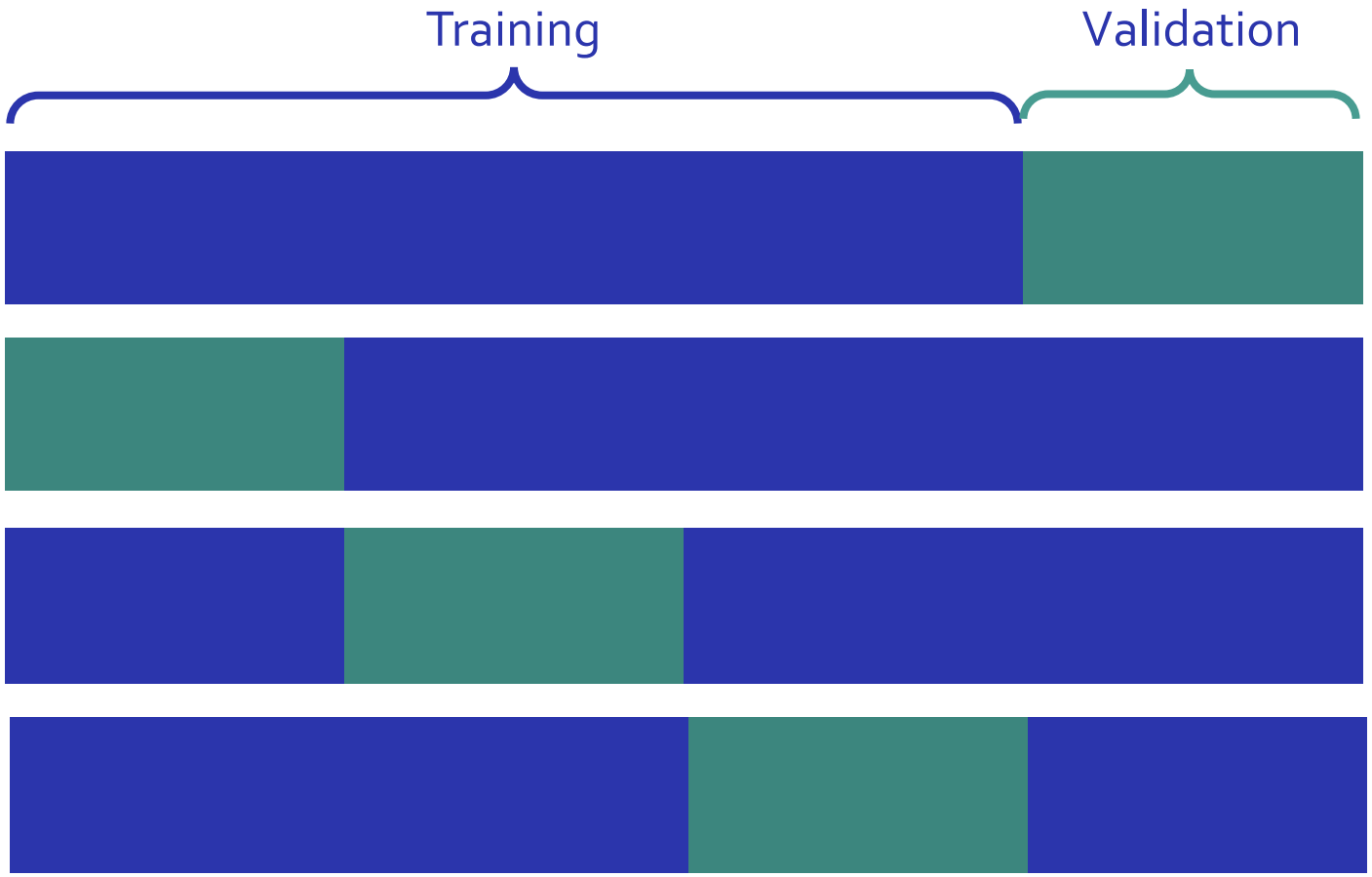
Learning rate :

A parameter that controls the rate at which the neural

network learns (ranges between 0.0 and 1.0)

Learning, Validation & Testing

Cross-Validation :



Cross-validation trains the model on subsets of the available data and evaluates it on the complementary subset that was not used for training.

