

AI in Culture and Arts – Tech Crash Course

Introduction to Deep Learning

Benedikt Zönnchen

9th of April 2025

MUC.DAI
Munich Center for
Digital Sciences and AI



myt Hochschule
für Musik und Theater
München

1. How Do Machines Learn?

2. How Do Humans Train Machines?

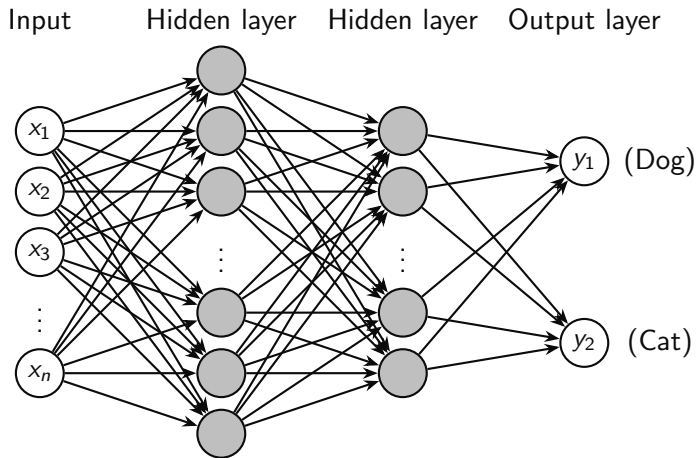
3. Interactions with ML

**1. How Do Machines
Learn?**

**2. How Do Humans Train
Machines?**

3. Interactions with ML

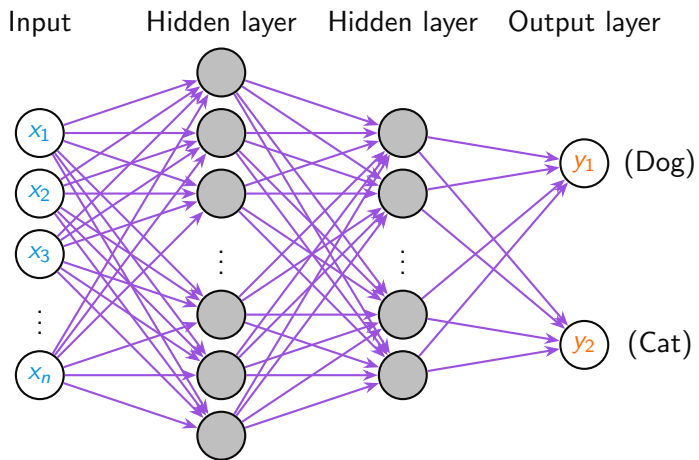
Synaptic Plasticity



$$h_{\theta}(\mathbf{x}) = \mathbf{y},$$

where $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_k)$

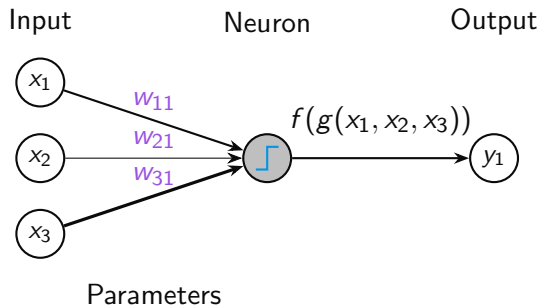
Synaptic Plasticity



$$h_{\theta}(\mathbf{x}) = \mathbf{y},$$

where $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_k)$

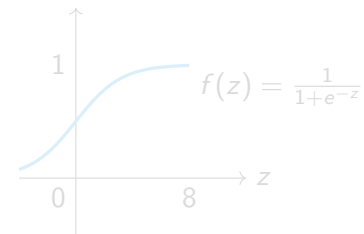
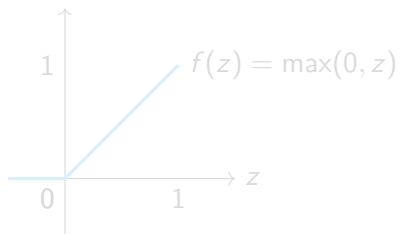
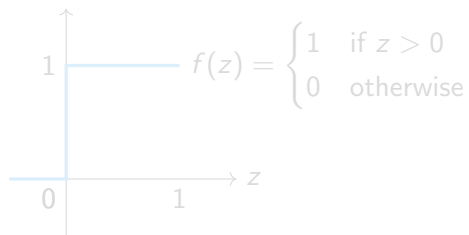
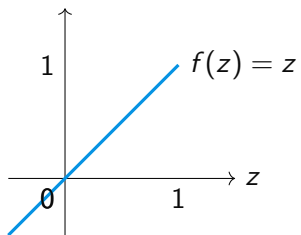
Synaptic Plasticity



