

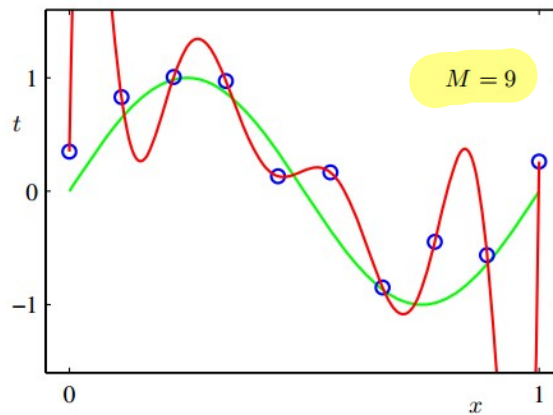
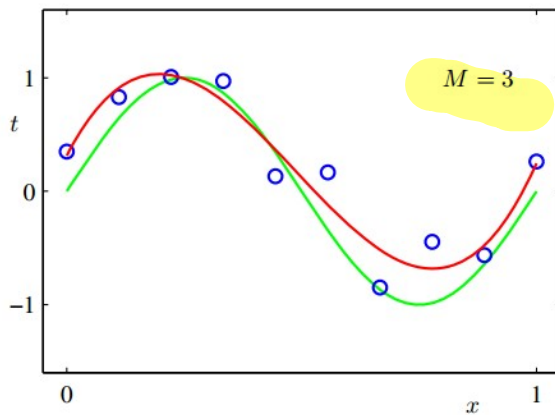
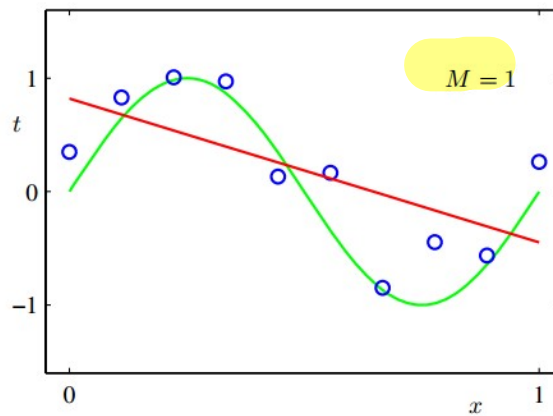
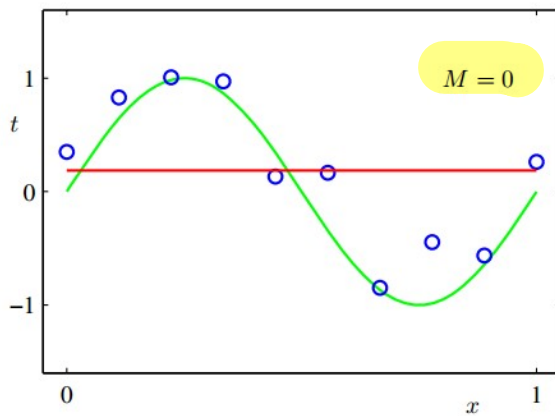
Lecture 4 – Part I

Regularization

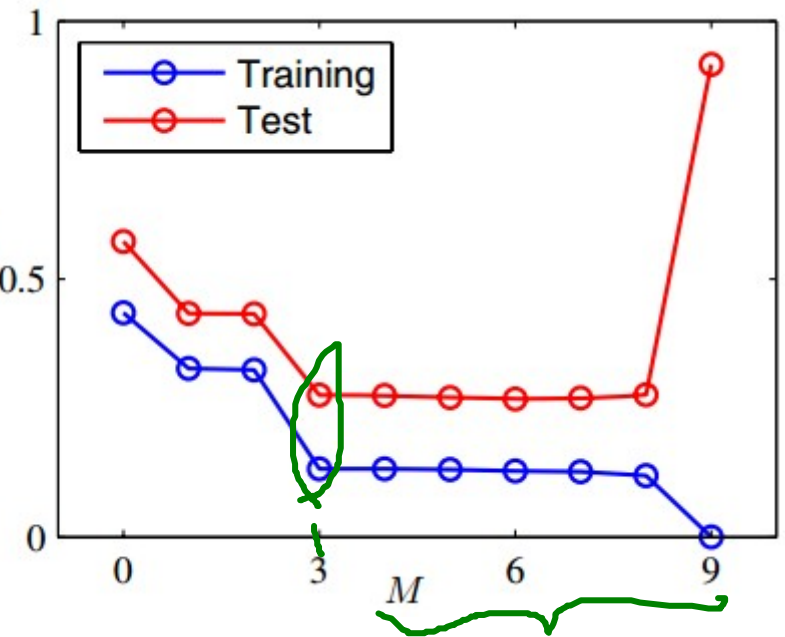
- Motivation, definition
- Observation: Large weights and overfitting
- Regularization: closed form in linear regression + intuitions
- Does it work ? A few examples
- The Bayesian interpretation
- Regul during GD: Parameter shrinkage, weight decay
- Lasso

