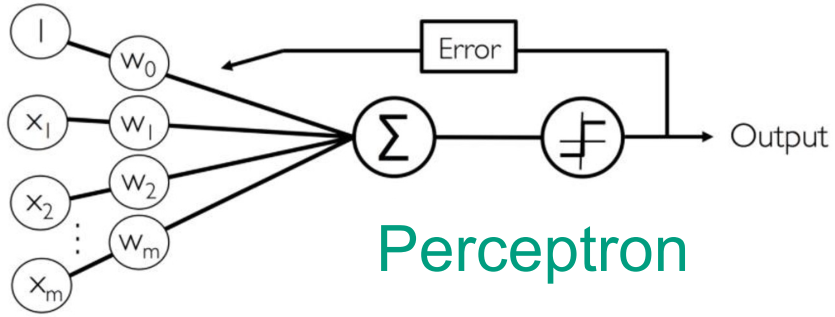
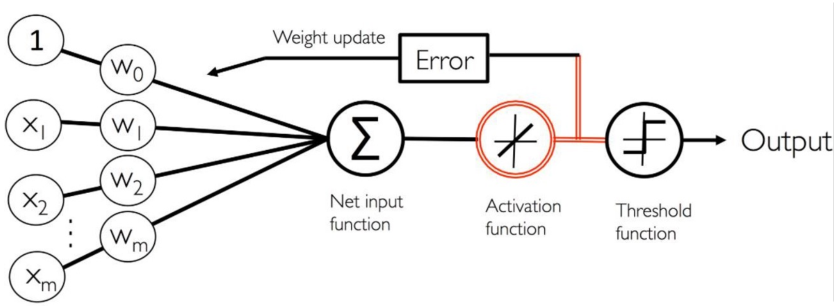
Recap

Quadratic loss/cost function

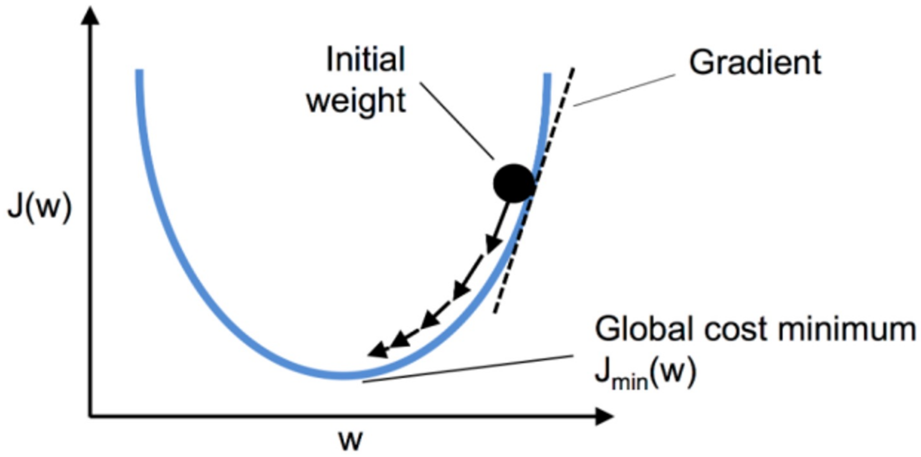
$$J(\boldsymbol{w}) = \frac{1}{2}\sum_i \left(y^{(i)} - \phi\left(z^{(i)} \right) \right)^2$$



Perceptron



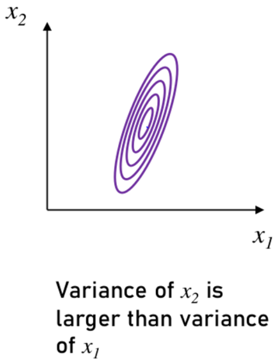
Adaptive linear neuron (Adaline)

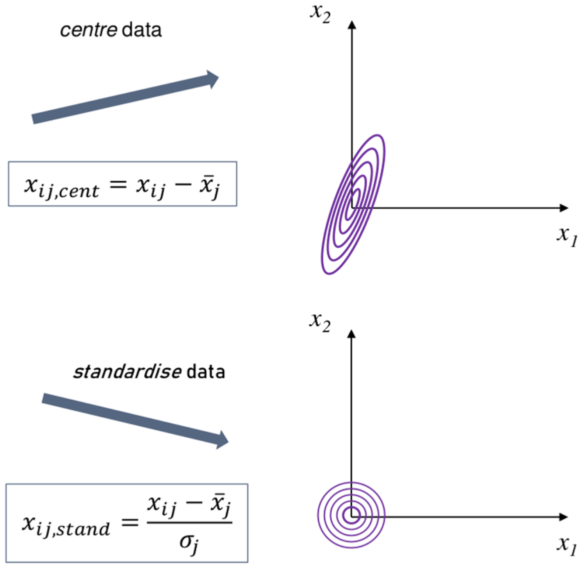


$$\Delta \boldsymbol{w} = -\eta \nabla J(\boldsymbol{w})$$

Gradient descent

Feature scaling





Linear regression & logistic regression

Basic notation (repetition)

