Transfer Learning on an Autoencoder-based Deep Network

Leandro F. Ariza Jiménez

Advisor: Olga Lucía Quintero, PhD.

Mathematical Modeling Research Group - GRIMMAT
School of Mathematical Sciences
Universidad EAFIT

September 6, 2016

Previous talk

- Big Data discussion
- Clustering algorithms overview
- Introduction to deep learning
- Future attendance to Cornell University workshop

Cornell-SJTU International Workshop

Cornell University
Department of Computer Science

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Research question

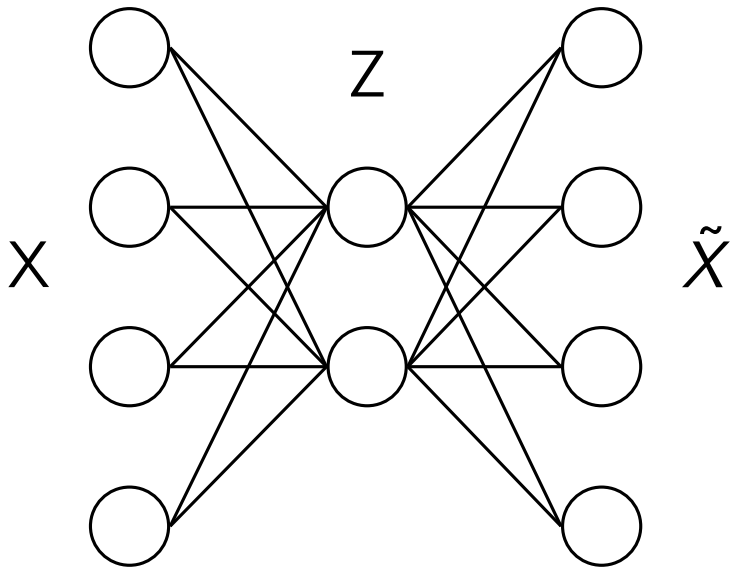
How well does a deep network work if we perform a classification task B based on what the deep network learned from a previous classification task A?

Project overview

- Two datasets of pictures (Digits and Leaves)
- Small deep network with two hidden layers
- Train each hidden layers using autoencoders
- Perform 3 experiments to conclude about

Autoencoders

Arquitecture



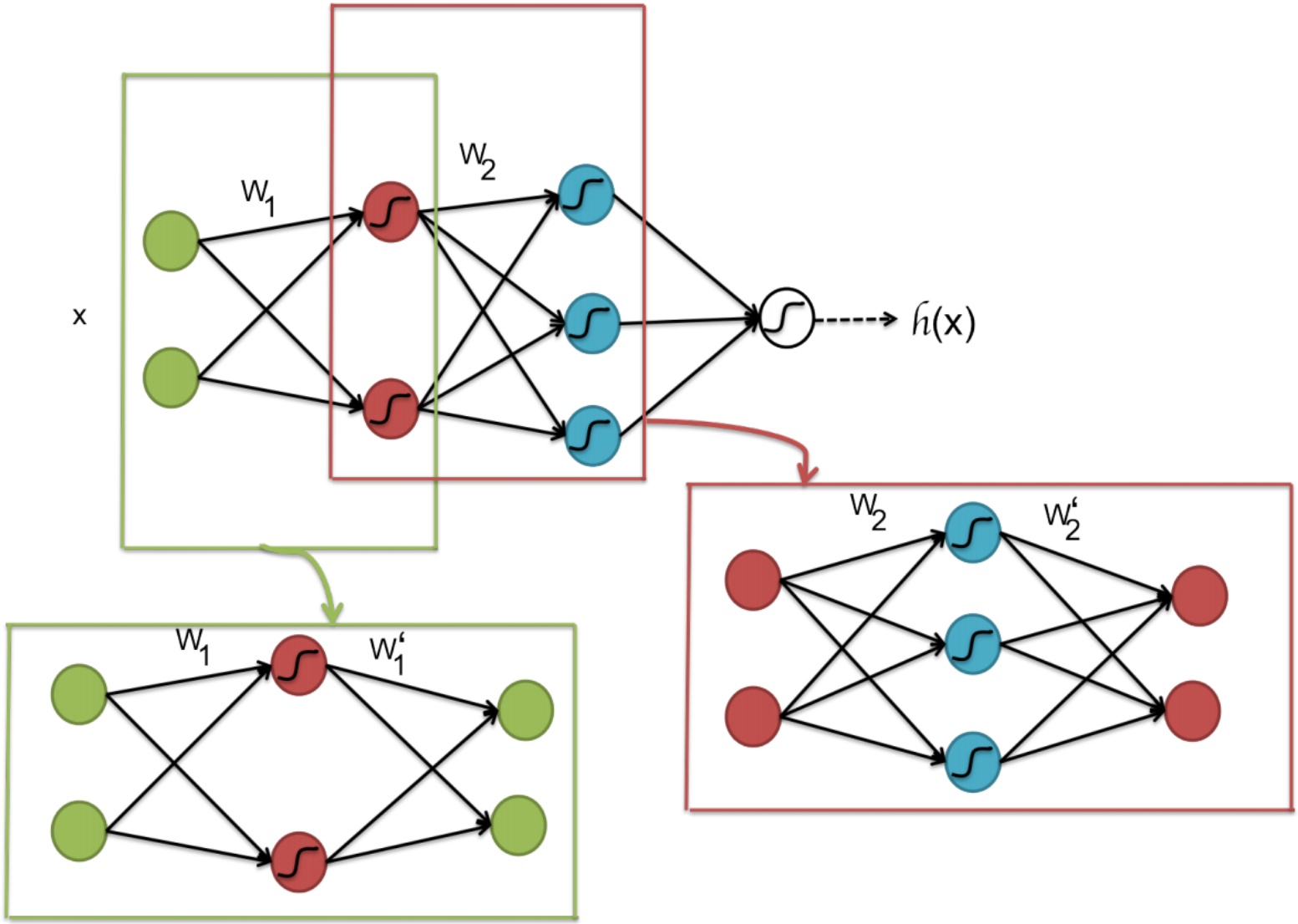
Model

$$z^{(i)} = W_1 x^{(i)} + b_1$$
$$\tilde{x}^{(i)} = W_2 z^{(i)} + b_2$$

Objective Function

$$J(W_1, b_1, W_2, b_2) = \sum_{i=1}^{m} \left(W_2 \left(W_1 x^{(i)} + b_1 \right) + b_2 - x^{(i)} \right)^2$$

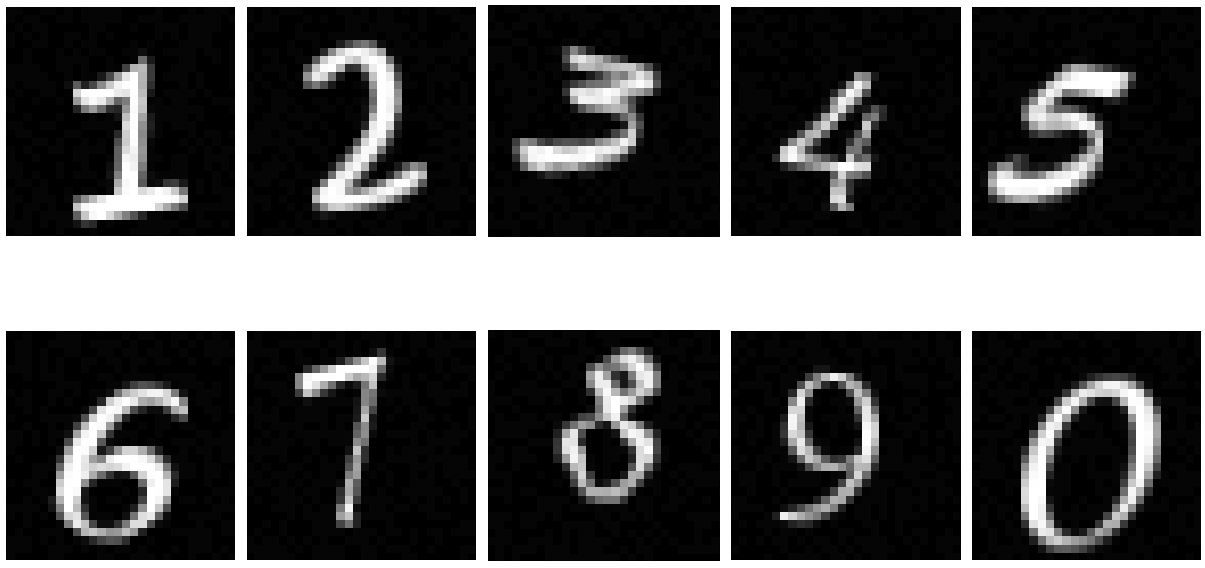
Autoencoders



Source:[3]

Digits dataset

MNIST database of handwritten digits [4]:



- Normalized greyscale images of 28x28 pixels
- Training Set (Tr. Set) with 5000 images (500 per digit approx.)
- Test Set (Te. Set) with 5000 images (500 per digit)

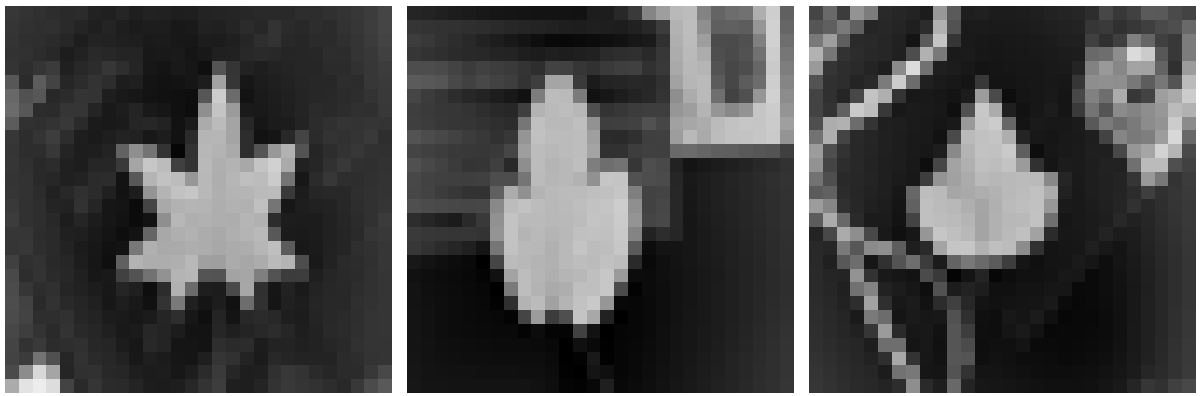
Leaves dataset

Leaves dataset taken by Markus Weber [5]:

- 186 RGB images of 896x592 pixels
- 66 images of type 1
- 60 images of both type 2 and type 3

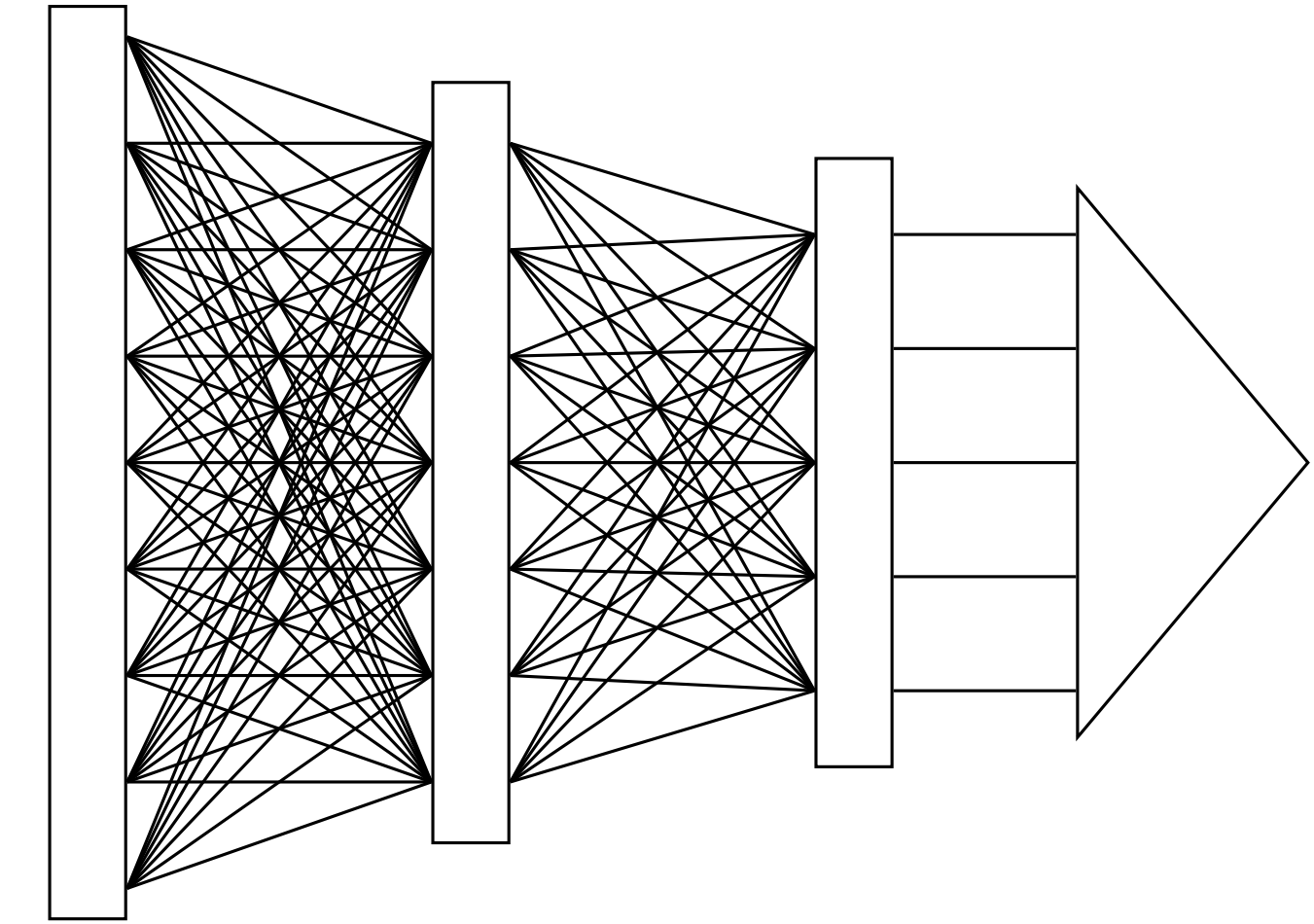
Leaves dataset - Pre-processed

Leaves dataset taken by Markus Weber [5]:

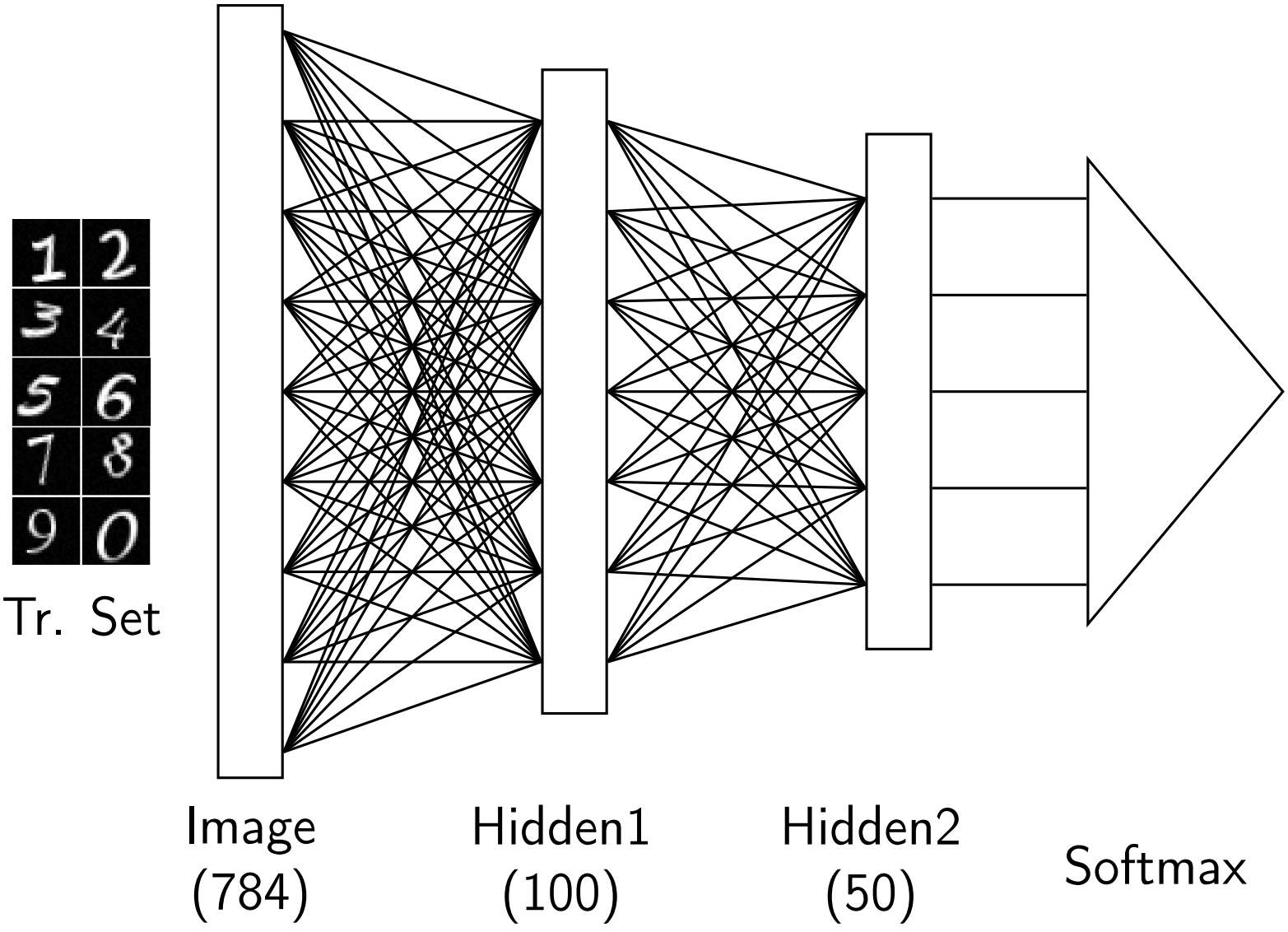


- Normalized and inverted greyscale images of 28x28 pixels
- Training Set (Tr. Set) with 120 images (40 per type)
- Test Set (Te. Set) with 60 images (20 per type)

