UNIVERSITÀ DEGLI STUDI DI TORINO

DIPARTIMENTO DI PSICOLOGIA

Corso di Laurea in Scienze del Corpo e della Mente

Tesi di Laurea Magistrale

Deep Learning and Neuroscience:
an integration

Relatore
Marco Tamietto

Correlatore:
C. Andrés Méndez

Laureando
Giancarlo Paoletti
matricola: 849899

Anno accademico 2018-2019

Contents

Chapter 1

Introduction

This thesis regards the interconnection between neuroscience and computer technology. The human brain, a complex computational system consisting by an estimated 100 billion neurons connected by 100 trillion synapses (Herculano-Houzel, 2012), represents one of the greatest frontiers in our understanding of ourselves (Cox and Dean, 2014). Efforts towards research in artificial intelligence (AI) have reached unprecedented levels during the last years, this progress came mainly from recent advancements from the deep learning sub-field of research. Deep learning refers to a particular type of neural networks which present multiple computational layers, capable of learning representations of data possessing multiple levels of abstraction (LeCun *et al.*, 2015).

These algorithms take inspirations and mimic features present in biological brains by the virtue of their brain-like computation. Their ability to emulate human learning and cognition lead, recently speaking, these neural networks to achieve human-level performance in many domains spanning through object recognition, image and video processing, speech recognition, motor control, adversarial games, and many more (LeCun *et al.*, 2015; Schmidhuber, 2015; Lake *et al.*, 2017). Deep learning algorithms demonstrate their effectiveness at discovering structures and pattern in high-dimensional data and proved to be reliably applicable in many domains, like science, government, business, and many more. Therefore, neuroscientists seek as well advantages over these neural networks:

- Neuroscientists look to artificial neural networks as a research tool for the interpretation of neurobiological phenomena, to help them to reveal the underlying processes of the brain.

- Engineers look to neurobiology for new ideas to solve problems more complex than those based on conventional design techniques.

A particular type of deep neural networks, called convolutional neural networks, possess features which are directly inspired by neuroscientific notions of simple cells and complex cells in visual neuroscience and his structure is reminiscent of the LGN–V1–V2–V4–IT hierarchy in the visual cortex ventral pathway (Felleman and Van, 1991; Cadieu *et al.*, 2014; LeCun *et al.*, 2015).

Although history has shown the ups and downs about interest in conjugate neuroscience and artificial intelligence, nowadays this gap is increasingly thinning. This growing importance of deep neural networks in neuroscience grants synergic work to shed light upon how the brain works, and neuroscientists seek to implement in their research work these neural networks as an effective tool, together with recent and state-of-the-art neuroimaging tools.

In line with the spirit of this aforementioned integration, the former part of this thesis introduces to the reader the key components of neural networks and the latter illustrate the application of deep learning algorithms in neuroscience.

Namely, chapter 2 is a compendium of neural networks features, explaining the classical algorithms (sections 2.2, 2.5, 2.6) and how the learning takes place (section 2.9).

Chapter 3 explains one of the most used deep neural networks, the convolutional neural networks, also including a historical section (3.3) where were reported the principal milestones towards current implementations of these neural networks.

Chapter 4 is a bibliographical survey about the implementation of deep neural networks in neuroscience, starting from the similarities between these networks and the human vision system (section 4.1) and progressing through the implementation of aspects like visual attention (section 4.2) and neuroimaging (section 4.4).

Chapter 2

Building blocks of neural networks

2.1 Basic introduction of neural networks

As explained in Haykin (2009), a **Neural Network** is a "*massively parallel distributor processor* capable of storing experiential knowledge and making it available for use". The structure of the network and its learning features resembles those present in the human brain. Either in a biological and artificial neuron, knowledge is obtained through a *learning process*, the information acquired from the surrounding environment is then stored thanks to the inter-neuron connection strengths (known as *synaptic weights*). Therefore, the *learning algorithm* permits this procedure, which its function is to modify the network's synaptic weights in order to attain the desired objective. This dissertation is focused on a popular paradigm of learning called **supervised learning** (or *learning with a teacher*), this type of learning involves the application of two sets of labeled examples, called *training set* and *test set*, each set example consists of a unique input signal and a corresponding desired response.

In the **training set**, the network is presented with an example or a subset chosen randomly from this set, the network's weights are modified accordingly to minimize the *error* produced (the discrepancy between the desired response and the actual response of the network produced by the input signal, see section 2.8). The training of the network is repeated until reaches a steady state where there are no further significant changes in the weights. The **test set** consists of inputs not encountered during the training process. The ability to generate outputs from this set is therefore called *generalization* (see section 2.11).

2.2 Biological and Artificial neuron comparison

2.2.1 The human nervous system

Arbib (2012) depicts the human nervous system as a three-stage system, the brain represents the central point of the system, which continually receives information, perceives it and makes appropriate decisions. Internal (coming from the human body itself) or external (coming from the environment) *stimuli* are converted into electrical impulses by the *receptors* which convey information to the brain. The *effectors* convert the impulses coming from the brain into *responses*, seen as system outputs.

Synapses mediate the interactions between neurons and its most common kind is a chemical one: a synapse converts a presynaptic electrical signal into a chemical signal and then back into a postsynaptic electrical signal (Sheperd and Koch, 1990). *Plasticity* permits the developing nervous system to adapt to its surrounding environment (Eggermont, 1990; Churchland and Sejnowski, 1992), through the creation of new synaptic connections between neurons and the modification of existing synapses. One of the most common types of cortical neuron is the *pyramidal neuron*. In the human nervous system, approximately 86 billion neurons are connected with approximately 10^{14} - 10^{15} synapses. Each neuron receives input signals from its dendrites and produces output signals along its (single) axon, which branches out and connects via synapses to dendrites of other neurons (Li *et al.*, 2018a).

2.2.2 The artificial neuron

The similarity between biological neuron (left) and the artificial one (right) is reported in figure 2.1. The first and basic model of an artificial neuron, which sets the grounding block for future AI research and neural networks development, was implemented by McCulloch and Pitts (1943).

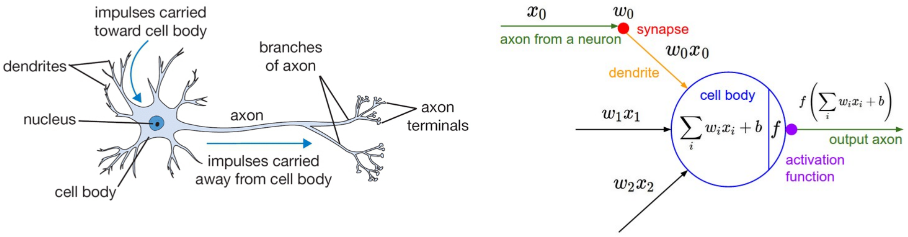


Figure 2.1: A biological and artificial neuron comparison, taken from Li *et al.* (2018a).

The **McCulloch and Pitts neuron** performs a series of mathematical operations depicted in figure 2.2 (left), some notations are required, to better understand the underlying functions of the artificial neuron:

A set of *inputs* x_j $(x_1, x_2, ..., x_m)$ are fed into the neuron. Each input has a *weight* w_{kj} $(w_{k1}, w_{k2}, ..., w_{km})$, a connecting link which determines its strength with a range between positive and negative real number value.

The *adder function* is a weighted sum operation, consist of the summing of all inputs multiplied with their respective weight, it is also referred as a linear combiner and the result of this operation is called *local net* u_k.

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.1}$$

The *activation function* $\varphi(\cdot)$ limits the amplitude of the neuron's *output* y_k, it is also referred as a *squashing function* because it limits the permissible amplitude range of the output signal to some finite value. The value range depends on which activation function is used (see section 2.3). In biologically-inspired neural networks, the activation function mimics the rate of action potential firing in the cell (Hodgkin and Huxley, 1952).

$$y_k = \varphi(v_k) = \varphi(u_k + b_k) \tag{2.2}$$

The *bias* b_k has the effect of applying an *affine transformation* to the output signal of the neuron, increasing or lowering the net input of the activation function (depicted in figure 2.2, right). It can be implemented into the neuron as an input x_0 with weight w_{k0} $(w_{k0} = b_k)$. Therefore, the *induced local field* v_k is defined as:

$$v_k = u_k + b_k = \sum_{j=0}^{m} w_{kj} x_j \tag{2.3}$$

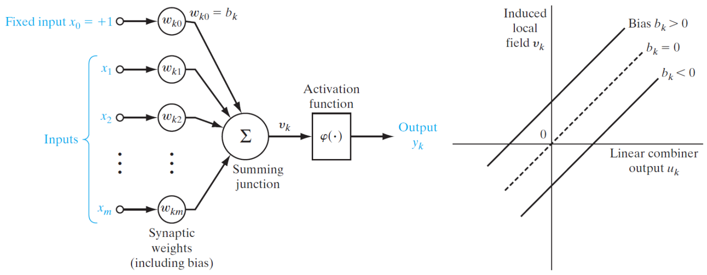


Figure 2.2: McCulloch and Pitts neuron (left) and affine transformation produced by the bias (right), taken from Haykin (2009).

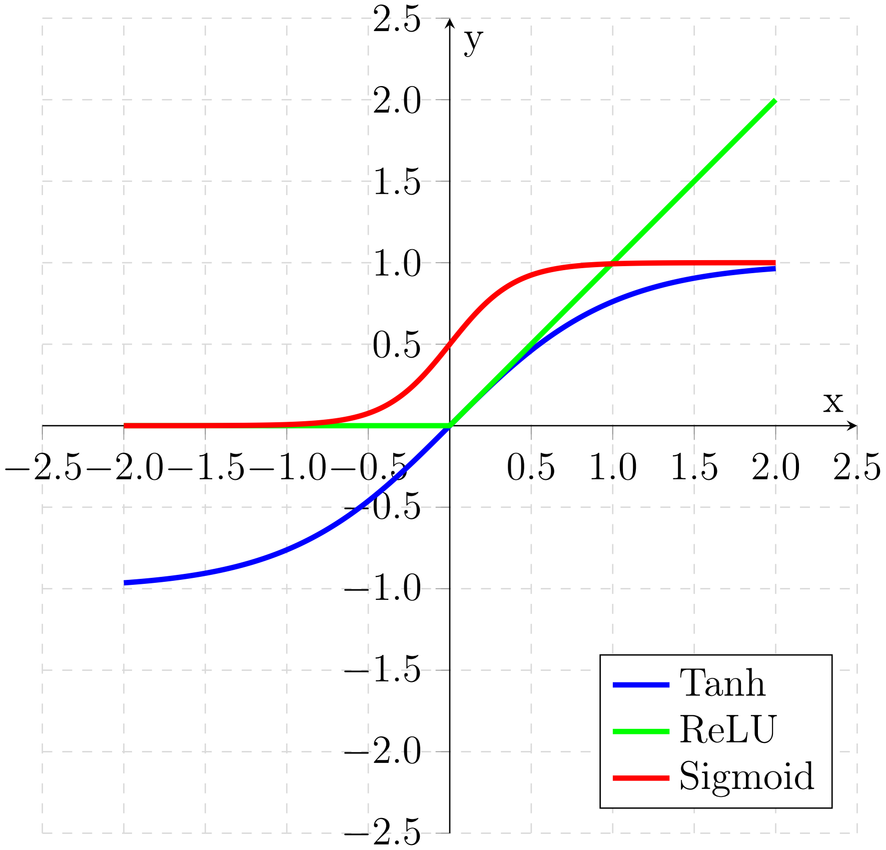


Figure 2.3: Three different activation functions.

2.3 The activation function

Figure 2.3 reports the most used activation functions in neural networks:

- *Sigmoid function*: a nonlinear function which amplitudes the local field of the neuron j within a range between 0 and 1 ($0 \leq y_j \leq 1$).

  $$\varphi_j(v_j(n)) = \frac{1}{1 + \exp(-av_j(n))} \tag{2.4}$$

  a is the slope parameter used to determine the steepness of the curve.

- *Hyperbolic tangent function*: another form of sigmoidal nonlinear function, can be viewed as a sigmoid function rescaled and biased to be fit in a range between -1 and 1 ($-1 \leq y_j \leq 1$)).

  $$\varphi_j(v_j(n)) = \frac{\exp(v_j(n)) - \exp(-v_j(n))}{\exp(v_j(n)) + \exp(-v_j(n))} \tag{2.5}$$

- *Rectified linear unit function*: relu, an activation function used extensively on convolutional neural networks (Jarrett *et al.*, 2009; Glorot *et al.*, 2011), it gives a linear output for all positive values, (or 0 if otherwise). This activation function does not suffer of the *vanishing gradient problem* (see section 3.1.2) which can occur when using sigmoid or tanh functions.

$$\varphi_j(v_j(n)) = max(0, v_j(n)) \tag{2.6}$$

- *Softmax function*: also referred as normalized exponential function, it is a probability distribution function used on the final layer of a classic or prototypical convolutional neural network architecture (Goodfellow *et al.*, 2016; Liu *et al.*, 2016a; Rawat and Wang, 2017). Softmax is a function that takes as input a vector of K real numbers and normalizes it into a probability distribution, consisting of K-probabilities i.

Each component will be in the interval (0,1), and the components will add up to 1, so that they can be interpreted as probabilities. Larger input components correspond to larger probabilities. The softmax function is used to map the non-normalized outputs to a probability distribution over j predicted different classes included in the training set.

$$softmax(z_i) = \frac{\exp{(z_i)}}{\sum_j \exp z_j} \quad for\ j = 1, \ldots, K \tag{2.7}$$

2.4 Layers in neural networks

The structure and organization of neurons are different between the various neural network architectures. Neurons are organized in the form of *layers*:

- *Single-layer* feed-forward networks: the simplest form of a layered network, (figure 2.4, left), composed only by an input layer and an output layer of neurons. The term feed-forward stands for the direction flow of the information: from the input layer to the output layer but not vice-versa.

- *Multi-layer* feed-forward networks: an augmented version of the above networks (figure 2.4, right), with the addition of one or more *hidden layers*, where this latter type of layers receives input from the previous layer of neurons and propagates the output data into the next layer of neurons.

The term **deep learning** indicates a network composed of more than 2 hidden layers, hence the term deep. Some algorithms like convolutional neural networks even contain hundreds of hidden layers.

2.5 The perceptron

Once the idea of the artificial neuron was introduced by McCulloch and Pitts (1943), in the later years the psychologist Frank Rosenblatt (1958) published an article which explains the first algorithmically described neural network.

The **perceptron** is a simple form of a neural network and it is used for classifying two types of data or patterns which are said to be *linearly separable* (i.e. data that

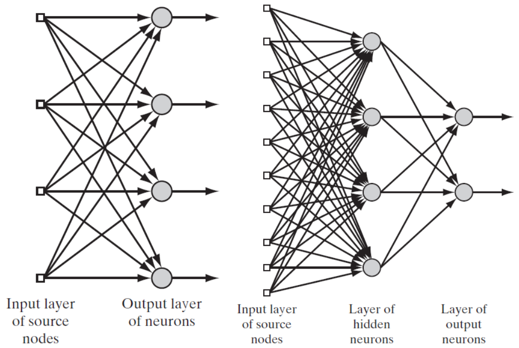


Figure 2.4: Single-layer and multi-layer feed-forward network, taken from Haykin (2009).

lies on a graph which can be separated into two distinct regions by a hyperplane). Its structure consists of a single layer of neurons which receive m inputs with adjustable weights and bias. Its architecture is built around the concept of McCulloch and Pitts neuron (see section 2.2.2).

The goal of the perceptron is to classify a set of inputs $x_1, x_1, ..., x_m$ into two distinct classes: $\mathscr{C}_1$ if the output y is $+1$ or $\mathscr{C}_2$ if the output y is -1. The activation function is a *hard limiter*: if the weighted sum of all inputs is positive the output y is equal to $+1$ (or -1 otherwise).

Graphically this classification can be viewed by the mean of a *decision boundary* equation, this hyperplane separates the graph space: a point (x_1,x_2) that lies above the boundary line is assigned to the class $\mathscr{C}_1$, otherwise the point that lies below is assigned to the class $\mathscr{C}_2$ (see figure 2.5).
The equation of the decision boundary is: $\sum_{i=1}^{m} w_i x_i + b = 0$

The major flaw of the Rosenblatt's perceptron lies where it works best: it generalizes well only for resolving a problem that is linearly separable (as famously pointed by Minsky and Papert, 1969), thus it is inherently incapable of making some global generalizations apart from the aforementioned classification problem. Minsky and Papert's book cast a long shadow on neural networks research (also referred as *the first AI winter*, see section 3.3.3) up until mid-1980s, where the multi-layer Perceptron and its backpropagation algorithm (Rumelhart *et al.*, 1986) gave new life and interest in this research area (see section 3.3.5).

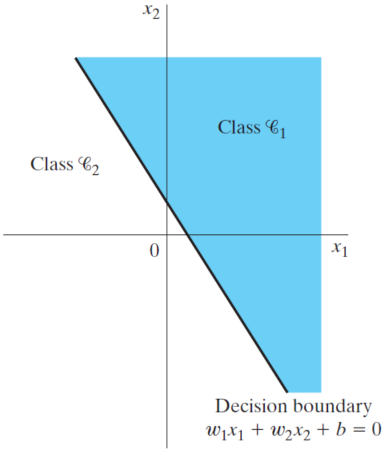


Figure 2.5: Hyperplane as a decision boundary, taken from Haykin (2009).

2.6 The multi-layer perceptron

Widrow and Hoff (1960), with their Least Mean Squared error algorithm, introduced the notion of neural network learning via *error reduction* (the iterative process of calculating the misclassification error between an output of the neural network and its desired value coming from the dataset). However, this network uses the same architecture of the perceptron, hence its limitations.

To overcome these limitations, but retaining the error-reduction concept, a **multi-layer perceptron** is introduced. Haykin (2009) highlights three major basic features of this network:

- The activation function of each neuron is *nonlinear* and *differentiable*. The introduction of a nonlinear function is less constraining than a hard limiter function like the one used in the perceptron (the function of choice of a multi-layer perceptron is usually a *sigmoid function*, see section 2.3).
- The network contains one or more *hidden layers* put in between the input and output layers.
- The network exhibits a high degree of connectivity thanks to the greater number of neurons and their weight connections between them.

An example model of a multi-layer perceptron is depicted in figure 2.6, consisting of (left to right) an input layer, two hidden layers, and an output layer. This network is *fully-connected*, meaning that a neuron in any layer is connected to all the neurons of the previous layer.

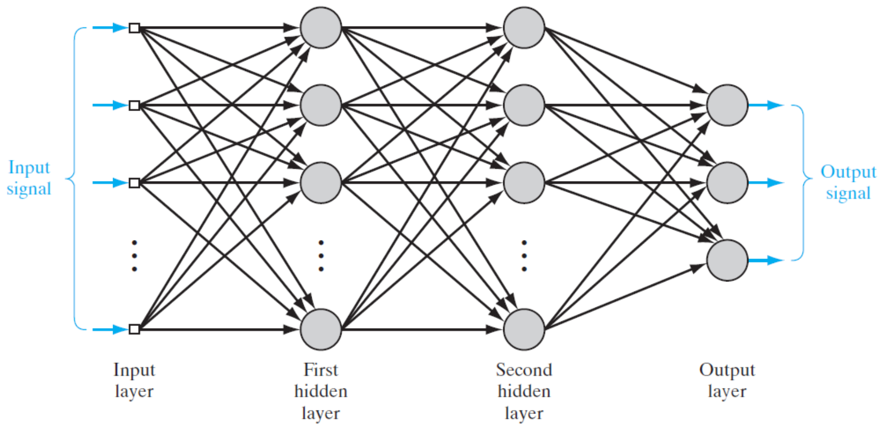


Figure 2.6: A fully-connected multi-layer perceptron with two hidden layers, taken from Haykin (2009).

The training process is possible through two distinct phases:

- *Forward phase*: the input signal is propagated through the network layer-to-layer up until reaches the output layer and produces a *function signal.*
- *Backward phase*: an error signal is generated in the output layer with a *loss function*: the comparison between the function signal with the desired response (see section 2.8). This error signal is backpropagated in opposite direction with the purpose of modifying the weights, after this phase a new forward phase starts again and therefore the error signal is reduced iteration by iteration.

The most popular method for training a multi-layer perceptron is the **backpropagation algorithm** (see section 2.9). This term was popularized through the publication of the seminal book of Rumelhart and McClelland (1986) and giving birth to the *second wave of AI* (see section 3.3.5).

The hidden layers are introduced because they act as feature detectors: as the learning process progresses the hidden neurons begin to discover gradually the salient features that characterize the training data set. The more the hidden layers, the more is the generalization achieved (at a cost of computational expense).

2.7 The feed-forward learning

The core of the multi-layer perceptron's learning lies on the automatic adjustment of the synaptic weights on the neuron, in accordance with the error signal generated at the end of the feed-forward phase. Since the multi-layer perceptron operates by the means of the *supervised learning*, the training and test datasets $\mathscr{T}$ contain the sample pairs of input data and their desired output.

$$\mathscr{T} = \{\mathbf{x}(n),\ \mathbf{d}(n)\}_{n=1}^{N} \tag{2.8}$$

where $\mathbf{x}(n)$ and $\mathbf{d}(n)$ represent the vectors contained in the training data set, respectively the n-th element of input data and its desired output value. To get an error signal the network must first be trained, this process is made possible thanks to a series of concatenated function (see figure 2.7 for an overview). During this feed-forward pass the synaptic weights remain unaltered (they will be adapted during the later backpropagation phase) and neuron-wise outputs are computed: the *function signals* $y(n)$ at neuron j are computed as

$$y_j(n) = \varphi(v_j(n)) \tag{2.9}$$

where $v_j(n)$ is the *induced local field* and $\varphi(\cdot)$ is the *activation function*, which limits the amplitude of the output in a value range depending on which activation function is used (see section 2.3).
The *induced local field* $v_j(n)$ is often referred as a *weighted sum* of all the neuron's inputs

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \tag{2.10}$$

m is the total number of inputs (excluding the bias) applied to neuron j. $w_{ji}(n)$ is the synaptic weight connecting neuron i to neuron j. $y_i(n)$ is the input signal of neuron j, it is the output signal which comes from the neuron i of the previous layer). If neuron j is in the first hidden layer of the network, the index i refers to the i-th input terminal of the network: hence $y_i(n) = x_i(n)$.