Complexity controlled **explicitly** (rare case)

M = polynomial order



E_{RMS}



Bishop, 2006

Regularization (general definition)

- A possible def: “Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.” From *Deep Learning*, by Ian Goodfellow and Yoshua Bengio and Aaron Courville

<https://www.deeplearningbook.org/>

- Goal: Regularization allows to **restrain a model's complexity, quantitatively**, without explicitly limiting the model (i.e. order of polynomial fitting, etc)

Not explicitly modify H , but do restrain the actual visited part of H (H =hypothesis space)

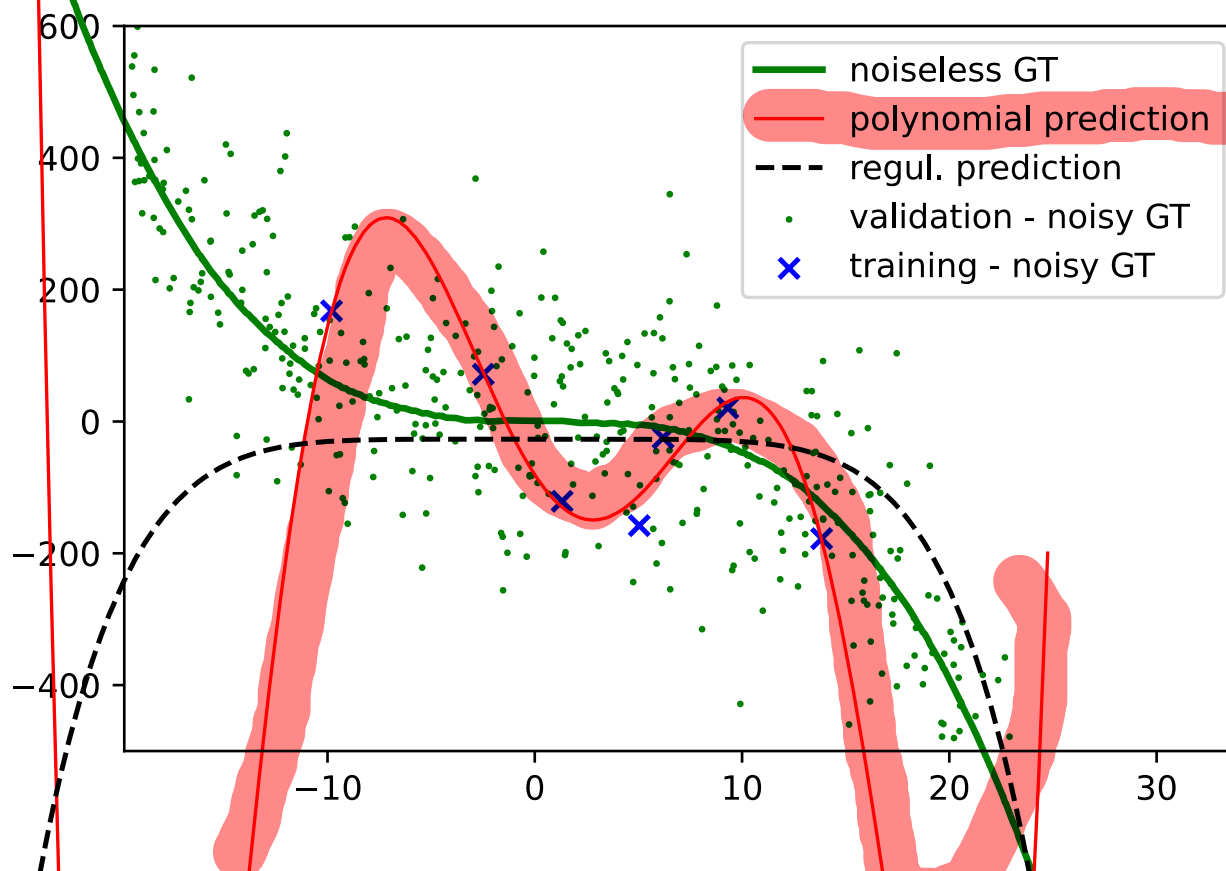
- Examples:
 - Lasso, Ridge, Elastic-Net
 - Dropout (see DeepNetworks)
 - feature selection procedures
 - ensemble methods

