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

HM

Hochschule
für Musik und Theater
München

1. How Do Machines Learn?

2. How Do Humans Train Machines?

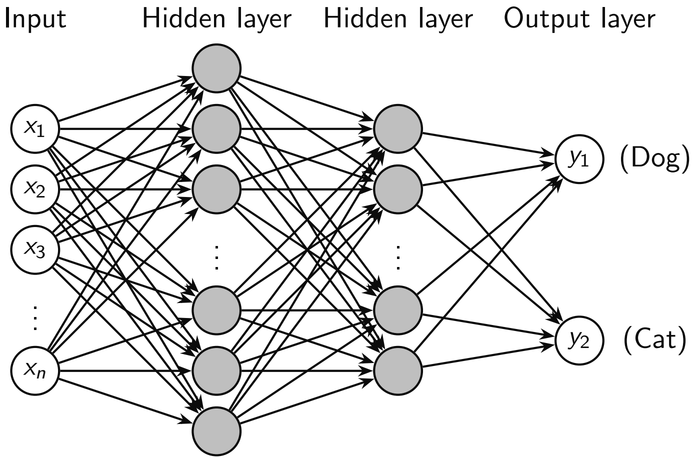
3. Interactions with ML

1. How Do Machines Learn?

2. How Do Humans Train Machines?

3. Interactions with ML

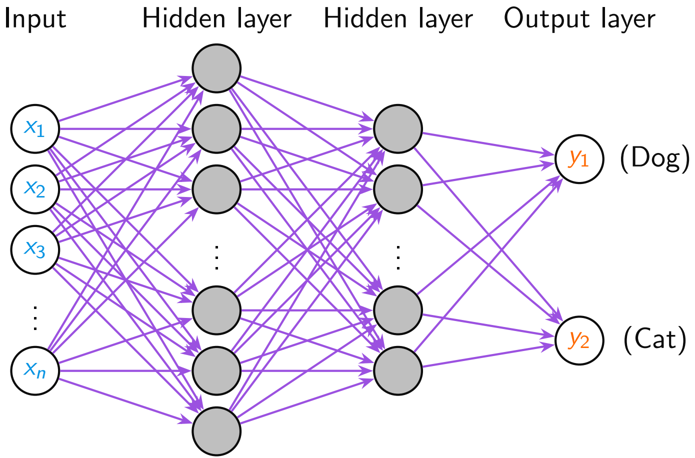
Synaptic Plasticity



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

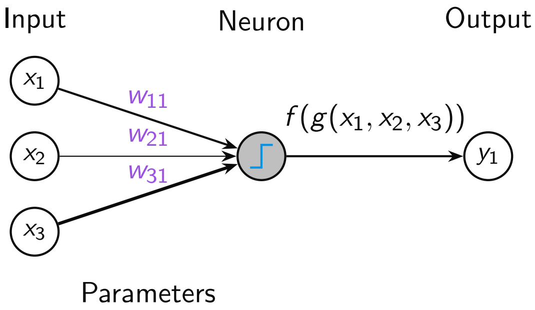
where $\mathbf{x} = (x_1, \ldots x_n)$ and $\mathbf{y} = (y_1, \ldots, y_k)$

Synaptic Plasticity



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

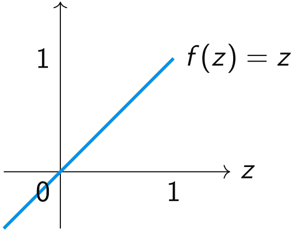
where $\mathbf{x} = (x_1, \ldots x_n)$ and $\mathbf{y} = (y_1, \ldots, y_k)$



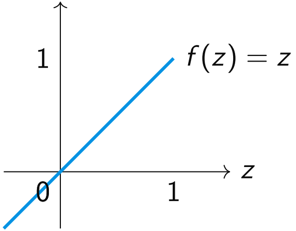
Parameters determine how strong neurons are wired together:

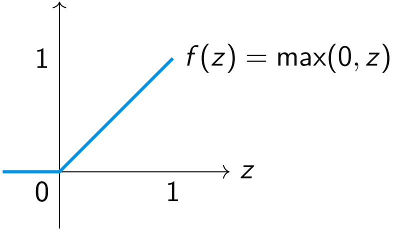
$$g(x_1, x_2, x_3) = x_1 \cdot w_{11} + x_2 \cdot w_{21} + x_3 \cdot w_{31}$$

