Linear regression
Polynomial regression
Time series prediction
and beyond

Beril Sirmacek
School of AI, Enschede

January 2019

Today's topics ☺

- Linear regression
- Piecewise linear regression
- Polynomial regression
- RNN
- LSTM
- Predicting housing prices
- Touching back to the autoencoders

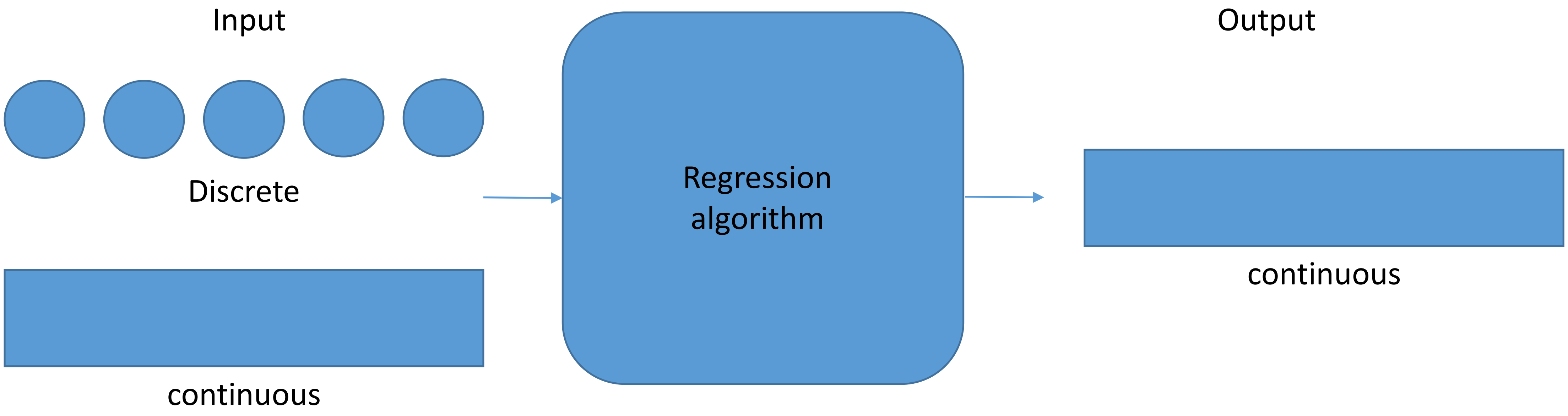
Linear regression

- In the most general sense, a regression algorithm tries to design a function, let's call it ***f***, that maps and input to an output.

- Regression can also be posed with multiple output, as opposed to just one real number. In that case, it is called *multivariate regression*.

- The input of the function can be continuous or discrete. But the output must be continuous.

Linear regression



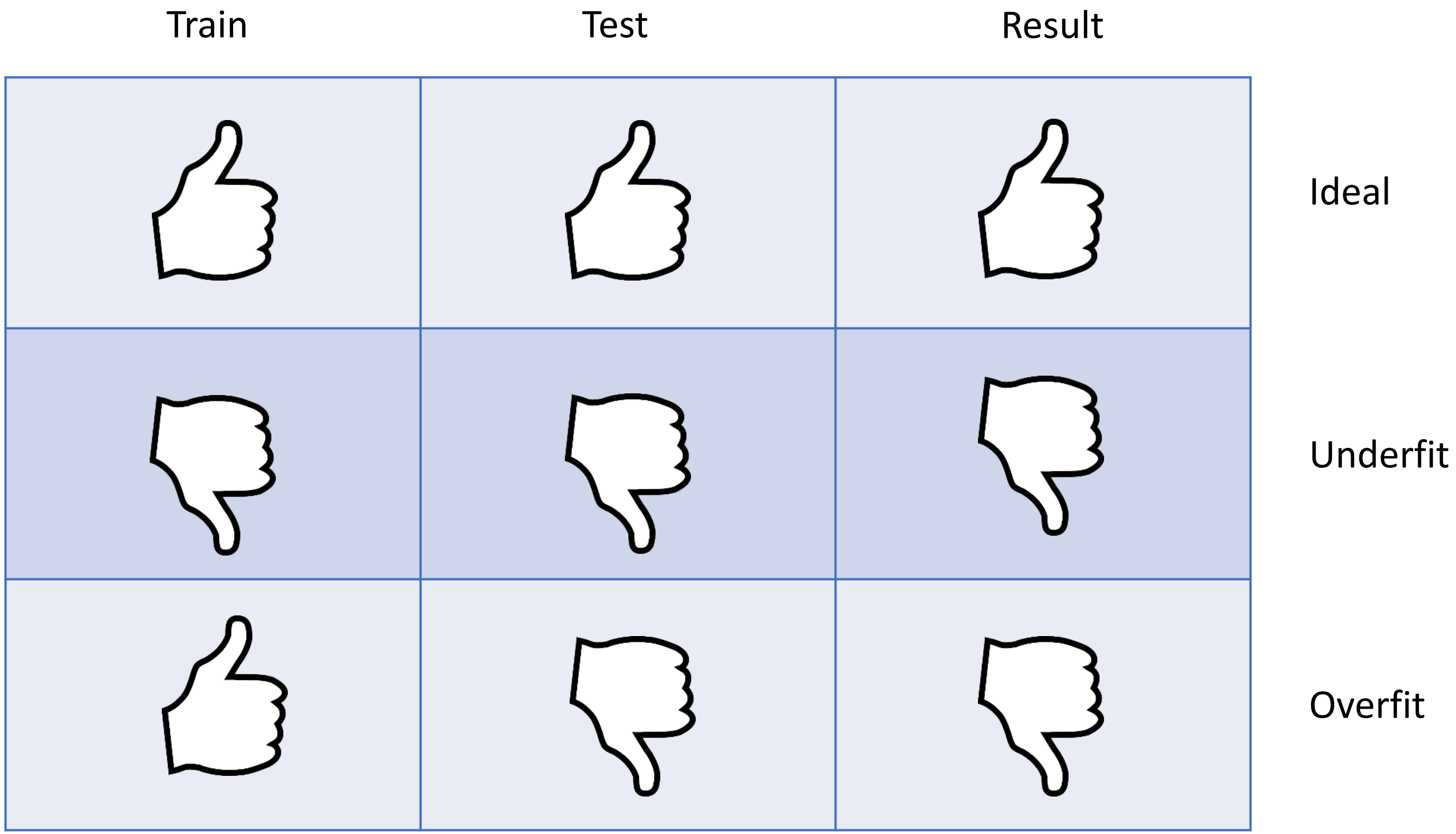
How do you know the regression algorithm is working?

