“Just Enough Delivery”

a neural network system for a better distribution of

single-copy newspapers

A product of SMART Research BV

JED

“Just Enough Delivery”

Better sales through improved quality of distribution

In the competitive environment of newspapers and magazines, the distribution of sales is playing an increasingly important role in obtaining higher profits. Higher distribution efficiency means lower cost. JED helps you to obtain better single-copy sales by reducing sellouts and returns.

JED produces better predictions of the number of copies to be delivered to individual points of sale. With the same amount of deliveries, it gives you a lower percentage of returns and sellouts, and thus more sales.

LESS SELLOUTS	⇒	**HIGHER PROFIT**
LESS RETURN COPIES		**LOWER COSTS**

JED is currently in full operation at “De Telegraaf”, a major newspaper in the Netherlands. With the same amount of sales, JED needed 5 % less single copy deliveries, compared to the system previously used.

The basic principles

JED discovers the relationship between explanatory variables and the actual sales results.

JED is based on an adaptive method: it *learns* from the past to *predict* the future. Learning means adapting a statistical model to historical sales figures. The statistical model is a so-called feed forward neural network. It consists of layers of neurons or units: an input layer, hidden layers and an output layer. Information is propagated from the input layer to the hidden layer and from the hidden layer to the outputs. Parameters in between layers of units are called weights. During training, the weights are adapted to give the closest fit between the model outputs and the observed sales figures. In this way, the model captures the relationship between the inputs (explanatory variables) and the outputs (sales figures).

This relationship is used for the weekly prediction. The inputs of the neural network are the current values of the explanatory variables for the point of sale we want to consider. The output of the model is the expected sales of the individual points of sale. An important feature of JED is that it not only yields sales prediction, but also an estimate of the uncertainty of this prediction. These two are then combined to compute the optimal delivery, which will usually be somewhat higher than the expected sales. How much higher depends both on the uncertainty in the prediction and the company's sales strategy.

An important point in the design of any prediction model is the choice of explanatory variables, the inputs of the model. Possible explanatory variables for predicting newspaper sales may include

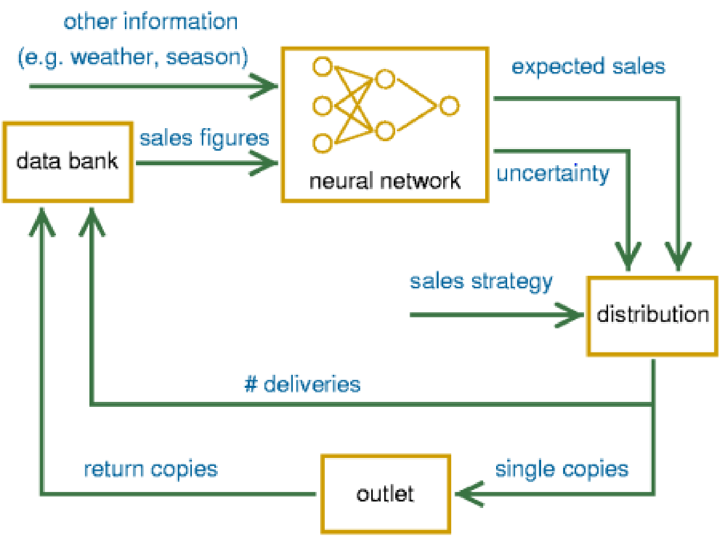
- recent sales figures
- season and holidays
- last year's sales figures
- information about sellouts
- weather information
- news content
- ....

Taking into account too many input variables may lead to an effect called overfitting, which results in poor prediction. JED makes use of several clever techniques to avoid this. These will be explained in more (technical) detail below.

JED within your organization

Data at the level of individual points of sale must be available to the system on a continuous basis.

The figure below sketches the flow of information for a single point of sale. The sales figures are stored in a data bank. This information, supplemented by other information, such as weather figures or seasonal information, is used to update the neural network. This is done on a weekly basis to account for the latest information. Using the most recent figures, the network gives a prediction of the upcoming single-copy sales. This prediction is combined with an estimate of its uncertainty and your sales strategy (focus on reducing sellouts or returns?) to yield the number of single copies to be delivered. Information about the realized sales, derived from the deliveries and the returns, is again stored in the data bank for future training and prediction.



