ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

On overfitting of multilayer perceptrons for classification

Joseph Rynkiewicz [1]

Université Paris I - SAMM
90 rue de tolbiac, Paris - France

Abstract. In this paper, we consider classification models involving multilayer perceptrons (MLP) with rectified linear (ReLU) functions for activation units. It is a difficult task to study the statistical properties of such models. The main reason is that in practice these models may be heavily overparameterized. We study the asymptotic behavior of the difference between the loss function of estimated models and the loss function of the theoretical best model. These theoretical results give us information on the overfitting properties of such models. Some simulations illustrate our theoretical finding and raise new questions.

1 Introduction

Feed-forward neural networks or are well known and popular tools (see Lecun et al. [2]). However, few theoretical results are available about such models, and we propose in this paper to study the overfitting of one hidden layer MLP with ReLU activation functions for classification purpose. We establish an asymptotic result which generalizes the result obtained in the regression framework (see Rynkiewicz [5]). Moreover, these results are complementary to some recent non-asymptotic results (see Neyshabur et al. [3]) where increasing the number of hidden unit may reduce the overfitting. This paper is organized as follows: Firstly, under suitable conditions, we provide the asymptotic distribution for the likelihood ratio test statistic of the estimated model and the best one. This result is general because it holds even if the model is not identifiable which is the case for over-parameterized MLP. Then we study on some simulations the overtraining of MLP, and we compare the asymptotic and non-asymptotic framework. Our results shed new light on the overfitting behavior of MLP models.

2 The model

Let x be an observation in $\mathbb{R}^d$, and let us write the d-dimensional vector of weights $w_{ki} := (w_{ki1}, \cdots, w_{kid})^T$, then an MLP function with H hidden units and K outputs can be written:

$$f_\theta(x) = (f_{\theta 1}(x), \cdots, f_{\theta K}(x))^T = \left(\beta_k + \sum_{i=1}^{H} a_{ki}\phi\left(b_{ki} + {w_{ki}}^T x\right)\right)_{1\leq k\leq K}$$

with

$$\theta = (\beta_1, \cdots, \beta_K, a_{11}, \cdots, a_{KH}, b_{11}, \cdots, b_{KH}, w_{11}, \cdots, w_{KH}) \in \mathbb{R}^{(2H+1+H\times d)\times K}$$

the parameter vector of the model. Let us denote Θ the set of possible parameters. The transfer function ϕ will be assumed to be a ReLU function: $\phi(z) = \max(0, z)$ for $z \in \mathbb{R}$. We observe a random sample of independent and identically distributed random vectors: $(X_1, Y_1), \cdots, (X_n, Y_n)$, from the distribution P of a vector (X, Y), with Y a categorical random variable, $Y \in \{1, \cdots, K\}$. The classification model can be written as:

$$P(Y = k|X) = \frac{\exp(f_k^0(X))}{\sum_{l=1}^{K} \exp(f_l^0(X))} \tag{1}$$

where

$$f^0 = \left(f_1^0, \cdots, f_K^0\right)^T = \left(\beta_k^0 + \sum_{i=1}^{H^0} a_{ki}^0 \phi\left(b_{ki}^0 + {w_{ki}^0}^T x\right)\right)_{1 \leq k \leq K} \tag{2}$$

is the best classification function. Since the variable Y is categorical the function f^0 exists and is unique. We assume that a minimal MLP, with H^0 number of hidden units, exists and realizes the best function f^0. Let us write $\mathbf{1}_k(y)$ the indicator function of the class k: $\mathbf{1}_k(y) = 1$ if $y = k$, and $\mathbf{1}_k(y) = 0$ if $y \neq k$, then

$$f^0 = \arg\min_{\theta \in \Theta} E\left(-\sum_{k=1}^{K} \mathbf{1}_k(Y) \log\left(\frac{\exp(f_{\theta k}(X))}{\sum_{l=1}^{K} \exp(f_{\theta l}(X))}\right)\right),$$

where the expectation $E\left(f(X, Y)\right) = \int f(x, y) dP(x, y))$ is taken under the law P of (X, Y). Hence, the function f^0 minimizes the theoretical negative log-likelihood or the multiclass cross entropy. Let us write Θ_0 the set of parameters realizing the best function f^0: $\forall \theta \in \Theta_0$, $f_\theta = f^0$. Note that we do not assume that Θ_0 is a finite set which means that loss of identifiability can occur, this is the case if the MLP has redundant hidden units (see Fukumizu [1] or Rynkiewicz [4]). The conditional probability function for a parameter θ will be:

$$g_\theta(x, y) = \sum_{k=1}^{K} \mathbf{1}_k(y) \log\left(\frac{\exp(f_{\theta k}(x))}{\sum_{l=1}^{K} \exp(f_{\theta l}(x))}\right). \tag{3}$$

