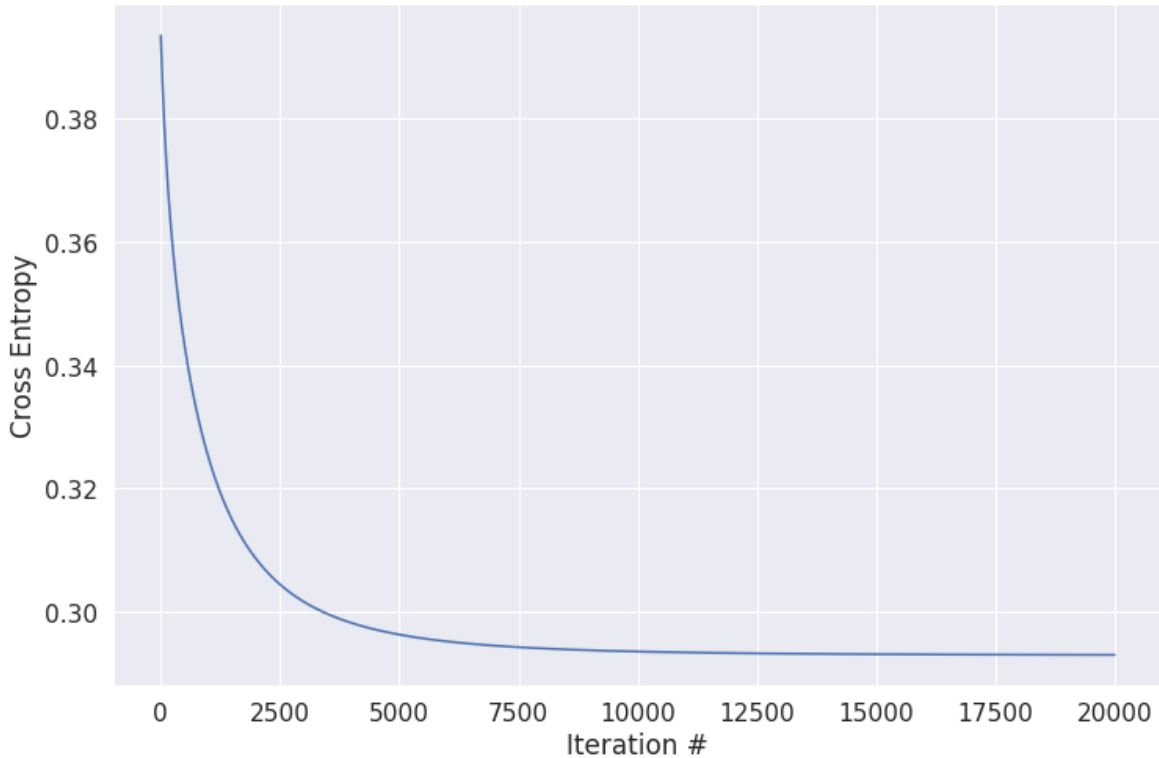
Overfitting

During training, we optimize a cost function J with respect to the training data

The cost computed on the training data is denoted E_{in} (in-sample error) by prof. Mostafa

Training loss

E_{in} (loss / cost) usually decreases along the iteration (for instance, when we are employing *gradient descent*)