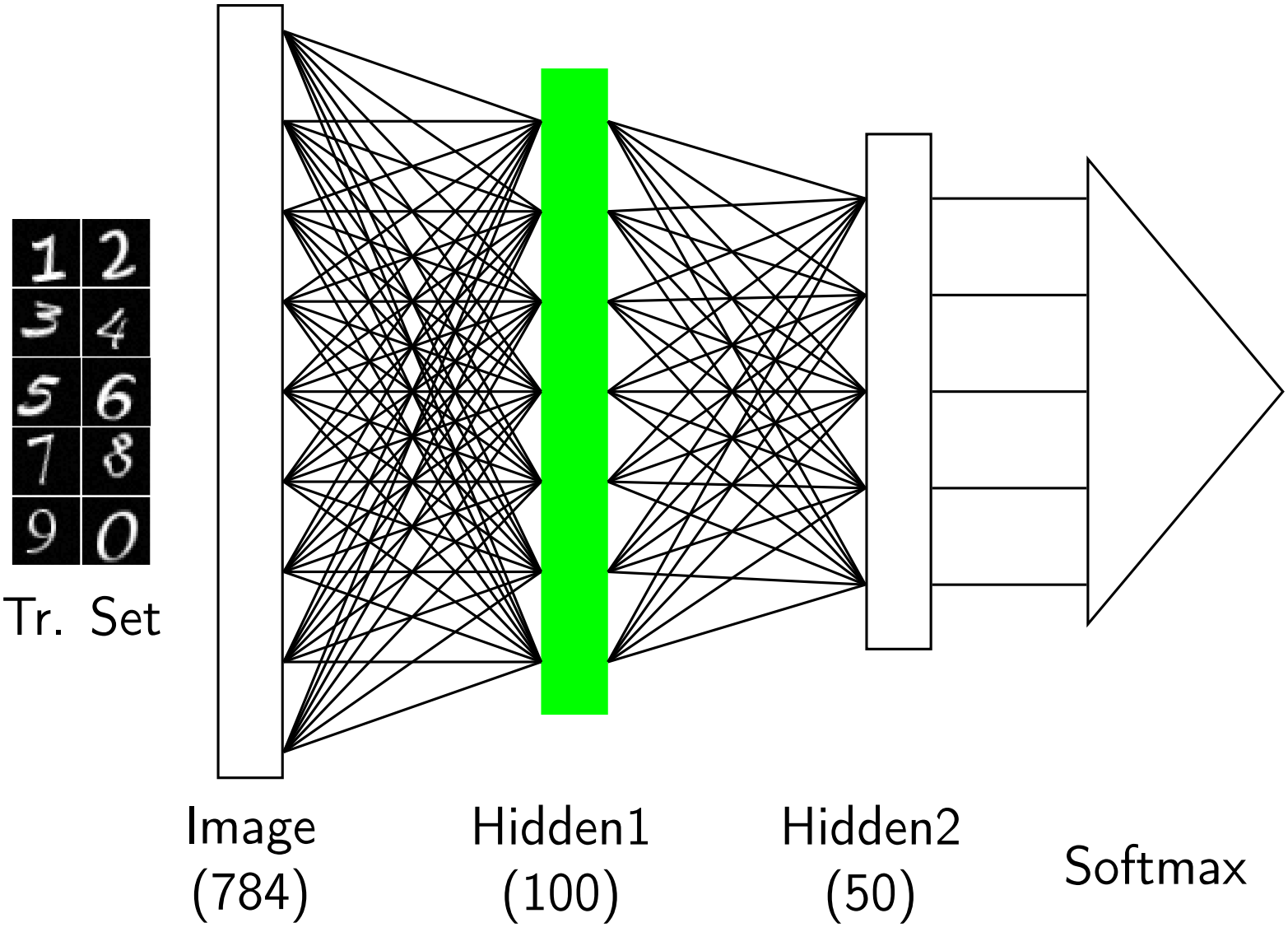
Proposed deep network



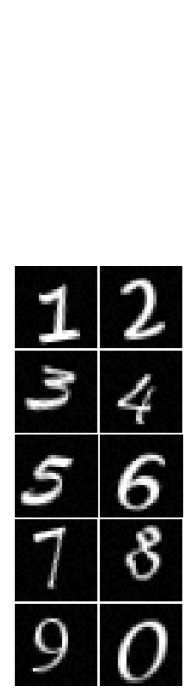
Exp. 1 - Digits dataset - Pre-training



Exp. 1 - Digits dataset - Pre-training



Exp. 1 - Digits dataset - Pre-training



Tr. Set

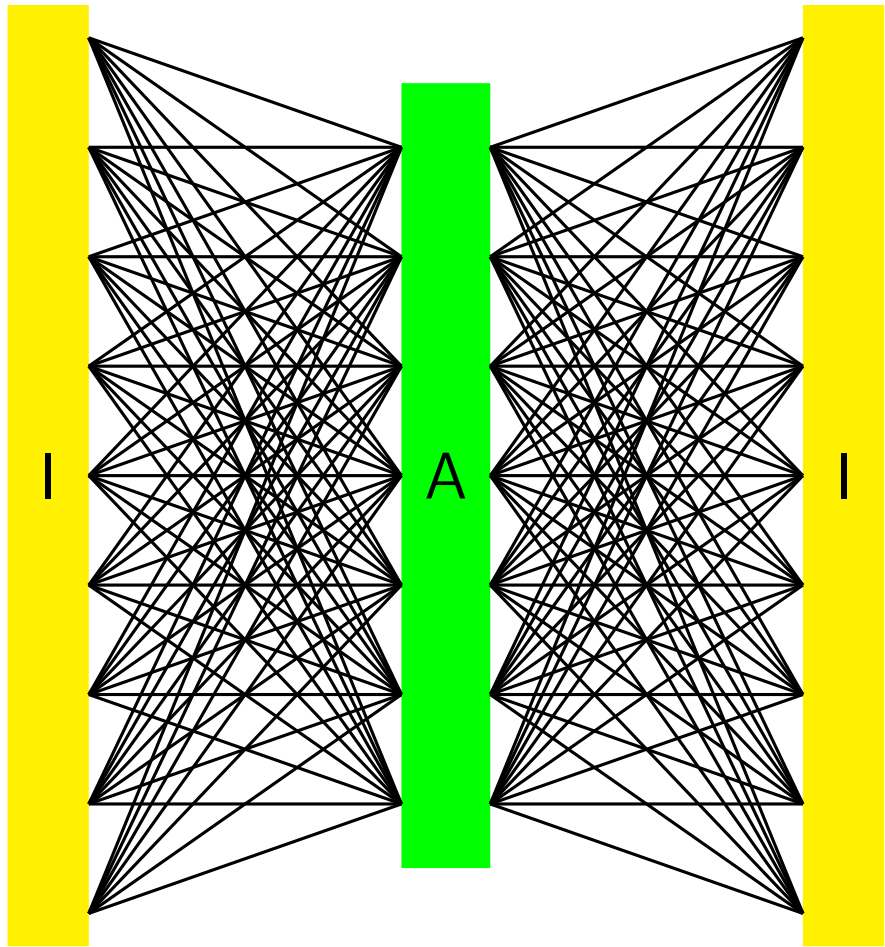
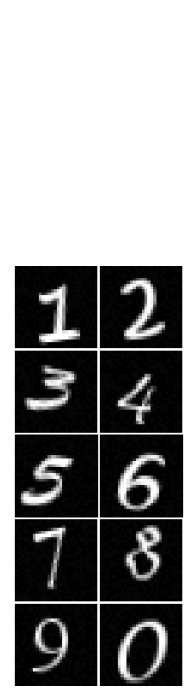


Image (784) Hidden1 (100) Image (784)

Exp. 1 - Digits dataset - Pre-training



Tr. Set

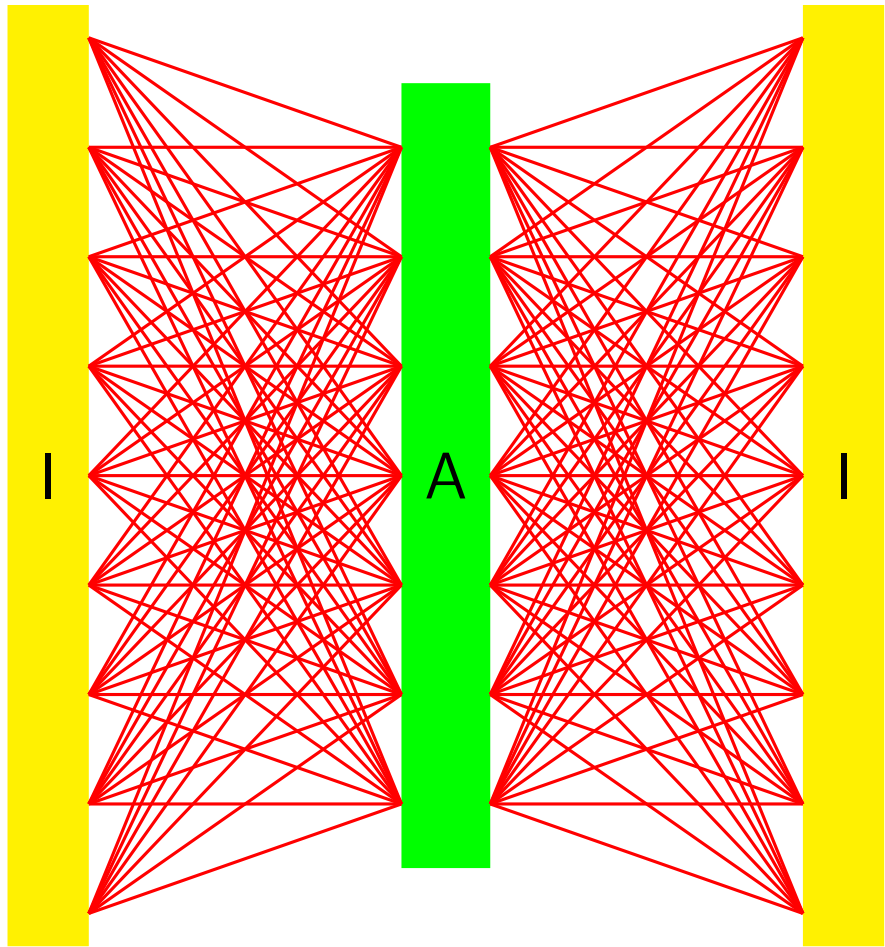
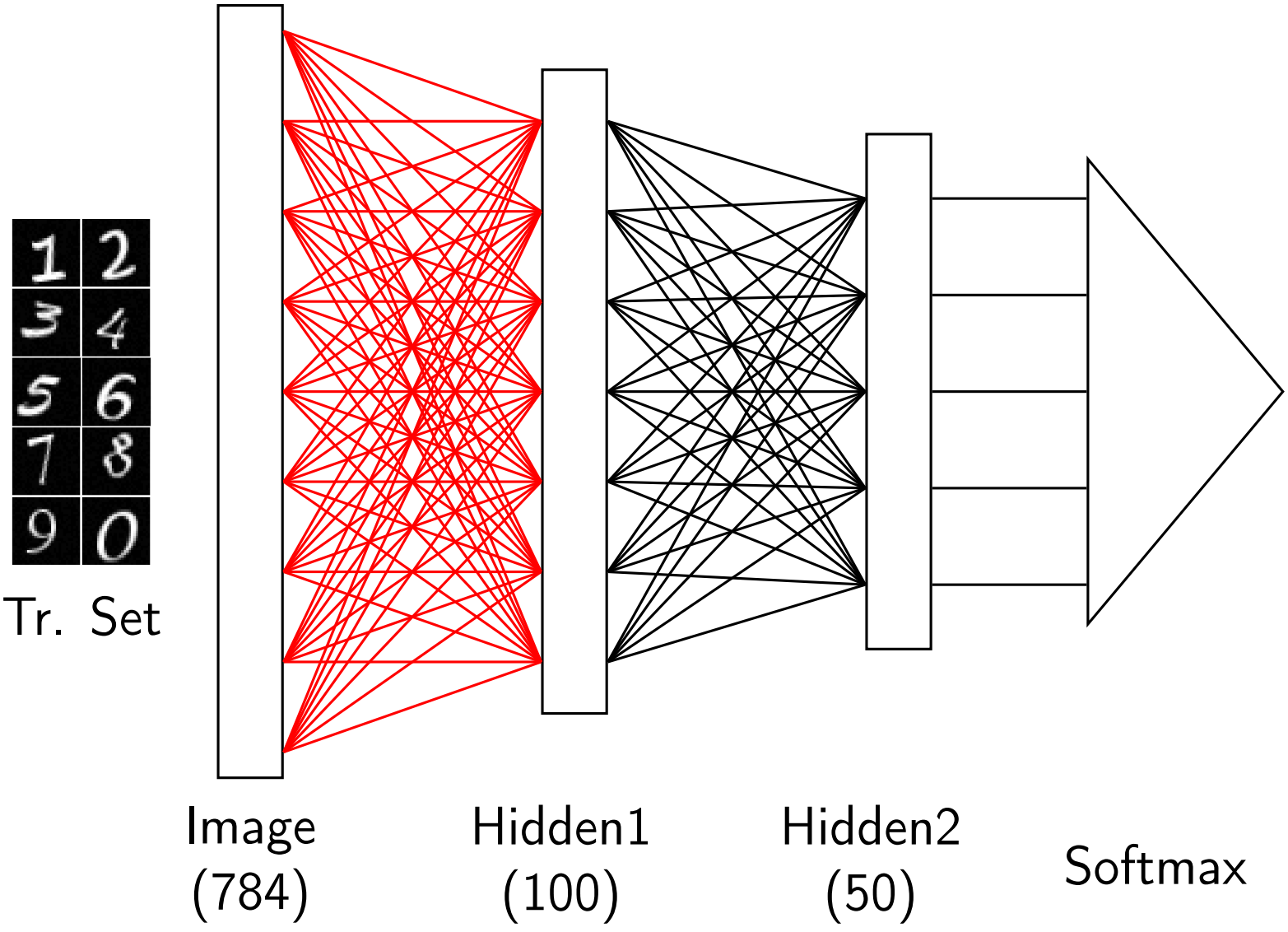
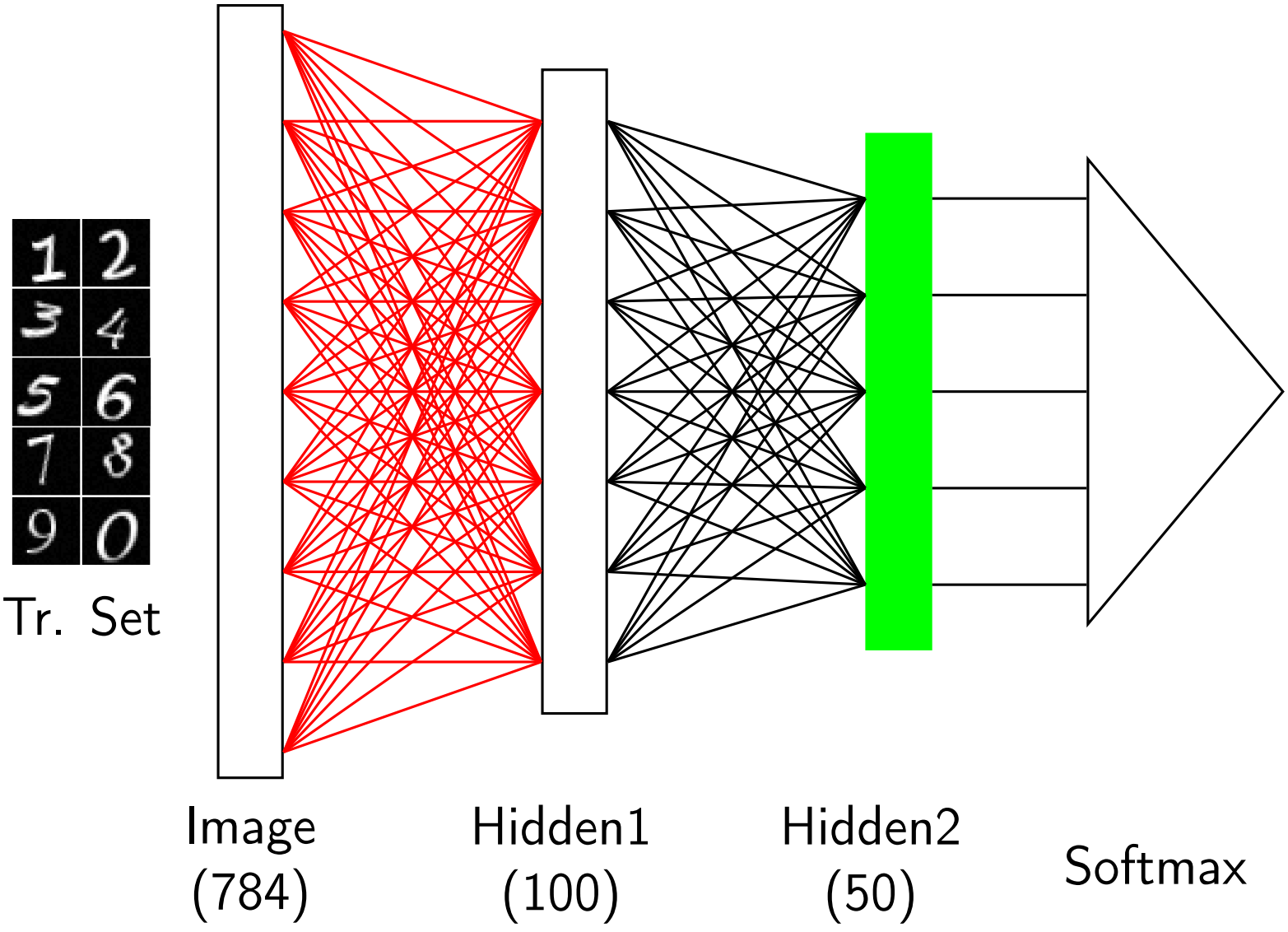


Image (784) Hidden1 (100) Image (784)

