Machine Learning

Intelligence and Learning: From Nature to Machine

Dr Chris Willcocks
Department of Computer Science

Durham University

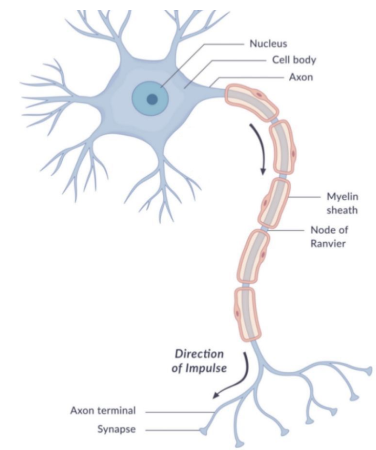
Lecture Overview

Course teaching style

- Concept driven, hands-on, practical/useful techniques
- Focus on recent research advances

Today's lecture

- Definitions of "artificial intelligence" and "machine learning"
- How learning works in nature (neuropsychology)
- How the brain works, how neurons work
- Neuroplasticity and Hebbian theory
- Basics of artificial neurons
- Fundamental (high-level) principles of machine learning
    - Introduction to "tensors"
    - Introduction to loss functions and so on



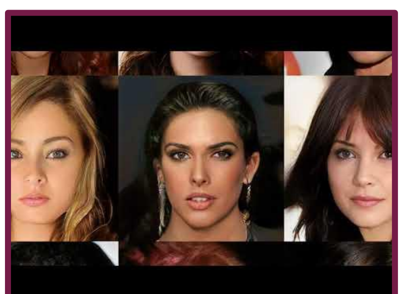
Artificial Intelligence

Durham University

- **Intelligence** is generally ill-defined:
  - "The ability to **acquire** and apply **knowledge** and **skills**"
  - "The **capacity** for **logic**, **understanding**, **self-awareness**, **learning**, **emotional knowledge**, **reasoning**, **planning**, **creativity**, and **problem solving**."
  - "The **ability to learn** or **understand** or to deal with **new** or trying **situations**"
- **Artificial** Intelligence (or **Machine** Intelligence)
  - Intelligence demonstrated by machines (in contrast to **natural intelligence**)

Hierarchy

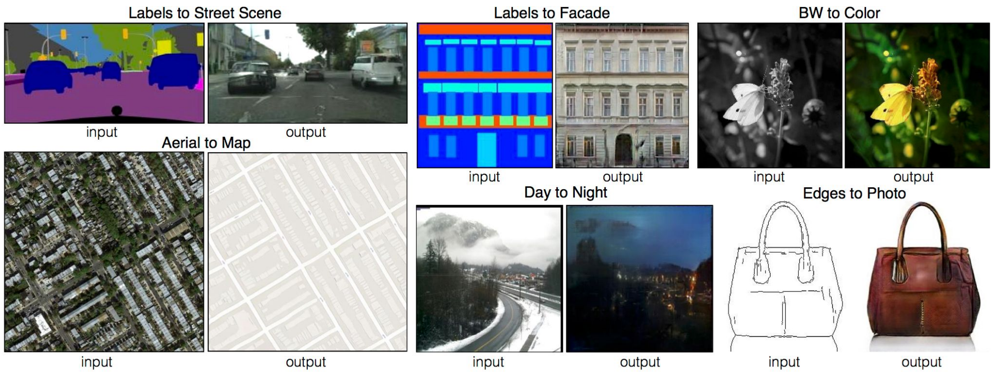
	e.g.
Artificial Intelligence	Knowledge bases
↓	
Machine Learning	Logistic regression
↓	
Representation Learning	Shallow autoencoders
↓	
Deep Learning	MLPs

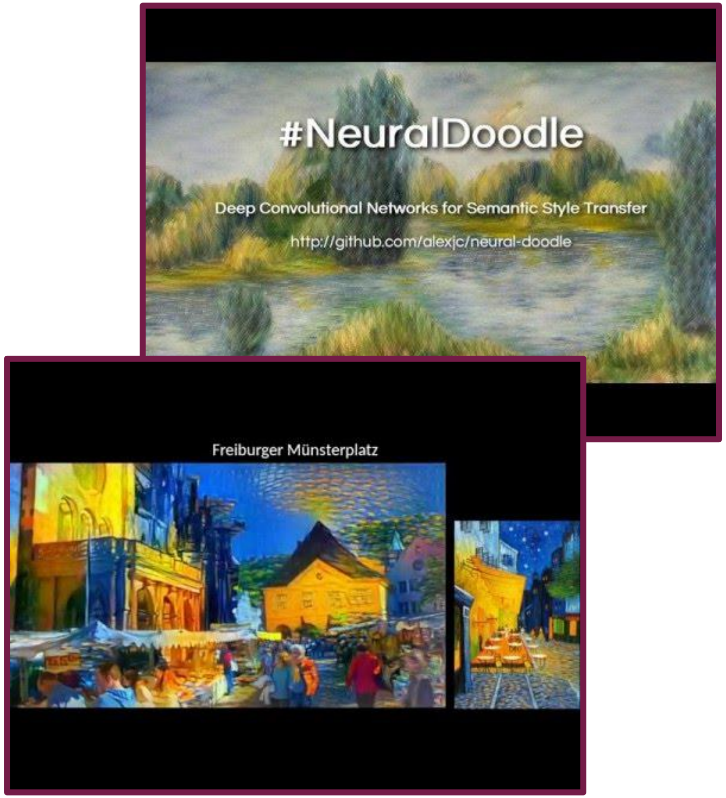


Progressive training of GANs

Are these tasks intelligent?

The ability to succeed at goals?
The ability to generalise?
The ability to create?
The ability to infer to new tasks?
The ability to comprehend?
What is creativity?





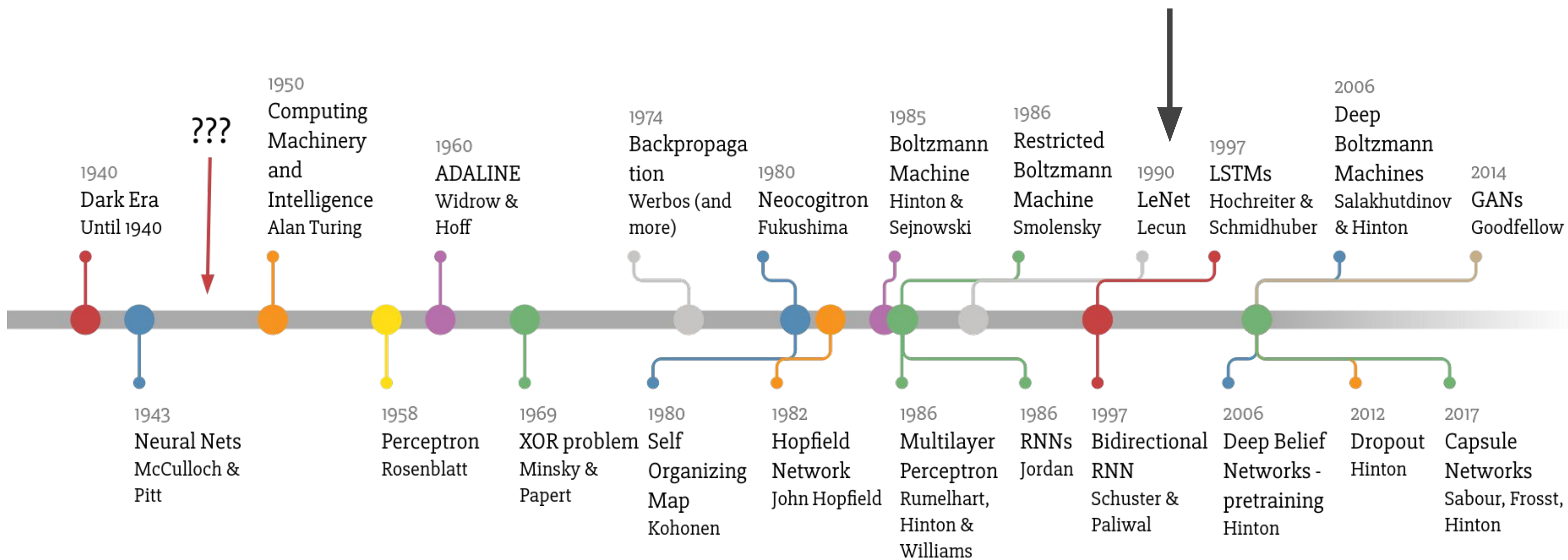
Style transfer in videos, neural doodle

The Waves of Artificial Intelligence

Durham University

The “recent” interest = GPU pricing, tooling, deep learning

*1998



Made by Favio Vázquez

What is Learning?

Environment

Experiences	Actions	Changed Actions
Delicious Cold Warm Burning	Put in mouth Get closer Touch	Eat it again! Don’t touch.

Time

What is Learning?

“We define learning as the **transformative process** of taking in information that—when internalized and mixed with what we have **experience**d—changes **what we know** and builds on **what we do.** It’s based on input, process, and reflection. It is what changes us.”