Vectors and Matrices

We often represent raw (numeric) data as vectors and matrices

Example: Iris data can be represented as a 150 by 4 matrix: $\boldsymbol{X} \in \mathbb{R}^{150x4}$

- Superscript means i-th training sample
- Subscript means j-th feature (dimension)
- Lowercase boldface → vectors ($\boldsymbol{x} \in \mathbb{R}^{nx1}$)
- Uppercase boldface → matrices ($\boldsymbol{X} \in \mathbb{R}^{mxn}$)
- Single element in a vector $x^{(i)}$
- Single element in a matrix $x_j^{(i)}$

$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

Vectors and Matrices

Row vectors (e.g. one row in $\boldsymbol{X}$, “flower” sample in Iris data set)

$$\boldsymbol{x}^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} \end{bmatrix}$$

$$\boldsymbol{x}^{i} \in \mathbb{R}^{1x4}$$

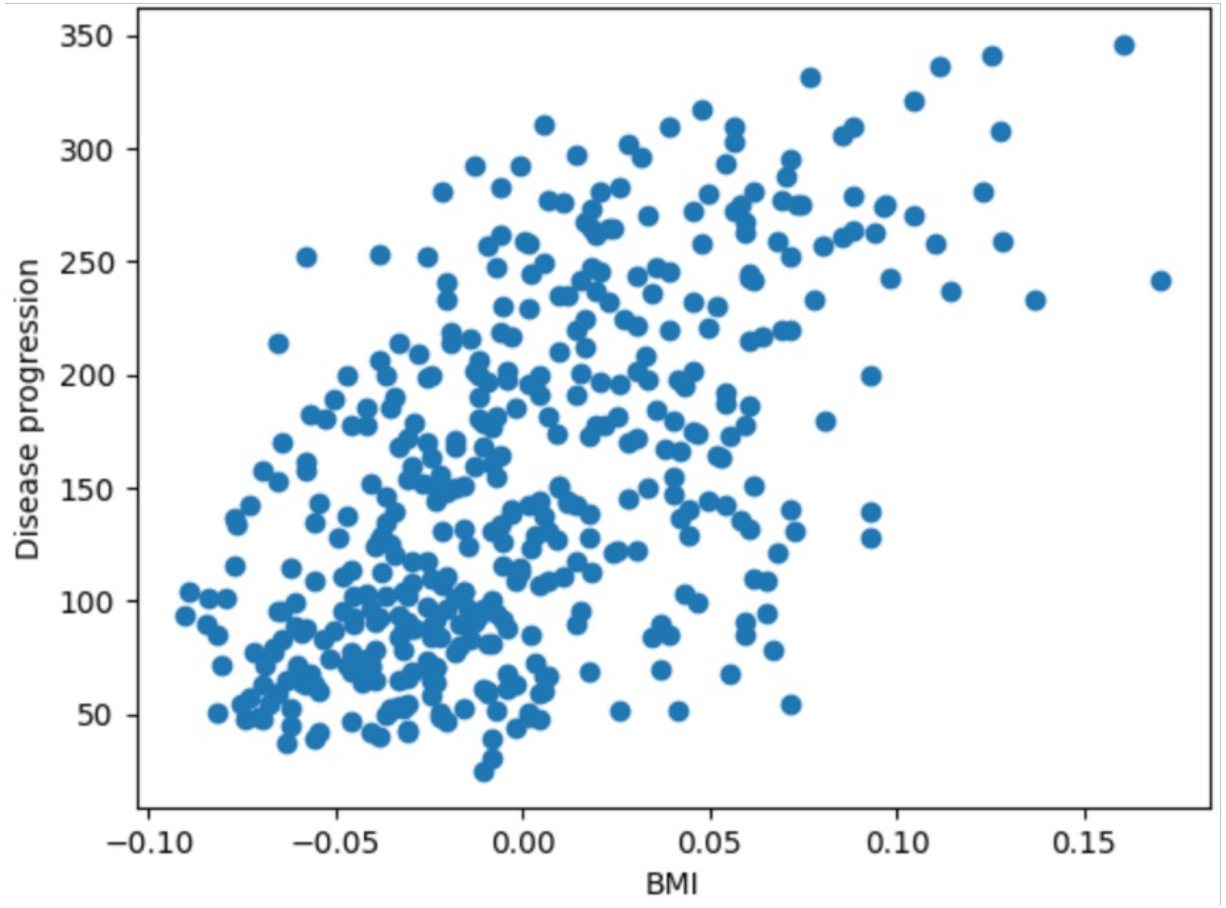
Column vectors (e.g. one column in $\boldsymbol{X}$, one feature)

