University of South Australia

Institute for Telecommunications Research

Deep Extreme Learning Machines: Supervised Auto-Encoding Architecture for Classification

Migel D. Tissera and Mark D. McDonnell

Computational and Theoretical Neuroscience Laboratory
Institute for Telecommunications Research, University of South Australia

University of South Australia

Institute for
Telecommunications Research

Computational and Theoretical Neuroscience Laboratory

Australian Government
Australian Research Council

University of South Australia
Institute for Telecommunications Research

www.itr.unisa.edu.au/ctnl


B

Outline

University of South Australia

- Brief introduction to Extreme Learning Machines (ELM)
- Motivation for Deep Extreme Learning Machines (Deep ELM)
- Deep ELM Architecture
- Improving accuracy using Weight Shaping Methods
- Experiments
- Discussion and conclusion

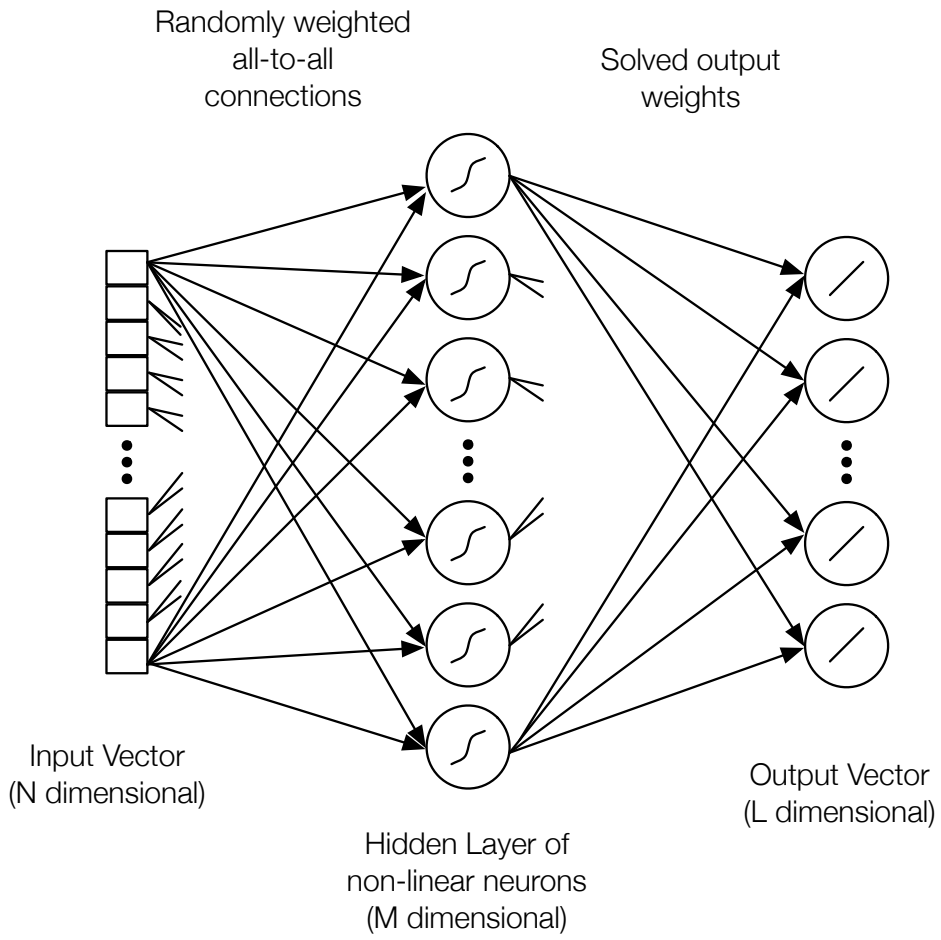
Extreme Learning Machines

University of South Australia

General Approach

1. Using random and fixed weights, connect an input layer to a hidden layer of sigmoidal neurons

2. Using training data, numerically solve for the output weights between the hidden and the output layers



Extreme Learning Machines

University of South Australia

Training an Extreme Learning Machine

Input to the hidden layer,

$$H_1 = \mathbf{W}_{P1} I$$

$$f[H_1]_i = \frac{1}{1 + \exp(-(H_1)_i)}$$

For *k* training data, we then form a matrix using the hidden layer activations,

$$\mathbf{A} \in \mathbb{R}^{M \times k}$$

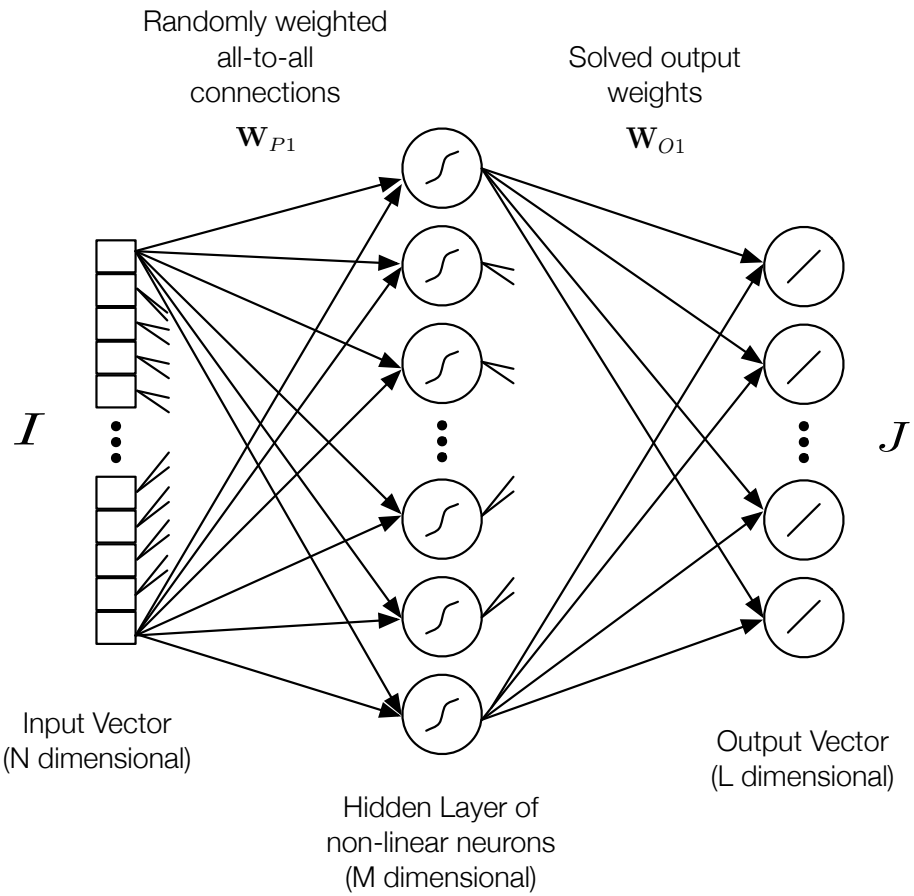
We then take the Moore-Penrose Pseudoinverse of A,

$$\mathbf{A}^{+} \in \mathbb{R}^{k \times M}$$

Now we can calculate the output weight matrix,

$$\mathbf{W}_{O1} = J\mathbf{A}^{+}$$

Where J represents the desired target output.



