1 - Introduction

Ludwig Krippahl

FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE NOVA DE LISBOA

Summary

- Course structure and assessment
- AI and the origin of Artificial Neural Networks
- Machine Learning
- The power of nonlinear transformations
- What deep learning offers

Course Overview

Overview

Objectives

- Foundations of deep neural networks
  - Activations, optimizers, training, implementation
- Different architectures and their applications
  - Dense, convolution, recurrent, generative models
- Training and regularization of deep networks
  - Supervised, unsupervised, reinforcement
- Practical experience
  - Tensorflow and Keras

Note: class plan is provisional

Overview

Instructors and assessment

- Ludwig Krippahl (ludi@fct.unl.pt)
  - Lectures, P1
- Cláudia Soares (cam.soares@fct.unl.pt)
  - P2
- Assessment:
  - 2 written tests: theoretical component, 50%
  - 2 assignments: lab component, 50%
  - Must have at least 9.5 on each component to pass.
- Tutorials start on March 24

Main Bibliography

- Course site: http://ap.ssdi.di.fct.unl.pt
- Goodfellow et. al., Deep Learning, MIT Press, 2016
- Skansi, Sandro: Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence , Springer, 2018
- Géron, Aurélien: Hands-on machine learning with Scikit-Learn and TensorFlow, O'Reilly Media, Inc, 2017
- Singh, Pramod and Manure, Avinash: Learn TensorFlow 2.0, Springer 2020

Overview

Software

- Python 3.x + Tensorflow 2; Google Colaboratory
- Easiest installation:
  - Anaconda: www.anaconda.com

Artificial Intelligence

The beginning of AI

- 1956: Dartmouth Summer Research Project on Artificial Intelligence
  - John McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon
  - "proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it"
- Initially, most successful approach of AI was to process rules
  - Expert systems, logic programming, ...
- Rule-based expert systems
  - Rules provided by humans
  - Computer does inference to reach conclusions

The beginning of AI

- Rule-based expert systems

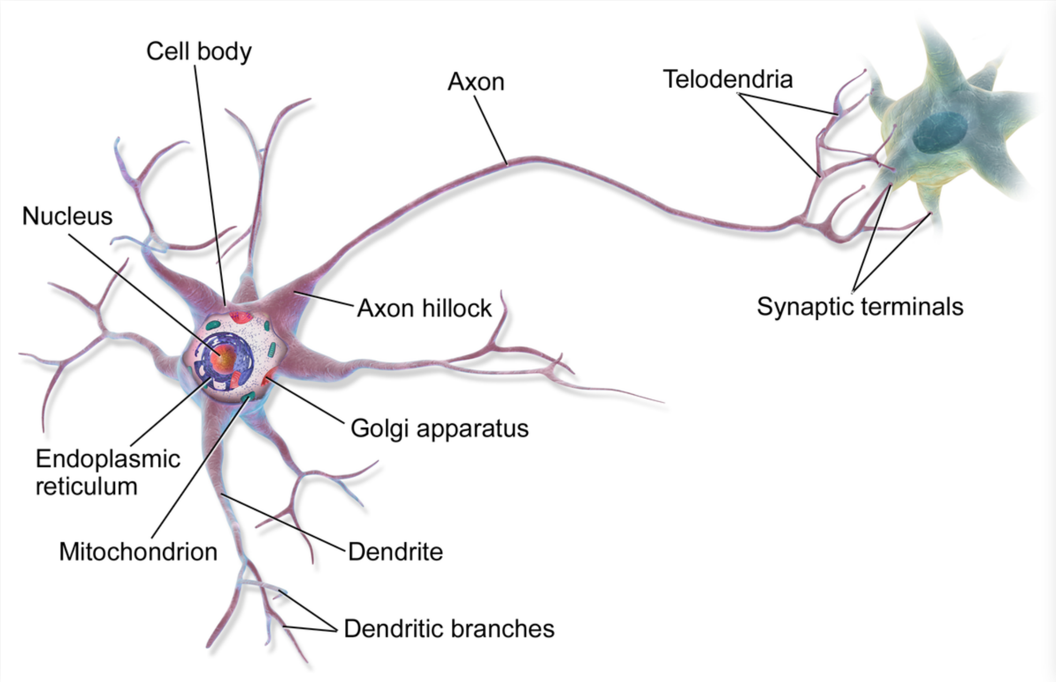
- Rules provided by humans
- Computer does inference to reach conclusions
- E.g. MYCIN, 1975 (Shortliffe, A model of inexact reasoning in medicine)

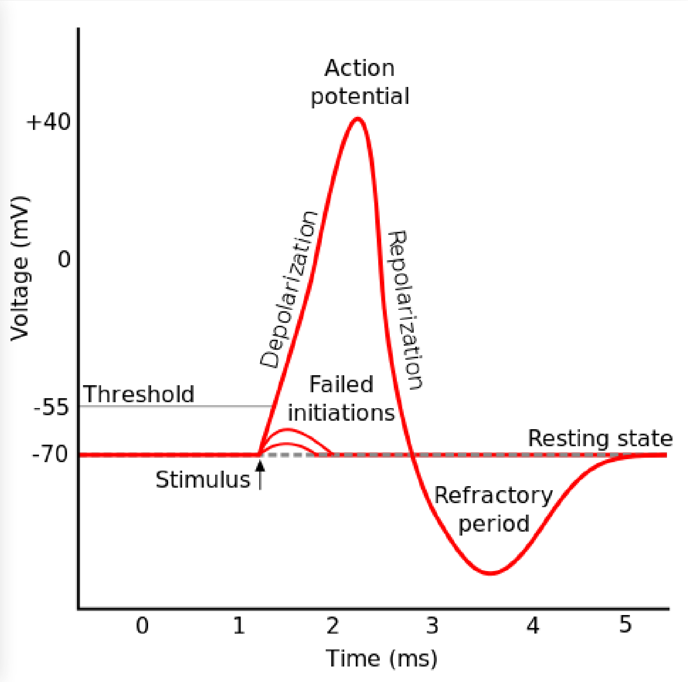
```
If:
(1) the stain of the organism is gram positive, and
(2) the morphology of the organism is coccus, and
(3) the growth conformation of the organism is chains
Then :
there is suggestive evidence (0.7) that the identity of
the organism is streptococcus
```

The beginning of Neural Networks

- The modelling of neurons predates modern AI

• 1943: McCulloch & Pitts, model of neuron

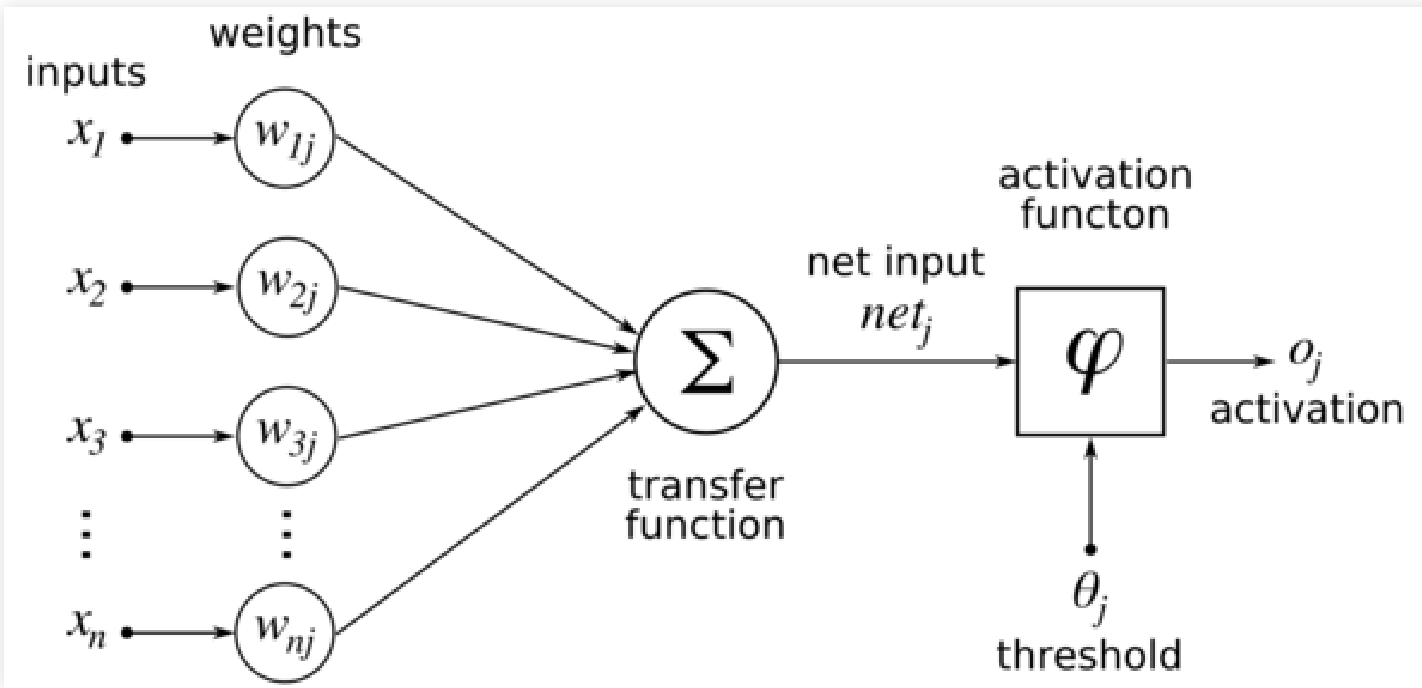




BruceBlaus, Chris 73: CC-BY, source Wikipedia

The beginning of Neural Networks

- The modelling of neurons predates modern AI
  - 1943: McCulloch & Pitts, model of neuron

