UiO : Department of Informatics
University of Oslo

INF 5860 Machine learning for image classification

Lecture 8: Generalization

Tollef Jahren

March 7 , 2018

Outline

- Is learning feasible?
- Model complexity
- Overfitting
- Evaluating performance
- Learning from small datasets
- Rethinking generalization
- Capacity of dense neural networks

About today

- Part1: Learning theory
- Part2: Practical aspects of learning

Readings

- Learning theory (caltech course):
  - https://work.caltech.edu/lectures.html
  - Lecture (Videos): 2,5,6,7,8,11

- Read: CS231n: section “Dropouts”
  - http://cs231n.github.io/neural-networks-2/

- **Optional**:
  - Read: The Curse of Dimensionality in classification
    - http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

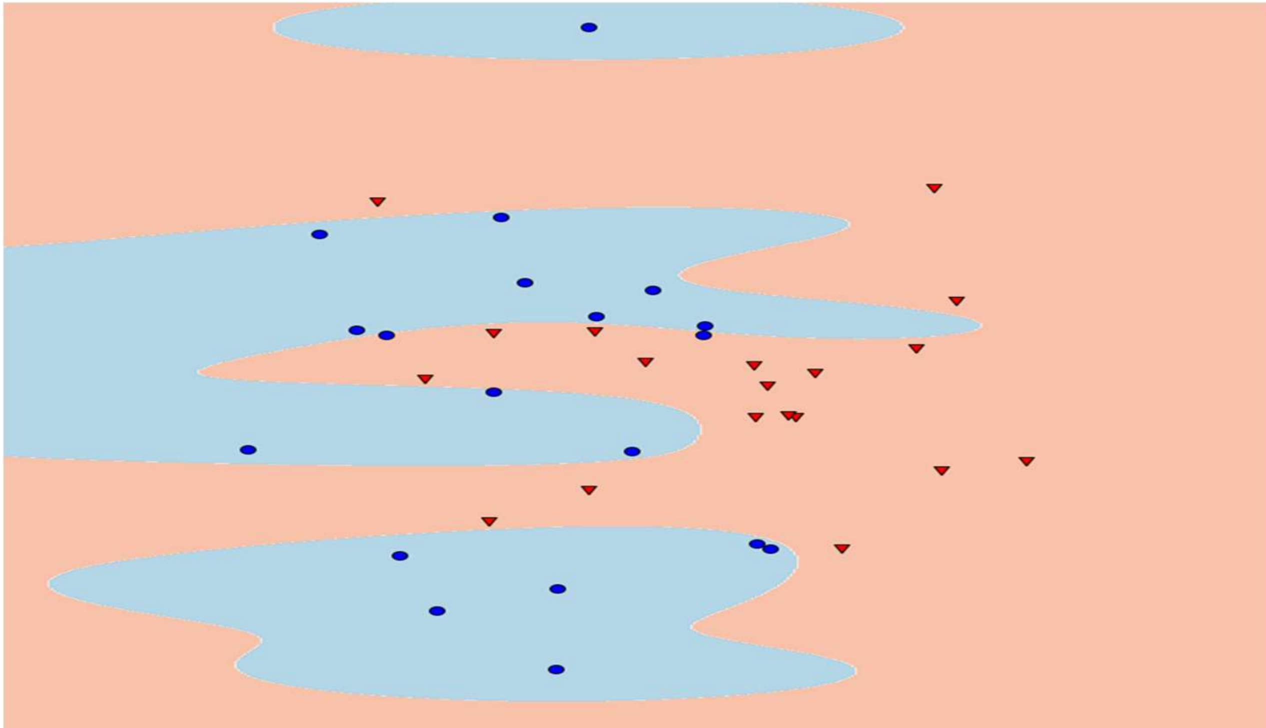
  - Read: Rethinking generalization
    - https://arxiv.org/pdf/1611.03530.pdf

Progress

- **Is learning feasible?**
- Model complexity
- Overfitting
- Evaluating performance
- Learning from small datasets
- Rethinking generalization
- Capacity of dense neural networks

Is learning feasible?

- Classification is to find the decision boundary
- But is it learning?



Notation

- **Formalization supervised learning:**
  - Input: $\boldsymbol{x}$
  - Output: $\boldsymbol{y}$
  - Target function: $f : \mathcal{X} \rightarrow \mathcal{Y}$
  - Data: $(\boldsymbol{x_1}, \boldsymbol{y_1}), (\boldsymbol{x_2}, \boldsymbol{y_2}) \cdots , (\boldsymbol{x_N}, \boldsymbol{y_N})$
  - Hypothesis: $\mathrm{h} : \mathcal{X} \rightarrow \mathcal{Y}$

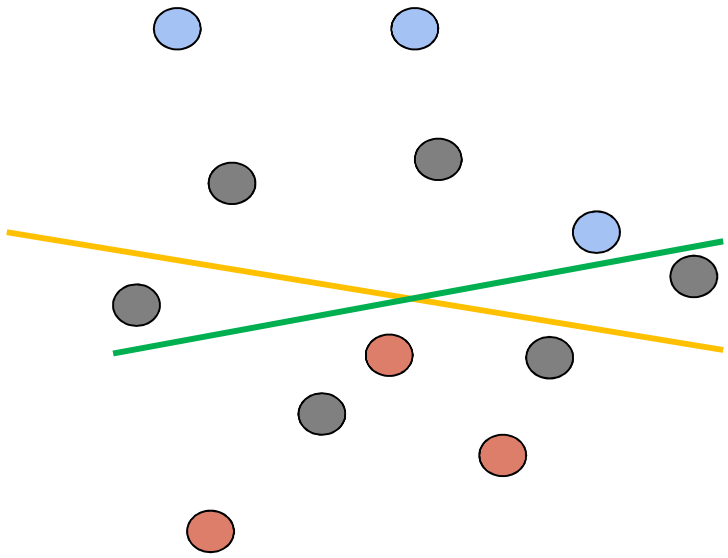
Example:

Hypothesis set: $y = w_1 x + w_0$

A hypothesis: $y = 2x + 1$

More notation

- **In-sample** (colored): Training data available to find your solution.
- **Out-of-sample** (gray): Data from the real world, the hypothesis will be used for.
- **Final hypothesis:** (yellow line)
- **Target hypothesis:** (green line)
- **Generalization:** Difference between the in-sample error and the out-of-sample error

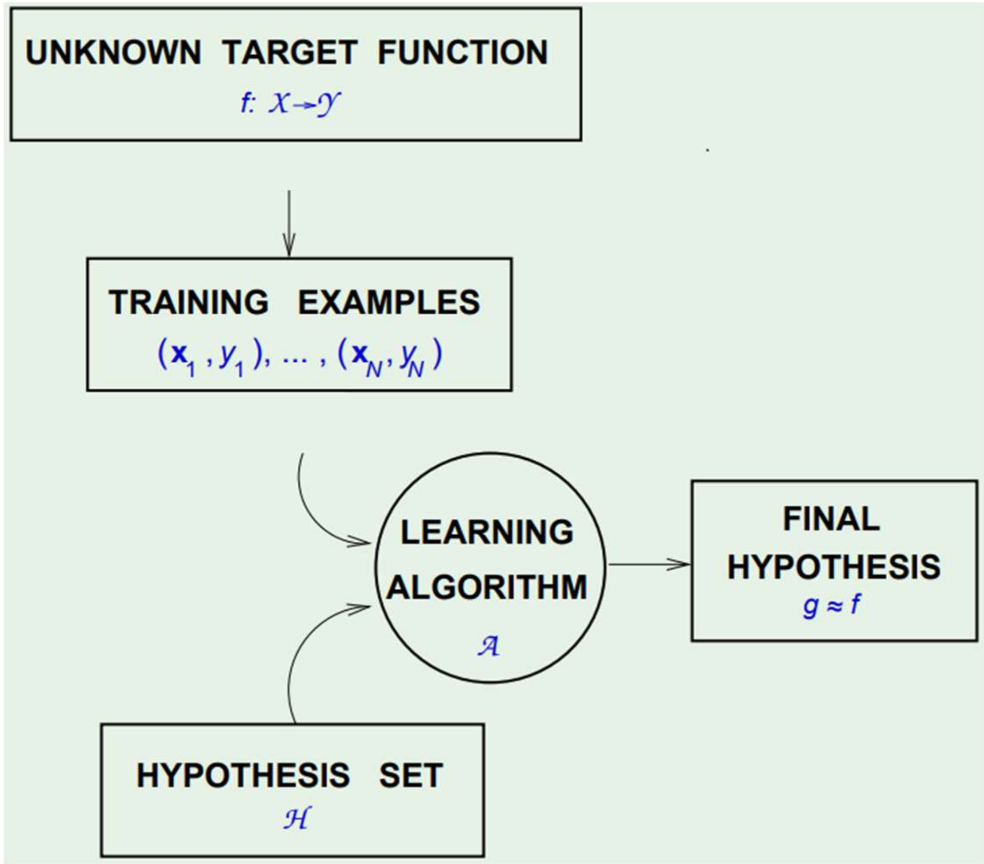


Learning diagram

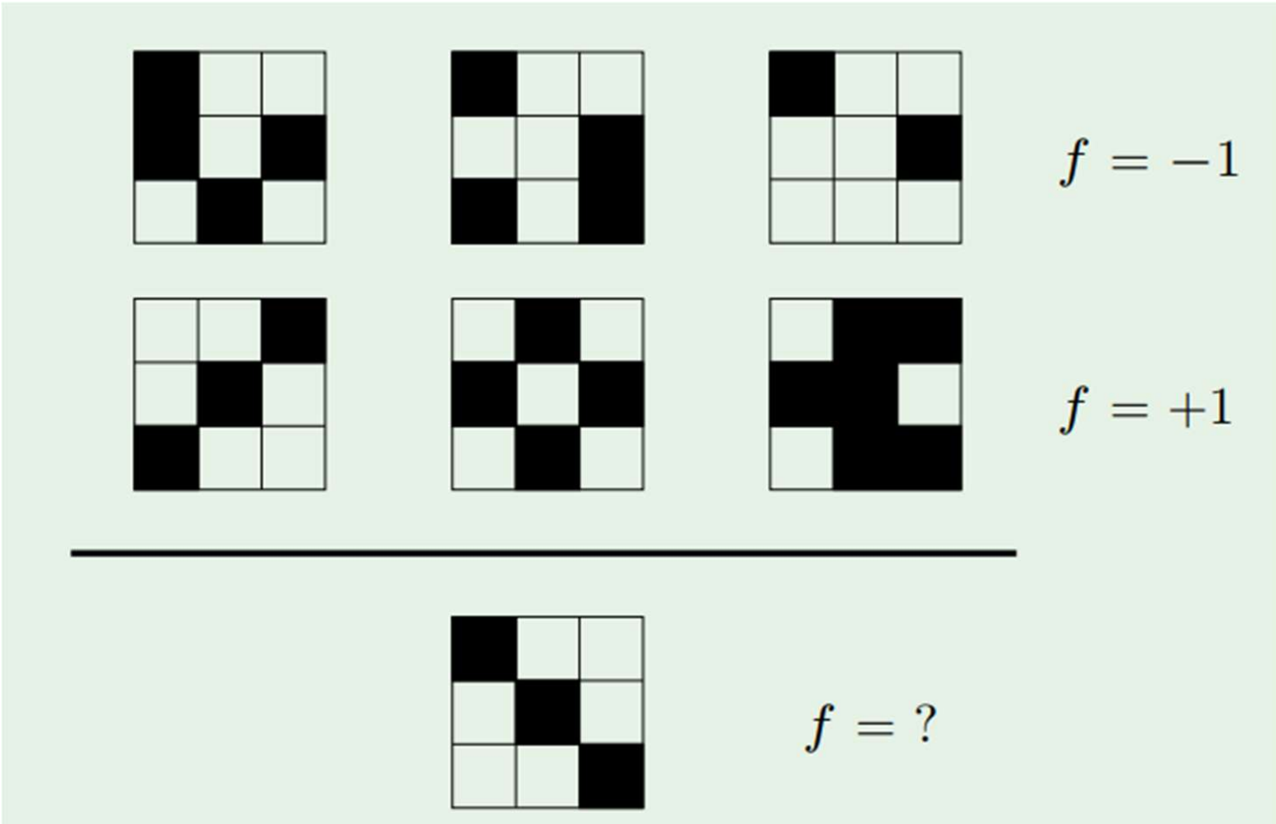
- **The Hypothesis Set**
  $\mathcal{H} = \{h\}$ $\quad g \in \mathcal{H}$

