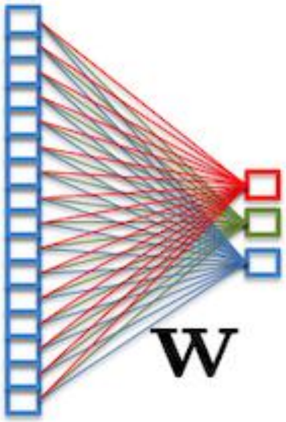
AI and Data Analysis

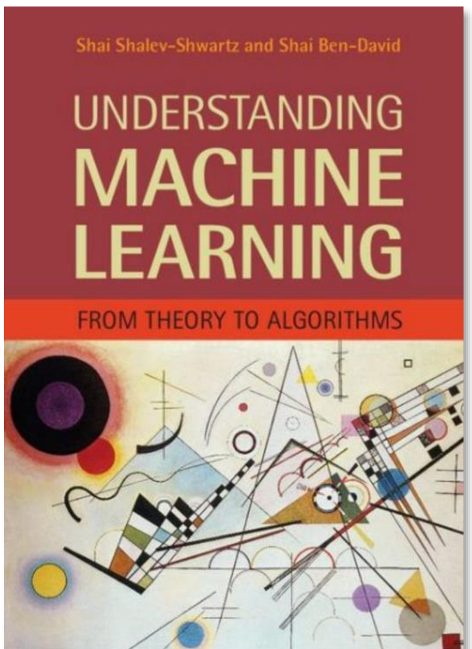
1.3 Some extremely short basics of machine learning



Christian Wolf

To go deeper ...

These next 15 (!!) slides will never be able to replace a full lecture in the theory of machine learning. The interested reader is referred to:



Shai Shalev-Shwartz and Shai Ben-David

Understanding Machine Learning, from Theory to Algorithms

Cambridge University Press, 2014

We would like to learn to predict a value y from observed input x

$$y = h(x, \theta)$$

Handcrafted from domain knowledge

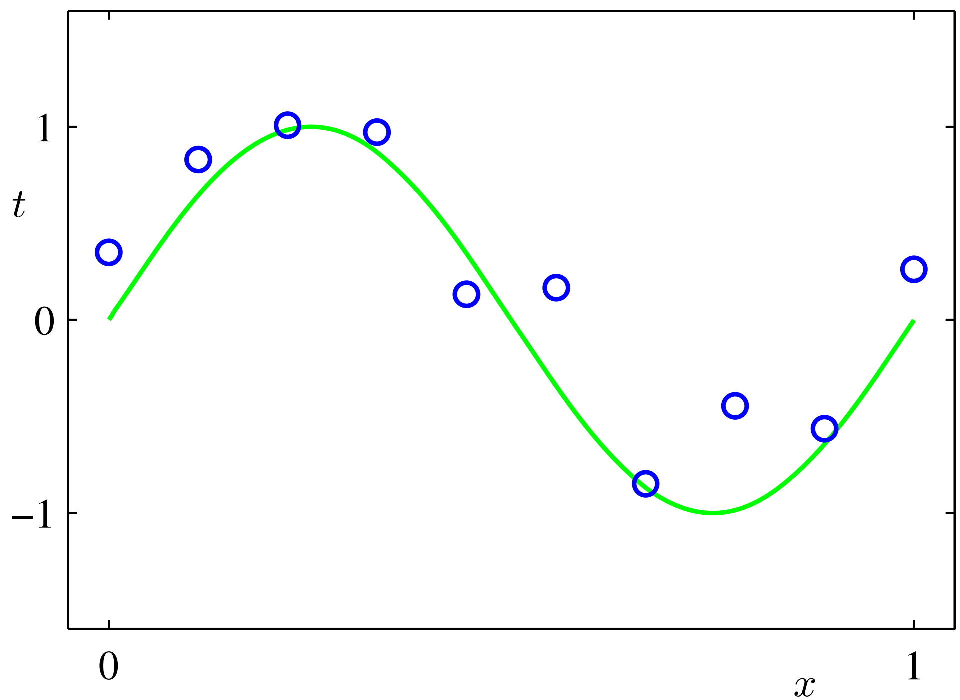
Learned from data or interactions

Fully handcrafted

Fully Learned

Fitting and Generalisation

- Data are generated with function $t = sin(2\pi x)$
- Noise has been added
- Objective: assuming the function unknown, predict t from x



[C. Bishop, Pattern recognition and Machine learning, 2006]

