Regression

Prototyping with Deep Learning

Learning outcomes

After this lesson you will be able to:

- Understand regression principles
- Identify appropriate evaluation metrics for regression tasks
- Recognize applications of regression models in DL

What is regression?

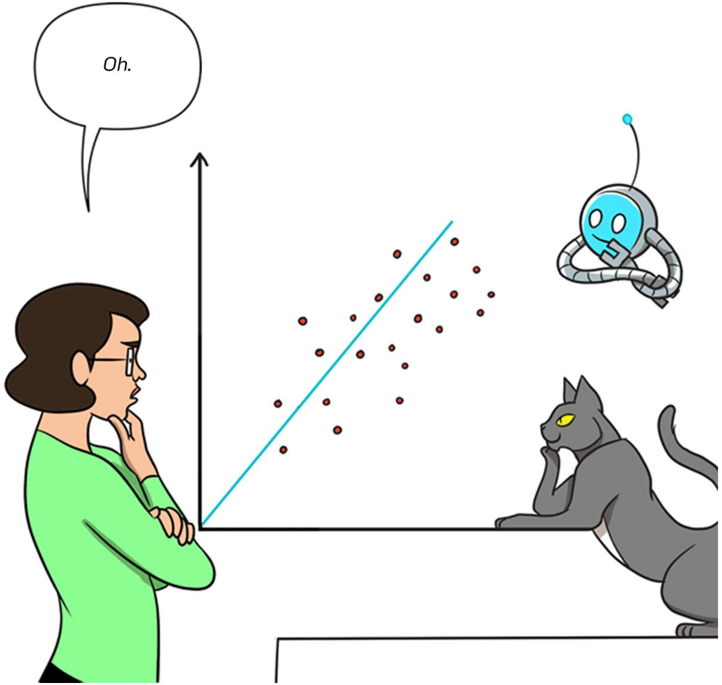
Predict a **continuous** value associated with a feature vector

Examples:

f(room) = temperature

f(trajectory) = time

…



https://cloud.google.com/products/ai/ml-comic-1/

Linear regression

Linear Regression: Single Variable

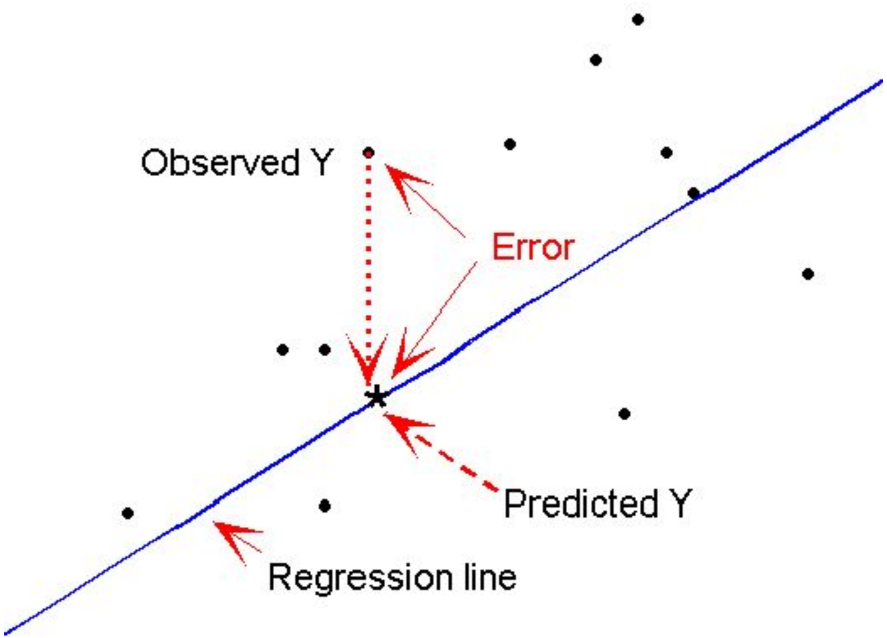
$$\underbrace{\boxed{\hat{y}}}_{\text{Predicted output}} = \underbrace{\beta_0 \; + \; \beta_1}_{\text{Coefficients}} \underbrace{\boxed{x}}_{\text{Input}} + \underbrace{\boxed{\epsilon}}_{\text{Error}}$$