To recap all the process, figure 2.7 depicts the forward phase of computation: it begins at the first hidden layer by presenting it with the input vector $x_i(n)$ and terminates at the output layer by computing the error signal for each neuron of this layer.

$$y_j(n) = \varphi\left(\sum_{i=0}^{m} w_{ji}(n) y_i(n)\right) \tag{2.11}$$

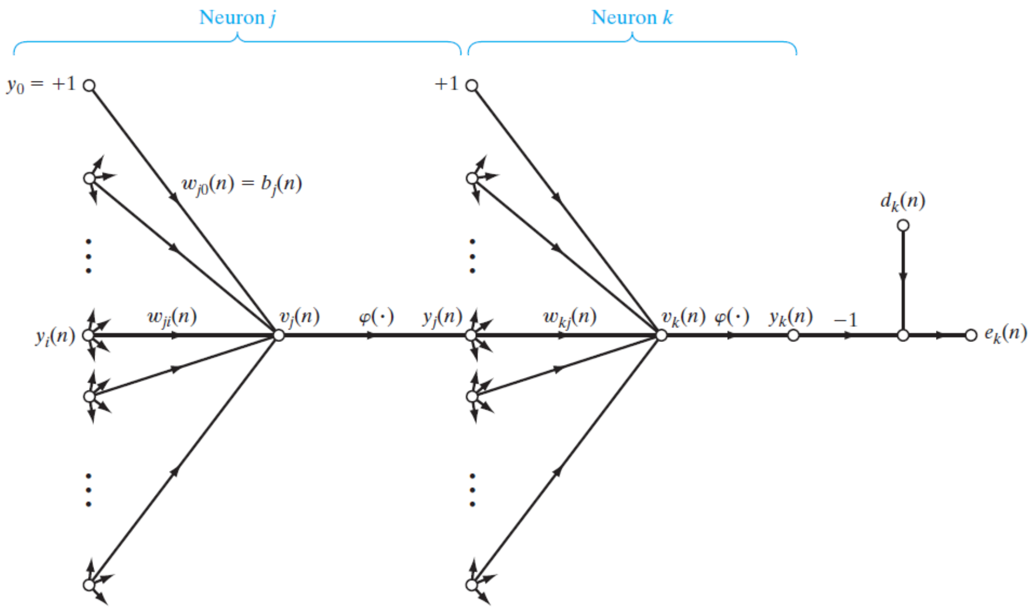


Figure 2.7: The signal graph of the multi-layer perceptron, taken from Haykin (2009).

2.8 The loss function

An optimization problem seeks to minimize a *loss function*: also referred as cost or objective function. It is a function that maps values of one or more variables onto a real number, intuitively representing some "cost" associated with the event.

- The ***mean squared error*** (MSE) measures the average of the squares of the errors: the average squared difference between the estimated values and what is estimated. It is a measure of the quality of an estimator: the closer the values are towards zero, the better is the estimation. It is typically used in classical neural networks such as the multi-layer perceptron.

  $$MSE(x,y) = \frac{1}{n}\sum_{i} |x_i - y_i|^2 \quad for\ i = 1, \dots, n \tag{2.12}$$

- The ***cross-entropy*** loss, also referred as log loss or multinomial logistic regression, measures the performance of a neural network trained for a classification task, whose output is a probability value between 0 and 1.

  It increases as the predicted probability diverges from the actual label: indicates the distance between what the network believes this distribution should be and what the teacher says it should be (Plunkett and Elman, 1997). The cross-entropy $CE(p,q)$ between two discrete probability distributions $p(x)$ and $q(x)$ measures how much the predicted value of the currently given distribution q differs from the true probability value p.

  $$CE(p,q) = -\sum_{x} \ p(x) \ * \ \log q(x) \tag{2.13}$$

  - Binary cross-entropy loss is used for a binary classification problem between two classes and represent the cross-entropy with a sigmoid activation function (see section 2.3).
  - Categorical cross-entropy loss is used for multi-class classification and represent the cross-entropy with a softmax activation function (see section 2.3). Categorical cross-entropy is typically used in convolutional neural networks because they often use a fully-connected classificator as output layers.

2.9 The backpropagation algorithm

The strength of a neural network's learning consists of the modification and update of the synaptic weights, so that in the next iteration the error signal is reduced. The goal of the network is to reduce the global error signal by the means of the **stochastic gradient descent**: the term *stochastic* refers to the random selections of input samples (instead of selecting them as in the order they appear in the training set). By lowering the error signal, the network consequently improves its accuracy rate (see section 2.10).

The graph reported on figure 2.8 (left) explains this correlation: adjusting the value of the weights (reported on axis X) affects also the value of the error signal (reported on axis Y). This procedure iterates many times (also referred as *epochs*) up until it reaches a point where no significant error changes occur (also called convergence to a *minimum*).

This error-reduction procedure is made possible by *backpropagating* the error signal throughout the network. The **backpropagation** phase starts at the output layer by passing the error signal throughout the network (layer by layer) and recursively computing the *local gradient* (δ, equation 2.20) for each neuron. This recursive process permits the synaptic weights of the network to change, in accordance with an operation called *delta rule.*

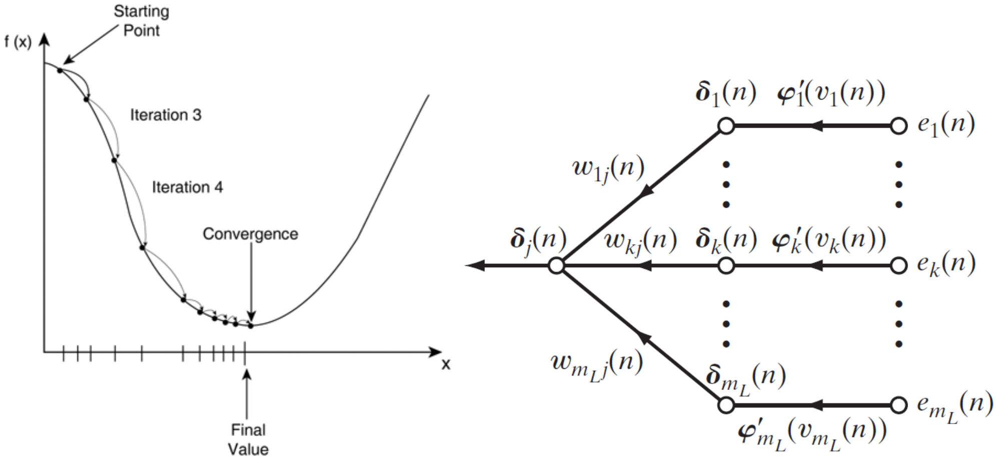


Figure 2.8: Gradient descent graph and backpropagation signal flow, taken from Caparrini (2017) and Haykin (2009).

The *error signal* $e_j(n)$ defines how much the *function signal* $y_j(n)$ (produced at the output neuron j) differs from the *desired value* $d_j(n)$ (the n-th element of the vector $\mathbf{d}(n)$).

$$e_j(n) = d_j(n) - y_j(n) \tag{2.14}$$

The total *instantaneous error energy* is defined by the sum of all error signals of the output layer neurons:

$$\mathscr{E}(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n) \tag{2.15}$$

where the set C includes all the neurons in the output layer and the $\frac{1}{2}$ is included for calculus purposes (the derivative of a squared value is $\partial x^2 = 2x$).

The weight $w_{ji}(n)$ correction, applied by the backpropagation algorithm, is called *delta rule*:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} \tag{2.16}$$

where η is the *learning rate* parameter of the backpropagation algorithm. The use of the *minus* sign denotes the *gradient descent* in weight space: the *opposite* direction of the error value $\mathscr{E}(n)$.

The derivative of the error with respect of the derivative of the weight which has to be updated, thanks to the *chain rule of calculus*, can be expressed as a series of partial derivative operations (refer to figure 2.7 for a better comprehension of the results of the operations):

$$\frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathscr{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{2.17}$$

$$\frac{\partial \mathscr{E}(n)}{\partial e_j(n)} = e_j(n), \; \frac{\partial e_j(n)}{\partial y_j(n)} = -1, \; \frac{\partial y_j(n)}{\partial v_j(n)} = \varphi_j'(v_j(n)), \; \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

$$\frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} = -e_j(n)\; \varphi_j'(v_j(n))\; y_i(n) \tag{2.18}$$

The inclusion of equation 2.18 in equation 2.16 updates the delta rule as:

$$\Delta w_{ji}(n) = \eta e_j(n)\; \varphi_j'(v_j(n))\; y_i(n) \tag{2.19}$$

The product of the corresponding error signal $e_j(n)$ of neuron j and the derivative of its activation function $\varphi'_j(v_j(n))$ is defined as *local gradient*:

$$\delta_j(n) = \frac{\partial \mathscr{E}(n)}{\partial v_j(n)} = \frac{\partial \mathscr{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n)\ \varphi'_j(v_j(n)) \tag{2.20}$$

Therefore, integrating equation 2.20 in equation 2.19 yields:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \tag{2.21}$$

The *weight update rule* of the successive adjustment $(n+1)$ applied to the synaptic weight w_{ji} can be expressed as:

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) \tag{2.22}$$

$$\begin{pmatrix} Weight \\ correction \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} learning \\ rate \\ \eta \end{pmatrix} \times \begin{pmatrix} local \\ gradient \\ \delta_j(n) \end{pmatrix} \times \begin{pmatrix} input\ signal \\ of\ neuron\ j, \\ y_i(n) \end{pmatrix}$$

The local gradient $\delta_j(n)$ depends on whether neuron j is an output node or a hidden node:

- If neuron j belongs to the *output layer*, the error signal $e_j(n)$ can be straightforwardly calculated using equation 2.14, therefore, $\delta_j(n)$ equals to:

$$\delta_j(n) = e_j(n)\ \varphi'_j(v_j(n)) \tag{2.23}$$

- If neuron j belongs to a *hidden layer*, the desired response $d_j(n)$ cannot be directly accessible (since neuron j is a hidden node), therefore $\delta_j(n)$ equals to the product of the associated derivative $\varphi'_j(v_j(n))$ and the weighted sum of the local gradient δ_k, *computed for the neurons k in the next hidden or output layer*:

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum\nolimits_{k\ in\ the\ next\ layer} \delta_k(n)\ w_{kj}(n) \tag{2.24}$$

Figure 2.8 (right) shows graphically the signal flow of equation 2.24

2.10 Gradient descent improvements

2.10.1 Learning rate of neural networks

To find the lowest value of signal error and to converge up to a point where no significant changes in error occur (the *global minimum*), the backpropagation algorithm approximates this trajectory in the weight space, this is referred as the *method of steepest descent*.

The *learning rate* η has a profound influence on this behavior of convergence and figure 2.9 shows two distinctive cases:

- If η is small, the weights change from one iteration to the next will be slow (and the trajectory in weight space will be smoother).
- If η is large, the learning speeds up but at the cost of an unstable weights update. The trajectory in weight space will follow an oscillatory or unstable path.

2.10.2 Gradient descent and its variants

Recalling from Ruder (2016), gradient descent minimizes an objective function $J(\theta)$ by updating its parameters in the opposite direction of the gradient of the objective function $\nabla_\theta J(\theta)$ (with respect to the parameters). The learning rate η determines the size of the steps adopted to reach the convergence towards a local minimum.

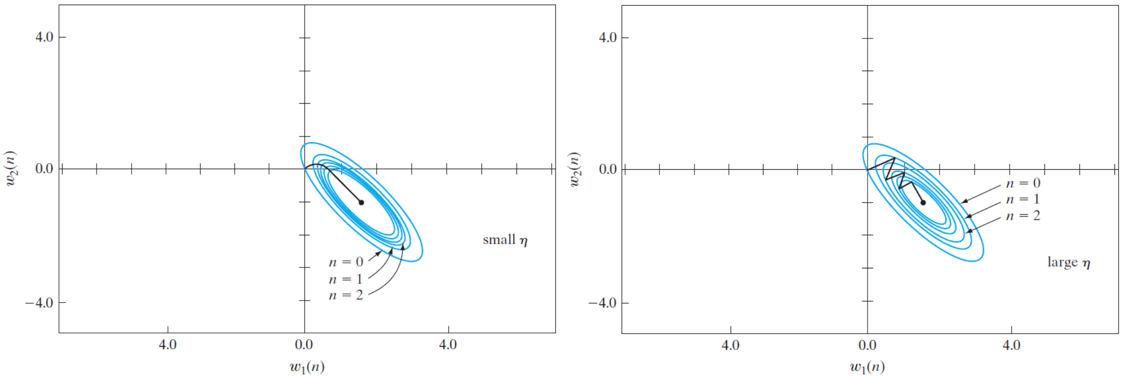


Figure 2.9: Steepest descent using a small and large learning rate, taken from Haykin (2009).

Gradient descent variants differ in how much data is used to compute the gradient of the objective function:

- ***Batch gradient descent***: computes the gradient of the cost function with respect to the parameters θ for the entire training set data. This method is slow and intractable for large datasets, since the gradient is calculated to perform the update in just one take, also not allowing to update the model *online* (i.e. with new examples on the fly).

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta) \tag{2.25}$$

- ***Stochastic gradient descent***: performs a parameter update for each training example $x^{(i)}$ and label $y^{(i)}$, performing one update at a time (online learning). Stochastic gradient descent performs frequent updates with a high variance that cause the objective function to fluctuate heavily, enabling it to converge to a better local minimum (Ruder, 2016). If the learning rate is slowly decreased the stochastic gradient descent shows the same convergence behavior as batch gradient descent, otherwise, due to his oscillatory nature, it could complicate convergence to the exact minimum.

$$\theta = \theta - \eta \nabla_\theta J(\theta; x^{(i)}; y^{(i)}) \tag{2.26}$$

- ***Mini-batch gradient descent***: splits the training dataset into small batches of n training examples and performs an update for every mini-batch, reducing the variance of the parameter updates and leading to a more stable convergence.

$$\theta = \theta - \eta \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)}) \tag{2.27}$$

  Mini-batch gradient descent is the most commonly used implementation of gradient descent, since it seeks to find a balance between the robustness of stochastic gradient descent and the efficiency of batch gradient descent (Brownlee, 2017).

2.10.3 Momentum

Momentum helps to accelerate the gradient descent towards relevant directions and dampens unstable oscillations (Polyak, 1964; Qian, 1999), adding a momentum term γ to the update vector of the past time step v_{t-1} and added over to the current update vector v_t:

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta) \tag{2.28}$$

This momentum term increases for gradients pointing in the same direction and reduces for gradients which change directions, granting faster convergence and reduction of oscillations. Figure 2.10 shows the application of the momentum term. The inclusion of momentum is useful to avoid unstable behavior and increase the learning rate process, granting many benefits:

- If the direction towards convergence proceeds with a steady downhill direction, the momentum tends to *accelerate* this process.
- If the direction towards convergence proceeds with an unstable oscillation, the momentum has a *stabilizing effect* property.
- The momentum term may also have the benefit of preventing the learning process from terminating in a shallow local minimum on the error surface (which prevents the network to reach a better *global minimum*).

2.10.4 Nesterov accelerated gradient

The Nesterov accelerated gradient was introduced by Sutskever *et al.* (2013) and it is inspired by the works of Nesterov about the accelerated gradient method (Nesterov, 1983, 2013). With respect to the standard momentum term γv_{t-1}, the Nesterov accelerated gradient performs a better gradient descent by computing $\theta - \gamma v_{t-1}$, which gives an approximation about where the next parameter θ could be located.

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta - \gamma v_{t-1}) \tag{2.29}$$

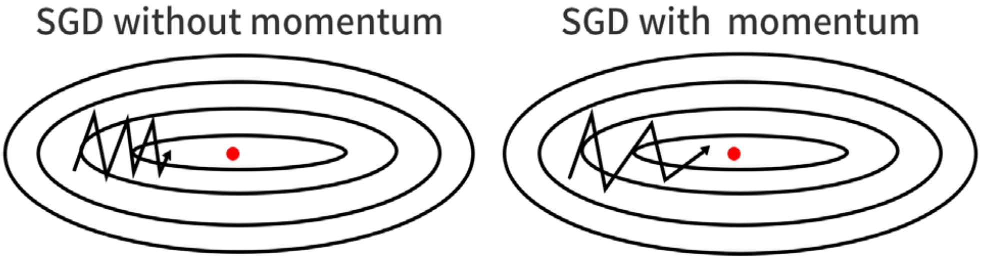


Figure 2.10: Adding a momentum term, taken from Du (2017).

Figure 2.11 depicts the benefits of the Nesterov accelerated gradient over standard momentum (the blue vector): the red vector is the correction applied by the Nesterov accelerated gradient to the brown vector, which the final vector is the green one.

2.10.5 Adagrad

Adagrad (short for ADAptive GRADient, Duchi *et al.* 2011) adapts the learning rate η of each model parameters, which is inversely proportional scaled to the square root of g_t, the sum of all of their past squared values (Goodfellow *et al.*, 2016). When a parameter is associated with frequent features, Adagrad performs smaller updates using a lower learning rate, otherwise performs larger updated using a higher learning rate.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{g_t + \epsilon}} \odot g_t \tag{2.30}$$

The main benefit of Adagrad lies on its adaptive learning rate but suffers from its gradual reduction up to an infinitesimally small value (caused by the accumulation of the squared gradients in the denominator, Ruder 2016).

2.10.6 Adadelta and RMSprop

Adadelta, an extension of Adagrad introduced by Zeiler (2012), aims to resolve the monotonical learning rate decrease of Adagrad. Unlike in the Adagrad algorithm (where all the past squared gradients are stored and accumulated) in Adadelta the sum of gradients is restricted to a fixed size and recursively defined as a decaying average of all past squared gradients (Ruder, 2016). Therefore, the running average at a time step depends only on the previous average and the current gradient. RMS represents the Root Mean Squared error criterion of the gradient g.

$$\theta_{t+1} = \theta_t - \frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t \tag{2.31}$$

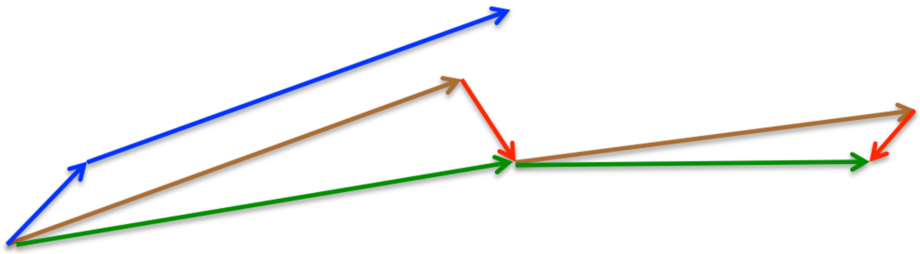


Figure 2.11: Nesterov accelerated gradient, taken from Hinton *et al.* (2014a).

A similar algorithm was also introduced by Hinton *et al.* (2014b) under the name of RMSprop, which they have been developed independently to issue the diminishing learning rate of Adagrad (Ruder, 2016).

2.10.7 Adam

Adam (short for ADAptive Moment estimation, Kingma and Ba 2014) features are inspired from momentum, Adadelta, and RMSprop (Ruder, 2016).

This algorithm stores momentum changes for each parameter separately, combining the storage of the exponentially decaying average of the past square gradients (the first moment v_t, like Adadelta and RMSprop) and the past gradients (the second moment m_t, like the momentum term).

$$\hat{m_t} = \frac{m_t}{1 - \beta_1^t} \tag{2.32}$$

$$\hat{v_t} = \frac{v_t}{1 - \beta_2^t} \tag{2.33}$$

Hence the name of this algorithm, m_t is the estimate of the *mean* of the gradients and v_t is the estimate of the *un-centered variance* of the gradients. The Adam parameter update is similar to the ones implemented in Adadelta and RMSprop:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v_t}} + \epsilon}\hat{m_t} \tag{2.34}$$

2.11 Generalization property and neural networks improvements

Whenever a biological system encounters a new situation it records the related responses, outcomes, and what new situations arose as a result. This ensures that the system keeps all the information in order to be referred back to in the future, detects regularities and eliminate redundancies. In this way, the system reaches a generalization where a less-complex sequence can achieve the same goal (Klahr, 1982; Siegler, 1991).

Once trained over a training set of data, a neural network is said to *generalize* well when the input–output mapping is also correct with inputs coming from the test set, never seen before by the network. **Generalization** is the ability of the neural network to perform well towards previously unobserved inputs (Goodfellow *et al.*, 2016). A good measure of this feature lies on the difference between the error produced from the training set learning and the one from the test set: the test error should be greater than or equal to the expected value of training error. If not, table 2.1 shows different behaviors that can occur:

- The first column shows a poor generalization feature of the network, this is referred as *underfitting*.
  The error of the training set is almost identical to the test set error.

- The center column shows a good generalization, the network produces a correct input–output mapping even when the input of the test set is slightly different from the examples of the training set.
  The error of the training set is slightly lower than the test set error.

- The last column shows the opposite effect: the network learns too many input–output examples, this effect is called *overfitting* and the ability to generalize is impaired.
  The error of the training set is much lower than the test set error.

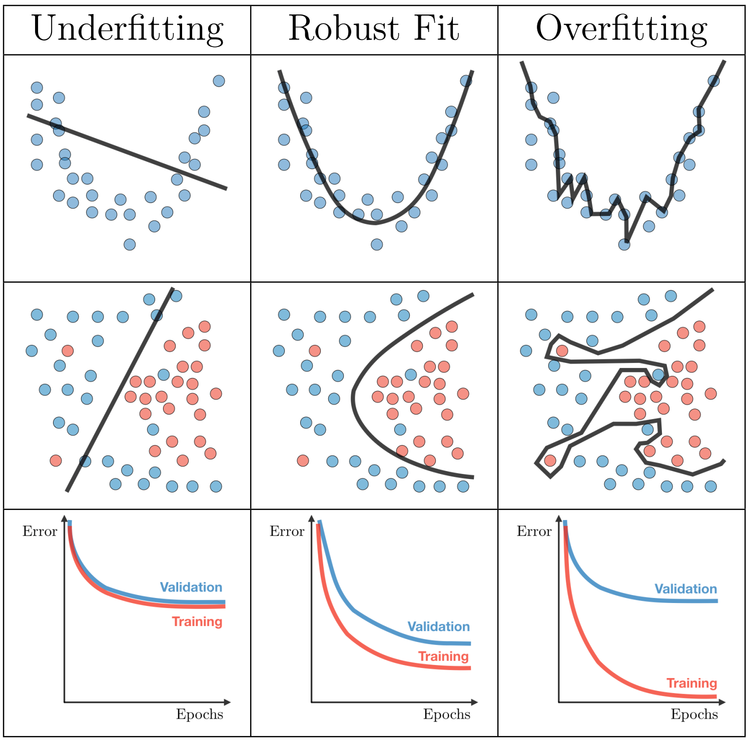


Table 2.1: Generalization examples, adapted from Amidi and Amidi (2018).

2.11.1 Data augmentation

One of the causes of overfitting is the lack of enough number of training data available, hence the model has poor generalization performance. If the training set cannot be augmented with new data, one solution could be reusing the existing data available via random transformation to generate new data. Data augmentation is considered as a *regularization* technique and it is done dynamically during training time over training data Dertat (2017): common image transformations are rotation, shifting, resizing, exposure adjustment, contrast change, etc.

2.11.2 Batch normalization

Normalization of the input layer by adjusting and scaling the activations of the previous layer at each batch: this technique applies a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1. Thanks to this *data whitening* the responses are all in the same range with zero mean, helping the next layers not to learn input data offsets, improving speed, performance, and stability of artificial neural networks (Ioffe and Szegedy, 2015; Culurciello, 2017).

2.11.3 Regularization methods

Also referred as *weight decay*, it's the most common form of regularization which heavily penalizes peaky weight vectors, to prefer diffuse weight vectors (Li *et al.*, 2018a). This regularization technique encourages the network to use all of its inputs, rather than use often some of it.

For every weight w in the network, a regularization strength term is added. During gradient descent parameter update, using the regularization term ultimately means that every weight is decayed linearly towards zero.