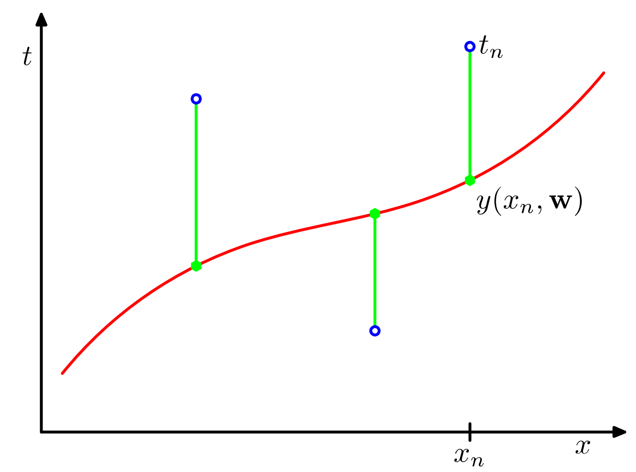
Fitting and Generalisation

Example: « Fitting » of a polynomial of order M

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

« Least squares » (of errors) criterion

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

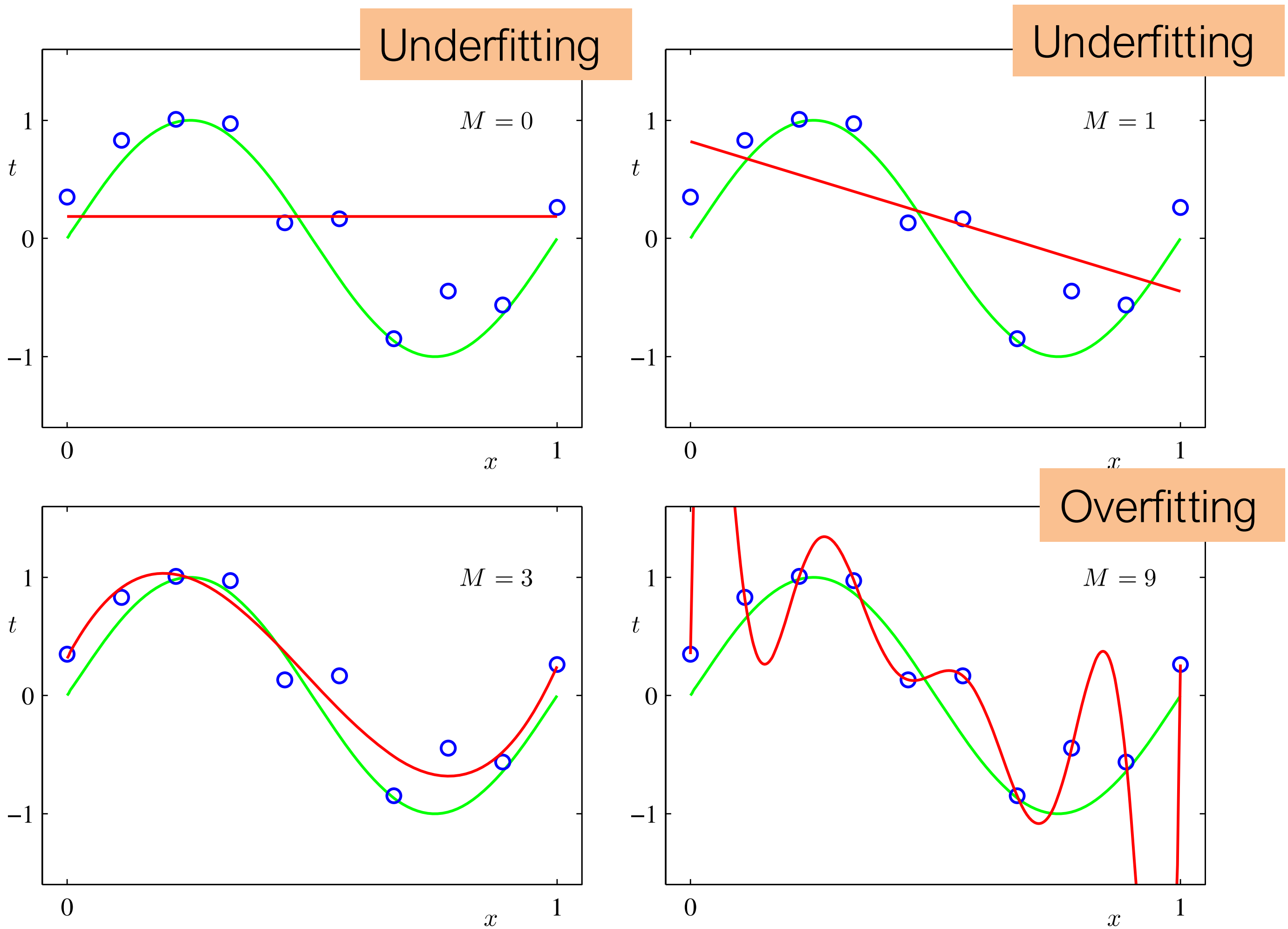


Linear derivative -> direct solution

[C. Bishop, Pattern recognition and Machine learning, 2006]