- **The Learning Algorithm**
  - e.g. Gradient descent

The hypothesis set and the learning algorithm are referred to as the Learning model



Learning puzzle

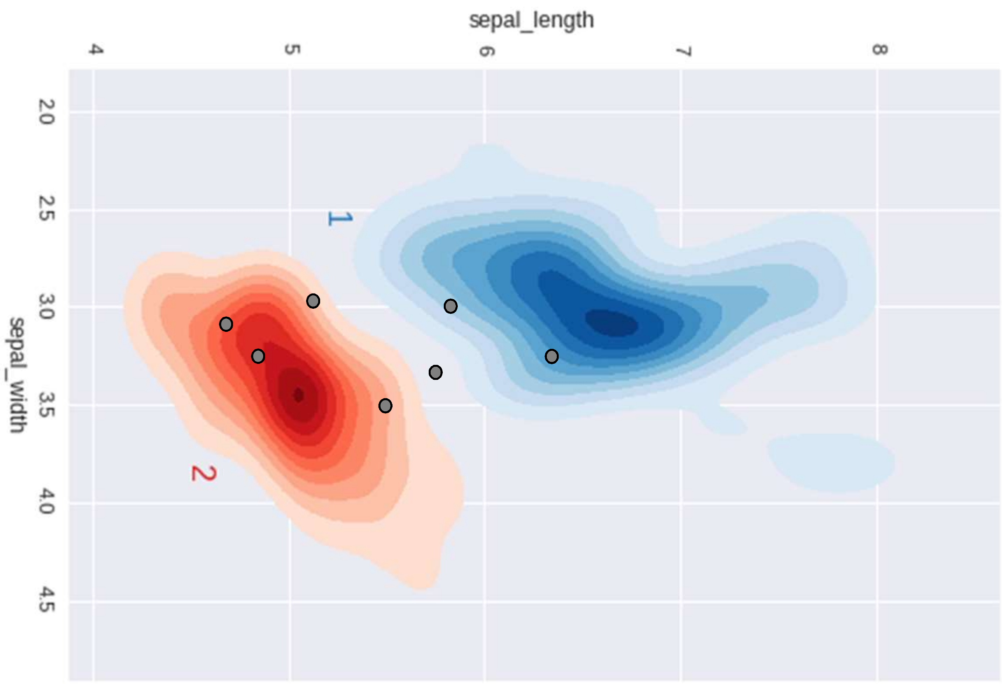


The target function is UNKNOWN

- We cannot know what we have not seen!
- What can save us?
  - Answer: **Probability**

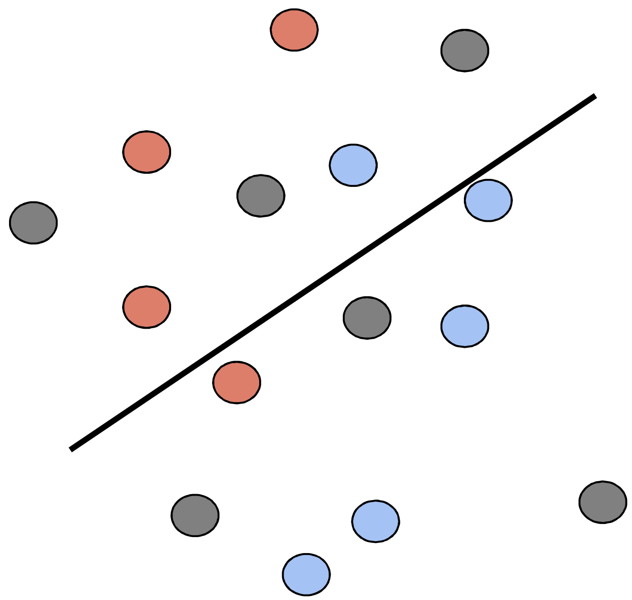
Drawing from the same distribution

- Requirement:
  - The **in-sample** and **out-of-sample** data must be drawn from the same distribution (process)



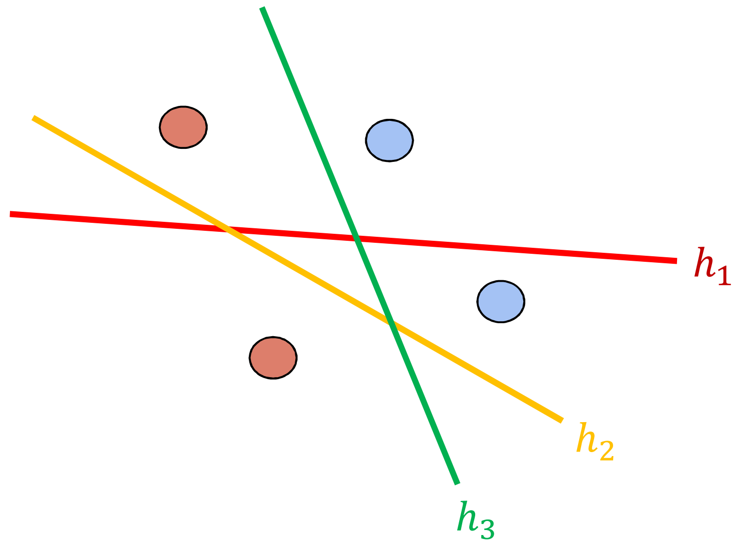
What is the expected out-of-sample error?

- For a randomly selected hypothesis
- The closest error approximation is the **in-sample** error



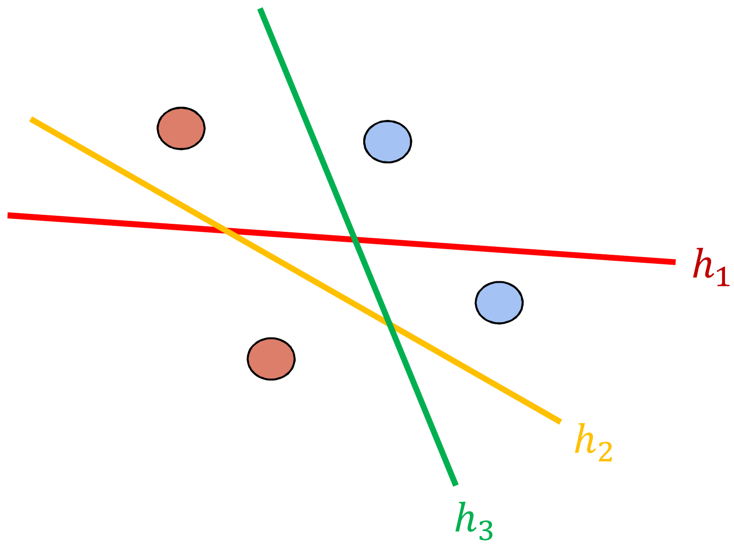
What is training?

- A general view of training:
  - Training is a search through possible hypothesis
  - Use in-sample data to find the best hypothesis



What is the effect of choosing the best hypothesis?

- Smaller **in-sample** error
- Increasing the probability that the result is a coincidence
- The expected **out-of-sample** error is greater or equal to the **in-sample** error



Searching through all possibilities

- The extreme case search through all possibilities
- Then you are guaranteed 0% **in-sample** error rate
- No information about the out-of-sample error

Progress

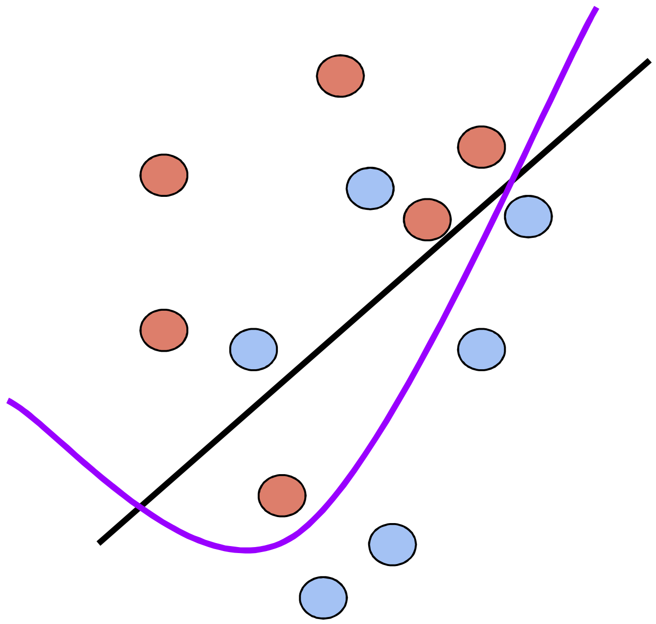
- Is learning feasible?
- **Model complexity**
- Overfitting
- Evaluating performance
- Learning from small datasets
- Rethinking generalization
- Capacity of dense neural networks

Capacity of the model (hypothesis set)

- The model restrict the number of hypothesis you can find
- Model capacity is a reference to how many possible hypothesis you have
- A linear model has a set of all linear functions as its hypothesis

$$\widehat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$$

$$\widehat{y} = \mathbf{x}^T W\mathbf{x} + \mathbf{w}^T x + b$$

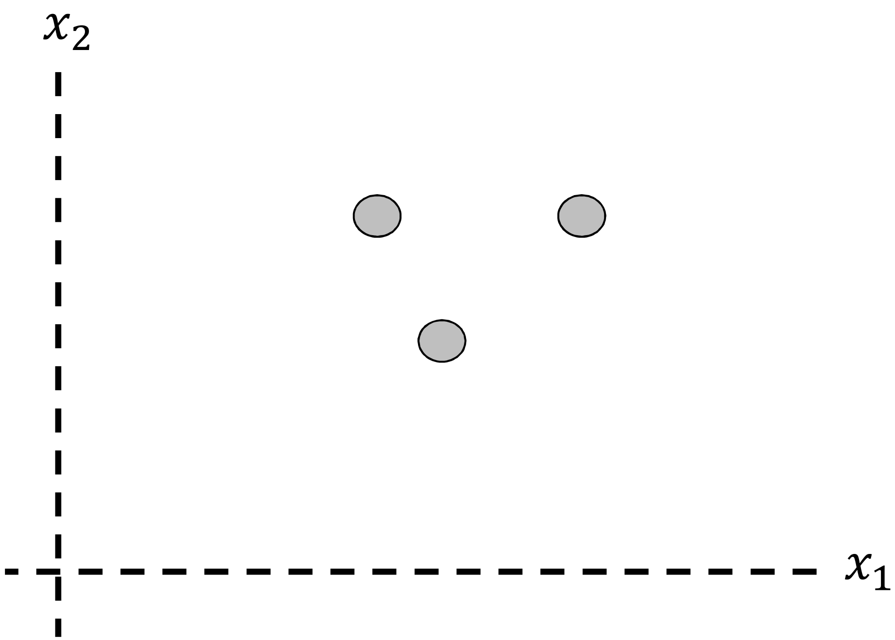


Measuring capacity

- **Vapnik-Chervonenkis (VC) dimension**
  - Denoted: $d_{VC}(\mathcal{H})$
  - Definition:
    - The maximum number of points that can be arrange such that $\mathcal{H}$ can shatter them.

