Data and Learning

Dr. Johan Hagelbäck

johan.hagelback@lnu.se

http://aiguy.org

Data and Data representation

Example (instance)

- Data consists of inputs and outputs
- Each set of inputs and outputs is an independent example (instance) of the data
  - One image of a cat
  - Test results for a patient with diabetes
  - …
- The inputs consists of one or more (often many) values, called attributes (features)
- Output consists of one or more values, called classes (categories)

Attributes (features)

- Attributes are variables describing an example of the data
- They can be of different types:
  - Numbers: integers or floats
  - Nominal (categorical): a finite set of discrete categories:
    car, boat, plane, the digit 2, the letter A, …

Common datasets

Iris dataset

- Learns to distinct between three subspecies of the iris flower based on measurements on the flowers

- Four numerical attributes
- Three categories
- 150 examples

Iris Versicolor Iris Setosa Iris Virginica

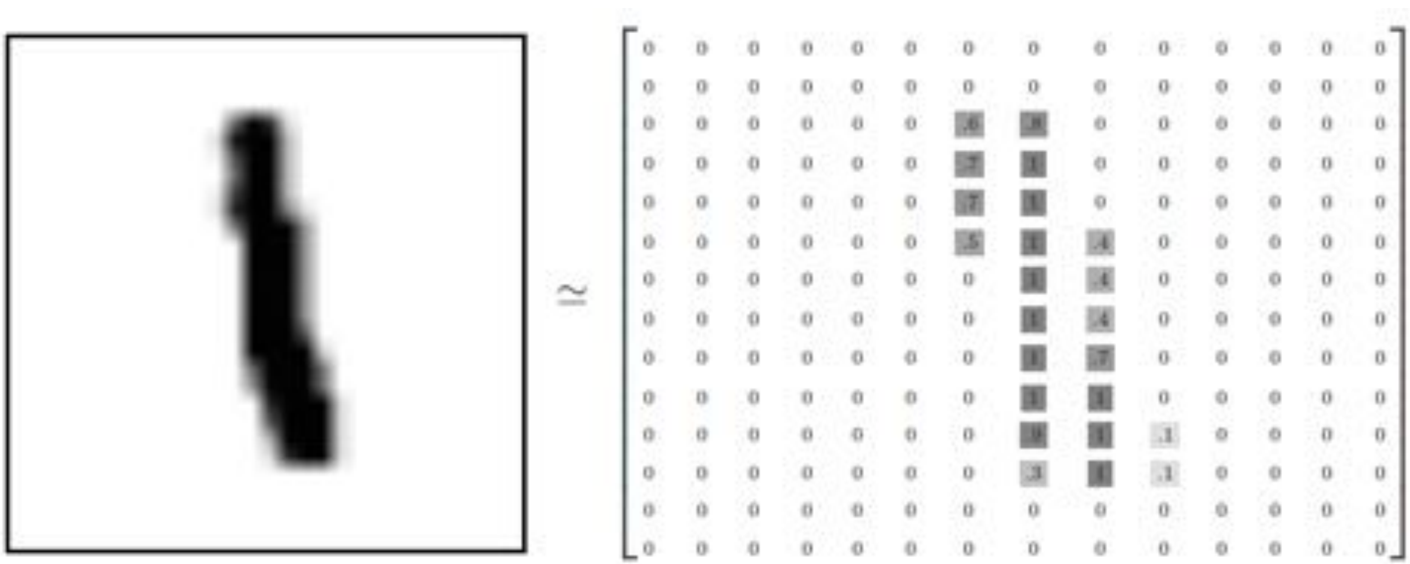
Iris dataset

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
7.0	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.9	3.1	4.9	1.5	Iris-versicolor
6.3	3.3	6.0	2.5	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica
7.1	3.0	5.9	2.1	Iris-virginica

MNIST dataset

- MNIST is a dataset containing images of handwritten digits
- It has a training set of 60000 examples and a test set of 10000 examples
- There are, of course, 10 categories (0, 1, … , 9)
- Each image is 28x28 pixels in grayscale

