Classification and Artificial Neural Networks

Anders Hansson and Martin Andersen

Linköping University and Technical Unversity of Denmark

April 18, 2024

Classification

Pairs of data (y_k, x_k) for $k \in \mathbf{N}_N$ with $x_k \in \mathbf{R}^n$, where $y_k \in \mathbf{N}_K$ are called *qualitative*, *categorial*, *discrete* or *factors*.

We say that (y_k, x_k) belongs to class l if $y_k = l$.

Determine functions $f_l : \mathbf{R}^n \to \mathbf{R}$, $l \in \mathbf{N}_K$: $f_l(x_k) > 0$ if $y_k = l$ and $f_l(x_k) < 0$ if $y_k \neq l$.

The set $\{x \mid f_l(x) = 0\}$ separates class l from the other classes.

Classification using Linear Regression

Model each class $l \in \mathbf{N}_K$ as a linear regression

$$z_l = a_l^T x + b_l$$

where $a_l \in \mathbf{R}^n$ and $b_l \in \mathbf{R}$.

LS problem wsikth variables (a, b):

$$\text{minimize} \frac{1}{2} \sum_{l=1}^{K} \left(\sum_{k:y_k=l} \left(a_l^T x_k + b_l - 1 \right)^2 + \sum_{k:y_k \neq l} \left(a_l^T x_k + b_l \right)^2 \right)$$

where $a = (a_1, \ldots, a_K)$ and $b = (b_1, \ldots, b_K)$.

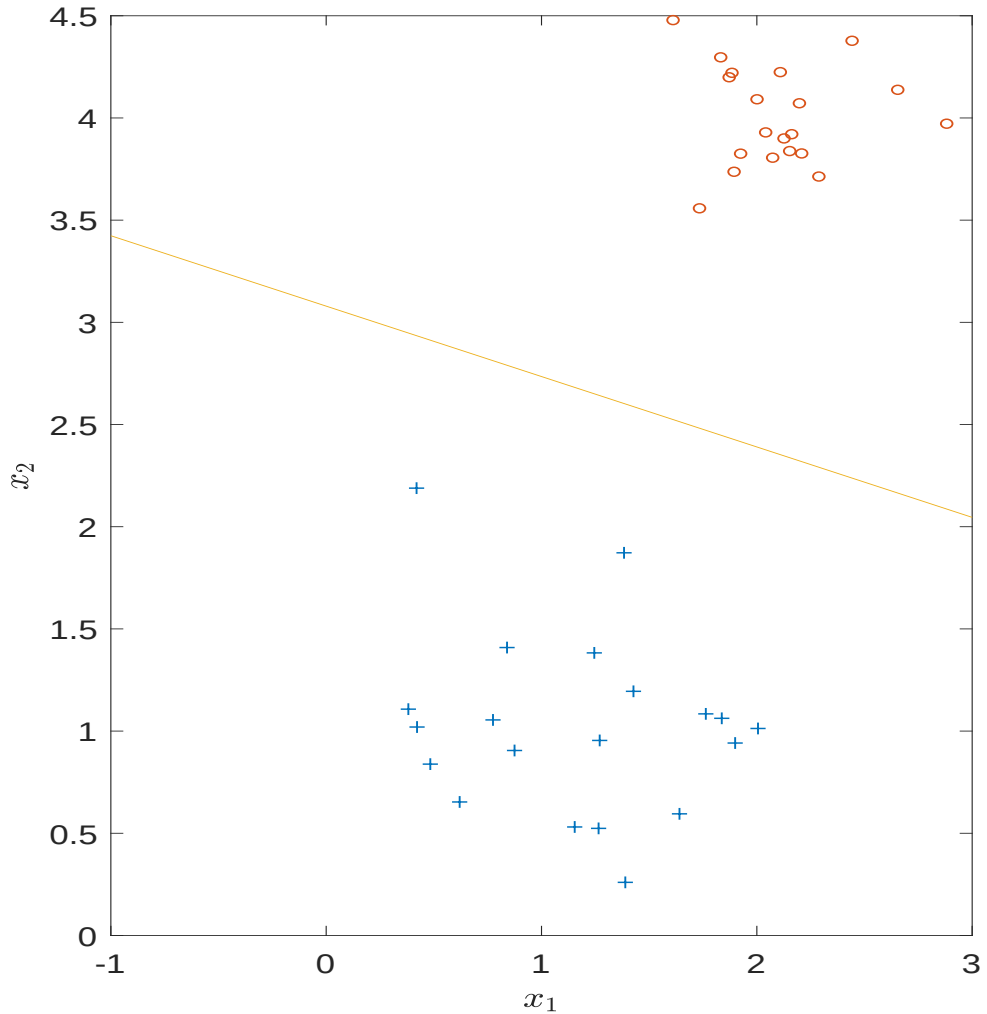
Discriminant functions: $\delta_l : \mathbf{R}^n \to \mathbf{R}$: $\delta_l(x) = a_l^T x + b_l$.