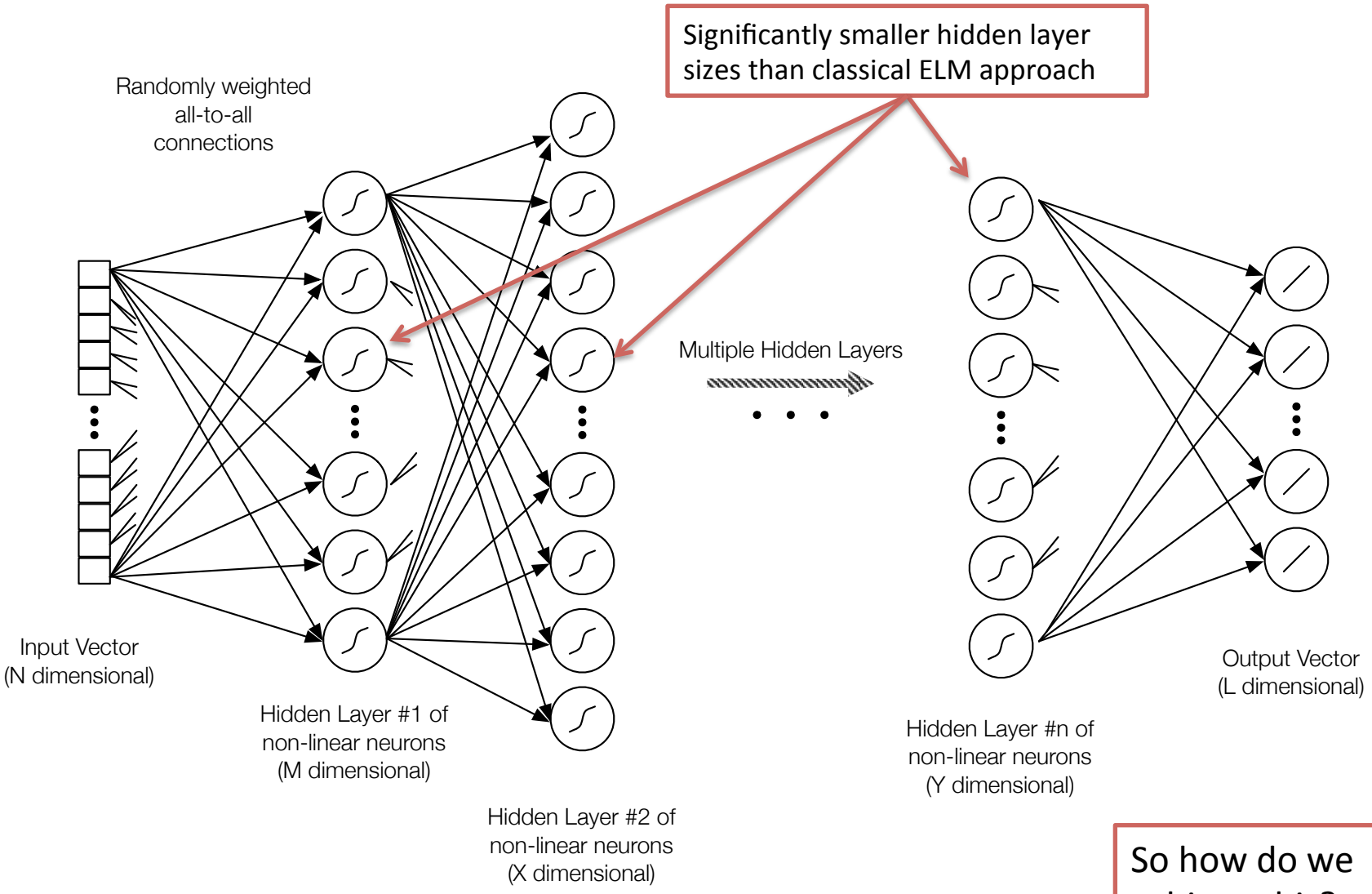
Motivation for Deep ELM

University of South Australia

- Drawbacks of classical single layer ELM
  - Large hidden layer size
  - Batch training (Restricts the use of it for real-time learning applications)
  - Combined batch training and a large hidden layer is computationally expensive, even for an extreme learning machine
  - Is it feasible for hardware implementations?
- Batch training have been addressed in on-line training algorithms
- How can we reduce the hidden layer size, and keep the same performance?
  - Can we have multiple hidden layers of smaller size?
    - Which would reduce the computational resources required, and significantly reduce the training time

