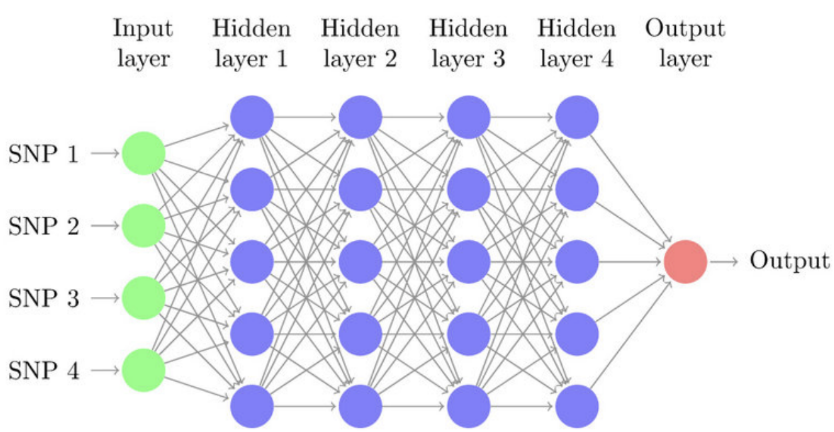
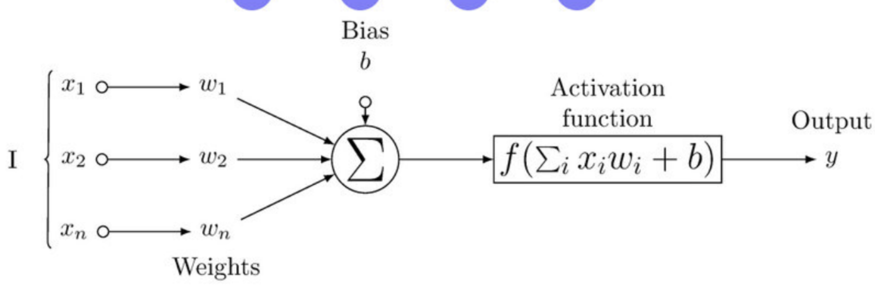
Alberto Andrés Valdés González.
Degree: Mathematical Engineer.
Work position: Data Scientist.
Mail: anvaldes@uc.cl/alberto.valdes.gonzalez.96@gmail.com
Location: Santiago, Chile.

Deep Learning Review

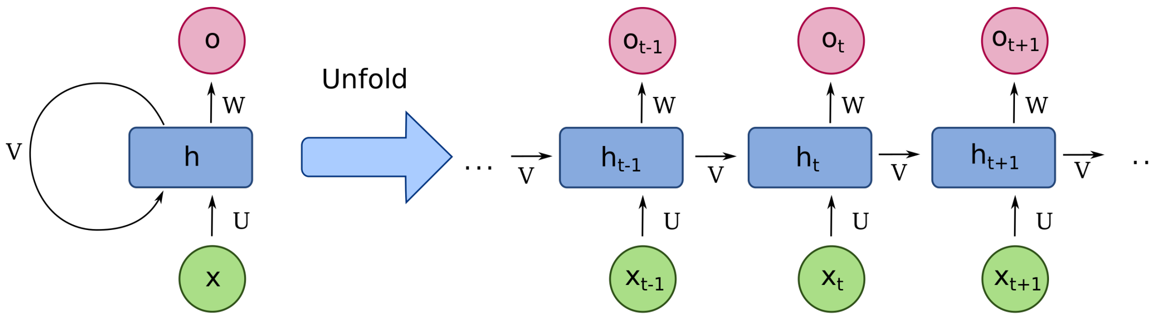
Deep learning is the subset of machine learning methods based on artificial neural networks with representation learning. The adjective ”deeprefers to the use of multiple layers in the network. Methods used can be either supervised, semi-supervised or unsupervised.

Multilayer Perceptron: A multilayer perceptron (MLP) is a misnomer for a modern feedforward artificial neural network, consisting of fully connected neurons with a nonlinear kind of activation function, organized in at least three layers, notable for being able to distinguish data that is not linearly separable.

