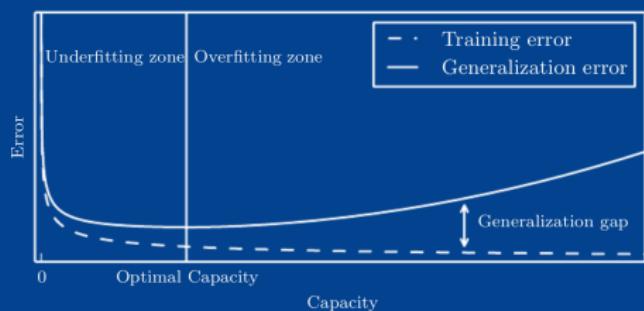




## LESSON 7: Pipelines, Model-capacity, Under/over-fitting, Generalization

CARSTEN EIE FRIGAARD

AUTUMN 2020



"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).

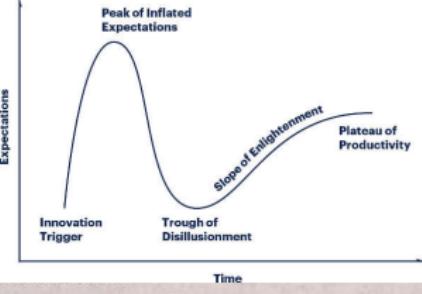
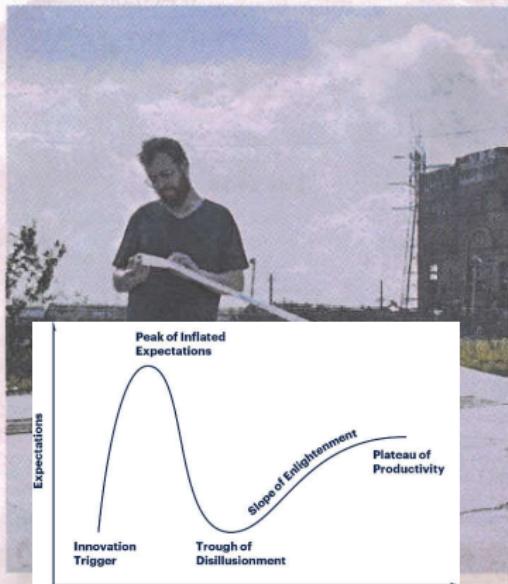
# L07: Pipelines, Model-capacity Under/over-fitting, Generalization

## Agenda

- ▶ Generelt om ML-systemer, og klasse diskussion vdr. ML
- ▶ Pipelines i Scikit-learn
  - Exercise: [L07/pipelines.ipynb](#) (NB: ny opgave)
- ▶ Model Capacity
- ▶ Under/over-fitting
  - Exercise: [L07/capacity\\_under\\_overfitting.ipynb](#)
- ▶ Generalization Error
  - Exercise: [L07/generalization\\_error.ipynb](#)







Werketlavisen

**Bøger**

Aman Boe: Interiørarkitekt og ark. Lone Marie Knudsen interview og andre rottende. Side 2-12

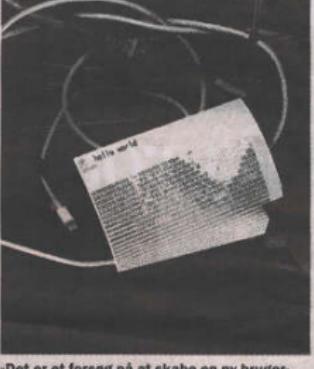
Karen, Katherine Winkler: "Hjemmet jeg er vokset op i har fået tilbage til livet igen". Side 13-14

Peter Petzold: "Måske har jeg ikke været den bedste i den øvrige efter alt udvalgtes pris". Side 14-15

Katharina: "For andet gange, hvilket ikke har været tilfældet de sidste gange jeg har rejst ud med min børn". Side 16-17

**INTERVIEW** Den engelske programværkskoder Ross Goodwin har forbi det en kærlig anlagt, som har skrevet en surrealistisk historie om en insekthjerne. **På roadtrip med en insekthjerne**

The following image is a dense, abstract graphic composed of numerous small, horizontal, black and white patterns, resembling a microscopic view of insect brain tissue.