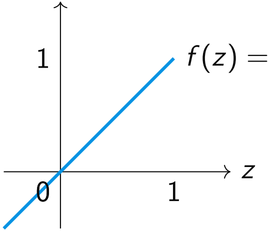
Activation Functions



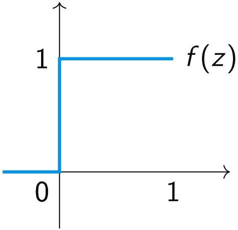
Activation Functions



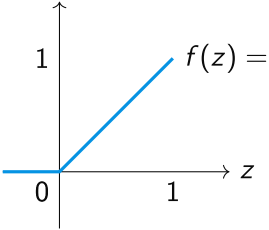




$f(z) = z$



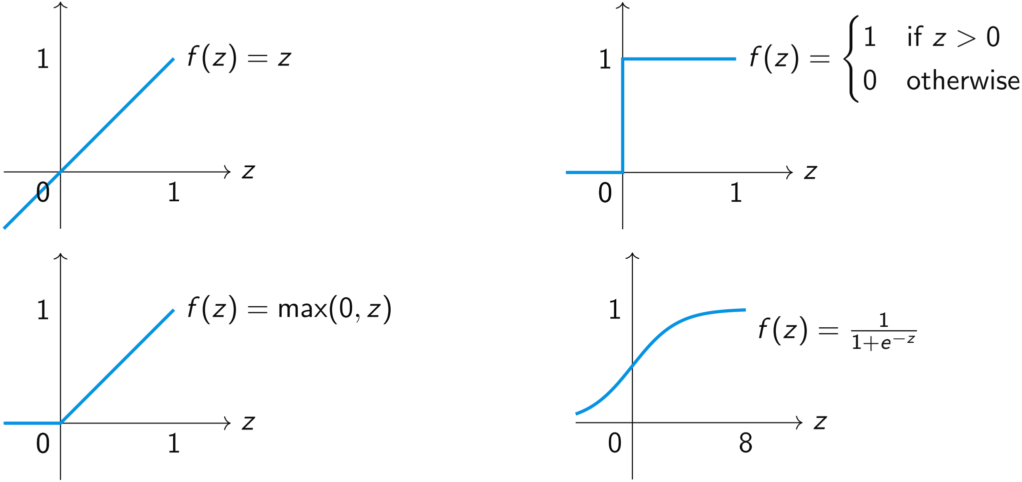
$f(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases}$



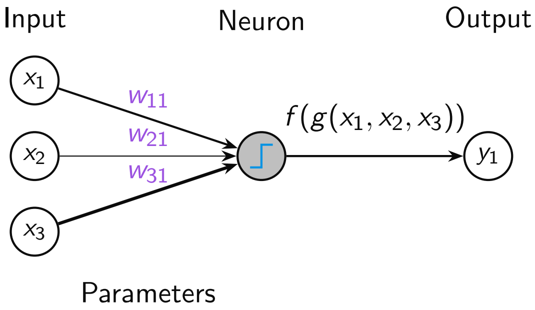
$f(z) = \max(0, z)$

$f(z) = \frac{1}{1+e^{-z}}$

Activation Functions

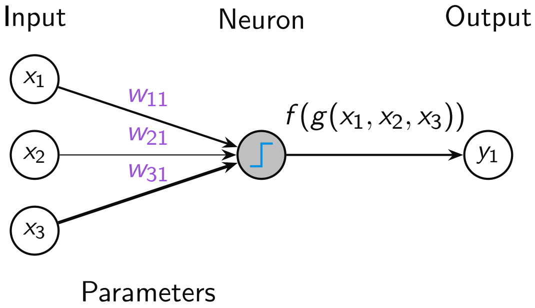


Synaptic Plasticity



"Neurons that fire together, wire together."

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

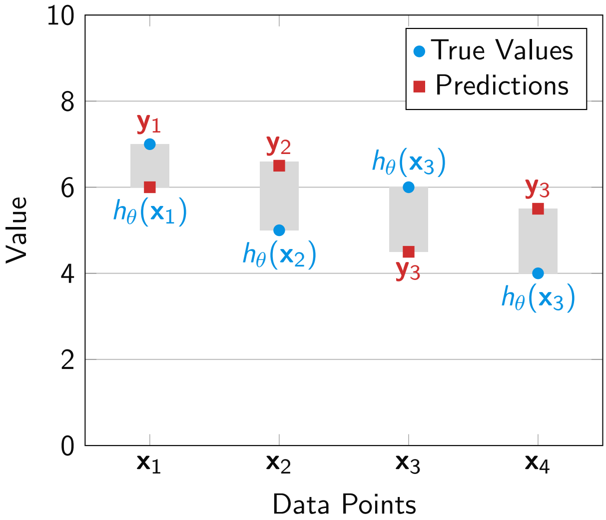


"Neurons that fire together, wire together."

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

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

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 $\mathbf{x}_i = (x_1, \ldots, x_n)$ in our training data.

Idea: Let's say our preduction clasifies 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 preduction clasifies 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:

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

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

Idea: Let's say our preduction clasifies 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:

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

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

Categorical Cross Entropy Cost:

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

where $\mathbf{y}_i$ is interpreted as the probability distribution of categories for $\mathbf{x}_i = (x_1, \ldots, x_n)$, 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!

