Loss Function

Dr Amit Sethi, IIT Bombay

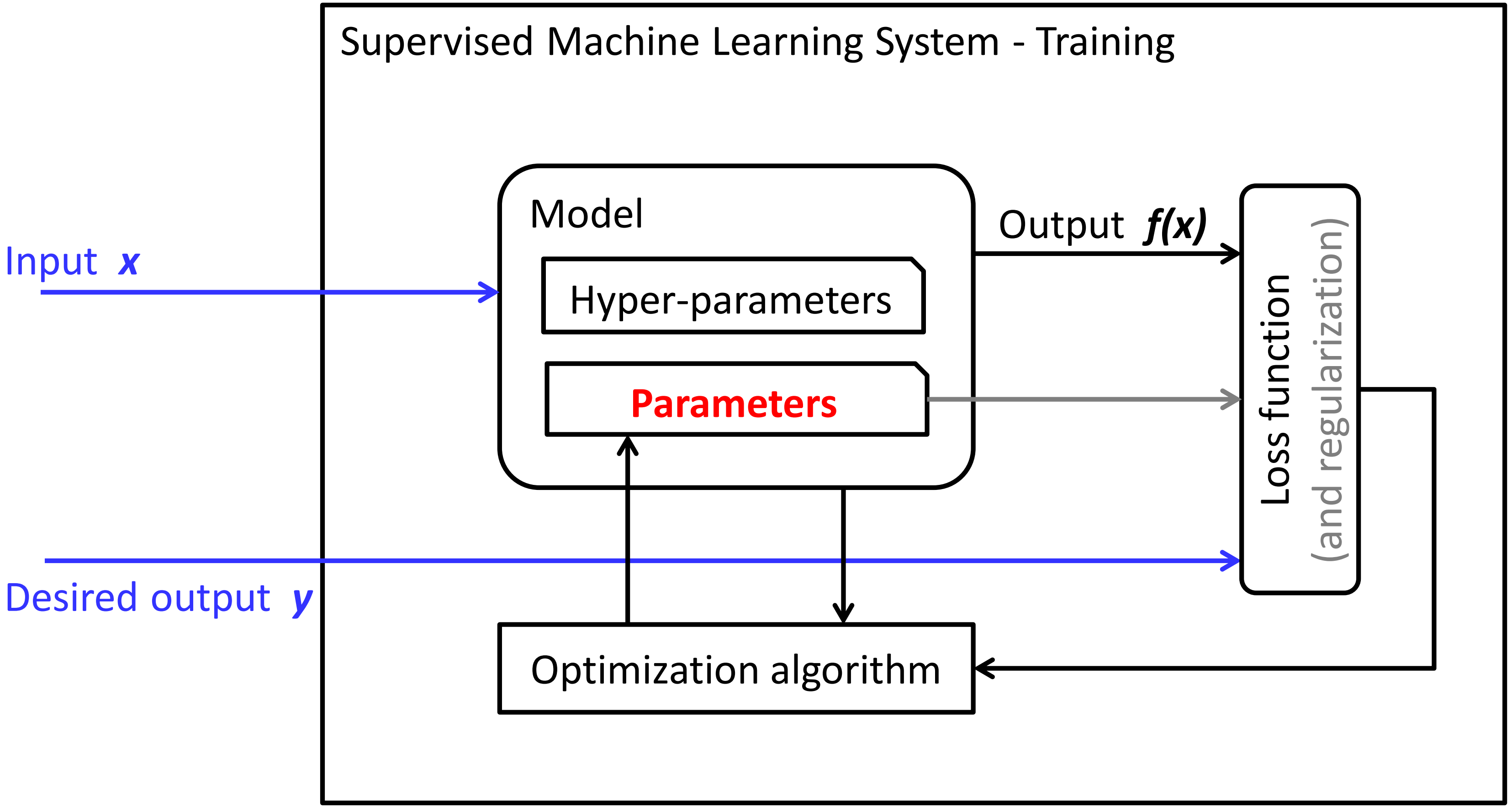
Module objectives

- Understand different components of supervised ML
- Learn to casting basic ML problems in terms of model and loss functions
- Appreciate the basics of regularization

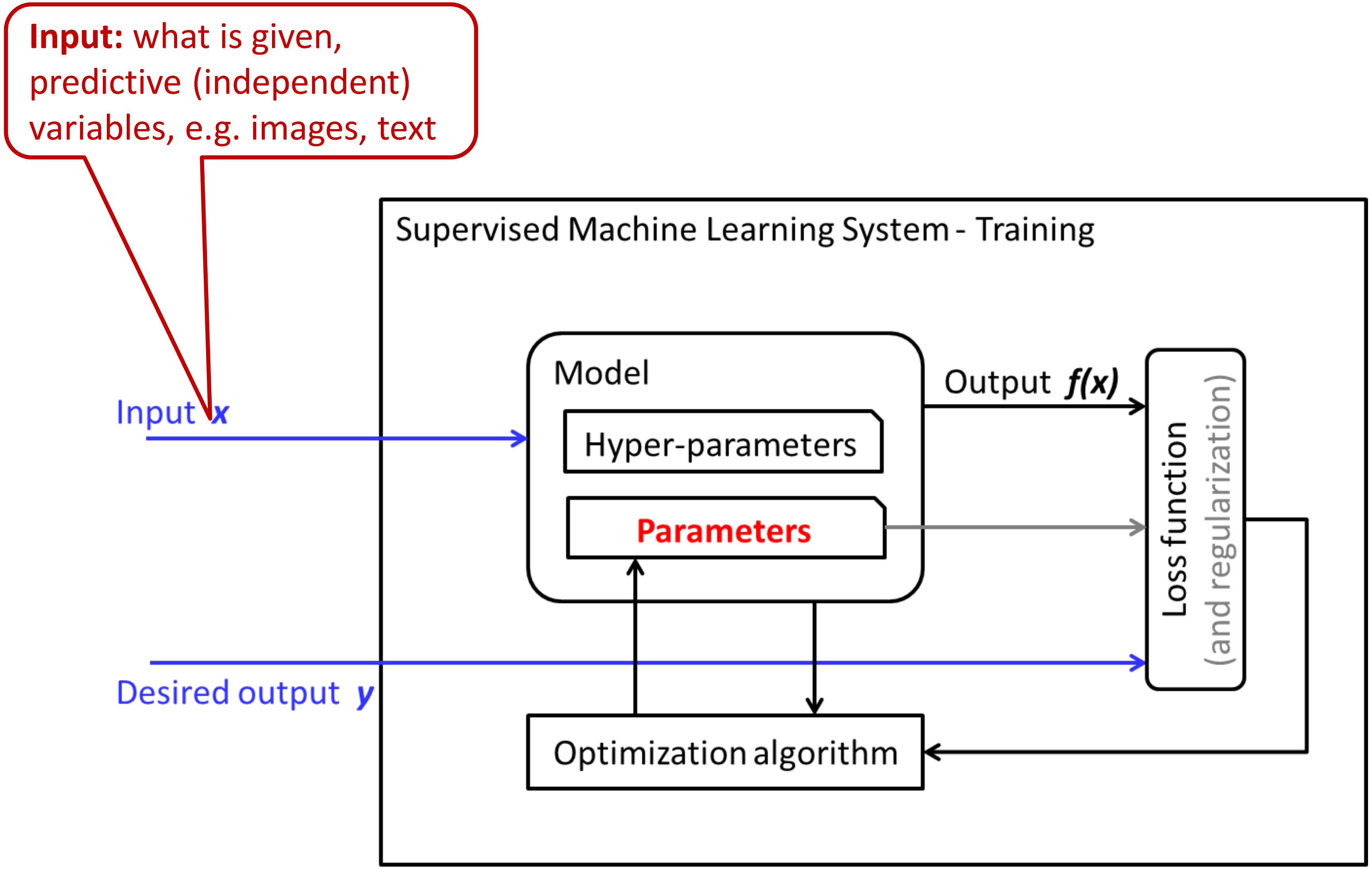
Outline

- Components of supervised machine learning
- Model
- Loss function
- Regularization

