Deep Learning - COSC2779/2972

Introduction to Deep Learning

Dr. Ruwan Tennakoon

RMIT UNIVERSITY

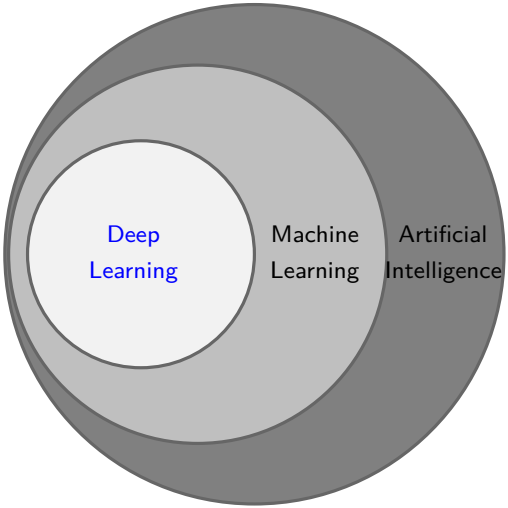
Semester 2, 2022

Reference: *Chapter 5: Ian Goodfellow et. al., "Deep Learning", MIT Press, 2016.*

Outline

Outline

What is Deep Learning?



AI: Any technique that enable computers to mimic human behaviour.

ML: Machine learning is the field of study that gives the Ability to learn without being explicitly programmed - Arthur Samuel (1959).

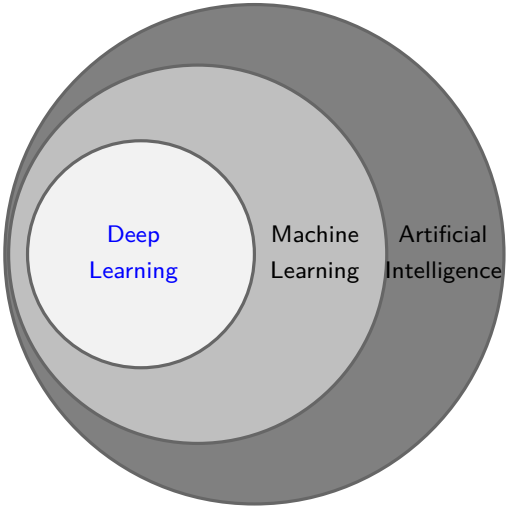
More technically: "A computer program is said to learn:

- Some class of **tasks T**
- From **experience E**, and
- **Performance measure P**

If its performance at tasks in T, as measured by P, improves with experience E".

- Tom Mitchell (1998)

What is Deep Learning?

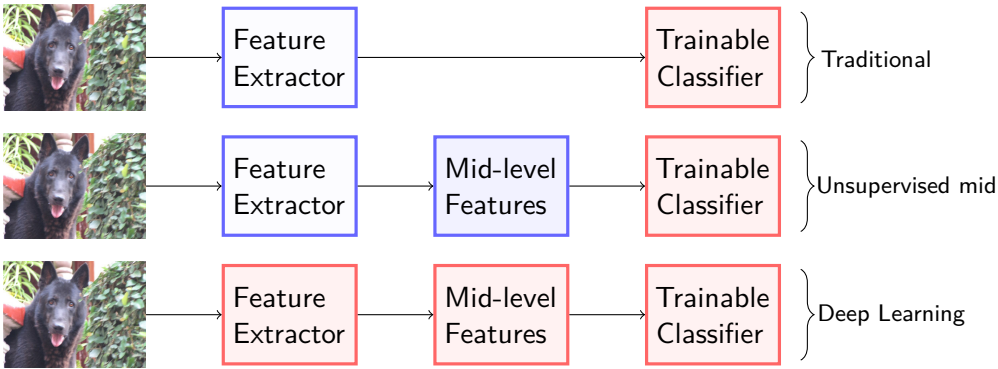


Deep learning is a ML technique that learns features and tasks directly from data using neural network framework.

