Deep learning

1.1. From neural networks to deep learning

François Fleuret

https://fleuret.org/dlc/

UNIVERSITÉ DE GENÈVE

Many applications require the automatic extraction of “refined” information from raw signal (e.g. image recognition, automatic speech processing, natural language processing, robotic control, geometry reconstruction).

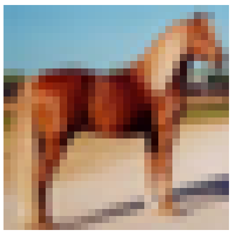
(ImageNet)

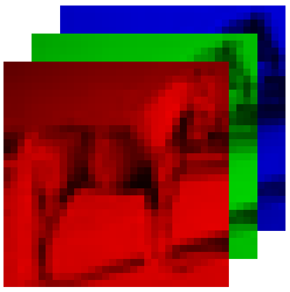
Notes

The core idea of machine learning is to write algorithms that depend on parameters whose values are let unspecified, and optimized to work on examples.
When the number of parameters is very large and the type of computation carefully chosen, these methods can “discover” rich and complex processings that would have been impossible to handcraft.
Although there are multiple forms of machine-learning models, most of them take as input a real world signal and output a refined information: semantic content (object classification), location of object (detection), word present in a audio signal (keyword spotting), meaning in a sentence (sentiment analysis). Some algorithms even take as input a random input to synthesis a structured signal: image, sound or text.
The task of automatically extracting the information of interest is difficult because of the large variability of the input signal for a given task.
Despite being obvious to the human eye that all the above images depict armchairs, it would be very difficult to come up with a hand-crafted algorithmic recipe taking as input the image pixels and predicting they represent an armchair.

Our brain is so good at interpreting visual information that the "semantic gap" is hard to assess intuitively.

This is a horse





Notes

When discussing the subject with people who are not from the field, it is intriguing to them that there is so much effort in making computers do what humans do so easily. Very often people do not realize that the problem actually exists.
The semantic gap is the difference there exists between a raw signal and its semantic content. For instance, two images can be very different in terms of pixel values, although depicting the same object. While it is even hard to be aware of the processing happening in our visual cortex when we look at an image such as the small vignette of a horse above, the larger pixelated image is slightly more difficult to parse since edges along the animal are not apparent anymore, while artificial pixel edges are. When the image is split into its three color component red/green/blue, that correspond to the representation in memory, our visual system has greater difficulty to understand the signal.

```
>>> from torchvision.datasets import CIFAR10
>>> cifar = CIFAR10('./data/cifar10/', train=True, download=True)
Files already downloaded and verified
>>> x = torch.from_numpy(cifar.data)[43].permute(2, 0, 1)
>>> x[:, :4, :8]
tensor([[[ 99,  98, 100, 103, 105, 107, 108, 110],
         [100, 100, 102, 105, 107, 109, 110, 112],
         [104, 104, 106, 109, 111, 112, 114, 116],
         [109, 109, 111, 113, 116, 117, 118, 120]],

        [[166, 165, 167, 169, 171, 172, 173, 175],
         [166, 164, 167, 169, 169, 171, 172, 174],
         [169, 167, 170, 171, 171, 173, 174, 176],
         [170, 169, 172, 173, 175, 176, 177, 178]],

        [[198, 196, 199, 200, 200, 202, 203, 204],
         [195, 194, 197, 197, 197, 199, 200, 201],
         [197, 195, 198, 198, 198, 199, 201, 202],
         [197, 196, 199, 198, 198, 199, 200, 201]]], dtype=torch.uint8)
```

Notes

In the memory of the computer, images are stored as tensors, which are multi-dimensional data structures storing the pixel values.
Tensors are truly what algorithms have access to operate on and solve the task they are trained for.
So an "image recognition" algorithms should predict that there is a horse in the input image from this table of integers.

Extracting semantic automatically requires models of extreme complexity, which cannot be designed by hand.

Techniques used in practice consist of

1. defining a parametric model, and
2. optimizing its parameters by "making it work" on training data.