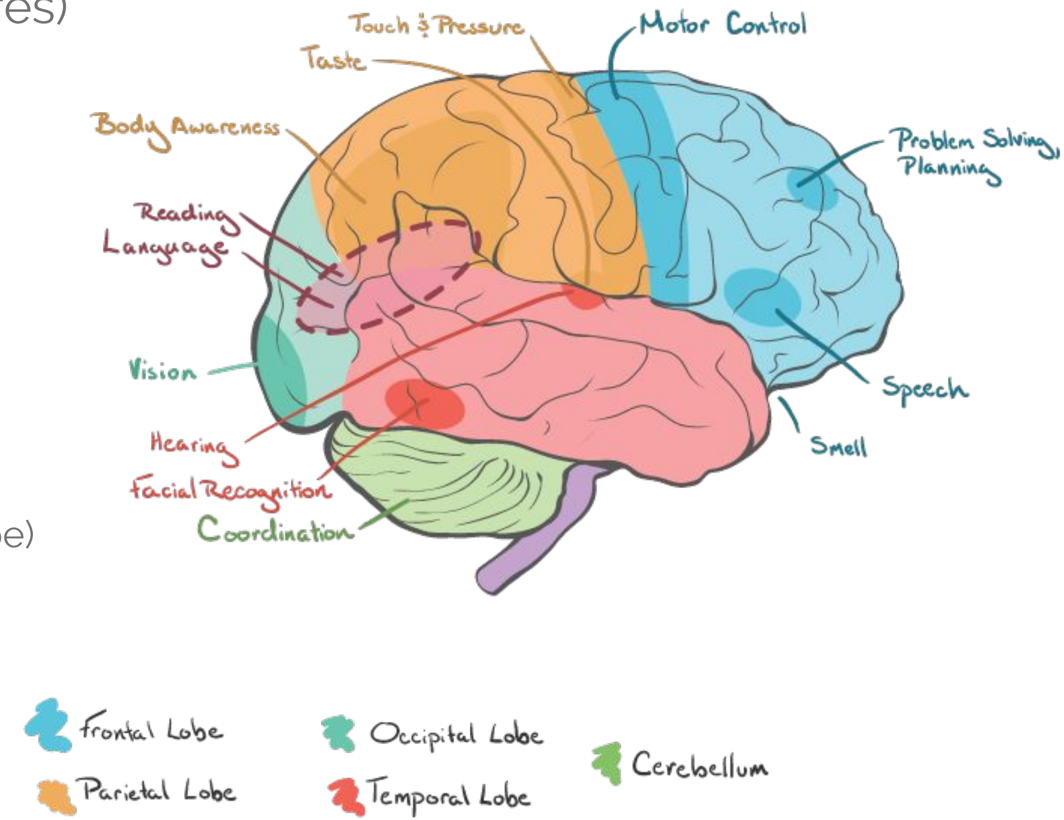
–From The New Social Learning by Tony Bingham and Marcia Conner.

“Learning is the relatively permanent **change** in a person’s **knowledge** or **behavior** due to **experience**. This definition has three components: 1) the duration of the change is long-term rather than short-term; 2) the locus of the change is the content and structure of knowledge in memory or the behavior of the learner; 3) the cause of the change is the learner’s experience in the environment rather than fatigue, motivation, drugs, physical condition or physiologic intervention.”

–From Learning in Encyclopedia of Educational Research, Richard E. Mayer.

How does the Brain Work?

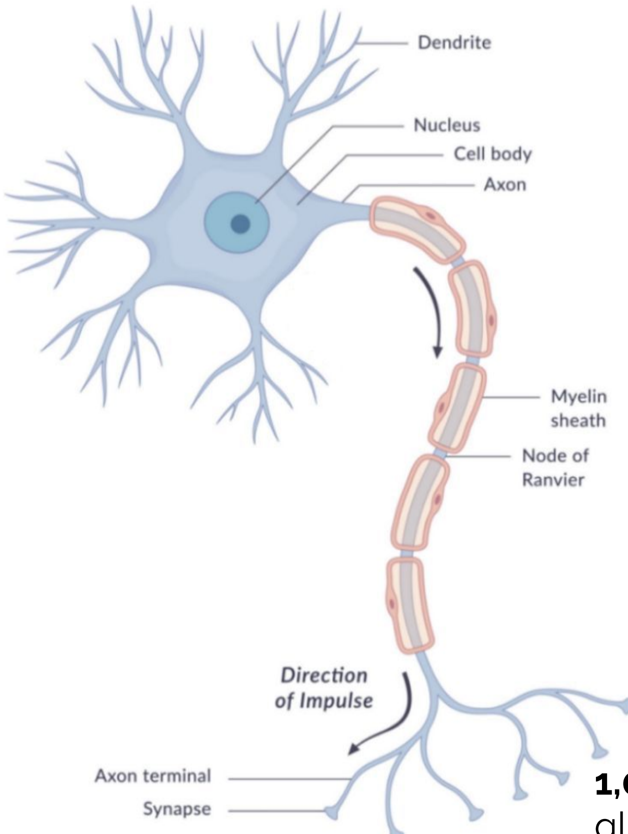
- Right side/left side (hemispheres)
  - Lobes
    - Frontal lobe
      - Executive functions
      - Memory
      - Learning
      - Planning
    - Parietal lobe
      - Sensation
      - Spatial awareness
    - Temporal lobe (banana shape)
      - Hearing
      - Language functions
    - Occipital lobe at back
      - Vision from front along optic nerves



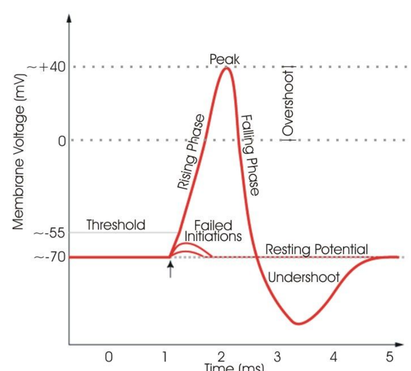
How does the Brain Work?

Durham University

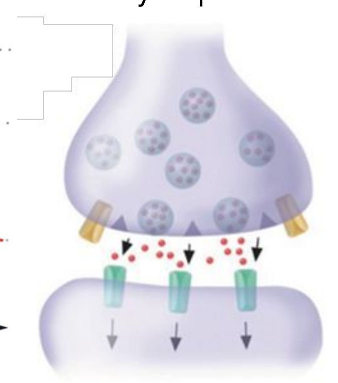
1,000’s of inputs (other neurons, sensory neurons e.g. taste buds from a salt or sugar molecule…)



- Neurons send out branches called **dendrites**, and a large output called an **axon**
- The axon is coated in myelin that helps it conduct electrical impulses.
- The places where the nerve cells make their connections with each other is called a **synapse**.



Synapse:



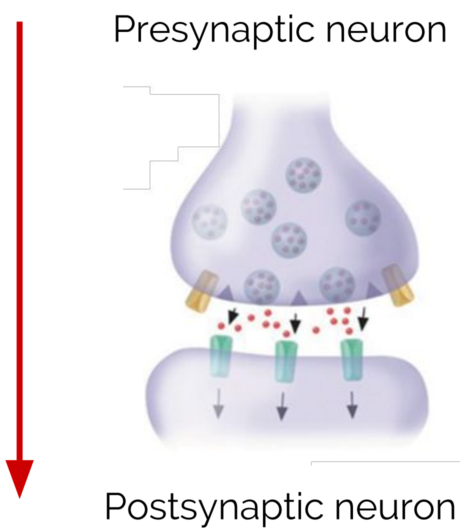
“Synapse” (from the greek meaning “to clasp together”)
Signals get **summed up**, and travel to the hillock (Axon neck)

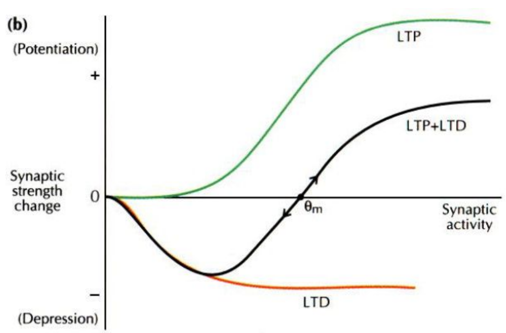
- If large enough, triggers an action potential travels down axon

1,000’s of targets (e.g. other neurons, muscle cells, gland cells, blood vessels to release hormones…)

Figure by wetcake (left) and Andrej Kral (right)

Synaptic Plasticity

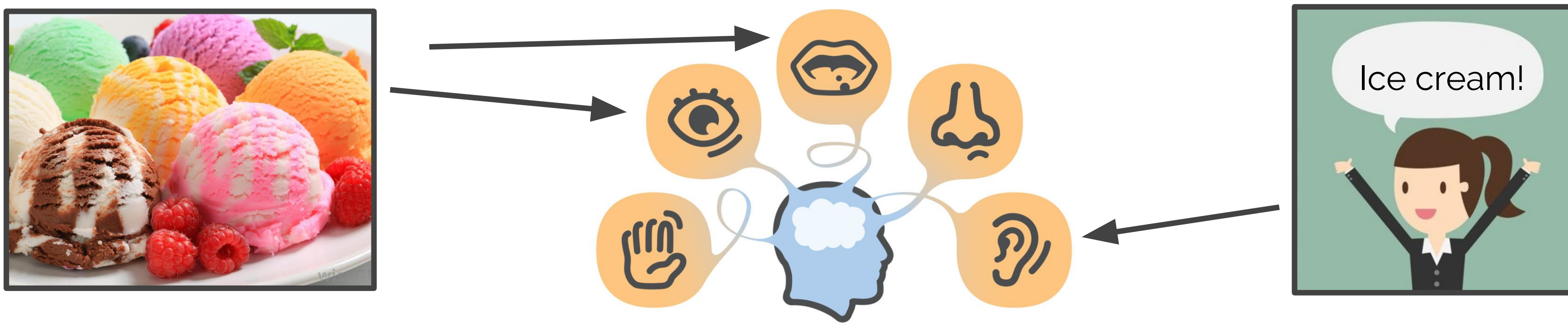




- What's very cool is that with frequent repeated stimulation, the same level of presynaptic stimulation converts into **greater** postsynaptic potential.
  - In other words, as a neuron gets a lot of practice sending signals to a specific target neuron, it gets better at sending those signals (the synapse strength increases).
    - Increased strength that lasts for a long time (from minutes to many months) is called **Long Term Potentiation** (weakening is **Long Term Depression**).
  - As synapses are strengthened and retain strength, we're able to more easily recall previous experiences.