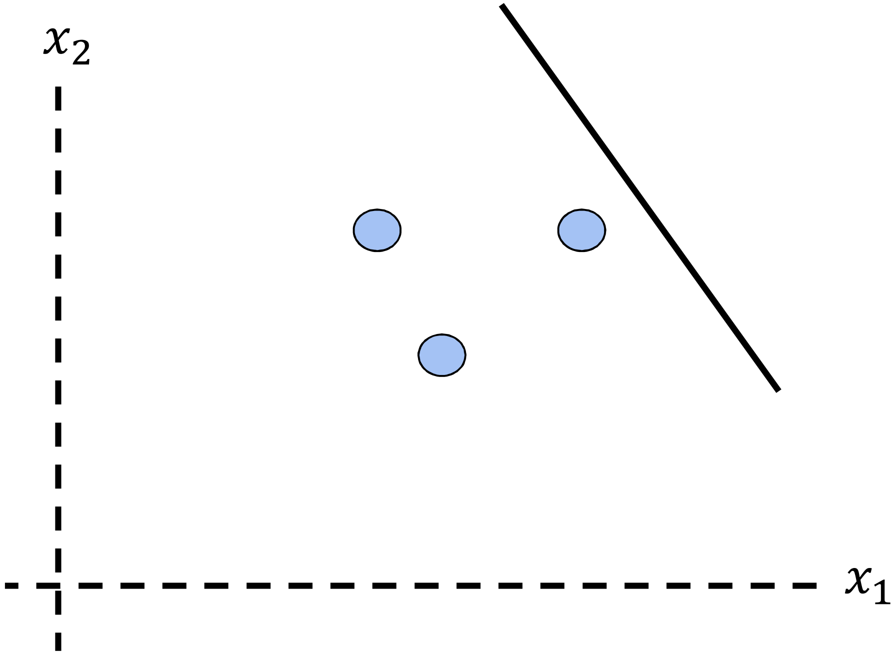
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration (N =3)



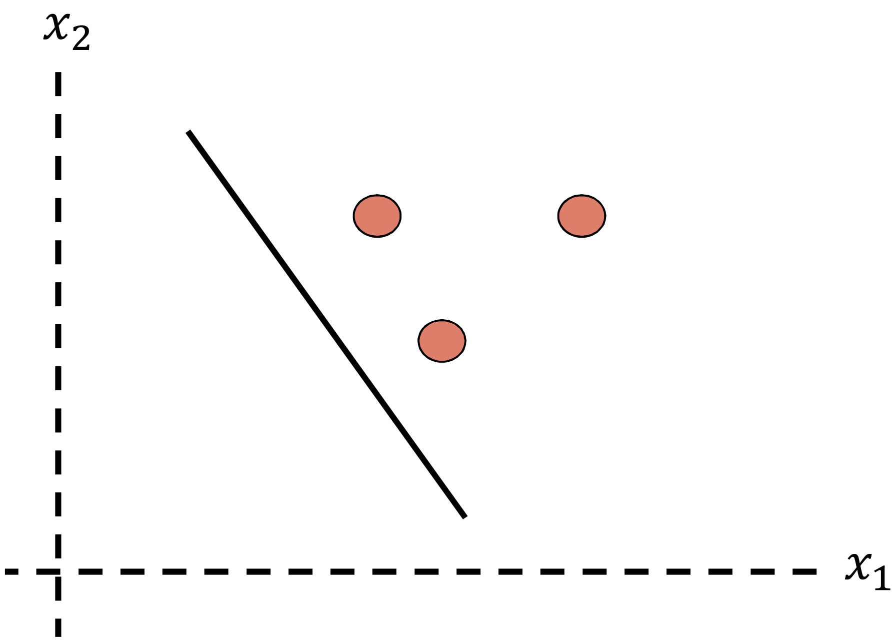
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration (N =3)



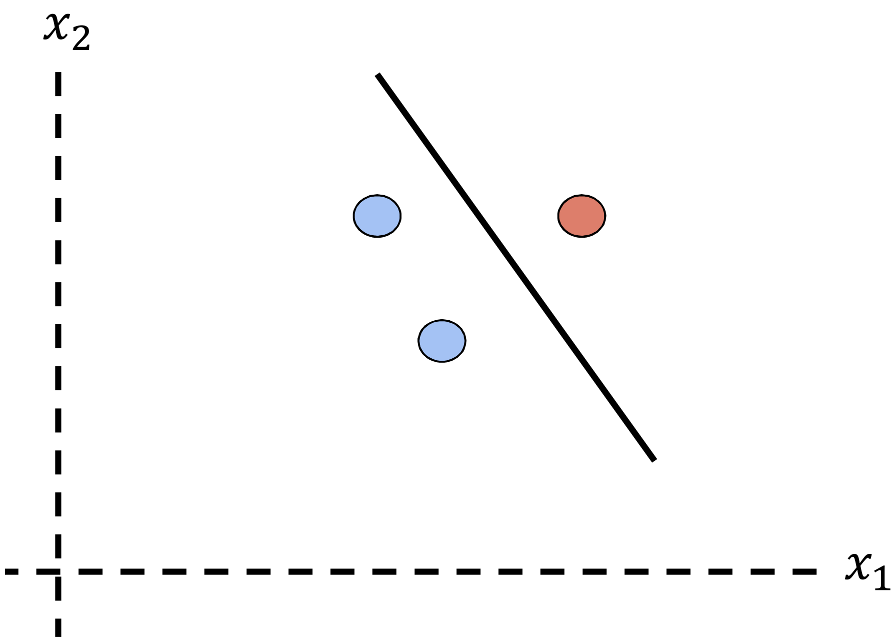
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration (N =3)



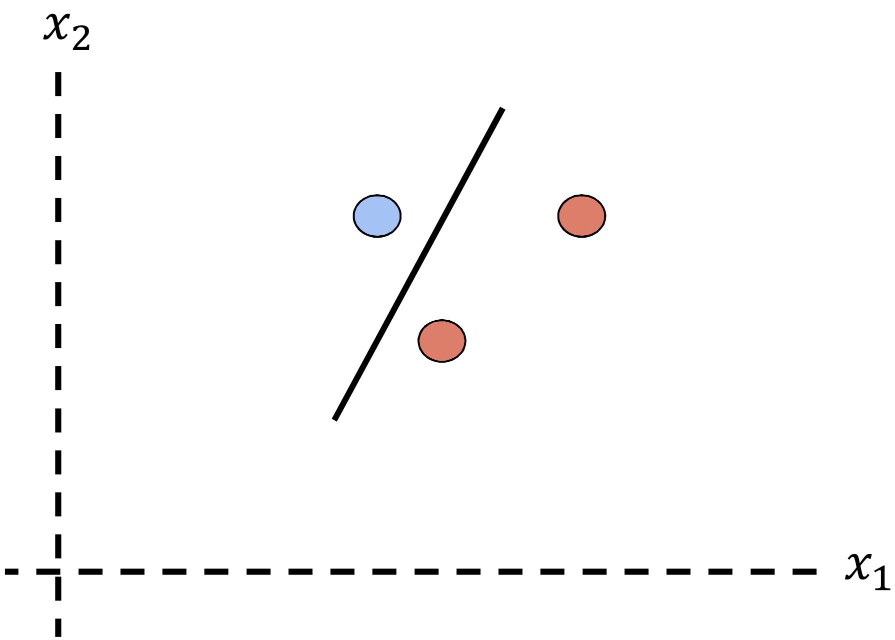
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration (N =3)



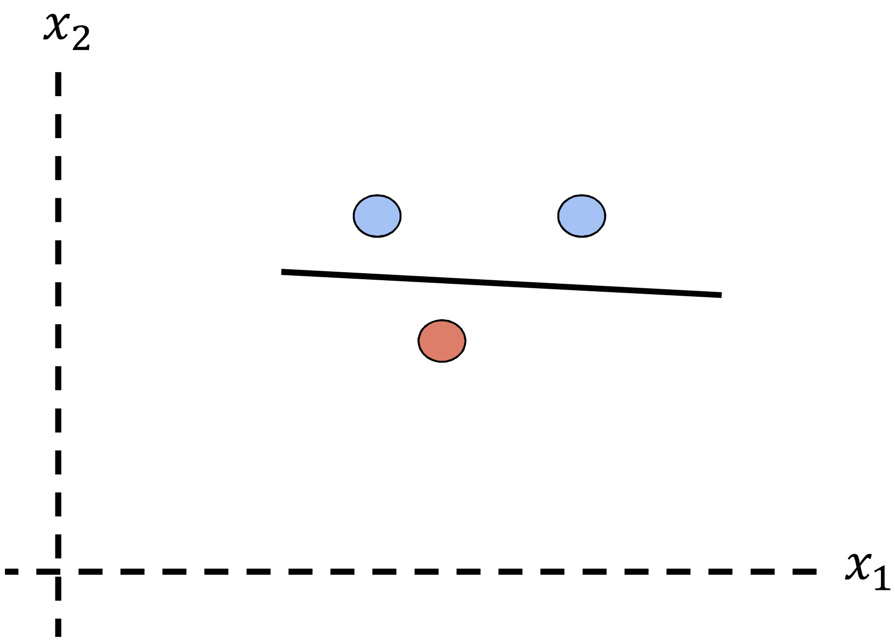
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration (N =3)



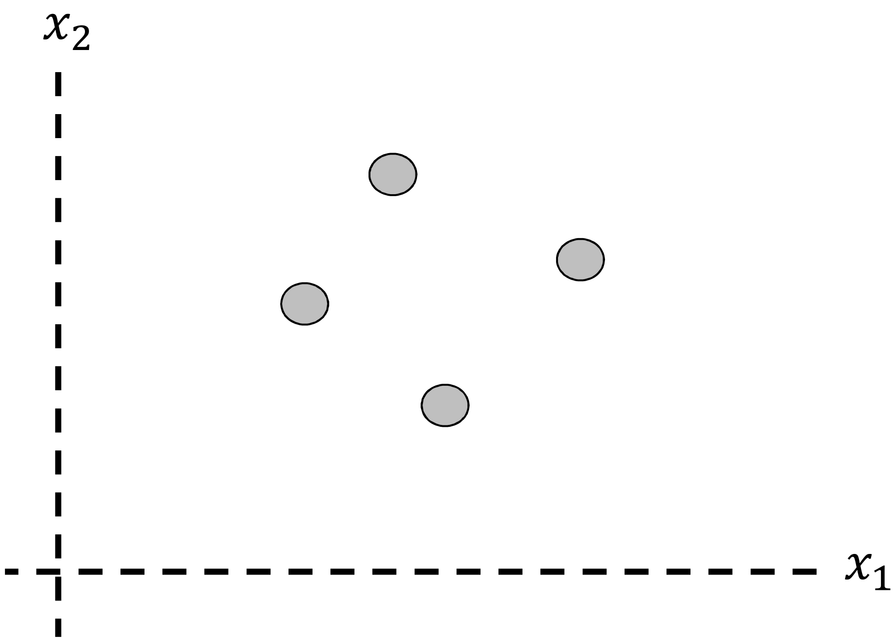
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration (N =3)



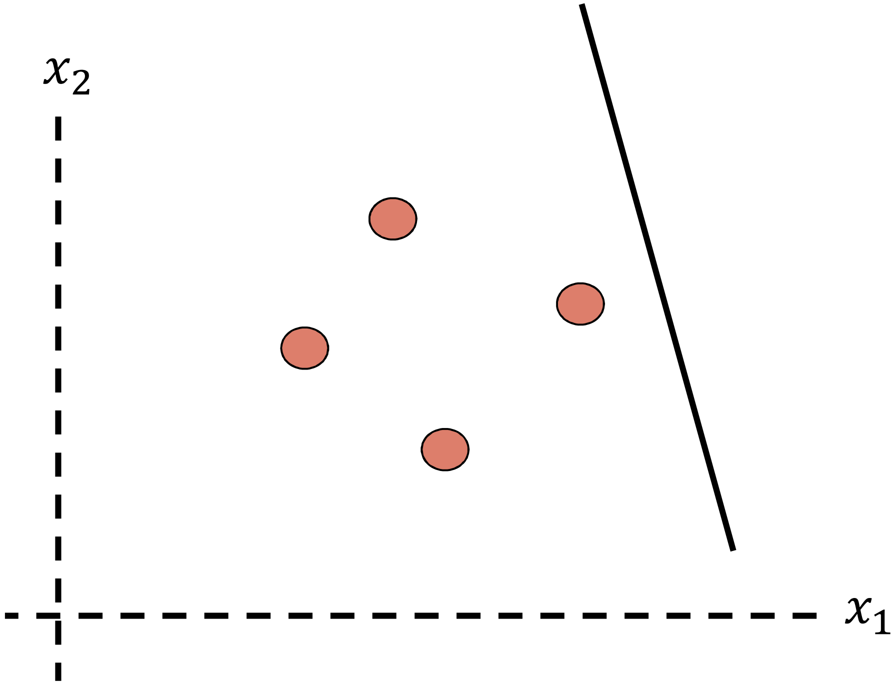
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration ($N = 4$)