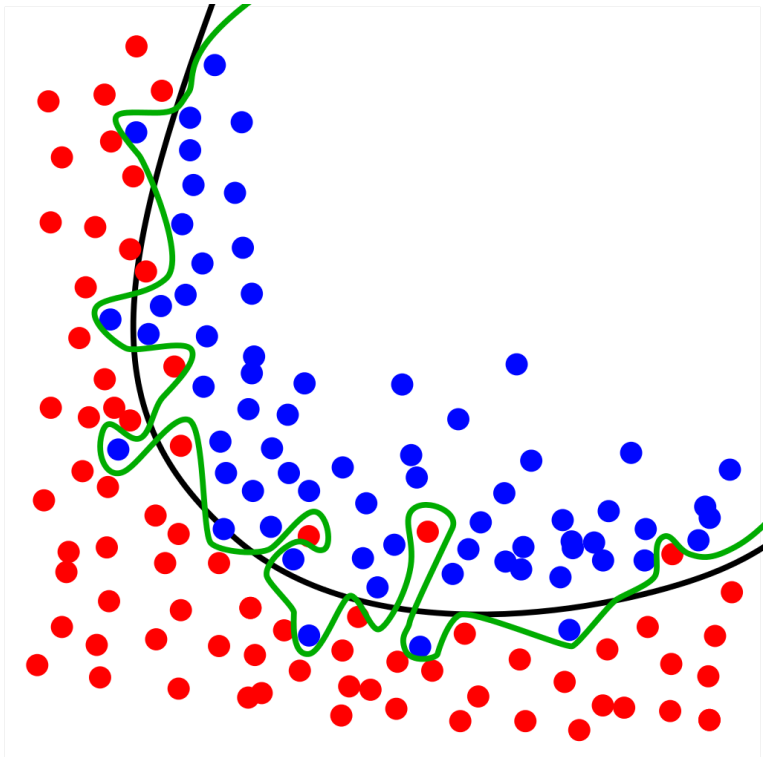


How eagerly should we try to optimize E_{in} ?

Is bringing E_{in} down, as closely as possible to 0, always a good thing ?

Overtraining

Overtraining may result in **overfitting**

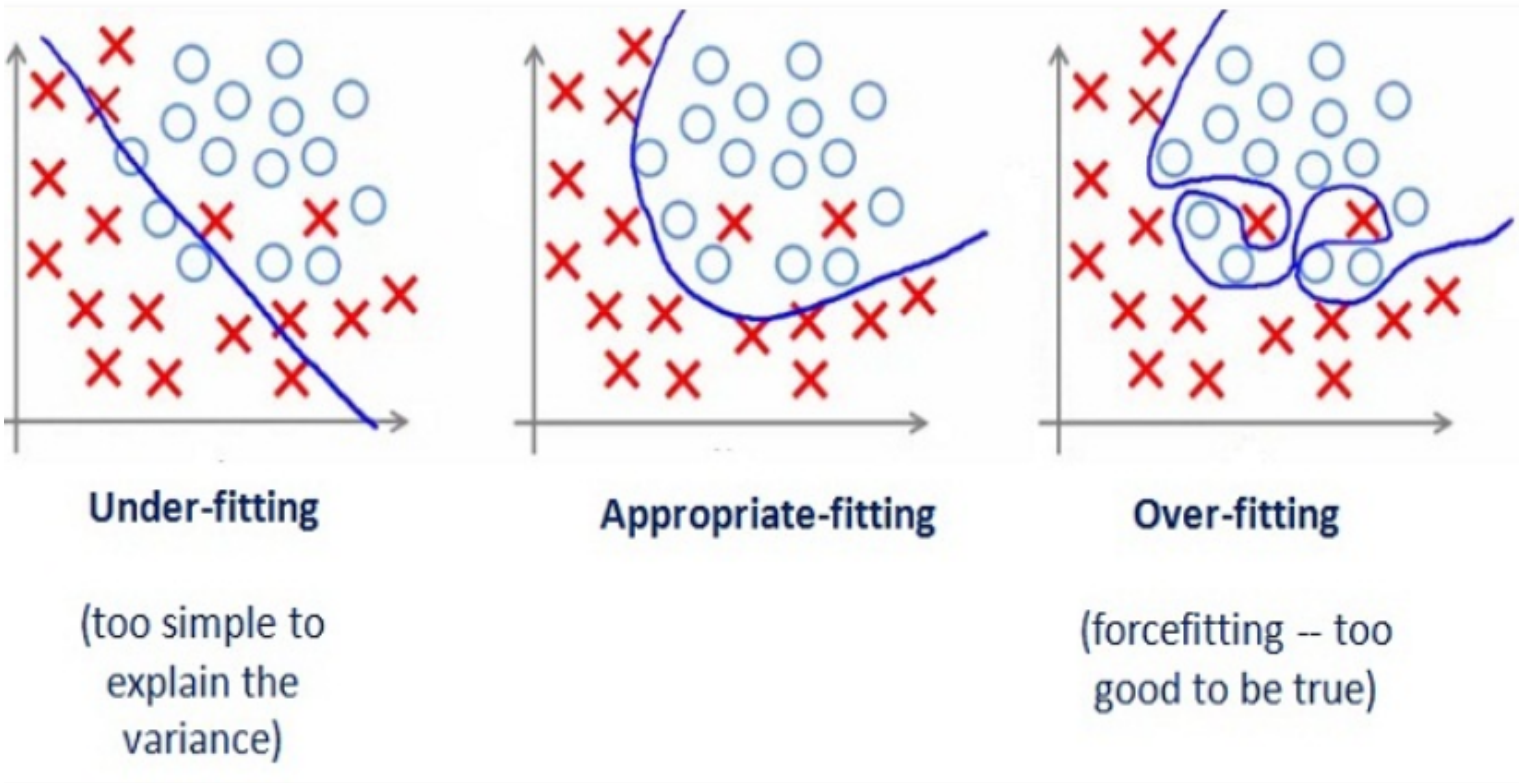


green illustrates overfitting

Fonte: Wikipedia

Overfitting / Underfitting

It is not just about number of iterations. It is also related to model complexity