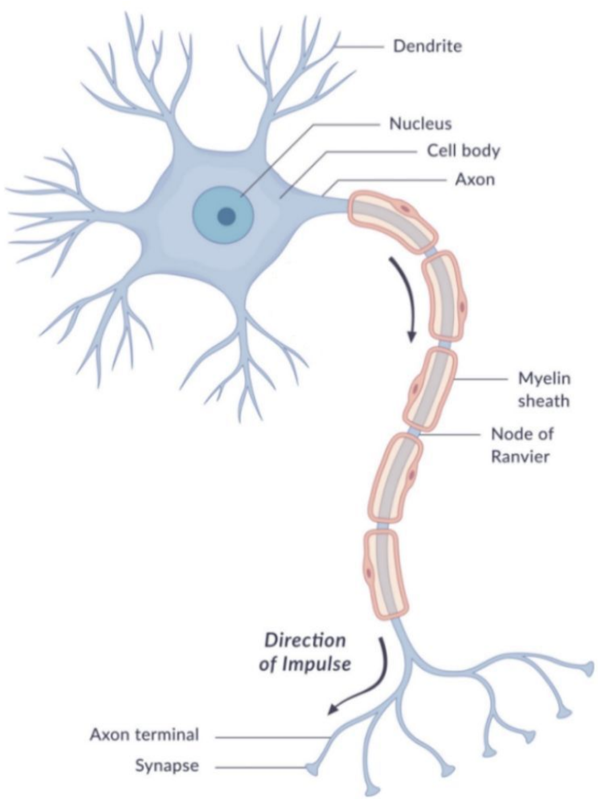
Figure from: "Synaptic Plasticity: A molecular mechanism for metaplasticity", Journal of Current Biology.

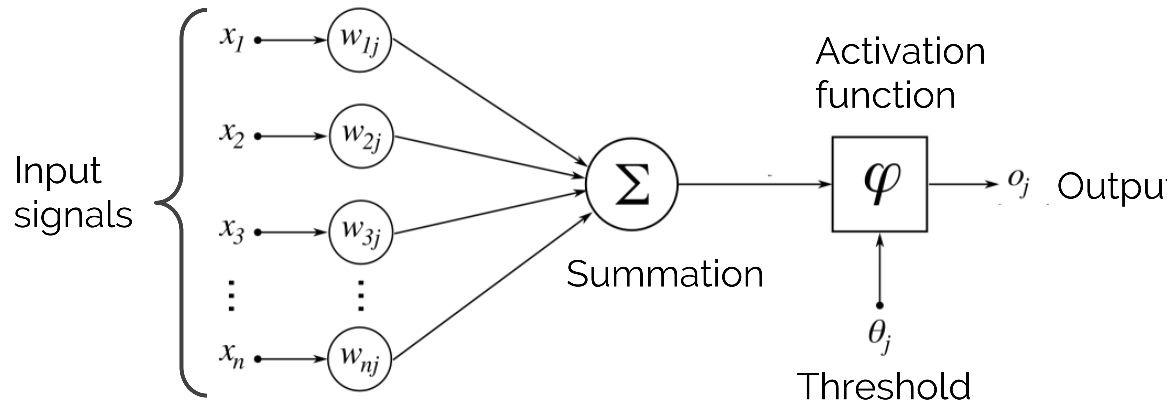
Hebbian Theory

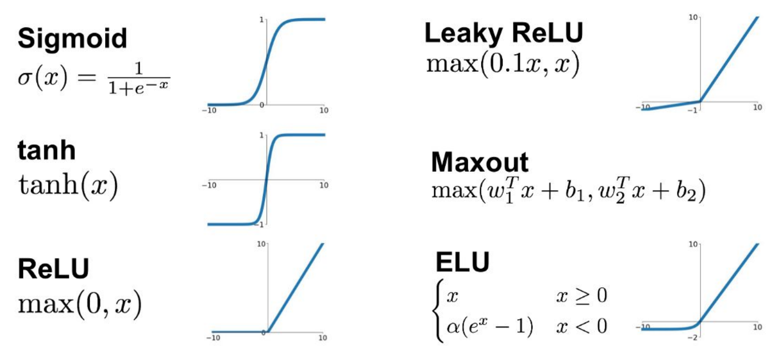


- **Hebbian theory**
  - If two neurons fire at the same time, the connections between them are strengthened, and thus are more likely to fire again together in the future.
  - If two neurons fire in an uncoordinated manner, their connections are weakened and their more likely to act independently in the future.
- **Updated hebbian hypothesis based on recent findings**
  - If the presynaptic neuron fires within a window of 20ms before the postsynaptic neuron, the synapse will be strengthened.
  - However if the presynaptic neuron fires within a window of 20ms after the postsynaptic neuron, the synapse will be weakened.

Artificial Neurons







What is Machine Learning?

...learning, but with a machine! (ok *throw in some stats, calculus, geometry, ...*)

"A computer program is said to learn from experience $\boldsymbol{E}$ with respect to some class of tasks $\boldsymbol{T}$ and performance measure $\boldsymbol{P}$, if its performance at tasks in $\boldsymbol{T}$ as measured by $\boldsymbol{P}$, improves with experience $\boldsymbol{E}$"

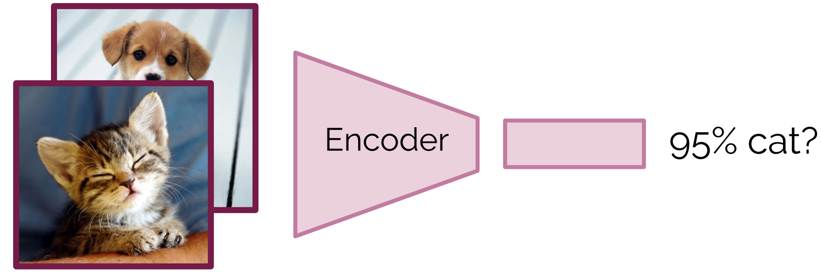
Mitchell, 1997

So we have 3 components:

1. The Task $\boldsymbol{T}$
2. The Experience $\boldsymbol{E}$
3. The Performance Measure $\boldsymbol{P}$

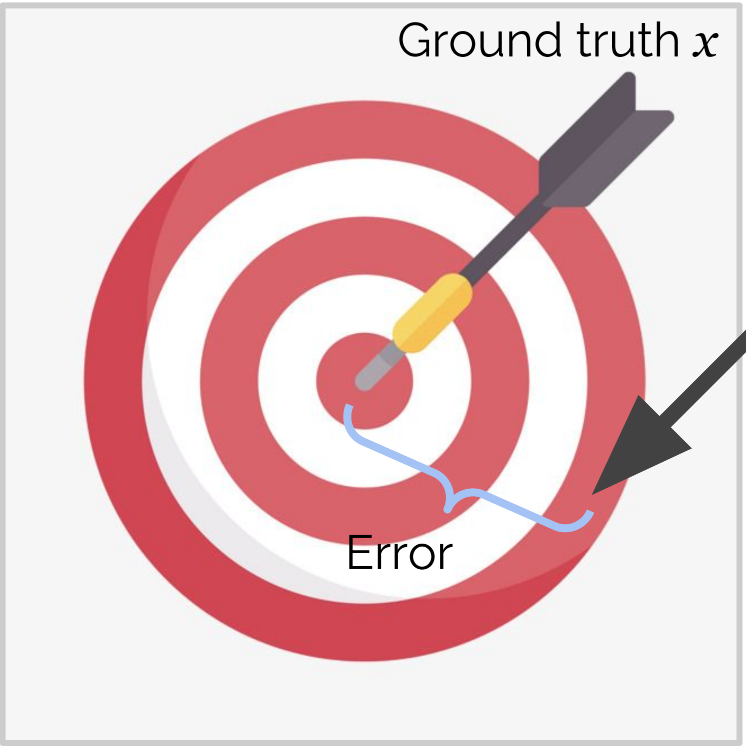
Tasks T

There is a large set of tasks, here's some listed by Goodfellow:



1. Classification
2. Regression
3. Transcription (image to characters, audio to characters)
4. Translation (english characters to french characters)
5. Anomaly detection (flag stuff that is unusual)
6. Synthesis (generating new examples similar to the training data)
7. Inpainting (with some data removed, can predict value of missing entries)
8. Denoising
9. Density estimation (learn where examples cluster tighly and where they are unlikely occur)

The Performance Measure *P*

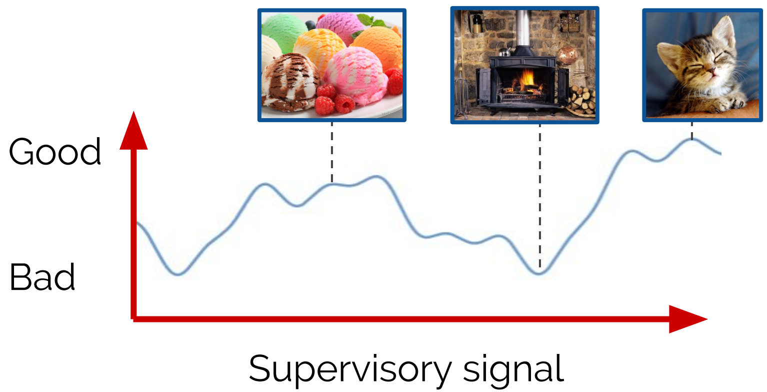


Model prediction p

- We need to define a quantitative measure of performance
- Typically we want it to give a **continuous**-valued score for each example
- Can be difficult to define for certain tasks