- L1 regularization, also referred as lasso regression, adds an absolute value of the magnitude of coefficient $\lambda \sum_{j=1}^{p} \beta_j$ as penalty term to the loss function. Lasso regression shrinks the less important coefficient of the feature to zero, removing some feature altogether, and works well for feature selection in case of a high number of features available. But if λ has a very large value then it will transform coefficient values to zero, leading to underfitting.

- L2 regularization, also referred ad ridge regression, adds a squared magnitude of coefficient $\lambda \sum_{j=1}^{p} \beta_j^2$ as penalty term to the loss function. This technique works very well to avoid over-fitting issue but if λ has a very large value then it will add too much weight, leading to underfitting.

2.11.4 Dropout

Dropout (Srivastava *et al.*, 2014) is the most popular regularization technique and used to prevent overfitting, at each iteration a neuron is temporarily disabled with probability p (by multiplying its output value by zero), as shown in figure 2.12.

The dropped-out neurons are resampled with probability p at every training step, therefore a dropped-out neuron at one step can be active at the next one (Dertat, 2017). The hyperparameter p is the dropout rate and has typically a 0,5 value (corresponding to 50% of the neurons being dropped out). Dropout prevents the network to be too dependent on a small number of neurons, therefore forces every neuron to be able to operate independently.

Dropout can be applied to input or hidden layer nodes during training. It is very computationally cheap but because it is a regularization technique, it reduces the effective capacity of a model (Goodfellow *et al.*, 2016). To offset this effect the size of the model must be increased (using, for example, data augmentation).

A spatial version of dropout performs the same function as standard dropout but drops entire 1D-2D-3D feature maps instead of individual elements. If adjacent frames (1D-2D) or voxels (3D) within feature maps are strongly correlated, typical in early convolution layers, the learning rate decreases. Therefore, spatial dropout helps to promote independence between feature maps.

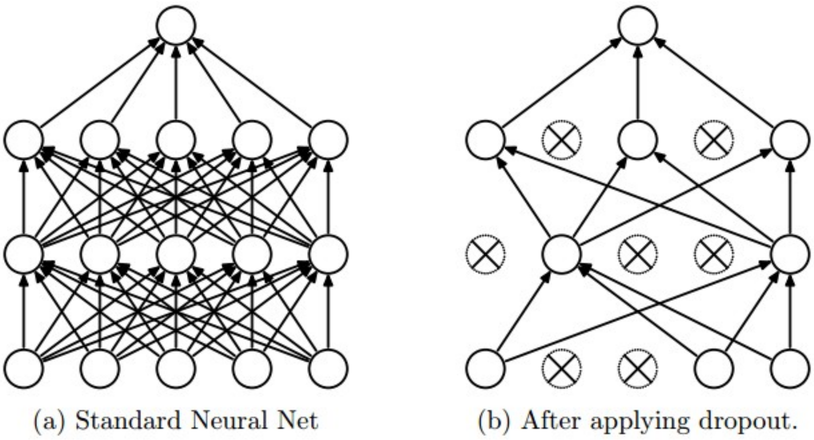


Figure 2.12: Dropout regularization technique, taken from Srivastava *et al.* (2014).

Chapter 3

Convolutional neural networks

Convolutional Neural Networks (LeCun *et al.*, 1989a) represent one of the most successful *Deep Neural Networks architectures* which gave rise of popularity and interest among the Artificial Intelligence field of study. The term "deep" refers to its internal architecture composed of more than two hidden layers (hence, models like the multi-layer perceptron are commonly referred as *shallow neural networks*).

Convolutional neural networks are commonly applied to visual imagery analysis and learn high-level and abstract representations from raw data by stacking hierarchical layers of neurons, relying on a matrix-multiplication called *convolution* (a specialized kind of linear operation) and it is mostly used in image classification, where spatially-coherent input data has a known, grid-like topology (Goodfellow *et al.*, 2016).

Convolutional neural networks are inspired by biological processes and connectivity pattern of the mammalian visual cortex: both systems respond to stimuli only in restricted regions called *receptive fields*, the entire visual field is covered thanks to the overlapping of these receptive fields (see section 4.1).

A convolutional neural network trained for image classification takes as input the image pixels (representing the first layer), the subsequent hierarchical layers capture progressive simple-to-abstract representations of the image, and the output layer serves as a classifier.

Figure 3.1 gives an example of how a convolutional neural network correctly classifies the object in the image as a person: The first layer takes as inputs the raw data consisting of image pixels.

- The first hidden layers detect simple relation-parts like edges, lines, curves, etc.
- The detection is progressively more complex and abstract towards the identification of object parts like eyes, ears, arms, hair, etc.
- The final layers of a convolutional neural network detect the whole-part relationships to correctly classify the image as a person.

Convolutional neural networks work with images which contain a spatial coherence. Elements of the image form relation parts and the translational-invariance property of the neural network keep the detection of these parts even if they are subject to change. *Invariance* means that object properties detected by the convolutional neural network can be maintained and re-detected even in the presence of image rotation, translation, etc.

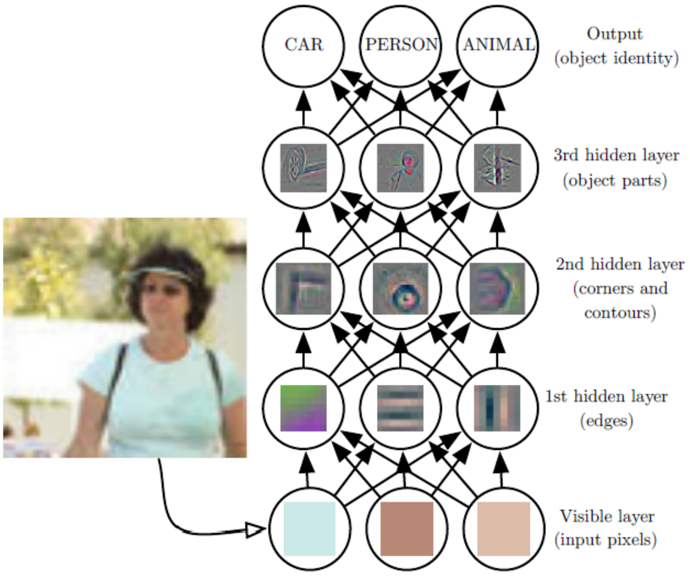


Figure 3.1: A scheme on how a convolutional neural network works, taken from Goodfellow *et al.* (2016).

As listed below, using convolutional neural networks grant several benefits over using multi-layer perceptrons, both in terms of better generalization properties and strong connections/weights reduction (Goodfellow *et al.*, 2016):

- *Sparse interactions*: neural networks use matrix multiplications to process data, these operations are computed differently between traditional and convolutional networks:

  - In traditional neural networks, every output unit interacts with every input unit: the network is *fully-connected.*
  - In convolutional neural networks, the information is processed *locally* since the convolution operation involves generally a kernel of a smaller size than the input matrix.

  I.e. for an image processing task, a multi-layer perceptron, given its fully-connected nature, has a connection and weight for each pixel of the input image. A convolutional neural network, on the other hand, when processes a specific portion of the image, uses only the local weights and connections. Therefore, computing the output requires fewer operations, which both reduces the memory requirements of the model and improves its statistical efficiency.

- *Parameter sharing*: refers to the use of the same parameter for more than one function in a model.

  - In traditional neural networks, each weight is used once to compute the output of a layer.
  - In convolutional neural networks, the weights are stored in a kernel, which will be used throughout every position of the input.

  Thanks to the parameter sharing, the convolution is dramatically more efficient than dense matrix multiplication of multi-layer perceptrons.

3.1 Convolutional neural networks architecture

A convolutional neural network model can be thought of as a combination of hierarchically structured layers, grouped into feature extraction and classification layers (Dertat, 2017).

The ***feature extraction layers*** detect semantic parts of an image (i.e. in figure 3.2, given an image of a car, they detect features like wheels, lights, windshield, etc.), the first layers detect edges, the next layers combine them to detect shapes, up to the following layers where they merge this information to infer that the image represents a car. The network doesn't have prior knowledge about what is a car wheel or how it's made of, but it learns to detect that as a feature by processing a usually extensive training set. This process is made possible by the integration of layers which compute different functions: *convolutional layers* (section 3.1.1), *activation layers* (section 3.1.2) and *pooling layers* (section 3.1.3). The input layer represents the image pixels, and the intermediate layers use local connections and shared weights. Local and shared connections make neurons processing, in the same way, on different portions of the image. The passage of the input data through layers of convolution and pooling creates a *bi-pyramidal effect* (LeCun *et al.*, 1995): at each layer of the neural network, comparing to the previous one, the number of feature maps increases while the spatial resolution of the layer decreases (see section 3.2). This concept of alternating the convolution and pooling layers was inspired by the work of Hubel and Wiesel (1962, 1977) on locally sensitive and orientation-selective neurons of the cat's and macaque's visual cortex (section 4.1).

The ***classification layers*** are typically used in classical convolutional neural networks architectures and act like a multi-layer perceptron (they are also referred as *fully-connected layers*) and the final layer assigns a probability score by using a *softmax* function, in order to correctly classify the images (section 3.1.4).

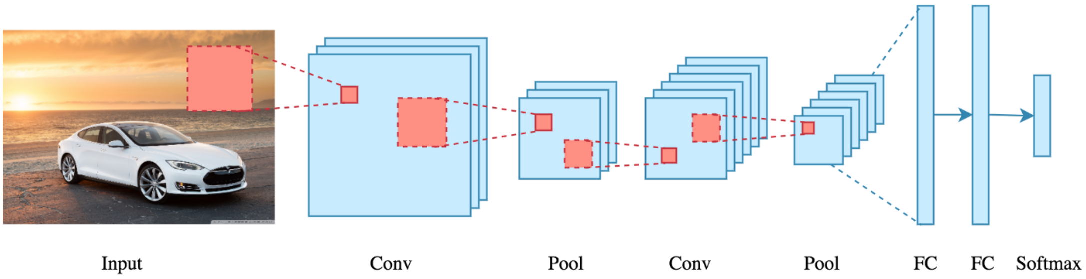


Figure 3.2: Architecture of a convolutional neural network, taken from Dertat (2017).

3.1.1 Convolutional layer

Convolution is a mathematical operation between two multidimensional matrices, defined as *tensors*: an *input matrix* (the input data, the blue matrix of figure 3.3) and a *kernel* (the parameters adapted by the learning algorithm, the green matrix of figure 3.3). The output of this operation is referred as a *feature map* (the red element blocks of figure 3.3).

The convolution is an element-wise matrix multiplication (also known as Hadamard product) between the input and the kernel, the result is added and stored in the feature map: starting from the first element the filter slides over the input matrix. The rate of this slide is referred as *stride* and the area where the convolution takes place is called *receptive field.*

- Starting from the left image of figure 3.3, the blue matrix is the input and the green one is the kernel filter.
- The other images of figure 3.3 show the convolution done on the first receptive field of the input matrix, where the output of this operation is stored on the feature map colored in red.
- The left image of figure 3.4 show how this operation is done in a tensor: it has a size of W×H×D (Width×Height×Depth). Since the input image is usually in color, the input size of figure 3.4 is 32×32×3, whereas the depth 3 corresponds to the RGB color channels.
- The depth of the feature map is determined by how many filters are used for the convolution operation: in the right image of figure 3.4 the feature map size is 32×32×10 since 10 different filters to detect different features in the image were used.

The convolution operation for each filter is performed independently.

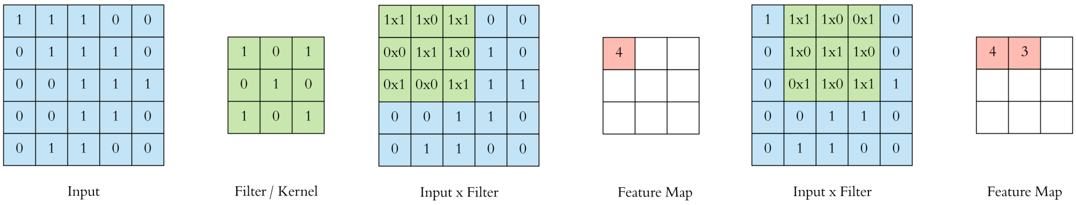


Figure 3.3: The convolution operation in a matrix, taken from Dertat (2017).

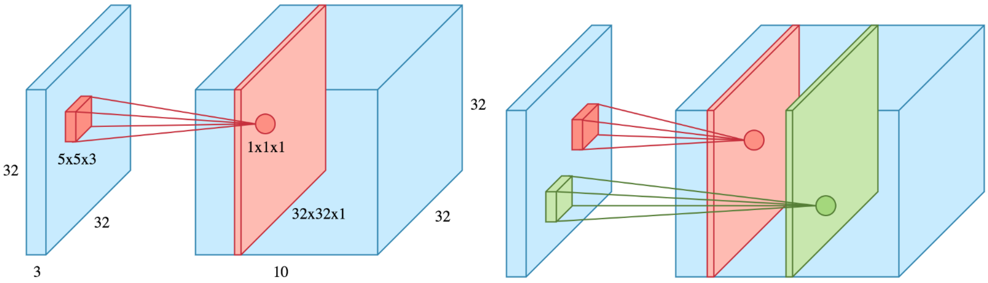


Figure 3.4: The convolution operation in a tensor, taken from Dertat (2017).

Stride specifies how much the filter moves along each step over the input, in the example reported on figure 3.3 the stride is 1 but its value can be adjusted accordingly. Bigger strides help to reduce overlapping between the receptive fields, but it reduces the feature map size with the risks of skipping potential feature locations and detection.

To overcome a reduction in size of the output matrix (the feature map) the solution is ***padding*** over the input with zeros, this operation is called *zero padding* and guarantees the exact dimensionality between the input and the feature map. Without this feature, the representation width shrinks by one pixel less than the kernel width at each layer, zero padding the input allows us to control the kernel width and the size of the output independently (Goodfellow *et al.*, 2016).

3.1.2 Activation layer

After the convolutional layer, the feature map is run through a nonlinear activation function, this stage is sometimes called the detector stage (Goodfellow *et al.*, 2016). Recalling from section 2.3 the rectified linear unit is the activation function of choice for a convolutional neural network (Jarrett *et al.*, 2009; Glorot *et al.*, 2011), giving a linear output for all positive values or 0 if otherwise.

Using the relu function leads to *sparse activations*: parts of the neurons are not activated and this can confer great robustness. The relu activation function is used primarily to overcome the *vanishing gradient problem*: it occurs in the backpropagation phase in deep neural networks using the sigmoid as the activation function. During the application of the derivation chain rule, the use of the sigmoid function leads to multiplying many terms which reduce a lot of gradient values: since the sigmoid derivative is typically < 1, this multiplication will result progressively towards a minimization tendency to zero, hence the vanishing gradient.

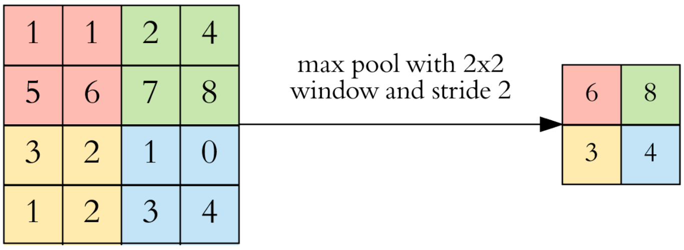
3.1.3 Pooling layer

Also referred as *subsampling layer*, the pooling function further modify the feature map, replacing the output net with a summary statistic of the nearby outputs (Goodfellow *et al.*, 2016). The *max pooling* (Zhou and Chellappa, 1988) operation reports the maximum output within a rectangular neighborhood, thus reducing the dimensionality and the number of parameters (which both shortens the training time and prevents overfitting). Pooling layers downsample each feature map independently, reducing the height and width, keeping the depth intact (Dertat, 2017).

An example of this subsampling is reported on figure 3.5 where an input 4×4 matrix, with a 2×2 max pooling and stride 2, is reduced to a 2×2 output matrix: only the highest value of the correspondent colored cells is stored on the output matrix.

Pooling helps to maintain the representation invariant to small input translations: ***invariance*** to translation means that if input translates by a small amount, the pooled output doesn't change. Invariance to local translation is useful when it is more important to know whether a feature is actually present, rather than know exactly where it is (Goodfellow *et al.*, 2016).

Spatial invariance represents a critical component of the theory of ecological perception (Gibson, 2014). The highly variable dynamic flux of visual stimulation is produced by changes in the world or by changes in viewing position. Characteristics which remains constant are the invariants of the optic array: picked up by direct perception and specify affordances to the perceiving agent. What is invariant in an optic array is the relationship between two of its varying components: they may vary but the relationship does not (see section 4.1).



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 3.5: The max-pooling operation, taken from Dertat (2017).

3.1.4 Fully-connected and output layer

Passing through the aforementioned convolutional and pooling layers, the input data decreased in spatial resolution but increases in features detected. Towards the end of the convolutional neural network architecture, thanks to this bi-pyramidal effect, the layers can be treated as a multi-layer perceptron for the classification task.

The output of the final pooling layer is flattened to a vector which becomes the input to the fully-connected layers: a passage from a three-dimensional matrix to a one-dimensional vector (Dertat, 2017).

The final layer of a classic or prototypical convolutional neural network architecture, used as the output of a classifier, relies on the exponential normalization function: the *softmax function* (Goodfellow *et al.* 2016; Liu *et al.* 2016a; Rawat and Wang 2017, see section 2.3).

3.1.5 Convolutional neural network hyperparameters

The hyperparameters (which are those values which are set before the learning process begins) are essentially four: number of filters, filter size, stride and the implementation of padding.

Figure 3.6 highlights the stacked layers of a convolutional neural network and its general architecture for image classification task. The input is represented by 227×227×3 images and the number of feature maps increases moving towards the softmax output (represented by 1000 classes). Layers 1 to 5 include a combination of convolutional, activation and pooling layers, with variable stride (stride 4 in the first convolutional layer and then stride 1 for the others) and kernel filters (with dimensions ranging from 11×11 to 3×3). Layers 6 and 7 are fully-connected.

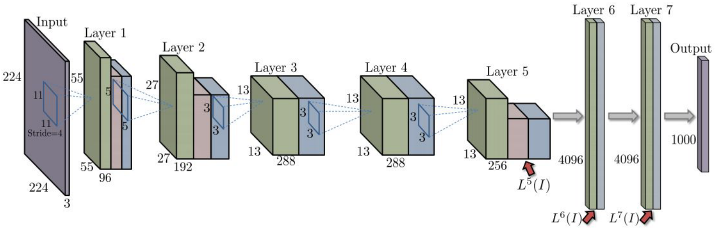


Figure 3.6: Layers of a convolutional neural network: the numbers indicate the image size and related features, taken from Babenko *et al.* (2014).

3.2 Visualizing convolutional neural network features

Dertat (2017) shows examples of what a convolutional neural network "sees": convolution feature maps (left to right) of the first layers retain most of the image information, acting usually as edge detectors and lower-level feature detection. Then they gradually show more abstraction and high-level representation of the image.

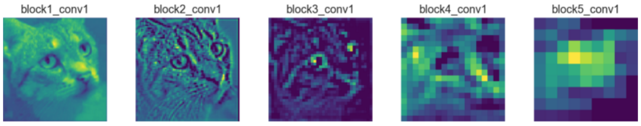


Figure 3.7: Feature maps, adapted from Dertat (2017).

First convolution kernel filters (left to right) mostly detect colors, edges, and simple shapes. In the deeper sections of the network, the filters build on top of each other and learn to encode complex pattern.

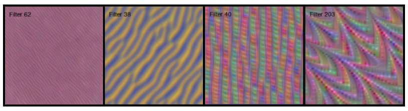


Figure 3.8: Kernel filters, adapted from Dertat (2017).

The convolutional neural network property of decomposing the visual input space (as a hierarchical and modular network of convolution filters) grants the probabilistic mapping, occurring at the final layers of the network, between certain combinations of the aforementioned filters and a set of arbitrary labels (Chollet, 2016). Figure 3.9 shows that, given a particular output class, every example shows an image that maximally represents its specific class: an object appears several times in the image because of the higher-class probability.

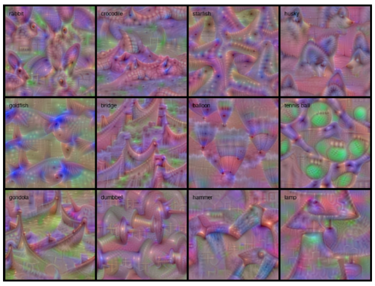


Figure 3.9: Output classes, adapted from Dertat (2017).

3.3 History and evolution of neural networks

The scope of this section is to illustrate an overview of how neural networks have evolved drastically over the course of time. This section is focused on explaining the major steps towards the current state-of-the-art deep learning architectures.

For the sake of keeping arguments as much as possible in line with this thesis many architectures will be omitted, for a complete survey and review see Schmidhuber (2015); Wang and Raj (2017).

3.3.1 1943: McCulloch & Pitts artificial neuron

McCulloch and Pitts (1943), in order a neurophysiologist and a mathematician, introduced and modeled in 1943 the first artificial neuron, a precursor of future neural networks.

Their model (see section 2.2.2) is built as electrical circuits and it is a linear step function upon weighted data, recognizing two different categories of inputs. The function completely prevents excitation of the neuron if the input is inhibitory. The weights are fixed and assigned with manual calculation (Wang and Raj, 2017).

3.3.2 1958-1960: Perceptron and ADALINE

Rosenblatt (1958) introduced in the 1950s the perceptron (see section 2.5): the first model where the weights are learned and defines the category outputs, given examples of inputs from each category (Goodfellow *et al.*, 2016).

Widrow and Hoff (1960) introduced ADALINE: adaptive linear element. Similar to the perceptron, this architecture uses stochastic gradient descent as a training algorithm for weights adaptation.

3.3.3 1969: The AI winter

Minsky and Papert (1969) highlighted the limitations of linear models: the perceptron and ADALINE, since they're limited to represent linear boundaries, cannot learn correctly non-linear functions like the XOR function. These flaws caused a backlash against interests on neural networks up until the 1980s.

3.3.4 1980: Neocognitron

Kunihiko Fukushima (1975, 1980) introduced the Cognitron and Neocogniton, a biologically-inspired neural network which is generally seen as the model that inspired convolutional neural networks (Wang and Raj, 2017). The model architecture of the Neocognitron (figure 3.10) takes inspiration from the works of Hubel and Wiesel (1959, 1962) on the structure of the mammalian visual system explained in section 4.1 (Rawat and Wang, 2017):

- Neurons in different stages of the primary visual cortex respond strongly to specific stimulus patterns while ignoring others. The visual cortex consists of simple cells having local receptive fields and complex cells invariant to shifted or distorted inputs, these cells are arranged hierarchically.
- Repeated stacks of two kinds of layers compose this neural network.
  - The Simple-cell layers serve as a feature extractor, where each cell is responsive to a particular feature presented in its receptive field.
  - The Complex-cell layers serve as structured connections to organize the extracted features.

  Local features (i.e. edges in particular orientations) are extracted in lower layers while global features are extracted in higher layers.

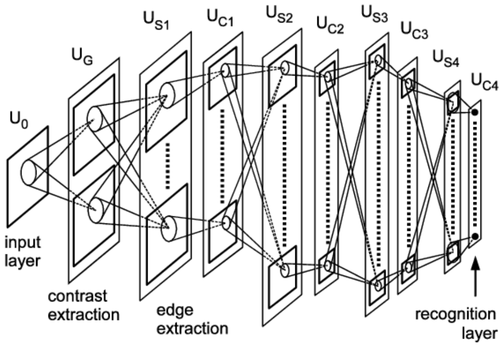


Figure 3.10: Neocognitron architecture, taken from Fukushima (2003).

