Deep Convolutional Neural Network for Survival Estimation of Amyotrophic Lateral Sclerosis patients

Enrico Grisan[1 2], Alessandro Zandonà[1] and Barbara Di Camillo[1]
the Pooled Resource Open-Access ALS Clinical Trials Consortium *

1- Department of Information Engineering - University of Padova
Via Gradenigo 6/b, Padova - Italy

2- School of Imaging Sciences and Biomedical Engineering - King's College London
Lambeth Wing, St. Thomas Hospital, London - UK

Abstract. We propose a convolutional neural network (CNN) coupled with a fully connected top layer for survival estimation. We design an objective function to directly estimate the probability of survival at discrete time intervals, conditional to the patient not having incurred any adverse event at previous time points. We test our CNN and objective function on a large dataset of longitudinal data of patients with Amyotrophic Lateral Sclerosis (ALS). We compare our CNN and the objective function against other neural networks designed for survival analysis, and against the optimization of Cox-partial-likelihood or a simple logistic classifier. The use of our objective function outperforms both Cox-partial-likelihood and logistic classifier, independently of the network architecture, and our deep CNN provides the best results in terms of AU-ROC, accuracy and mean absolute error.

1 Introduction

Survival analysis models the probability that certain event of interest (e.g. death) occurs after a specific time interval during the follow ups of patient. Its importance is primarily linked to the need of estimating the effect of different conditions (therapy scheme or others) on the time of occurrence of a specific event (e.g. death, recurrence) when all other subject specific characteristics are factored out. Traditional approaches to survival analysis, such as the Cox proportional hazards (CPH) model [1], assume that the risk of the event of interest of a patient can be obtained as a linear combination of the patient's characteristics. The advent and development of deep learning approaches, together with a formulation of the Cox hazard function in terms of partial likelihood [2] has opened the possibility of a further performance increase in the estimation

*Data used in the preparation of this article were obtained from the Pooled Resource Open-Access ALS Clinical Trials (PRO-ACT) Database. As such, the following organizations and individuals within the PRO-ACT Consortium contributed to the design and implementation of the PRO-ACT Database and/or provided data, but did not participate in the analysis of the data or the writing of this report: Neurological Clinical Research Institute (MGH), Northeast ALS Consortium, Novartis, Prize4Life, Regeneron Pharmaceuticals, Inc., Sanofi, Teva Pharmaceutical Industries, Ltd.

of the relative risk of patients [3, 4], by using a nonlinear combination of covariates to drive the patients' risk, outperforming standard linear CPH model. The key idea is to formulate an objective function that is able to combine the deep (nonlinear) features and to preserve the ordering between patient score and time of events: similarly to classical proportional hazard regression, the patients who are experiencing the event of interest first should be ideally those with the highest score. However the probability of an event at a specific time is still estimated post-hoc by modulating the population baseline cumulative survival function. In this work, we developed a deep-learning architecture to directly estimate the probability of an event at (pre)defined time points, removing the need to estimate the relative risk as intermediate step. In detail, we proposed a convolutional neural network (CNN) coupled with a fully connected layer as input, in order to exploit all possible relationships among the patients' features, which might be unordered and unstructured. The utilization of CNNs greatly reduces the number of parameters to be estimated with respect to those required by fully connected networks as those proposed in [2, 3], allowing an increase in the number of hidden layers.

2 Proportional hazard methods

Proportional hazard models for risk estimation assume that any subject has an hazard function whose ratio with respect to a baseline population hazard is a constant proportion depending on the patient specific characteristics (covariates) x_j. Being $h_j(t)$ the hazard function for the j^{th} subject, the proportional model assumes that it is linked to the baseline (population) hazard function $h_0(t)$ only through a scale factor: $h_j(t) = e^{f_\theta(x_j)} h_0(t)$ where $f_\theta(x_j)$ is a generic parametric function of the covariates x_j and of the model parameters θ. Cox regression [1] assumes a linear function $f_\theta(x_j) = \theta \cdot x_j$. Under the assumption that no censoring and no tied event times exist in the observed subjects, the regression parameters θ can be estimated maximizing the partial likelihood $L_p(\theta)$ of the events over the entire set of patients, that is maximizing the joint probability of the ordering of patients' events instead of the joint probability of the actual time of events:

$$\hat{\theta} = \underset{\theta}{max} L_p(\theta) = \underset{\theta}{max} \prod_{t_i} \frac{e^{\theta \cdot x_j}}{\sum_{j \in R_i} e^{\theta \cdot x_j}} \tag{1}$$