"*Deep learning methods aim at* ***learning feature hierarchies*** *with features from higher levels of the hierarchy formed by the composition of lower level features. Automatically learning features at multiple levels of abstraction allow a system to learn complex functions mapping the input to the output* ***directly from data, without depending completely on human-crafted features.***"

- Yoshua Bengio, "Deep learning of representations for unsupervised and transfer learning", 2012.

Learning Hierarchical Representations



Handcrafted features are **time consuming, brittle and not scalable** in practice. DL learn underlying features directly from data.

What is Deep Learning?

According to Yann LeCun, Yoshua Bengio & Geoffrey Hinton, "Deep Lerning ", Nature 2015.

- Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.
- Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer.
- Deep convolutional nets have brought about **breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.**

Why use Deep Multi Layered Models?

Theoretical result [Cybenko, 1989]: one hidden-layer NN can approximate any continuous function over compact domain to arbitrary accuracy given enough hidden units!.

Why use Deep Multi Layered Models?

Why use Deep Multi Layered Models?

Theoretical result [Cybenko, 1989]: one hidden-layer NN can approximate any continuous function over compact domain to arbitrary accuracy given enough hidden units!.

Why use Deep Multi Layered Models?

- Argument 1: Visual scenes are hierarchially organized.
- Argument 2: Biological vision is hierarchically organized.
- Argument 3: Shallow representations are inefficient at representing highly varying functions.

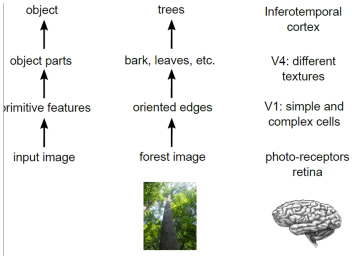
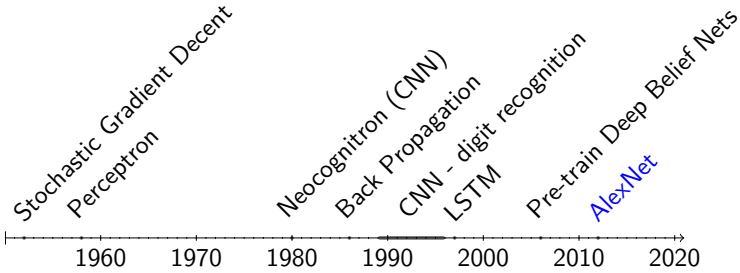


Image: Richard E. Turner

Some History



Neural Networks date back decades, so Why the resurgence?

Big Data

Larger Data sets.

Easier collection and storage.

IMAGENET

Computation

Graphic Processing Units.

Massively parallelizable.

Software

Improved Algorithms

Widely available open source frameworks.

TensorFlow

PyTorch

Why Now?

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is an annual competition helped between 2010 and 2017.

The datasets comprised approximately 1 million images and 1,000 object classes.

The annual challenge focuses on multiple tasks for image classification.

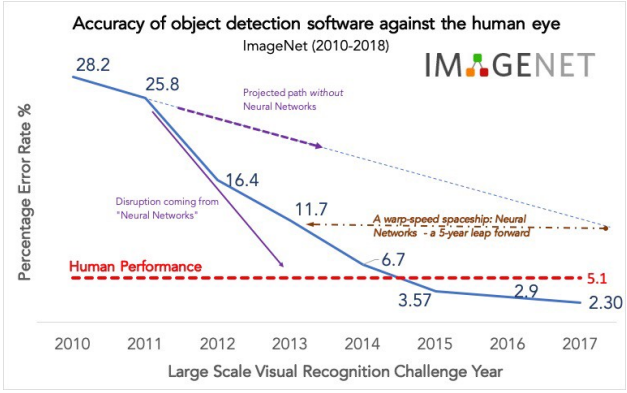
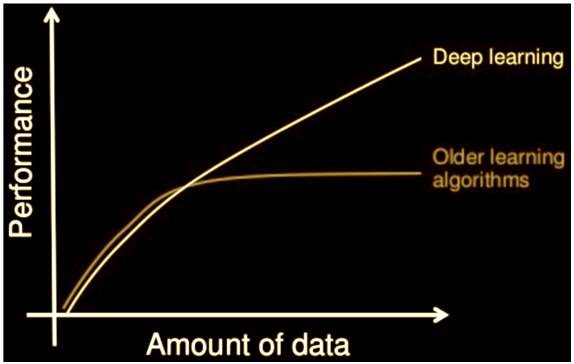


Image source: ImageNet
Alex Krizhevsky, et al. "ImageNet Classification with Deep Convolutional Neural Networks" developed a convolutional neural network that achieved top results on the ILSVRC-2010 and ILSVRC-2012 image classification tasks.