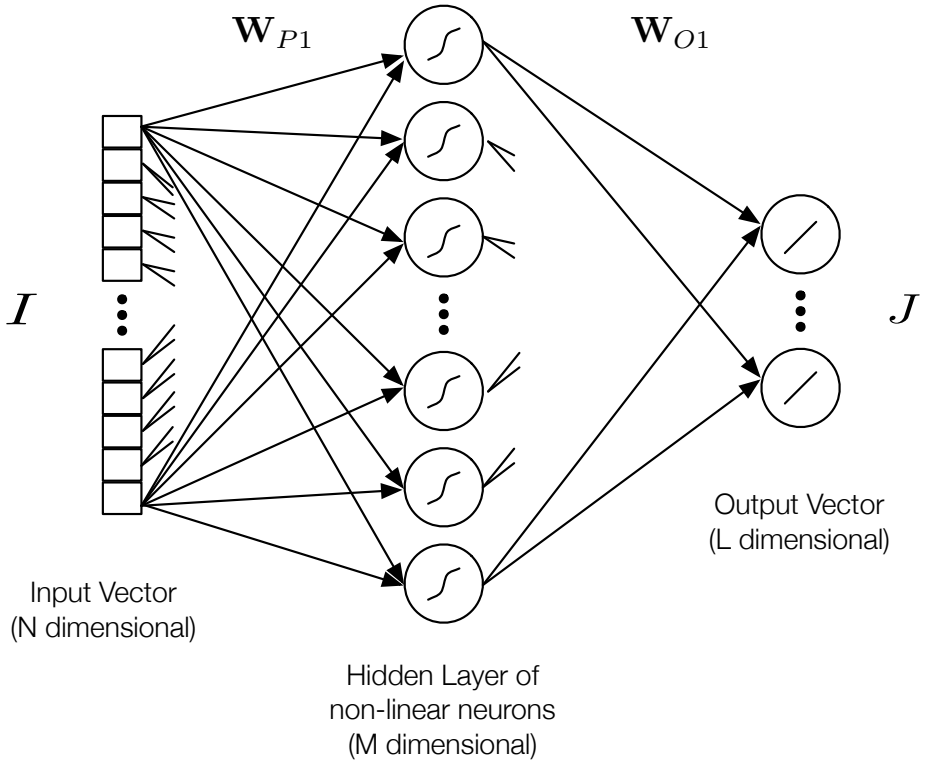
Deep ELM Architecture

University of South Australia



Deep ELM Architecture

University of South Australia



- We start by constructing a single layer ELM

$$H_1 = \mathbf{W}_{P1} I$$

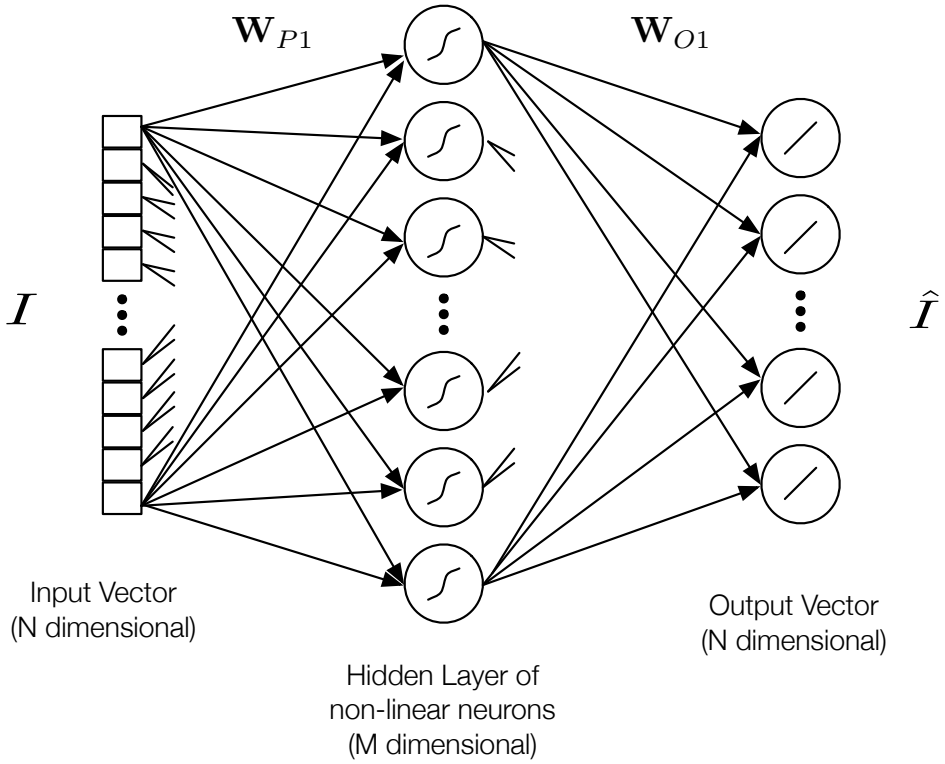
$$f[H_1]_i = \frac{1}{1 + \exp(-(H_1)_i)}$$

$$\mathbf{W}_{O1} = J\mathbf{A}^{+}$$

- Here, J is the target vectors. In most ELMs, this will be a K dimensional vector, where K is equal to the number of classes in the training dataset.

Deep ELM Architecture

University of South Australia



- Instead of performing classification at the output layer, we train the output weight matrix such that the target outputs are auto-encoded versions of the inputs

$$H_1 = \mathbf{W}_{P1} I$$

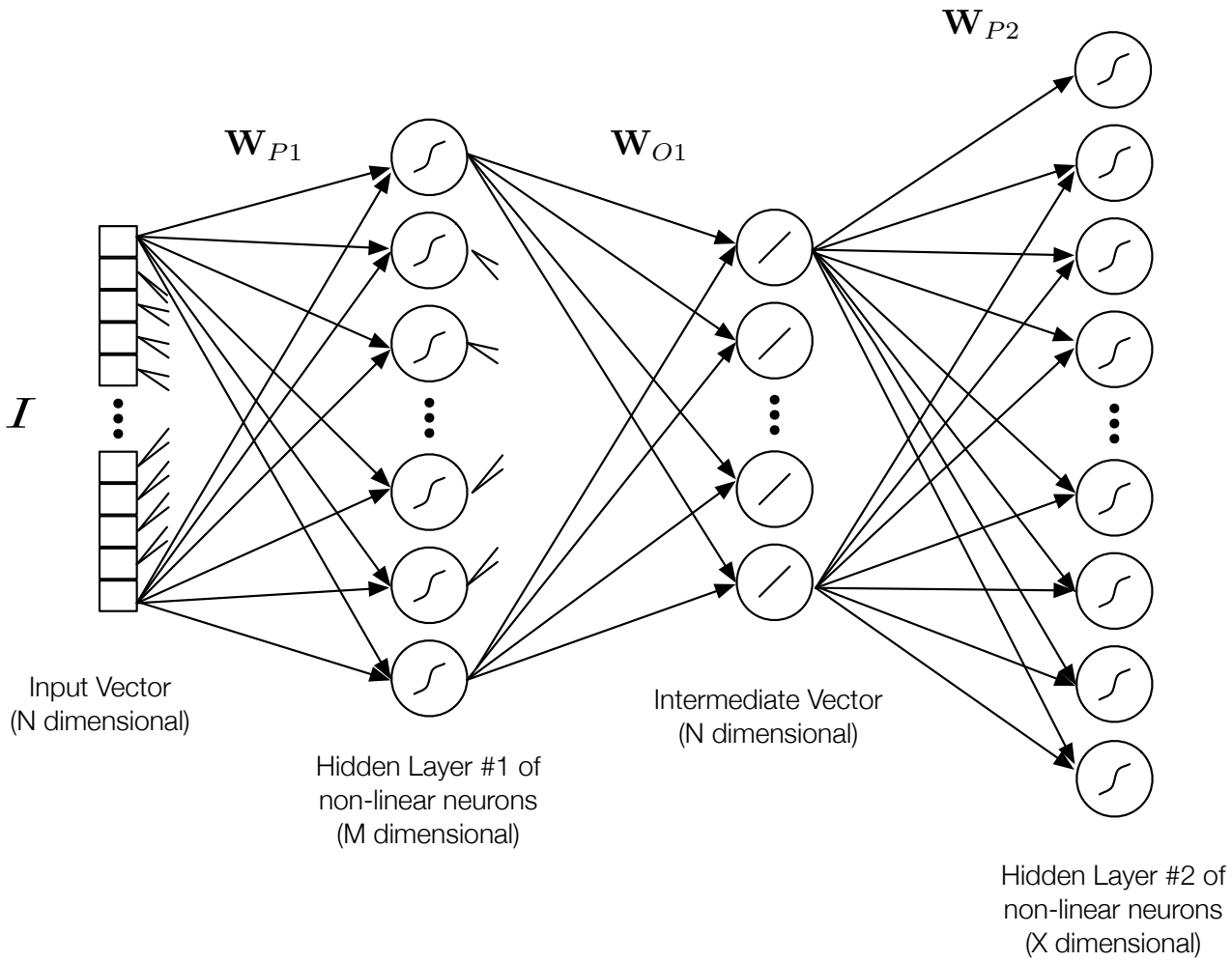
$$f[H_1]_i = \frac{1}{1 + \exp(-(H_1)_i)}$$