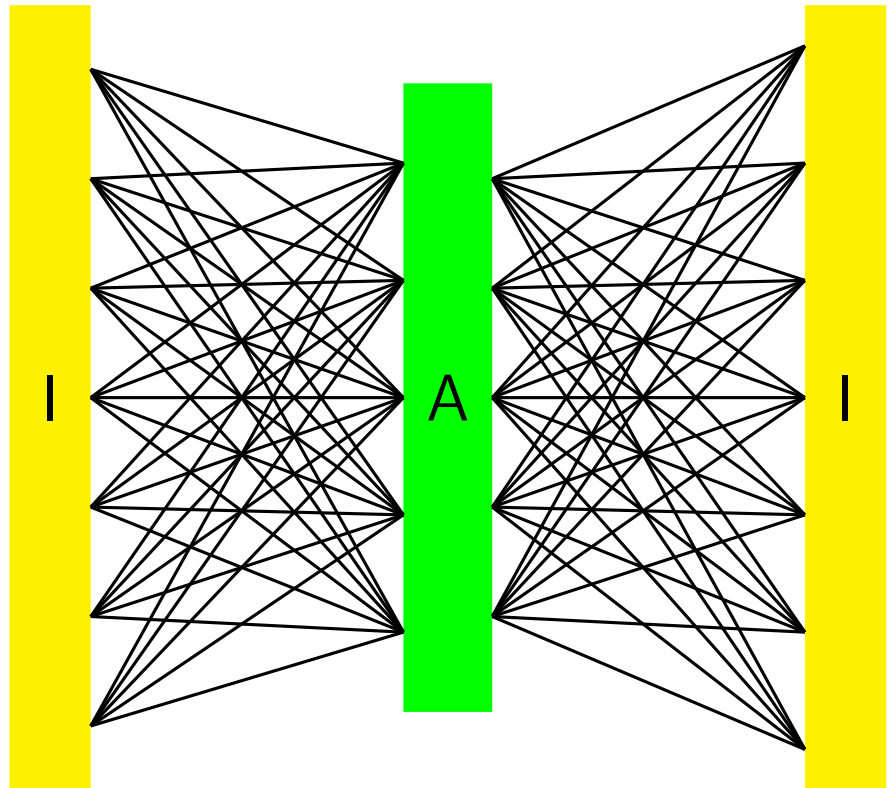
Exp. 1 - Digits dataset - Pre-training



Exp. 1 - Digits dataset - Pre-training

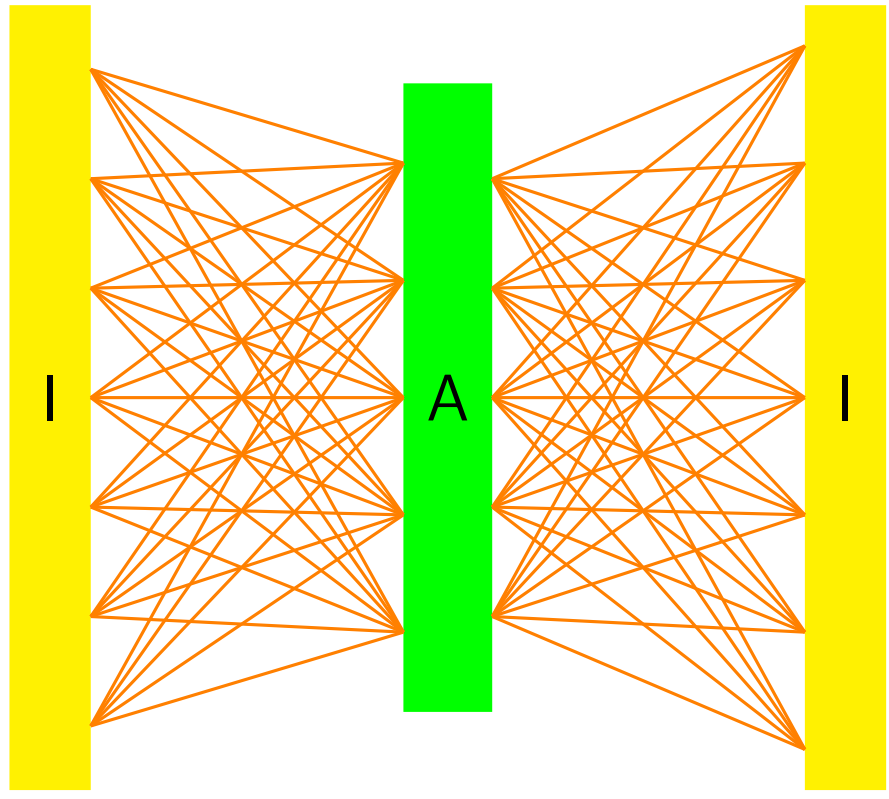


Exp. 1 - Digits dataset - Pre-training



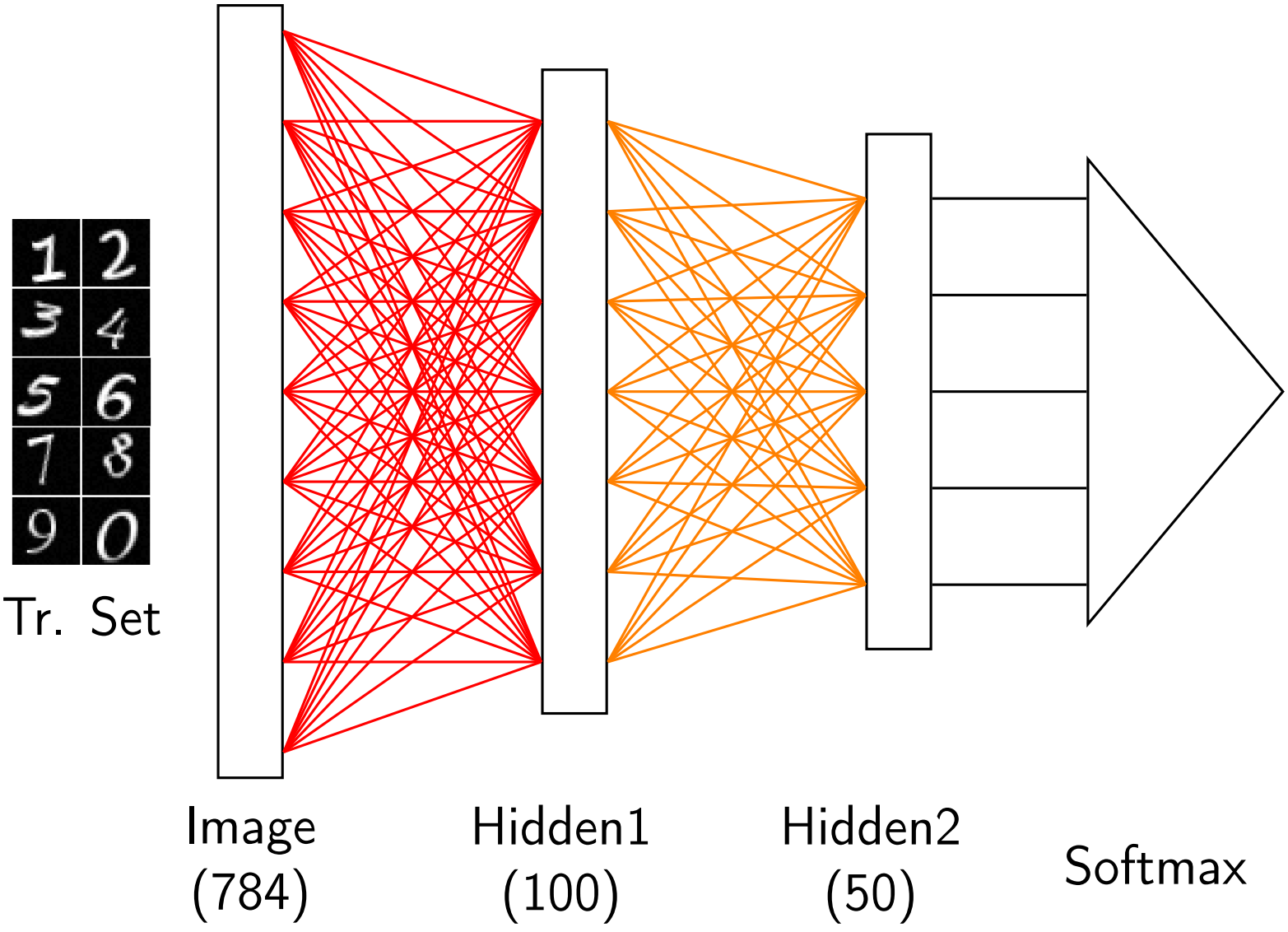
Hidden1 (100) Hidden2 (50) Hidden1 (100)

Exp. 1 - Digits dataset - Pre-training

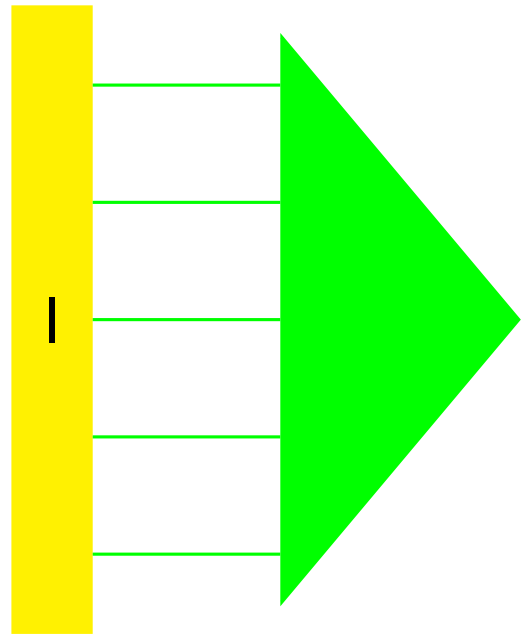


Hidden1 (100) Hidden2 (50) Hidden1 (100)

Exp. 1 - Digits dataset - Pre-training



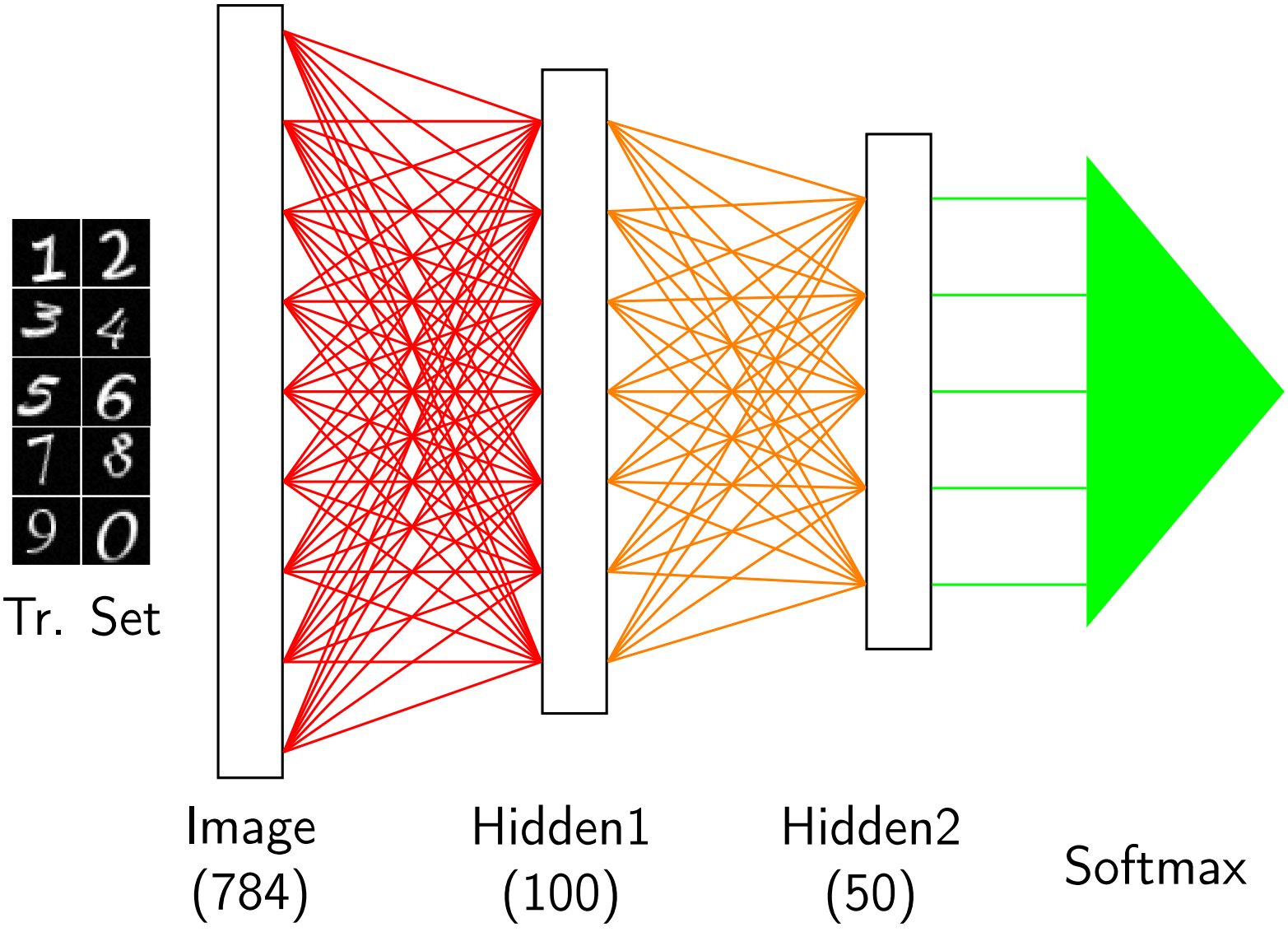
Exp. 1 - Digits dataset - Training



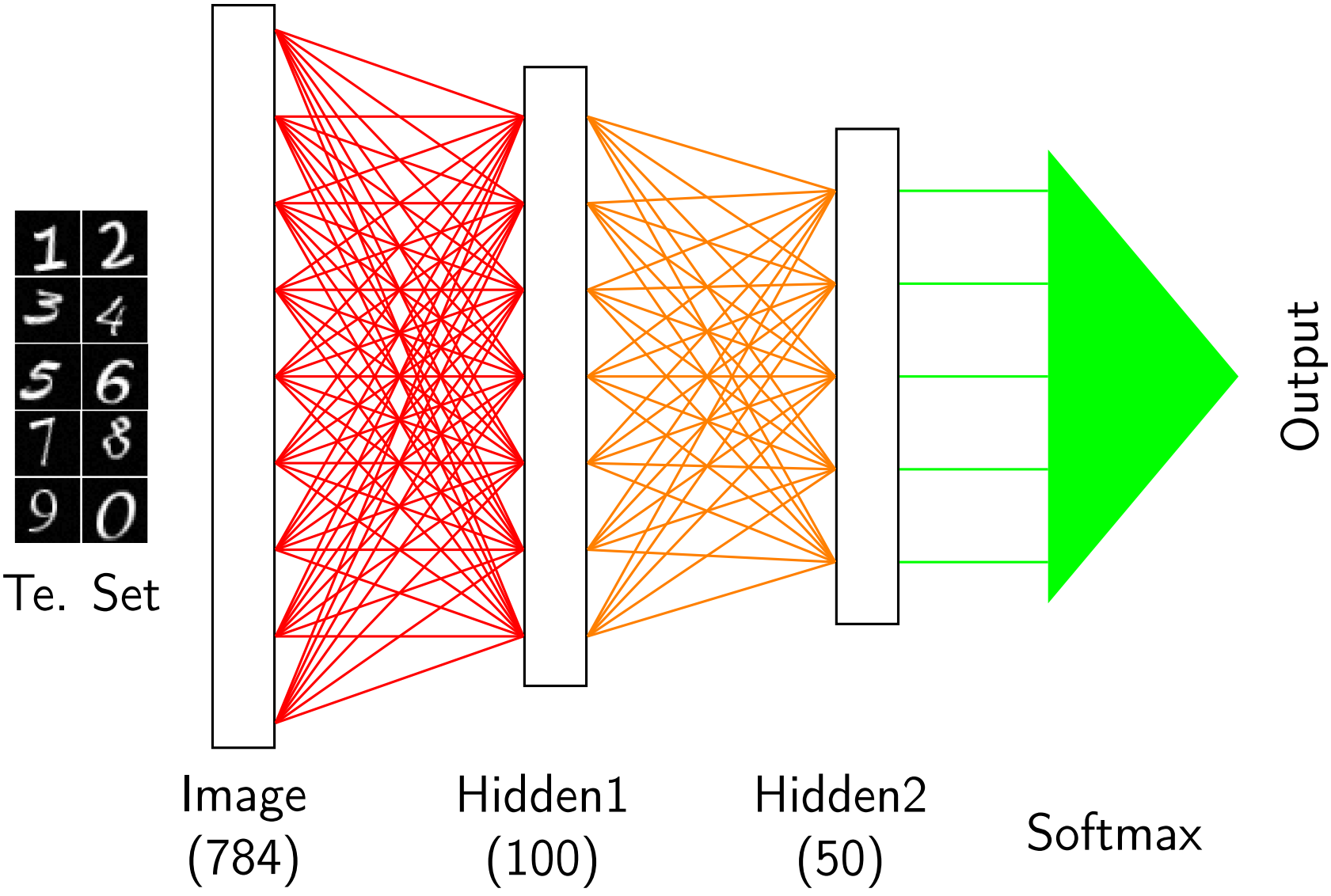
Hidden2
(50)

