Determining Output With Missing Data in Machine Learning Using Modified Neural Networks

Kenneth Marino
Georgia Institute of Technology

As the project manager of our university's Intelligent Ground Vehicle Competition team, I have often had to deal with problems in which partially observable states and uncertain or unknown sensory information is a factor. In particular, when our robot's agent program builds a real-time map of the environment, showing obstacles and local goals, it often has to contend with areas within the frame of the map where there is no measured data from vision or LIDAR (Light Detection and Ranging). As a result, our program must run path-planning steps without the benefit of complete knowledge of its environment. The path-planning algorithm treats areas without recorded data as an average of clear and impassible areas, and although this allows the robot to deal with the missing data, the simplification essentially creates false data and distorts the model of the environment. This treatment often causes suboptimal paths to be chosen because the program has made an incorrect assumption about the environment. If the missing data could be intelligently handled without attempting to replace unknown data, the robot would be able to make better decisions in path planning. An algorithm that could learn to predict output with a partial subset of the necessary input data could not only substantially improve our robot's planning strategy but also help solve similar problems in the field of robotics.

The normal practice in a number of supervised machine learning algorithms is to determine a discriminate function or regression function given a set of N training examples and results $\boldsymbol{X} = \{\boldsymbol{x}^t, \boldsymbol{y}^t\}_{t=1}^{N}$, where $\boldsymbol{x}$ is a vector of inputs or features, and $\boldsymbol{y}$ is a classification or real valued output [1]. Because of incomplete data caused by faulty sensors, peculiarities in data collection, or another reason depending on the application, a particular feature in the input $\boldsymbol{x}$ may not be present in all of the training examples. Most machine learning algorithms of this type do not respond well to missing input data and cannot evaluate particular data samples with missing features unless the data is preprocessed to replace the missing data.

For large enough data sets, the state-of-the-art methods for dealing with missing at random features are the multiple imputation method and the expectation maximization algorithm [2]. Although these methods provide reasonable estimates of missing data for the purpose of analyzing or learning from incomplete data sets, they will not usually allow one to effectively predict the output of single examples with missing features. In these methods and others used to replace missing data, whichever algorithm is predicting the output based on the recovered data assumes that the input data for each feature is measured and that useful information can be derived from them. In fact, these data replacement methods actually generate missing features using probabilistic methods and treat these features as collected data. Artificial Intelligence also has many algorithms that allow agents to operate in partially observable environments; however, these algorithms do not predict the values of output variables as in machine learning, but rather find a path to the solution despite missing information [3].

One possible solution to the problem of predicting output with missing input features is a specialized feed-forward neural network that handles missing data and is trained on data with missing features. Neural networks are a simplified representation of neurons and are commonly used to generate discriminate or regression functions in machine learning. Each neuron is represented by a transfer function of the inputs, the most common being $\sum_j w_{ij} x_j$, the weighted sum of inputs into the i$^{\text{th}}$ node from each of the anterior nodes, and an activation function, such as the sigmoid function $s(x) = \frac{1}{1+e^{-x}}$, which is common in classification networks [4].

The proposed network works much like a standard feed-forward network, except each environmental input is paired with a Boolean value, the missing-data input. Its value is 1 when the variable is known and 0 when the variable is unknown. Instead of the standard weighted sum of inputs, the first hidden layer uses a weight reassignment node which still uses weighted sum, but does some important processing steps beforehand. For each of the 1's from the missing-data inputs, the function $w_r' = w_r \left(\frac{\sum_{k=1}^{n} w_k}{\sum_{k=1}^{n} w_k - \sum_m w_m} \right)$ is used to alter each of the weights w_r in the node whose corresponding inputs are known given the weights $\sum w_m$ that correspond to the missing inputs. After this step, the transfer function calculates the weighted sum of the known inputs, using the new weights calculated by this operation as well as the missing-data inputs. Implicitly, each node in the network computes a function based on the input it receives from the environment or from other nodes, so using the "missing-ness" as an input allows the function to make trade-offs and compute the output based on which of the inputs are missing as well as the known inputs. The methods mentioned previously allow machine learning algorithms such as neural networks to ignore these details and compute the output based on the recovered data sets. In contrast, this method allows the network to consider the effect of the missing data. Since the weights corresponding to the missing-data inputs determine how the nodes handle missing data, and these weights, along with the others, can be trained by an algorithm such as gradient descent, the neural network itself can optimize the way in which the missing data is handled.

One possible application of this type of network in the field of medicine is a diagnostic system that reduces the number of tests needed to diagnose the cause of a condition such as anemia, which has dozens of possible causes and tests. Rather than performing dozens of tests and analyzing the results, tests will be performed one by one, and once each has been completed, the results will be fed into the network and used to predict the patient's medical condition. Based on the outcome of the last prediction, another test will be chosen automatically and performed immediately afterward until there is enough information to make a diagnosis. Standard neural networks are perfectly capable of doing this, but if it is not known ahead of time which tests should be performed in which order, $2^n - 1$ networks would be needed to specify all the combinations of missing inputs from tests to predict output. For a large number of tests, this is impractical. Since many of these tests are automated and can use the same blood samples, a program using this algorithm could drastically reduce the number of tests needed to make a diagnosis. Choosing which tests to perform with an algorithm, rather than running a battery of tests, would save hospitals money and would allow for more affordable coverage, especially important for those with little or no insurance.

The idea of using "missing-ness" as an additional input has the potential for strengthening the usefulness of neural networks. I hope that in the course of my professional career, I, along with others in the field, can combine new missing data techniques with developments in the self-generation of neural network structures and other advances to make these techniques more useful in general problem solving. The overall development of the field will hopefully enable us to achieve greater success for a wide variety of problems, even beyond the areas in which neural networks have typically been used. Perhaps the development of this field in parallel with other improvements in artificial intelligence and machine learning will bring us closer to the original promise of artificial intelligence: solutions for a multitude of problems without a priori knowledge of the solution or problem space, and the creation of agents that are able to act intelligently without human intervention.

Whether or not advances in neural networks are a factor, it is possible that in the near future we will have agent programs capable of understanding human language at a deep level. Robots may be able to navigate in any arbitrary, real-world environment without prior knowledge, reacting to unexpected events around them in a systematic way. I look forward to being a part of the field of Artificial Intelligence and doing exciting research on the cutting edge. I do not just want to read about the next great advance in Artificial Intelligence, I want to be part of the team responsible for it.

References

[1] Alpaydın, E. (2010). *Introduction to Machine Learning* (2nd ed.). Cambridge, Massachusetts: MIT Press.

[2] Wang, T. and Wu, L. (2011), Multiple Imputation Methods for Multivariate One-Sided Tests with Missing Data. Biometrics. doi: 10.1111/j.1541-0420.2011.01597.x

[3] Russell, S. and Norvig P. (2010), *Artificial Intelligence A Modern Approach* (3rd ed.). Upper Saddle River, New Jersey: Prentice Hall.

[4] Reiterer, Alexander et all (2010). Application of artificial intelligence in Geodesy – A review of theoretical foundations and practical examples. *Journal of Applied Geodesy,* 4, 201-2