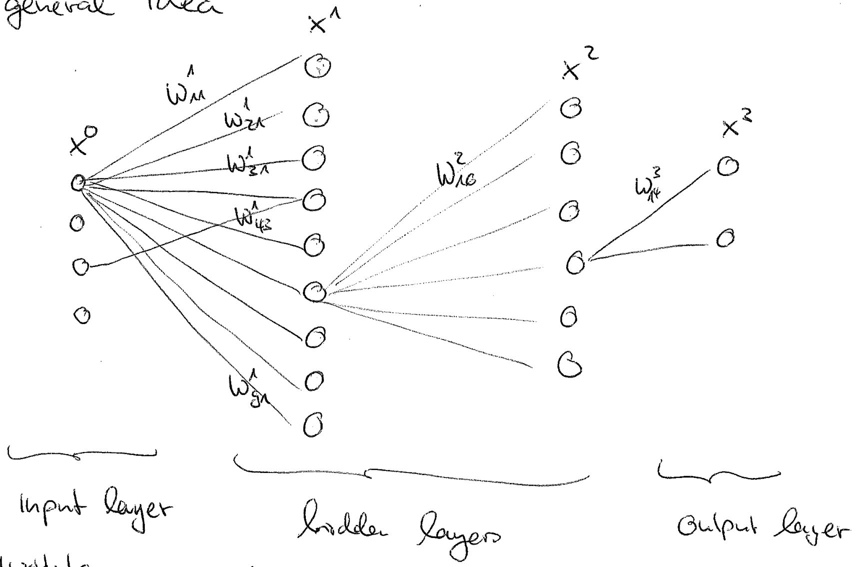
Multilayer Networks

general idea



networks's forward pass:

$$\mathbb{R}^{n^0} \ni x^0 \longmapsto \underbrace{\overset{\in \mathbb{R}^{n^1 \times n^0}}{W^1} \cdot x^1 + \overset{\in \mathbb{R}^{n^1}}{b^1}} =: z^1$$

$$\downarrow$$

$$x^1 := \sigma^1(z^1) \longmapsto \underbrace{W^2 x^1 + b^1} =: z^2$$

$$\downarrow$$

$$x^2 := \sigma^2(z^2) \longmapsto \underbrace{W^3 x^2 + b^3 =: z^3}$$

$$\downarrow$$

$$x^3 = \sigma^3(z^3)$$

- each layer takes as input x^{k-1} and applies the maps
$$x^{k-1} \mapsto z^k := W^k x^{k-1} + b^k \longrightarrow \sigma^k(z^k)$$
where $W^k \in \mathbb{R}^{n^k \times n^{k-1}}$, $b^k \in \mathbb{R}^k$, $\sigma^k : \mathbb{R}^k \to \mathbb{R}^k$
$(z_i)_{1 \le i \le k} \mapsto (\bar{\sigma}^k(z_i))_{1 \le i \le k}$
for some $\bar{\sigma}^k : \mathbb{R} \to \mathbb{R}$

- the output of an N-layer neural network is
$$x^N = \sigma(z^N) = \sigma(W^N x^{N-1} + b^N) = \dots = x^N(x^0)$$
this is what we usually called the hypothesis function $h(x^0) \equiv x^N(x^0)$

- given the training data $(x^{(i)}, y^{(i)})_{1 \le i \le M}$ with $x^{(i)} \in \mathbb{R}^{n^0}$ being the features and $y^{(i)} \in \mathbb{R}^{n^N}$ the "labels".
  - for classification we use the convention
  $$\text{class } i \in \mathbb{N} \longleftrightarrow \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{n^N}$$
  i-th position
  - for regression $y^{(i)} \in \mathbb{R}^{n^N}$ are the n^N continuous values of the output that is desired

After the forward pass we want to compare the output of the network to the labels of our training data and update the weights W^l and biases b^l accordingly:

$$L = \frac{1}{M} \sum_{i=1}^{M} l_i \quad \text{as loss function}$$

for some function l_i depending on the i-th pair $(x^{(i)}, y^{(i)})$, for example $l_i = \frac{1}{2}\left(y^{(i)} - x^N(x^{(i)})\right)^2$

How should the weights and biases be updated? (2)

Gradient descent: let p^k be a place holder for some parameter in the k-th layer, i.e. W^k_{ij}, b^k_i for $1 \le i \le n^k$, $1 \le j \le n^{k-1}$

$$\frac{\partial L}{\partial p^k} = \frac{1}{M}\sum_{i=1}^{M} \frac{\partial l_i}{\partial p^k}$$

recall $l_i = l(y^{(i)}, x^N(x^{(i)}))$

depending on all the weights W^l and biases b^l

$$\frac{\partial l}{\partial p^k} = \underbrace{\sum_{i=1}^{n^N} \frac{\partial}{\partial x_i^N} l(y, x^N)}_{=:\Delta(x^N)} \cdot \frac{\partial x_i^N}{\partial p^k}$$

$$\frac{\partial x^N}{\partial p^k} = \frac{\partial[\sigma^N(z^N)]}{\partial p^k} = \sum_{i=1}^{n^N} \sigma^{N\prime}(z_i^N)\,\frac{\partial z_i^N}{\partial p^k}$$

$$= \sum_{i=1}^{n^N} \sigma^{N\prime}(z_i^N)\,\frac{\partial\left[\sum_{j=1}^{n^{N-1}} W_{ij}^N x_j^{N-1} + b_i^N\right]}{\partial p^k}$$

$$= \sum_{i=1}^{n^N} \sigma^{N\prime}(z_i^N)\left[\sum_{j=1}^{n^{N-1}} W_{ij}^N \frac{\partial x_j^{N-1}}{\partial p^k} + \sum_{j=1}^{n^{N-1}} \frac{\partial W_{ij}^N}{\partial p^k}\, x_j^{N-1} + \frac{\partial b_i^N}{\partial p^k}\right]$$