This is similar to biological systems for which the model (e.g. brain structure) is DNA-encoded, and parameters (e.g. synaptic weights) are tuned through experiences.

Deep learning encompasses software technologies to scale-up to billions of model parameters and as many training examples.

Notes

In some very controlled environments such as in an automatic factory assembly line, it may sometimes be possible to design models by hand, but in most real-world vision problems images are prone to many variations due to illumination, geometric pose, occlusion, texture, articulated bodies, etc. which makes it impossible to design a model by hand to extract their semantic content.
The standard way of addressing the task of extracting a "refined" information from a high dimensional input signal consists of designing an algorithm with a lot of free parameters, that is known, for theoretical reasons, or by experience to compute the proper responses for adequate values of the parameters. These values are then optimized by a procedure on available training examples.
This process of designing a system whose parameters are changed to make it better at a task shares similarities with biological nervous systems, whose structure is fixed (DNA-encoded), but whose processing is modulated by quantities (synaptic weights) that are tuned through experiences.

There are strong connections between standard statistical modeling and machine learning.

Classical ML methods combine a “learnable” model from statistics (e.g. “linear regression”) with prior knowledge in pre-processing.

“Artificial neural networks” pre-dated these approaches, and do not follow this dichotomy. They consist of “deep” stacks of parametrized processing.

Notes

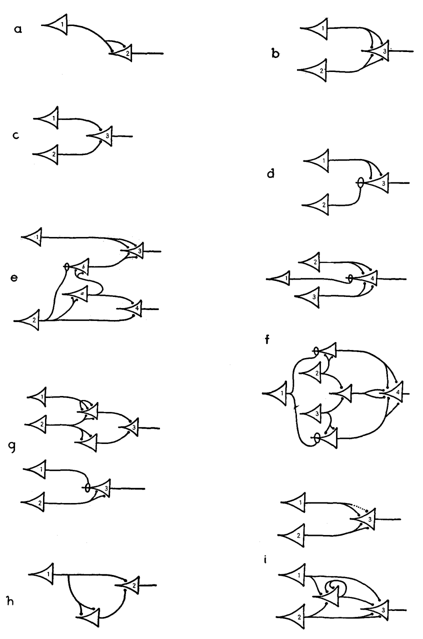
Most of standard statistical methods (e.g. logistic regression, linear regression) do not allow to deal with signals of very high dimensions such as images.
Therefore, we usually combine them with a hand designed pre-processing step which extracts a small number of meaningful quantities from the raw signal. Hopefully, this pre-processing step retains all the information content useful to make the prediction.
Classical machine learning methods follow this dichotomy of

- first, processing the signal to extract features in a ad-hoc manner,
- second, feeding these features to a statistical processing that makes a prediction, and can be tuned to work on training examples.

as opposed to artificial neural networks, which are series of parametrized processing units, each of them extracting meaningful values and making predictions at the same time.
The term “deep” in “deep learning” refers to the fact that many of these modules are stacked together.

From artificial neural networks to “Deep Learning”

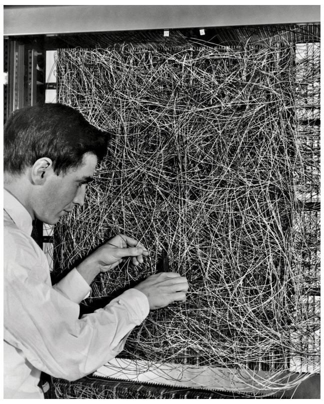
Networks of “Threshold Logic Unit”



(McCulloch and Pitts, 1943)

Notes

We can trace back the origins of neural networks to McCulloch and Pitts (1943) who proposed to model the nervous system as a network of “threshold logic units.” They suggested that one can put all the intelligence in the connections: elementary units doing very simple computation can perform an arbitrary mathematical function by being combined in an appropriate manner.
This opened the way to the notion that one can have a class of processing methods which are parameterized through the connections between units.