The error that really matters is E_{out} (the error computed over the entire domain) – *out-of-sample error*

Generalization: We minimize E_{in} hoping to also minimize E_{out} (ou-of-sample error). We would like to have E_{out} as close as possible to E_{in}

In general, the following equality holds:

$$E_{out} = E_{in} + \text{generalization\_error}$$

Since in practice we can not compute E_{out}, we can use an independent set of examples (validation set) and compute the cost on it, E_{val}

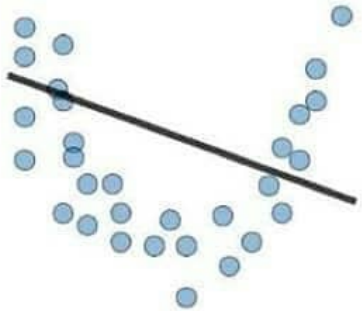
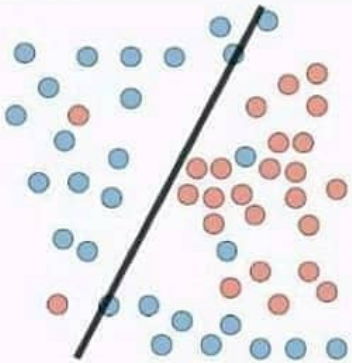
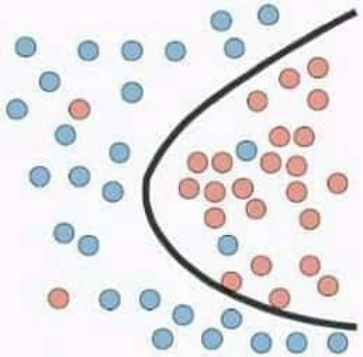
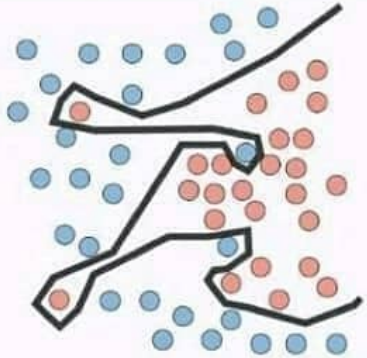
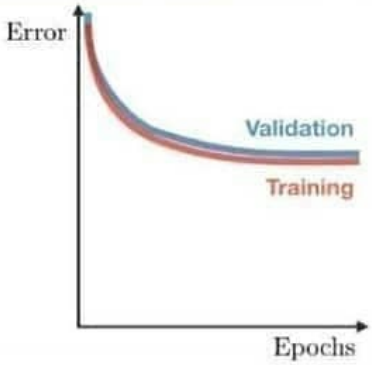
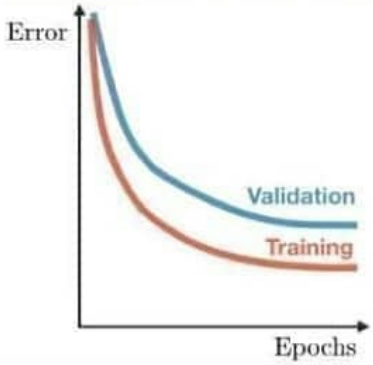
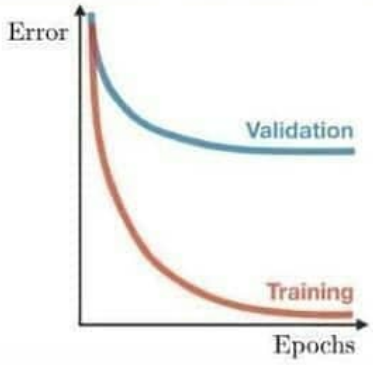
E_{val} can be thought as a proxy of E_{out}

Validation error and overfitting

Underfitting: large E_{in} and E_{val} indicate strong model *bias* (model is too constrained)

Overfitting: when the curves of E_{in} and E_{val} start to get separated each other along the iterations, it is an indication of overfitting (model is too sophisticated)

