<https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c>

<https://towardsdatascience.com/over-fitting-and-regularization-64d16100f45c>

<https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/>

[**https://medium.com/datadriveninvestor/l1-l2-regularization-7f1b4fe948f2**](https://medium.com/datadriveninvestor/l1-l2-regularization-7f1b4fe948f2)

[**https://www.quora.com/What-is-the-difference-between-L1-and-L2-regularization-How-does-it-solve-the-problem-of-overfitting-Which-regularizer-to-use-and-when**](https://www.quora.com/What-is-the-difference-between-L1-and-L2-regularization-How-does-it-solve-the-problem-of-overfitting-Which-regularizer-to-use-and-when)

[**https://www.quora.com/What-is-the-difference-between-L1-and-L2-regularization-How-does-it-solve-the-problem-of-overfitting-Which-regularizer-to-use-and-when/answer/Justin-Solomon**](https://www.quora.com/What-is-the-difference-between-L1-and-L2-regularization-How-does-it-solve-the-problem-of-overfitting-Which-regularizer-to-use-and-when/answer/Justin-Solomon)

**Different Regularization techniques in Deep Learning**

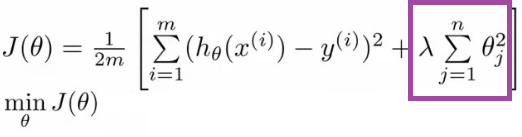
* **L2 and L1 regularization**
* **Dropout**
* **Data augmentation**
* **Early stopping**

In supervised machine learning, models are trained on a subset of data aka training data. The goal is to compute the target of each training example from the training data.

Now, overfitting happens when model learns signal as well as noise in the training data and wouldn’t perform well on new data on which model wasn’t trained on.

ways you can avoid overfitting your model on training data like cross-validation sampling, reducing number of features, pruning, regularization etc.

Regularization basically adds the penalty as model complexity increases. Regularization parameter (lambda) penalizes all the parameters except intercept so that model generalizes the data and won’t overfit.



Regularization in cost function

In above gif as the complexity is increasing, regularization will add the penalty for higher terms. This will decrease the importance given to higher terms and will bring the model towards less complex equation.

**in machine learning, you will have a fair idea that regularization penalizes the coefficients. In deep learning, it actually penalizes the weight matrices of the nodes.**

### L2 & L1 regularization

**L1 and L2 are the most common types of regularization. These update the general cost function by adding another term known as the regularization term.**

***Cost function = Loss (say, binary cross entropy) + Regularization term***

**Due to the addition of this regularization term, the values of weight matrices decrease because it assumes that a neural network with smaller weight matrices leads to simpler models. Therefore, it will also reduce overfitting to quite an extent.**

**However, this regularization term differs in L1 and L2.**

**In L2, we have:**

****

**Here, lambda is the regularization parameter. It is the hyperparameter whose value is optimized for better results. L2 regularization is also known as *weight decay* as it forces the weights to decay towards zero (but not exactly zero).**

**In L1, we have:**

****

**In this, we penalize the absolute value of the weights. Unlike L2, the weights may be reduced to zero here. Hence, it is very useful when we are trying to compress our model. Otherwise, we usually prefer L2 over it.**

**The key difference between these techniques is that Lasso shrinks the less important feature’s coefficient to zero thus, removing some feature altogether. So, this works well for feature selection in case we have a huge number of features.**

the two vectors are equivalent with respect to the L1 norm but different with respect to the L2 norm. This is because squaring a number punishes large values more than it punishes small values.

Thus, solving the minimization problem above with ∥x∥2 (so-called "Tikhonov regularization") really wants small values in all slots of x, whereas solving the L1 version doesn't care if it puts all the large values into a single slot of