

Title: Dilution (neural networks)

URL: [https://en.wikipedia.org/wiki/Dilution\\_\(neural\\_networks\)](https://en.wikipedia.org/wiki/Dilution_(neural_networks))

PageID: 47349395

Categories: Category:2012 in artificial intelligence, Category:Deep learning

Source: Wikipedia (CC BY-SA 4.0).

-----

Dropout and dilution (also called DropConnect [ 1 ] ) are regularization techniques for reducing overfitting in artificial neural networks by preventing complex co-adaptations on training data . They are an efficient way of performing model averaging with neural networks. [ 2 ] Dilution refers to randomly decreasing weights towards zero, [ 3 ] while dropout refers to randomly setting the outputs of hidden neurons to zero. Both are usually performed during the training process of a neural network, not during inference. [ 4 ] [ 5 ] [ 2 ]

#### Types and uses

Dilution is usually split in weak dilution and strong dilution . Weak dilution describes the process in which the finite fraction of removed connections is small, and strong dilution refers to when this fraction is large. There is no clear distinction on where the limit between strong and weak dilution is, and often the distinction is dependent on the precedent of a specific use-case and has implications for how to solve for exact solutions.

Sometimes dilution is used for adding damping noise to the inputs. In that case, weak dilution refers to adding a small amount of damping noise, while strong dilution refers to adding a greater amount of damping noise. Both can be rewritten as variants of weight dilution.

These techniques are also sometimes referred to as random pruning of weights, but this is usually a non-recurring one-way operation. The network is pruned, and then kept if it is an improvement over the previous model. Dilution and dropout both refer to an iterative process. The pruning of weights typically does not imply that the network continues learning, while in dilution/dropout, the network continues to learn after the technique is applied.

#### Generalized linear network

Output from a layer of linear nodes, in an artificial neural net can be described as

$y_i$  – output from node  $i$

$w_{ij}$  – real weight before dilution, also called the Hebb connection strength

$x_j$  – input from node  $j$

This can be written in vector notation as

$\mathbf{y}$  – output vector

$\mathbf{W}$  – weight matrix

$\mathbf{x}$  – input vector

Equations (1) and (2) are used in the subsequent sections.

#### Weak dilution

During weak dilution, the finite fraction of removed connections (the weights) is small, giving rise to a tiny uncertainty. This edge-case can be solved exactly with mean field theory . In weak dilution the impact on the weights can be described as

$\hat{w}_{ij}$  – diluted weight

$w_{ij}$  – real weight before dilution

$P(c)$  – the probability of  $c$  , the probability of keeping a weight

The interpretation of probability  $P(c)$  can also be changed from keeping a weight into pruning a weight.

In vector notation this can be written as

where the function  $g(\cdot)$  imposes the previous dilution.

In weak dilution only a small and fixed fraction of the weights are diluted. When the number of terms in the sum goes to infinite (the weights for each node) it is still infinite (the fraction is fixed), thus mean field theory can be applied. In the notation from Hertz et al. [3] this would be written as

$\langle h_i \rangle$  the mean field temperature

$c$  – a scaling factor for the temperature from the probability of keeping the weight

$w_{ij}$  – real weight before dilution, also called the Hebb connection strength

$\langle S_j \rangle$  – the mean stable equilibrium states

There are some assumptions for this to hold, which are not listed here. [6] [7]

Strong dilution

When the dilution is strong, the finite fraction of removed connections (the weights) is large, giving rise to a huge uncertainty.

Dropout

Dropout is a special case of the previous weight equation (3), where the aforementioned equation is adjusted to remove a whole row in the vector matrix, and not only random weights

$P(c)$  – the probability  $c$  to keep a row in the weight matrix

$\mathbf{w}_j$  – real row in the weight matrix before dropout

$\hat{\mathbf{w}}_j$  – diluted row in the weight matrix

Because dropout removes a whole row from the vector matrix, the previous (unlisted) assumptions for weak dilution and the use of mean field theory are not applicable.

The process by which the node is driven to zero, whether by setting the weights to zero, by “removing the node”, or by some other means, does not impact the end result and does not create a new and unique case. If the neural net is processed by a high-performance digital array-multiplicator, then it is likely more effective to drive the value to zero late in the process graph. If the net is processed by a constrained processor, perhaps even an analog neuromorphic processor, then it is likely a more power-efficient solution is to drive the value to zero early in the process graph.

Google's patent

Although there have been examples of randomly removing connections between neurons in a neural network to improve models, [3] this technique was first introduced with the name dropout by Geoffrey Hinton, et al. in 2012. [2] Google currently holds the patent for the dropout technique. [8] [note 1]

See also

AlexNet

Convolutional neural network § Dropout

Notes

References