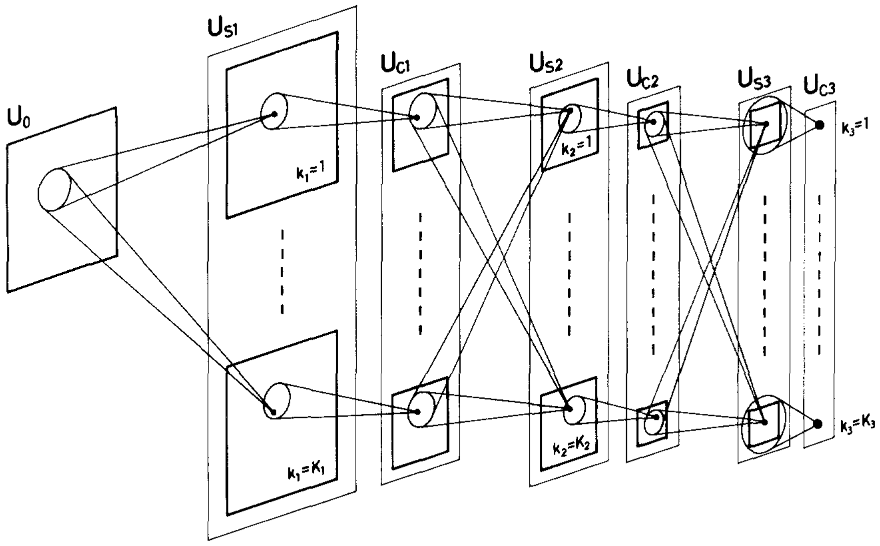

Frank Rosenblatt working on the Mark I perceptron (1956)

- 1949 – Donald Hebb proposes the Hebbian Learning principle (Hebb, 1949).
- 1951 – Marvin Minsky creates the first ANN (Hebbian learning, 40 neurons).
- 1958 – Frank Rosenblatt creates a perceptron to classify 20×20 images.
- 1959 – David H. Hubel and Torsten Wiesel demonstrate orientation selectivity and columnar organization in the cat's visual cortex (Hubel and Wiesel, 1962).
- 1982 – Paul Werbos proposes back-propagation for ANNs (Werbos, 1981).

Notes

The Hebbian Learning principle is a simple rule that allows to learn patterns and decision rules by reinforcing the connections between neurons when they tend to activate simultaneously. Although biologically plausible it is not used nowadays in machine learning.
A perceptron is the simplest form of neural network, composed of a single neuron.
Hubel and Wiesel's studies of the visual cortex of a cat showed that the visual information goes through a series of several processing steps: edge detections, combination of edges, detection of motion of edges, etc. These results built a strong bridge between the neural processing and the mathematical world, in particular signal processing.
The key component of deep learning is the back-propagation algorithm which was proposed by Werbos. Back-propagation is used to train neural networks and is a straight-forward application of the chain rule from differential calculus.

Neocognitron



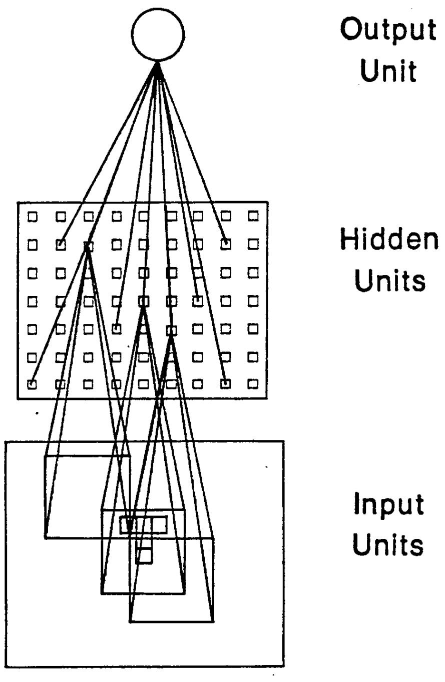
(Fukushima, 1980)

This model follows Hubel and Wiesel's results.

Notes

Fukushima (1980) implemented the results of Hubel and Wiesel in a model called the Neocognitron. It was used for handwritten character recognition and can be viewed as the precursor of modern convolution networks.

