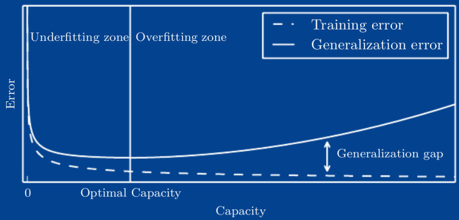
AARHUS UNIVERSITY

LESSON 08: Model-capacity, Under/over-fitting, Generalization

CARSTEN EIE FRIGAARD

AUTUMN 2024



"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E." — Mitchell (1997).

L08: Model-capacity, Under/over-fitting, Generalization

Agenda

- Resumé af GD og NN's.
- Model Capacity,
- Under/over-fitting,
- Generalization Error,
- Exercise: `L08/learning_curves.ipynb`,

  that replaces the old excercises:

  ~~Exercise: `L08/capacity_under_overfitting.ipynb`~~

  ~~Exercise: `L08/generalization_error.ipynb`~~

RESUMÉ: GD

The numerically Gradient decent [GD] method is based on the gradient vector

$$\nabla_{\mathbf{w}} J(\mathbf{w})$$

for the gradient oprator

$$\nabla_{\mathbf{w}} = \left[\frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \ldots, \frac{\partial}{\partial w_m} \right]^\top$$

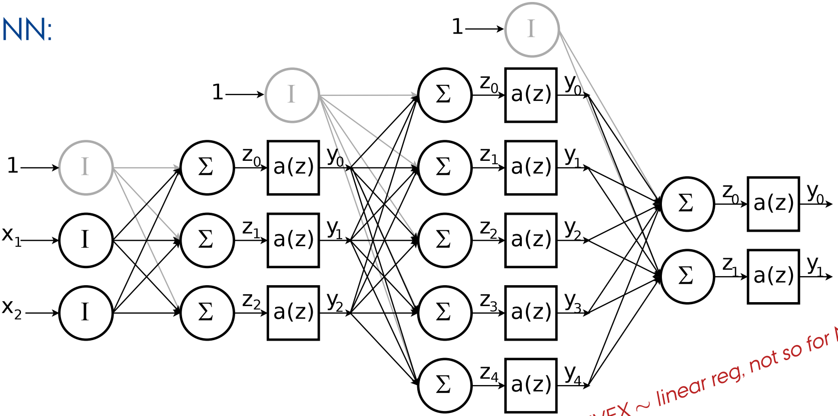
The algoritmn for updating via steps reads

$$\mathbf{w}^{(\text{next step})} = \mathbf{w} - \eta \nabla_{\mathbf{w}} J(\mathbf{w})$$

with η being the step size.

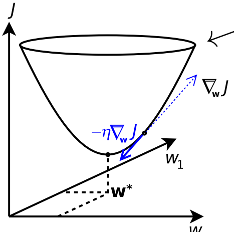
RESUMÉ: Training Deep Neural Networks

NN:



CONVEX ∼ linear reg, not so for NNs

GD (via BPROP):

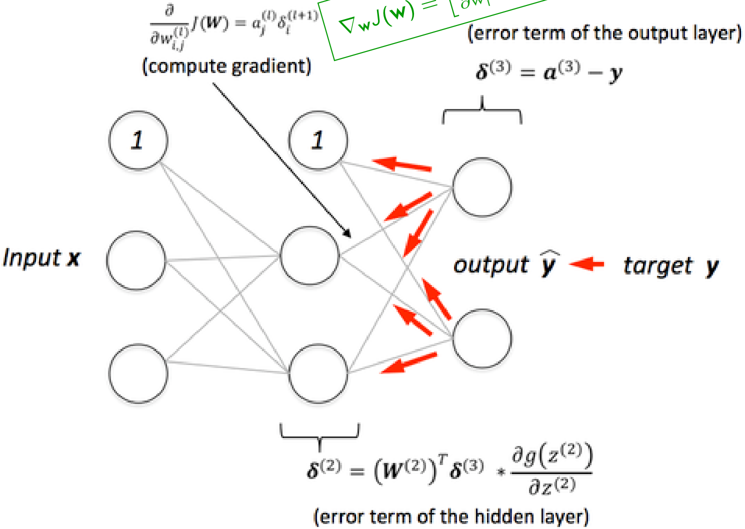


$$\mathbf{w}^{(\text{next})} = \mathbf{w} - \eta \nabla_{\mathbf{w}} J(\mathbf{w})$$

Backpropagation (BProp)

