

Introduction to Machine Learning

Image recognition in Python with Tensorflow

Esten H. Leonardsen

22.05.23



Introduction

Theory session (~1 hour):

- What is a statistical learning model?
- What is a loss function?
- How do we train a model?
- How does a deep neural network work?
- How does a convolutional neural network process images?
- What is transfer learning?
- What is overfitting, and how do we avoid it?

Practical session (~2 hours):

1. Configure a Python-environment containing Tensorflow
2. Set up a pretrained convolutional neural network for prediction
3. Fit a flower classifier using transfer learning
4. Improve the flower classifier

Introduction

Bear



Moose



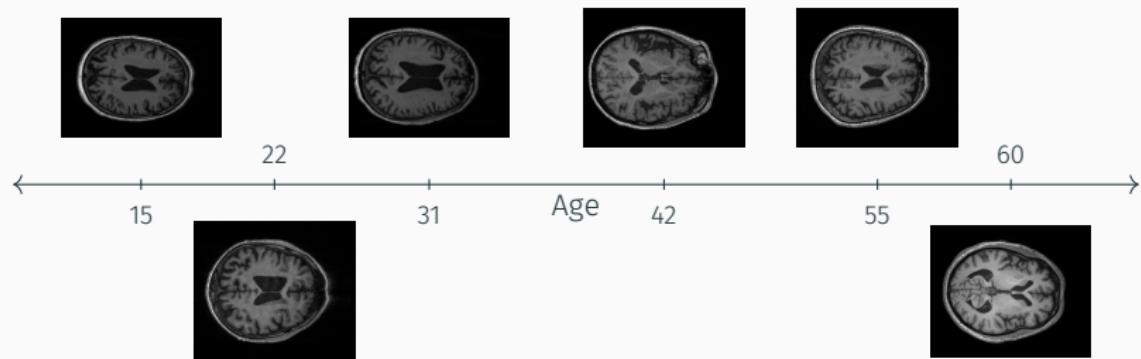
Bobcat

Introduction



Crack: 95%

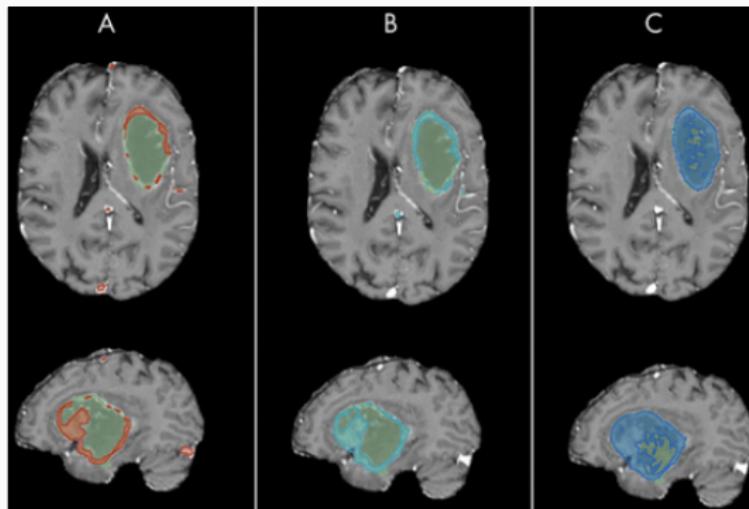
Introduction



Introduction

Pose estimation

Introduction



Eijgelaar, Roelant S., et al. "Robust deep learning-based segmentation of glioblastoma on routine clinical MRI scans using sparsified training." Radiology: Artificial Intelligence 2.5 (2020)

Introduction



Statistical learning

Statistical learning: Models

Muligheten markeds

Venskaper Ny annonse Meldinger Min profil

Søgesstatus:

- Til salgs (910)
- Søgt siden 3 dager (9)
- Kommer for salg (1)

Nyhetsbrev:

- Ønsket biling (22)
- Nyhetsbrev (33)

Prisprøving:

Rader: 70 kr

Antall rom:

Rader: 70 kr

Feltprøving per måned:

Rader: 70 kr

Glemt:

Rader: 70 kr

Antall soverom:

1+ 2+ 3+ 4+ 5+

Hjemprå:

Rader: 70 kr

Boligtype:

- Lefteghet (514)
- Garage/Parkerings (11)
- Høkerhus (2)
- Erlending (1)
- Tunemråsing (1)
- Oppgradering (2)
- Andre (2)

Vis alle

Interne:

- Akjøp (5)
- Andre (134)
- For salg/Leiesalg (22)

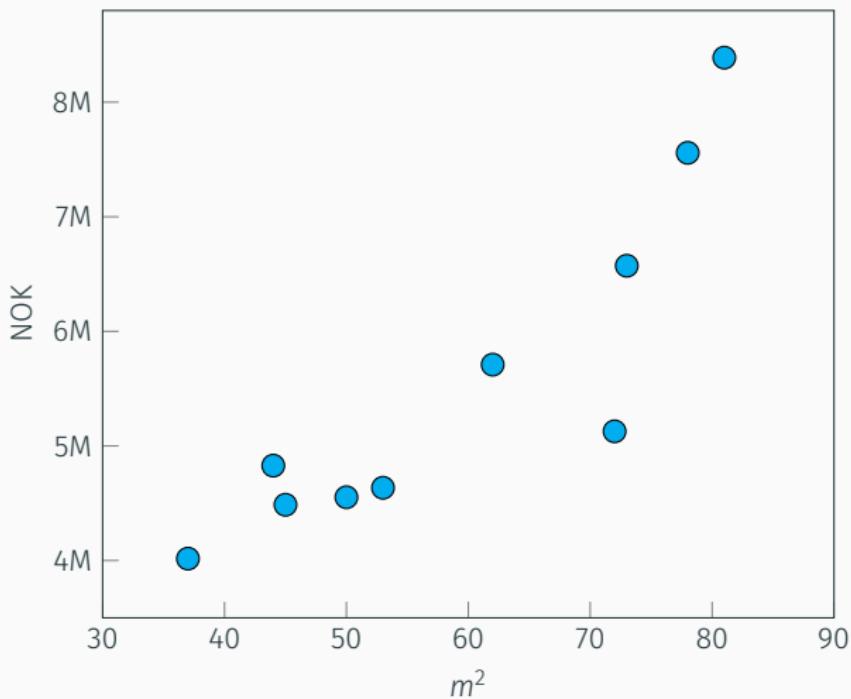
Leren - Titallaten 3-roms med god planløsning og flott beliggenhet - Solrik..
73 m² 6 400 000 kr
Til salgs i 572 200 kr • Flere etasjer • 3 rom (alle) • Leiegjeld: 2 vinterpris - 22 januar kl 13:00
Lærveien 41 G, Oslo

SP Schalit & Partner Grunerløkka
Rodenrikkha / Grunerløkka - Lys, luftig 3-roms hjemmet med innre gjerd...
67 m² 5 500 000 kr
Til salgs i 5 127 kr • Flere etasjer • 3 rom (alle) • Leiegjeld: 2 vinterpris - 22 januar kl 13:00
Rodeløkka gate 11, Oslo

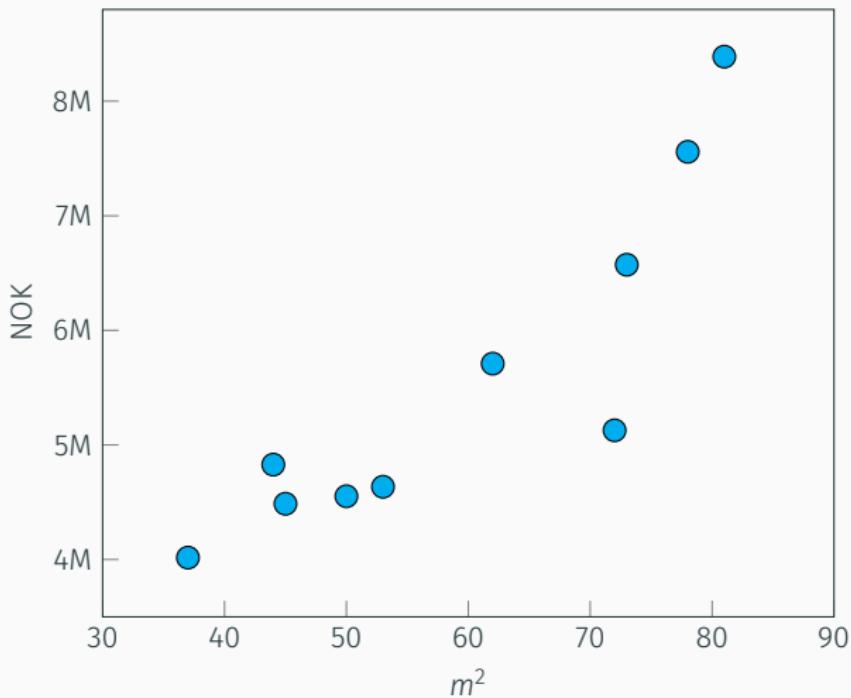
Statistical learning: Models

m^2	Price
72	5.127.379
50	4.552.170
45	4.486.654
62	5.709.276
53	4.634.912
81	8.388.570
44	4.828.170
78	7.557.770
37	4.016.520
73	6.572.351