Classify x to belong to class l if $\delta_l(x) > \delta_k(x)$ for all $k \neq l$. Hence $f_l(x) = \delta_l(x) - \max_{k \neq l} \delta_k(x)$.

Prone to give bad results for $K \geq 3$. More general basis functions better

Example

Let $K = 2$ and $n = 2$. In total 20 data points from each class. In figure data points shown together with the line $\delta_1(x) = \delta_2(x)$.



Logistic Regression

Categorical distribution:

$$p_l(z_1, z_2, \ldots, z_{K-1}) = \begin{cases} \frac{e^{z_l}}{1+\sum_{k=1}^{K-1} e^{z_k}}, & l \in \mathbf{N}_{K-1} \\ \frac{1}{1+\sum_{k=1}^{K-1} e^{z_k}}, & l = K \end{cases}$$

where $p_l : \mathbf{R}^{K-1} \to [0,1]$ for $l \in \mathbf{N}_K$. Let $z_l = a_l^T x + b_l$ and define likelihood function $\ell : \mathbf{R}^{Nn} \times \mathbf{R}^{(K-1)(n+1)} \to [0,1]$:

$$\begin{aligned} &\ell(x_1, \ldots, x_N; a_1, \ldots, a_{K-1}, b_1, \ldots, b_{K-1}) \\ &= \prod_{l=1}^{K} \prod_{k:y_k=l} p_l(a_1^T x_k + b_1, \ldots, a_{K-1}^T x_k + b_{K-1}) \end{aligned}$$

Log-likelihood function:

$$\sum_{l=1}^{K-1} \sum_{k:y_k=l} a_l^T x_k + b_l - \sum_{k=1}^{N} \ln\left(1 + \sum_{l=1}^{K-1} e^{a_l^T x_k + b_l}\right)$$

is concave. Discriminant functions: $\delta_l : \mathbf{R}^n \to \mathbf{R}$ defined by $\delta_l(x) = p_l(a_1^T x + b_1, \ldots, a_{K-1}^T x + b_{K-1})$.

Separating Hyperplane

Given pairs of data (y_k, x_k), $k \in \mathbf{N}_N$ with $x_k \in \mathbf{R}^n$ and $y_k \in \{-1, 1\}$.

Find function $f : \mathbf{R}^n \to \mathbf{R}$ such that $f(x_k) > 0$ if $y_k = 1$ and $f(x_k) < 0$ if $y_k = -1$.

The set $\{x \mid f(x) = 0\}$ separates the two classes.

Let $f : \mathbf{R}^n \times \mathbf{R}^n \times \mathbf{R} \to \mathbf{R}$:

$$f(x; a, b) = a^T x + b$$

Objective: choose (a, b):

$$y_k f(x_k; a, b) > 0, \quad k \in \mathbf{N}_N$$

Feasibility problem which has solution iff the two classes can be separated with the *hyperplane* $\{x \mid a^T x + b = 0\}$.

Hebbian Learning

Inequalities homogeneous in (a, b). Equivalent to consider feasibility problem

$$y_k f(x_k; a, b) \geq 1, \quad k \in \mathbf{N}_N$$

Let $g : \mathbf{R} \to \mathbf{R}_+$ be the Rectifier Linear Unit (ReLU):

$$g(y) = \begin{cases} y, & y \geq 0 \\ 0, & y < 0 \end{cases}$$

and define equivalent optimization problem

$$\text{minimize} \sum_{k=1}^{N} g(1 - y_k f(x_k; a, b))$$

with variables (a, b). Has a solution with objective function value zero iff the two classes can be separated by a hyperplane. However, not unique. Also non-differentiable objective function.

Unique Separating Hyperplane

Maximize the distance from x_k to the hyperplane.

Distance given by $y_k f(x_k; a, b)/\|a\|_2$, and since f is homogeneous in (a, b) maximizing distance equivalent to minimizing $\|a\|_2^2$:

$$\text{minimize } \frac{1}{2}\|a\|_2^2 \tag{1}$$

$$\text{subject to } y_k f(x_k; a, b) \geq 1, \quad k \in \mathbf{N}_N \tag{2}$$

with variables (a, b). However, no solution if separating hyperplane not exists.

Differentiable Formulation

Let $\xi = (\xi_1, \ldots, \xi_N) \in \mathbf{R}^N$, and consider

$$\text{minimize} \quad \sum_{k=1}^{N} \xi_k \tag{3}$$

$$\text{subject to} \quad y_k f(x_k; a, b) \geq 1 - \xi_k, \quad k \in \mathbf{N}_N \tag{4}$$

$$\xi_k \geq 0, \quad k \in \mathbf{N}_N \tag{5}$$

with variables (a, b, ξ). However, possible non-uniqueness of solution.

Support Vector Machine (SVM)

Remedy:

$$\text{minimize} \quad \sum_{k=1}^{N} \xi_k + \frac{\nu}{2}\|a\|_2^2 \tag{6}$$

$$\text{subject to} \quad y_k f(x_k; a, b) \geq 1 - \xi_k, \quad k \in \mathbf{N}_N \tag{7}$$

$$\xi_k \geq 0, \quad k \in \mathbf{N}_N \tag{8}$$

with variables (a, b, ξ), where $\nu \geq 0$ provides tradeoff.