Scheme sketching the flow of information for a single point of sale.

JED relies heavily on the availability of historical sales figures. This makes JED only useful for newspaper companies and journal distributors that

- work with a right of return;
- have sales figures for at least one year, but preferably for two or more years;
- collect sales figures at the level of individual points of sale.

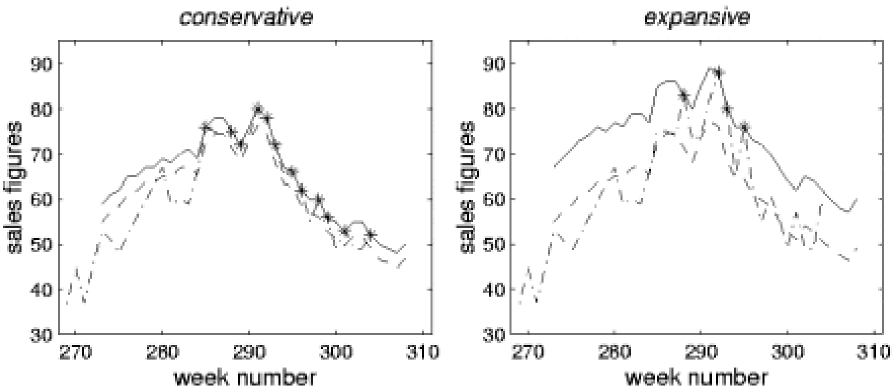
Each newspaper or magazine has its own characteristics: a different reader group, different distribution channels and so on. Therefore, JED is custom made for each newspaper or magazine.

Efficient distribution

JED can be tuned to your sales strategy to compute the most efficient distribution of single copies.

The figures below explain JED's operation for two possible sales strategies: a conservative strategy which mostly focuses on reducing return copies, and an expansive strategy which aims at higher sales figures through a decrease of sellouts. We illustrate these strategies on data from De Telegraaf, regarding the sales of Saturday's newspaper. There is a delay of 4 weeks between the most recent sales figure and the upcoming delivery.

The dashed and solid lines show JED's predictions and deliveries, respectively. The realized sales figures are the dash-dotted lines. Stars indicate sellouts.



Realized sales (·-), expected sales (--), and deliveries (—) for different sales strategies.

The deliveries corresponding to a conservative strategy are shown on the left-hand side. The conservative strategy in this case yields only 113 return copies (6%), but 12 days with sellouts (41%).

The right-hand side gives the deliveries for a more expansive strategy. Now we have only 4 days with sellouts (14%), but 323 returns (15%). The expansive strategy corresponds to higher costs (261 more copies delivered than with the conservative strategy), but also higher sales (51 more copies sold than with the conservative strategy).

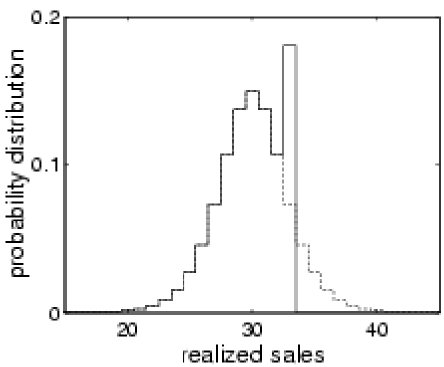
JED in more detail

JED is based on state-of-the-art technology to cope with sellouts and overfitting.

The methods implemented in JED are more advanced than what is typically encountered in a standard neural network application for time series prediction. It not only yields predictions for the upcoming sales, but also estimates the uncertainty of these predictions. The uncertainty consists of two parts: the uncertainty in the model parameters (which is particularly relevant for points of sale that just started) and the noise inherent to the problem. This allows for the computation of the optimal delivery.

Another technical, but quite relevant problem is the incorporation of historical sellouts. We want to predict how many copies can be sold and thus have to fill in the number of copies that could have been sold (that we do not know). Neglect of this effect leads to a structural underestimation of the realizable sales. The solution to this problem is based on a probabilistic approach that allows for an elegant treatment of partially missing information.