Statistical learning: Models

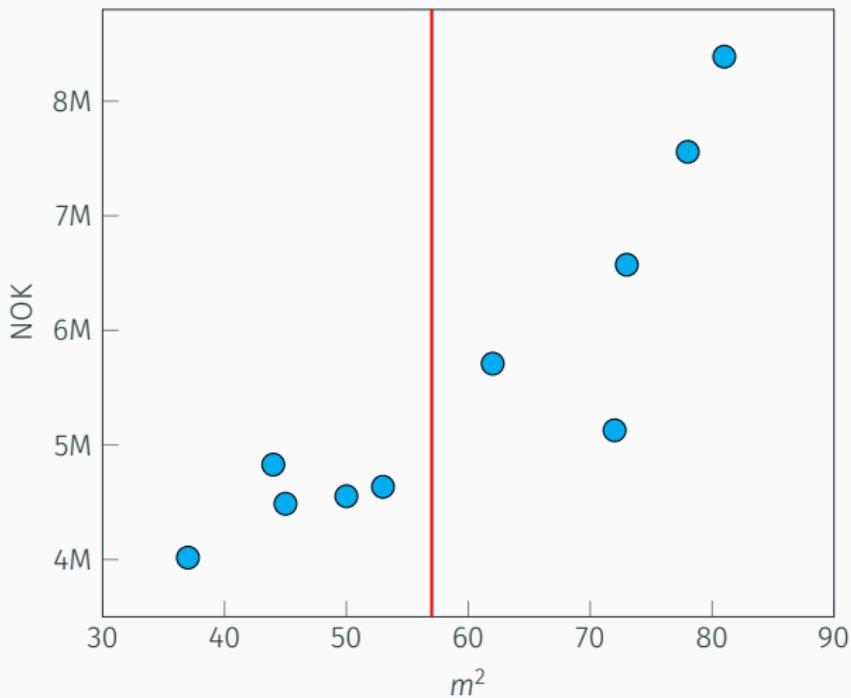


Statistical learning: Models



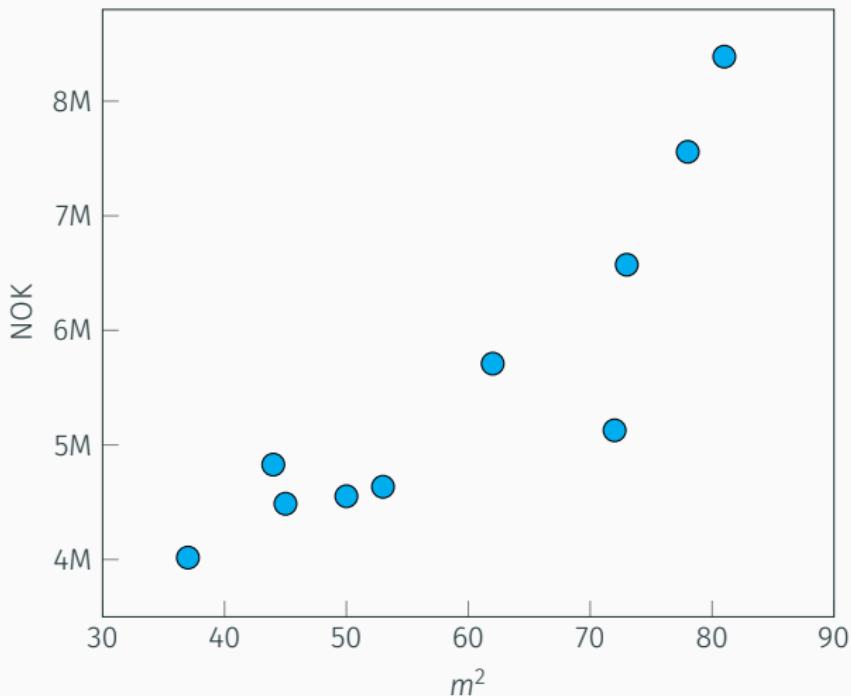
$$\hat{y} = f(x)$$

Statistical learning: Models



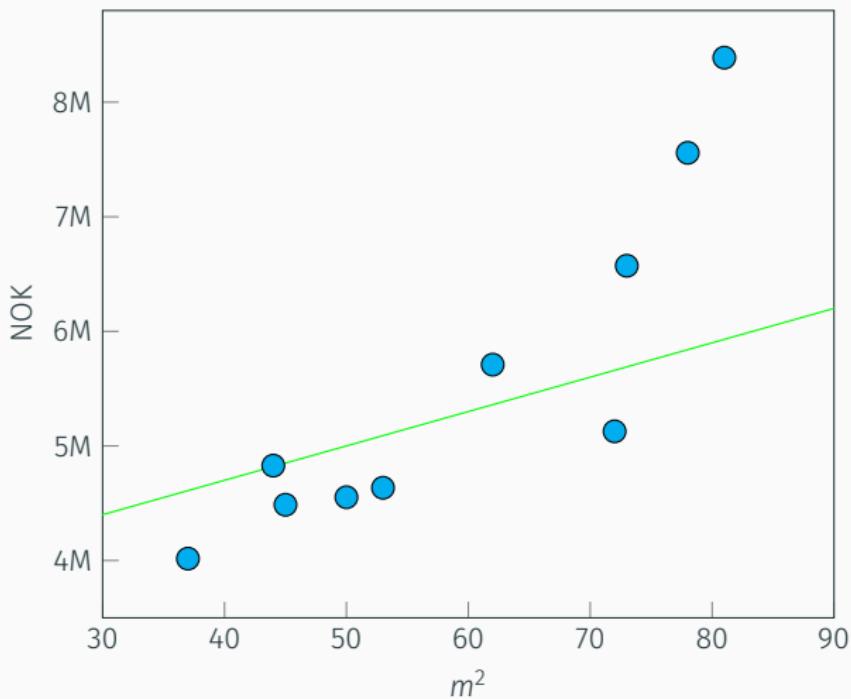
$$\hat{y} = f(57)$$

Statistical learning: Models



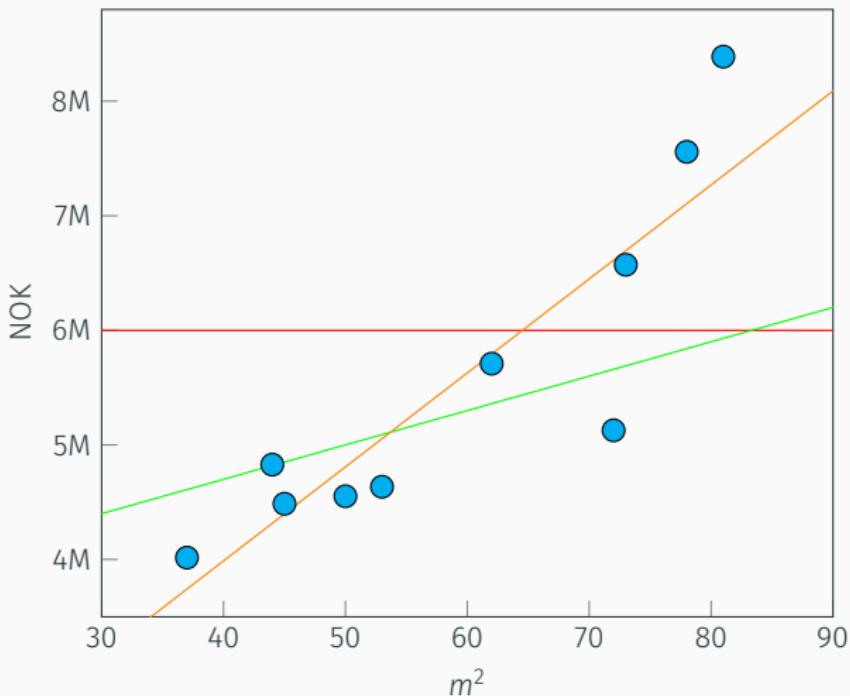
$$\hat{y} = wx + b$$

Statistical learning: Models



$$\hat{y} = 30000x + 3500000$$

Statistical learning: Models

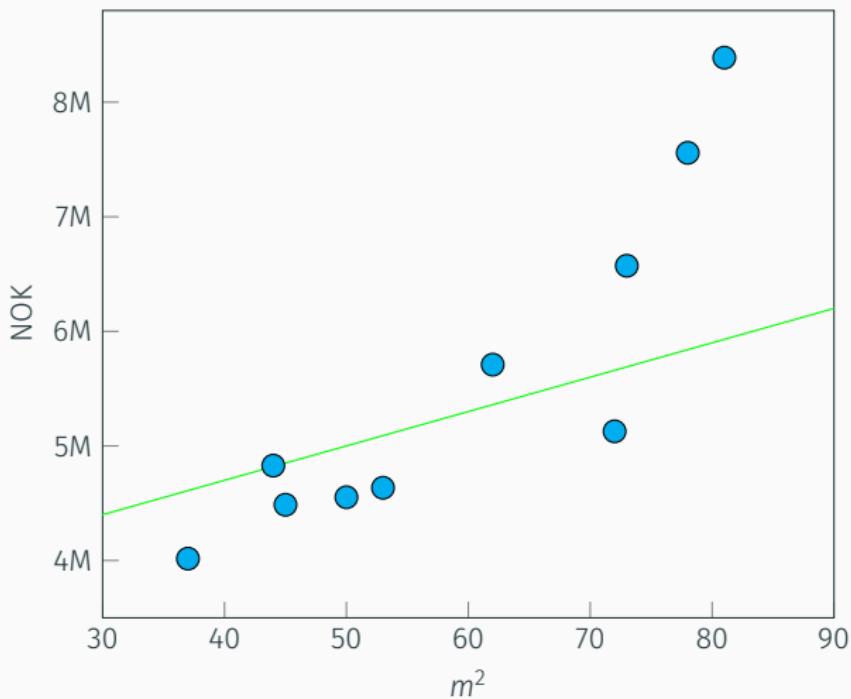


$$\hat{y} = 0x + 6000000$$

$$\hat{y} = 30000x + 3500000$$

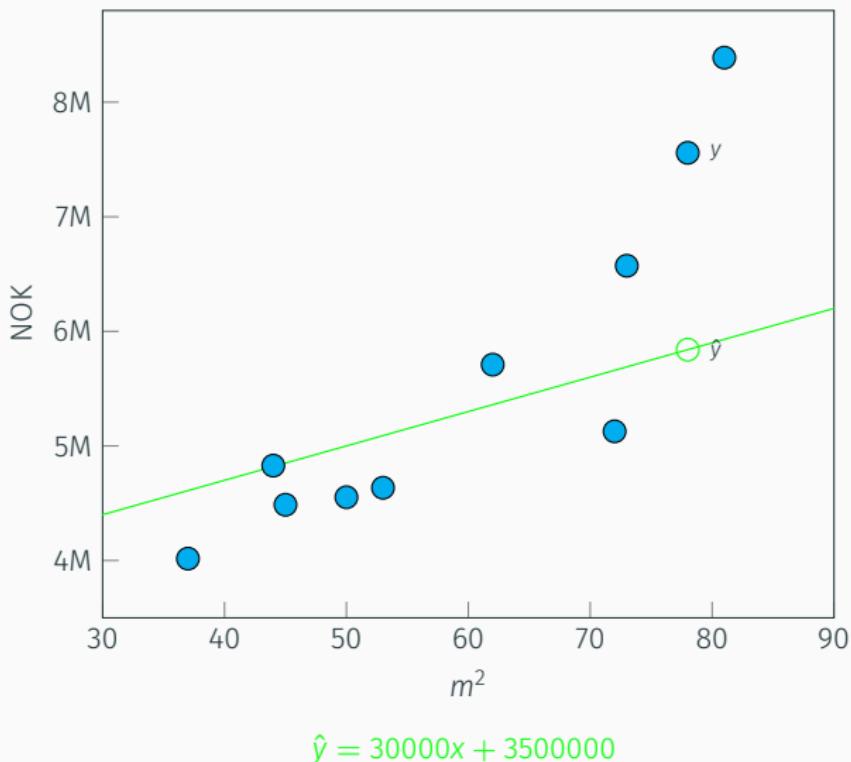
$$\hat{y} = 82031x + 706495$$

Statistical learning: Loss functions

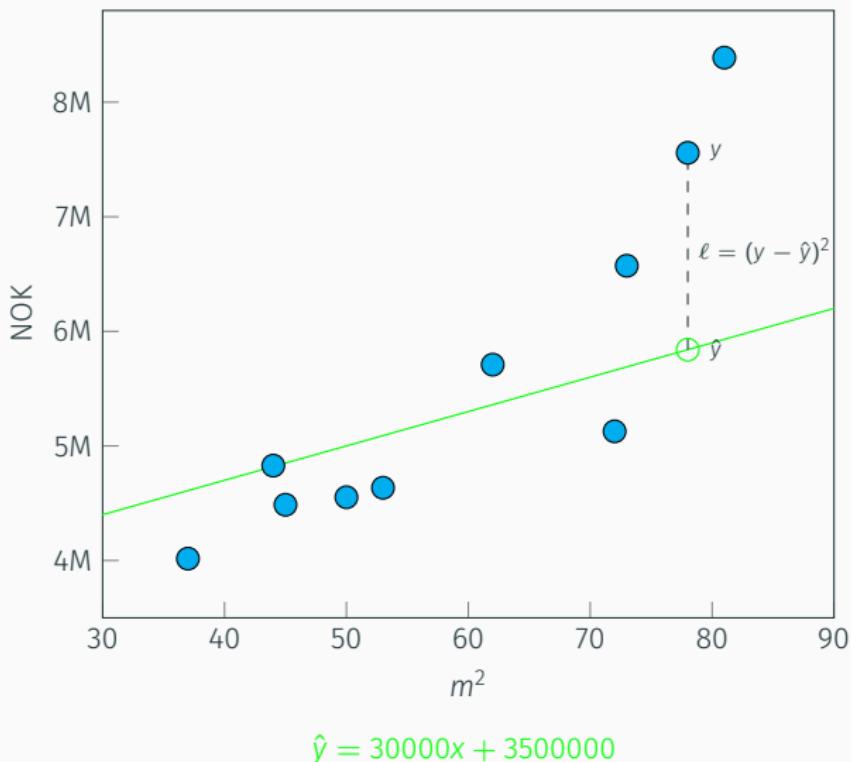


$$\hat{y} = 30000x + 3500000$$

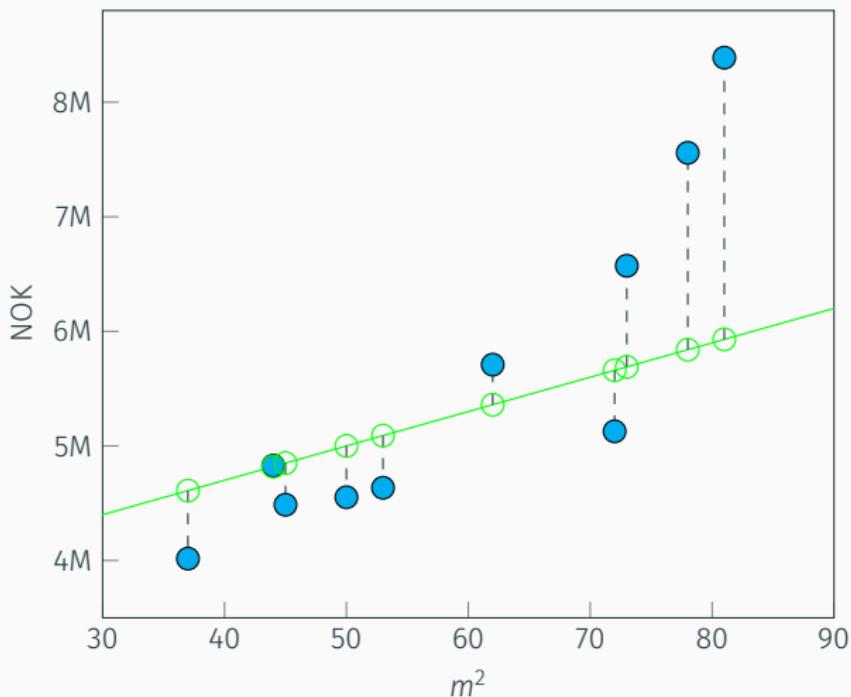