Network for the T-C problem



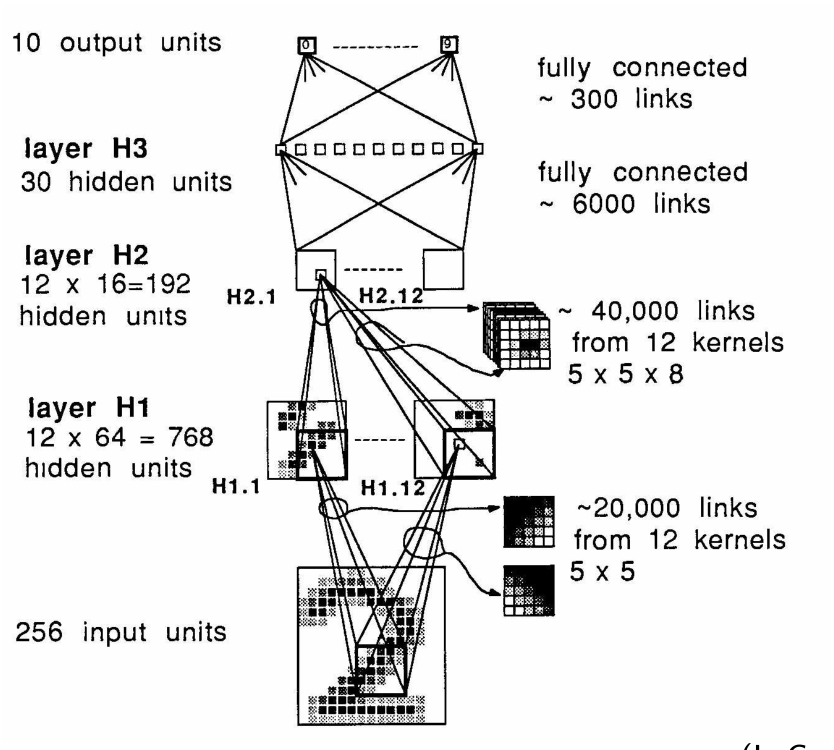
Trained with back-prop.

(Rumelhart et al., 1988)

Notes

Rumelhart et al. (1988) used back-propagation to train a network similar to the Neocognitron, and showed that the so-called "hidden" units, which are neither input nor output neurons, learn meaningful representation of the data.

LeNet family



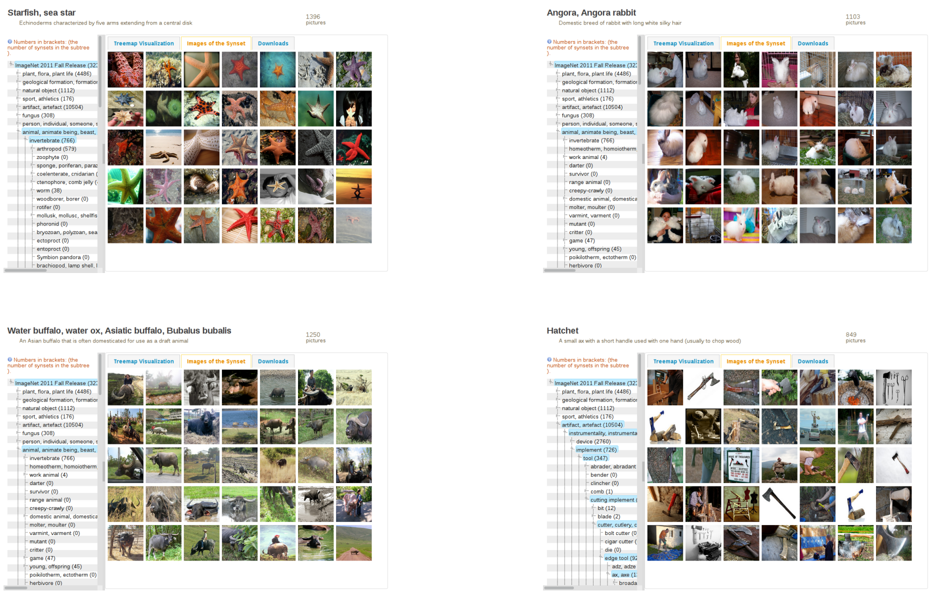
(LeCun et al., 1989)

Notes

LeCun et al. (1989) proposed a convolution neural network (CNN, or "convnet") very similar to modern architectures used nowadays.
As we shall see later on, a convnet is a series of "layers" which compute at every location of their input matching scores with small templates, and propagate the said matching scores to the next layer. These templates are optimized with variants of the back-propagation algorithm.