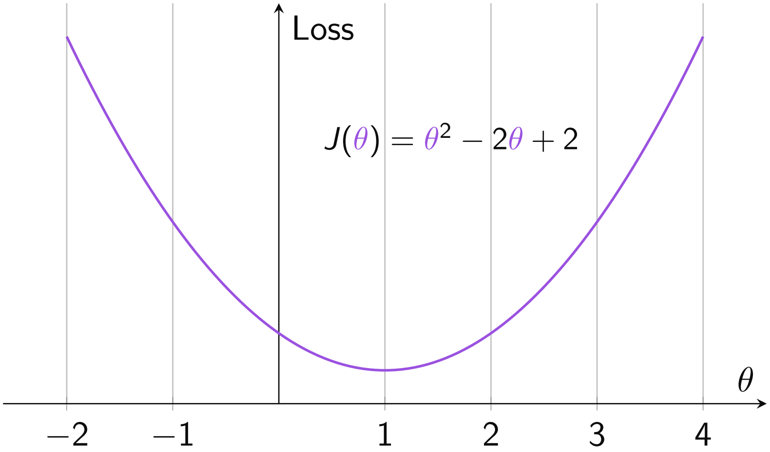
Interactive Tutorial

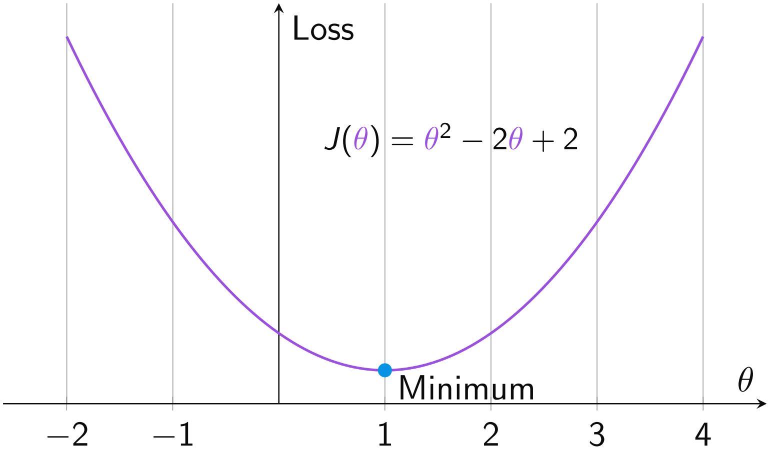
To improve the model's prediction, we try to minimize the cost function. One way to do this is **gradient decent**:

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

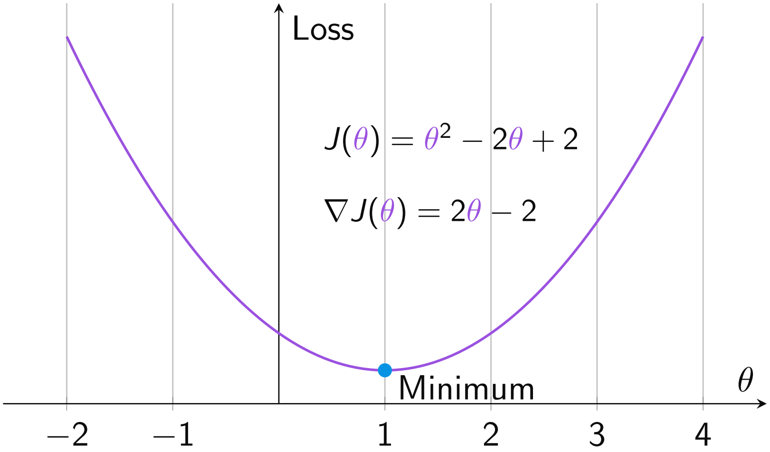
Condition: $\nabla J(\theta_t)$ exits!