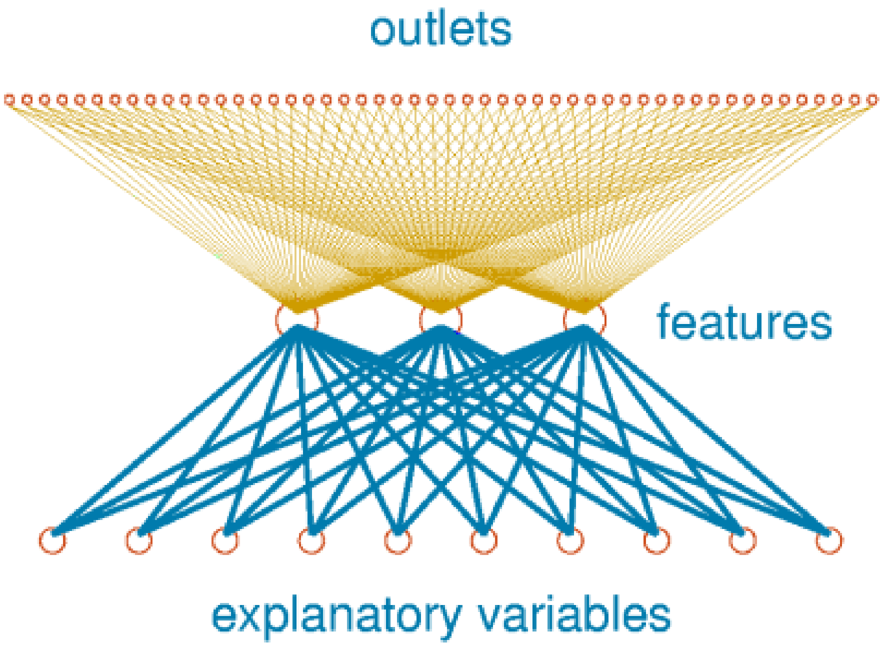
Sales is modeled within a probabilistic framework.



An even more important problem is the risk of overfitting. If we include many explanatory variables in our model, we will typically find some strong correlations between the inputs and the output. These correlations seem to explain the data in the training set almost perfectly, but are completely useless and wrong when it comes to prediction, the thing that we are really interested in. JED avoids overfitting by making the points of sale “learn from each other”. This idea is implemented in two ways.

Firstly, all points of sale are combined in one huge network architecture. Each output in this network corresponds to a single point of sale. All points of sale share the weights from the input – representing the explanatory variables – to the hidden layer. Since all points of sale cooperate to learn the input-to-hidden weights, there is sufficient data to prevent overfitting this part of the network, even with a relatively large number of inputs. The hidden units represent typical features for the task of predicting newspaper sales and are shared by all points of sale. Each newspaper has its own typical features. These can be learned on a representative set of points of sale.

The weights from hidden to output are specific to each point of sale. They are updated weekly, i.e., based on the most recent available information. A small number of hidden units controls the risk of overfitting in this part.



The neural architecture.

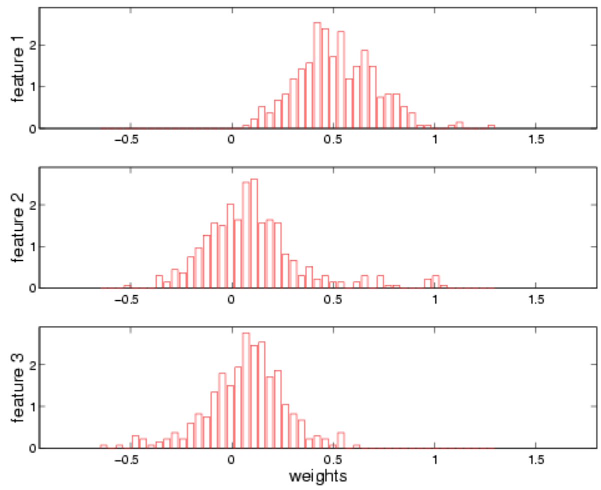
The second technique is based on the observation that, prior to any information about sales figures, there is no way to distinguish between the different points of sale. This allows for a method called hierarchical Bayesian modeling in statistics, which is somewhat similar to the use of weight decay in neural networks, or more generally, regularization.

