Training of data driven prediction methods

Morten Nielsen,
Department of Health Technology
DTU
and
Instituto de Investigaciones Biotecnológicas, Universidad Nacional de San Martín, Argentina

What is a data driven prediction method?

- Learn predictive rules from data
- These rules allow you to make predictions for new data
- Examples
  - Weather forecast
  - Stock marked
  - Prediction of Signal peptides
  - Protein function
  - MHC:peptide binding
  - Protein folding/structure

What methods exist?

- Homology / Alignment
- Simple pattern ("word") recognition
- Statistical methods
  - Weight matrices: calculate amino acid probabilities
  - Other examples: Regression, variance analysis, clustering
- Machine learning
  - Like statistical methods, but parameters are estimated by iterative training rather than direct calculation
  - Examples: Neural Networks (NN), Hidden Markov Models (HMM), Support Vector Machines (SVM)
- Combinations

Example. Where is the active site?

CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

Align using sequence profiles

ALN 1K7C.A 1WAB._ RMSD = 5.29522. 14% ID

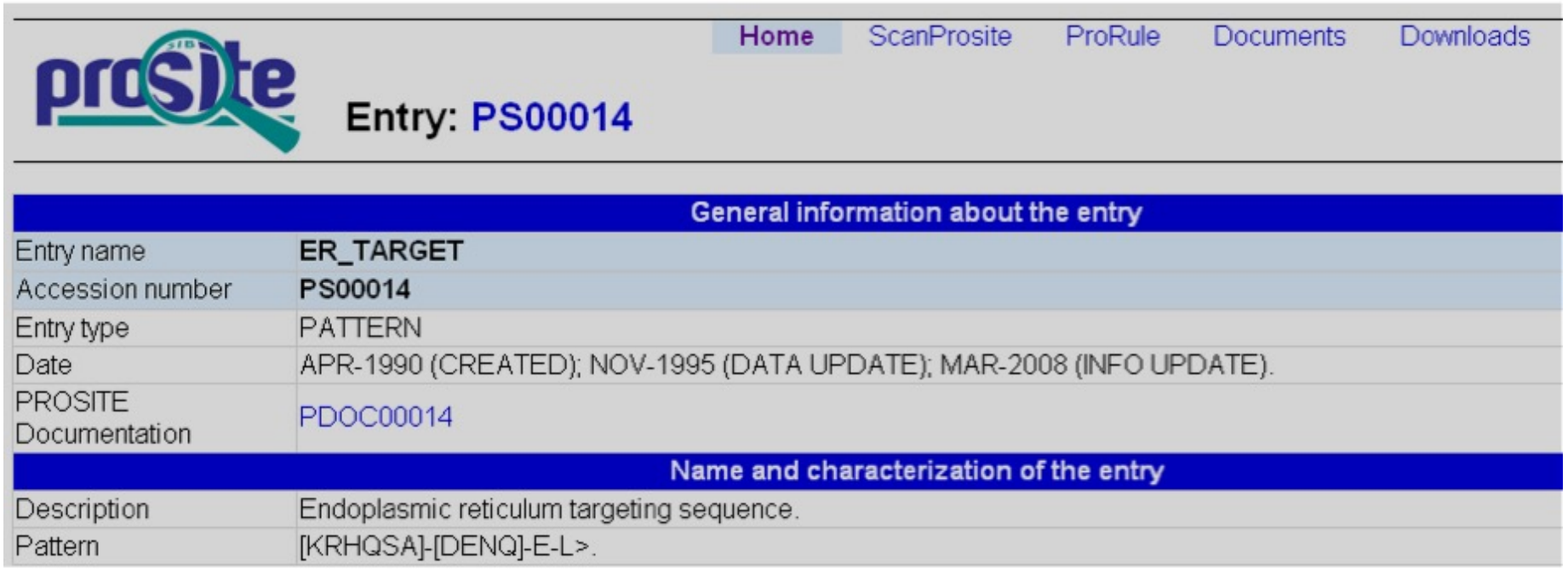
```
1K7C.A TVYLAGDSTMAKNGGGSGTNGWGEYLASYLSATVVNDAVAGRSARSYTREGRFENIADVVTAGDYVIVEFGHNDGGSLSTDN
               S                                 G                                 N
1WAB._ EVVFIGDSLVQLMHQCE---IWRELFS---PLHALNFGIGGDSTQHVLW--RLENGELEHIRPKIVVVWVGTNNHG------

1K7C.A GRTDCSGTGAEVCYSVYDGVNETILTFPAYLENAAKLFTAK--GAKVILSSQTPNNPWETGTFVNSPTRFVEYAEL-AAEVA
1WAB._ ---------------------HTAEQVTGGIKAIVQLVNERQPQARVVVLGLLPRGQ-HPNPLREKNRRVNELVRAALAGHP

1K7C.A GVEYVDHWSYVDSIYETLGNATVNSYFPIDHTHTSPAGAEVVAEAFLKAVVCTGTSL
                                          H
1WAB._ RAHFLDADPG---FVHSDG--TISHHDMYDYLHLSRLGYTPVCRALHSLLLRL---L
```

Does it work?

- Does it work
  - Yes, otherwise the field of bioinformatics would not exist
  - but you have to be very careful to avoid overfitting
- Critical issues
  - How do you construct a good training data set?
  - How do you evaluate how good you are?

Regular expression ("words")

- Example:
  - PROSITE entry PS00014, ER_TARGET:
  - Endoplasmic reticulum targeting sequence ("KDEL-signal").
  - Pattern: [KRHQSA]-[DENQ]-E-L>



Home ScanProsite ProRule Documents Downloads

prosite

Entry: PS00014

General information about the entry	
Entry name	**ER_TARGET**
Accession number	**PS00014**
Entry type	PATTERN
Date	APR-1990 (CREATED); NOV-1995 (DATA UPDATE); MAR-2008 (INFO UPDATE).
PROSITE Documentation	PDOC00014
Name and characterization of the entry	
Description	Endoplasmic reticulum targeting sequence.
Pattern	[KRHQSA]-[DENQ]-E-L>.

The answer is just yes/no

Simple motifs

Yes/No rules

10 MHC restricted peptides

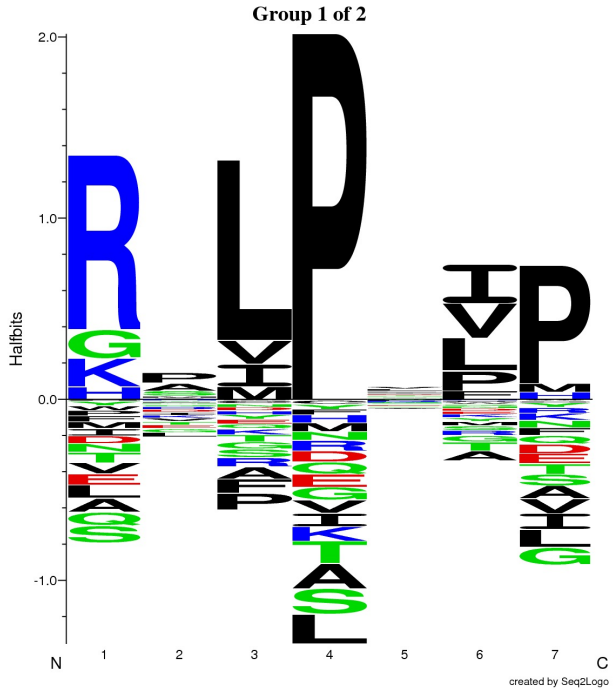
ALAKAAAAM
ALAKAAAAN
ALAKAAAAR
ALAKAAAAT
ALAKAAAAV
GMNERPILT
GILGFVFTM
TLNAWVKVV
KLNEPVLLL
AVVPFIVSV

$$[AGTK]_1[LMIV]_2[ANLV]_3 \ldots [MNRTVL]_9$$

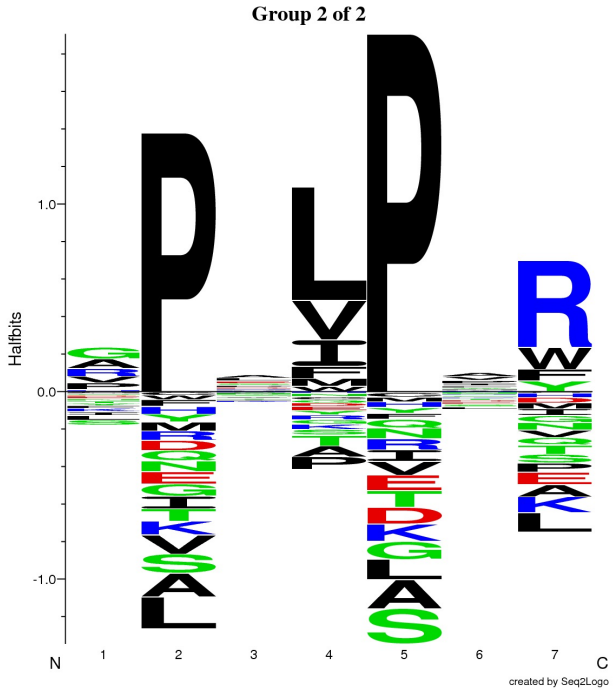
- Only 11 of 212 peptides identified!
- Need more flexible rules
  - If not fit P1 but fit P2 then ok
- Not all positions are equally important
  - We know that P2 and P9 determines binding more than other positions
- Cannot discriminate between good and very good binders