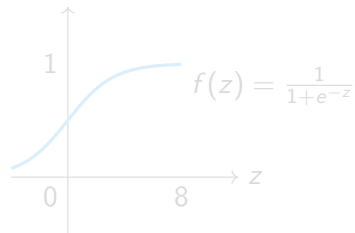
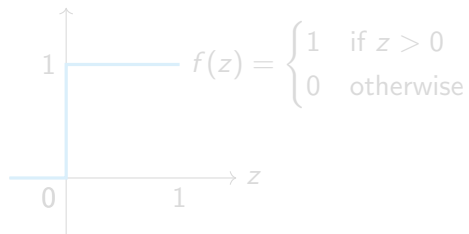
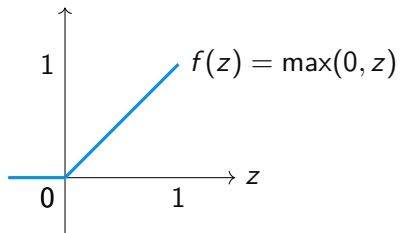
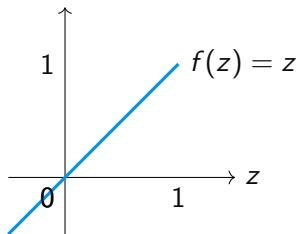
Parameters determine how strong neurons are wired together:

$$g(x_1, x_2, x_3) = x_1 \cdot w_{11} + x_2 \cdot w_{21} + x_3 \cdot w_{31}$$

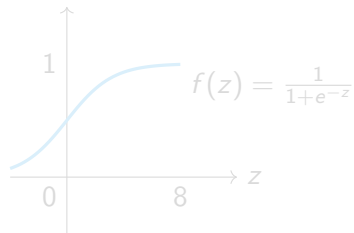
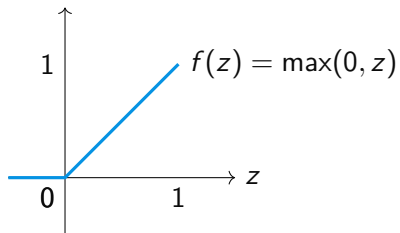
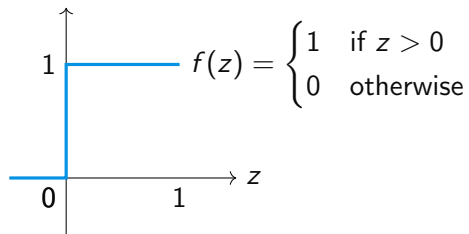
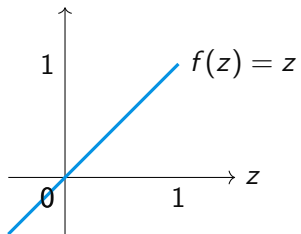
Activation Functions



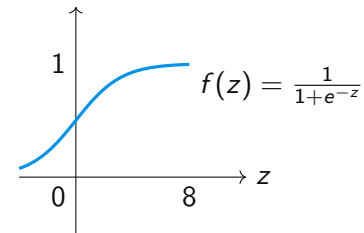
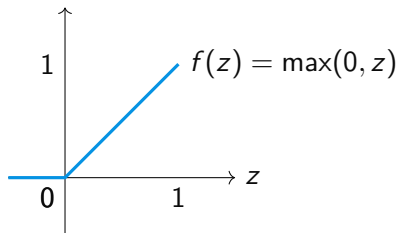
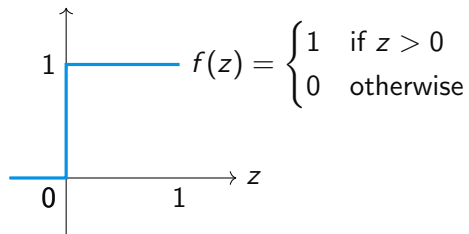
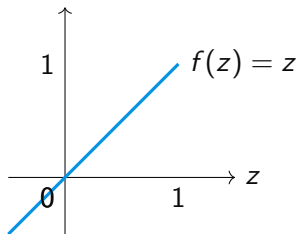
Activation Functions



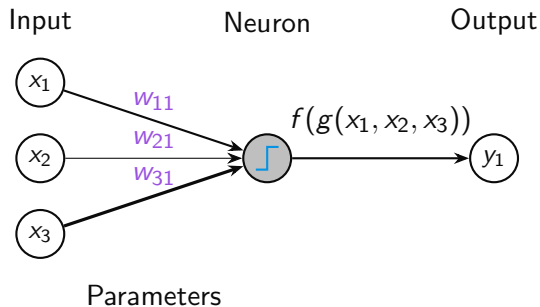
Activation Functions



Activation Functions



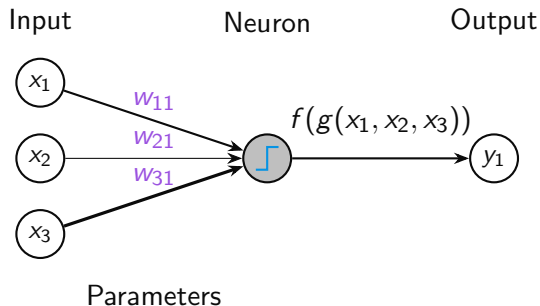
Synaptic Plasticity



"Neurons that fire together, wire together."

$$w_{ij} = w_{ij} - \eta \cdot x_i \cdot y_j$$

Synaptic Plasticity



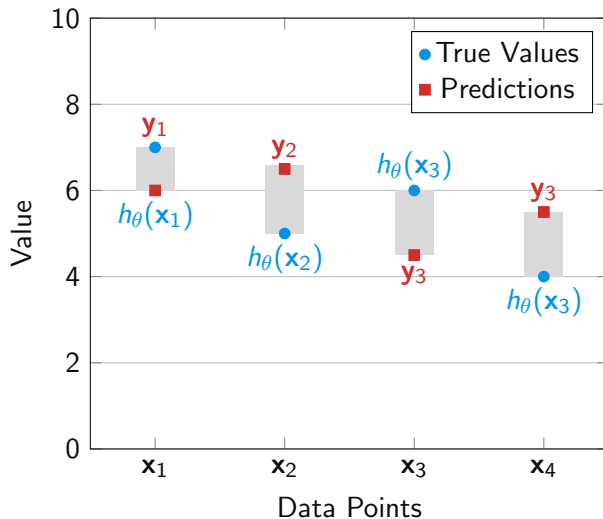
"Neurons that fire together, wire together."

