# Deep Learning - Foundations and Concepts

## Chapter 4. Single-layer Networks: Regression

nonlineark@github

February 13, 2025

# Outline

# Basis functions

Consider the linear combinations of fixed nonlinear functions of the input variables:

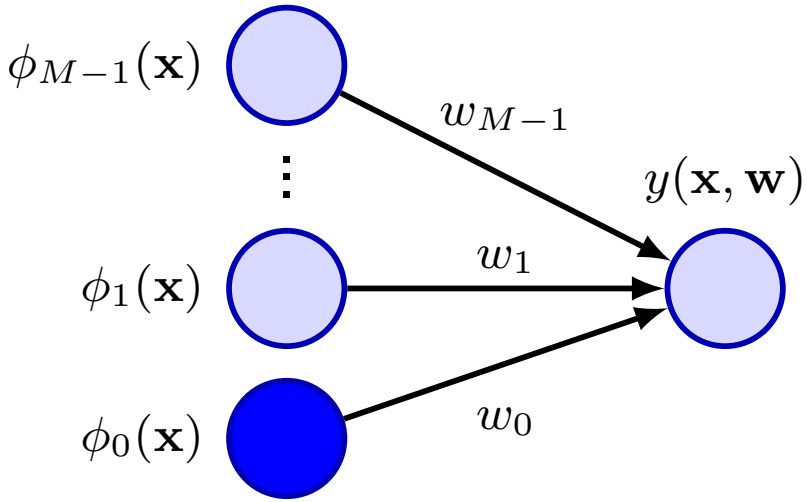
$$y(x; w) = w_0 + \sum_{m=1}^{M-1} w_m \phi_m(x)$$

where $\phi_m(x)$ are known as basis functions. The parameter $w_0$ allows for any fixed offset in the data and is sometimes called a bias parameter. If we define $\phi_0(x) = 1$ then $y(x; w)$ becomes:

$$y(x; w) = \sum_{m=0}^{M-1} w_m \phi_m(x) = w^T \phi(x)$$

# Basis function

Figure: The linear regression model as a single-layer network



# Basis function

Here are some possible choices of basis functions:

- Polynomial: $\phi_m(x) = x^m$.
- Gaussian: $\phi_m(x) = \exp(-\frac{(x-\mu_m)^2}{2s^2})$.
- Sigmoidal: $\phi_m(x) = \frac{1}{1+\exp(-\frac{x-\mu_m}{s})}$.

# Maximum likelihood

Consider a data set of inputs $\{x^1, \ldots, x^N\}$ with corresponding target values $t_1, \ldots, t_N$. Assume that given the value of $x^n$, the corresponding value of $t_n$ has a Gaussian distribution. The likelihood function takes the form:

$$p(t_1, \ldots, t_N | x^1, \ldots, x^N; w, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(t_n; w^T \phi(x^n), \sigma^2 I)$$

The negative log of the likelihood function is given by:

$$\begin{aligned} L &= -\log p(t_1, \ldots, t_N | x^1, \ldots, x^N; w, \sigma^2) \\ &= \frac{N}{2} \log(2\pi) + \frac{N}{2} \log \sigma^2 + \frac{1}{2\sigma^2} \sum_{n=1}^{N} (t_n - w^T \phi(x^n))^2 \end{aligned}$$

# Maximum likelihood

Let's maximize the likelihood function (for simplicity, we will denote $\phi(x^n)$ by $\phi_n$):

$$\frac{\partial L}{\partial w} = \frac{1}{\sigma^2}(w^T \sum_{n=1}^{N} \phi_n \phi_n^T - \sum_{n=1}^{N} t_n \phi_n^T) = \frac{1}{\sigma^2}(w^T \Phi^T \Phi - t^T \Phi)$$

where:

$$\Phi = \begin{pmatrix} \phi_1 & \phi_2 & \dots & \phi_N \end{pmatrix}^T$$

We see that:

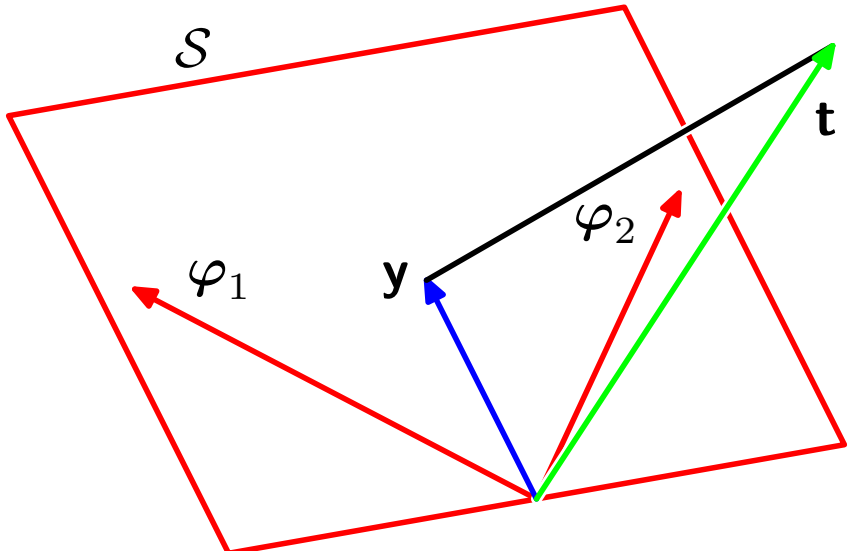
$$w_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t$$

The quantity $(\Phi^T \Phi)^{-1} \Phi^T$ is known as the Moore-Penrose pseudo-inverse of the matrix $\Phi$. It's easy to calculate $\sigma^2_{ML}$ as well:

$$\sigma^2_{ML} = \frac{1}{N} \sum_{n=1}^{N} (t_n - w_{ML}^T \phi_n)^2$$

# Geometry of least squares

Figure: Geometrical interpretation of the least squares solution



# Geometry of least squares

Let $\Phi_m$ be the $m$th column of the matrix $\Phi$, and let $y_{ML} \in \mathbb{R}^N$ be the best approximation to $t$ we obtained by maximizing the likelihood function:

$$y_{ML} = \begin{pmatrix} w_{ML}^T \phi_1 \\ w_{ML}^T \phi_2 \\ \vdots \\ w_{ML}^T \phi_N \end{pmatrix} = \Phi w_{ML} = \sum_{m=0}^{M-1} (w_{ML})_m \Phi_m$$

Here we clearly see that $y_{ML} \in \text{span}(\Phi_0, \ldots, \Phi_{M-1})$. In addition, we have:

$$\Phi^T y_{ML} = \Phi^T \Phi w_{ML} = (\Phi^T \Phi)(\Phi^T \Phi)^{-1} \Phi^T t = \Phi^T t$$

$$(t - y_{ML})^T \Phi = 0 \qquad (t - y_{ML})^T \Phi_m = 0$$

That is, $t - y_{ML}$ is orthogonal to each $\Phi_m$, or put another way, $y_{ML}$ is the orthogonal projection of $t$.

# Sequential learning

The maximum likelihood estimator for $w$ involves processing the entire training set in one go. Sometimes we want the data points to be considered one at a time and the model parameters updated after each such presentation. The technique of stochastic (sequential) gradient descent:

- The error function comprises a sum over data points: $E = \sum_n E_n$.
- After presentation of data point $n$, updates the parameter $w$ using: $w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E_n$.
- $\tau$ denotes the iteration number, and $\eta$ is a training rate parameter.