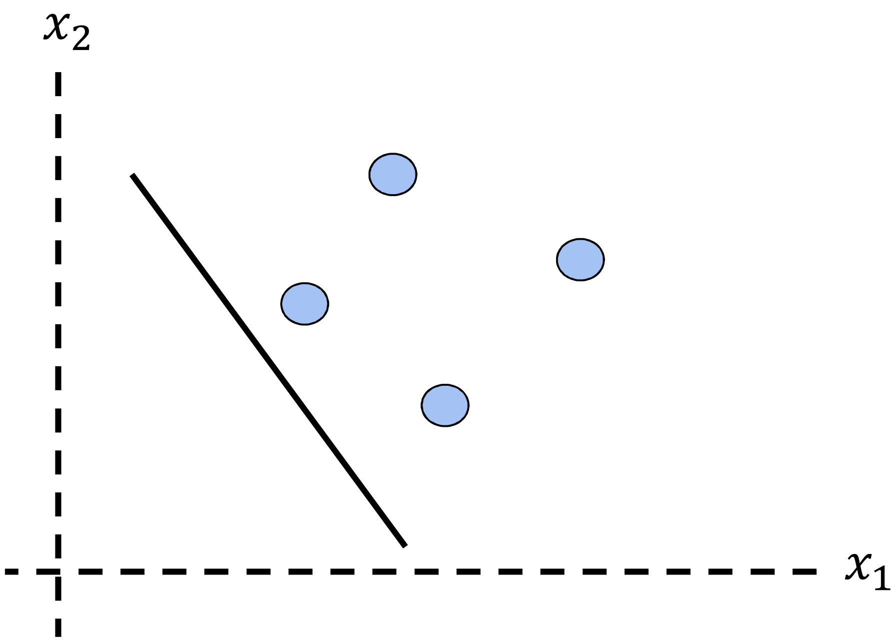
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration ($N = 4$)



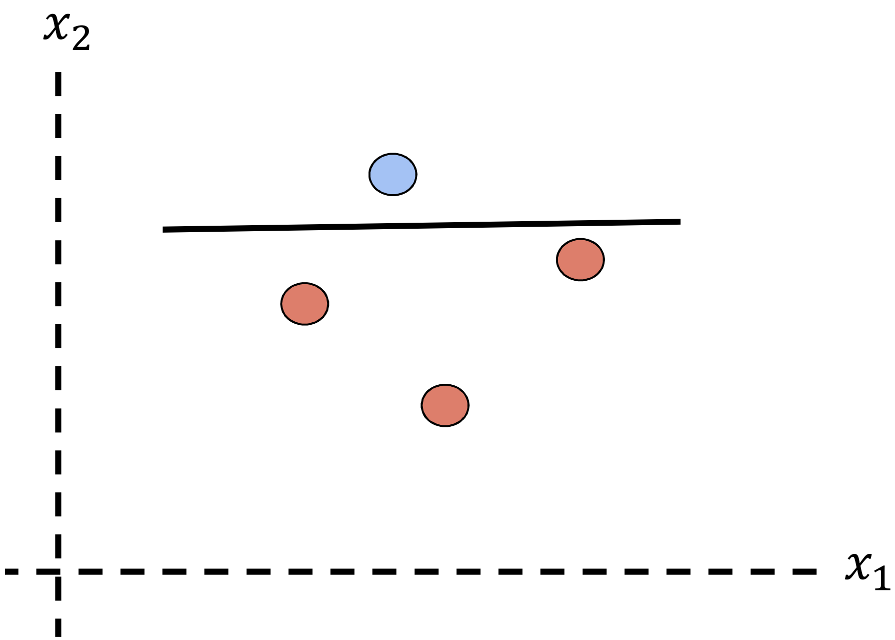
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration ($N = 4$)



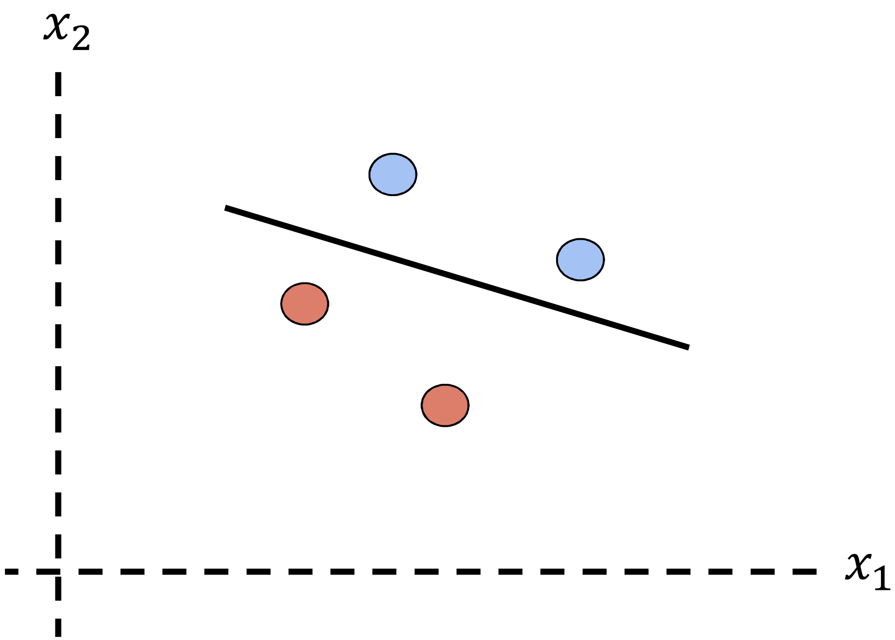
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration ($N = 4$)



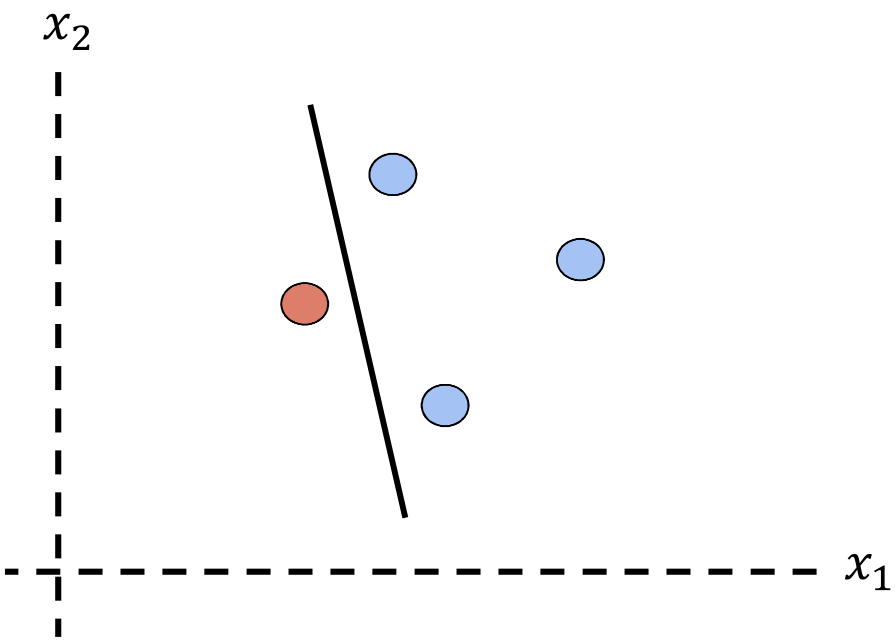
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T \mathbf{x} + b)$
- Configuration ($N = 4$)



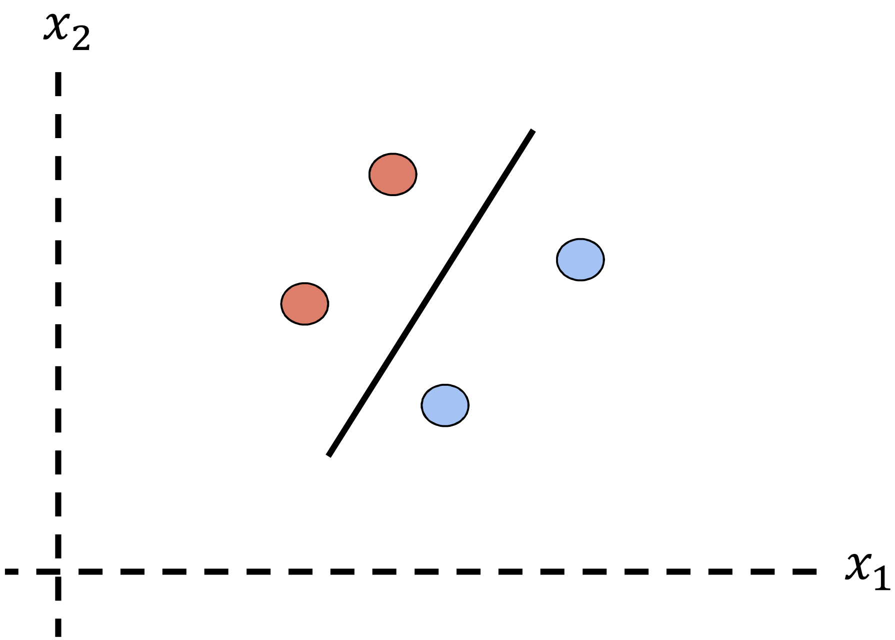
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration ($N = 4$)



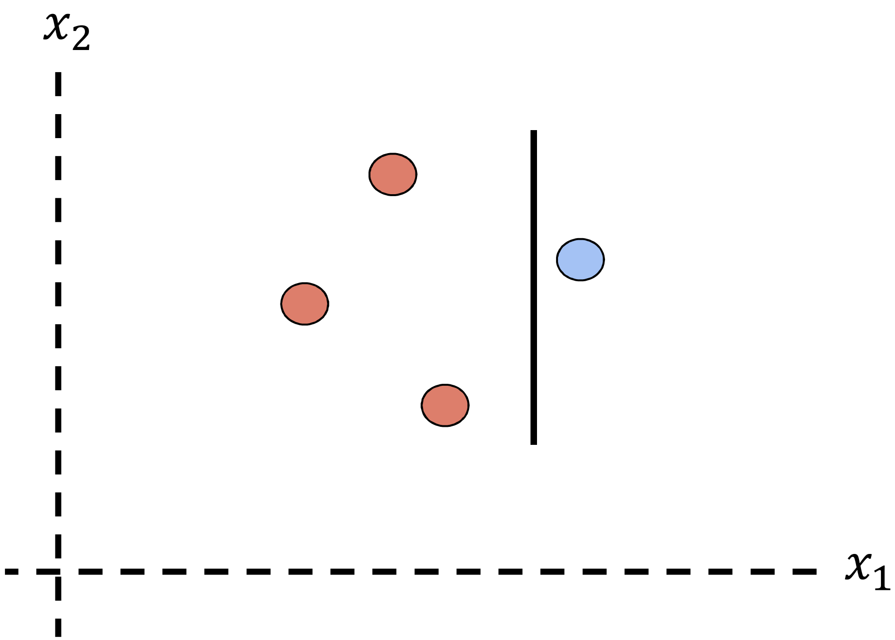
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T \mathbf{x} + b)$
- Configuration ($N = 4$)