For an observed sample $(x_1, y_1), \cdots, (x_n, y_n)$, a natural estimator of f^0 is the Maximum Likelihood Estimator (MLE) $f_{\hat{\theta}}$ that minimizes the negative log-likelihood:

$$f_{\hat{\theta}} = \arg\min_{\theta \in \Theta} -\sum_{i=1}^{n} g_\theta(x_i, y_i) \tag{4}$$

Let us introduce some assumptions:

H-1: Let $f_\theta, \theta \in \Theta$ be MLP functions with H hidden units. We assume that Θ is a closed and bounded set and the set of parameters Θ_0, realizing the best function f^0 is assumed to be a subset of the interior of Θ.

H-2: The explicative random vector X admits a strictly positive density with respect to the Lebesgue measure of $\mathbb{R}^d$ and $P(|X|^2) < \infty$.

Under **H-1** $f_{\hat{\theta}}$ converges to the function f^0: $f_{\hat{\theta}} \xrightarrow{a.s.} f^0$. To assess the asymptotic behavior of this convergence we can study the difference between the log-likelihood of $f_{\hat{\theta}}$ and the log-likehood for the best function f^0, this is done in the next section.

3 The likelihood ratio test statistic (LRTS)

Let us denote by

$$g_0(x, y) = \sum_{k=1}^{K} \mathbf{1}_k(y_i) \log\left(\frac{\exp(f_k^0(x))}{\sum_{l=1}^{K} \exp(f_l^0(x))} \right) \tag{5}$$

the true conditional probability function, the likelihood ratio test statistic (LRTS) will be:

$$\sum_{i=1}^{n} g_{\hat{\theta}}(x_i, y_i) - \sum_{i=1}^{n} g_0(x_i, y_i). \tag{6}$$

3.1 Asymptotic behavior of the LRTS

Under assumptions **H-1** and **H-2**, following the same line than Rynkiewicz [4], [5], we can then prove the following theorem:

Theorem 3.1 *Let the map $\Omega : \mathcal{L}^2(P) \to \mathcal{L}^2(P)$ be defined as $\Omega(f) = \frac{f}{\|f\|_2}$, and $\mathbb{I}_{\mathbb{R}^+}$ be the indicator function of $\mathbb{R}^+$. Under the assumptions* **H-1** *and* **H-2***, a centered Gaussian process $\{W(s), s \in \mathcal{S}\}$ with continuous sample paths and a covariance kernel $P\left(W(s_1)W(s_2)\right) = P\left(s_1 s_2\right)$ exists so that*

$$\lim_{n\to\infty} \sup_{\theta\in\Theta} \sum_{i=1}^{n} g_\theta(X_i, Y_i) - \sum_{i=1}^{n} g_0(X_i, Y_i) = \sup_{s\in\mathcal{S}} \left(\max\left\{W(s); 0\right\}\right)^2.$$

The index set $\mathcal{S}$ is defined as $\mathcal{S} = \cup_t \mathcal{S}_t$, the union runs over any possible integers $H^0 \leq t \leq H$ with

$$\begin{aligned}&\mathcal{S}_t = \Big\{\Omega\left(\textstyle\sum_{k=1}^{K}\left(\gamma_k + \sum_{i=0}^{H^0} \mathbb{I}_{\mathbb{R}^+}({w_{ki}^0}^T X + b_{ki}^0)({\zeta_{ki}}^T X + \alpha_{ki})\right.\right.\\ &\quad \left.\left.+ \textstyle\sum_{i=t}^{H} \mu_{ki}\phi({w_{ki}}^T X + b_{ki})\right)\right),\\ &\gamma_1, \cdots, \gamma_K, \alpha_{11}, \cdots, \alpha_{KH^0} \in \mathbb{R}, \mu_{1t}, \cdots, \mu_{KH} \in \mathbb{R}^+; \zeta_{11}, \cdots, \zeta_{KH^0} \in \mathbb{R}^d,\\ &(w_{1t}, b_{1t}), \cdots, (w_{KH}, b_{KH}) \in \Theta \backslash \left\{(w_{11}^0, b_{11}^0), \cdots, (w_{KH^0}^0, b_{KH^0}^0)\right\}\Big\}.\end{aligned}$$

