Fundamentals of Artificial Intelligence

Machine Learning and Applications

Adam Dahlgren

Department of Computing Science

Umeå University

Outline

- Overview of the machine learning process
  - Looking at code together
- Examples of interesting applications
- Some notes on deep learning
- General summary of current state of machine learning

Rules during this lecture

- Whenever there is a question, at least one person must answer.
- I might lose track of space and time if interrupted
- But, whenever you do not understand, interrupt me.
  - You are most likely not alone
  - I might not understand either
  - Ask in Swedish if you are uncomfortable with English.
  - I want two volunteers that keep an eye on the chat in case I miss questions.
- Asking the right questions is as important as knowing the right answers.
- Remind me when you need a break

Introduction

Some repetition

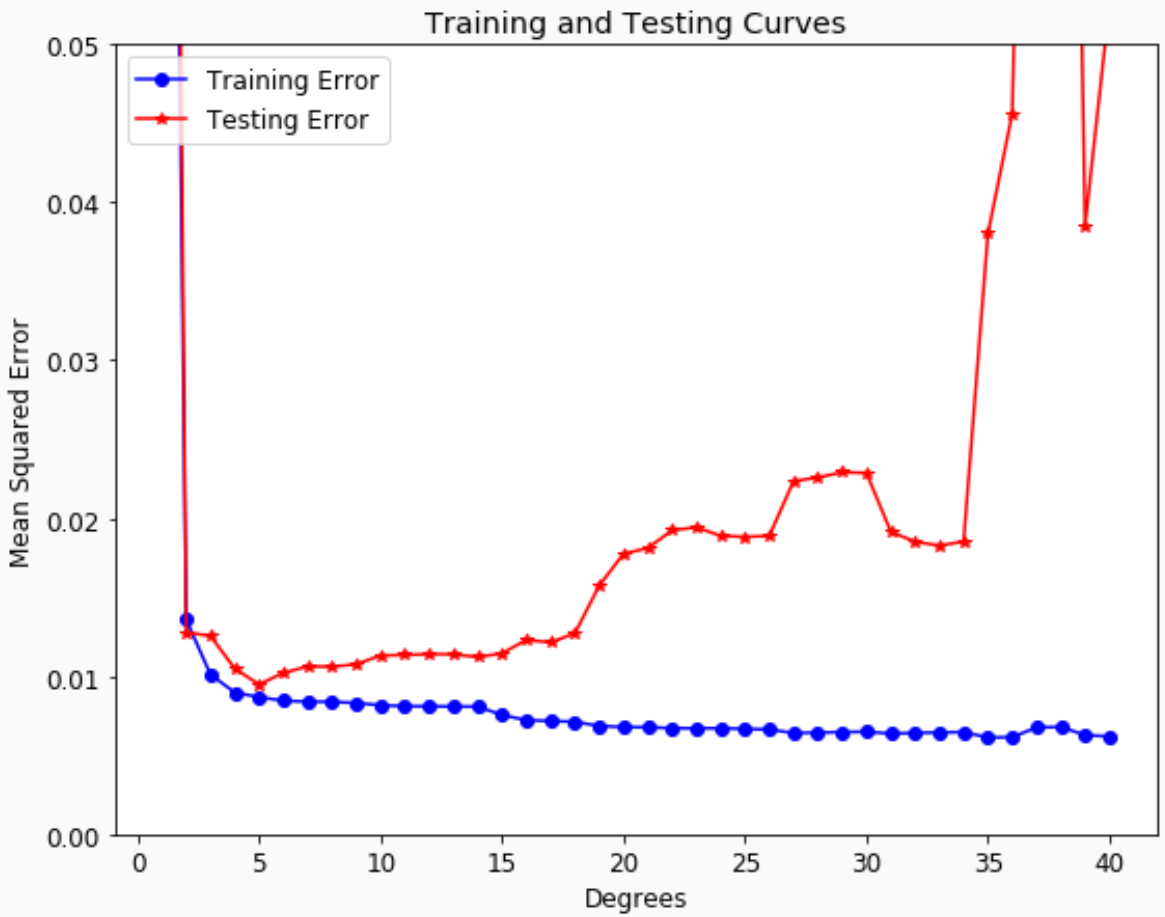


Figure 1: What is this an example of?

The Machine Learning Process

Recap

How do we define a machine learning?

A process that

- improves behaviour (performance)
- on a given task
- by experience

Can someone give me an example of another way of measuring performance than accuracy?

Recap cont'd.

