Multi-task modular backpropagation for feature-based pattern classification

Dr. Rohitash Chandra

1. Centre for Translational Data Science
http://sydney.edu.au/data-science
2. Discipline of Business Analytics
School of Business

International Conference on Neural Information Processing
Ramada Pearl Hotel, Guangzhou, China
www.iconip2017.org

Overview

Background

- Embedded control systems guided with machine learning can encounter states that have disruptions in stream of data from sensors. This can as a result provide poor decision making for actuators.
- Due to knowledge representation as modules, the system should be able to make decisions with some degree of uncertainty even if some of the modules are damaged or missing.
- Such motivations come from biological neural systems, i.e due to modular knowledge representation, one is able to see even if one eye is damaged.

Background and Motivation

- Multi-task learning employs shared representation knowledge for learning multiple instances from the same problem. In the case of time series, multi-task learning can consider different a set of embedding dimensions as tasks that have shared knowledge representation.
- The motivations from related learning techniques is incorporated, in particular, multi-task learning for modular knowledge representation in neural networks.
- A multi-task modular back-propagation algorithm is presented that takes into account feature groups that are partitioned from the data. The method produces a modular network that provides decision making for the partitioned feature groups.

The Robot Control Motivation

Consider a robot navigation problem that depends on input sensors that are controlled through a neural network.

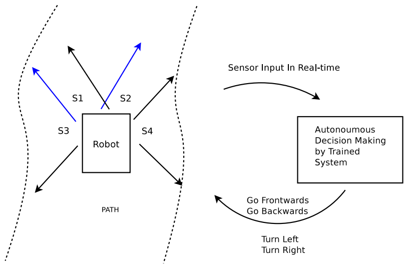
S1
S2
S3
S4
Robot
PATH
Sensor Input In Real-time
Autonoumous Decision Making by Trained System
Go Frontwards
Go Backwards
Turn Left
Turn Right

Figure: Control problem for autonomous robots based on sensor input (S1, S2, S3, and S4). The decision making can be done through a control system that features modular pattern classification.

Feature groups: Multi-task modular backpropagation

A feature group, X_m, is a subset of features. The overlapping subtasks $\Omega_1, \ldots, \Omega_m$ are defined as the union of selected feature groups $X_1, \ldots X_m$, for $m = 1, \ldots, M$, where M is the total number of feature groups, so that;

$$\begin{aligned} \Omega_1 &= [X_1] \\ \Omega_2 &= [X_1, X_2] \\ \Omega_3 &= [X_1, X_2, X_3] \\ \Omega_M &= [X_1, X_2, ..., X_M] \end{aligned} \tag{1}$$

Multi-task modular backpropagation

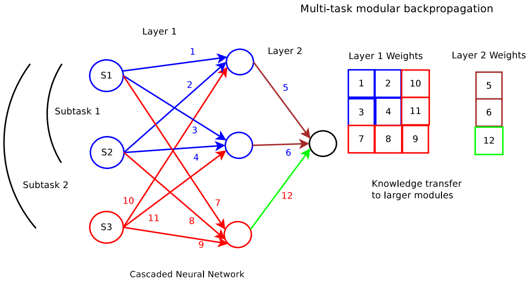

Figure: This cascaded modular neural network which can also be viewed and implemented as an ensemble of neural networks. The colours associated with the synapses in the network are linked to their encoding that are given as different modules. Subtask 1 employs a network topology with 2 hidden neurons while the rest of the modules add extra input and hidden neurons.

Proposed Method: Multi-task modular backpropagation

The cascaded network module θ_m of subtask m is constructed by combining with current Φ_m and previous network module Φ_{m-1} as follows.

$$\begin{aligned}
\Phi_1 &= [\omega_1, \upsilon_1]; \;\; \theta_1 = (\Phi_1) \\
\Phi_2 &= [\omega_2, \upsilon_2]; \;\; \theta_2 = [\theta_1, \Phi_2] \\
&\vdots \\
\Phi_M &= [\omega_M, \upsilon_M]; \;\; \theta_M = [\theta_{M-1}, \Phi_M]
\end{aligned} \tag{2}$$

The list of network modules considered for training or optimisation is therefore $\boldsymbol{\Phi} = (\Phi_1, \ldots, \Phi_M)$.