SH3 domain binding

Class I



Class II



class I
+xϕPxϕP

class II
ϕPxϕPx+

Weight matrices

- A weight matrix is given as

  $W_{ij} = \log(p_{ij}/q_j)$

  - where i is a position in the motif, and j an amino acid. q_j is the background frequency for amino acid j.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	0.6	0.4	-3.5	-2.4	-0.4	-1.9	-2.7	0.3	-1.1	1.0	0.3	0.0	1.4	1.2	-2.7	1.4	-1.2	-2.0	1.1	0.7
2	-1.6	-6.6	-6.5	-5.4	-2.5	-4.0	-4.7	-3.7	-6.3	1.0	5.1	-3.7	3.1	-4.2	-4.3	-4.2	-0.2	-5.9	-3.8	0.4
3	0.2	-1.3	0.1	1.5	0.0	-1.8	-3.3	0.4	0.5	-1.0	0.3	-2.5	1.2	1.0	-0.1	-0.3	-0.5	3.4	1.6	0.0
4	-0.1	-0.1	-2.0	2.0	-1.6	0.5	0.8	2.0	-3.3	0.1	-1.7	-1.0	-2.2	-1.6	1.7	-0.6	-0.2	1.3	-6.8	-0.7
5	-1.6	-0.1	0.1	-2.2	-1.2	0.4	-0.5	1.9	1.2	-2.2	-0.5	-1.3	-2.2	1.7	1.2	-2.5	-0.1	1.7	1.5	1.0
6	-0.7	-1.4	-1.0	-2.3	1.1	-1.3	-1.4	-0.2	-1.0	1.8	0.8	-1.9	0.2	1.0	-0.4	-0.6	0.4	-0.5	-0.0	2.1
7	1.1	-3.8	-0.2	-1.3	1.3	-0.3	-1.3	-1.4	2.1	0.6	0.7	-5.0	1.1	0.9	1.3	-0.5	-0.9	2.9	-0.4	0.5
8	-2.2	1.0	-0.8	-2.9	-1.4	0.4	0.1	-0.4	0.2	-0.0	1.1	-0.5	-0.5	0.7	-0.3	0.8	0.8	-0.7	1.3	-1.1
9	-0.2	-3.5	-6.1	-4.5	0.7	-0.8	-2.5	-4.0	-2.6	0.9	2.8	-3.0	-1.8	-1.4	-6.2	-1.9	-1.6	-4.9	-1.6	4.5

- W is a L x 20 matrix, L is motif length

Fitting a model


```
AAAFVNQHL 0.212499
AATEAEKQL 0.000000
FLNAWIPPV 0.989885
VLRDDLLEA 0.396155
TLDGQQFYW 0.287704
KQINPPTVY 0.084687
KSHNVSLIW 0.084687
KLFFAKCLV 0.695578
NVDNLIMIV 0.525266
GEMWAQDAA 0.123103
EVGTNFGTI 0.084324
EMADYIFFV 0.850274
RIGTAATKR 0.000000
TMGPHPAGV 1.000000
SLLADVQQL 0.691397
FVLATGDFV 0.576216
TLDSQVMSL 0.530360
CLPACVYGL 0.601055
FYLFTFTIY 0.084687
LIRILQRAL 0.239440
```

Fitting a model (quantitative data)

CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

```
AAAFVNQHL 0.212499
AATEAEKQL 0.000000
FLNAWIPPV 0.989885
VLRDDLLEA 0.396155
TLDGQQFYW 0.287704
KQINPPTVY 0.084687
KSHNVSLIW 0.084687
KLFFAKCLV 0.695578
NVDNLIMIV 0.525266
GEMWAQDAA 0.123103
EVGTNFGTI 0.084324
EMADYIFFV 0.850274
RIGTAATKR 0.000000
TMGPHPAGV 1.000000
SLLADVQQL 0.691397
FVLATGDFV 0.576216
TLDSQVMSL 0.530360
CLPACVYGL 0.601055
FYLFTFTIY 0.084687
LIRILQRAL 0.239440
```

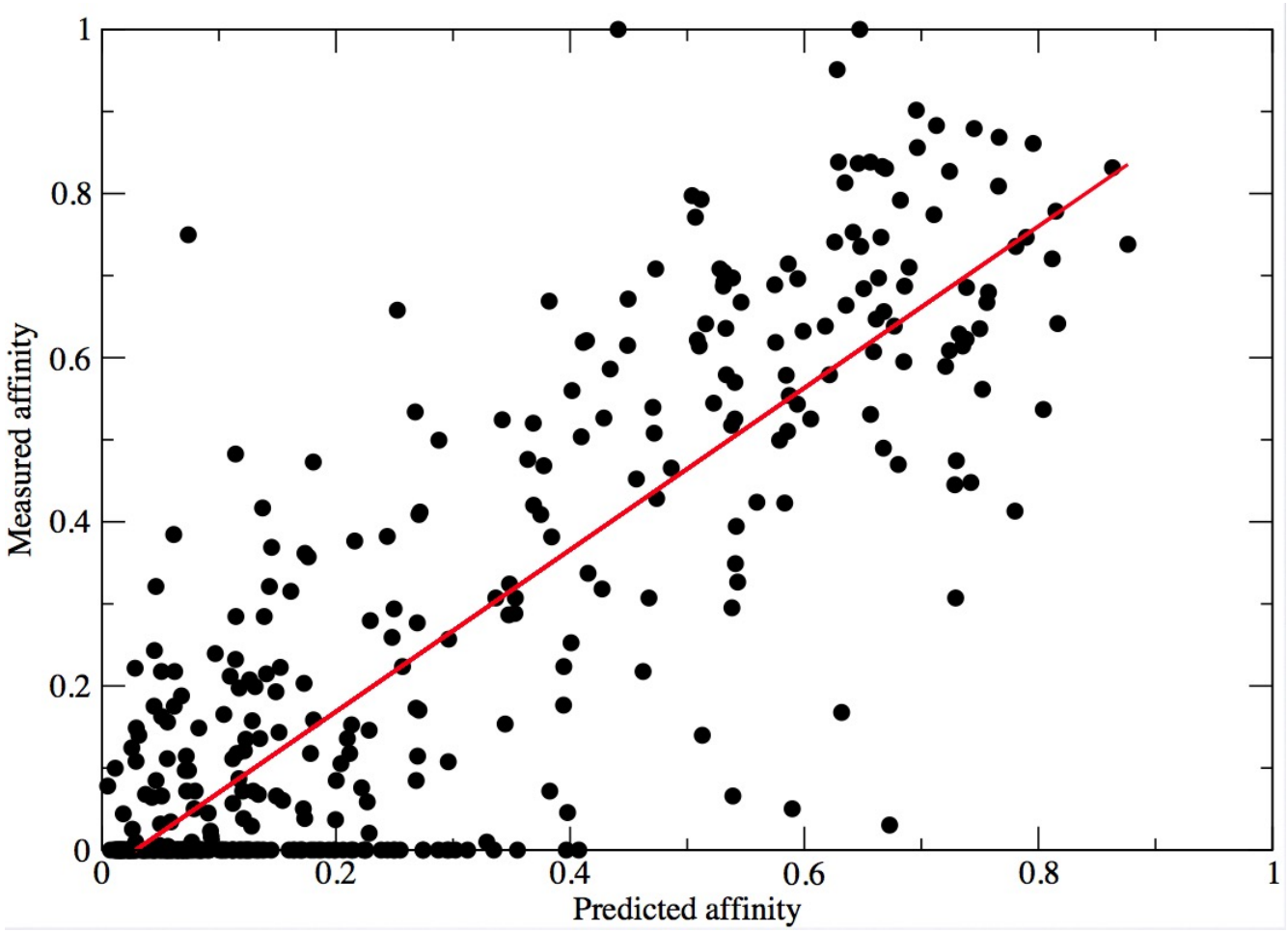
- Fit model to minimize error between prediction and target values =>
- Matrix 9*20

1	-0.024	-0.031	-0.071	-0.206	-0.054	-0.055	-0.165	-0.029	-0.058	-0.011	0.002	0.009	0.033	0.147	-0.166	-0.024	-0.044	-0.027	0.117	-0.016
2	0.004	-0.045	-0.105	-0.083	-0.123	0.063	-0.092	-0.024	-0.110	0.124	0.261	-0.024	0.275	0.005	-0.075	0.010	0.031	-0.031	-0.041	0.058
3	0.035	-0.070	0.006	0.046	-0.052	-0.019	-0.079	-0.025	-0.023	0.081	0.085	-0.094	0.145	0.070	-0.017	0.033	-0.013	0.114	0.078	0.023
4	0.021	-0.063	-0.020	0.066	0.013	-0.011	0.042	0.010	-0.010	-0.018	-0.032	-0.012	-0.016	0.004	-0.000	0.020	-0.002	0.025	0.025	-0.027
5	0.004	0.011	-0.019	0.012	0.034	-0.017	0.010	0.004	-0.007	0.049	0.023	-0.012	0.009	0.065	-0.044	-0.008	-0.007	0.122	0.112	0.038
6	-0.005	-0.075	0.015	-0.017	0.004	0.016	-0.023	-0.030	-0.019	0.112	0.085	-0.070	0.069	0.065	0.017	0.003	0.054	0.002	0.024	0.076
7	0.014	-0.076	-0.036	-0.022	0.008	-0.027	-0.024	-0.071	0.045	0.069	0.054	-0.102	0.047	0.147	0.039	-0.006	0.018	0.065	0.100	0.046
8	-0.000	-0.013	-0.000	-0.066	0.010	-0.010	0.018	0.013	-0.016	-0.010	0.042	-0.007	0.001	0.085	0.041	0.032	0.012	0.062	0.093	-0.019
9	0.132	-0.065	-0.122	-0.031	-0.058	-0.031	0.021	-0.025	-0.049	0.177	0.152	-0.076	0.068	-0.016	0.012	-0.022	0.036	-0.040	-0.071	0.254