Linear Regression: Multiple Variables

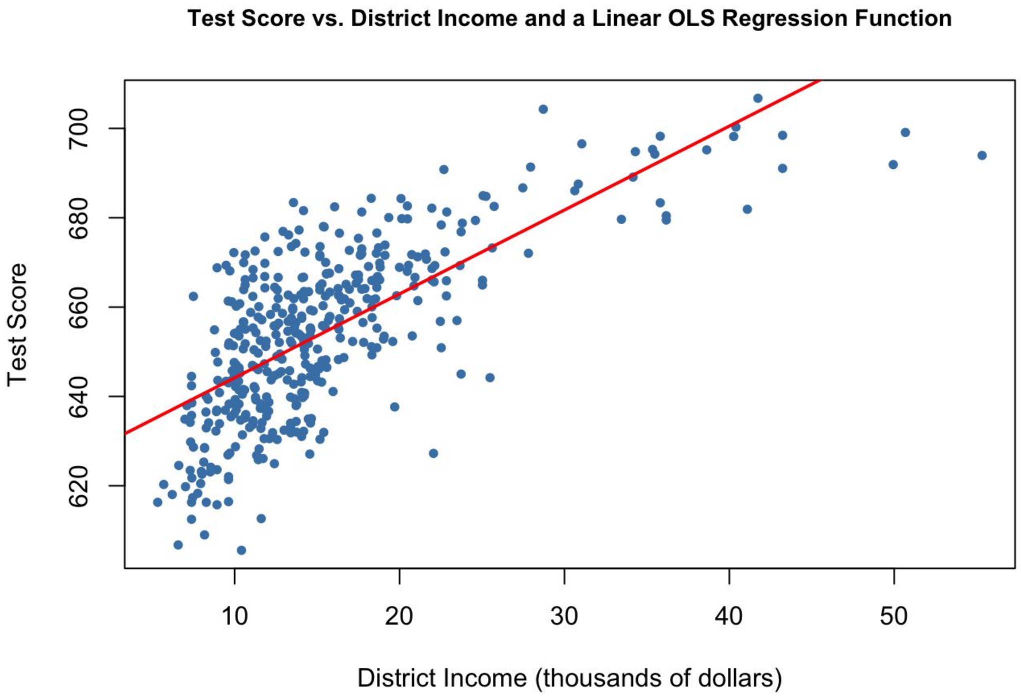
$$\boxed{\hat{y}} = \underbrace{\beta_0 \; + \; \beta_1}_{} \boxed{x_1} + \ldots + \underbrace{\beta_p}_{} \boxed{x_p} + \boxed{\epsilon}$$

https://www.dataquest.io/blog/understanding-regression-error-metrics/

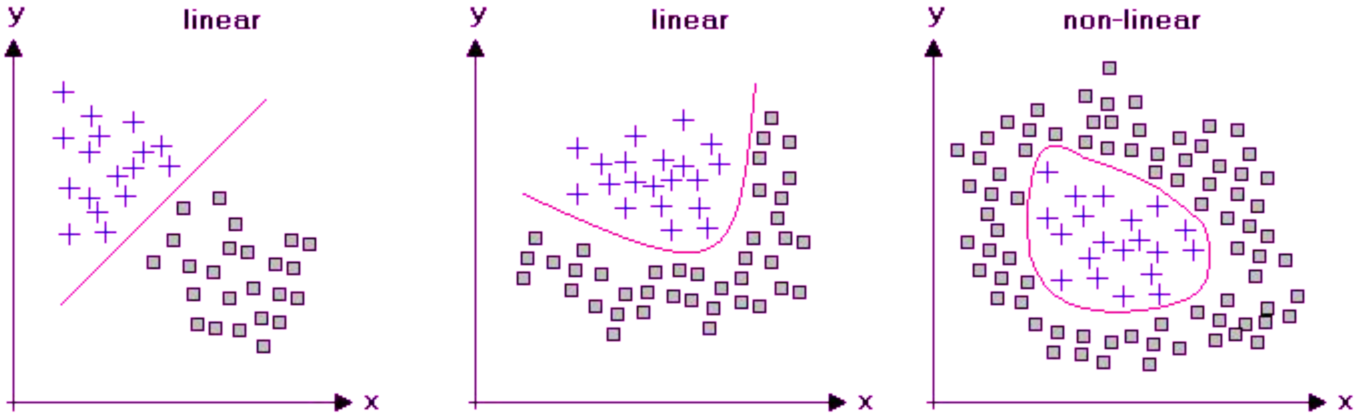
Residuals



Non-linear regression



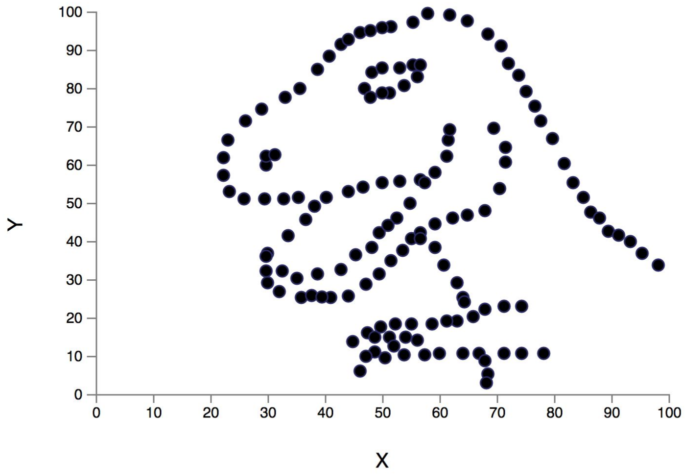
Non-linear regression with linear models