Softmax

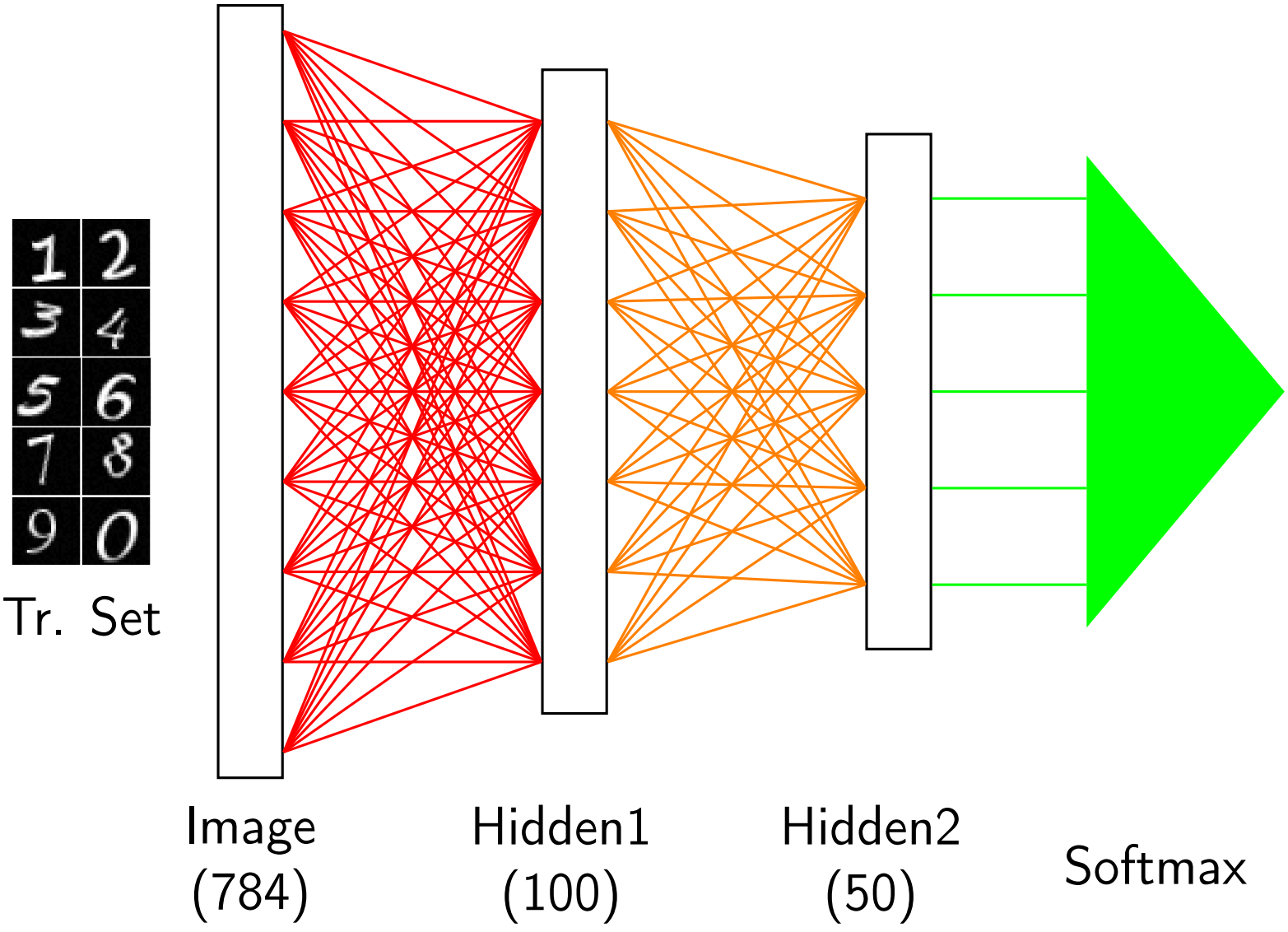
Exp. 1 - Digits dataset - Training



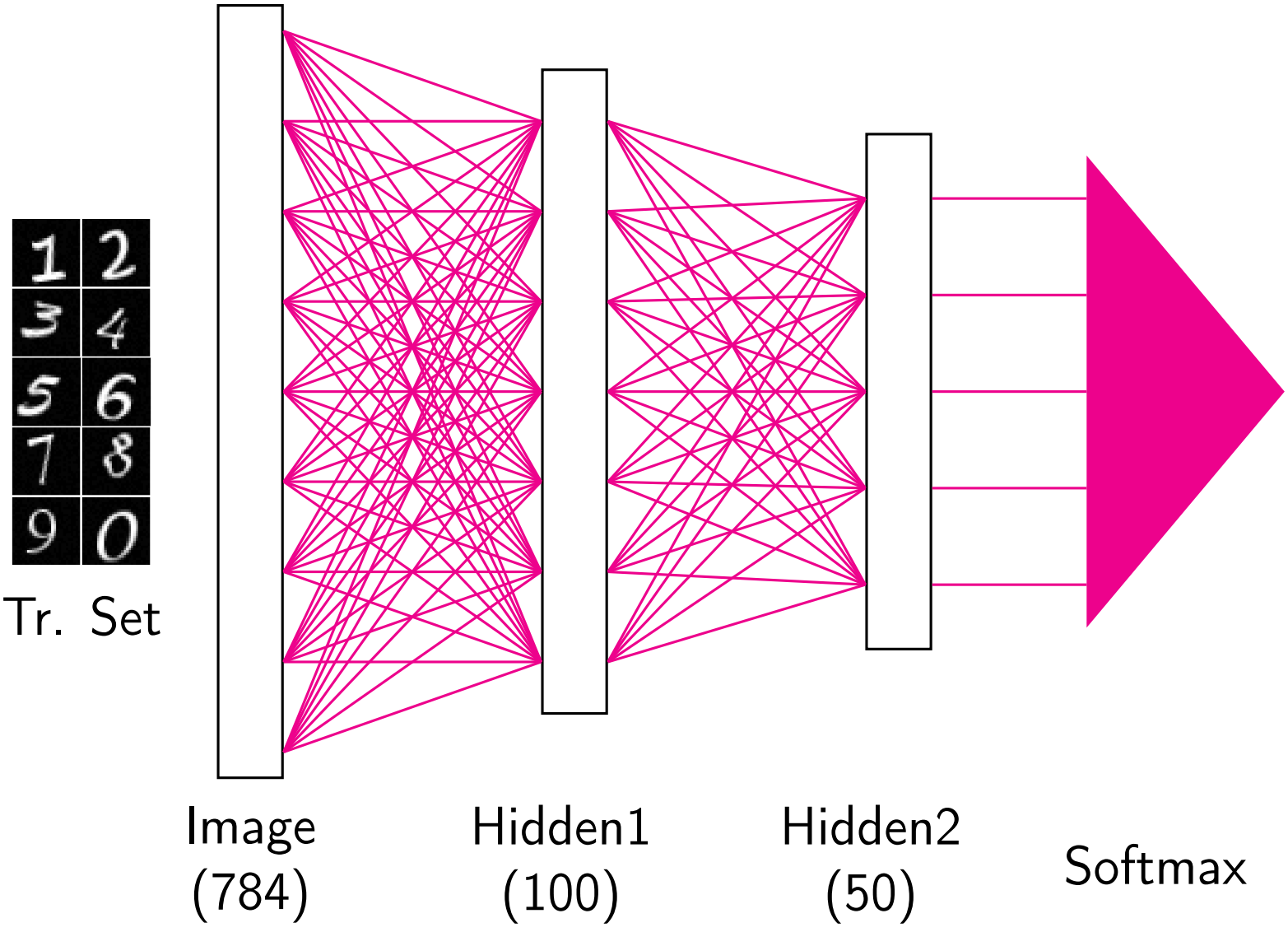
Exp. 1 - Digits dataset - Testing



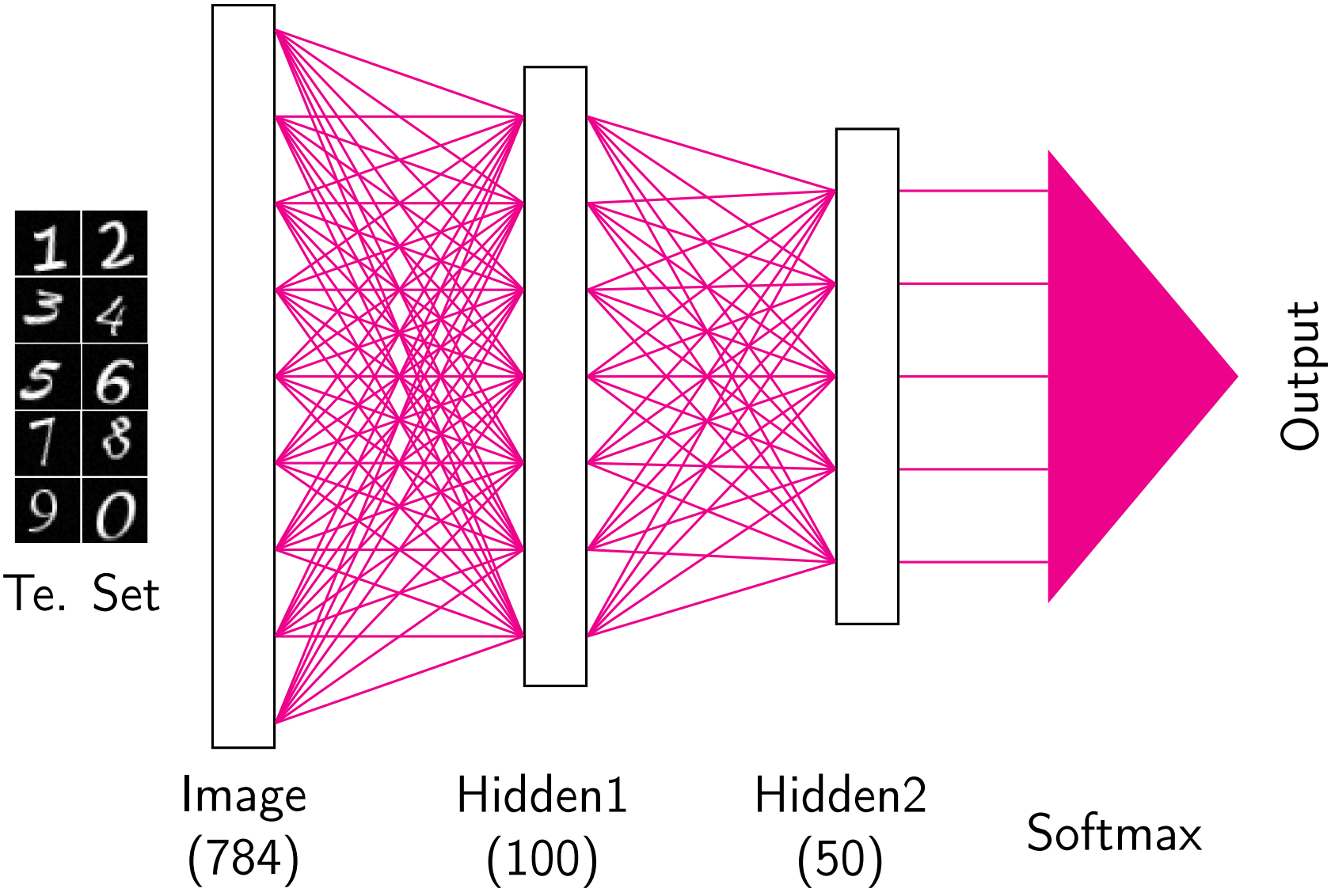
Exp. 1 - Digits dataset - Fine tuning



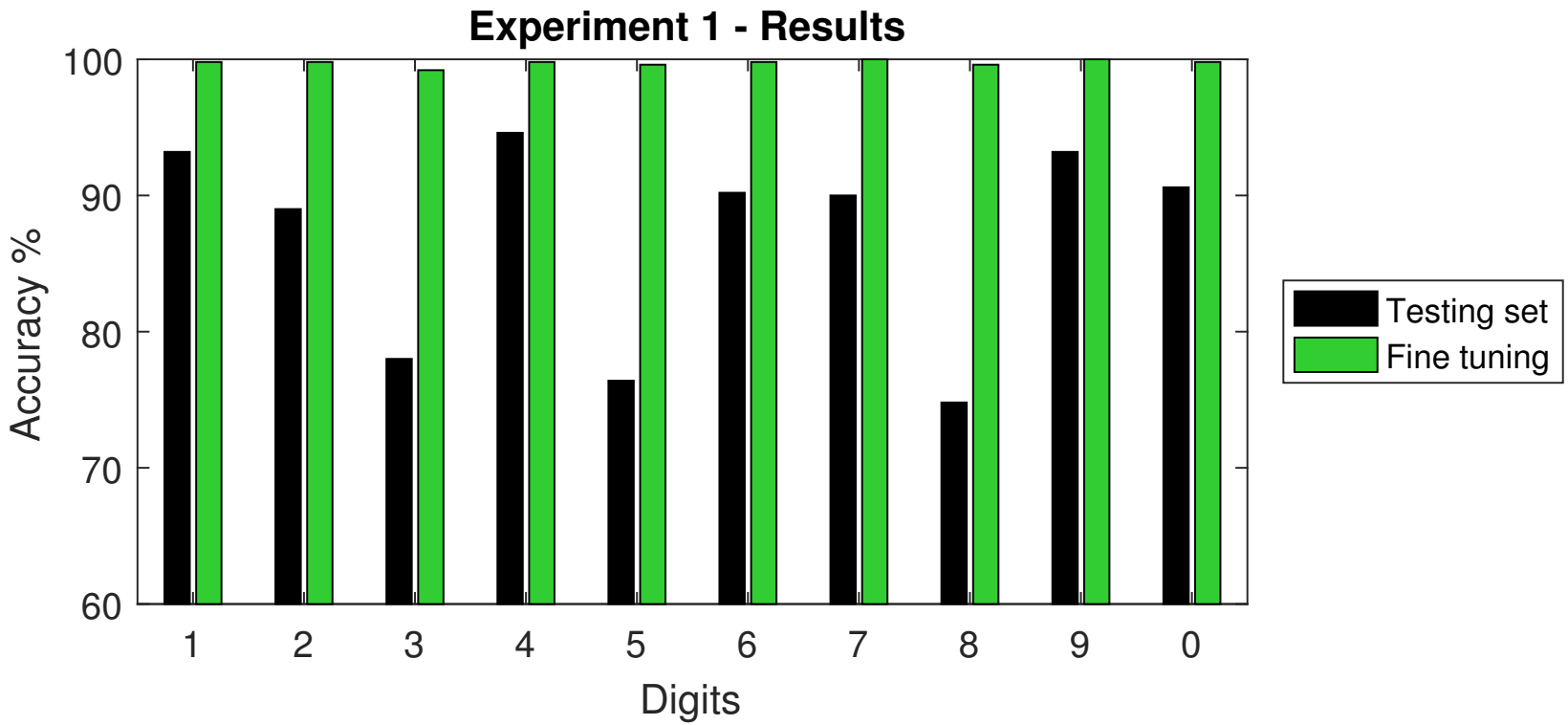
Exp. 1 - Digits dataset - Fine tuning



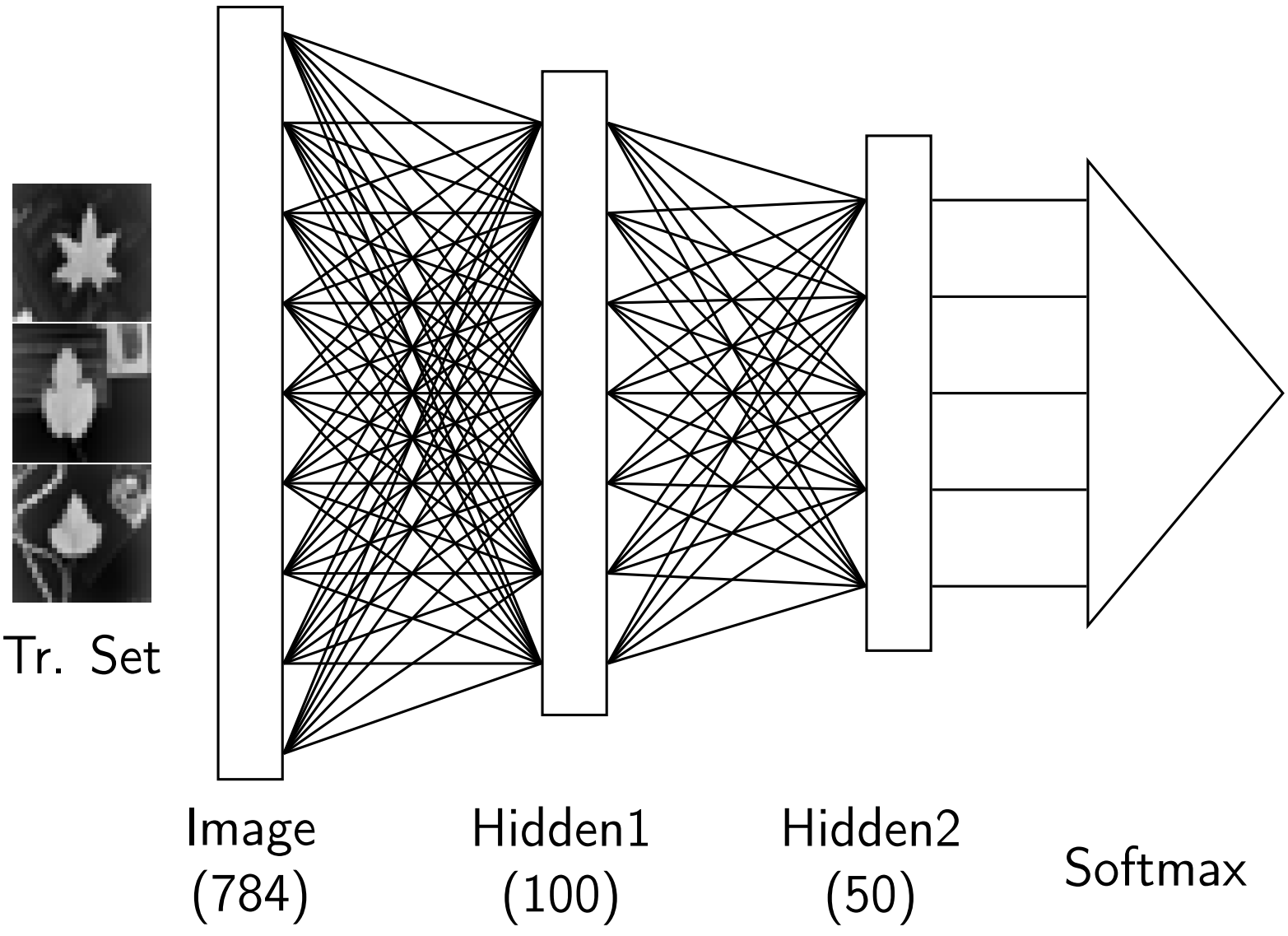
Exp. 1 - Digits dataset - Fine tuning



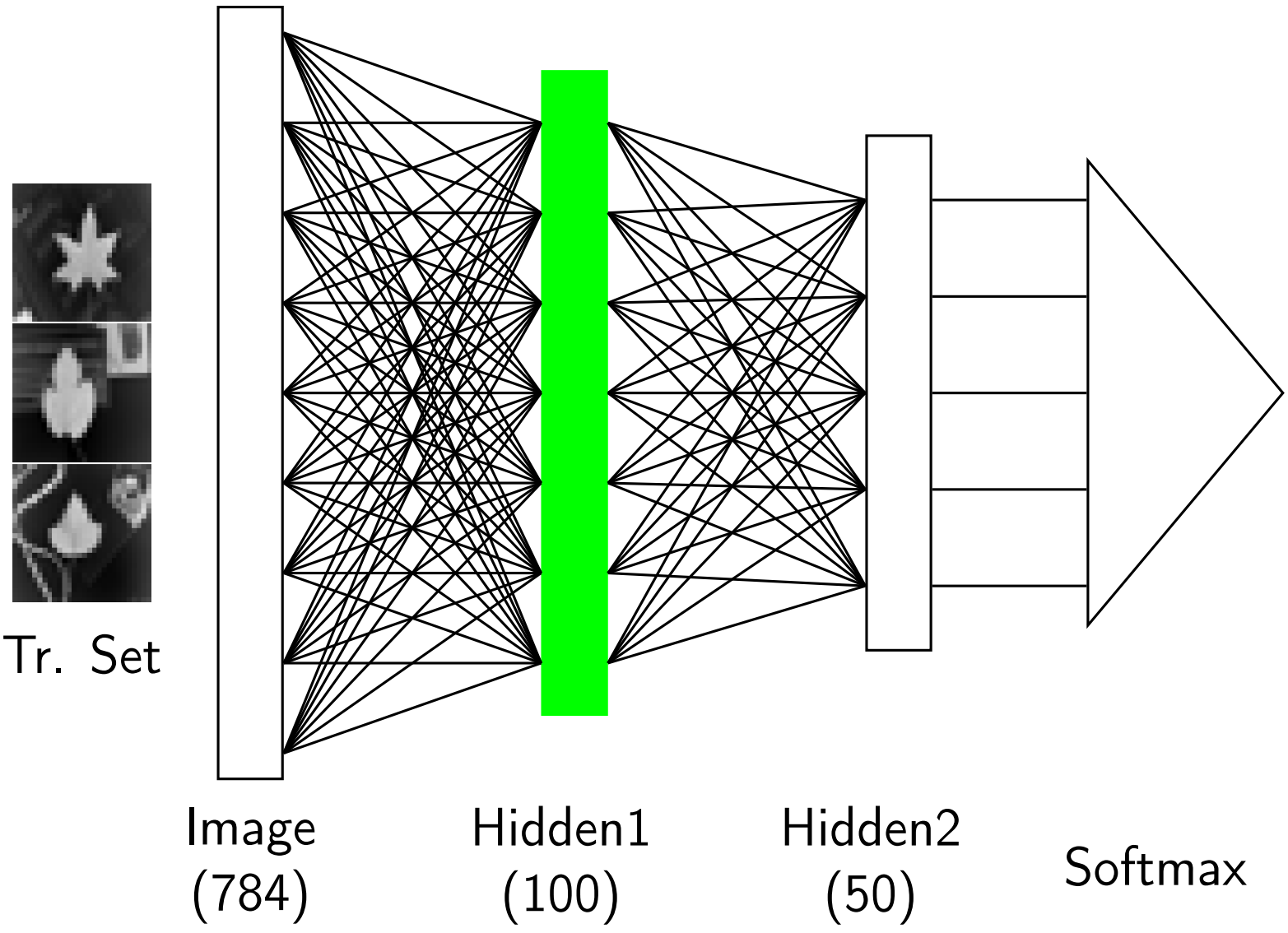
Exp. 1 - Digits dataset - Results



Exp. 2 - Digits dataset - Pre-training



Exp. 2 - Digits dataset - Pre-training



Exp. 2 - Digits dataset - Pre-training

Tr. Set

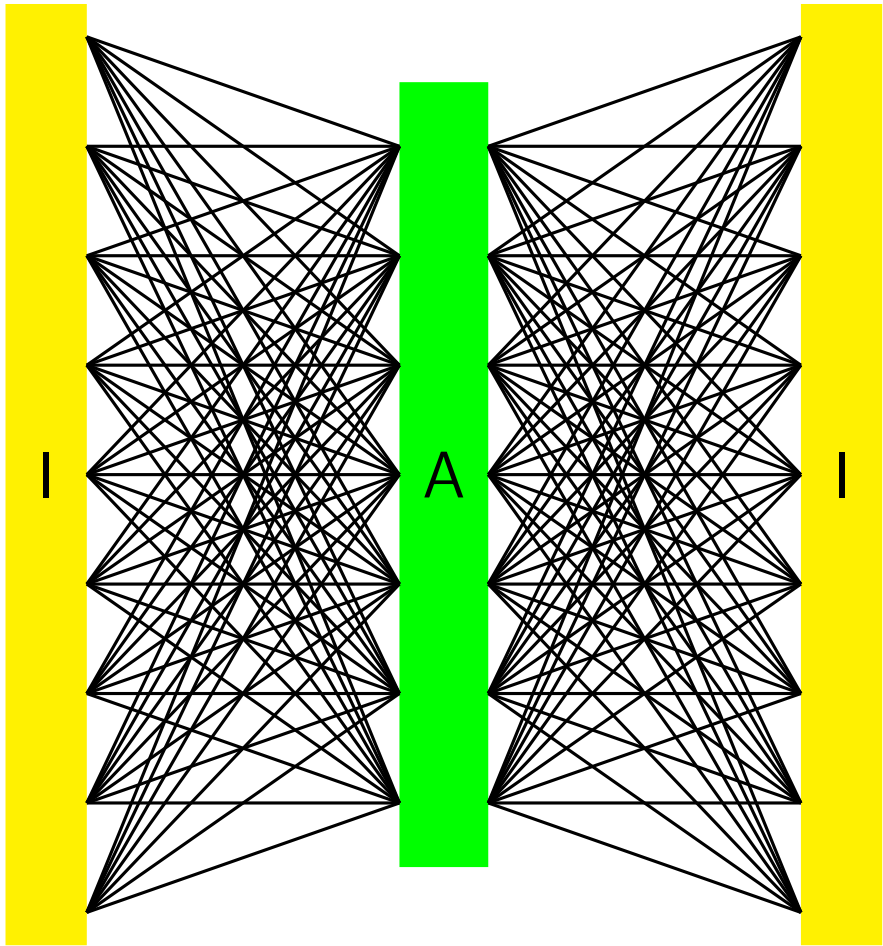


Image (784)

Hidden1 (100)

Image (784)