Training MLPs

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = \left[\frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \ldots, \frac{\partial}{\partial w_m}\right]^T J(\mathbf{w})$$



NOTE: [https://sebastianraschka.com/images/faq/visual-backpropagation/backpropagation.png

RESUMÉ: Training Deep Neural Networks

Equation 4-6. Gradient vector of the cost function

$$\nabla_{\boldsymbol{\theta}} \text{MSE}(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial}{\partial \theta_0} \text{MSE}(\boldsymbol{\theta}) \\ \frac{\partial}{\partial \theta_1} \text{MSE}(\boldsymbol{\theta}) \\ \vdots \\ \frac{\partial}{\partial \theta_n} \text{MSE}(\boldsymbol{\theta}) \end{pmatrix} = \frac{2}{m} \mathbf{X}^T (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$$

Notice that this formula involves calculation
set **X**, at each Gradient Descent step! This i
called *Batch Gradient Descent*: it uses the w
data at every step (actually, *Full Gradient D*
be a better name). As a result it is terribly sl
ing sets (but we will see much faster Gradient Descent algorithms
shortly). However, Gradient Descent scales we
features; training a Linear Regression model
dreds of thousands of features is much fa
Descent than using the Normal Equation or SV

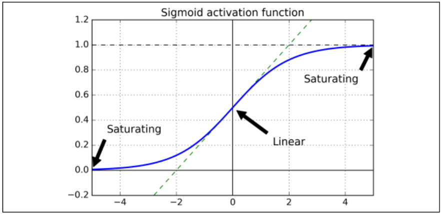


Figure 11-1. Logistic activation function saturation

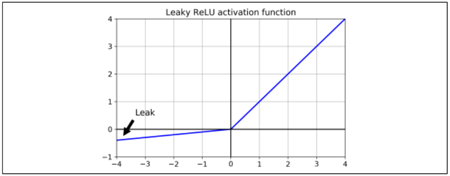


Figure 11-2. Leaky ReLU

Once you have the gradient vector, which points uphill, ju
tion to go downhill. This means subtracting $\nabla_{\boldsymbol{\theta}}\text{MSE}(\boldsymbol{\theta})$
learning rate η comes into play:[6] multiply the gradient v
size of the downhill step (Equation 4-7).

Equation 4-7. Gradient Descent step

$$\boldsymbol{\theta}^{(\text{next step})} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \text{MSE}(\boldsymbol{\theta})$$

$$\mathbf{w}^{(\text{next})} = \mathbf{w} - \eta \nabla_{\mathbf{w}} J(\mathbf{w})$$

MODEL CAPACITY

Model capacity

Dummy and Paradox classifier:
capacity *fixed* ~ 0, cannot generalize at all!

Linear regression for a polynomial model:
capacity $\sim$ degree of the polynomial, x^n

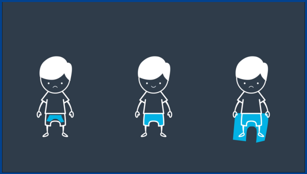
Neural Network model:
capacity $\propto$ number of neurons

Homo sabiens (”modern humans”):
capacity $\propto$ the IQ 'score' function?

$\Rightarrow$ **Capacity** can be hard to express as a quantity for some models, but you need to choose..

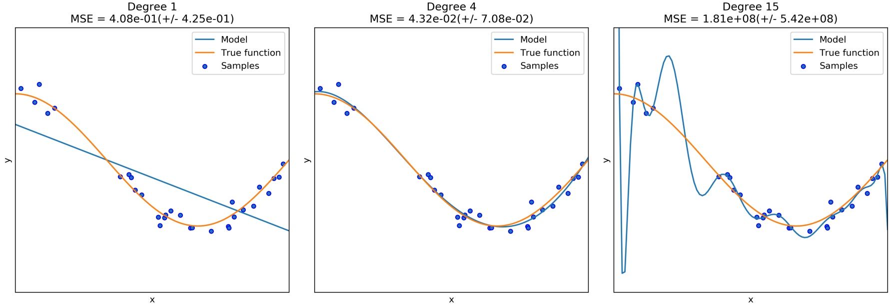
$\Longrightarrow$ how to choose the **optimal** *capacity*?

UNDER- AND OVERFITTING



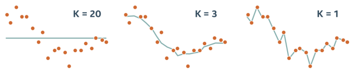
Under- and overfitting

Polynomial linear reg. fit for underlying model: `cos(x)`



- underfitting:
  capacity of model too low,
- overfitting:
  capacity to high.

k-NN from L01:



$\Longrightarrow$ how to choose the **optimal** capacity?