# Sequential learning

For the sum-of-squares error function:

$$E_n = \frac{1}{2}(t_n - w^T \phi_n)^2$$
$$\nabla E_n = -(t_n - w^T \phi_n)\phi_n$$
$$w^{(\tau+1)} = w^{(\tau)} + \eta(t_n - (w^{(\tau)})^T \phi_n)\phi_n$$

# Regularized least squares

Adding a regularization term to an error function to control over-fitting:

$$E_D(w) + \lambda E_W(w)$$

For example, if we use the sum-of-squares error function, the total error function becomes:

$$\frac{1}{2}\sum_{n=1}^{N}(t_n - w^T\phi_n)^2 + \frac{\lambda}{2}w^T w$$

Minimizing this total error function, we obtain:

$$w_{ML} = (\lambda I + \Phi^T\Phi)^{-1}\Phi^T t$$

# Multiple outputs

We have considered situations with a single target variable. In some applications, we may wish to predict $K > 1$ target variables. Let's first get the dimensions right:

- There are $N$ input data: $x^1, \ldots, x^N$, where $x^n \in \mathbb{R}^D$.
- There are $N$ target data: $t^1, \ldots, t^N$, where $t^n \in \mathbb{R}^K$.
  - Let $T = \begin{pmatrix} t^1 & t^2 & \ldots & t^N \end{pmatrix}^T \in \mathbb{R}^{N \times K}$
- There is a basis $\phi$: $\mathbb{R}^D \to \mathbb{R}^M$, $x \to \phi(x)$. For simplicity, we denote $\phi(x^n)$ by $\phi_n$.
  - Let $\Phi = \begin{pmatrix} \phi_1 & \phi_2 & \ldots & \phi_N \end{pmatrix}^T \in \mathbb{R}^{N \times M}$
- There is a matrix of parameters: $W \in \mathbb{R}^{M \times K}$.

# Multiple outputs

Now, let's maximize the likelihood for $y(x; W) = W^T \phi(x)$:

$$
\begin{aligned}
L &= -\log p(t^1, \ldots, t^N | x^1, \ldots, x^N; W, \sigma^2) \\
&= -\log \prod_{n=1}^{N} \mathcal{N}(t^n; W^T \phi_n, \sigma^2 I) \\
&= \frac{NK}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} \sum_{n=1}^{N} ||t^n - W^T \phi_n||^2 \\
\frac{\partial L}{\partial W}(W)H &= \frac{1}{\sigma^2} \sum_{n=1}^{N} (\mathrm{tr}(W^T \phi_n \phi_n^T H) - \mathrm{tr}(t^n \phi_n^T H)) \\
&= \frac{1}{\sigma^2} \mathrm{tr}((W^T \Phi^T \Phi - T^T \Phi) H) \\
W_{ML} &= (\Phi^T \Phi)^{-1} \Phi^T T
\end{aligned}
$$

# Decision theory

- We have learned from data using maximum likelihood, and the result is a predictive distribution.
- However, for many practical applications we need to predict a specific value.
- In the inference stage, we use the training data to determine a predictive distribution.
- In the decision stage, we use this predictive distribution to determine a specific value.

# Decision theory

**Problem**

Given a predictive distribution $p(t|x)$, determine a specific value $f(x)$, which will be dependent on the input $x$, that is optimal according to some criterion.

# Decision theory

Because we do not know the true value of $t$, we cannot minimize the loss $L = (f(x) - t)^2$ itself, instead let's minimize the expected loss:

$$E(L) = \iint (f(x) - t)^2 p(x, t) \mathrm{d}x \mathrm{d}t$$

We want to find $f(x)$ that minimizes $E(L)$:

$$\frac{\delta E(L)}{\delta f(x)} = 2 \int (f(x) - t) p(x, t) \mathrm{d}t = 0$$

$$f(x) = \frac{\int t p(x, t) \mathrm{d}t}{\int p(x, t) \mathrm{d}t} = \frac{\int t p(x, t) \mathrm{d}t}{p(x)} = \int t p(t|x) \mathrm{d}t = E(t|x)$$

which is the conditional average of $t$ conditioned on $x$ and is known as the regression function.