$$\boldsymbol{x}_j = \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \\ \vdots \\ x_j^{(150)} \end{bmatrix}$$

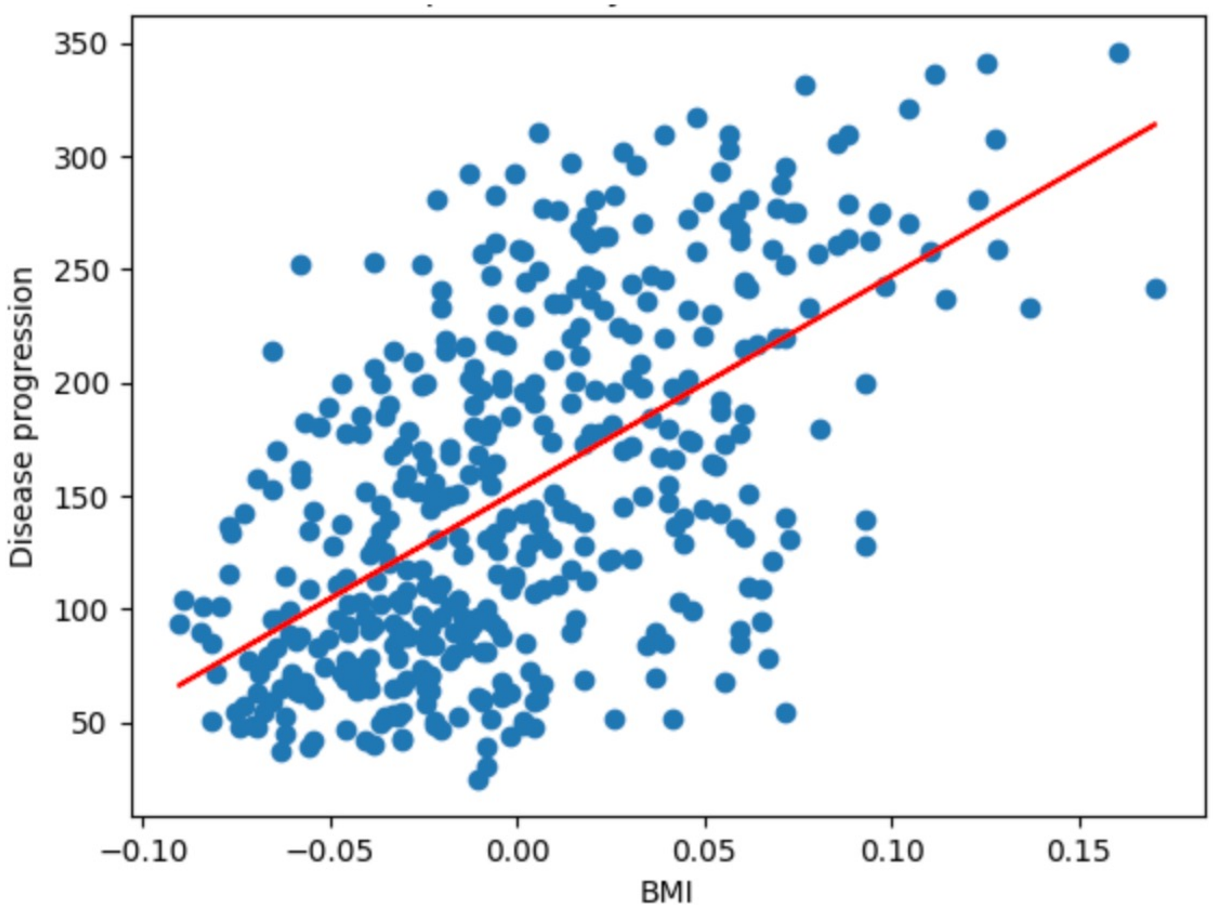
$$\boldsymbol{x}_j \in \mathbb{R}^{150x1}$$