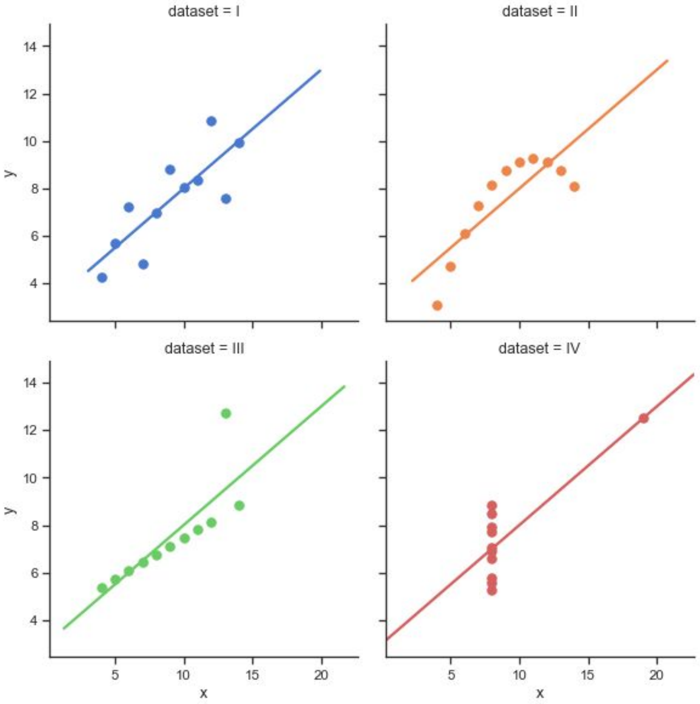
http://www.statistics4u.info/fundstat_eng/cc_linvsnonlin.html

https://blog.minitab.com/en/adventures-in-statistics-2/what-is-the-difference-between-linear-and-nonlinear-equations-in-regression-analysis

Anscombe’s quartet



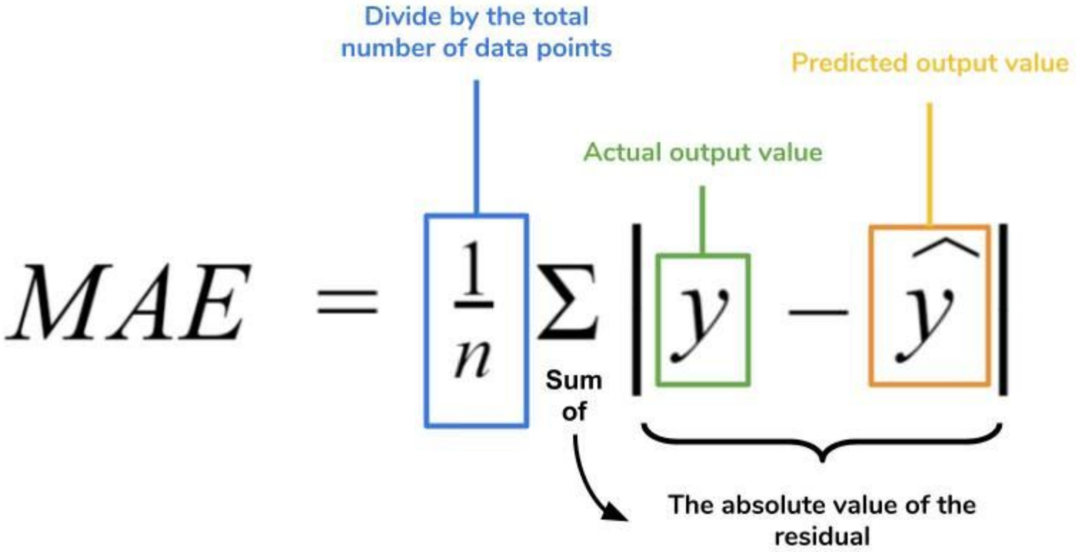
https://www.autodeskresearch.com/publications/samestats



https://www.heap.io/blog/anscombes-quartet-and-why-summary-statistics-dont-tell-the-whole-story

Evaluation metrics

Mean Average Error



Evaluation metrics

Mean Square Error

$$MSE = \frac{1}{n} \Sigma \underbrace{\left(y - \widehat{y} \right)^2}_{\text{The square of the difference between actual and predicted}}$$

Evaluation metrics

Mean Absolute Percentage Error

Multiplying by 100% converts to percentage

$$MAPE = \frac{\boxed{100\%}}{n} \Sigma \left| \frac{\overbrace{y - \hat{y}}^{\text{The residual}}}{\boxed{y}} \right|$$

Each residual is scaled against the actual value

Evaluation metrics

Mean Percentage Error

$$MPE = \frac{100\%}{n} \Sigma \left(\frac{y - \hat{y}}{y} \right)$$

Evaluation metrics

UNIVERSITÉ DU LUXEMBOURG

CASE 1: Evenly distributed errors

ID	Error	|Error|	Error^2
1	2	2	4
2	2	2	4
3	2	2	4
4	2	2	4
5	2	2	4
6	2	2	4
7	2	2	4
8	2	2	4
9	2	2	4
10	2	2	4

MAE	RMSE
2.000	2.000

CASE 2: Small variance in errors

ID	Error	|Error|	Error^2
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	3	3	9
7	3	3	9
8	3	3	9
9	3	3	9
10	3	3	9

MAE	RMSE
2.000	2.236

CASE 3: Large error outlier

ID	Error	|Error|	Error^2
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	20	20	400

