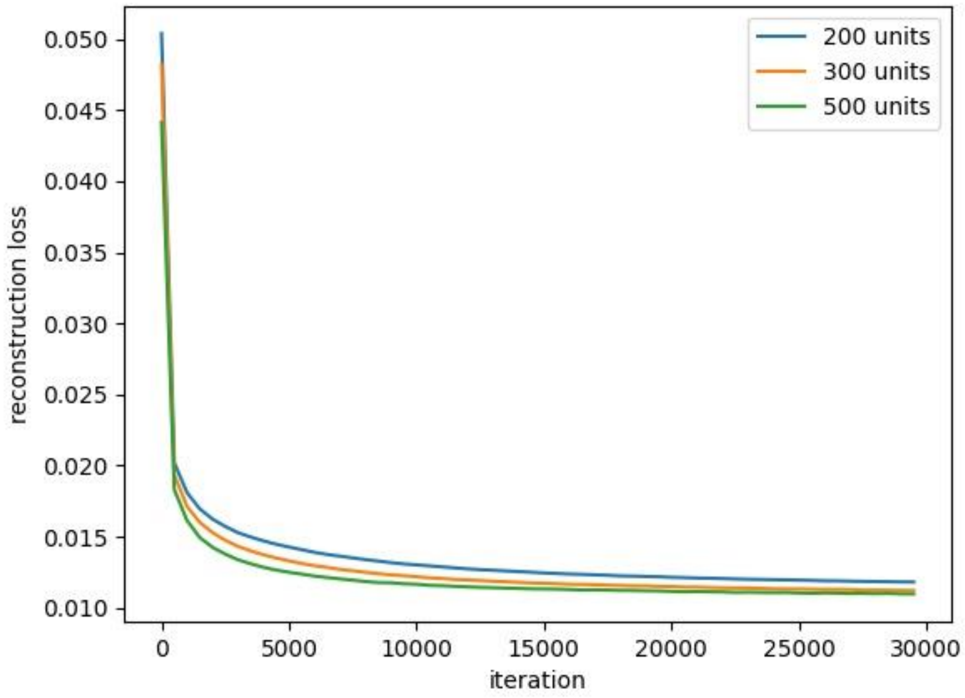
Restricted Boltzmann Machines and Deep Belief Nets