PCA as pre-processing
early stopping

- Here we focus on classics, i.e. **Ridge** and **Lasso**

Empirical Observation

Large weights \approx overfitting



- No regularization : bad score, typically high weights (esp. coeffs of large order are too high)
- cf `lecture4-unregularized regression has large coefficients.ipynb` .

Intuition: Large weights \approx overfitting

(it's actually more complicated)

$$x \rightarrow x + \delta x, \quad y \rightarrow y + \underbrace{W \cdot \delta x}_{O(1) \text{ or } \Delta}$$

- **Large weights** : output $W \cdot x$ is **very sensitive** to small changes in data x . So, small perturbation of training data \rightarrow big changes in weights \rightarrow big changes in output (\rightarrow overfitting)

- **Small weights** : output $W \cdot x$ is less sensitive, i.e. is **more robust** w.r.t. change in data : not so different output for slightly different data \rightarrow less overfitting (=better generalization)

$$y \xrightarrow{x \rightarrow x + \delta x} y + W \cdot \delta x \approx y \pm o(1)$$

- Remark: actually, the value of weights itself is meaningless. But, that's the spirit.

$$x \rightarrow a_0 + x a_1 + x^2 a_2 + \dots + x^p a_p \quad x \in \mathbb{R}$$

Adding a Regularization term

There are two standard regularization terms. For a ML problem with a given Loss L :

- **Ridge** regul.: $+ \lambda \|\vec{w}\|_2^2$ (L2-Regul)

$$L_{\text{Ridge}} = L + \lambda \underbrace{\|\vec{w}\|_2^2}_{\sum_{d=1}^D |w_d|^2}$$

- **Lasso** regul.: (L1 regul)

$$L_{\text{Lasso}} = L + \lambda \|\vec{w}\|_1$$

$$\|\vec{w}\|_1 = \sum_{d=1}^D |w_d|$$

- Elastic-Net: a mix of them both:

$$L_{\text{Elastic}} = L + \alpha_1 \|\vec{w}\|_1 + \alpha_2 \|\vec{w}\|_2^2$$

Linear Regression: **One-shot solution**

- (without regularization)