# På roadtrip med en insekthjerne

AF MIKKEL BORIS

T 2016 slog computeren AlphaGo den 18-dobbelte verdensmester i brætspillet Go, Lee Sedol. Go er et kompliceret og abstrakt spil, som kræver intuition og kreativitet, men det kunstige intelligens vandt med en række innovative træk overlegen.

Underveis i størrøede kommentarerne

Goodwin til Weekendavisen fra sin lejlighed i Los Angeles.

Inden køreturen havde han brugt måneder på at træne maskinen. Han satte den til at læse et stort korpus af moderne litteratur fra hele verden, så den kunne lære at skrive af de store forfattere.

»Det fungerer ligesom autokorrekturen på din telefon, bare klogere og trænet på en mere litterær kilde. Den skriver bogstav for bogstav, så den har lært sig selv at formulere det næste

når du dekonstruerer dem. Efter at have læst den i ét stæk og fået turen lidt på afstand har romanen fået en universalitet, så jeg kan projicere mine egne oplevelser ind i teksten,« uddyber Goodwin.

– *Det har beskrevet projektet som at leve en insekthjerne at skrive. Hvad betyder det?*

»Jeg forsøgte at pointere, at maskinen ikke er på niveau med den menneskelige hjerne. Et artificielt næret net er en algoritme, der er lavet, som vi troer hjernen fungerer. Ieo synes

»Det er et forsøg på at skabe en ny brugerflade for at skrive. På en måde har jeg jo skrevet en roman med en bil«, fortæller Ross Goodwin om sin AI-forfattede bog 1 the road

## Banker vil vurdere boligpriser med AI



#3 Poul-Henning Kamp | 7. oktober 2020 - 08:54 Blogger

"Fordi vi ved det ikke virker"

Jeg kan ikke forstå at Finanstilsynet ikke simpelthen har henvist til at Skat har prøvet i årevæs uden at få det til at virke ?

16    0

#4 Poul-Henning Kamp | 7. oktober 2020 - 08:56 Blogger

Re: Voodoo?

Databeskyttelsesloven fastlægger, at du har ret til ikke at være genstand for en afgørelse, der alene er baseret på automatisk behandling

Det er faktisk mere fundamentalt end som så: Forvaltningsloven sikrer at du altid får at vide *hvorfor* afgørelsen er taget som den er.

Intet af det juks der idag kaldes "AI" er i stand til at forklare eller dokumentere *hvorfor* resultatet er som det er.

18    1

(Illustration: Puttachat Kumkrong // Bigstock)

Finanstilsynet har endnu ikke godkendt nyudviklet AI, der kan hjælpe bankerne med at vurdere ejendomspriser, siger Michael Hald Graversen fra E-nettet.

### Mest debatterede

1984, Ghostbusters og ...

Regningen for Smittestop-appen stiger

Frankrig og Holland støtter EU-kamp mod amerikanske it-gigant

Forslag om øget it-kontrol i de offentlige: »Det her stykke arbejde SKAL laves«

### WEBINARS AND WHITEPAPERS



#### WEBINAR

Øget effektivitet, fleksibilitet og muligheder for hjemmearbejd  
Cisco Webex



# A computer vision system to monitor the infestation level of Varroa destructor in a honeybee colony

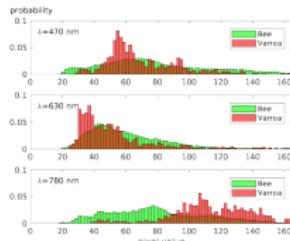


Figure 5: Histograms of bee and varroa pixel intensity values, for the spectral wavelengths 470 nm, 630 nm, and 780 nm respectively, recorded with the JAI camera. The image path is via the mirror-window-mirror, i.e., data were sampled with the setup given in figure 6. The image data for the histogram is the single bee with mite seen in figure 6.



Figure 6: The actual unprocessed camera view of the bees

spectively. The CM analysis was able to rank all wavelengths combinations, using one, two, three or four distinct wavelengths to give a ranking list of ‘best’ combination also taking the JAI camera spectrum into account.