Interactive Tutorial

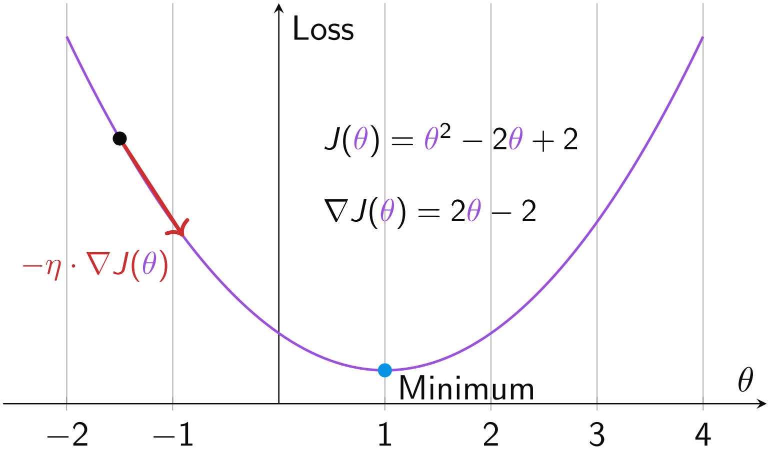

Loss
$J(\theta) = \theta^2 - 2\theta + 2$
θ
-2
-1
1
2
3
4

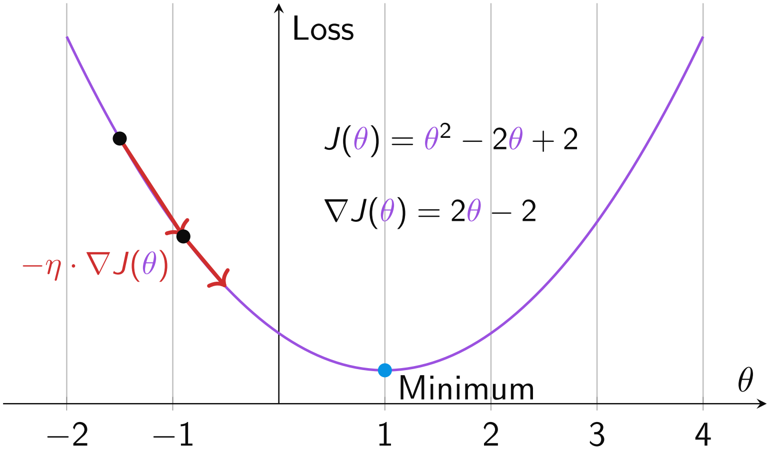

Loss
$J(\theta) = \theta^2 - 2\theta + 2$
Minimum
θ
−2
−1
1
2
3
4

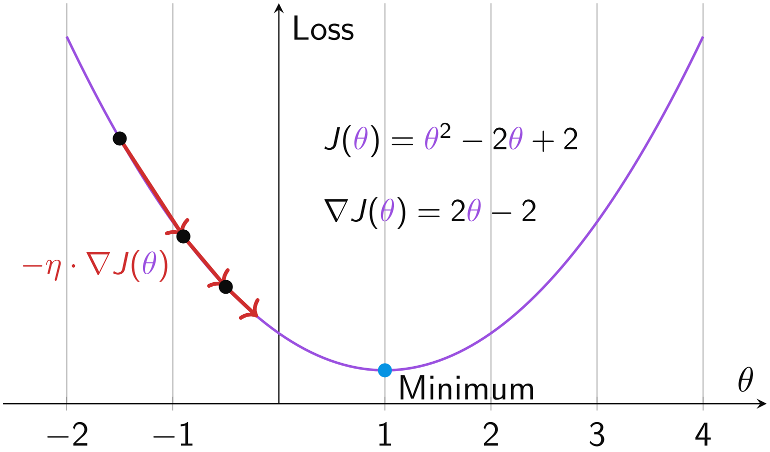
Gradient Decent



Gradient Decent

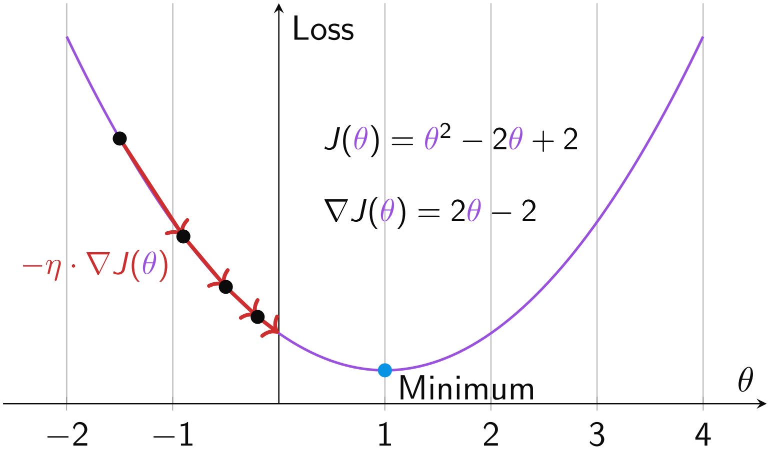



Loss
$J(\theta) = \theta^2 - 2\theta + 2$
$\nabla J(\theta) = 2\theta - 2$
$-\eta \cdot \nabla J(\theta)$
Minimum
θ
-2
-1
1
2
3
4