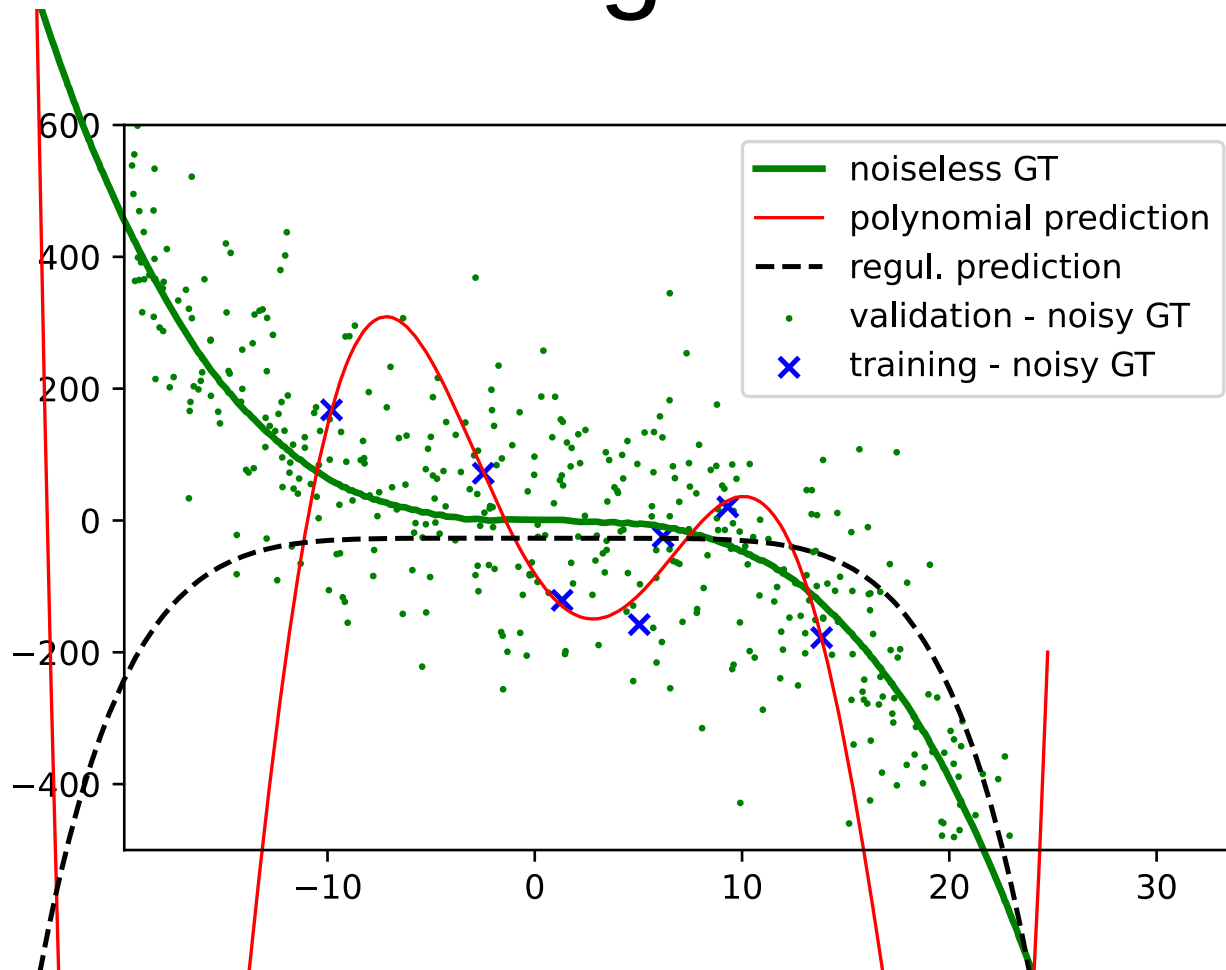
Linear Regression: **One-shot solution**

- (**with** regularization)

Linear Regression: **One-shot solution**

- (**with** regularization, in $D=1$ – even more intuitive)