Remark: May use regressor f nonlinear in x_k, i.e.
$f : \mathbf{R}^m \times \mathbf{R}^n \times \mathbf{R} \to \mathbf{R}$ given by

$$f(x; a, b) = a^T \beta(x) + b$$

where $\beta : \mathbf{R}^m \to \mathbf{R}^n$

Duality

Lagrangian $L : \mathbf{R}^n \times \mathbf{R} \times \mathbf{R}^N \times \mathbf{R}^N \times \mathbf{R}^N \to \mathbf{R}$ for (6–8) is given by

$$L(a, b, \xi, \lambda, \mu) = \sum_{k=1}^{N} \xi_k + \frac{\nu}{2}\|a\|_2^2 \tag{9}$$

$$+ \sum_{k=1}^{N} \lambda_k \left(1 - \xi_k - y_k f(x_k; a, b)\right) - \sum_{k=1}^{N} \mu_k \xi_k \tag{10}$$

$$= \frac{\nu}{2}\|a\|_2^2 - \sum_{k=1}^{N} \lambda_k y_k a^T \beta(x_k) - \sum_{k=1}^{N} \lambda_k y_k b \tag{11}$$

$$+ \sum_{k=1}^{N} (1 - \lambda_k - \mu_k)\, \xi_k + \sum_{k=1}^{N} \lambda_k \tag{12}$$

Minimization of Lagrangian

Minimum of Lagrangian is unbounded from below unless $1 - \lambda_k - \mu_k = 0$, $k \in \mathbf{N}_N$ and $\sum_{k=1}^{N} \lambda_k y_k = 0$. When bounded minimizing a solution of

$$\frac{\partial L}{\partial a} = \nu a - \sum_{k=1}^{N} \lambda_k y_k \beta(x_k) = 0$$

i.e.

$$a = \frac{1}{\nu} \sum_{k=1}^{N} \lambda_k y_k \beta(x_k)$$

If (7) satisfied strictly at optimality for index k complementary slackness implies $\lambda_k = 0$. Hence optimal solution a only depends on the x_k for which the constraint is satisfied with equality, called *support vectors*.

Dual Problem

Lagrange dual function $g : \mathbf{R}^N \times \mathbf{R}^N \to \mathbf{R}$:

$$g(\lambda, \mu) = \sum_{k=1}^{N} \lambda_k - \frac{1}{2\nu} \sum_{k=1}^{N} \sum_{l=1}^{N} \lambda_k \lambda_l y_k y_l \beta^T(x_k)\beta(x_l)$$

with domain $\{(\lambda, \mu) | 1 - \lambda_k - \mu_k = 0,\ k \in \mathbf{N}_N,\ \sum_{k=1}^{N} \lambda_k y_k = 0\}$.

Slater's conditions fulfilled and we obtain optimal (λ, μ) from

$$\text{maximize } g(\lambda, \mu) \tag{13}$$

$$\text{subject to } \lambda_k + \mu_k = 1, \quad k \in \mathbf{N}_N \tag{14}$$

$$\sum_{k=1}^{N} \lambda_k y_k = 0 \tag{15}$$

$$\lambda_k \geq 0; \quad \mu_k \geq 0, \quad k \in \mathbf{N}_N \tag{16}$$

The so-called *sequential minimal optimization* algorithm extremely efficient.

Solution to Primal Problem from Dual

We already have a expressed in dual variables.

For any $\lambda_k > 0$, complementary slackness implies $1 - \xi_k - y_k f(x_k; a, b) = 0$, i.e.

$$b = \frac{-\xi_k + 1}{y_k} - a^T \beta(x_k), \quad k \in \mathcal{I}$$

where $\mathcal{I} = \{k \in \mathbf{N}_N | \lambda_k > 0\}$.

If $\mu_k > 0$, complementary slackness $\mu_k \xi_k = 0 \Rightarrow \xi_k = 0$. Hence there is a nonempty set $\mathcal{J} = \{k \in \mathcal{I} | \xi_k = 0\}$ for which

$$b = \frac{1}{y_k} - a^T \beta(x_k), \quad k \in \mathcal{J}$$

Any such k can be used to compute b.

For numerical stability take average.

Kernel Trick Again

Dual problem depends only on $\beta^T(x_k)\beta(x_l)$. Also true for resulting regressor:

$$f(x; a, b) = \frac{1}{\nu}\sum_{k=1}^{N} \lambda_k y_k \beta^T(x_k)\beta(x) + b$$

Hence sufficient to know $K : \mathbf{R}^m \times \mathbf{R}^m \to \mathbf{R}_+$ defined by $K(x, \bar{x}) = \beta^T(x)\beta(\bar{x})$.