To measure the success of the learning algorithm, you need to measure two important concepts; **variance** and **bias**.

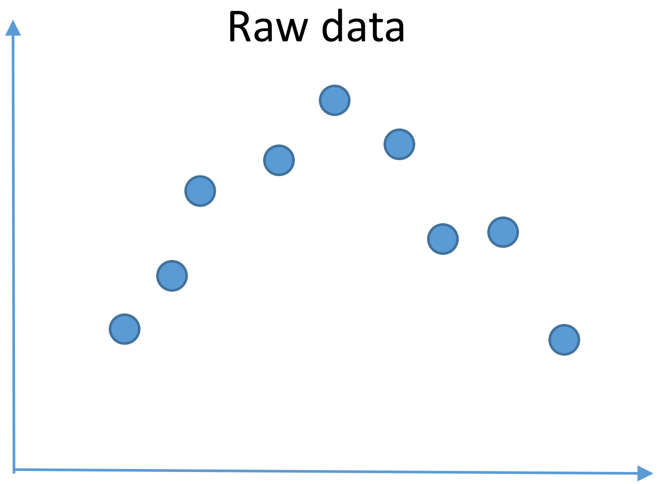
Variance indicates how sensitive a prediction is to the training set that was used. Ideally, how you choose the training set shouldn’t matter, meaning a lower variance is desired.

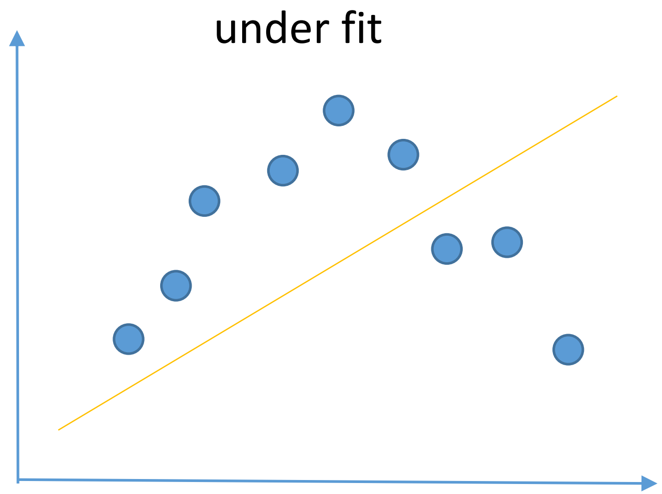
Bias indicates the strength of assumptions made about the training data set. Making too many assumption might make the model unable to generalize, so you should prefer low bias.

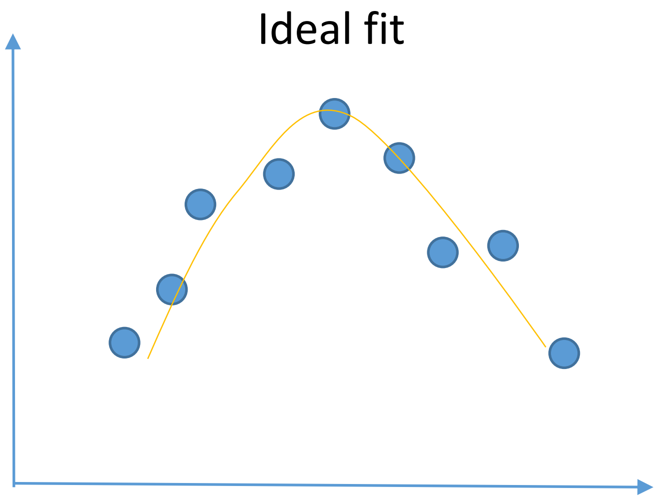
How do you know the regression algorithm is working?

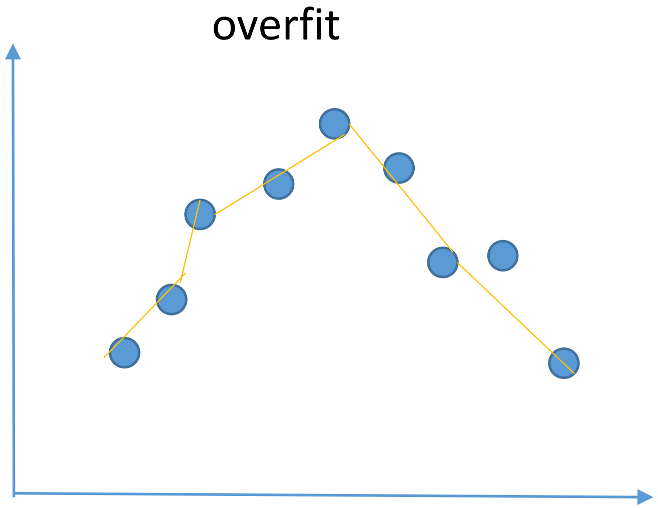


How do you know the regression algorithm is working?