3.3.5 1989-1990s: Connectionism and the backpropagation

This renewed interest upon neural networks emerged in great part via a movement called connectionism or parallel distributed processing (Rumelhart *et al.*, 1985; McClelland *et al.*, 1988). The central idea in connectionism is that a large number of simple computational units can achieve intelligent behavior when networked together, unlike using symbolic models where were difficult to explain in terms of how the brain could actually implement them using neurons (Goodfellow *et al.*, 2016).

The use of backpropagation (section 2.9, Rumelhart *et al.* 1986; LeCun 1987) successfully applied upon the first convolutional neural networks (LeCun *et al.*, 1989b, 1990, 1998) shows with promising results, using the MNIST handwritten digits dataset which over time has become one of the benchmark datasets and it is still used today.

3.3.6 1998: LeNet5

The first convolutional neural network, LeNet5, was implemented by LeCun *et al.* (1998), after previous successful iteration since 1988. Its architecture is inspired by the insights of the Neocognitron (Fukushima, 1980) and its core features are still used for today's state-of-the-art convolutional neural networks.

Instead of using each image pixel as a separate input neuron (like a multi-layer perceptron architecture) the image features are distributed over the entire picture using the convolution operation, in order to extract similar feature at different portions of the image and to reduce parameters and computation expense (Culurciello, 2017). The convolution, pooling and non-linear activation of LeNet5 (figure 3.11) extract and sub-sample the spatial features of the input image, the final layers classify the image as the same method as used in a multi-layer perceptron.

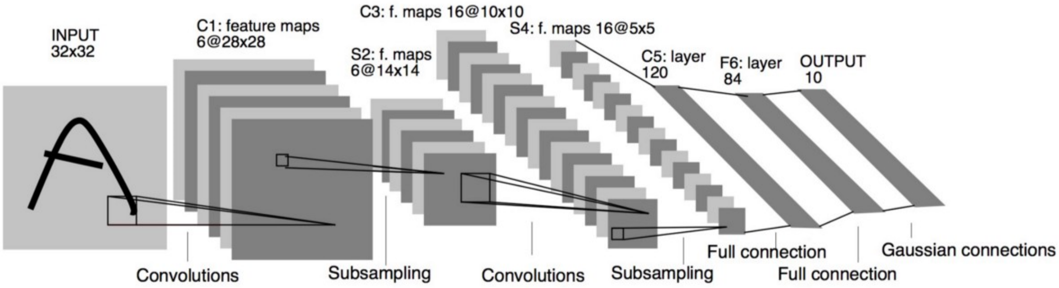


Figure 3.11: LeNet5 architecture, taken from LeCun *et al.* (1998).

3.3.7 2010s: Deep learning renaissance

After a period of latent interest in neural networks between the mid-1990s and 2000s, the current wave of neural networks research is popularized under the use of the term deep learning, which key aspects are the follow (Goodfellow *et al.*, 2016):

- Increased amount of data: the increasing digitalization of society helped, along with increasingly networked computers, the creation of bigger and detailed datasets for machine learning applications.

- Increased neural networks model size: due to the availability of faster hardware (i.e. CPUs, GPUs, faster network connectivity, etc.), deep learning models have grown in size over time, doubling in size roughly every 2.4 years (Goodfellow *et al.*, 2016).

3.3.8 2012: AlexNet

The breakthrough of convolutional neural networks came from the work of Krizhevsky *et al.* (2012), their paper is regarded as one of the most influential publication in the deep learning research field.

Their architecture, called AlexNet, won the 2012 ILSVRC competition (ImageNet Large Scale Visual Recognition Challenge) of visual tasks like classification, localization, detection, etc. This competition evaluates the top-5 test-error rate: an error at which, given an image, the model doesn't output the correct label with its top 5 predictions (Deshpande, 2016).

The ImageNet dataset contains more than 15 million labelled images over 22 thousand categories (Deng *et al.*, 2009). AlexNet achieved a top-5 test-error rate of 15.4% and largely outperformed the best other competitor, where it achieved a 26.2% error.

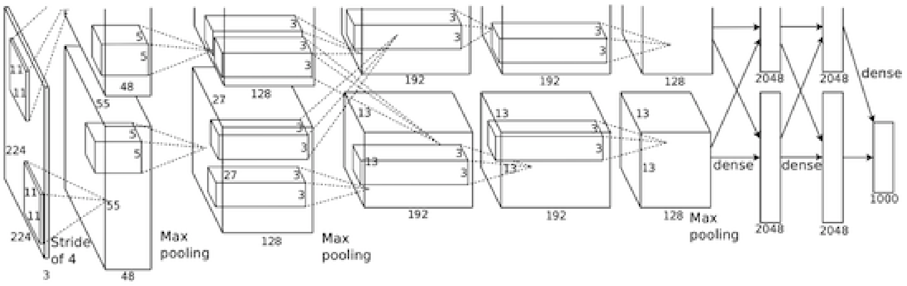


Figure 3.12: AlexNet architecture, taken from Krizhevsky *et al.* (2012).

This outstanding performance was made possible by implementing techniques like:

- ReLU used as nonlinear activation function (see section 2.3).
- Dropout used to avoid overfitting of the training data (see section 2.11.4).
- Implementation of data-augmentation like image translations, horizontal reflections, patch extraction (see section 2.11.1).
- Overlap of max-pooling as subsampling layer (instead of the average-pooling of LeNet5, see section 3.1.3).
- Two GPU (Graphics Processing Unit) cards used for the training process.

The architecture of AlexNet is shown in figure 3.12 and consists of 5 convolutional, max-pooling and dropout layers, with 3 final fully-connected layers (the last one is used for classification upon 1000 image categories). The success of AlexNet started the deep learning era, thanks to the contribution of larger datasets and the extensive use of GPUs to accelerate training time for faster performance.

3.3.9 2013-2014: VGG and Network in Network (NiN)

VGG networks (developed from the University of Oxford by Simonyan and Zisserman 2014) is one of the convolutional neural networks where the dimensionality of the convolution filters is reduced, inverting the trend of earlier networks where larger convolutions were used. Instead of using large filters (like the 9×9 or 11×11 of AlexNet), VGG adopted the smaller 3×3 filter in each convolutional layer (Culurciello, 2017). The use of smaller filters and multiple sequential convolutions grants the emulation of larger receptive fields (i.e. 5×5 or 7×7) and this insight is also used in later architectures like Inception and ResNet but at the expense of an impressive number of parameters and computational power.

The Network in Network (NiN) architecture (Lin *et al.*, 2013) uses 1×1 convolution layers with an immediate follow-up multi-layer perceptron layer after each

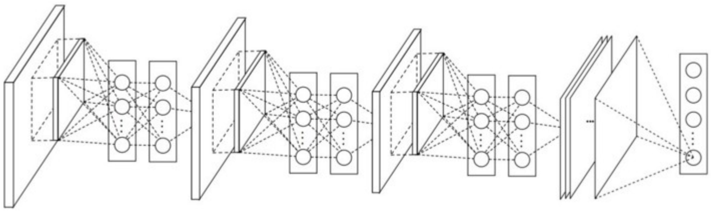


Figure 3.13: Network in Network architecture, taken from Lin *et al.* (2013).

convolution, in order to produce a better feature combination to pass on the latter layers (as shown in figure 3.13). This peculiar architecture prevents the use of a large stack of layers and parameters like VGG and the insight of using 1 × 1 convolution is used also on other architectures like ResNet, Inception and their derivatives (Culurciello, 2017).

3.3.10 2014-2017: GoogleNet and the Inception module

GoogleNet, a deep convolutional neural network architecture developed by Szegedy *et al.* (2015), won the 2014 ILSRVC image contest with a top-5 error rate of 6.7%. Its architecture, shown in figure 3.14 consists of the sequential concatenation of different parallel operations performed within the *inception modules*.

The inception module (figure 3.15, left) is a parallel combination of different convolutional filters, which are stacked in a similar way used on NiN networks: a 1 × 1 convolutional layer before a 3 × 3 or 5 × 5 convolutional layers in order to reduce the number of features to provide for the filter concatenation layer.

This particular layer structure can be referred as a *bottleneck layer*: reducing the number of features and computational operations at each layer, the convolutional neural network can keep a low inference time without losing generalization properties.

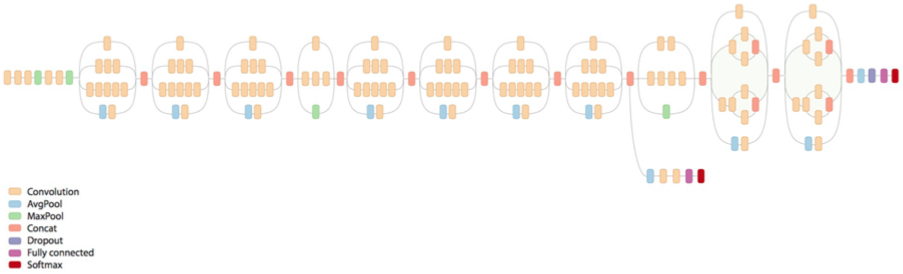


Figure 3.14: GoogleNet architecture, adapted from Deshpande (2016).

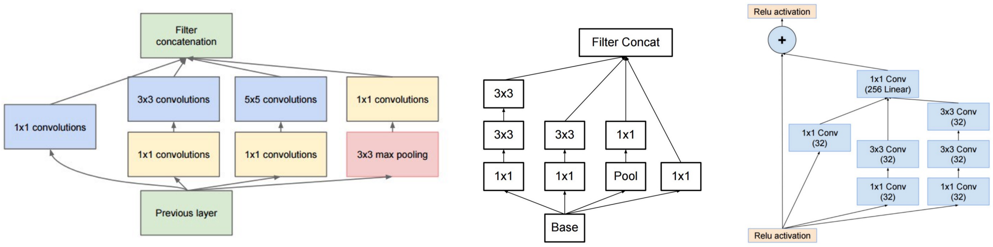


Figure 3.15: Inception V1 (left), Inception V3 (middle) and Inception V4 (right) module, taken from Szegedy *et al.* (2015, 2016, 2017).

The correlated and redundant input features can be removed by combining them with 1 × 1 convolution and the expanded again into meaningful combinations for the successive layers (Culurciello, 2017). Inception V2 (Ioffe and Szegedy, 2015), introduces the batch normalization (see section 2.11.2) and Inception V3 (Szegedy *et al.* 2016; figure 3.15, middle), decompose 5 × 5 and 7 × 7 filters with multiple 3 × 3 filters. Inception V4 (Szegedy *et al.* 2017; figure 3.15, right), represent a combination between the Inception module with the ResNet module.

3.3.11 2015: ResNet and the residual module

ResNet architecture (He *et al.*, 2016) is a very deep convolutional neural network, a 152-layers deep networks that won the ILSVRC 2015 contest with an astounding 3.6% error rate (whereas human performance sets in a 5% range).

This architecture relies on the *residual modules* (figure 3.16, left): an input x passes upon convolutional-relu-convolutional layers, the output $F(x)$ generated is then added with the original input x called *identity layer*. The final outcome $F(x) + x$ represents the input for the successive residual module. This layer-bypass was also an intuition from Sermanet and LeCun (2011) and this process can be seen as a small classifier or a Network in Network, with hundreds up until thousand-layers deep architecture. This very deep architecture started to use also a bottleneck layer similar to Inception: this module uses a 1 × 1 convolutional layer to reduce the number of features per-layer, an intermediate 3 × 3 convolutional layer and a final 1 × 1 convolutional layer to a large number of features (as seen in figure 3.16, right). ResNet can also be seen as multiple combinations of both serial and parallel modules (Veit *et al.*, 2016).

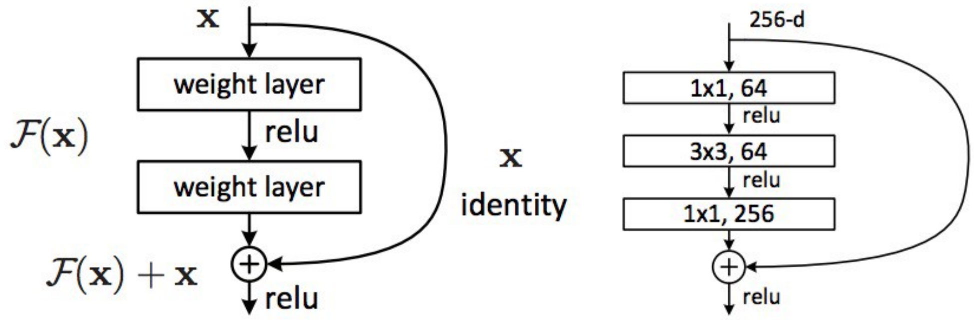


Figure 3.16: ResNet residual module, taken from He *et al.* (2016).

Chapter 4

Integration between neuroscience and artificial neural networks

Rapid progress in related fields of neuroscience and artificial intelligence has been made in recent years, however, the interaction between them has become less commonplace since they both have grown in complexity, consolidating disciplinary boundaries (Hassabis *et al.*, 2017). Despite the efforts, the broad fields of neuroscience and deep learning have evolved greatly in mostly not overlapping paths, also because of the several technical obstacles like the limited capacity of experimental tools used to probe visual information processing, or the limited computational power available for simulations (Medathati *et al.*, 2016).

With the recent wave of new experimental and analysis techniques, the gap between the two fields of research is reducing (Victor, 2005). Many benefits could arise: neuroscience provides rich sources of inspiration for new types of neural networks architectures, and also provide validation for the currently existing techniques (Hassabis *et al.*, 2017). Much of the current computational understandings of biological vision is based on the theoretical framework of Marr *et al.* (1982). Complex systems like brains or computers must be studied and understood at three levels of description: a task carried on from observable behavior, an algorithm used to solve the task and its implementation. Examples of opinions about the importance of neural networks in neuroscience can be found in Hinton (2011); Paninski and Cunningham (2018); Vogt (2018); Vu *et al.* (2018). Glaser *et al.* (2019) pointed out four roles of supervised machine learning to be used in neuroscience: neural networks can create solutions to engineering problems, identify predictive variables, set benchmarks for simple brain models, and serve as a brain model. Barrett *et al.* (2019) reviewed how concepts developed by computational neuroscientists can be useful for analyzing and understanding representations in biological neural networks, since both research fields show empirical similarities (Khaligh-Razavi and Kriegeskorte, 2014; Yamins *et al.*, 2014; Güçlü and van Gerven, 2015; Kriegeskorte, 2015; Pospisil *et al.*, 2016; Yamins and DiCarlo, 2016; Geirhos *et al.*, 2018).

To better understand neural computations powerful analysis tools are required (Cunningham and Byron, 2014), in order to interpret the generated complex data coming from these modern neuroimaging techniques. Researchers can take advantage of deep neural networks in order to develop computational theories for cognitive science and neuroscience: for example, the development of visual theories addressing core-vision problems, which can be tested on realistic images without having to restrict themselves to the study on simplified stimuli (Yuille and Liu, 2018).

The recording of the simultaneous activity of hundreds of neurons is made possible by using electrophysiological technologies (Jun *et al.*, 2017) and imaging tools (Ahrens *et al.*, 2012), and targeted perturbations of neural activities were enabled by optogenetic techniques (Packer *et al.*, 2015; Lerman *et al.*, 2018).

Techniques to identify selective neuronal populations and dissecting their circuitry at synaptic level is now possible by combining functional and structural imaging, in order to understand visual circuits both at retinal and cortical levels (Helmstaedter *et al.*, 2013; Bock *et al.*, 2011). This description of connectivity patterns between different cortical areas, done in a quantitative fashion (Markov *et al.*, 2013), allows neuroscientists to obtain large-scale models of visual networks, both in temporal and spatial manners (Potjans and Diesmann, 2012; Kim *et al.*, 2014; Chaudhuri *et al.*, 2015).

Recent neuroscience tools give access to the activity of large populations of cells, and the same cell across large spans of time (Ohki *et al.*, 2005; Barretto *et al.*, 2009; Margolis *et al.*, 2012). Connectivity between identified cells can be measured, stimulated and silenced directly with high precision, thanks to optogenetic tools (Bernstein and Boyden, 2011; Deisseroth, 2011).

To predict primates' neural ventral stream responses about the estimation of 3D-structure of objects and parts (Biederman, 1987; Yamane *et al.*, 2008), deep neural networks were applied along with electro-physiology measurements (Yamins *et al.*, 2014). Deep neural networks can improve these estimates (McCann *et al.*, 2017) also by including techniques like improved image denoising (Burger *et al.*, 2012; Xie *et al.*, 2012), and resolution (Dong *et al.*, 2015), often replacing the entire image processing pipeline (Golkov *et al.*, 2016; Vito *et al.*, 2005).

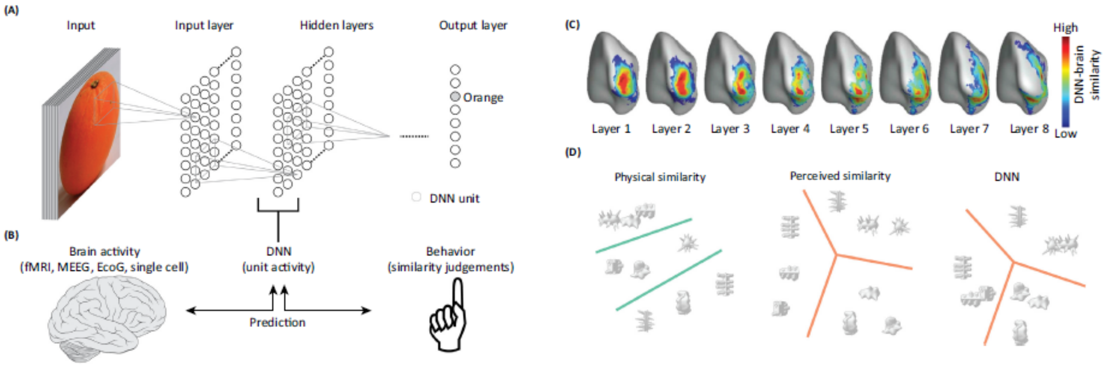


Figure 4.1: Examples of relations between neural networks and cognitive science, taken from Cichy and Kaiser (2019).

Figure 4.1 (a - b) show an example of how a convolutional neural network can be used to categorize objects and how researchers in cognitive science use them to predict brain activity and behavior (Cichy and Kaiser, 2019), outperforming any other machine learning methods for neural activity in primate sensory cortices (Khaligh-Razavi and Kriegeskorte, 2014; Yamins *et al.*, 2014; Güçlü and van Gerven, 2015; Cichy *et al.*, 2016; Eickenberg *et al.*, 2017; Horikawa and Kamitani, 2017a; Cadena *et al.*, 2019).

Figure 4.1 (c) shows an example of the relation between deep neural networks (of eight layers trained for object categorization) and brain activity (using fMRI) (Cichy *et al.*, 2016): the hierarchical relationship shows that earlier layers of the neural network predict low-level visual brain regions and the later layers predict the higher ones.

Figure 4.1 (d) shows an example of how a deep neural network trained for object categorization predict human behavior about perceptual similarity judgments of shapes and representations (Kubilius *et al.*, 2016; Peterson *et al.*, 2017), predicting the perceived similarity (judgment) rather than the physical similarity of the visual stimuli. Kubilius *et al.* (2016) demonstrate that the output layers of deep neural networks develop representations closely related to the human perceptual shape judgments, even though the neural network were never explicitly trained for shape processing.

4.1 The human vision system and deep neural networks

Visual processing in humans takes place by a multilevel aggregation of information, which is processed forward and backward across cortical brain regions (Bullier, 2001; Kourtzi and Connor, 2011; Kravitz *et al.*, 2011; DiCarlo *et al.*, 2012). Figure 4.2 shows the robustness of the human visual system: since it is capable of object recognition across wide variations in pose, light, and occlusion.

The ambiguities that arise in projecting 3D images on the retina representations are also dealt effortlessly. Any given image on the retina can correspond to infinitely many possible objects in the world (Cox and Dean, 2014). The neural activity, occurring during the first 100ms after a stimulus change, unfolds as a cascade along a series of anatomically distinguishable areas connected to each other (Felleman and Van, 1991; Malach *et al.*, 2002; Carandini *et al.*, 2005; DiCarlo *et al.*, 2012; Sharpee *et al.*, 2013; Yamins and DiCarlo, 2016).

Neuroscience and cognitive science were influential in the development of convolutional neural networks, as they take insights and inspiration from the canonical properties of the ventral visual stream, as an ensemble of deep cortical hierarchies (Kruger *et al.*, 2012; Poggio and Serre, 2013; Cox and Dean, 2014; Wang and Raj, 2017; Barrett *et al.*, 2019).

The visual cortex contains over 30 visual areas in the occipital, temporal and parietal lobes, including the primary and extrastriate visual cortex. These visual areas process different aspects of visual information and they form two major pathways. This hierarchical organization and division into parallel streams are supported by a large body of anatomical and physiological evidences (Mishkin and Ungerleider, 1982; Goodale and Milner, 1992; Ungerleider and Haxby, 1994; Van Essen, 2003; Markov *et al.*, 2013).

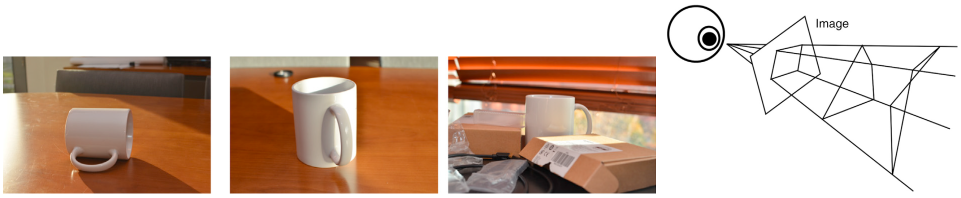


Figure 4.2: The robustness of the human visual system, taken from Cox and Dean (2014).

The ***dorsal pathway*** extends from V1 to the parietal cortex, through motion areas MT and MST, is referred as the "where" pathway and processes information regarding location, motion perception and analysis of spatial structures of the visual scene. The ***ventral pathway*** extends from V1 to the temporal cortex, through area V4 and reaching area IT, is referred as the "what" pathway and processes information regarding the form perception and identity of visual objects, including object and face recognition.

Figure 4.3 (a) shows a cartoon anatomical view of the information flow of both dorsal and ventral pathways, and the diagram of figure 4.3 (b) represents the parallel information flow of these pathways. Starting from the retina, the light-sensitive tissue in the back of the eye where their neurons perform a preprocessing of the image without alteration (Enroth-Cugell and Robson, 1984; Goodfellow *et al.*, 2016), the information passes through the optic nerve and the lateral geniculate nucleus (LGN), in order to be fed into visual cortex V1 thanks to parallel pathways (Ungerleider, 1982; Ungerleider and Haxby, 1994; Milner and Goodale, 2008): the *magnocellular* dorsal pathway conveys coarse and achromatic luminance-based spatial inputs with strong temporal sensitivity, with small cells and receptive size. The information is transmitted to higher cortical areas involved in motion and space processing. The *parvocellular* ventral pathway conveys slower inputs with high spatial resolution but low temporal sensitivity, with large cells and receptive size, in order to flow into cortical areas involved in form and color processing (Livingstone and Hubel, 1984, 1988; Zeki and Shipp, 1988; Kaplan, 2004).

