University of Amsterdam
Informatics Institute

AMLAB Amsterdam Machine Learning Lab

Machine Learning 1

Lecture 8.3 - Supervised Learning
Neural Networks - Losses

Erik Bekkers

(Bishop 5.2.0)

Image credit: Kirillm | Getty Images

Slide credits: Patrick Forré and Rianne van den Berg

Network Training

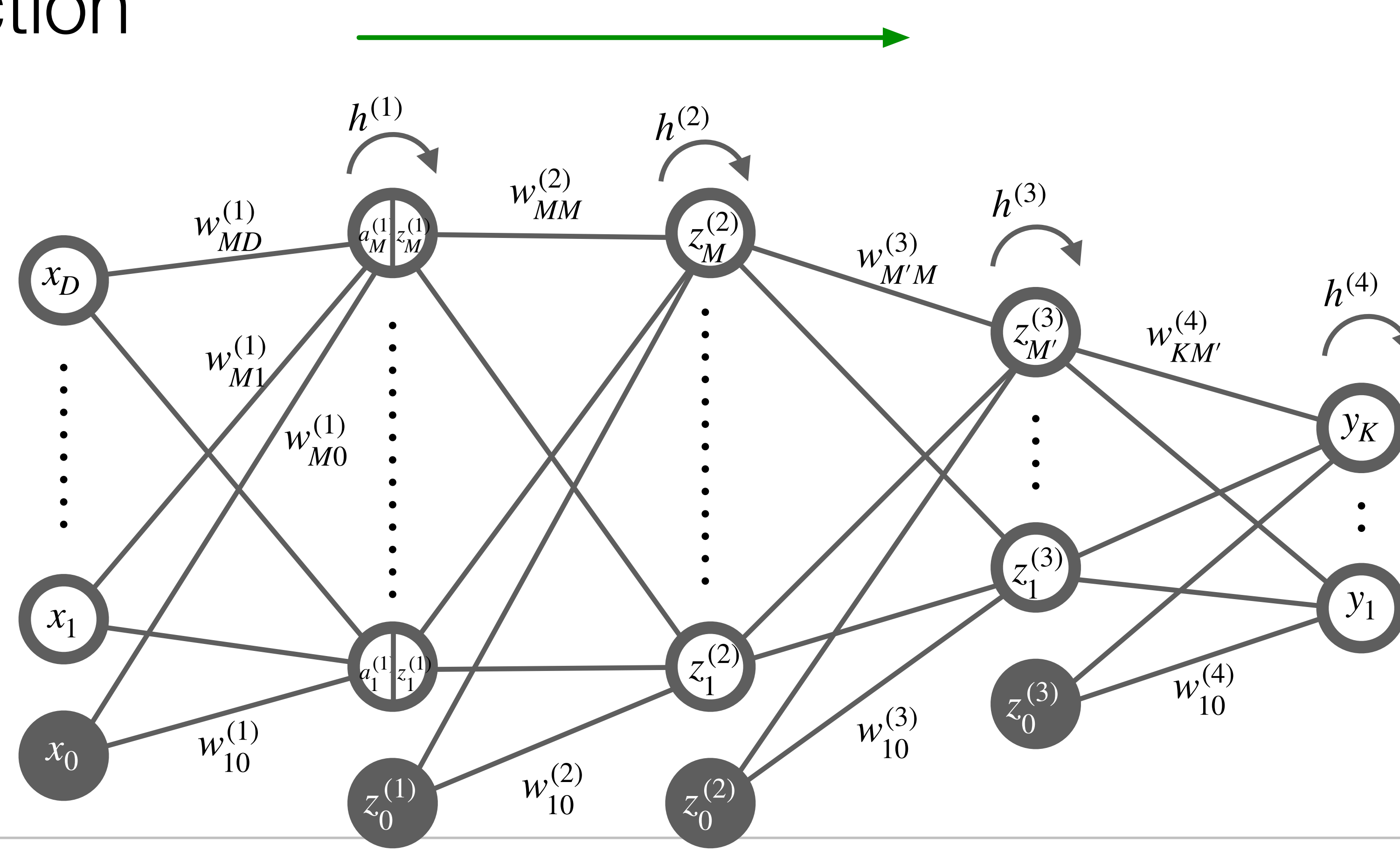
- Dataset: inputs $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)^T \quad \mathbf{x}_n \in \mathbb{R}^D$