MNIST dataset



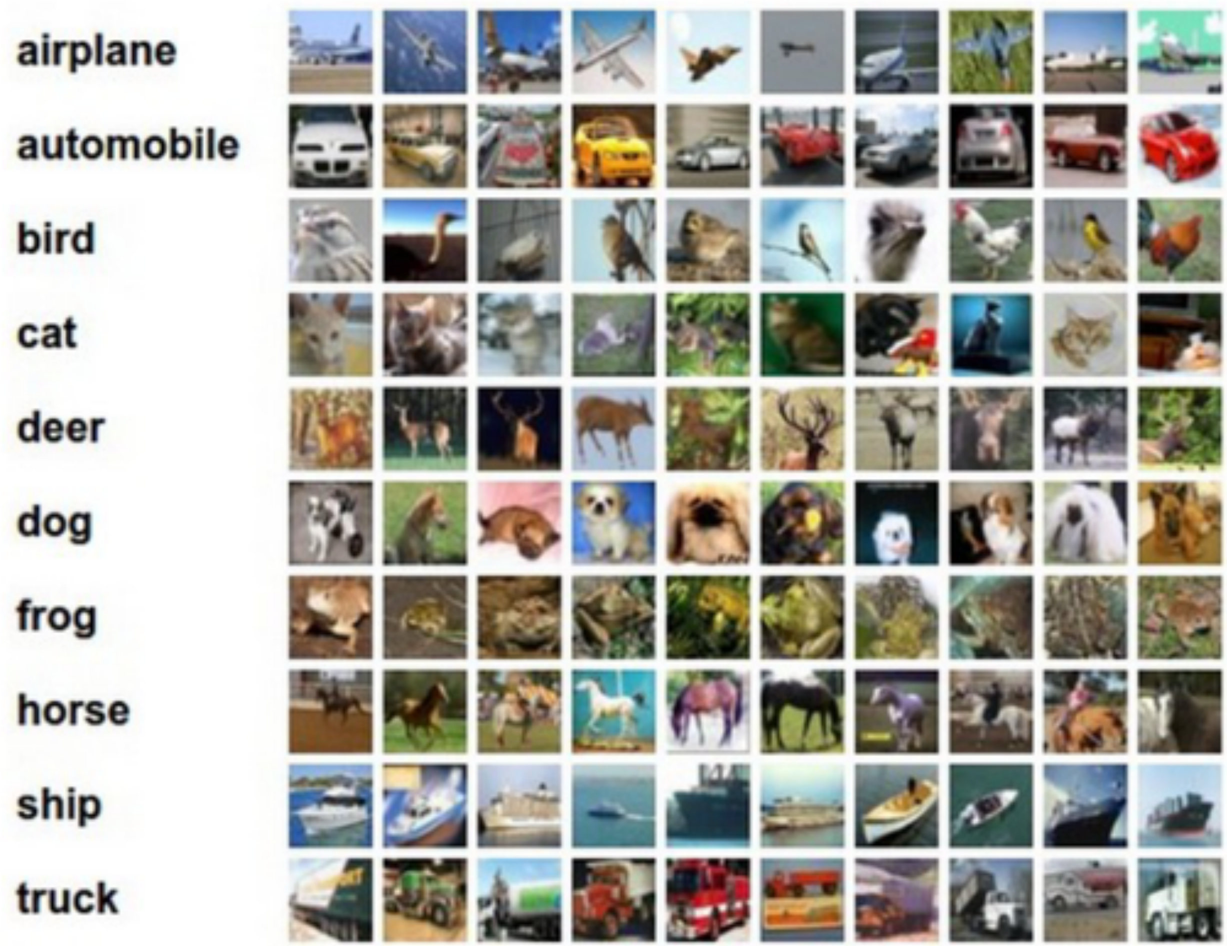
MNIST dataset

- Each image can be seen as a 28x28 matrix of float values
- Each value represents the darkness of a pixel:
  - 0.0: white
  - 1.0: black
- Sometimes values are integers between 0 and 255 instead
- To use it we flatten the array to a 28x28 = 784 input vector:
  - [0.0, 0.0, 0.0, 0.0, 0.1, 0.1, 0.15, … , 0.0, **1**]
- MNIST is an image recognition problem