The CM value of the actual choose wavelengths combination (470-630-780 nm) gave a rank just below the CM average score. This CM analysis was conducted after picking the actual used wavelengths, so later versions of the VMU might want to investigate a CM combination with a higher rank.

A specially designed diffuser and a number of narrow spectral LI were mounted in the camera focal diffuse illuminant fixtures.

Figure 6 disp along the passa era, with the gre with the NIR in

### 2.3.3. Real-time processing

A color and motion of 1296×96 from the camera over two separate sustained by ing frames real-

These data will post-process first matching rally coalescing producing a 24 the later image

Lossless real-time can be applied bandwidth that

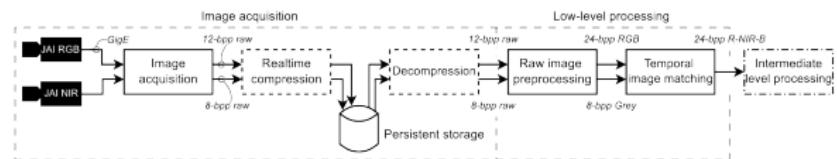


Figure 7: The low-level image processing pipeline. Raw camera images are stored on disk for later retrieval and post-processing. 12- and 8-bits per-pixel are used as the raw JAI/Bayer packed pixel format for the RGB and IR images respectively. Lossless, real-time compression can be introduced if persistent storage bandwidth is less than the raw-stream image rate of 93 MB/sec. The 12- and 8-bpp raw images from the network arrives out-of-order with respect to each other, hence the need for the temporal image matching.