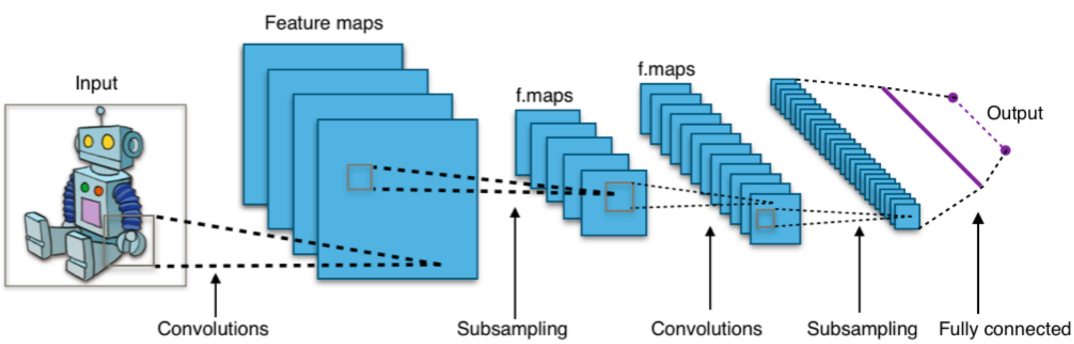




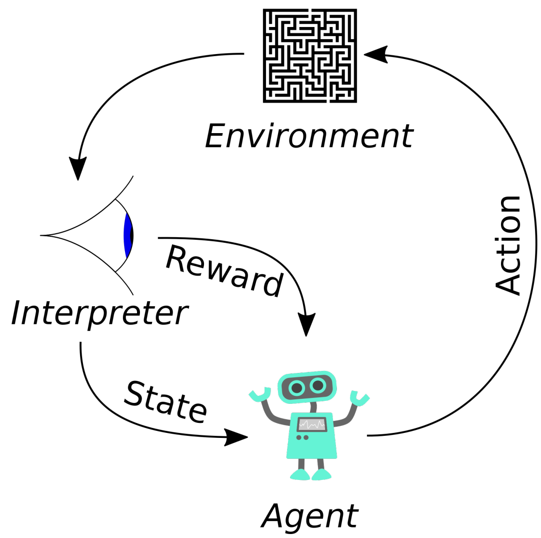
RNN: A recurrent neural network (RNN) is one of the two broad types of artificial neural network, characterized by direction of the flow of information between its layers. In contrast to the uni-directional feedforward neural network, it is a bi-directional artificial neural network, meaning that it allows the output from some nodes to affect subsequent input to the same nodes.



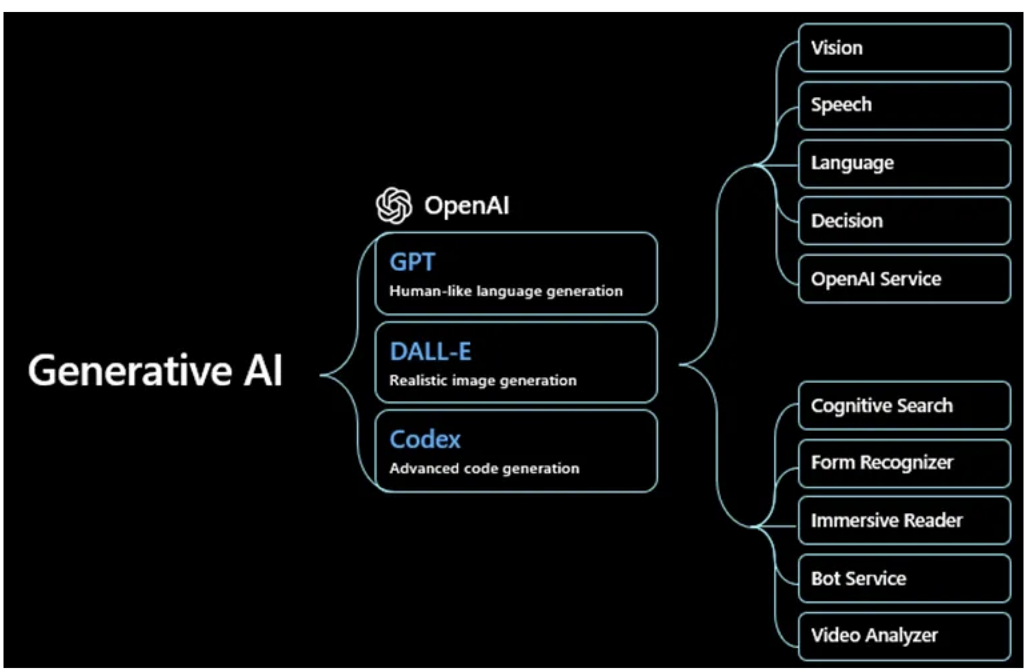
CNN: Convolutional neural network (CNN) is a regularized type of feed-forward neural network that learns feature engineering by itself via filters (or kernel) optimization. Vanishing gradients and exploding gradients, seen during backpropagation in earlier neural networks, are prevented by using regularized weights over fewer connections.