Model selection

Which order M for the polynomial?



[C. Bishop, Pattern recognition and Machine learning, 2006]

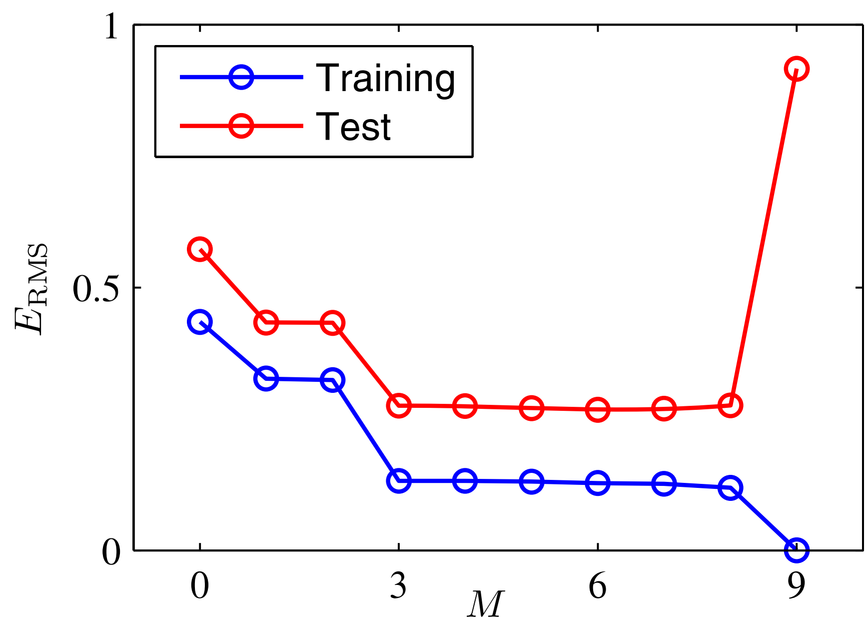
Model selection

Separation into (at least) two sets

- Training set
- Validation set (hold out set)

Root Mean Square Error (RMS)

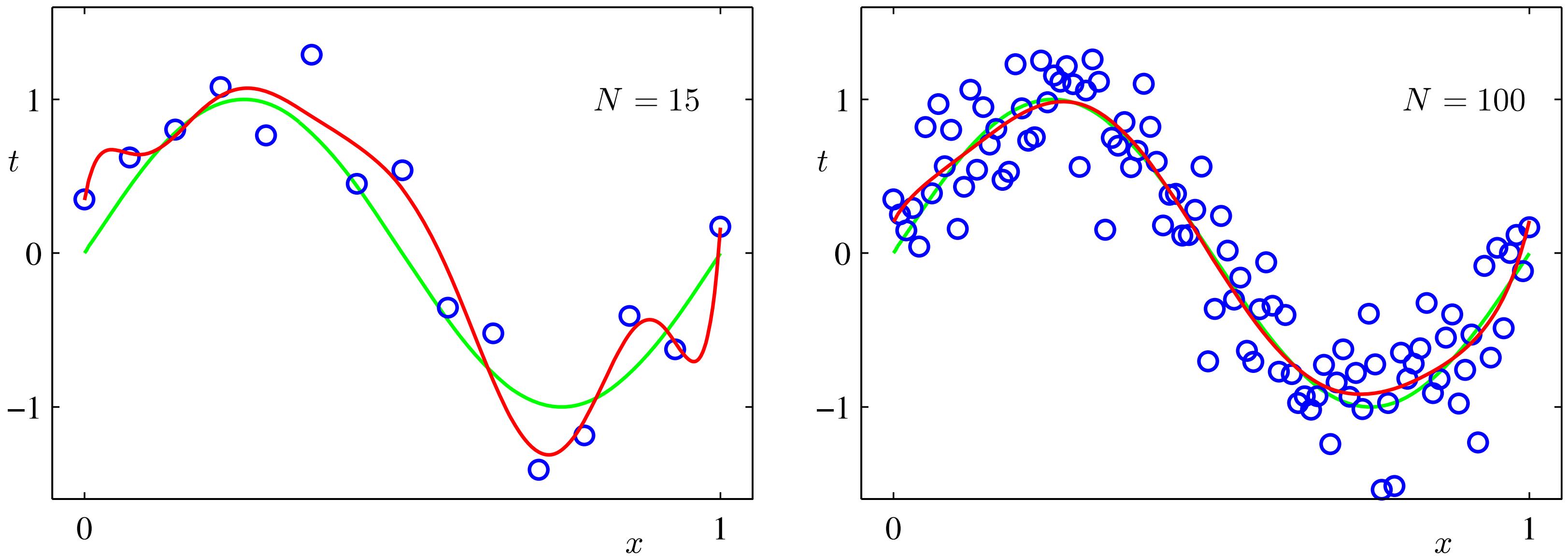
$$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^{\star})/N}$$