Simple linear regression
(and least squares)

Linear regression



Linear regression



Model (parametric)

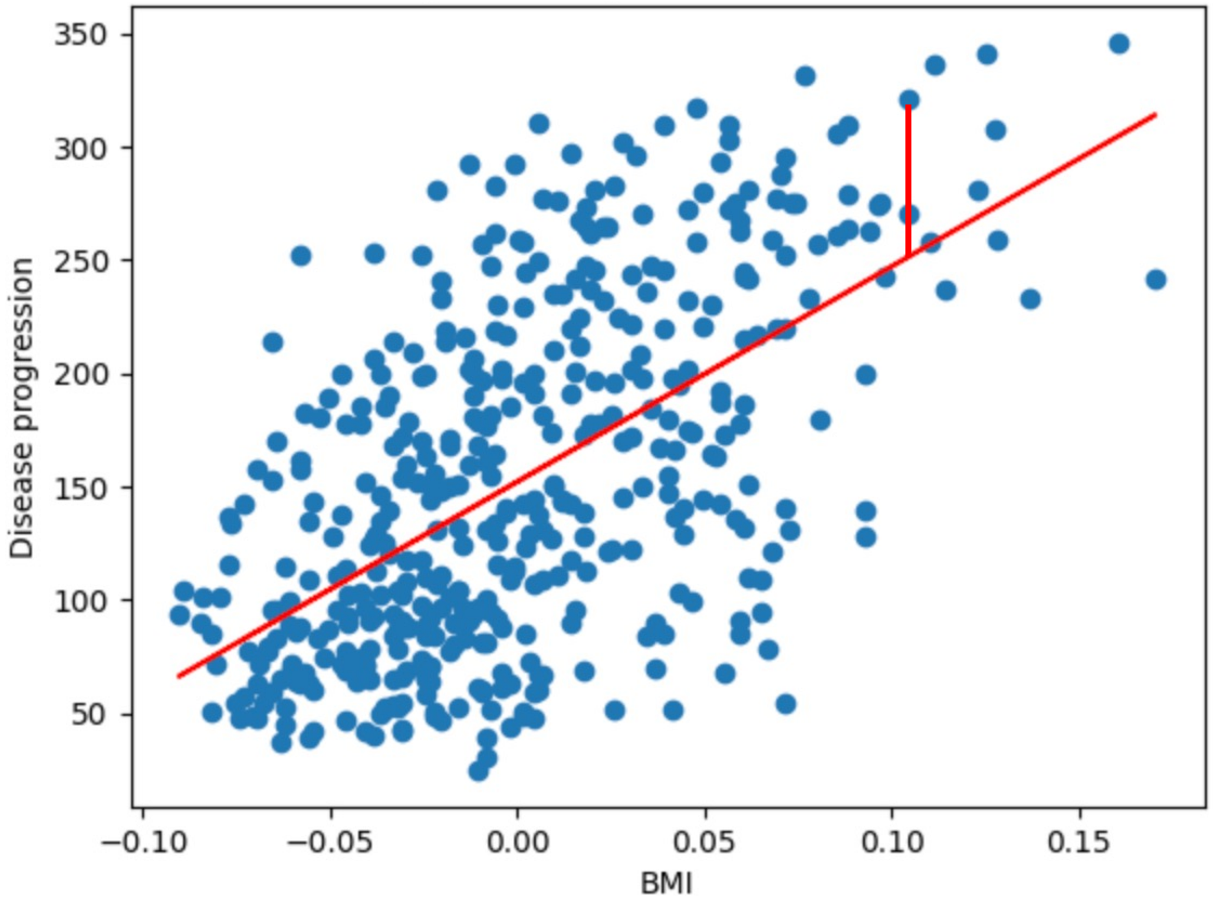
$$f(x) = \theta_0 + \theta_1 x$$

Error

$$\varepsilon^{(i)} = y^{(i)} - (\theta_0 + \theta_1 x^{(i)})$$

Linear regression

$$\varepsilon^{(i)} = y^{(i)} - (\theta_0 + \theta_1 x^{(i)})$$



Generalization to m features

$$\varepsilon^{(i)} = y^{(i)} - \mathbf{x}^{(i)}\boldsymbol{\theta},$$

$$\mathbf{x}^{(i)} = [1, x_1] \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

$$\mathbf{x}^{(i)}\boldsymbol{\theta} := \sum_{j=0}^{m} x_j^{(i)} \theta_j$$

Alternative equivalent parameter representation

$$\mathbf{x}^{(i)}\boldsymbol{\theta} = b + \mathbf{x}^{(i)}\mathbf{w},$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_m \end{bmatrix} := \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{bmatrix}, \quad b = \theta_0$$

"weights" "bias"

"normal" "offset" (hyperplanes)

Linear regression

Error in matrix notation

$$\boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{X}\boldsymbol{\theta}.$$

Linear regression

Goal: Minimize the error

We have one error for each sample, how to measure a global error?

