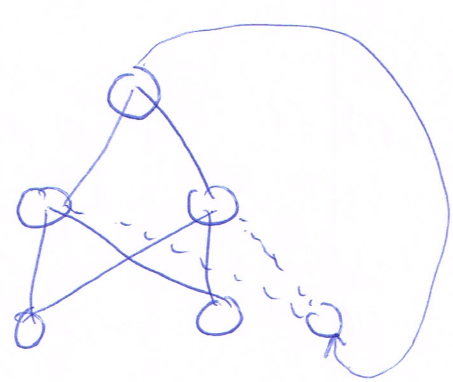
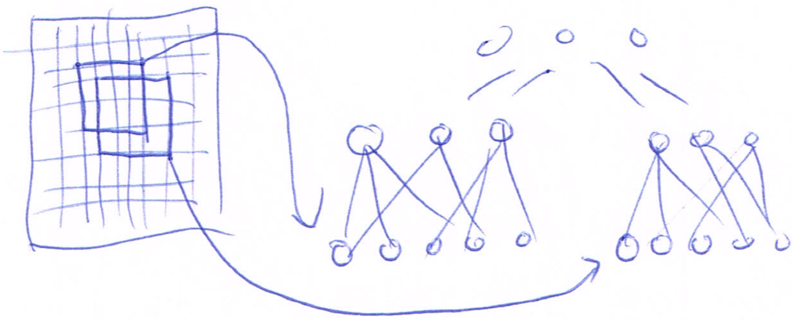
April 20

# Modern neural nets

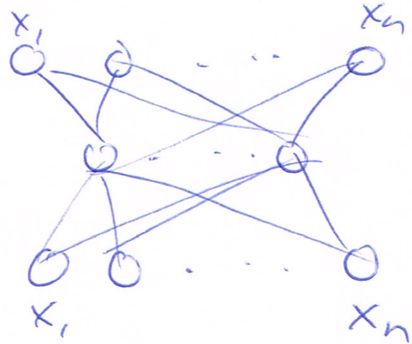
1. Minimize other objective functions
   - $Loss_{\bar{w}}(\bar{x}) = -h_{\bar{w}}(\bar{x}) \ln f(\bar{x}) - (1 - h_{\bar{w}}(\bar{x}))(1 - f(\bar{x}))$
     - Nice properties for optimization
   - $Cost_{\bar{w}}$ = Empirical Loss of $\bar{w}$ + $\lambda$ Complexity of $\bar{w}$
     - Balance accuracy and simplicity
     - "Regularization"
2. Other activation functions
   - Rectified linear units
     $$f(x) = \max(0, x)$$
     - addresses vanishing/exploding gradients
     - biological plausibility
   - Softmax
     $$f_i(\bar{x}) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$
3. Recurrent neural nets



4. Convolutional neural nets

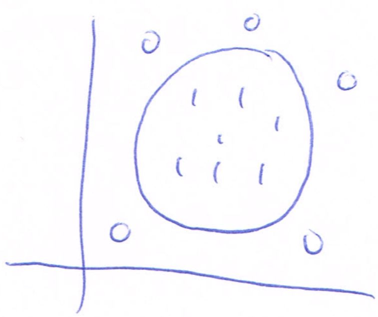


5. Autoencoders




"Representation learning"

Rough intuition



$x_1^2 + x_2^2 \le 1$

Not linearly separable

Convert to

$u_1 = x_1^2$

$u_2 = x_2^2$

$u_3 = \sqrt{2}\, x_1 x_2$

Linearly separable

Classification learning: Classifying data

Given $D = (\bar{x}^1, y^1), \dots, (\bar{x}^m, y^m)$ $\quad y^i = f(\bar{x}^i)$

$x^i \in \{0,1\}^n$ Boolean attributes

$y^i \in C$ Set of categories

$(x_1, \dots, x_n)$ New example

Predict $f(x_1, \dots, x_n)$

Bayesian learning:

Predict $\underset{c \in C}{\operatorname{argmax}}\, p(c \mid x_1, \dots, x_n)$

Bayes rule $\quad p(a \mid b) = \dfrac{p(b \mid a) \times p(a)}{p(b)}$

$$p(c \mid x_1, \dots, x_n) = \frac{p(x_1, \dots, x_n \mid c) \times p(c)}{p(x_1, \dots, x_n)}$$

$$\underset{c \in C}{\operatorname{argmax}}\, p(c \mid x_1, \dots, x_n)$$

$$= \underset{c \in C}{\operatorname{argmax}} \frac{p(x_1, \dots, x_n \mid c) \times p(c)}{p(x_1, \dots, x_n)}$$

Doesn't depend on c

$$= \underset{c \in C}{\operatorname{argmax}}\, p(x_1, \dots, x_n \mid c) \times p(c)$$

estimate from data

"Naive Bayes"

Conditional independence

$$p(a_1, a_2, \dots, a_n \mid b)$$

$$= p(a_1 \mid b) \times p(a_2 \mid b) \times \dots \times p(a_n \mid b)$$

$$= \prod_{i=1}^{n} p(a_i \mid b)$$