ImageNet Large Scale Visual Recognition Challenge.

Started 2010, 1 million images, 1000 categories



(http://image-net.org/challenges/LSVRC/2014/browse-synsets)

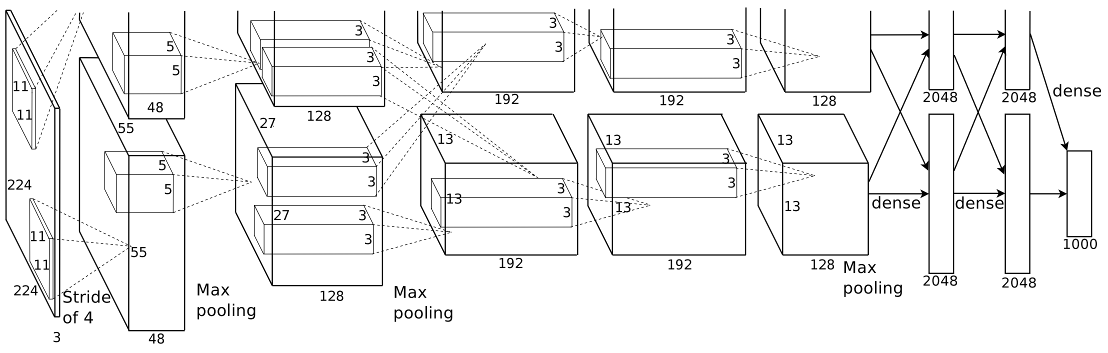
Notes

The availability of large amount of training data is critical to the success of deep-learning methods. ImageNet was started precisely to fullfill the need of machine learning, and the subset used to benchmark models is composed of more than a million of images organized in 1000 categories as diverse as "angora rabbit", "German shepherd", "acoustic guitar", or "school bus".
ImageNet was key in the development of deep learning because it is of the size required to train deep architectures.
Most image classification models are trained on this dataset, which is split in three parts: the training set, the validation (or dev) set, and the test set. The overall goal is to train a model on the training data, tune the hyper-parameters on the validation set, and finally evaluate the performance of the final model on the test set.
The testing part consists in:

- applying the model on each test image: the model returns a value between 0 and 999, corresponding to the class the model believes the image belongs to;
- then counting how many times the prediction of the model is right.

There are variants as well, such as the top-5 error rate, which is considering the prediction correct if the correct class is among the 5 first classes predicted by the network.
It is also common practice for many computer vision tasks, to start from a network that was trained on ImageNet, and to refine its training on another task and/or extend it.