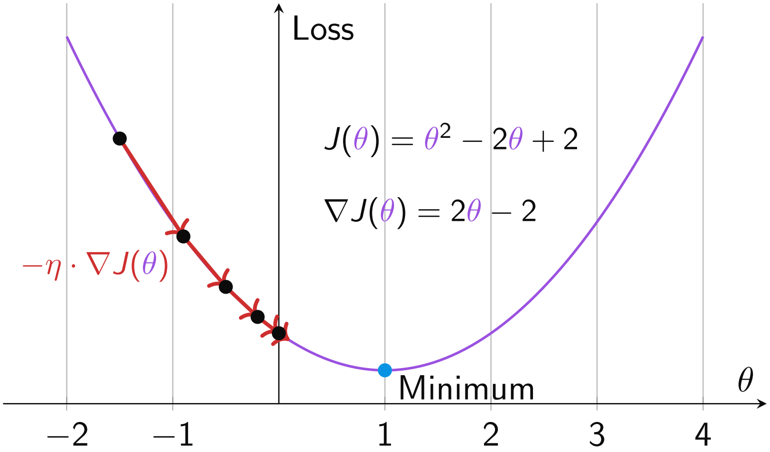


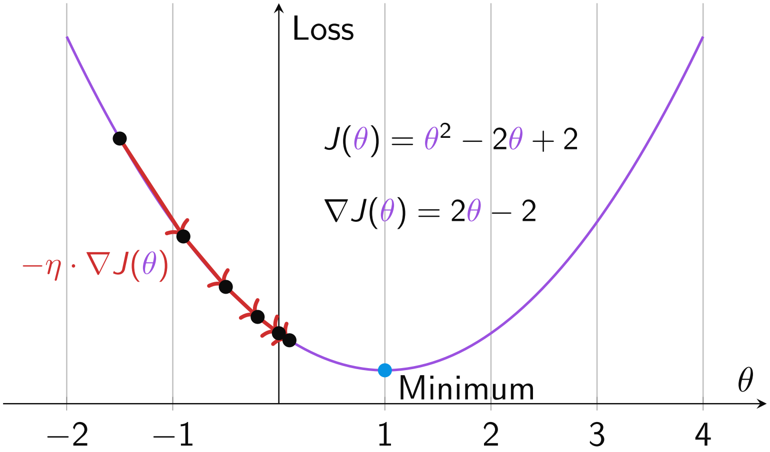
Loss
$J(\theta) = \theta^2 - 2\theta + 2$
$\nabla J(\theta) = 2\theta - 2$
$-\eta \cdot \nabla J(\theta)$
Minimum
θ
-2
-1
1
2
3
4

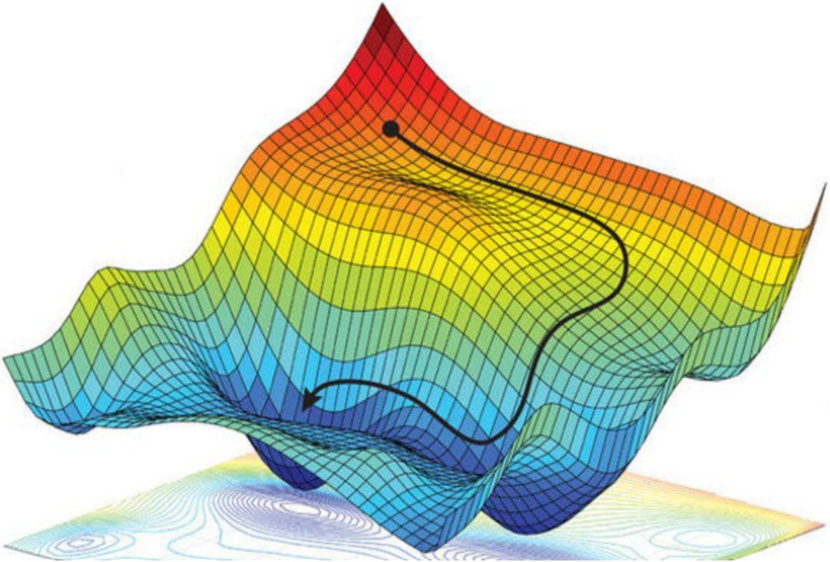
Gradient Decent



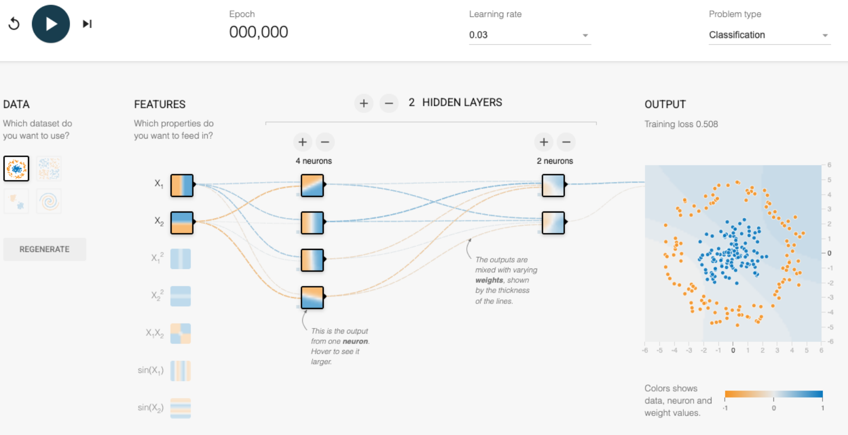
Gradient Decent




Loss
$J(\theta) = \theta^2 - 2\theta + 2$
$\nabla J(\theta) = 2\theta - 2$
$-\eta \cdot \nabla J(\theta)$
Minimum
θ
-2
-1
1
2
3
4