The Experience *E*

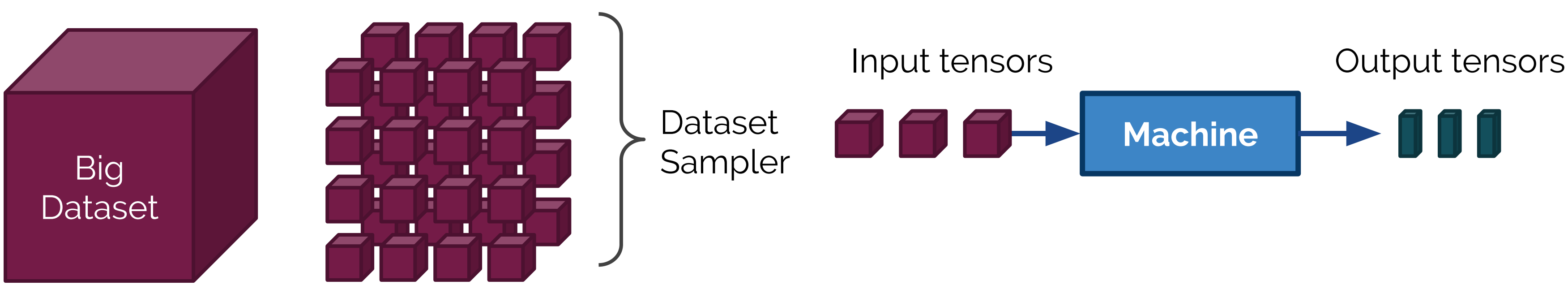
- Supervised
  - Supervisory signal/labels
    Guide the learning
- Unsupervised
  - Clustering
  - Dimensionality reduction
  - Generative models
- Semi-supervised
- Reinforcement learning
- One-shot learning
- Batch learning vs Online (incremental) learning



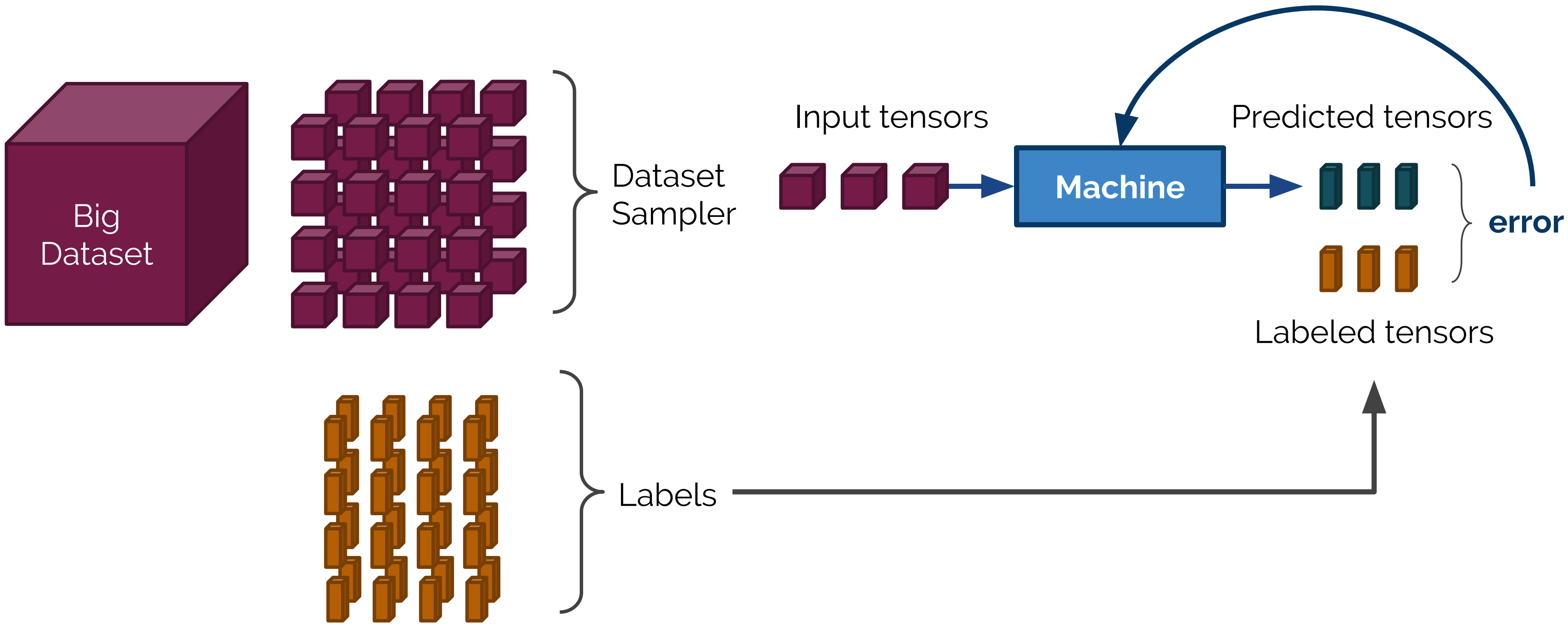
Experiencing a Dataset

In **machine** learning in practice, <u>we have constraints</u>:

- In most practical ML applications we don't have access to a continuous influx of **data over time**
- We have a static dataset that we wish to learn from, which can be quite large
- The **machine** we want to use has a "tiny" amount of available memory
  - Therefore we split our dataset down into **tensors** represented as multidimensional arrays