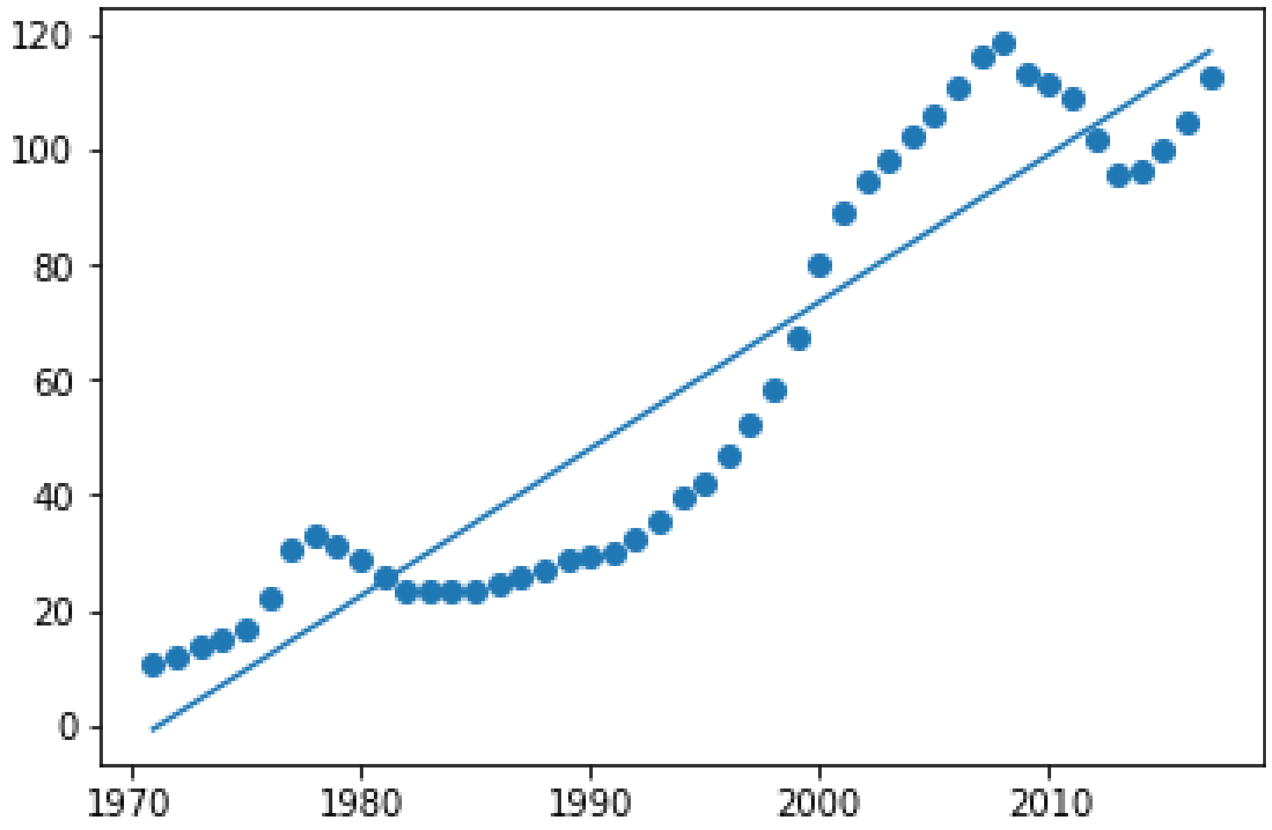
Linear regression

In linear models, no matter what parameters are learned, the function remains linear.

The nonlinear neural network model with a hidden layer, on the other hand, is flexible enough to approximately represent any function.

(Adding more hidden layers greatly improves the expressive power of the network.)

Linear regression



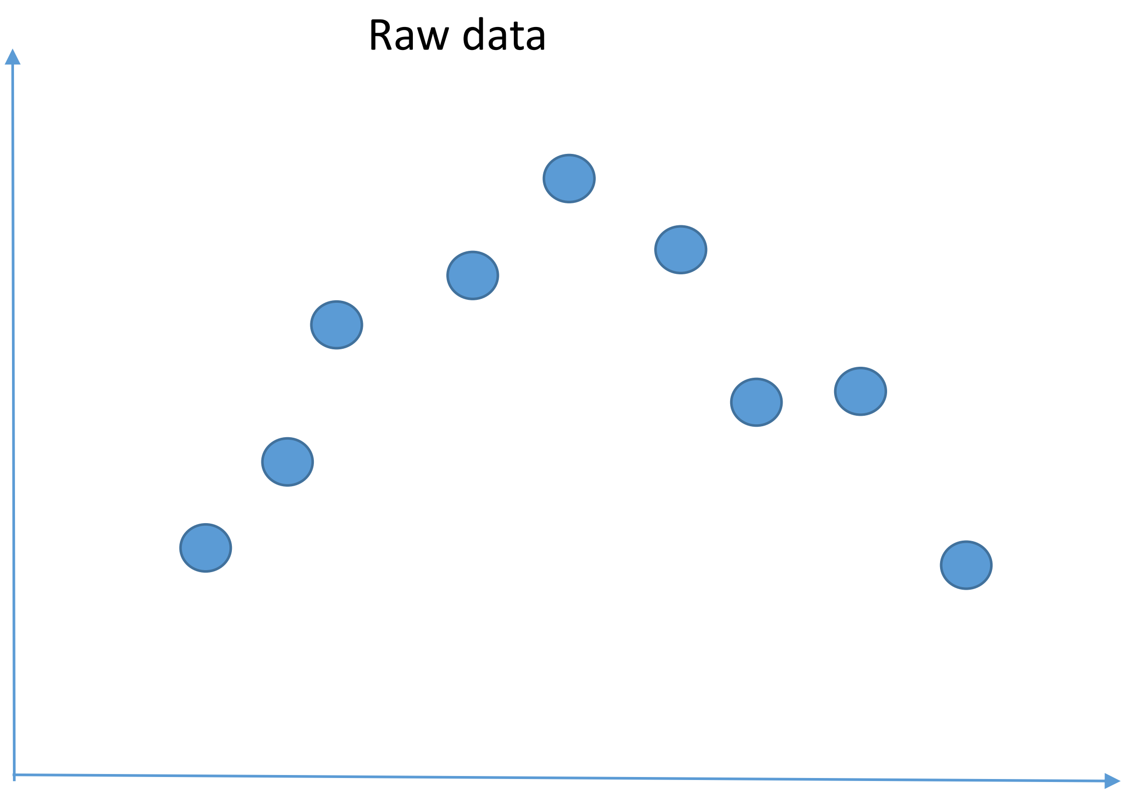
https://data.oecd.org/price/housing-prices.htm