Artificial Neural Networks and Deep Architectures – Lab 4

Franco Ruggeri, Fredrik Danielsson

Restricted Boltzmann Machines

Number of hidden units

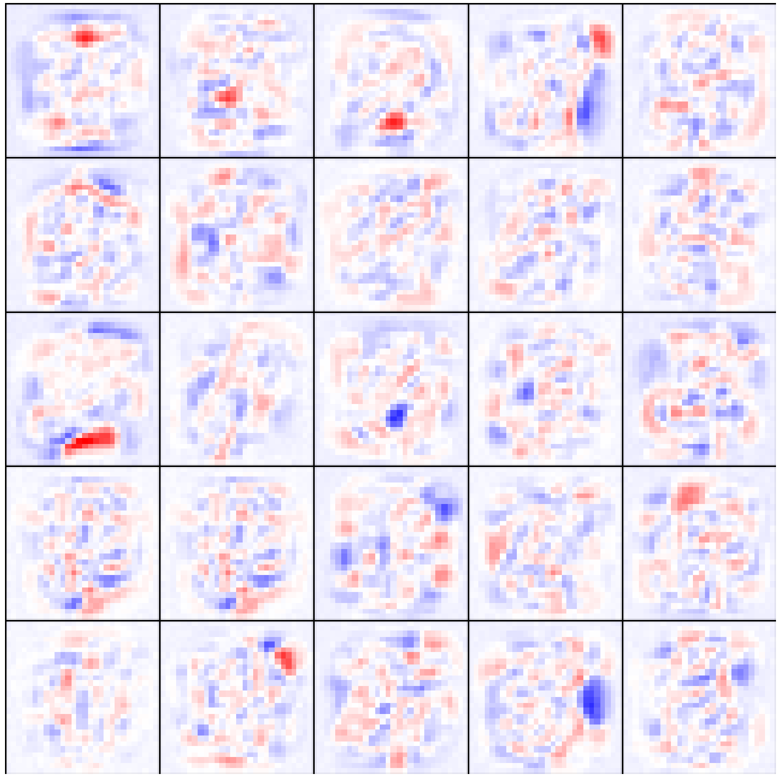


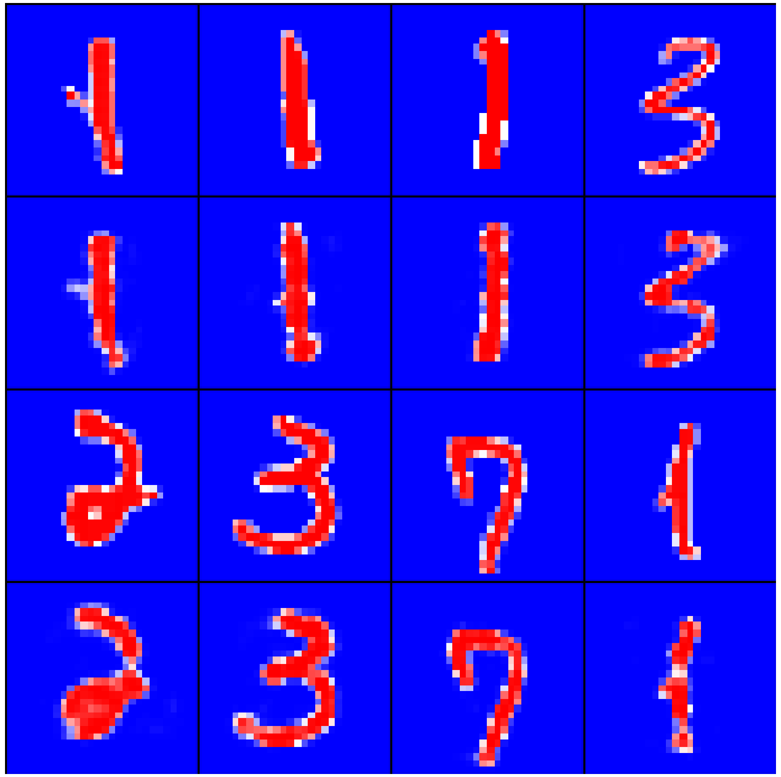
Dataset	# hidden units	Reconstruction loss
Training	200	0.01180 ± 0.00002
Training	300	0.01118 ± 0.00002
Training	500	0.01095 ± 0.00001
Test	200	0.02888 ± 0.00005
Test	300	0.02740 ± 0.00003
Test	500	0.02681 ± 0.00002