[C. Bishop, Pattern recognition and Machine learning, 2006]

Big Data!

Overfitting decreases if we increase the size of the training set.



M=9

[C. Bishop, Pattern recognition and Machine learning, 2006]

The 3 problems of Machine Learning

1. Expressivity
   – What is the complexity of the functions my model can represent?

2. Trainability
   – How easy is training of my model (i.e. solving the optimization problem)?

3. Generalization
   – How does my model behave on unseen data?
   – In presence of a shift in distributions?

(D’après Eric Jang & Jascha Sohl-Dickstein)

Car kept jamming on the brakes thinking this was a person 🤦‍♂️
The NN was dreaming @greentheonly

Learning formulations

Supervised learning — Labels y^* are available during training:

$$\hat{\theta} = \min_{\theta} \mathcal{L}\left(h(x,\theta), y^*\right)$$

Learning formulations

Supervised learning — Labels y^* are available during training:

$$\hat{\theta} = \min_{\theta} \mathcal{L}\left(h(x, \theta), y^*\right)$$

Unsupervised learning — no labels, discovery of regularities in the data. Different objectives are possible.

Learning formulations

Supervised learning — Labels y^* are available during training:

$$\hat{\theta} = \min_{\theta} \mathcal{L}\left(h(x, \theta), y^*\right)$$

Unsupervised learning — no labels, discovery of regularities in the data. Different objectives are possible.

Self-supervised learning — prediction of masked parts of the data itself, for instance the future:

$$\hat{\theta} = \min_{\theta} \mathcal{L}\left(h(x_{t-\Delta:t-1}, \theta), x_t\right)$$

$\Rightarrow$ Pretraining step, usually followed by task oriented training.

Learning formulations

Supervised learning — Labels y^* are available during training:

$$\hat{\theta} = \min_{\theta} \mathcal{L}\left(h(x, \theta), y^*\right)$$

Unsupervised learning — no labels, discovery of regularities in the data. Different objectives are possible.

Self-supervised learning — prediction of masked parts of the data itself, for instance the future:

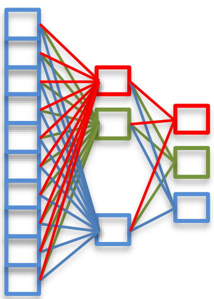
$$\hat{\theta} = \min_{\theta} \mathcal{L}\left(h(x_{t-\Delta:t-1}, \theta), x_t\right)$$

$\Rightarrow$ Pretraining step, usually followed by task oriented training.

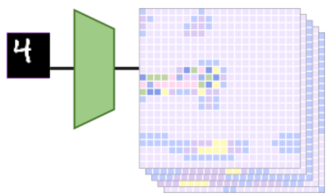
Reinforcement learning — learning from interactions, maximizing the cummulated reward R over a horizon:

$$\hat{\theta} = J\left(\pi_\theta\right) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta}\left[R(\tau)\right]$$

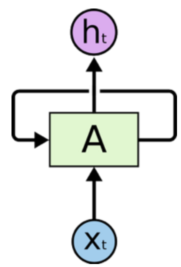
The Deep Toolbox



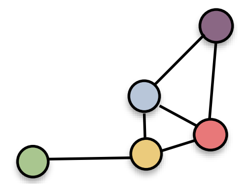
MLP



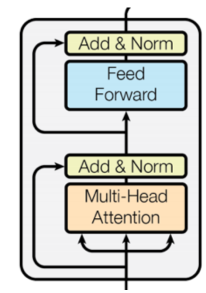
CNN /
Convolutions



RNN /
Recurrence



GN, GCN /
Graphs, geometry



Transformers /
Self-attention

What do I know about the data and the task?