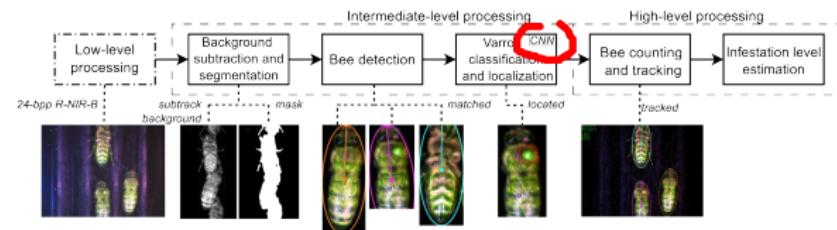
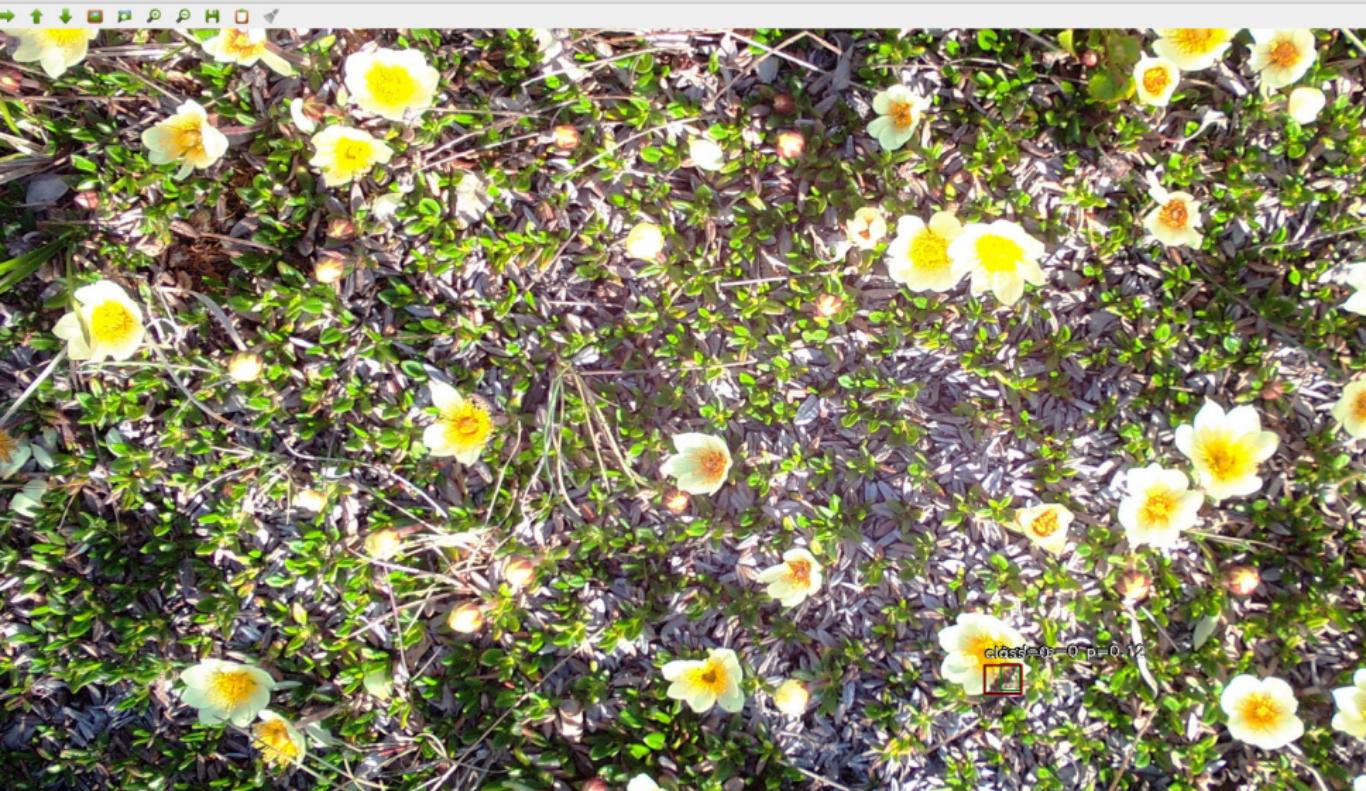
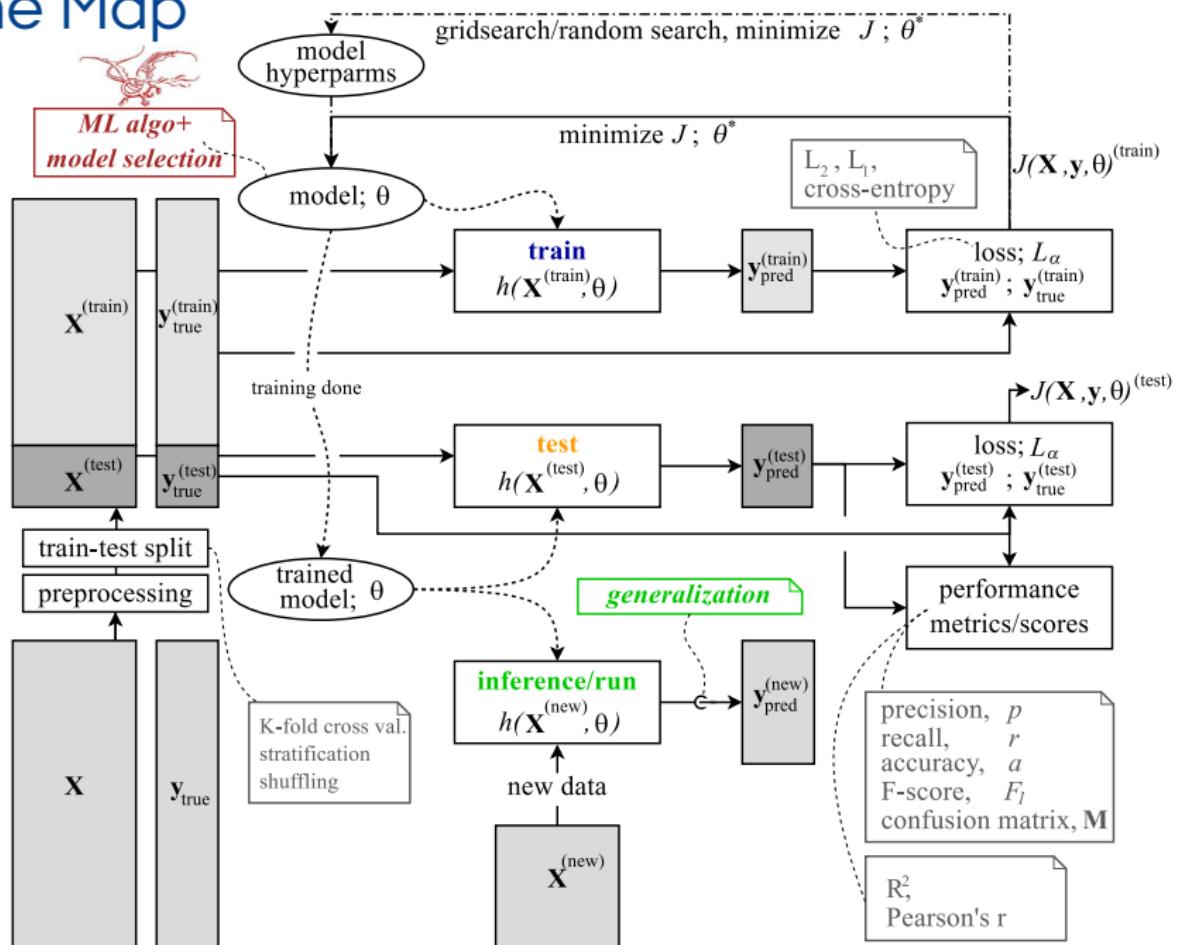