RL: Reinforcement learning (RL) is an interdisciplinary area of machine learning and optimal control concerned with how an intelligent agent ought to take actions in a dynamic environment in order to maximize the cumulative reward.



Generative AI: Generative artificial intelligence (also generative AI or GenAI) is artificial intelligence capable of generating text, images, or other media, using generative models.[2][3][4] Generative AI models learn the patterns and structure of their input training data and then generate new data that has similar characteristics.



Tensorflow: TensorFlow is an end-to-end deep learning framework. It was developed by the Google Brain team, initially for internal use at Google for research and production; however, it was made open-source in 2015 under the Apache License. In September 2019, a new updated version of TensorFlow was released by Google under the name TensorFlow 2.0.

Keras: Keras is a popular open-source high-level neural network API developed by François Chollet and released in 2015. The documentation refers to it as, "an API designed for human beings, not machines."

In mid-2017, the framework was adopted and integrated into TensorFlow, making it accessible to TensorFlow users through the tf.keras module.

Pytorch: PyTorch is the newest deep learning framework of the bunch. Developed by Facebook's (now Meta) AI research group and open-sourced in 2016, PyTorch is now part of the Linux Foundation umbrella.

Over the years, PyTorch has built a reputation as a framework of simplicity, flexibility, and efficiency. It enables developers to rapidly build complex neural networks for applications such as computer vision and natural language processing.

Activation Function: Activation function of a node in an artificial neural network is a function that calculates the output of the node (based on its inputs and the weights on individual inputs). Nontrivial problems can be solved only using a nonlinear activation function.

Name	**Function**
Sigmoid	$\sigma(x) = \dfrac{1}{1+e^{-x}}$
Hyperbolic Tangent	$tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$
GELU	$f(x) = x \cdot \Phi(x)$
Softplus	$f(x) = ln(1+e^x)$
SiLU	$f(x) = \dfrac{x}{1+e^{-x}}$
Gaussian	$f(x) = e^{-x^2}$

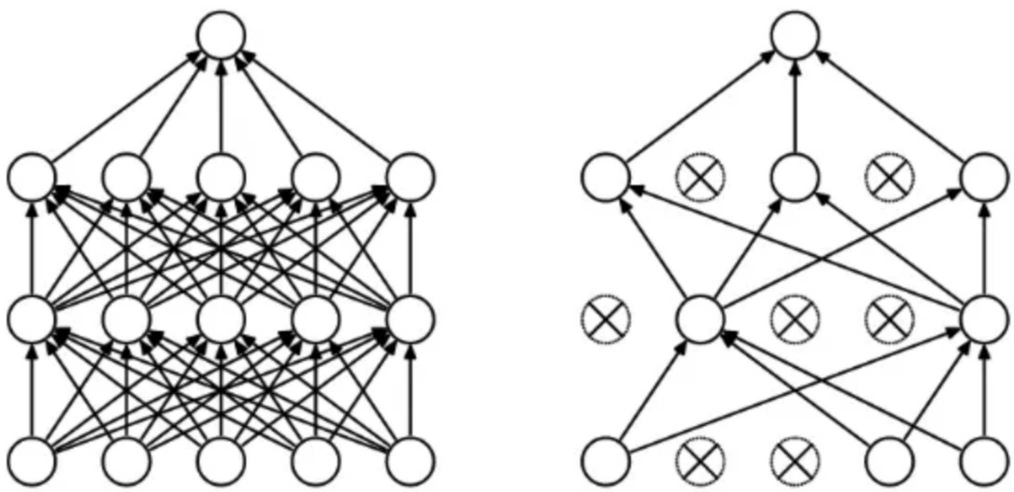
Vanishing Gradients: In machine learning, the vanishing gradient problem is encountered when training artificial neural networks with gradient-based learning methods and backpropagation. In such methods, during each iteration of training each of the neural networks weights receives an update proportional to the partial derivative of the error function with respect to the current weight. The problem is that in some cases, the gradient will be vanishingly small, effectively preventing the weight from changing its value. In the worst case, this may completely stop the neural network from further training.