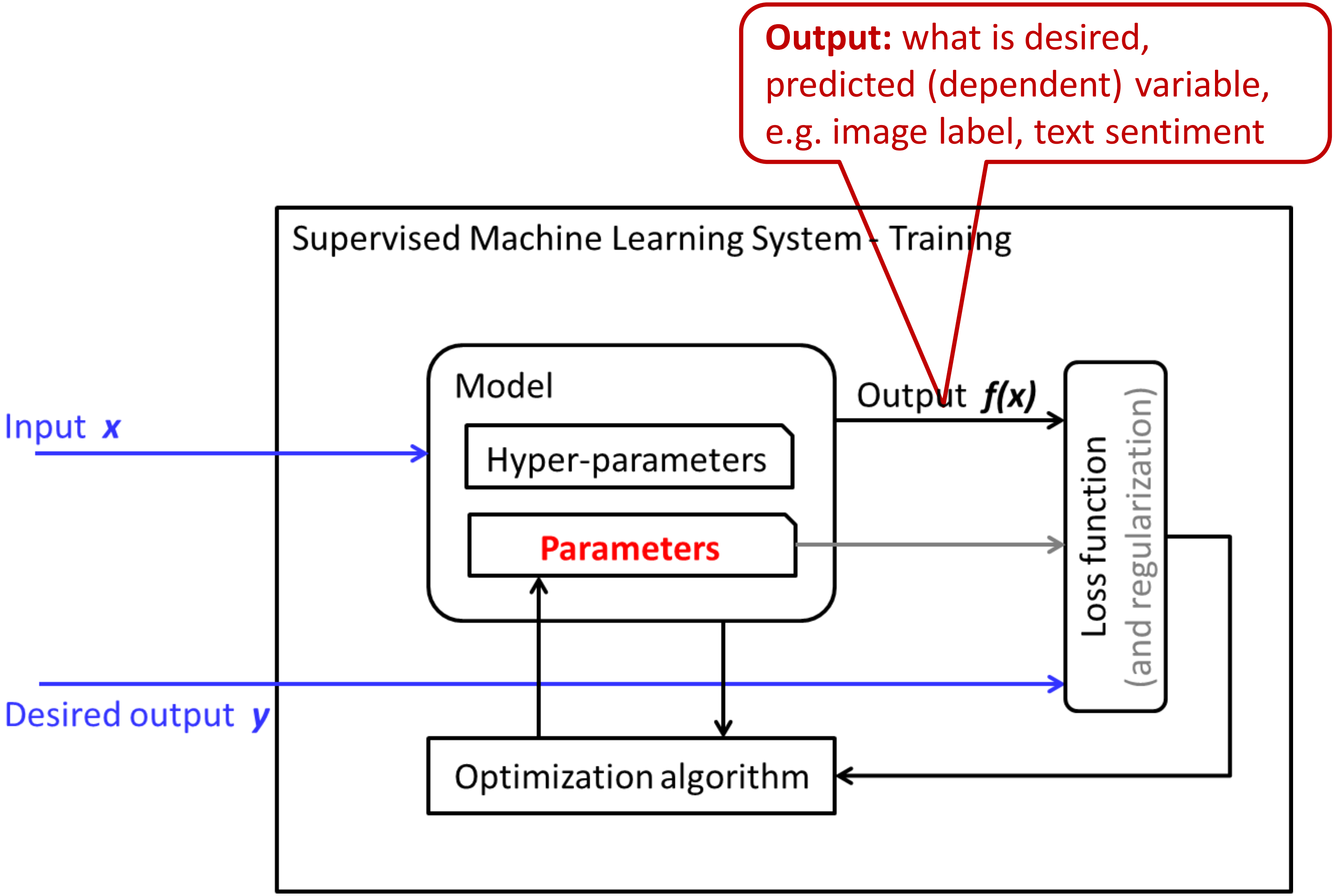
Components of Supervised ML



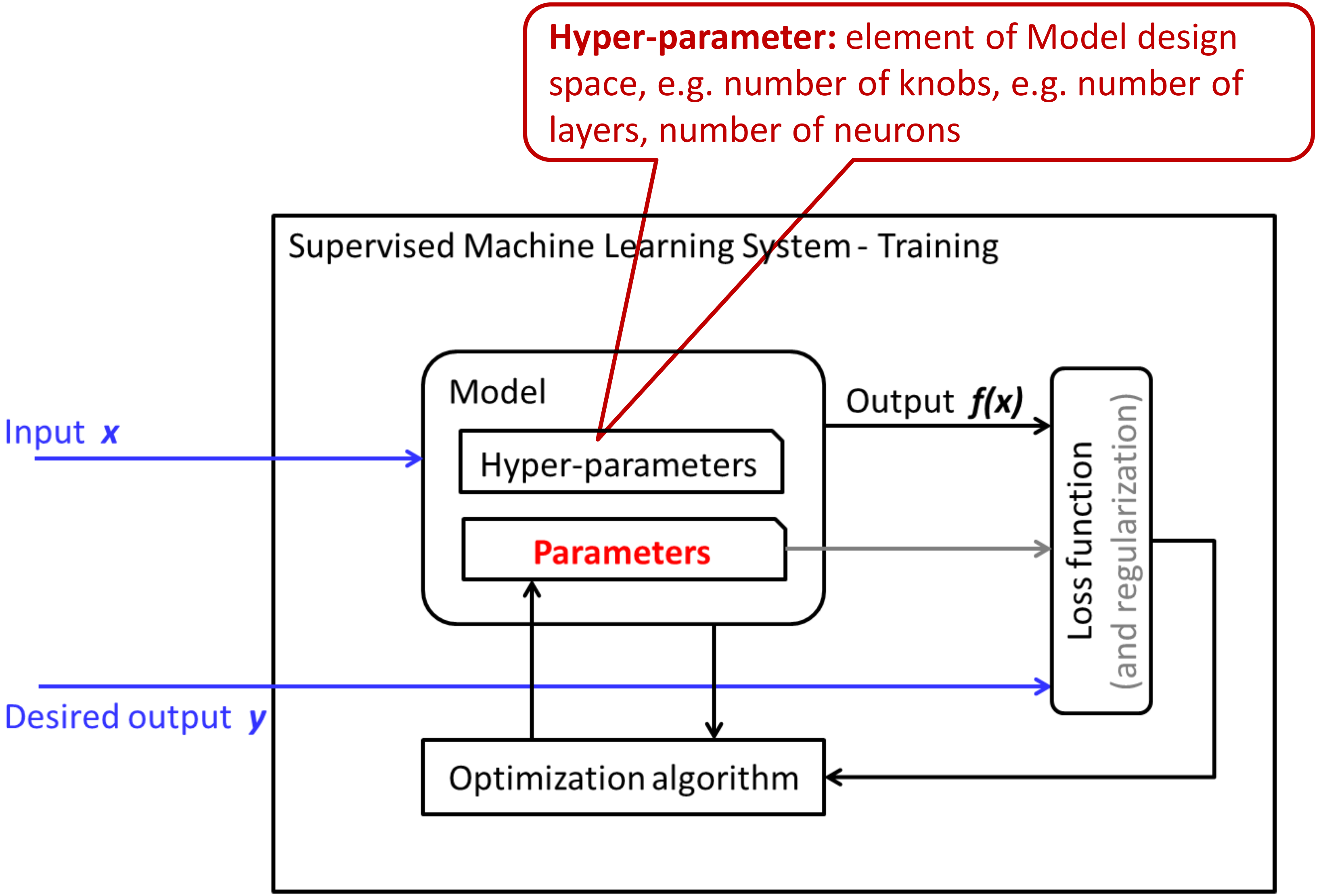
Components of Supervised ML



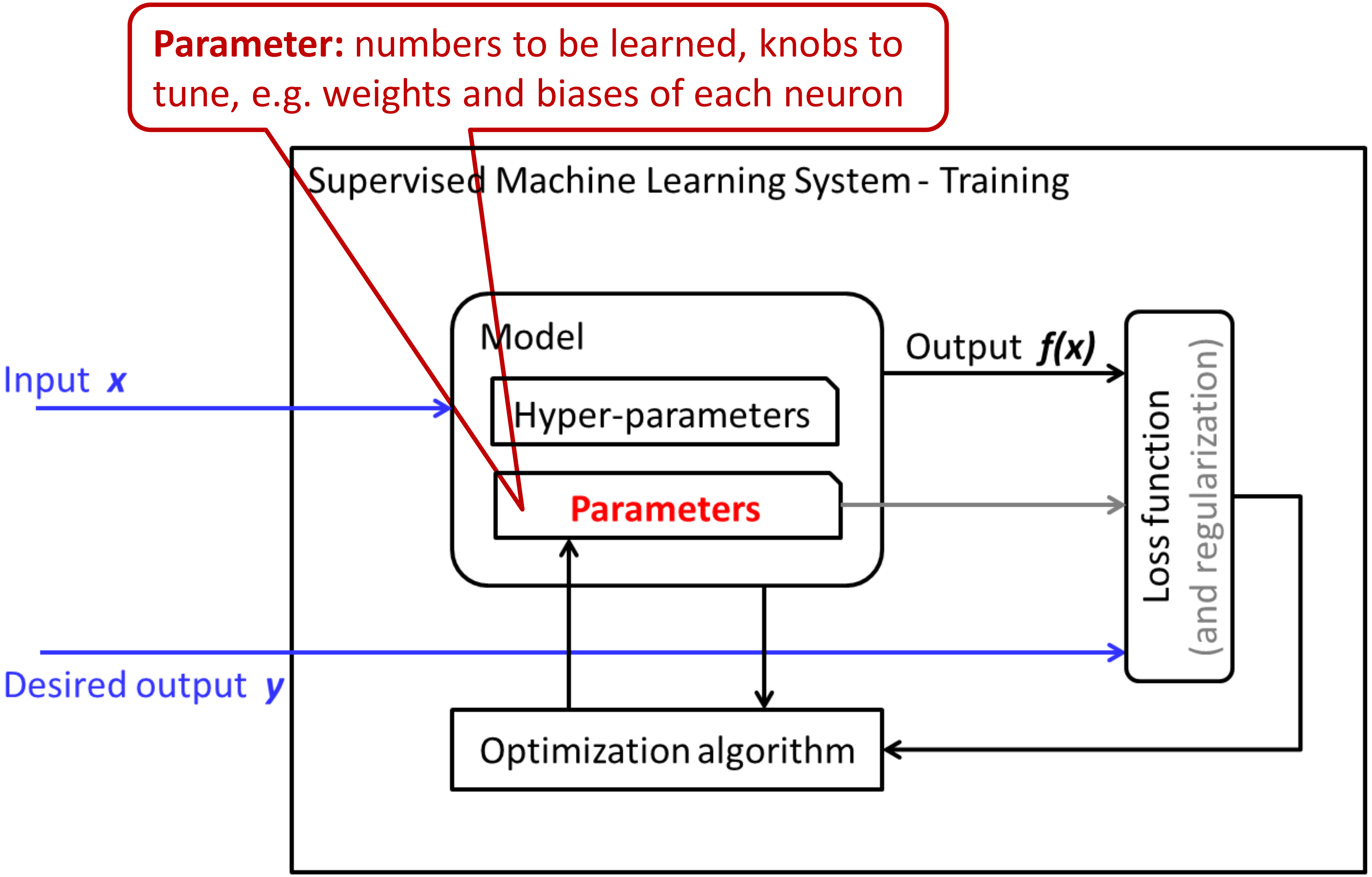
Components of Supervised ML



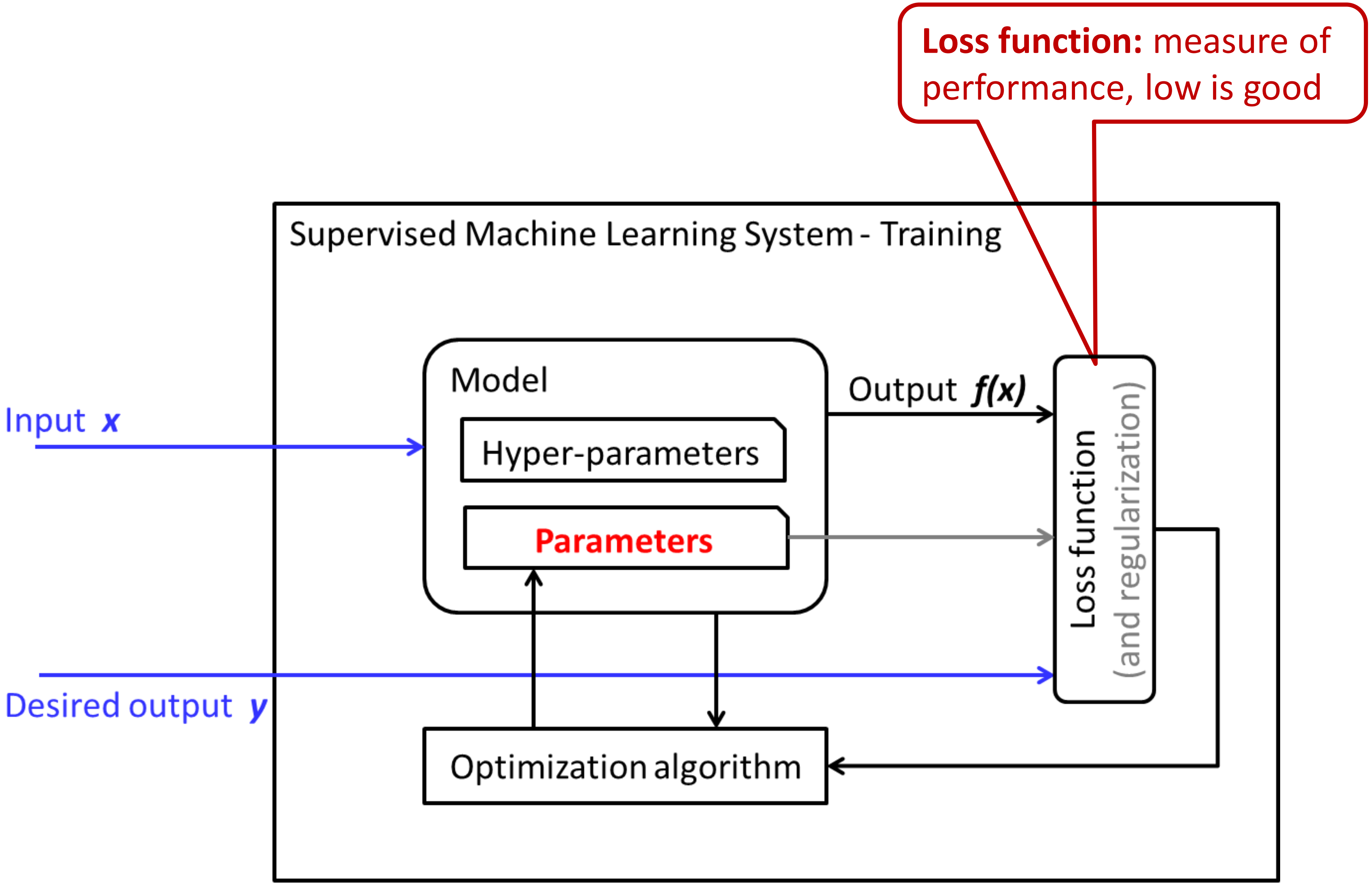
Components of Supervised ML



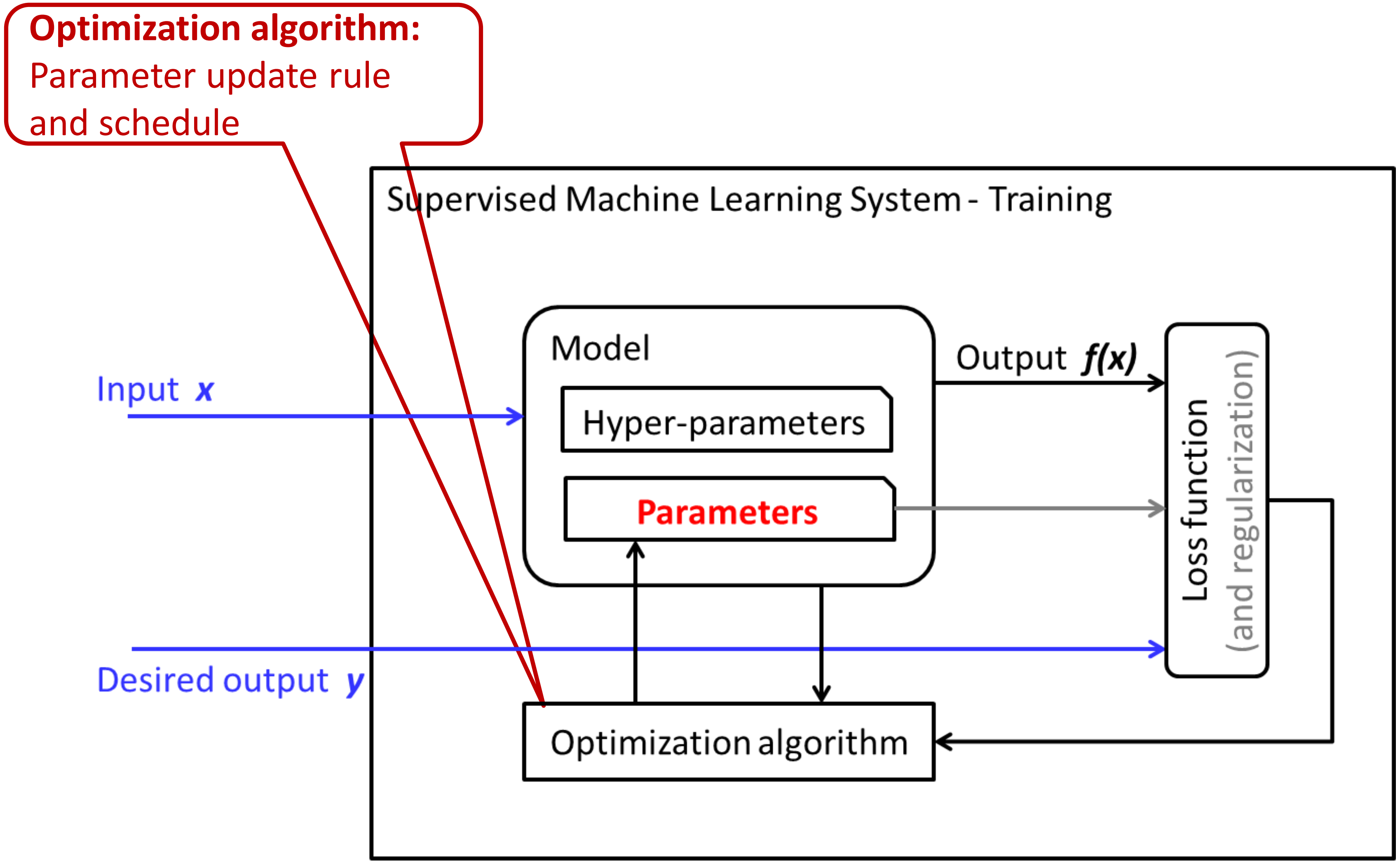
Components of Supervised ML



Components of Supervised ML



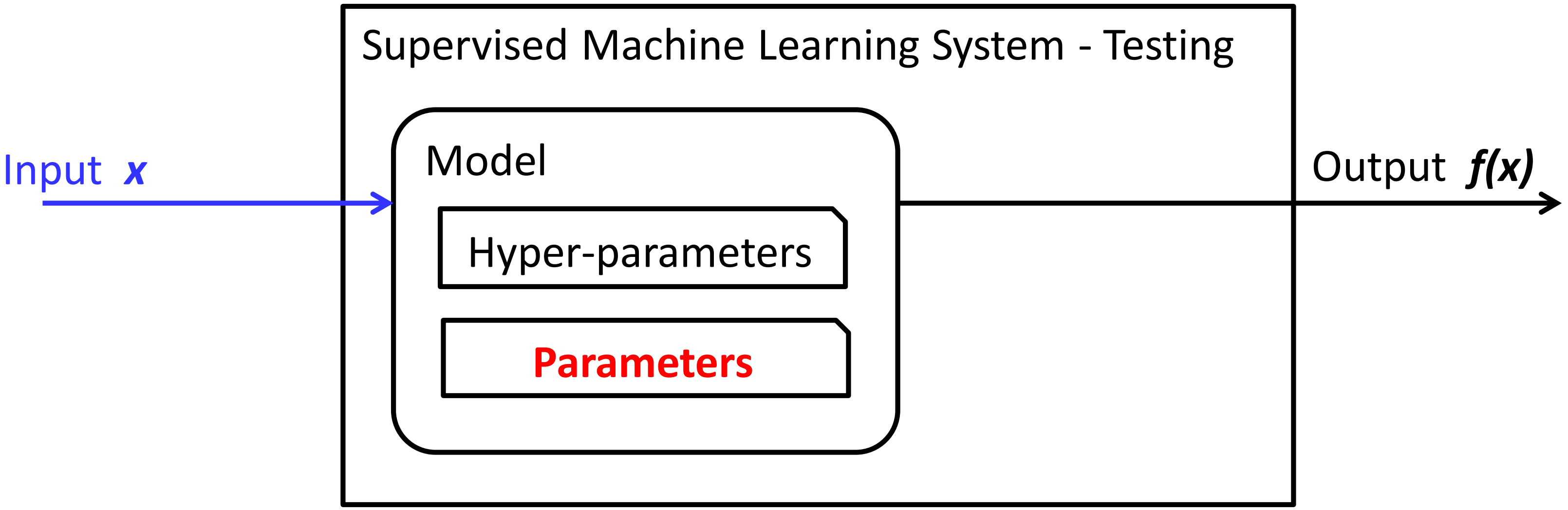
Components of Supervised ML



Definitions of Components of an ML System

- **Input:** what is given, predictive (independent) variables, e.g. images, text
- **Output:** what is desired, predicted (dependent) variable, e.g. image label, text sentiment