Nothing (vector space)

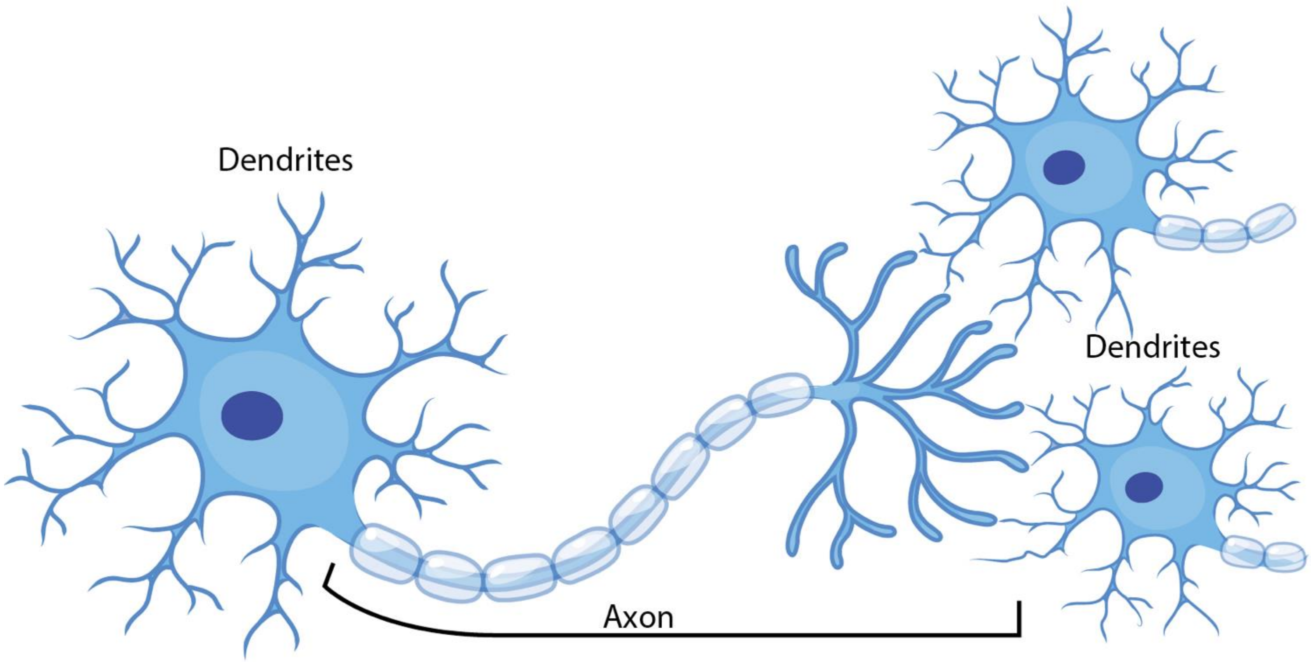
Translation equivariance

Sequential data, Markov property

Graph structured data

Permutation equivariance