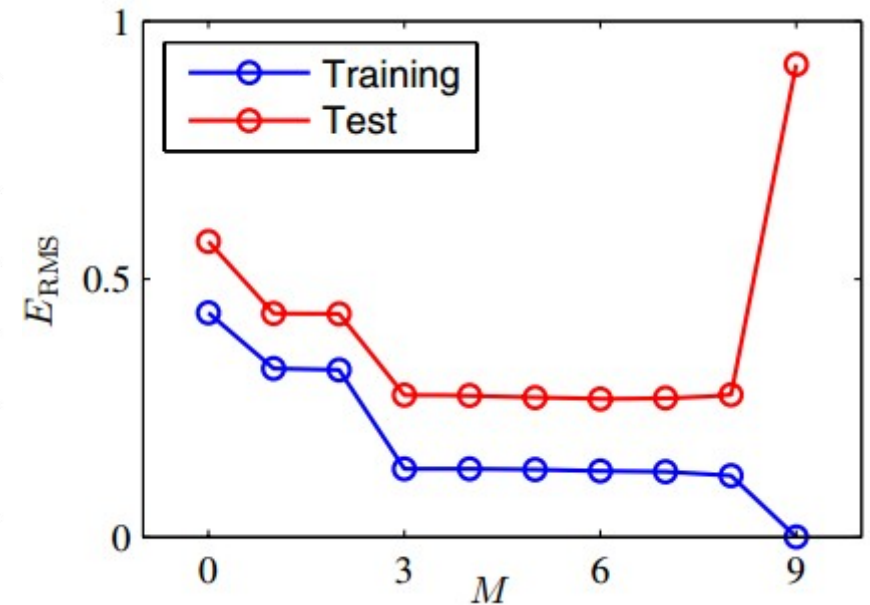
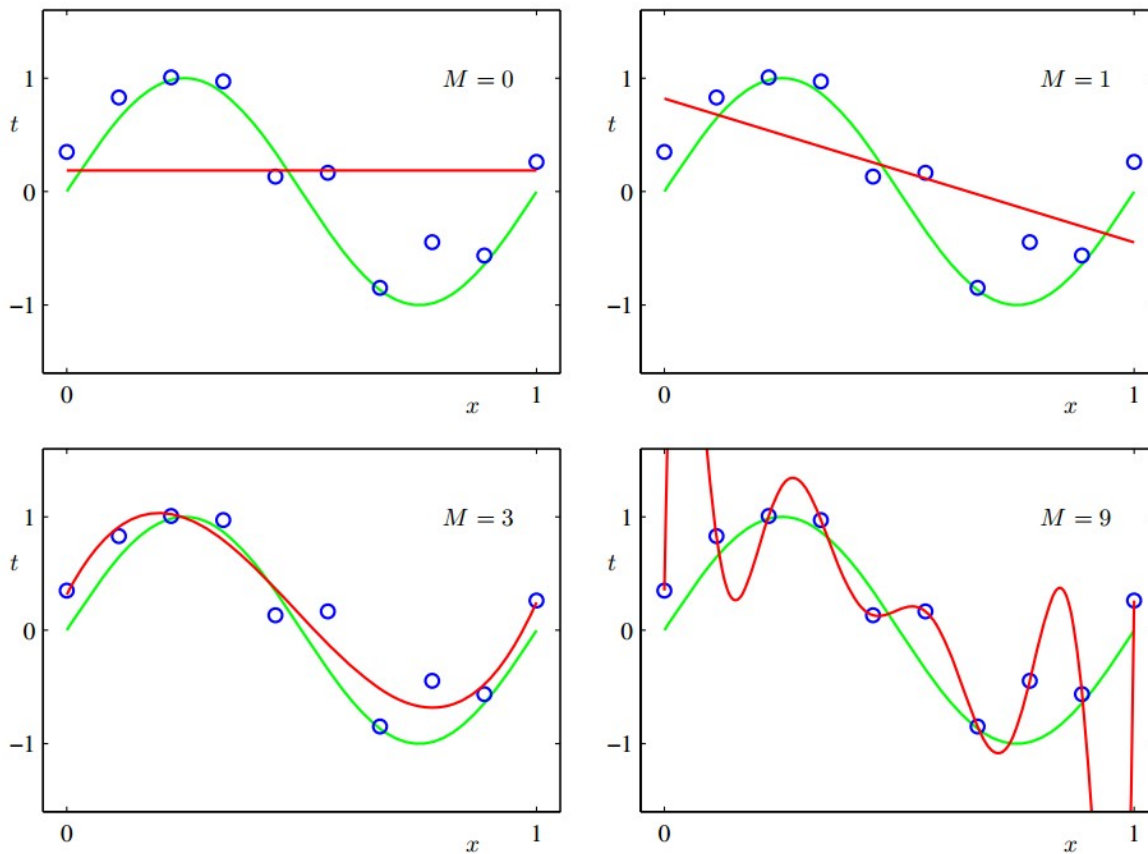
Does regul. work ?



- No regularization : bad score, typically high weights (esp. coeffs of large order are too high)
- **With regularization: better score, all coeffs. shrink a lot (towards 0)**

Complexity controlled **explicitly** (rare case)