Exp. 2 - Digits dataset - Pre-training

Tr. Set

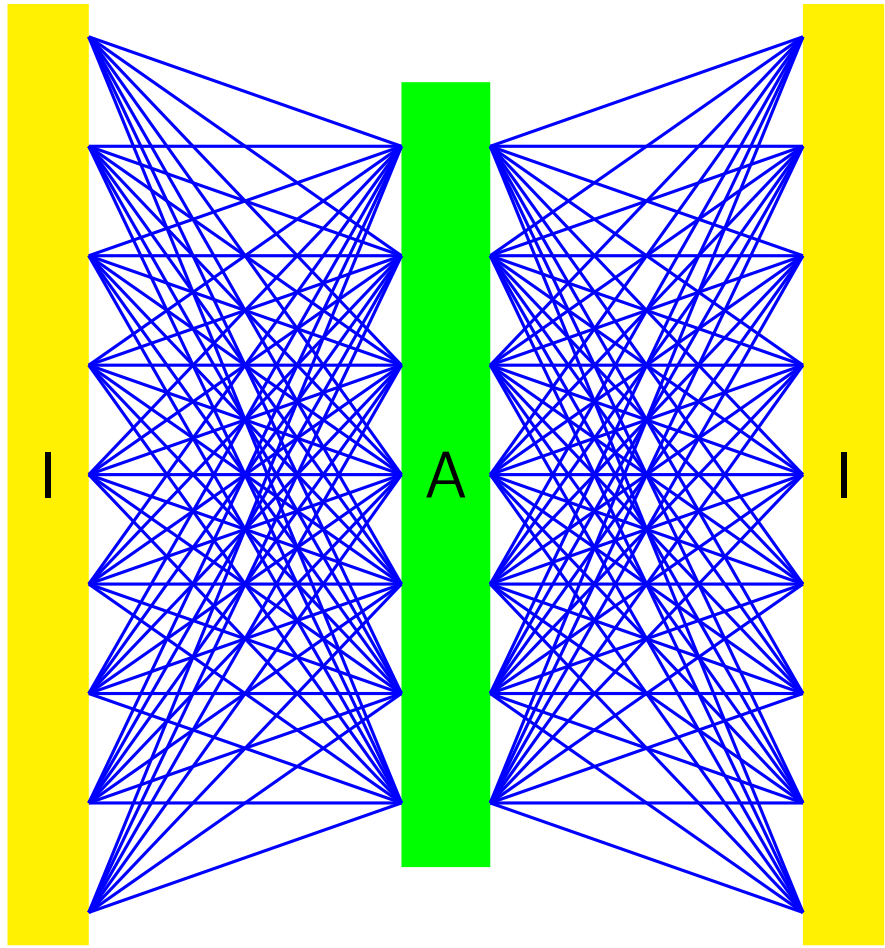
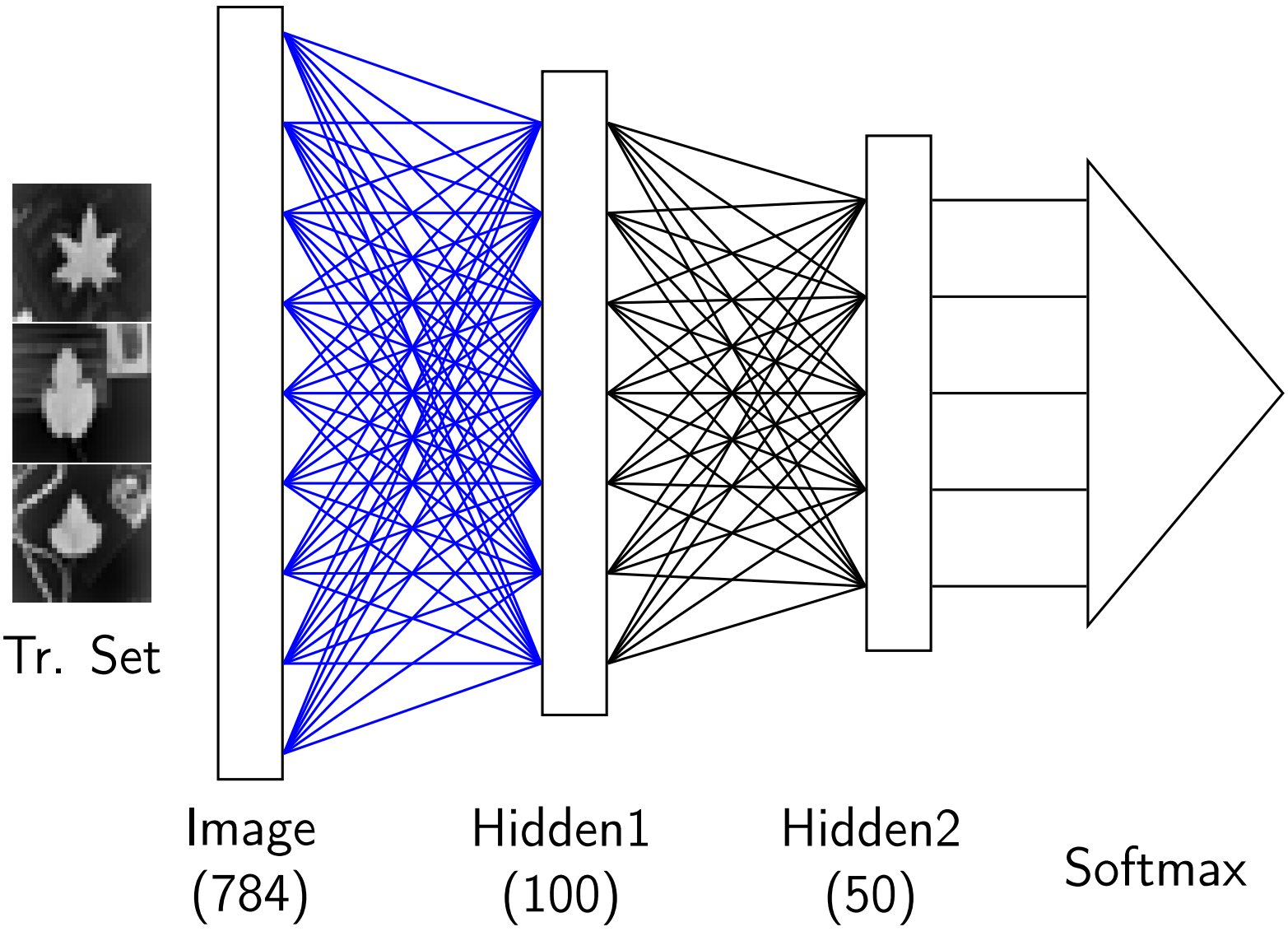
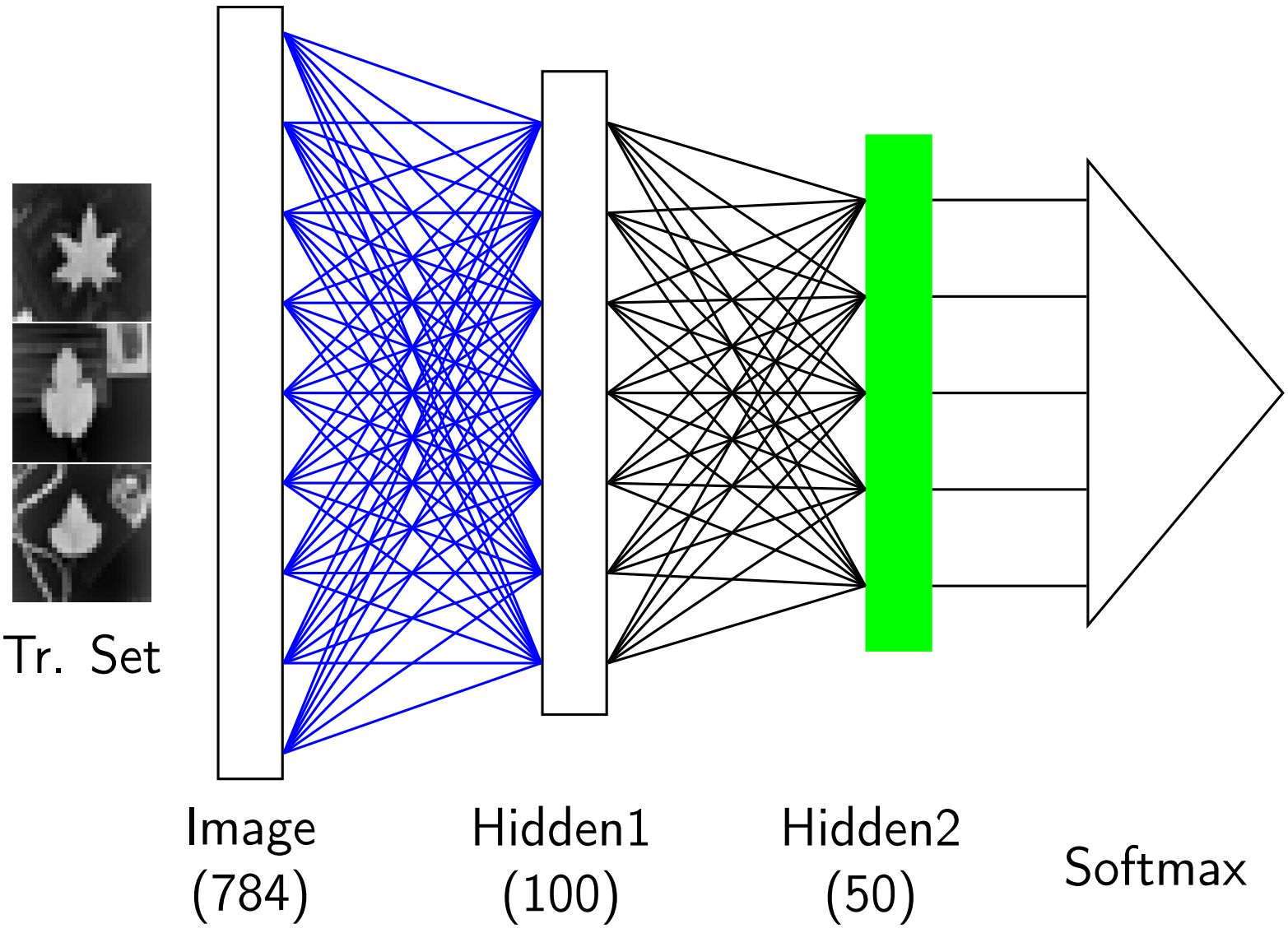


Image (784) Hidden1 (100) Image (784)

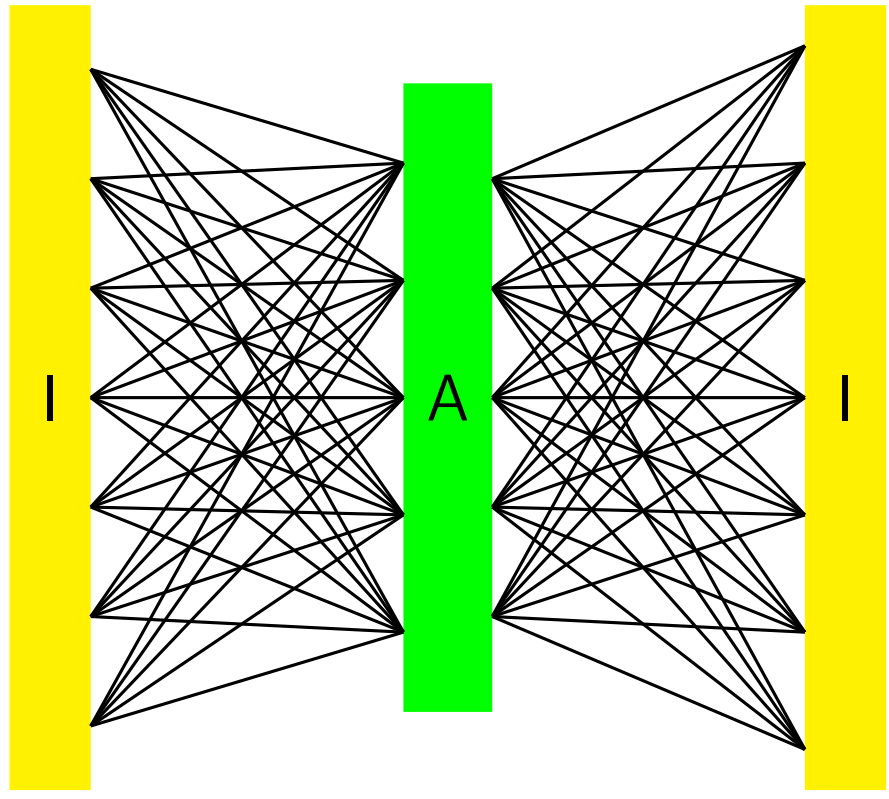
Exp. 2 - Digits dataset - Pre-training



Exp. 2 - Digits dataset - Pre-training

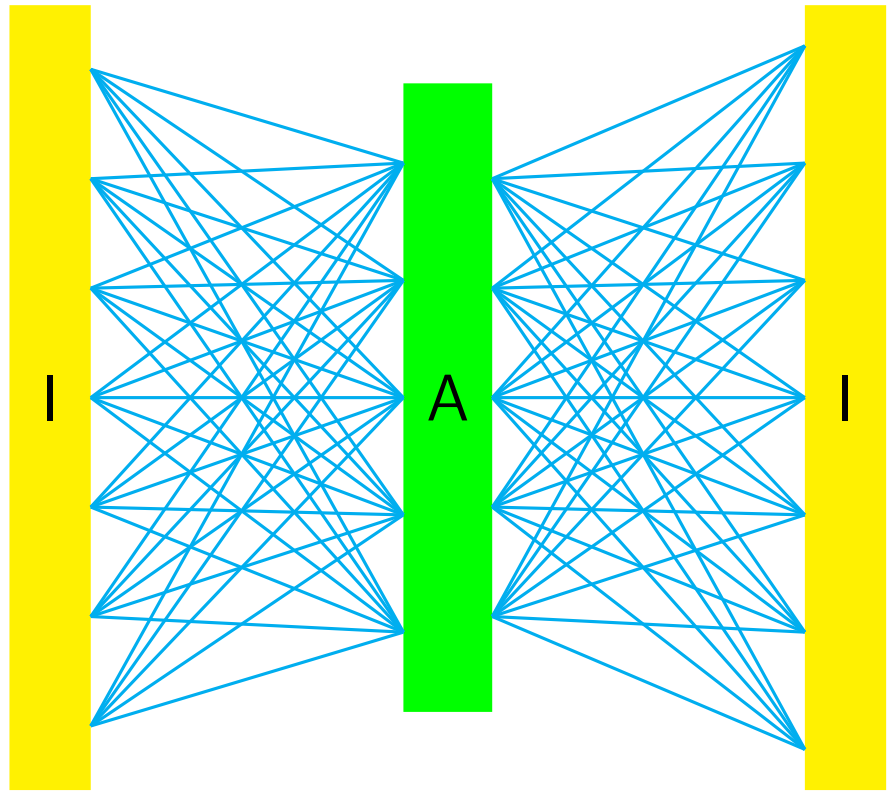


Exp. 2 - Digits dataset - Pre-training



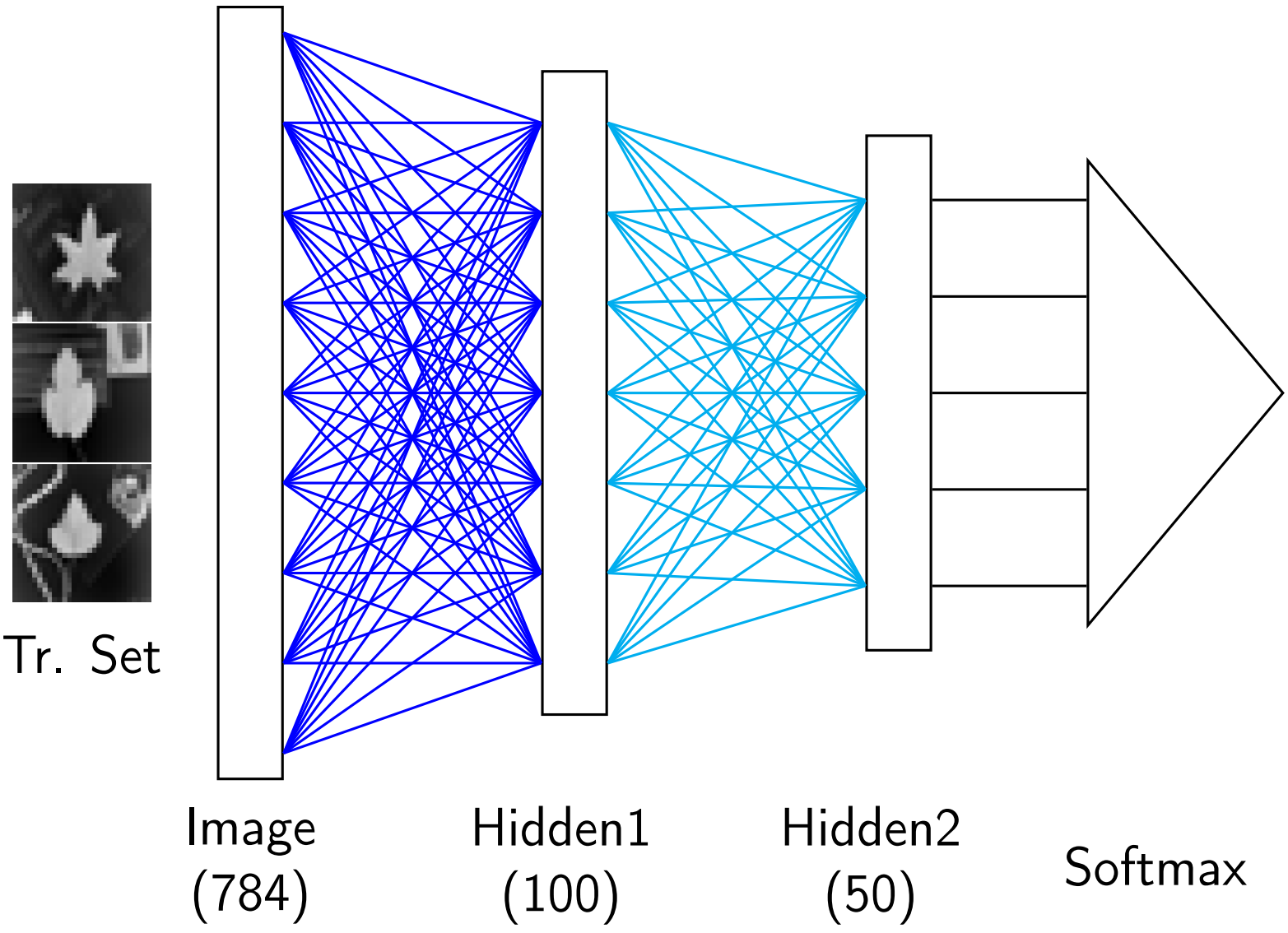
Hidden1 (100) Hidden2 (50) Hidden1 (100)

Exp. 2 - Digits dataset - Pre-training

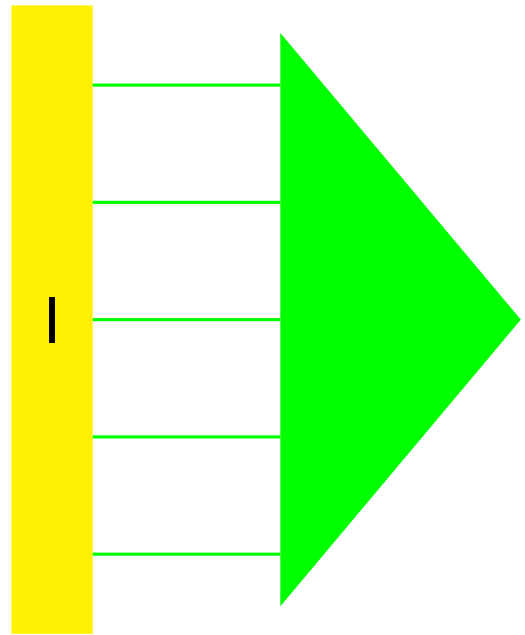


Hidden1 (100) Hidden2 (50) Hidden1 (100)