Algorithm: Multi-task modular backpropagation

Alg. 1 Multi-task modular backpropagation

```
Step 1: Partition $n$ subtasks $\Omega_n$ from data.
Step 2: Define cascaded neural network of modules: $\theta_n = f(i_n, h_n, o)$
while until termination do
    for each Module $\theta_n$ do
        for Depth $d$ do
            i. Forwardpropagate($\theta_n$, $\Omega_n$)
            ii. Backpropagate using Gradient-descent($\theta_n$, $\Omega_n$)
            iii. a) Calculate gradients
            b) Weight updates
        end for
        transfer-knowledge($\theta_n$, $\theta_{n+1}$)
    end for
end while
```

Design of Experiments

- Performance evaluation of the proposed multi-task modular backpropgation method for feature-based and modular pattern classification. We compare the results with canonical backpropagation that features gradient descent for pattern benchmark classification tasks.
- The size δ of each subtask (Ω_m) is given by the proportion β considering the total number of features for the given problem p.

$$\beta = [0.25, 0.5, 0.75, 1] \tag{3}$$

$$\delta_n = \beta_n * p \tag{4}$$

Design of Experiments

- The number of hidden neurons for foundation module that corresponds to Ω_1 is given by $\hat{h}$ in Table given in the next page. The rest of hidden neurons in the respective network modules are incremented by $h_n = \hat{h} + h_{n-1} + \epsilon$, given that $h_1 = \hat{h}$. $\epsilon = 2$ is used in all the experiments.
- Note that the two major strategies in multi-task modular back-propagation algorithm deal with number of iterations used for each module while they are evolved or trained in a round-robin fashion. The performance is evaluated where adaptive and fixed depth of search strategies are used.
- The fixed depth of $f = 5$ is used. The adaptive depth α considers a set of values where $\alpha = [10, 7, 4, 1]$.

Dataset

Table: Configuration

Dataset	Features	Classes	Instances	$\hat{h}$	Max. Time
Iris	4	3	150	6	500
Wine	13	3	178	6	500
Cancer	9	2	699	6	1000
Heart	13	2	270	16	2000
Credit	15	2	690	20	3000
Baloon	4	2	20	5	500
Tic-Tac-Toe	9	2	269	30	2000
Ionosphere	34	2	351	8	500
Zoo	17	7	101	6	300
Lenses	4	3	24	5	500
Balance	4	3	625	8	200
Robot (Four)	4	4	5456	14	2000
Robot(TwentyFour)	24	4	5456	14	2000

