In this lecture

- Hopfield networks
- Feed forward neural networks
- Backpropagation
- Deep neural networks
- Autoencoders
Hopfield networks

Recurrent (feedback) network

Example of memory

Set weights (Hebbian rule): \( w_{ij} = \sum_{\alpha=1}^{m} \frac{x_i^\alpha x_j^\alpha}{mN} \)

Update rule (synchronous or asynchronous): \( x_i \leftarrow 2 \theta(\sum_j w_{ij} x_j) - 1 \)

Also continuous version...
Feedforward neural networks

Parallel/serial stack of single neuron units (cf. perceptron):

$$h_i = f(w_{ij}x_j + b_i)$$

Activation function $f$ can be: step, sigmoid, ReLU, gaussian, ...

Define cost function $C(w)$, then optimize through backpropagation.
Backpropagation

Given a network and some new example:

- Calculate current predictions
- Calculate cost of current prediction (supervised/unsupervised)
- Calculate gradients with chain rule

Can have problems with vanishing gradients.
NN concepts

- Softmax
- Dropout
- Regularization (weight decay)
- Batch normalization
- Early stopping
Simplest deep neural network

Stack multiple hidden units.

Why does this work?
Autoencoders

Unsupervised network to learn compressed representation.
That’s all for now...