$$\mathbf{W}_{O1} = \hat{I}\mathbf{A}^{+}$$

Note our target has now changed

Deep ELM Architecture

University of South Australia

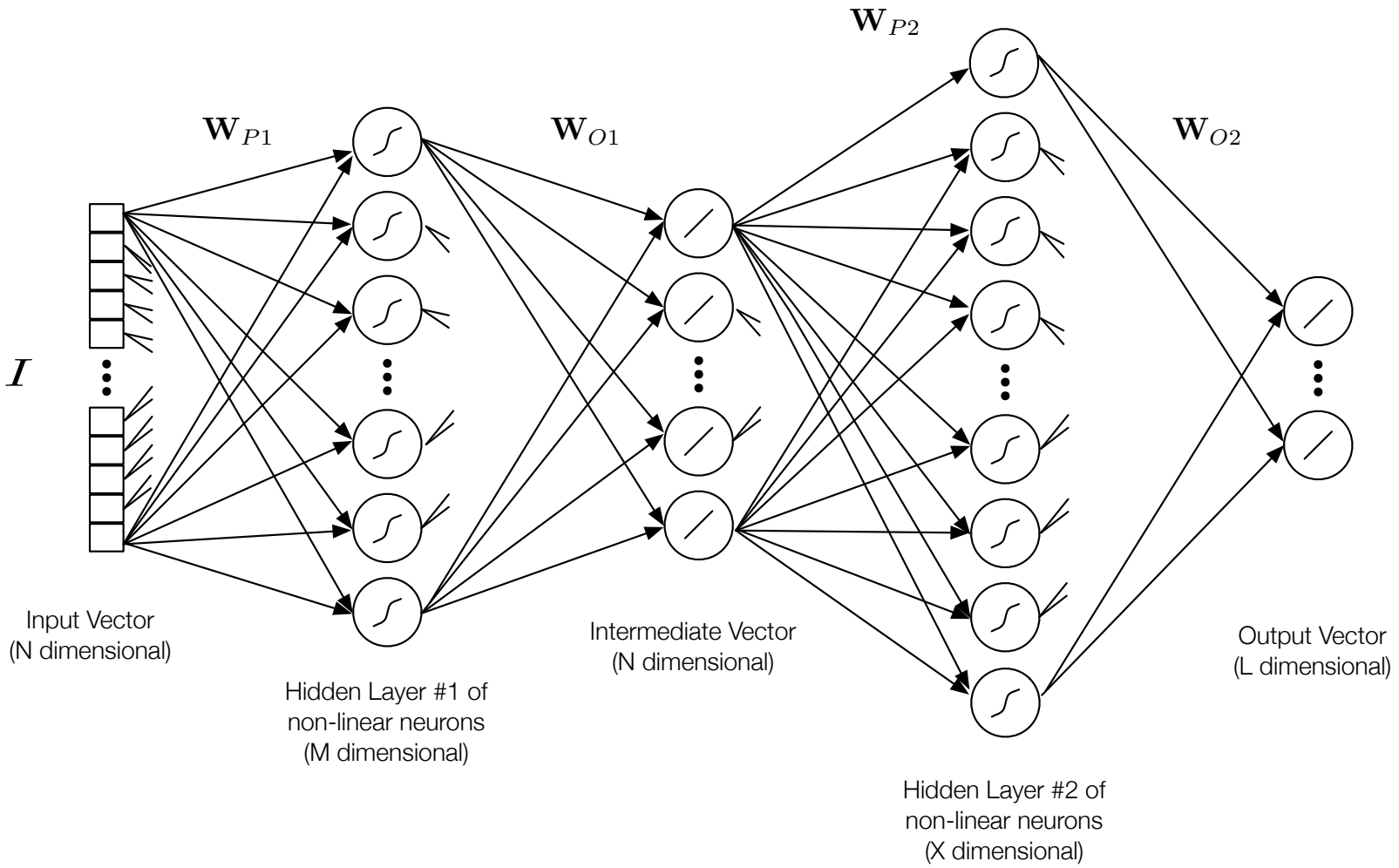


- Compute the pseudoinverse of the second hidden layer, which we denote as $\mathbf{B}^{+}$