Design and Try Your Perceptron

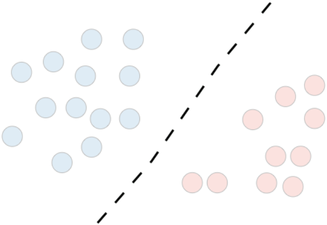


Simplified Tensorflow Playground

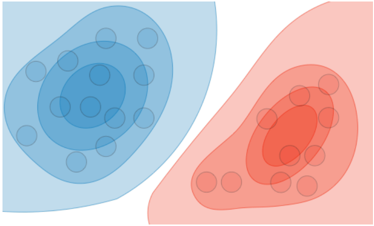
Extended Tensorflow Playground

Modeltypes

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



Discriminative modelling



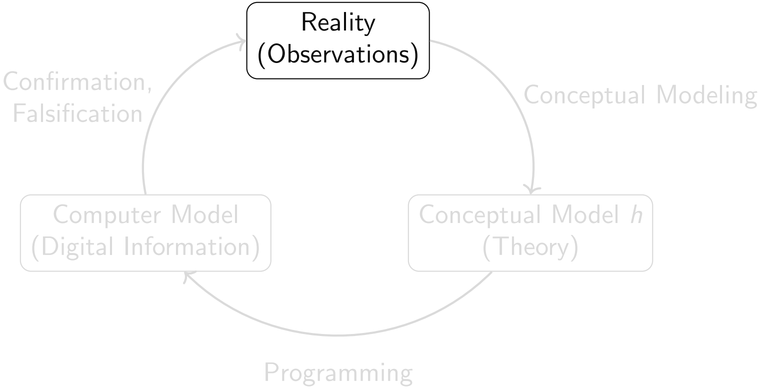
Generative modelling

1. How Do Machines Learn?

2. How Do Humans Train Machines?

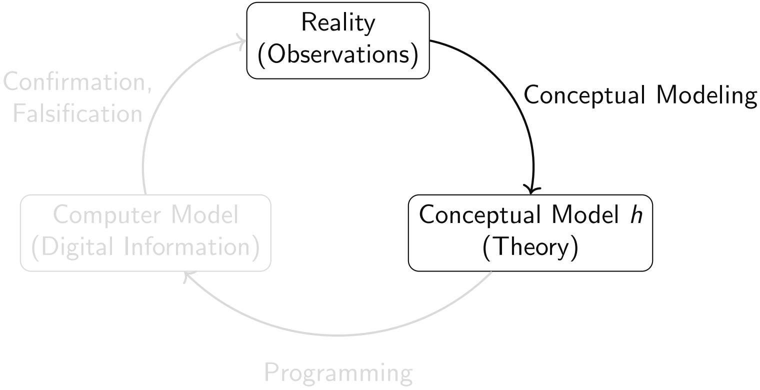
3. Interactions with ML

Theory-driven Modeling



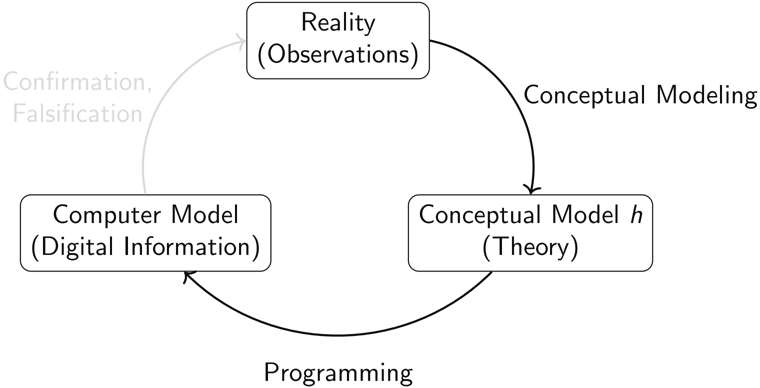
Minds contructs a (falsifiable) theory or hypothesis about reality to test against.

Theory-driven Modeling



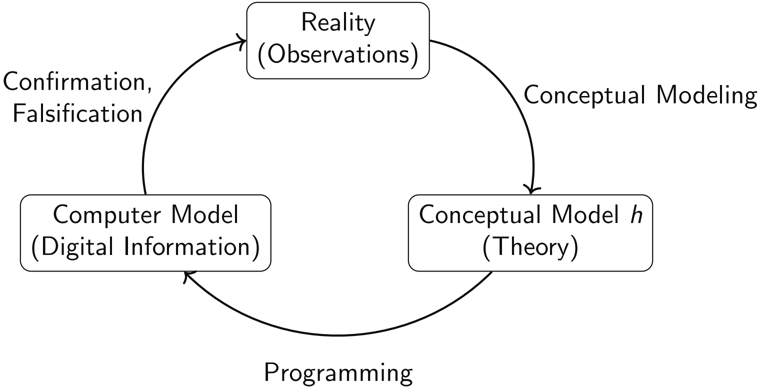
Minds contructs a (falsifiable) theory or hypothesis about reality to test against.