Linear regression

```
#train model on data
housing_reg = linear_model.LinearRegression()
housing_reg.fit(x_values, y_values)

#visualize results
plt.scatter(x_values, y_values)
plt.plot(x_values, housing_reg.predict(x_values))

plt.show()
```


https://data.oecd.org/price/housing-prices.htm

Linear regression

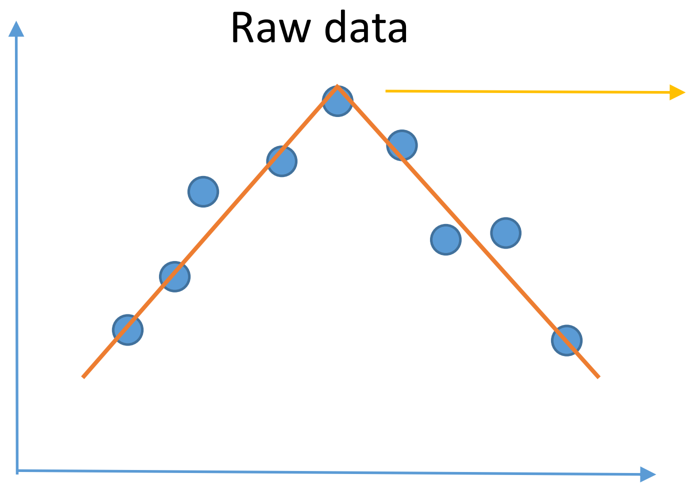


https://data.oecd.org/price/housing-prices.htm

Piecewise linear regression

Segmented regression

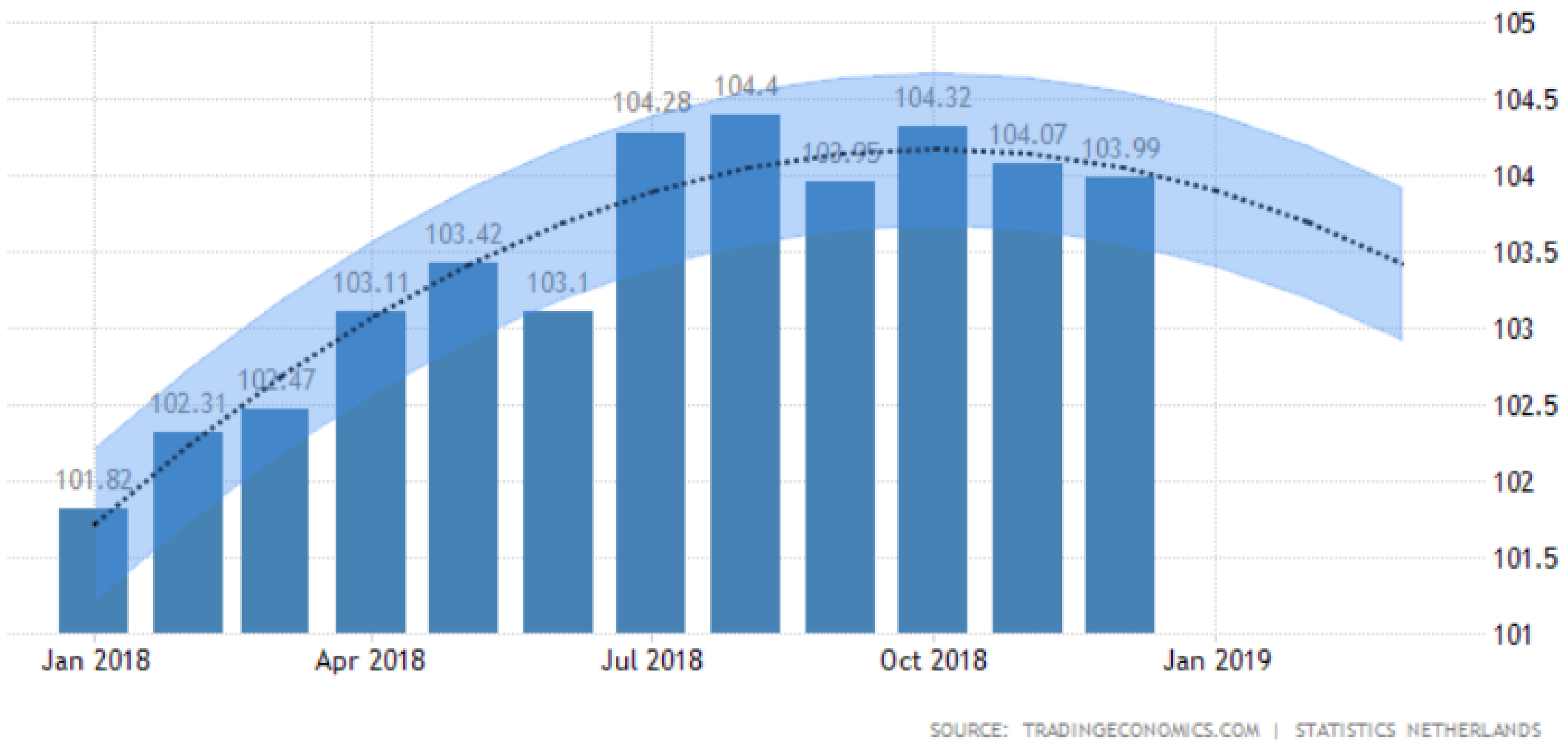
Broken stick regression



Problem: How to find the breakpoints?
(One approach, start with a single line and break down into pieces)