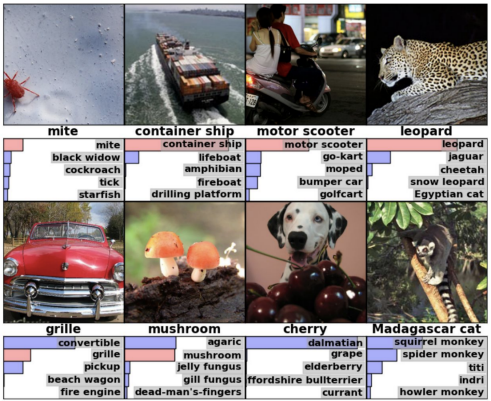
Why Now?



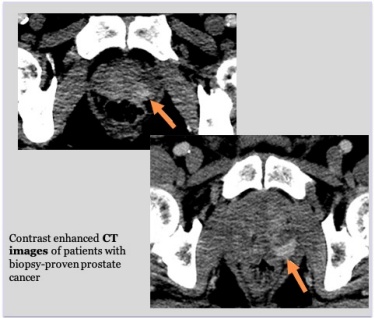
When a large neural networks is trained with more and more data, their performance continues to increase. This is generally different to other machine learning techniques that reach a plateau in performance. - Andrew Ng.

Outline

ImageNet



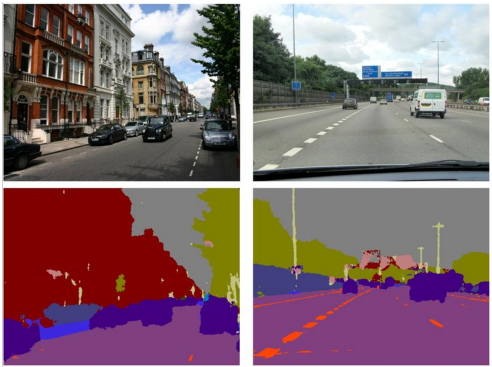
Early Detection of Prostate Cancer



Korevaar, S., Tennakoon, R., Page, M., Brotchie, P., Thangarajah, J., Florescu, C., Sutherland, T., Kam, N.M. and Bab-Hadiashar, A., 2021. Incidental detection of prostate cancer with computed tomography scans. Scientific Reports.

7 News clip

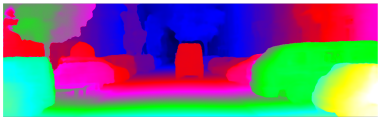
SegNet



https://mi.eng.cam.ac.uk/projects/segnet/

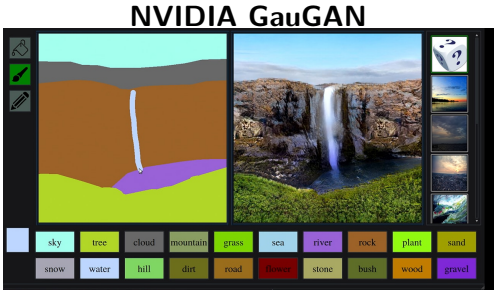
Depth Estimation - Autonomous Navigation

Image



Distance from camera or Depth

Chuah, W., Tennakoon, R., Hoseinnezhad, R. and Bab-Hadiashar, A., 2021. Deep Learning-Based Incorporation of Planar Constraints for Robust Stereo Depth Estimation in Autonomous Vehicle Applications. IEEE Transactions on Intelligent Transportation Systems.

