Neural Networks

Unsupervised Learning

CompNeuro 2012

Unsupervised Learning

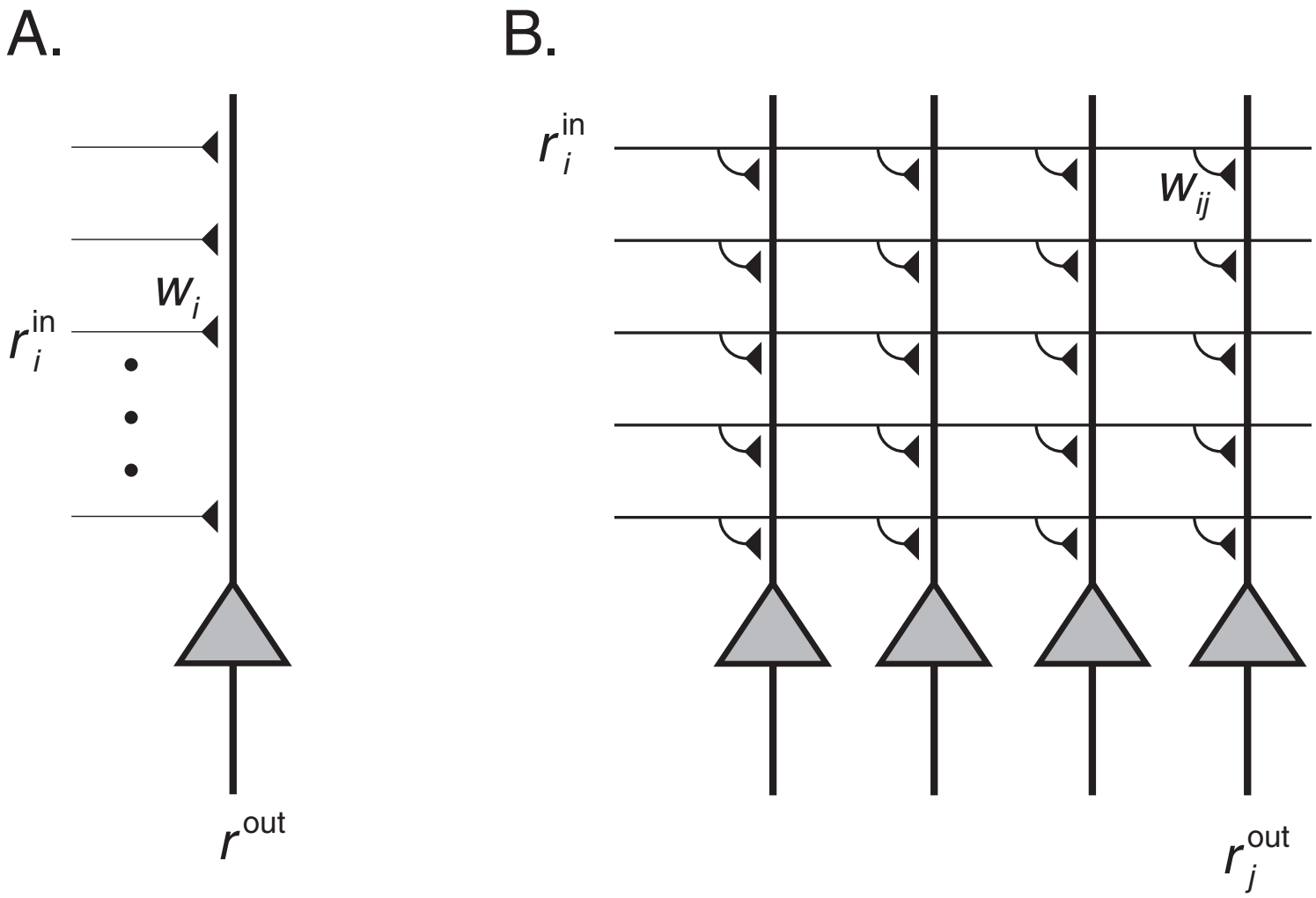
- networks restructure connections to model the correlational structure of the environment
- no explicit teaching signal or error signal

Hebbian Learning

- learning rule: $\delta w_{ij} = \gamma x_i y_y$
- weights adjusted in proportion to pre- and post-synaptic activities
- good for learning associations
- e.g. can account for classical conditioning

Hopfield Networks

- recurrent connections feed back outputs to input layers



Hopfield Networks

- input is presented to network
- network output pattern will resemble a previously learned input
- output is then fed back as input
- network repeats cycle

Hopfield Networks

- can be thought of as a dynamical system
- after presenting an input, system settles into a low-energy state
- equal to previously experienced (trained) input pattern
- pattern completion

Hopfield Networks

- simple model of human memory
- “content-addressible” memory
- MATLAB demo: hopfield.tgz

Self-Organizing Maps

- networks learn to classify continuous-valued inputs into discrete categories, **without an explicit teaching signal**
- networks become tuned to the statistical features of the input patterns such that a meaningful coordinate system is created
- same kind of idea as principal components analysis (PCA) or ICA

Self-Organizing Maps

- SOMs form topographic maps of input patterns
- spatial locations of neurons in a lattice are indicative of intrinsic similarities in features of the input space
- e.g. somatotopy in motor cortex, retinotopy in visual cortex, etc.

