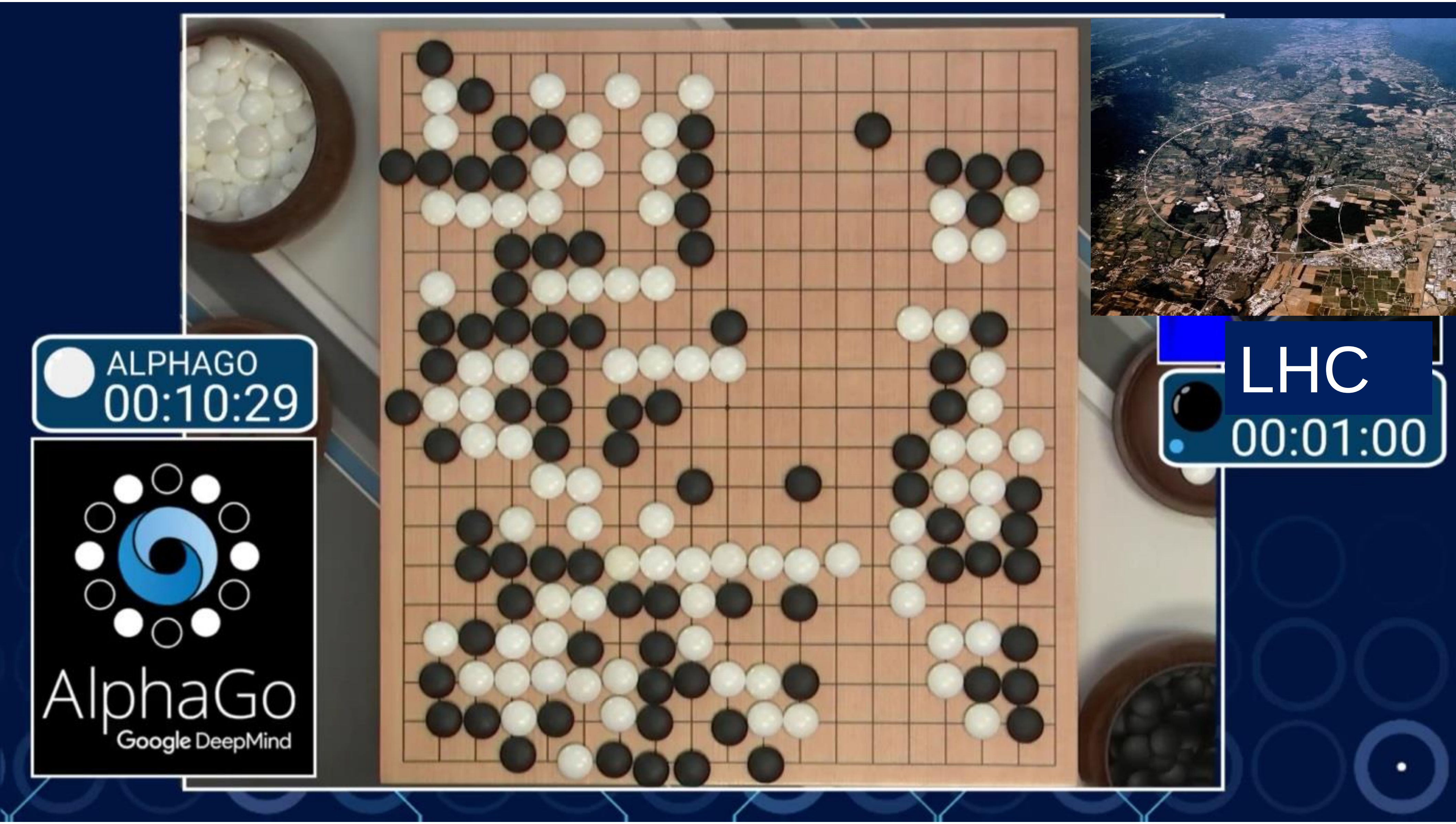
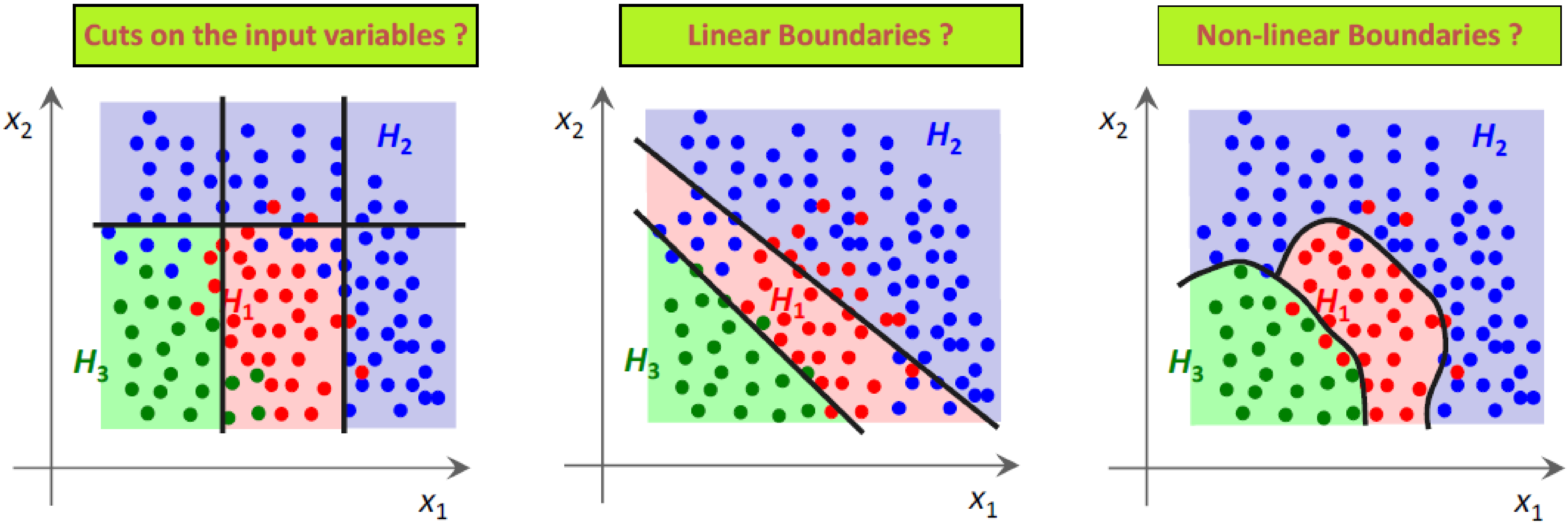
Deep learning at LHC