Things to consider:
Error criteria
Stopping criteria

Polynomial regression



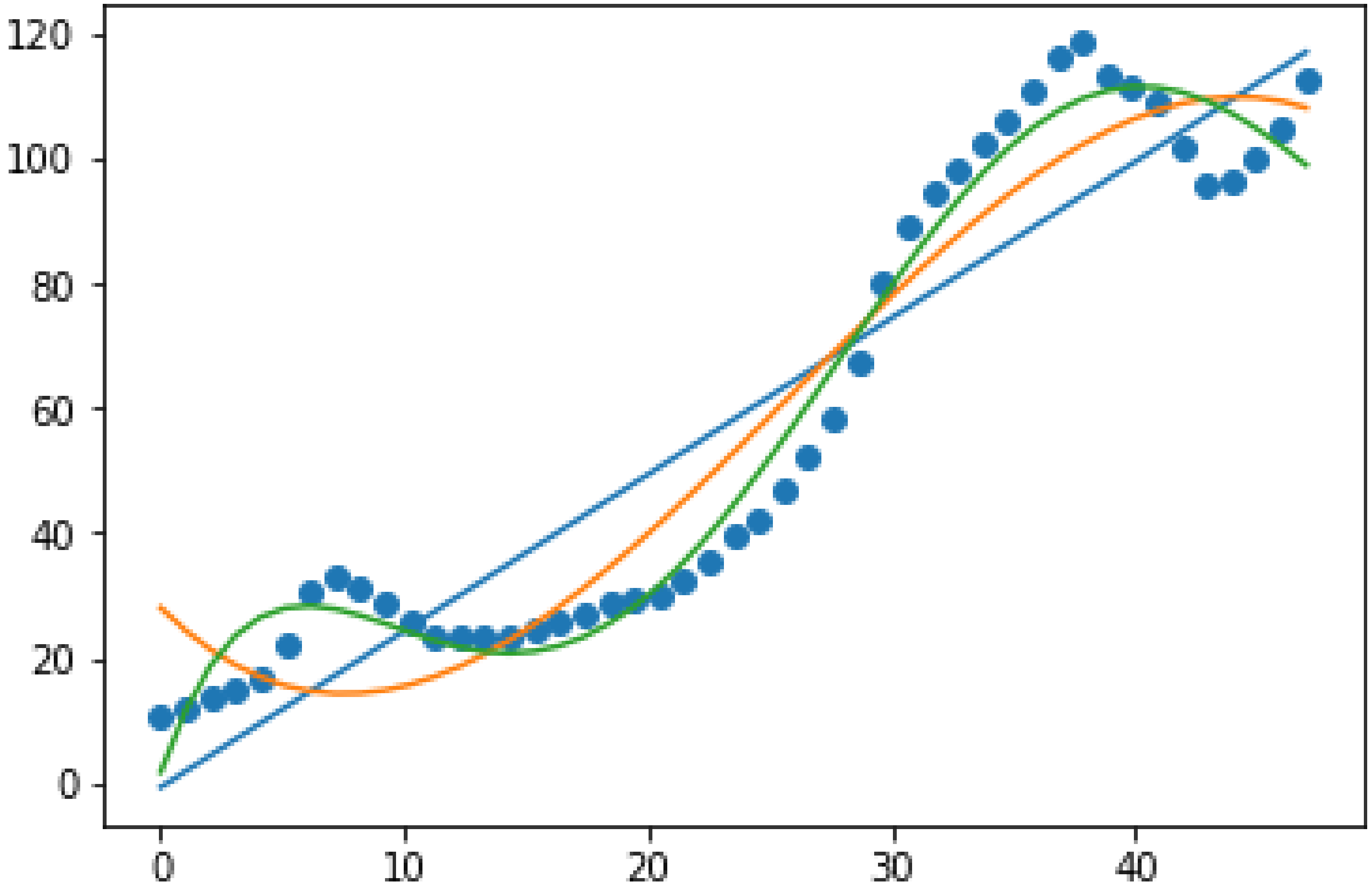
https://tradingeconomics.com/netherlands/ (API available)

Polynomial regression

```
from sklearn.linear_model import Ridge
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline

y_train =y_values
x_plot =  np.linspace(0,len(y_train), len(y_train))
plt.scatter(x_plot, y_train )

x_plot = x_plot.reshape(-1,1)

for count, degree in enumerate([1, 3, 5]):
  model = make_pipeline(PolynomialFeatures(degree), Ridge())
  model.fit(x_plot, y_train)
  y_plot = model.predict(x_plot)
  plt.plot(x_plot, y_plot)


plt.show()
```

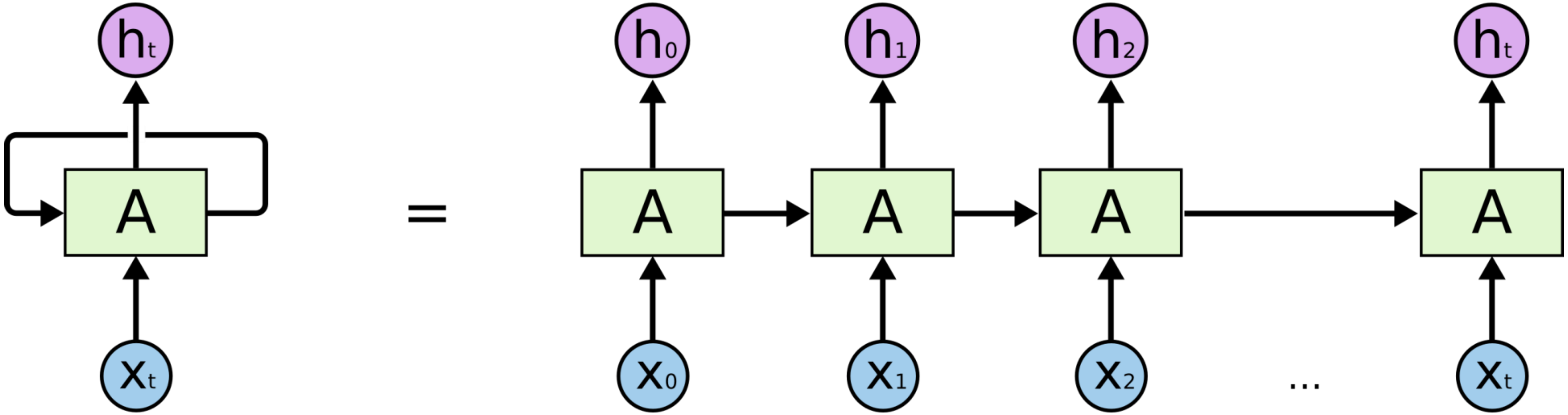
https://scikit-learn.org/stable/

Polynomial regression



Can already be used for estimation!