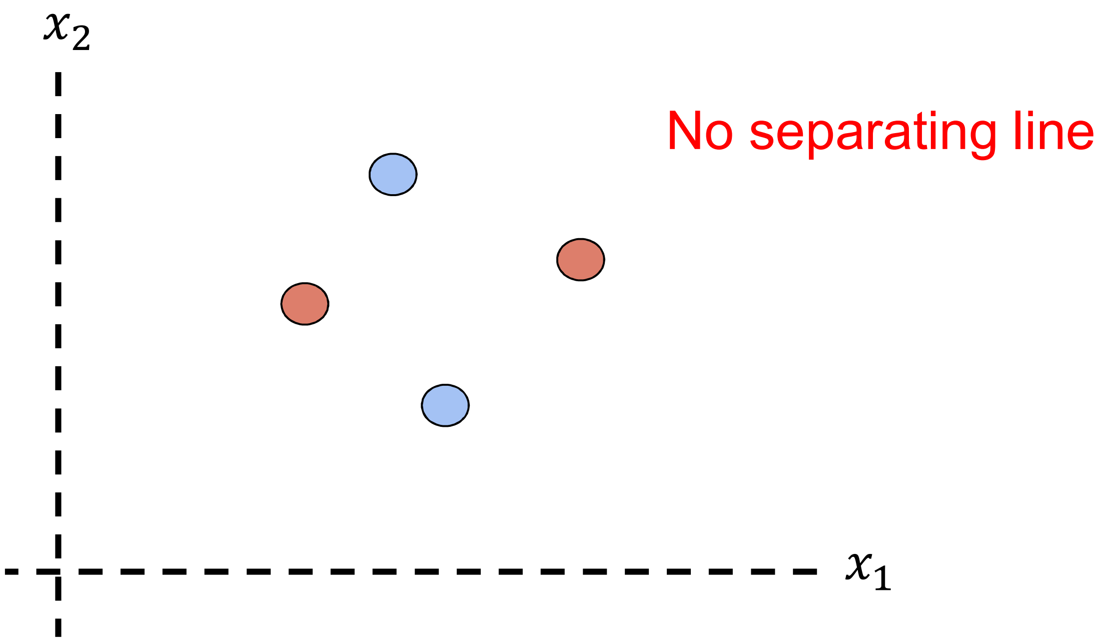
Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration ($N = 4$)



Example VC dimension

- (2D) Linear model $\hat{y} = sign(\mathbf{w}^T\mathbf{x} + b)$
- Configuration ($N = 4$)



VC dimension

- Definition
  - The maximum number of points that can be arrange such that $\mathcal{H}$ can shatter them.

- The VC dimension of a linear model in dimension d is:
  - $d_{VC}(\mathcal{H}_{lin}) = d + 1$

- Capacity increases with the number of **effective** parameters

Growth function

- The **growth function** is a measure of the capacity of the hypothesis set.
- Given a set of N samples and an unrestricted hypothesis set, the value of the growth function is:

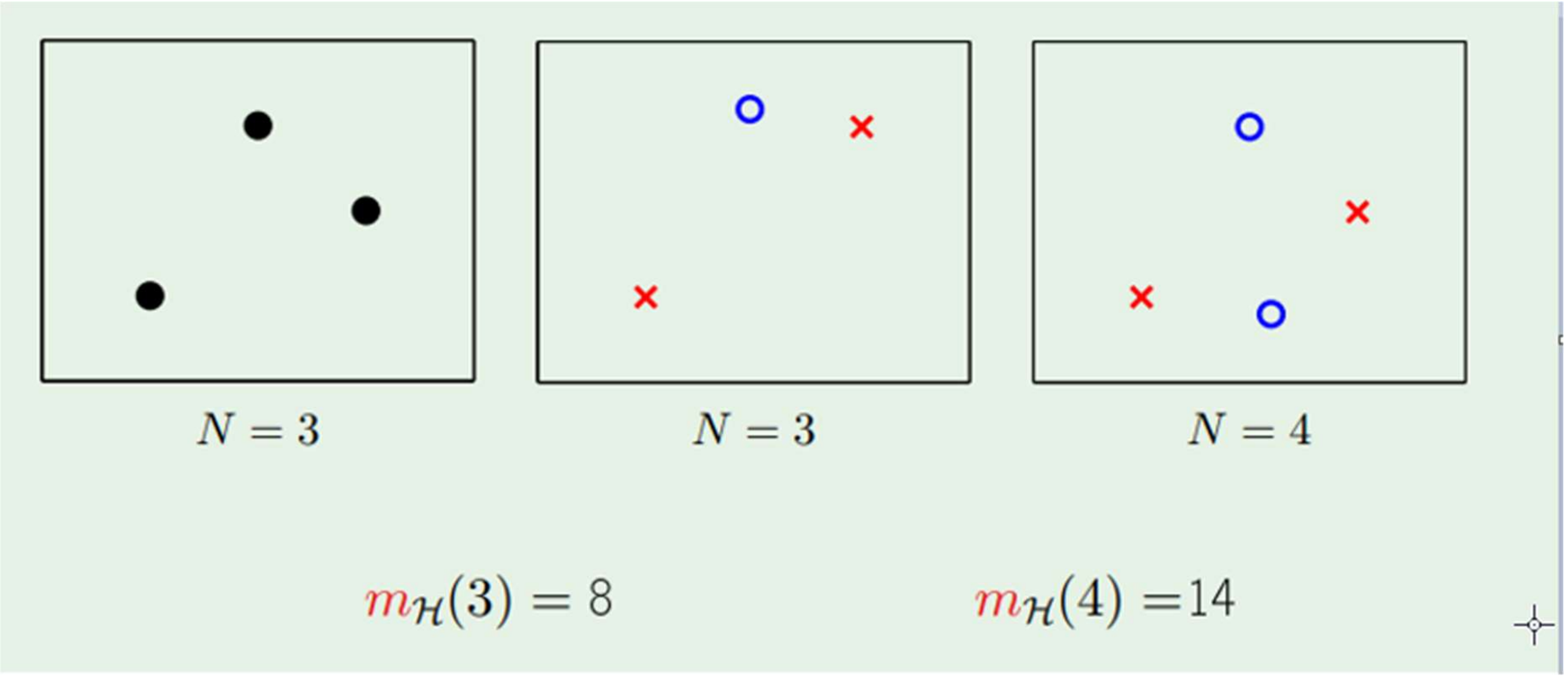
$$m_{\mathcal{H}}(N) = 2^N$$

- For a restricted (limited) hypothesis set the growth function is bounded by:

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{d_{VC}} \binom{N}{i}$$

Maximum power is $N^{d_{VC}}$

Growth function for linear model



Generalization error

- ***Error measure binary classification:***

$$e\big(g(\mathbf{x}_n), f(\mathbf{x}_n)\big) = \begin{cases} 0, & if\ g(\mathbf{x}_n) = f(\mathbf{x}_n) \\ 1, & if\ g(\mathbf{x}_n) \neq f(\mathbf{x}_n) \end{cases}$$

- ***In-sample error:***

$$E_{in}(g) = \frac{1}{N}\sum_{n=1}^{N} e\big(g(\mathbf{x}_n), f(\mathbf{x}_n)\big)$$

- ***Out-of-sample error:***

$$E_{out}(g) = \mathrm{E}_{\mathbf{x}}\Big[e\big(g(\mathbf{x}), f(\mathbf{x})\big)\Big]$$

- **Generalization error:**

$$G(g) = E_{in}(g) - E_{out}(g)$$

Upper generalization bound

- Number of **In-sample** samples, N
- Generalization threshold, ϵ
- Growth function: $m_{\mathcal{H}}()$

- **The Vapnik-Chervonenkis Inequality**

$$P\left[|E_{in}(g) - E_{out}(g)| > \varepsilon\right] \leq 4\, m_{\mathcal{H}}(2N)\, e^{-\frac{1}{8}\varepsilon^2 N}$$

Maximum power is $N^{d_{VC}}$

What makes learning feasible?

- Restricting the capacity of the hypothesis set!
- But are we satisfied?
  - No!
- The overall goal is to have a small $E_{out}(g)$

The goal is small $E_{out}(g)$

$$P\left[|E_{in} - E_{out}| > \varepsilon\right] \leq 4\, m_{\mathcal{H}}(2N)\, e^{-\frac{1}{8}\varepsilon^2 N} = \delta$$

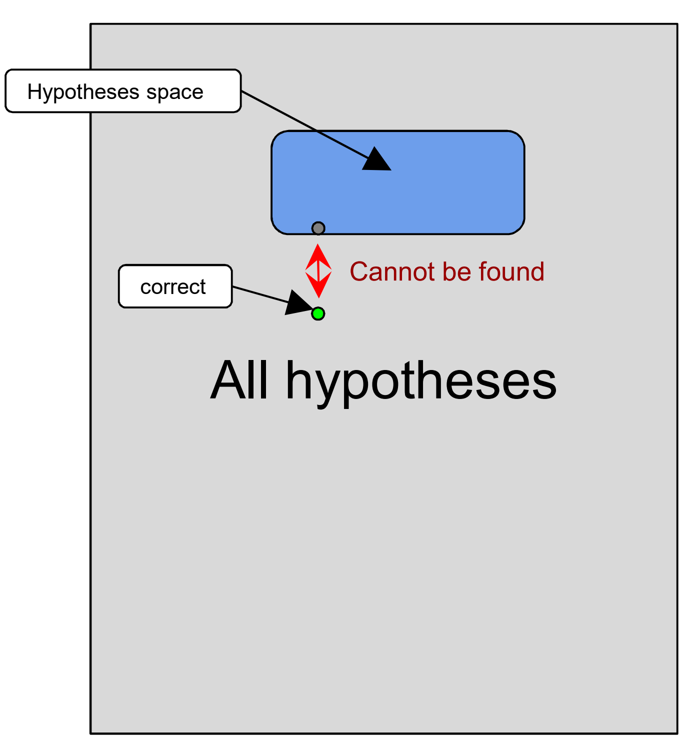
$$\varepsilon = \sqrt{\frac{8}{N} \ln \frac{4 m_{\mathcal{H}}(2N)}{\delta}} = \Omega(N, \mathcal{H}, \delta)$$

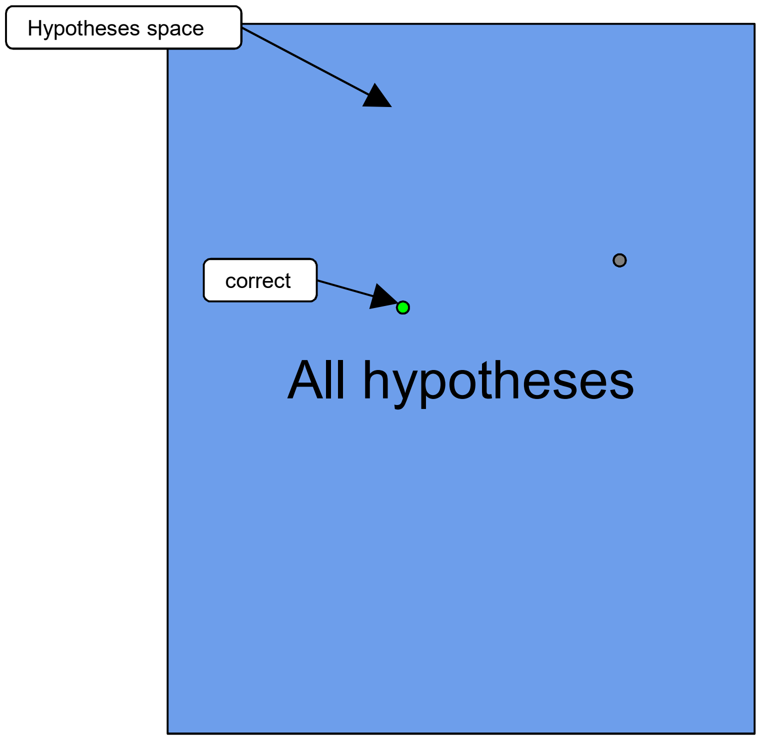
$$P\left[|E_{in} - E_{out}| < \Omega\right] > 1 - \delta$$