$$H_2 = \mathbf{W}_{P2}\hat{I} \qquad f[H_2]_i = \frac{1}{1+\exp(-(H_2)_i)}$$

Deep ELM Architecture

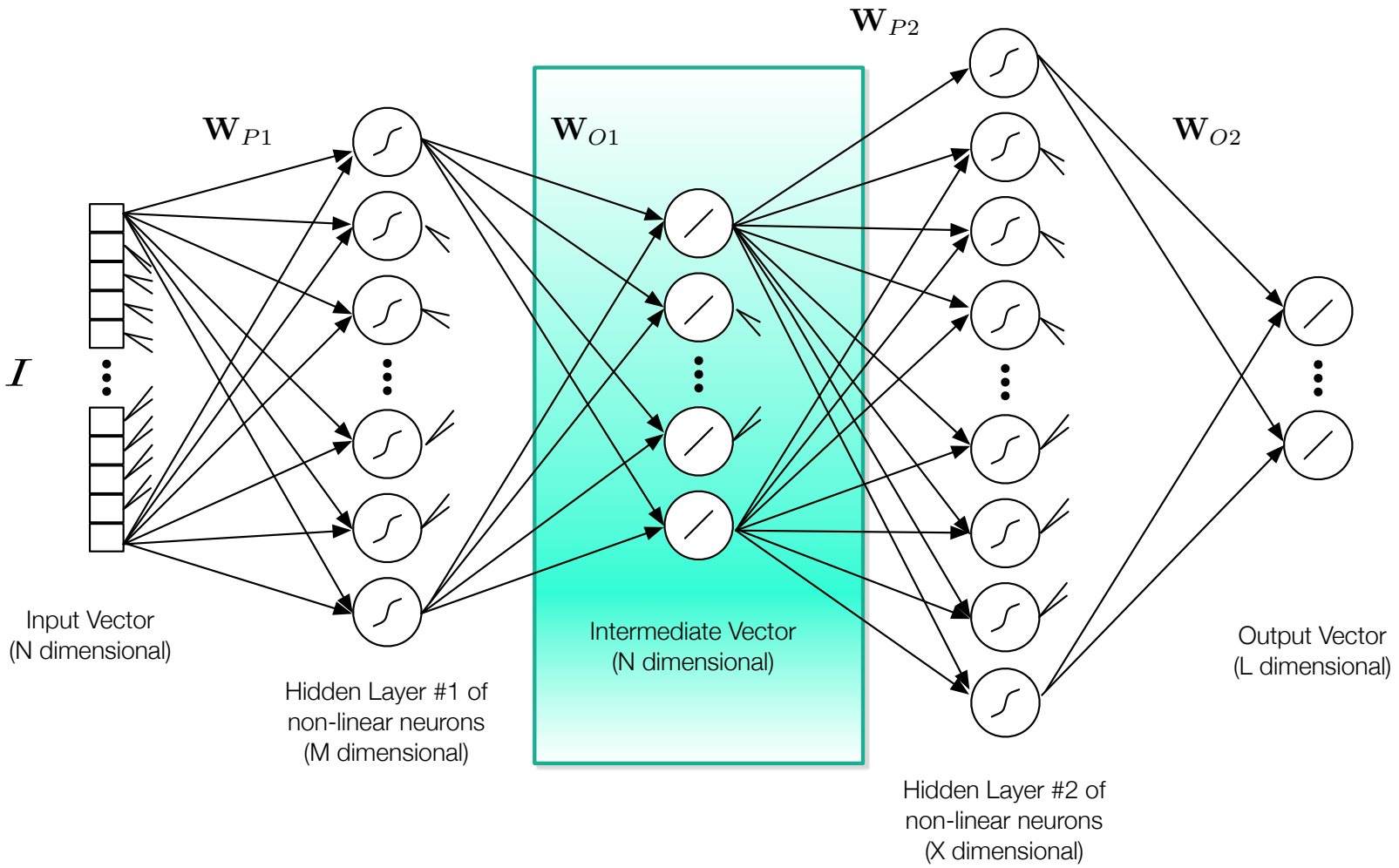
University of South Australia



$$\mathbf{W}_{O2} = J\mathbf{B}^{+}$$

Deep ELM Architecture

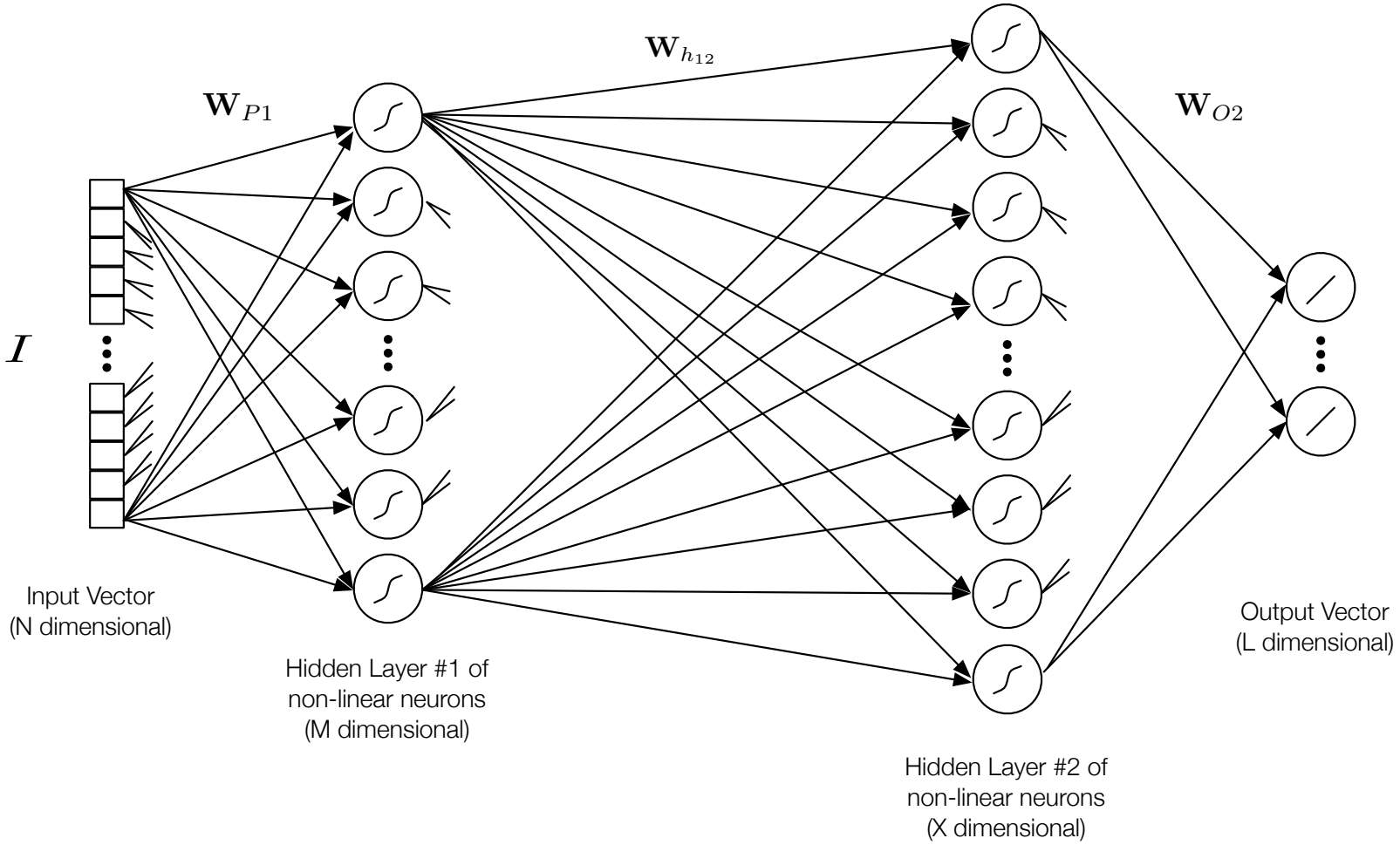
University of South Australia



$$\mathbf{W}_{h_{12}} = \mathbf{W}_{O1} \times \mathbf{W}_{P2}$$

Deep ELM Architecture

University of South Australia

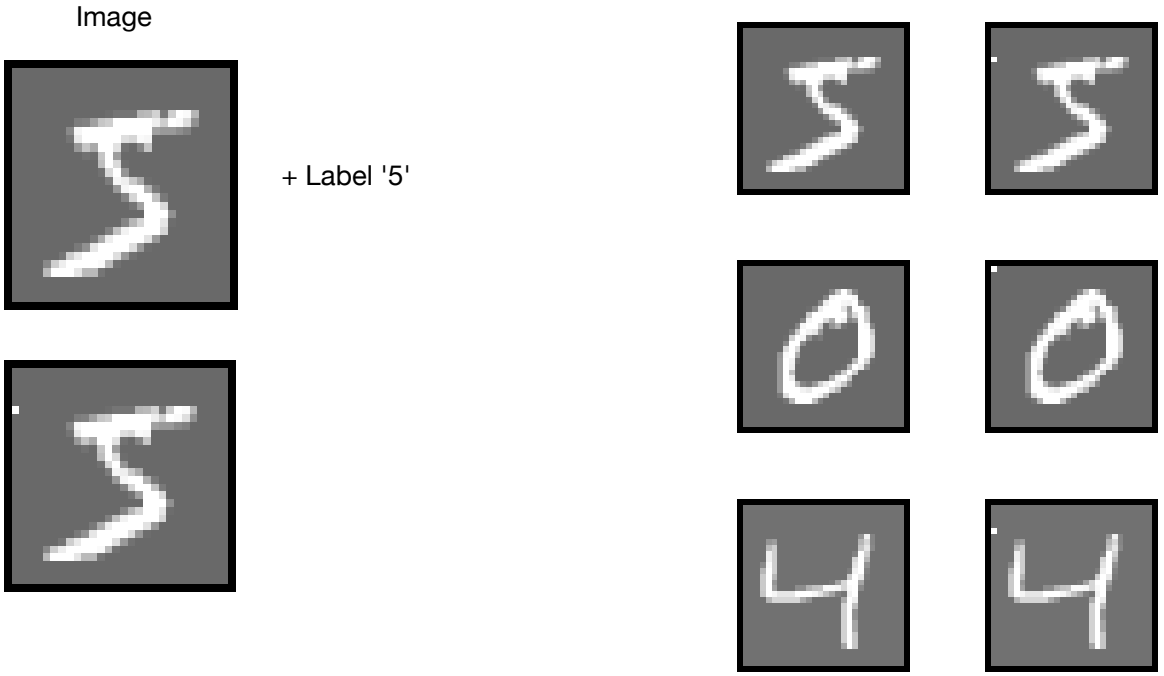


$$\mathbf{W}_{h_{12}} = \mathbf{W}_{O1} \times \mathbf{W}_{P2}$$

Adding supervision to the network

University of South Australia

- If we do this purely using the training images (without labels), the advantages become unclear. So what can we do?
- Supervised Auto-Encoding
- We add training labels combined with training images to create a single training dataset.



Weight Shaping Methods

University of South Australia

- In standard ELM approach, the input weights are initialised randomly. For example between -0.5 and +0.5
- Methods
  - Computed Input Weights for ELM (CIW-ELM)[1]
  - Receptive Fields (RF-ELM)[2]