NOTE: HOML: *Constraining a model [..] reduce risk of overfitting [via] regularization => L09*

Generalization Err., Over-, Underfit, Capacity

The Bias/Variance Tradeoff

An important theoretical result of statistics and Machine Learning is the fact that a model's generalization error can be expressed as the sum of three very different errors:

Bias

This part of the generalization error is due to wrong assumptions, such as assuming that the data is linear when it is actually quadratic. A high-bias model is most likely to underfit the training data.[10]

Variance

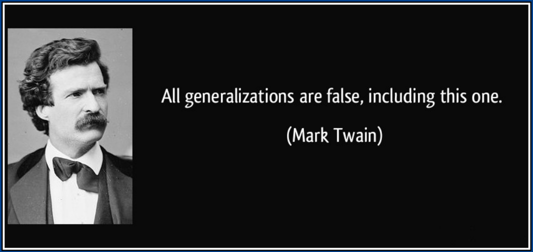
This part is due to the model's excessive sensitivity to small variations in the training data. A model with many degrees of freedom (such as a high-degree polynomial model) is likely to have high variance, and thus to overfit the training data.

Irreducible error

This part is due to the noisiness of the data itself. The only way to reduce this part of the error is to clean up the data (e.g., fix the data sources, such as broken sensors, or detect and remove outliers).

Increasing a model's complexity will typically increase its variance and reduce its bias. Conversely, reducing a model's complexity increases its bias and reduces its variance. This is why it is called a tradeoff.

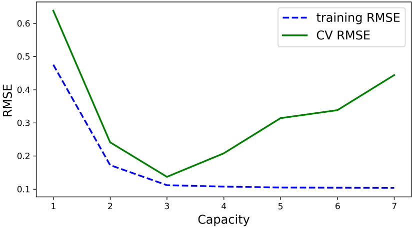
GENERALIZATION ERROR



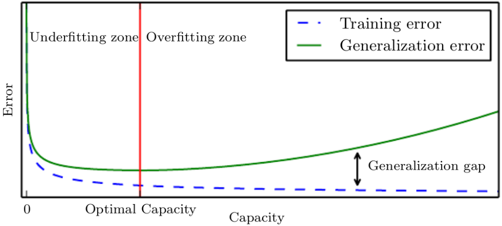
Generalization Error

RMSE-capacity plot for lin. reg. with polynomial features

(capacity $\sim$ degree of poly)



(Figure 5.3 from [DL])



Inspecting the plots from the exercise (`.ipynb`) and [DL], extracting the concepts:

- training/generalization error,
- generalization gab,
- underfit/overfit zone,
- optimal capacity (best-model, early stop),
- (and the two axes: x/capacity, y/error.)

Definition of Machine Learing [ML] def of AI?

"A machine learning algorithm is an algorithm that is able to learn from data. But what do we mean by learning?" [DL]

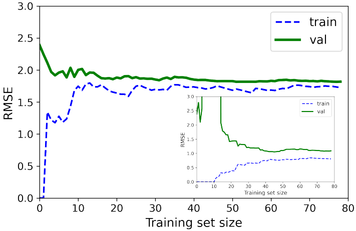
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T, as measured by P, improves with experience E."

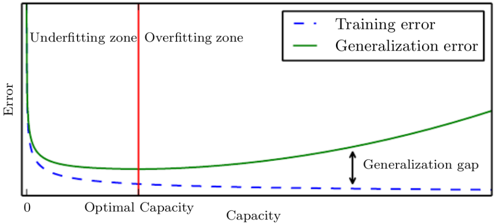
— Mitchell (1997).

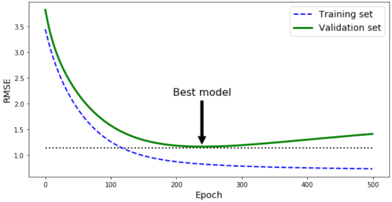
Generalization Error

NOTE: three methods/plots:

i) via **learning curves** as in [HOML],
ii) via an **error-capacity** plot as in [GITHOML] and [DL],
iii) via an **error-epoch** plot as in [GITHOML].







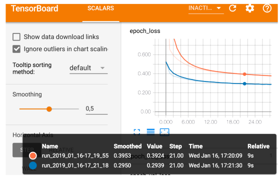


Figure 10-16. Visualizing Learning Curves with TensorBoard