Statistical learning: Loss functions



Statistical learning: Loss functions



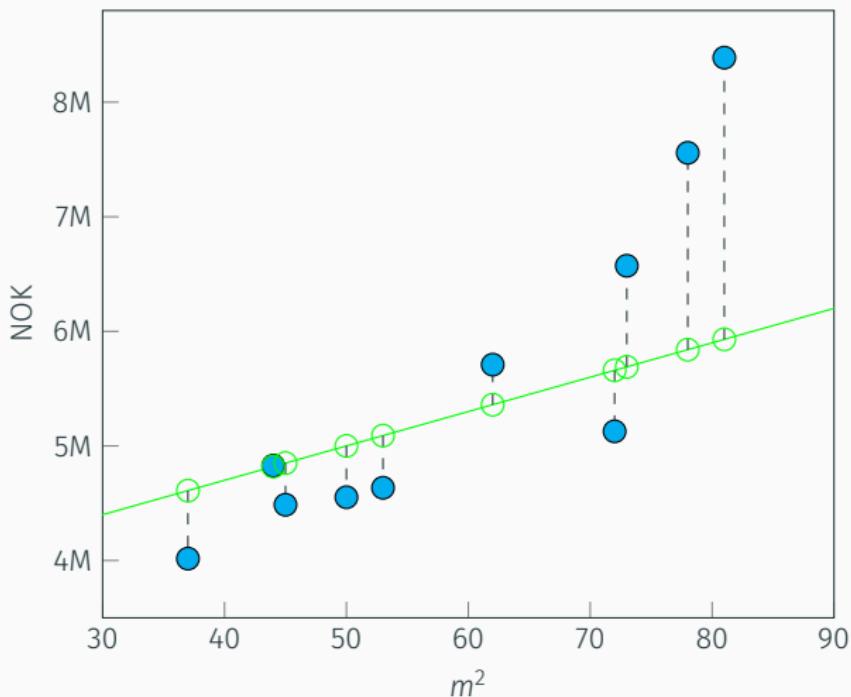
Statistical learning: Loss functions



$$\hat{y} = 30000x + 3500000$$

$$\ell = \sum(y - \hat{y})^2$$

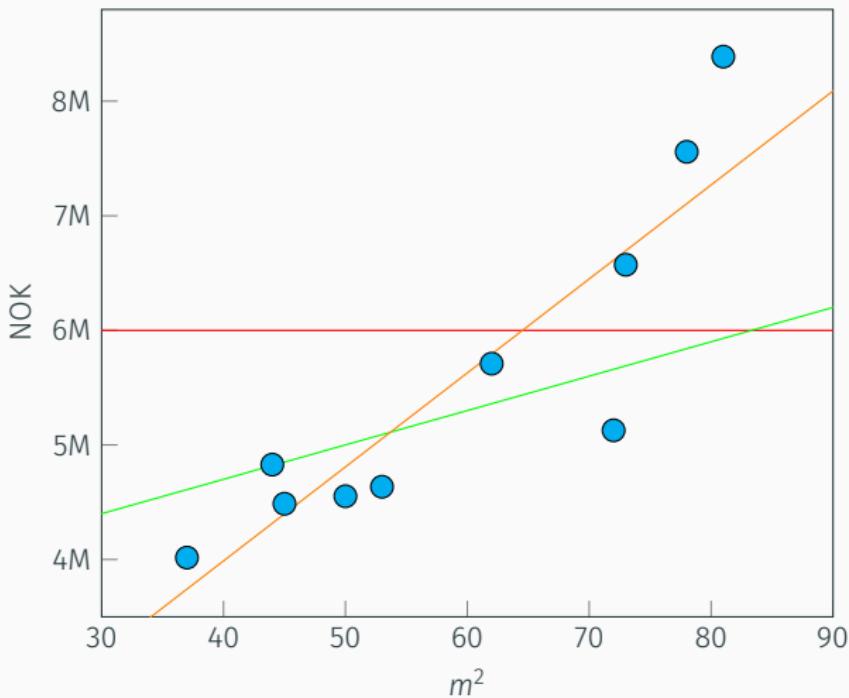
Statistical learning: Loss functions



$$\hat{y} = 30000x + 3500000$$

$$\ell = 1.10 \times 10^{13}$$

Statistical learning: Loss functions

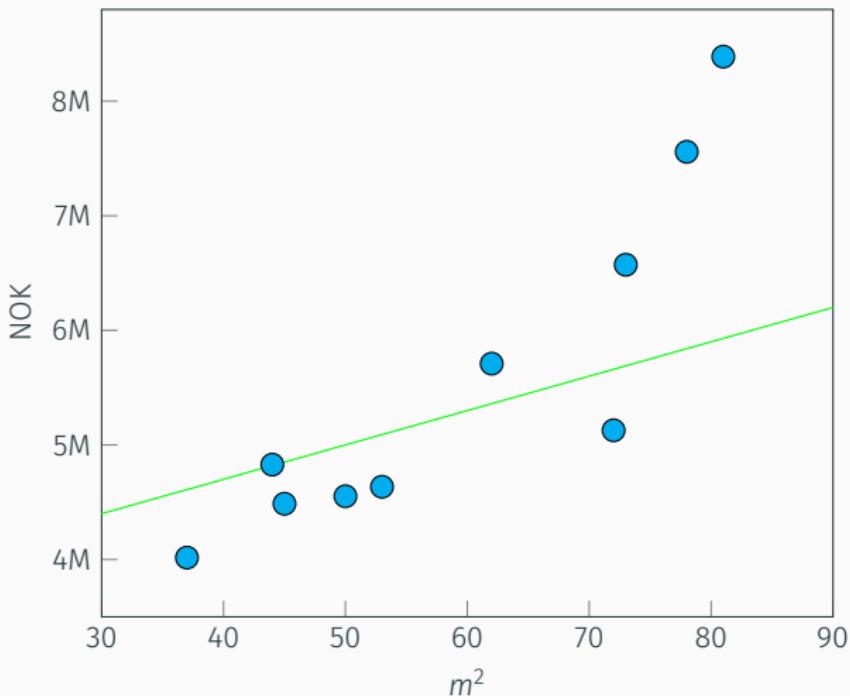


$$\hat{y} = 0x + 6000000$$
$$\ell = 2.08 \times 10^{13}$$

$$\hat{y} = 30000x + 3500000$$
$$\ell = 1.10 \times 10^{13}$$

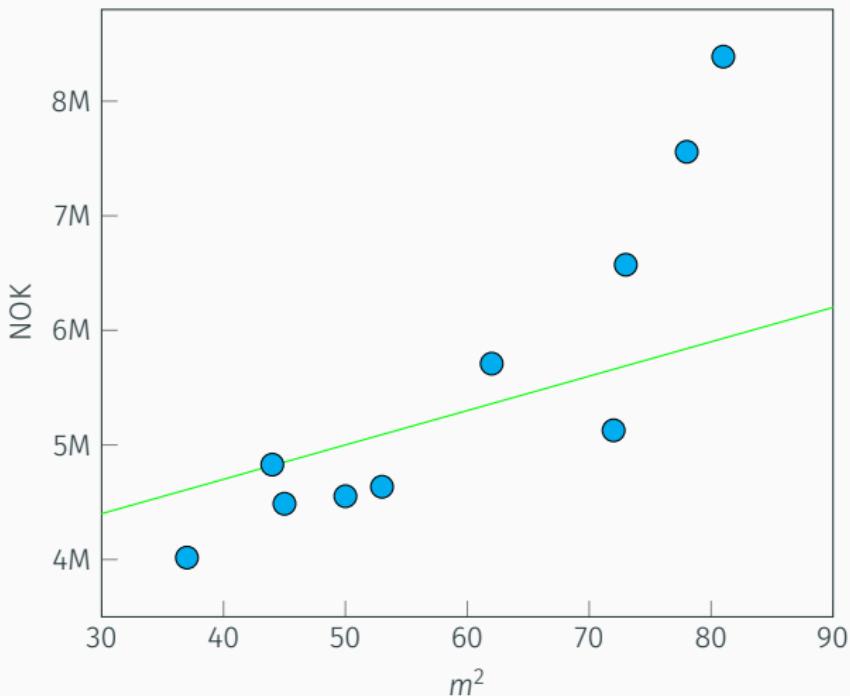
$$\hat{y} = 82031x + 706495$$
$$\ell = 4.09 \times 10^{12}$$

Statistical learning: Training



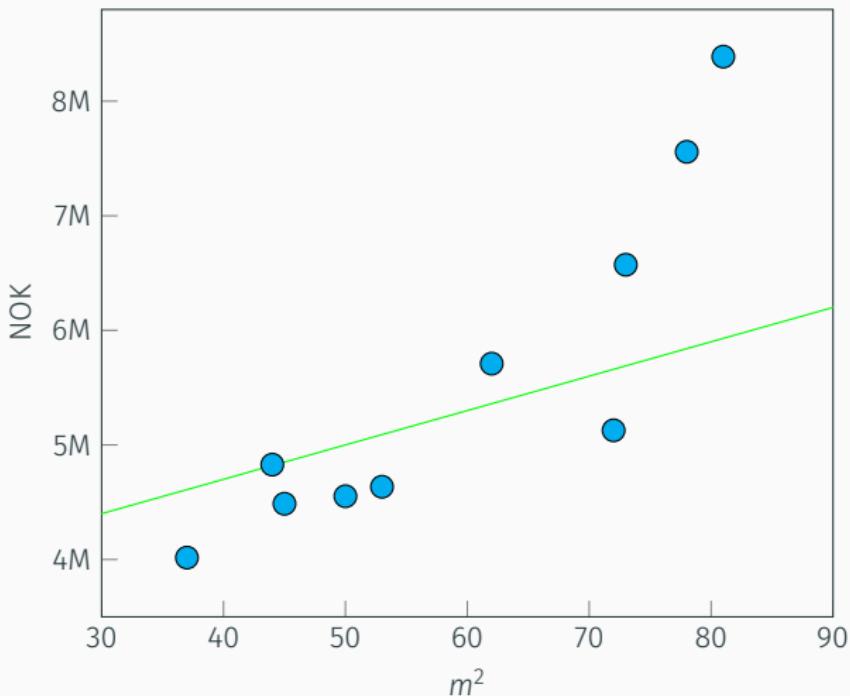
$$\hat{y} = wx + b$$
$$\ell = \sum(y - \hat{y})^2$$