- More hidden units → more degrees of freedom → the model fits better the data

Restricted Boltzmann Machines

Receptive fields and reconstructions

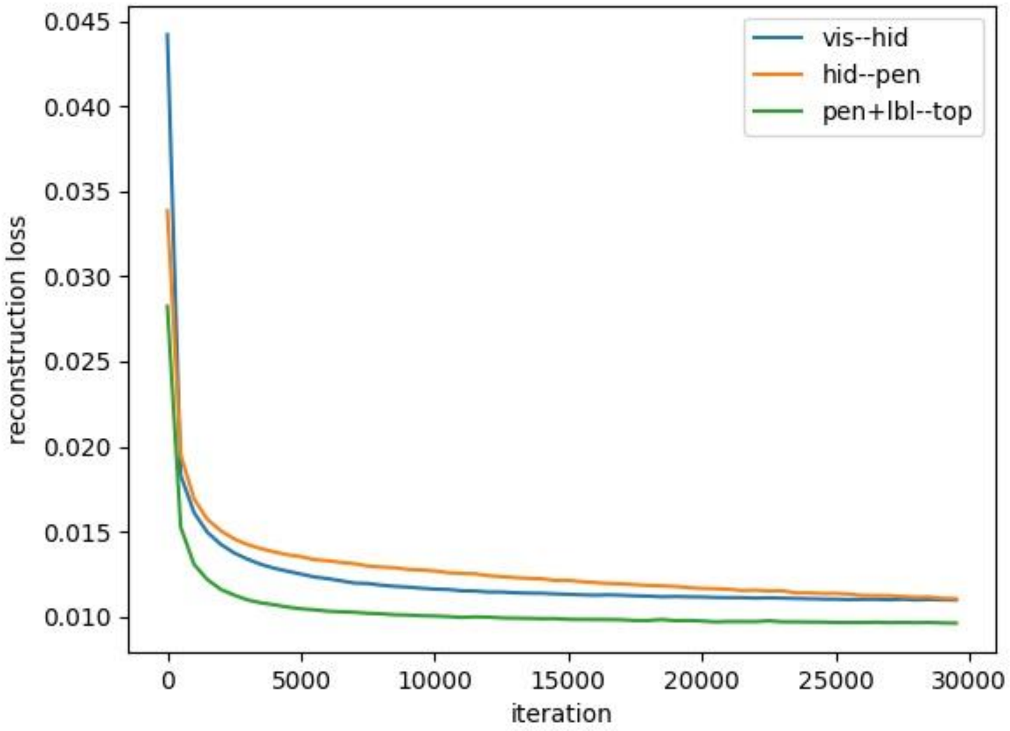


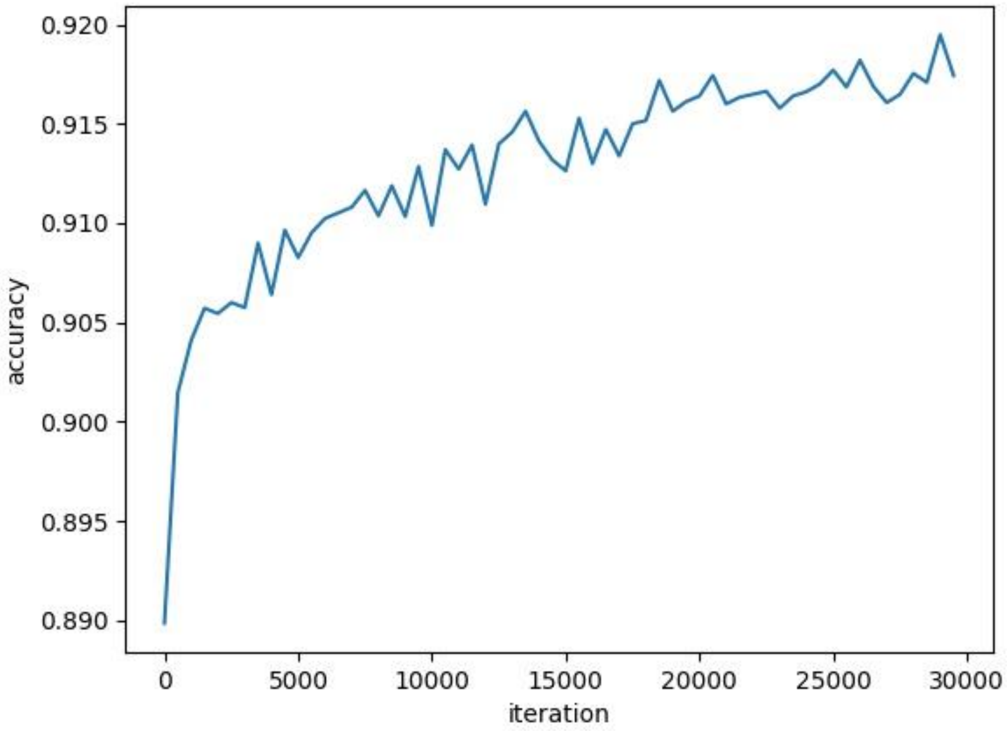


- Receptive fields (left) show some templates of several digits mixed together
- Reconstructions (right) of test images very very good → representation learnt

Deep Belief Nets

Training - Greedy layer-wise pre-training and fine-tuning





- Pre-training (left): each layer (pre-)trained until convergence to build a stack (freezing layers below)
- Fine-tuning (right): whole network trained simultaneously, using pre-trained weights as initializations

Deep Belief Nets

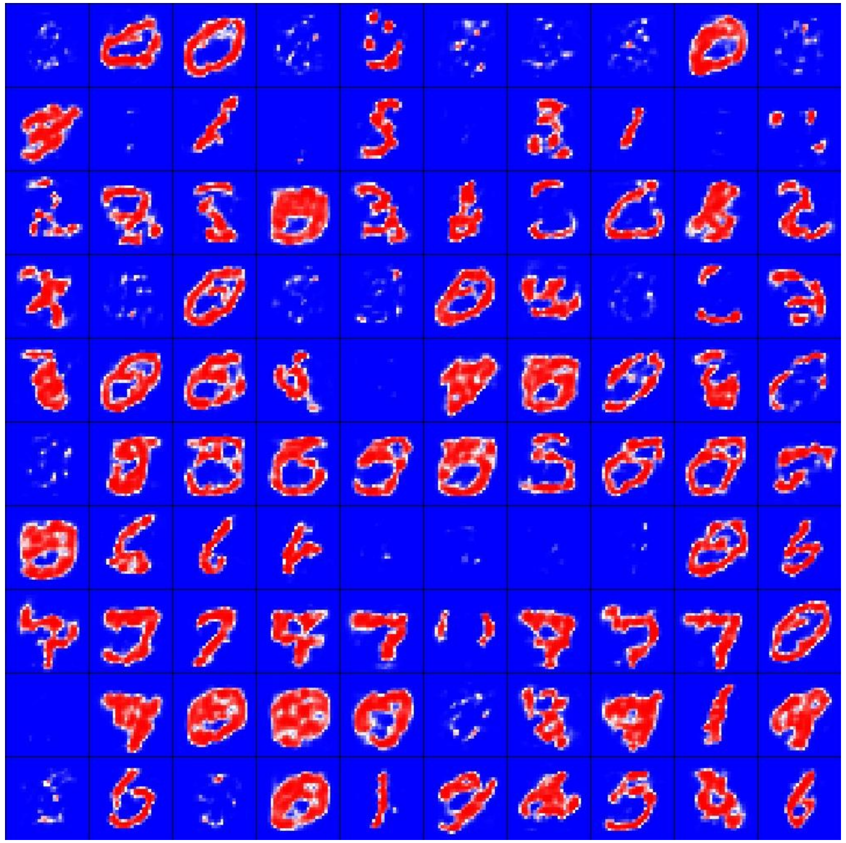
Evaluation - Recognition mode – Pre-training and fine-tuning