Define/explain

- Supervised learning
- Hypothesis space
- Overfitting
- Underfitting
- Ockham's razor
- Frame Problem

Machine Learning Process - Overview

- It is a *highly* iterative process
- Usually requires lots of computational resources
  - GPUs
  - RAM
  - Time constraints
- Difficult to know how to improve performance
- Might be a hassle to actually fit the data and the model together
- Dependent on specialized software

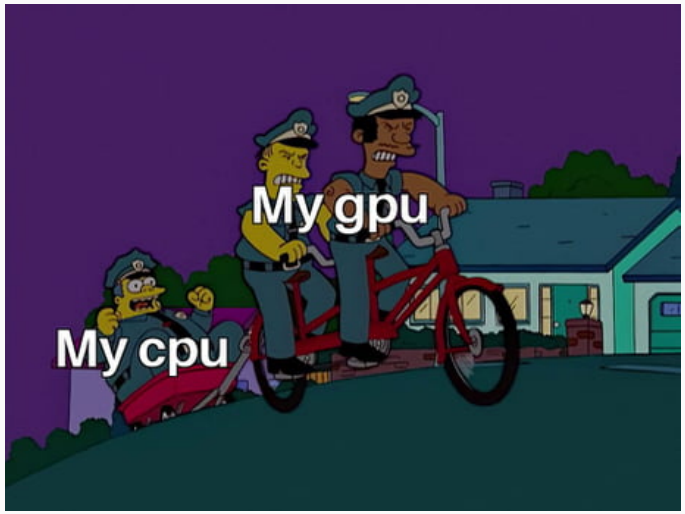


Figure 2: The usual setup.

Software

We have a couple of established (open-source) libraries that we can use

- Python is a defacto standard for machine learning
  - Wrapper of C/C++ implementations
- Tensorflow, developed by Google and optimized against Nvidias GPUs
- Keras, built on Tensorflow, more high-level, contains all standard tools
- Pytorch, developed by Facebook
- Honorable mentions
  - Scikit-learn, R, Caffe, Weka, Numpy/Scipy, Theano, et c.
- Cloud based - Amazon, Microsoft, Google, IBM, Oracle

Your best bet is probably Keras in the beginning, but it becomes trickier when you move closer to the cutting edge.

Repetition: An algorithm for learning

1. Collect a (large) set of examples
2. Divide them randomly into a training set and a test set
3. Generate a hypothesis using the training set (training our model further)
    - E.g. update the parameters of our model using gradient descent.
4. Measure the performance on examples from the test set (e.g. number of correctly classified examples).
5. Repeat until satisfactory performance is achieved.

This produces a model of the relationship between the input/output space that the classifier can use later.

In its simplest form

```
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import np_utils
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

iris = load_iris()

(x_train, x_test, y_train, y_test) = train_test_split(iris.data, iris.target, test_size=0.3)

y_train = np_utils.to_categorical(y_train, 4)
y_test = np_utils.to_categorical(y_test, 4)

model = Sequential()
model.add(Dense(8, input_dim=4, activation='relu'))
model.add(Dense(4, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()

model.fit(x_train, y_train, epochs=100)

print(''Accuracy: '', model.evaluate(x_test, y_test)[1])
```

How do we prioritize?

Figure 3: One soup (model) with too many chefs (things to tweak) requires a systematic approach.

Traning- and test data

As we've seen, the way that we split our dataset can be paramount

- Training data - do we have enough?
- Validation data - what do we validate on during training?
- Test data - is it representative of the problem? How do we compare results?

When we have lots of classes, it is important to make sure that each class is sufficiently represented in each of these sets. How can we do that?

Cross validation

Many of you will have noticed strange behaviour in your plots from the k-NN assignment.

- Why is it that the `random_state` you choose was so important?
- Give an example of how a bad selection of training-/test data could look like
- What can be done?

K-fold cross validation.

K-fold cross validation

- Instead of choosing just one random division
- We split the dataset into k parts, using each part as the test set once.
- We then report the average over all k iterations as the accuracy or error.