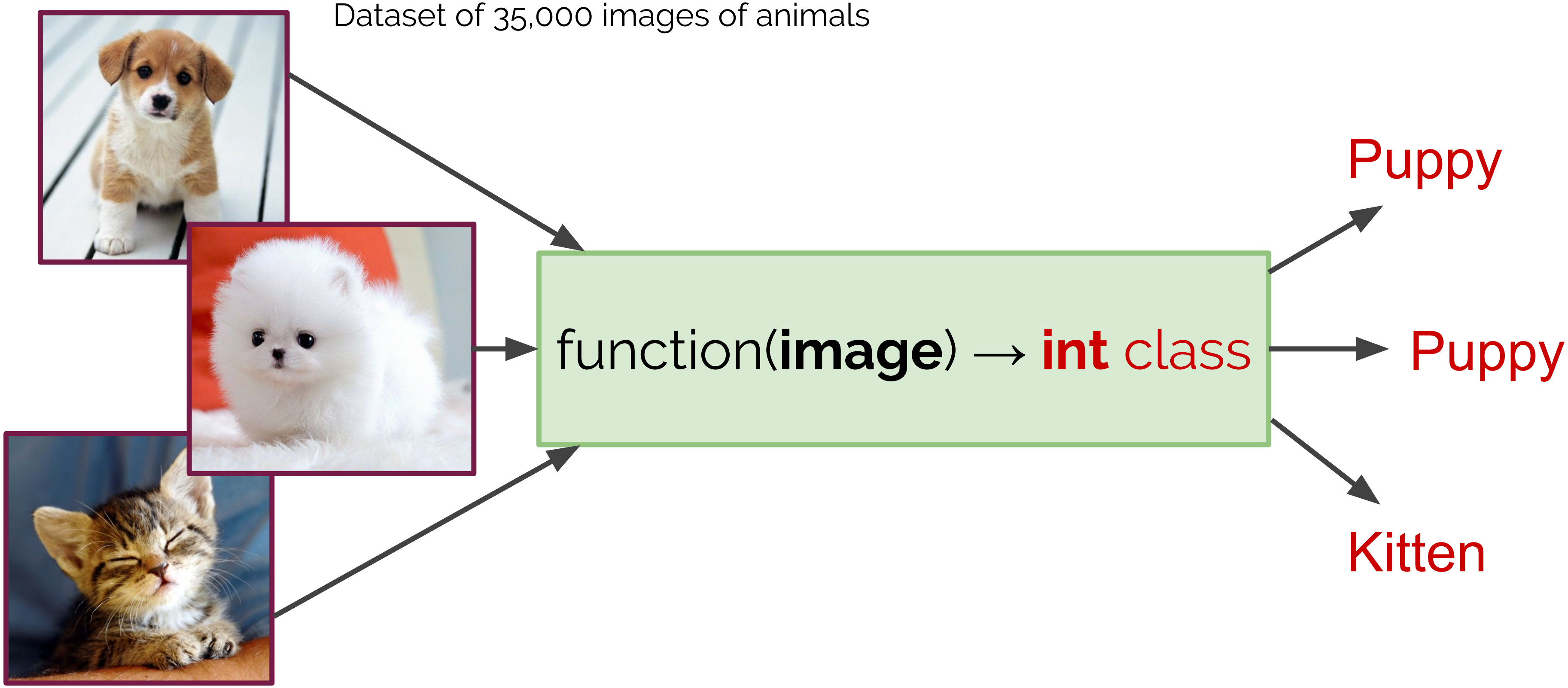


Supervised Learning

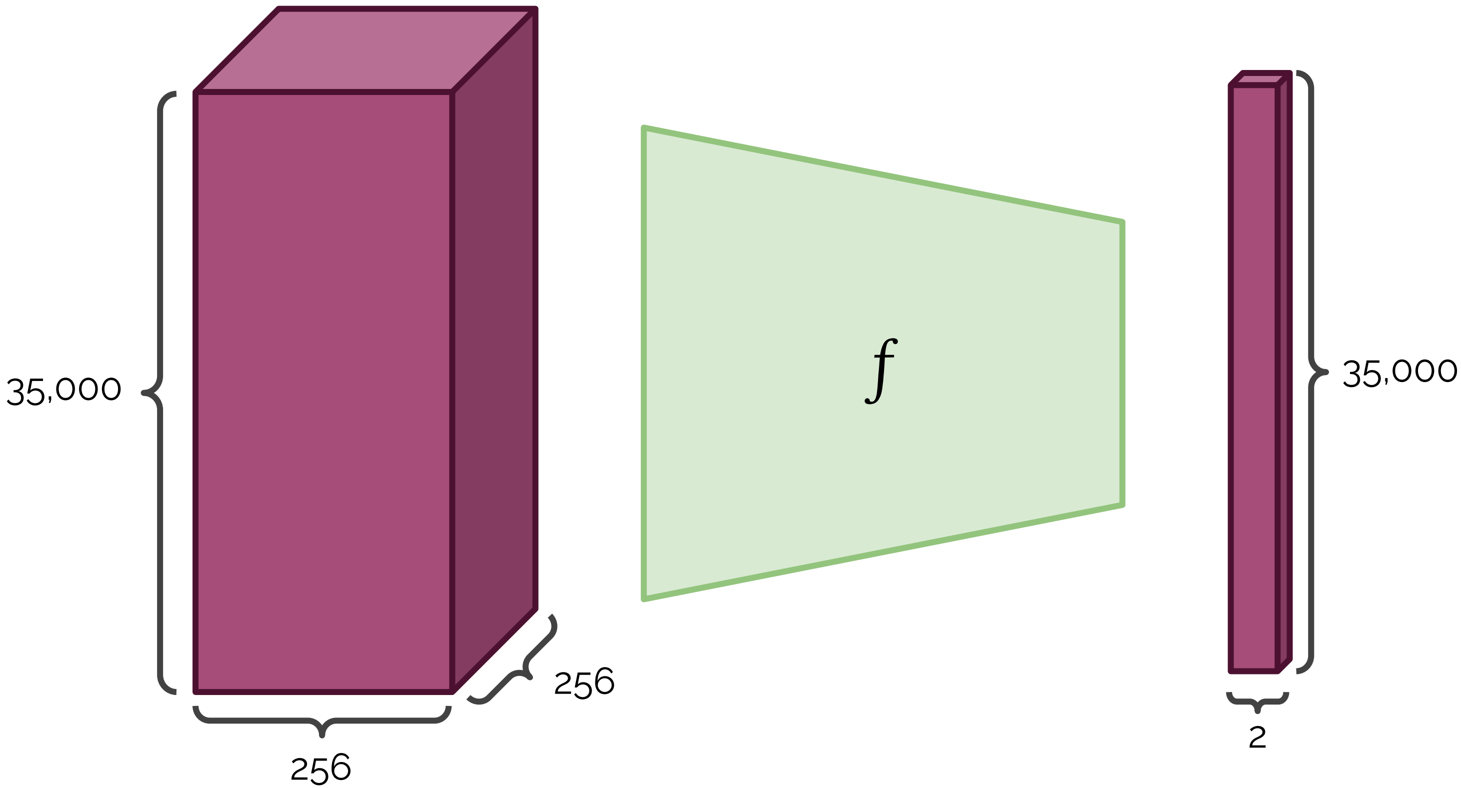


Classification Example

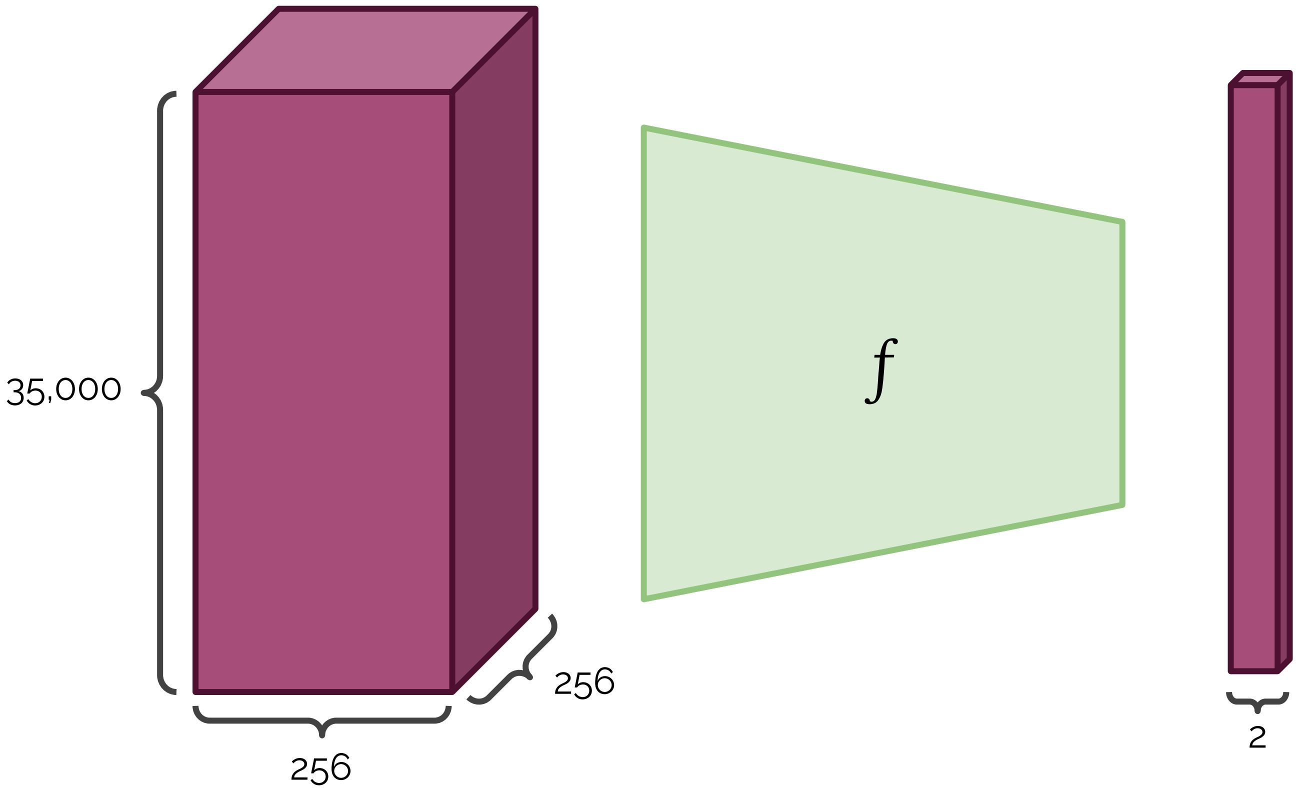
Dataset of 35,000 images of animals



Tensors: Images → Classes

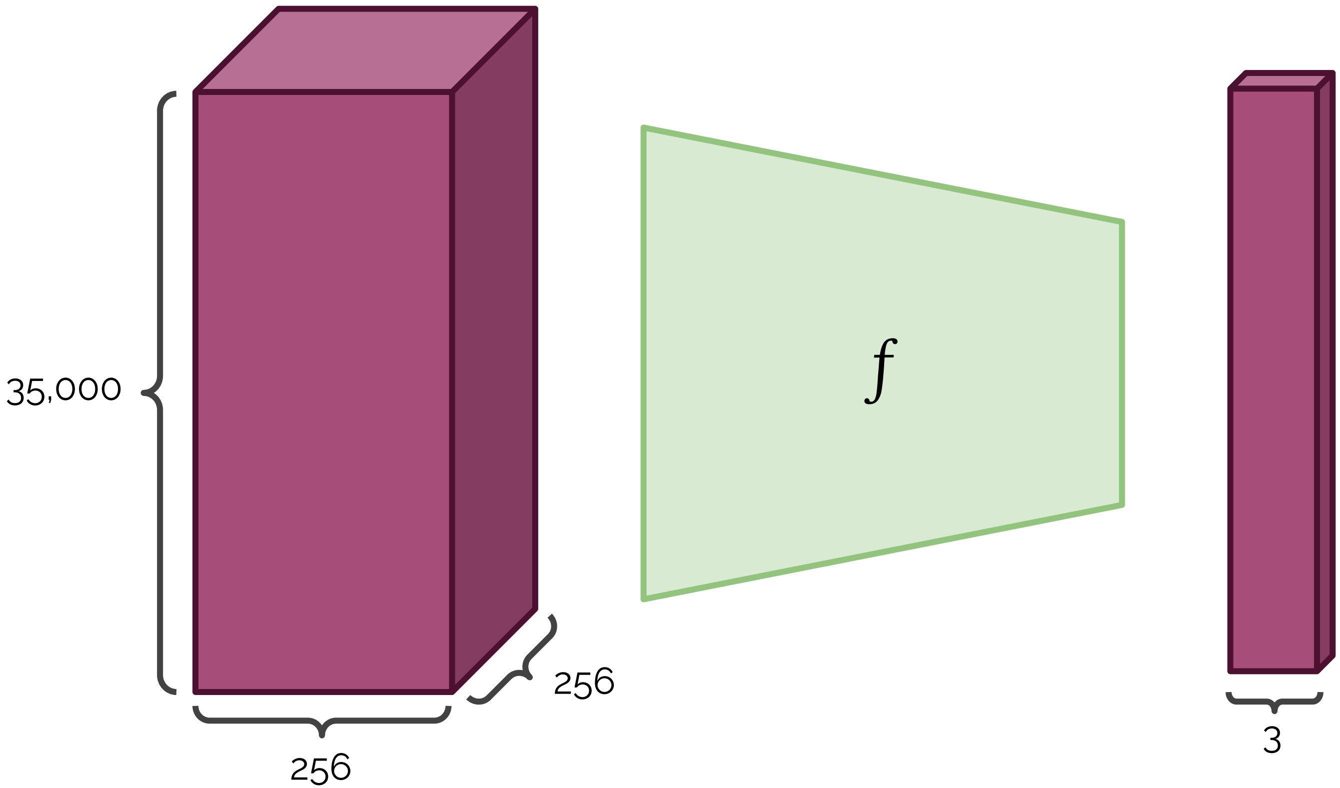


Tensors: Images → Classes