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla J(\theta_t)$$

In this case $\theta_t = (w_{11}, w_{21}, w_{31})$.

Cost Function (Regression)

Mean Squared Error (MSE):



Mean Squared Error (MSE):

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N (\mathbf{y}_i - h_{\theta}(\mathbf{x}_i))^2$$

where \mathbf{y}_i is the correct label of a data point $\mathbf{x}_i = (x_1, \dots, x_n)$ in our training data.

Idea: Let's say our prediction classifies our i -th image, that is \mathbf{x}_i , as 0.3 dog and 0.7 cat:

$$h_{\theta}(\mathbf{x}_i) = (0.3, 0.7)$$

but in reality it is most certainly a dog, that is, $(0.95, 0.05)$. A good error would be:

$$- [0.95 \cdot 0.3 \cdot (1 - 0.95) \cdot (1 - 0.3)] \cdot [0.05 \cdot 0.7 \cdot (1 - 0.05) \cdot (1 - 0.7)]$$

This term is minimal for $\mathbf{x}_i = (0.95, 0.05)$

Idea: Let's say our prediction classifies our i -th image, that is \mathbf{x}_i , as 0.3 dog and 0.7 cat:

$$h_{\theta}(\mathbf{x}_i) = (0.3, 0.7)$$

but in reality it is most certainly a dog, that is, $(0.95, 0.05)$. A good error would be:

$$- [0.95 \cdot 0.3 \cdot (1 - 0.95) \cdot (1 - 0.3)] \cdot [0.05 \cdot 0.7 \cdot (1 - 0.05) \cdot (1 - 0.7)]$$

This term is minimal for $\mathbf{x}_i = (0.95, 0.05)$

Idea: Let's say our prediction classifies our i -th image, that is \mathbf{x}_i , as 0.3 dog and 0.7 cat:

$$h_{\theta}(\mathbf{x}_i) = (0.3, 0.7)$$

but in reality it is most certainly a dog, that is, $(0.95, 0.05)$. A good error would be:

$$- [0.95 \cdot 0.3 \cdot (1 - 0.95) \cdot (1 - 0.3)] \cdot [0.05 \cdot 0.7 \cdot (1 - 0.05) \cdot (1 - 0.7)]$$

This term is minimal for $\mathbf{x}_i = (0.95, 0.05)$

Categorical Cross Entropy Cost:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N [\mathbf{y}_i \cdot \log(h_{\theta}(\mathbf{x}_i)) + (\mathbf{1} - \mathbf{y}_i) \cdot (\mathbf{1} - \log(h_{\theta}(\mathbf{x}_i)))]$$

where \mathbf{y}_i is interpreted as the probability distribution of categories for $\mathbf{x}_i = (x_1, \dots, x_n)$,
i.e. a data point.

To improve the model's prediction, we try to minimize the cost function. One way to do this is **gradient decent**:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla J(\theta_t)$$

Condition: $\nabla J(\theta_t)$ exits!

Interactive Tutorial

To improve the model's prediction, we try to minimize the cost function. One way to do this is **gradient decent**:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla J(\theta_t)$$

Condition: $\nabla J(\theta_t)$ exits!

Interactive Tutorial

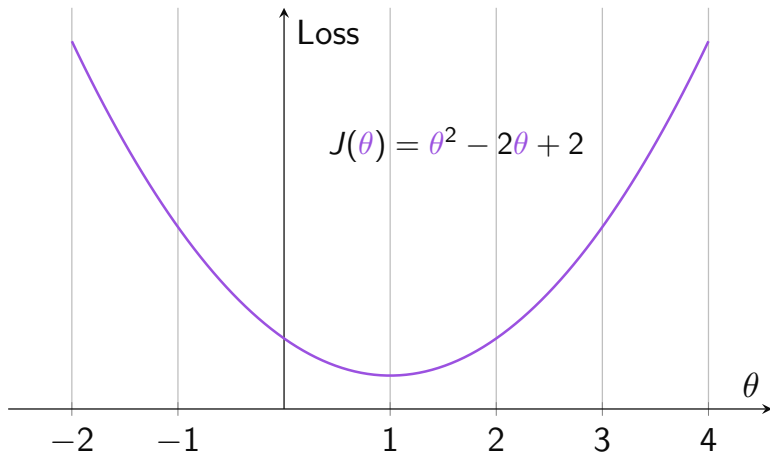
To improve the model's prediction, we try to minimize the cost function. One way to do this is **gradient decent**:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla J(\theta_t)$$

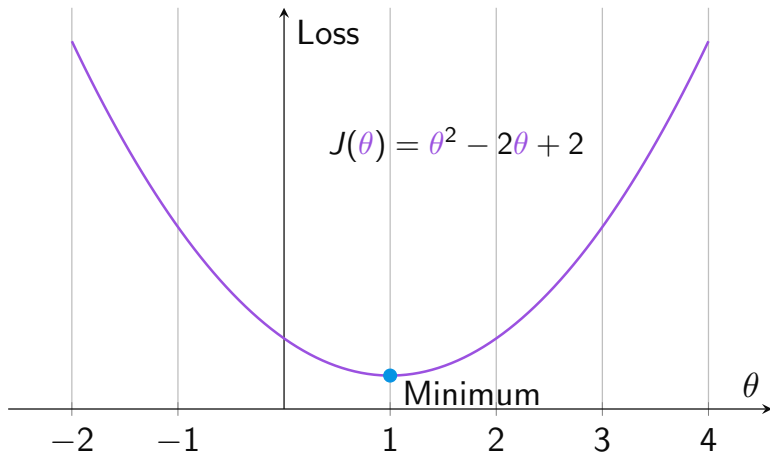
Condition: $\nabla J(\theta_t)$ exits!

