

Data efficiency in deep learning

Grégory Châtel

Disaitek
Intel AI Software Innovator

@rodgzilla
github.com/rodgzilla

February 26th, 2019

- 1 Deep learning basics
- 2 Popular ML tasks and their dataset
- 3 Data efficiency
 - Transfer learning
 - Multi-task learning
 - Semi-supervised learning

Machine learning

Machine learning is a subfield of artificial intelligence.

Intuitively We want to *learn from* and *make predictions on* data.

Technically We want to build a model that approximate well (e.g. minimize a loss function) an unknown function.

Application examples

- Regression

Polynomial $(x, y, z) \rightarrow f(x, y, z)$

House price (surface, nb rooms, city) \rightarrow price

- Classification

Image classification pixel values \rightarrow cat or dog

Text classification list of words \rightarrow spam or valid email

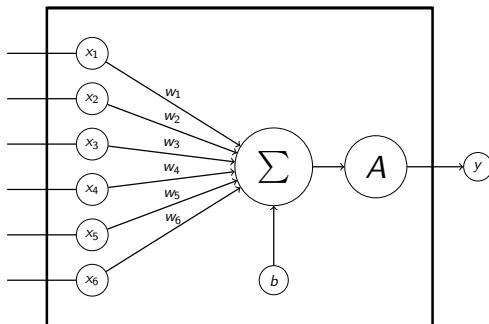
Deep learning

Concept

Deep learning is a subfield of machine learning in which we use artificial neural networks to make predictions.

An artificial neural networks is a computation model loosely based on the human brain. It aims to mimic electric signals travelling through neurons in order to make computations.

Neuron with activation

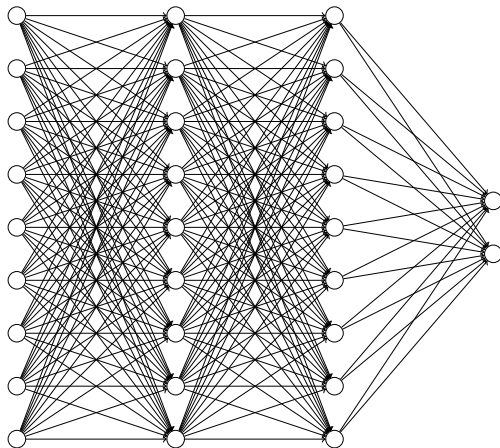


$$A(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$y = A(w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6 + b)$$