- **Hyper-parameter:** element of the model design space, e.g. number of knobs, e.g. number of layers, number of neurons
- **Parameter:** numbers to be learned, knobs to tune, e.g. weights and biases of each neuron
- **Loss function:** measure of performance, low is good
- **Optimization algorithm:** Parameter update rule and schedule

Components of a Trained ML System



Outline

- Components of supervised machine learning
- Model
- Loss function
- Regularization

A model is an estimate of something

- Model is a mathematical function
  - Input $\boldsymbol{x}$
  - Output $f(\boldsymbol{x})$
  - Desired output y approximated by $f(\boldsymbol{x})$, i.e. $y \approx f(\boldsymbol{x})$
- Examples:
  - $f(x) = w\,x + b$ or $w\,x + b\,1$
  - $f(x) = w_2 x^2 + w_1 x^1 + w_0 x^0$
  - $f(x) = \boldsymbol{w}^T\boldsymbol{x}+b$
  - $f(x) = g(\boldsymbol{w}^T\boldsymbol{x}+b)$, where g is a nonlinear function

Examples of hyper-parameters and parameters

- $f(x) = w_2 x^2 + w_1 x^1 + w_0 x^0$
  - Hyper-parameter is degree 2
  - Parameters are w_2, w_1, and w_0

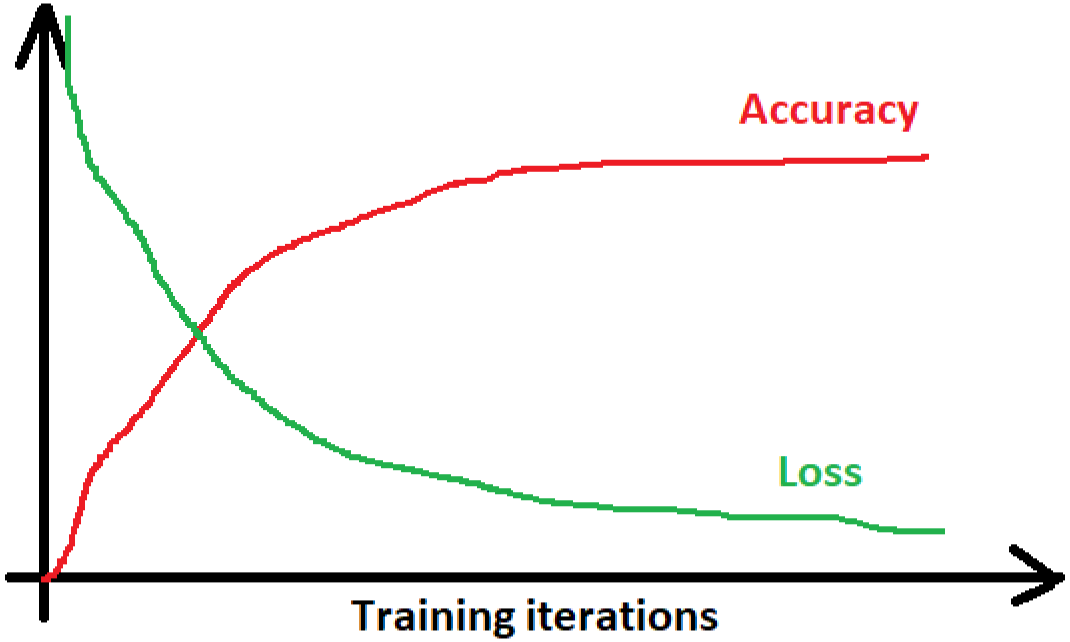
- $f(\mathbf{x}) = h(\mathbf{W}_3\, g(\mathbf{W}_2\, g(\mathbf{W}_1\, \mathbf{x})))$
  - Parameters are elements of $\mathbf{W}_3$, $\mathbf{W}_2$, and $\mathbf{W}_1$
  - Hyper-parameters are number of layers 3, and the number of neurons in each layer (rows of $\mathbf{W}_3$, $\mathbf{W}_2$, and $\mathbf{W}_1$)

Outline

- Components of supervised machine learning
- Model
- Loss function
- Regularization