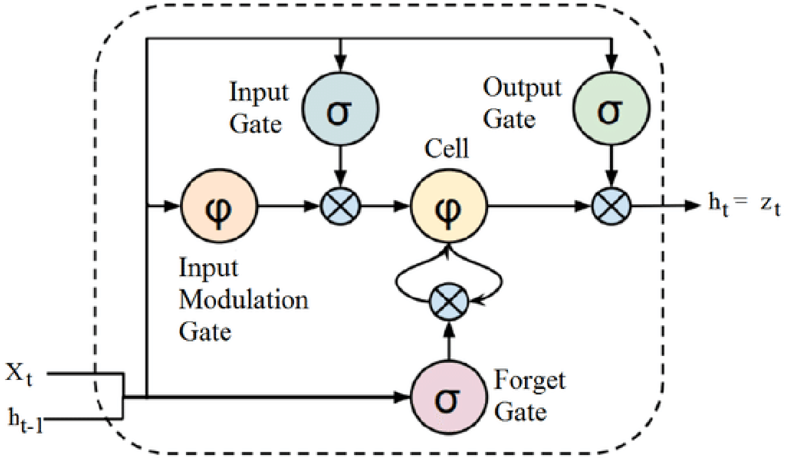
RNN



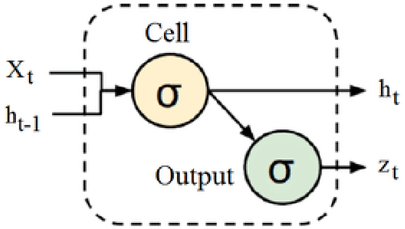
I am French…………………I speak very good French…………. I can speak French.