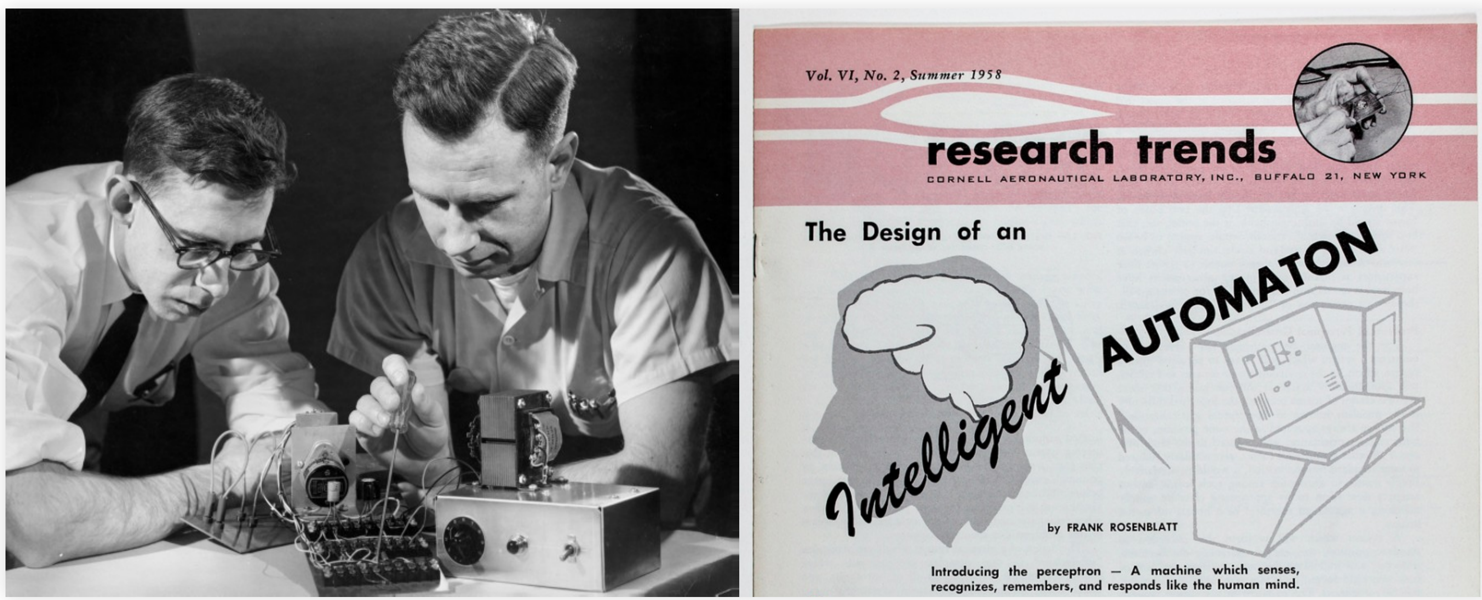


Chrislb, CC-BY, source Wikipedia

The perceptron, the first learning machine

- 1958: Rosenblatt, perceptron could learn to distinguish examples

«the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.»
New York Times, 1958



Wightman and Rosenblatt. Source: Cornell Chronicle

Perceptron

- Linear combination of the inputs and a threshold function:

$$y = \sum_{j=1}^{d} w_j x_j + w_0 \qquad s(y) = \begin{cases} 1, & y > 0 \\ 0, & y \leq 0 \end{cases}$$

- Training rule for the perceptron:

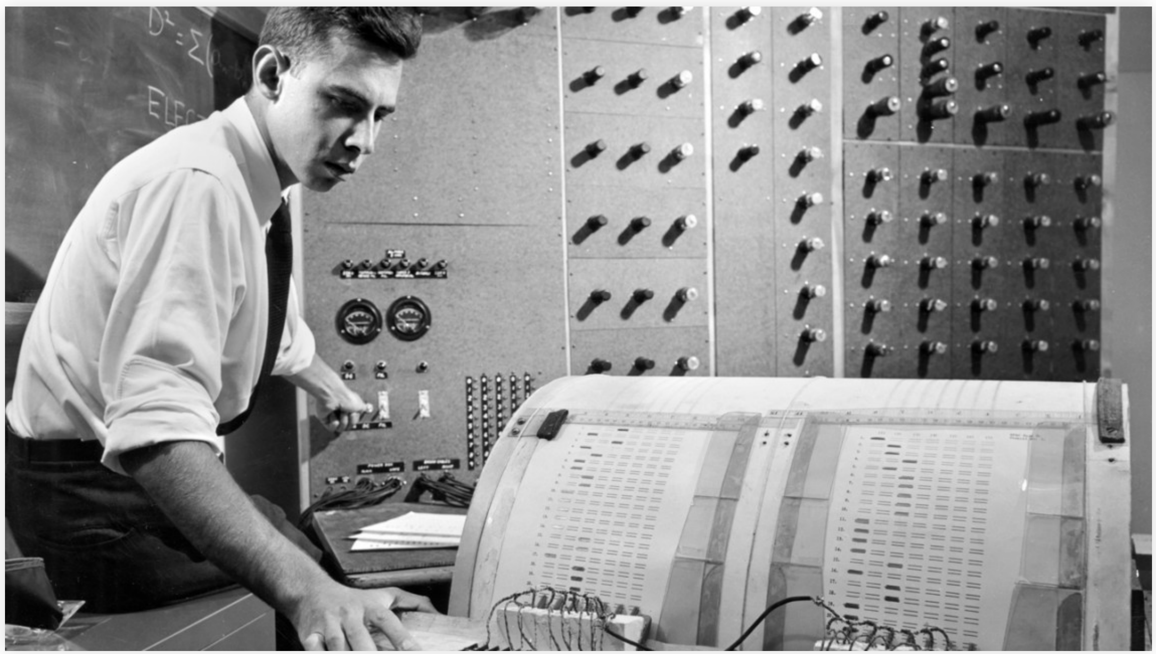
$$w_i = w_i + \Delta w_i \qquad \Delta w_i = \eta(t - o)x_i$$

- Adjust weights slightly to correct misclassified examples.
- Greater adjustment to those with larger inputs.

Perceptron

- First implemented on an IBM 704, 1958

- Learned to distinguish between cards punched on the right and punched on the left after 50 examples



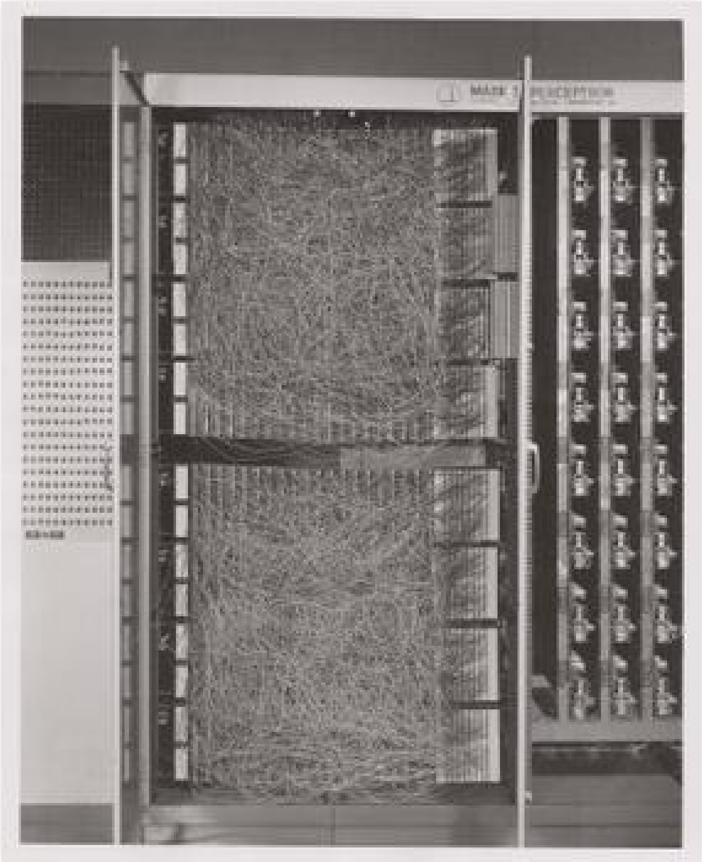
Rosenblatt and IBM 704. Source: Cornell Chronicle

Perceptron

- But then was actually built as a machine

Camera with 20x20 pixels, for image recognition

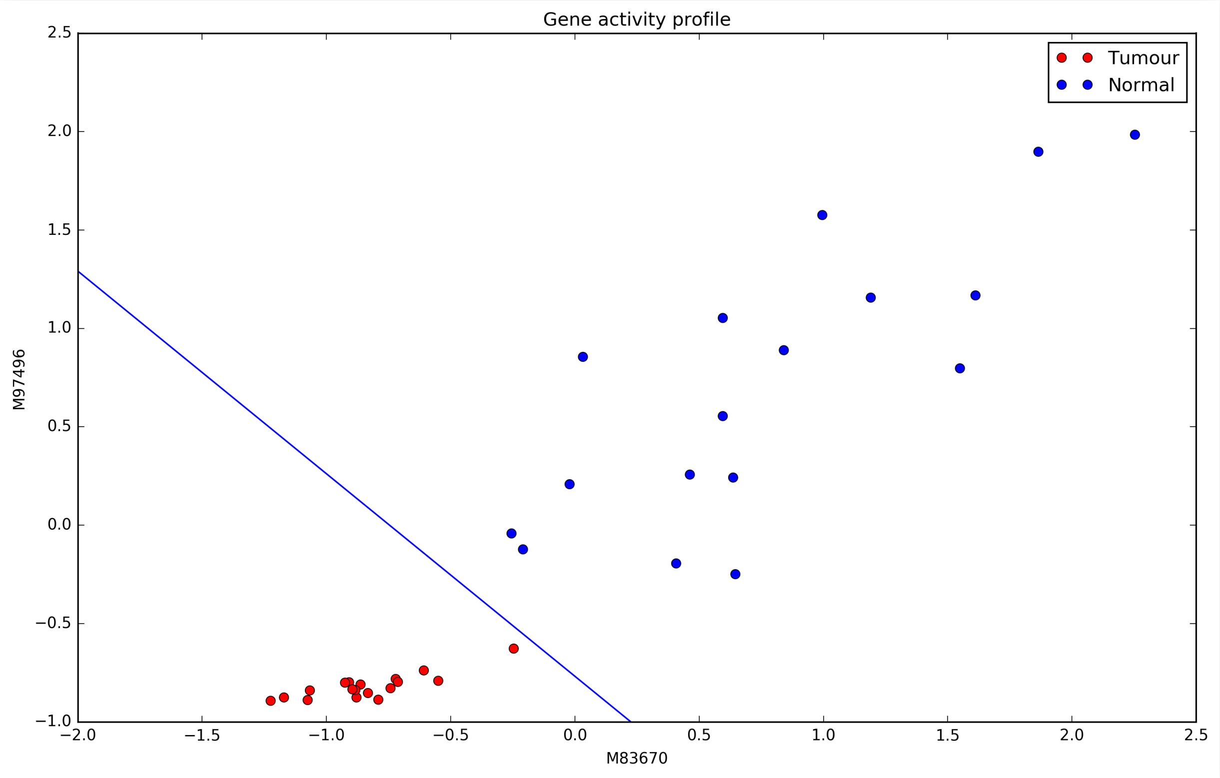
Electric motors to adjust potentiometers for the weights of the inputs