Exp. 2 - Digits dataset - Pre-training



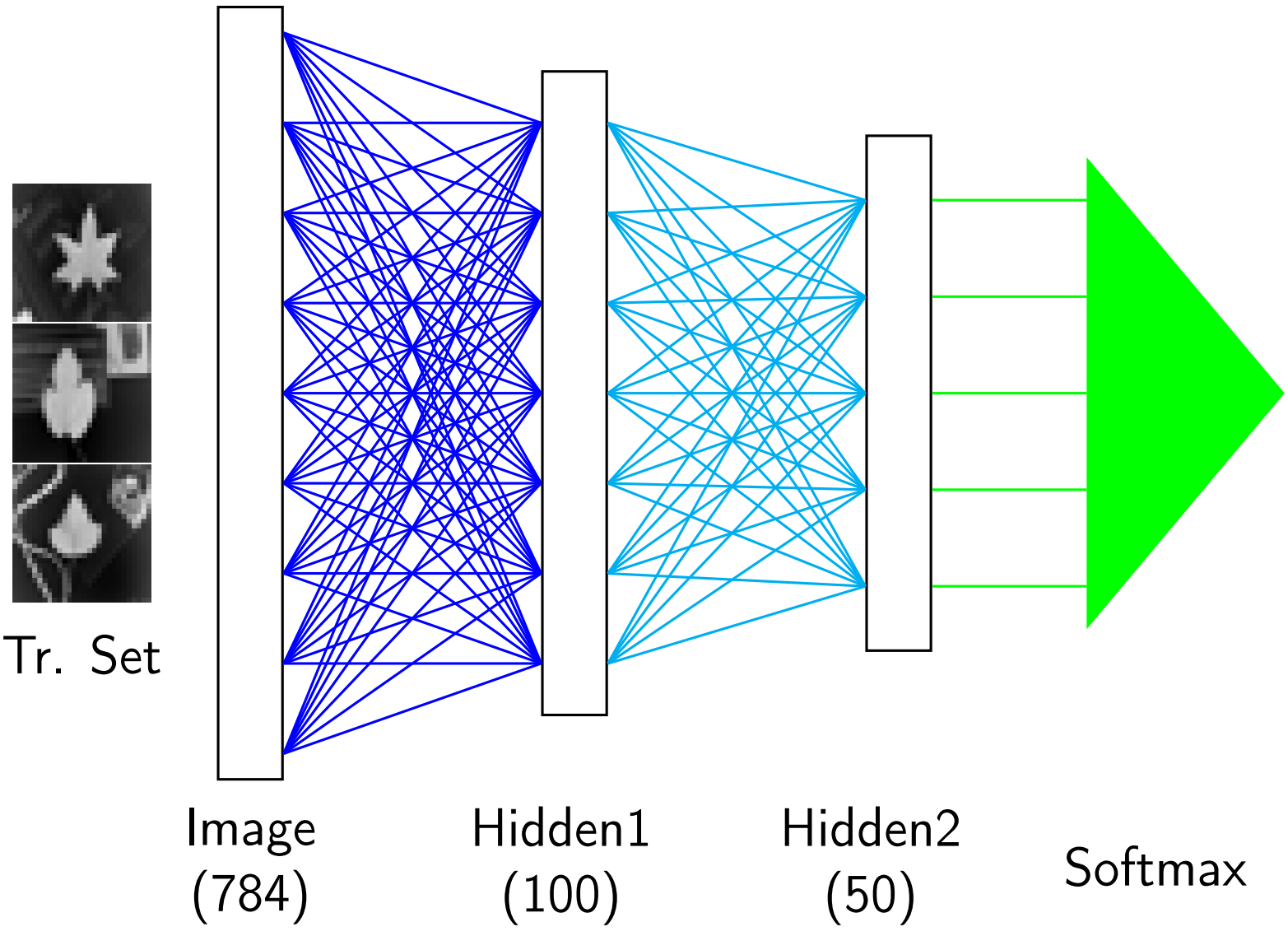
Exp. 2 - Digits dataset - Training



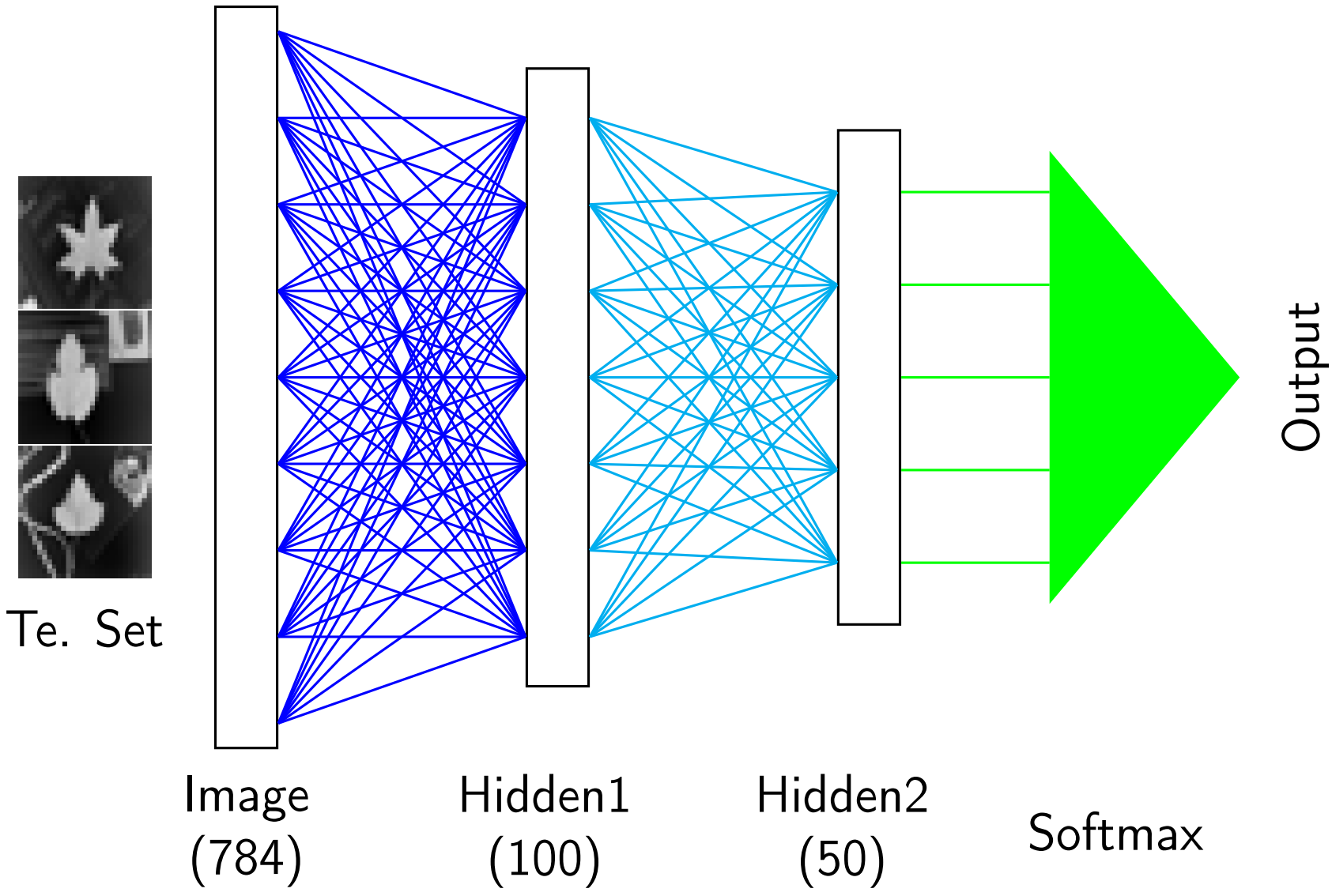
Hidden2
(50)

Softmax

Exp. 2 - Digits dataset - Training



Exp. 2 - Digits dataset - Testing



Exp. 2 - Leaves dataset - Fine tuning

Tr. Set

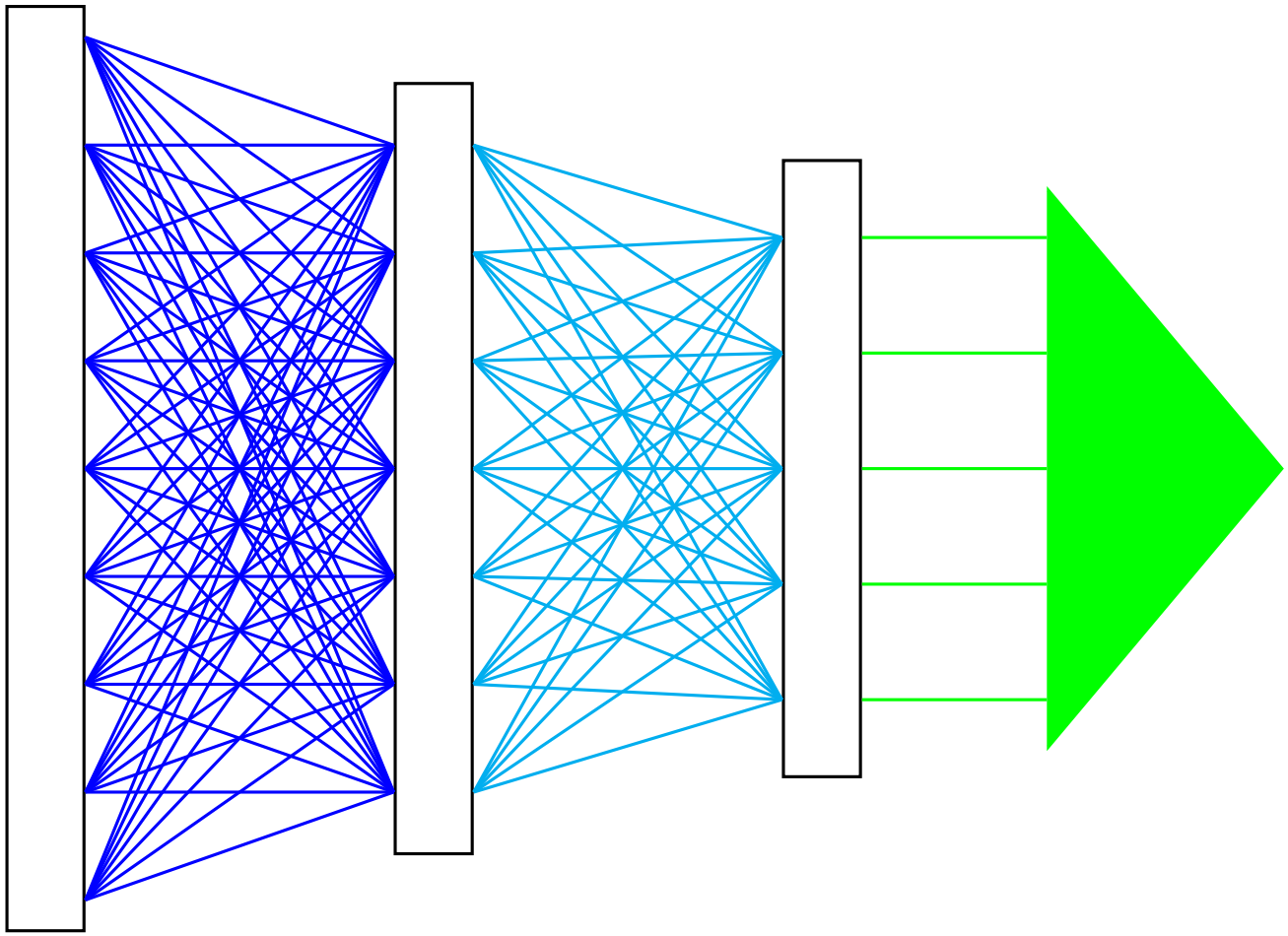


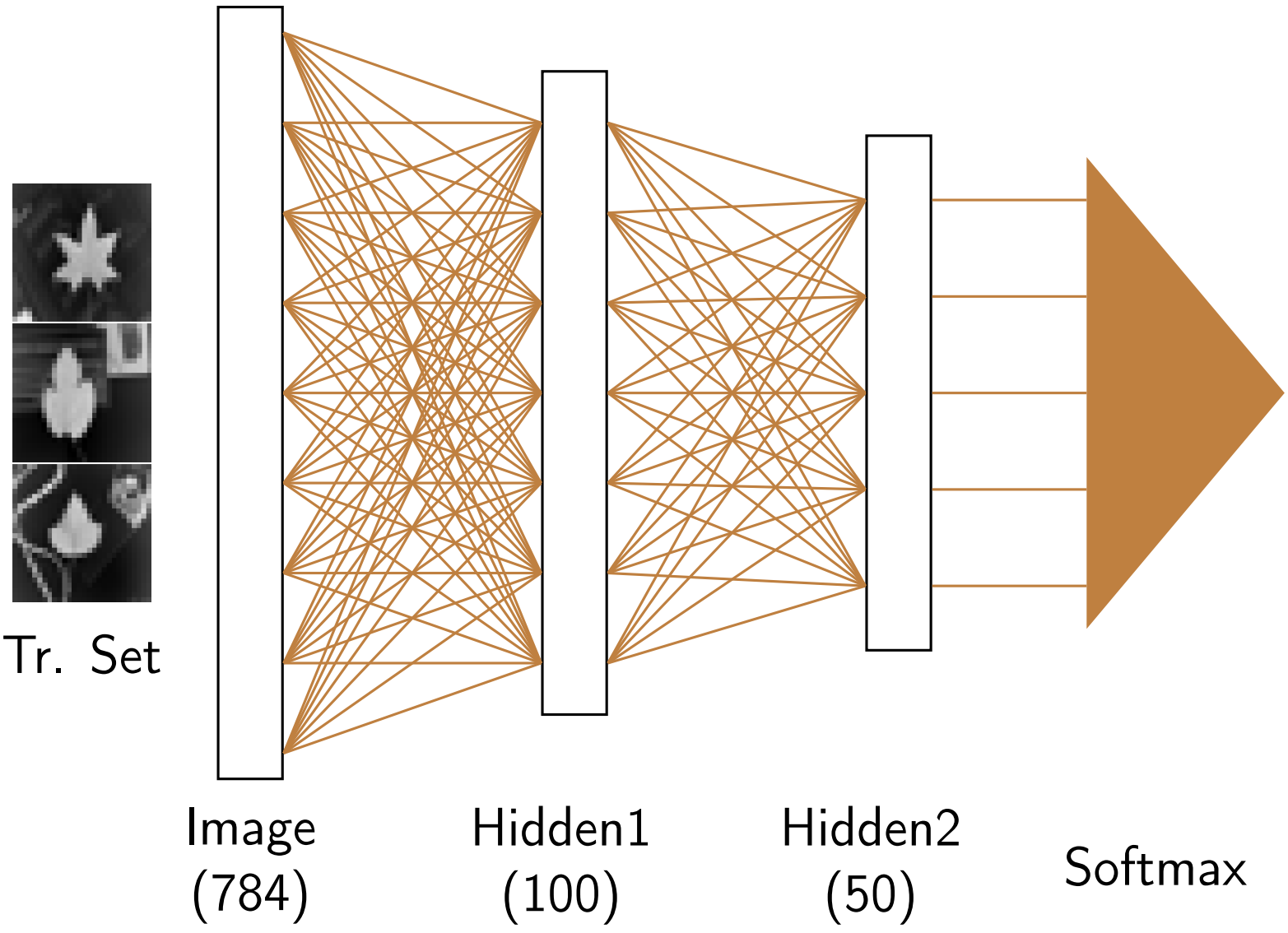
Image (784)

Hidden1 (100)

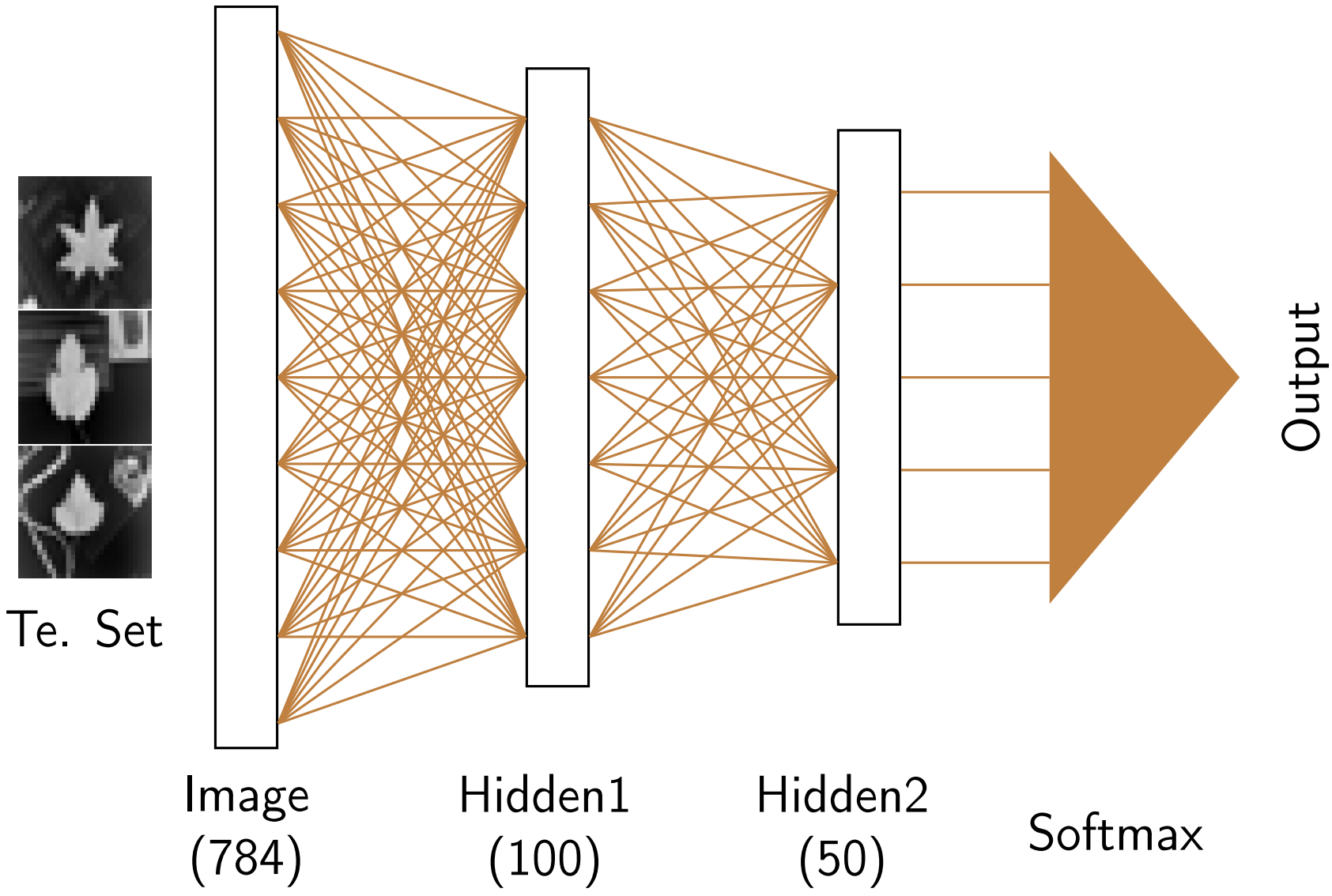
Hidden2 (50)