  - Combined CIW and RF ELM

[1]. J. Tapson, P. de Chazal, and A. van Schaik, "Explicit computation of input weights in Extreme Learning Machines," Proceedings of ELM2014, Accepted, vol. arXiv:1406.2889, 2014.

[2]. M. D. McDonnell, M. D. Tissera, A. van Schaik, and J. Tapson, "Do we need deep networks? Pushing the envelope with Extreme Learning Machines," Submitted, 2014.

Experiments

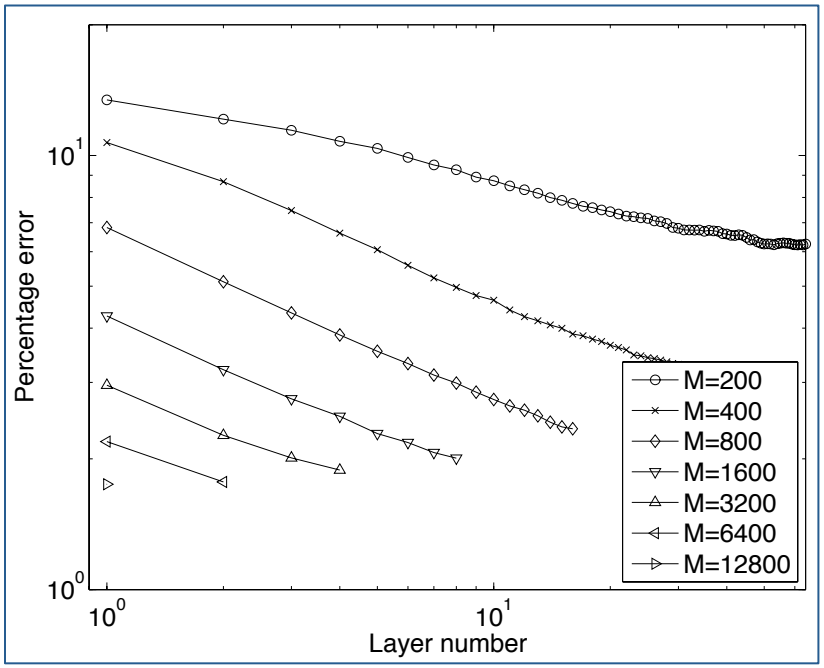
University of South Australia

- Image Classification
  - MNIST (10 Classes)
    - 60,000 Training and 10,000 Test Images
    - 28 x 28 Greyscale
  - CIFAR-10 (10 Classes)
    - 50,000 Training and 10,000 Test Images
    - 32 x 32 RGB, treated as one image in our experiments, 32 x 32 x 3
  - SVHN (Google Street View House Numbers)
    - 73,257 Training images used
    - 32 x 32 RGB, converted to greyscale in our experiments, 32 x 32

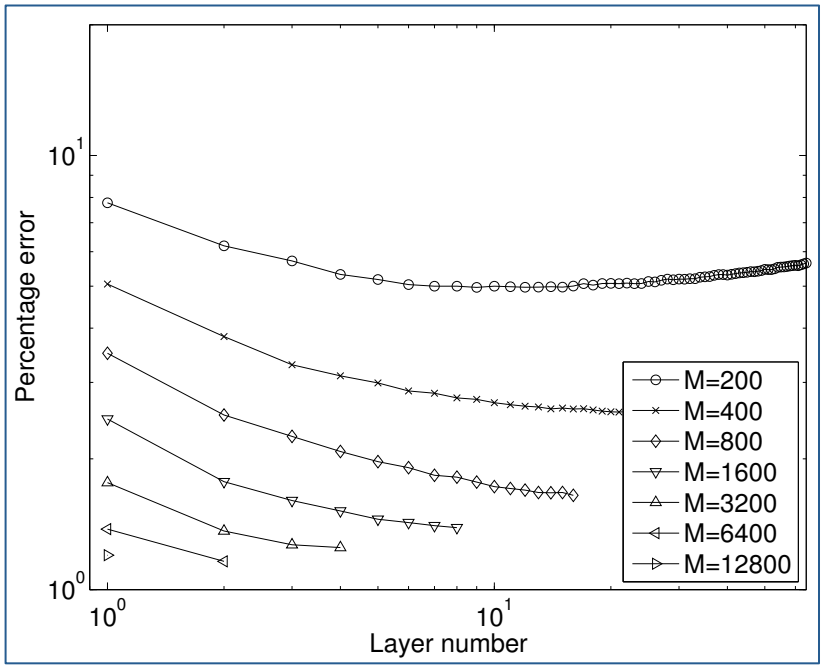
Results - MNIST

University of South Australia

Classification error rates on MNIST for increasing number of hidden layers



Random Input Weights
(Gaussian)