Statistical learning: Training



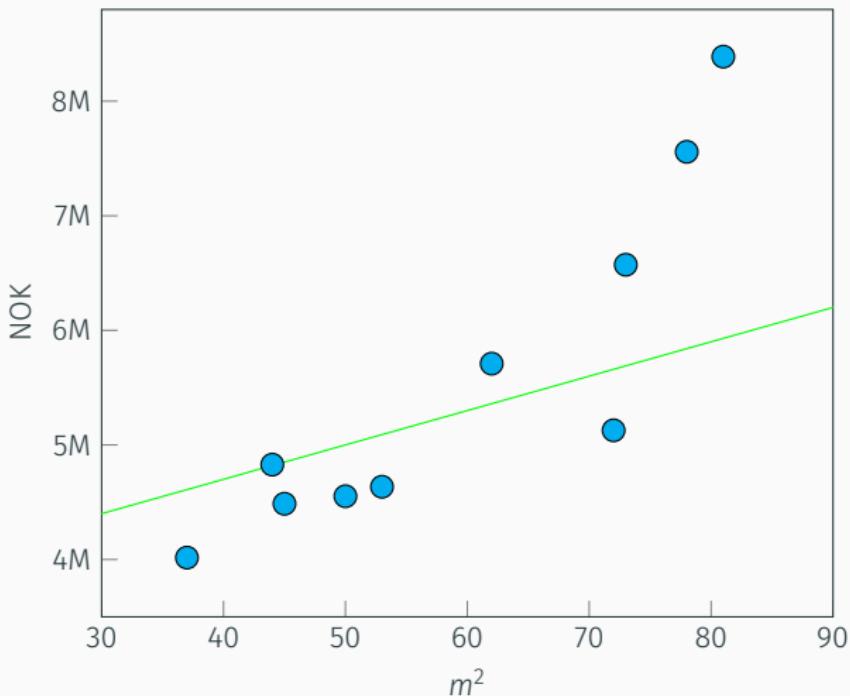
$$\hat{y} = wx + b$$
$$\ell = \sum(y - \hat{y})^2$$

Statistical learning: Training



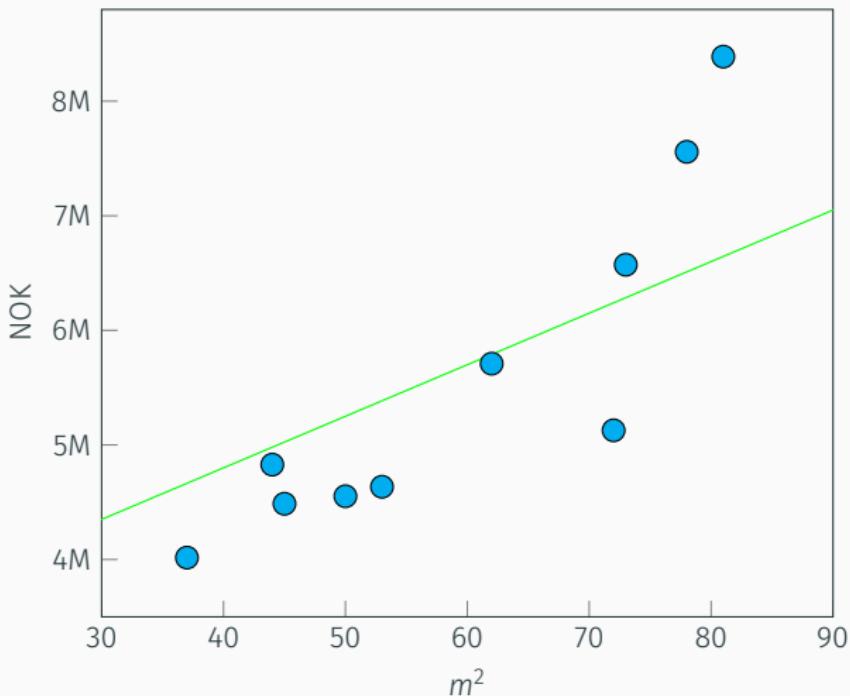
$$\ell = \sum(y - (wx + b))^2$$

Statistical learning: Training



$$1.10 \times 10^{13} = \sum(y - (30000x + 3500000))^2$$

Statistical learning: Training

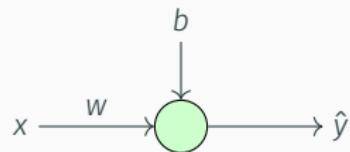


$$7.24 \times 10^{12} = \sum(y - (45000x + 3000000))^2$$

Demo

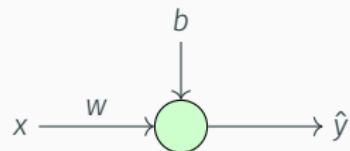
Artificial neural networks

Artificial neural networks: Building blocks

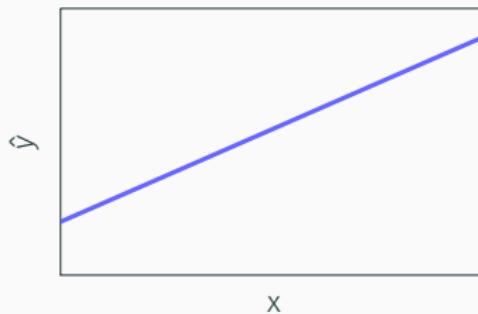


$$\hat{y} = wx + b$$

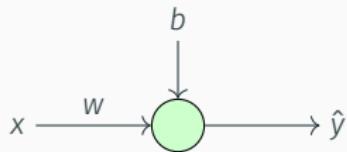
Artificial neural networks: Building blocks



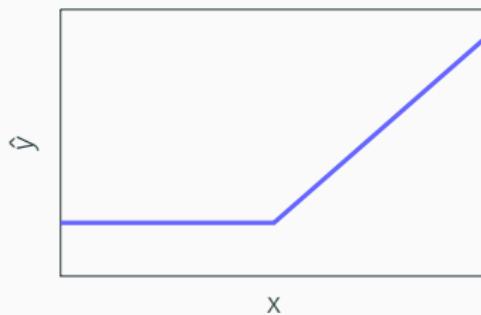
$$\hat{y} = wx + b$$



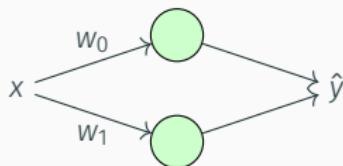
Artificial neural networks: Building blocks



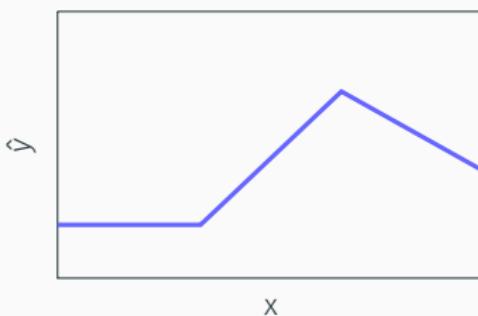
$$\hat{y} = \max(0, wx + b)$$



Artificial neural networks: Building blocks



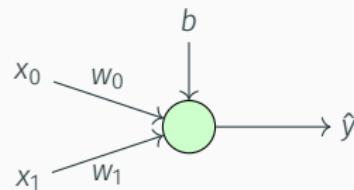
$$\hat{y} = \max(0, w_0x + b_0) + \max(0, w_1x + b_1)$$



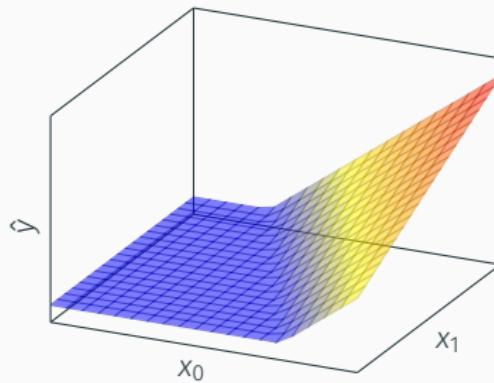
Artificial neural networks: Building blocks

"Any relationship that can be described with a polynomial function can be approximated by a neural network with a single hidden layer."

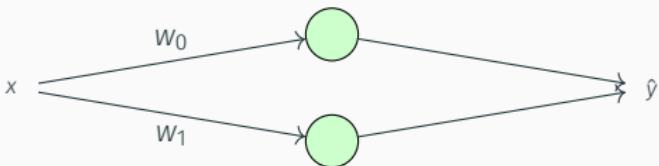
Artificial neural networks: Building blocks



$$\hat{y} = \max(0, w_0x_0 + w_1x_1 + b)$$

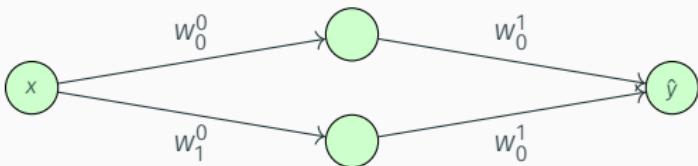


Artificial neural networks: Building blocks



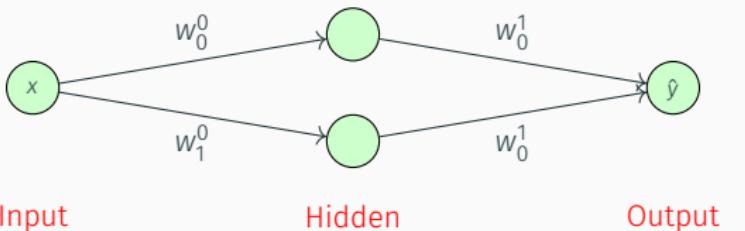
$$\hat{y} = \max(0, w_0x + b_0) + \max(0, w_1x + b_1)$$

Artificial neural networks: Building blocks



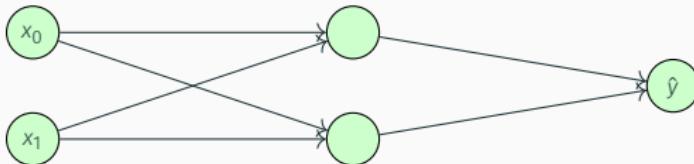
$$\hat{y} = \max(0, w_{0,0}^1 * \max(0, w_{0,0}^0 * x + b_{0,0}) + w_{1,0}^1 * \max(0, w_{0,1}^0 * x + b_{1,0}) + b_1)$$

Artificial neural networks: Building blocks



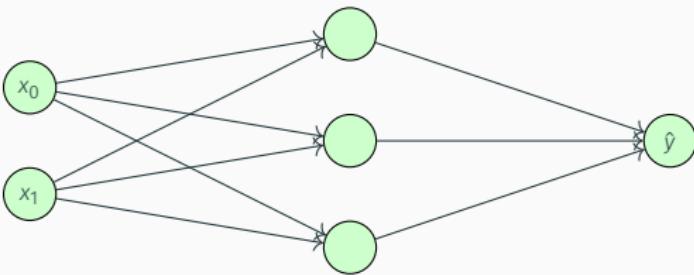
$$\hat{y} = \max(0, w_{0,0}^1 * \max(0, w_{0,0}^0 * x + b_{0,0}) + w_{1,0}^1 * \max(0, w_{0,1}^0 * x + b_{1,0}) + b_1)$$