Classification using Boltzmann Machine

Given data (y_k, x_k), $k \in \mathbf{N}_N$ with $x_k \in \{0,1\}^M$ and $y_k \in \{0,1\}^K$.

(y_k, x_k) belongs to class $l \in \mathbf{N}_K$ if $y_k = e_l$, where e_l is the lth unit vector.

We let $v_k = (y_k, x_k) \in \{0,1\}^{M+N}$ be the visible variables in the Boltzmann machine.

Use data (y_k, x_k), $k \in \mathbf{N}_N$ to find optimal values of the parameters (λ, Λ) in Ising distribution $p(y, x, h)$, where h are the hidden variables

Let $\delta_l(x) = \sum_h p(e_l, x, h)$ be discriminant function and say that x belongs to class l if $\delta_l(x) > \delta_k(x)$ for all $k \neq l$.

Restricted Boltzmann Machine (RBM)

Graphical Ising model which is obtained by considering a graph on the variables (v, h) such that there are no edges between the variables within v or h, respectively. Then

$$\Lambda = \begin{bmatrix} 0 & \Lambda_{12} \\ \Lambda_{12}^T & 0 \end{bmatrix}$$

which means Ising distribution reads:

$$p(v, h) = \frac{1}{Z} e^{\lambda_1^T v + \lambda_2^T h + 2 v^T \Lambda_{12} h}$$

where Z normalizes p to a distribution.

Factorization of Conditional Distribution

Marginal distribution of v:

$$\sum_h p(v,h) = \frac{e^{\lambda_1^T v}}{Z} \sum_h e^{(\lambda_2 + 2\Lambda_{12}^T v)^T h} = \frac{e^{\lambda_1^T v}}{Z} \sum_h \prod_i e^{(\lambda_2 + 2\Lambda_{12}^T v)_i^T h_i}$$

$$= \frac{e^{\lambda_1^T v}}{Z} \prod_i \sum_{h_i} e^{(\lambda_2 + 2\Lambda_{12}^T v)_i^T h_i} = \frac{e^{\lambda_1^T v}}{Z} \prod_i \left(1 + e^{(\lambda_2 + 2\Lambda_{12}^T v)_i^T}\right)$$

Conditional distribution:

$$\frac{p(v,h)}{\sum_h p(v,h)} = \frac{\frac{1}{Z} e^{\lambda_1^T v + \lambda_2^T h + 2v^T \Lambda_{12} h}}{\frac{e^{\lambda_1^T v}}{Z} \prod_i \left(1 + e^{(\lambda_2 + 2\Lambda_{12}^T v)_i^T}\right)} = \frac{\prod_i e^{(\lambda_2 + 2\Lambda_{12}^T v)_i h_i}}{\prod_i \left(1 + e^{(\lambda_2 + 2\Lambda_{12}^T v)_i^T}\right)}$$

Conditional distribution for h_j given v:

$$s_j(h_j, v) = \frac{e^{(\lambda_2 + 2\Lambda_{12}^T v)_j h_j}}{\prod_i \left(1 + e^{(\lambda_2 + 2\Lambda_{12}^T v)_i^T}\right)}$$

Logistic Function

We have

$$s_j(0, v) = \frac{1}{\prod_i (1 + e^{x_i})}; \qquad s_j(1, v) = \frac{e^{x_j}}{\prod_i (1 + e^{x_i})}$$

respectively, where $x_i = (\lambda_2 + 2\Lambda_{12}^T v)_i$. Since $s_j(0, v) + s_j(1, v) = 1$, we have that $\prod_i (1 + e^{x_i}) = e^{x_j} + 1$, and hence

$$s_j(1, v) = \frac{e^{x_j}}{1 + e^{x_j}} = \frac{1}{1 + e^{-x_j}} = \sigma(x_j)$$

where $\sigma : \mathbf{R} \to \mathbf{R}$ is the *logistic function*, which is an example of a so-called *sigmoid function*.

Gradient Expressions

Assume only one observation v_k, which we write as v. Case of several observations is obtained by summing up the gradients for the different observations.[1]

Gradients:

$$\frac{\partial Q}{\partial \lambda} = \sum_h s(h, v, \lambda^-, \Lambda^-) \begin{bmatrix} v \\ h \end{bmatrix} - \sum_{\xi_v} p_v(\xi_v) \sum_{\xi_h} s(\xi_h, \xi_v, \lambda^-, \Lambda^-) \begin{bmatrix} \xi_v \\ \xi_h \end{bmatrix}$$

$$\frac{\partial Q}{\partial \Lambda_{12}} = 2 \sum_h s(h, v, \lambda^-, \Lambda^-) v h^T - \sum_{\xi_v} p_v(\xi_v) \sum_{\xi_h} s(\xi_h, \xi_v, \lambda^-, \Lambda^-) \xi_v \xi_h^T$$

where $s(h, v, \lambda^-, \Lambda^-) = \prod_k s_k(h_k, v)$, $\xi = (\xi_v, \xi_h)$, and where $p_v(\xi_v) = \sum_{\xi_h} p(\xi_v, \xi_h)$ marginal distribution for the observations.