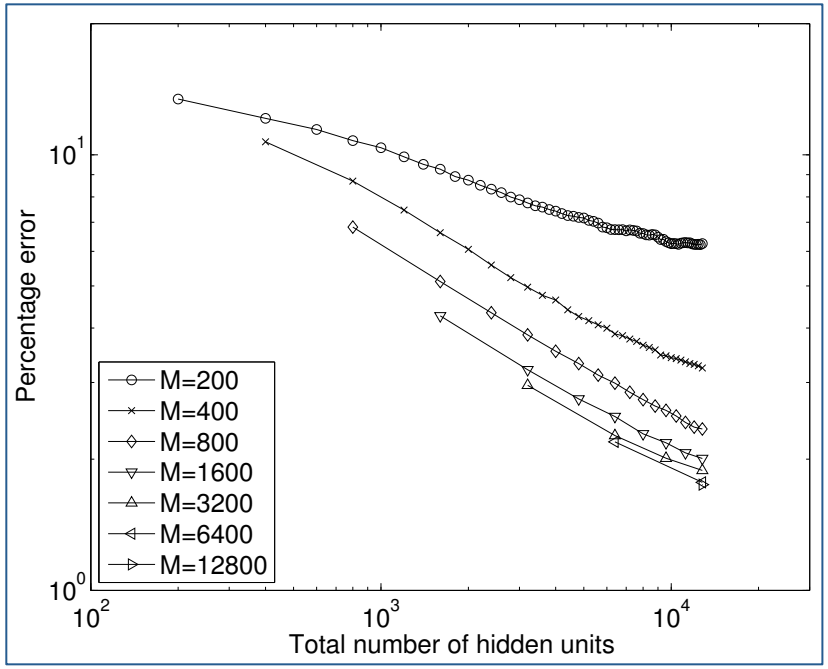
Shaped Input Weights
(CIW and RF Combined)

[1]. All data was calculated as the average over 10 repeats for each condition.

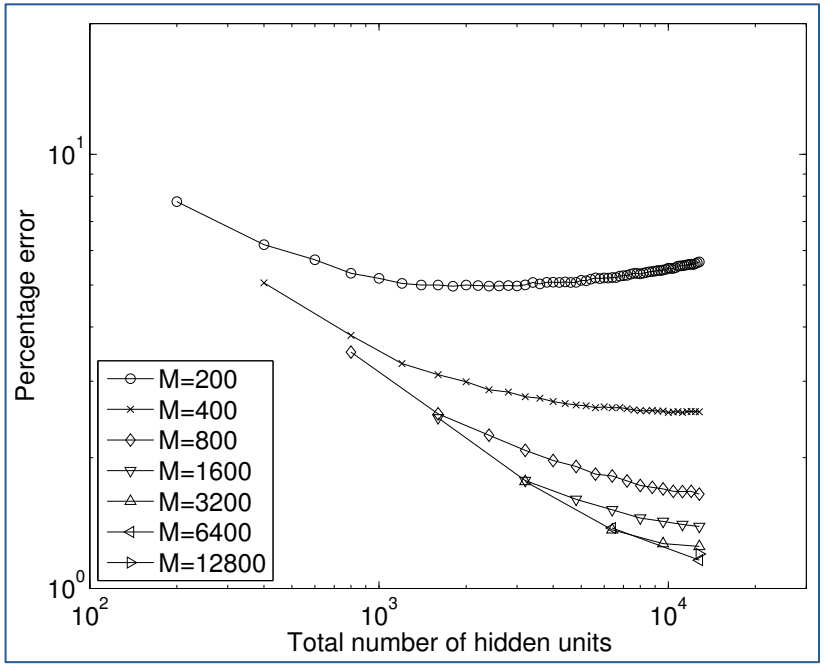
Results - MNIST

University of South Australia

Classification error rates on MNIST for increasing total number of hidden units



Random Input Weights
(Gaussian)



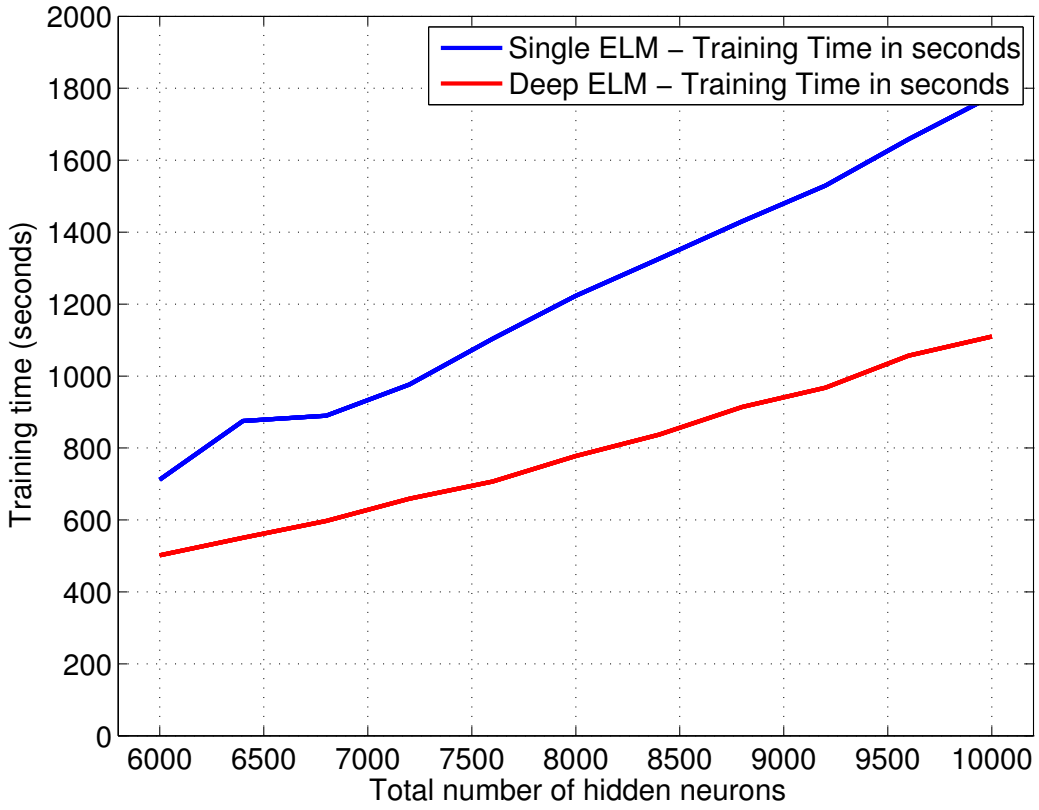
Shaped Input Weights
(CIW and RF Combined)

[1]. All data was calculated as the average over 10 repeats for each condition.