Va

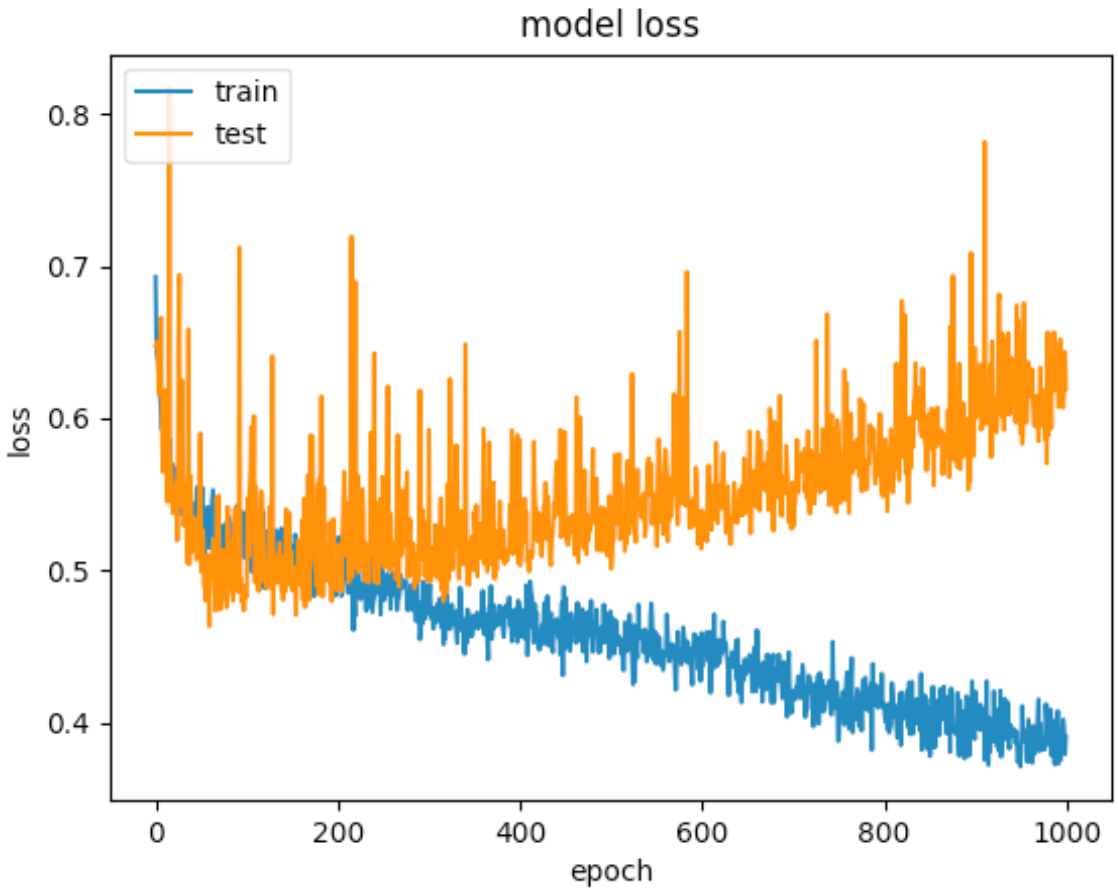
	Underfitting	Just right	Overfitting
Symptoms	• High training error • Training error close to test error • High bias	• Training error slightly lower than test error	• Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
ssible ne fra	• Complexify model • Add more features • Train longer		• Perform regularization • Get more data

model

arated

ng

Learning curve



Learning curve in many practical situations

(test = error on validation set)

How to deal with overfitting (prof. Mostafa's view):

- **regularization** – add a penalty term (contra-peso) in the cost function (to be seen later)
- **validation** – error on the validation set, E_{val}, can be used to choose a family of hypotheses $\mathcal{H}$ of "right complexity"

Validation versus regularization

In one form or another, $E_{\text{out}}(h) = E_{\text{in}}(h) + \text{overfit penalty}$

Regularization:

$$E_{\text{out}}(h) = E_{\text{in}}(h) + \underbrace{\text{overfit penalty}}_{\text{regularization estimates this quantity}}$$

Validation:

$$\underbrace{E_{\text{out}}(h)}_{\text{validation estimates this quantity}} = E_{\text{in}}(h) + \text{overfit penalty}$$

Overfitting – Lecture 11 Mostafa

Validation – Lecture 13 Mostafa

For now, we will consider E_{val} as a means to evaluate the performance of the models

Later we will look at it from the perspective of generalization error.