Artificial neural networks: Building blocks



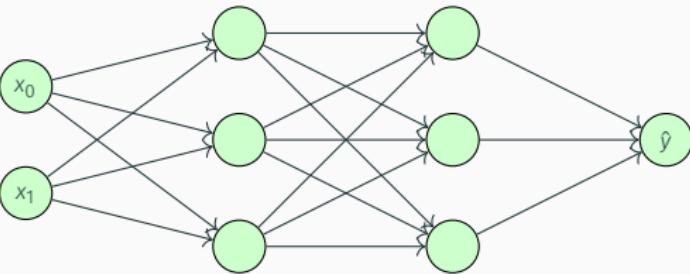
$$\begin{aligned}\hat{y} = \max(0, & w_{0,0}^1 * \max(0, w_{0,0}^0 * x_0 + w_{1,0}^0 * x_1 + b_{0,0}) + \\ & w_{1,0}^1 * \max(0, w_{0,1}^0 * x_0 + w_{1,1}^0 * x_1 + b_{0,1}) + \\ & b_1)\end{aligned}$$

Artificial neural networks: Building blocks



$$\begin{aligned}\hat{y} = \max(0, & w_{0,0}^1 * \max(0, w_{0,0}^0 * x_0 + w_{1,0}^0 * x_1 + b_{0,0}) + \\ & w_{1,0}^1 * \max(0, w_{0,1}^0 * x_0 + w_{1,1}^0 * x_1 + b_{0,1}) + \\ & w_{2,0}^1 * \max(0, w_{0,2}^0 * x_0 + w_{1,2}^0 * x_1 + b_{0,2}) + \\ & b_1)\end{aligned}$$

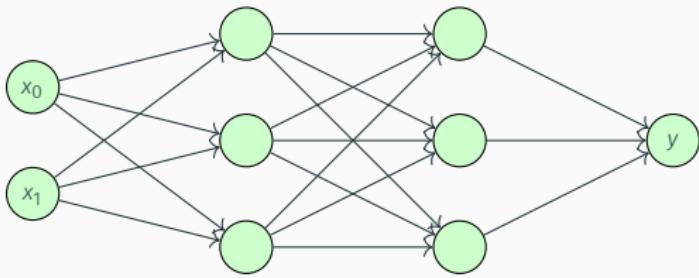
Artificial neural networks: Building blocks



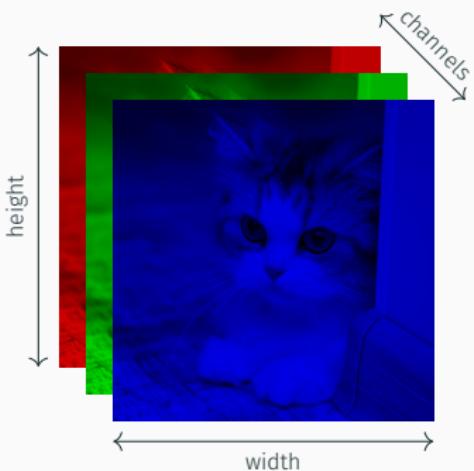
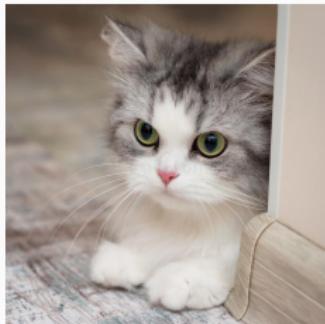
$$\begin{aligned}\hat{y} = & \max(0, w_{0,0}^2 * \max(0, w_{0,0}^1 * \max(0, w_{0,0}^0 * x_0 + w_{1,0}^0 * x_1 + b_{0,0}) + \\& w_{1,0}^1 * \max(0, w_{0,1}^0 * x_0 + w_{1,1}^0 * x_1 + b_{0,1}) + \\& w_{2,0}^1 * \max(0, w_{0,2}^0 * x_0 + w_{1,2}^0 * x_1 + b_{0,2}) + \\& b_{1,0}) + \\& w_{0,1}^2 * \max(0, w_{0,1}^1 * \max(0, w_{0,0}^0 * x_0 + w_{1,0}^0 * x_1 + b_{0,0}) + \\& w_{1,1}^1 * \max(0, w_{0,1}^0 * x_0 + w_{1,1}^0 * x_1 + b_{0,1}) + \\& w_{2,1}^1 * \max(0, w_{0,2}^0 * x_0 + w_{1,2}^0 * x_1 + b_{0,2}) + \\& b_{1,1}) + \\& w_{0,2}^2 * \max(0, w_{0,2}^1 * \max(0, w_{0,0}^0 * x_0 + w_{1,0}^0 * x_1 + b_{0,0}) + \\& w_{1,2}^1 * \max(0, w_{0,1}^0 * x_0 + w_{1,1}^0 * x_1 + b_{0,1}) + \\& w_{2,2}^1 * \max(0, w_{0,2}^0 * x_0 + w_{1,2}^0 * x_1 + b_{0,2}) + \\& b_{1,2}) + \\& b_2)\end{aligned}$$

Convolutional neural networks

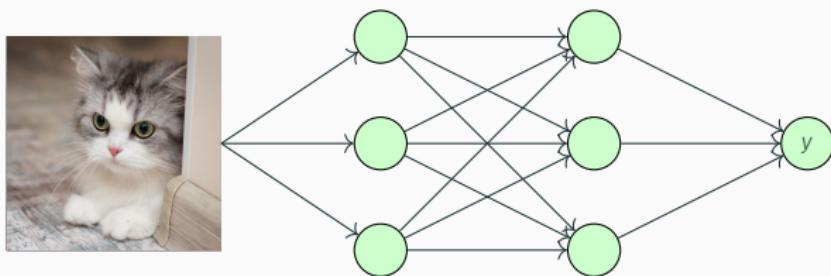
Convolutional neural networks



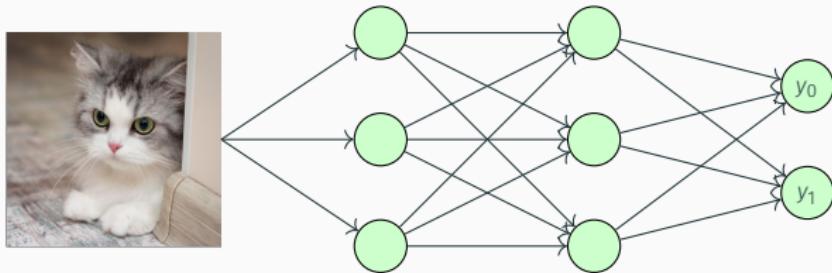
Convolutional neural networks: Images



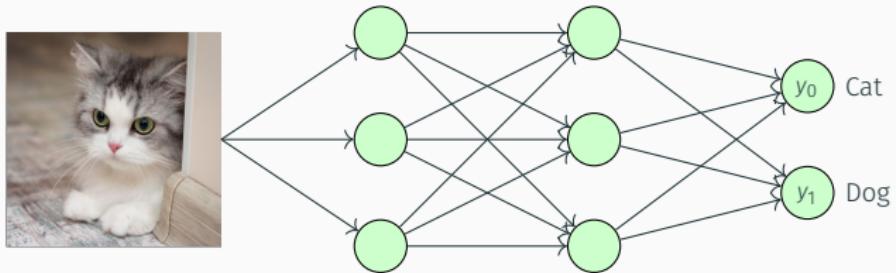
Convolutional neural networks: Images



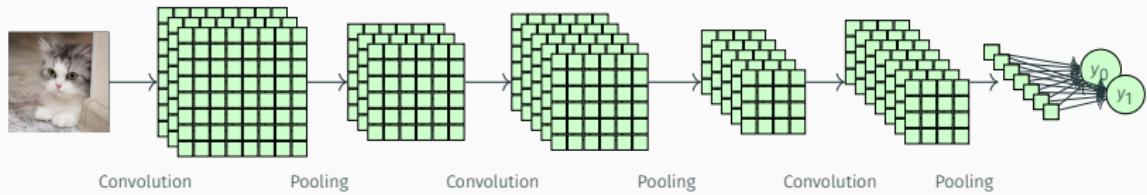
Convolutional neural networks: Classification



Convolutional neural networks: Classification

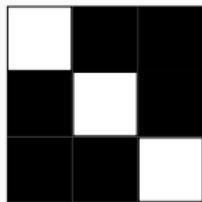
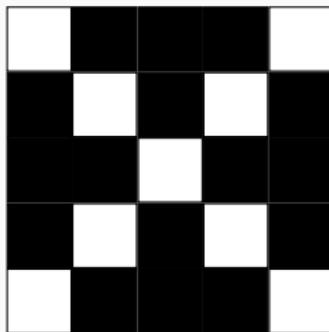


Convolutional neural networks: Architecture



Convolutional neural networks: Convolution

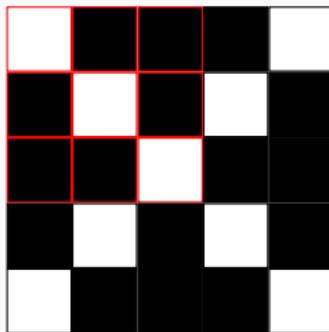
Image



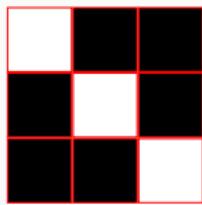
Pattern 1

Convolutional neural networks: Convolution

Image



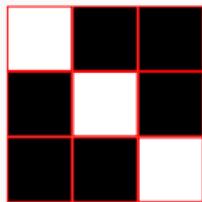
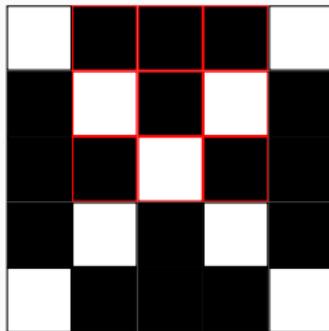
3



Pattern 1

Convolutional neural networks: Convolution

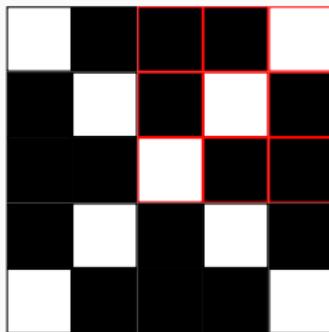
Image



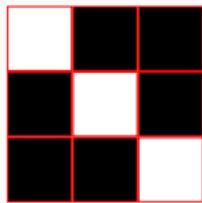
Pattern 1

Convolutional neural networks: Convolution

Image



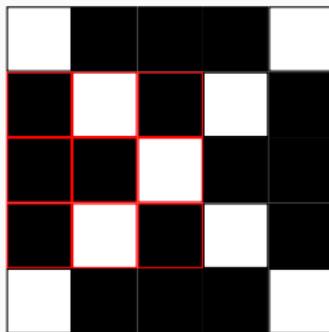
3	0	1
---	---	---



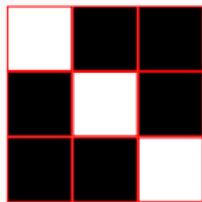
Pattern 1

Convolutional neural networks: Convolution

Image



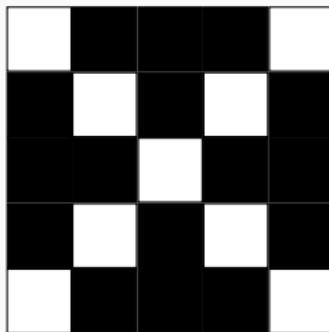
3	0	1
0		



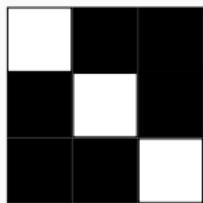
Pattern 1

Convolutional neural networks: Convolution

Image



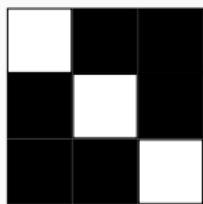
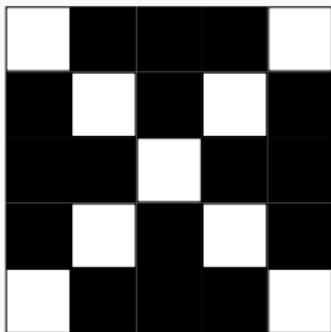
3	0	1
0	3	0
1	0	3