Results - MNIST

University of South Australia

Training time on MNIST as the total number of hidden units increased

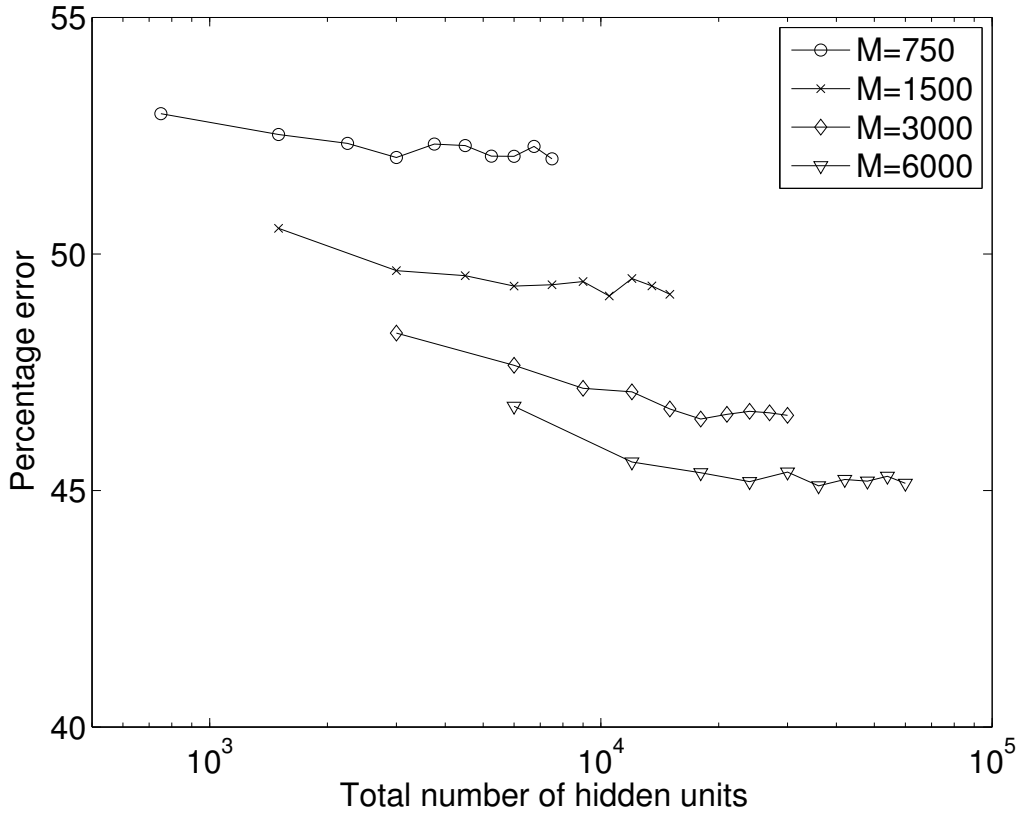


Deep ELM Vs Single Layer ELM

Results – CIFAR 10

University of South Australia

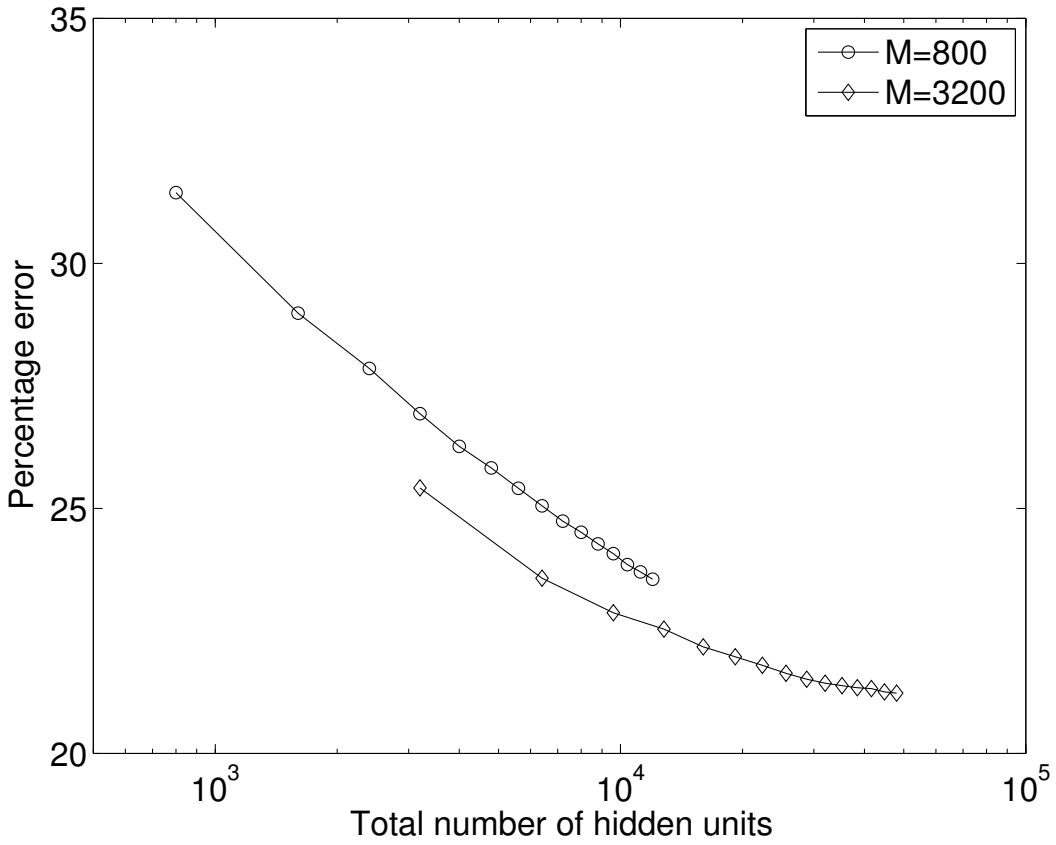
Classification error rates on CIFAR-10 for increasing total number of hidden units



Results – SVHN

University of South Australia

Classification error rates on SVHN dataset for increasing total number of hidden units



Discussion

University of South Australia

- We have presented a method for synthesising deep neural networks using Extreme Learning Machines (ELMs) as a stack of supervised autoencoders.
- Using standard multi-class image classification benchmark tasks, we show that the classification error rate can progressively improve with inclusion of additional layers.
- Very good performance on MNIST in very short time
  - 98% accuracy in less than 2 minutes
  - 99% accuracy in less than 10 minutes
- The method can potentially be applied in a resource-constrained hardware implementation to advantages in terms of significantly reduced network training time and memory usage.

University of South Australia

Institute for Telecommunications Research

Questions?

University of South Australia

Institute for Telecommunications Research

Thank you!

Deep Extreme Learning Machines: Supervised Auto-Encoding Architecture for Classification

Migel D. Tissera
migel.tissera@mymail.unisa.edu.au