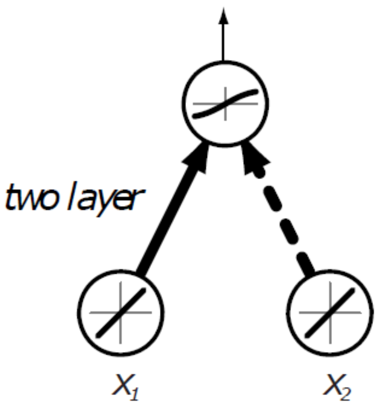
Event classification problem (applied to HEP)

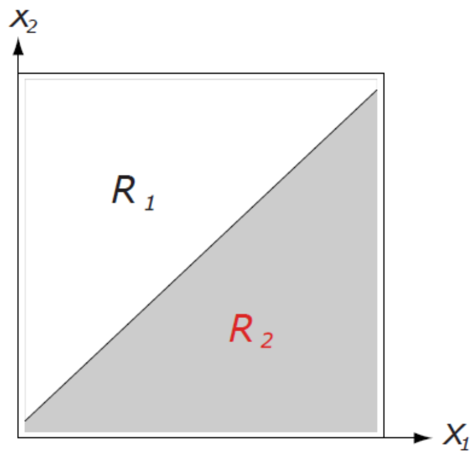
The question: what ‘decision boundary’ should we use to accept/reject events as belonging to event types *H1, H2 or H3?*



Methods available (up to 2015): Rectangular cut optimization, Projective likelihood estimation, Multidimensional probability density estimation, Multidimensional k-nearest neighbor classifier, Linear discriminant analysis (H-Matrix and Fisher discriminants), Function discriminant analysis, Predictive learning via rule ensembles, Support Vector Machines, Artificial neural networks, Boosted/Bagged decision trees (BDT)…