Figure 8: The processing pipeline of the intermediate- to high-level image processing algorithms to analyze and count the number of bees with *Varroa destructor*. A trained convolutional neural network (CNN) was used for the Varroa classification and localization stage.

# BA Project: generic annotation tool



# The Map



# Preprocessing of Data

Scaling, Standardization, Normalization...

Why the need for preprocessing?

*Standardization of datasets is a **common requirement** for many machine learning estimators [...] they might **behave badly** if the individual features do not more or less look like standard normally distributed data. [...]*

[<https://scikit-learn.org/stable/modules/preprocessing.html>]

**Standardization** of a feature vector  $\mathbf{x}$ , giving  $\mathbf{x}'$  mean zero, and standard deviation one

$$\mathbf{x}' = \frac{\mathbf{x} - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}}$$

What kind of estimators needs standardization?

→ **Neural networks (NNs) in particular!**

# Pipelines

Exercise: pipelines.ipynb: Revisit the OECD data in O1s for MLPs

Feature: GDP per capita feature in range 10K to 50K \$.

But MLP expects input in the range [0;1] or perhaps [-1;1].

```
1 # Manual scaling..
2 X_min = np.min(X)
3 X_max = np.max(X)
4 s = X_max - X_min
5
6 print(f"X_min={X_min:.0f}, X_max={X_max:.0f}, s={s:.0f}")
7
8 X_scaled = (X-X_min)/s
9 print(f"X_scaled.shape={X_scaled.shape}")
10 print(f"np.min(X_scaled)={np.min(X_scaled)}")
11 print(f"np.max(X_scaled)={np.max(X_scaled)}")
12
13 mlp.fit(X_scaled ,y.ravel())
14 y_pred_mlp = mlp.predict((M-X_min)/s)
15
16 plt.plot(m, y_pred_lin, "r")
17 plt.plot(m, y_pred_mlp, "k")
18
19 print(f"mpl.score={mlp.score(X_scaled, y.ravel()):0.2f}")
```

```
Prints:
X_min=9055,
X_max=55805, s=46750
X_scaled.shape=(29, 1)
np.min(X_scaled)=0.0
np.max(X_scaled)=1.0
mpl.score=0.70
```

# Pipelines

## OECD Data and MLPs: introducing a MinMaxScaler

```
1 # Now, do the same but via a pipeline..
2 from sklearn.preprocessing import MinMaxScaler
3
4 scaler = MinMaxScaler()
5 scaler.fit(X)
6 X_scaled = scaler.transform(X)
7 M_scaled = scaler.transform(M)
8
9 mlp.fit(X_scaled, y)
10 y_pred_mlp = mlp.predict(M_scaled)
11
12 print(f"mpl.score={mlp.score(X_scaled, y):0.2f}")
13
14 # PRINTS: mpl.score=0.71
```

# Pipelines

OECD Data and MLPs: putting everything in a Full Pipeline

```
1 # Or even better, in a full pipeline..
2 from sklearn.pipeline import Pipeline
3
4 pipe = Pipeline( # indent pipeline as VHDL port mappings!
5     [
6         ('scaler', MinMaxScaler()),
7         ('mlp', mlp)
8     ]
9 )
10
11 pipe.fit(X, y)
12
13 print(f"pipe.score(..)={pipe.score(X, y):0.2f}")
14
15 # PRINTS: pipe.score(..)=0.68
```

# Pipelines

Brief intro to Scikit-learn pipelines..