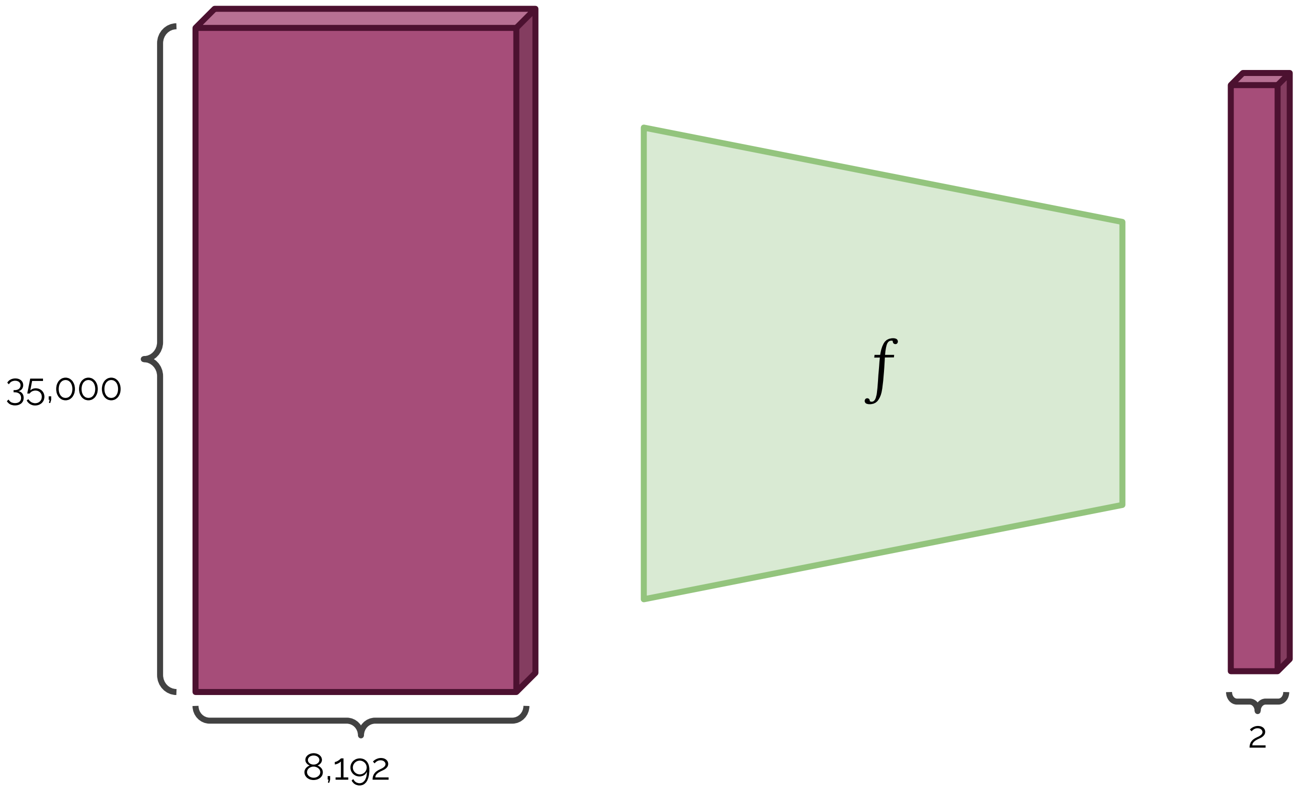
Dog	Cat
1	0
0	1
0	1
1	0
0	1
0	1
1	0
...	...

Tensors: Images → Classes



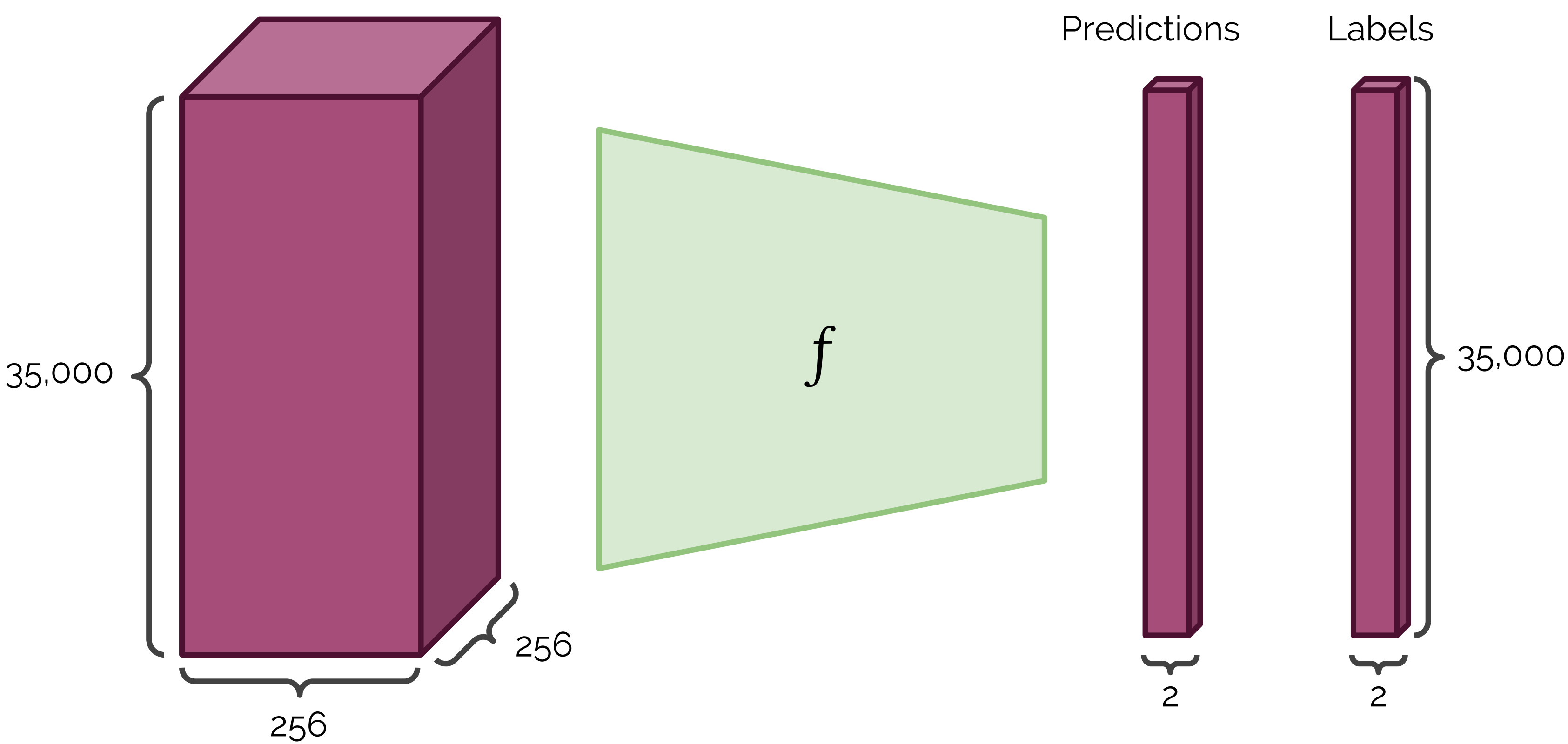
Dog	Cat	Fish
0	0	1
1	0	0
0	1	0
0	0	1
0	1	0
0	0	1
0	1	0
...	...	

Tensors: Audio → Classes



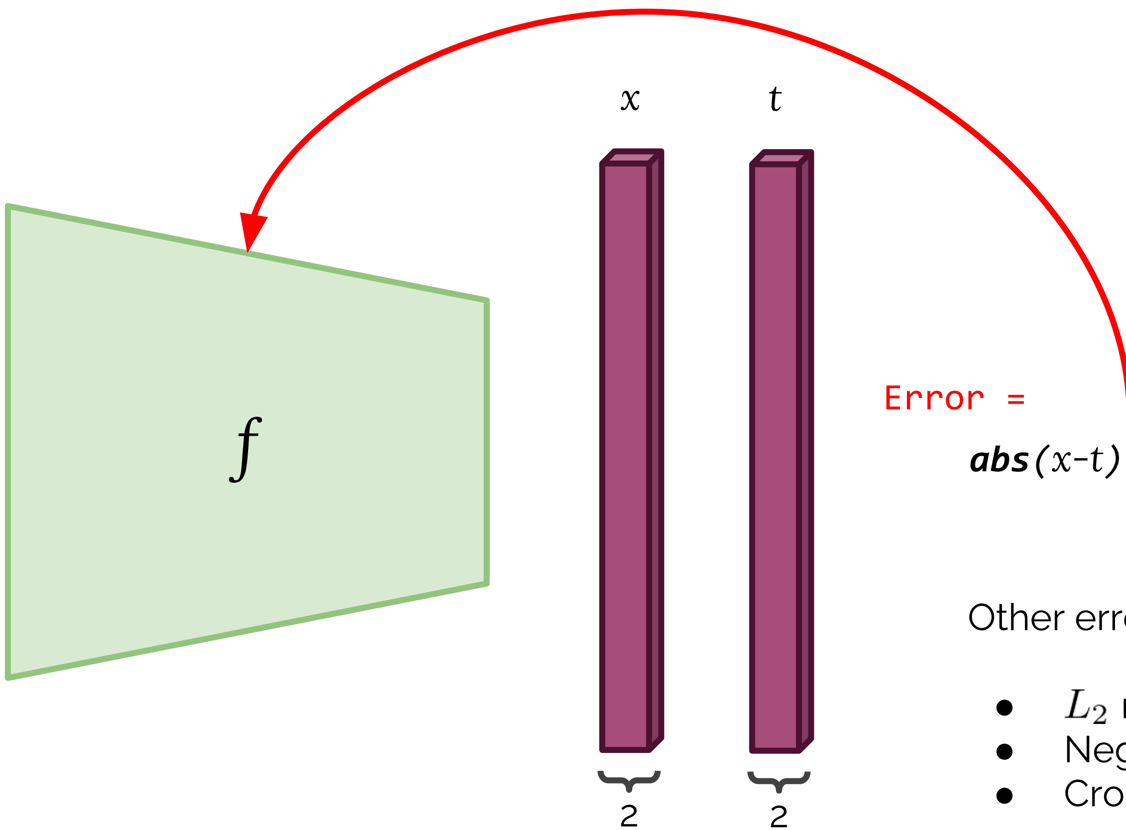
English	French
1	0
0	1
0	1
1	0
0	1
0	1
1	0
...	...