Learning, Validation & Testing

Cross-Validation :

K-fold cross-validation :

- The data is split into k subsets/folds.
- Training is done on k-1 subsets
- Evaluation is done on the left-out subset.
- This process is repeated k times with a different subset left for evaluation each time.

Learning, Validation & Testing

Cross-Validation :

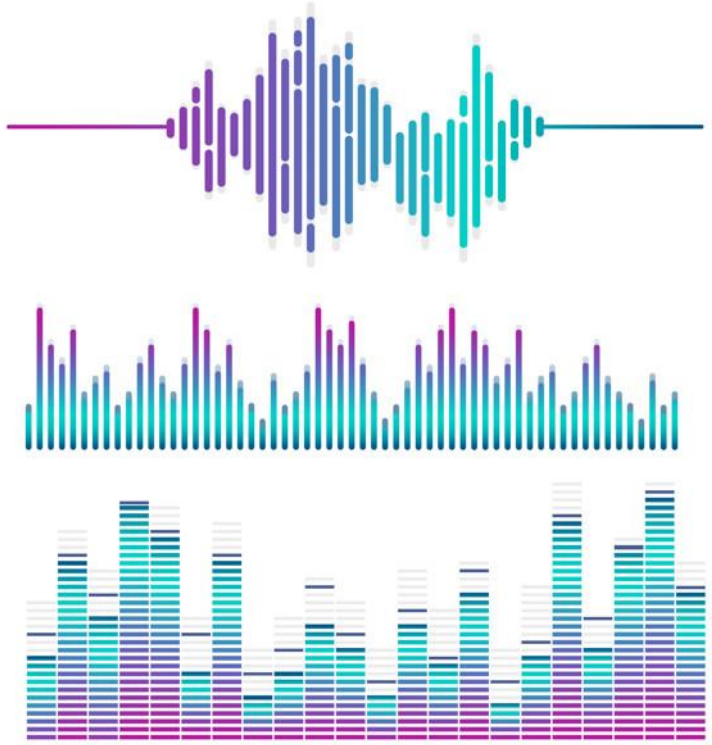
- Helps detect overfitting
- Is used to optimize hyperparameters of a model
- Is used for model selection

Learning, Validation & Testing

Example : Model selection with k-fold cross-validation (with k = 4)

Model 1	Model 2	Model 3	Model 4
% Quality of performance	% Quality of performance	% Quality of performance	% Quality of performance
85%	87%	87%	87%
87%	88%	88%	88%
90%	91%	91%	91%
88%	89%	89%	89%
Overall performance = 87,5% (average of performances)	88,75%	93,5%	97,0%

Regularisation



Regularisation

Regularisation :

- is a technique that introduces slight modifications to the learning algorithm for it to generalize better

- also improves the model’s performance on new unseen data

- helps reduce overfitting by reducing the model's complexity

- introduces a small bias to our model that, in turn, reduces the variance significantly

Regularisation

1. L1 Regularisation (Lasso Regression) :

$$L1Regularisation = Lossfunction + \lambda \sum_{i=1}^{n} |w_i|$$

Regularisation term that adds bias to the model to avoid perfect fitting

Regularisation

1. L1 Regularisation (Lasso Regression) :

$$L1Regularisation = Lossfunction + \lambda \sum_{i=1}^{n} |w_i|$$

Regularisation term that adds bias to the model to avoid perfect fitting

λ determines how strongly the regularization influences the network's training.

- If λ = 0 --> no regularization at all
- If λ = 1,000,000,000 --> very high regularization (too much bias, model hardly learns anything)

Reasonable choices of the regularization strength λ (0.001, 0.01, 0.1, 1.0, etc.)