Interactive Tutorial

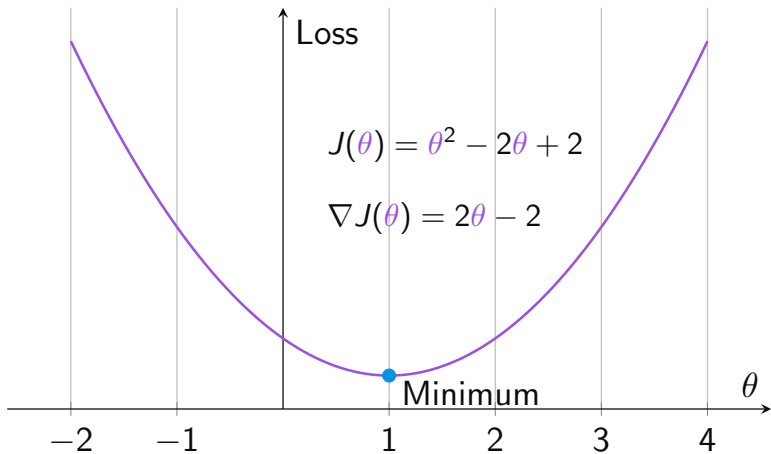
Gradient Decent



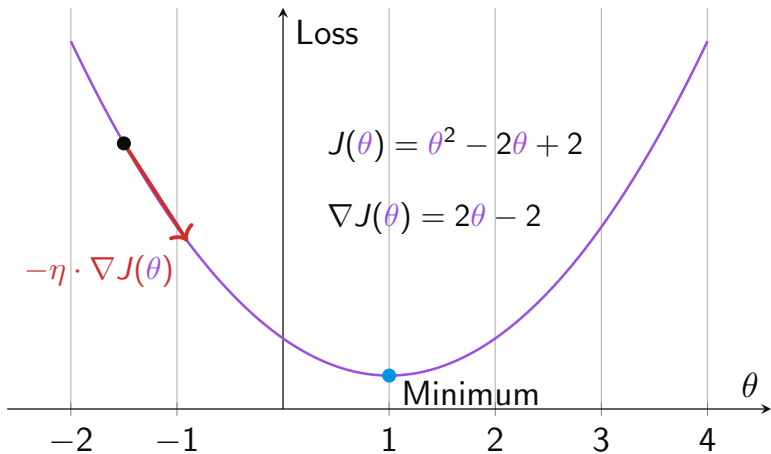
Gradient Decent



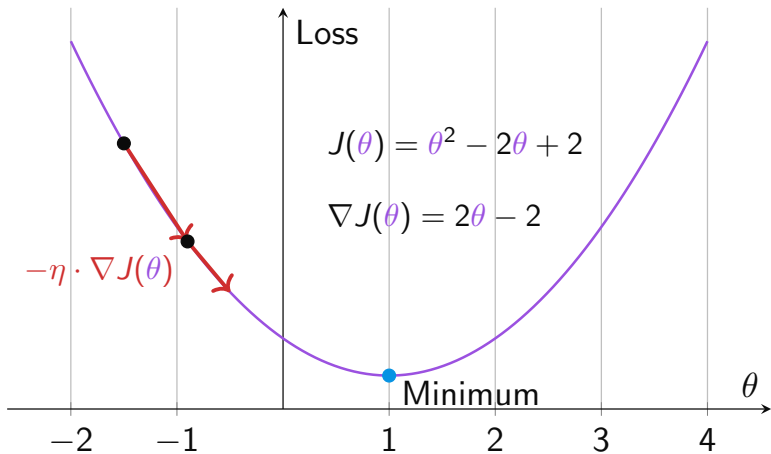
Gradient Decent



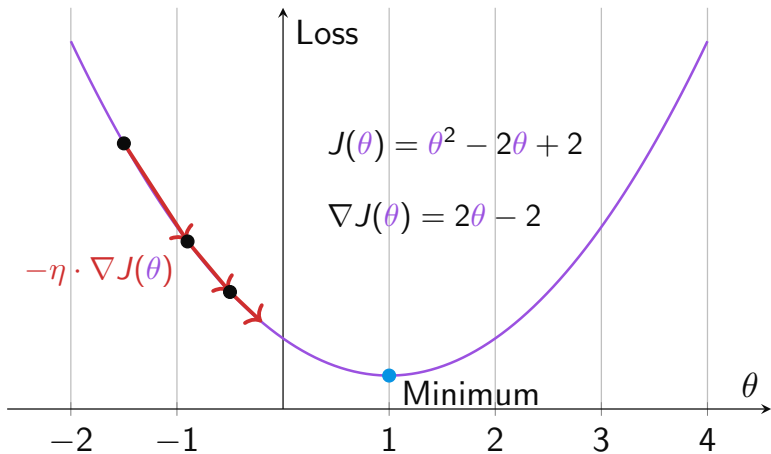
Gradient Decent



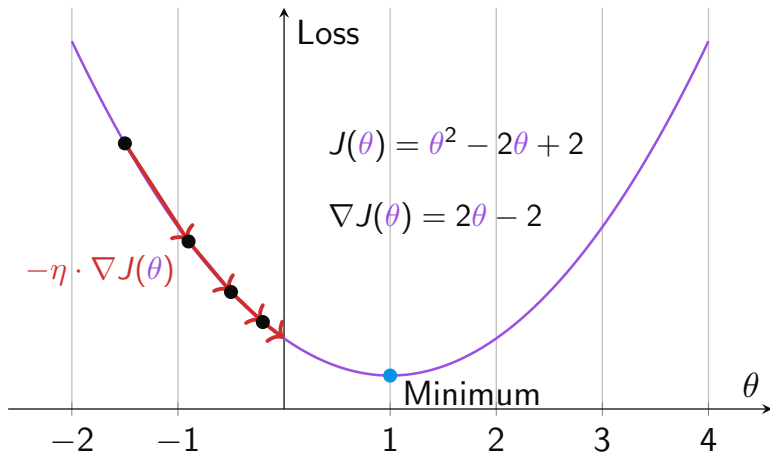
Gradient Decent



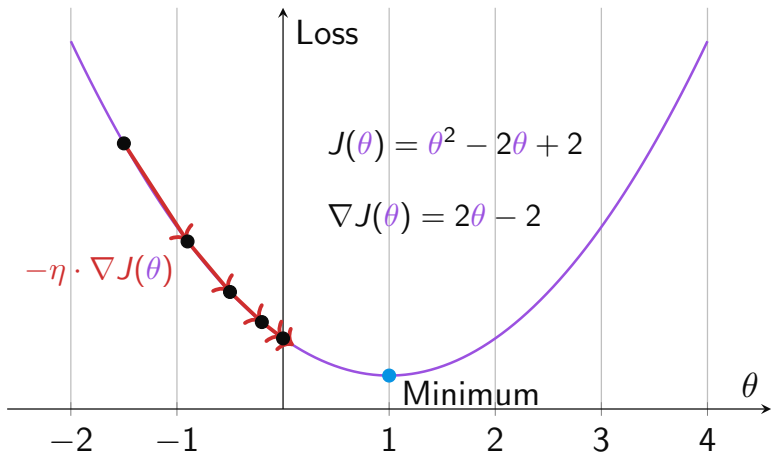
Gradient Decent



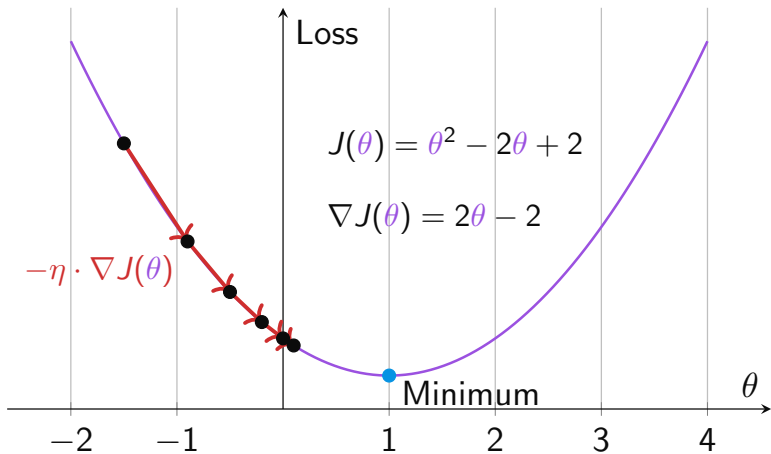