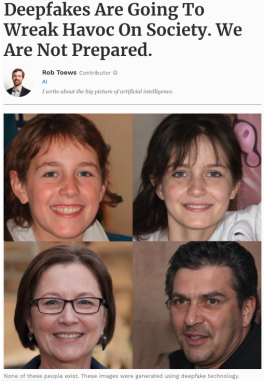


GauGAN: Changing Sketches into Photorealistic Masterpieces

Interactive Demo

Computer Vision - Other

- Object Detection
- Style Transfer
- Image synthesis
- 3D point cloud analysis and scene understanding
- Many more ...



Forbes article on Deep Fake
It's Getting Harder to Spot a Deep Fake Video

- Translation
- Speech recognition
- Transcription
- Speech synthesis

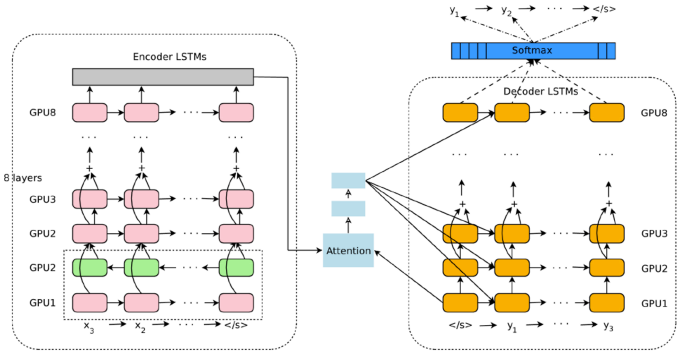


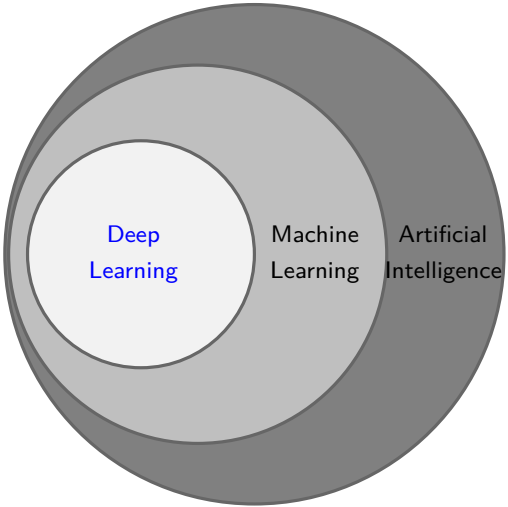
Image: Googles Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Many Other Fields

- Deep learning for biology
- Applications of machine learning in drug discovery and development
- Deep Learning for Physical Sciences - Radio astronomy, Jet Physics, Quantum physics.
- Weather Forcasting

Outline

Learning Algorithms



Deep learning is a specific kind of machine learning. To understand deep learning well, one must have a solid understanding of the basic principles of machine learning

ML: Machine learning is the field of study that gives the Ability to learn without being explicitly programmed - Arthur Samuel (1959).

More technically: "A computer program is said to learn:

- Some class of **tasks T**
- From **experience E**, and
- **Performance measure P**

If its performance at tasks in T, as measured by P, improves with experience E".

- Tom Mitchell (1998)

The Task can be expressed an **unknown target function**:

$$\mathbf{y} = f(\mathbf{x})$$

- Attributes (features) of the task: $\mathbf{x} \in \mathbb{R}^d$
- Unknown target function: $f(\mathbf{x})$
- Output of the function: $\mathbf{y} \in \mathbb{R}^c$

ML finds a Hypothesis (model), $h(\cdot)$, from hypothesis space $\mathcal{H}$, which approximates the unknown target function.

$$\hat{\mathbf{y}} = h^*(\mathbf{x}) \approx f(\mathbf{x})$$

The (optimal) hypothesis is learnt from the Experience. The hypothesis generalises to predict the output of instances from outside of the Experience.