Quadratic / Squared-error loss function

$$L = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = ||\boldsymbol{\epsilon}||_2^2 = \sum_{i=1}^{n} {\varepsilon^{(i)}}^2$$

Least squares solution

Goal: Minimize the sum of squared errors

$$L = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = ||\boldsymbol{\epsilon}||_2^2 = \sum_{i=1}^{n} \varepsilon^{(i)^2}$$

Insert error

$$L(\boldsymbol{\theta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})$$

$$= \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{X}\boldsymbol{\theta} - \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{y} + \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{X}\boldsymbol{\theta}$$

Least squares solution

Goal: Minimize the sum of squared errors

$$L(\boldsymbol{\theta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{y} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = \frac{\partial\left(\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{y} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}} = \mathbf{0}$$

$$= -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} = \mathbf{0}$$

Least squares solution

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} = \mathbf{0}$$

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} = \mathbf{X}^T\mathbf{y}$$ (Normal equations)

$$\boxed{\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$ Least squares solution

Least squares solution

Learning / Training / Fitting

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

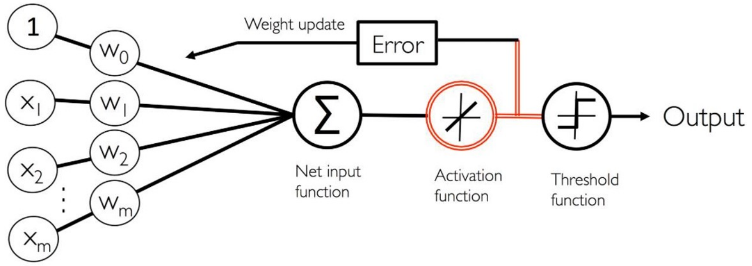
Prediction

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta}$$

Remark: remember that we added a "1" feature to ***X***

`lin_regression.ipynb`

Linear regression and Adaline?



Adaptive Linear Neuron (Adaline)

→ Adaline is linear regression with a threshold

Same loss function:

$$L = \sum_{i=1}^{n} \varepsilon^{(i)^2} = \sum_{i=1}^{n} \left(y^{(i)} - \sum_{k=0}^{m} x_k^{(i)} \theta_k \right)^2 = \sum_{i=1}^{n} \left(y^{(i)} - \mathbf{x}^{(i)} \boldsymbol{\theta} \right)^2$$

Why is Adaline not always so good in practice?

– sensitive to outliers

Linear regression again

Probabilistic interpretation

Probabilistic interpretation

Linear model

$$y^{(i)} = \mathbf{x}^{(i)}\boldsymbol{\theta} + \varepsilon^{(i)}$$

Error model (assumption)

$$\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$$

$$p(\varepsilon^{(i)}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{{\varepsilon^{(i)}}^2}{2\sigma^2}\right)$$

$\varepsilon^{(i)}$ are i.i.d.

independent and identically distributed

Probabilistic interpretation

Probability of a single outcome, given the sample, and parametrized by $\boldsymbol{\theta}$

$$P(y^{(i)} \mid x^{(i)}; \boldsymbol{\theta}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \mathbf{x}^{(i)}\boldsymbol{\theta}\right)^2}{2\sigma^2}\right)$$

Likelihood of $\boldsymbol{\theta}$ (defined in terms of probability of the data)

$$\mathcal{L}(\boldsymbol{\theta}) = P(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\theta})$$

Probabilistic interpretation

Goal: We want to maximize the likelihood of our parameters

$$\mathcal{L}(\boldsymbol{\theta}) = P(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\theta})$$

$$= \prod_{i=1}^{n} P(y^{(i)} \mid x^{($$

$$\ell(\boldsymbol{\theta}) := \log \mathcal{L}(\boldsymbol{\theta}) = \log \prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}; \boldsymbol{\theta})$$