Loss and accuracy

- Training accuracy saturates to a maximum
- Training loss saturates to a minimum
- Loss is a measure of error



Loss function tells how bad the model is

- Loss trends opposite of accuracy
  - Loss is low when accuracy is high
  - Loss is zero for perfect accuracy (by convention)
  - Loss is high when accuracy is low

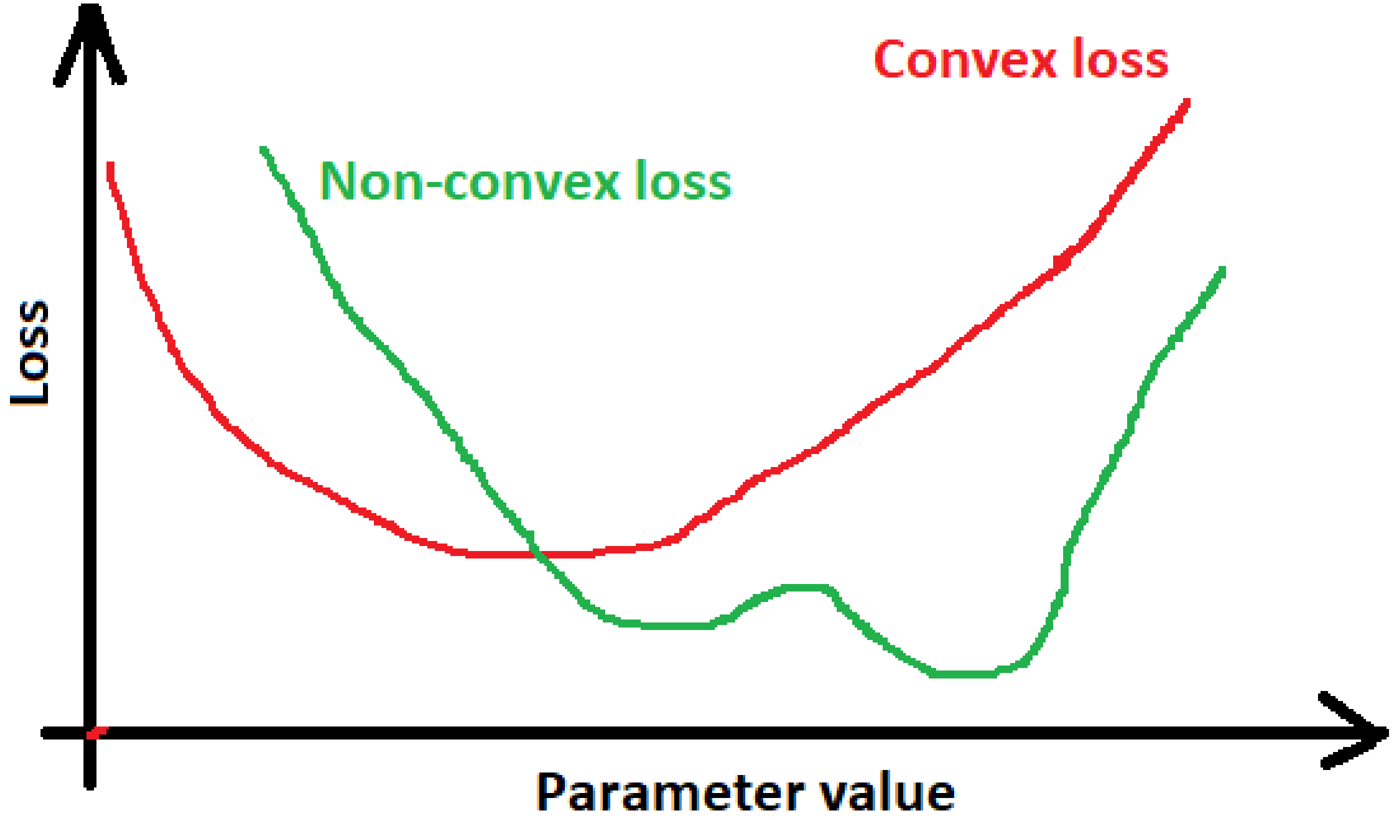
- Loss is a function of actual and desired output

- Minimizing the loss function with respect to parameters leads to good parameters

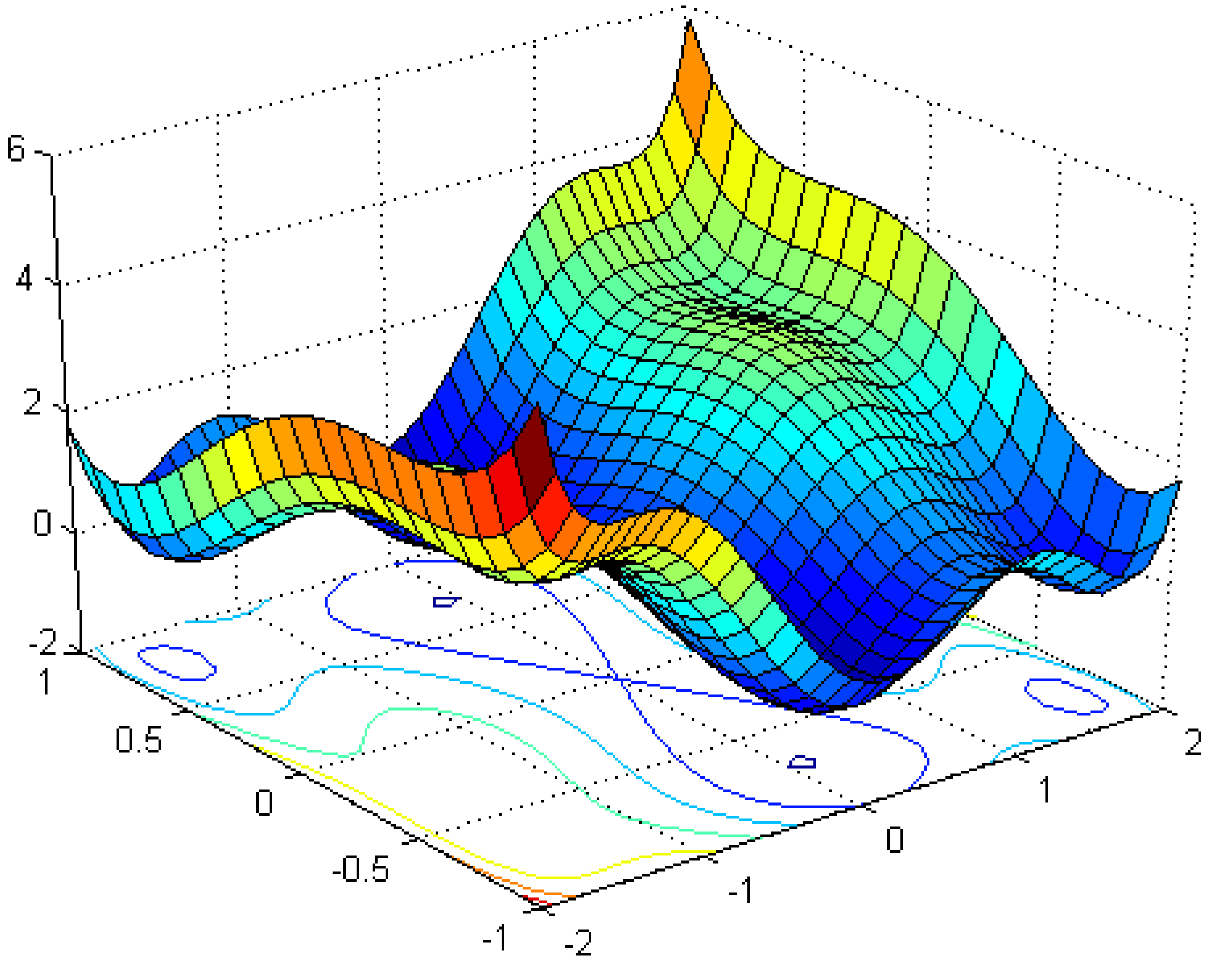
Properties of a good loss function

- Minimum value for perfect accuracy
  - Usually zero
  - *Note:* low loss on training does not guarantee low loss on validation or testing
- Varies smoothly with input
- Varies smoothly with parameters
- Good to be convex in parameters (but is usually not)
  - Like a paraboloid