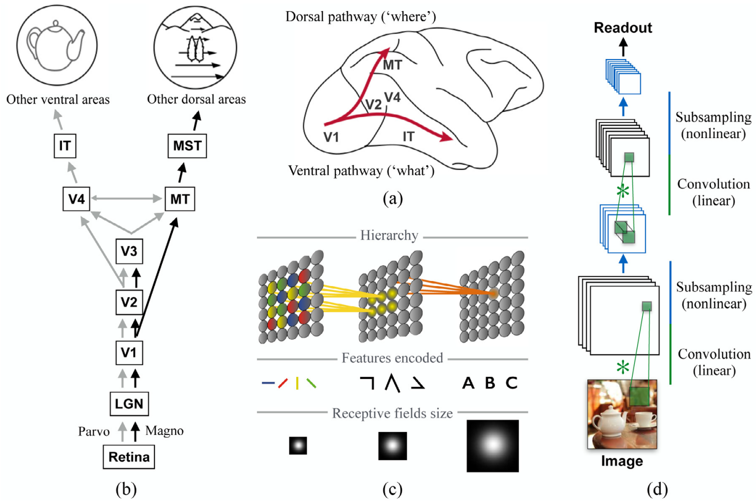


Figure 4.3: An overview of hierarchical feed-forward processing, adapted from Cox and Dean (2014).

Figure 4.4 shows the hierarchical ventral pathway of visual cortex: starting from retinas and LGN, visual information passes through V1 which conveys feed-forward connections to cortex area V2, V4 and IT. The image presented to the retina is kept topographically intact in the next processing steps on the cortical surface (Wandell *et al.*, 2007). V1, the primary visual cortex, performs the edge detection task: an edge is an area with the strongest local contrast in the visual signals (Wang and Raj, 2017). The most famous receptive field analysis is carried by the findings of neurophysiologists David Hubel and Torsten Wiesel about the mammalian vision system (Hubel and Wiesel, 1959, 1962, 1968) reveal how the cat's cortical neurons responded strongly to very specific patterns of light (i.e. precisely oriented bars): these are referred as *simple cells.* They also found *complex cells*, which are more invariant to small shifts in the position of the feature than simple cells.

Layers of convolutional neural networks are designed to capture properties of V1 like the spatial mapping (a two-dimensional structure similar to the retina's structure image), the inclusion of aforementioned simple cells (the spatially localized receptive field property), and complex cells (pooling units, Goodfellow *et al.* 2016). From these insights and from seminal works of Hubel and Wiesel (1959, 1962, 1968, 1977), several biologically-inspired neural networks were created (Fukushima, 1980; LeCun *et al.*, 1995; Riesenhuber and Poggio, 1999; Serre *et al.*, 2007a; Bengio *et al.*, 2009; Pinto *et al.*, 2009).

To analyze the single neuron activity, receptive field analysis represents a canonical method in neuroscience (Victor, 2005): referring to the region of the stimulus space which causes neural firing responses (Barrett *et al.*, 2019). Canonical understandings of visual processing show that neuron receptive fields, along the ventral stream, become progressively and increasingly larger (Yamins and DiCarlo, 2016), complex (Pasupathy and Connor, 2001; Güçlü and van Gerven, 2015) and invariant to changes in the input-statistics (Quiroga *et al.*, 2005; Rust and DiCarlo, 2010), as shown in figure 4.3 (c).

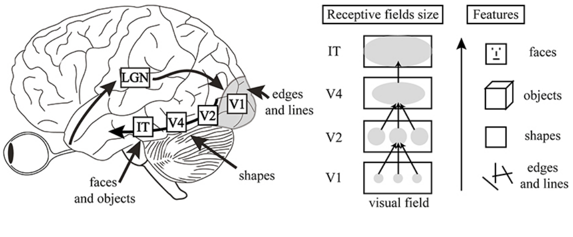


Figure 4.4: Receptive fields of the human vision system, taken from Manassi *et al.* (2013).

One of the basic frameworks in which neuroscientists study the sensory cortex is the encoding-decoding model: the former is the process where stimuli are transformed into neural activity patterns, whereas the latter is the process where neural activity generates behavior (Yamins and DiCarlo 2016; Wen *et al.* 2017, figure 4.6).

Figure 4.5 compares similarities between the ventral visual stream of macaque (top) and convolutional neural networks (down). V1 neurons have small receptive fields, paving a high-resolution retinotopic map, this spatio-temporal structure of each receptive field corresponds to a processing unit that locally filters a given property of the image (Medathati *et al.*, 2016).

Low-level features such as orientation, direction, color or disparity are encoded in different sub-populations forming a sparse and overcomplete representation of local feature dimensions. Moving along hierarchies, receptive fields become larger and encode features of increasing complexities and conjunctions (DeYoe and Van Essen, 1988; Roelfsema *et al.*, 2000).

The same properties are also found in convolutional neural networks receptive fields (Luo *et al.*, 2016): earlier units have edge-like receptive fields, later units respond to complex textures (Zeiler and Fergus, 2014; Mahendran and Vedaldi, 2016), and final layer units have class-specific receptive fields corresponding to specific object categories (Quiroga *et al.*, 2005; Le *et al.*, 2011; Mahendran and Vedaldi, 2016).

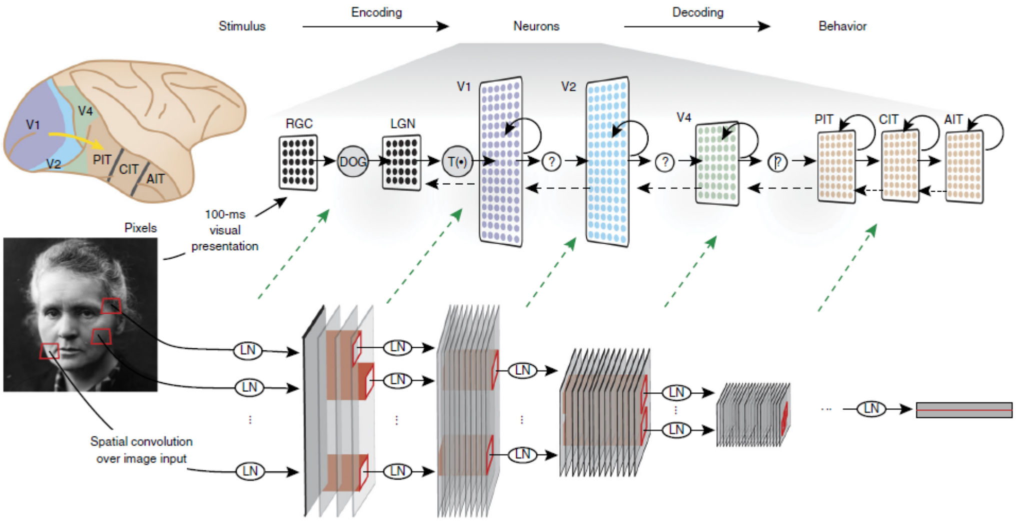


Figure 4.5: Convolutional neural network as models of human sensory cortex, taken from Yamins and DiCarlo (2016).

Convolutional neural networks trained for image classification have their units which resemble neurons in the ventral visual pathway of mammalians (Yamins and DiCarlo, 2016): units of earlier layers present Gabor-like receptive fields reminiscent of edge detectors neurons seen in V1 (Güçlü and van Gerven, 2015). Intermediate and later layers also show reminiscence of V4 and IT predictions, both in individual neurons and fMRI responses: using the same set of stimuli, researchers compared the representational dissimilarity matrices of neural networks models with the ones obtained from human IT (measured with fMRI) and monkey IT (measured with cell recording) (Khaligh-Razavi and Kriegeskorte, 2014; Yamins *et al.*, 2014; Güçlü and van Gerven, 2015; Yamins and DiCarlo, 2016).

Figure 4.3 (d) illustrate these computational models of receptive fields and its analysis has also recently become a canonical method for analyzing response properties in neural networks (Hinton and Salakhutdinov, 2006; Cadieu *et al.*, 2007; Serre *et al.*, 2007a,b; Nandy *et al.*, 2013; Cox and Dean, 2014; Zeiler and Fergus, 2014; Mahendran and Vedaldi, 2015; Yosinski *et al.*, 2015; Luo *et al.*, 2016; Mahendran and Vedaldi, 2016; Nguyen *et al.*, 2016; Cadena *et al.*, 2018).

V2 cells, the secondary visual cortex, are tuned to extract simple properties of visual signal like orientation, spatial frequency, and color. V4 detects intermediate object features like simple geometric shapes, owns strong attentional modulation features (Moran and Desimone, 1985), and receives direct inputs also from V1.

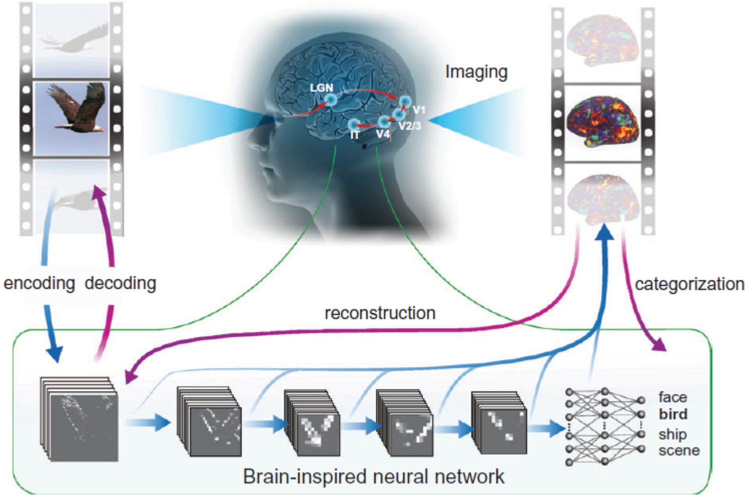


Figure 4.6: Neural encoding and decoding through a deep learning model, adapted from Wen *et al.* (2017).

Neural responses to stimuli in V2 are largely similar to those in V1, but this visual area is more sensitive to line conjunctions or corners (Peterhans and von der Heydt, 1989; Hegdé and Van Essen, 2000; Lee and Nguyen, 2001; Ito and Komatsu, 2004).

Also, intermediate layers of convolutional neural networks turn out to be state-of-the-art predictors of neural responses in V4 cortex (Yamins *et al.*, 2014), and lower layers contain a Gabor-wavelet-like activation pattern, providing effective models of voxel responses in V1 and V3 (Khaligh-Razavi and Kriegeskorte, 2014; Güçlü and van Gerven, 2015). fMRI measurements in humans demonstrated that neurons in V2 and V4 prefer stimuli with higher-order statistical dependencies (Freeman *et al.*, 2013; Okazawa *et al.*, 2015).

Inferior Temporal gyrus (IT) is one of the higher levels of the ventral stream, performs semantic-level tasks and it is associated with the representation of complex object features such as object recognition, identification, and categorization (Haxby *et al.*, 2000), comparing the processed information to stored memories (Kolb and Whishaw, 2001). Neurons in the IT cortex are also sensitive to whole patterns like faces (Kanwisher *et al.*, 1997; Epstein and Kanwisher, 1998; Gauthier *et al.*, 2000).

When glimpsing an object, the information arrives to IT within 100ms and will later begin to flow backward, thanks to the top-down feedback of the brain who updates the activations in the lower level brain areas (Goodfellow *et al.*, 2016). Convolutional neural networks can predict IT firing rates and perform very similarly to humans on object recognition tasks (Afraz *et al.*, 2014).

Convolutional neural networks that perform better on object recognition tasks also better predict cortical spiking data: top hidden layers represent an accurate image-computable model of spiking responses in IT cortex (Yamins *et al.*, 2013; Cadieu *et al.*, 2014; Khaligh-Razavi and Kriegeskorte, 2014; Yamins *et al.*, 2014; Güçlü and van Gerven, 2015; Kheradpisheh *et al.*, 2016). Kalfas *et al.* (2018) revealed a high similarity and correspondence between the activation of convolutional neural network units and the response of macaque IT neurons.

Also, Soltani and Koch (2010) implemented a salience computational model using spiking neural circuits, where cortical areas like V1, V2, and V4 perform only lateral excitation or inhibition. This full-scale model provides support for local computations to experimentally look for in each brain area (Veale *et al.*, 2017).

4.2 Visual attention

Biological brains allow the interaction and learning with the surrounding environment, being modular systems with distinct but interacting subsystems which underpins key functions such as cognitive control, language, and memory (Shallice, 1988; Anderson *et al.*, 2004; Marblestone *et al.*, 2016).

Most convolutional neural networks work directly on entire images or video frames, giving equal priority to all pixels at the earliest stage of processing (Hassabis *et al.*, 2017). Conversely, the primate visual system works differently: it shifts strategically among locations and objects, centering resources and coordinates on a series of regions (Desimone and Duncan, 1995; Ungerleider and G, 2000; Anderson, 2005). This behavior isolates and prioritizes the information that is relevant at any given moment (Koch and Ullman, 1987; Posner and Petersen, 1990; Salinas and Abbott, 1997; Olshausen *et al.*, 1993; Petersen and Posner, 2012; Moore and Zirnsak, 2017).

Top-down attention components are context-driven and goal-driven (Land and Lee, 1994; Ballard *et al.*, 1995; Land and Hayhoe, 2001; Hayhoe and Ballard, 2005; Borji *et al.*, 2013b). Implementing top-down attention components is a complex task, involving degrees of cognitive reasoning to attend objects and to recognize them in their context, in order to incrementally update the model's understanding and to planning the next most task-relevant shift of attention (Navalpakkam and Itti, 2005; Itti and Arbib, 2006; Kimura *et al.*, 2008; Yu *et al.*, 2008; Beuter *et al.*, 2009; Akamine *et al.*, 2010; Ban *et al.*, 2010; Yu *et al.*, 2011; Borji *et al.*, 2013b; Itti and Borji, 2015; Zhang *et al.*, 2018b).

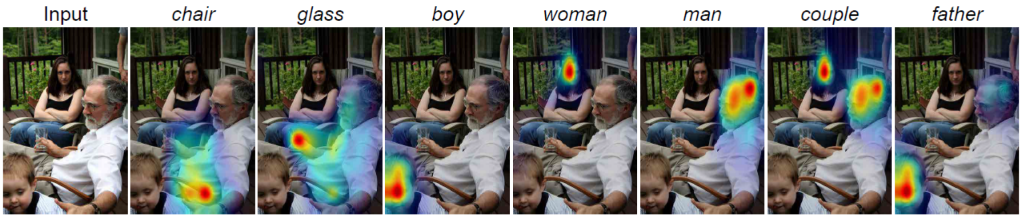


Figure 4.7: Top-down attention maps: given a selected semantic label, the neural network generates and highlights the related attention map, taken from Zhang *et al.* (2018b).

Instead, bottom-up attention components are stimulus-driven, saliency-based, and operates in a feed-forward manner: rapid parallel operations happen throughout the entire visual field and a series of successive transformations are applied to the entire visual field to highlight salient regions (Treisman and Gelade, 1980; Koch and Ullman, 1987; Itti and Koch, 2001; Borji, 2018).

Overt attention refers to the sensory apparatuses redirection towards features and locations that contain relevant information. This sampling of the visual world by moving eyes, head and body allows humans to perform selective modulation of the visual signal since it cannot be processed all at once.

The feature maps that represent basic visual features (like color, orientation, luminance, motion, etc.) compete within themselves to determine which map locations are significantly different from their surroundings. These maps are then normalized and combined into a "feature-agnostic" saliency map, which serves to determine the most likely target for attention (Parkhurst *et al.*, 2002; Peters *et al.*, 2005; Borji *et al.*, 2013a; Veale *et al.*, 2017). In cortical pathways, the information is processed through feature-to-priority maps, conversely the sub-cortical pathways process information in a feature-agnostic manner.

The *superior colliculus* (Schiller *et al.*, 1974) is a sub-cortical structure where the visual saliency 2D-map topographically encodes the conspicuity of a stimulus over the visual scene, this structure has proven to be an effective eye-movements predictor (Veale *et al.*, 2017). It is a phylogenetically-old midbrain structure in the visual control of orienting movements: its superficial layers have strong visual responses, whereas the deeper layers activity is related to eye-movements orientation (Veale *et al.*, 2017).

The superficial layers of the superior colliculus, through a center-surround inhibition mechanism, feed a priority-selection mechanism in its deeper layers, affecting the saccadic and micro-saccadic eye movements. They receive input from the retina and visual cortex, in order to send outputs to deeper layers. Studies demonstrate that its activity is correlated with the bottom-up salience of the visual input (White *et al.*, 2017).

The deeper layers of the superior colliculus receive associative inputs converging from the cortex, basal ganglia and the superficial layers of superior colliculus (Jayaraman *et al.*, 1977; Fries, 1984; Moschovakis *et al.*, 1988; Lee *et al.*, 1997; Tardif *et al.*, 2005). Physiologically its neurons are related to eye movements and can evoke overt attention (Veale *et al.*, 2017).

The goal of saliency modeling (Itti *et al.*, 1998) is to create a saliency map of an image (static saliency, Cornia *et al.* 2018; Huang *et al.* 2015; Jia and Bruce 2018; Kruthiventi *et al.* 2017; Kümmerer *et al.* 2014; Liu *et al.* 2015a; Vig *et al.* 2014) and taking into account, if possible, temporal relation (dynamic saliency, Bak *et al.* 2017; Bazzani *et al.* 2016; Chaabouni *et al.* 2016; Gorji and Clark 2018; Jiang *et al.* 2017; Sun *et al.* 2018; Wang *et al.* 2018). Based on these insights, several computational models of bottom-up attention were proposed in literature (Koch and Ullman, 1987; Borji and Itti, 2012; Borji *et al.*, 2012, 2013a; Li *et al.*, 2014; Borji *et al.*, 2015; Itti and Borji, 2015; Li and Yu, 2015; Pan and Giró-i Nieto, 2015; Pan *et al.*, 2016; Wang *et al.*, 2015; Zhao *et al.*, 2015; Wang and Shen, 2017; Xu *et al.*, 2017; Anderson *et al.*, 2018).

Recent focus in neuroscience on studies about attention as an orienting mechanism for perception (Summerfield *et al.*, 2006) gave inspiration in neural network architectures where attentional mechanisms have been used to select information to be read out from the internal memory of the network, leading to important advancements on memory and reasoning tasks (Graves *et al.*, 2014).

The attentional process of taking "glimpses" of input images at each step, update internal state representations, and select next locations to sample, allows convolutional neural networks to ignore irrelevant objects in a scene and perform better even in the presence of object clutter (Larochelle and Hinton, 2010; Mnih *et al.*, 2014).

Different attention mechanisms for fine-grained recognition exist in literature, using spatial transformers network or top-down feed-forward attention mechanisms (Jaderberg *et al.*, 2015; Liu *et al.*, 2016b; Rodríguez *et al.*, 2017). Rodríguez *et al.* (2018) proposed a modular attention mechanism capable of learning to attend lower-level feature activations without requiring part annotations, using these activations to update and rectify the output likelihood distribution of the convolutional neural network.

Attentional features allow to rapidly infer if another person is making eye contact, follow their gaze to identify their gaze target, categorize quick glances to objects, and identify when other people are paying attention (Land and Tatler, 2009). Literature shows efforts about the automatic detection and identification of these features from images and video, like gaze estimation and following (Funes Mora *et al.*, 2014; Recasens *et al.*, 2015; Zhang *et al.*, 2015; Krafka *et al.*, 2016; Chong *et al.*, 2017; Gatys *et al.*, 2017; Gorji and Clark, 2017; Recasens *et al.*, 2017; Zhang *et al.*, 2017; Chong *et al.*, 2018).

Also, in literature different attention mechanisms were proposed (Benfold and Reid, 2009; Cristani *et al.*, 2011; Soo Park and Shi, 2015; Chen and Grauman, 2018), including spatial attention mechanism for image captioning (Vinyals *et al.*, 2015; Xu *et al.*, 2015; Yang *et al.*, 2016; Lu *et al.*, 2017; Pedersoli *et al.*, 2017; Rennie *et al.*, 2017), for image classification (Cao *et al.*, 2015; Xiao *et al.*, 2015; Almeida *et al.*, 2017; Wang *et al.*, 2017; Jetley *et al.*, 2018), for egocentric activity recognition (Sudhakaran and Lanz, 2018), for semantic image segmentation (Chen *et al.*, 2014; Dai *et al.*, 2015; Liu *et al.*, 2015b; Long *et al.*, 2015; Noh *et al.*, 2015; Chen *et al.*, 2016; Lin *et al.*, 2016; Harley *et al.*, 2017; Hou *et al.*, 2017; Li *et al.*, 2018b), and for visual tracking and recognition (Smeulders *et al.*, 2013; Ba *et al.*, 2014; Luo *et al.*, 2014; Ma *et al.*, 2015; Choi *et al.*, 2016; Qi *et al.*, 2016; Choi *et al.*, 2017; Chu *et al.*, 2017; Kiani Galoogahi *et al.*, 2017; Kosiorek *et al.*, 2017; Song *et al.*, 2017; Pu *et al.*, 2018).

4.3 Biological plausibility of the backpropagation algorithm

The *credit assignment problem* concerns to determine, examining a system, how the success of the overall performance is determined by the various contributions of the system's components like planning, reasoning, learning, etc. (Minsky, 1961).

Three credit assignment problems can be identified: temporal (the identification, given a long sequence of actions, of which ones are useful or useless in obtaining the final feedback), structural (the finding of sensory situations set which a given actions sequence yield the same outcome), and transfer credit assignment (learning to generalize a given action sequence across tasks).

This kind of problem analysis led to the resolution of old connectionism problems (see section 3.3.3) and helped the rise of a new connectionist movement, also thanks to the backpropagation algorithm (see section 3.3.5).

Backpropagation offers an efficient solution to the credit assignment problem within artificial neural networks (LeCun *et al.*, 2015), but several aspects of this algorithm were viewed to be biologically implausible (Dayan and Abbott, 2001; Bengio *et al.*, 2015; Lee *et al.*, 2015; Lillicrap *et al.*, 2016; Bartunov *et al.*, 2018; Whittington and Bogacz, 2019).

Backpropagation is thought to require a perfectly symmetrical feedback and feed-forward connectivity, and these features were not observed in mammalian brains (Song *et al.*, 2005; Hassabis *et al.*, 2017), although recent work demonstrates that these constraints can be in some ways softened (Liao *et al.*, 2016; Lillicrap *et al.*, 2016).

Another critic on biological implausibility of backpropagation relies on the accessibility of information: the weight update process on neural networks requires the access of non-local information like the error signal generated from distant layers, whereas cortical plasticity in biological synapses depends primarily on local information like pre-synaptic and post-synaptic neuronal activity (Bi and Poo, 1998; Bengio *et al.*, 2015).

Speaking of analogies and insights between neural networks and biological brains, Marblestone *et al.* (2016) address issues on how the brain optimizes cost functions, how are diverse across cortex areas and how they change over development. Cost functions optimization in the human brain means that neurons in brain areas can change, somehow, the properties of their synapses.