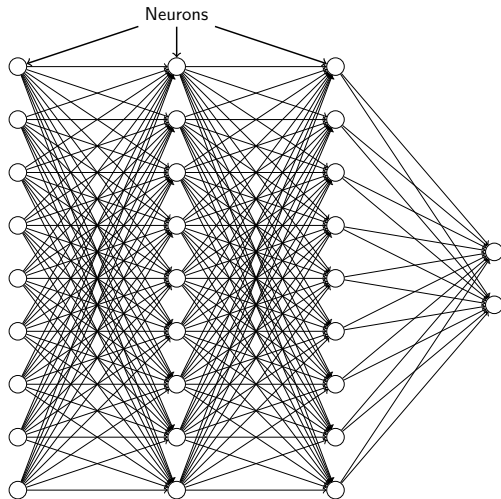
Artificial neural network

Network



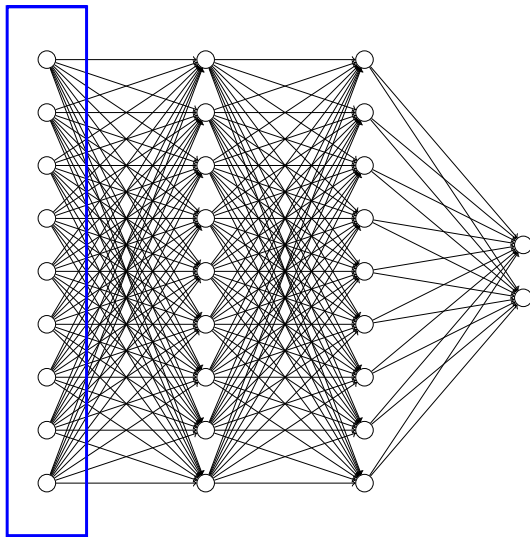
Artificial neural network

Network



Artificial neural network

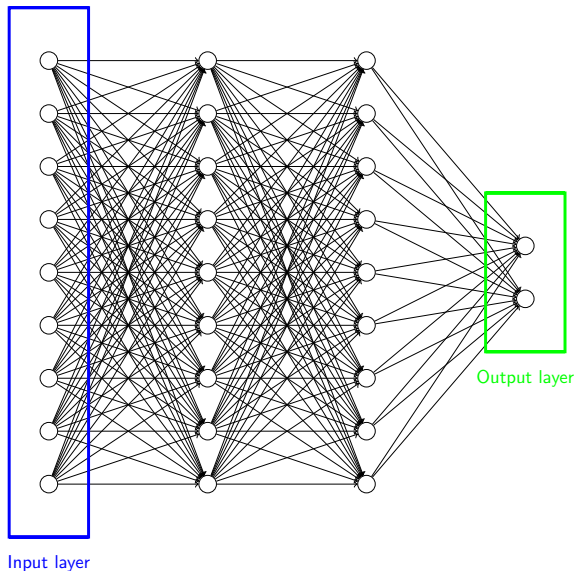
Network



Input layer

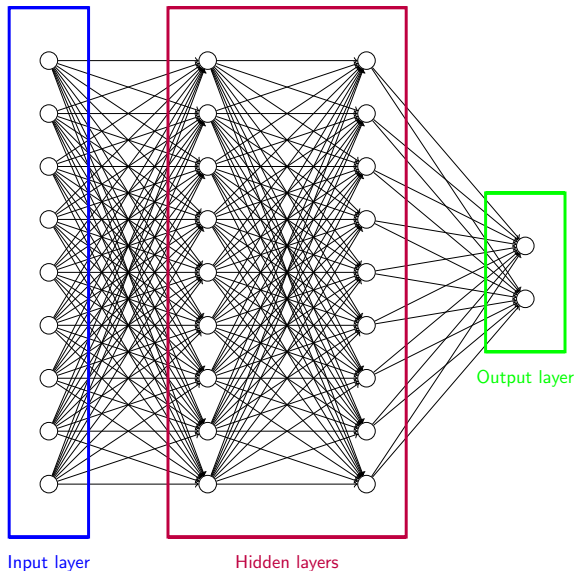
Artificial neural network

Network



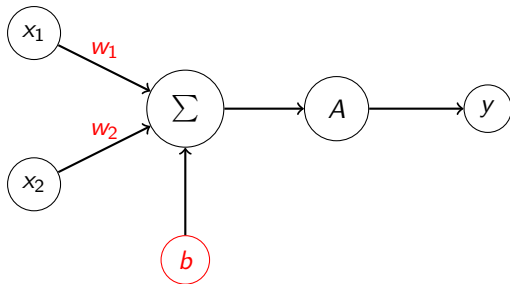
Artificial neural network

Network



Computation example

Binary AND gate

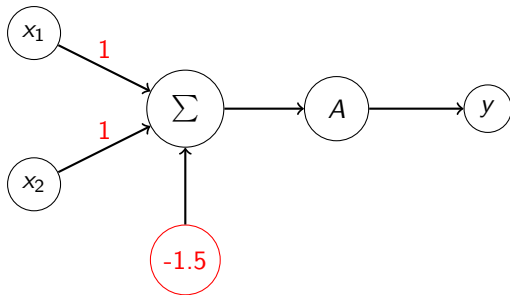


We want to set w_1 , w_2 and b such that:

$$A(w_1x_1 + w_2x_2 + b) = x_1 \text{ AND } x_2$$

Computation example

Binary AND gate

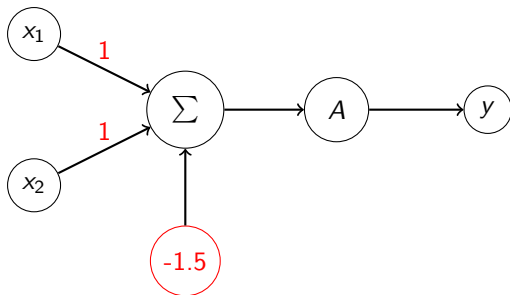


We want to set w_1 , w_2 and b such that:

$$A(w_1x_1 + w_2x_2 + b) = x_1 \text{ AND } x_2$$

Computation example

Binary AND gate



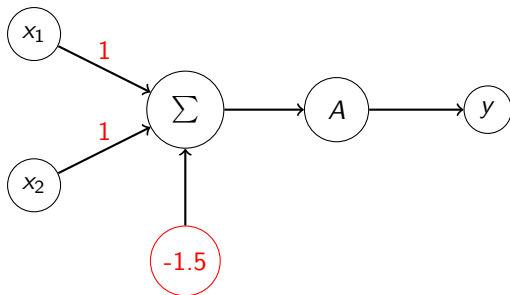
We want to set w_1 , w_2 and b such that:

$$A(w_1x_1 + w_2x_2 + b) = x_1 \text{ AND } x_2$$

$$x_0 = 0, x_1 = 1. \quad y = A(0 + 1 - 1.5) = A(-0.5) = 0$$

Computation example

Binary AND gate



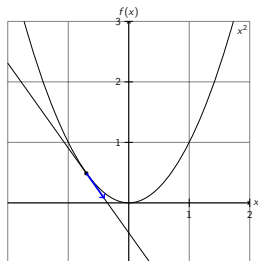
We want to set w_1 , w_2 and b such that:

$$A(w_1x_1 + w_2x_2 + b) = x_1 \text{ AND } x_2$$

$$x_0 = 0, x_1 = 1. \quad y = A(0 + 1 - 1.5) = A(-0.5) = 0$$

$$x_0 = 1, x_1 = 1. \quad y = A(1 + 1 - 1.5) = A(0.5) = 1$$

Neural networks



To train and evaluate neural networks, we use a **loss function** $L(\theta, x, y)$ with θ the parameters of the models, x an input and y the target. This function measures *how good* the prediction of the model is on a specific example.