We note that

$$\frac{\partial x^l}{\partial p^k} = 0 \quad \text{for} \quad l \le k$$

$$\frac{\partial W^l}{\partial p^k} = 0 = \frac{\partial b^l}{\partial p^k} \quad \text{for} \quad l \ne k$$

Hence, the last expression simplifies depending on layer number k:

$$\frac{\partial x^N}{\partial p^k} = \sigma^{N\prime}(z^N) \odot W^N \cdot \overbrace{\frac{\partial x^{N-1}}{\partial p^k}}^{\text{if } k \le N-1 \text{ this has to be expanded}}$$

$$+ \sigma^{N\prime}(z^N) \odot \underbrace{\left[\frac{\partial W^N}{\partial p^k}\, x^{N-1} + \frac{\partial b^N}{\partial p^k}\right]}_{\text{if } k \ne N \text{ this term is zero}}$$

for $k < N$:

$$\frac{\partial x^N}{\partial p^k} = \sigma^{N\prime}(z^N) \odot W^N \cdot \frac{\partial x^{N-1}}{\partial p^k}$$

$$= \sigma^{N\prime}(z^N) \odot W^N \cdot \left[\sigma^{N-1\prime}(z^{N-1}) \odot W^{N-1} \cdot \frac{\partial x^{N-1}}{\partial p^k}\right.$$

$$\left. + \sigma^{N-1\prime}(z^{N-1}) \odot \left(\frac{\partial W^{N-1}}{\partial p^k} \cdot x^{N-2} + \frac{\partial b^{N-1}}{\partial p^k}\right)\right]$$

and, ~~hence~~, we get the following structure:

$$\frac{\partial l}{\partial p^k} = \Delta(x^N) \cdot \left[\sigma^{N\prime}(z^N) \odot W^N\right] \cdot \frac{\partial x^{N-1}}{\partial p^k}$$

$$= \Delta(x^N) \cdot \left[\sigma^{N\prime}(z^N) \odot W^N\right] \cdot \left[\sigma^{N-1\prime}(z^{N-1}) \odot W^{N-1}\right]$$

$$\cdot \left[\dots\right] \cdot \left[\sigma^{k+1\prime}(z^{k+1}) \odot W^{k+1}\right]$$

$$\cdot \left[\sigma^{k\prime}(z^k) \odot \left(\frac{\partial W^k}{\partial p^k} \cdot x^{k-1} + \frac{\partial b^k}{\partial p^k}\right)\right]$$

An efficient update rule: Backpropagation

3

Algorithm: 1) Input $x^0 \leftarrow x^{(i)}$

2) Forward pass: $x^0 \mapsto z^1 \mapsto x^1 \mapsto \dots \mapsto z^N \mapsto x^N$

3) Compute $\Delta(x^N) = \frac{\partial}{\partial x_j} \ell(y_j^{(i)}, x_j) \Big|_{x_j = x^N_{j(x^{(i)})}}$

4) Compute backward pass

$$\Delta^N := \Delta(x^N)$$

$$\Delta^{k-1} := \Delta^k \cdot [\sigma^{k\prime}(z^k) \odot W^k]$$

5) Compute

$$\frac{\partial \ell_i}{\partial p^k} = \Delta^k \frac{\partial x^k}{\partial p^k}$$

$$= \Delta^k \left[\sigma^{k\prime}(z^k) \odot \left(\frac{\partial W^k}{\partial p^k} \cdot x^{k-1} + \frac{\partial b^k}{\partial p^k} \right) \right]$$

6) Compute ℓ_i for $i \in I$ (mini-batch $I \subseteq \{1 \dots n\}$) and average

$$\frac{\partial L_I}{\partial p^k} = \frac{1}{|I|} \sum_{i \in I} \frac{\partial \ell_i}{\partial p_k}$$

7) Update weights accordingly

$$W_{ij}^k \mapsto W_{ij}^k - \eta \frac{\partial L_I}{\partial W_{ij}^k}$$

$$b_i^k \mapsto b_i^k - \eta \frac{\partial L_I}{\partial b_i^k}$$

8) repeat for all mini-batches and epochs.

$\boxed{\text{HW}}$ Derive this backpropagation algorithm from the KKT condition for the Lagrangian formulation of:

$$\min_{W^k, b^k} \mathcal{L} = \frac{1}{n} \sum_{i=1}^{M} \ell(y^{(i)}, x^N)$$

under constraints $x^k = \sigma(W^k x^{k-1} + b^k)$

1) formulate Lagrangian

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{M} \ell(y^{(i)}, x^N) + \sum_{i=1}^{M} \sum_{k=1}^{N} \beta_i^k \cdot [x^k - \sigma(W^k x^{k-1} + b^k)]$$

2) $\frac{\partial \mathcal{L}}{\partial \beta_i}$ gives forward pass

3) $\frac{\partial \mathcal{L}}{\partial x_i}$ gives backward pass

4) $\frac{\partial \mathcal{L}}{\partial W^k}, \frac{\partial \mathcal{L}}{\partial b^k}$ give update rule