The best λ can be chosen using cross validation

Regularisation

2. L2 Regularisation (Ridge Regression) :

$$L2Regularisation = Lossfunction + \lambda \sum_{i=1}^{n} w_i^2$$

Regularisation term that adds bias to the model to avoid perfect fitting

The best λ can be chosen using cross validation

Regularisation

L1 versus L2 Regularisation :

L1 Regularisation shrinks less important parameters to 0

L2 Regularisation shrinks less important parameters close to 0 but not equal to 0

L2 Regularisation penalises larger weights more severely :

** Example : A weight of 10 get a penalty = 10 in L1 Regularisation and a penalty = 100 in L2 Regularisation

Regularisation

3. Dropout Regularisation :

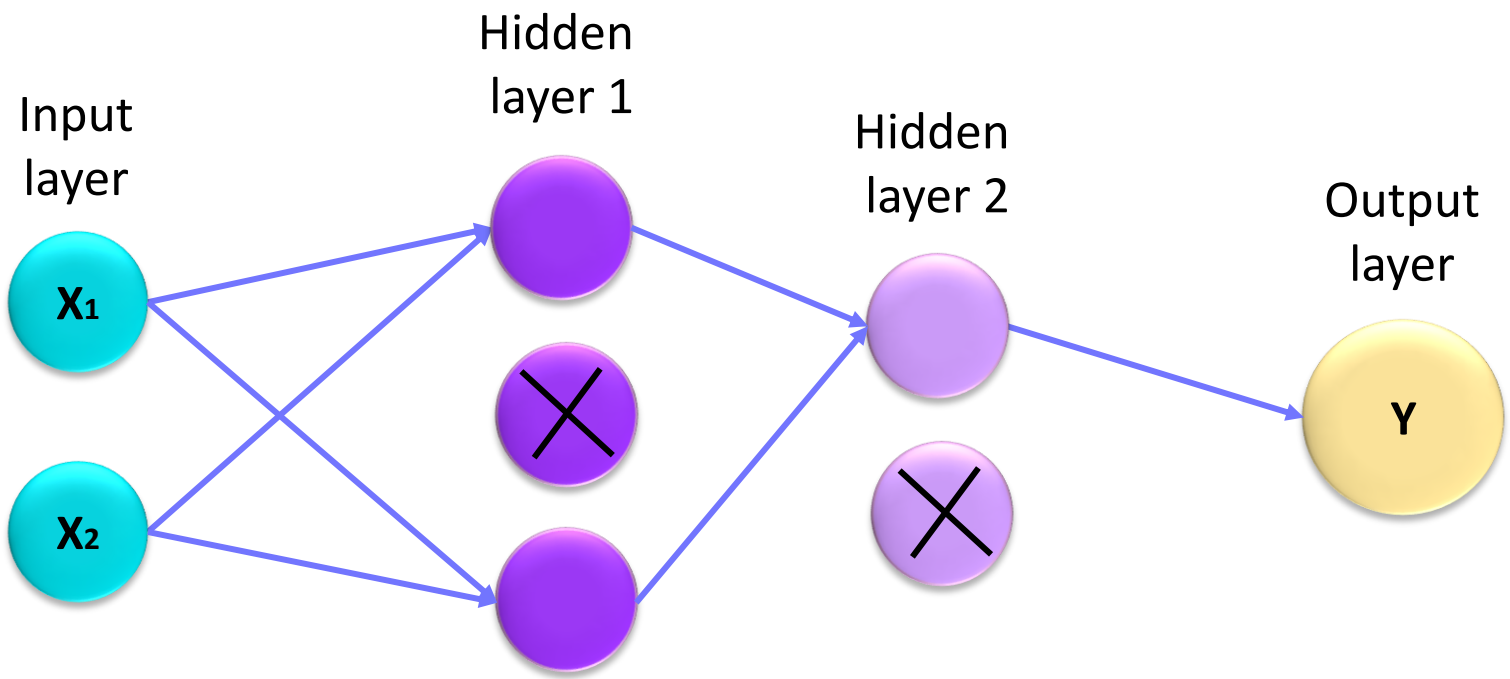
Dropout regularisation :

- is a computationally cheap type of regularisation
- Produces very good results
- Is the most frequently used regularisation technique in deep learning

Regularisation

3. Dropout Regularisation :

Dropout regularisation : At every iteration, the algorithm randomly selects some nodes and removes all connections (inputs & outputs) to them.