[1] Sub-indexes will from now on refer to component of vectors.

Simplifications

We have:

$$\sum_h s(h, v, \lambda^-, \Lambda^-) h_l = \sum_h \prod_k s_k(h_k, v) h_l = \sum_h \prod_{k \neq l} s_k(h_k, v) s_l(h_l, v) h_l$$
$$= \prod_{k \neq l} \sum_{h_k} s_k(h_k, v) \sum_{h_l} s_l(h_l, v) h_l = s_l(1, v)$$

where the last equality follows from the fact that $s_k(0, v) + s_k(1, v) = 1$ and $s_l(0, v) \times 0 = 0$, and hence

$$\frac{\partial Q}{\partial \lambda_1} = v - \sum_{\xi_v} p_v(\xi_v) \xi_v$$

$$\left(\frac{\partial Q}{\partial \lambda_2} \right)_i = s_i(1, v) - \sum_{\xi_v} p_v(\xi_v) s_i(1, \xi_v)$$

$$\left(\frac{\partial Q}{\partial \Lambda_{12}} \right)_{i,j} = 2 s_j(1, v) v_i - \sum_{\xi_v} p_v(\xi_v) s_j(1, \xi_v) (\xi_v)_i$$

Contrastive Divergence Method

Approximated second term in gradients using Gibbs sampling from p_v.

1. Initialize Gibbs sampler with $\xi_v^{(0)} = v$.
2. Because of graphical structure of RBM we then draw a sample $h^{(1)}$ from the conditional distribution of h given v, i.e. from $s_j(h_j, \xi_v^{(0)})$.
3. Draw a sample $\xi_v^{(1)}$ from the conditional distribution of v given h, where we use $h = h^{(1)}$.[2]

Approximation of sums for gradients:

$$\sum_{\xi_v} \xi_v^{(1)}$$

$$\sum_{\xi_v} s_i(1, \xi_v^{(1)})$$

$$\sum_{\xi_v} s_j(1, \xi_v^{(1)})(\xi_v^{(1)})_i$$

[2] This conditional distribution can be obtained similarly to $s_j(h_j, v)$ above.

Artificial Neural Networks (ANN)s

We consider an ANN with L layers and n_i neurons for layer i.

Let $x_i \in \mathbf{R}^{n_i}$ for $i \in \mathbf{Z}_{L-1}$ be the *input activation* of layer i, and let

$$z_i = \Phi_i(x_{i-1}), \quad i \in \mathbf{N}_L,$$

where $z_i \in \mathbf{R}^{n_i}$ be the *output* of layer i.

The function $\Phi_i : \mathbf{R}^{n_{i-1}} \to \mathbf{R}^{n_i}$ is called the *propagation function*. Typically

$$\Phi_i(x) = W_i x + v_i, \quad i \in \mathbf{N}_L, \tag{17}$$

where $W_i \in \mathbf{R}^{n_i \times n_{i-1}}$ and $v_i \in \mathbf{R}^{n_i}$.