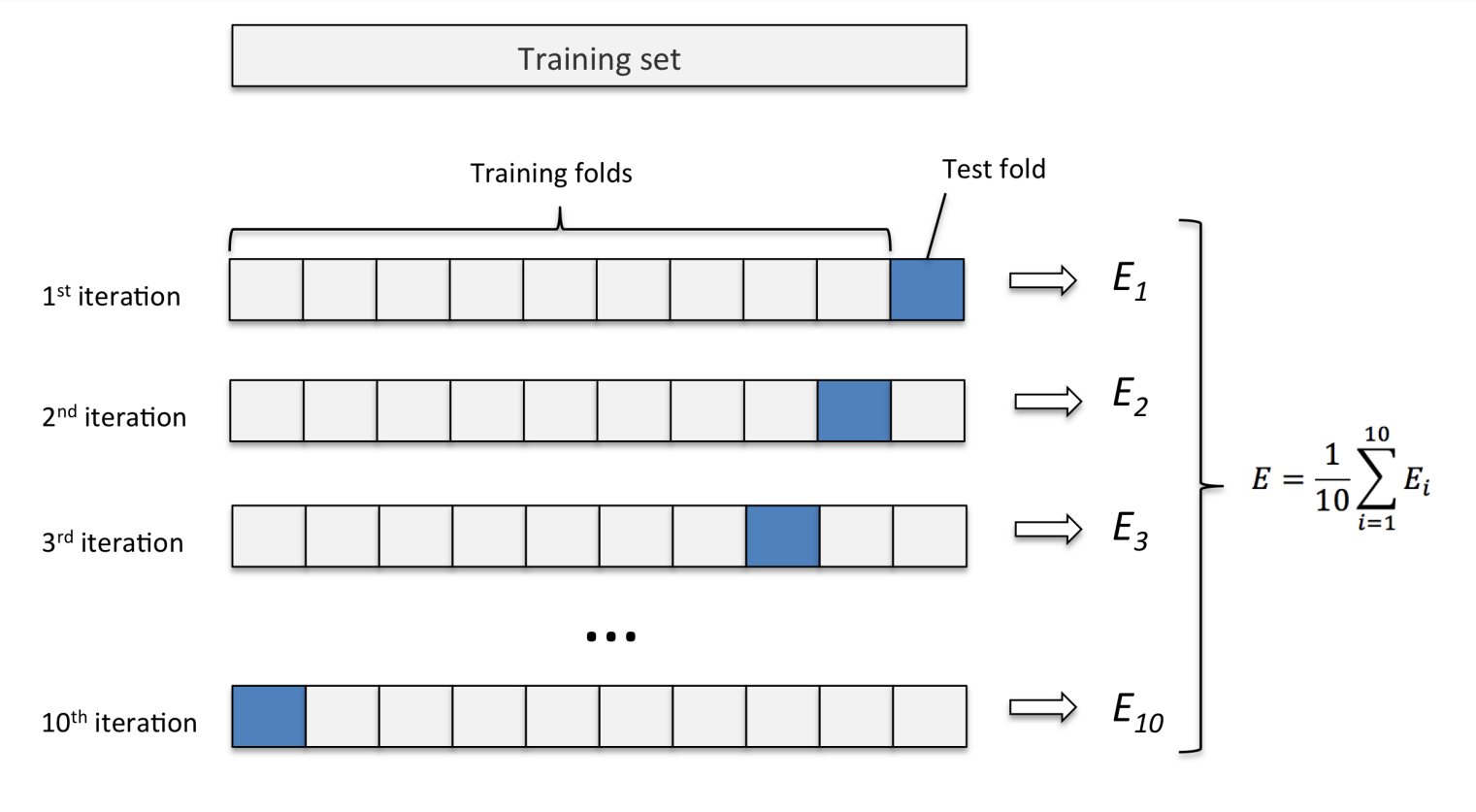


Figure 4: 10-fold cross validation. CC-license [1].

[1] http://karlrosaen.com/ml/learning-log/2016-06-20/

Traning parameters

We have some parameters that are related to the training process

- Learning rate
  - How drastic can our parameter updates be? Step size in our optimization.
- Number of epochs
  - How many times should we go through the training data?
- Batch size
  - How many samples should we look at simultanously?
  - Useful to normalize over batches [1]

Measuring performance

After each training iteration we want to measure performance

- Precision and recall can be defined per class
- Precision
  - Conceptually different from accuracy, although really similar
  - The number of correct predictions
  - Example where this is important?
- Recall
  - The fraction of total number in class correctly predicted
  - Example where this is important?
- Resources
  - RAM or disk consumption
  - Ecological footprint

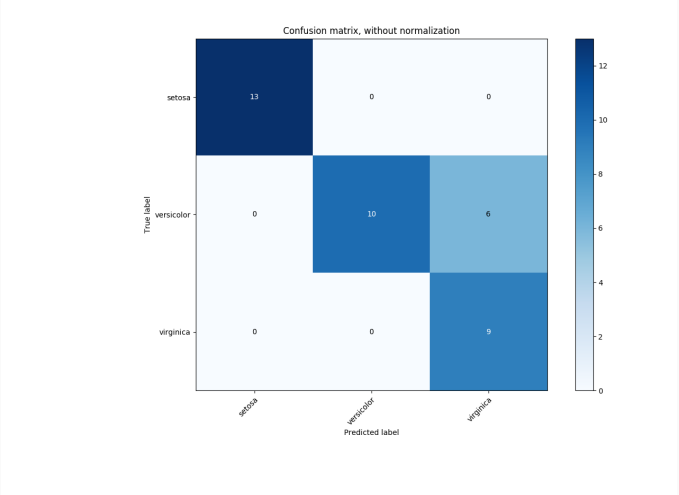


Figure 5: Example confusion matrix for Iris dataset.