Gradient Decent



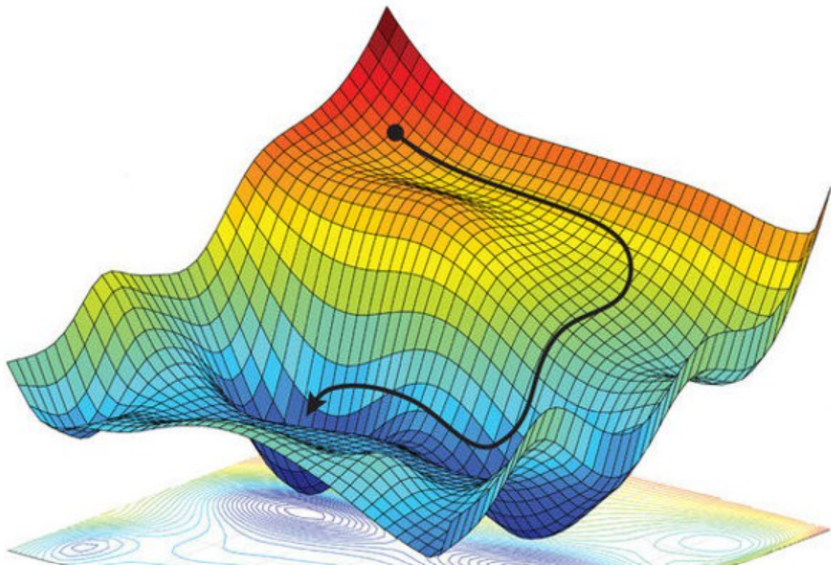
Gradient Decent



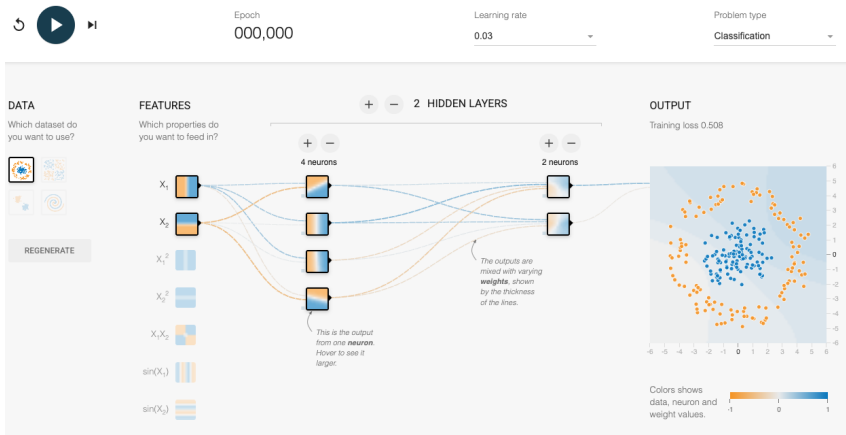
Gradient Decent



Gradient Decent



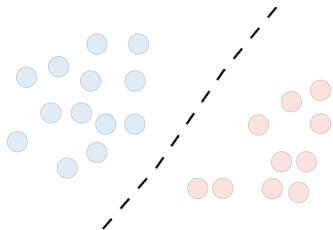
Design and Try Your Perceptron



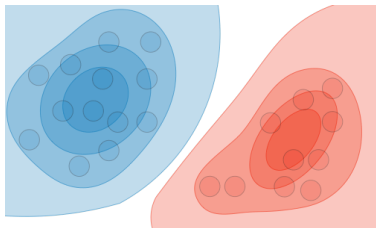
Simplified Tensorflow Playground

Extended Tensorflow Playground

- **Discriminative models:** Learn the boundaries of decisions.