The input to the next layer is obtained by

$$x_i = h_i(z_i), \quad i \in \mathbf{N}_{L-1}, \tag{18}$$

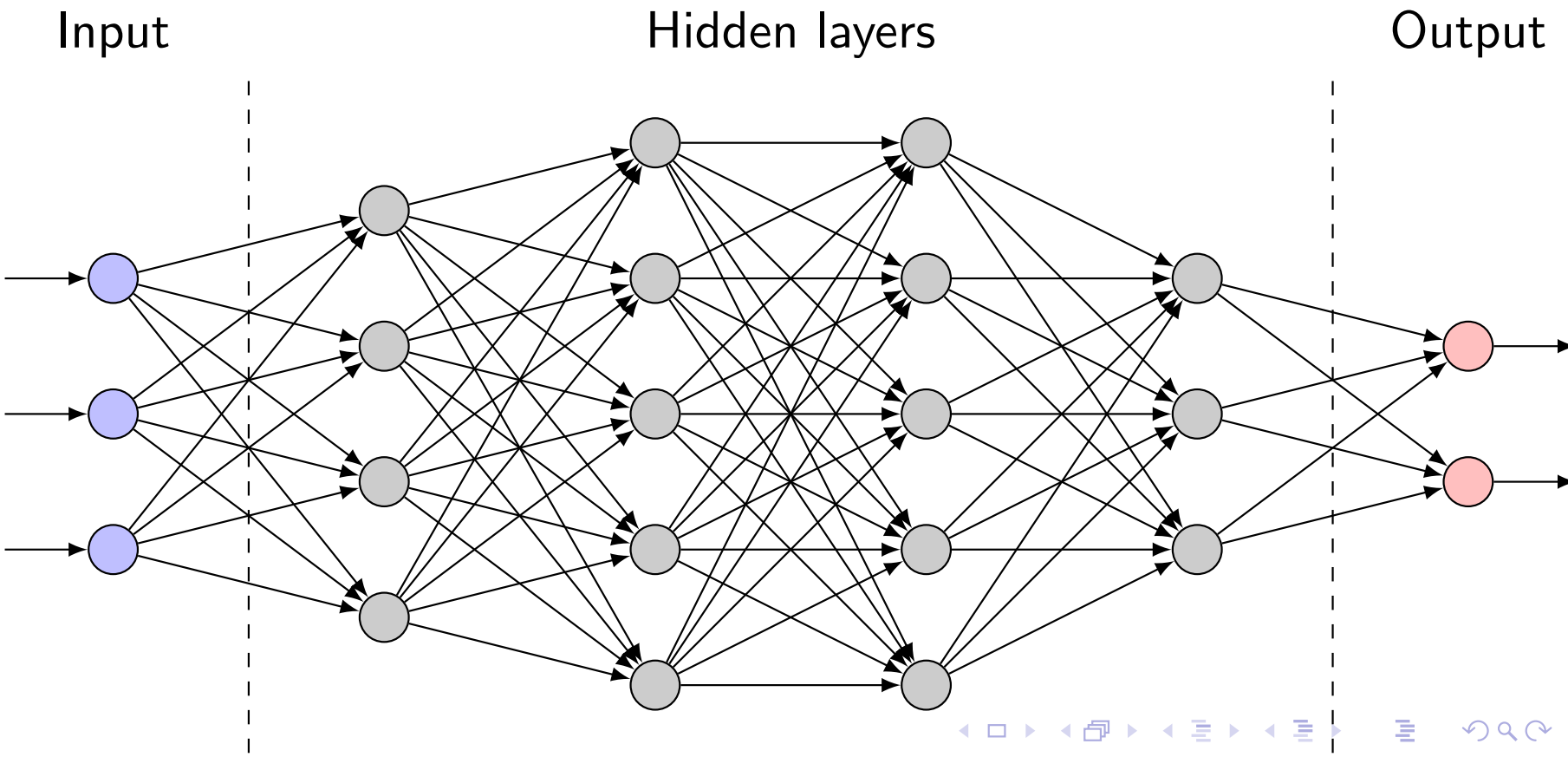
where $h_i : \mathbf{R}^{n_i} \to \mathbf{R}^{n_i}$ is called the *activation function*.

Activation Function

Often

$$h_i(z) = (h_{i1}(z_1), \ldots, h_{in_i}(z_{n_i}))\,, \tag{19}$$

Each component h_{ij} is typically saturation function or sigmoid function, *i.e.*, a function $\sigma : \mathbf{R} \to [0, 1]$ such that $\lim_{t\to\infty} \sigma(t) = 1$ and $\lim_{t\to-\infty} \sigma(t) = 0$. Another popular choice is the ReLU.



Predictor

Let $f_i : \mathbf{R}^{n_{i-1}} \times \mathbf{R}^{p_i} \to \mathbf{R}^{n_i}$, $i \in \mathbf{N}_L$, as

$$f_i(x_{i-1}, \theta_i) = h_i(\Phi_i(x)) = h_i(W_i x_{i-1} + v_i), \tag{20}$$

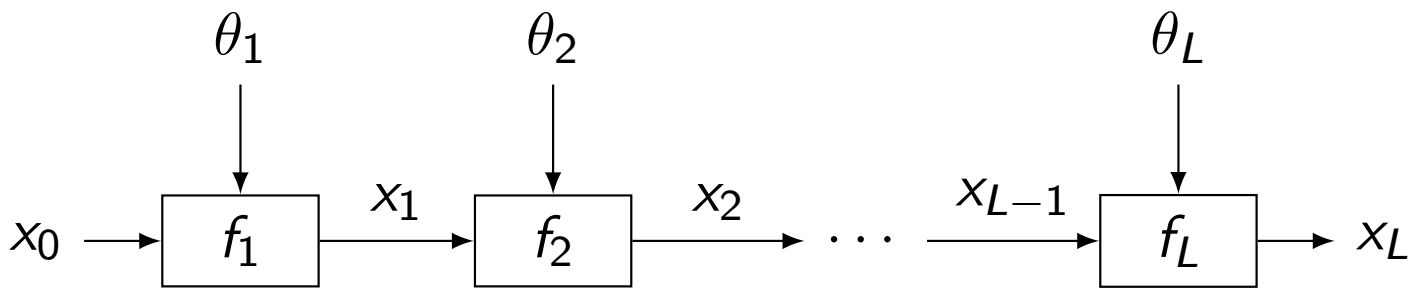
where

$$\theta_i = \mathbf{vec}\left(\begin{bmatrix} W_i & v_i \end{bmatrix}\right) \in \mathbf{R}^{p_i}. \tag{21}$$

Then

$$x_i = f_i(x_{i-1}; \theta_i), \quad i \in \mathbf{N}_L, \tag{22}$$

Let $f : \mathbf{R}^{n_0} \times \mathbf{R}^{p_1} \times \cdots \times \mathbf{R}^{p_L} \to \mathbf{R}^{n_L}$ where $f(x_0; \theta_1, \ldots, \theta_L) = x_L$. With $x = x_0 \in \mathbf{R}^{n_0}$ and $\theta = (\theta_1, \ldots, \theta_L) \in \mathbf{R}^p$, where $p = \sum_{i=1}^{L} p_i$, we have defined a nonlinear regression model or predictor $f(x; \theta)$.