Mark I Perceptron (Wikipedia)

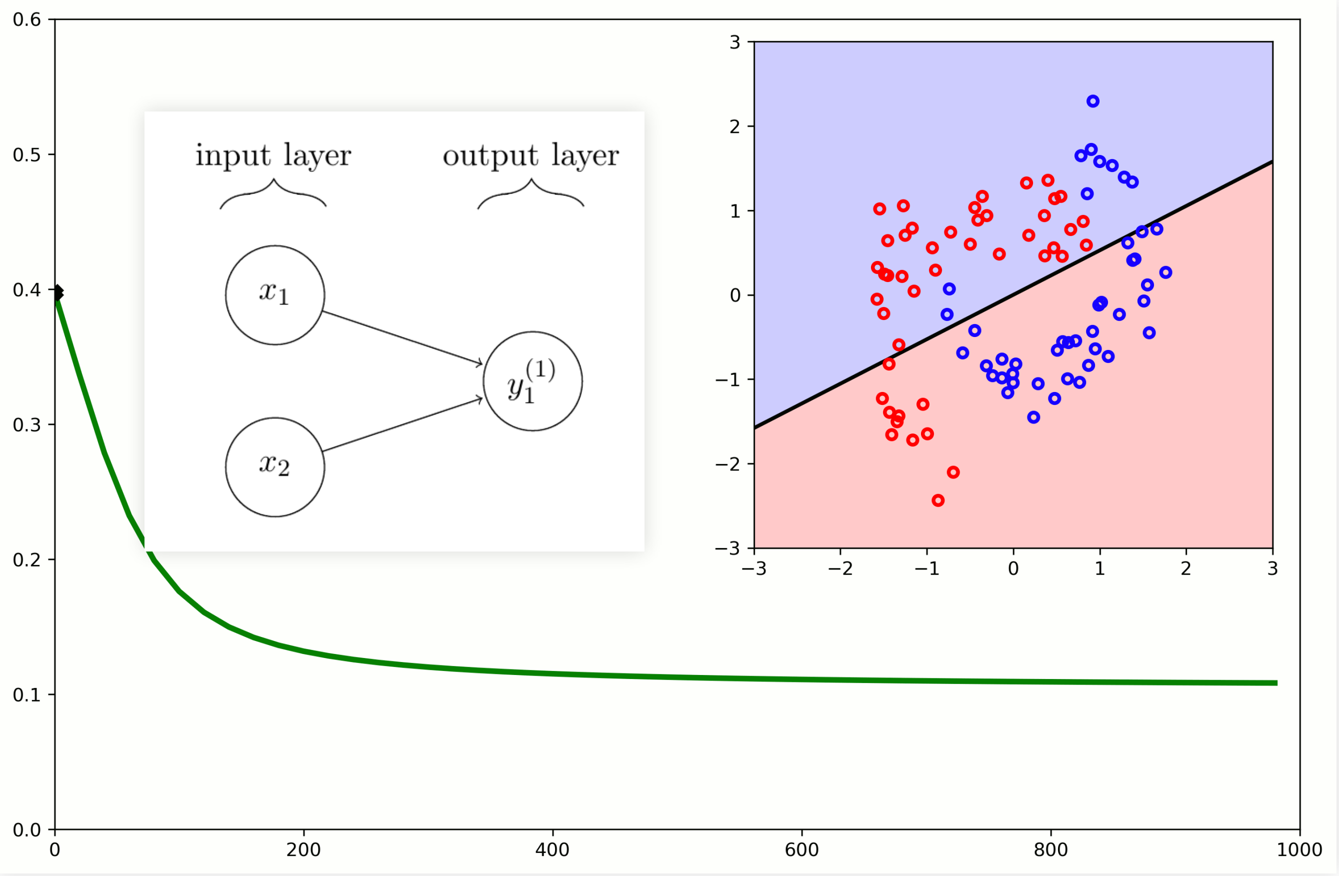
Perceptron

- Seemed a promising start
- But the perceptron is just a linear model



Perceptron

- It's a single neuron, so a linear classifier:



Perceptron is similar to Logistic Regression

- Let $g(\vec{x}, \widetilde{w})$ be a function of the probability of $\vec{x}$ in class 1

$$g(\vec{x}, \widetilde{w}) = P(C_1|\vec{x})$$

- We want to separate classes on:

$$P(C_1|\vec{x}) = P(C_0|\vec{x}) = 1 - P(C_1|\vec{x})$$

- This is solved by minimizing using this function:

$$g(\vec{x}, \widetilde{w}) = \frac{1}{1 + e^{-(\vec{w}^T \vec{x} + w_0)}}$$

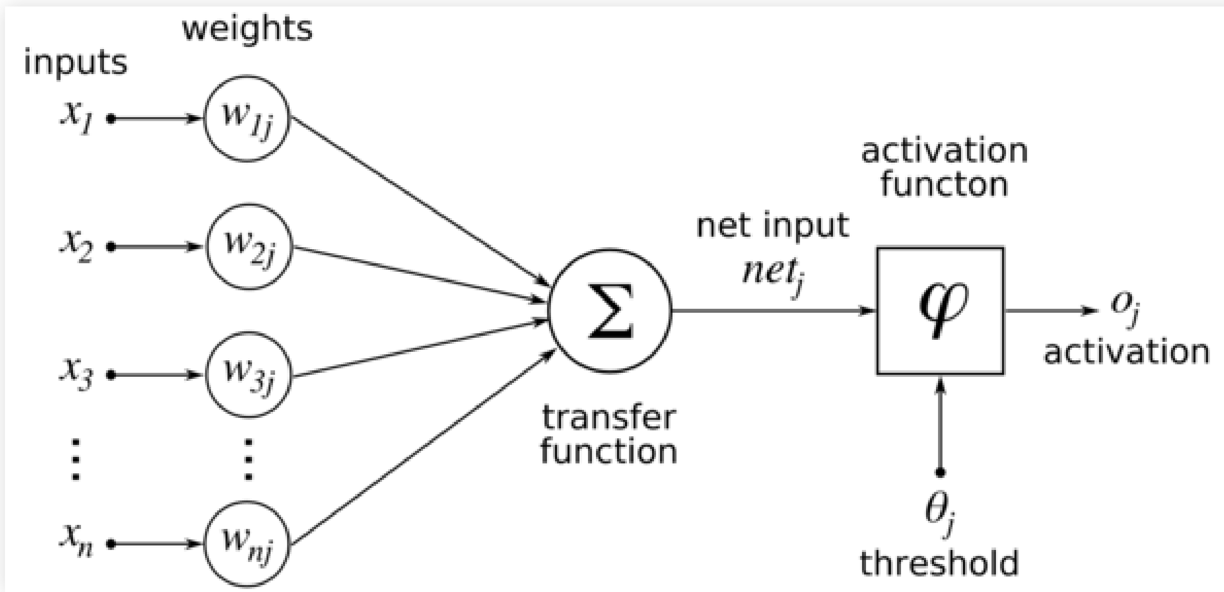
- And minimizing error measured as the logistic loss:

$$E(\widetilde{w}) = -\frac{1}{N} \sum_{n=1}^{N} [t_n \ln g_n + (1 - t_n) \ln(1 - g_n)]$$

Logistic Regression

- We can represent logistic regression in this way:

$$g(\vec{x}, \widetilde{w}) = \frac{1}{1 + e^{-(\vec{w}^T \vec{x} + w_0)}}$$



- This is an artificial neuron with sigmoid activation

Neural Networks

- A very promising early start with neuron and perceptron:
  - 1943: McCulloch & Pitts, model of neuron
  - 1958: Rosenblatt, perceptron and learning algorithm
- But these turned out to be equivalent to generalized linear models
  - And in 1969 Perceptrons (Minsky, Papert): need fully connected networks

1960-1980s: "AI Winter", in particular ANN

- Logic systems ruled AI in this period (e.g. Prolog)
- 1970s: The equivalent of backpropagation appeared in other contexts
- 1986: Rumelhart, Hinton, Williams, backpropagation can be used for multi-layer networks

Machine Learning

What is machine learning?

- "Field of study that gives computers the ability to learn without being explicitly programmed"
(Samuel, 1959)
- "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)

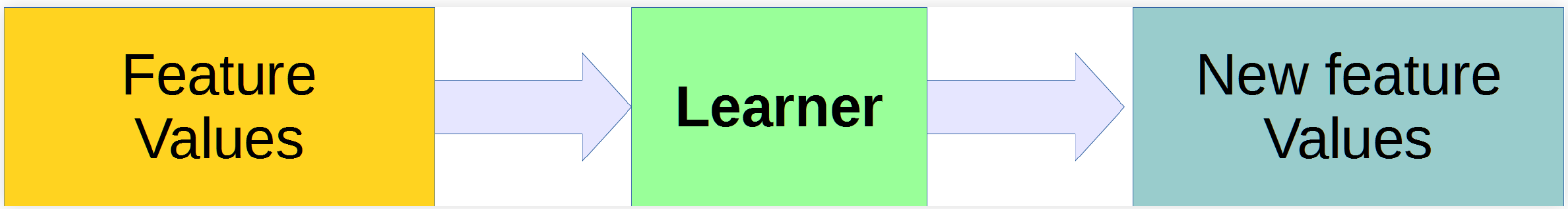
Machine Learning problem

- A task that the system must perform.
- A measure of its performance
- The data used to improve its performance
- Examples:
  - Spam filtering
  - Image classification
  - Medical diagnosis
  - Speech recognition
  - Autonomous driving
  - Clustering, feature representation, ...