MAE	RMSE
2.000	6.325

https://medium.com/human-in-a-machine-world/e60ac3bde13d

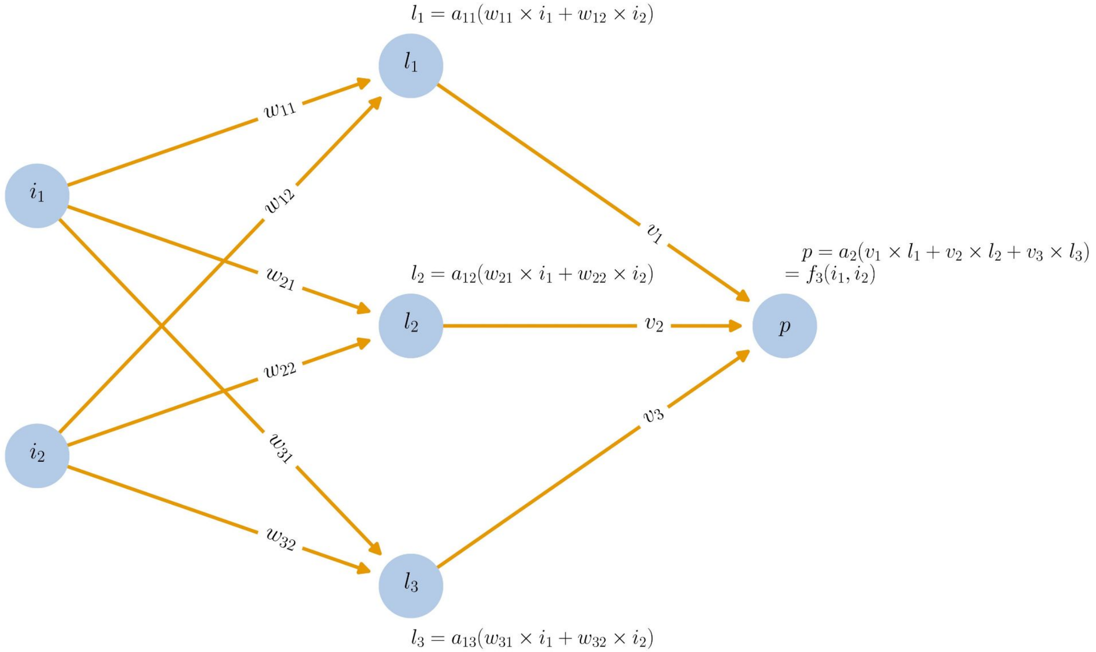
Evaluation metrics

UNIVERSITÉ DU LUXEMBOURG

Acronym	Name	Residual Operation	Robust To Outliers
MAE	Mean Absolute Error	Abs. diff	yes
MSE	Mean Squared Error	Squared diff	no
RMSE	Root Mean Squared Error	Squared diff	no
MAPE	Mean Absolute Percentage Error	Abs. diff	yes
MPE	Mean Percentage Error	Raw diff	yes

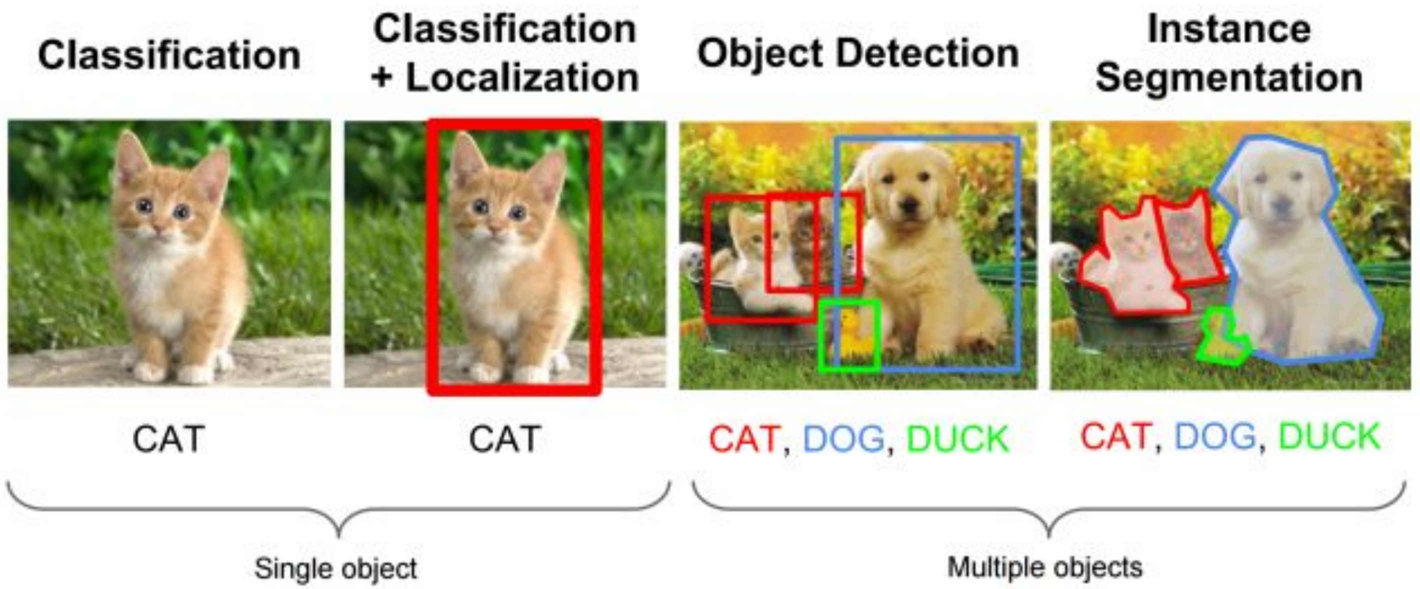
https://towardsdatascience.com/cdc5703d242d