This theorem shows that the degree of overfitting is bounded in probability, but depends on the size of the asymptotic set $\mathcal{S}$. Intuitively the set $\mathcal{S}$ is the degree of freedom of the estimated model when it is very near to the best function. The set $\mathcal{S}$ depends on the difference of size between the model in use and the true one. Since the dimension of the true one is fixed, the overfitting depends on the following hyperparameters of the estimated model: the number of hidden neurons, and the size of the parameters. The bigger the hyperparameters, the bigger the set $\mathcal{S}$. Hence, to limit the size of $\mathcal{S}$, we can reduce any of this hyperparameter, but they have to be large enough so that the best function f^0 belongs to the set of possible parameters. We will investigate the meaning of this theorem on simulations in the next section.

4 An empirical investigation

In this section, we assess the effect of overparameterization on the training set and the influence of the form of the best classification function on it. Since, in practice, the data are often high dimensional, we chose to simulate inputs of size 200. Let us write $0_{\mathbb{R}^{200}}$ the null vector of $\mathbb{R}^{200}$ and $I_{\mathbb{R}^{200}}$ the identity matrix of $\mathbb{R}^{200} \times \mathbb{R}^{200}$. We trained fully connected feedforward networks with one hidden layer and ReLU transfer functions on two sets of data:

1. For the first set, the input X_t is a Gaussian random vector of size 200, with each component centered, normalized and independent from each other: $X_t \sim \mathcal{N}(0_{\mathbb{R}^{200}}, I_{\mathbb{R}^{200}})$. The output Y_t is a Bernoulli variable: $Y_t \sim b(0.5)$, independent of X_t. We simulate a sample of independent vectors (X_t, Y_t) of length 100000: $(X_t, Y_t)_{1\leq t\leq 100000}$. In this case, the best classification function is $f^0(x) = (0, 0)$.
2. For the second set, we first create an MLP function M^0 with two outputs, 200 as input size, one hidden layers of size 8 neurons and randomly chosen weights between -0.5 and 0.5. The input X_t is a Gaussian random vector of size 200, with each component centered, normalized and independent from each other: $X_t \sim \mathcal{N}(0_{\mathbb{R}^{200}}, I_{\mathbb{R}^{200}})$. The output of the network is $M^0(x) = \left(M_1^0(x), M_2^0(x)\right)$ so the random variable Y_t is a Bernoulli variable with parameter $\frac{\exp(M_2^0(X_t))}{\exp(M_1^0(X_t))+\exp(M_2^0(X_t))}$. We simulate a sample of independent vectors (X_t, Y_t) of length 100000: $(X_t, Y_t)_{1\leq t\leq 100000}$. In this case, the best classification function is the MLP function: $f^0(x) = M^0(x)$.

Moreover, for each set of data, we simulate 100000 supplementary data for using it to assess the models on a test set.

Training the models On both sets, we trained 8 architectures with one hidden layer from 2^3 to 2^{10} hidden units, each time increasing the number of hidden units of each layer by factor 2. For the small architectures, 2^3 to 2^6 hidden units, the amount of data is much greater than the number of parameters of the MLP; it is the asymptotic framework where our results apply. For the great architectures, 2^9 to 2^{10} hidden units, the number of parameters of the MLP is greater than the amount of data; it is the not-asymptotic framework, where our results do not apply, but we did it for comparing with the experiments of Neyshabur et al. [3]. Note that, since we generated the second data set with the best function f^0 using the smallest architecture, all the MLP of the second experiment can realize the best function.