Softmax

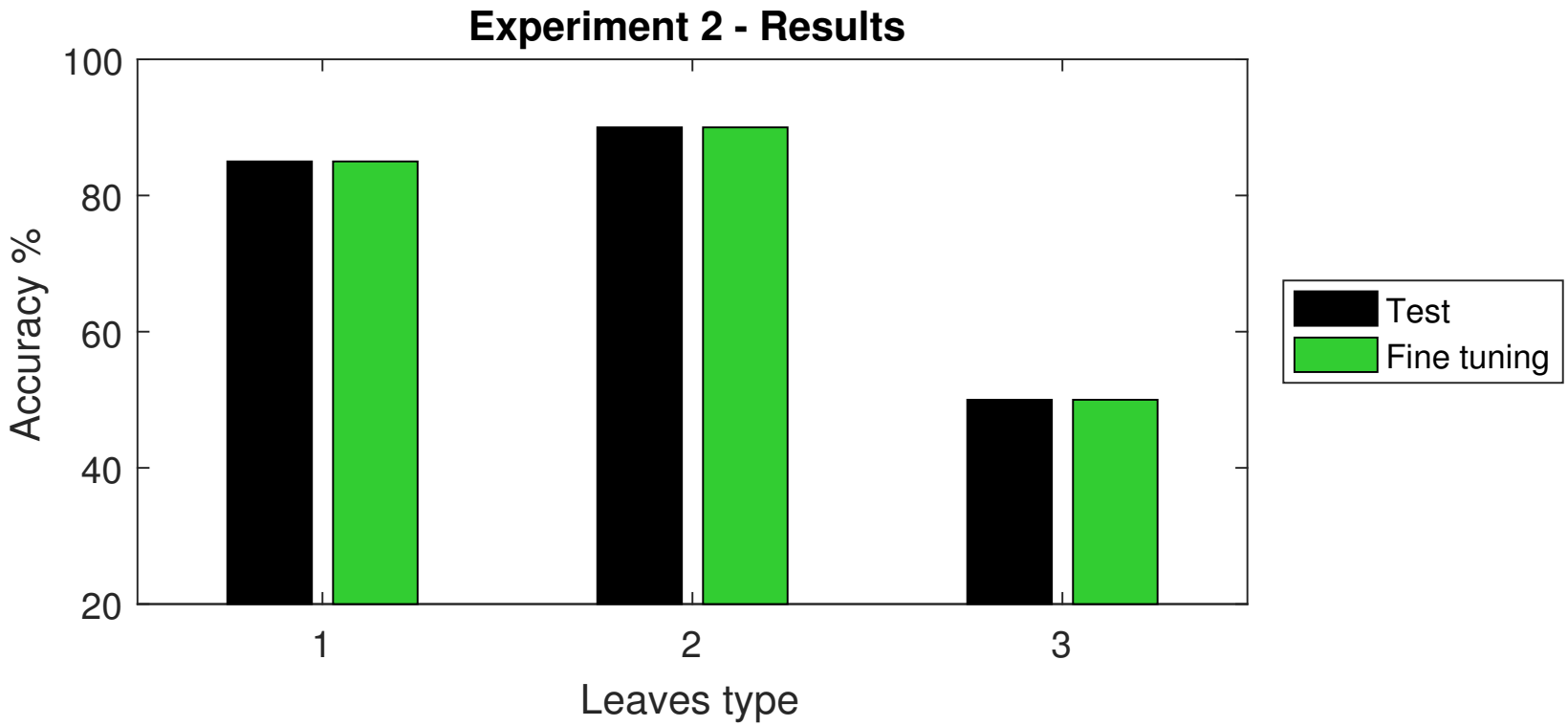
Exp. 2 - Leaves dataset - Fine tuning



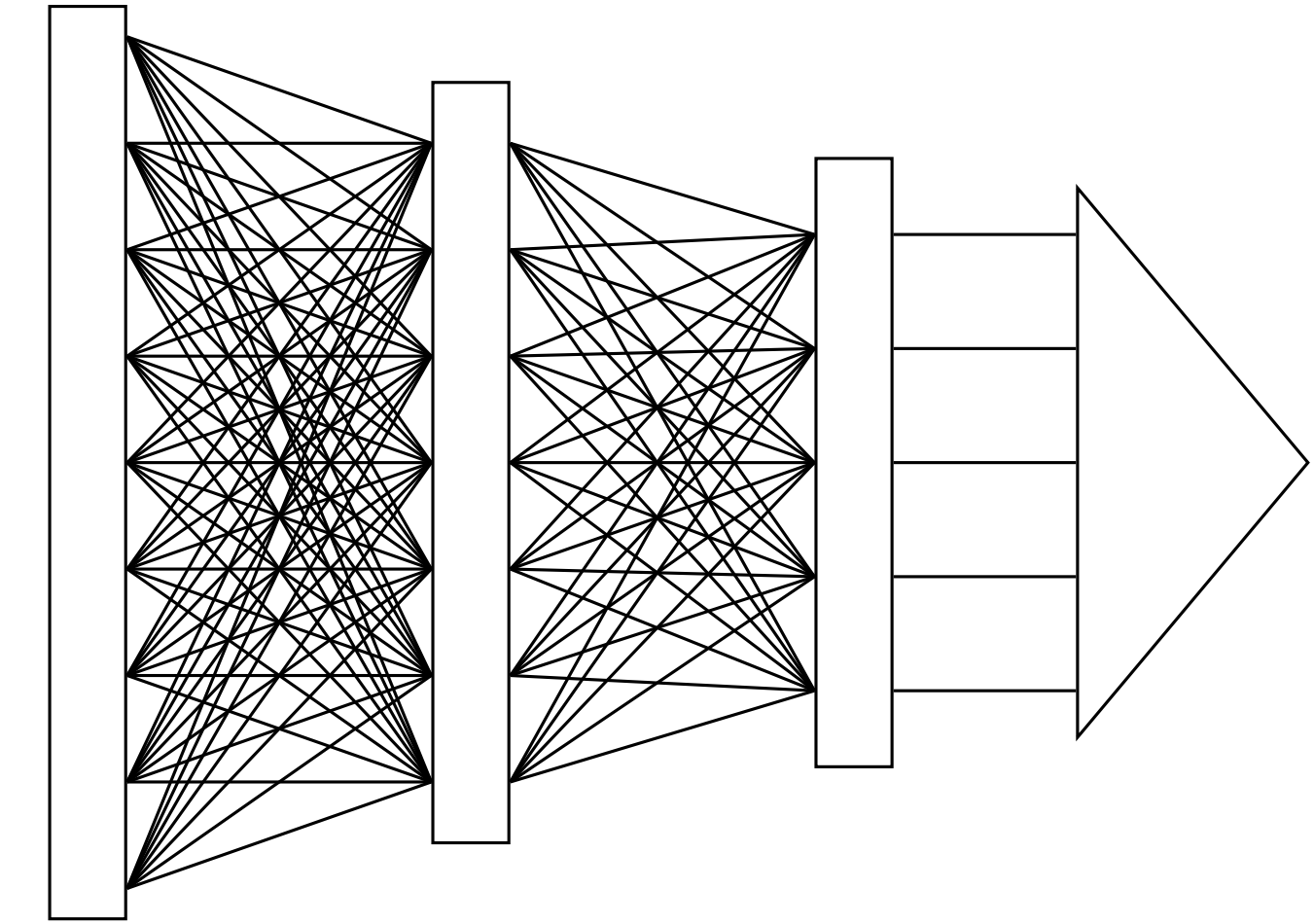
Exp. 2 - Digits dataset - Fine tuning



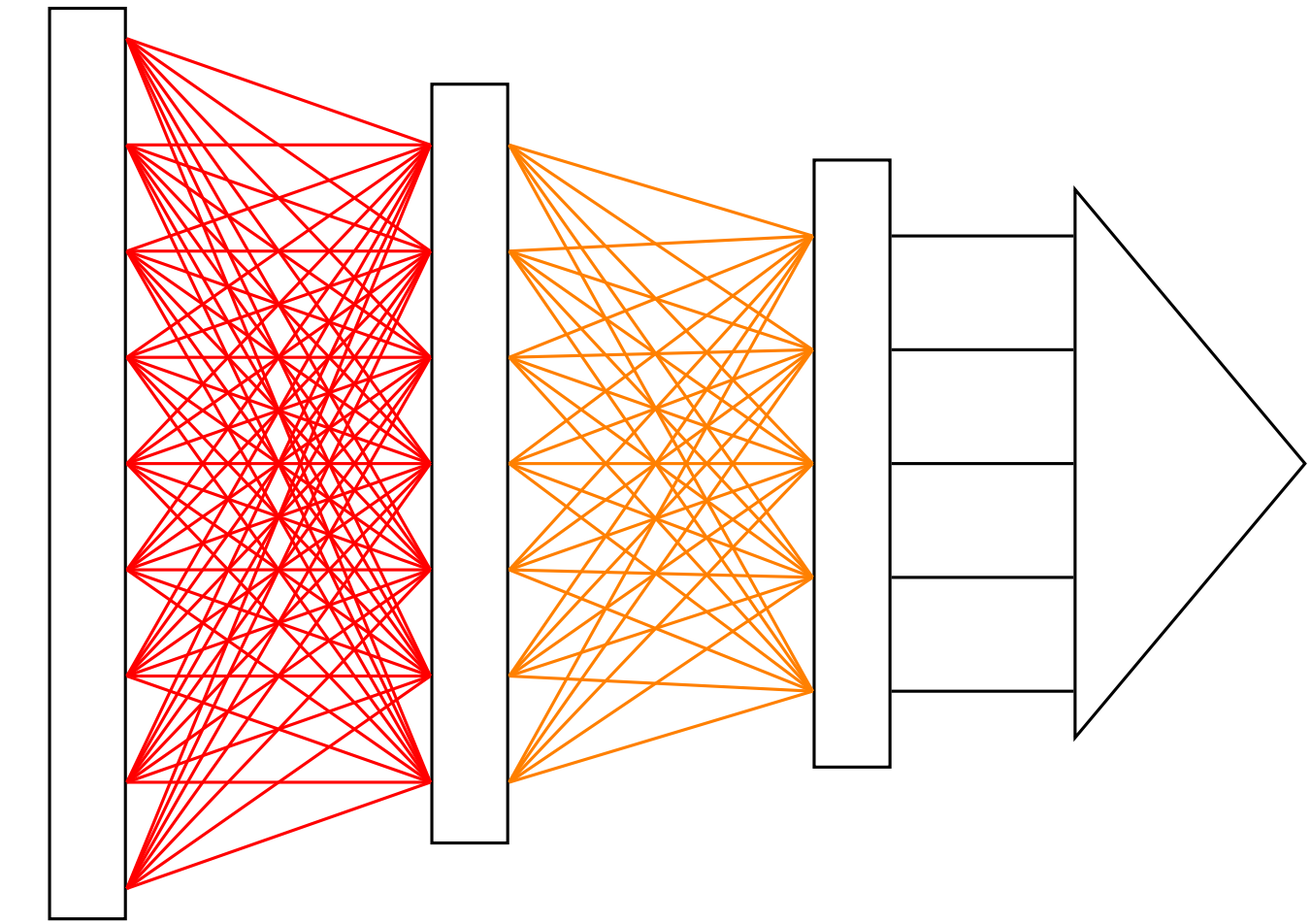
Exp. 2 - Leaves dataset - Results



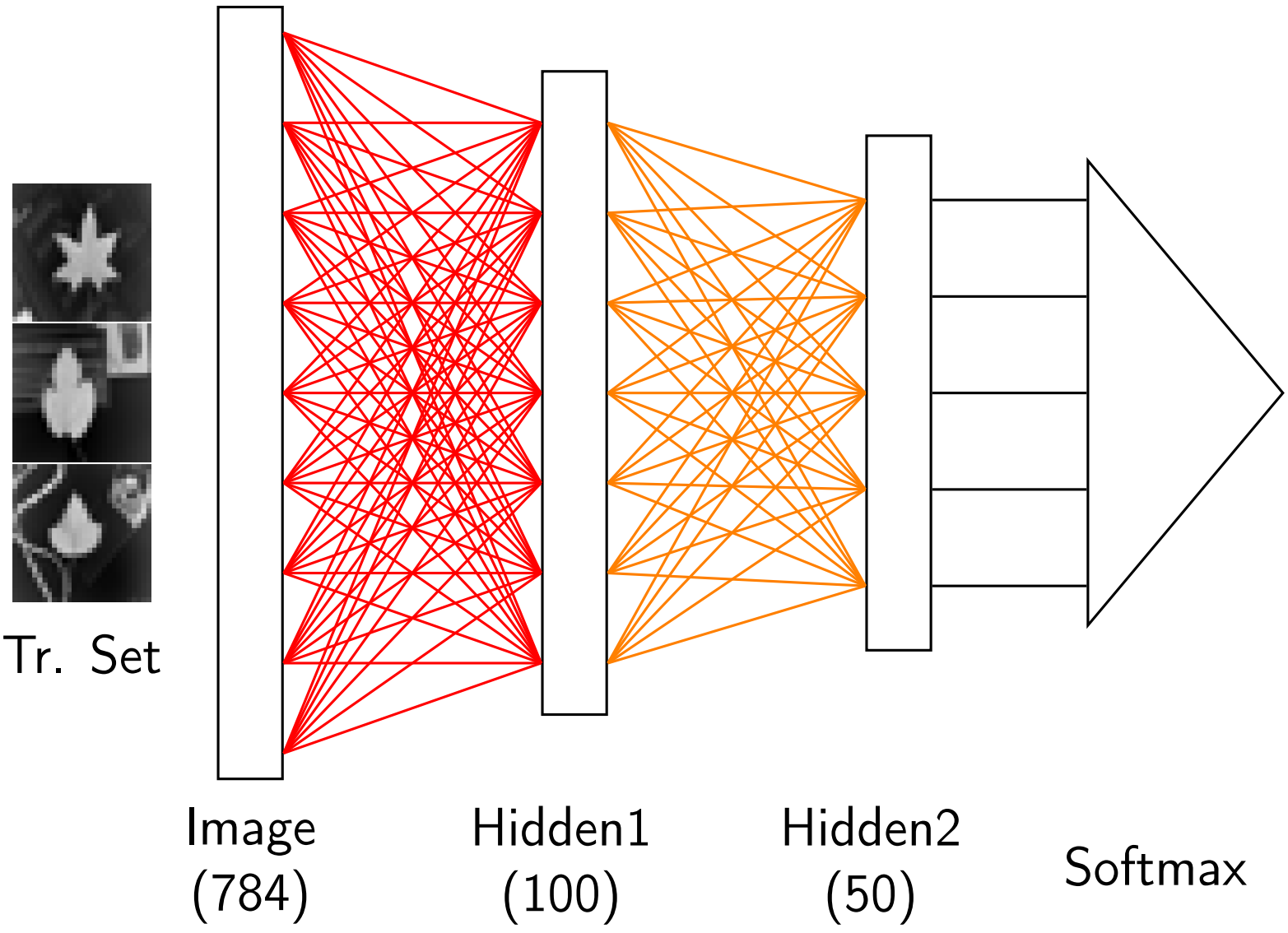
Exp. 3 - Leaves dataset - Transfer Learning



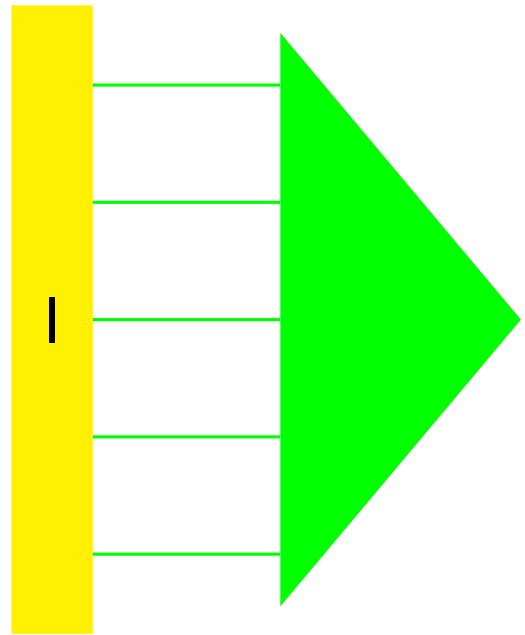
Exp. 3 - Leaves dataset - Transfer Learning