CIFAR-10 dataset

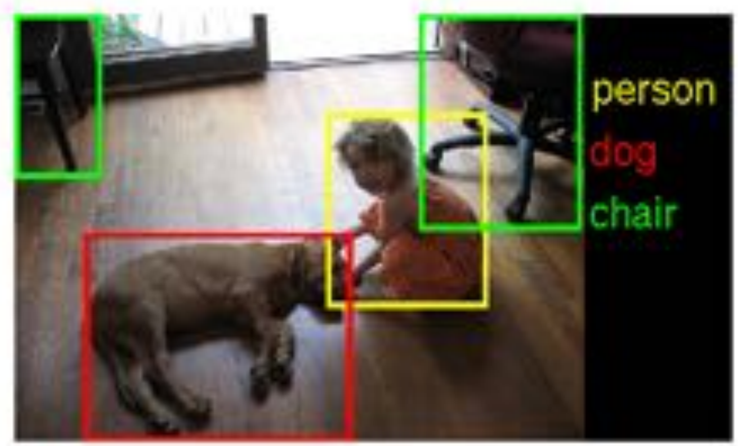
- A much more complex image recognition problem is CIFAR-10
- It consists of 60000 images (50000 for training, 10000 for testing) of size 32x32 pixels
- It has 10 categories:
  - airplane, automobile, bird, …
- The input vector must be flattened to 32x32 pixels times 3 color channels (RGB):
  - 32x32x3 = 3072 input values

CIFAR-10 dataset

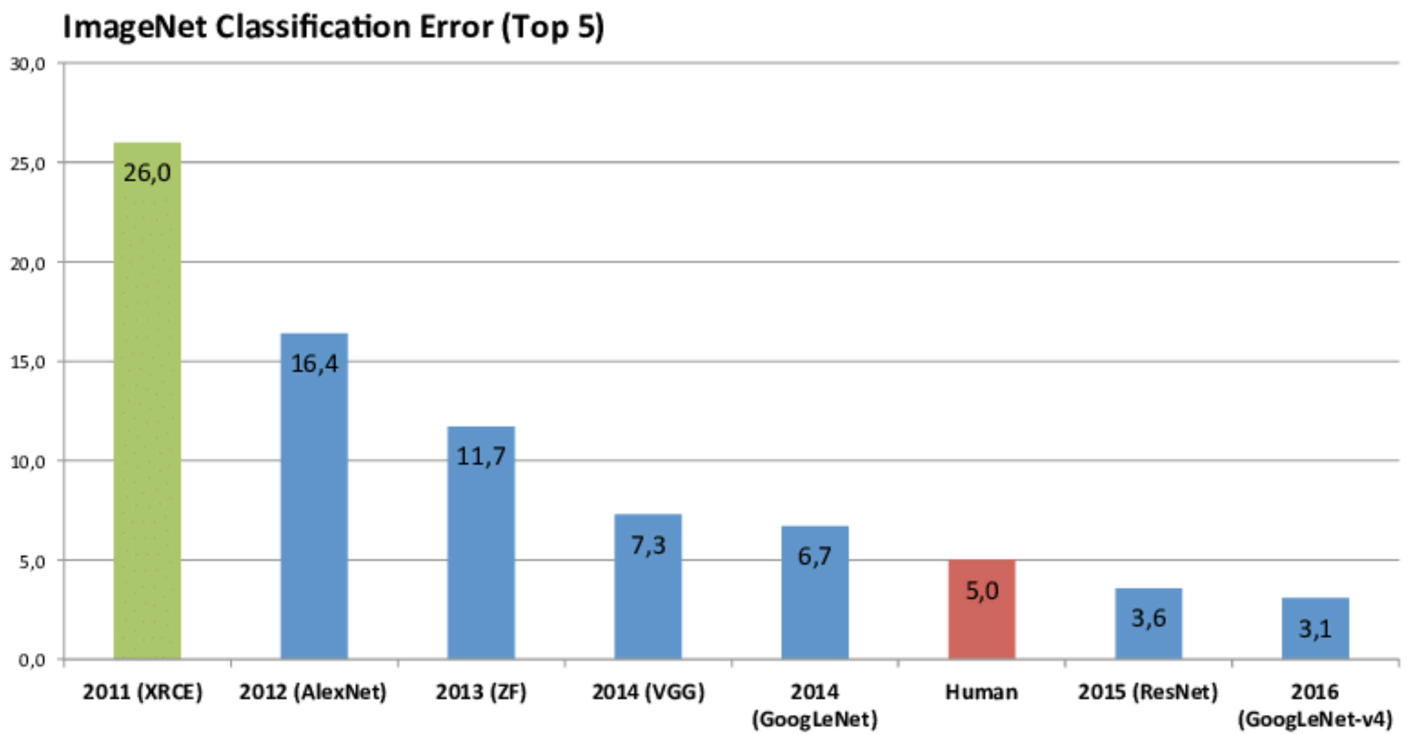


ImageNet challenge

- The ImageNet challenge is an annual contest for image recognition
- The training dataset consists of 1.2 million images and 1000 possible categories
- The validation set for the challenge is a random subset of 50000 images
- Images can differ in size, but in average the resolution is 482x415 pixels
- Two tasks:
  - Recognition: what's in an image
  - Localization: where an object is