AlexNet

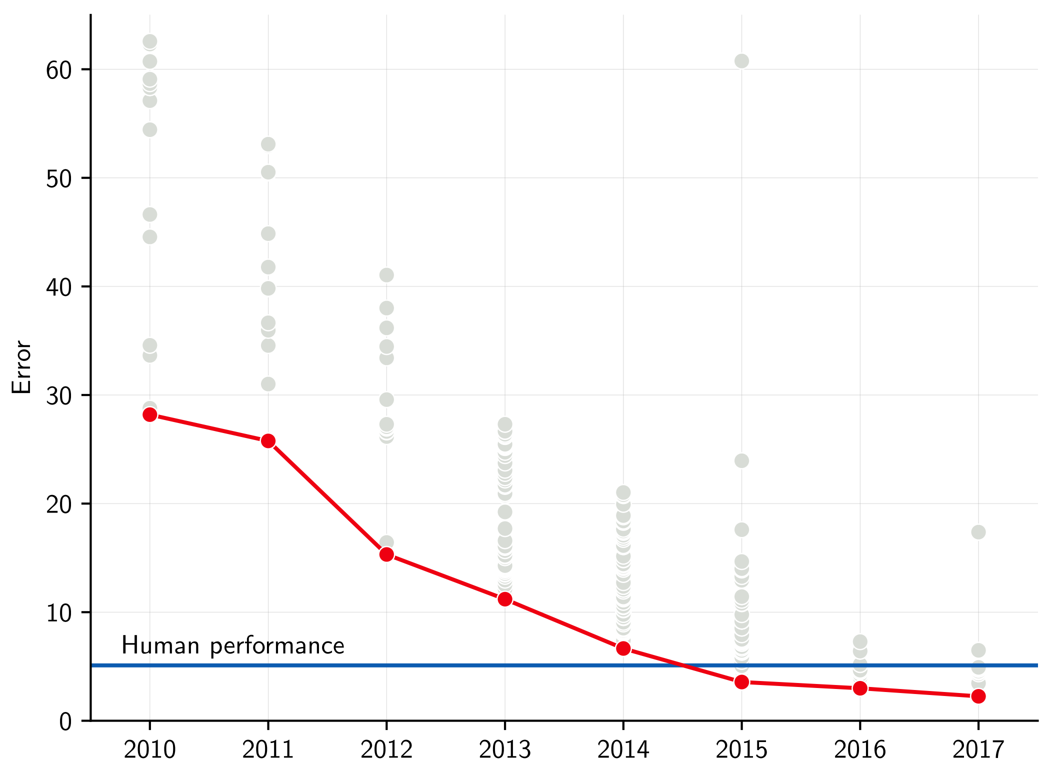


(Krizhevsky et al., 2012)

Notes

Following some earlier work from Cireșan et al. (2011), the work of Krizhevsky et al. (2012) showed that a network very similar to a LeNet5, but of far greater size, implemented on a graphical card could beat by a large margin state-of-the-art image classification methods on what was the reference benchmark of the community.
This work opened the way of training bigger networks on GPUs and started a new era of artificial neural networks.

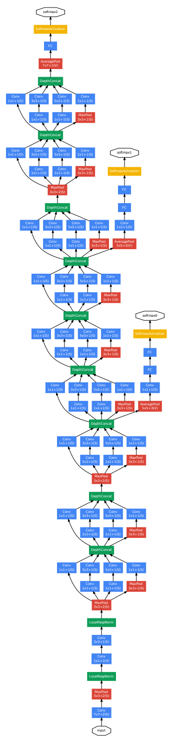
Top-5 error rate on ImageNet



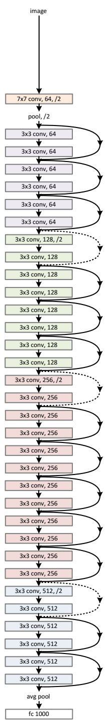
(Gershgorn, 2017)

Notes

Each gray dot on this graph shows the error rate of a model. The red line indicates the state-of-the-art performance each year, and the blue line shows the performance of humans asked to make the prediction, which can be seen as a gold standard.
A model may outperform humans if it picks statistical regularities that humans do not perceive, probably because of a bias in the data set.