Python code from capacity\_under\_overfitting.ipynb

```
1  from sklearn.pipeline import Pipeline
2  from sklearn.preprocessing import PolynomialFeatures
3  from sklearn.linear_model import LinearRegression
4  from sklearn.model_selection import cross_val_score
5
6  polynomial_features = PolynomialFeatures(degree=degrees[i], ..
7  linear_regression = LinearRegression()
8
9  pipeline = Pipeline(
10    [
11      ("polynomial_features", polynomial_features),
12      ("linear_regression", linear_regression)
13    ]
14 )
15
16 pipeline.fit(X[:, np.newaxis], y)
17
18 scores = cross_val_score(
19   pipeline, X[:, np.newaxis], y,
20   scoring="neg_mean_squared_error", cv=10
21 )
22 score_mean = -scores.mean()
```

# RESUMÉ: L02/performance\_metrics.ipynb

## Classification metrics

See the [Classification metrics](#) section of the user guide for further details.

<code>metrics.accuracy_score(y_true, y_pred[, ...])</code>	Accuracy classification score.
<code>metrics.auc(x, y[, reorder])</code>	Compute Area Under the Curve (AUC) using the trapezoidal rule
<code>metrics.average_precision_score(y_true, y_score)</code>	Compute average precision (AP) from prediction scores
<code>metrics.cohen_kappa_score(y1, y2[, labels, ...])</code>	Cohen's kappa: a statistic that measures inter-annotator agreement.
<code>metrics.confusion_matrix(y_true, y_pred[, ...])</code>	Compute confusion matrix to evaluate the accuracy of a classification
<code>metrics.f1_score(y_true, y_pred[, labels, ...])</code>	Compute the F1 score, also known as balanced F-score or F-measure
<code>metrics.log_loss(y_true, y_pred[, eps, ...])</code>	Log loss, aka logistic loss or cross-entropy loss.
<code>metrics.precision_score(y_true, y_pred[, ...])</code>	Compute the precision
<code>metrics.recall_score(y_true, y_pred[, ...])</code>	Compute the recall
<code>metrics.roc_auc_score(y_true, y_score[, ...])</code>	Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.
<code>metrics.roc_curve(y_true, y_score[, ...])</code>	Compute Receiver operating characteristic (ROC)
<code>metrics.zero_one_loss(y_true, y_pred[, ...])</code>	Zero-one classification loss.

## Regression metrics

See the [Regression metrics](#) section of the user guide for further details.

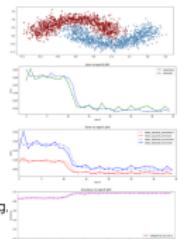
<code>metrics.explained_variance_score(y_true, y_pred)</code>	Explained variance regression score function
<code>metrics.max_error(y_true, y_pred)</code>	max_error metric calculates the maximum residual error.
<code>metrics.mean_absolute_error(y_true, y_pred)</code>	Mean absolute error regression loss
<code>metrics.mean_squared_error(y_true, y_pred[, ...])</code>	Mean squared error regression loss
<code>metrics.mean_squared_log_error(y_true, y_pred)</code>	Mean squared logarithmic error regression loss
<code>metrics.median_absolute_error(y_true, y_pred)</code>	Median absolute error regression loss
<code>metrics.r2_score(y_true, y_pred[, ...])</code>	R <sup>2</sup> (coefficient of determination) regression

## Notes on Keras MLPs

Typical Keras MLP Supervised Classifier setup:

- ▶ loss function  
`loss='categorical_crossentropy'`
- ▶ metrics collected via history  
`metrics=[`  
  `'categorical_accuracy',`  
  `'mean_squared_error',`  
  `'mean_absolute_error']`  
`]`
- ▶ input lay.: categorical encoding,
- ▶ output lay.: softmax function.