A simple regression model architecture



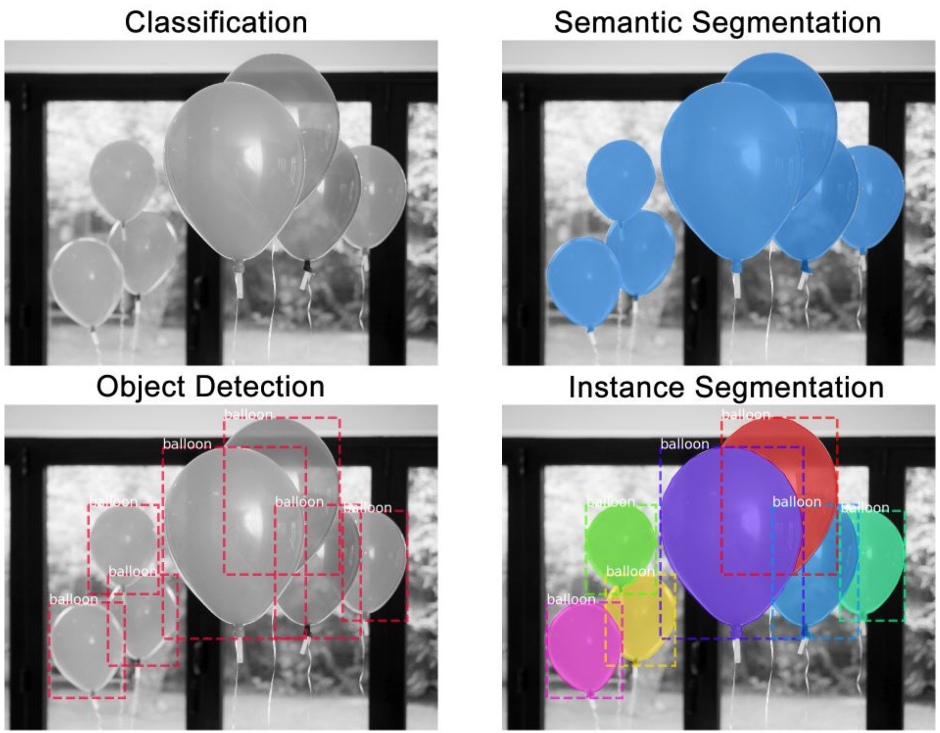
https://towardsdatascience.com/68881590760e

There is more to regression!



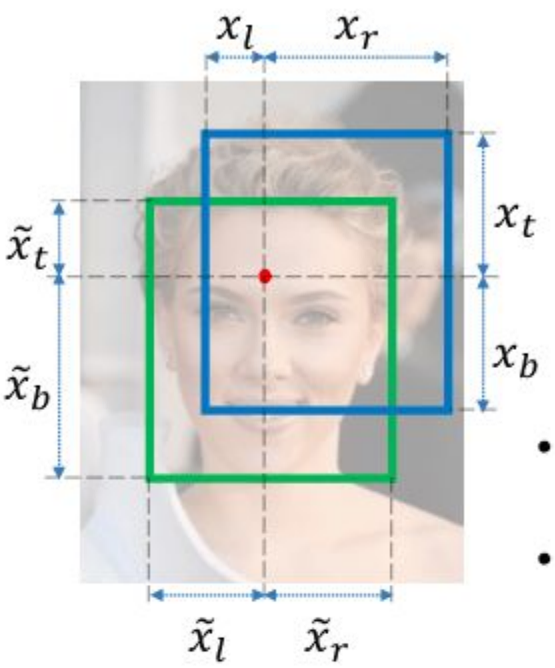
https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/object_localization_and_detection.html

Practical use cases



https://engineering.matterport.com/7c761e238b46

Evaluation metric: Intersection over Union



$\square$ Ground truth: $\tilde{x} = (\tilde{x}_t, \tilde{x}_b, \tilde{x}_l, \tilde{x}_r)$

$\square$ Prediction: $x = (x_t, x_b, x_l, x_r)$

- $\ell_2 \ \ loss = ||\square - \square||_2^2$
- $IoU \ loss = -\ln \dfrac{Intersection(\square, \square)}{Union(\square, \square)}$

Classic architectures

FCN (2014)

DeconvNet (2015)