- **Generative models:** Learn the whole distribution of the data.



Discriminative modelling



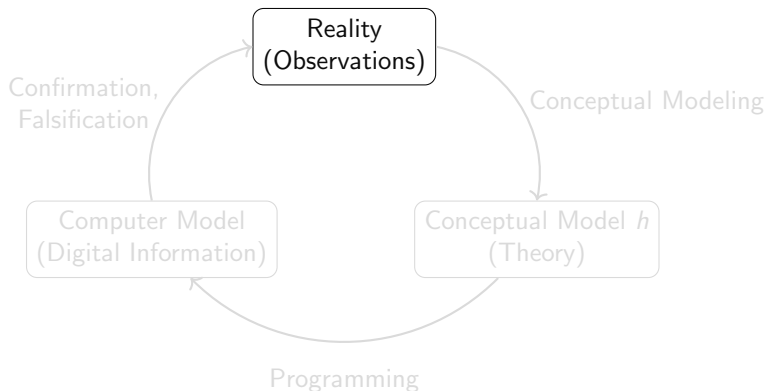
Generative modelling

1. How Do Machines Learn?

2. How Do Humans Train Machines?

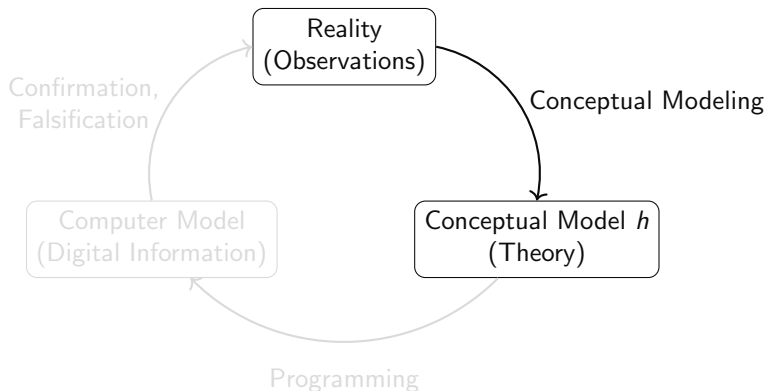
3. Interactions with ML

Theory-driven Modeling



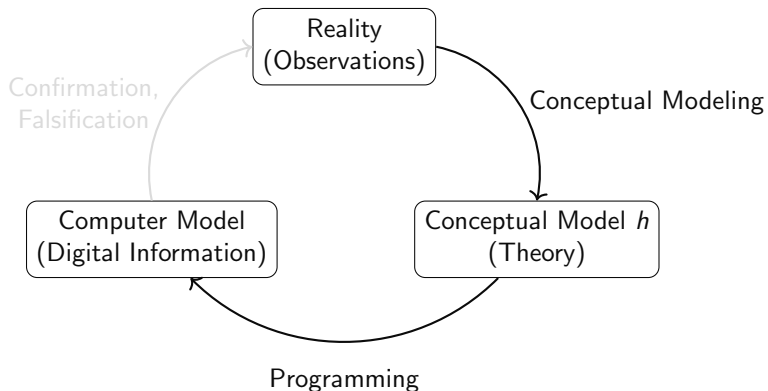
Minds constructs a (falsifiable) theory or hypothesis about reality to test against.