ANN architecture: heuristic selection based on complexity adjustment and parameter estimation



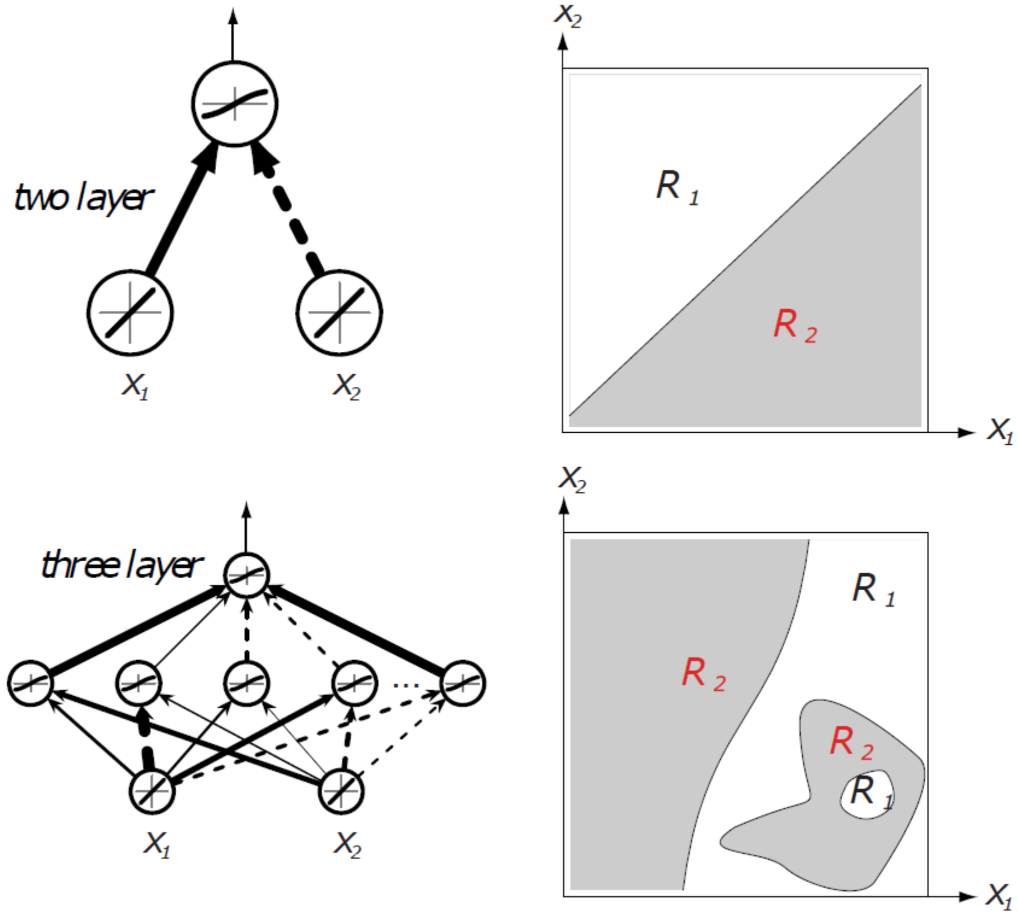


Theoretical basis:

Arnold - Kolmogorov (1957): if f is a multivariate continuous function, then f can be written as a finite composition of continuous functions of a single variable and the binary operation of addition

Gorban (1998): it is possible to obtain arbitrarily exact approx. of any continuous function of several variables using operations of summation and multiplication by number, superposition of functions, linear functions and one arbitrary continuous nonlinear function of one variable.