Dying ReLU: A "deadReLU always outputs the same value (zero as it happens, but that is not important) for any input. Probably this is arrived at by learning a large negative bias term for its weights. In turn, that means that it takes no role in discriminating between inputs. For classification, you could visualise this as a decision plane outside of all possible input data.

Dropouts: The deep neural networks have different architectures, sometimes shallow, sometimes very deep trying to generalise on the given dataset. But, in this pursuit of trying too hard to learn different features from the dataset, they sometimes learn the statistical noise in the dataset. This definitely improves the model performance on the training dataset but fails massively on new data points (test dataset). This is the problem of overfitting. To tackle this problem we have various regularisation techniques that penalise the weights of the network but this wasn't enough.

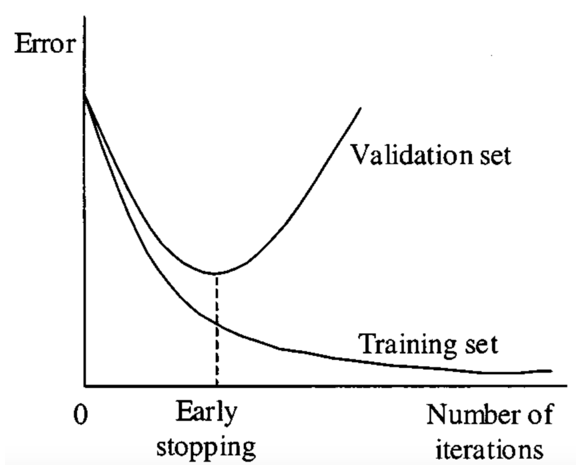
The best way to reduce overfitting or the best way to regularise a fixed-size model is to get the average predictions from all possible settings of the parameters and aggregate the final output. But, this becomes too computationally expensive and isn't feasible for a real-time inference/prediction.

The other way is inspired by the ensemble techniques (such as AdaBoost, XGBoost, and Random Forest) where we use multiple neural networks of different architectures. But this requires multiple models to be trained and stored, which over time becomes a huge challenge as the networks grow deeper. So, we have a great solution known as Dropout Layers.



(a) Standard Neural Net (b) After applying dropout.

Early Stopping: Early Stopping is a regularization technique for deep neural networks that stops training when parameter updates no longer begin to yield improves on a validation set.



Vectorization: In the context of high-level languages like Python, Matlab, and R, the term vectorization describes the use of optimized, pre-compiled code written in a low-level language (e.g. C) to perform mathematical operations over a sequence of data. This is done in place of an explicit iteration written in the native language code (e.g. a "for-loop" written in Python). Vectorization allows the elimination of the for-loops in python code. It is especially important in Deep learning as we are dealing with large numbers of datasets.

Tokenization: Tokenization breaks text into smaller parts for easier machine analysis, helping machines understand human language.

Tokenization methods vary based on the granularity of the text breakdown and the specific requirements of the task at hand. These methods can range from dissecting text into individual words to breaking them down into characters or even smaller units. Here's a closer look at the different types:

<u>Word tokenization:</u> This method breaks text down into individual words. It's the most common approach and is particularly effective for languages with clear word boundaries like English.

<u>Character tokenization:</u> Here, the text is segmented into individual characters. This method is beneficial for languages that lack clear word boundaries or for tasks that require a granular analysis, such as spelling correction.

Data augmentation: Data augmentation is a technique in machine learning used to reduce overfitting when training a machine learning model, by training models on several slightly-modified copies of existing data.

Transfer Learning: Transfer learning (TL) is a technique in machine learning (ML) in which knowledge learned from a task is re-used in order to boost performance on a related task.

Fine Tuning: In deep learning, fine-tuning is an approach to transfer learning in which the weights of a pre-trained model are trained on new data.