Theory-driven Modeling



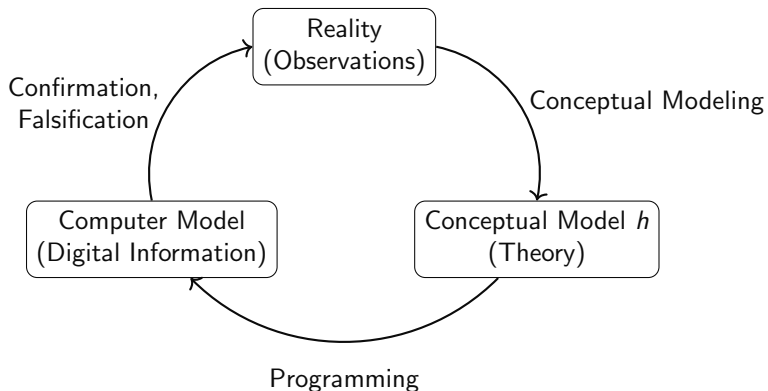
Minds constructs a (falsifiable) theory or hypothesis about reality to test against.

Theory-driven Modeling



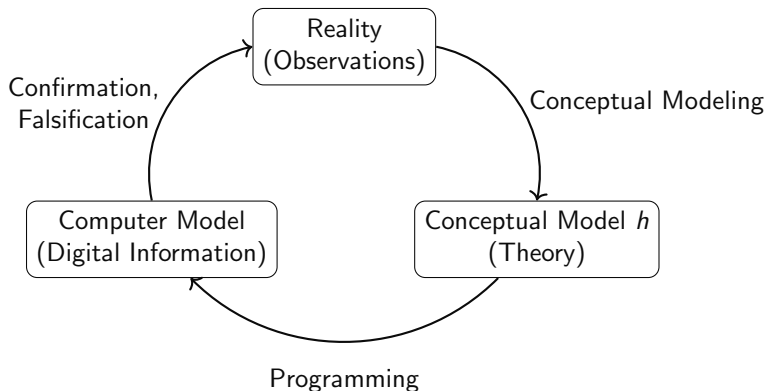
Minds constructs a (falsifiable) theory or hypothesis about reality to test against.

Theory-driven Modeling



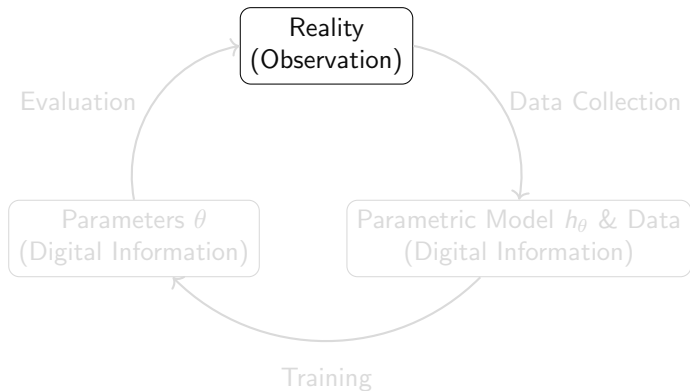
Minds constructs a (falsifiable) theory or hypothesis about reality to test against.

Theory-driven Modeling



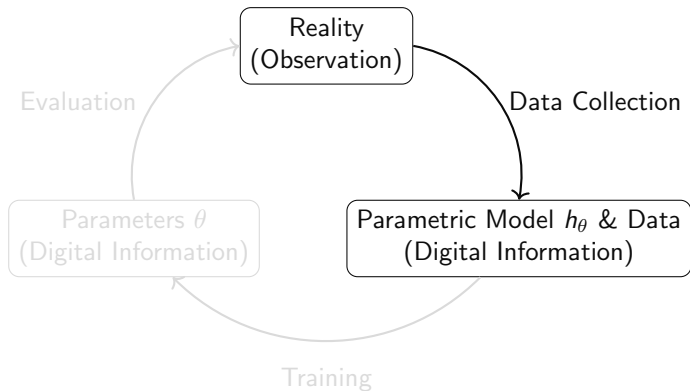
Minds constructs a (falsifiable) theory or hypothesis about reality to test against.

Data-driven Modeling

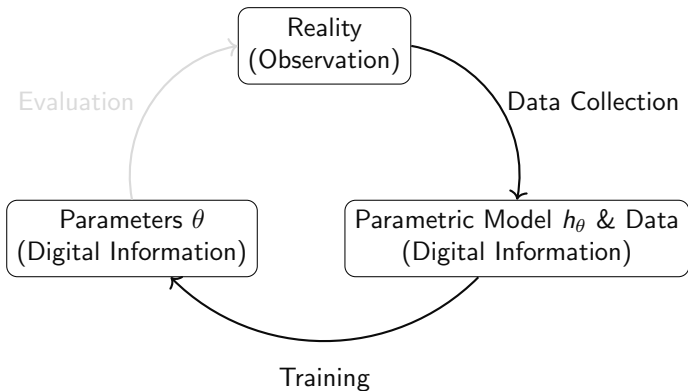


Algorithms (directly) fit a parametric model to the data. **Minds** are usually unable to conceptualize the trained model.