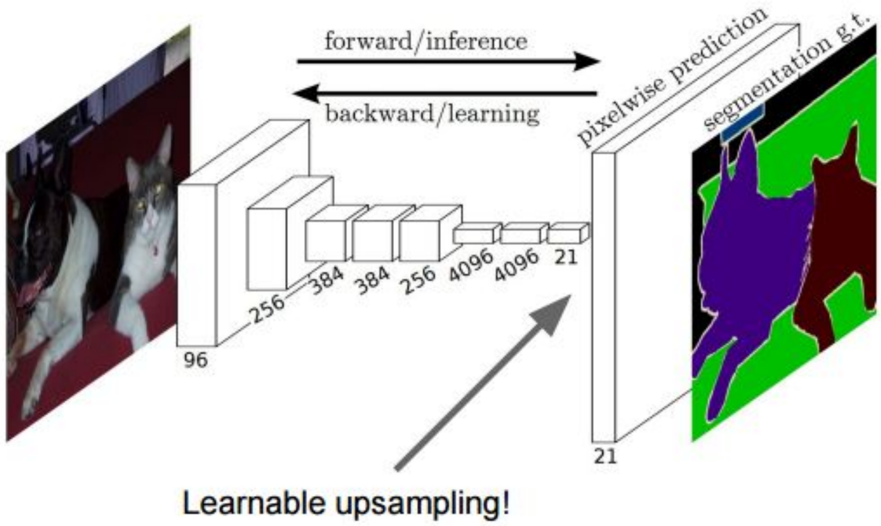
U-Net (2015)

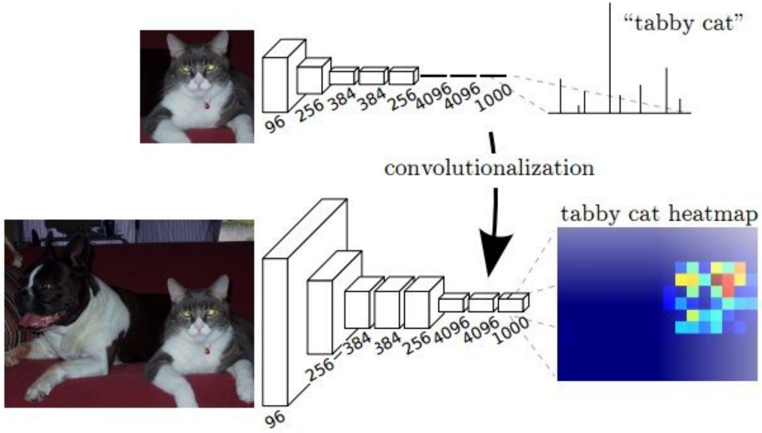
Fast R-CNN (2014) and Faster R-CNN (2016)

Mask R-CNN (2017)

YOLO (2016)