Four basic kinds of ML problems

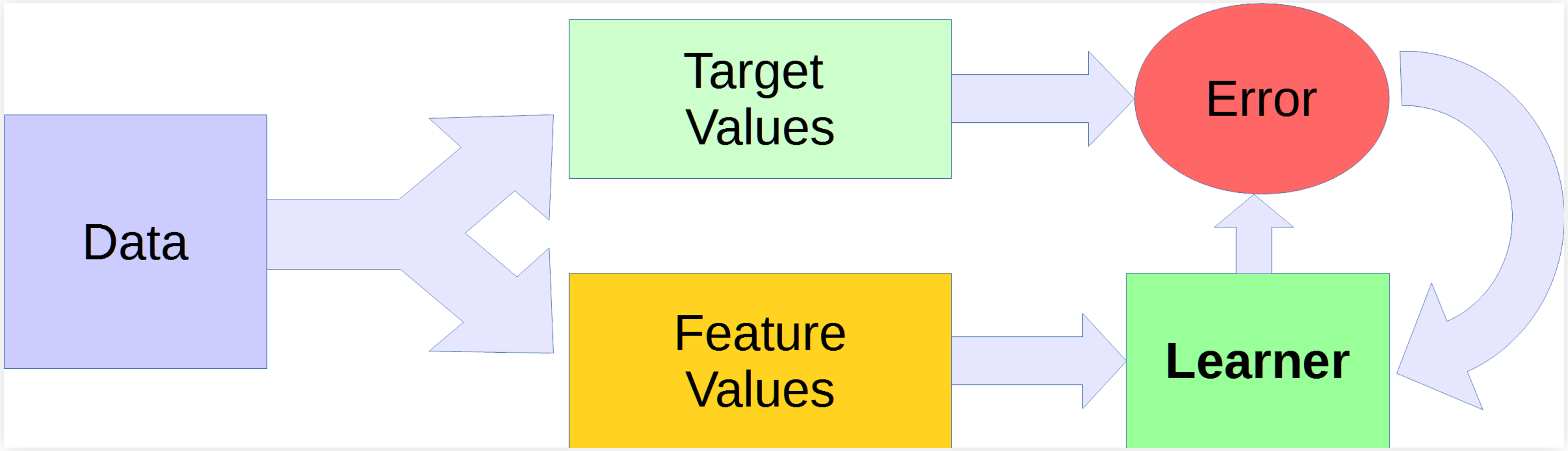
- Unsupervised learning

- No need for labels in data; model structure
- Clustering is a common example, but we will see applications in deep learning



Four basic kinds of ML problems

- Unsupervised learning
- Supervised learning
  - Uses labelled data and aims at predicting classes or values
  - Continuous values: Regression
  - Discrete classes: Classification



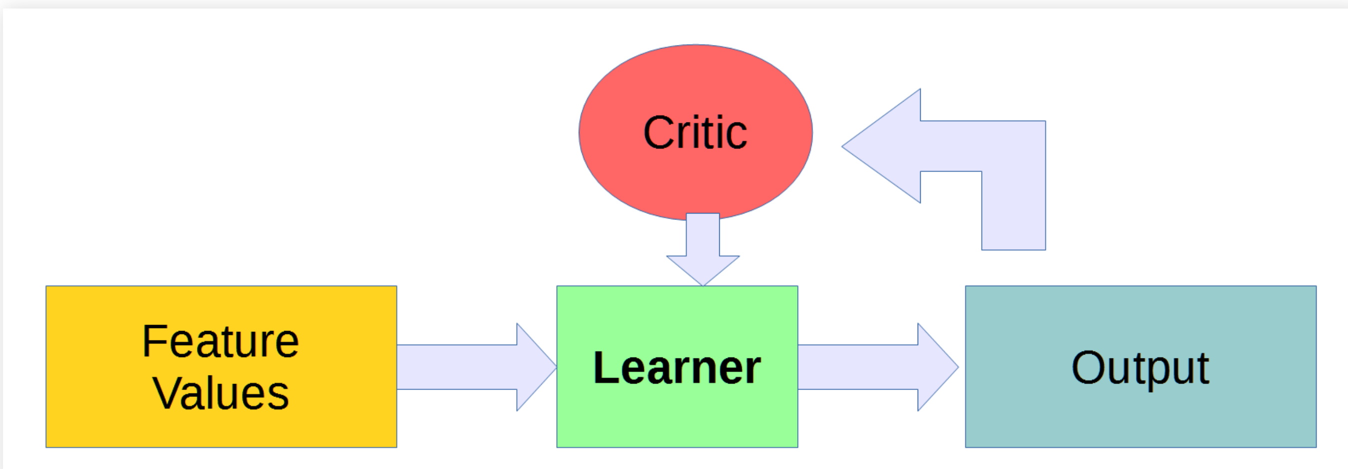
Four basic kinds of ML problems

- Unsupervised learning
- Supervised learning
- Semi-supervised learning
  - Mixes labeled and unlabeled data
  - Can be useful to increase size of data set

Four basic kinds of ML problems

- Unsupervised learning
- Supervised learning
- Semi-supervised learning
- Reinforcement learning

• Optimize output without immediate feedback for each instance



Can solve different kinds of problems

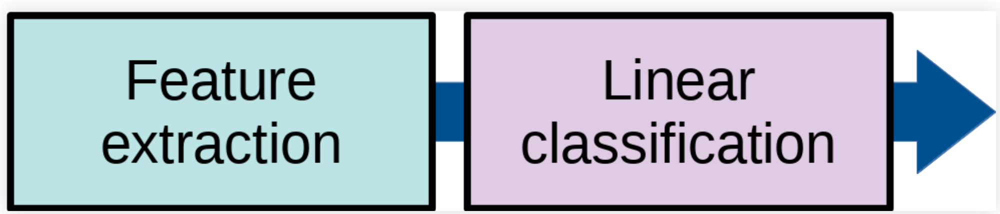
- Extracting new features and finding relations
  - Unsupervised learning
- Approximating a target
  - Supervised learning
- Optimizing policy
  - Reinforcement learning

Traditional approach:

- Use very different models, optimizations, etc.

The rise of machine learning

- Statistical methods for regression and classification predate AI
- Least squares linear regression used since 1800s
  - Legendre (1805) and Gauss (1809), for predicting planetary movement
- Probit regression (Bliss, 1934; Fisher 1935)
  - Like logistic regression but uses inverse of cumulative normal
- Logistic regression (Wilson & Worcester, 1943; Berkson, 1944)



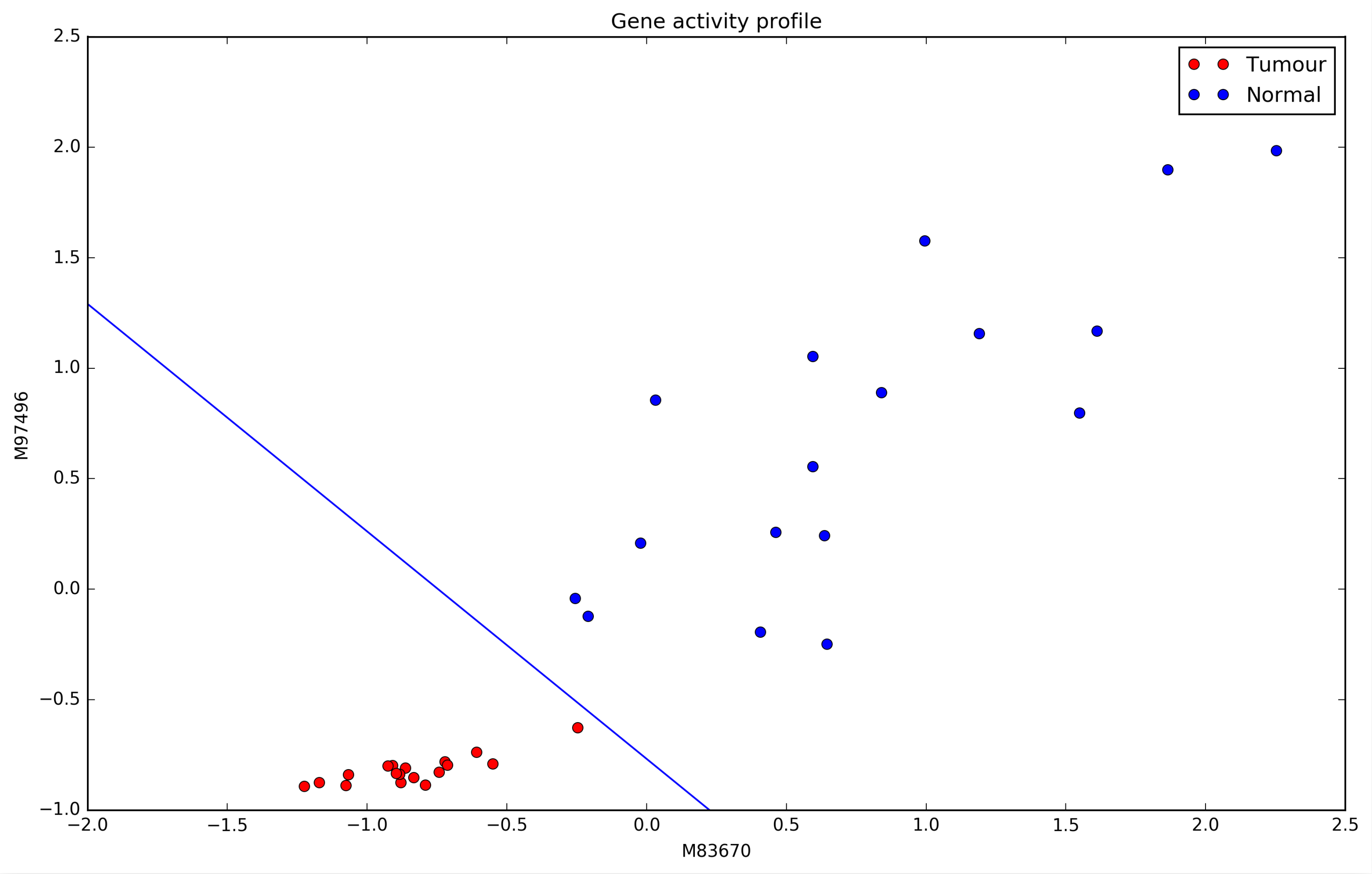
The rise of machine learning

- In the 1990s, AI shifted from knowledge-driven to data-driven with new ML algorithms
- E.g. 1992 Vapnik et. al. publish the kernel trick for SVM

- 1995: SVM (Cortes & Vapnik), Random Forest (Ho)
- 1997: Multi-layered and convolution networks for check processing USA (leCun)
- 1998: MNIST database (LeCun). Benchmarks, libraries and competitions