With probability $> 1 - \delta$:

$$E_{out} < E_{in} + \Omega$$

A model with wrong hypothesis will never be correct





Progress

- Is learning feasible?
- Model complexity
- **Overfitting**
- Evaluating performance
- Learning from small datasets
- Rethinking generalization
- Capacity of dense neural networks

Noise

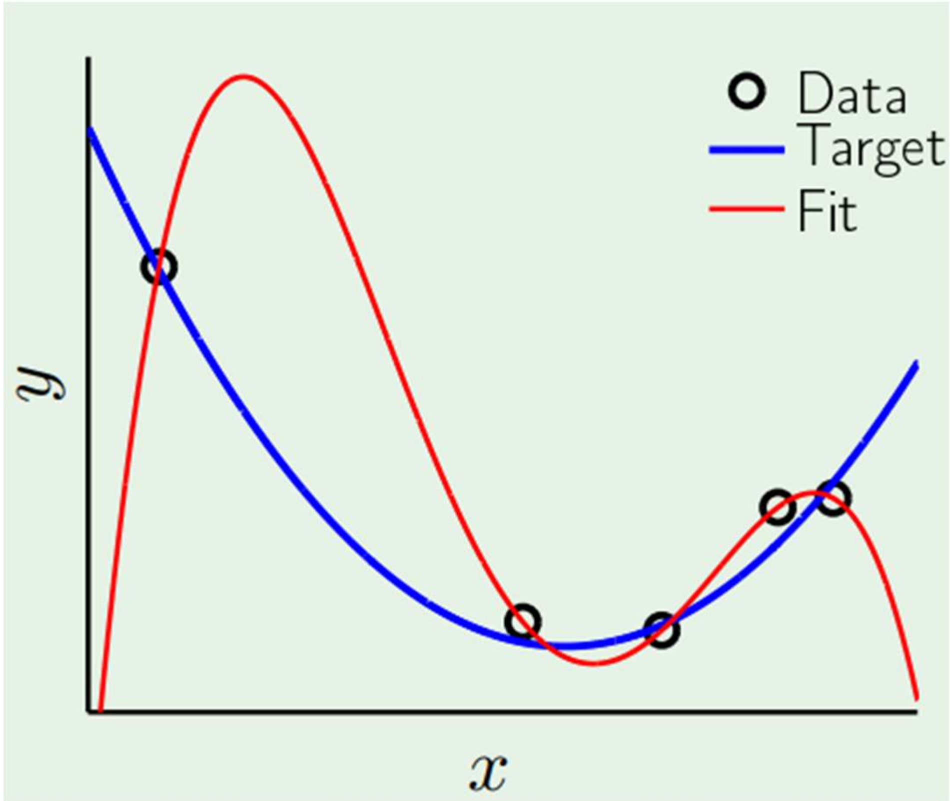
- The **in-sample** data will contain noise.

- Origin of noise:
  - Measurement (sensor) noise
  - The **in-sample** data may not include all parameters
  - Our $\mathcal{H}$ has not the capacity to fit the target function

The role of noise

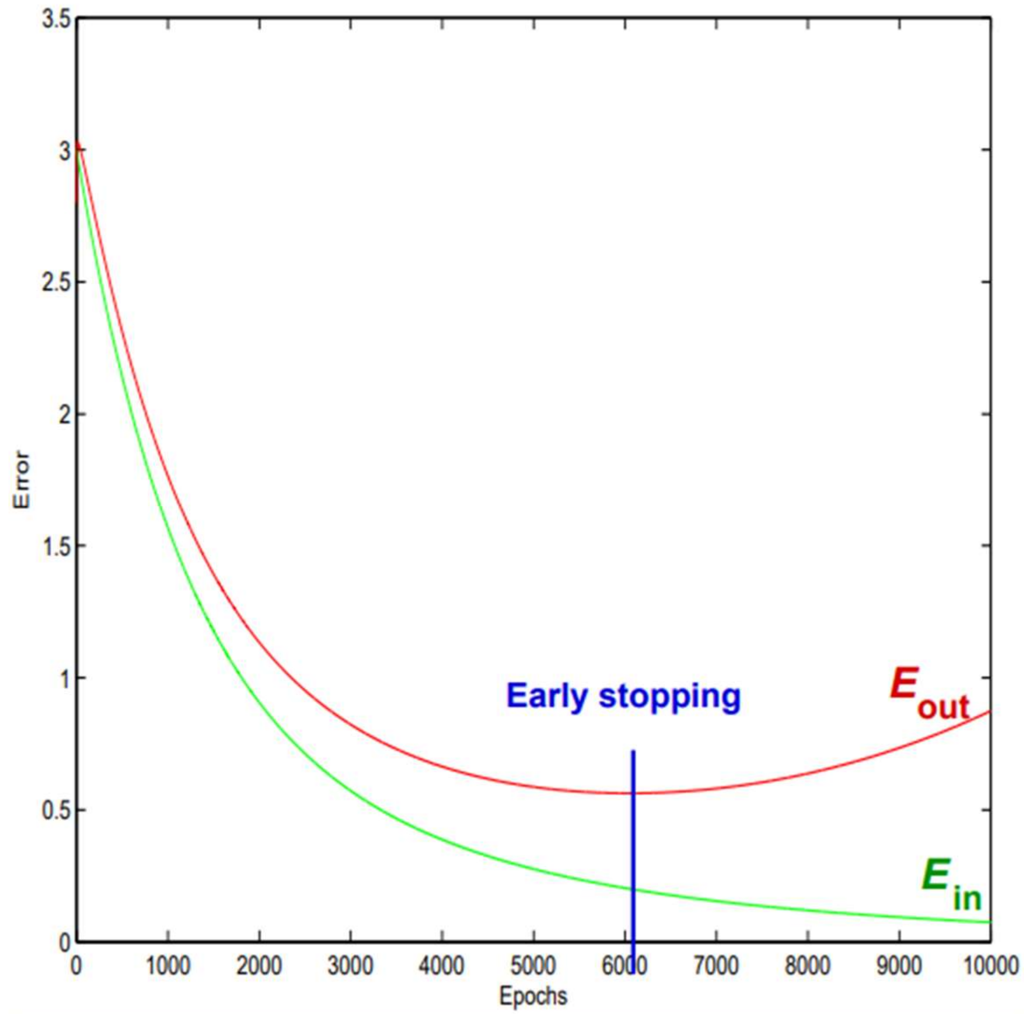
- We want to fit our hypothesis to the target function, not the noise
- Example:
  - Target function: second order polynomial
  - Noisy **in-sample** data
  - Hypothesis: Fourth order polynomial

Result: $E_{in} = 0,\ E_{out}\ is\ huge$



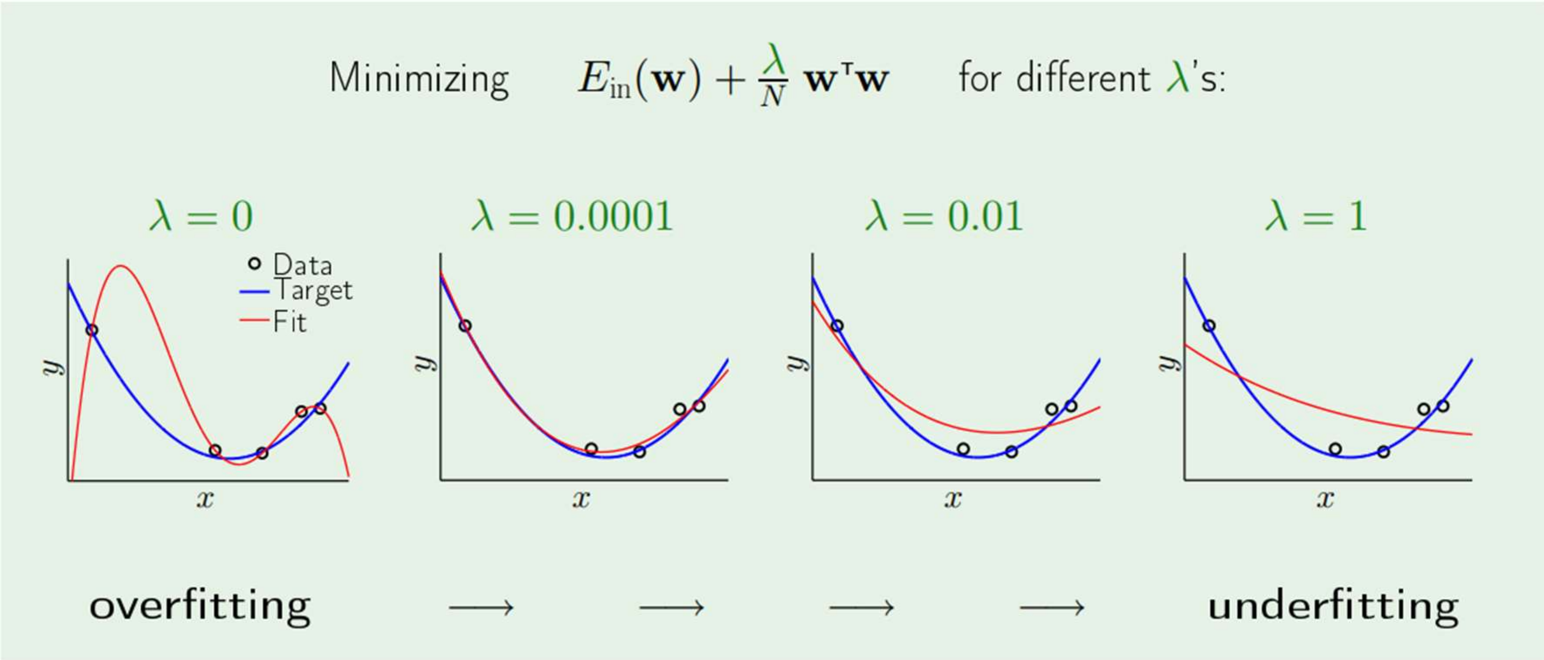
Overfitting - Training to hard

- Initially, the hypothesis is not selected from the data and E_{in} and E_{out} are similar.
- While training, we are exploring more of the hypothesis space
- The effective VC dimension is growing at the beginning, and defined by the number of free parameters at the end



Regularization

- With a tiny weight penalty, we can reduce the effect of noise significantly.