Exp. 3 - Leaves dataset - Training



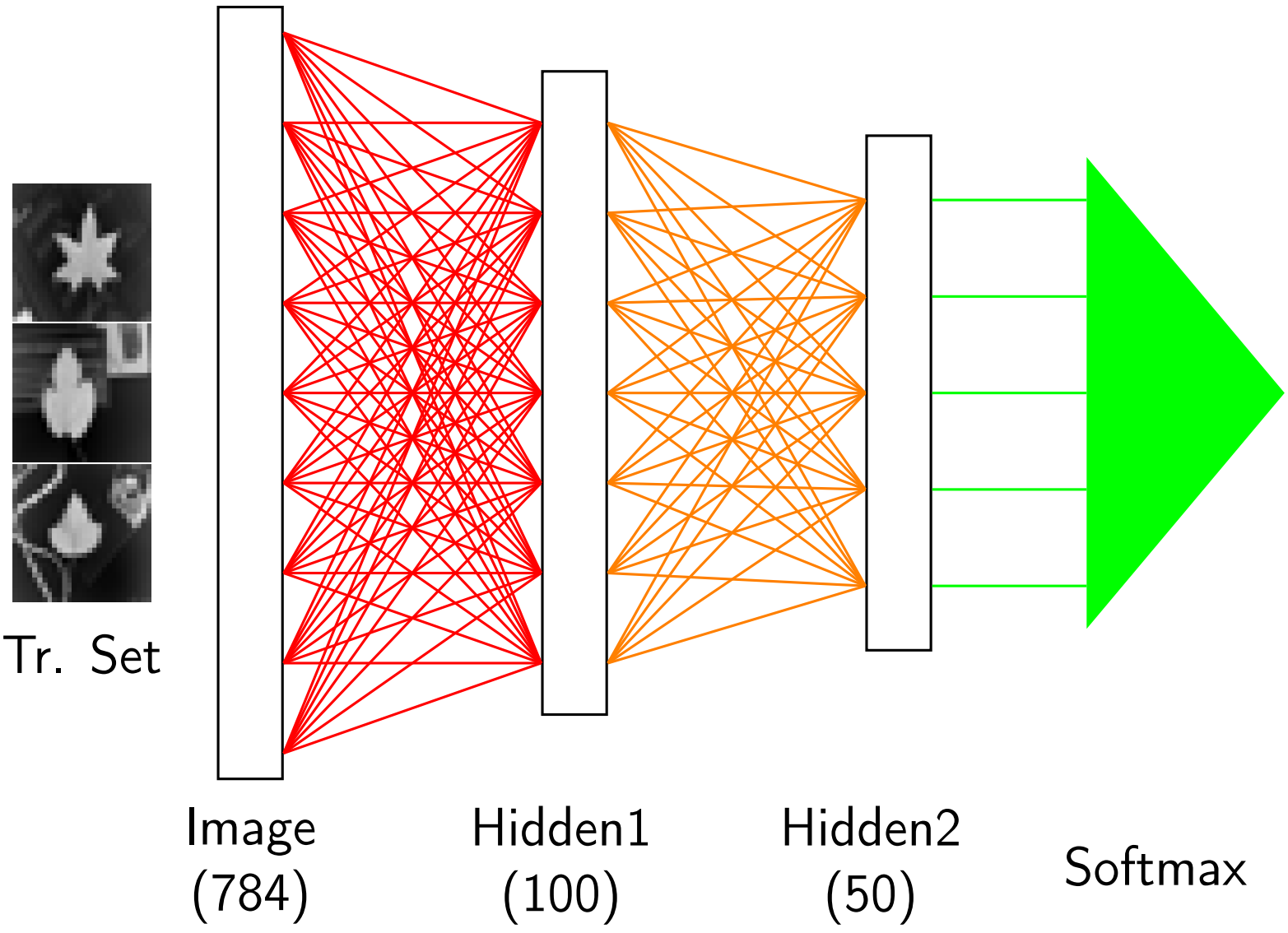
Exp. 3 - Leaves dataset - Training



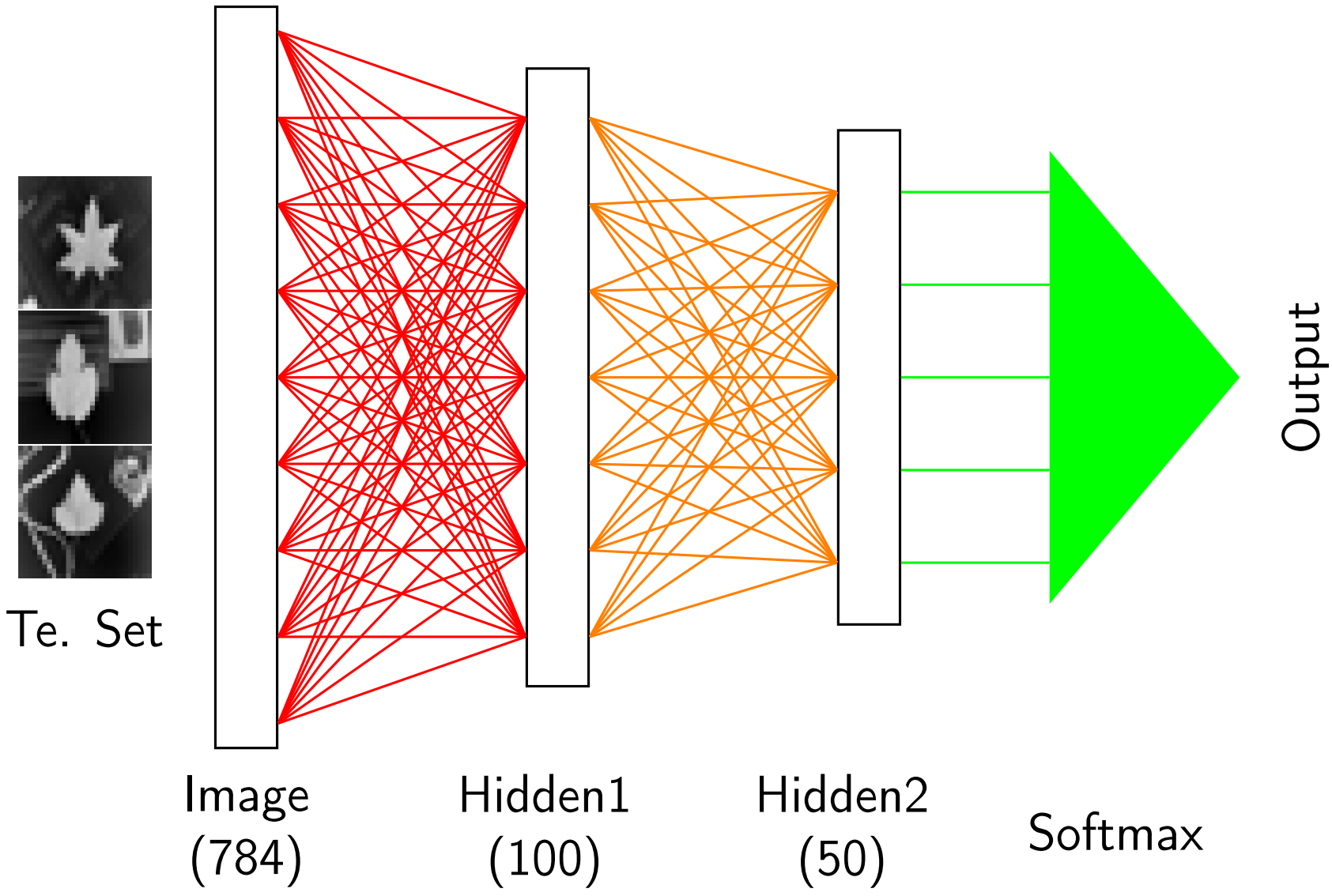
Hidden2
(50)

Softmax

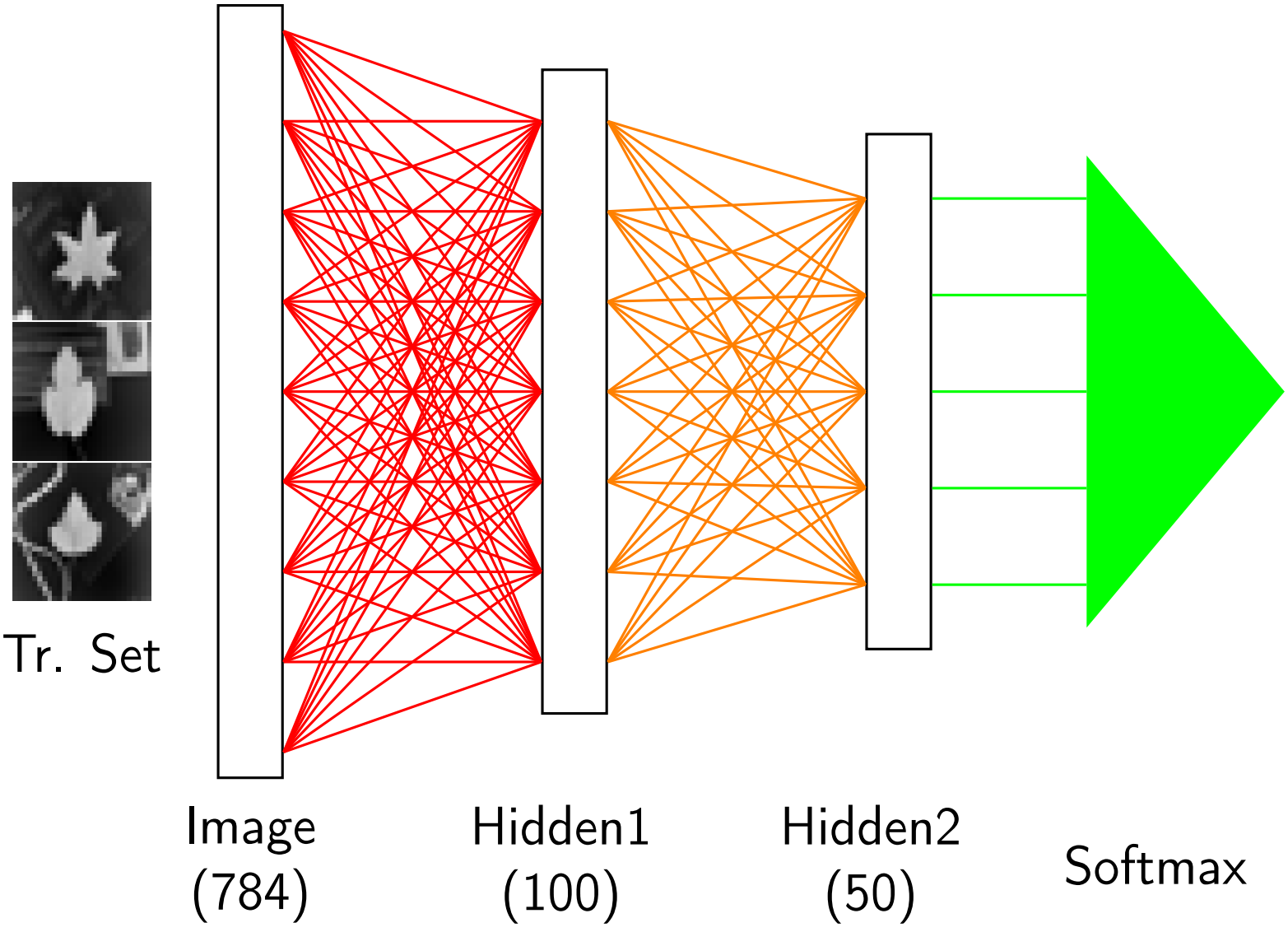
Exp. 3 - Leaves dataset - Training



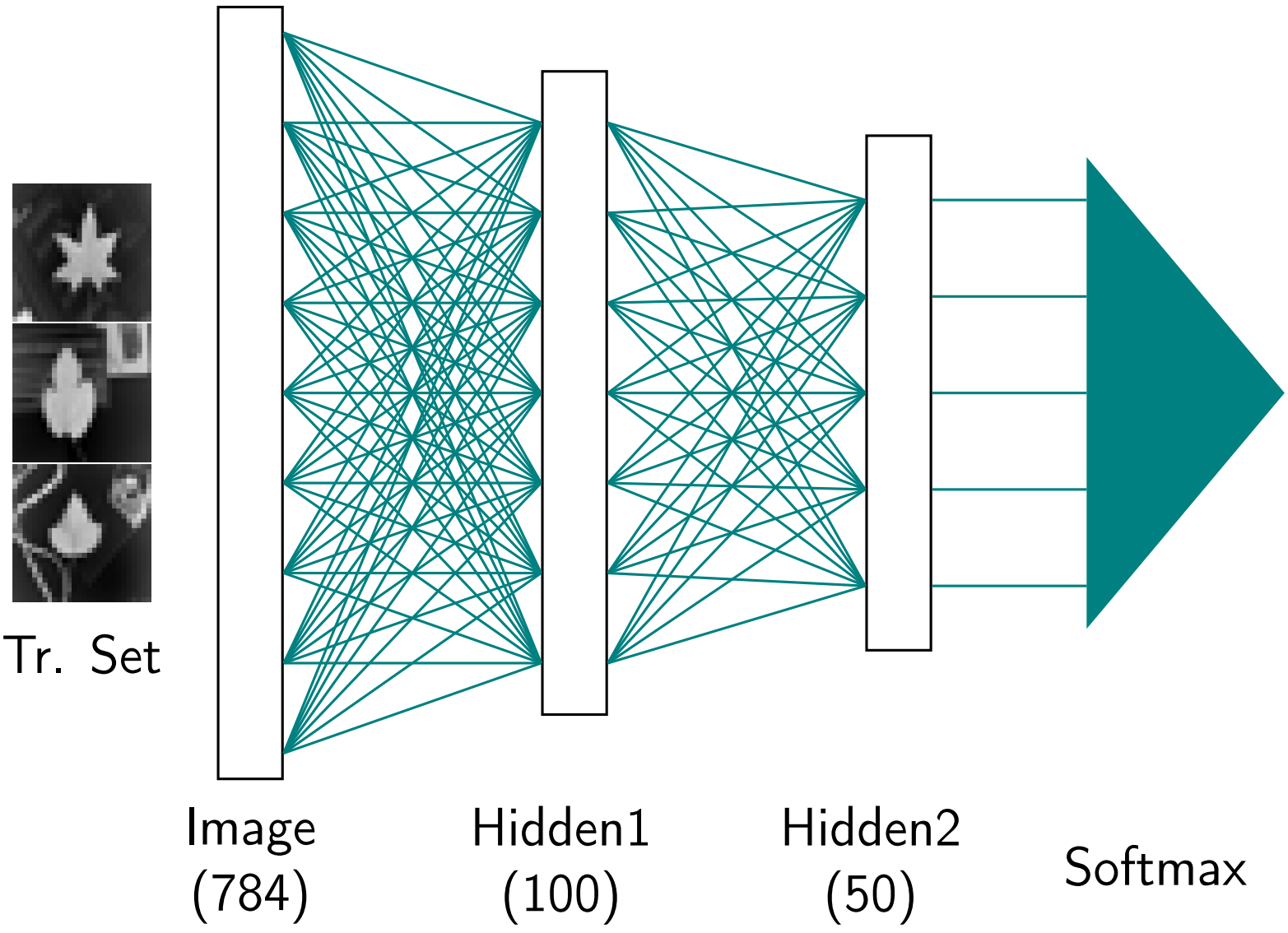
Exp. 3 - Leaves dataset - Testing



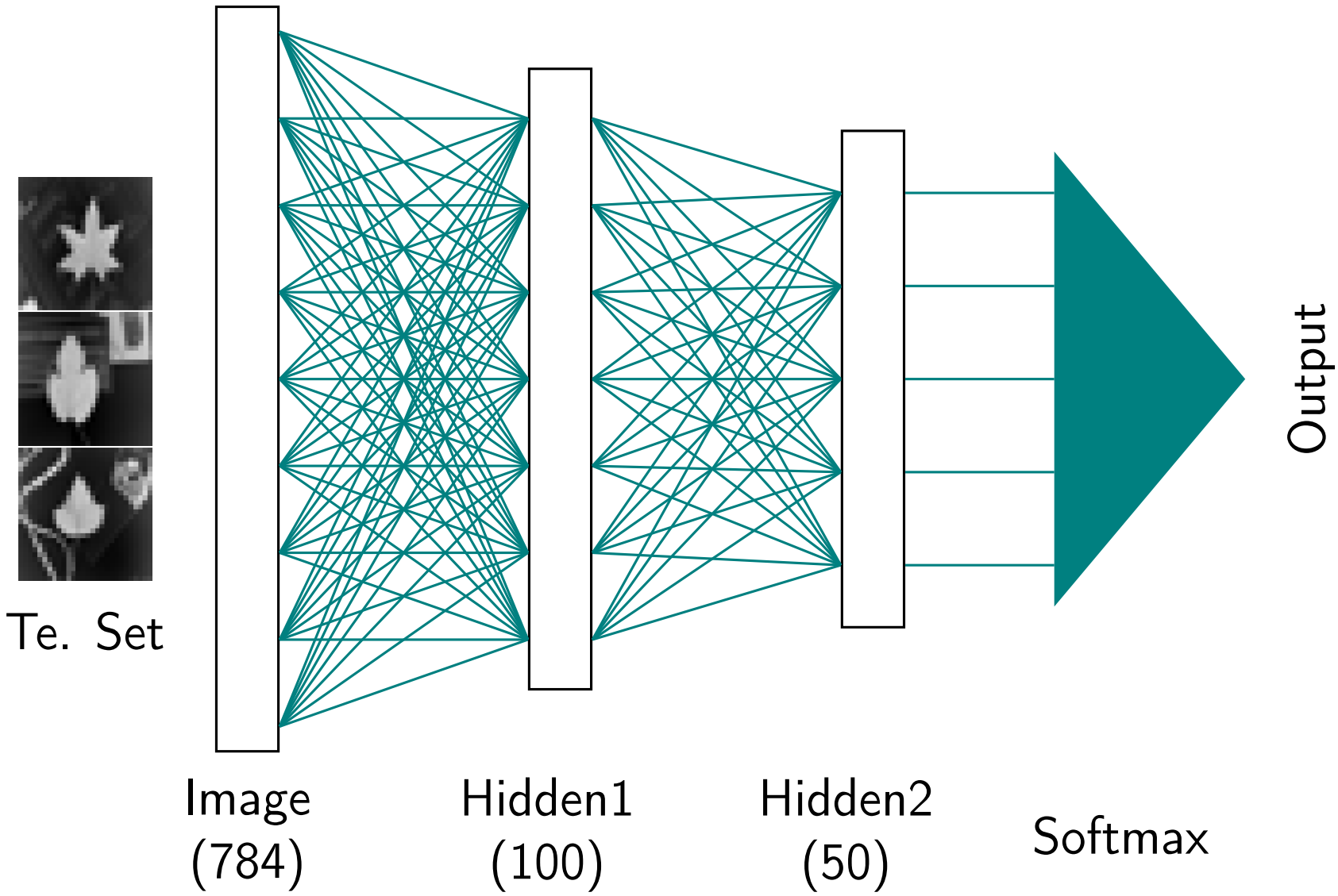
Exp. 3 - Leaves dataset - Fine tuning



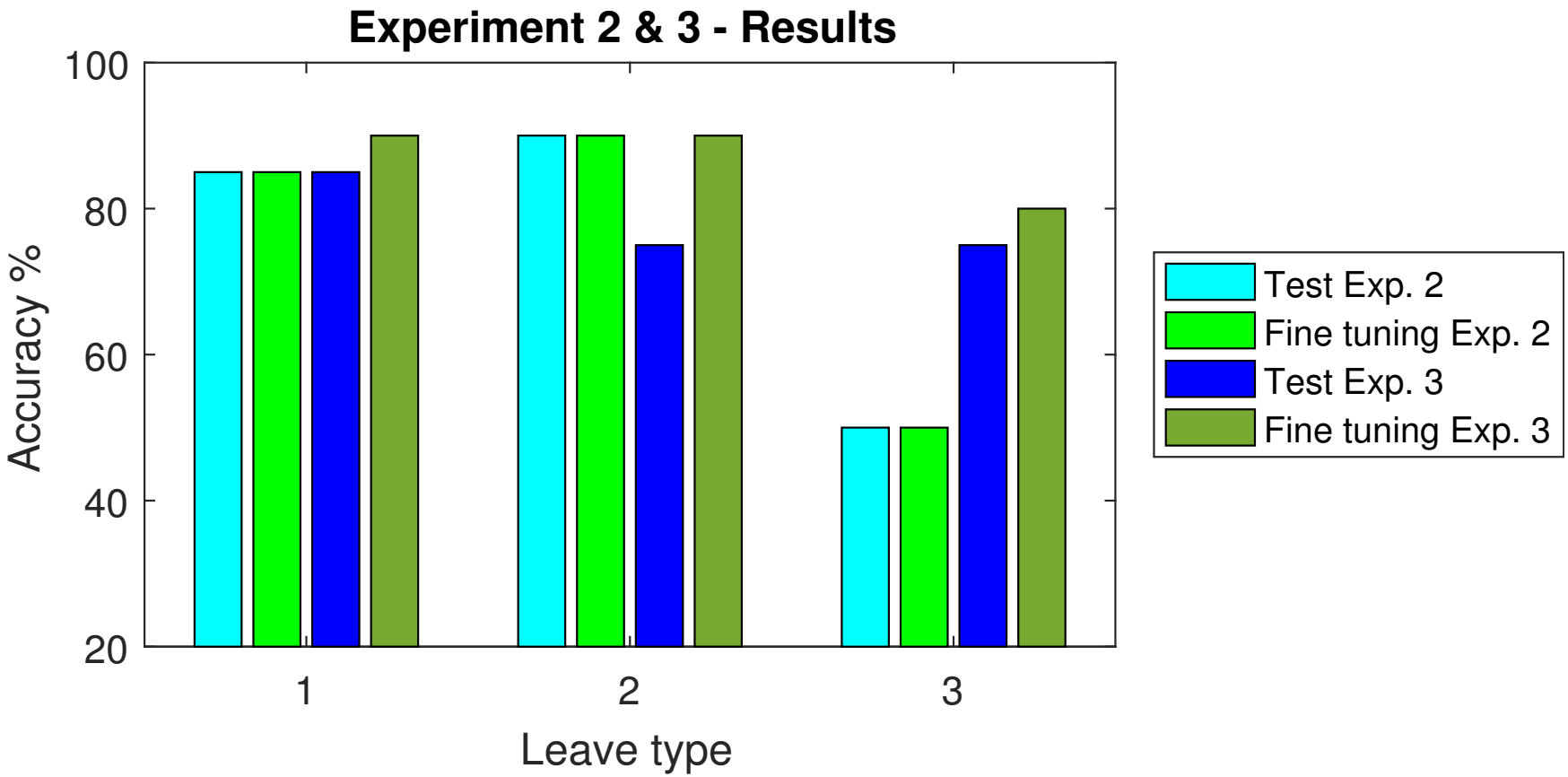
Exp. 3 - Leaves dataset - Fine tuning



Exp. 3 - Digits dataset - Fine tuning



Exp. 3 - Leaves dataset - Results



Conclusions

- We can effectivily train a deep network by means of autoencoders.
- A deep network can work well when we apply a transfer learning procedure.
- Fine tuning procedure can improves the overall performance of a deep network.
- MATLAB® allows a straightforward implementation of deep networks.

Future work

- Implement more experiments, e.g. interchange roles in Experiment 3.
- Manipulate the technical specifications of each component in the deep network.
- Repeat this experiments using diferente datasets.

References I

[1] Taylor B. Arnold.
Lecture 18 - Transfer Learning and Computer Vision I.
http://euler.stat.yale.edu/~tba3/stat665/lectures/lec18/lecture18.pdf.
Accesed: 24/07/2016.

[2] Li Fei-Fei, R. Fergus, and P. Perona.
Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories.
In *Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04.*, pages 178–178, 2004.

References II

[3] Quoc V. Le.
A Tutorial on Deep Learning, Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks.
https://cs.stanford.edu/~quocle/tutorial2.pdf.
Accesed: 24/07/2016.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner.
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324, 1998.

[5] The Computational Vision Group.
Leaves dataset taken by Markus Weber (1999).
http://www.vision.caltech.edu/Image_Datasets/leaves/leaves.tar.
Accesed: 24/07/2016.

References III

[6] The MathWorks, Inc.
Training a Deep Neural Network for Digit Classification.
http://www.mathworks.com/help/nnet/examples/training-a-deep-neural-network-for-digit-classifica
html.
Accesed: 24/07/2016.