Where R_i is the set of patients still at risk of death for any time t larger than the time of event T_i of the i^{th} subject: $R_i = \{j : T_j > T_i\}$. Once the risk function $\hat{f}_\theta(x)$ has been estimated, the baseline cumulative hazard function can be obtained with different methods [1, 6, 7]; we used Breslow's method [8] throughout the paper:

$$\hat{H}_0(t) = \sum_{t_i < t} \hat{h}_0(t) = \sum_{t_i < t} \frac{1}{\sum_{j \in R_i} e^{\hat{f}_\theta(x_j)}} \tag{2}$$

from which the baseline survival function becomes $\hat{S}_0(t) = e^{-\hat{H}_0(t)}$ and the probability of survival of the patient j at time t is:

$$\hat{S}_j(t) = \hat{S}_j(t|x_j) = P[t > t_i|x_j] = \hat{S}_0(t)^{exp(\hat{f}_\theta(x_j))} \tag{3}$$

In the case in which multiple patients die at a specific time, or in which we are interested in predicting the survival at discrete time [9, 10], binning all events in time intervals, the probability $\pi_{ij} = P[t < t_{i+1}|t \geq t_i, x_j]$ of the patient j to die in the interval $[t_i, t_{i+1})$ given that he survived up to time t_i is given by:

$$log(-log(1-\pi_{ij})) = \hat{h}_\theta(x_j) + log(log(\hat{S}_0(t_{i+1})) - log(\hat{S}_0(t_i))) = \hat{f}_\theta(x_j) + c_i \tag{4}$$

3 Fixed-interval survival estimation

At variance with the hazard estimation problem, where the aim is to estimate how much a patient survival probability deviates from the baseline (population) survival function, we cast the problem of estimating the survival probability of a patient within a time interval as a multinomial classification.In particular, given the time point t_i, and the corresponding time T_j of death of patient j, we define K disjoint time intervals $[t_i, t_{i+1})), [t_{i+1}, t_{i+2}), ...[t_{i+K-1}, t_{i+K})$ over which we want to estimate the survival probability $d_j(k)$ for the k^{th} interval:

$$d_j(k) = 1 - P_j[t > t_{i+k}|x_j] = \begin{cases} 0 & T_j < t_{i+k} \\ 1 & T_j \geq t_{i+k} \end{cases} \tag{5}$$

In the case of right-censored data, the formulation accommodates naturally for the situation defining the survival probabilities of the patient as 1 for each time interval.

3.1 Deep-learning architecture

In this work, we proposed a deep-learning architecture ConvSurv for survival estimation based on a convolutional network [11, 12] in which each neuron can be interpreted as a filter. This allows on one side to have shared filters across the network, so that each neuron is computing a *feature map* when it is applied to its input signal, on the other side it allows for keeping the number of weights at a reasonable level when they have to be estimated in very deep architecture, and finally allows to exploit the possible presence of *local structure* in the data. In order to exploit the advantages of convolutional networks even in data not showing an apparent and obvious structure, such as those coming from clinical, demographic and laboratory, we inserted on top of the convolutional architecture a fully connected layer mapping through a linear combination the input data $X \in \mathbb{R}^{1\times M}$ into a vector of the same dimension $X' \in \mathbb{R}^{1\times M}$. The element at position m of the resulting feature vector $X'(m) = w_m X^T$ with $w_m \in \mathbb{R}^{1\times M}$ is a vector of coefficients: in the extreme case where the weights w_m from the input to each node $m = 1, ..., M$ are such that $||w_m||_0 = 1$ for every m, is simply

a shuffling of the input vector. This layer allows to find the combination of measurements that are better exploited by the subsequent filters, and to have the deeper filters to (possibly) have access to evet available input features, even if it is outside the receptive field of the filter.