Convex vs. non-convex loss



Non-convex loss can have multiple minima



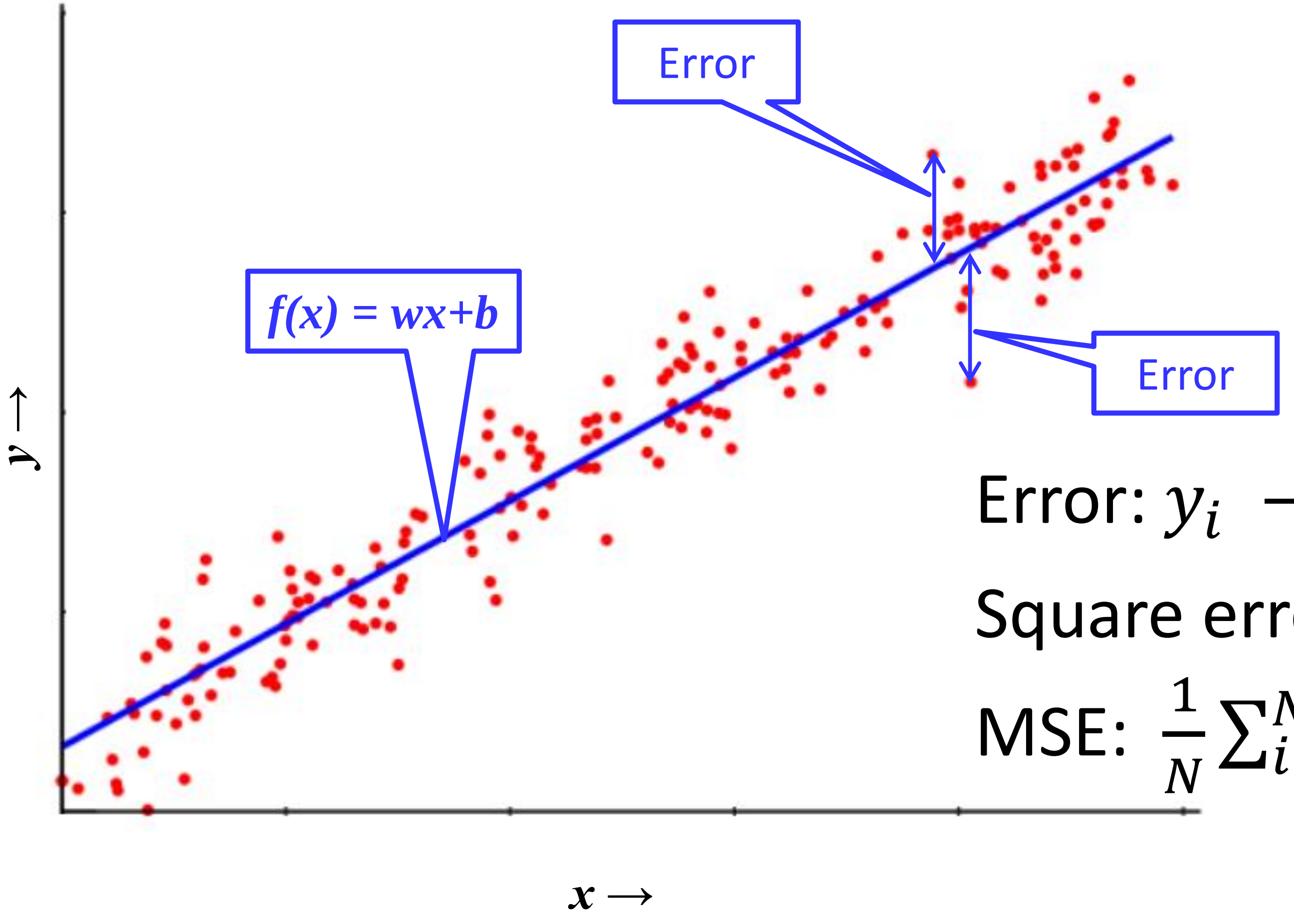
More about loss function

- Choice of loss function depends on:
  - Desired output type: continuous or categorical?
  - Predicted output type: continuous or categorical?
  - Goal: supervised or unsupervised?
- Loss function over a set is the average of loss over each sample in the set
- Loss function over the validation set is the most important thing to monitor during training
- High training loss means under-fitting
- Large gap between training and validation losses means over-fitting

Examples of loss functions

- Regression with continuous output
  - Mean square error (MSE), log MSE, mean absolute error
- Classification with probabilistic output
  - Cross entropy (negative log likelihood), hinge loss
- Similarity between vectors or clustering
  - Euclidean distance, cosine

