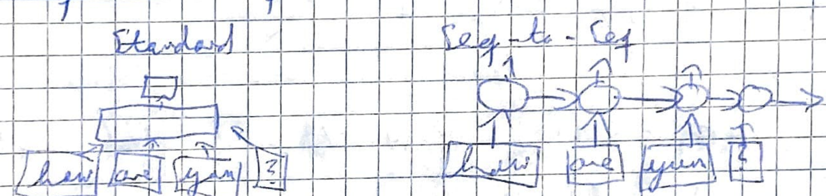
Deep Learning 2 (Structured Prediction)

Why structured output?

- usually NNs predict real values or classes
- sometimes, more complex and structured outputs, e.g. sentences
- output has constraints
- structured output NN examples: sequence-to-sequence models, mixture density networks, energy-based learning

Sequence-to-sequence models: recurrent neural network



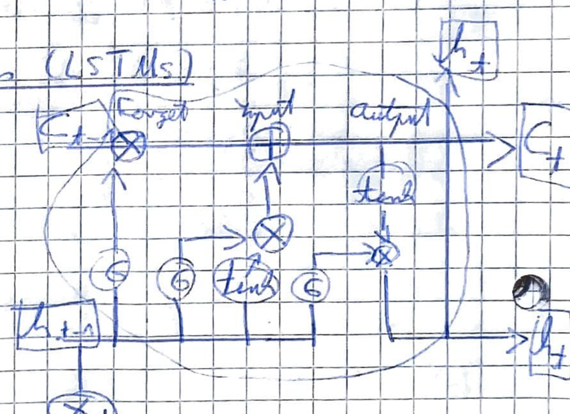
$h_t = f(h_{t-1}, x_t)$

$h_{t+1} = f(h_t, x_{t+1})$

⋮

Long Short Term Memories (LSTMs)

- core dynamics accessed via gating functions $\otimes$
- cell output h_t → prediction
  ↳ reinjection to the cell



Mixture Density Networks

$$p(t|x) = \sum_{i=1}^{m} \alpha_i(x)\, \phi_i(t|x) \qquad \phi_i(t|x) = \frac{1}{(2\pi)^{\frac{c}{2}} \sigma_i(x)^c} e^{-\frac{\|t-\mu_i(x)\|^2}{2\sigma_i(x)^2}}$$

- specialized for predicting conditional probabilities
- output parameters for distribution

Gaussian Mixture Models → local distribution
Boltzmann Machines → global distributions

Energy-Based Learning

- outputs a score for each input - output pair
  (like kernel-based structured prediction)

Structured Prediction: Kernels vs. Networks

Pro Kernel:	Pro NN:
- margin is maximized in well defined feature space $\phi(x)$	- more flexibility in representation extraction
- convex	- output can be continuous
	- inference via gradient descent

Summary:

- NNs need to be adapted to perform structured prediction (like kernel machines)
- simple: mixture density networks, conditional RBMs
- more general: energy-based learning