https://youtu.be/cdLUzrjnlr4

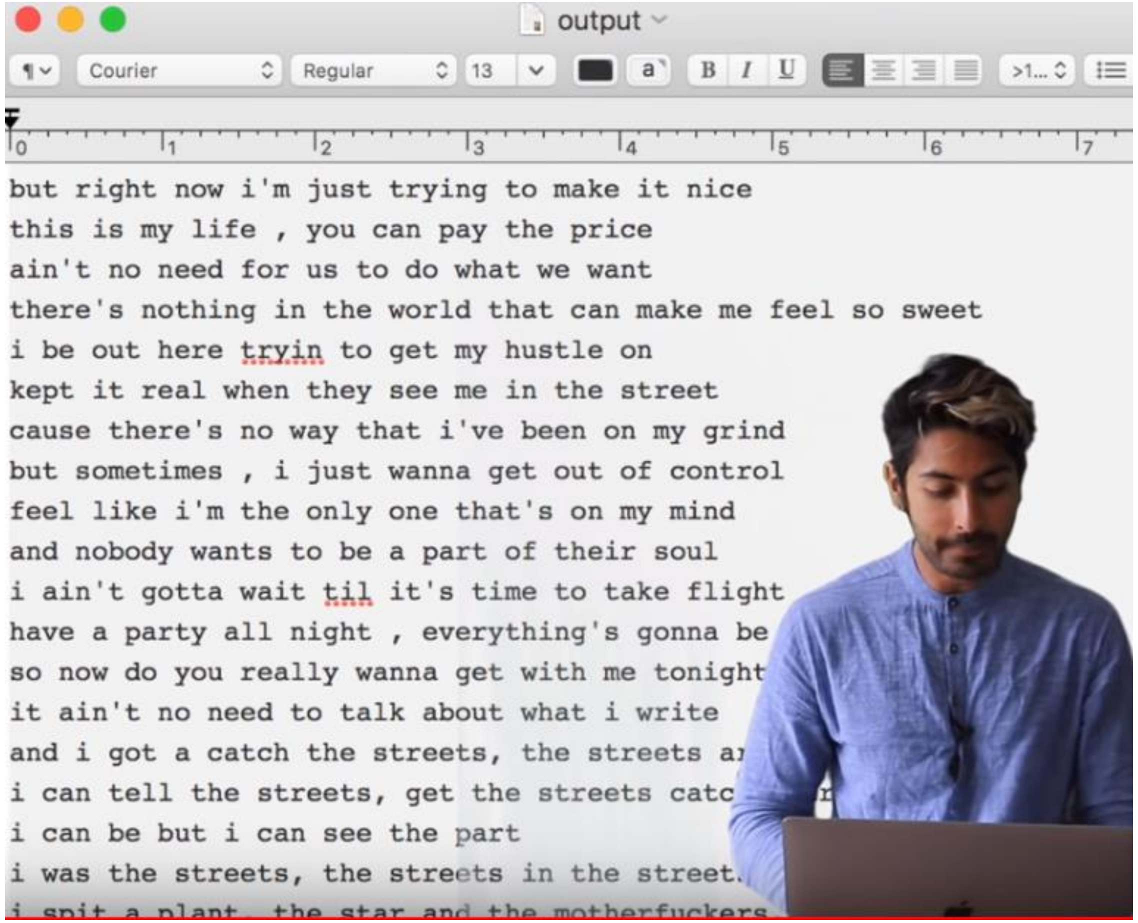
LSTM



LSTM

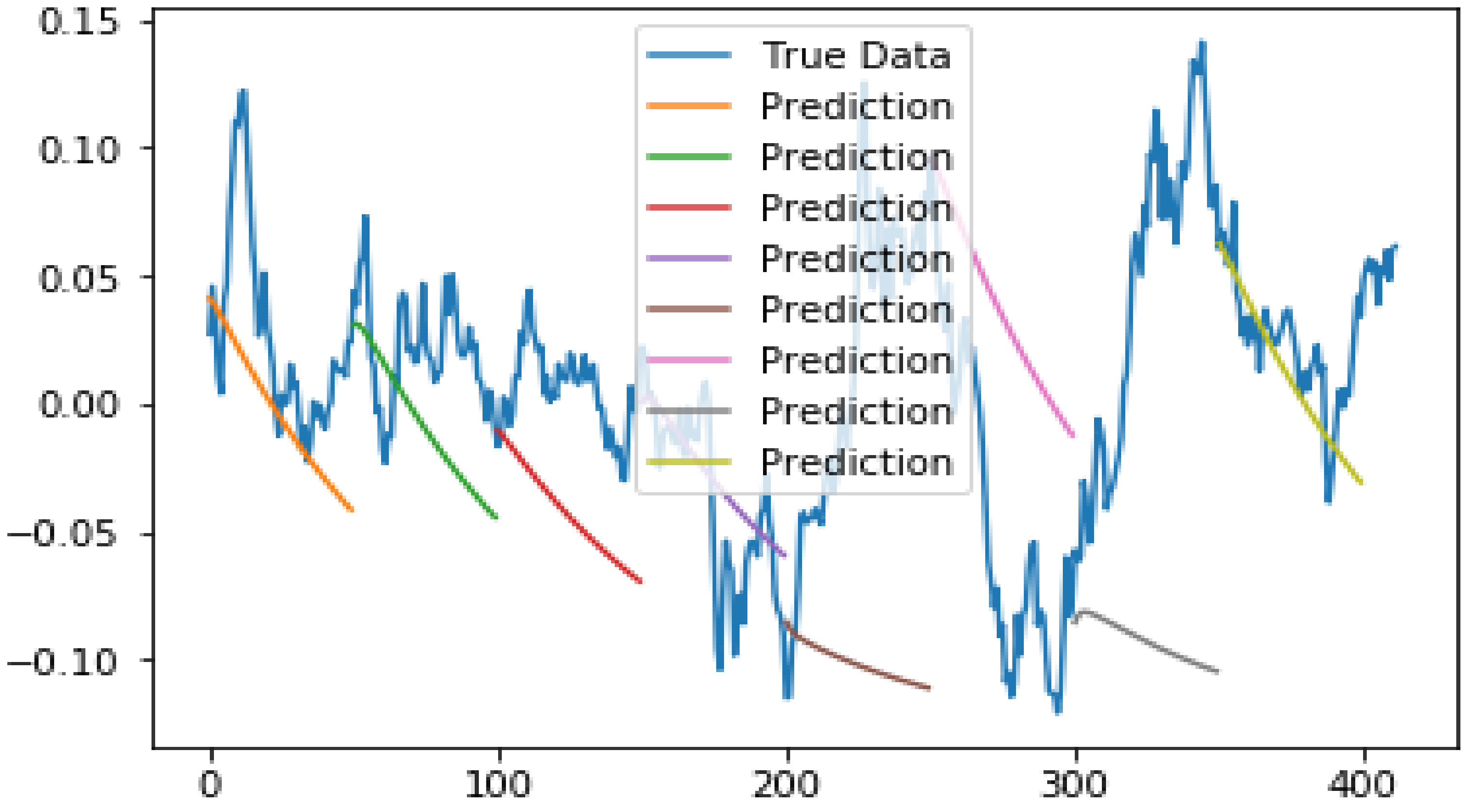


RNN



https://youtu.be/9zhrxE5PQgY

LSTM



https://github.com/bsirmacek/How-to-Predict-Stock-Prices-Easily-Demo/blob/master/stockdemo.ipynb

LSTM

If you want to apply on real data;

StatLine

Themes   Recent   Help   Search...

Trade and industry; employment and finance per sector, SIC 2008

Changed on: 29 March 2018

Variables can be dragged to the header, rows or columns of the table. In the header only one item of a variable can be selected. X