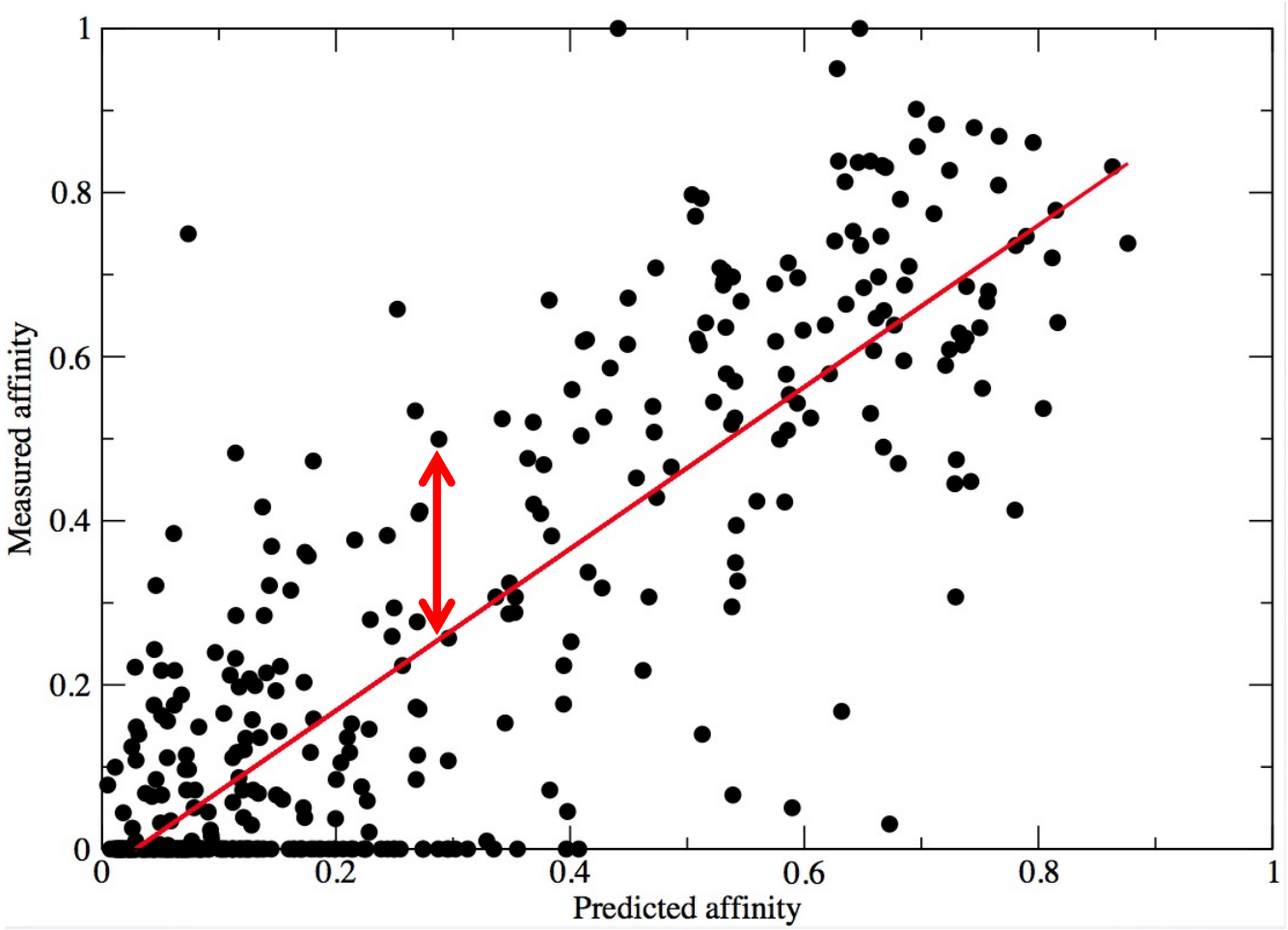
Performance measures

CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

Meas	Pred
0.4050	0.8344
0.9373	1.0000
0.8161	0.6388
0.6752	0.9841
0.0253	0.0000
0.3196	0.5388
0.6764	0.6247
0.1872	0.1921
0.4220	0.6546
0.6545	0.6546
0.7917	0.1342
0.4405	0.3551
0.1548	0.0000
0.2740	0.1993
0.4399	0.6461
0.1725	0.3916
0.0539	0.0000
0.3795	0.5623
0.2242	0.1968
0.3108	0.2114
0.2260	0.0336
0.2780	0.5647
0.0198	0.1224
0.5890	0.5538
0.5120	0.4349
0.7266	1.0000
0.1136	0.0000
0.0456	0.2128
0.0069	0.4100
0.4502	0.3848

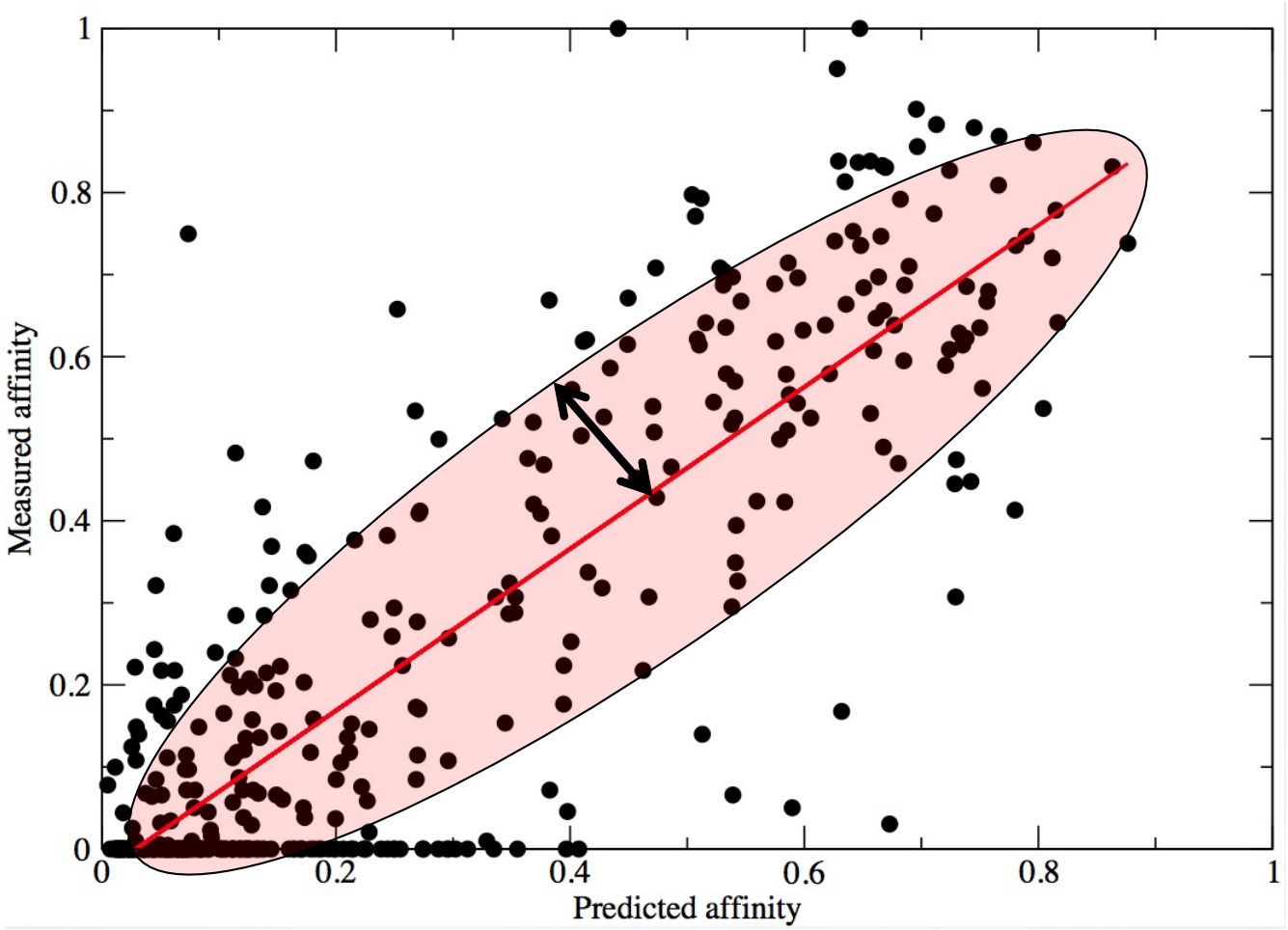


RMSE (root mean square error)



$$RMSE = \sqrt{\frac{1}{N}\sum_i (p_i - t_i)^2}$$

Pearsons' correlation coefficient



$$PCC = \frac{\sum_i (a_i - \bar{a}) \cdot (p_i - \bar{p})}{\sqrt{\sum_i (a_i - \bar{a})^2} \cdot \sqrt{\sum_i (p_i - \bar{p})^2}}$$

Performance measures (quantitative data)

- Accuracy of prediction method

Prediction score	Binder/non-binder
0,94	1
0,88	1
0,12	0
0,43	1
0,58	0
0,66	1
0,77	0
0,11	0
0,77	0
0,06	1
0,71	1
0,43	0
0,05	0
0,79	1
0,55	0
0,32	1
0,22	1
0,33	0
0,35	0
0,57	1
0,21	0
0,91	1
0,16	0
0,45	1
0,88	0
0,66	1
0,98	1

Prediction score	Binder/non-binder
0,98	1
0,94	1
0,91	1
0,88	1
0,88	0
0,79	1
0,77	0
0,77	0
0,71	1
0,66	1
0,66	1
0,58	0
0,57	1
0,55	0
0,45	1
0,43	1
0,43	0
0,35	0
0,33	0
0,32	1
0,22	1
0,21	0
0,16	0
0,12	0
0,11	0
0,06	1
0,05	0

Matthews correlation - Threshold of 0.5

Prediction score	Binder/non-binder
0,98	1
0,94	1
0,91	1
0,88	1
0,88	0
0,79	1
0,77	0
0,77	0
0,71	1
0,66	1
0,66	1
0,58	0
0,57	1
0,55	0
0,45	1
0,43	1
0,43	0
0,35	0
0,33	0
0,32	1
0,22	1
0,21	0
0,16	0
0,12	0
0,11	0
0,06	1
0,05	0

False negative	True positive
True negative	False positive

$$Sensitivity = TP / AP$$

$$Specificity = TN / AN$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}}$$