This process occurs generally in shaping the internal representations and processes used by the brain, and the cost functions are highly tunable since they were shaped by evolution and ethological needs. Whereas, cost functions optimization in neural networks occurs primarily thanks to the use of error backpropagation (Werbos, 1974; Rumelhart and McClelland, 1986; Hinton, 1990; Baldi and Sadowski, 2015).

Although common assumptions (Bengio *et al.*, 2015), literature shows the proposal of many biologically-plausible mechanisms of gradient descent like the use of local activation differences, contrastive and local Hebbian plasticity learning, difference-target propagation and local learning, the use of random synaptic feedback weights and direct feedback alignment, spike-timing-dependent plasticity, the use of segregated dendrites and dendritic cortical microcircuits, etc. (O'Reilly, 1996; Xie and Seung, 2003; Balduzzi *et al.*, 2015; Bengio *et al.*, 2015; Lee *et al.*, 2015; Baldi and Sadowski, 2016; Liao *et al.*, 2016; Lillicrap *et al.*, 2016; Nøkland, 2016; Scellier and Bengio, 2016; Bengio *et al.*, 2017; Guerguiev *et al.*, 2017; Luo *et al.*, 2017; Whittington and Bogacz, 2017; Wiseman *et al.*, 2017; Betti *et al.*, 2018; Mostafa *et al.*, 2018; Sacramento *et al.*, 2018).

Above methods still lack some key aspects of biological realism: for example, brain neurons tend to be either excitatory or inhibitory but not both, whereas in artificial neural networks this could happen at the same time. Moreover, biological neurons are highly non-linear with dendrites which implements something similar to a three-layer neural network: therefore, individual biological neuron should be regarded as multi-component sub-networks, rather than single nodes (Mel, 1992; Antic *et al.*, 2010; Marblestone *et al.*, 2016). Backpropagation in neural networks is purely linear, whereas biological neurons interleave linear and non-linear operations. Moreover, biological neurons communicate by stochastic binary values (spikes) instead of continuous values (Bengio *et al.*, 2015).

The use of large-scale brain maps could help researchers to better understand how the brain implements cost optimization, where the training signals originates and what structures exist to constrain this optimization in order to find solutions about specific kinds of problems (Marblestone *et al.*, 2016). Various techniques exist to compare receptive fields that are optimized in a simulation of a particular cost function: including fMRI studies comprising of population receptive field estimation (where the population receptive field is computed from stimuli responses, estimates a highly accurate and precise visual field map, and represent a useful tool to link fMRI signals in the visual pathways to neuronal receptive fields), and representational dissimilarity matrices (where multi-channel measures of neural activity are quantitatively related to each other by comparing representational dissimilarity matrices, which characterize the information carried by a given brain representation) (Dumoulin and Wandell, 2008; Kriegeskorte *et al.*, 2008; Khaligh-Razavi and Kriegeskorte, 2014; Güçlü and van Gerven, 2015).

4.4 Machine learning for neuroimaging

Research in neuroimaging is directed towards data-driven feature learning methods like seed-based and canonical correlation analysis, independent component analysis, and many more (Biswal *et al.*, 1995; McKeown *et al.*, 1998; Plis *et al.*, 2014; Raichle *et al.*, 2001; Van Den Heuvel and Pol, 2010; Brookes *et al.*, 2011; Sui *et al.*, 2012).

Brain imaging techniques are highly successful in revealing known brain features with new details and supporting their credibility. They are diagnosis and disease assisting tools, revealing consistent patterns in data from uncontrolled resting-state experiments.

Machine learning techniques come in aid on the multivariate analysis of fMRI and MEG neuroimaging datasets (Cukur *et al.*, 2013; Kriegeskorte and Kievit, 2013; Cichy *et al.*, 2014), also with the encouraging promise to speed up connectomic analysis (Glasser *et al.*, 2016).

Functional magnetic resonance imaging (fMRI) represents a non-invasive brain scan which relies on the correlation between neuronal activation and its hemodynamic response: when a brain area is active, fMRI measures its intensity by detecting changes associated with the blood flow occurring to that brain region. This technique is referred as the blood oxygen level dependent (BOLD) contrast.

Figure 4.8 illustrates how neural networks can identify predictive variables: Lebedev *et al.* (2014) demonstrate that, by using MRI data, it is possible to identify which brain regions are most predictive for Alzheimer's disease diagnosis.

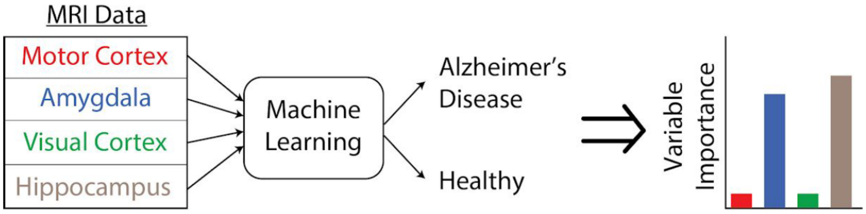


Figure 4.8: Identification of predictive variables by using machine learning techniques, adapted from Glaser *et al.* (2019).

fMRI, by using multi-voxel pattern analysis (Zafar *et al.*, 2015), represents an effective tool for decoding visual activities, stimulus categories classification, memories, imagination, dreams, and many more (Kay *et al.*, 2008; Mitchell *et al.*, 2008; Reddy *et al.*, 2010; Huth *et al.*, 2012; Horikawa *et al.*, 2013; Postle, 2015).

Researchers also used machine learning classifiers to diagnose neurological conditions from the functional activity of the brain and neuroimaging data, obtaining and improving the accurate estimates of neural activity from raw measurements: neuroanatomical measurements such as structural, functional, and diffusion MRI (sMRI, fMRI, dMRI) can distinguish healthy from unhealthy patients across conditions like schizophrenia, depression, autism, Alzheimer's disease, and ADHD (Sarraf *et al.*, 2016; Arbabshirani *et al.*, 2017; Rathore *et al.*, 2017; Vieira *et al.*, 2017; Bhagwat, 2018; Campese *et al.*, 2019).

Deep neural networks have been used to predict brain activities for visual tasks, using fMRI and other non-invasive measurements (Cichy *et al.*, 2016; Wen *et al.*, 2017). Research in neuroanatomy relies heavily on imaging techniques and advancements in deep neural networks represent an important asset for this task, consisting for example in approaches used to segment and label image parts (Ronneberger *et al.*, 2015; Litjens *et al.*, 2017).

From MRI scans Ghafoorian *et al.* (2017) and Fong *et al.* (2018) applied deep neural networks to identify white matter tracts from these scans and to exploit representations found in visual cortex.

Cichy *et al.* (2016), about the capture of visual perception process from brain's ventral and dorsal pathways, showed a comparison and correlation between temporal (MEG) and spatial (fMRI) brain visual representations, and visual representations coming from deep neural networks.

For visual image reconstruction tasks, Thirion *et al.* (2006) estimated the response model of each fMRI voxel in a retinotopic mapping experiment, reconstructing simple images composed of quickly rotating Gabor filters.

Miyawaki *et al.* (2008) reconstructed simple letters and graphics by solving the linear mapping model from visual cortex voxels to each image pixel, similar techniques were applied on letters (Schoenmakers *et al.*, 2013, 2014, 2015) and handwritten digits (Hossein-Zadeh *et al.*, 2016; Yargholi and Hossein-Zadeh, 2016).

Literature also shows the use of deep neural networks for visual images and video decoding (Nishimoto *et al.*, 2011; Yamins and DiCarlo, 2016; Svanera *et al.*, 2017). Agrawal *et al.* (2014) encoded fMRI signals using image features extracted from convolutional neural networks. Güçlü and van Gerven (2015, 2017) used deep neural networks to probe neural responses to naturalistic stimuli, showing an explicit gradient for feature complexity occurring from early visual areas toward the ventral and dorsal streams.

Horikawa and Kamitani (2017a,b) proposed decoding models based on hierarchical visual features of deep neural networks: they could be predicted from fMRI patterns and used them to identify seen or imagined object categories. Features decoded from the dream fMRI data also have a strong positive correlation with the intermediate and advanced deep neural network layer features of the dreamed objects (see figure 4.9).

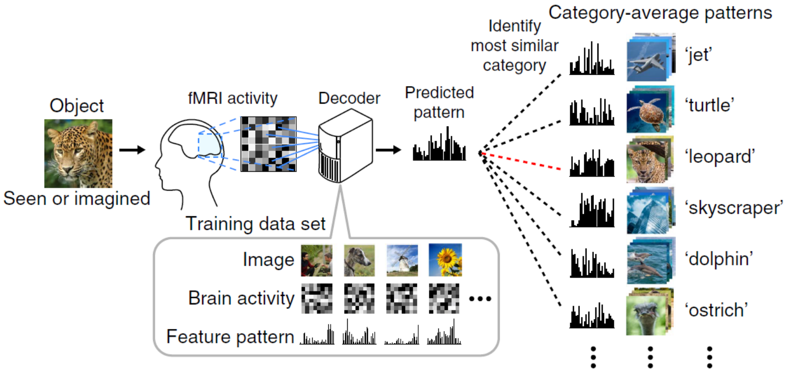


Figure 4.9: Decoding of seen or imagined objects using fMRI data, adapted from Horikawa and Kamitani (2017a).

Shen *et al.* (2019) proposed a method of visual image reconstruction from the brain: based of deep neural networks, both seen and imagined contents are revealed by capitalizing on multiple levels of visual cortical representations.

Zhang *et al.* (2018a) developed a method of natural images reconstruction from fMRI signals of human visual cortex, by using convolutional neural networks.
Figure 4.10 shows the main process of this visual image reconstruction.

During a natural-scenes display task, the subject's fMRI responses were acquired through an MRI scanner. A convolutional neural network is then used to extract hierarchical visual features from the fMRI stimuli responses acquired earlier.

To decode fMRI Signals to these features, the authors estimated multivariate regression models to map the distributed fMRI signals into output features of the convolutional neural network's artificial neurons.

The iterative optimization of the convolutional neural network serves to find the matched image (the natural scenes visual stimulation), whose unit features became the most similar to those predicted from the fMRI responses.
Finally, the matched image is chosen as the reconstruction result from brain activity.

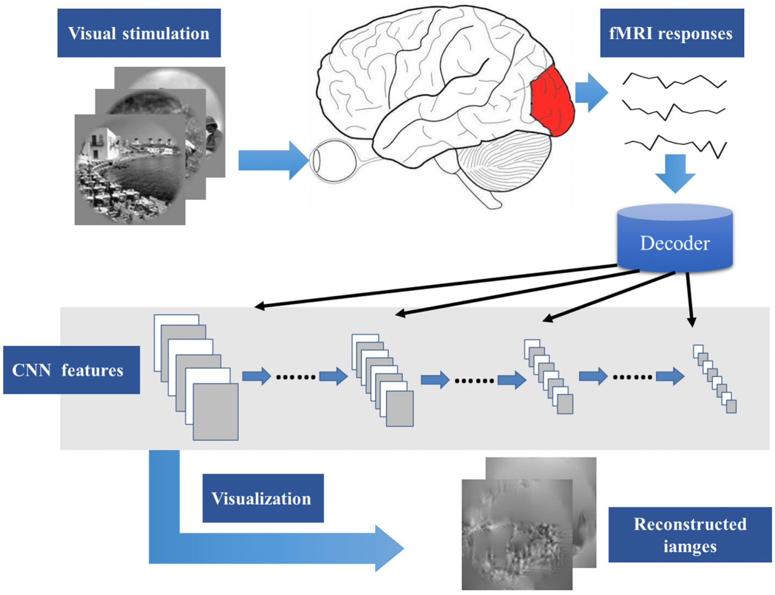


Figure 4.10: Image reconstruction from fMRI data, taken from Zhang *et al.* (2018a).

Convolutional neural network representations of visual stimuli using fMRI imaging show correspondence to processing stages in the ventral and dorsal streams of the visual system.

In order to take account of the high temporal resolution of neuroimaging data, a combination of encoding models based on deep neural networks with MEG data represents a useful approach to probe object processing dynamics in the human brain (Cichy *et al.*, 2017; Cichy and Teng, 2017; Seeliger *et al.*, 2018).

Neuroscientists and artificial intelligence researchers aim their efforts, using EEG data, to reverse-engineering human adaptive capabilities like brain responses, behavior, impulses, etc. (Palazzo *et al.*, 2017, 2018).

Using EEG data, deep learning models can predict imminent seizures caused by epilepsy (Nigam and Graupe, 2004; Brinkmann *et al.*, 2016; Talathi, 2017). Spampinato *et al.* (2017), for automated visual classification tasks, mapped visual features coming from EEG data evoked by visual object stimuli with visual features learned from deep neural networks.

Bashivan *et al.* (2015), to preserve the spatial, spectral, and temporal structure of EEG, proposed a deep neural network approach for learning invariant to inter-and-intra subjects representation from multi-channel EEG data: transforming them into a sequence of topology-preserving multi-spectral images and therefore learning robust representations from the sequence of images (see figure 4.11).

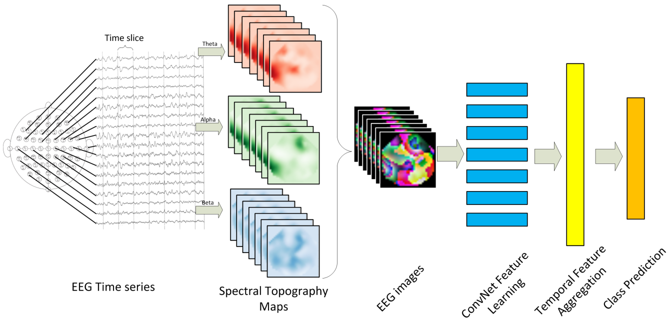


Figure 4.11: EEG time-series data for representational learning and classification, taken from Bashivan *et al.* (2015).

Chapter 5

Concluding remarks

Neural networks, as explained in this thesis, contribute to the understanding of the many dynamical aspects displayed by the human brain. But despite their success on explaining basic aspects of the human perception, feed-forward theories of convolutional neural networks still remain very schematic.

Many behavioral and anatomical aspects of the human biological visual processing still remain neglected and not yet fully understood nor appropriately modeled. For example, aspects like active vision, detail preservation, multi-stability, representation of concepts, space perception, more detailed aspects of top-down attention elements, and many more.

Literature about the visual pathways in biological organisms is extensive and continue to grow. This ever-growing knowledge is beneficial for improvements in deep neural networks modeling, understanding of transformations, for information representation and so on.

Ignoring the neural processing of these structures and oversimplification of many biological processing methods could also mislead about the real efficiency of biological visual systems.

Despite the adversities encountered by both fields of neuroscience and computer science, progresses made thus far constitutes undoubtedly a precious contribution on building deeply bio-inspired adaptive and versatile artificial systems, paving the way on the long journey ahead for the benefit of mankind.

Bibliography

Afraz, A., Yamins, D. L., and DiCarlo, J. J. (2014). Neural mechanisms underlying visual object recognition. In *Cold Spring Harbor symposia on quantitative biology*, volume 79, pages 99–107. Cold Spring Harbor Laboratory Press.

Agrawal, P., Stansbury, D., Malik, J., and Gallant, J. L. (2014). Pixels to voxels: Modeling visual representation in the human brain. *arXiv preprint arXiv:1407.5104*.

Ahrens, M. B., Li, J. M., Orger, M. B., Robson, D. N., Schier, A. F., Engert, F., and Portugues, R. (2012). Brain-wide neuronal dynamics during motor adaptation in zebrafish. *Nature*, **485**(7399), 471.

Akamine, K., Fukuchi, K., Kimura, A., and Takagi, S. (2010). Fully automatic extraction of salient objects from videos in near real time. *The Computer Journal*, **55**(1), 3–14.

Almeida, A. F., Figueiredo, R., Bernardino, A., and Santos-Victor, J. (2017). Deep networks for human visual attention: A hybrid model using foveal vision. In *Iberian Robotics conference*, pages 117–128. Springer.

Amidi, A. and Amidi, S. (2018). Cs 229 - machine learning tips and tricks cheatsheet. `https://github.com/afshinea/stanford-cs-229-machine-learning/raw/master/en/cheatsheet-machine-learning-tips-and-tricks.pdf`.

Anderson, J. R. (2005). *Cognitive psychology and its implications*. Macmillan.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, **111**(4), 1036.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086.

Antic, S. D., Zhou, W.-L., Moore, A. R., Short, S. M., and Ikonomu, K. D. (2010). The decade of the dendritic nmda spike. *Journal of neuroscience research*, **88**(14), 2991–3001.

Arbabshirani, M. R., Plis, S., Sui, J., and Calhoun, V. D. (2017). Single subject prediction of brain disorders in neuroimaging: promises and pitfalls. *Neuroimage*, **145**, 137–165.

Arbib, M. A. (2012). *Brains, machines, and mathematics*. Springer Science & Business Media.

Ba, J., Mnih, V., and Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*.

Babenko, A., Slesarev, A., Chigorin, A., and Lempitsky, V. (2014). Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer.

Bak, C., Kocak, A., Erdem, E., and Erdem, A. (2017). Spatio-temporal saliency networks for dynamic saliency prediction. *IEEE Transactions on Multimedia*, **20**(7), 1688–1698.

Baldi, P. and Sadowski, P. (2015). The ebb and flow of deep learning: a theory of local learning. *arXiv preprint arXiv:1506.06472*.

Baldi, P. and Sadowski, P. (2016). A theory of local learning, the learning channel, and the optimality of backpropagation. *Neural Networks*, **83**, 51–74.

Balduzzi, D., Vanchinathan, H., and Buhmann, J. (2015). Kickback cuts backprop's red-tape: biologically plausible credit assignment in neural networks. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Ballard, D. H., Hayhoe, M. M., and Pelz, J. B. (1995). Memory representations in natural tasks. *Journal of cognitive neuroscience*, **7**(1), 66–80.

Ban, S.-W., Kim, B., and Lee, M. (2010). Top-down visual selective attention model combined with bottom-up saliency map for incremental object perception. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Barrett, D. G., Morcos, A. S., and Macke, J. H. (2019). Analyzing biological and artificial neural networks: challenges with opportunities for synergy? *Current opinion in neurobiology*, **55**, 55–64.

Barretto, R. P., Messerschmidt, B., and Schnitzer, M. J. (2009). In vivo fluorescence imaging with high-resolution microlenses. *Nature methods*, **6**(7), 511.

Bartunov, S., Santoro, A., Richards, B., Marris, L., Hinton, G. E., and Lillicrap, T. (2018). Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In *Advances in Neural Information Processing Systems*, pages 9368–9378.

Bashivan, P., Rish, I., Yeasin, M., and Codella, N. (2015). Learning representations from eeg with deep recurrent-convolutional neural networks. *arXiv preprint arXiv:1511.06448*.

Bazzani, L., Larochelle, H., and Torresani, L. (2016). Recurrent mixture density network for spatiotemporal visual attention. *arXiv preprint arXiv:1603.08199*.

Benfold, B. and Reid, I. D. (2009). Guiding visual surveillance by tracking human attention. In *BMVC*, volume 2, page 7.

Bengio, Y. *et al.* (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, **2**(1), 1–127.

Bengio, Y., Lee, D.-H., Bornschein, J., Mesnard, T., and Lin, Z. (2015). Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*.

Bengio, Y., Mesnard, T., Fischer, A., Zhang, S., and Wu, Y. (2017). Stdp-compatible approximation of backpropagation in an energy-based model. *Neural computation*, **29**(3), 555–577.

Bernstein, J. G. and Boyden, E. S. (2011). Optogenetic tools for analyzing the neural circuits of behavior. *Trends in cognitive sciences*, **15**(12), 592–600.

Betti, A., Gori, M., and Marra, G. (2018). Backpropagation and biological plausibility. *arXiv preprint arXiv:1808.06934*.

Beuter, N., Lohmann, O., Schmidt, J., and Kummert, F. (2009). Directed attention-a cognitive vision system for a mobile robot. In *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pages 854–860. IEEE.

Bhagwat, N. (2018). *Prognostic applications for Alzheimer's disease using magnetic resonance imaging and machine-learning*. Ph.D. thesis.

Bi, G.-q. and Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, **18**(24), 10464–10472.

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological review*, **94**(2), 115.

Biswal, B., Zerrin Yetkin, F., Haughton, V. M., and Hyde, J. S. (1995). Functional connectivity in the motor cortex of resting human brain using echo-planar mri. *Magnetic resonance in medicine*, **34**(4), 537–541.

Bock, D. D., Lee, W.-C. A., Kerlin, A. M., Andermann, M. L., Hood, G., Wetzel, A. W., Yurgenson, S., Soucy, E. R., Kim, H. S., and Reid, R. C. (2011). Network anatomy and in vivo physiology of visual cortical neurons. *Nature*, **471**(7337), 177.

Borji, A. (2018). Saliency prediction in the deep learning era: An empirical investigation. *arXiv preprint arXiv:1810.03716*.

Borji, A. and Itti, L. (2012). State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, **35**(1), 185–207.

Borji, A., Sihite, D. N., and Itti, L. (2012). Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, **22**(1), 55–69.

Borji, A., Sihite, D. N., and Itti, L. (2013a). What stands out in a scene? a study of human explicit saliency judgment. *Vision research*, **91**, 62–77.

Borji, A., Sihite, D. N., and Itti, L. (2013b). What/where to look next? modeling top-down visual attention in complex interactive environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **44**(5), 523–538.

Borji, A., Cheng, M.-M., Jiang, H., and Li, J. (2015). Salient object detection: A benchmark. *IEEE transactions on image processing*, **24**(12), 5706–5722.

Brinkmann, B. H., Wagenaar, J., Abbot, D., Adkins, P., Bosshard, S. C., Chen, M., Tieng, Q. M., He, J., Muñoz-Almaraz, F., Botella-Rocamora, P., *et al.* (2016). Crowdsourcing reproducible seizure forecasting in human and canine epilepsy. *Brain*, **139**(6), 1713–1722.

Brookes, M. J., Woolrich, M., Luckhoo, H., Price, D., Hale, J. R., Stephenson, M. C., Barnes, G. R., Smith, S. M., and Morris, P. G. (2011). Investigating the electrophysiological basis of resting state networks using magnetoencephalography. *Proceedings of the National Academy of Sciences*, **108**(40), 16783–16788.

Brownlee, J. (2017). A gentle introduction to mini-batch gradient descent and how to configure batch size. `https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/`.

Bullier, J. (2001). Integrated model of visual processing. *Brain research reviews*, **36**(2-3), 96–107.

Burger, H. C., Schuler, C. J., and Harmeling, S. (2012). Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE conference on computer vision and pattern recognition*, pages 2392–2399. IEEE.

Cadena, S. A., Weis, M. A., Gatys, L. A., Bethge, M., and Ecker, A. S. (2018). Diverse feature visualizations reveal invariances in early layers of deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–232.

Cadena, S. A., Denfield, G. H., Walker, E. Y., Gatys, L. A., Tolias, A. S., Bethge, M., and Ecker, A. S. (2019). Deep convolutional models improve predictions of macaque v1 responses to natural images. *PLoS computational biology*, **15**(4), e1006897.

Cadieu, C., Kouh, M., Pasupathy, A., Connor, C. E., Riesenhuber, M., and Poggio, T. (2007). A model of v4 shape selectivity and invariance. *Journal of neurophysiology*, **98**(3), 1733–1750.

Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J., and DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, **10**(12), e1003963.