The power of nonlinearity

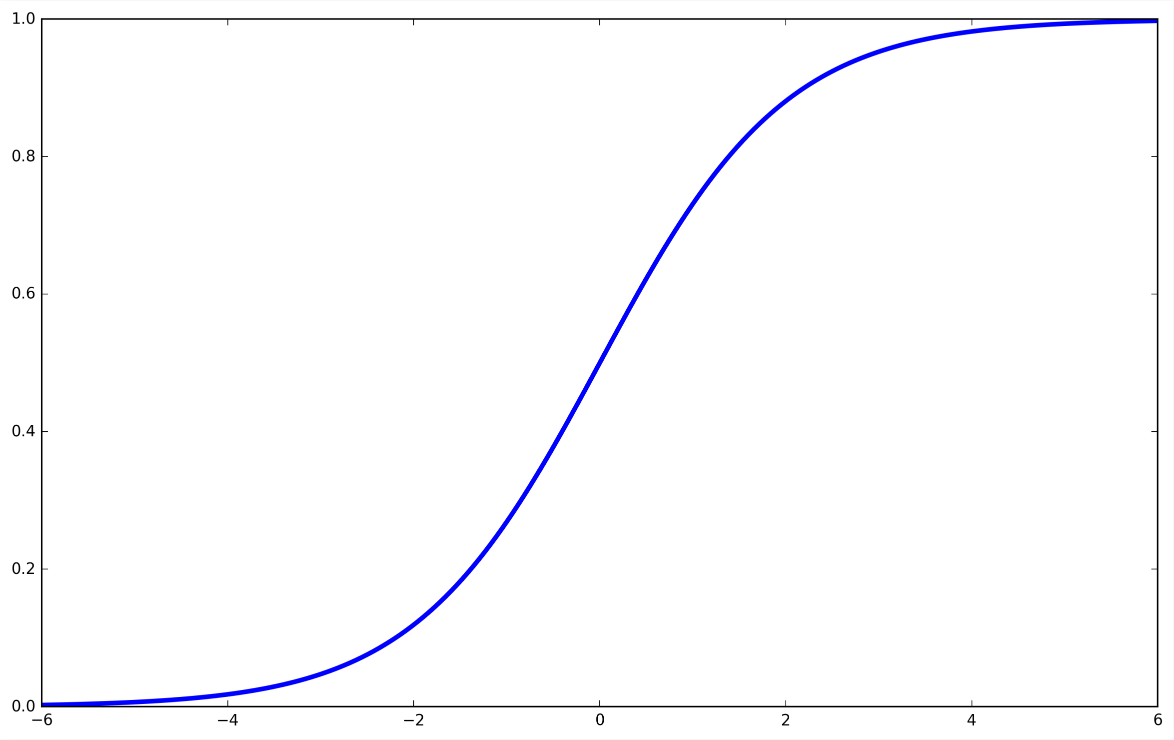
Linear classification



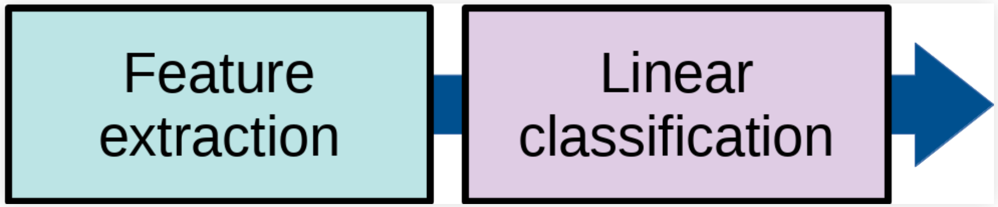
Nonlinearity

Linear classification, e.g. Logistic Regression

$$g(\vec{x}, \widetilde{w}) = P(C_1|\vec{x}) \qquad g(\vec{x}, \widetilde{w}) = \frac{1}{1 + e^{-(\vec{w}^T \vec{x} + w_0)}}$$



Linear classification, e.g. Logistic Regression



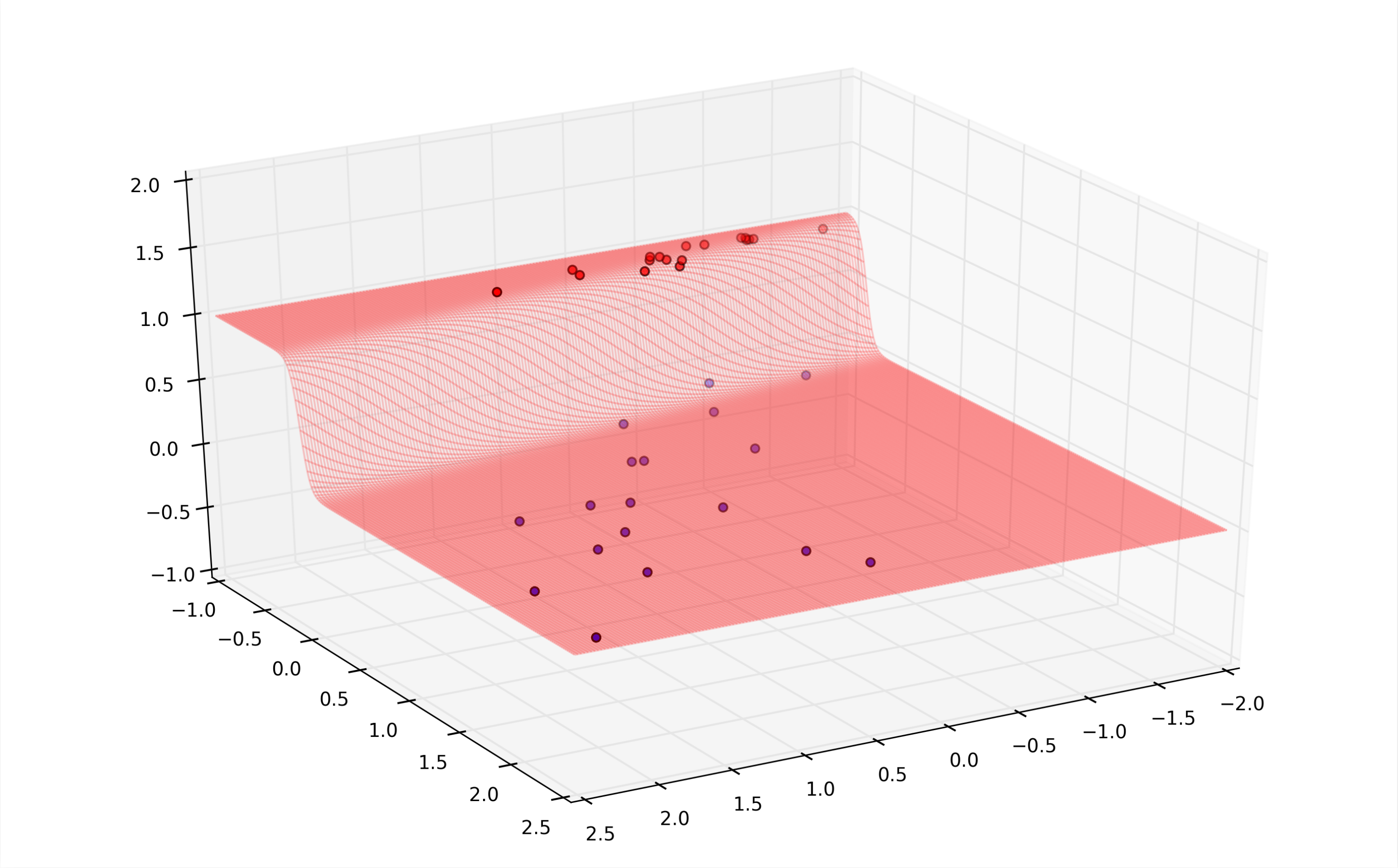
$$g(\vec{x}, \widetilde{w}) = P(C_1 | \vec{x}) \qquad g(\vec{x}, \widetilde{w}) = \frac{1}{1 + e^{-(\vec{w}^T \vec{x} + w_0)}}$$

- Find $\widetilde{w}$ by minimizing logistic loss:

$$E(\widetilde{w}) = -\frac{1}{N} \sum_{n=1}^{N} \left[t_n \ln g_n + (1 - t_n) \ln(1 - g_n) \right]$$

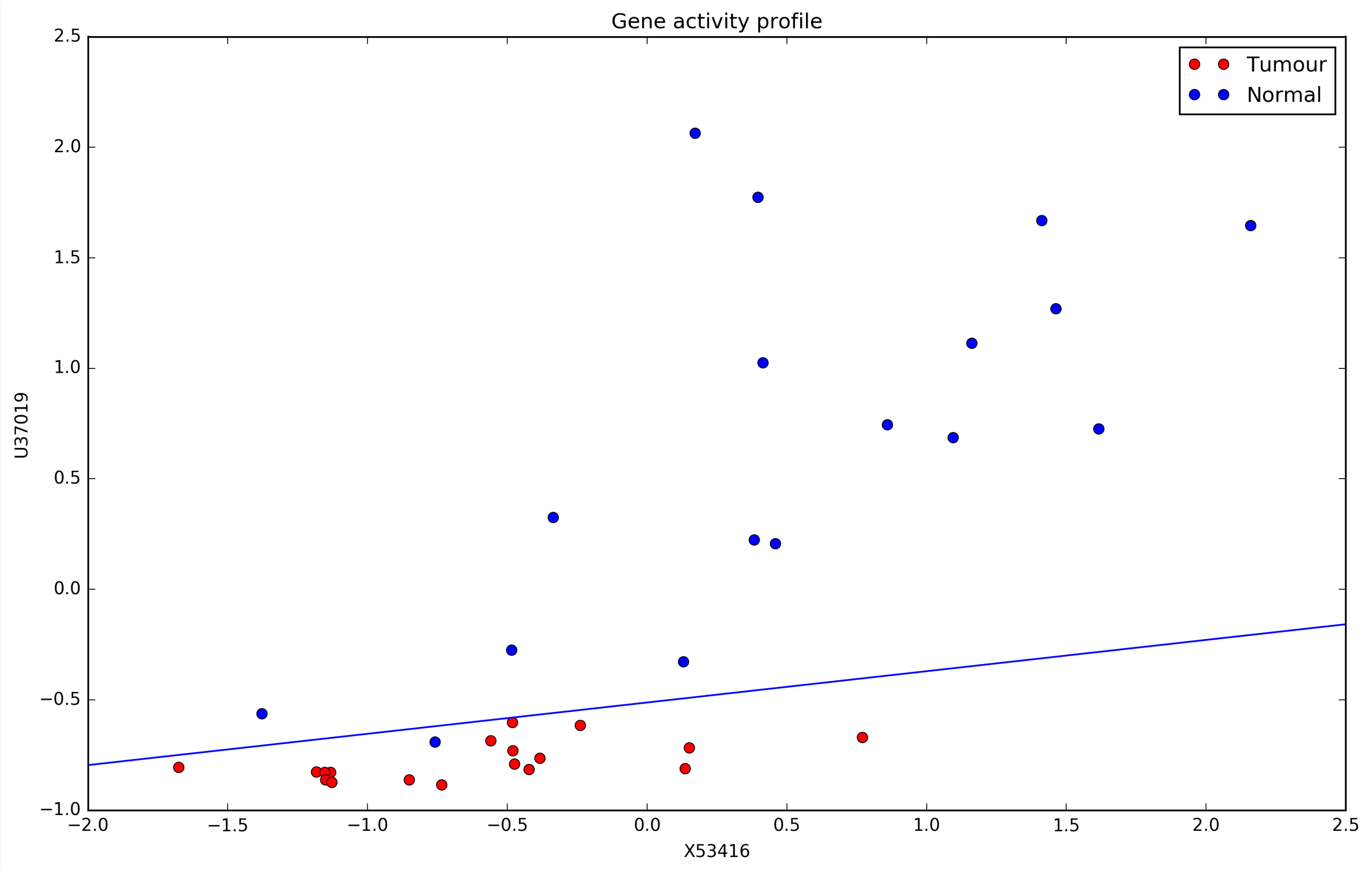
- This works fine for linearly separable sets
  - but not so well otherwise