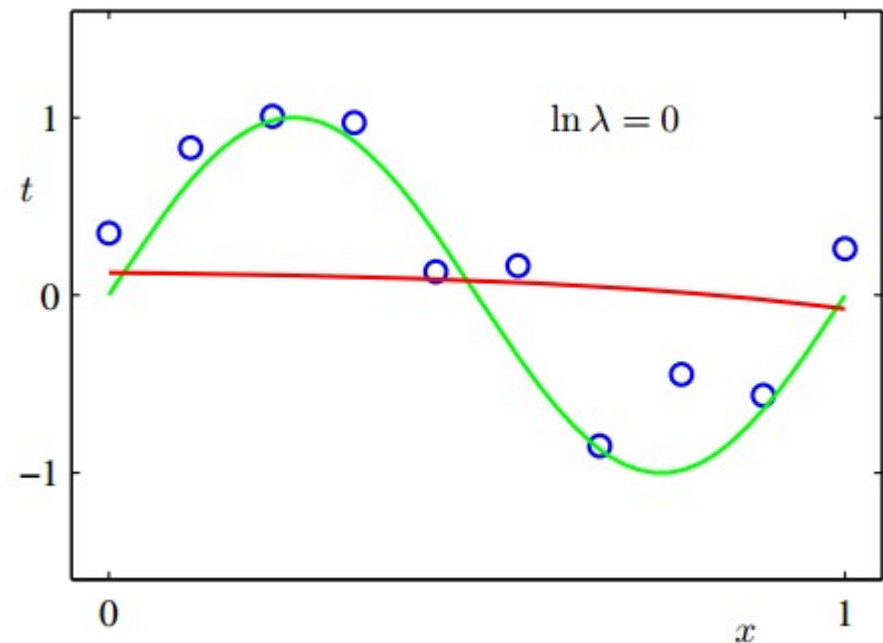
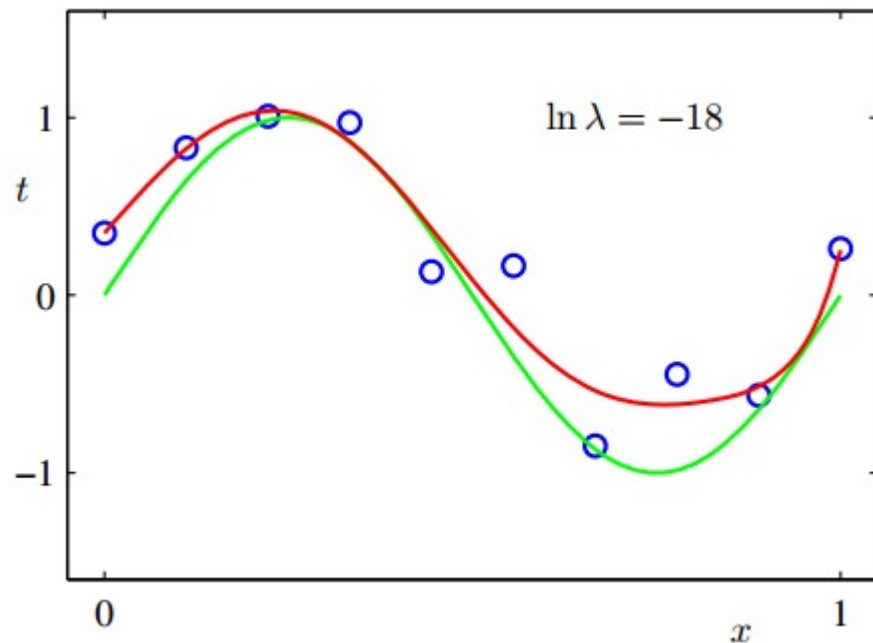
M = polynomial order



Bishop, 2006

Complexity controlled indirectly : regularization

$M =$ polynomial order: still $M=9$
But, with **regularization parameter λ** changed



Important: practice with the **tutorial** to do more tests / play with:
`lecture4-regularization-dependence on lambda.ipynb`

Bayesian computations

The basics (prerequisite)

- MLE + an *a priori* opinion on what things should be = Maximum A Posteriori = MAP
- i.e. estimate a random variable, with the opinion (a priori) that its mean is of order τ :

Regularization

Bayesian interpretation

- MLE + an *a priori* opinion on what the model is = MAP \rightarrow we can get the L2 regul from that !
- Assume model's weights follow a Gaussian distribution

Regularization during GD:

Parameter shrinkage, weight decay

- What does regularization do **during a GD** ?

Lasso Regularization

- If we use the L1 norm: (or L0 norm)
- Effect: tends to set some weight to 0 exactly
→ it's already feature selection !

References

- **Algebra** reminder: *Bishop*, appendix C, p. 695-701 (only 6 pages !!)
- **Regularization**: *Bishop*, sec. 3.1.4, p. 144-146
See also Sec. 5.5, p. 256-271, for much much more (Neural Nets).
- Another good **book**: (more recent, 2016): *Deep Learning*, by Ian Goodfellow and Yoshua Bengio and Aaron Courville
<https://www.deeplearningbook.org/> , in particular the chapter 5,
<https://www.deeplearningbook.org/contents/ml.html>
-