ANN architecture: heuristic selection based on complexity adjustment and parameter estimation



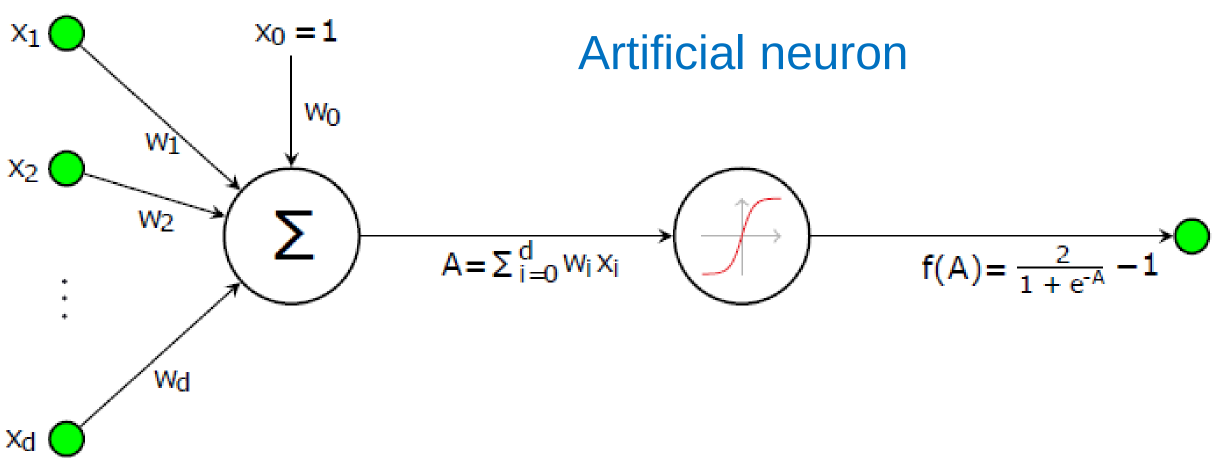
An example of a two and three-layer networks with two input nodes. Given an adequate number of hidden units, arbitrary nonlinear decision boundaries between regions R1 and R2 can be achieved

Theoretical basis:

Arnold - Kolmogorov (1957): if f is a multivariate continuous function, then f can be written as a finite composition of continuous functions of a single variable and the binary operation of addition

Gorban (1998): it is possible to obtain arbitrarily exact approx. of any continuous function of several variables using operations of summation and multiplication by number, superposition of functions, linear functions and one arbitrary continuous nonlinear function of one variable.