Linear classification, e.g. Logistic Regression



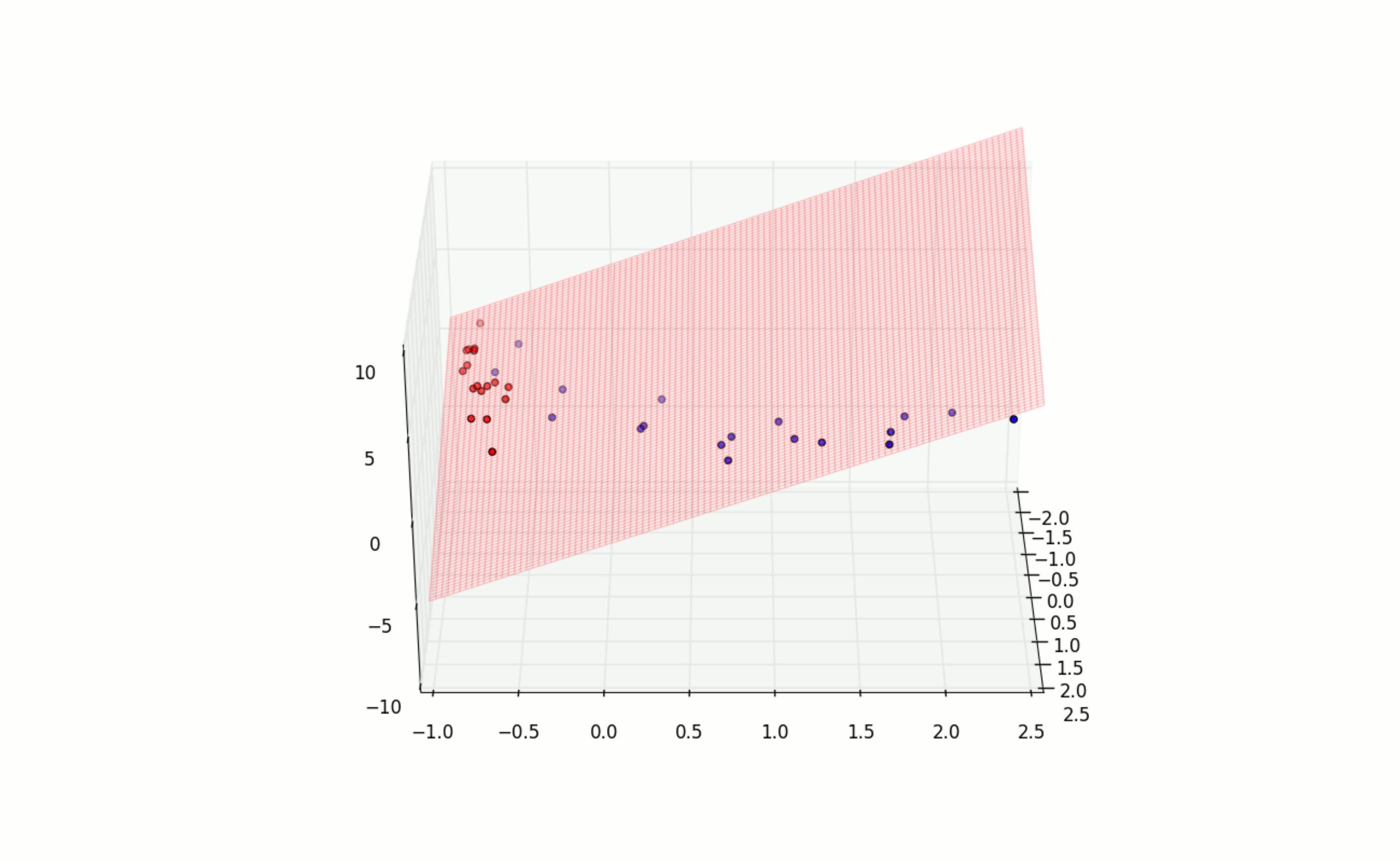
Nonlinearity

Linear classification, e.g. Logistic Regression



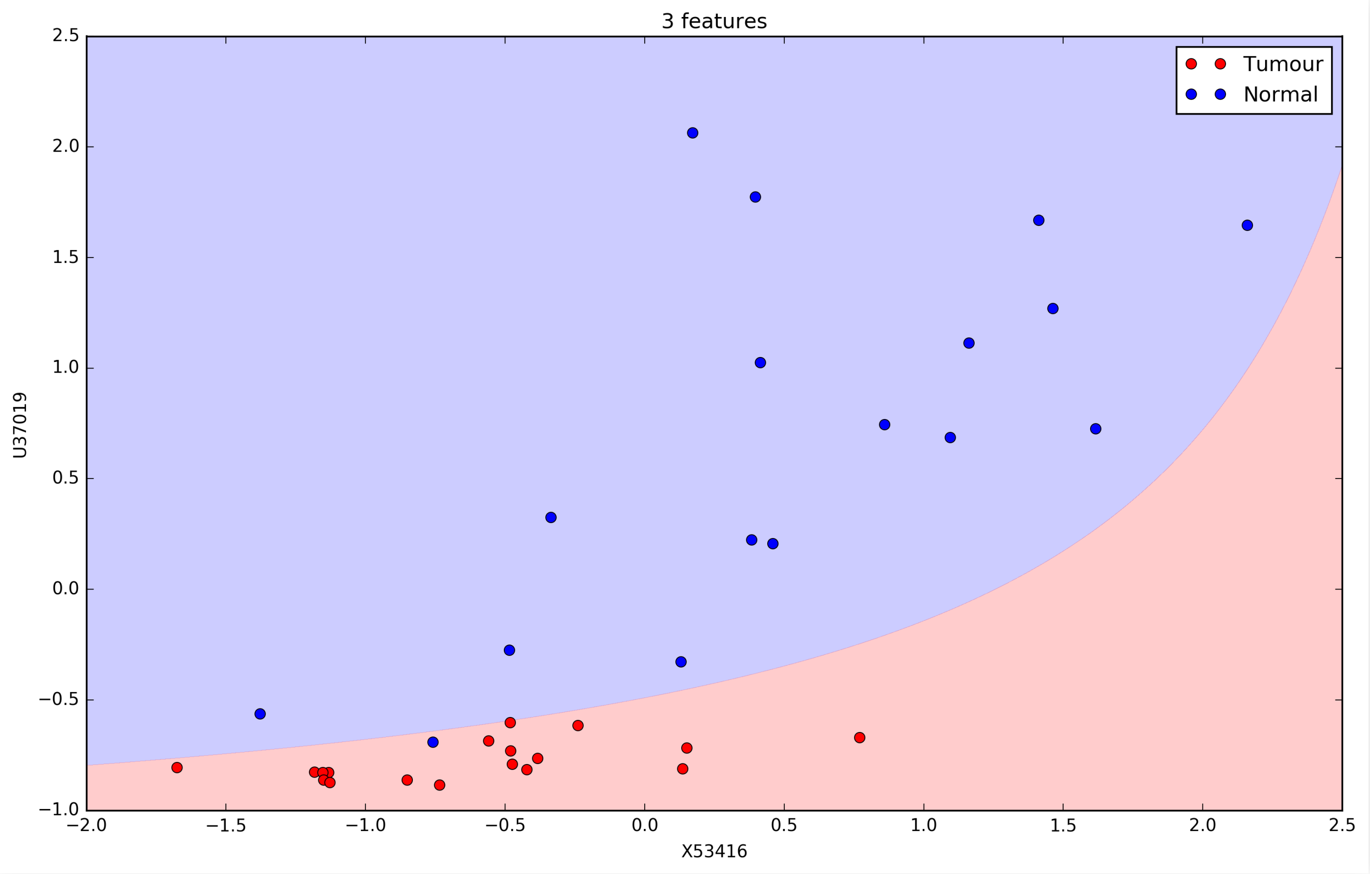
Nonlinear expansion of attributes

- We can expand the attributes non-linearly ($x_1 \times x_2$)