Representation and Approximation Capability

By Kolmogorov (1957) and Lorentz (1976) any continuous function $f : [0,1]^n \to \mathbf{R}$ can be represented as

$$f(x) = \sum_{q=0}^{2n} g\left(\sum_{p=1}^{n} \phi_{p,q}(x_p)\right) \tag{23}$$

where $\phi_{p,q} : [0,1] \to [0,1]$ are continuous increasing functions, and where $g : \mathbf{R} \to \mathbf{R}$ is a continuous function, Hence a two layer ANN is sufficient.

By Cybenko (1989) any continuous function $f : [0,1]^n \to \mathbf{R}$ can be approximated arbitrarily well with

$$f(x) = \sum_{j=1}^{N} \alpha_j \sigma(a_j^T x + b_j) \tag{24}$$

with any continuous sigmoid function, where $\alpha_j \in \mathbf{R}$, $a_j \in \mathbf{R}^n$ and $b_j \in \mathbf{R}$, $j \in \mathbf{N}_N$.

Regression with ANN

Regression using ANNs is about selecting a $\theta \in \mathbf{R}^p$ such that for pairs of data $(x_k, y_k) \in \mathbf{R}^{n_0} \times \mathbf{R}^{n_L}$, $k \in \mathbf{N}_N$, it holds that the output $f(x_k; \theta)$ of the ANN is close to y_k.

The goodness of the closeness is measured with a function $V : \mathbf{R}^p \to \mathbf{R}$ which could typically be the sum of squared norms of $y_k - f(x_k; \theta)$, *e.g.*,

$$V(\theta) = \frac{1}{2} \sum_{k=1}^{N} \|y_k - f(x_k; \theta)\|_2^2, \tag{25}$$

the minimization of which is a nonlinear LS problem.

Interpretation of Logistic Regression as ANN

Let

$$W(1) = \begin{bmatrix} a_1^T \\ \vdots \\ a_{K-1}^T \end{bmatrix}; \qquad v(0) = \begin{bmatrix} b_1 \\ \vdots \\ b_{K-1} \end{bmatrix}$$

and $h(z) = (z(1)/s, z(2)/s, \ldots, 1/s)$, where $s = 1 + \sum_{k=1}^{K-1} e^{z(k)}$.

Hence output x_1 of the ANN are the probabilities $p_l(z_1, z_2, \ldots, z_{K-1})$, $l \in \mathbf{N}_K$ of the categorial distribution, i.e. $x_{1l} = p_l(z_1, z_2, \ldots, z_{K-1})$.

The training data is $(x_k, y_k) \in \mathbf{R}^n \times \{0, 1\}$ for $k \in \mathbf{N}_N$

Function to maximize is the likelihood function for the data.

Interpretation of RBM as ANN

Let $x_0 = v$, $v_1 = \lambda_2$, $W_1 = 2\Lambda_{12}^T$ and $h_i(z_i) = \sigma(z_i$.

Then $x_{1i} = s_i(1, v)$, and hence the outputs of the ANN are the conditional probabilities for $h_i = 1$ given v or equivalently the expected values of the hidden variables given the visible variables.

Training data is $v = (v_1, \ldots, v_N)$ for the visible layer of the RBM

Function to maximize is the likelihood function for the data.

Optimization can be performed using the EM algorithm, and the outputs of the ANN are used to compute gradients.

Interpretation of Hebbian Learning as ANN

Let $v_1 = b$, $W_1 = a^T$ and $h(z_1) = g(z_1)$.

Training data is $(x_k, y_k) \in \mathbf{R}^n \times \{-1, 1\}$, $k \in \mathbf{N}_N$

Function to maximize is the sum of the outputs of the ANN over all training data.

Consequence

Because of these interpretations it easy to see how one can generalize logistic regression, the RBM, and the perceptron using multi-layer ANNs.

Interpolation of Data

Consider linear regression problem where the pairs of data (x_k, y_k) with $x_k \in \mathbf{R}^n$ and $y_k \in \mathbf{R}$, $k \in \mathbf{N}_N$ should satisfy the linear regression

$$y_k = a^T x_k$$

This means that we are able to *interpolate* the data. This is only possible if $N \leq n$; common practice when using ANNs.

Collect all pairs of data in

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix} \in \mathbf{R}^{N \times n}; \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \in \mathbf{R}^N$$

Then a should satisfy

$$Xa = y$$

which has a non-unique solution since the linear system of equations is under-determined.