Matthews correlation - Threshold of 0.5

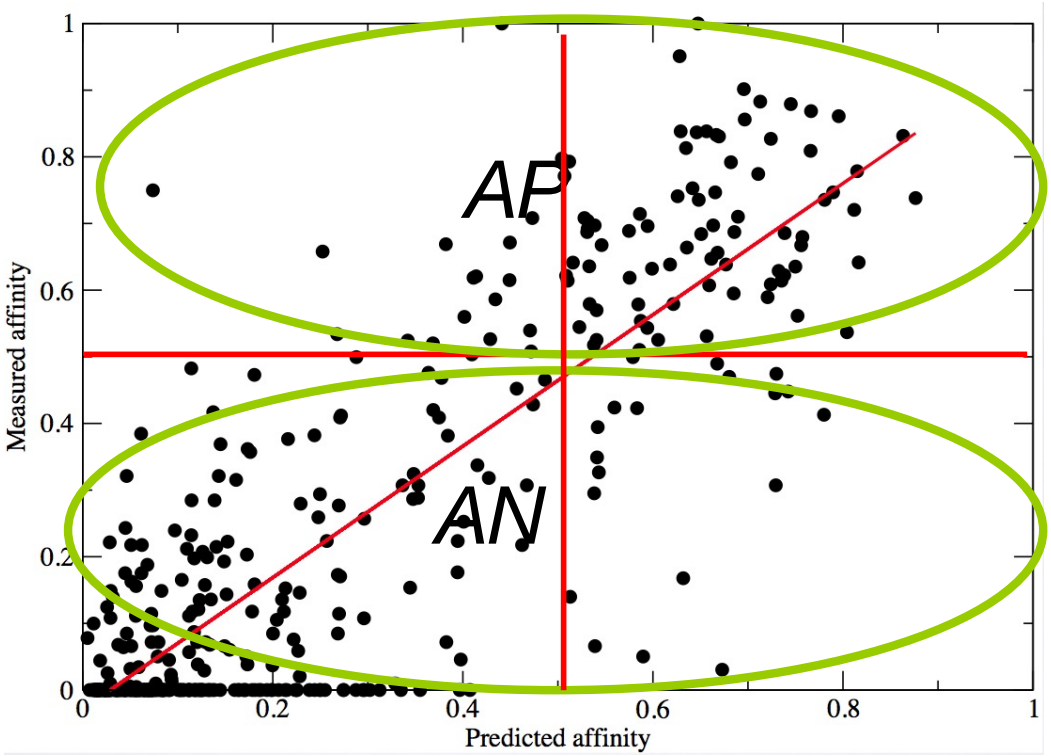
Prediction score	Binder/non-binder
0,98	1
0,94	1
0,91	1
0,88	1
0,88	0
0,79	1
0,77	0
0,77	0
0,71	1
0,66	1
0,66	1
0,58	0
0,57	1
0,55	0
0,45	1
0,43	1
0,43	0
0,35	0
0,33	0
0,32	1
0,22	1
0,21	0
0,16	0
0,12	0
0,11	0
0,06	1
0,05	0

5	9
8	5

$$CC = \frac{9 \cdot 8 - 5 \cdot 5}{\sqrt{(9+5)(8+5)(9+5)(8+5)}} = 0.258$$

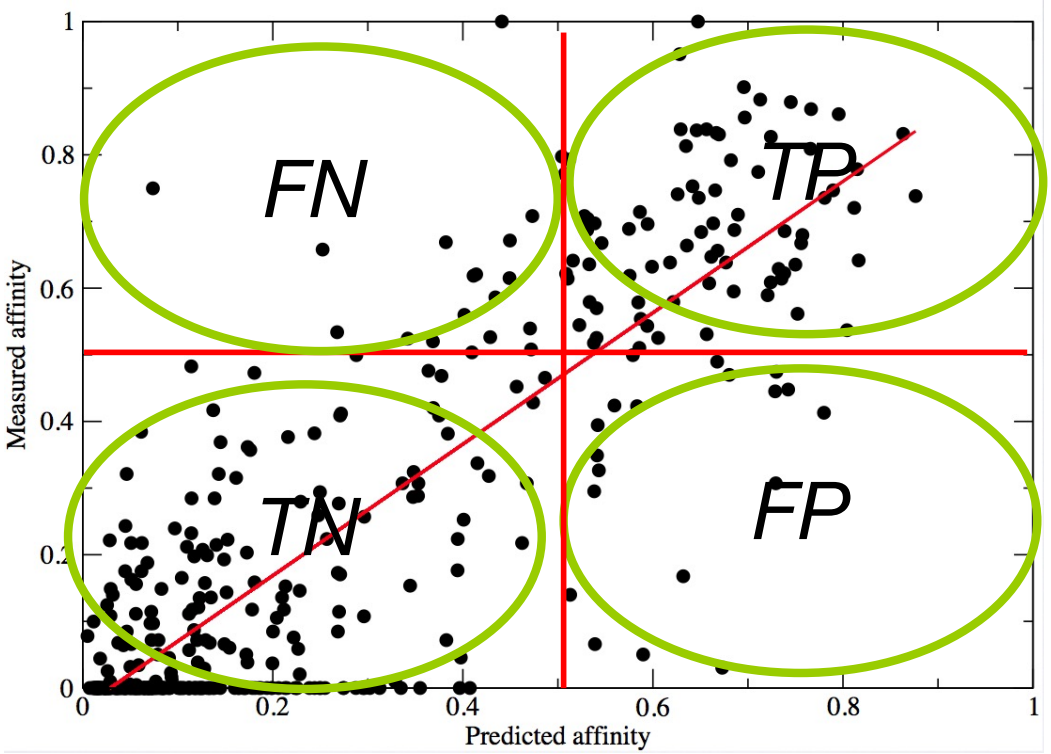
Evaluation of prediction accuracy

$$Sensitivity = TP / AP \qquad Specificity = TN / AN$$



Evaluation of prediction accuracy

$$Sensitivity = TP / AP \qquad Specificity = TN / AN$$



Performance measure – Roc curve

CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

Prediction score	Binder/non-binder
0,98	1
0,94	1
0,91	1
0,88	1
0,88	0
0,79	1
0,77	0
0,77	0
0,71	1
0,66	1
0,66	1
0,58	0
0,57	1
0,55	0
0,45	1
0,43	1
0,43	0
0,35	0
0,33	0
0,32	1
0,22	1
0,21	0
0,16	0
0,12	0
0,11	0
0,06	1
0,05	0

False negative	True positive
True negative	False positive

Threshold	TP	FN	TP/(TP+FN)	FP	TN	FP/(FP+TN)
>0,8	*4*	*10*	*0.29*	*1*	*12*	*0.08*
>0,6						
>0,4						
>0,2						
>0						

Performance measure – Roc curve

CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

Prediction score	Binder/non-binder
0,98	1
0,94	1
0,91	1
0,88	1
0,88	0
0,79	1
0,77	0
0,77	0
0,71	1
0,66	1
0,66	1
0,58	0
0,57	1
0,55	0
0,45	1
0,43	1
0,43	0
0,35	0
0,33	0
0,32	1
0,22	1
0,21	0
0,16	0
0,12	0
0,11	0
0,06	1
0,05	0

False negative	True positive
True negative	False positive

Threshold	TP	FN	TP/(TP+FN)	FP	TN	FP/(FP+TN)
>0,8	*4*	*10*	*0.29*	*1*	*12*	*0.08*
>0,6						
>0,4						
>0,2						
>0						

Performance measure – Roc curve

CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

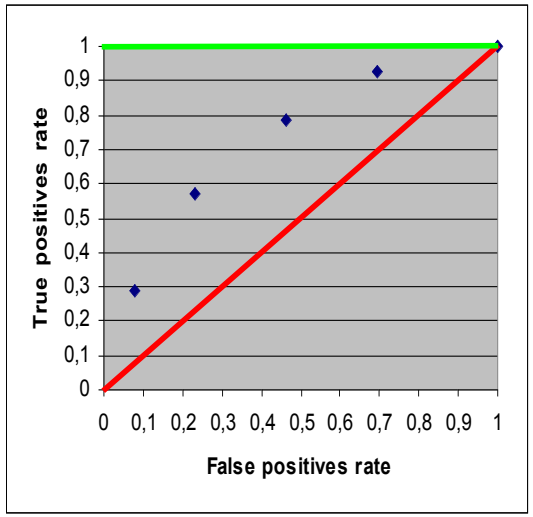
Prediction score	Binder/non-binder
0,98	1
0,94	1
0,91	1
0,88	1
0,88	0
0,79	1
0,77	0
0,77	0
0,71	1
0,66	1
0,66	1
0,58	0
0,57	1
0,55	0
0,45	1
0,43	1
0,43	0
0,35	0
0,33	0
0,32	1
0,22	1
0,21	0
0,16	0
0,12	0
0,11	0
0,06	1
0,05	0

False negative	True positive
True negative	False positive

Threshold	TP	FN	TP/(TP+FN)	FP	TN	FP/(FP+TN)
>0,8	4	10	0.29	1	12	0.08
>0,6	8	6	0.57	3	10	0.23
>0,4						
>0,2						
>0						