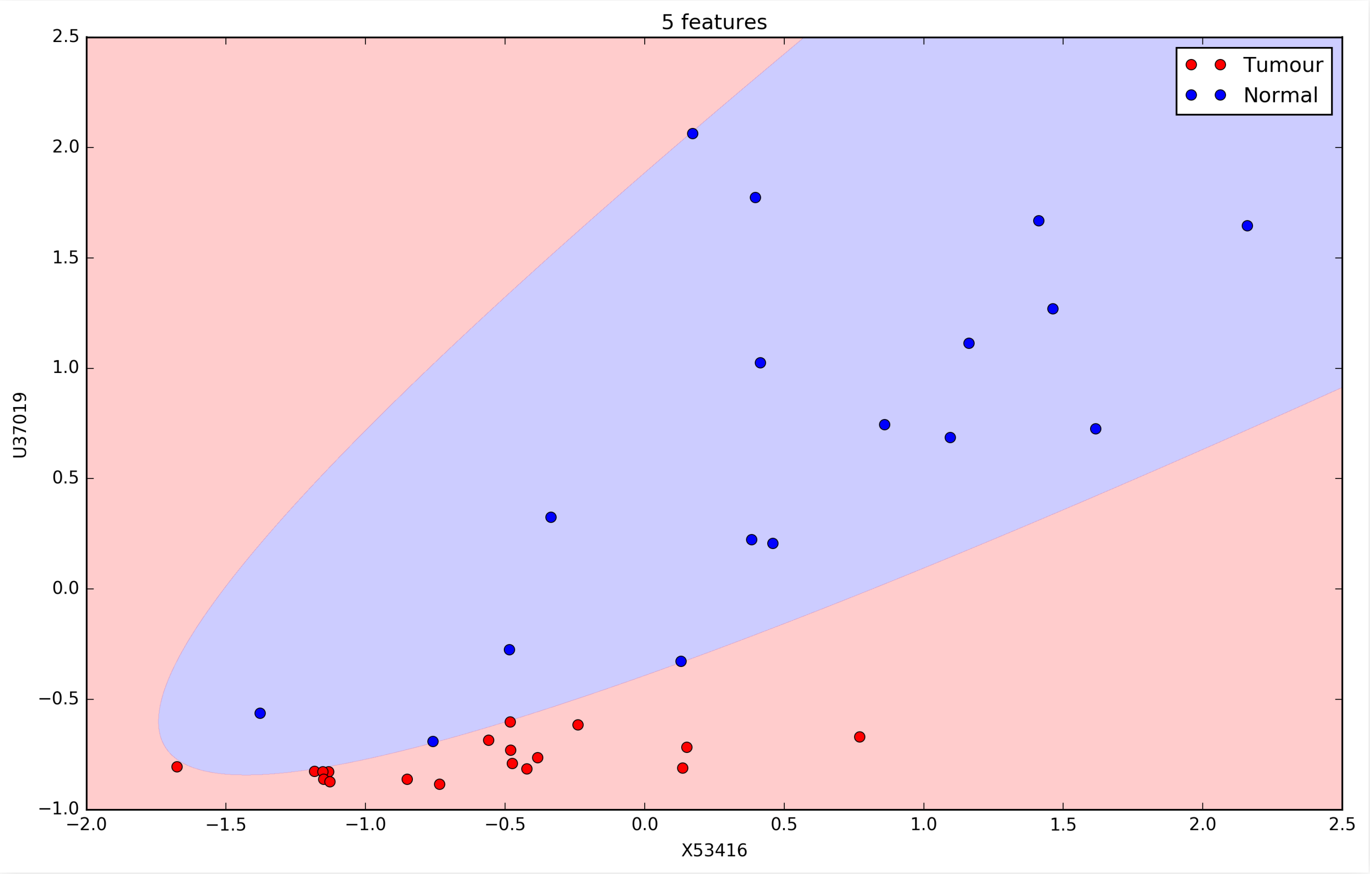
Nonlinear expansion of attributes

- We can expand the attributes non-linearly ($x_1 \times x_2$)



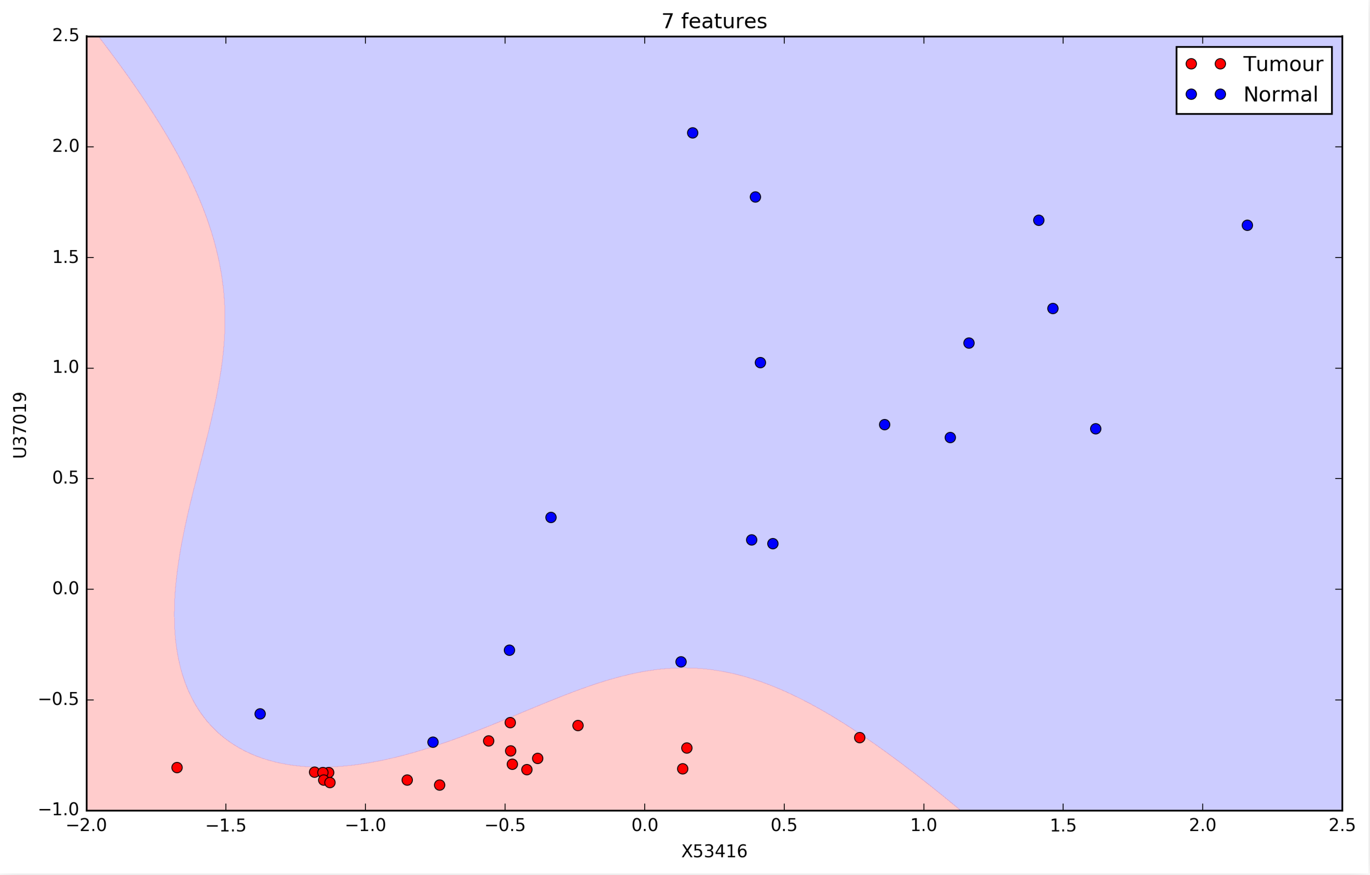
Nonlinear expansion of attributes

- We can expand further $(x_1, x_2, x_1x_2, x_1^2, x_2^2)$



Nonlinear expansion of attributes

- We can expand further $(x_1, x_2, x_1x_2, x_1^2, x_2^2, x_1^3, x_2^3)$



Nonlinearity

Nonlinear expansion of attributes

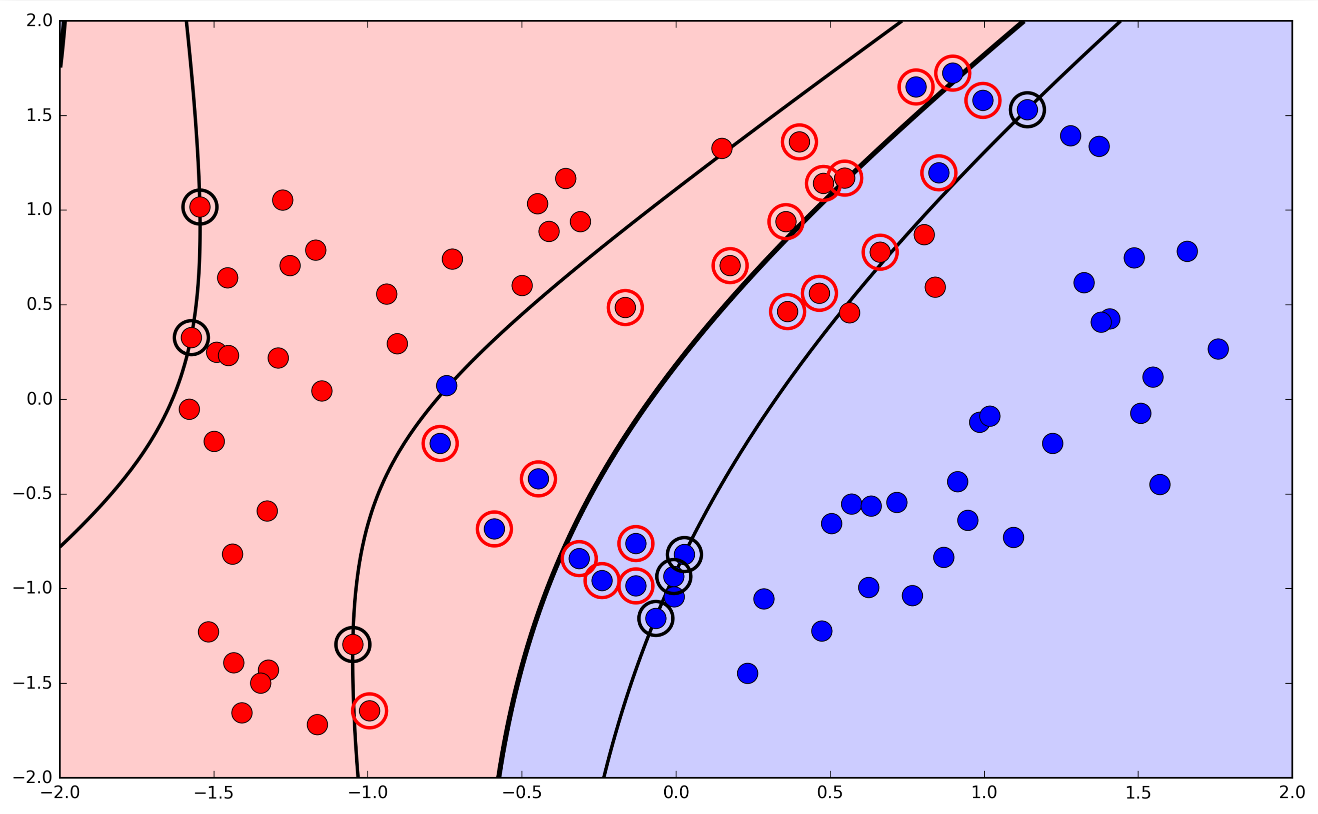
- With logistic regression this is not practical
  - We have to do it by hand
- Support Vector Machines do this automatically

$$\arg\max_{\vec{\alpha}} \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m K(\vec{x}_n, \vec{x}_m)$$

- Where $K(\vec{x}_n, \vec{x}_m)$ is the kernel function for some non-linear expansion ϕ of our original data