Confusion matrix

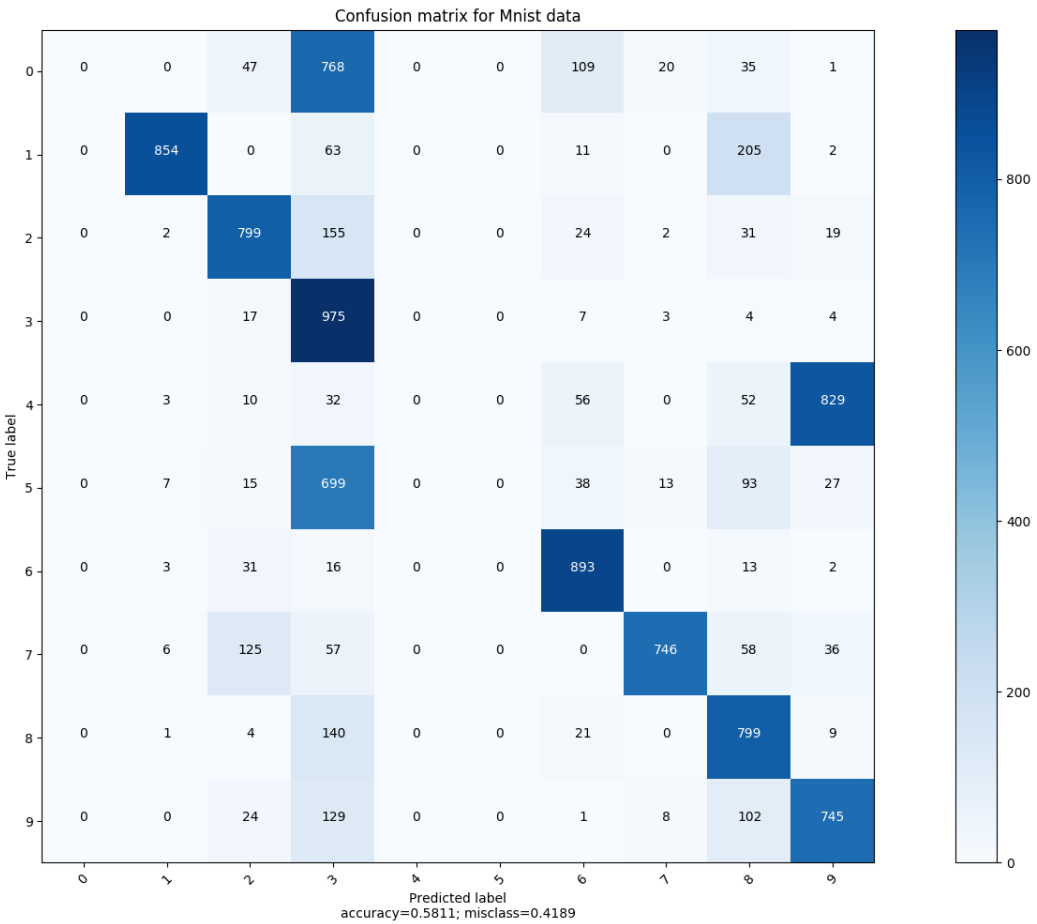


Figure 6: Confusion matrix for MNIST dataset.

When should we stop?

Discussion: at what point should we stop training our model?

- Could we iterate until below a certain error?
- Always a fixed number of epochs
  - Can someone give me a reason why this can be bad?
- Early stopping
  - We want to avoid overfitting
  - We stop training after a number of training cycles has increased our test error

Complexity of model

Can we increase the complexity of the model somehow?

- We can only vary the k in k-NNs
- In regression we could choose a higher-degree polynomial
- In neural networks we can increase the number of nodes

Deep learning

Deep Learning

- Neural network comeback in seminal paper from 2012 on Convolutional Neural Networks [2].
- Fueled by advancements in hardware, it became possible to train much larger models
- Bengio, Hinton, and LeCun were awarded the Turing award for their work in this field
- Common saying: almost certainly the next best hammer to all nails

Object detection and transfer learning

- Object detection is more or less a solved problem, with better-than-human performance [3]
- Has opened up for more complex tasks such as Visual Question Answering
- Transfer learning
  - Standard is to initialize our model with random values
  - With transfer learning we initialize with the parameters from an already trained model
  - The idea is that we only need to finetune to our problem
  - The previous training (for a different problem) should have picked up some general structure we can reuse
  - Especially useful if we have little data
- Transfer Learning common with models trained on ImageNet [4]

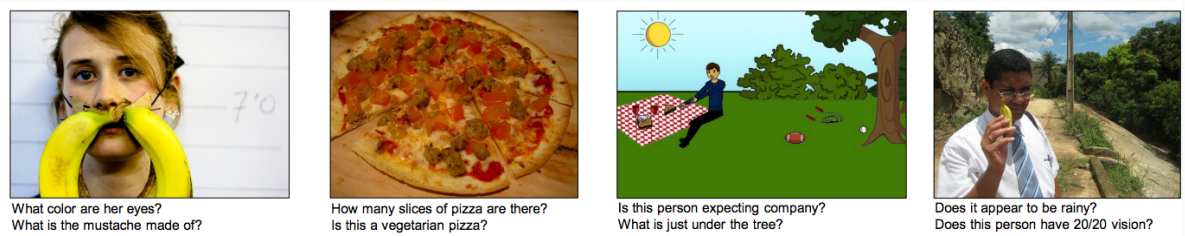


Figure 7: Example of tasks we are able to solve with deep learning.