# Decision theory

Now that we know that the optimal solution is the conditional expectation, we can expand the square term as follows:

$$
\begin{aligned}
(f(x) - t)^2 &= ((f(x) - E(t|x)) + (E(t|x) - t))^2 \\
&= (f(x) - E(t|x))^2 + 2(f(x) - E(t|x))(E(t|x) - t) + (E(t|x) - t)^2
\end{aligned}
$$

# Decision theory

Let's examine the expectation for each term:

$$\iint (f(x) - E(t|x))^2 p(x,t) \mathrm{d}x\mathrm{d}t = \int (f(x) - E(t|x))^2 (\int p(x,t)\mathrm{d}t)\mathrm{d}x$$

$$= \int (f(x) - E(t|x))^2 p(x)\mathrm{d}x$$

$$\iint (f(x) - E(t|x))(E(t|x) - t)p(x,t)\mathrm{d}x\mathrm{d}t$$

$$= \int (f(x) - E(t|x))p(x)(\int (E(t|x) - t)p(t|x)\mathrm{d}t)\mathrm{d}x = 0$$

$$\iint (E(t|x) - t)^2 p(x,t)\mathrm{d}x\mathrm{d}t = \int p(x)(\int (t - E(t|x))^2 p(t|x)\mathrm{d}t)\mathrm{d}x$$

$$= \int \mathrm{var}(t|x)p(x)\mathrm{d}x$$

# Decision theory

Let's interpret what we have derived here:

$$E(L) = \int (f(x) - E(t|x))^2 p(x) \mathrm{d}x + \int \mathrm{var}(t|x) p(x) \mathrm{d}x$$

- The first term shows that the optimal least-squares predictor is given by the conditional expectation.
- The second term is the variance of $t$ averaged over $x$, and represents the intrinsic variability of the target data.

# The bias-variance trade-off

For a regression problem:

- Given a data set $\mathcal{D}$, we can run our learning algorithm and obtain a prediction function $f(x; \mathcal{D})$.
  - Note that this prediction function contains both the inference and decision stages.
- We could view the uncertainty of our model in two ways:
  - Bayesian: The uncertainty is expressed through a poterior distribution over the parameters.
  - Frequentist: The uncertainty comes from the data set $\mathcal{D}$. If we are given an ensemble of data sets, we can average out the uncertainty.

# The bias-variance trade-off

From previous analysis we know that the expected squared loss can be written in the form:

$$E(L) = \int (f(x) - E(t|x))^2 p(x) \mathrm{d}x + \int \mathrm{var}(t|x) p(x) \mathrm{d}x$$

To better understand the first term, let's consider its expectation over an ensemble of data sets. If we denote the average prediction function over the ensemble of data sets as $\bar{f}(x) = E_{\mathcal{D}}(f(x;\mathcal{D}))$, then:

$$\begin{aligned}
&E_{\mathcal{D}}((f(x;\mathcal{D}) - E(t|x))^2) \\
&= E_{\mathcal{D}}(((f(x;\mathcal{D}) - \bar{f}(x)) + (\bar{f}(x) - E(t|x)))^2) \\
&= E_{\mathcal{D}}((f(x;\mathcal{D}) - \bar{f}(x))^2) + E_{\mathcal{D}}((\bar{f}(x) - E(t|x))^2) \\
&= \mathrm{var}_{\mathcal{D}}(f(x;\mathcal{D})) + (\bar{f}(x) - E(t|x))^2
\end{aligned}$$

# The bias-variance trade-off

Let's examine the terms:

- $(\bar{f}(x) - E(t|x))^2$: The squared bias, represents the extent to which the average prediction over all data sets differs from the desired regression function.
- $\mathrm{var}_{\mathcal{D}}(f(x;\mathcal{D}))$: The variance, measures the extent to which the solutions for individual data sets vary around their average.

# The bias-variance trade-off

We obtain the following decomposition of the expected squared loss:

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

where:

$$\text{bias}^2 = \int (\bar{f}(x) - E(t|x))^2 p(x) \mathrm{d}x$$

$$\text{variance} = \int \text{var}_{\mathcal{D}}(f(x;\mathcal{D})) p(x) \mathrm{d}x$$

$$\text{noise} = \int \text{var}(t|x) p(x) \mathrm{d}x$$

# The bias-variance trade-off

To minimize the expected loss, there will be a trade-off between bias and variance:

- Very flexible models have low bias and high variance.
- Relatively rigid models have high bias and low variance.