Progress

- Is learning feasible?
- Model complexity
- Overfitting
- **Evaluating performance**
- Learning from small datasets
- Rethinking generalization
- Capacity of dense neural networks

Splitting of data

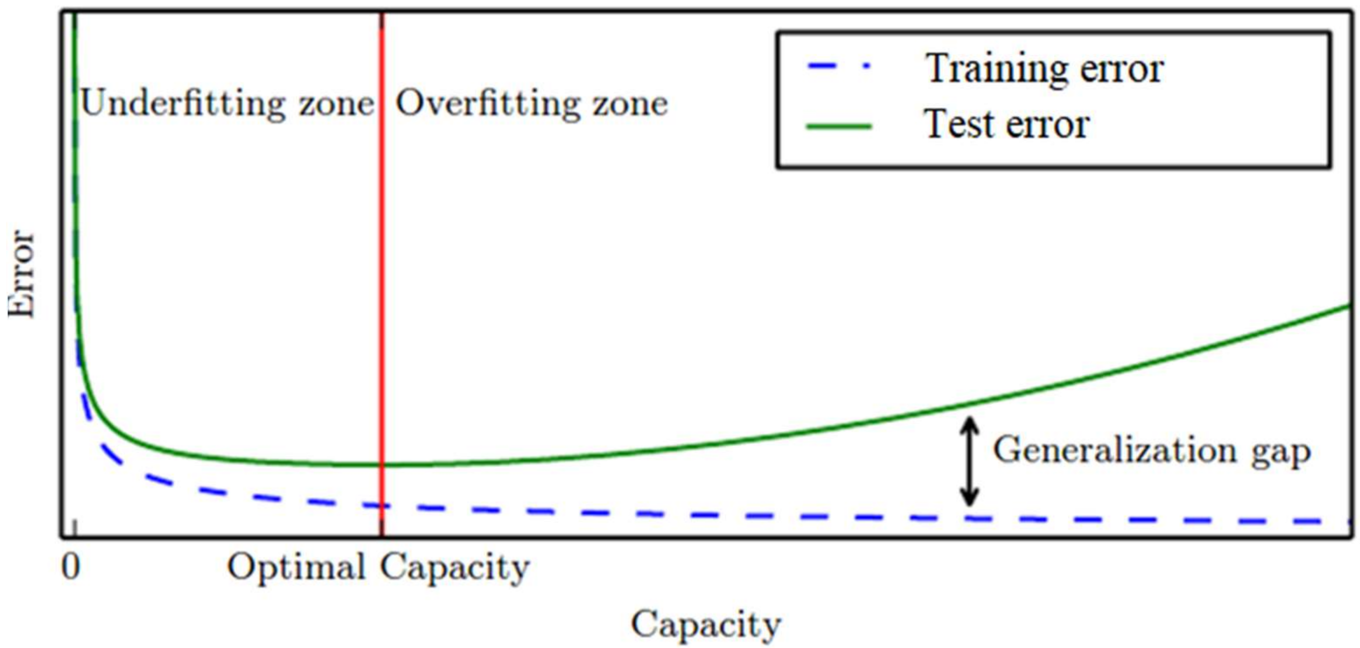
- Training set (60%)
  - Used to train our model
- Validation set (20%)
  - Used to select the best hypothesis
- Test set (20%)
  - Used to get a representative **out-of-sample** error

Important! No peeking

- Keep a dataset that you don’t look at until evaluation (**test set**)

- The test set should be as different from your **training set** as you expect the real world to be

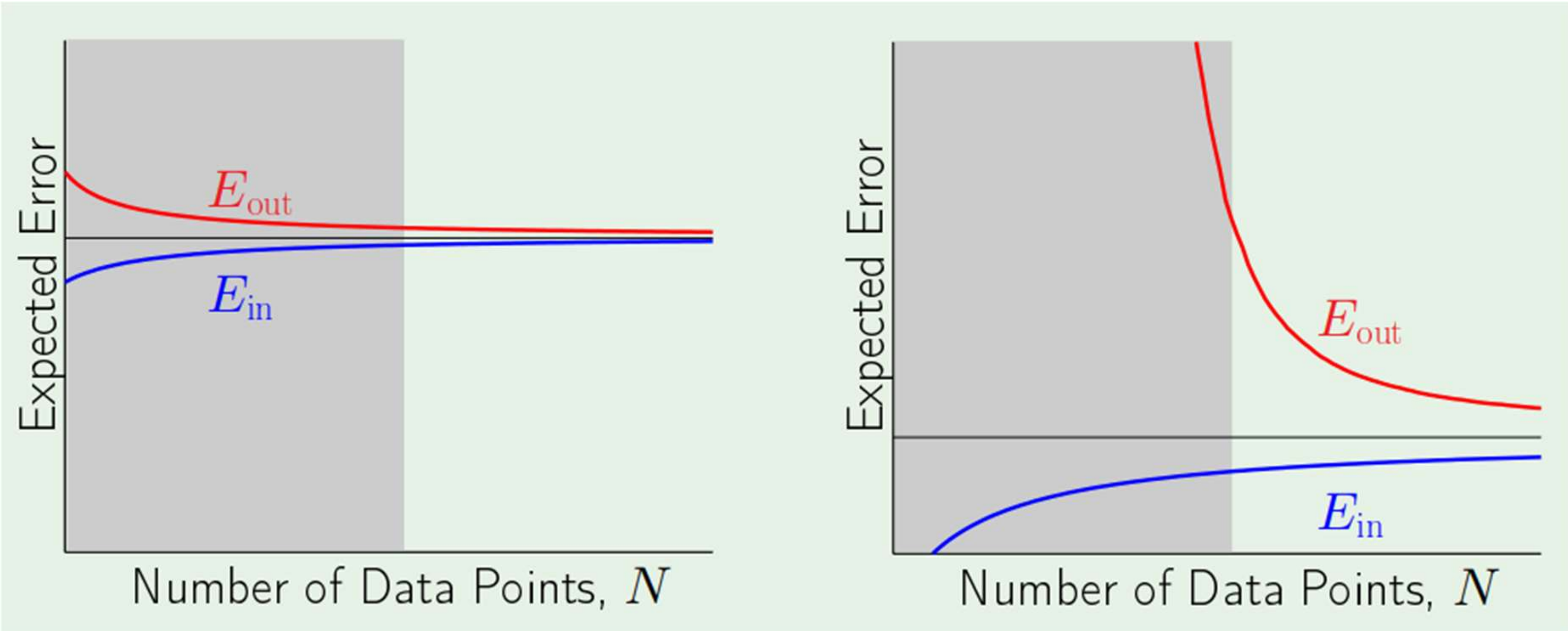
A typical scenario



Learning curves

Simple hypothesis

Complex hypothesis



Progress

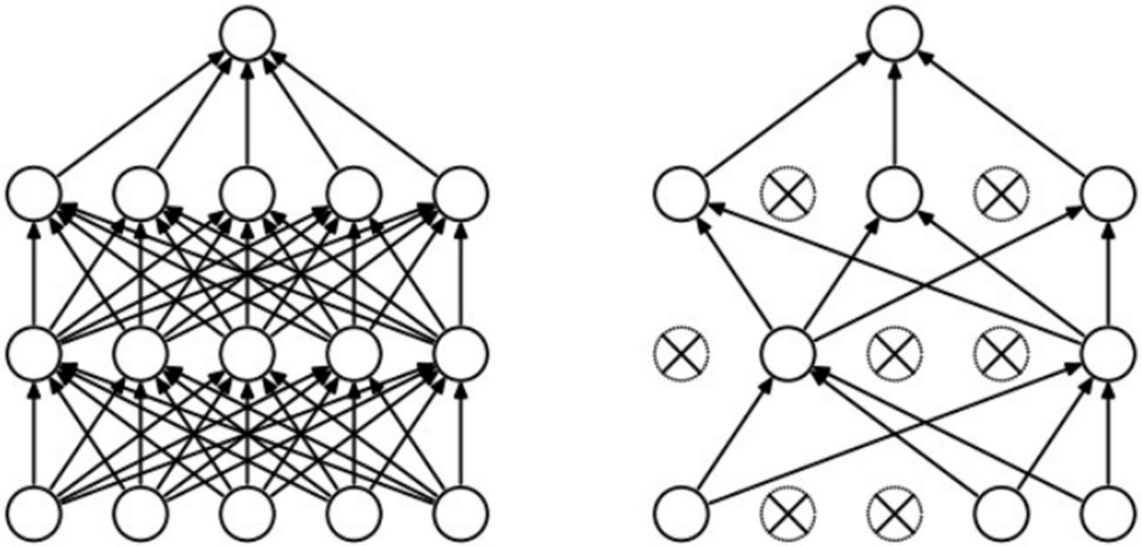
- Is learning feasible?
- Model complexity
- Overfitting
- Evaluating performance
- **Learning from small datasets**
- Rethinking generalization
- Capacity of dense neural networks

Learning from a small datasets

- Regularization
- Dropouts
- Data augmentation
- Transfer learning
- Multitask learning

Dropouts

- Regularization technique
- Drop nodes with probability, p

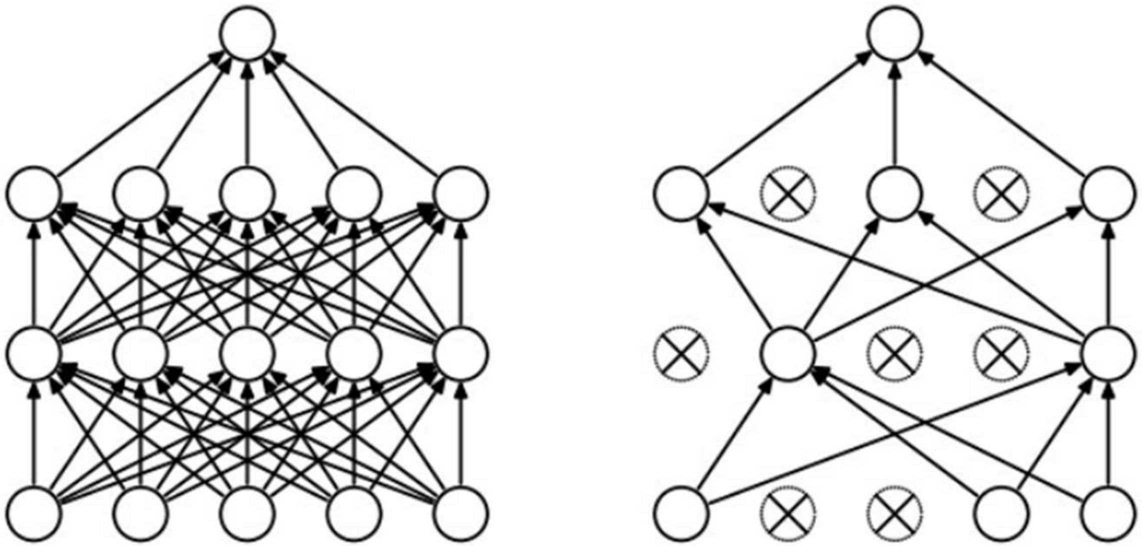


(a) Standard Neural Net

(b) After applying dropout.

Dropouts

- Force the network to make redundant representations
- Stochastic in nature, difficult for the network to memories.
- Remember to scale with $1/p$



(a) Standard Neural Net

(b) After applying dropout.

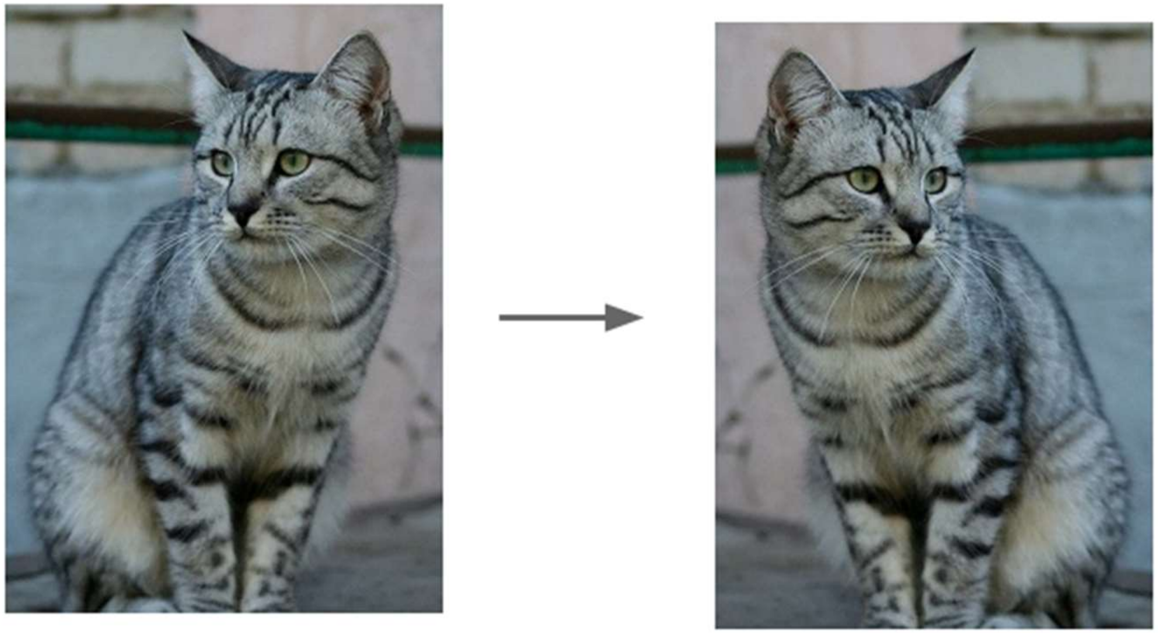
Data augmentation

- Increasing the dataset!