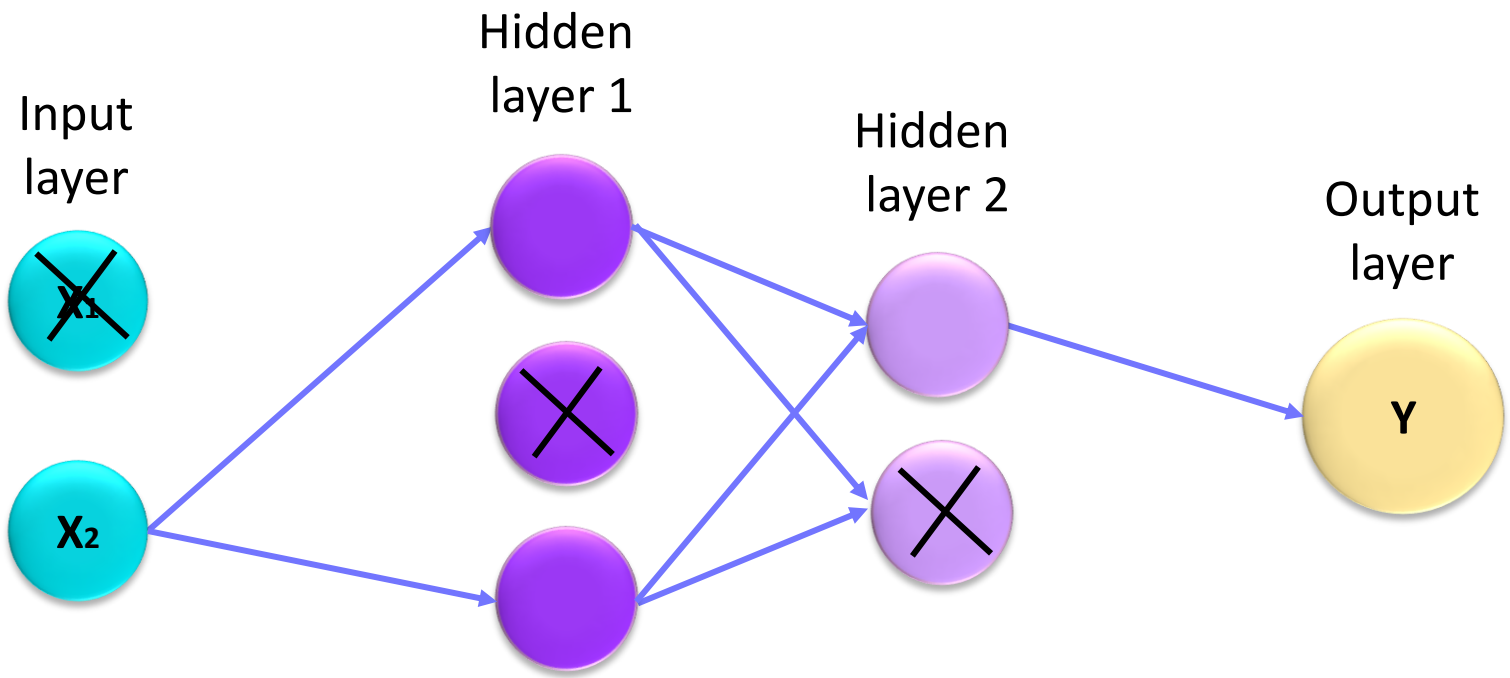


Each iteration has a different set of dropped out nodes

Regularisation

3. Dropout Regularisation :

The probability of choosing the number of nodes to drop out is a hyper parameter that we can assign to the dropout function.



Dropout can be applied to both the hidden and the input layers

Dropout is usually preferred for large neural networks

Regularisation

4. Early Stopping :

As soon as the performance on the validation dataset starts to decrease (loss starts to increase), the training is stopped immediately, hence the **early stopping**.