3.2 Objective functions

In this work, two objective functions have been considered for the deep-learning architectures to estimate survival at specific time intervals. The first is a classical logistic loss, the second is our proposed derivation of a conditional probability quasi-logistic loss, explicitly translating a fixed-interval survival model into a loss function.

For the j^{th} patient the output of the deep-neural network would be a K-dimensional vector of probabilities $f_\theta(x_j) \in [0,1]^{1\times K}$ so that the logistic loss over all intervals and over all patients becomes:

$$l(\theta) = -\sum_{jk} w_{jk}\Big(log\Big(\frac{1}{1+e^{-f_\theta(x_j k)}}\Big)d_j(k) + log\Big(\frac{1}{1+e^{f_\theta(x_j k)}}\Big)(1-d_j(k))\Big) \quad (6)$$

Where $f_\theta(x_j, k)$ is the k^{th} element of the output vector $f_\theta(x_j)$.

The logistic loss assumes that the survival probabilities in each interval are independent from each other, whereas it is not the case for survival analysis. In particular, the estimated probability of survival at the k^{th} interval would be:

$$p_\theta(x_j, k) = \prod_{l=0}^{k} f_\theta(x_j, l) \cong 1 - \prod_{l=0}^{k} P[t_{j+l}|x_j] \quad (7)$$

The loss function can be rewritten in terms of a sigmoidal function:

$$l(\theta) = \sum_{jk} w_{jk} l(p_\theta(x_j, k), d_j(k)) = \sum_{jk} w_{jk} log(1 + e^{|p_\theta(x_j,k) - d_j(k)| - \Delta p}) \quad (8)$$

Where $d_j(k)$ is target vector and $p_\theta(x_j, k)$ the estimated survival probability for each interval and for each patient. This formulation is defined as quasi-logistic loss; it is worth noting that this network output does not add any parameter to the architecture, but rather forces an interplay among the parameters of the output-layer providing the probabilities at the different intervals.

4 Experiment

In order to compare the performance of both the proposed architecture and the proposed objective function Eq.8, we tested 4 network architectures (DeepSurv [2], SurvivalNet[3], ConvSurv, ConvSurv with fully connected initial layer) to optimize either Cox-Partial-Likelihood, or the logistic loss, or the quasi logistic loss. We applied the models on data of patients affected by ALS, obtained from

the PRO-ACT Database, which includes data from 10723 patients, on multiple visits. The data were first pre-processed in order to remove all records with missing values, resulting in a dataset of 1936 patients and 16177 records, and then split to cross-validate the methods into 11 folds each containing all visits of a different set of 176 patients each: all records of a patient belong at each run either to the train or to the test set, so that correlations among features of the same patient can not be exploited by the network. Following [2], we tested DeepSurv by using one fully connected layer with a number of neurons $3.5 \cdot D$, where D is the number of available features. SurvivalNet was equipped with two fully connected layers, and a number of neurons per layer equal to $1.3 \cdot D$, following [3]. The number of output neurons were decided according to the type of objective function: one in case of Cox-Partial Likelihood, and six otherwise to predict survival probability from time 0 to 18 months at interval of 3 months, in line with the visiting sampling grid used in PRO-ACT. All experiments have been conducted in Matlab (The Mathworks, Inc) using Matconvnet [13], run on a NVIDIA® Titan Xp.

4.1 Results

The predictive performance is reported in Table 1 in terms of Areas Under the ROC Curve (AU-ROC), accuracy, Concordance index (CI) and mean absolute error (MAE). AU-ROC defines the ability of methods to classify correctly the state of a subjects based on their survival probability, averaged over different time points (months 3 to 18). Accuracy is the ability of correctly predicting a survival probability higher than 0.5 for each interval in which a patient is alive, and lower than 0.5 otherwise. CI [14] is defined as the probability that subjects with lower risk score have higher survival time; values of CI near 1 indicate a good ranking ability. Mean absolute error provides a measure of the discrepancy between the predicted time of death (in days from the current visit) and the real event.