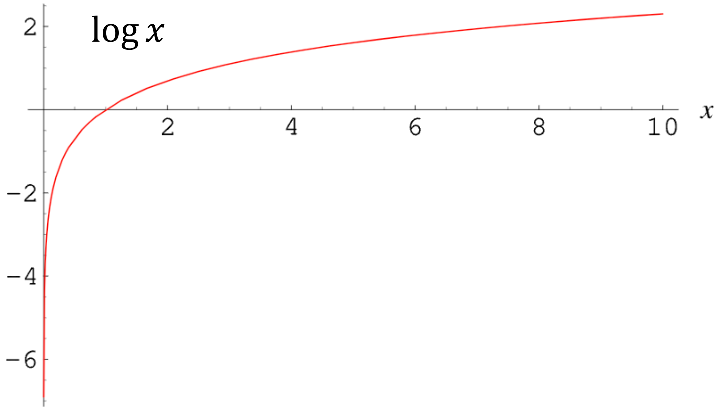
$$= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \mathbf{x}^{(i)}\boldsymbol{\theta}\right)^2}{2\sigma^2}\right)$$

Probabilistic interpretation

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}; \boldsymbol{\theta})$$

In practice it's often beneficial to look at the log-likelihood

Since the natural logarithm (log) is a strictly monotone function, likelihood and log-likelihood attain maximum at the same $\boldsymbol{\theta}$



$$\ell(\boldsymbol{\theta}) := \log \mathcal{L}(\boldsymbol{\theta}) = \log \prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}; \boldsymbol{\theta})$$

$$= \log \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \mathbf{x}^{(i)}\boldsymbol{\theta}\right)^2}{2\sigma^2}\right)$$

$$= n \log \frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y^{(i)} - \mathbf{x}^{(i)}\boldsymbol{\theta}\right)^2$$

Probabilistic interpretation

Maximizing a function is the same as minimizing the negative function

$$\ell(\boldsymbol{\theta}) := \log \mathcal{L}(\boldsymbol{\theta}) \;= n \log \frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y^{(i)} - \mathbf{x}^{(i)}\boldsymbol{\theta}\right)^2$$

New goal: Minimize negative log-likelihood

(leave away scaling factors and constant)

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left(y^{(i)} - \mathbf{x}^{(i)}\boldsymbol{\theta}\right)^2 = \sum_{i=1}^{n} {\varepsilon^{(i)}}^2$$

→ For example, solve with least squares

Probabilistic interpretation

Under the assumption that the errors are Gaussian and i.i.d.,
the **maximum likelihood estimator** for $\boldsymbol{\theta}$ is given by the least squares solution

$$\boxed{\boldsymbol{\theta}_{\text{MLE}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$

What is the variance σ^2?

$$\frac{\partial \ell(\boldsymbol{\theta}, \sigma)}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}\left(y^{(i)} - \mathbf{x}^{(i)T}\boldsymbol{\theta}\right)^2 = 0$$

$$\rightarrow \quad \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \mathbf{x}^{(i)T}\boldsymbol{\theta}\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\varepsilon^{(i)2}$$

variance is the mean squared error

Logistic regression

Logistic regression, a binary classifier

Labels $y^{(i)} \in \{0, 1\}$

Probability of the data

$$p := P(y^{(i)} = 1 \,|\, \mathbf{x}^{(i)}) \rightarrow P(y^{(i)} = 0 \,|\, \mathbf{x}^{(i)}) = 1 - p.$$

What are the odds?

$$\frac{p}{p-1}$$

Useful in practice: log-odds

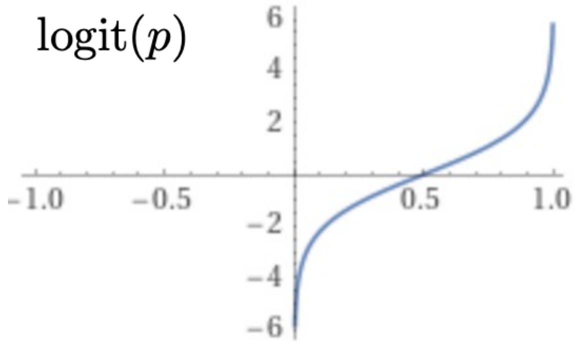
$$\operatorname{logit}(p) := \log \frac{p}{1-p}, \quad \operatorname{logit} : (0, 1) \rightarrow \mathbb{R}$$