Predictions and Labels



Norms and Loss Functions

Durham University



Error =

```
abs(x-t).sum()
```

= 0.3914

L_p norm:

$$\|x\|_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}$$

$$\|x\|_1 = \sum_i |x_i|$$

$$\|x\| = \|x\|_2 = \sqrt{\sum_i x_i^2}$$

Other error functions:

- L_2 norm (p=2 Euclidean norm)
- Negative Log Likelihood
- Cross Entropy Loss

...

More loss functions

Durham University

https://pytorch.org/docs/stable/nn.html#loss-functions

Docs > torch.nn

CrossEntropyLoss

CLASS `torch.nn.CrossEntropyLoss`(*weight=None, size_average=None, ignore_index=-100, reduce=None, reduction='mean'*) [SOURCE]

This criterion combines `nn.LogSoftmax()` and `nn.NLLLoss()` in one single class.

It is useful when training a classification problem with *C* classes. If provided, the optional argument `weight` should be a 1D *Tensor* assigning weight to each of the classes. This is particularly useful when you have an unbalanced training set.

The *input* is expected to contain scores for each class.

input has to be a Tensor of size either $(minibatch, C)$ or $(minibatch, C, d_1, d_2, ..., d_K)$ with $K \geq 2$ for the *K*-dimensional case (described later).

This criterion expects a class index (0 to *C-1*) as the *target* for each value of a 1D tensor of size *minibatch*

The loss can be described as:

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right)$$

or in the case of the *weight* argument being specified:

$$\text{loss}(x, class) = weight[class]\left(-x[class] + \log\left(\sum_j \exp(x[j])\right)\right)$$

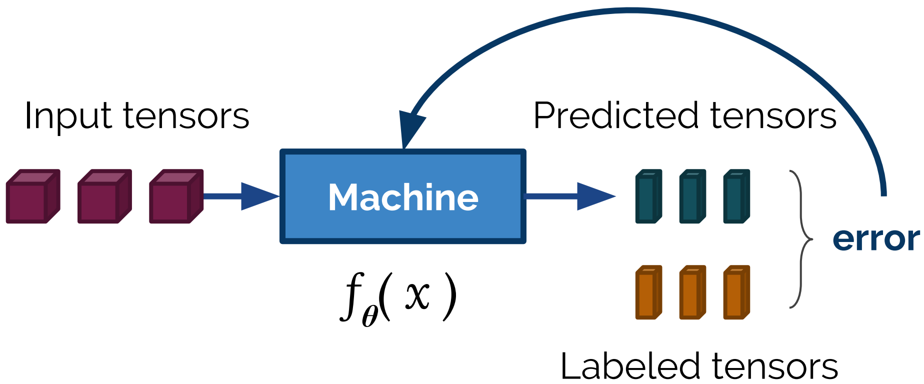
The losses are averaged across observations for each minibatch.

Can also be used for higher dimension inputs, such as 2D images, by providing an input of size $(minibatch, C, d_1, d_2, ..., d_K)$ with $K \geq 2$, where K is the number of dimensions, and a target of appropriate shape

Shortcuts

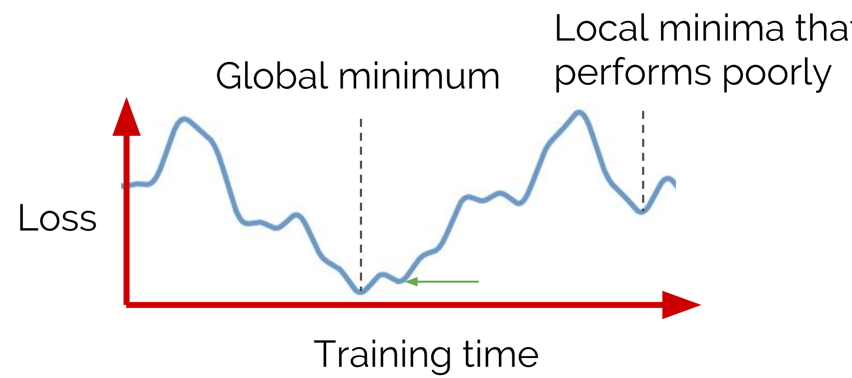
Further reading

Optimization

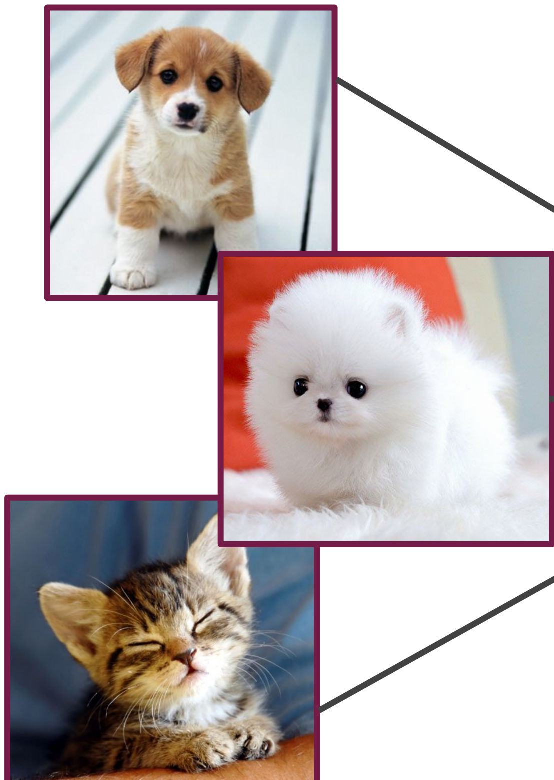


- We update the machine parameters $\boldsymbol{\theta}$ such as to minimise the objective function.
- We can choose an optimisation strategy based on the shape of the output space.

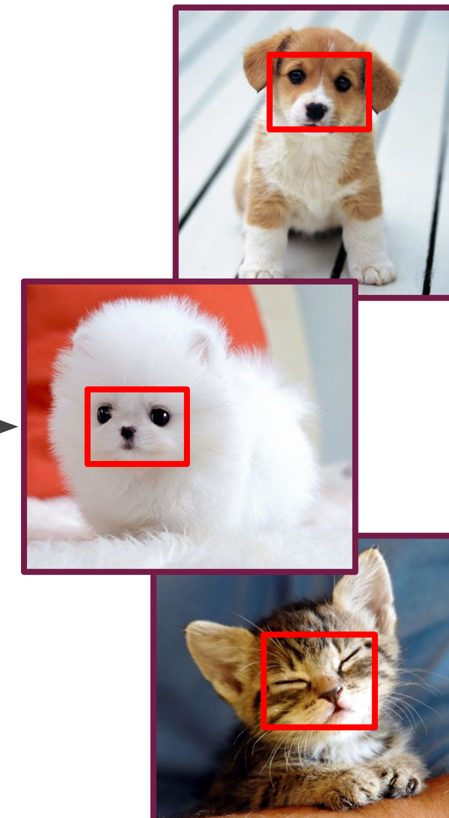
- We don't always want to find the global minimum - often this means the machine has simply memorised the input dataset
- Instead a good local minima such as the green arrow, may be sufficient.



Regression Example

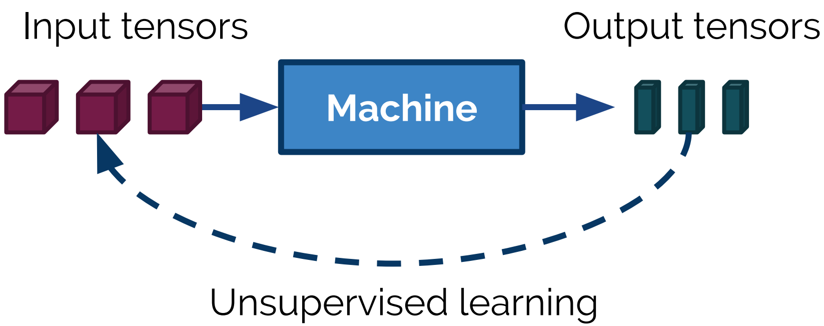


$$\text{func}(\textbf{image}) \rightarrow {\color{red}x,y,w,h}$$

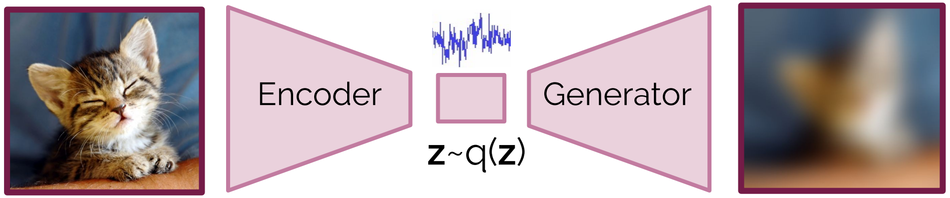


Note: this regression example is not a good way to "detect" boxes.

