

# NEURAL NETWORKS AND DEEP LEARNING CST 395 CS 5TH SEMESTER HONORS COURSE- Dr Binu V P, 9847390760

CS 5th Semester Honors course for the Computer Science at KTU- Dr Binu V P

## Hyperparameters and Validation Sets



September 30, 2022

Most machine learning algorithms have several settings that we can use to control the behavior of the learning algorithm. These settings are called hyperparameters. The values of hyperparameters are not adapted by the learning algorithm itself (though we can design a nested learning procedure where one learning algorithm learns the best hyperparameters for another learning algorithm).

In the polynomial regression example we saw early, there is a single hyperparameter: the degree of the polynomial, which acts as a capacity hyperparameter. The  $\lambda$  value used to control the strength of weight decay is another example of a hyperparameter.

Sometimes a setting is chosen to be a hyper parameter that the learning algorithm does not learn because it is difficult to optimize. More frequently, the setting must be a hyper parameter because it is not appropriate to learn that hyper parameter on the training set. This applies to all hyper parameters that control model capacity. If learned on the training set, such hyper parameters would always choose the maximum possible model capacity, resulting in overfitting . For example, we can always fit the training set better with a higher degree polynomial and a weight decay setting of  $\lambda = 0$  than we could with a lower degree polynomial and a positive weight decay setting.

To solve this problem, we need a **validation set** of examples that the training algorithm does not observe.

Earlier we discussed how a held-out test set, composed of examples coming from the same distribution as the training set, can be used to estimate the generalization error of a learner, after the learning process has completed. It is important that the test examples are not used in any way to make choices about the model, including its hyperparameters. For this reason, no example from the test set can be used in the validation set. Therefore, we always construct the validation set from the training data.

Specifically, we split the training data into two disjoint subsets. One of these subsets is used to learn the parameters. The other subset is our validation set, used to estimate the generalization error during or after training, allowing for the hyperparameters to be updated accordingly.

The subset of data used to learn the parameters is still typically called the training set, even though this may be confused with the larger pool of data used for the entire training process. The subset of data used to guide the selection of hyperparameters is called the validation set.

**Typically, one uses about 80% of the training data for training and 20% for validation.**

Since the validation set is used to “train” the hyperparameters, the validation set error will underestimate the generalization error, though typically by a smaller amount than the training error. After all hyperparameter optimization is complete, the generalization error may be estimated using the test set.

In practice, when the same test set has been used repeatedly to evaluate performance of different algorithms over many years, and especially if we consider all the attempts from the scientific community at beating the reported state-of-the-art performance on that test set, we end up having optimistic evaluations with the test set as well. Benchmarks can thus become stale and then do not reflect the true field performance of a trained system. Thankfully, the community tends to move on to new (and usually more ambitious and larger) benchmark datasets.

### Cross-Validation

Dividing the dataset into a fixed training set and a fixed test set can be problematic if it results in the test set being small. A small test set implies statistical uncertainty around the estimated average test error, making it difficult to claim that algorithm A works better than algorithm B on the given task.

When the dataset has hundreds of thousands of examples or more, this is not a serious issue. When the dataset is too small, alternative procedures enable one to use all of the examples in the estimation of the mean test error, at the price of increased computational cost. These procedures are based on the idea of repeating the training and testing computation on different randomly chosen subsets or splits of the original dataset. The most common of these is the **k-fold cross-validation** procedure, shown in algorithm below, in which a partition of the dataset is formed by splitting it into  $k$  non-overlapping subsets. The test error may then be estimated by taking the average test error across  $k$  trials. On trial  $i$ , the  $i$ -th subset of the data is used as the test set and the rest of the data is used as the training set. One problem is that there exist no unbiased estimators of the variance of such average error estimators ([Bengio and Grandvalet](#),

2004), but approximations are typically used.

---

**Define**  $\text{KFoldXV}(\mathbb{D}, A, L, k)$ :

**Require:**  $\mathbb{D}$ , the given dataset, with elements  $\mathbf{z}^{(i)}$

**Require:**  $A$ , the learning algorithm, seen as a function that takes a dataset as input and outputs a learned function

**Require:**  $L$ , the loss function, seen as a function from a learned function  $f$  and an example  $\mathbf{z}^{(i)} \in \mathbb{D}$  to a scalar  $\in \mathbb{R}$

**Require:**  $k$ , the number of folds

Split  $\mathbb{D}$  into  $k$  mutually exclusive subsets  $\mathbb{D}_i$ , whose union is  $\mathbb{D}$ .

**for**  $i$  from 1 to  $k$  **do**

$f_i = A(\mathbb{D} \setminus \mathbb{D}_i)$

**for**  $\mathbf{z}^{(j)}$  in  $\mathbb{D}_i$  **do**

$e_j = L(f_i, \mathbf{z}^{(j)})$

**end for**

**end for**

**Return**  $e$

---

The  $k$ -fold cross-validation algorithm: It can be used to estimate generalization error of a learning algorithm  $A$  when the given dataset  $D$  is too small for a simple train/test or train/valid split to yield accurate estimation of generalization error, because the mean of a loss  $L$  on a small test set may have too high variance. The dataset  $D$  contains as elements the abstract examples  $\mathbf{z}^{(i)}$  (for the  $i$ -th example), which could stand for an (input,target) pair  $\mathbf{z}^{(i)} = (x^{(i)}, y^{(i)})$  in the case of supervised learning, or for just an input  $\mathbf{z}^{(i)} = (x^{(i)})$  in the case of unsupervised learning.

The algorithm returns the vector of errors  $e$  for each example in  $D$  whose mean is the estimated generalization error. The errors on individual examples can be used to compute a confidence interval around the mean. While these confidence intervals are not well-justified after the use of cross-validation, it is still common practice to use them to declare that algorithm  $A$  is better than algorithm  $B$  only if the confidence interval of the error of algorithm A lies below and does not intersect the confidence interval of algorithm B.



### Popular posts from this blog

## NEURAL NETWORKS AND DEEP LEARNING CST 395 CS 5TH SEMESTER HONORS COURSE NOTES - Dr Binu V P, 9847390760

*October 03, 2022*

About Me Syllabus Question Paper Dec 2022 Module 1 ( Basics of Machine Learning)  
Overview of Machine Learning Machine Learning Algorithm Linear Regression Capacity,  
Overfitting and Underfitting Regularization Hyperparameters and Validation ...

[READ MORE](#)

---

## Machine Learning Algorithm

*October 01, 2022*

A machine learning algorithm is an algorithm that is able to learn from data. But what do we mean by learning? Mitchell (1997) provides the definition "A computer program is said to learn from experience E with respect to some class of tasks T and ...

[READ MORE](#)

---

## Syllabus

*October 01, 2022*

Syllabus Module - 1 (Basics of Machine Learning ) Machine Learning basics - Learning algorithms - Supervised, Unsupervised, Reinforcement, overfitting, Underfitting, Hyper parameters and Validation sets, Estimators -Bias and Variance. Challenges ...

[READ MORE](#)

 [Powered by Blogger](#)

Theme images by [Michael Elkan](#)





**DR.BINU V P-9847390760**

Associate Professor and Head Dept of  
Computer Science-IHRD

Karunagappally. Love to share the  
thoughts and views .I play lot of Games  
and basically an Athlet in my school and  
college days.I never keep my studies as  
a second option.Stood first In MTech  
and BTech.I did my resear

...

[VISIT PROFILE](#)

**Archive**



[Report Abuse](#)