ROC curves

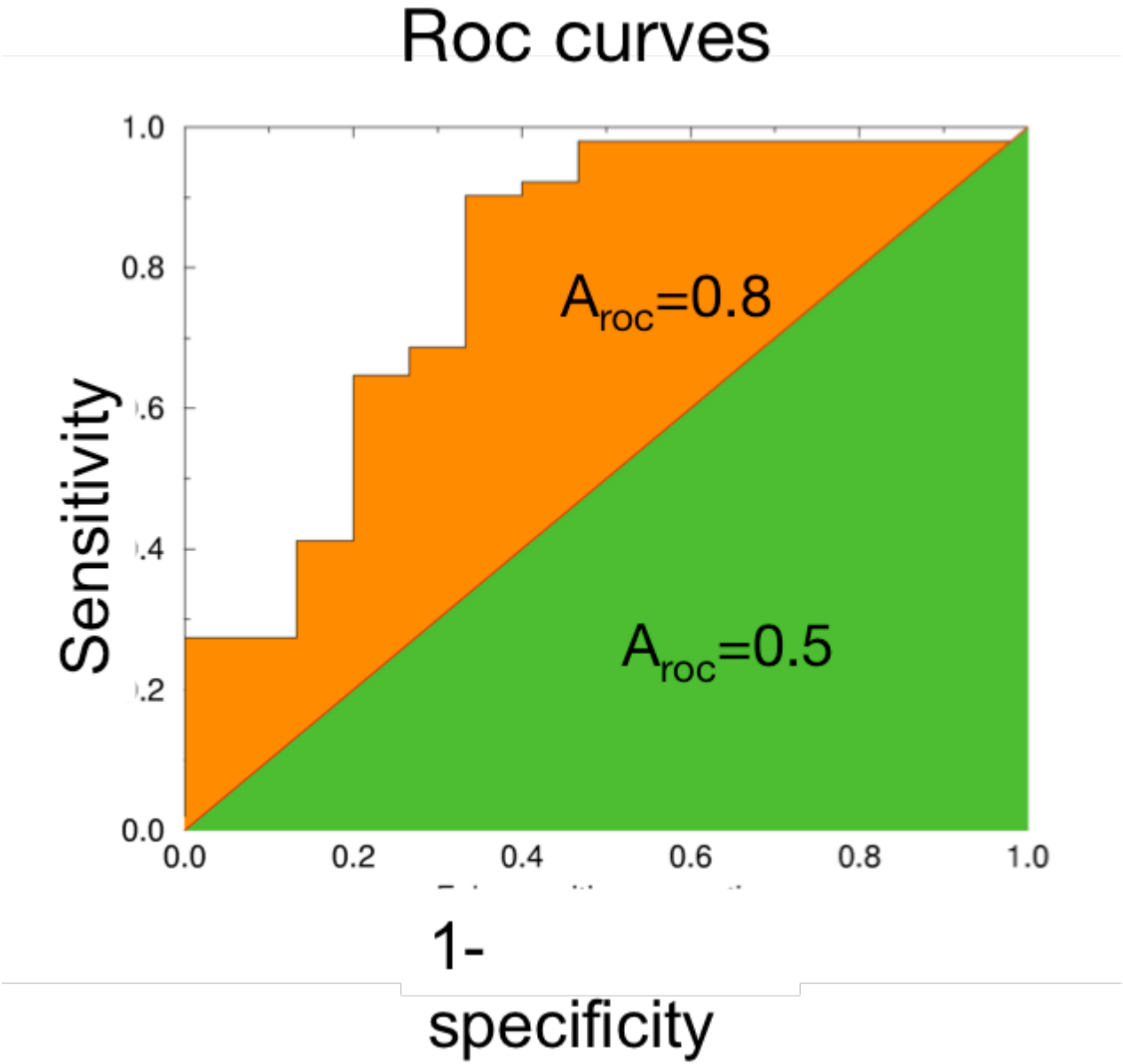
Threshold	TP	FN	TP/(TP+FN)	FP	TN	FP/(FP+TN)
>0.8	4	10	0.29	1	12	0.08
>0.6	8	6	0.57	3	10	0.23
>0.4	11	3	0.79	6	7	0.46
>0.2	13	1	0.93	9	4	0.69
>0	14	0	1	13	0	1



AUC = 0.5

AUC = 1.0

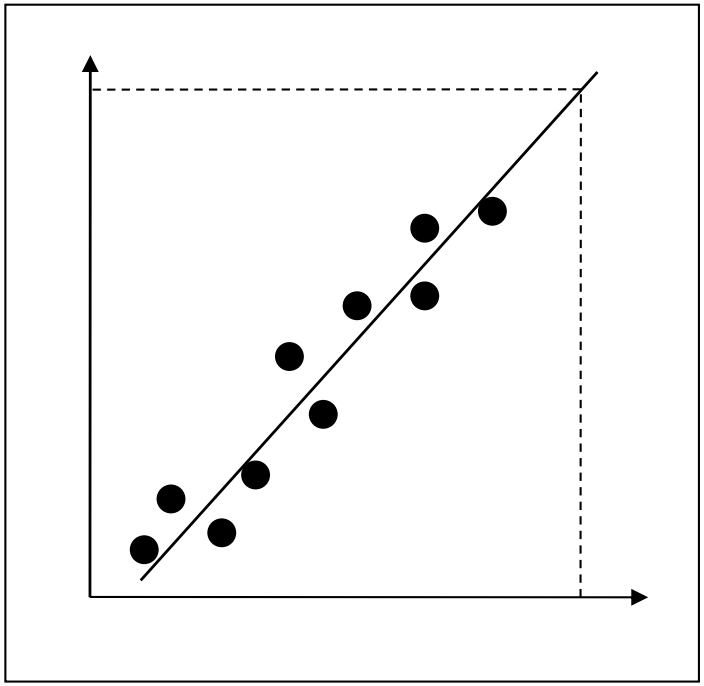
AUC (area under the ROC curve)



Performance measures. Summary

- MCC and PCC
  - Random = 0.0
  - Perfect = 1.0 (or -1)
- RMSE
  - Perfect = 0.0
  - Random >> 0
- ROC (AUC)
  - Random = 0.5
  - Perfect = 1 (or 0)

How to training a method. A simple statistical method: Linear regression



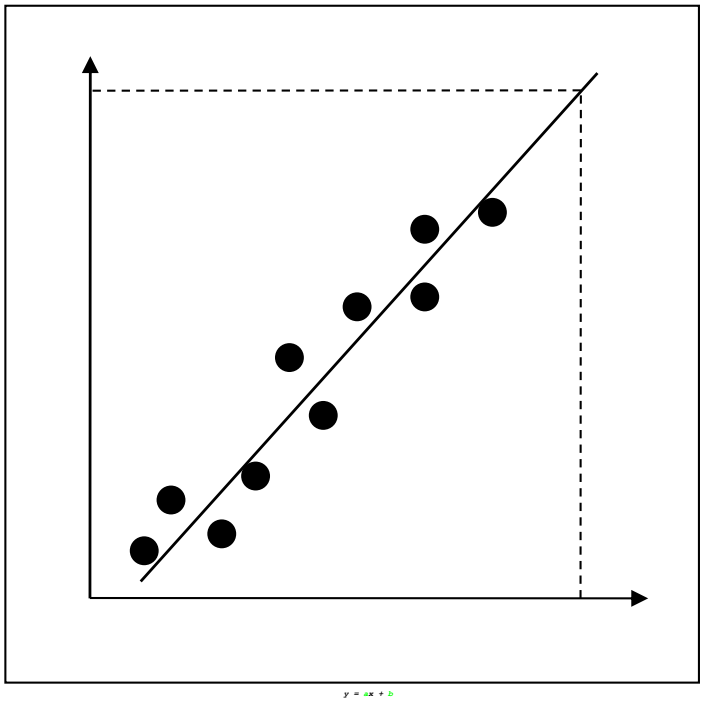
Observations (training data): *a set of x values (input) and y values (output).*

Model: y = ax + b *(2 parameters, which are estimated from the training data)*

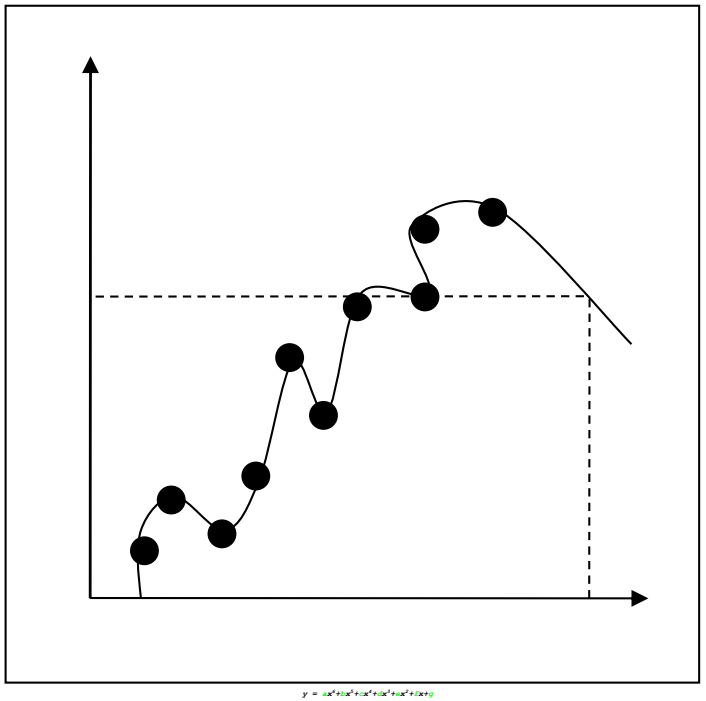
Prediction: *Use the model to calculate a y value for a new x value*

Note: *the model does not fit the observations exactly. Can we do better than this?*

Overfitting



2 parameter model
Good description, poor fit



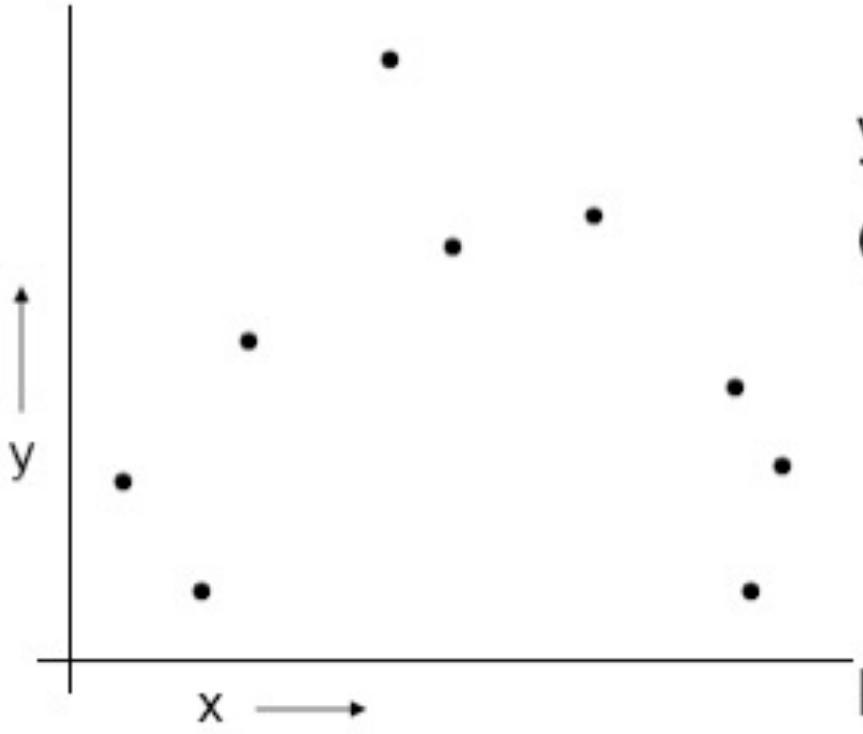
7 parameter model
Poor description, good fit

Note: *It is not interesting that a model can fit its observations (training data) exactly.*

To function as a prediction method, a model must be able to generalize, i.e. produce sensible output on new data.

