Learning Sequence Neighbourhood Metrics

Justin Bayer, Christian Osendorfer, Patrick van der Smagt
Fakultät für Informatik
Technische Universität München
bayer.justin@googlemail.com, osendorf@in.tum.de, smagt@tum.de

1. Introduction

Storing short descriptors of sequential data has several benefits. First, they typically require much less memory and thus make processing of large data sets much more efficient. Second, if the descriptors are formed as vectors, e.g. $x \in \mathbb{R}^n$, numerous algorithms tailored towards static data can be applied. Instead of applying static data algorithms to dynamic data, we propose to learn a mapping from sequential data to static data first. This can be done by combining recurrent neural networks (RNNs), a pooling operation and any differentiable objective function for static data. In this work, we present how neigbourhood components analysis (NCA) (Goldberger et al. 2004) can be used to learn meaningful representations which lead to excellent classification results and visualizations on a speech dataset.

2. RNNs for fixed length objective functions

Recurrent neural networks are a state space model extension of feedforward networks. The inputs to an RNN are given as a sequence $(x_1, x_2, \dots, x_T)$. Subsequently, a sequence of hidden states $(h_1, h_2, \dots, h_T)$ and a sequence of outputs $(o_1, o_2, \dots, o_T)$ is calculated via the following equations:

$$\begin{aligned} h_t &= \sigma(W_{xh} x_t + W_{hh} h_{t-1} + b_h) \\ o_t &= W_{ho} h_t + b_o \end{aligned}$$

where $t = 1, 2, \dots, T$ and σ is a suitable transfer function, typically the tangent hyperbolic, applied element-wise. $W_.$ are weight matrices $b_.$ bias terms. x_i, h_i and o_i are real valued.

We reduce a sequence to a point similar to (Collobert et al. 2011) with a pooling operator. Given a network f parametrized by W, a data set $\mathcal{D} = \{x_i\}$, a pooling operator p and an objective function $\mathcal{O}$ we proceed as follows: (1) Process input sequences $\mathbf{x}_i = (x_{i1}, \dots, x_{iT})$ to produce output sequences $f(\mathbf{x}_i; W) = \mathbf{o}_i = (o_{i1}, \dots, o_{iT})$. (2) Use a pooling operation p to reduce the output sequences to points via $p(o_{i1}, \dots, o_{iT}) = e_i$. (3) Calculate the objective function $\mathcal{O}(\{e_i\})$.

One important point is that the whole calculation is differentiable. We can thus evaluate the derivative of the objective function with respect to the parameters of the RNN via $\frac{\partial \mathcal{O}}{\partial p} \frac{\partial p}{\partial f} \frac{\partial f}{\partial W}$. Subsequently, we can use the gradients to find embeddings $\{e_i\}$ of our data that optimize any objective function commonly used for static data.

3. Sequential Neighbourhood Components Analysis

Given a set of sequences with an associated class label $\mathcal{D} = \{x_i, c_i\}$ mapped to a set of embeddings $\mathcal{E} = \{e_i\}$, we define the probability that a point a selects another point b as its neighbour based on Euclidean pairwise distances as $p_{ab} = \frac{\exp(-||e_a - e_b||^2)}{\sum_{z \neq a} \exp(-||e_a - e_z||^2)}$, while the probability that a point selects itself as a neighbour is set to zero: $p_{aa} = 0$. The probability that a point i is assigned to

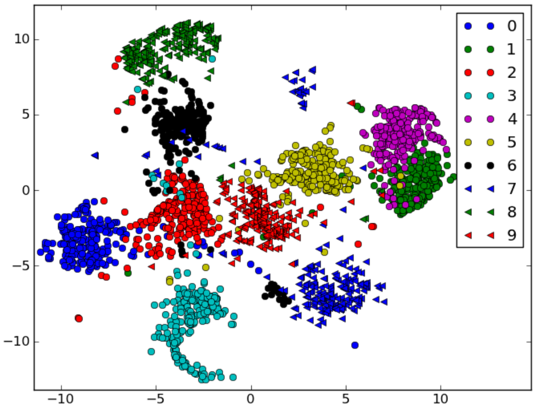

Figure 1: Two dimensional embeddings for the TIDIGITs test data. The data was first reduced to 30 dimensions using an NCA trained RNN and afterwards visualized with tSNE. Note how the samples for digits 6 and 7 arrange themselves in more than one cluster. The similar sounding 1 and 4 are placed near each other, suggesting a semantically meaningful metric.

class k depends on the classes of the points in its neighbourhood $p(c_i = k) = \sum_j p_{ij}\mathbb{I}(c_j = k)$ [1]. The overall objective function is then the expected number of correctly classified points $\mathcal{O} = \sum_i \sum_j p_{ij}\mathbb{I}(c_i = c_j)$.

4. Experiments

We use TIDIGTS to show that our method is useful for practical applications. TIDIGITs is a dataset consisting of the spoken digits from 0 to 9. The data we use consists of 4500 samples, partitioned into 2000 samples for training, 240 samples for validation and 2260 samples for testing. The raw audio signal was preprocessed with a mel frequency filter bank and additional whitening. For pooling we chose the $\max$ operator. Due to the vanishing gradient problem that makes learning of long range dependencies difficult, we resort to a more sophisticated type of hidden units for RNNs, Long Short-Term Memory cells(Hochreiter and Schmidhuber 1997).

In terms of optimization, the method reliably found good minima in few iterations, rarely overfitting. The best achieved classification accuracy on the test set we achieved was 97.9%.

5. Conclusion and Future Work

We presented a solution to an important problem—by combining two well established methods we introduced a new approach to embed sequential data into a semantically meaningful metric feature space. We think that investigating further objective functions, especially unsupervised ones, is a promising direction of research for metric learning on structured data. We want to emphasize that calculation of the descriptor of a new sequence takes time proportional to the length of the sequence, and is independent of the amount of already seen data.

References

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research* (to appear).

Goldberger, Jacob, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. 2004. Neighbourhood components analysis. *Advances in Neural Information Processing Systems 17*. MIT Press.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9: 1735–80. doi:10.1162/neco.1997.9.8.1735.

[1] Here $\mathbb{I}$ is the indicator function that returns 1 if the argument is true and 0 otherwise.