Campese, S., Lauriola, I., Scarpazza, C., Sartori, G., and Aiolli, F. (2019). Psychiatric disorders classification with 3d convolutional neural networks. In *INNS Big Data and Deep Learning conference*, pages 48–57. Springer.

Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., Wang, Z., Huang, Y., Wang, L., Huang, C., Xu, W., *et al.* (2015). Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2956–2964.

Caparrini, F. S. (2017). Entrenamiento de redes neuronales: mejorando el gradiente descendiente. `http://www.cs.us.es/~fsancho/?e=165`.

Carandini, M., Demb, J. B., Mante, V., Tolhurst, D. J., Dan, Y., Olshausen, B. A., Gallant, J. L., and Rust, N. C. (2005). Do we know what the early visual system does? *Journal of Neuroscience*, **25**(46), 10577–10597.

Chaabouni, S., Benois-Pineau, J., and Amar, C. B. (2016). Transfer learning with deep networks for saliency prediction in natural video. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1604–1608. IEEE.

Chaudhuri, R., Knoblauch, K., Gariel, M.-A., Kennedy, H., and Wang, X.-J. (2015). A large-scale circuit mechanism for hierarchical dynamical processing in the primate cortex. *Neuron*, **88**(2), 419–431.

Chen, C.-Y. and Grauman, K. (2018). Subjects and their objects: Localizing interactees for a person-centric view of importance. *International Journal of Computer Vision*, **126**(2-4), 292–313.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C., Yang, Y., Wang, J., Xu, W., and Yuille, A. L. (2016). Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649.

Choi, J., Jin Chang, H., Jeong, J., Demiris, Y., and Young Choi, J. (2016). Visual tracking using attention-modulated disintegration and integration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4321–4330.

Choi, J., Jin Chang, H., Yun, S., Fischer, T., Demiris, Y., and Young Choi, J. (2017). Attentional correlation filter network for adaptive visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4807–4816.

Chollet, F. (2016). How convolutional neural networks see the world. `https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html`.

Chong, E., Chanda, K., Ye, Z., Southerland, A., Ruiz, N., Jones, R. M., Rozga, A., and Rehg, J. M. (2017). Detecting gaze towards eyes in natural social interactions and its use in child assessment. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, **1**(3), 43.

Chong, E., Ruiz, N., Wang, Y., Zhang, Y., Rozga, A., and Rehg, J. M. (2018). Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 383–398.

Chu, Q., Ouyang, W., Li, H., Wang, X., Liu, B., and Yu, N. (2017). Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4836–4845.

Churchland, P. S. and Sejnowski, T. J. (1992). *The computational brain*. Cambridge,MA:MIT Press.

Cichy, R. M. and Kaiser, D. (2019). Deep neural networks as scientific models. *Trends in cognitive sciences*.

Cichy, R. M. and Teng, S. (2017). Resolving the neural dynamics of visual and auditory scene processing in the human brain: a methodological approach. *Philosophical Transactions of the Royal Society B: Biological Sciences*, **372**(1714), 20160108.

Cichy, R. M., Pantazis, D., and Oliva, A. (2014). Resolving human object recognition in space and time. *Nature neuroscience*, **17**(3), 455.

Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., and Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific reports*, **6**, 27755.

Cichy, R. M., Khosla, A., Pantazis, D., and Oliva, A. (2017). Dynamics of scene representations in the human brain revealed by magnetoencephalography and deep neural networks. *NeuroImage*, **153**, 346–358.

Cornia, M., Baraldi, L., Serra, G., and Cucchiara, R. (2018). Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, **27**(10), 5142–5154.

Cox, D. D. and Dean, T. (2014). Neural networks and neuroscience-inspired computer vision. *Current Biology*, **24**(18), R921–R929.

Cristani, M., Bazzani, L., Paggetti, G., Fossati, A., Tosato, D., Del Bue, A., Menegaz, G., and Murino, V. (2011). Social interaction discovery by statistical analysis of f-formations. In *BMVC*, volume 2, page 4.

Cukur, T., Nishimoto, S., Huth, A. G., and Gallant, J. L. (2013). Attention during natural vision warps semantic representation across the human brain. *Nature neuroscience*, **16**(6), 763.

Culurciello, E. (2017). Neural network architectures. `https://medium.com/p/156e5bad51ba`.

Cunningham, J. P. and Byron, M. Y. (2014). Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, **17**(11), 1500.

Dai, J., He, K., and Sun, J. (2015). Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643.

Dayan, P. and Abbott, L. F. (2001). Theoretical neuroscience: computational and mathematical modeling of neural systems.

Deisseroth, K. (2011). Optogenetics. *Nature methods*, **8**(1), 26.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Dertat, A. (2017). Applied deep learning - part 4: Convolutional neural networks. `https://towardsdatascience.com/584bc134c1e2`.

Deshpande, A. (2016). The 9 deep learning papers you need to know about (understanding cnns part 3). `https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html`.

Desimone, R. and Duncan, J. (1995). Neural mechanisms of selective visual attention. *Annual review of neuroscience*, **18**(1), 193–222.

DeYoe, E. A. and Van Essen, D. C. (1988). Concurrent processing streams in monkey visual cortex. *Trends in neurosciences*, **11**(5), 219–226.

DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, **73**(3), 415–434.

Dong, C., Loy, C. C., He, K., and Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, **38**(2), 295–307.

Du, J. (2017). The frontier of SGD and its variants in machine learning. In *2017 2nd International Conference on Mechatronics and Information Technology (ICMIT 2017)*. Francis Academic Press.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, **12**(Jul), 2121–2159.

Dumoulin, S. O. and Wandell, B. A. (2008). Population receptive field estimates in human visual cortex. *Neuroimage*, **39**(2), 647–660.

Eggermont, J. J. (1990). The correlative brain. In *The correlative brain*, pages 267–281. Springer.

Eickenberg, M., Gramfort, A., Varoquaux, G., and Thirion, B. (2017). Seeing it all: Convolutional network layers map the function of the human visual system. *NeuroImage*, **152**, 184–194.

Enroth-Cugell, C. and Robson, J. G. (1984). Functional characteristics and diversity of cat retinal ganglion cells. basic characteristics and quantitative description. *Investigative ophthalmology & visual science*, **25**(3), 250–267.

Epstein, R. and Kanwisher, N. (1998). A cortical representation of the local visual environment. *Nature*, **392**(6676), 598.

Felleman, D. J. and Van, D. E. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, **1**(1), 1–47.

Fong, R. C., Scheirer, W. J., and Cox, D. D. (2018). Using human brain activity to guide machine learning. *Scientific reports*, **8**(1), 5397.

Freeman, J., Ziemba, C. M., Heeger, D. J., Simoncelli, E. P., and Movshon, J. A. (2013). A functional and perceptual signature of the second visual area in primates. *Nature neuroscience*, **16**(7), 974.

Fries, W. (1984). Cortical projections to the superior colliculus in the macaque monkey: a retrograde study using horseradish peroxidase. *Journal of Comparative Neurology*, **230**(1), 55–76.

Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, **20**(3-4), 121–136.

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

Fukushima, K. (2003). Neocognitron for handwritten digit recognition. *Neurocomputing*, **51**, 161–180.

Funes Mora, K. A., Monay, F., and Odobez, J.-M. (2014). Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 255–258. ACM.

Gatys, L. A., Kümmerer, M., Wallis, T. S., and Bethge, M. (2017). Guiding human gaze with convolutional neural networks. *arXiv preprint arXiv:1712.06492*.

Gauthier, I., Skudlarski, P., Gore, J. C., and Anderson, A. W. (2000). Expertise for cars and birds recruits brain areas involved in face recognition. *Nature neuroscience*, **3**(2), 191.

Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 7538–7550.

Ghafoorian, M., Karssemeijer, N., Heskes, T., van Uden, I. W., Sanchez, C. I., Litjens, G., de Leeuw, F.-E., van Ginneken, B., Marchiori, E., and Platel, B. (2017). Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities. *Scientific Reports*, **7**(1), 5110.

Gibson, J. J. (2014). *The ecological approach to visual perception: classic edition*. Psychology Press.

Glaser, J. I., Benjamin, A. S., Farhoodi, R., and Kording, K. P. (2019). The roles of supervised machine learning in systems neuroscience. *Progress in neurobiology*.

Glasser, M. F., Smith, S. M., Marcus, D. S., Andersson, J. L., Auerbach, E. J., Behrens, T. E., Coalson, T. S., Harms, M. P., Jenkinson, M., Moeller, S., *et al.* (2016). The human connectome project's neuroimaging approach. *Nature neuroscience*, **19**(9), 1175.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

Golkov, V., Dosovitskiy, A., Sperl, J. I., Menzel, M. I., Czisch, M., Sämann, P., Brox, T., and Cremers, D. (2016). Q-space deep learning: twelve-fold shorter and model-free diffusion mri scans. *IEEE transactions on medical imaging*, **35**(5), 1344–1351.

Goodale, M. A. and Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in neurosciences*, **15**(1), 20–25.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press. `http://www.deeplearningbook.org`.

Gorji, S. and Clark, J. J. (2017). Attentional push: A deep convolutional network for augmenting image salience with shared attention modeling in social scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2510–2519.

Gorji, S. and Clark, J. J. (2018). Going from image to video saliency: Augmenting image salience with dynamic attentional push. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7501–7511.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Güçlü, U. and van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, **35**(27), 10005–10014.

Güçlü, U. and van Gerven, M. A. (2017). Increasingly complex representations of natural movies across the dorsal stream are shared between subjects. *NeuroImage*, **145**, 329–336.

Guerguiev, J., Lillicrap, T. P., and Richards, B. A. (2017). Towards deep learning with segregated dendrites. *ELife*, **6**, e22901.

Harley, A. W., Derpanis, K. G., and Kokkinos, I. (2017). Segmentation-aware convolutional networks using local attention masks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5038–5047.

Hassabis, D., Kumaran, D., Summerfield, C., and Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron*, **95**(2), 245–258.

Haxby, J. V., Hoffman, E. A., and Gobbini, M. I. (2000). The distributed human neural system for face perception. *Trends in cognitive sciences*, **4**(6), 223–233.

Hayhoe, M. and Ballard, D. (2005). Eye movements in natural behavior. *Trends in cognitive sciences*, **9**(4), 188–194.

Haykin, S. S. (2009). *Neural networks and learning machines*, volume 3. Pearson Upper Saddle River.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hegdé, J. and Van Essen, D. C. (2000). Selectivity for complex shapes in primate visual area v2. *Journal of Neuroscience*, **20**(5), RC61–RC61.

Helmstaedter, M., Briggman, K. L., Turaga, S. C., Jain, V., Seung, H. S., and Denk, W. (2013). Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, **500**(7461), 168.

Herculano-Houzel, S. (2012). The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost. *Proceedings of the National Academy of Sciences*, **109**(Supplement 1), 10661–10668.

Hinton, G., Srivastava, N., and Swersky, K. (2014a). Lecture 6c: The momentum method. `http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`.

Hinton, G., Srivastava, N., and Swersky, K. (2014b). Lecture 6e: rmsprop: Divide the gradient by a running average of its recent magnitude. `http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`.

Hinton, G. E. (1990). Connectionist learning procedures. In *Machine learning*, pages 555–610. Elsevier.

Hinton, G. E. (2011). Machine learning for neuroscience. *Neural systems & circuits*, **1**(1), 12.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, **313**(5786), 504–507.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, **117**(4), 500–544.

Horikawa, T. and Kamitani, Y. (2017a). Generic decoding of seen and imagined objects using hierarchical visual features. *Nature communications*, **8**, 15037.

Horikawa, T. and Kamitani, Y. (2017b). Hierarchical neural representation of dreamed objects revealed by brain decoding with deep neural network features. *Frontiers in computational neuroscience*, **11**, 4.

Horikawa, T., Tamaki, M., Miyawaki, Y., and Kamitani, Y. (2013). Neural decoding of visual imagery during sleep. *Science*, **340**(6132), 639–642.

Hossein-Zadeh, G.-A. *et al.* (2016). Reconstruction of digit images from human brain fmri activity through connectivity informed bayesian networks. *Journal of neuroscience methods*, **257**, 159–167.

Hou, Q., Massiceti, D., Dokania, P. K., Wei, Y., Cheng, M.-M., and Torr, P. H. (2017). Bottom-up top-down cues for weakly-supervised semantic segmentation. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 263–277. Springer.

Huang, X., Shen, C., Boix, X., and Zhao, Q. (2015). Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 262–270.

Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, **148**(3), 574–591.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, **160**(1), 106–154.

Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, **195**(1), 215–243.

Hubel, D. H. and Wiesel, T. N. (1977). Ferrier lecture: Functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, pages 1–59.

Huth, A. G., Nishimoto, S., Vu, A. T., and Gallant, J. L. (2012). A continuous semantic space describes the representation of thousands of object and action categories across the human brain. *Neuron*, **76**(6), 1210–1224.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Ito, M. and Komatsu, H. (2004). Representation of angles embedded within contour stimuli in area v2 of macaque monkeys. *Journal of Neuroscience*, **24**(13), 3313–3324.

Itti, L. and Arbib, M. A. (2006). Attention and the minimal subscene. *Action to language via the mirror neuron system*, pages 289–346.

Itti, L. and Borji, A. (2015). Computational models of attention. *arXiv preprint arXiv:1510.07182*.

Itti, L. and Koch, C. (2001). Computational modelling of visual attention. *Nature reviews neuroscience*, **2**(3), 194.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11), 1254–1259.

Jaderberg, M., Simonyan, K., Zisserman, A., *et al.* (2015). Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025.

Jarrett, K., Kavukcuoglu, K., LeCun, Y., *et al.* (2009). What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, pages 2146–2153. IEEE.

Jayaraman, A., Batton III, R. R., and Carpenter, M. B. (1977). Nigrotectal projections in the monkey: an autoradiographic study. *Brain research*, **135**(1), 147–152.

Jetley, S., Lord, N. A., Lee, N., and Torr, P. H. (2018). Learn to pay attention. *arXiv preprint arXiv:1804.02391*.

Jia, S. and Bruce, N. D. (2018). Eml-net: An expandable multi-layer network for saliency prediction. *arXiv preprint arXiv:1805.01047*.

Jiang, L., Xu, M., and Wang, Z. (2017). Predicting video saliency with object-to-motion cnn and two-layer convolutional lstm. *arXiv preprint arXiv:1709.06316*.

Jun, J. J., Steinmetz, N. A., Siegle, J. H., Denman, D. J., Bauza, M., Barbarits, B., Lee, A. K., Anastassiou, C. A., Andrei, A., Aydın, Ç., *et al.* (2017). Fully integrated silicon probes for high-density recording of neural activity. *Nature*, **551**(7679), 232.

Kalfas, I., Vinken, K., and Vogels, R. (2018). Representations of regular and irregular shapes by deep convolutional neural networks, monkey inferotemporal neurons and human judgments. *PLoS computational biology*, **14**(10), e1006557.

Kanwisher, N., McDermott, J., and Chun, M. M. (1997). The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of neuroscience*, **17**(11), 4302–4311.

Kaplan, E. (2004). The m, p, and k pathways of the primate visual system. *The visual neurosciences*, **1**, 481–493.

Kay, K. N., Naselaris, T., Prenger, R. J., and Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature*, **452**(7185), 352.

Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, **10**(11), e1003915.

Kheradpisheh, S. R., Ghodrati, M., Ganjtabesh, M., and Masquelier, T. (2016). Deep networks can resemble human feed-forward vision in invariant object recognition. *Scientific reports*, **6**, 32672.

Kiani Galoogahi, H., Fagg, A., and Lucey, S. (2017). Learning background-aware correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1135–1143.

Kim, J. S., Greene, M. J., Zlateski, A., Lee, K., Richardson, M., Turaga, S. C., Purcaro, M., Balkam, M., Robinson, A., Behabadi, B. F., *et al.* (2014). Space–time wiring specificity supports direction selectivity in the retina. *Nature*, **509**(7500), 331.

Kimura, A., Pang, D., Takeuchi, T., Yamato, J., and Kashino, K. (2008). Dynamic markov random fields for stochastic modeling of visual attention. In *2008 19th International Conference on Pattern Recognition*, pages 1–5. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klahr, D. (1982). Nonmonotone assessment of monotone development: An information processing analysis. In *U-shaped behavioral growth*, pages 63–86. Elsevier.

Koch, C. and Ullman, S. (1987). Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer.

Kolb, B. and Whishaw, I. Q. (2001). *An introduction to brain and behavior.* Worth Publishers.

Kosiorek, A., Bewley, A., and Posner, I. (2017). Hierarchical attentive recurrent tracking. In *Advances in Neural Information Processing Systems*, pages 3053–3061.

Kourtzi, Z. and Connor, C. E. (2011). Neural representations for object perception: structure, category, and adaptive coding. *Annual review of neuroscience*, **34**, 45–67.

Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., and Torralba, A. (2016). Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184.

Kravitz, D. J., Saleem, K. S., Baker, C. I., and Mishkin, M. (2011). A new neural framework for visuospatial processing. *Nature Reviews Neuroscience*, **12**(4), 217.

Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual review of vision science*, **1**, 417–446.

Kriegeskorte, N. and Kievit, R. A. (2013). Representational geometry: integrating cognition, computation, and the brain. *Trends in cognitive sciences*, **17**(8), 401–412.

Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, **2**, 4.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kruger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., Rodriguez-Sanchez, A. J., and Wiskott, L. (2012). Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE transactions on pattern analysis and machine intelligence*, **35**(8), 1847–1871.

Kruthiventi, S. S., Ayush, K., and Babu, R. V. (2017). Deepfix: A fully convolutional neural network for predicting human eye fixations. *IEEE Transactions on Image Processing*, **26**(9), 4446–4456.

Kubilius, J., Bracci, S., and de Beeck, H. P. O. (2016). Deep neural networks as a computational model for human shape sensitivity. *PLoS computational biology*, **12**(4), e1004896.

Kümmerer, M., Theis, L., and Bethge, M. (2014). Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *arXiv preprint arXiv:1411.1045*.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and brain sciences*, **40**.

Land, M. and Tatler, B. (2009). *Looking and acting: vision and eye movements in natural behaviour*. Oxford University Press.

Land, M. F. and Hayhoe, M. (2001). In what ways do eye movements contribute to everyday activities? *Vision research*, **41**(25-26), 3559–3565.

Land, M. F. and Lee, D. N. (1994). Where we look when we steer. *Nature*, **369**(6483), 742.

Larochelle, H. and Hinton, G. E. (2010). Learning to combine foveal glimpses with a third-order boltzmann machine. In *Advances in neural information processing systems*, pages 1243–1251.

Le, Q. V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y. (2011). Building high-level features using large scale unsupervised learning. *arXiv preprint arXiv:1112.6209*.

Lebedev, A., Westman, E., Van Westen, G., Kramberger, M., Lundervold, A., Aarsland, D., Soininen, H., Kłoszewska, I., Mecocci, P., Tsolaki, M., *et al.* (2014). Random forest ensembles for detection and prediction of alzheimer's disease with a good between-cohort robustness. *NeuroImage: Clinical*, **6**, 115–125.

LeCun, Y. (1987). *Modèles Connexionnistes de lapprentissage*. Ph.D. thesis, PhD thesis, These de Doctorat, Universite Paris VI.

LeCun, Y. *et al.* (1989a). Generalization and network design strategies. In *Connectionism in perspective*, volume 19. Citeseer.

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

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404.

LeCun, Y., Bengio, Y., *et al.* (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, **3361**(10), 1995.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., *et al.* (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, **521**(7553), 436.

Lee, D.-H., Zhang, S., Fischer, A., and Bengio, Y. (2015). Difference target propagation. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 498–515. Springer.

Lee, P. H., Helms, M. C., Augustine, G. J., and Hall, W. C. (1997). Role of intrinsic synaptic circuitry in collicular sensorimotor integration. *Proceedings of the National Academy of Sciences*, **94**(24), 13299–13304.

Lee, T. S. and Nguyen, M. (2001). Dynamics of subjective contour formation in the early visual cortex. *Proceedings of the National Academy of Sciences*, **98**(4), 1907–1911.

Lerman, G. M., Gill, J. V., Rinberg, D., and Shoham, S. (2018). Spatially and temporally precise optical probing of neural activity readout. In *Optics and the Brain*, pages BTu2C–3. Optical Society of America.

Li, F.-F., Johnson, J., and Yeung, S. (2018a). Cs231n: Convolutional neural networks for visual recognition. `http://cs231n.stanford.edu/`.

Li, G. and Yu, Y. (2015). Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5455–5463.

Li, H., Xiong, P., An, J., and Wang, L. (2018b). Pyramid attention network for semantic segmentation. *arXiv preprint arXiv:1805.10180*.

Li, Y., Hou, X., Koch, C., Rehg, J. M., and Yuille, A. L. (2014). The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287.

Liao, Q., Leibo, J. Z., and Poggio, T. (2016). How important is weight symmetry in backpropagation? In *Thirtieth AAAI Conference on Artificial Intelligence*.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, **7**, 13276.

Lin, G., Shen, C., Van Den Hengel, A., and Reid, I. (2016). Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203.

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, **42**, 60–88.

Liu, N., Han, J., Zhang, D., Wen, S., and Liu, T. (2015a). Predicting eye fixations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 362–370.

Liu, W., Wen, Y., Yu, Z., and Yang, M. (2016a). Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7.

Liu, X., Xia, T., Wang, J., Yang, Y., Zhou, F., and Lin, Y. (2016b). Fully convolutional attention networks for fine-grained recognition. *arXiv preprint arXiv:1603.06765*.

Liu, Z., Li, X., Luo, P., Loy, C.-C., and Tang, X. (2015b). Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE international conference on computer vision*, pages 1377–1385.

Livingstone, M. and Hubel, D. (1988). Segregation of form, color, movement, and depth: anatomy, physiology, and perception. *Science*, **240**(4853), 740–749.

Livingstone, M. S. and Hubel, D. H. (1984). Specificity of intrinsic connections in primate primary visual cortex. *Journal of Neuroscience*, **4**(11), 2830–2835.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Lu, J., Xiong, C., Parikh, D., and Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383.

Luo, H., Fu, J., and Glass, J. (2017). Adaptive bidirectional backpropagation: Towards biologically plausible error signal transmission in neural networks. *arXiv preprint arXiv:1702.07097*.

Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X., and Kim, T.-K. (2014). Multiple object tracking: A literature review. *arXiv preprint arXiv:1409.7618*.

Luo, W., Li, Y., Urtasun, R., and Zemel, R. (2016). Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906.

Ma, C., Huang, J.-B., Yang, X., and Yang, M.-H. (2015). Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 3074–3082.

Mahendran, A. and Vedaldi, A. (2015). Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196.

Mahendran, A. and Vedaldi, A. (2016). Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, **120**(3), 233–255.

Malach, R., Levy, I., and Hasson, U. (2002). The topography of high-order human object areas. *Trends in cognitive sciences*, **6**(4), 176–184.

Manassi, M., Sayim, B., and Herzog, M. H. (2013). When crowding of crowding leads to uncrowding. *Journal of Vision*, **13**(13), 10–10.

Marblestone, A. H., Wayne, G., and Kording, K. P. (2016). Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience*, **10**, 94.

Margolis, D. J., Lütcke, H., Schulz, K., Haiss, F., Weber, B., Kügler, S., Hasan, M. T., and Helmchen, F. (2012). Reorganization of cortical population activity imaged throughout long-term sensory deprivation. *Nature neuroscience*, **15**(11), 1539.