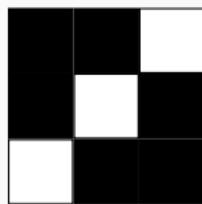
Pattern 1

Convolutional neural networks: Convolution

Image



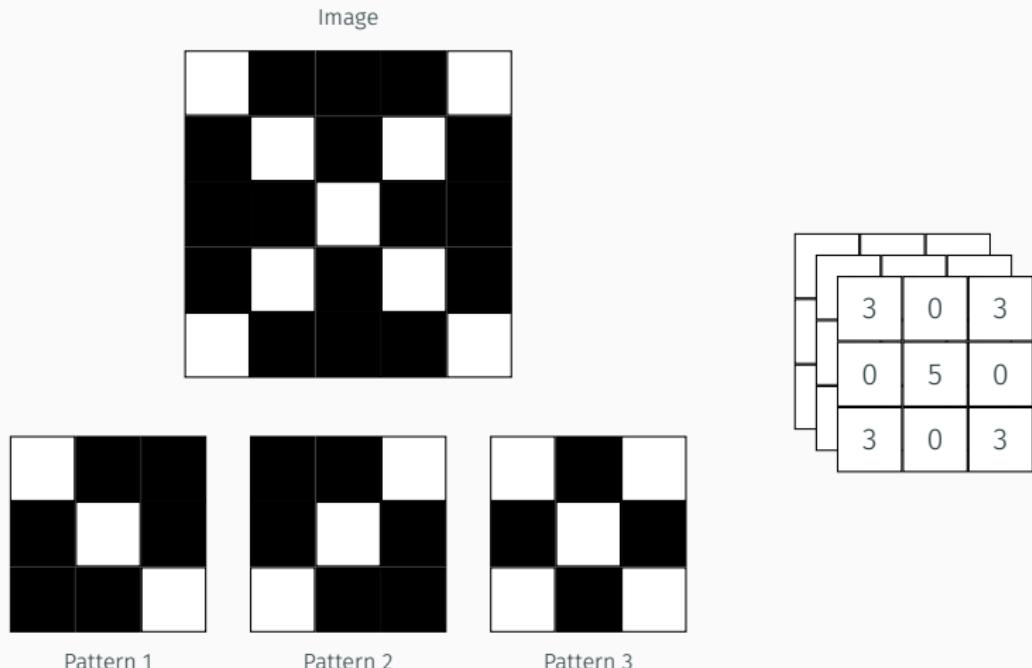
Pattern 1



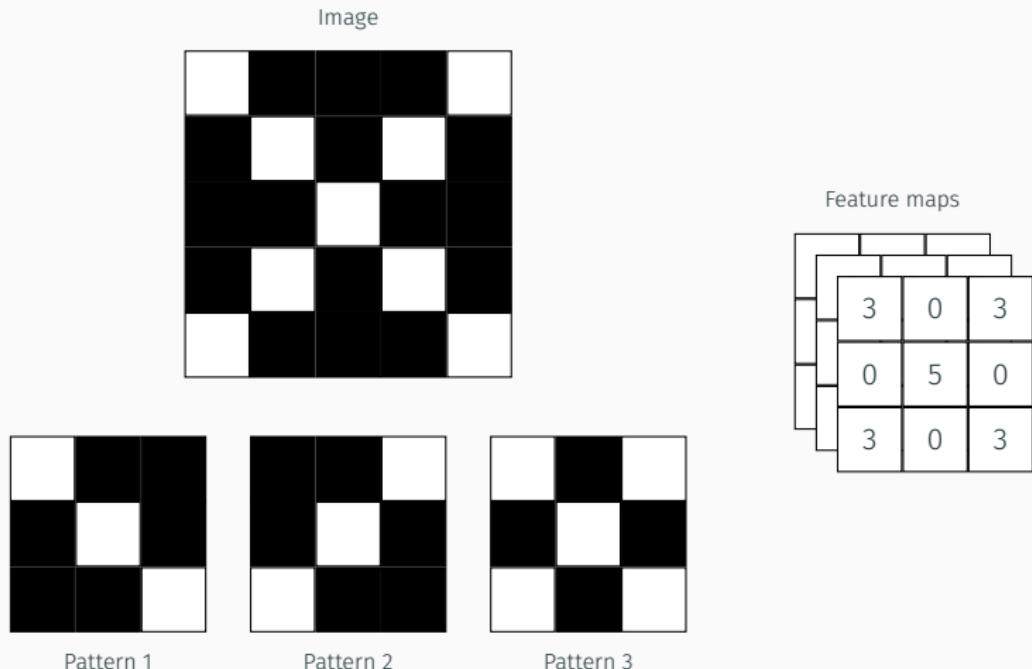
Pattern 2

1	0	3
0	3	0
3	0	1

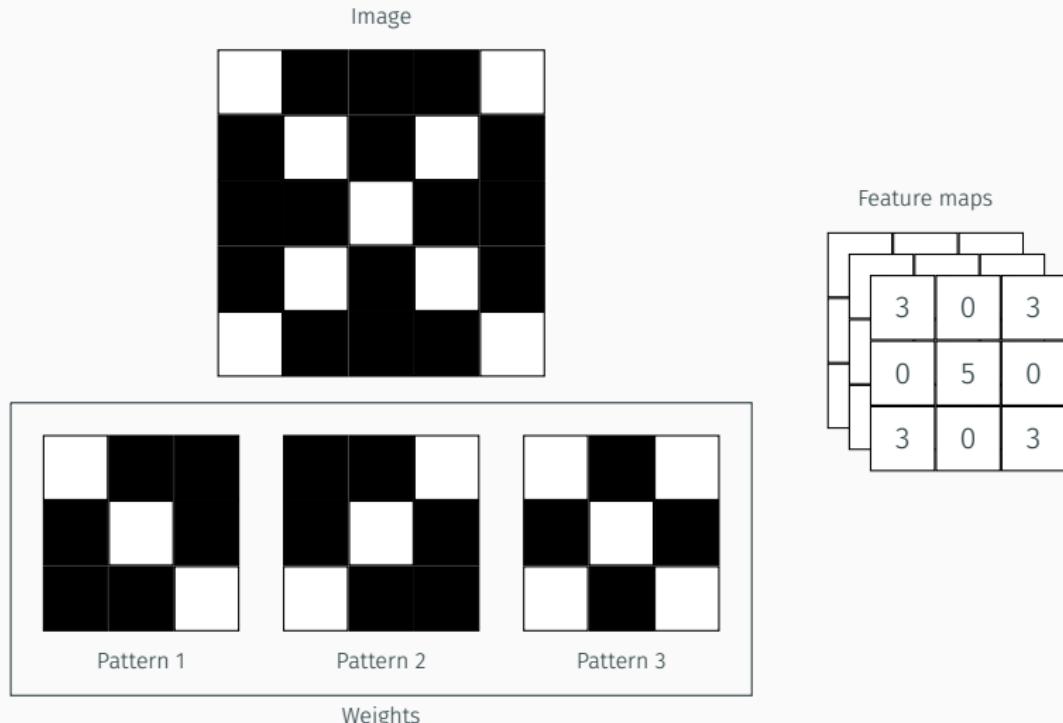
Convolutional neural networks: Convolution



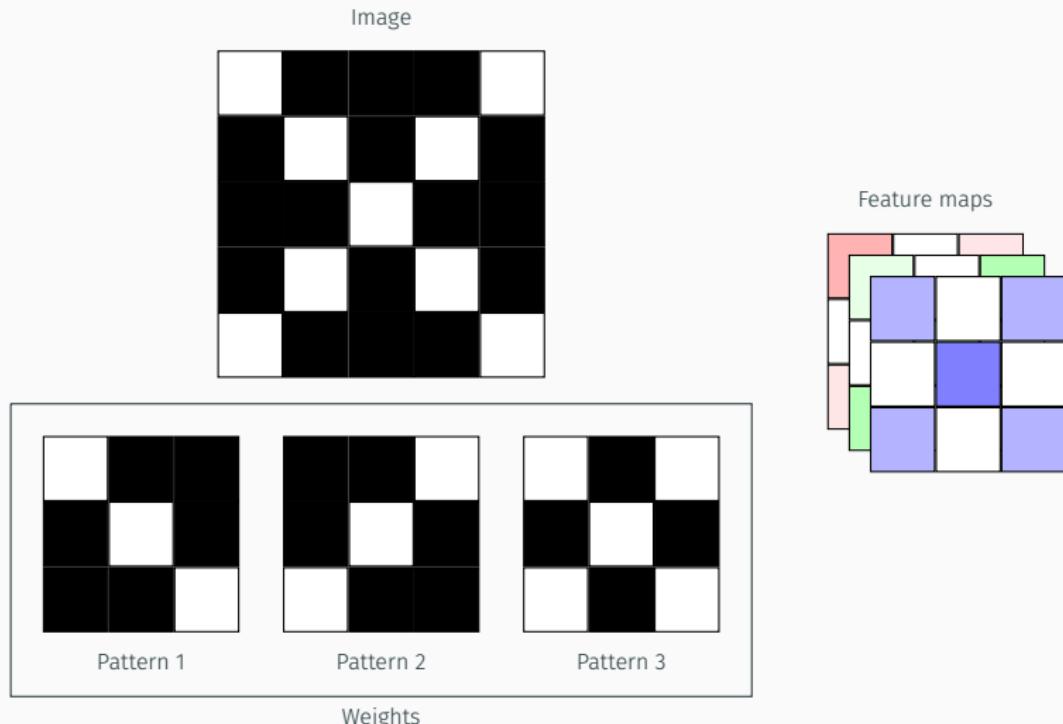
Convolutional neural networks: Convolution



Convolutional neural networks: Convolution



Convolutional neural networks: Convolution



Convolutional neural networks: Pooling

Feature map

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Convolutional neural networks: Pooling

Feature map

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15



Convolutional neural networks: Pooling

Feature map

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

5	7
---	---

Convolutional neural networks: Pooling

Feature map

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

5	7
13	

Convolutional neural networks: Pooling

Feature map

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

5	7
13	15

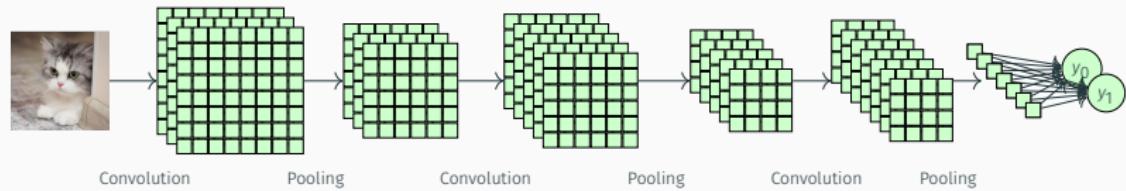
Convolutional neural networks: Pooling

Feature map

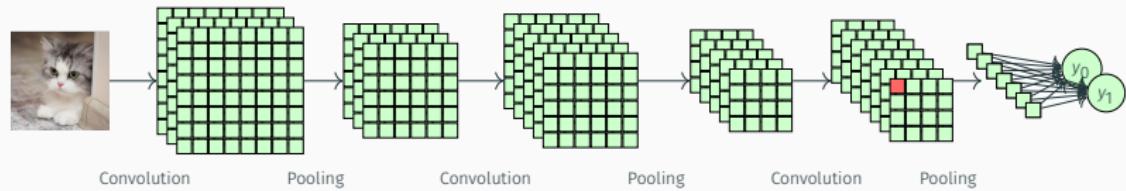
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