Minimum norm solution

Look for a solution that minimizes the norm $\|a\|_2$, i.e.

$$a = X^T \left(XX^T \right)^{-1} y$$

Solution can be computed using an incremental stochastic optimization method on

$$\text{minimize} \sum_{k=1}^{N} V_k(a^T x_k, y_k)$$

with variable a. Here $V_k : \mathbf{R} \times \mathbf{R} \to \mathbf{R}$ are functions such that $V_k(0,0) = 0$ and such that the incremental stochastic optimization method converges to an a such that $Xa = y$. This holds true for many convex functions. One example is

$$V_k(z_k, y_k) = \frac{1}{2} \|y_k - z_k\|_2^2 \tag{26}$$

which corresponds to the LS criterion.

Minimum norm solution

Let $f_k(a) = V_k(a^T x_k, y_k)$. Then the incremental stochastic gradient method reads

$$a^{k+1} = a^k - t_k \nabla f_{i_k}(a^k)$$

We have that

$$\nabla f_{i_k}(a) = \frac{\partial V_{i_k}(a^T x_{i_k})}{\partial z_{i_k}} x_{i_k}$$

and hence if $a^0 = 0$, we obtain

$$a = \sum_{k=1}^{N} \alpha_k x_k$$

for some $\alpha_k \in \mathbf{R}$ or equivalently $a = X^T \alpha$. Together with $Xa = y$ we obtain:

$$XX^T \alpha = y$$

which has a unique solution if X has full row rank. Then it follows that

$$a = X^T \left(XX^T\right)^{-1} y$$

Dropout

Consider $V_k(z, y)$ given by (26) and resulting LS problem

$$\text{minimize}_a \frac{1}{2} \sum_{k=1}^{N} (y_k - a^T x_k)^2$$

Dropout obtained by replacing a_i with $\delta_i a_i$, where δ_i is an outcome of a random variable Δ_i with a Bernoulli distribution, i.e. $P(\Delta_i = 1) = p_i$ and $P(\Delta_i = 0) = 1 - p_i$

Assume that Δ_i is independent of Δ_j for $i \neq j$. Let $\Delta = \mathbf{diag}(\Delta_i)$ and $\delta = \mathbf{diag}\, \delta_i$. We also assume that we have a different outcomes δ of Δ for each k that are independent of one another.

The kth term in the objective function is the outcome of $\frac{1}{2}(y_k - x_k^T \Delta a)^2$, but since we only need to analyze on fixed value of k we neglect all dependence on k and consider for $y \in \mathbf{R}$ and $x \in \mathbf{R}^n$ $V^d : \mathbf{R} \to \mathbf{R}$ defined as

$$V^d(a) = \frac{1}{2}(y - x^T \Delta a)_2^2$$

Predictor

Define the predictor $\hat{Y} : \mathbf{R}^n \times \{0,1\}^n \to \mathbf{R}$ as

$$\hat{Y}(x; \Delta_1, \ldots, \Delta_n) = x^T \Delta a$$

Not so useful, since it involves random variable Δ.

The *ensemble average predictor* $\hat{y} : \mathbf{R}^n \to \mathbf{R}$ defined as

$$\hat{y}(x) = E\,\hat{Y}(x) = x^T P a$$

where $P = \mathbf{diag}\, p_i$ is more useful.

Will running the incremental stochastic gradient method on terms of the form V^d result in a good ensemble average predictor?

Goodness of Ensemble Average Predictor

Introduce the function $V^e : \mathbf{R}^n \to \mathbf{R}$ as

$$V^e(a) = \frac{1}{2}(y - x^T P a)^2$$

which should be small for the ensemble average predictor to be good.

We will now compare the gradient of this function with the expected value of the gradient of V^d.

Remember that this expected value is what determines the behaviour of the incremental stochastic gradient method.

Gradients

We have

$$\frac{\partial V^e(a)}{\partial a} = -(y - x^T Pa)Px$$
$$\frac{\partial V^d(a)}{\partial a} = -(y - x^T \Delta a)\Delta x$$

From this we obtain

$$E\left(\frac{\partial V^d(a)}{\partial a}\right) = -yPx + E(\Delta x x^T \Delta)a$$

where element (i, j) of $E(\Delta x x^T \Delta)$ is given by

$$E\Delta_i \Delta_j x_i x_j = \begin{cases} p_i p_j x_i x_j, & i \neq j \\ p_i x_i^2, & i = j \end{cases}$$

This implies that

$$E(\Delta x x^T \Delta) = Pxx^T P + \mathbf{diag}(\sigma_i^2 x_i^2)$$

where $\sigma_i^2 = p_i(1 - p_i)$ is the variance of Δ_i.

Regularization

This means that we have

$$E\left(\frac{\partial V^d(a)}{\partial a}\right) = \frac{\partial V^e(a)}{\partial a} + \mathbf{diag}(\sigma_i^2 x_i^2)a$$

which is the gradient of

$$V^e(a) + \frac{1}{2}a^T \mathbf{diag}(\sigma_i^2 x_i^2)a \tag{27}$$

We see that the this is a Ridge regularization, and this explains how dropout implicitly provides regularization.

The largest possible regularization is obtained when $p_i = 1/2$, since this value maximizes σ_i^2.