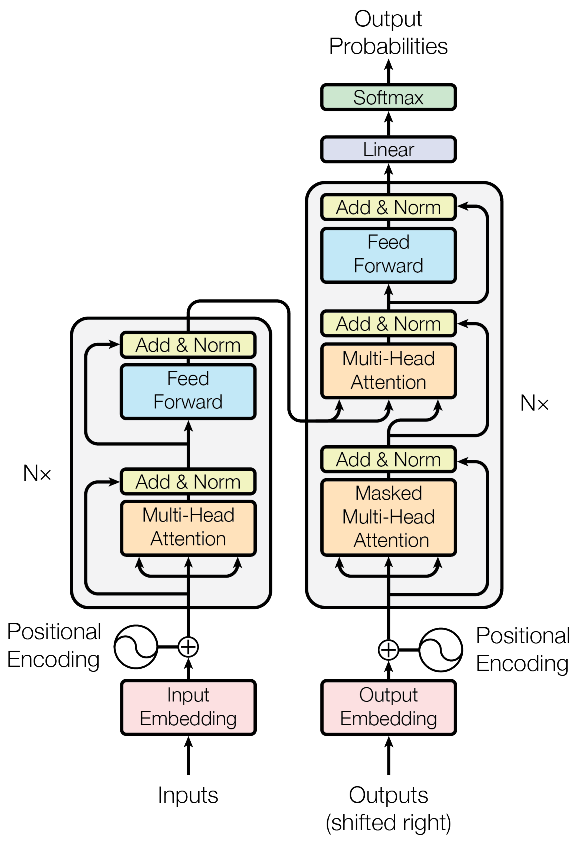
GoogleNet (Szegedy et al., 2015)



ResNet (He et al., 2015)

Notes

Alexnet initiated a trend toward more complex and bigger architectures.
GoogLeNet (Szegedy et al., 2015) contains several "inception" modules in a kind of fractal structure.
Residual networks (He et al., 2015) allow very deep networks thanks to "passthrough" connections which add the input of a layer to its output, and facilitate the training.



(Vaswani et al., 2017)

Notes

The Transformers are a class of deep architectures using attention-based computation, very popular for Natural Language Processing (Vaswani et al., 2017).
Some of these models for language modeling are of extremely large size, e.g. GPT-3 having 175 billion parameters (Brown et al., 2020).

Deep learning is built on a natural generalization of a neural network: **a graph of tensor operators**, taking advantage of

- the chain rule (aka "back-propagation"),
- stochastic gradient decent,
- convolutions,
- parallel operations on GPUs.

This does not differ much from networks from the 90s.

Notes

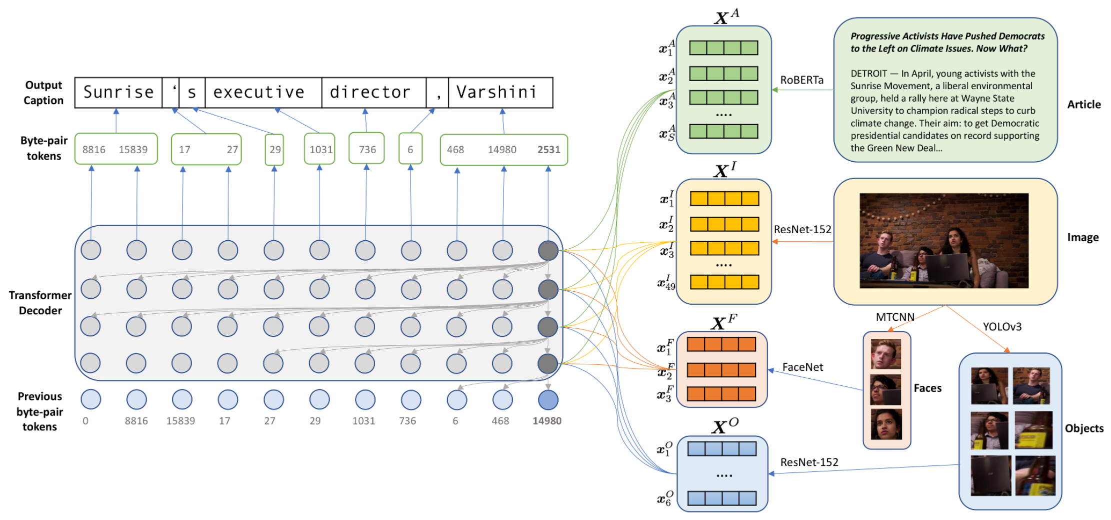
As we will see later in the course, an artificial neural network is a series of layers of neurons, each neuron connected to several neurons in the previous layer and sending activations to neurons that follow in the network.
Deep learning is "simply" a natural generalization of artificial neural networks by viewing the activities of a group of neurons as a multidimensional matrix, called a tensor.
A "deep model" can be formalized as a graph of tensor operators in which

- the nodes of the graph are operations,
- the results of the operation are propagated along the edges of the graph, until it reaches the output node.