The Experience is typically a data set, $\mathcal{D}$, of values

$$\mathcal{D} = \left\{ \left(\mathbf{x}^{(i)}, f\left(\mathbf{x}^{(i)} \right) \right) \right\}_{i=1}^{N} \quad \triangleright \text{ (Supervised learning)}$$

- Attributes (features) of the task: $\mathbf{x}^{(i)} \in \mathbb{R}^d$
- Output of the function (**Target**): $\mathbf{y}^{(i)} = f\left(\mathbf{x}^{(i)} \right)$

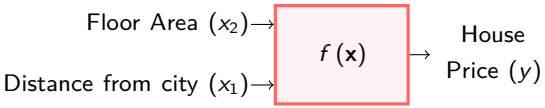
What does success look like? To evaluate the abilities of a machine learning algorithm, we must design quantitative measure of its performance.

$$\text{We like to measure:} \quad h^*(\mathbf{x}) \approx f(\mathbf{x})$$

The Performance is typically numerical measure that determines how well the hypothesis matches the experience. Note, the performance is measured against the experience NOT the unknown target function!

Usually we are interested in how well the machine learning algorithm performs when deployed in the real world - unseen data. We therefore evaluate these performance measures **using a test set** of data that is separate from the data used for training the machine learning system.

Example: Linear Regression



We need to design an algorithm that will improve the **weights**, w, in a way that reduces MSE of testset when the algorithm is allowed to gain experience by observing a training set.

One intuitive way of doing this is just to minimize the mean squared error on the training set.

How can we find the w which minimize the performance measure (Mean Squared Error) on train data?

Hypothesis (Model):

$$\hat{y}^{(i)} = h\left(\mathbf{x}^{(i)}\right) = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)}$$

Hypothesis space: $\mathcal{H} \in$
All possible combinations of (w_0, w_1, w_2)

Experience:

$$\mathcal{D} = \left\{ \left(\left[x_1^{(i)}, x_2^{(i)} \right], y^{(i)} \right) \right\}_{i=1}^{N}$$

Performance Measure:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - \hat{y}^{(i)} \right)^2$$

performance is measured over the test set.

Common Machine Learning Problems

- **Classification:** Specify which of k categories some input belongs to.
- **Regression:** Predict a numerical value given some input.
- **Anomaly detection:** Observes a set of events and flags some of them as being unusual or atypical.
- **Synthesis and sampling:** Generate new examples that are similar to those in the training data.
- **Denoising:** Predict a clean example x from its corrupted version.
- **Machine translation:** Given an input consisting of a sequence of symbols in some language, convert this into a sequence of symbols in another language.
- ...

Building a Machine Learning Algorithm

Nearly all machine learning algorithms can be described with the following fairly simple recipe:

- Dataset
- Cost function (Objective, loss)
- Model
- Optimization procedure

The first step in solving a ML problem is to **analyse the data and task** to identify the above components.

Design Choices:

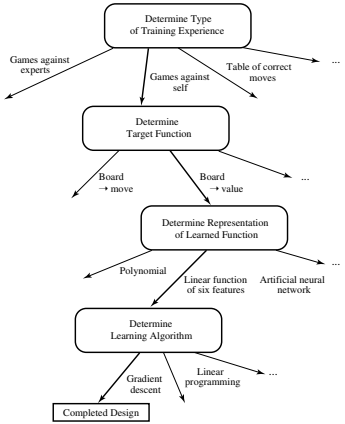


Image: Tom Mitchell, "Machine Learning", 1997.

The central challenge in machine learning is that our algorithm must perform well on new, previously unseen inputs (not just those on which our model was trained). The ability to perform well on previously unseen inputs is called generalization.

- Generalization error is related to the true error of a hypothesis (cannot be measured).
- The generalization error of a machine learning model is typically estimated by measuring its performance on a **test set** collected separately from the training set.

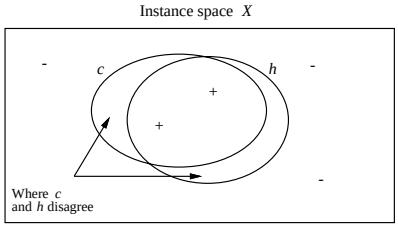


Image: Tom Mitchell, "Machine Learning", 1997.
Here $c := f$ is the unknown target function.