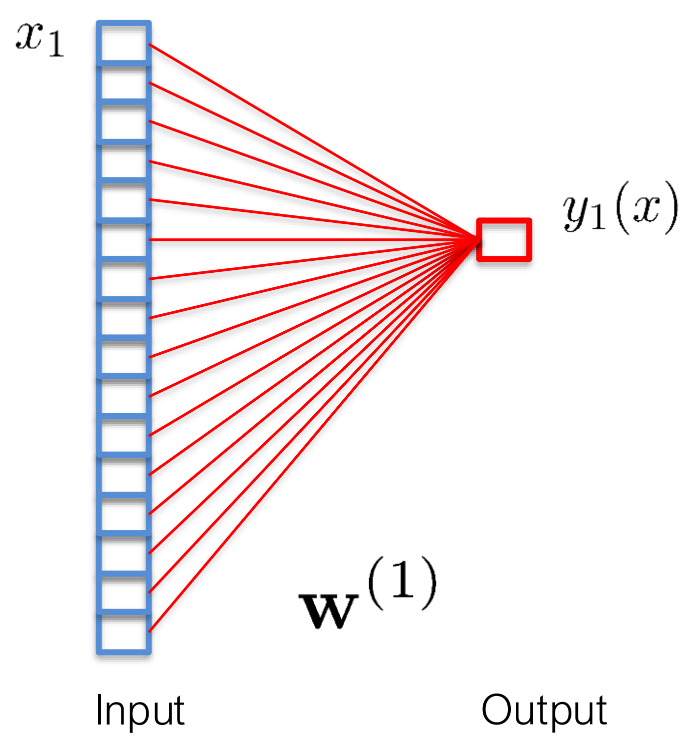
Biological neurons



Devin K. Phillips

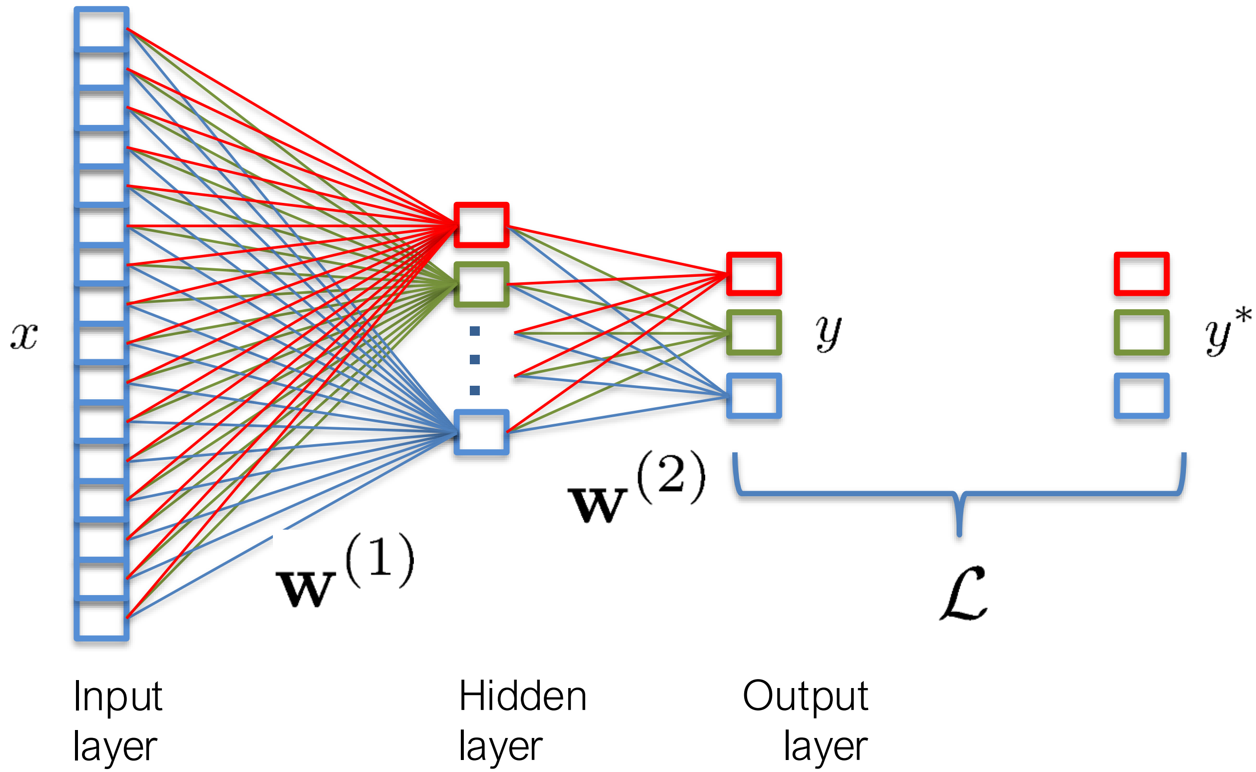
Neural networks

« Perceptron »