- Examples:
  - Horizontal flips
  - Cropping and scaling
  - Contrast and brightness
  - Rotation
  - Shearing

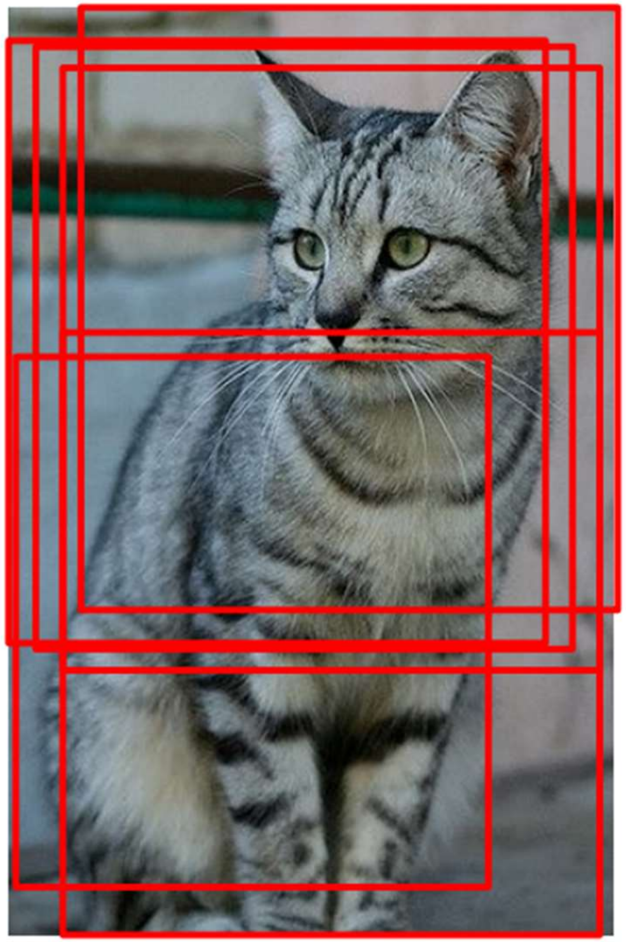
Data augmentation

- Horizontal Flip



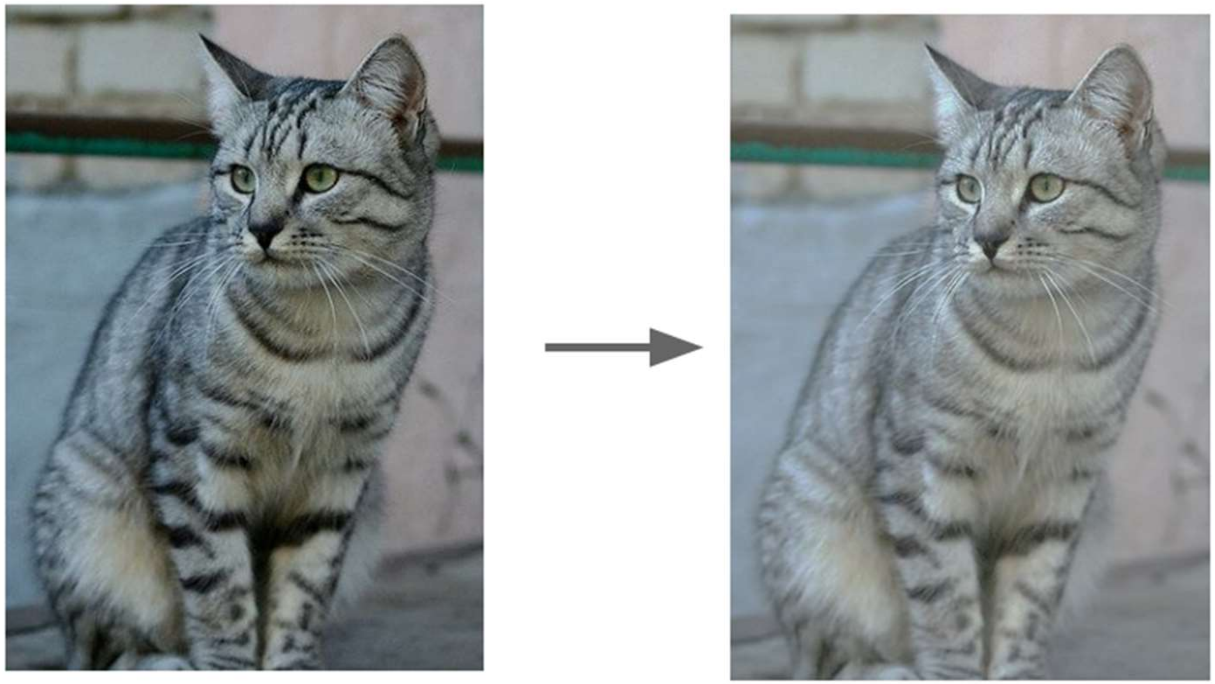
Data augmentation

- Cropping and scaling



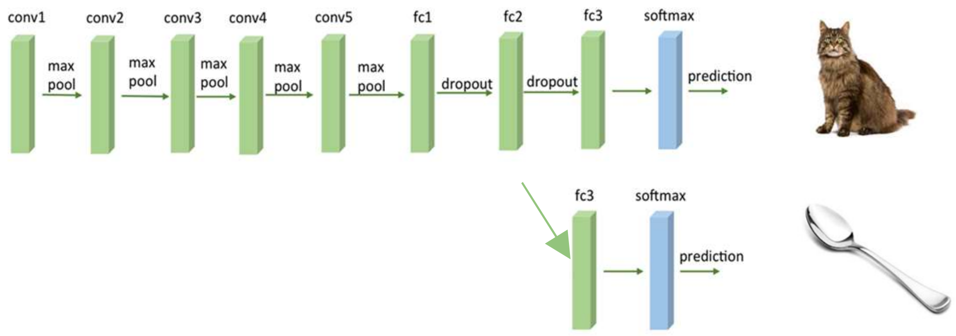
Data augmentation

- Change Contrast and brightness



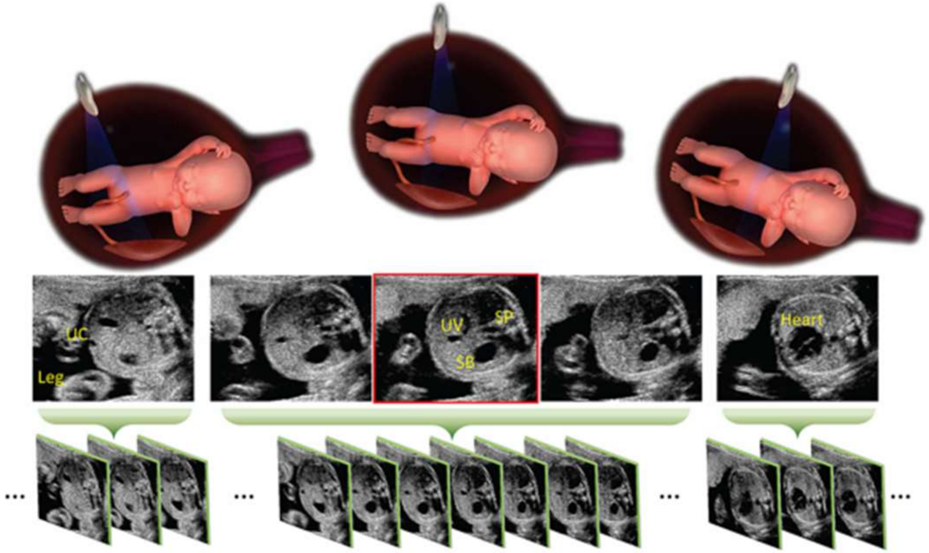
Transfer learning

- Use a pre-trained network
- Neural networks share representations across classes
- You can reuse these features for many different applications
- Depending on the amount of data, finetune:
  - the last layer only
  - the last couple of layers



What can you transfer to?

- Detecting special views in Ultrasound
- Initially far from ImageNet
- Benefit from fine-tuning imagenet features



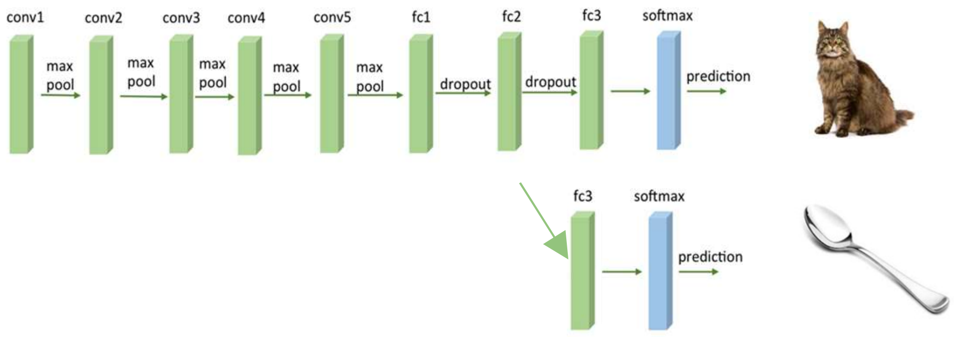
[Standard Plane Localization in Fetal Ultrasound via Domain Transferred Deep Neural Networks]

Transfer learning from pretrained network

- Since you have less parameters to train, you are less likely to overfit.

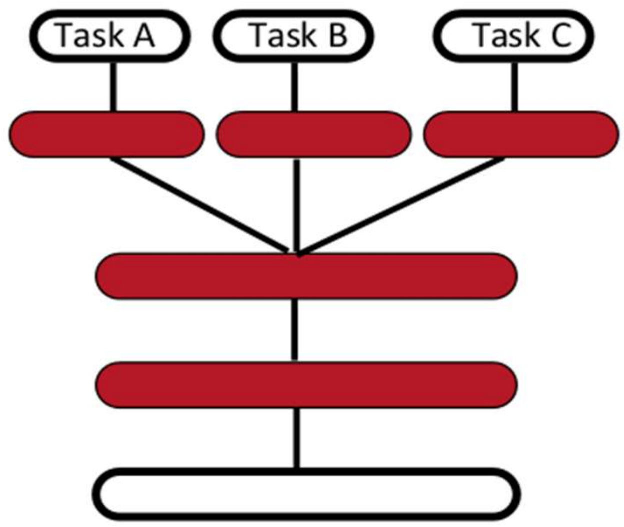
- Need a lot less time to train.

OBS! Since networks trained on ImageNet have a lot of layers, it is still possible to overfit.



Multitask learning

- Many small datasets
- Different targets
- Share base-representation



Progress

- Is learning feasible?
- Model complexity
- Overfitting
- Evaluating performance
- Learning from small datasets
- **Rethinking generalization**
- Capacity of dense neural networks

Is traditional theory valid for deep neural networks?

- “UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION”

- Experiment:
  - Deep neural networks have the capacity to memories many datasets
  - Deep neural networks show small generalization error

Progress

- Is learning feasible?
- Model complexity
- Overfitting
- Evaluating performance
- Learning from small datasets
- Rethinking generalization
- **Capacity of dense neural networks**

Have some fun

- Capacity of dense neural networks
- http://playground.tensorflow.org

Tips for small data

1. Try a pre-trained network

2. Get more data
   a) 1000 images with 10 mins per label is 20 working days…
   b) Sounds like a lot, but you can spend a lot of time getting transfer learning to work

1. Do data-augmentation

2. Try other stuff (Domain-adaption, multitask learning, simulation, etc.)