Logistic regression, the model

The log-odds are modelled by a linear function

$$\text{logit}(p) = \mathbf{x}^{(i)}\boldsymbol{\theta} = b + \mathbf{x}^{(i)}\mathbf{w}$$

$$\text{logit}(p) := \log \frac{p}{1-p}, \quad \text{logit} : (0,1) \to \mathbb{R}$$



Logistic regression, the model

But we are interested in the probability

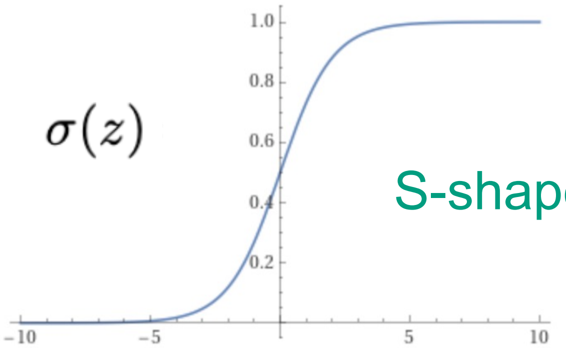
$$p = \sigma(z) = \frac{1}{1+\exp(-z)}$$

Logistic function / sigmoid function

$$\sigma : \mathbb{R} \to (0,1), \quad \sigma(z) = \frac{1}{1+\exp(-z)}$$

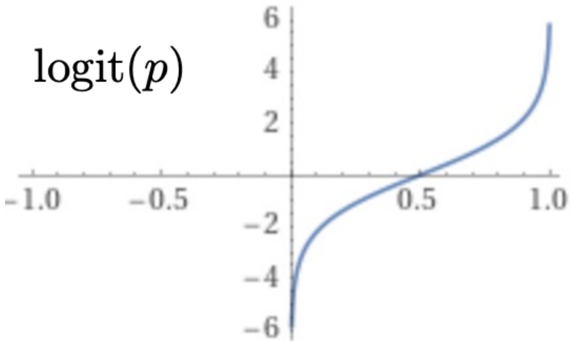
Sigmoid is the inverse of logit

$\sigma(z)$



S-shape

$\mathrm{logit}(p)$



Logistic regression, the plan

Use linear model for the log-odds

Convert to probability using logistic function

Use thresholding to predict class label

$$\hat{y}(z) = \begin{cases} 1 & if\ \sigma(z) \leq 0.5 \\ 0 & otherwise \end{cases} = \begin{cases} 1 & if\ z \leq 0 \\ 0 & otherwise. \end{cases}$$

Logistic regression

Probabilities of the data

$$P(y^{(i)} = 1|\mathbf{x}^{(i)};\boldsymbol{\theta}) = \sigma(z) \qquad P(y^{(i)} = 0 \mid \mathbf{x}^{(i)};\boldsymbol{\theta}) = 1 - \sigma(z)$$

Combined

$$P(y^{(i)} \mid \mathbf{x}^{(i)};\boldsymbol{\theta}) = \sigma(z)^{y^{(i)}}(1 - \sigma(z))^{(1-y^{(i)})} \qquad y^{(i)} \in \{0, 1\}$$

Like for linear regression: maximize likelihood of the parameters

$$\mathcal{L}(\boldsymbol{\theta}) = P(\mathbf{y} \mid \mathbf{X};\boldsymbol{\theta})$$

Logistic regression

Goal: maximize likelihood of the parameters

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= P(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\theta}) \\
&= \prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}; \boldsymbol{\theta}), \qquad \text{(samples are i.i.d.)} \\
&= \prod_{i=1}^{n} \left[\sigma(z)^{y^{(i)}} (1 - \sigma(z))^{(1-y^{(i)})} \right].
\end{aligned}
$$

Logistic regression

Goal: maximize likelihood of the parameters

Use log-likelihood instead

$$\ell(\boldsymbol{\theta}) = \log \mathcal{L}(\boldsymbol{\theta}) = \log P(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\theta})$$

$$= \log \prod_{i=1}^{n} \left[\sigma(z)^{y^{(i)}} (1 - \sigma(z))^{(1-y^{(i)})} \right]$$

$$= \sum_{i=1}^{n} \left[y^{(i)} \log(\sigma(z)) + (1 - y^{(i)}) \log(1 - \sigma(z)) \right]$$

Logistic regression

Goal: maximize likelihood of the parameters

Use log-likelihood instead

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &= \log \mathcal{L}(\boldsymbol{\theta}) = \log P(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\theta}) \\
&= \log \prod_{i=1}^{n} \left[\sigma(z)^{y^{(i)}} (1 - \sigma(z))^{(1-y^{(i)})} \right] \\
&= \sum_{i=1}^{n} \left[y^{(i)} \log(\sigma(z)) + (1 - y^{(i)}) \log(1 - \sigma(z)) \right]
\end{aligned}$$