The histograms on the next page visualize the weights in the neural network architecture specific to each point of sale. Each histogram shows the connections of one of the hidden units to a group of about 300 points of sale, after training and Bayesian inference. The Bayesian machinery automatically determines the appropriate mean and width of each distribution. As a result, each point of sale keeps its specific character, with a tendency to act like an average point of sale in order to prevent overfitting.

What makes JED better than any other approach?

JED deals with differences and similarities between points of sale in a special way. This makes JED better than other techniques.

Alternative approaches such as Box-Jenkins, ARMA time-series modeling, and Holt-Winters exponential smoothing techniques, are similar in spirit to neural networks. In all these methods, a statistical model is fitted to historical data. Neural networks are generally considered to be more powerful techniques. It is the special architecture of JED that makes the difference: it allows for the combination of many explanatory variables, while controlling the risk of overfitting. The more traditional approaches consider each point of sale individually, which makes them much more restricted in the type of relationships that they can find. The traditional alternative to Bayesian techniques is cross-validation. Cross-validation at the level of individual points of sale is inefficient and very time-consuming.



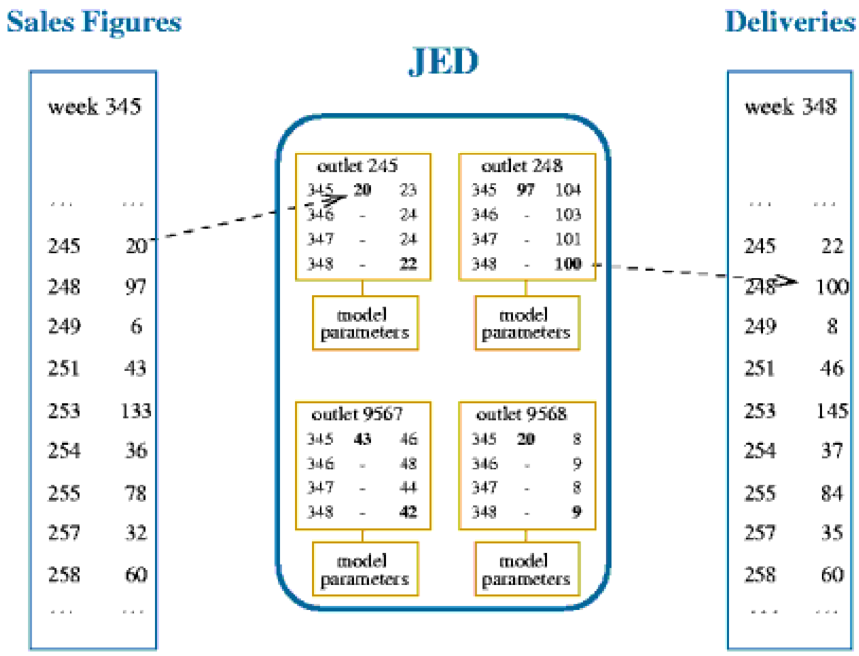
Histograms of weights specific to individual points of sale: each combination of weights gives rise to a different "rule".

At the other extreme are rule-based methods, in which the upcoming delivery is computed according to a predetermined set of rules. Usually these rules are the same for each point of sale, with only minor exceptions for e.g. points of sale that are known to be sensitive to the time of year. The basic question for rule-based techniques is how to determine which rules to use and how to distinguish between obviously different points of sale. The way in which JED specifies the deliveries can also be interpreted as the result of a set of rules. The important advantage is that these rules are learned and updated instead of predetermined. Moreover, the Bayesian techniques let the data decide to what extent the various points of sale should apply different rules instead of the same rule for all points of sale.

Software description

JED communicates with other information systems through files. It keeps track of its own databases to allow for rather easy implementation and maintenance on any platform.

As one can tell from the previous few pages, JED is based on advanced technology. This, however, should not imply that you have to become an expert on neural networks in order to benefit from them. The task is simple: determine the optimal delivery given the available information.