5	7
13	15

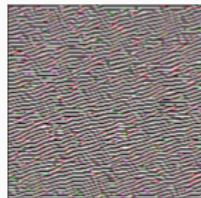
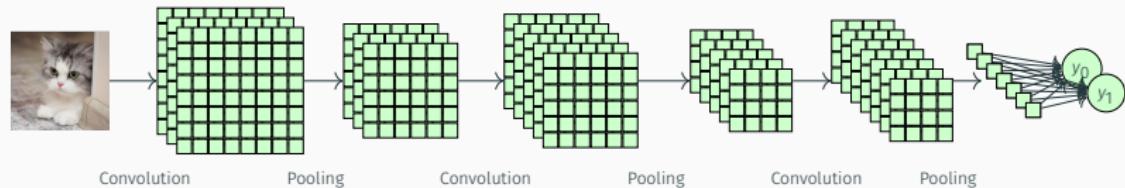
Convolutional neural networks: Architecture



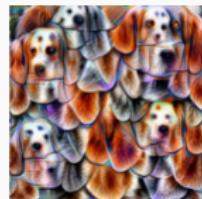
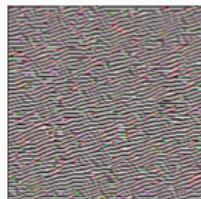
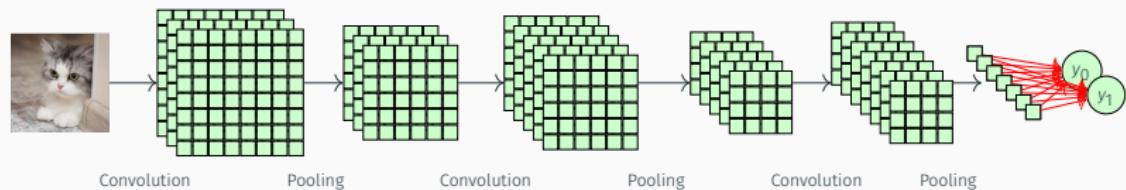
Convolutional neural networks: Architecture



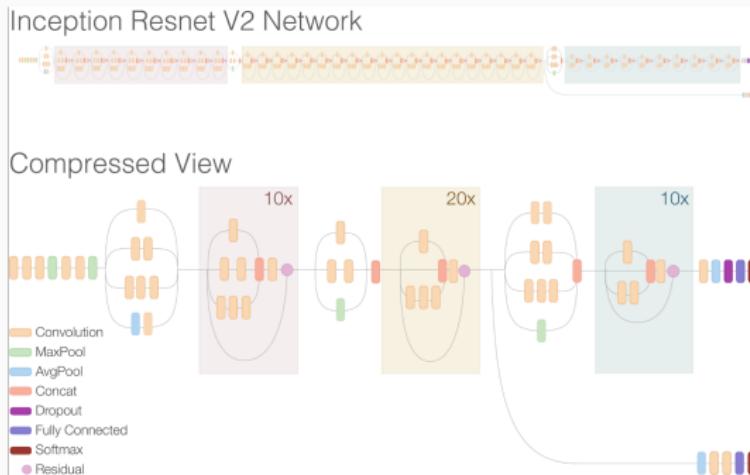
Convolutional neural networks: Architecture



Convolutional neural networks: Architecture

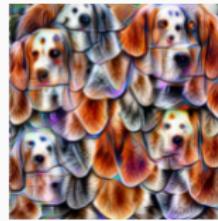
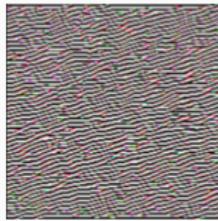
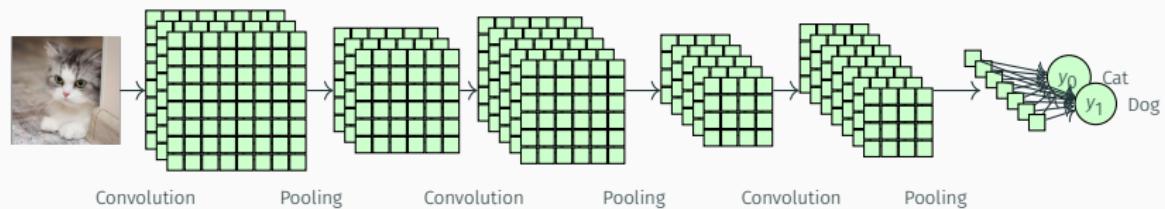


Convolutional neural networks: Architecture

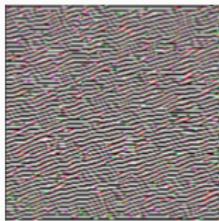
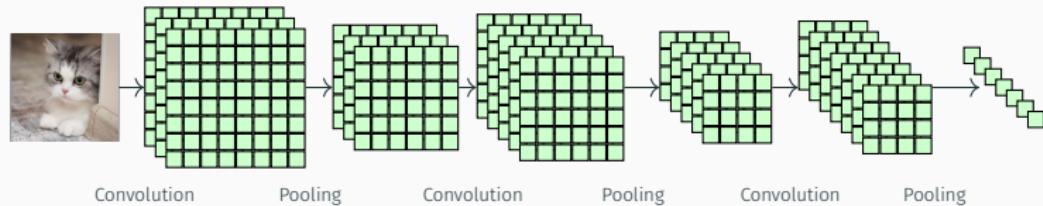


Practicalities

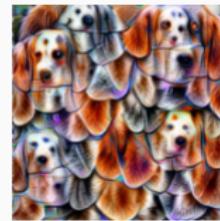
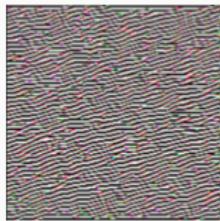
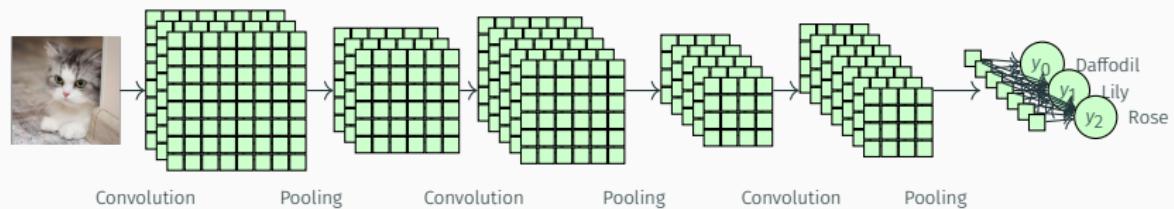
Practicalities: Transfer learning



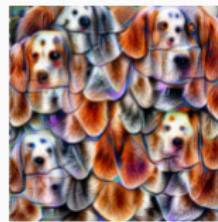
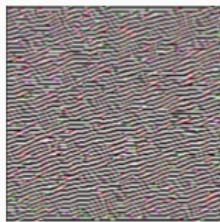
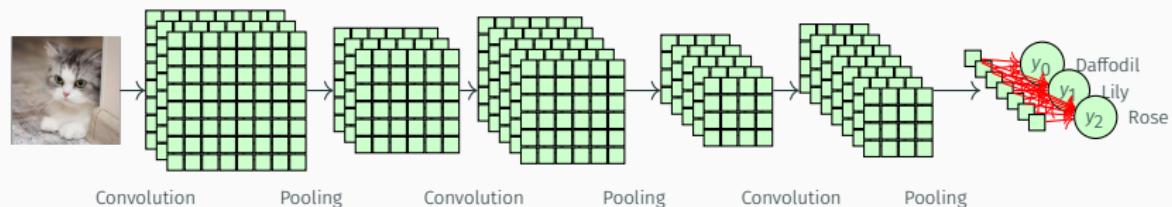
Practicalities: Transfer learning