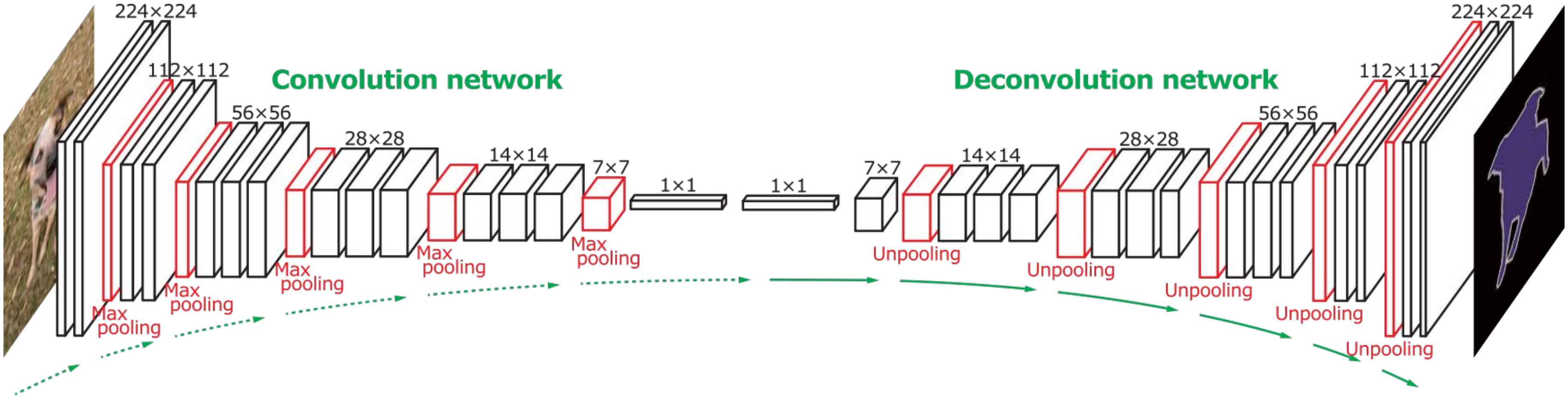
Classic architecture: **FCN**





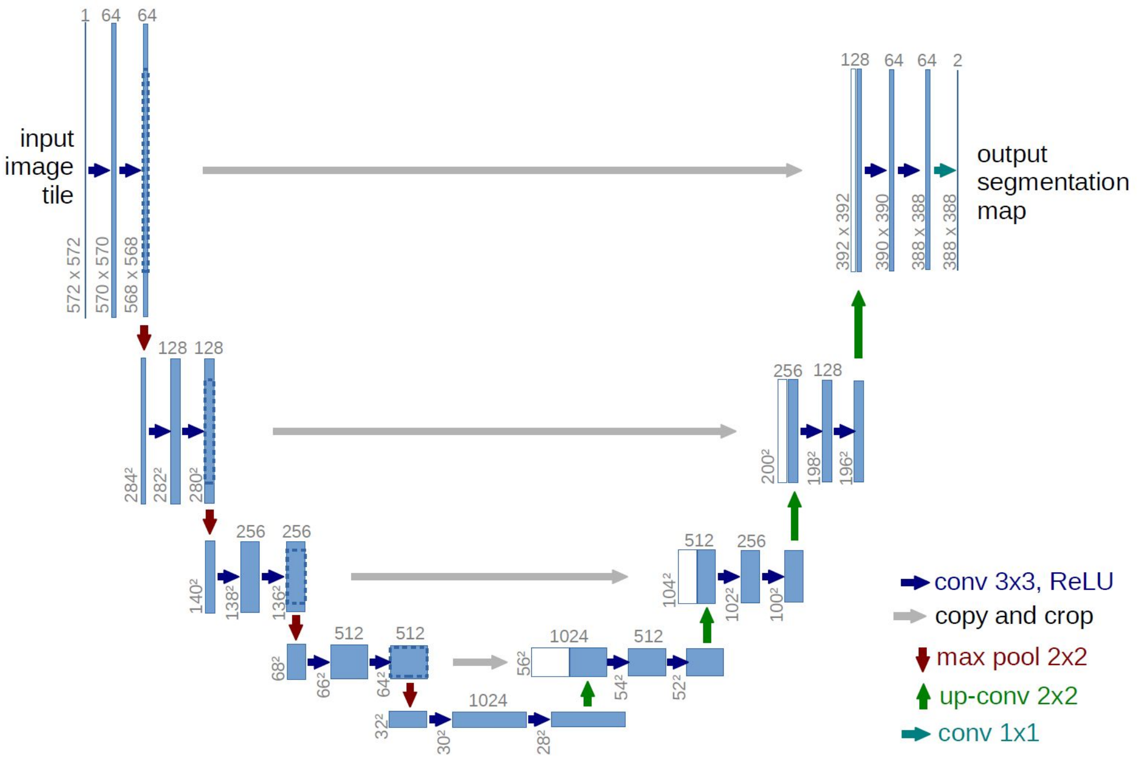
https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/image_segmentation

Classic architecture: **DeconvNet**



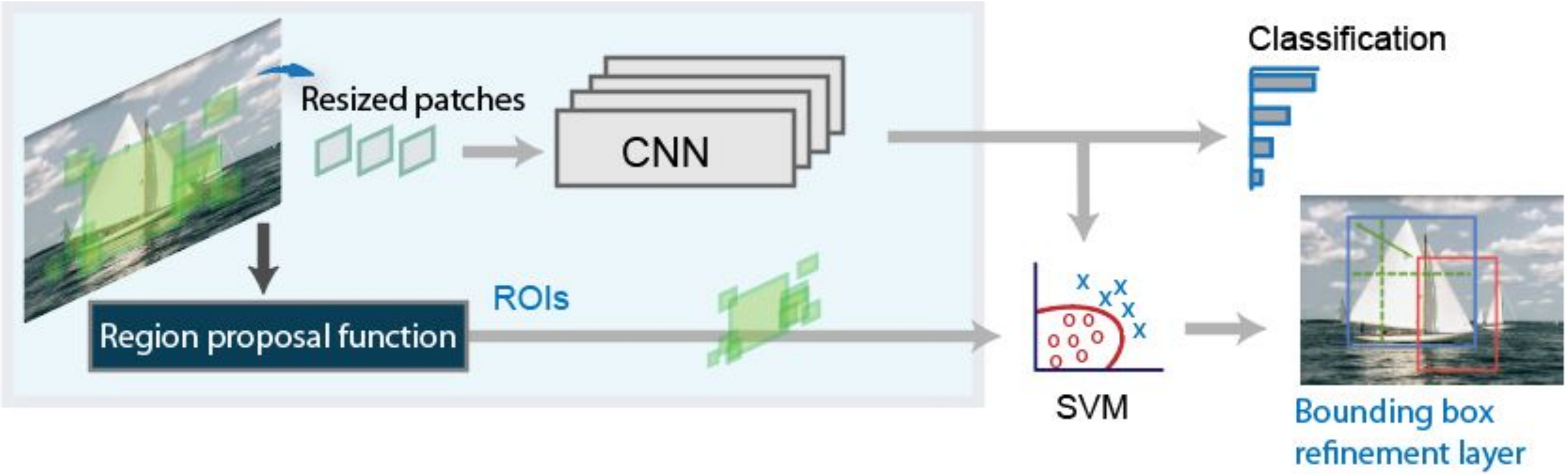
https://towardsdatascience.com/55cf8a6e380e