MSE loss for regression

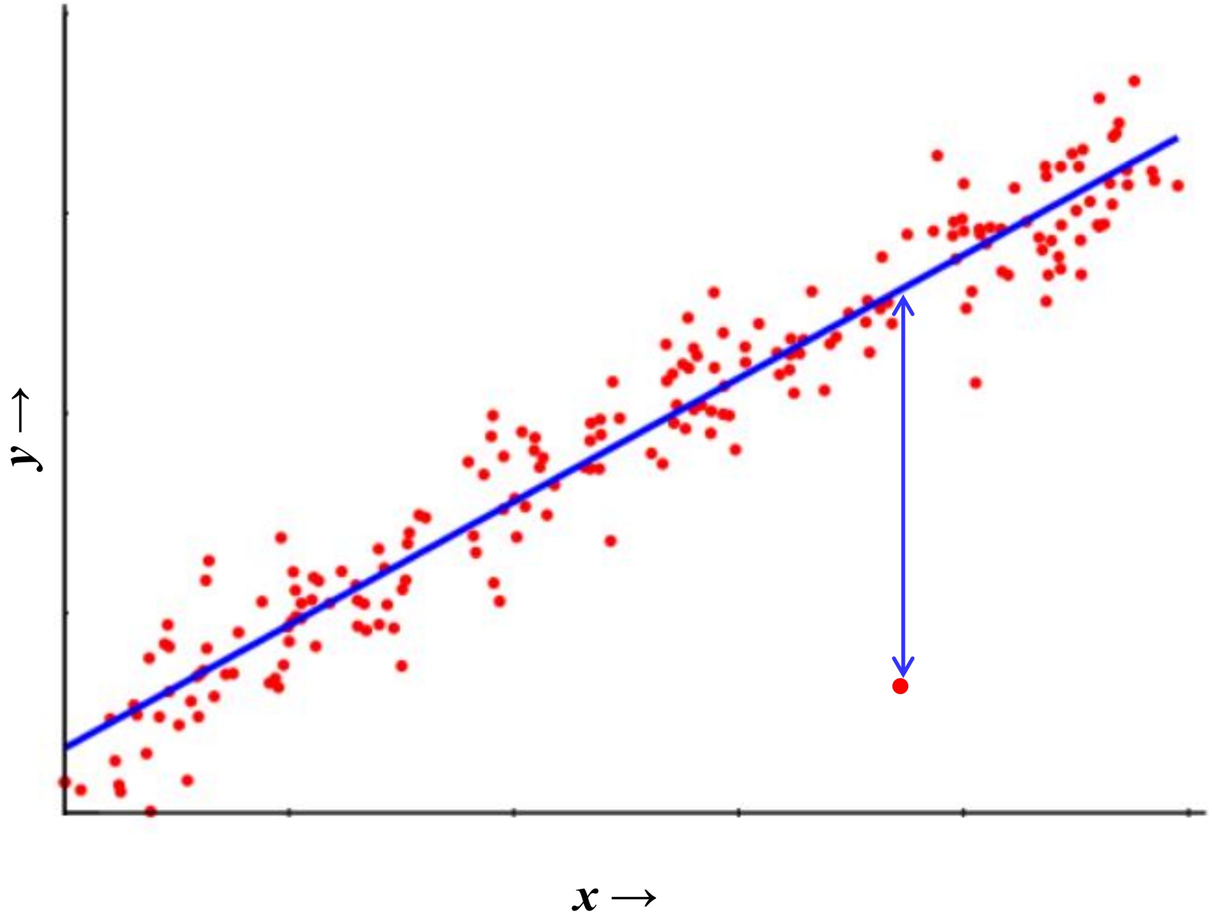


Error: $y_i - f(x_i)$

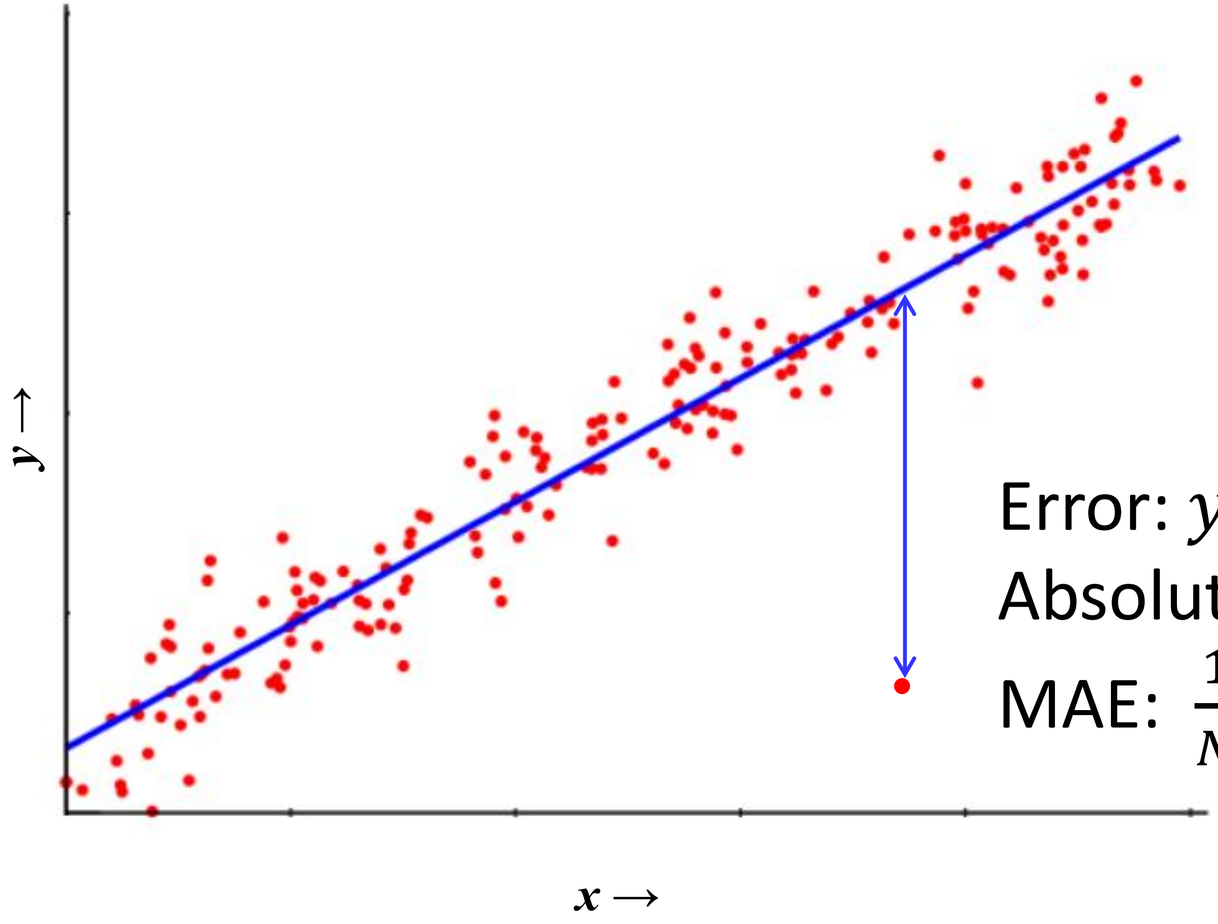
Square error: $\left(y_i - f(x_i)\right)^2$

MSE: $\frac{1}{N}\sum_{i=1}^{N}\left(y_i - f(x_i)\right)^2$

Is MSE always appropriate?



MAE loss is less affected by outliers than MSE

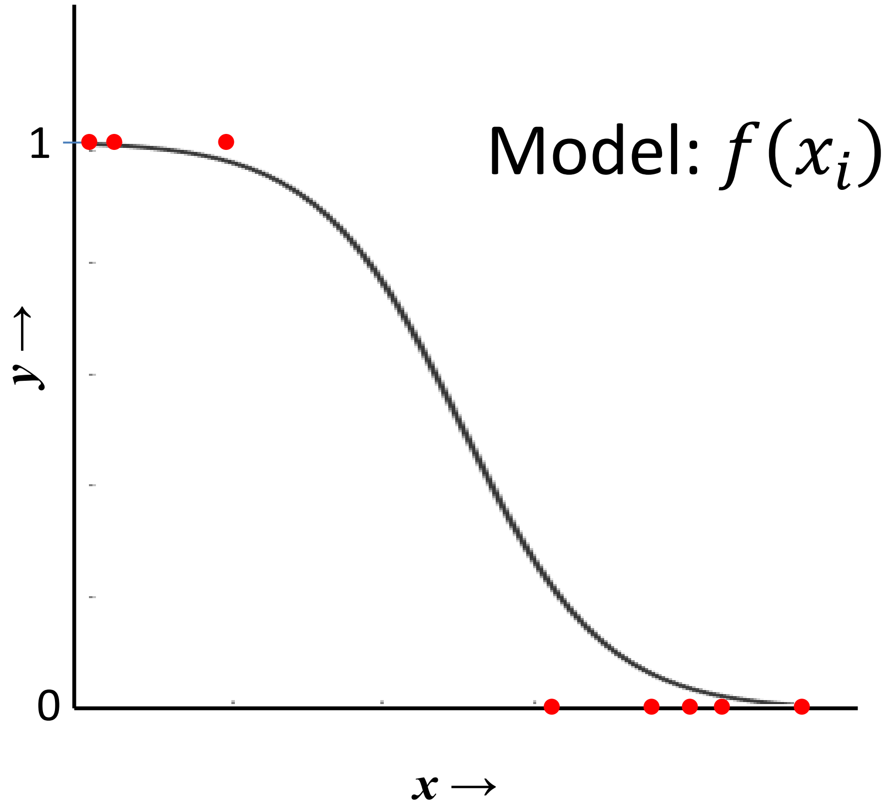


Error: $y_i - f(x_i)$

Absolute error: $|y_i - f(x_i)|$

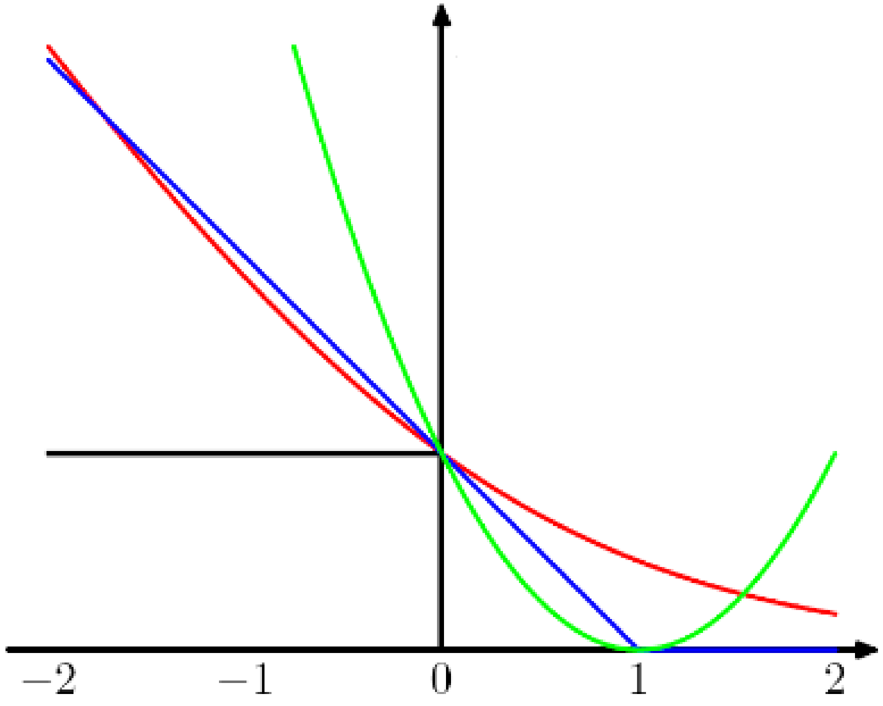
MAE: $\frac{1}{N}\sum_{i=1}^{N}|y_i - f(x_i)|$

Is MSE appropriate for classification?