And notice that Keras do *not* provide metrics like  
precision, recall, F1  
but instead  
`categorical_accuracy`, `binary_accuracy`



# Model capacity

Exercise: `capacity_under_overfitting.ipynb`

Dummy and Paradox classifier:

*capacity fixed  $\sim 0$ , 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/layers*

NOTE: 'score' function for human brain capacity: **the IQ!**

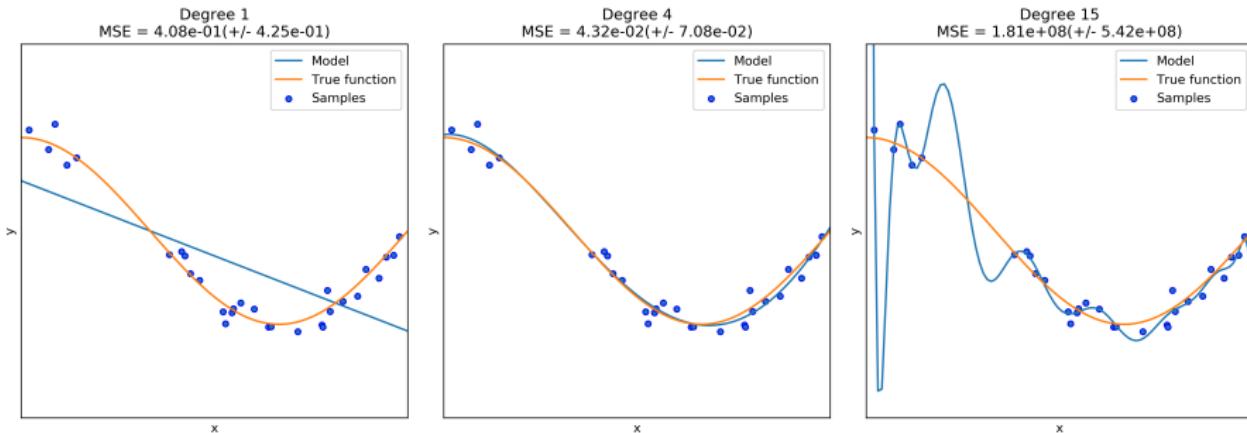
⇒ **Capacity** can be hard to express as a quantity for some models, but you need to choose..

⇒ how to choose the **optimal capacity**?

# Under- and overfitting

Exercise: `capacity_under_overfitting.ipynb`

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



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

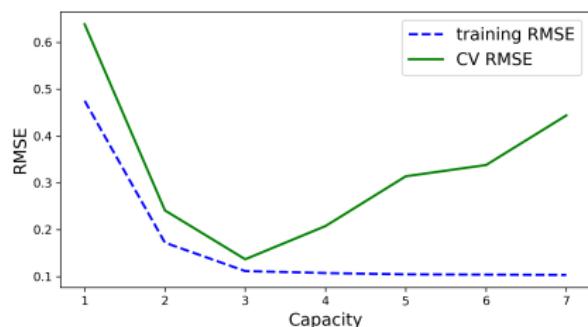
⇒ how to choose the **optimal** capacity?

# Generalization Error

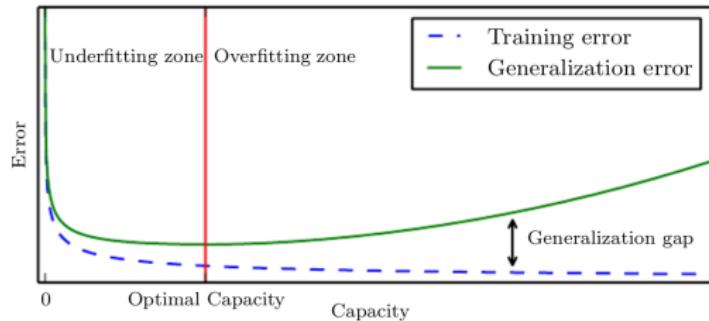
Exercise: generalization\_error.ipynb

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.)

# Generalization Error

Definition of ML:

“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

Exercise: generalization\_error.ipynb

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].