Logistic regression

Goal: maximize likelihood of the parameters → minimize negative log-likelihood

Use log-likelihood instead

$$L(\boldsymbol{\theta}) = -\ell(\boldsymbol{\theta}) \qquad \ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left[y^{(i)} \log(\sigma(z)) + (1 - y^{(i)}) \log(1 - \sigma(z)) \right]$$

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{0}$$

condition for minimum

Logistic regression

$$\sum_{i=1}^{n} \left[y^{(i)} \log(\sigma(z)) + (1 - y^{(i)}) \log(1 - \sigma(z)) \right]$$

$$\frac{\partial \sigma(z)}{\partial z} =$$

$$\frac{\partial (\log \sigma(z))}{\partial \boldsymbol{\theta}} =$$

$$\frac{\partial (\log(1 - \sigma(z)))}{\partial \boldsymbol{\theta}} =$$

Logistic regression

Goal: maximize likelihood of the parameters → minimize negative log-likelihood

Use log-likelihood instead

$$L(\boldsymbol{\theta}) = -\ell(\boldsymbol{\theta}) \qquad \ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left[y^{(i)} \log(\sigma(z)) + (1 - y^{(i)}) \log(1 - \sigma(z)) \right]$$

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\sum_{i=1}^{n} \left[y^{(i)} (1 - \sigma(z)) \mathbf{x}^{(i)} - (1 - y^{(i)}) \sigma(z) \mathbf{x}^{(i)} \right]$$

$$= -\sum_{i=1}^{n} \left[(y^{(i)} - \sigma(z)) \mathbf{x}^{(i)} \right] = \mathbf{0},$$

Same gradient as for linear regression / Adaline, except for $\sigma(z)$!

Component-wise

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \theta_j} = -\sum_{i=1}^{n} \left(x_j^{(i)} \left[(y^{(i)} - \sigma(\mathbf{x}^{(i)} \boldsymbol{\theta})) \right] \right) = 0$$

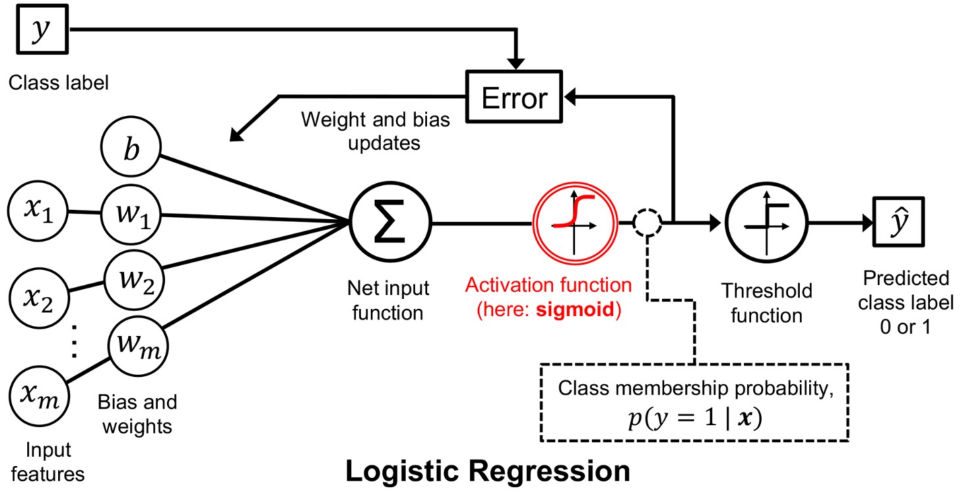
But nonlinear, no explicit solution!

Logistic regression summary

Learn / train / fit

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \theta_j} = -\sum_{i=1}^{n} \left(x_j^{(i)} \left[(y^{(i)} - \sigma(\mathbf{x}^{(i)} \boldsymbol{\theta})) \right] \right) = 0.$$

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta} - \eta \frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$ (Batch) gradient descent



Predict

$$z = \mathbf{X}\boldsymbol{\theta} \qquad \mathbf{y} = \sigma(z) = \frac{1}{1 + \exp(-z)}$$

`log_regression.ipynb`

Logistic regression summary

Logistic regression gives us label **and** probability

Very popular e.g. in health section, but really everywhere

`log_regression.ipynb`