Markov, N. T., Ercsey-Ravasz, M., Van Essen, D. C., Knoblauch, K., Toroczkai, Z., and Kennedy, H. (2013). Cortical high-density counterstream architectures. *Science*, **342**(6158), 1238406.

Marr, D. *et al.* (1982). Vision: A computational investigation into the human representation and processing of visual information.

McCann, M. T., Jin, K. H., and Unser, M. (2017). Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine*, **34**(6), 85–95.

McClelland, J. L., Rumelhart, D. E., and Hinton, G. E. (1988). *The appeal of parallel distributed processing.* Morgan Kaufmann.

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

McKeown, M. J., Makeig, S., Brown, G. G., Jung, T.-P., Kindermann, S. S., Bell, A. J., and Sejnowski, T. J. (1998). Analysis of fmri data by blind separation into independent spatial components. *Human brain mapping*, **6**(3), 160–188.

Medathati, N. K., Neumann, H., Masson, G. S., and Kornprobst, P. (2016). Bio-inspired computer vision: Towards a synergistic approach of artificial and biological vision. *Computer Vision and Image Understanding*, **150**, 1–30.

Mel, B. W. (1992). The clusteron: toward a simple abstraction for a complex neuron. In *Advances in neural information processing systems*, pages 35–42.

Milner, A. D. and Goodale, M. A. (2008). Two visual systems re-viewed. *Neuropsychologia*, **46**(3), 774–785.

Minsky, M. (1961). Steps toward artificial intelligence. *Proceedings of the IRE*, **49**(1), 8–30.

Minsky, M. L. and Papert, S. A. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Mishkin, M. and Ungerleider, L. G. (1982). Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. *Behavioural brain research*, **6**(1), 57–77.

Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K.-M., Malave, V. L., Mason, R. A., and Just, M. A. (2008). Predicting human brain activity associated with the meanings of nouns. *science*, **320**(5880), 1191–1195.

Miyawaki, Y., Uchida, H., Yamashita, O., Sato, M.-a., Morito, Y., Tanabe, H. C., Sadato, N., and Kamitani, Y. (2008). Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. *Neuron*, **60**(5), 915–929.

Mnih, V., Heess, N., Graves, A., *et al.* (2014). Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212.

Moore, T. and Zirnsak, M. (2017). Neural mechanisms of selective visual attention. *Annual review of psychology*, **68**, 47–72.

Moran, J. and Desimone, R. (1985). Selective attention gates visual processing in the extrastriate cortex. *Science*, **229**(4715), 782–784.

Moschovakis, A., Karabelas, A., and Highstein, S. (1988). Structure-function relationships in the primate superior colliculus. ii. morphological identity of presaccadic neurons. *Journal of neurophysiology*, **60**(1), 263–302.

Mostafa, H., Ramesh, V., and Cauwenberghs, G. (2018). Deep supervised learning using local errors. *Frontiers in neuroscience*, **12**, 608.

Nandy, A. S., Sharpee, T. O., Reynolds, J. H., and Mitchell, J. F. (2013). The fine structure of shape tuning in area v4. *Neuron*, **78**(6), 1102–1115.

Navalpakkam, V. and Itti, L. (2005). Modeling the influence of task on attention. *Vision research*, **45**(2), 205–231.

Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Nesterov, Y. E. (1983). A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Soviet Mathematics Doklady*, **27**, 372–376.

Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., and Clune, J. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395.

Nigam, V. P. and Graupe, D. (2004). A neural-network-based detection of epilepsy. *Neurological research*, **26**(1), 55–60.

Nishimoto, S., Vu, A. T., Naselaris, T., Benjamini, Y., Yu, B., and Gallant, J. L. (2011). Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology*, **21**(19), 1641–1646.

Noh, H., Hong, S., and Han, B. (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528.

Nøkland, A. (2016). Direct feedback alignment provides learning in deep neural networks. In *Advances in neural information processing systems*, pages 1037–1045.

Ohki, K., Chung, S., Ch'ng, Y. H., Kara, P., and Reid, R. C. (2005). Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. *Nature*, **433**(7026), 597.

Okazawa, G., Tajima, S., and Komatsu, H. (2015). Image statistics underlying natural texture selectivity of neurons in macaque v4. *Proceedings of the National Academy of Sciences*, **112**(4), E351–E360.

Olshausen, B. A., Anderson, C. H., and Van Essen, D. C. (1993). A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, **13**(11), 4700–4719.

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural computation*, **8**(5), 895–938.

Packer, A. M., Russell, L. E., Dalgleish, H. W., and Häusser, M. (2015). Simultaneous all-optical manipulation and recording of neural circuit activity with cellular resolution in vivo. *Nature methods*, **12**(2), 140.

Palazzo, S., Spampinato, C., Kavasidis, I., Giordano, D., and Shah, M. (2017). Generative adversarial networks conditioned by brain signals. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3410–3418.

Palazzo, S., Spampinato, C., Kavasidis, I., Giordano, D., and Shah, M. (2018). Decoding brain representations by multimodal learning of neural activity and visual features. *arXiv preprint arXiv:1810.10974*.

Pan, J. and Giró-i Nieto, X. (2015). End-to-end convolutional network for saliency prediction. *arXiv preprint arXiv:1507.01422*.

Pan, J., Sayrol, E., Giro-i Nieto, X., McGuinness, K., and O'Connor, N. E. (2016). Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 598–606.

Paninski, L. and Cunningham, J. P. (2018). Neural data science: accelerating the experiment-analysis-theory cycle in large-scale neuroscience. *Current opinion in neurobiology*, **50**, 232–241.

Parkhurst, D., Law, K., and Niebur, E. (2002). Modeling the role of salience in the allocation of overt visual attention. *Vision research*, **42**(1), 107–123.

Pasupathy, A. and Connor, C. E. (2001). Shape representation in area v4: position-specific tuning for boundary conformation. *Journal of neurophysiology*, **86**(5), 2505–2519.

Pedersoli, M., Lucas, T., Schmid, C., and Verbeek, J. (2017). Areas of attention for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1242–1250.

Peterhans, E. and von der Heydt, R. (1989). Mechanisms of contour perception in monkey visual cortex. ii. contours bridging gaps. *Journal of Neuroscience*, **9**(5), 1749–1763.

Peters, R. J., Iyer, A., Itti, L., and Koch, C. (2005). Components of bottom-up gaze allocation in natural images. *Vision research*, **45**(18), 2397–2416.

Petersen, S. E. and Posner, M. I. (2012). The attention system of the human brain: 20 years after. *Annual review of neuroscience*, **35**, 73–89.

Peterson, J. C., Abbott, J. T., and Griffiths, T. L. (2017). Adapting deep network features to capture psychological representations: An abridged report. In *IJCAI*, pages 4934–4938.

Pinto, N., Doukhan, D., DiCarlo, J. J., and Cox, D. D. (2009). A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS computational biology*, **5**(11), e1000579.

Plis, S. M., Hjelm, D. R., Salakhutdinov, R., Allen, E. A., Bockholt, H. J., Long, J. D., Johnson, H. J., Paulsen, J. S., Turner, J. A., and Calhoun, V. D. (2014). Deep learning for neuroimaging: a validation study. *Frontiers in neuroscience*, **8**, 229.

Plunkett, K. and Elman, J. L. (1997). *Exercises in rethinking innateness: A handbook for connectionist simulations*. Mit Press.

Poggio, T. and Serre, T. (2013). Models of visual cortex. *Scholarpedia*, **8**(4), 3516.

Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, **4**(5), 1–17.

Posner, M. I. and Petersen, S. E. (1990). The attention system of the human brain. *Annual review of neuroscience*, **13**(1), 25–42.

Pospisil, D., Pasupathy, A., and Bair, W. (2016). Comparing the brains representation of shape to that of a deep convolutional neural network. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 516–523. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering.

Postle, B. R. (2015). The cognitive neuroscience of visual short-term memory. *Current opinion in behavioral sciences*, **1**, 40–46.

Potjans, T. C. and Diesmann, M. (2012). The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cerebral cortex*, **24**(3), 785–806.

Pu, S., Song, Y., Ma, C., Zhang, H., and Yang, M.-H. (2018). Deep attentive tracking via reciprocative learning. In *Advances in Neural Information Processing Systems*, pages 1931–1941.

Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., and Yang, M.-H. (2016). Hedged deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4303–4311.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, **12**(1), 145–151.

Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C., and Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, **435**(7045), 1102.

Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A., and Shulman, G. L. (2001). A default mode of brain function. *Proceedings of the National Academy of Sciences*, **98**(2), 676–682.

Rathore, S., Habes, M., Iftikhar, M. A., Shacklett, A., and Davatzikos, C. (2017). A review on neuroimaging-based classification studies and associated feature extraction methods for alzheimer's disease and its prodromal stages. *NeuroImage*, **155**, 530–548.

Rawat, W. and Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, **29**(9), 2352–2449.

Recasens, A., Khosla, A., Vondrick, C., and Torralba, A. (2015). Where are they looking? In *Advances in Neural Information Processing Systems*, pages 199–207.

Recasens, A., Vondrick, C., Khosla, A., and Torralba, A. (2017). Following gaze in video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1435–1443.

Reddy, L., Tsuchiya, N., and Serre, T. (2010). Reading the mind's eye: decoding category information during mental imagery. *Neuroimage*, **50**(2), 818–825.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature neuroscience*, **2**(11), 1019.

Rodríguez, P., Cucurull, G., Gonfaus, J. M., Roca, F. X., and Gonzalez, J. (2017). Age and gender recognition in the wild with deep attention. *Pattern Recognition*, **72**, 563–571.

Rodríguez, P., Gonfaus, J. M., Cucurull, G., XavierRoca, F., and Gonzàlez, J. (2018). Attend and rectify: a gated attention mechanism for fine-grained recovery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 349–364.

Roelfsema, P. R., Lamme, V. A., and Spekreijse, H. (2000). The implementation of visual routines. *Vision research*, **40**(10-12), 1385–1411.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, **65**(6), 386.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Rumelhart, D. E. and McClelland, J. L. (1986). Parallel distributed processing: explorations in the microstructure of cognition. volume 1. foundations.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, **323**(6088), 533–536.

Rust, N. C. and DiCarlo, J. J. (2010). Selectivity and tolerance ("invariance") both increase as visual information propagates from cortical area v4 to it. *Journal of Neuroscience*, **30**(39), 12978–12995.

Sacramento, J., Costa, R. P., Bengio, Y., and Senn, W. (2018). Dendritic cortical microcircuits approximate the backpropagation algorithm. In *Advances in Neural Information Processing Systems*, pages 8721–8732.

Salinas, E. and Abbott, L. (1997). Invariant visual responses from attentional gain fields. *Journal of Neurophysiology*, **77**(6), 3267–3272.

Sarraf, S., Tofighi, G., *et al.* (2016). Deepad: Alzheimer's disease classification via deep convolutional neural networks using mri and fmri. *BioRxiv*, page 070441.

Scellier, B. and Bengio, Y. (2016). Towards a biologically plausible backprop. *arXiv preprint arXiv:1602.05179*, **914**.

Schiller, P. H., Stryker, M., Cynader, M., and Berman, N. (1974). Response characteristics of single cells in the monkey superior colliculus following ablation or cooling of visual cortex. *Journal of Neurophysiology*, **37**(1), 181–194.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, **61**, 85–117.

Schoenmakers, S., Barth, M., Heskes, T., and van Gerven, M. (2013). Linear reconstruction of perceived images from human brain activity. *NeuroImage*, **83**, 951–961.

Schoenmakers, S., van Gerven, M., and Heskes, T. (2014). Gaussian mixture models improve fmri-based image reconstruction. In *2014 International Workshop on Pattern Recognition in Neuroimaging*, pages 1–4. IEEE.

Schoenmakers, S., Güçlü, U., Van Gerven, M., and Heskes, T. (2015). Gaussian mixture models and semantic gating improve reconstructions from human brain activity. *Frontiers in computational neuroscience*, **8**, 173.

Seeliger, K., Fritsche, M., Güçlü, U., Schoenmakers, S., Schoffelen, J.-M., Bosch, S., and van Gerven, M. (2018). Convolutional neural network-based encoding and decoding of visual object recognition in space and time. *NeuroImage*, **180**, 253–266.

Sermanet, P. and LeCun, Y. (2011). Traffic sign recognition with multi-scale convolutional networks. In *IJCNN*, pages 2809–2813.

Serre, T., Oliva, A., and Poggio, T. (2007a). A feedforward architecture accounts for rapid categorization. *Proceedings of the national academy of sciences*, **104**(15), 6424–6429.

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007b). Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (3), 411–426.

Shallice, T. (1988). *From neuropsychology to mental structure*. Cambridge University Press.

Sharpee, T. O., Kouh, M., and Reynolds, J. H. (2013). Trade-off between curvature tuning and position invariance in visual area v4. *Proceedings of the National Academy of Sciences*, **110**(28), 11618–11623.

Shen, G., Horikawa, T., Majima, K., and Kamitani, Y. (2019). Deep image reconstruction from human brain activity. *PLoS computational biology*, **15**(1), e1006633.

Sheperd, G. and Koch, C. (1990). Introduction to synaptic circuits. *The synpatic organization of the brain. Oxford University Press, New York Oxford*, pages 3–31.

Siegler, R. S. (1991). *Children's thinking*. Prentice-Hall, Inc.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M. (2013). Visual tracking: An experimental survey. *IEEE transactions on pattern analysis and machine intelligence*, **36**(7), 1442–1468.

Soltani, A. and Koch, C. (2010). Visual saliency computations: mechanisms, constraints, and the effect of feedback. *Journal of Neuroscience*, **30**(38), 12831–12843.

Song, S., Sjöström, P. J., Reigl, M., Nelson, S., and Chklovskii, D. B. (2005). Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS biology*, **3**(3), e68.

Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R. W., and Yang, M.-H. (2017). Crest: Convolutional residual learning for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2555–2564.

Soo Park, H. and Shi, J. (2015). Social saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4777–4785.

Spampinato, C., Palazzo, S., Kavasidis, I., Giordano, D., Souly, N., and Shah, M. (2017). Deep learning human mind for automated visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6809–6817.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, **15**(1), 1929–1958.

Sudhakaran, S. and Lanz, O. (2018). Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. *arXiv preprint arXiv:1807.11794*.

Sui, J., Adali, T., Yu, Q., Chen, J., and Calhoun, V. D. (2012). A review of multivariate methods for multimodal fusion of brain imaging data. *Journal of neuroscience methods*, **204**(1), 68–81.

Summerfield, J. J., Lepsien, J., Gitelman, D. R., Mesulam, M. M., and Nobre, A. C. (2006). Orienting attention based on long-term memory experience. *Neuron*, **49**(6), 905–916.

Sun, M., Zhou, Z., Hu, Q., Wang, Z., and Jiang, J. (2018). Sg-fcn: A motion and memory-based deep learning model for video saliency detection. *IEEE transactions on cybernetics*, (99), 1–12.

Sutskever, I., Martens, J., Dahl, G. E., and Hinton, G. E. (2013). On the importance of initialization and momentum in deep learning. *ICML (3)*, **28**(1139-1147), 5.

Svanera, M., Benini, S., Raz, G., Hendler, T., Goebel, R., and Valente, G. (2017). Deep driven fmri decoding of visual categories. *arXiv preprint arXiv:1701.02133*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Talathi, S. S. (2017). Deep recurrent neural networks for seizure detection and early seizure detection systems. *arXiv preprint arXiv:1706.03283*.

Tardif, E., Delacuisine, B., Probst, A., and Clarke, S. (2005). Intrinsic connectivity of human superior colliculus. *Experimental brain research*, **166**(3-4), 316–324.

Thirion, B., Duchesnay, E., Hubbard, E., Dubois, J., Poline, J.-B., Lebihan, D., and Dehaene, S. (2006). Inverse retinotopy: inferring the visual content of images from brain activation patterns. *Neuroimage*, **33**(4), 1104–1116.

Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, **12**(1), 97–136.

Ungerleider, L. G. (1982). Two cortical visual systems. *Analysis of visual behavior*, pages 549–586.

Ungerleider, L. G. and Haxby, J. V. (1994). 'what'and 'where'in the human brain. *Current opinion in neurobiology*, **4**(2), 157–165.

Ungerleider, S. K. and G, L. (2000). Mechanisms of visual attention in the human cortex. *Annual review of neuroscience*, **23**(1), 315–341.

Van Den Heuvel, M. P. and Pol, H. E. H. (2010). Exploring the brain network: a review on resting-state fmri functional connectivity. *European neuropsychopharmacology*, **20**(8), 519–534.

Van Essen, D. C. (2003). Organization of visual areas in macaque and human cerebral cortex. *The visual neurosciences*, **1**, 507–521.

Veale, R., Hafed, Z. M., and Yoshida, M. (2017). How is visual salience computed in the brain? insights from behaviour, neurobiology and modelling. *Philosophical Transactions of the Royal Society B: Biological Sciences*, **372**(1714), 20160113.

Veit, A., Wilber, M. J., and Belongie, S. (2016). Residual networks behave like ensembles of relatively shallow networks. In *Advances in neural information processing systems*, pages 550–558.

Victor, J. D. (2005). Analyzing receptive fields, classification images and functional images: challenges with opportunities for synergy. *Nature neuroscience*, **8**(12), 1651.

Vieira, S., Pinaya, W. H., and Mechelli, A. (2017). Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications. *Neuroscience & Biobehavioral Reviews*, **74**, 58–75.

Vig, E., Dorr, M., and Cox, D. (2014). Large-scale optimization of hierarchical features for saliency prediction in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2798–2805.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Vito, E. D., Rosasco, L., Caponnetto, A., Giovannini, U. D., and Odone, F. (2005). Learning from examples as an inverse problem. *Journal of Machine Learning Research*, **6**(May), 883–904.

Vogt, N. (2018). Machine learning in neuroscience. *Nature Methods*, **15**(1), 33.

Vu, M.-A. T., Adalı, T., Ba, D., Buzsáki, G., Carlson, D., Heller, K., Liston, C., Rudin, C., Sohal, V. S., Widge, A. S., *et al.* (2018). A shared vision for machine learning in neuroscience. *Journal of Neuroscience*, **38**(7), 1601–1607.

Wandell, B. A., Dumoulin, S. O., and Brewer, A. A. (2007). Visual field maps in human cortex. *Neuron*, **56**(2), 366–383.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., and Tang, X. (2017). Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164.

Wang, H. and Raj, B. (2017). On the origin of deep learning. *arXiv preprint arXiv:1702.07800*.

Wang, L., Lu, H., Ruan, X., and Yang, M.-H. (2015). Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3183–3192.

Wang, W. and Shen, J. (2017). Deep visual attention prediction. *IEEE Transactions on Image Processing*, **27**(5), 2368–2378.

Wang, W., Shen, J., Guo, F., Cheng, M.-M., and Borji, A. (2018). Revisiting video saliency: A large-scale benchmark and a new model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4894–4903.

Wen, H., Shi, J., Zhang, Y., Lu, K.-H., Cao, J., and Liu, Z. (2017). Neural encoding and decoding with deep learning for dynamic natural vision. *Cerebral Cortex*, **28**(12), 4136–4160.

Werbos, P. (1974). Beyond regression:" new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*.

White, B. J., Berg, D. J., Kan, J. Y., Marino, R. A., Itti, L., and Munoz, D. P. (2017). Superior colliculus neurons encode a visual saliency map during free viewing of natural dynamic video. *Nature Communications*, **8**, 14263.

Whittington, J. C. and Bogacz, R. (2017). An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, **29**(5), 1229–1262.

Whittington, J. C. and Bogacz, R. (2019). Theories of error back-propagation in the brain. *Trends in cognitive sciences*.

Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. In *IRE WESCON Convention Record*, volume 4, pages 96–104.

Wiseman, S., Chopra, S., Ranzato, M., Szlam, A., Sun, R., Chintala, S., and Vasilache, N. (2017). Training language models using target-propagation. *arXiv preprint arXiv:1702.04770*.

Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., and Zhang, Z. (2015). The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 842–850.

Xie, J., Xu, L., and Chen, E. (2012). Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems*, pages 341–349.

Xie, X. and Seung, H. S. (2003). Equivalence of backpropagation and contrastive hebbian learning in a layered network. *Neural computation*, **15**(2), 441–454.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*.

Xu, Y., Gao, S., Wu, J., Li, N., and Yu, J. (2017). Personalized saliency and its prediction. *arXiv preprint arXiv:1710.03011*.

Yamane, Y., Carlson, E. T., Bowman, K. C., Wang, Z., and Connor, C. E. (2008). A neural code for three-dimensional object shape in macaque inferotemporal cortex. *Nature neuroscience*, **11**(11), 1352.

Yamins, D. L. and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, **19**(3), 356.

Yamins, D. L., Hong, H., Cadieu, C., and DiCarlo, J. J. (2013). Hierarchical modular optimization of convolutional networks achieves representations similar to macaque it and human ventral stream. In *Advances in neural information processing systems*, pages 3093–3101.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, **111**(23), 8619–8624.

Yang, Z., Yuan, Y., Wu, Y., Cohen, W. W., and Salakhutdinov, R. R. (2016). Review networks for caption generation. In *Advances in Neural Information Processing Systems*, pages 2361–2369.

Yargholi, E. and Hossein-Zadeh, G.-A. (2016). Brain decoding-classification of hand written digits from fmri data employing bayesian networks. *Frontiers in human neuroscience*, **10**, 351.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.

Yu, Y., Mann, G. K., and Gosine, R. G. (2008). An object-based visual attention model for robots. In *2008 IEEE International Conference on Robotics and Automation*, pages 943–948. IEEE.

Yu, Y., Mann, G. K., and Gosine, R. G. (2011). A goal-directed visual perception system using object-based top–down attention. *IEEE Transactions on Autonomous Mental Development*, **4**(1), 87–103.

Yuille, A. L. and Liu, C. (2018). Deep nets: What have they ever done for vision? *arXiv preprint arXiv:1805.04025*.

Zafar, R., Malik, A. S., Kamel, N., Dass, S. C., Abdullah, J. M., Reza, F., and Abdul Karim, A. H. (2015). Decoding of visual information from human brain activity: A review of fmri and eeg studies. *Journal of integrative neuroscience*, **14**(02), 155–168.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Zeki, S. and Shipp, S. (1988). The functional logic of cortical connections. *Nature*, **335**(6188), 311.

Zhang, C., Qiao, K., Wang, L., Tong, L., Zeng, Y., and Yan, B. (2018a). Constraint-free natural image reconstruction from fmri signals based on convolutional neural network. *Frontiers in human neuroscience*, **12**.

Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., and Sclaroff, S. (2018b). Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, **126**(10), 1084–1102.

Zhang, X., Sugano, Y., Fritz, M., and Bulling, A. (2015). Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4511–4520.

Zhang, X., Sugano, Y., and Bulling, A. (2017). Everyday eye contact detection using unsupervised gaze target discovery. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 193–203. ACM.

Zhao, R., Ouyang, W., Li, H., and Wang, X. (2015). Saliency detection by multi-context deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1265–1274.

Zhou, Y.-T. and Chellappa, R. (1988). Computation of optical flow using a neural network. In *IEEE International Conference on Neural Networks*, volume 1998, pages 71–78.