It is worth noting that the ConvSurv network with a fully connected top layer and quasi logistic loss obtained the highest AU-ROC and accuracy and the best MAE; indeed, the convolutional layers coupled with a top fully-connected layer ensures to learn the most informative structures in the data whereas quasi logistic loss guarantees the modeling of dependencies among time intervals. Besides, all architectures optimizing both logistic and quasi-logistic objectives outperformed those optimizing Cox partial likelihood. As expected, the methods optimizing the ranking of subjects have higher C-index compared to those optimizing the survival probability estimation.

Ackowledgments

This work was funded by the bilateral Italian-Israel project CompALS (Computational analysis of the clinical manifestations and predictive modelling of ALS), supported by the Italian Ministry of Foreign Affairs and International Cooperation and the Ministry of Science, Technology and Space of the State of Israel.

		AU-ROC	Accuracy	C-index	MAE [days]
		Mean (StD)	Mean (StD)	Mean (StD)	Mean (StD)
CPL	SurvNet	0.67 (0.07)	0.63 (0.07)	0.58 (0.03)	566 (166)
	DeepSurv	0.62 (0.06)	0.68 (0.05)	0.59 (0.02)	544 (182)
	CNN	0.58 (0.09)	0.77(0.03)	0.55 (0.03)	676 (5)
Logistic	SurvNet	0.89 (0.01)	0.87 (0.01)	0.72 (0.04)	139 (9)
	DeepSurv	0.90 (0.01)	0.88 (0.01)	**0.73** (0.04)	140 (7)
	CNN	0.88 (0.01)	0.87 (0.01)	**0.73** (0.04)	157 (6)
	CNN+FC	0.88 (0.01)	0.85 (0.01)	0.72 (0.03)	151 (7)
Quasi Logistic	SurvNet	0.94 (0.01)	0.88 (0.01)	0.71 (0.04)	98 (6)
	DeepSurv	0.93 (0.01)	0.88 (0.01)	0.71 (0.04)	97 (7)
	CNN	0.92 (0.01)	0.87 (0.02)	0.70 (0.04)	104 (8)
	CNN+FC	**0.95** (0.01)	**0.89** (0.02)	0.71 (0.04)	**86** (9)

Table 1: Performance in predicting survival time. FC: fully connected layer. We report the cross-validated results in term of mean and standard deviation of the performance metrics over the splits used as test sets.

We gratefully acknowledge the support of NVIDIA Corp. with the donation of the Titan Xp GPU used for this research.

References

[1] Cox DR. Regression models and life-tables. In *Breakthroughs in statistics*, pages 527–541. Springer, 1992.

[2] Katzman JL et al. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Med Red Methodol*, (1):24, 2018.

[3] Safoora Y et al. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Scientific Reports*, (1):11707, 2017.

[4] Hazlett HC et al. Early brain development in infants at high risk for autism spectrum disorder. *Nature*, (7641):348, 2017.

[5] Atassi N et al. The pro-act database design, initial analyses, and predictive features. *Neurology*, (19):1719–1725, 2014.

[6] Breslow N. Covariance analysis of censored survival data. *Biometrics*, pages 89–99, 1974.

[7] Kalbfleisch JD. *The statistical analysis of failure time data*. John Wiley & Sons, 2011.

[8] Breslow N. Analysis of survival data under the proportional hazards model. *Int Stat Rev*, pages 45–57, 1975.

[9] Huang J et al. Interval censored survival data: a review of recent progress. In *Proceedings of the First Seattle Symposium in Biostatistics*, pages 123–169. Springer, 1997.

[10] Lindsey JC et al. Survival models: Methods for interval-censored data. *Tutorials in Biostatistics: Statistical Methods in Clinical Studies*, pages 141–160, 2004.

[11] LeCun Y et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, (10):1995, 1995.

[12] Krizhevsky A et al. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[13] Vedaldi A et al. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.

[14] Harrell JFE et al. Evaluating the yield of medical tests. *Jama*, (18):2543–2546, 1982.