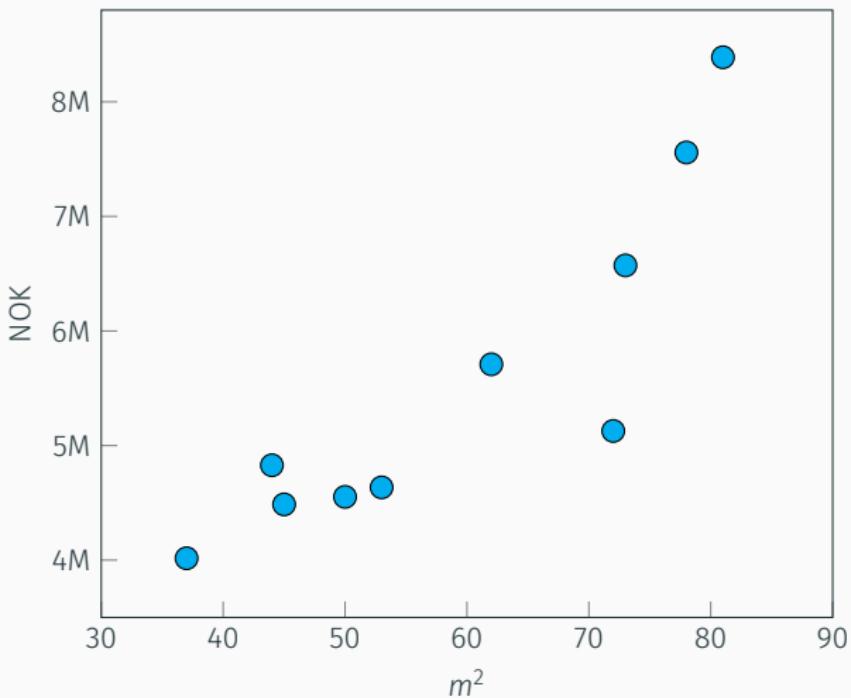
Practicalities: Transfer learning



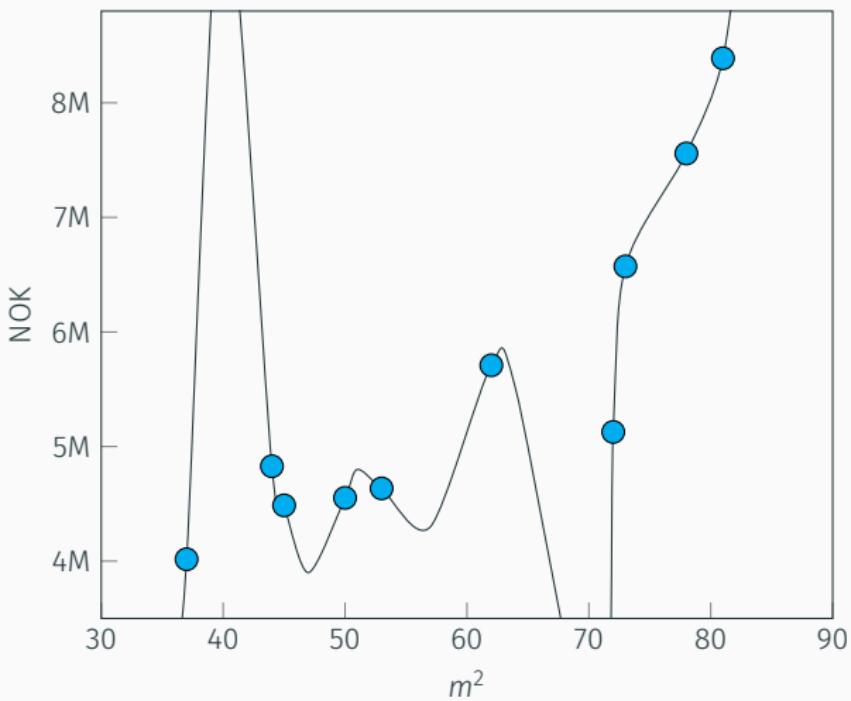
Practicalities: Transfer learning



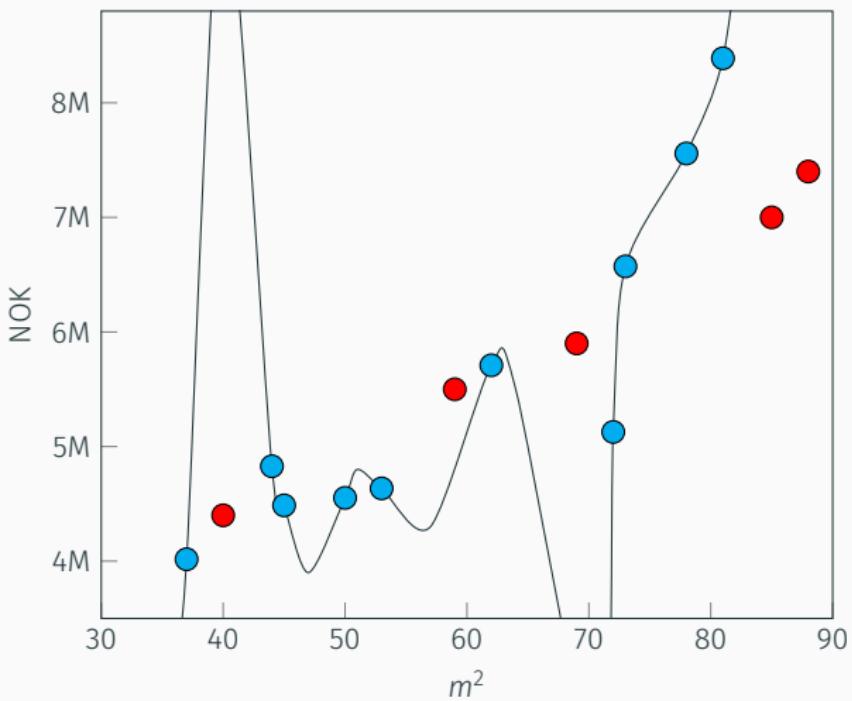
Practicalities: Overfitting



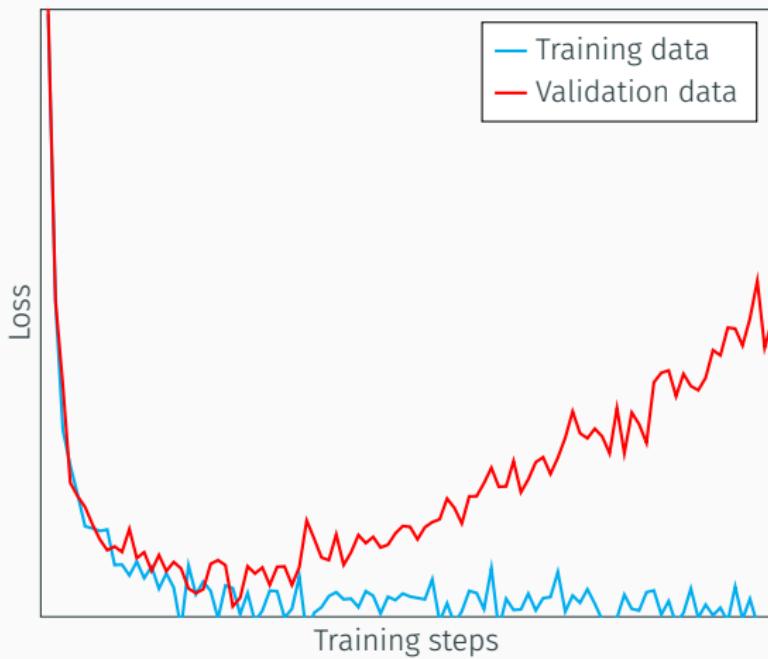
Practicalities: Overfitting



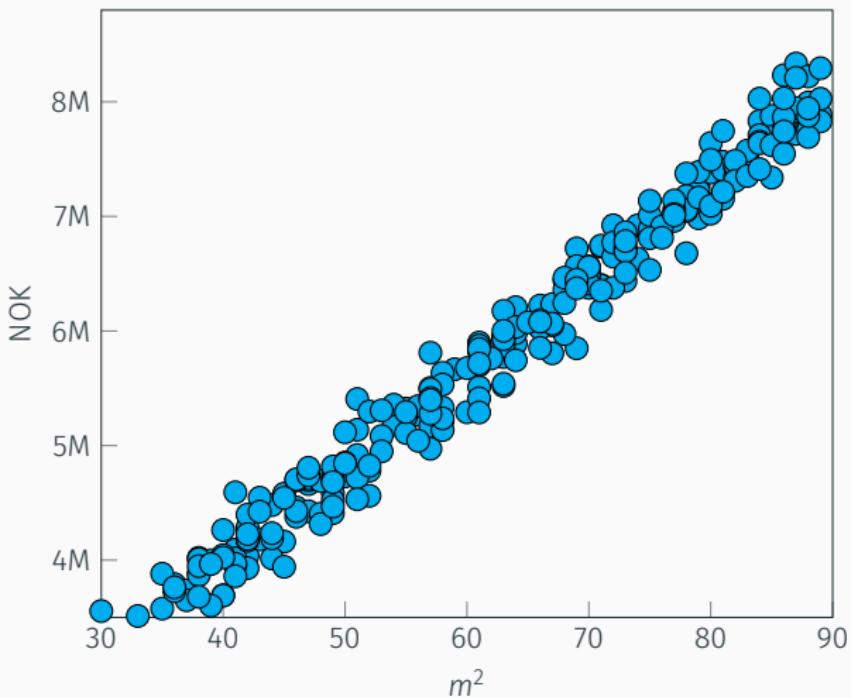
Practicalities: Overfitting



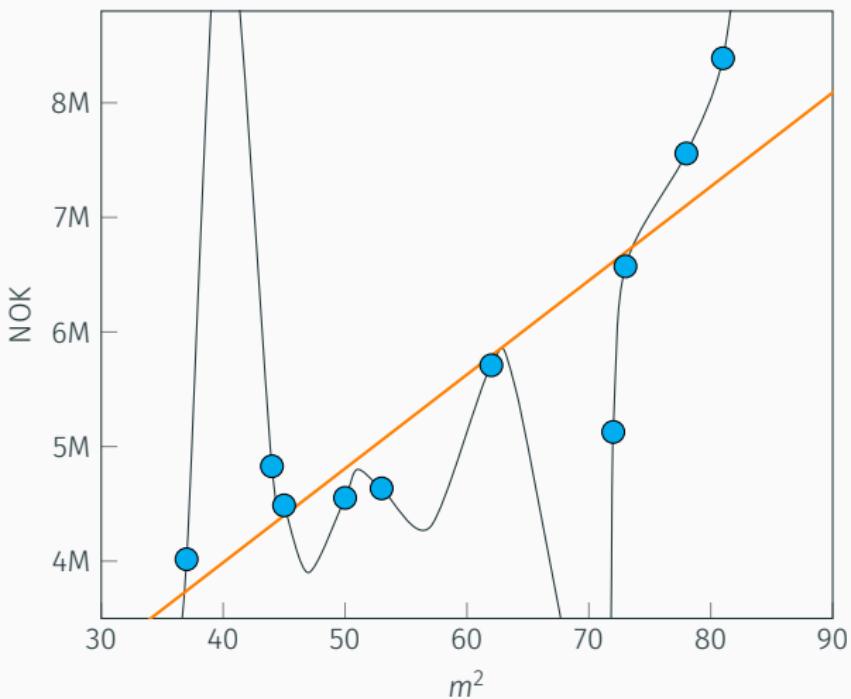
Practicalities: Overfitting



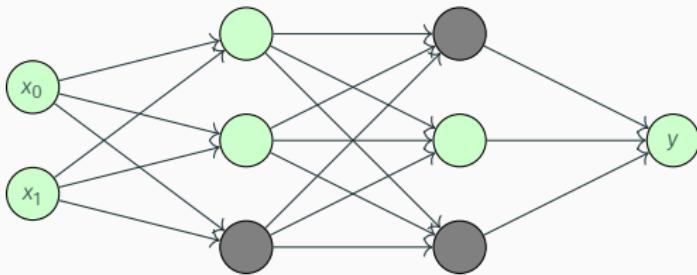
Practicalities: Overfitting



Practicalities: Regularization



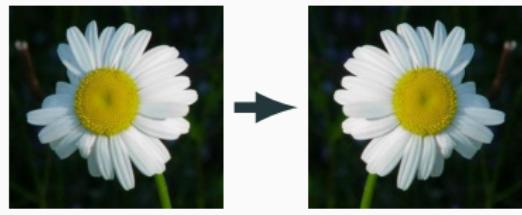
Practicalities: Regularization



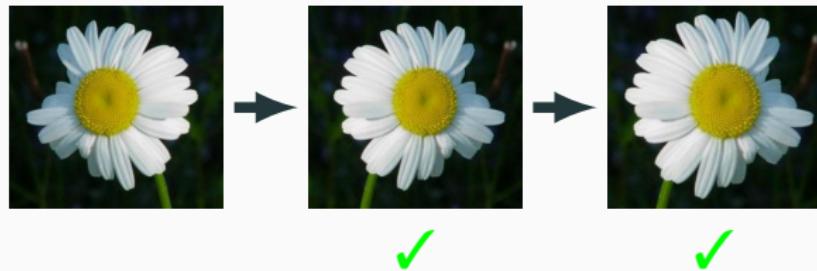
Practicalities: Augmentation



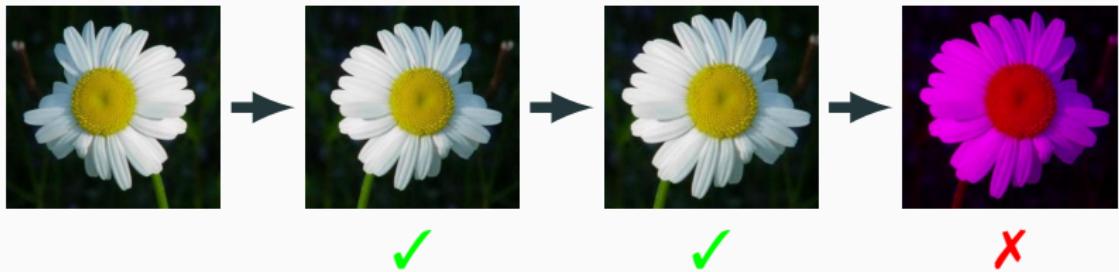
Practicalities: Augmentation



Practicalities: Augmentation



Practicalities: Augmentation



Summary

- What is a statistical learning model?

A formula expressing a relationship between inputs and outputs

- What is a loss function?

A function quantifying how good a set of predictions are

- How do we train a model?

By applying gradual updates of parameters using gradient descent

- How does a deep neural network work?

Sequentially applying (non-linear) artificial neurons to transform inputs

- How does a convolutional neural network process images?

Alternating convolutions and pooling, to match patterns in the input

- What is transfer learning?

Retraining (parts of) an already trained model for a new problem

- What is overfitting?

When a model learns patterns in the training data that does not hold generally

- How do we avoid it?

Rigorous testing, regularization and data augmentation