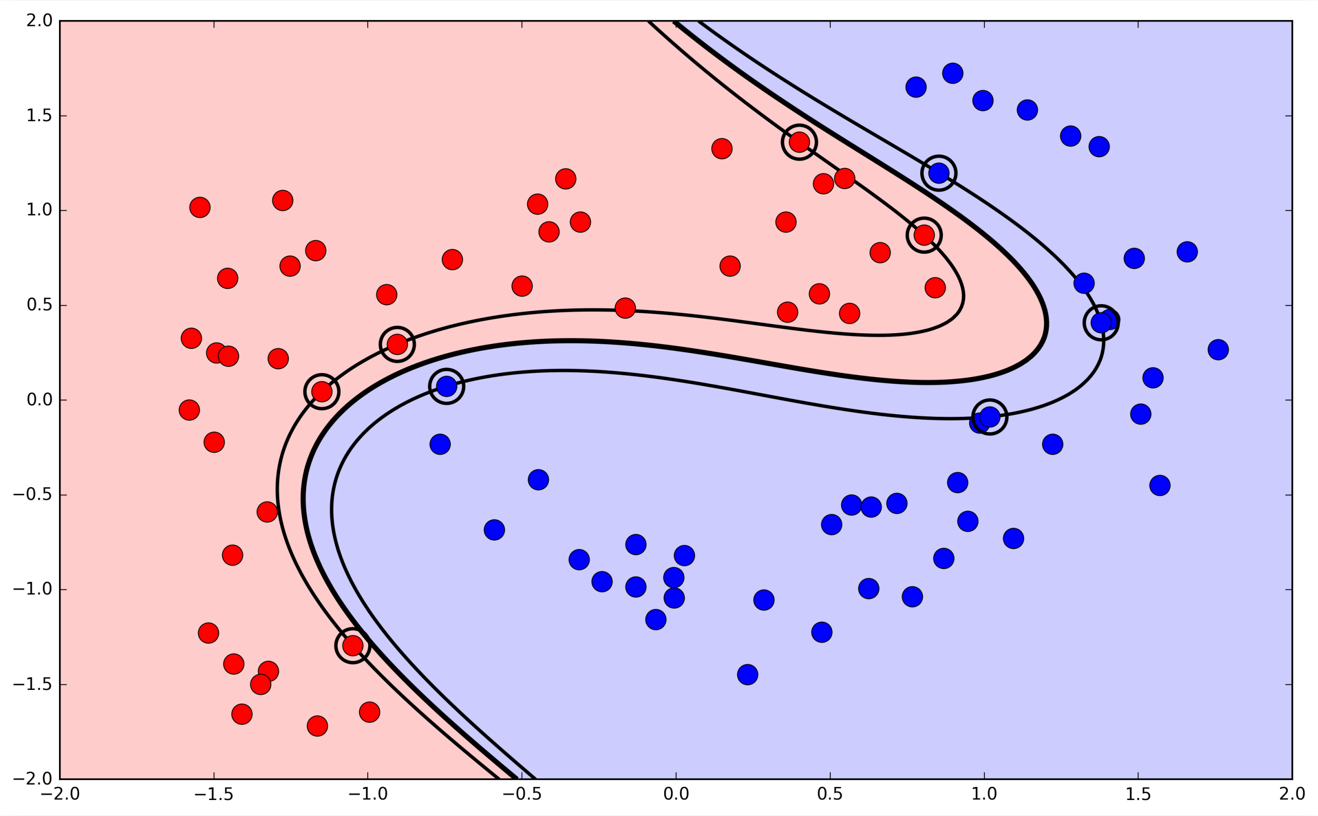
Nonlinear expansion of attributes

- Example, using a polynomial kernel: $K_{\phi(\vec{x}^n)} = (\vec{x}^T\vec{z} + 1)^2$



Nonlinear expansion of attributes

- Example, using a polynomial kernel: $K_{\phi(\vec{x}^n)} = (\vec{x}^T \vec{z} + 1)^3$



No free lunch

No free lunch

No-free-lunch theorems (Wolpert and MacReady, 1997)

"[I]f an algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems."

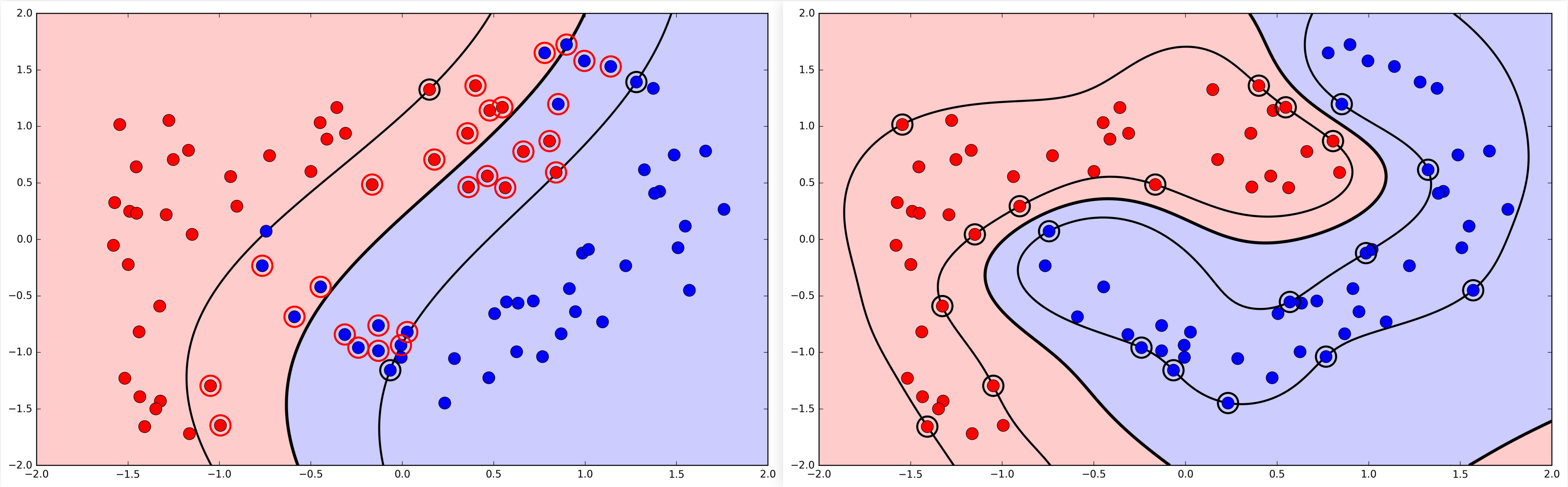
Important for two reasons:

- No single model can be best at all tasks:
  - We need to create different models optimized for different tasks
- Overfitting
  - The hypothesis chosen may be so adjusted to the training data it does not generalize

Overfitting

Nonlinearity is important for capturing patterns in data

- But can lead to loss of generalization



"With great power comes great overfitting"

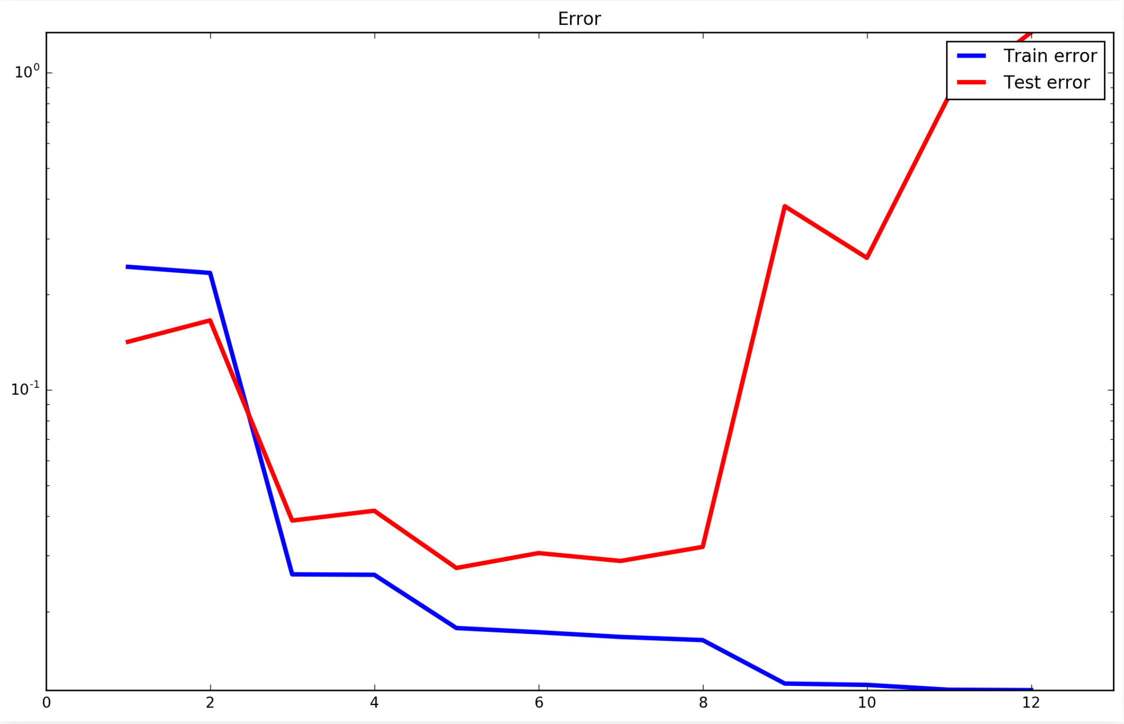
Benjamin Parker (attributed)

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Overfitting

Occurs when model adjusts to noise

- Some details are informative about patterns in the population
- Some are particular to the data sample and do not generalize



Overfitting

Occurs when model adjusts to noise

- Measuring overfitting:
  - Evaluate outside the training set
  - Validation set: used for selecting best model, hyperparameters, ...
  - Test set: used to obtain unbiased estimate of the true error
- Preventing overfitting:
  - Adjust training (regularization)
  - Select adequate model
  - Use more data (allows more powerful models)

What do we want to use machine learning?

- Nonlinear transformations to power our models
- Different models for different problems
  - And a good way to adjust the model to the problem
- The right features
  - Feature selection and extraction is important
- Preventing overfitting
  - Model selection and regularization
- Ability to use large amounts of data
  - Incremental learning

What do we have in "classic" machine learning?

- Many algorithms do nonlinear transformations
- Many different models
  - Great diversity, with different algorithms
- The right features
  - Feature extraction usually done by the user
- Preventing overfitting
  - Method depends on the algorithm
- Ability to use large amounts of data
  - Some do, some don't

Deep learning helps solve these problems

- Nonlinear transformations, stacked
- Many different models
  - but all built from artificial neurons
- The right features
  - can be done automatically determined by the model during training
- Preventing overfitting
  - Many ways to regularize
- Ability to use large amounts of data
  - Yes!

Summary

Summary

- Overview of the course
- AI and Machine learning
- Nonlinear transformations and Overfitting
- The promise of deep learning

Further reading:

- Skansi, Introduction to Deep Learning, Chapter 1
- Goodfellow et al, Deep Learning, Chapters 1 and 5