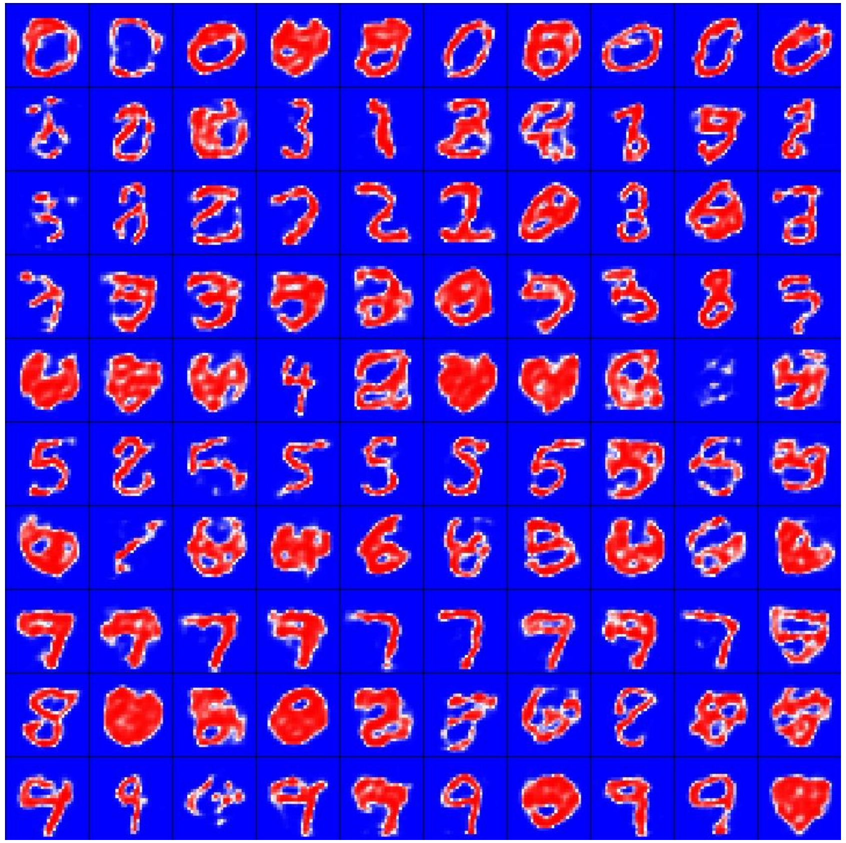
Dataset	Learning step	Accuracy
Training	Pre-training	0.888 ± 0.006
Training	Fine-tuning	0.924 ± 0.001
Test	Pre-training	0.891 ± 0.006
Test	Fine-tuning	0.925 ± 0.002

- Pre-training: very good performance considering the greedy approach
- Fine-tuning: improved performance

Deep Belief Nets

Evaluation - Generative mode – Pre-training and fine-tuning





- Pre-training: few good generated images
- Fine-tuning: qualitatively high improvement, many good generated images

Deep Belief Nets

Simplified architecture

Dataset	Learning step	Accuracy	Accuracy with simplified architecture
Training	Pre-training	0.888 ± 0.006	0.916 ± 0.001
Training	Fine-tuning	0.924 ± 0.001	0.942 ± 0.001
Test	Pre-training	0.891 ± 0.006	0.919 ± 0.003
Test	Fine-tuning	0.925 ± 0.002	0.945 ± 0.002

- The simplified architecture outperforms the deeper one
- Possible reasons:
  - Deeper network → more complex landscape of loss function → more local minima
  - Deeper network → small gradients (vanishing gradients problem)

THANK YOU！