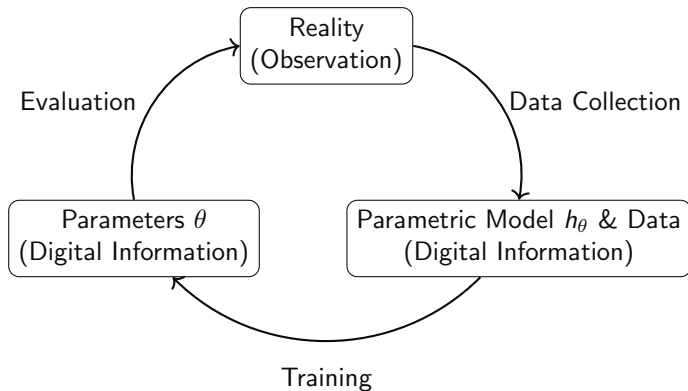
Data-driven Modeling



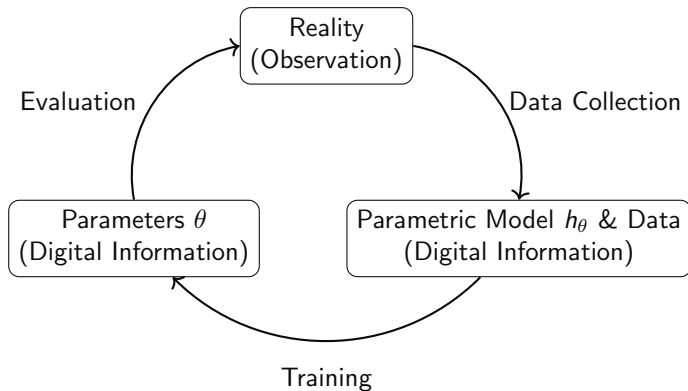
Algorithms (directly) fit a parametric model to the data. **Minds** are usually unable to conceptualize the trained model.



Algorithms (directly) fit a parametric model to the data. **Minds** are usually unable to conceptualize the trained model.

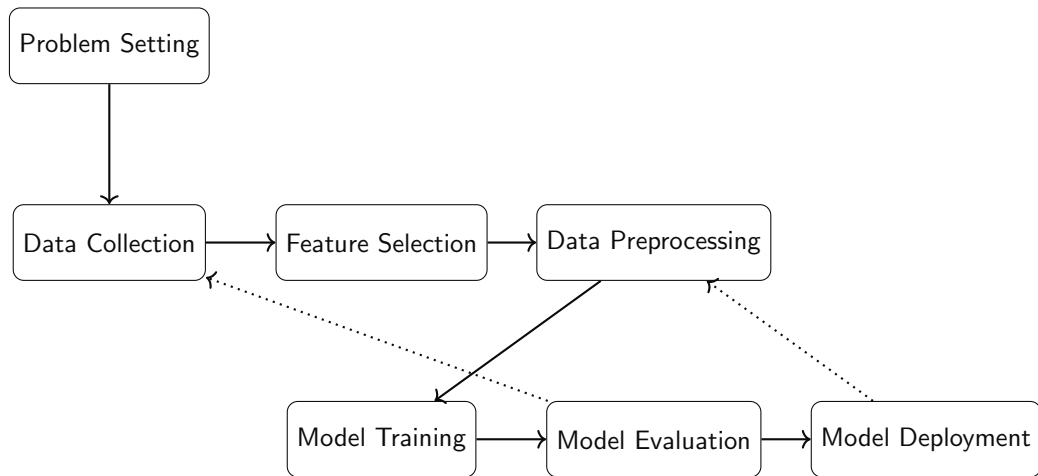


Algorithms (directly) fit a parametric model to the data. **Minds** are usually unable to conceptualize the trained model.



Algorithms (directly) fit a parametric model to the data. **Minds** are usually unable to conceptualize the trained model.

Development Cycle



```
100 class fast_qlinear(torch.autograd.Function):
101     def forward(ctx, a, b, scales, zeros):
102
103         m, k = a.shape
104         _, n = b.shape
105
106         quant_groupsize = 128
107         block_size_m = 16
108         block_size_n = 32 # [N = 4096 // 32] = 128 blocks
109         block_size_k = 256
110         group_size_m = 8
111         num_warps = 4
112         num_stages = 8
113         total_blocks_m = triton.cdiv(m, block_size_m)
114         total_blocks_n = triton.cdiv(n, block_size_n)
```

Python and ML libraries (PyTorch, tensorflow, JAX etc.)

Train a Model with Python

1. How Do Machines Learn?

2. How Do Humans Train Machines?

3. Interactions with ML

Marcelle: composing interactive machine learning workflows and interfaces (?, ?).

<https://marcelle.dev/>

The Marcelle Toolkit

Marcelle Example - Dashboard

Data Management

Training

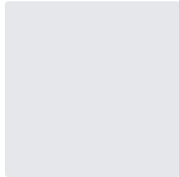
Batch Prediction

Real-time Prediction



webcam

☐ activate video



mobileNet

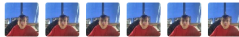
Using Mobilenet v1 with alpha = 1.

Instance label

Capture instances to the training set

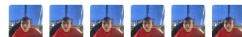
dataset browser

This dataset contains 65 instances.



[View More](#)

Bene



[View More](#)

Nicht Bene



[View More](#)

**1. How Do Machines
Learn?**

**2. How Do Humans Train
Machines?**

3. Interactions with ML

**Any
questions?**

Françoise, J., Caramiaux, B., & Sanchez, T. (2021). Marcelle: Composing interactive machine learning workflows and interfaces. In *The 34th annual acm symposium on user interface software and technology* (pp. 39–53). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3472749.3474734> doi: 10.1145/3472749.3474734