Generative models

So far we have mostly looked at supervised learning

- Can you give me three reasons why this is insufficient?
  - Labeling data is expensive, time consuming, and error prone
  - Unsupervised learning to understand underlying (*hidden*) structure
- Here we want to generate new examples given previous data
  - Generating new viewpoints
  - Completely new examples
  - Autocomplete in images (colorization, adding/removing segments)
- Many examples of really impressive applications in recent years

pix2pix [5] - GAN example application

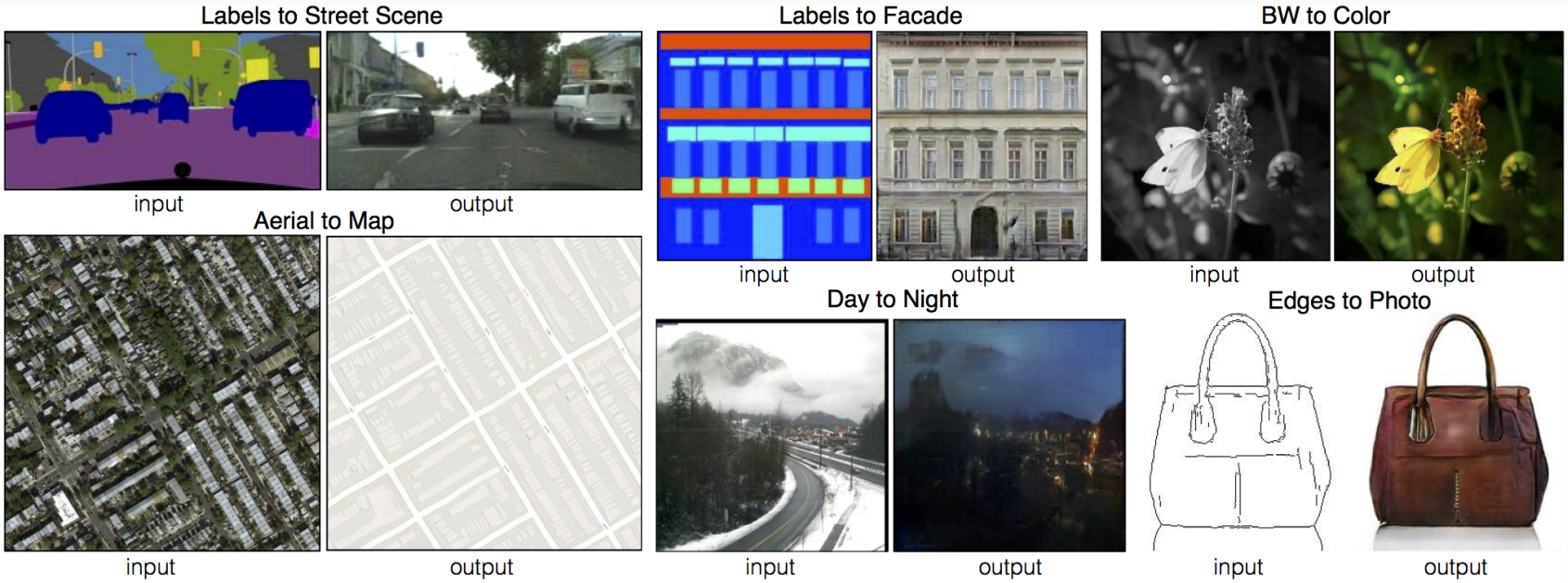


Figure 8: Example application built on GANs. Generate a realistic rendering of a given sketch. Take a look at Github [2].