For each experiment, we trained the network using Stochastic Gradient Descent (SGD) with mini-batch size 64, momentum 0.9 and fixed step size 0.01. We did not use any technic of regularization. We stopped the training when the number of epochs reached 1000. All the computations are done with Torch7 using a GPU.

Evaluations For the trained architectures we give the number of hidden units of each hidden layer, the corresponding number of parameters, the mean LRTS on the training data set and the mean LRTS on the test data set:

$$\frac{1}{n}\left(\sum_{i=1}^{n} g_{\hat{\theta}}(x_i, y_i) - \sum_{i=1}^{n} g_0(x_i, y_i)\right). \tag{7}$$

Note that, since the amount of data is large (100000) the mean LRTS on the test set will be very near to the expectation of the LRTS, which is called the Kullback-Leibler divergence:

$$E\left(g_{\hat{\theta}}(X, Y) - g_0(X, Y)\right) := K(g_0, g_{\hat{\theta}}). \tag{8}$$

We summarize the results for the two data sets on table 1. As expected by our results the training error depends not only of the architecture of the MLP but also of the best function f^0. Indeed, for all models, for the same number of parameters and data, the training overfitting is smaller when the best function is more complicated. However, we can see that the relationship between the

Table 1: Comparison of overtraining in function of architectures and data sets

Nb of hidden units	Nb of parameters	LRTS	Data set $f^0(x) = 0$	Data set $f^0(x) = M_0(x)$
2^3	1626	training	0.015146	0.008396
		test	0.014629	0.020987
2^4	3250	training	0.029139	0.024180
		test	0.033930	0.040766
2^5	6498	training	0.061729	0.054064
		test	0.073892	0.090174
2^6	12994	training	0.126840	0.114709
		test	0.186349	0.294475
2^7	25986	training	0.278007	0.244434
		test	0.739622	1.682036
2^8	51970	training	0.688570	0.250662
		test	7.874922	1.431996
2^9	103948	training	0.692167	0.250892
		test	3.711566	0.993170
2^{10}	207874	training	0.692419	0.250924
		test	2.704526	0.873004

test error and the training error also depends on this best function and even if the learning overfitting is smaller for the second data set, the test error may be more significant in this case. This fact, unexpected, remains to explain. Finally, we can see that when the number of parameters becomes greater than the amount of data, the test error seems to decrease when the amount of parameters increases, as in Neyshabur et al. [3]. However, the test error of the biggest model doesn't reach the test error of the smallest and less overparameterized model. Surprisingly, the behavior of the asymptotic case seems the inverse of the behavior of the not-asymptotic case.

5 Conclusion

The asymptotic overfitting of MLP functions depends on the size of set $\mathcal{S}$ which is a function of the difference between the complexity of the MLP function in use and the complexity of the best classification function f^0 (the number of redundant hidden units). This fact explains the apparent contradiction noticed by some authors (cf Zhang et al. [6]), where an MLP does not overfit too much for a complex task but overfits a lot if you randomize the output data. Moreover, in the experiments, we have seen that the relationship between the overtraining on the learning set and the error on the test set is not obvious and seems also depends on the best function f^0. Finally, the behavior of the overfitting also relies on the comparison of the amount of data and number of parameter of the model and seems different in the asymptotic or not-asymptotic framework. It will be interesting to understand these surprising facts which we leave for future work.

References

[1] Fukumizu, K., Likelihood ratio of unidentifiable models and multilayer neural networks, Ann. Statist. 31 (2003) 833-851.

[2] LeCun, Yann, Bengio, Y. and Hinton, G., Deep learning, Nature. 521 (7553) (2015) 436-444.

[3] Neyshabur, B., Li, Z., Bhojanapalli, S. LeCun, Y., Srebro, N. Towards Understanding the Role of Over-Parametrization in Generalization of Neural Networks. arXiv preprint arXiv:1805.12076, 2018.

[4] Rynkiewicz, J., Asymptotics for Regression Models Under Loss of Identifiability, Sankhya A, 78 (2) (2016) 155-179.

[5] Rynkiewicz, J., Asymptotic statistics for multilayer perceptron with ReLu hidden units, ESANN 2018, p. 491-496.

[6] Zhang, C. Bengio, S. Hardt, M. Recht, B. and Vinyals, O., Understanding deep learning requires rethinking generalization. ICLR 2017 (2017).