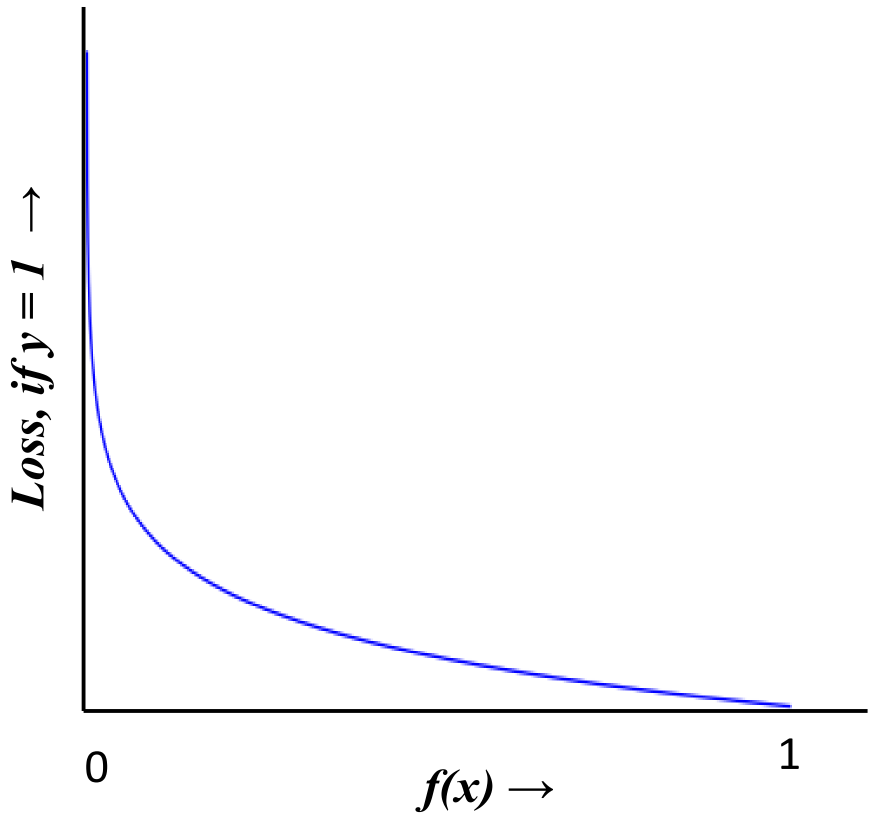
Model: $f(x_i) = \frac{1}{1+e^{-(wx_i+b)}}$

Some loss functions



- Problem: binary classification
- Assumption: desired output is 1
- Notice rate of convergence at different points

Cross entropy loss is preferred for classification



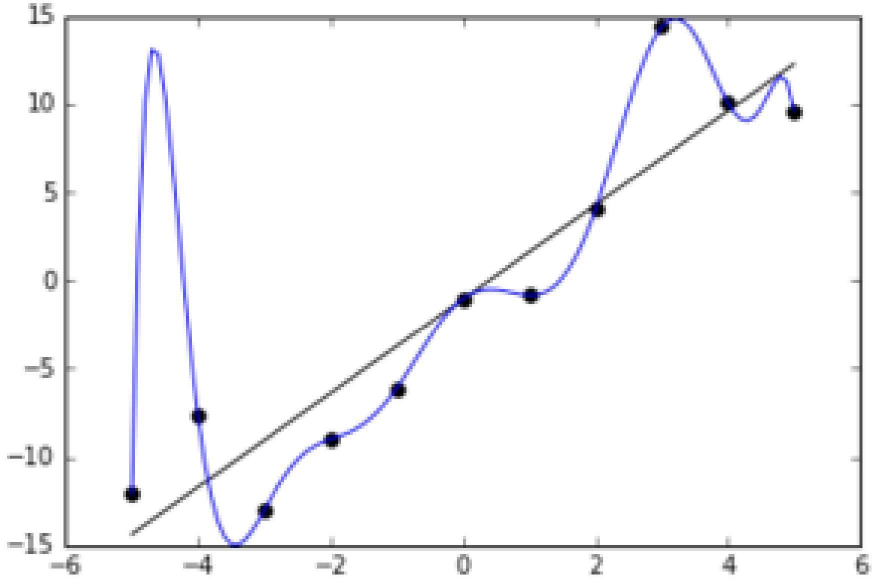
- How much does one (estimated) probability distribution *q(x)* deviates from another (real) *p(x)*
- KL-divergence of *q(x)* from *p(x)*
- For binary classification:

$$-\{y \log f(\boldsymbol{x}) + (1-y)\log\big(1-f(\boldsymbol{x})\big)\}$$

Outline

- Components of supervised machine learning
- Model
- Loss function
- Regularization

Under-constrained models lead to overfitting



- An *n*-degree polynomial can fit *n* points perfectly
- But, is it overfitting?
- Is it being swayed by outliers?
- "Models should be as simple as possible, but not simplistic"
- To make model simpler:
  - Restrict number of parameters,
  - Or, restrict the set of values that they can take
- Always check validation performance

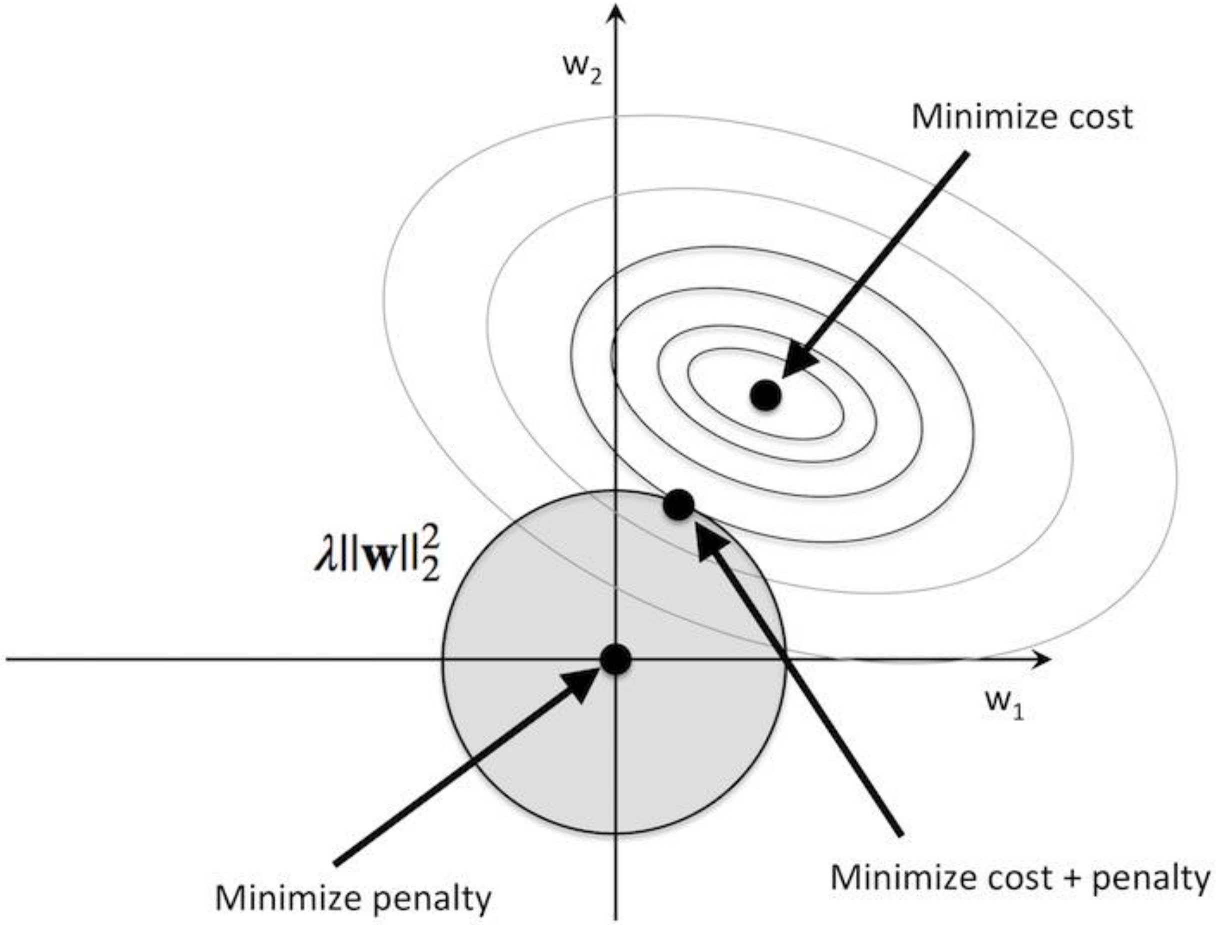
greatlearning
Learning for Life

Regularization is constraining a model

- How to regularize?
  - Reduce the number of parameters
    - Share weights in structure
  - Constrain parameters to be small
  - Encourage sparsity of output in loss
- Most commonly Tikhonov (or L2, or ridge) regularization (a.k.a. weight decay)
  - Penalty on sums of squares of individual weights

$$J = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f(x_i)\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n} w_j^2 \quad ; f(x_i) = \sum_{j=0}^{n} w_j \; x_i^j \quad ;$$

L2-regularization visualized



Other forms of regularization

- Convolutional filter structure in CNN neurons
- Max-pooling
- Dropout
- L1-regularization (sparsity inducing norm)
  - Penalty on sums of absolute values of weights