How to estimate parameters for prediction?

A Regression Problem

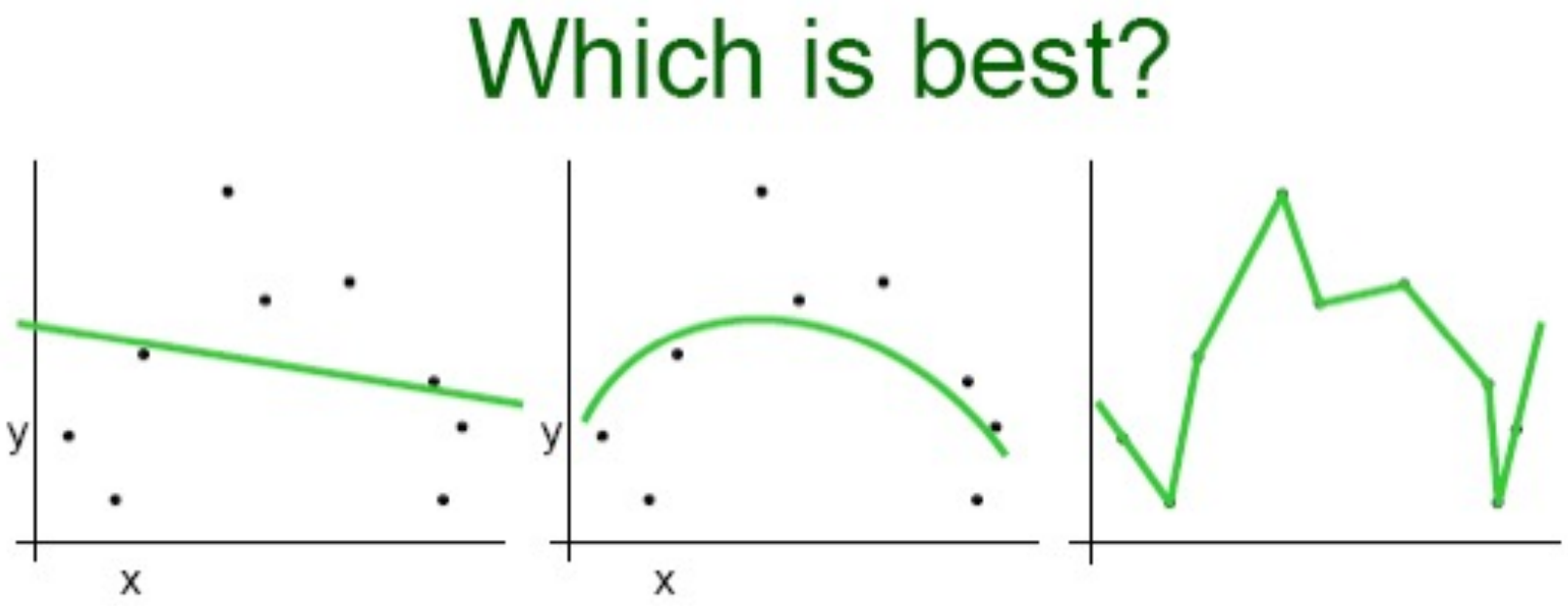


y = f(x) + noise

Can we learn f from this data?

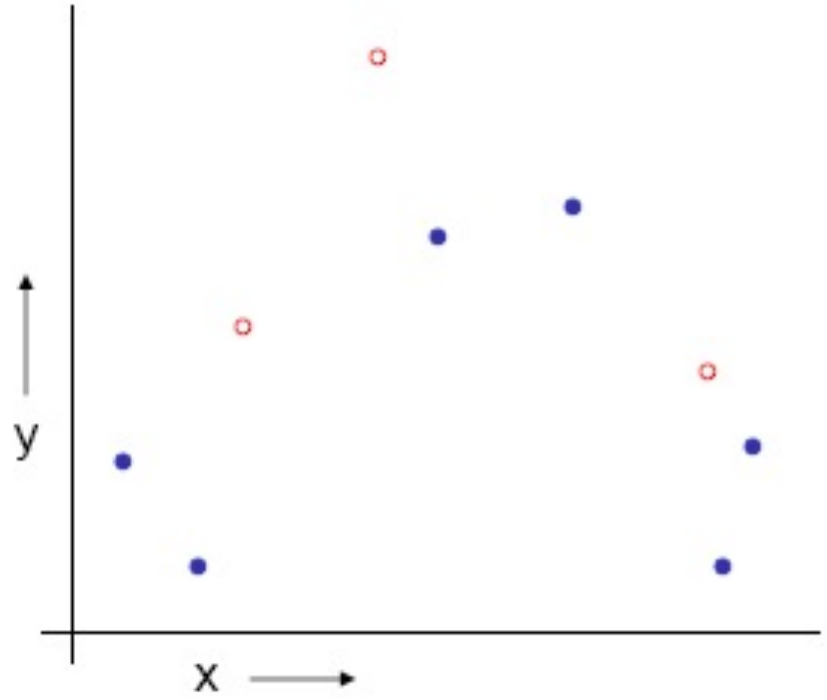
Let's consider three methods…

Model selection



Linear Regression *Quadratic Regression* *Join-the-dots*

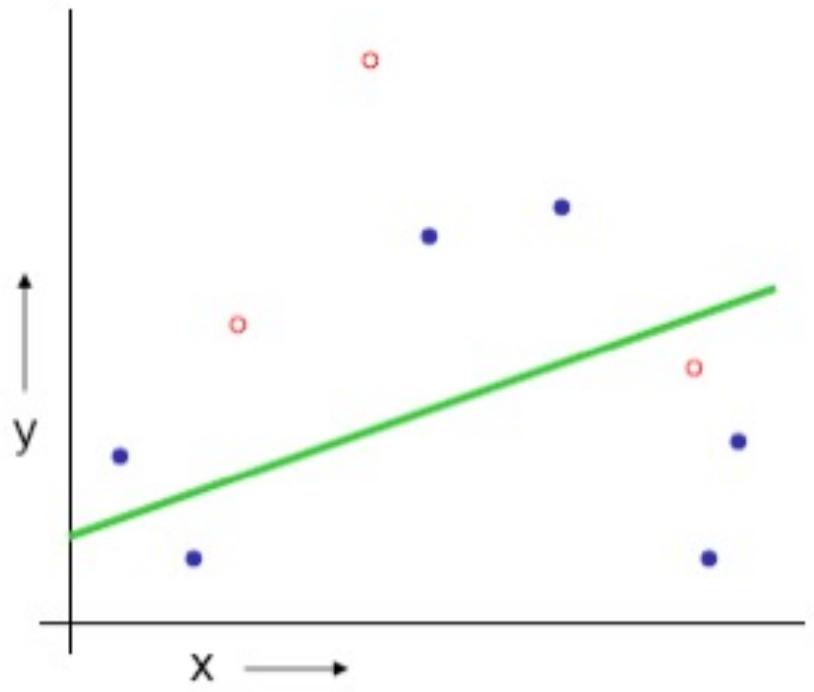
The test set method



1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

The test set method



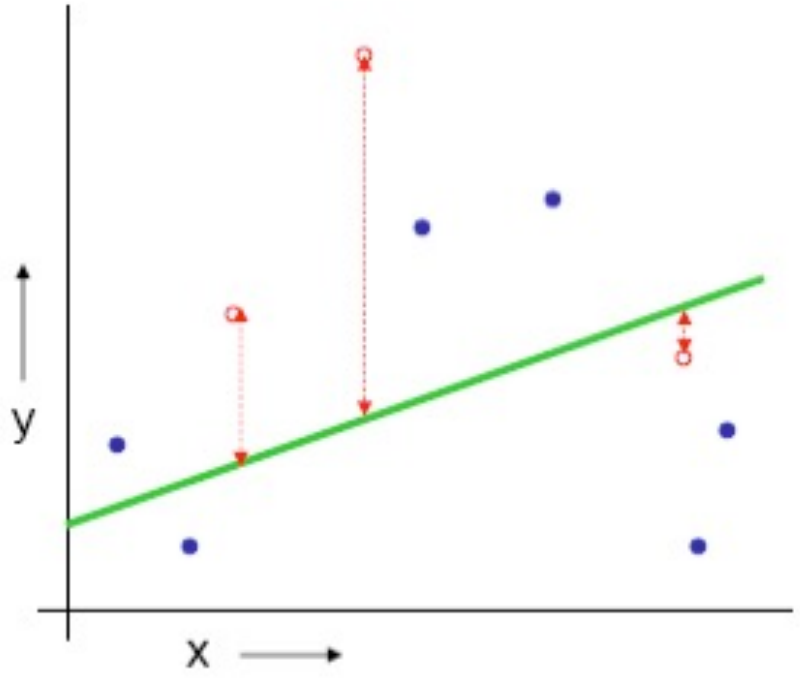
(Linear regression example)

1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

The test set method



(Linear regression example)

Mean Squared Error = 2.4

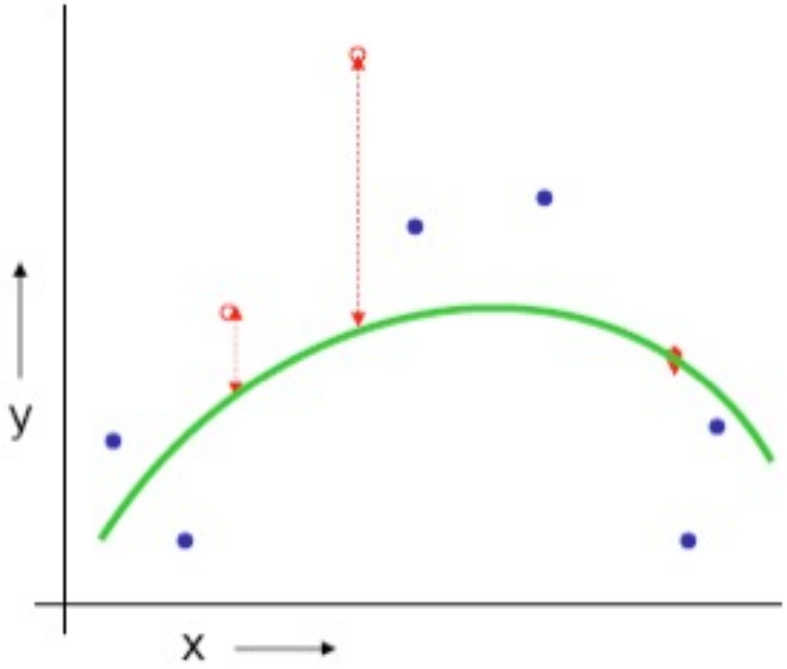
1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