Neural Network is an universal approximator for any continuous function

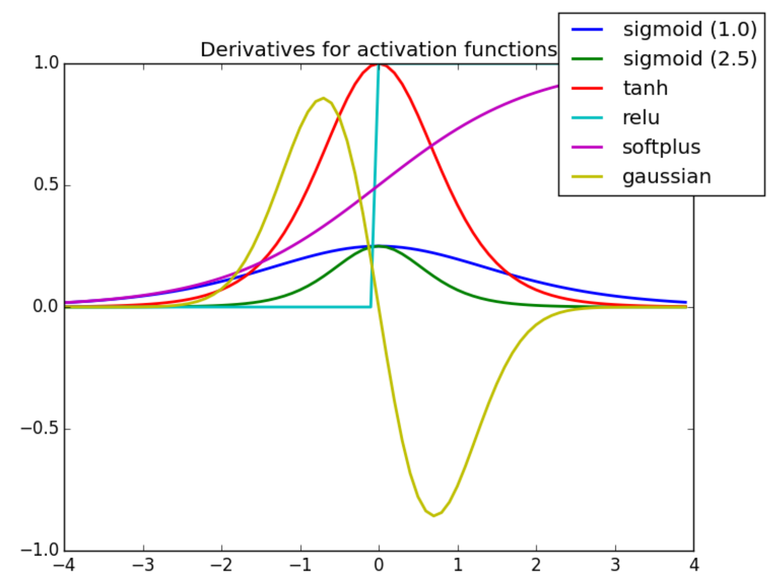


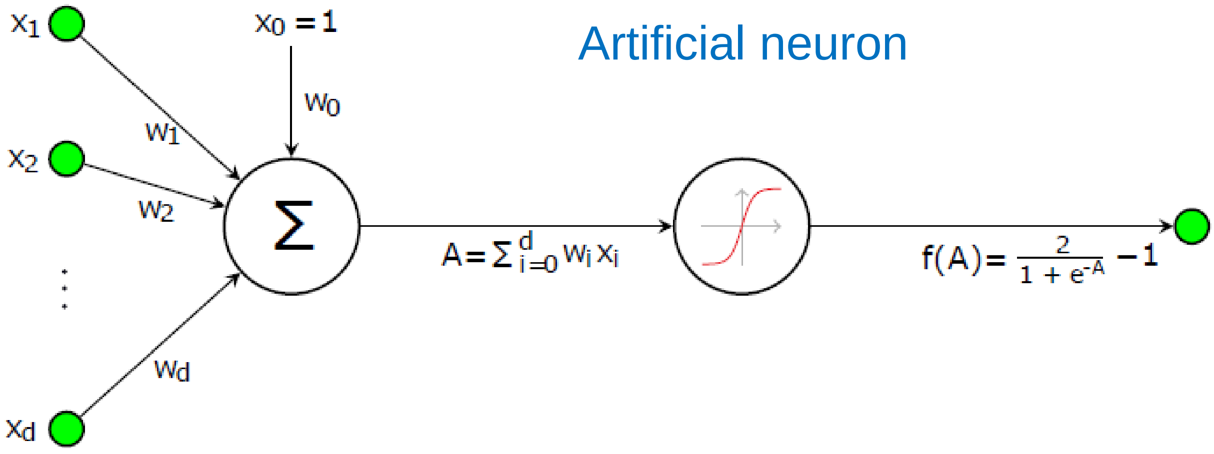
An ANN mimics the behaviour of the biological neuronal networks and consists of an interconnected group of processing elements (referred to as neurons or nodes) arranged in layers.

The first layer, known as the input layer, receives the input variables (x1; x2; …xd). Each connection to the neuron is characterised by a weight (w1; w2; … wd) which can be excitatory (positive weight) or inhibitory (negative weight). Moreover, each layer may have a bias (x0 = 1), which can provide a constant shift to the total neuronal input net activation (A), in this case a sigmoid function:

$$A = \sum_{i=1}^{d} w_i x_i + w_0 = \sum_{i=0}^{d} w_i x_i .$$

$$f(A) = \frac{2}{1+e^{-A}} - 1,$$





Artificial neuron

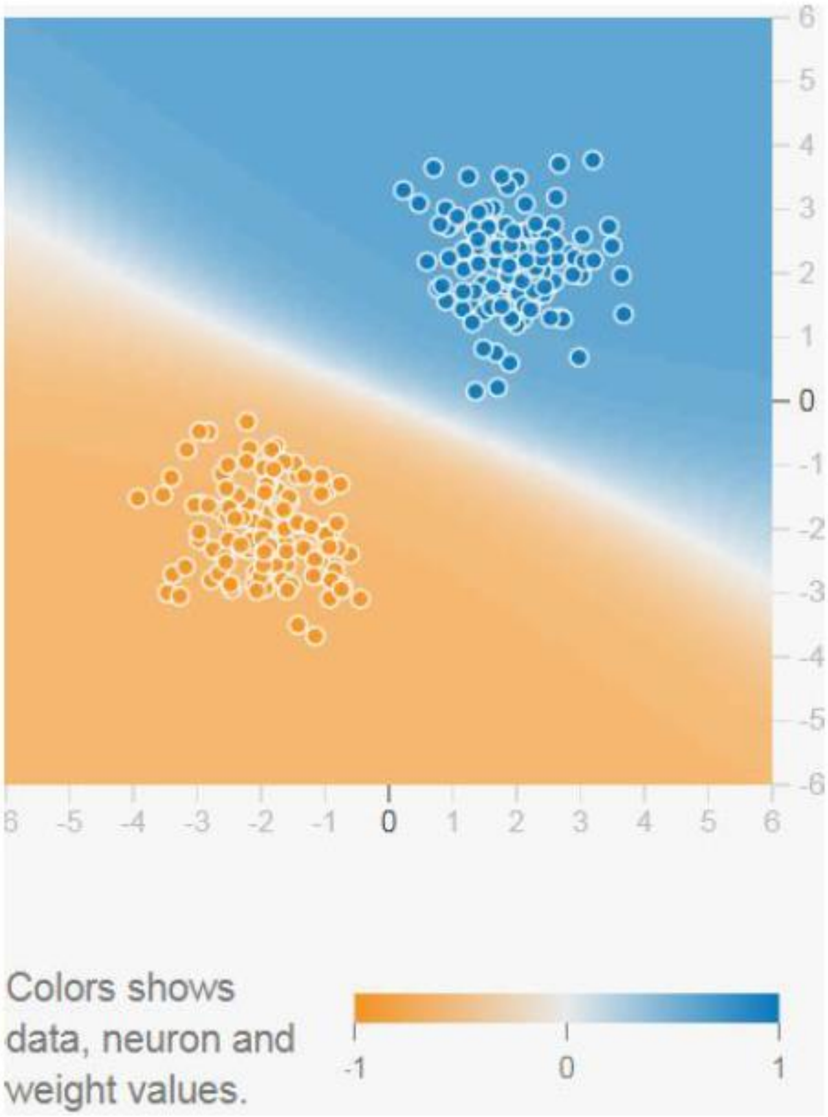
The last layer represents the final response of the ANN, which in the case of d input variables and nH nodes in the hidden layer can be expressed as:

$$o = f\left(\sum_{j=0}^{n_H} w_j f(\sum_{i=0}^{d} w_i x_i)\right)$$

The weights and thresholds are the network parameters, whose values are learned during the training phase by looping through the training data several hundreds of times. These parameters are determined by minimising an empirical loss function over all the events N in the training sample and adjusting the weights iteratively in the multidimensional space, such that the deviation E of the actual network output o from the desired (target) output y is minimal

$$E = \frac{1}{N}\sum_{\mu=1}^{N} \log\left(\frac{1}{2}(1 + y_\mu o_\mu + \epsilon)\right)$$

Decision boundaries with TensorFlow



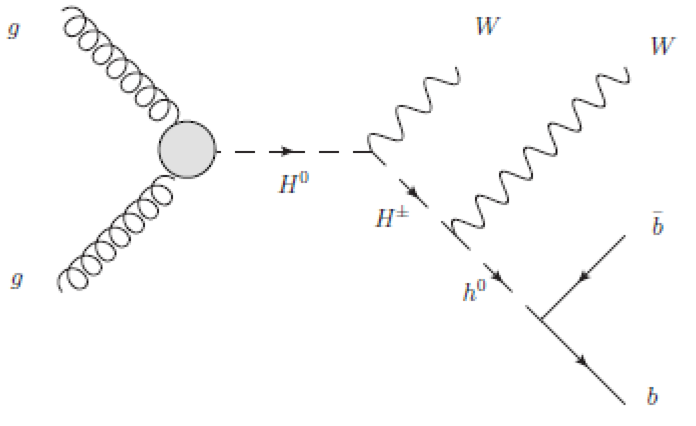
Orange shows negative values

Bue shows positive values

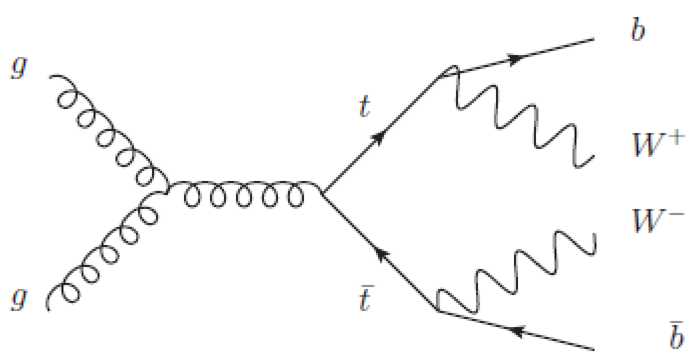
The data points (represented by small circles) are initially colored orange or blue, which correspond to positive one and negative one.

https://playground.tensorflow.org

Reproducing a Deep Learning result



(a)



nature COMMUNICATIONS

ARTICLE

Received 19 Feb 2014 | Accepted 4 Jun 2014 | Published 2 Jul 2014

DOI: 10.1038/ncomms5308

Searching for exotic particles in high-energy physics with deep learning

P. Baldi[1], P. Sadowski[1] & D. Whiteson[2]

https://www.nature.com/articles/ncomms5308

https://www.kaggle.com/lserkin/kaggle-mltutorial