ImageNet challenge



Types of learning problems

Supervised learning

- Algorithms are presented with example inputs and known outputs:

Input 1	Input 2	Output
1	3	4
2	1	3
3	5	8

- The learning task is to map the inputs to the output
- The output can consist of categories (classification) or a continuous number (regression)

Unsupervised learning

- In contrast to supervised learning, no known output is given
- The algorithms are left on their own to find patterns or structures in the input data
- An example is to group news articles discussing similar topics together (called *clustering*)

Reinforcement learning

- In reinforcement learning, systems learn from trial and error
- The system executes an action in its environment, and is given feedback on how well it worked out:
  - If the action was a success, a positive reward is given
  - If the action was a failure, a negative reward (punishment) is given
- Over time, the system learns what actions are successful in its environment
- Similar to how small children learn
- An example is creating a robot that can learn how to best navigate in an environment

Training and Validation

Training and Validation

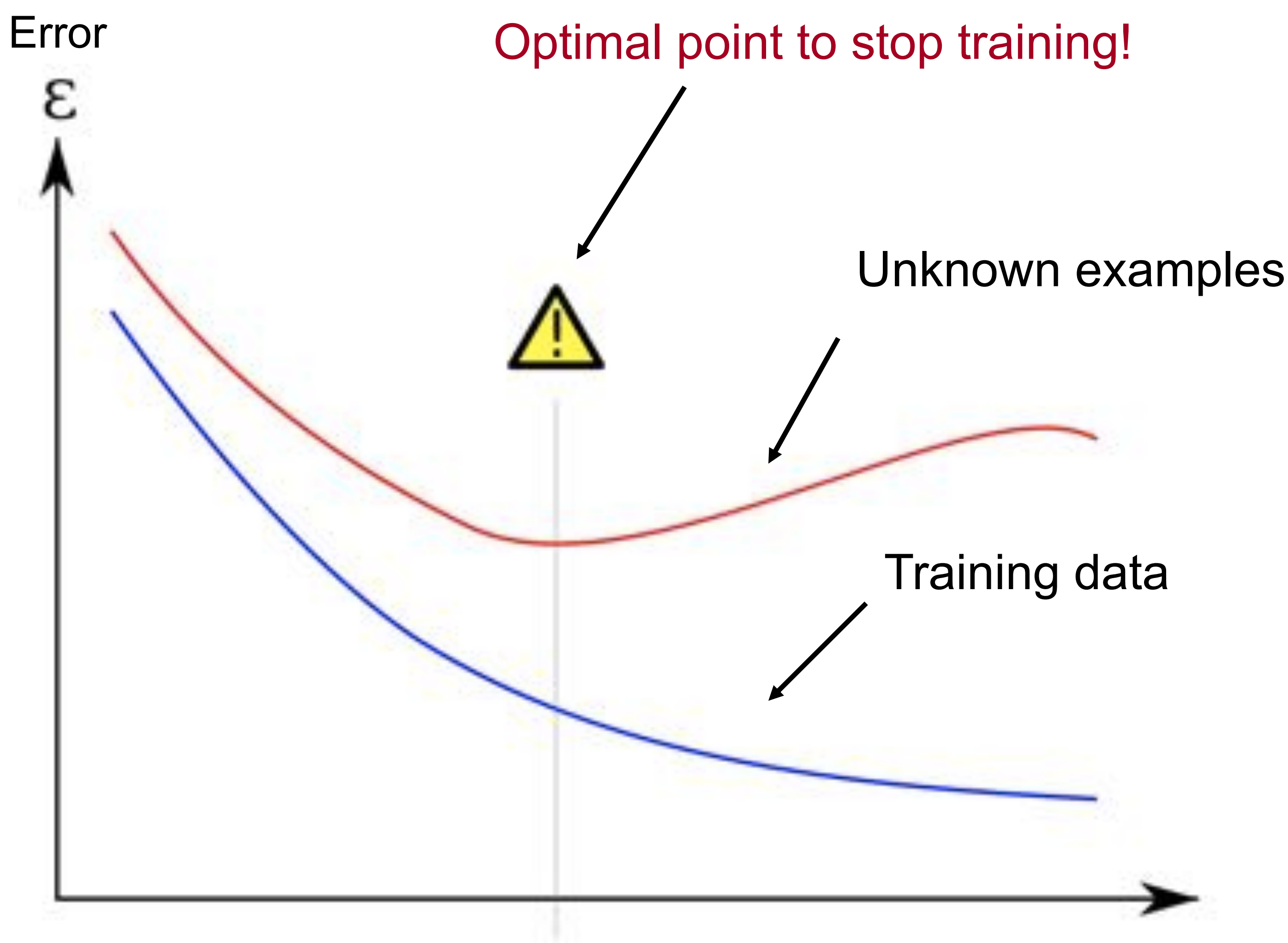
- The machine learning algorithm is trained on a dataset
- The dataset consists of a number of *examples* with known output
- The trained algorithm is called a *model*
- The model is used to classify new examples
- We can check how good the model is by calculating the *accuracy*
- Accuracy means the percentage correctly classified examples:

Accuracy: 95.33% (143/150 correctly classified)

Training

- If we use the same dataset for both training and testing, we don’t know how good the model is on new, unknown examples
- We only know how good the model is at mapping known examples to output
- How good the model is on unknown examples is called its *generalization ability*
- If we train too much on the data, the accuracy on the data increases but we can get worse accuracy on unknown examples
- This is called overfitting:

Overfitting



http://en.wikipedia.org/wiki/Overfitting_(machine_learning)

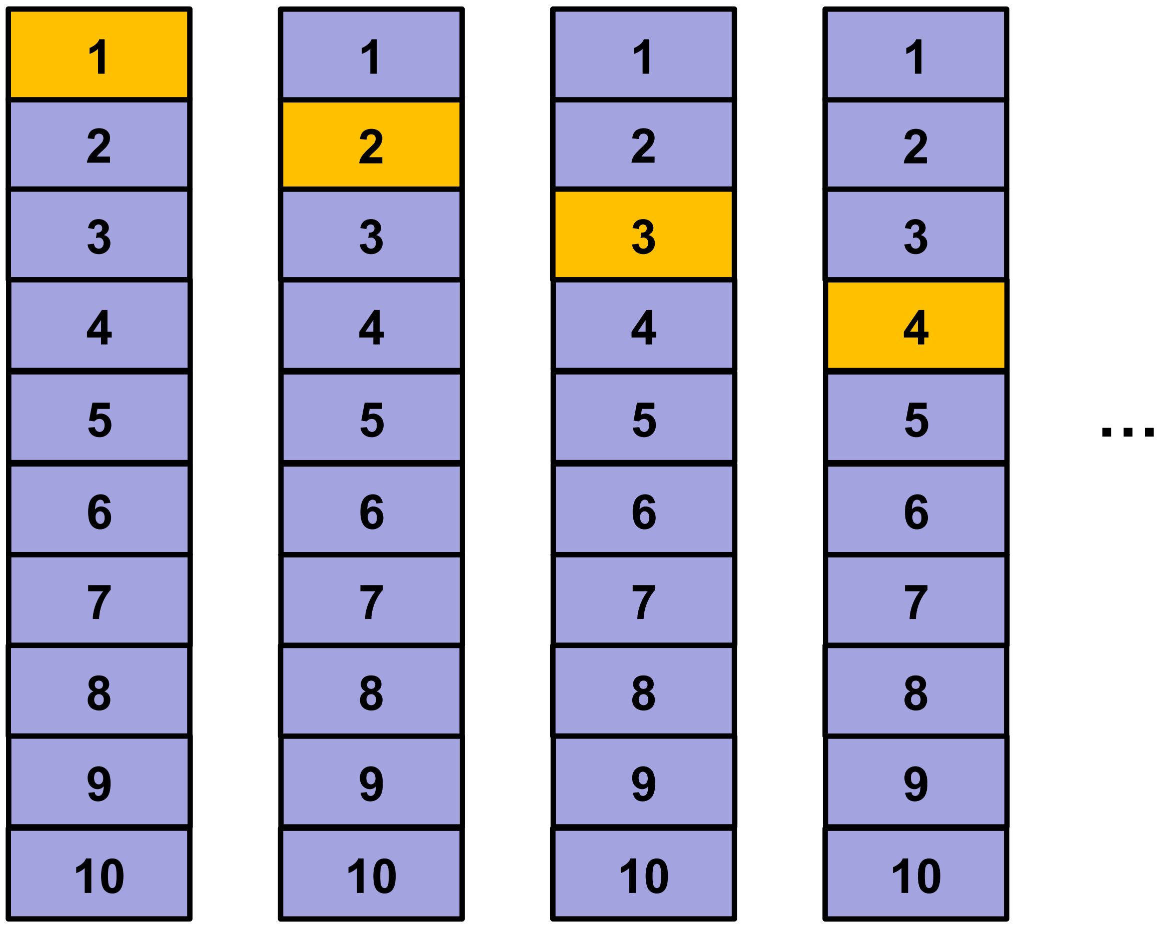
Using separate datasets

- One way to improve the generalization ability and reduce overfitting is to use two datasets
- The first, larger dataset is used to train the model
- It typically contains around 70-80% of the examples
- The rest is used to test the model performance
- We select the model candidate with the best performance on the test dataset