4. Estimate your future performance with the test set

The test set method



(Quadratic regression example)

Mean Squared Error = 0.9

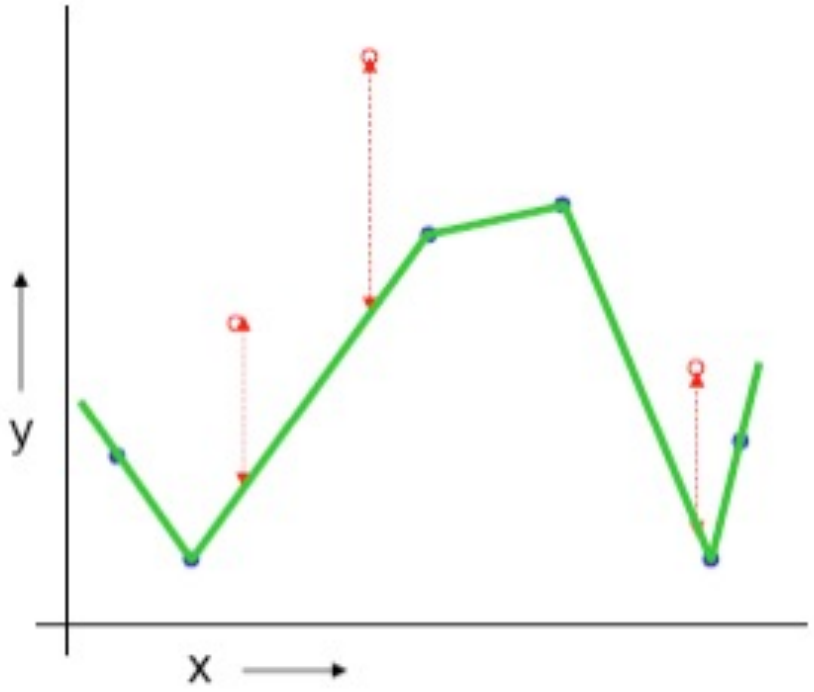
1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

4. Estimate your future performance with the test set

The test set method



(Join the dots example)

Mean Squared Error = 2.2

1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

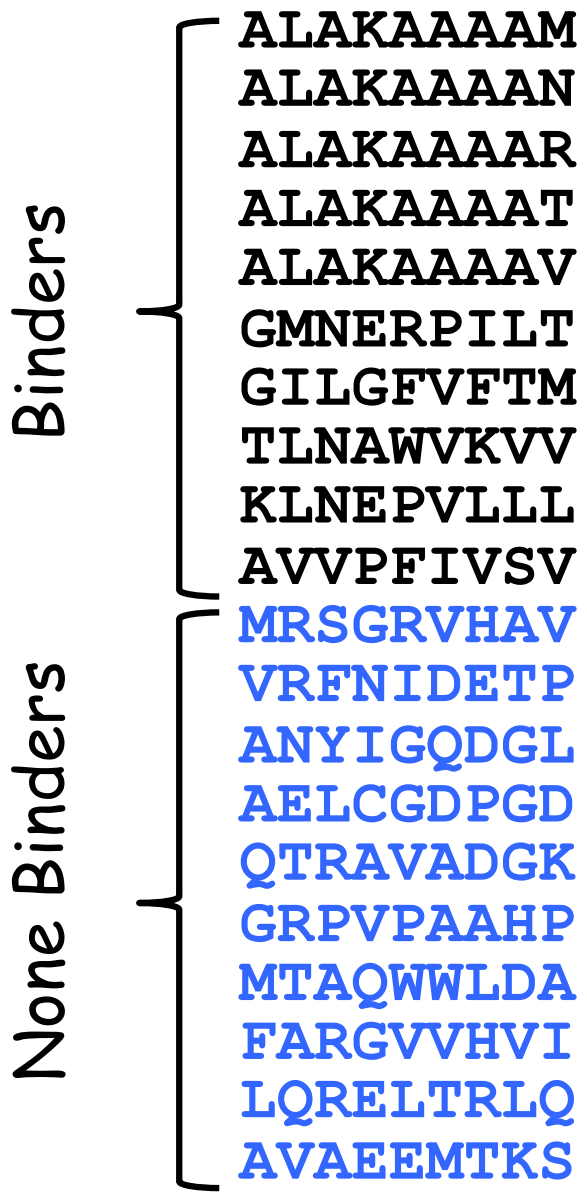
3. Perform your regression on the training set

4. Estimate your future performance with the test set

So quadratic function is best

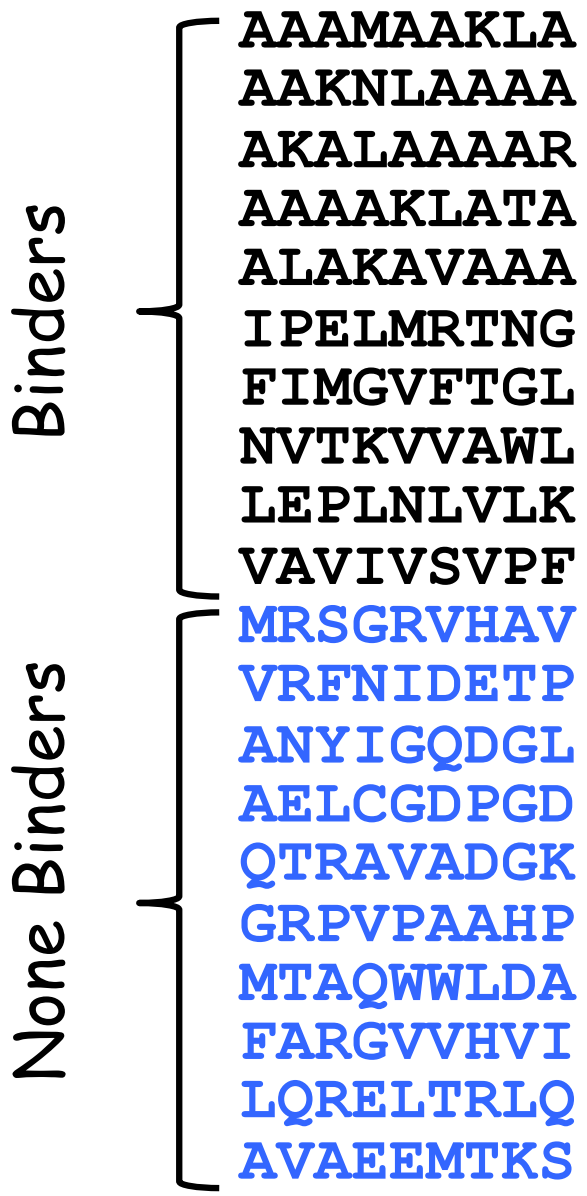
Learning noise

- Train PSSM on raw data
  - Fit 9*20 parameters to 9*10 data points
- Evaluate on training data
  - PCC = 0.93
  - AUC = 1.0
- Close to a perfect prediction method



Learning noise

- Train PSSM on Permuted (random) data
  - -Fit 9*20 parameters to 9*10 data points
- Evaluate on training data
  - -PCC = 0.95
  - -AUC = 1.0
- Close to a perfect prediction method AND
- Same performance as one the original data



Pattern Association

Pattern *association*.
Input is associated with output.
Classification, categorization, discrimination.

Goal: Find weights and thresholds.
Method: Training, not programming.

Training examples: I_j^α $(\alpha = 1, 2, \ldots; j = 1, 2, \ldots, N)$.

Desired targets: T_i^α $(\alpha = 1, 2, \ldots; i = 1, 2, \ldots, M)$.

Actual output: O_i^α $(\alpha = 1, 2, \ldots; i = 1, 2, \ldots, M)$.

Define quadratic error

$$E = \frac{1}{2} \sum_{\alpha, i} (O_i^\alpha - T_i^\alpha)^2$$

Measures least square deviation between desired result and actual output.

Minimize error by varying weights and thresholds.

$$\delta w = -\epsilon \frac{\partial E}{\partial w}$$

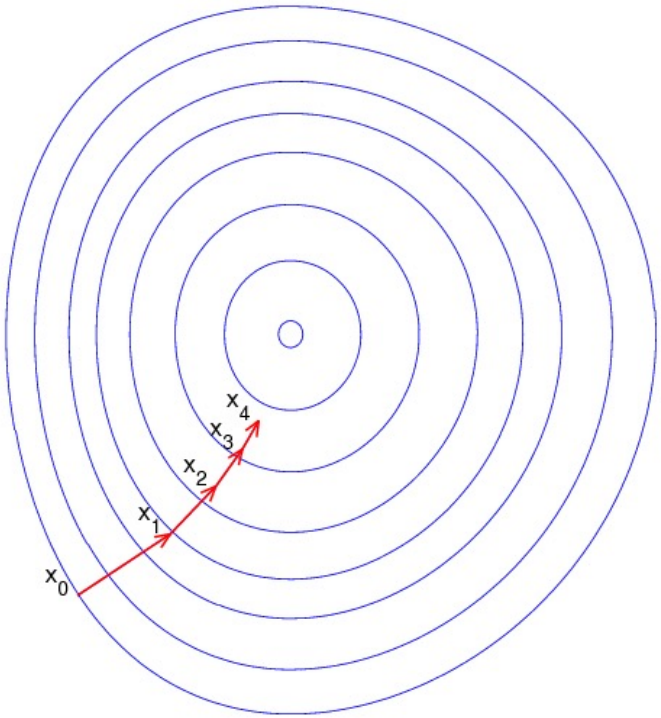
Gradient descent method.