Self-Organizing Maps

- input space is mapped onto a 2-dimensional (usually) lattice of neurons
- for each input pattern, weights determine activities of neurons in lattice
- neuron with greatest activity defines the spatial location of a topological **neighborhood** of excited neurons
- excited neurons adjust local weights in relation to input pattern so that next time those neurons are excited even more
- “neighborhood function” defines how much each neuron cares about neighboring inputs
- individual neurons become “tuned” to particular features of the input space

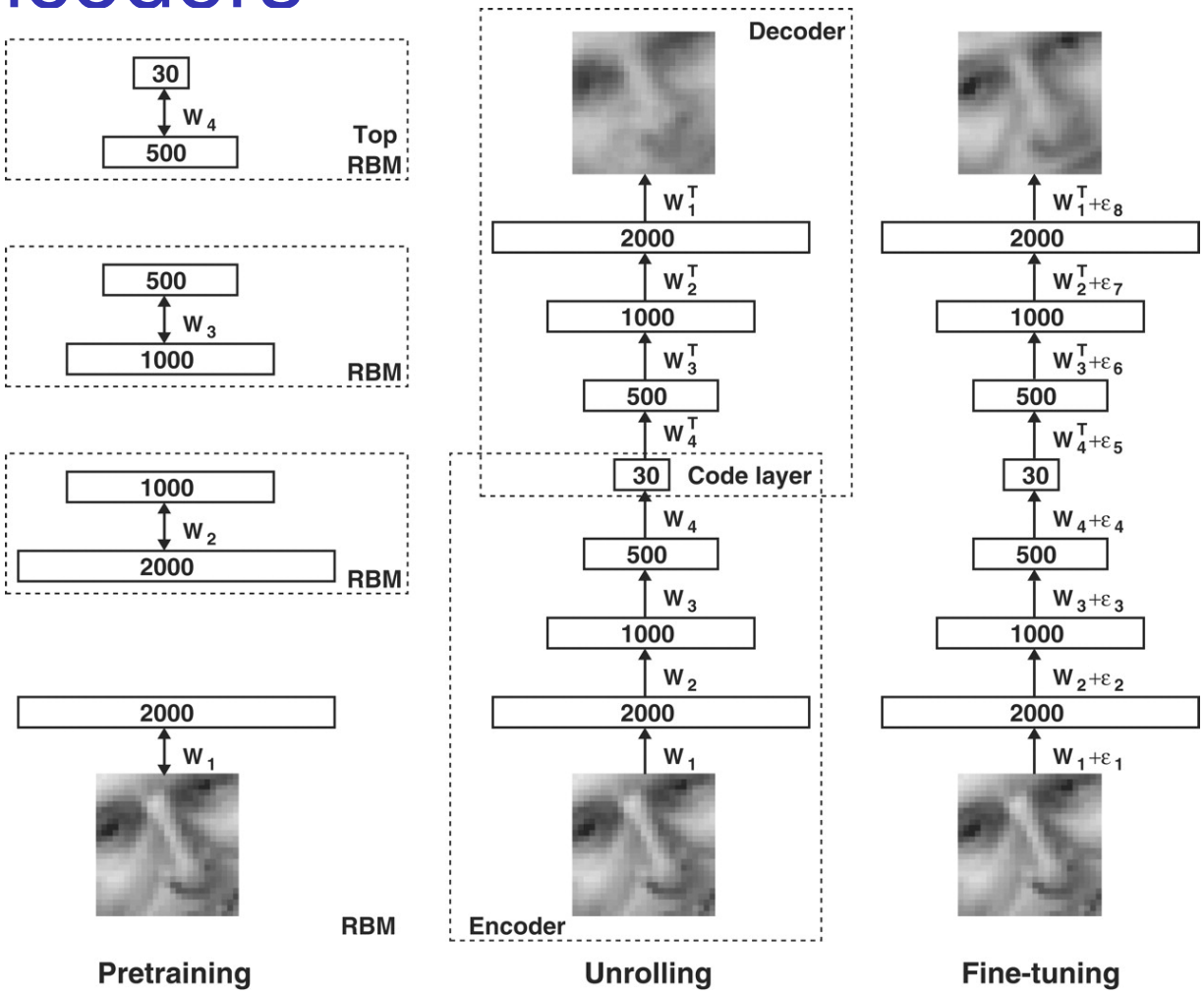
Self-Organizing Maps

- MATLAB demo code som1.m
- Aflalo, T. N., & Graziano, M. S. (2006). Possible origins of the complex topographic organization of motor cortex: reduction of a multidimensional space onto a two-dimensional array. Journal of Neuroscience 26(23), 6288-6297.

Autoencoders

- feedforward multi-layer neural network
- high-dimensional input (e.g. an image from a retina)
- low dimensional hidden layer(s)
- instead of low-dimensional output (e.g. N classes), **output is same dimensionality as input**
- train network to reproduce input, as output
- what makes this interesting: low-dimensional hidden layers act as bottleneck, forcing the network to represent the inputs in a lower-dimensional space

Autoencoders



Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.

Boltzmann Machines

- Boltzmann machine: stochastic neural network
- generative model
- neurons fully interconnected
- binary units fire with probability dependent on input
- network settles in low energy states
- much like a hopfield network

Restricted Boltzmann Machine

- RBMs like BMs but neurons not fully interconnected
- RBMs organized into layers
- one layer fully connected to next layer but no connections within a layer
- through training of weights, higher layer learns to represent lower layer
- training similar to hebbian learning rule

Deep Belief Networks

- use RBMs and simple layer-to-layer training rule to efficiently train “deep” networks with many hidden layers
- works much better than backprop and gradient descent
- networks run in reverse can generate inputs
- see Hinton demos, video and demo code on course webpage