The four main elements of a the deep learning technology are:

- the back-propagation which allows to compute how the quantity to optimize will change when changing slightly the model parameters. This directly comes from the chain rule from differential calculus;
- the stochastic gradient descent algorithm, which is a recipe to iteratively update the parameters of the network, until it fulfills its tasks;
- the convolutions, which leverage the fact that the signal is structured, and often has some stationarity properties. Convolutions allow the processing of large signals such as image, videos, or chunks of text. In an image for instance, it makes sense to use the same filter detecting an edge everywhere;
- the parallelization of operations to take advantage of highly efficient computing hardware (GPUs/TPUs).

This generalization allows to design complex networks of operators dealing with images, sound, text, sequences, etc. and to train them end-to-end.


Output Caption
Sunrise ' s executive director , Varshini
Byte-pair tokens
8816 15839 17 27 29 1031 736 6 468 14980 2531
Transformer Decoder
Previous byte-pair tokens
0 8816 15839 17 27 29 1031 736 6 468 14980
X^A
x_1^A x_2^A x_3^A x_S^A
RoBERTa
Progressive Activists Have Pushed Democrats to the Left on Climate Issues. Now What?
DETROIT — In April, young activists with the Sunrise Movement, a liberal environmental group, held a rally here at Wayne State University to champion radical steps to curb climate change. Their aim: to get Democratic presidential candidates on record supporting the Green New Deal...
Article
X^I
x_1^I x_2^I x_3^I x_{49}^I
ResNet-152
Image
MTCNN
YOLOv3
X^F
x_1^F x_2^F x_3^F
FaceNet
Faces
Objects
X^O
x_1^O x_6^O
ResNet-152

(Tran et al., 2020)

Notes

The paradigm of graph of operators allows to design architectures at a new level, where sub-modules themselves perform very complicated operations.
The work of Tran et al. (2020) aims at doing auto captioning from images, which is given an input image should produce a piece of text describing the content of it. The architecture they devised illustrates the modularity of complex deep models, and embeds for instance a full ResNet152 as a sub-processing for the image part.

References

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. **Language models are few-shot learners**. CoRR, abs/2005.14165, 2020.

D. C. Cireșan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. **Flexible, high performance convolutional neural networks for image classification**. In American Association for Artificial Intelligence Conference, pages 1237–1242, 2011.

K. Fukushima. **Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position**. Biological Cybernetics, 36(4):193–202, April 1980.

D. Gershgorn. **The data that transformed AI research—and possibly the world**, July 2017.

K. He, X. Zhang, S. Ren, and J. Sun. **Deep residual learning for image recognition**. CoRR, abs/1512.03385, 2015.

D. O. Hebb. The organization of behavior: A neuropsychological theory. Wiley, 1949. ISBN 0-8058-4300-0.

D. Hubel and T. Wiesel. **Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex**. Journal of Physiology, 160:106–154, 1962.

A. Krizhevsky, I. Sutskever, and G. Hinton. **Imagenet classification with deep convolutional neural networks**. In Neural Information Processing Systems (NIPS), 2012.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. **Backpropagation applied to handwritten zip code recognition**. Neural Computation, 1(4): 541–551, 1989.

W. S. McCulloch and W. Pitts. **A logical calculus of the ideas immanent in nervous activity**. The bulletin of mathematical biophysics, 5(4):115–133, 1943.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Neurocomputing: Foundations of Research, chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, 1988.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. **Going deeper with convolutions**. In Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

A. Tran, A. Mathews, and L. Xie. **Transform and tell: Entity-aware news image captioning**. In Conference on Computer Vision and Pattern Recognition (CVPR), pages 13035–13045, 2020.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. **Attention is all you need**. CoRR, abs/1706.03762, 2017.

P. J. Werbos. **Applications of advances in nonlinear sensitivity analysis**. In Proceedings of the 10th IFIP Conference, pages 762–770, 1981.