Results

Table: Results

Problem	Domain	Adapt	Depth	MTMB		Fixed	Depth	MTMB		BP-GD
		Ω_1	Ω_2	Ω_3	Ω_4	Ω_1	Ω_2	Ω_3	Ω_4	
Iris	Train	55.73	60.52	74.79	80.18	52.70	59.27	89.76	95.09	93.30
	(std)	4.89	3.72	4.96	4.39	4.95	5.04	3.16	1.38	0.89
	Test	85.25	88.67	89.42	93.08	88.75	90.00	91.08	94.42	83.50
	(std)	5.38	5.07	3.14	4.46	5.31	4.38	4.36	2.56	2.47
Wine	Train	98.26	99.81	99.83	99.95	98.09	99.86	100.00	100.00	98.45
	(std)	0.91	0.42	0.55	0.26	1.25	0.34	0.00	0.00	0.87
	Test	99.75	100.00	100.00	100.00	99.75	99.83	100.00	100.00	83.25
	(std)	0.75	0.00	0.00	0.00	0.75	0.62	0.00	0.00	9.90
Cancer	Train	86.32	90.03	93.59	93.84	86.03	89.56	94.00	94.56	94.07
	(std)	0.75	0.90	0.65	0.81	0.53	0.85	0.48	0.73	0.54
	Test	97.73	97.49	98.37	98.44	97.46	97.67	98.22	98.41	96.24
	(std)	0.47	0.87	0.27	0.39	0.55	0.91	0.27	0.41	0.57
Heart	Train	49.61	56.76	68.39	77.03	47.89	55.21	71.99	88.20	90.92
	(std)	3.72	4.36	2.11	2.45	4.21	5.15	2.96	2.69	1.44
	Test	74.78	75.67	72.56	79.11	75.22	76.22	73.41	77.56	70.63
	(std)	1.84	1.63	2.65	2.19	1.70	1.74	1.88	2.42	3.14
Credit	Train	19.79	42.53	82.34	84.35	18.94	40.62	83.24	87.44	89.64
	(std)	2.91	4.46	1.86	2.26	4.15	5.18	1.97	1.39	1.07
	Test	56.97	61.82	82.22	82.64	55.23	62.40	82.71	82.06	78.79
	(std)	3.32	2.96	1.73	1.43	4.59	2.76	1.67	1.48	1.56

Results

Table: Results

Problem	Domain	Adapt	Depth	MTMB		Fixed	Depth	MTMB		BP-GD
		Ω_1	Ω_2	Ω_3	Ω_4	Ω_1	Ω_2	Ω_3	Ω_4	
Zoo	Train	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.81
	(std)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.60
	Test	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.44
	(std)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.93
Lenses	Train	83.12	75.42	83.96	61.67	79.17	77.92	94.17	95.21	99.81
	(std)	15.15	17.60	21.51	24.72	14.99	15.46	9.12	5.97	0.60
	Test	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.44
	(std)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.93
Balance	Train	94.09	95.62	95.98	93.35	93.94	95.36	96.11	94.65	92.29
	(std)	0.00	0.72	0.63	1.05	0.57	0.97	0.53	1.60	7.85
	Test	100.00	100.00	99.98	100.00	100.00	100.00	100.00	100.00	88.33
	(std)	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	8.50
Robot (Four)	Train	99.95	100.00	100.00	100.00	99.93	100.00	100.00	100.00	95.95
	(std)	0.15	0.00	0.00	0.00	0.17	0.00	0.00	0.00	1.69
	Test	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.81
	(std)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.39
Robot (Twenty Four)	Train	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.69
	(std)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75
	Test	100.00	100.00	100.00	100.00	1.17	100.00	100.00	100.00	94.35
	(std)	0.00	0.00	0.00	0.00	2.17	0.00	0.00	0.00	6.21

Discussion

- Modularity enforced through backpropagation helped to retain the knowledge of the smaller network modules that are used as building blocks of knowledge for multi-task learning.
- Heterogeneous form of transfer learning is employed to transfer knowledge from small network modules to larger network modules which link with the respective subtasks.
- Multi-task modular back-propagation implements an ensemble of cascaded network modules that employ heterogeneous transfer learning for utilizing knowledge in smaller modules.
- This can also be viewed as a form of dynamic programming (optimisation) as the problem is decomposed into modules and knowledge from the base network modules are used in larger modules through transfer learning.

Conclusions and Future Work

- The results shows that the multi-task modular backpropgation algorithm can deliver similar or better performance to that of canonical (non-modular) backpropagation network. Hence, the proposed algorithm can train neural networks that do not lose performance although their knowledge representation featured modularity.
- Moreover, although unexpected, the algorithm outperformed canonical backpropagation in several cases.
- In future work, the modular method needs to be adapted further so that the trained network is operational even when the foundation or base subtask is unavailable during test phase.
- Uncertainty quantification through Bayesian inference methods for modular networks can also be explored.

Thank You

More information: https://rohitash-chandra.github.io