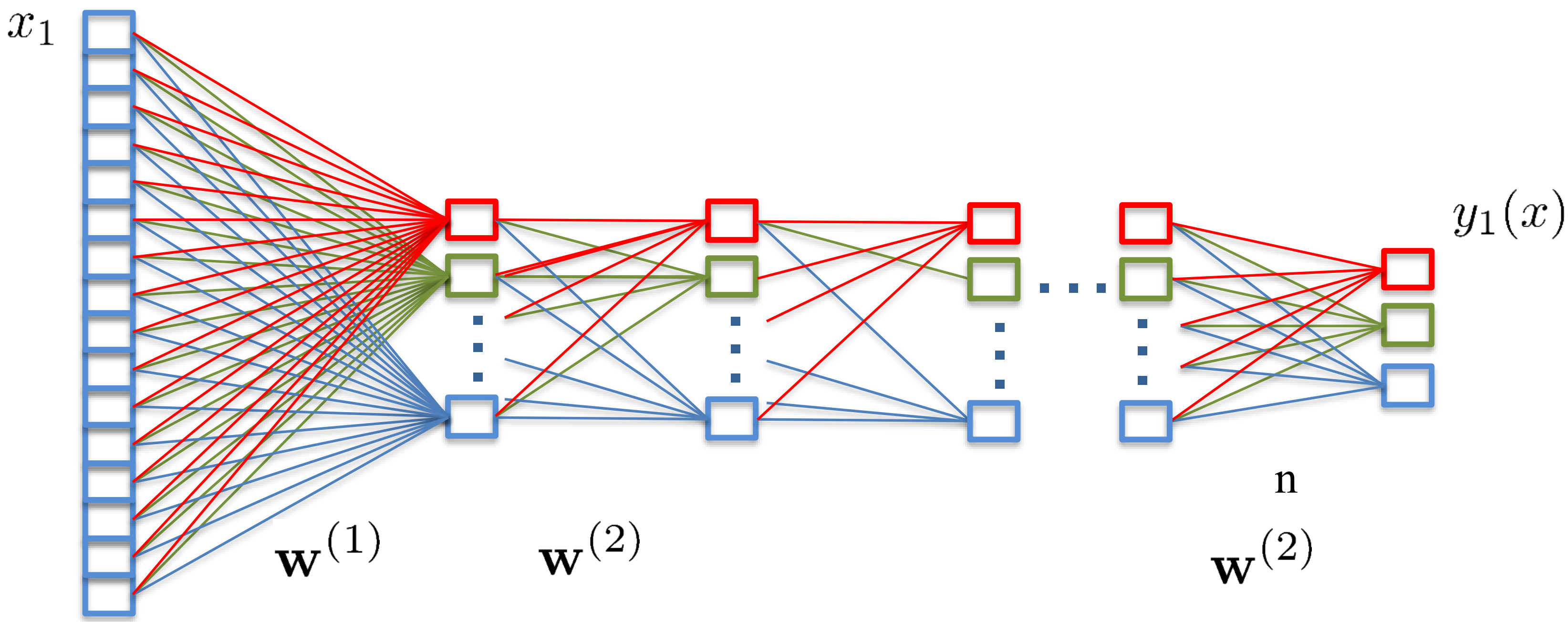


$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=0}^{D} \boldsymbol{w}_i \boldsymbol{x}_i$$

Deep neural networks



Deep neural networks



Gradient descent

One optimizer step:

$$\theta^{[t+1]} = \theta^{[t]} + \nu \nabla \mathcal{L}\left(h(x,\theta), y^*\right)$$

The gradient is a vector of partial derivatives:

$$\nabla \mathcal{L} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \theta_0} \\ \\ \frac{\partial \mathcal{L}}{\partial \theta_1} \\ \\ \vdots \\ \\ \frac{\partial \mathcal{L}}{\partial \theta_N} \end{bmatrix}$$

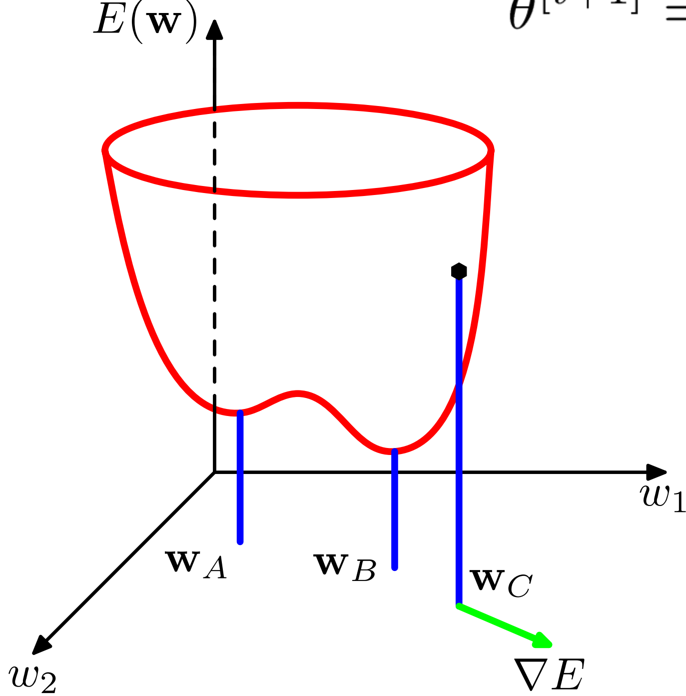
Gradient descent

Minimize the error on known data

"Empirical Risk Minimization"

$$\theta^{[t+1]} = \theta^{[t]} + \boxed{\nu}\nabla\mathcal{L}(h(x,\theta), y^*)$$