[2] https://phillipi.github.io/pix2pix/

Generative Adversial Networks

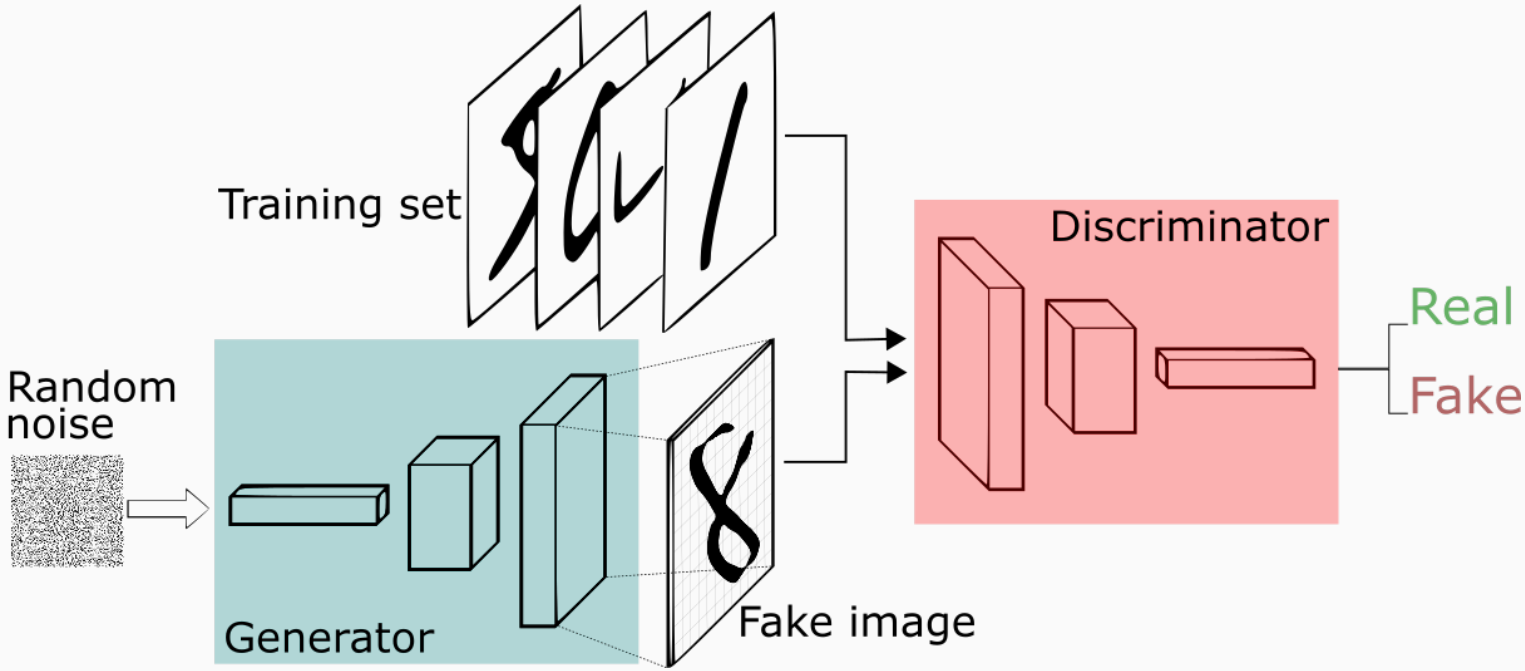


Figure 9: Overview of the general GAN architecture. Image credit: Thalles Silva.

Example application with CycleGANs

CycleGANs introduce the idea that in image-to-image translation we want to do style transfer, not just generate good output.

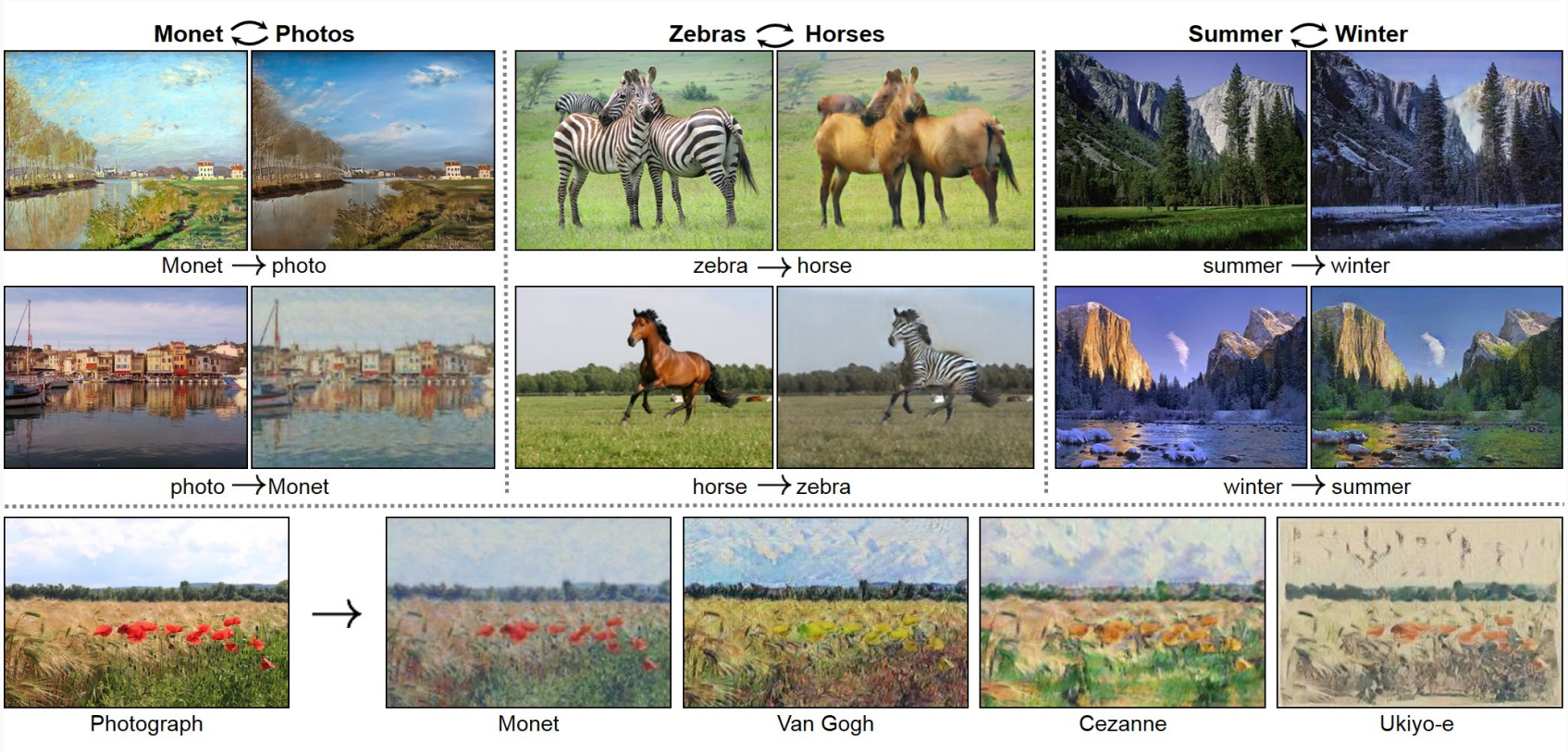


Figure 10: Image-to-image translation using CycleGANs [5].