How can we affect performance on the test set when we can observe only the training set?

If the training and the test set are collected arbitrarily, there is indeed little we can do. In practice, the learning algorithm does not actually find the best function, but merely one that significantly reduces the training error.

However, If we are allowed to make some assumptions about how the training and test set are collected, then we can use the field of *statistical learning theory* to obtain some answers.

Assumptions about the data-generating process:

- Each dataset are independent from each other.
- Training set and test set are identically distributed. We call that shared underlying distribution the data-generating distribution (p_{data}).

Example: Data-generating Process

We are interested in **classifying traffic signs in Melbourne Australia**. In order to train ML model, we obtained data using the following process: Randomly pick a traffic sign (signID) and image it three times (instanceID) from slightly different angles, crop the traffic sign and save it as "signID_instanceID.jpg".

Discuss the following two scenarios:

1. All images with instanceID equal to 1 is selected as the test set and, the remaining images are used as training set.
2. All images are used for training and "German traffic sign dataset" in Kaggle is used as test set.

Under-fitting and Over-fitting

The factors determining how well a machine learning algorithm will perform are its ability to

- Make the training error small.
- Make the gap between training and test error small (generalization)

Two main challenges in machine learning:

- **Under-fitting:** Both training and test error is large. The model does not have enough capacity to capture the target function.
- **Over-fitting:** Test error is large but the training error is small (large gap). Learning vs. memorizing.

$*$The expected test error is greater than or equal to the expected value of training error.

Under-fitting and Over-fitting

We can control whether a model is more likely to over-fit or under-fit by altering its **capacity**.

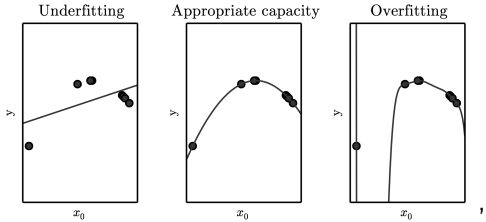


Image: Goodfellow 2016

One way to control the capacity of a learning algorithm is by choosing its hypothesis space $\mathcal{H}$. Linear $\leftrightarrow$ polynomial.

Machine learning algorithms will generally perform best when their capacity is appropriate for the true complexity of the task they need to perform and the amount of training data they are provided with.

Generalization Gap

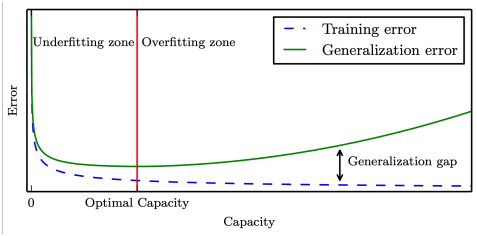


Image: Goodfellow, 2016.

Simpler functions are more likely to generalize, but a sufficiently complex hypothesis is needed to achieve low training error.

$*$ It is possible for the model to have optimal capacity and yet still have a large gap between training and generalization errors. In this situation, we may be able to reduce this gap by gathering more training examples.

Occam's razor: Among competing hypotheses that explain known observations equally well, we should choose the "simplest" one.

Statistical learning theory provides various means of quantifying model capacity. e.g. Vapnik-Chervonenkis dimension (VC-dimension)

- Discrepancy between training error and generalization error is bounded from above by a quantity that grows as the model capacity grows but shrinks as the number of training examples increases.
- These bounds are rarely used in practice when working with deep learning algorithms.
- This is because the bounds are often quite loose and it can be quite difficult to determine the capacity of deep learning algorithms.

Regularization

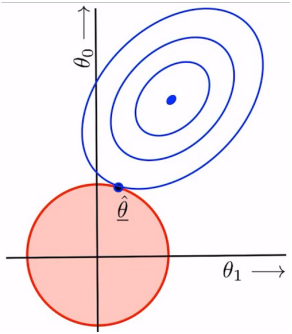
Another method to control whether a model is more likely to over-fit or under-fit is regularization.