Cross-validation

- In cross-validation, the dataset is divided into a number of buckets of roughly equal size
  - Common is to use 3, 5 or 10 buckets
- The model is trained on 9 buckets, and tested on the last bucket
- In the next iteration, another bucket is used for testing and the rest for training
- We repeat until all buckets have been used for testing, and calculate an average accuracy

Cross-validation



Train-test split or cross-validation?

- Cross-validation is a better measurement of the model's generalization ability, but takes longer time than train-test split
- Use CV if it doesn't take too long time
- Often we use cross-validation to fine-tune the model when training on the training dataset, and use the test dataset for final validation of the model performance

Good or bad result?

- How good the accuracy is depends on how many possible categories the output can have
- An accuracy of 50-60% on a binary classification problem (2 categories) is hardly better than random chance!
- The same accuracy can however be rather good if we have 25 possible categories!

ZeroR

- We can use the ZeroR classifier as baseline when comparing the results for different classifiers
- ZeroR simply classifies all examples as the most frequent category in the dataset
- ZeroR has an accuracy of 33.3% on the iris dataset, since we have an equal amount of examples from the three categories

Other Performance Metrics

Accuracy

- Accuracy is one of the most common performance metric for classification
- It means the percentage correctly classified instances
- If we have 150 examples in the test dataset, and 138 of them are correctly classified
- … we calculate accuracy as 138/150 = 92%

- The problem is that we only know the number of errors, not what type of errors the model produces

Type of errors

- TP = true positives
  - we classify a correct example as correct
  - cat is classified as cat
- TN = true negatives
  - we classify an incorrect example as incorrect
  - dog is classified as not cat

- FP = false positives (type 1 error)
  - we classify an incorrect example as correct
  - dog is classified as cat
- FN = false negatives (type 2 error)
  - we classify a correct example as incorrect
  - cat is classified as non cat

Type of errors

- Depending on the task, knowing if an error is of type 1 or 2 can be important
- In for example an earthquake detection system we don't want to start the alarm if there is no earthquake, spreading fear among people (type 1 error)
- It is better that we miss an early sign, and most likely detect the earthquake later
- We aim to minimize type 1 errors

Type of errors

- Performance metrics that takes the type of errors into consideration are *Precision* and *Recall*
- Precision is calculated for each category, and measures how often the model is correct when classifying examples as belonging to the category:

$$Precision = \frac{TP}{TP + FP}$$

- Recall is also calculated for each category, and measures the amount of examples that belong to the category that are correctly classified:

$$Recall = \frac{TP}{TP + FN}$$

F1-score

- Precision and Recall is combined into a single performance metric called F1-score (sometimes just F-score):

$$F1 = 2 \cdot \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Confusion Matrix

- A confusion matrix shows the correct and incorrect classifications for each category:

Confusion Matrix

A	B		
48	2	A	= Category A
4	46	B	= Category B

Example: Iris dataset

Accuracy: 96.67% (29/30 correctly classified)

Confusion Matrix:

	[0]	[1]	[2]	
[0]	13	0	0	→ Iris-setosa
[1]	0	8	1	→ Iris-versicolor
[2]	0	0	8	→ Iris-virginica

Metrics by category:

	Precision	Recall	F1 score	
[0]	1.000	1.000	1.000	→ Iris-setosa
[1]	1.000	0.889	0.941	→ Iris-versicolor
[2]	0.889	1.000	0.941	→ Iris-virginica
Avg:	0.963	0.963	0.961	

Other important characteristics

- There are other things we need to take into consideration when selecting an algorithm for a problem:
  - Training and classification time
  - Space consumption of the trained model
  - Explainability – can we understand what the model has learned?
  - Possibility of online learning – can we continue training a model with new examples without having access to all data?

Data and Learning

Dr. Johan Hagelbäck

johan.hagelback@lnu.se

http://aiguy.org

Linnæus University