Unsupervised Learning



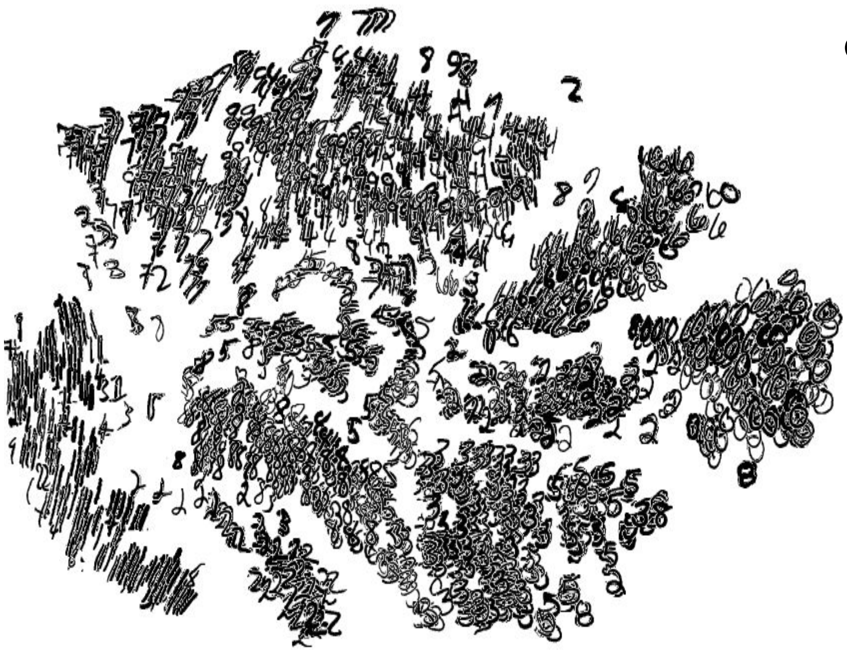
Example: reconstructive autoencoder

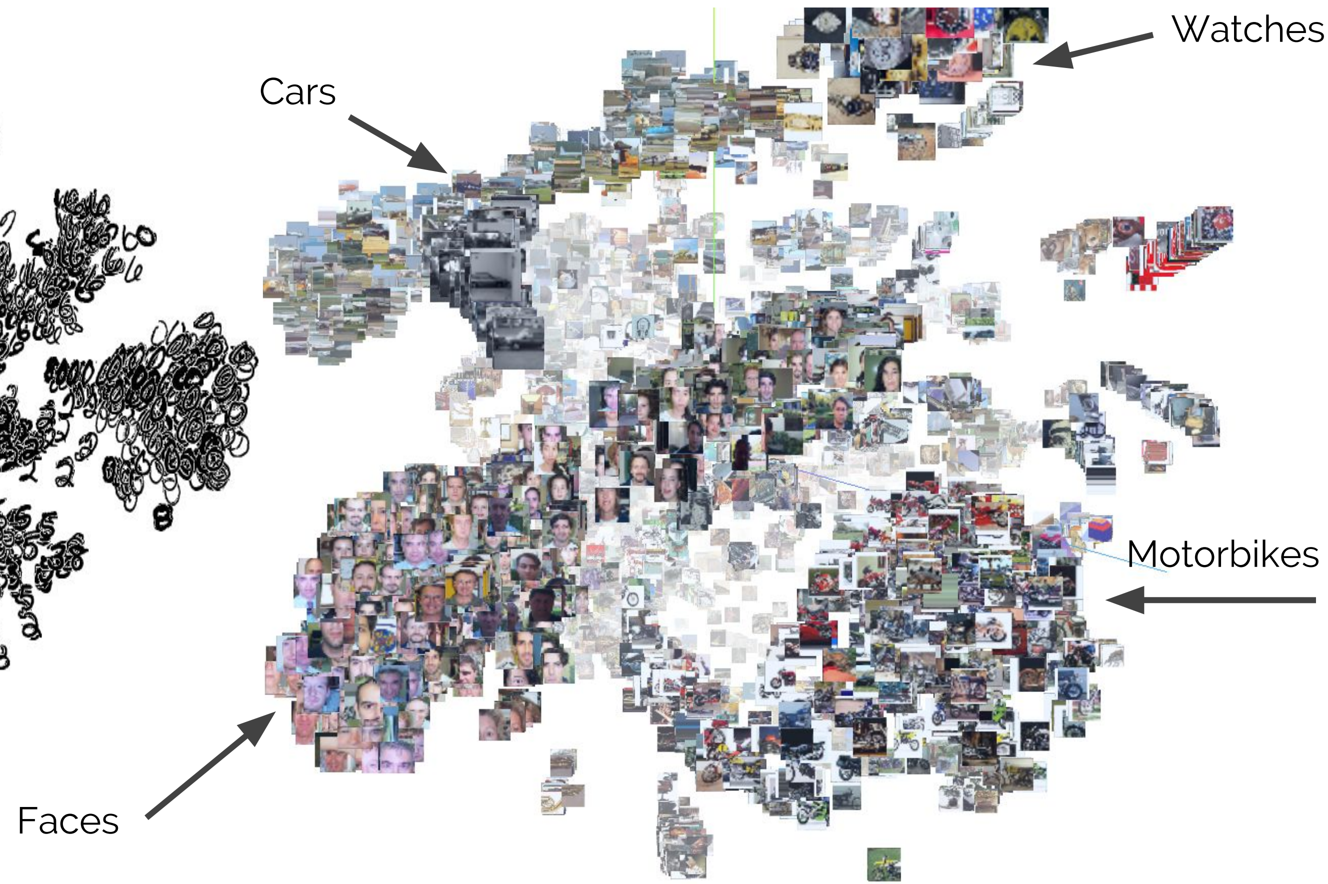


Example tasks:

- Generating new examples similar to the training data
- Inpainting (with some data removed, can predict value of missing entries)
- Denoising
- Density estimation (learn where examples cluster tighly and where they are unlikely occur)

Manifolds, Density Estimation, and Dimensionality Reduction





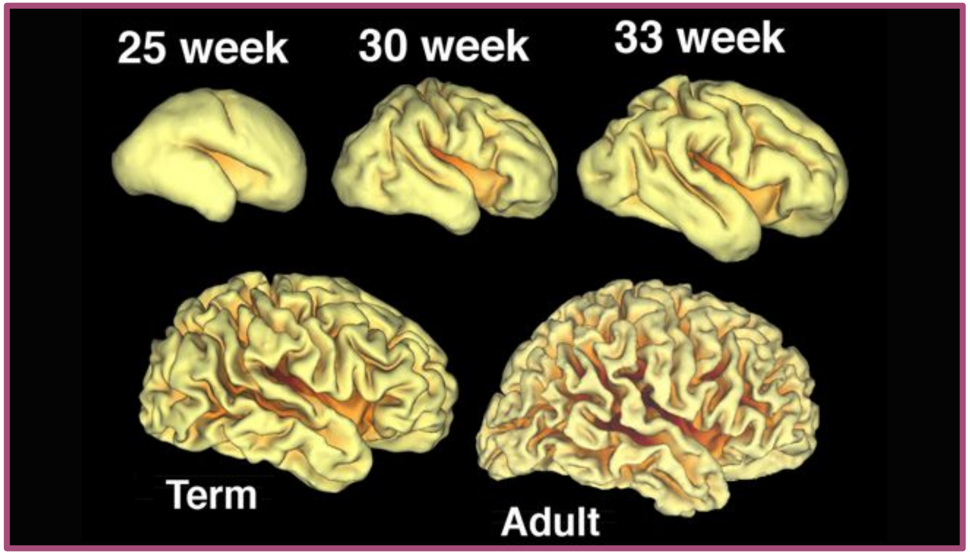
Transfer Learning

Durham University

```
# Load the model
model = torch.nn.DataParallel(resnet.resnet34(pretrained=True)).cuda()
```

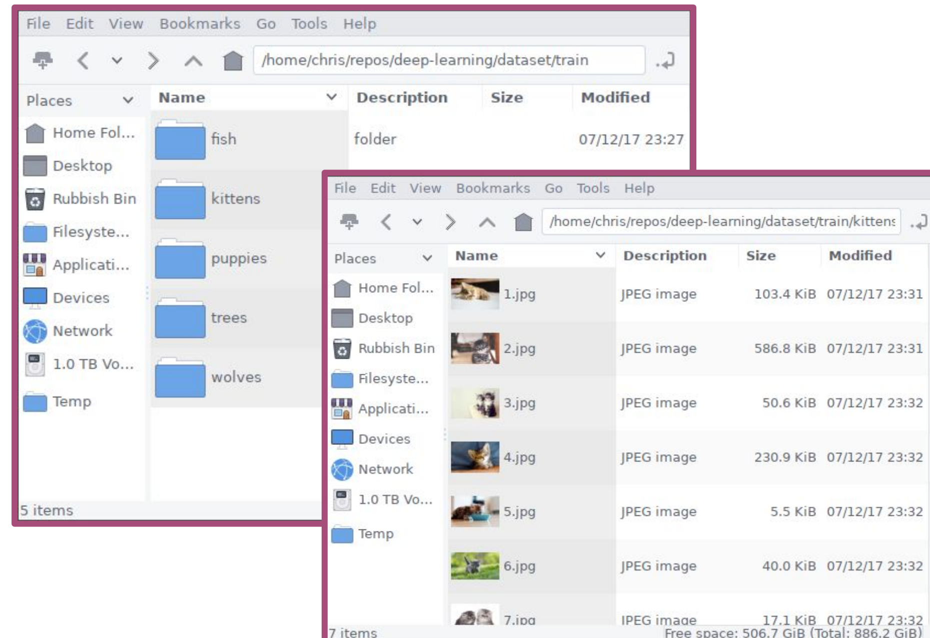
Nice!

- Is it easier to train a **baby** to detect cancer or an **adult** to detect cancer?
- Train model on complex tasks with **<u>lots</u>** of public data (even until they outperform humans)
- Then change the data to our tasks, and continue training



The Dataset

- Preparing a **good dataset is hard**
  - Garbage in, garbage out
- Learning the data distribution
- Train, test, and validate
- Overfitting and underfitting
- Bias
- Balance
- Augmentation



Source: https://github.com/albu/albumentations

Take away points

- Machine learning has lots of crossover with learning in nature
- Most research is very different and uses overly simplistic models
- Mainly advances in compute hardware and tooling have given rise of the past waves
    - GPUs
    - Automatic differentiation
- The field is currently mostly data driven
    - Especially for supervised learning
    - It's very easy to think a model is doing well, when it is actually just overfitting
- Next week, PyTorch!

Good books:
1. Deep Learning Book, Goodfellow
2. Pattern Recognition and Machine Learning, Bishop