And sometimes they do

https://thispersondoesnotexist.com/

Figure 11: Artwork generated using GANs - sold for $432,500 in 2018 [3].

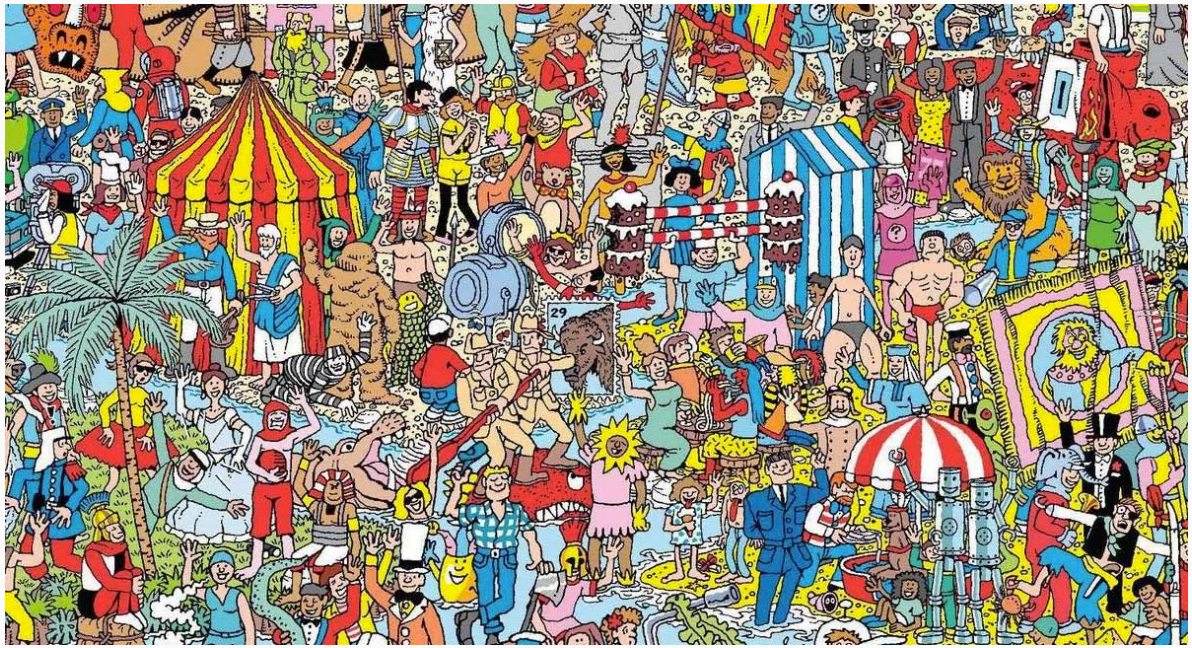
[3] https://edition.cnn.com/style/article/obvious-ai-art-christies-auction-smart-creativity/index.html

The relationship between GOFAI and Deep Learning

Kahneman's System 1 and System 2

An interesting parallel that many researchers relate to is the work by Daniel Kahneman and Amos Tversky that won them the Nobel prize in economics

- The idea is that we have a *System 1* that is fast, automatic, and unconcious
  - Riding a bike
  - 2+2=?
- *System 2* is slow, logical
  - Determine social behaviour



Parallel to machine learning

The argument now is that AI systems can be modelled the same

- Machine learning models are highly specialized in narrow tasks, similar to system 1
- Good Old Fashion AI (GOFAI) such as logic, the more deliberate system 2
- A surge in interest in combining GOFAI and ML, see e.g. [6]

Machine Learning Today

Some of the hot topics in the machine learning community

- Safe AI
- More interdisciplinary work
  - Important for explainable AI et c.
- Building models that need less data
  - Again, comming back to the Frame Problem
- Really large models (e.g. GPT3 from OpenAI) using unlabeled data

Guacamole cat

- Neural networks is traditionally a black box, unclear what part of the data decides the outcome
- Or; Current deep-learning mechanisms are unable to link decisions to inputs

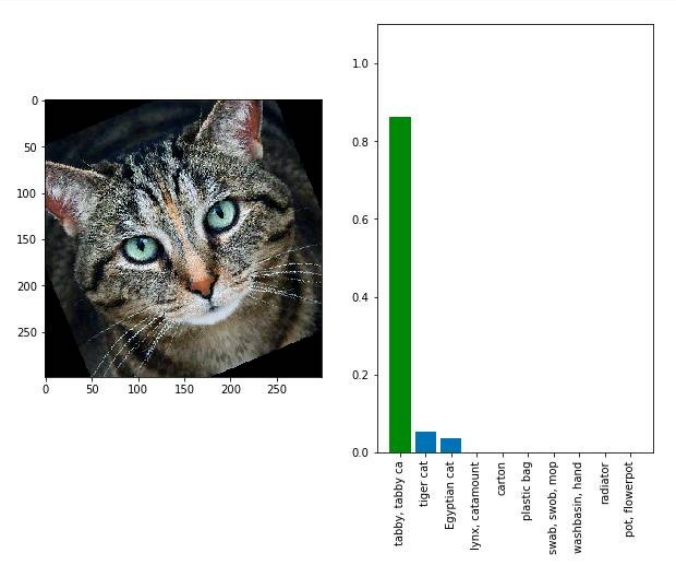


Figure 12: Image classified as a tabby cat.