Learning rate

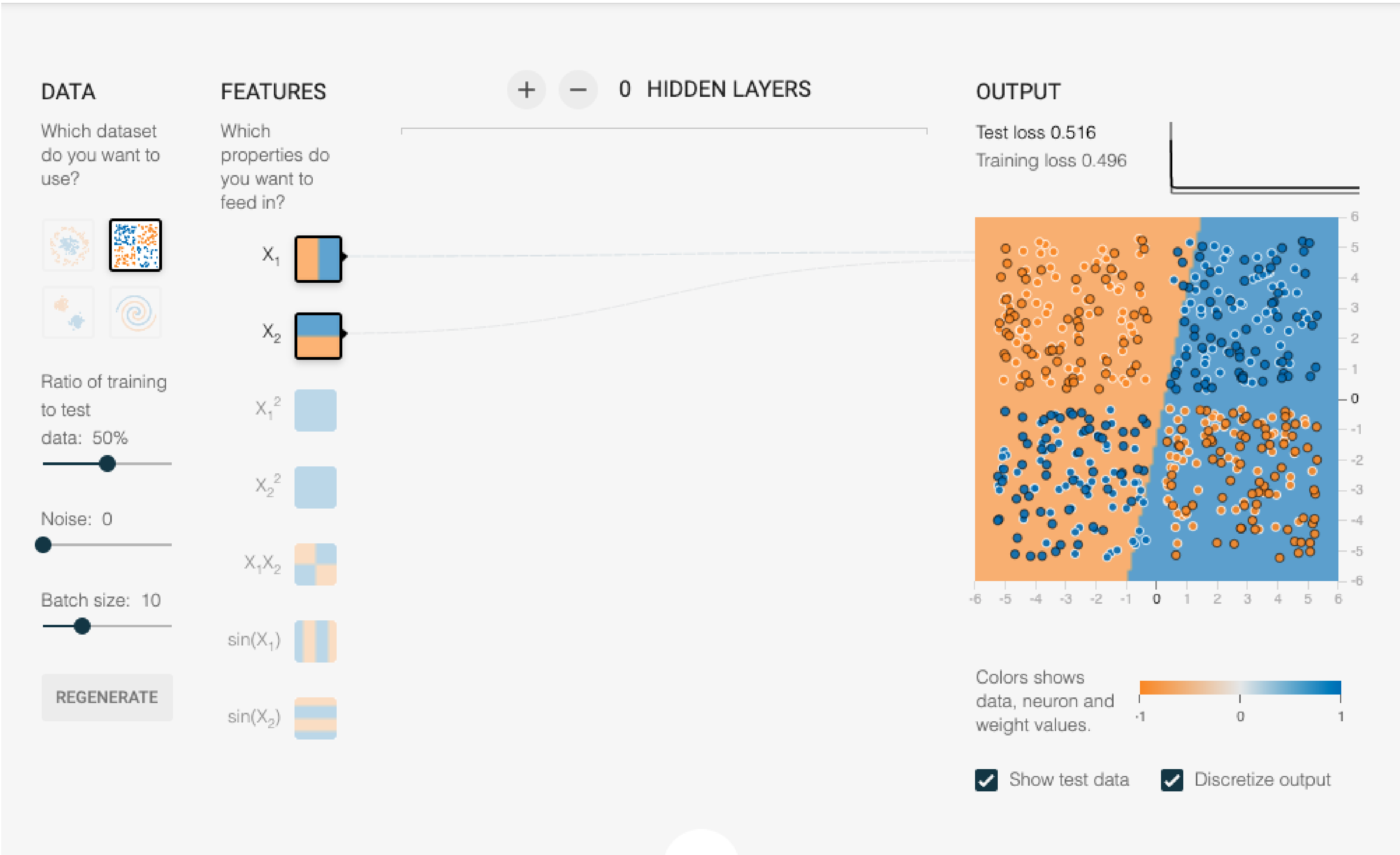


Can be blocked in a local minimum

[C. Bishop, Pattern recognition and Machine learning, 2006]

Demo session:
Tensorflow playground

Tensorflow Playground



https://playground.tensorflow.org