To **train** a neural network we compute the derivative of L according to **the weights** (θ) and use the result to update θ in order to **decrease** the loss value.

Popular datasets for computer vision

- 1990, Statlog ~2k outdoor images,
- 1998, MNIST 60k B&W images of handwritten digits,
- 2005, LabelMe ~187k scenes images,
- 2009, ImageNet ~14M images with 1000 different categories,
- 2017, JFT-300M ~300M RGB images ~18k categories (internal dataset @ Google).

Popular datasets for Natural Language Processing (NLP)

1997, Car evaluation dataset $\sim 2\text{k}$ textual car evaluations,

2005, Stanford Sentiment Treebank $\sim 11\text{k}$ movie reviews,

2011, IMDB Reviews $\sim 50\text{k}$ movie reviews,

2012, Youtube Comedy Slam $\sim 1.1\text{M}$ pairs of video metadata,

2015, Amazon reviews $\sim 82\text{M}$ product reviews.

Dataset creation strategies

Creating new high quality datasets is both **hard** and **expensive**.

Some researchers experiment with training models using **low quality data** (weakly supervised learning).

Amazon offers a **dataset creation service** (Amazon Mechanical Turk) where you can pay to get your dataset labelled by humans.

These strategies are **not** the focus of this talk.

Data efficiency

Knowing that datasets are so important and hard to create, it is important to squeeze *every last bit of value* out of them.

To do this, three ideas are explored:

- Transfer learning,
- Multi-task learning,
- Semi-supervised learning.

Transfer learning

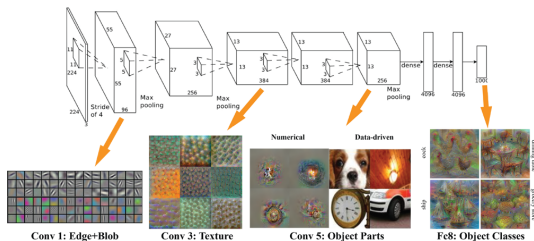
The application of skills, knowledge, and/or attitudes that were learned in one situation to another learning situation. (Perkins, 1992)

Transfer learning consists in taking an artificial neural network that has been trained on a *generic* task and *transferring* its knowledge (retraining it) to perform a new task.

The idea behind this method is that the information learned on a generic task will probably be useful for a new task of the same domain.

Transfer learning is actually the base of the [Google Cloud AutoML service](#).

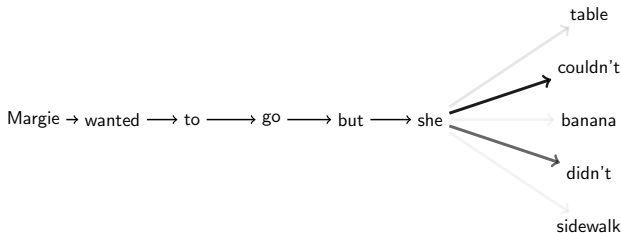
Transfer learning in computer vision



Using this trained model as a base to build a dogs vs cats picture classifier **greatly reduce the need of labelled data**.

The knowledge about **basic shapes and textures** that has been learned on ImageNet will be useful to almost all task involving real world images.

Transfer learning in NLP



The **language modeling task** is currently the most generic task that NLP researcher have found to perform transfer learning.

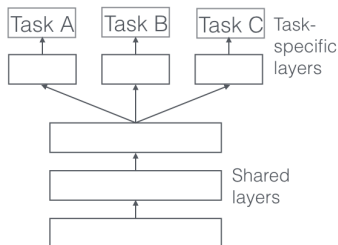
Knowing how to predict the most likely following word requires to understand, to some extent, the **meaning of words**, **the syntax of the language** and **the way concepts interact**.

Typical language models are trained on Wikipedia content, books or Internet Common Crawl.

Multi-task learning

Multitask Learning is an approach to inductive transfer that improves generalization [...] It does this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better. (Rich Caruana, 1997)

Instead of just training the network to perform the desired task, we also optimize it to perform *auxiliary tasks*.