Periods 2015

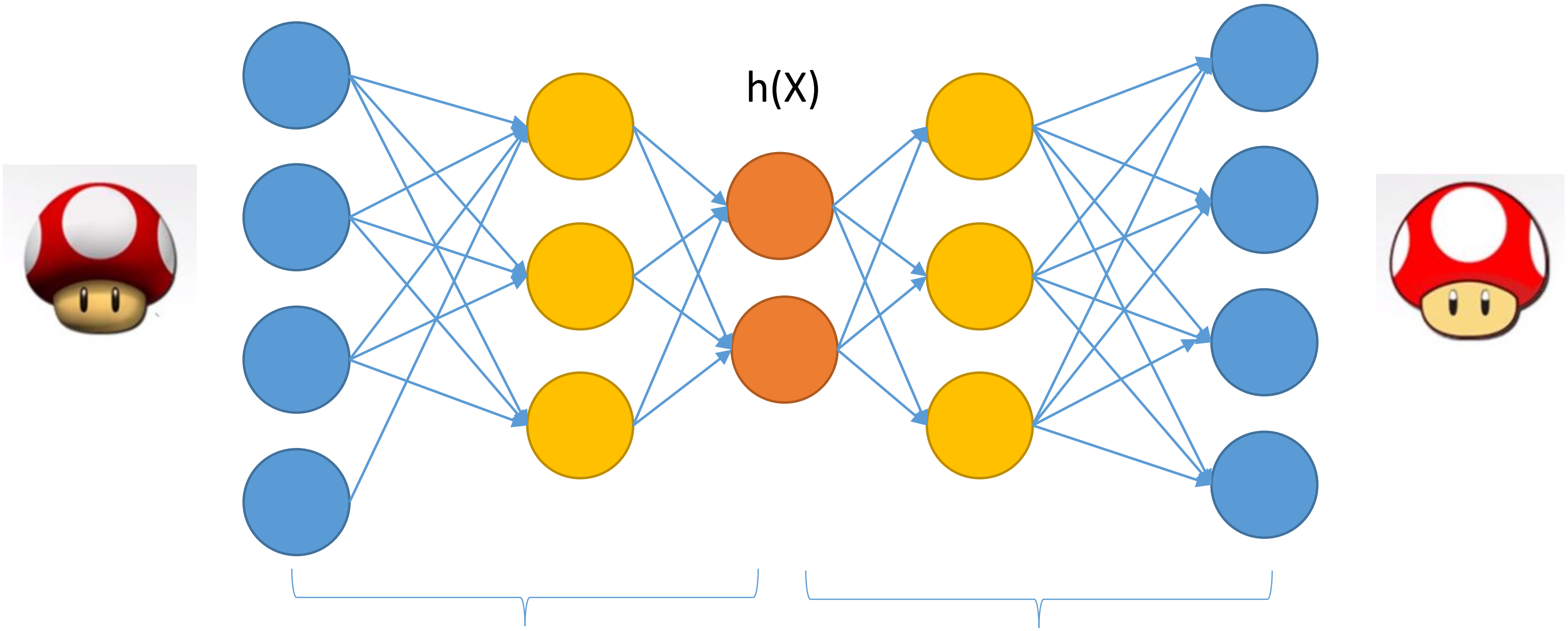
Topic

Sector/branches (SIC 2008)	Jobs Employee	Employed person	Labour volume persons employed Employee	Employed person	Operating returns Total	Net turnover	Other revenues	Operating costs Total	Purchase value of sales Total	Purchase value not elsewhere classified	Personnel costs Total	Gross wages and salaries	Other operating costs Total	Costs of energy use	Housing costs
	x 1 000				x mln euro										
F Construction	313.4	515.0	288.6	466.3	82,715	80,595	2,120	75,517	46,466	116	19,529	12,840	8,031	359	1,04[illegible]
41 Construction buildings, development	87.7	159.8	79.6	144.3	32,621	32,552	69	29,746	21,820	85	5,377	3,639	2,216	74	34[illegible]
412-439 Construction (no development)	305.7	503.7	282.1	456.6	78,225	76,173	2,052	71,691	43,473	115	19,048	12,484	7,731	350	99[illegible]
412 Construction of buildings	80.1	148.6	73.1	134.6	28,132	28,131	1	25,920	18,828	84	4,896	3,284	1,916	65	29[illegible]
42 Civil engineering	50.8	66.4	48.2	61.5	14,169	13,128	1,041	13,681	7,829	14	3,836	2,570	1,597	119	15[illegible]
421 Construction of roads and railways	27.0	36.8	25.6	33.9	8,197	8,025	172	7,892	4,939	14	1,892	1,292	918	52	8[illegible]
422 Construction of utility projects	15.6	19.8	14.7	18.3	3,161	2,975	187	3,052	1,476	0	1,109	734	417	9	4[illegible]
43 Specialised construction activities	174.9	288.8	160.9	260.4	35,925	34,914	1,011	32,090	16,817	17	10,317	6,631	4,218	165	54[illegible]
431 Demolition and site preparation	12.8	20.2	11.8	18.5	2,901	2,835	66	2,577	1,200	4	734	460	519	48	4[illegible]
432 Construction installation	105.9	136.1	98.9	124.7	18,608	18,034	574	17,506	9,222	9	6,113	4,111	1,901	46	28[illegible]
433 Building completion	31.3	86.9	27.5	76.3	8,267	8,139	128	6,553	3,495	3	1,924	1,110	984	28	12[illegible]
439 Other specialised construction	25.0	45.6	22.7	40.9	6,149	5,906	243	5,455	2,900	2	1,546	950	815	44	8[illegible]

Autoencoder

- An autoencoder is a type of neural network that ties to learn parameters that make the output as close to the input as possible.
- It contains a small hidden layer. This hidden layer compresses the data (called encoding).
- The process of reconstructing the input from the hidden layer is called decoding.

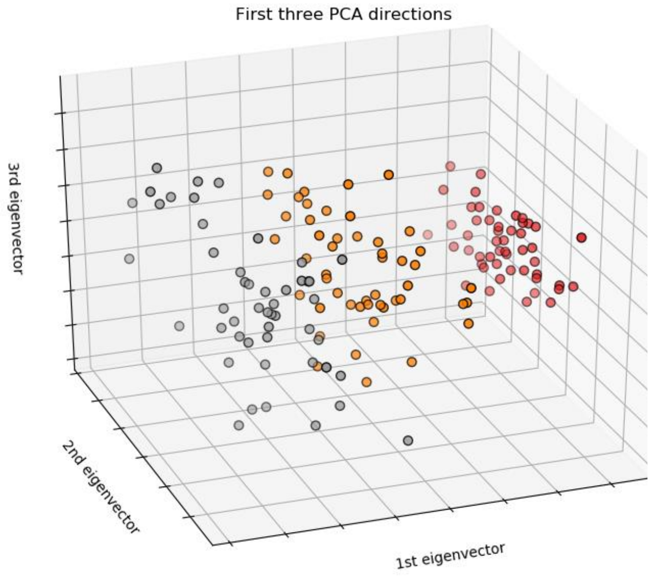
Autoencoder

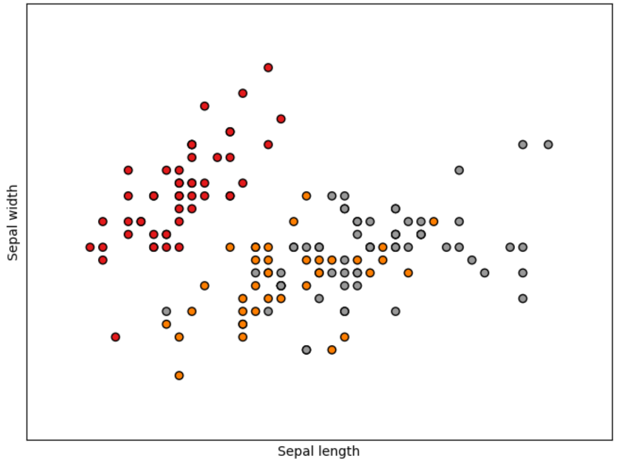


Encoder:
$h(X) = \mathrm{sigm}(b+Wx)$

Decoder:
$\bar{\bar{X}} = \mathrm{sigm}(b+W^{out}h(x))$

Autoencoder





Ronald Fisher

Sepal Length, Sepal Width, Petal Length and Petal Width

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets

Autoencoder

```
from autoencoder import Autoencoder
from sklearn import datasets

hidden_dim = 1
data = datasets.load_iris().data
input_dim = len(data[0])

ae = Autoencoder(input_dim, hidden_dim)

ae.train(data)

ae.test([[8,4,6,2]])
```

```
input [[8, 4, 6, 2]]
compressed [[0.72539085]]
reconstructed [[6.458398  2.8213227 5.43477   1.9124408]]
```

Why Autoencoder re-mentioned here?

https://www.kaggle.com/c/house-prices-advanced-regression-techniques

Practice Skills
- Creative feature engineering
- Advanced regression techniques

Data looks like…

- Close to the road
- Agriculture
- Universities
- Markets
- Built year
- Garage
- Crime rate
- House conditions
- Land slope
- Etc.