Theory-driven Modeling



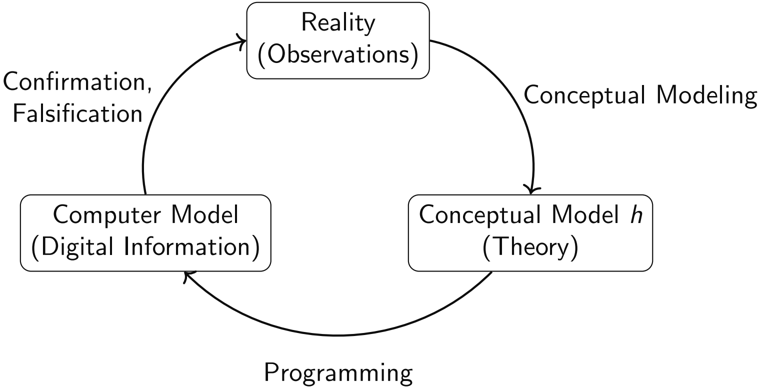
Minds contructs a (falsifiable) theory or hypothesis about reality to test against.

Theory-driven Modeling



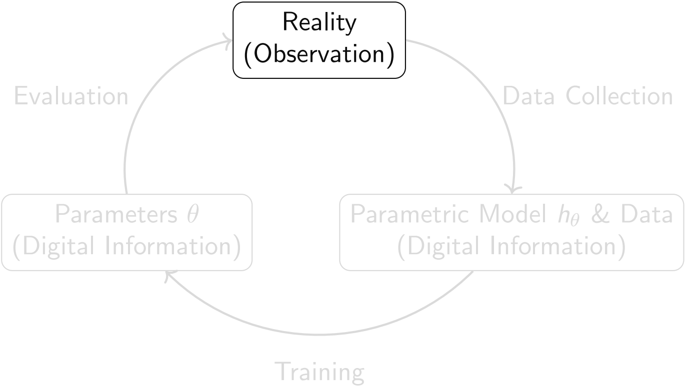
Minds contructs a (falsifiable) theory or hypothesis about reality to test against.

Theory-driven Modeling



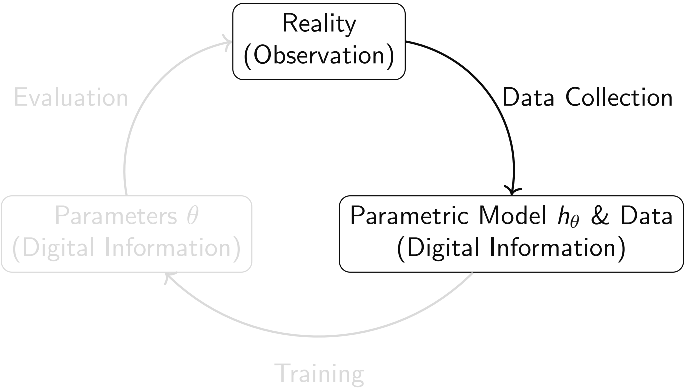
Minds contructs a (falsifiable) theory or hypothesis about reality to test against.

Data-driven Modeling



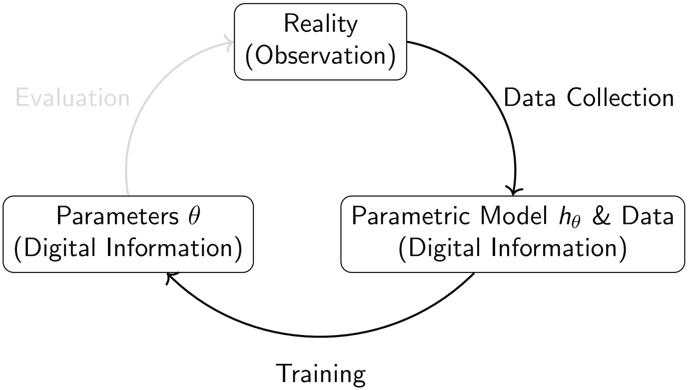
Algorithms (directly) fit a parametric model to the data. **Minds** are usually unable to conceptualize the trained model.

Data-driven Modeling

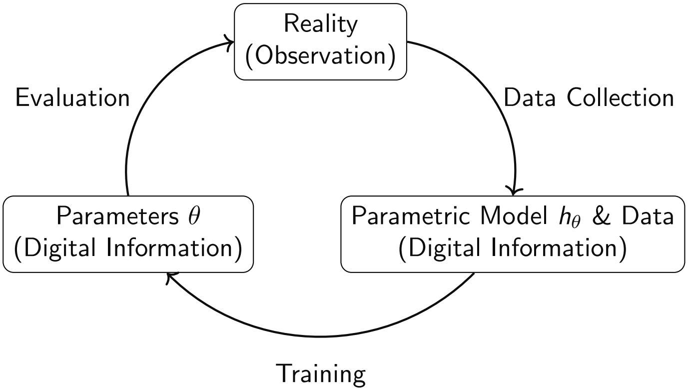


Algorithms (directly) fit a parametric model to the data. **Minds** are usually unable to conceptualize the trained model.