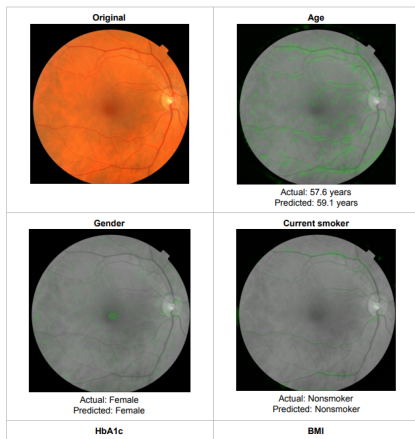
Multi-task as a regularization technique



Informally, the goal of the multi-task learning is to force the model to use its **computing power** to perform something **meaningful** instead of using it to learn the **noise** of the data (overfitting).

Image from <https://hackernoon.com/memorizing-is-not-learning-6-tricks-to-prevent-overfitting-in-machine-learning-820b091dc42>

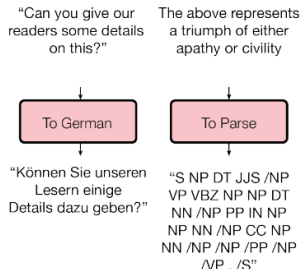
Multi-task learning in computer vision



Some researchers discovered that by asking a model to predict the gender and age of patient in addition to detect *cardiovascular diseases* they got strong performance improvements.

Poplin, Ryan, et al. "Predicting cardiovascular risk factors from retinal fundus photographs using deep learning." arXiv preprint arXiv:1708.09843 (2017).

Multi-task learning in NLP

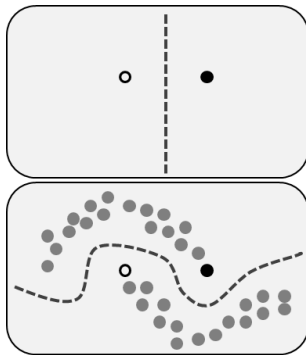


In NLP, translation can be used as an auxiliary task to improve models that perform tasks that have relatively small datasets such as sentence parsing.

By making the model perform translation, a task with huge datasets, we allow it to gain access to a much richer [structure of the language](#).

Kaiser, Lukasz, et al. "One model to learn them all." arXiv preprint arXiv:1706.05137 (2017).

Semi-supervised learning



The idea of semi-supervised learning is to use *unlabelled data* to improve our model.

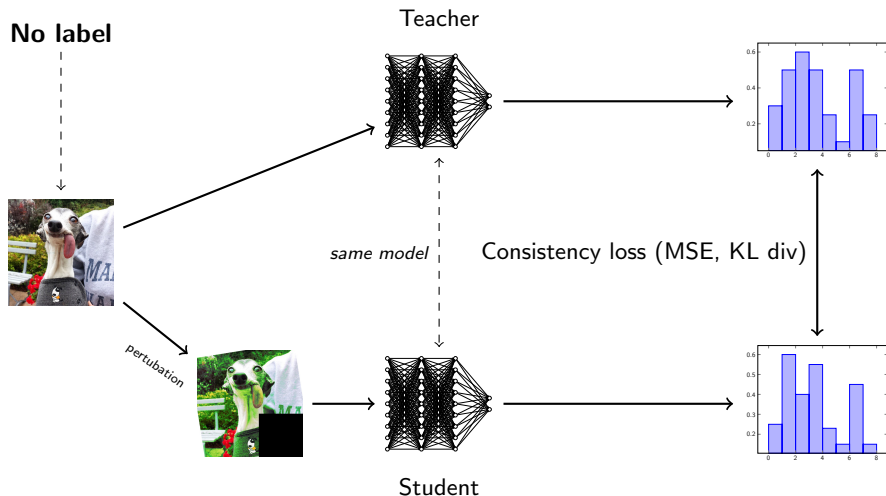
Image from https://en.wikipedia.org/wiki/Semi-supervised_learning

Concepts of semi-supervised learning

The main concept of semi-supervised learning is to train a **weaker student** to imitate a **stronger teacher**.

Technically, we apply *mean-squared error* or a *Kullback-Liebler divergence* between the logits output by the student and the teacher. We typically alternate between supervised and semi-supervised steps of training.

Semi-supervised learning in practice



We do not need a label for the clean image, we want to teach the model to be noise invariant.

Applying deep learning to *small* datasets

- 1 Establish a strong baseline using basic algorithms (logistic regression, random forest, etc.)
- 2 Choose a model architecture (MLP, CNN, RNN, Transformer)
- 3 Try to find (or build) a pre-trained version of this model that performs a related task (*transfer learning*)
- 4 Try to find a related auxiliary task to regularize and improve the model learning abilities. (*multi-task learning*)
- 5 Once the performance of the model is relatively good, try to use unlabelled data to improve noise invariance (*semi-supervised learning*)