Classic architecture: **U-Net**



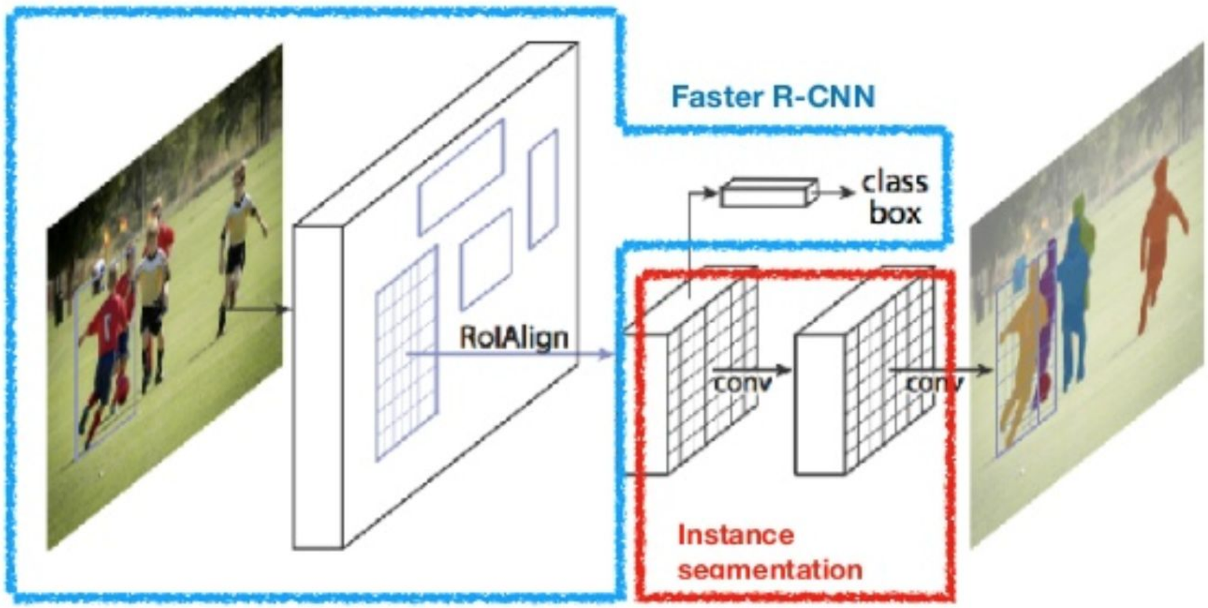
https://heartbeat.fritz.ai/ff17f6e4c1cf

Classic architecture: **Fast(er) R-CNN**



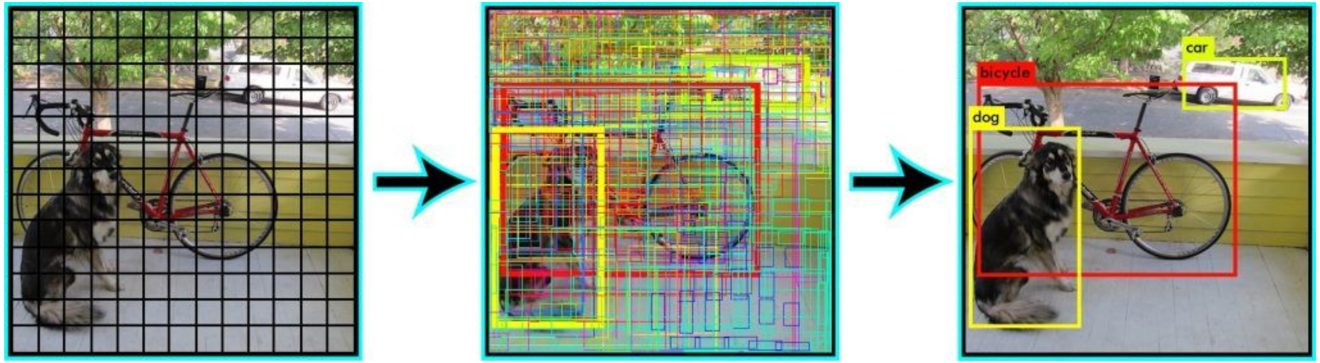
https://www.mathworks.com/help/vision/ug/getting-started-with-r-cnn-fast-r-cnn-and-faster-r-cnn.html

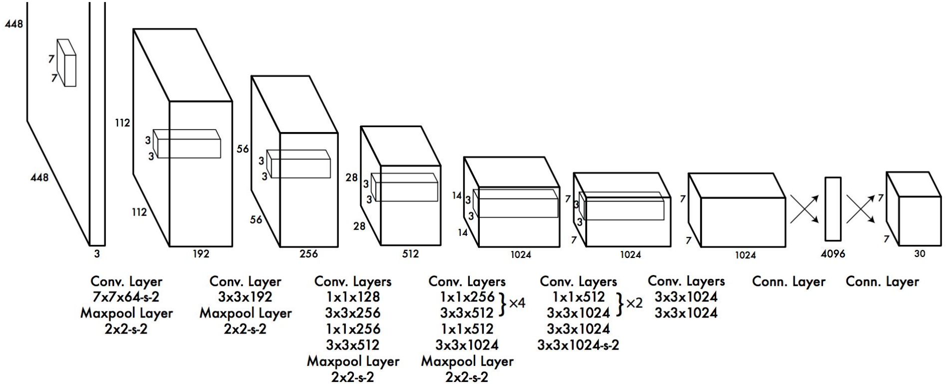
Classic architecture: **Mask R-CNN**



https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html

Classic architecture: **YOLO**





https://dzone.com/articles/understanding-object-detection-using-yolo

Faster R-CNN vs YOLO

https://www.youtube.com/watch?v=Vrx2rKt1xSc

https://www.youtube.com/watch?v=Vrx2rKt1xSc