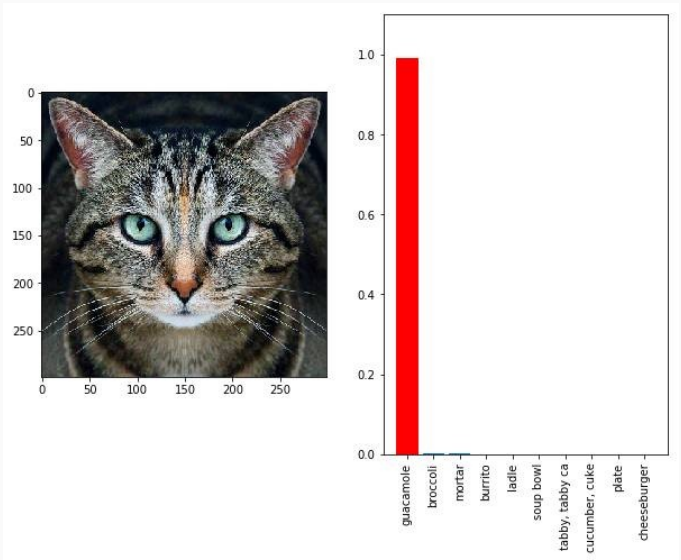


Figure 13: Image of the same cat, slightly rotated, classified as guacamole.

Slightly more uncomfortable example

Could someone give an example that is more serious?

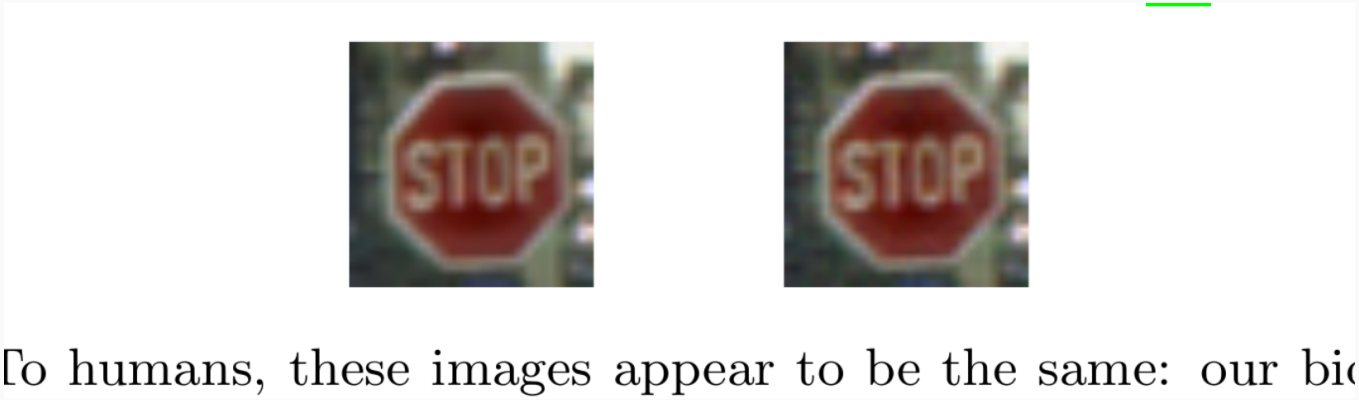


Figure 14: Two versions of the same image. One is recognized as a stop sign, the other as a yield sign [7].

The last decade has seen an increased need for what is known as Explainable AI (XAI).

Theory of mind

Relationship between AI and philosophy

- Philosopher John C. Searles' distinction between *Strong* and *Weak* AI [8]
  - A system can: (1) *have a mind and mental states* or (2) *act intelligently*. Back to Chinese Room
- Traditional theory of mind is a framework within belief, knowledge, et c., are defined, and the ability to attribute this to oneself and others
- Rabinowitz et al. [9] has worked on a machine theory of mind, showcasing learning by observing
- The problem of other minds (see [10]): Partially the sollipsism argument, partially the problem of recognizing alien (as in unknown to us) intelligence

Discussion

Is there anything you'd like us to elaborate on? Something we haven't talked about? Something we have talked about?

Last slide

QUESTIONS?

References i

- S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.
- C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.
- O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.
- J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networkss," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

References ii


H. Geffner, "Model-free, model-based, and general intelligence," *CoRR*, vol. abs/1806.02308, 2018.


N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, (New York, NY, USA), pp. 506–519, ACM, 2017.


J. Searle, *Mind, Language and Society: Philosophy in the Real World*. **MasterMinds series, Basic Books, 1999.**


N. C. Rabinowitz, F. Perbet, H. F. Song, C. Zhang, S. M. A. Eslami, and M. Botvinick, "Machine theory of mind," *CoRR*, vol. abs/1802.07740, 2018.


P. Godfrey-Smith, *Other Minds: The Octopus, the Sea, and the Deep Origins of Consciousness*. **Farrar, Straus and Giroux, 2016.**