- Use a probabilistic interpretation of the network outputs to choose

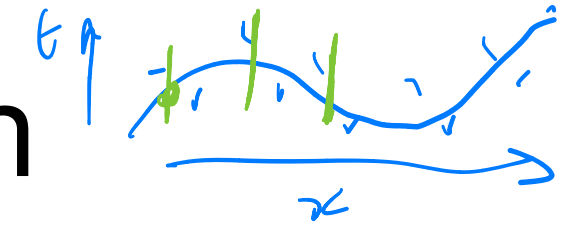
1. Number of outputs

2. Output activation function

3. Loss function!



Network Training: Regression



- Data: inputs $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)^T$, and targets $\mathbf{t} = (t_1, ..., t_N)^T$

  $\underline{x}_n \in \mathbb{R}^D$ $t_n \in \mathbb{R}$

- Assume target distribution: $p(t|\mathbf{x}, \mathbf{w}) =$ $\mathcal{N}(t \mid \underline{y(\underline{x}, w)}, \beta^{-1})$

  NN

- Single target —> Single output unit: $y(\mathbf{x}, \mathbf{w}) = h^{(L)}(a^{\mathrm{out}})$

- Targets are real valued: identity output activation function:

$$y(\mathbf{x}, \mathbf{w}) = h^{(L)}(a^{\mathrm{out}}) = a^{\mathrm{out}}$$

- Maximum Likelihood/minimum negative log likelihood:

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \frac{\beta}{2}\sum_{n=1}^{N}\{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2 - \frac{N}{2}\ln\beta + \frac{N}{2}\ln 2\pi$$

Equivalently: $$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2$$

Network Training: Binary Classification

- Data: inputs $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)^T$, and targets $\mathbf{t} = (t_1, ..., t_N)^T$

  $\underline{x}_n \in \mathbb{R}^D$ $t_n \in \{0, 1\}$

- Assume target distribution: $\overbrace{y(\mathbf{x}, \mathbf{w})}^{NN} = p(t = 1|\mathbf{x})$

  Bernoulli

  $$p(t|\mathbf{x}, \mathbf{w}) = y(\underline{x}, \underline{w})^t (1 - y(\underline{x}, \underline{w}))^{1-t}$$

- Single target —> Single output unit: $y(\mathbf{x}, \mathbf{w}) = h^{(L)}(a^{\text{out}})$

- Targets are binary: sigmoid output activation function:

  $$p(t=1|\underline{x}) = y(\mathbf{x}, \mathbf{w}) = h^{(L)}(a^{\text{out}}) = \sigma(a^{\text{out}}) \in [0, 1]$$

- Maximum Likelihood/minimum negative log likelihood:

  Cross-entropy loss

  $$E(\mathbf{w}) = -\sum_{n=1}^{N} t_n \ln y(\underline{x}_n, \underline{w}) + (1 - t_n) \ln(1 - y(\underline{x}_n, \underline{w}))$$

Network Training: Classification with K classes

one-hot encoding

- Data: inputs $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)^T$, and targets $\mathbf{T} = (\mathbf{t}_1, ..., \mathbf{t}_N)^T$

$\underline{x}_n \in \mathbb{R}^p$, $\underline{t}_n = (t_{n1}, t_{n2}, \ldots, t_{nK})^T = (0, \ldots, 1, \ldots, 0)^T$

- Assume target distribution: $p(\mathbf{t}_n|\mathbf{x}_n, \mathbf{w}) = \prod_{k=1}^{K} y_k(\underline{x}_n, \underline{w})^{t_{nk}}$

"Generalized Bernoulli"

$$y_k(\mathbf{x}, \mathbf{w}) = p(\mathcal{C}_k|\mathbf{x})$$

- K targets —> K output units: $y_k(\mathbf{x}, \mathbf{w}) = h^{(L)}(a_k^{\text{out}})$

$\sum_{k=1}^{K} p(C_k | \underline{x}) = y_k(\underline{x}, \underline{w}) = 1$

- Categorical targets: softmax output activation function

$$y_k(\mathbf{x}, \mathbf{w}) = h^{(L)}(\mathbf{a}^{\text{out}}) = \frac{\exp(a_k^{\text{out}})}{\sum_{j=1}^{K} \exp(a_j^{\text{out}})}$$

- Maximum Likelihood/minimum negative log likelihood:

cross-entropy loss

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(\underline{x}_n, \underline{w})$$

Losses overview

To minimize

- Regression
  - Assume Gaussian target distribution
  - NN makes prediction for the mean
  - Output activation is identity

Least squares errors

- Binary classification
  - Assume Bernoulli target distribution
  - NN makes prediction for probability for class 1
  - Output activation is logistic sigmoid

Cross-entropy loss

- Multi-class classification
  - Assume generalized Bernoulli target distribution
  - NN makes prediction for probability for each class
  - Output activation is soft max function

(Multi-class) Cross-entropy loss