Regularization is any modification made to a learning algorithm that is intended to reduce its generalization error but not its training error.

We do so by building a set of preferences (biases) into the learning algorithm. e.g. Preference one solution over another in its hypothesis space.

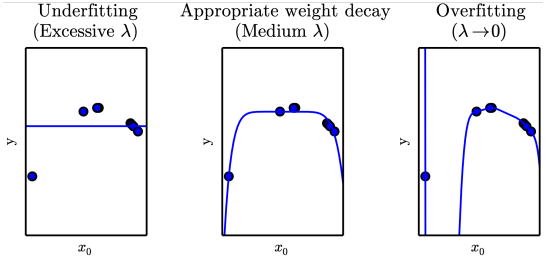
For example, we can modify the training criterion for linear regression to include weight decay.

$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (y - \hat{y})^2 + \lambda w^\top w$



Hyperparameters and Validation Sets

Most machine learning algorithms have hyperparameters - settings that we can use to control the algorithm's behavior.



It is not appropriate to learn hyperparameters on the training set. e.g. If hyperparameters that control model capacity are learned on the training set, it would always choose the maximum possible model capacity, resulting in over-fitting.

$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (y - \hat{y})^2 + \lambda w^\top w$

To solve this problem, we need a **validation set**.

Cross-Validation

Dividing the dataset into a fixed training set and a fixed test set (Hold-out validation) can be problematic if it results in the test set being small.

Cross validation provides an alternative to use all the examples in the estimation of the mean test error, at the price of increased computational cost.

In CV training and testing computation are repeated on different randomly chosen subsets or splits of the original dataset.

In k-fold cross-validation procedure, partition the dataset is generated by splitting it into k non-overlapping subsets.

Hold-out: Dividing the dataset into a fixed training set and a fixed test set.

K-fold-CV: partition the dataset is generated by splitting it into k non-overlapping subsets.

Independent (i.i.d) test set to make the final evaluation.

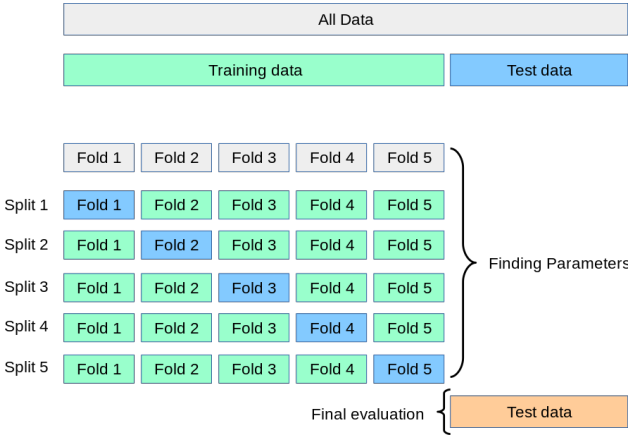


Image: scikit-learn

Review Questions

- Is the performance evaluated over training examples? why?
- What are the key ingredients of a general ML recipe?
- What is generalization-gap?
- If a model shows low train error and high test error is it over-fitting or under-fitting?
- What are the methods that can be used to control whether a model is more likely to over-fit or under-fit?
- Can we use train error to identify the best value for the regularization parameter? Why?
- When will you chose hold-out validation over cross-validation.

Example


Parcel Post | Prepaid
Within Australia
Tracking*
2+ business days*
Max. weight 5kg
Recipient mobile
Company name
Contact name
Traditional place name (if known)
Street address or PO Box number
Suburb or town, State, Postcode
To:
Mr A Sample
Wurundjeri
1 Sample Street
MELBOURNE 3000
Aviation Security and Dangerous Goods Declaration*
Sign here:
*I, the sender, acknowledge that this article may be carried by air and will be subject to aviation security and screening procedures, and I declare that it does not contain any dangerous goods, explosives or incendiary devices. A false declaration is a criminal offence. If the declaration is not signed this article cannot be carried by air.
NOT FOR RESALE
Parcel Post
For use with domestic parcels only
63256424794097
63256424794097
Australia Post