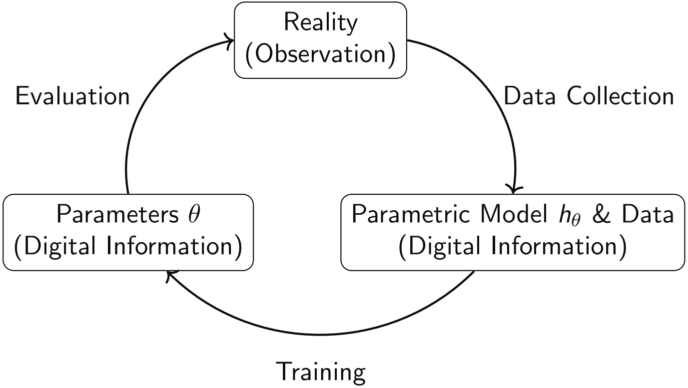
Data-driven Modeling



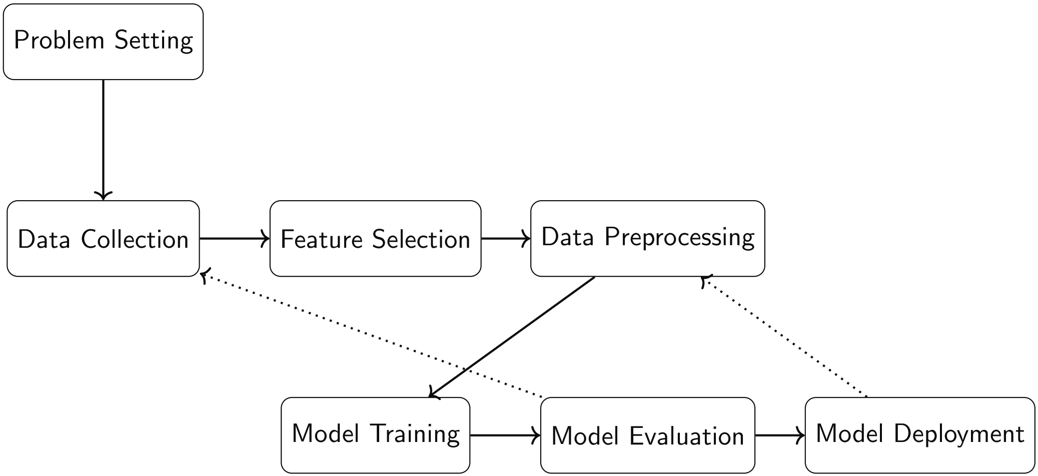
Algorithms (directly) fit a parametric model to the data. **Minds** are usually unable to conceptualize the trained model.



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.


Problem Setting
Data Collection
Feature Selection
Data Preprocessing
Model Training
Model Evaluation
Model Deployment

Programming Libraries

```python
class fast_qlinear(torch.autograd.Function):
    def forward(ctx, a, b, scales, zeros):

        m, k = a.shape
        _, n = b.shape

        quant_groupsize = 128
        block_size_m = 16
        block_size_n = 32 # [N = 4096 // 32] = 128 blocks
        block_size_k = 256
        group_size_m = 8
        num_warps = 4
        num_stages = 8
        total_blocks_m = triton.cdiv(m, block_size_m)
        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?

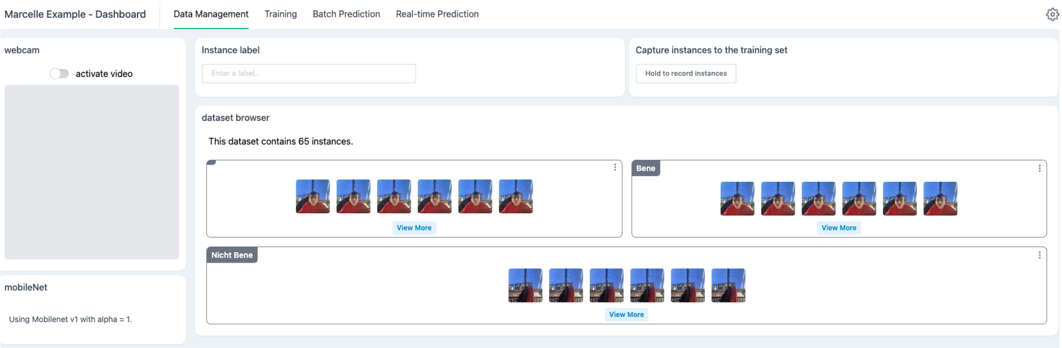
3. Interactions with ML

The Marcelle Toolkit

Marcelle: composing interactive machine learning workflows and interfaces (Françoise, Caramiaux, & Sanchez, 2021).

https://marcelle.dev/

The Marcelle Toolkit



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