Regularisation

4. Early Stopping :

Example :

The training is stopped at the dotted line

source : Regularization Techniques | Regularization In Deep Learning (analyticsvidhya.com)

Regularisation

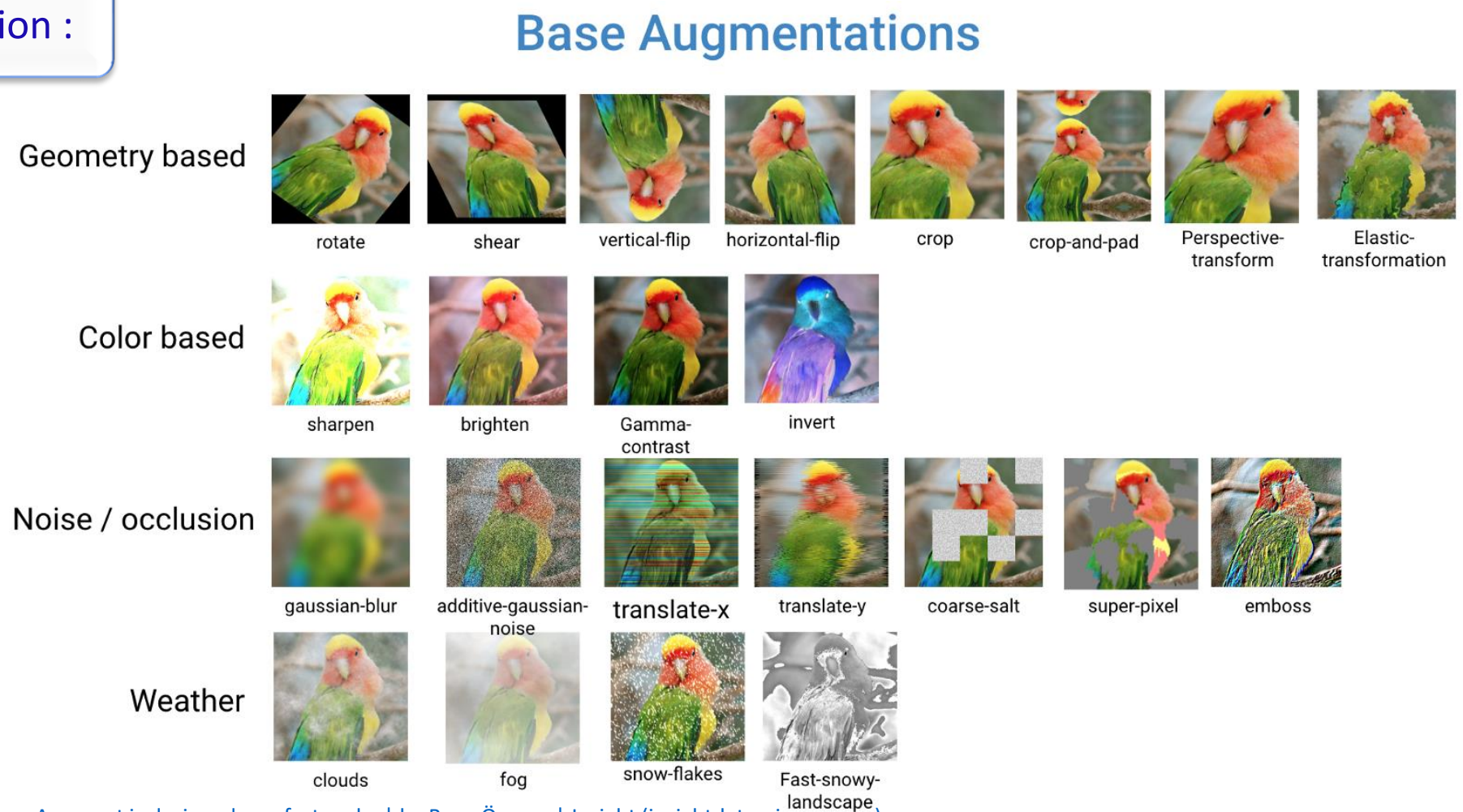
5. Data Augmentation :

Data augmentation corresponds to increasing the size of the training dataset

** Example : We can introduce variations in images : rotation, flipping, rescaling, shifting

Regularisation

5. Data Augmentation :



source : AutoML for Data Augmentation. DeepAugment is designed as a fast and… | by Barış Özmen | Insight (insightdatascience.com)

Artificial Neural Networks

by

Dr. Vera DAMERJIAN PIETERS