Sketch of how JED organizes the information.

Communication between JED and other information systems within your organization goes through databases. It is sketched in the figure above. The input to JED is a file containing the sales figures of all points of sale in a particular week and databases containing information about explanatory variables not dealing with sales figures (weather information, for example). JED's output is a similar file with deliveries for the upcoming week. For each point of sale, JED keeps track of two files: a data file and a model file. The data file contains the historical sales figures and the deliveries. The model file contains the model parameters specific to this point of sale.

On a weekly basis, the following main commands have to be executed:

1. **Store** (previous_week)

- transfers the sales figures realized in "previous_week" from the main file to the data files for the individual points of sale.

2. **Compute** (next_week)

- computes the delivery for "next_week" for all points of sale.

3. **Retrieve** (next_week)

- transfers the deliveries from the points of sale' data files to the main file.

The crux is of course in the computations involved in the second command. Computing the delivery to "point_of_sale" for "next_week" can be split up in three consecutive operations:

2a. **Update** (point_of_sale, next_week)

- updates the model parameters (weights from hidden units to output) to account for the latest information.

2b. **Predict** (point_of_sale, next_week)

- uses the most recent model parameters to compute the expected sales and its uncertainty given the input information.

2c. **Deliver** (point_of_sale, next_week)

- combines the expected sales and the uncertainty with the sales strategy to compute the optimal delivery.

The "sales strategy" is a factor completely under control of the user. It is represented in JED through a relative cost factor, relating the cost of a sellout (a copy which could have been sold with a higher delivery) to the cost of a return (which could have been avoided with a lower delivery). The higher the cost factor, the stronger the focus on reducing sellouts instead of reducing returns, and thus the higher the delivery.

JED keeps track of about two to three years of sales figures for each point of sale. Depending on the number of points of sale, this may require quite a lot of disk space. Computation time will not be a problem: computing the optimal delivery for a point of sale takes no more than two seconds on a PC.

A typical implementation process

After building a prototype of JED optimized to your specific situation, JED is installed and maintained by SMART Research BV at your offices. In about 4 months, the system is operational.

Introductory meetings

We are always willing to give you more information about JED and/or to come over to give a presentation and a computer demo explaining its main features. For us, this is also an opportunity to listen to your specific problems and wishes.

Data collection

For building a prototype, we will need some of your data. Each journal title or each day of the week for newspapers has its own characteristics. To build a prototype, we need a representative set of about 500 points of sale and at least one, but preferably two or more years of sales figures.

Prototype building

A prototype is developed based on the collected data and your specific wishes. During the development of this prototype, we obtain lots of information about the optimal settings of the architecture and other global parameters. At the end of this phase, we present our results in the form of a report and presentation. The report contains

- an estimate of the profit compared to the realized sales in the past;
- an analysis of the relative importance of the explanatory variables;
- recommendation for implementation.

Implementation

In the implementation phase the optimal settings, obtained during the prototype development, are built as constants into the software. This is why the software itself is quite transparent, flexible and easy to work with. JED's basic building blocks are written in C/C++. The software can be implemented on any platform and fully customized to your wishes. It comes with full documentation.

Service

We offer training, maintenance and upgrades. We like to think of JED as a perfect combination between a solid piece of software and our expertise.

The organizations behind JED

JED is a commercial product of SMART Research BV. SMART stands for Statistical Modeling and Artificial Reasoning Technology. We build and maintain software systems based on neural networks and related technologies. Our main activities are in an area that can be summarized as “the prediction of consumer behavior”.

SMART Research BV is a spin-off company of SNN, the Dutch Foundation for Neural Networks. SNN is one of Europe’s leading institutions in neural network research, interested in its theoretical aspects and with an open eye towards industrial applications. SMART Research BV and SNN are affiliated with the University of Nijmegen.

Please check out our website at *http://www.smart-research.nl* or contact

Wim Wiegerinck
SMART Research BV
P.O. Box 31070
6503 CB Nijmegen
The Netherlands

Phone: 31-(0)24-3617497
Fax: 31-(0)24-3541435
Email: jed@smart-research.nl