Gradient decent (from wekipedia)

Gradient descent is based on the observation that if the real-valued function F(x) is defined and differentiable in a neighborhood of a point **a**, then F(x) decreases fastest if one goes from **a** in the direction of the negative gradient of F at **a**.
It follows that, if

$$b = a - \varepsilon \cdot \nabla F(a)$$

for $\varepsilon > 0$ a small enough number, then F(b)<F(a)



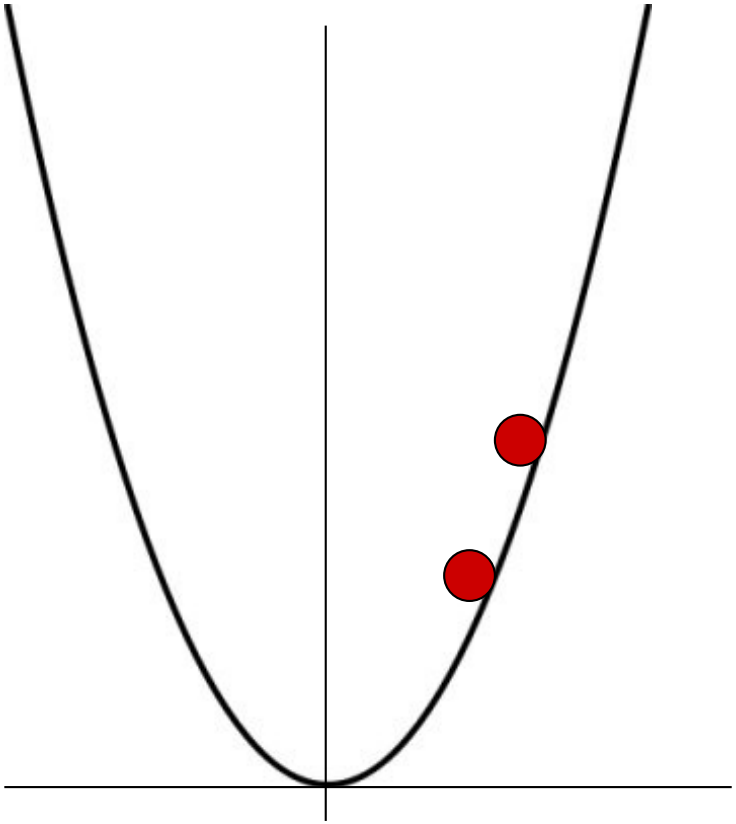
Gradient decent (example)

$$F(x) = x^2$$

$$\frac{\partial F}{\partial x} = 2 \cdot x$$

$$a = 2$$

$$F(a) = 4$$



$$b = a - \varepsilon \cdot \nabla F(a) = 2 - 0.1 \cdot 2 \cdot 2 = 1.6$$

Gradient decent

Weights are changed in the opposite direction of the gradient of the error

$$w_i' = w_i + \Delta w_i$$

$$\Delta w_i = -\varepsilon \cdot \frac{\partial E}{\partial w_i}$$

$$E = \frac{1}{2} \cdot (O - t)^2$$

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial w_i}$$

$$\frac{\partial E}{\partial O} = (O - t)$$

$$\frac{\partial O}{\partial w_i} = ?$$

Gradient decent (Linear function)

Weights are changed in the opposite direction of the gradient of the error

$$E = \frac{1}{2} \cdot (O - t)^2$$

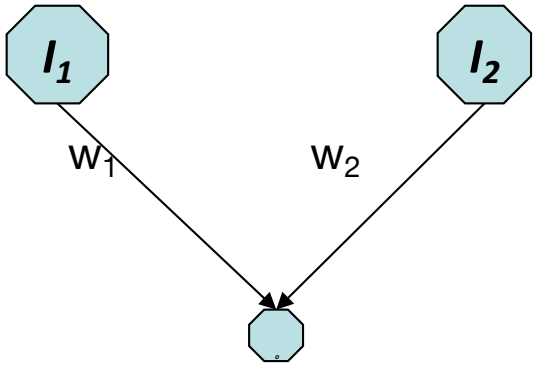
$$O = \sum_i I_i \cdot w_i$$

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial w_i}$$

$$\frac{\partial E}{\partial w_i} = (O - t) \cdot \frac{\partial O}{\partial w_i} = ?$$

Linear function

$$O = I_1 \cdot w_1 + I_2 \cdot w_2$$



Gradient decent

Weights are changed in the opposite direction of the gradient of the error

$$E = \frac{1}{2} \cdot (O - t)^2$$

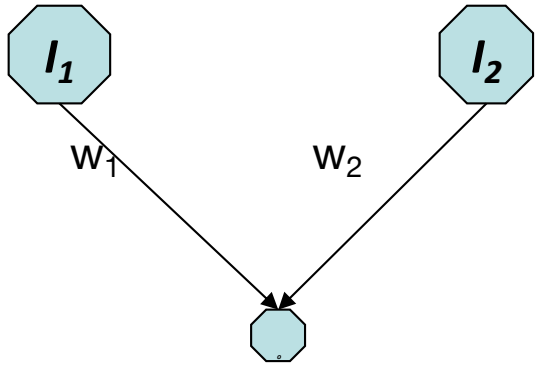
$$O = \sum_i I_i \cdot w_i$$

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial w_i}$$

$$\frac{\partial E}{\partial w_i} = (O - t) \cdot \frac{\partial O}{\partial w_i} = (O - t) \cdot I_i$$

Linear function

$$o = I_1 \cdot w_1 + I_2 \cdot w_2$$



Gradient decent. Example

Weights are changed in the opposite direction of the gradient of the error

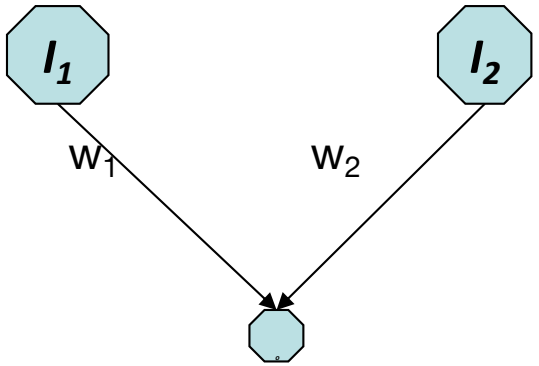
$$w_i' = w_i + \Delta w_i$$

$$E = \frac{1}{2} \cdot (O - t)^2$$

$$O = \sum_i w_i \cdot I_i$$

Linear function

$$O = I_1 \cdot w_1 + I_2 \cdot w_2$$



$$\Delta w_i = -\varepsilon \cdot \frac{\partial E}{\partial w_i} = -\varepsilon \cdot (O - t) \cdot I_i$$

Gradient decent.

Weights are changed in the opposite direction of the gradient of the error

$$w_i' = w_i + \Delta w_i$$

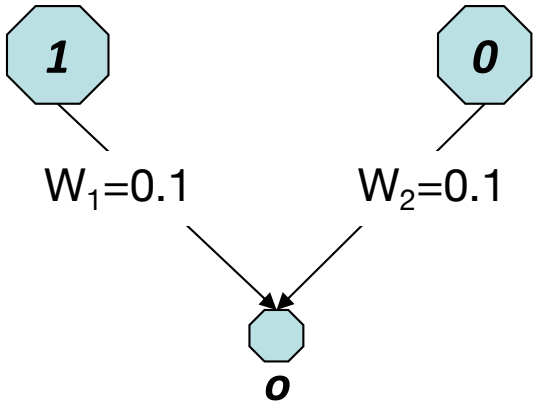
$$E = \frac{1}{2} \cdot (O - t)^2$$

$$O = \sum_i w_i \cdot I_i$$

$$\Delta w_i = -\varepsilon \cdot \frac{\partial E}{\partial w_i} = -\varepsilon \cdot (O - t) \cdot I_i$$

Linearfunction

$$O = I_1 \cdot w_1 + I_2 \cdot w_2$$



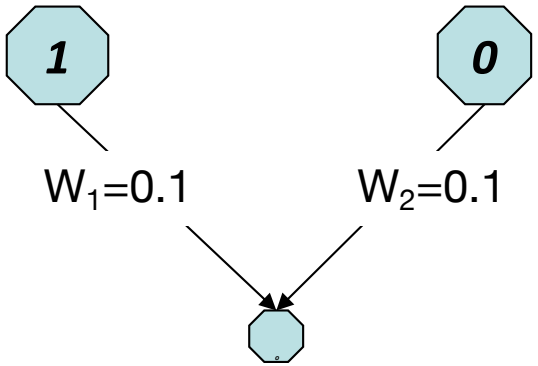
What are the weights after 2 forward (calculate predictions) and backward (update weights) iterations with the given input, and has the error decrease (use ε=0.1, and t=1)?

Fill out the table

What are the weights after 2 forward/backward iterations with the given input, and has the error decrease (use ε=0.1, t=1)?

Linear function

$$O = I_1 \cdot w_1 + I_2 \cdot w_2$$



itr	W1	W2	O
0	0.1	0.1	
1			
2			

Fill out the table

CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

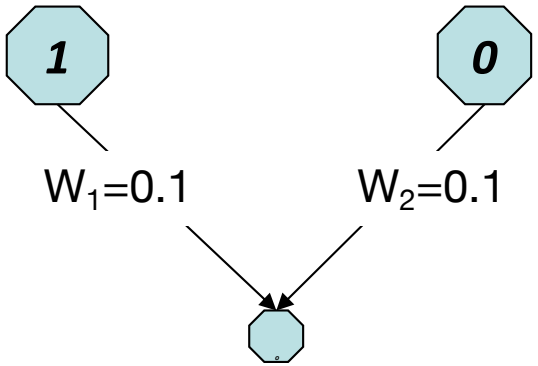
Hand out

Fill out the table

What are the weights after 2 forward/backward iterations with the given input, and has the error decrease (use ε=0.1, t=1)?

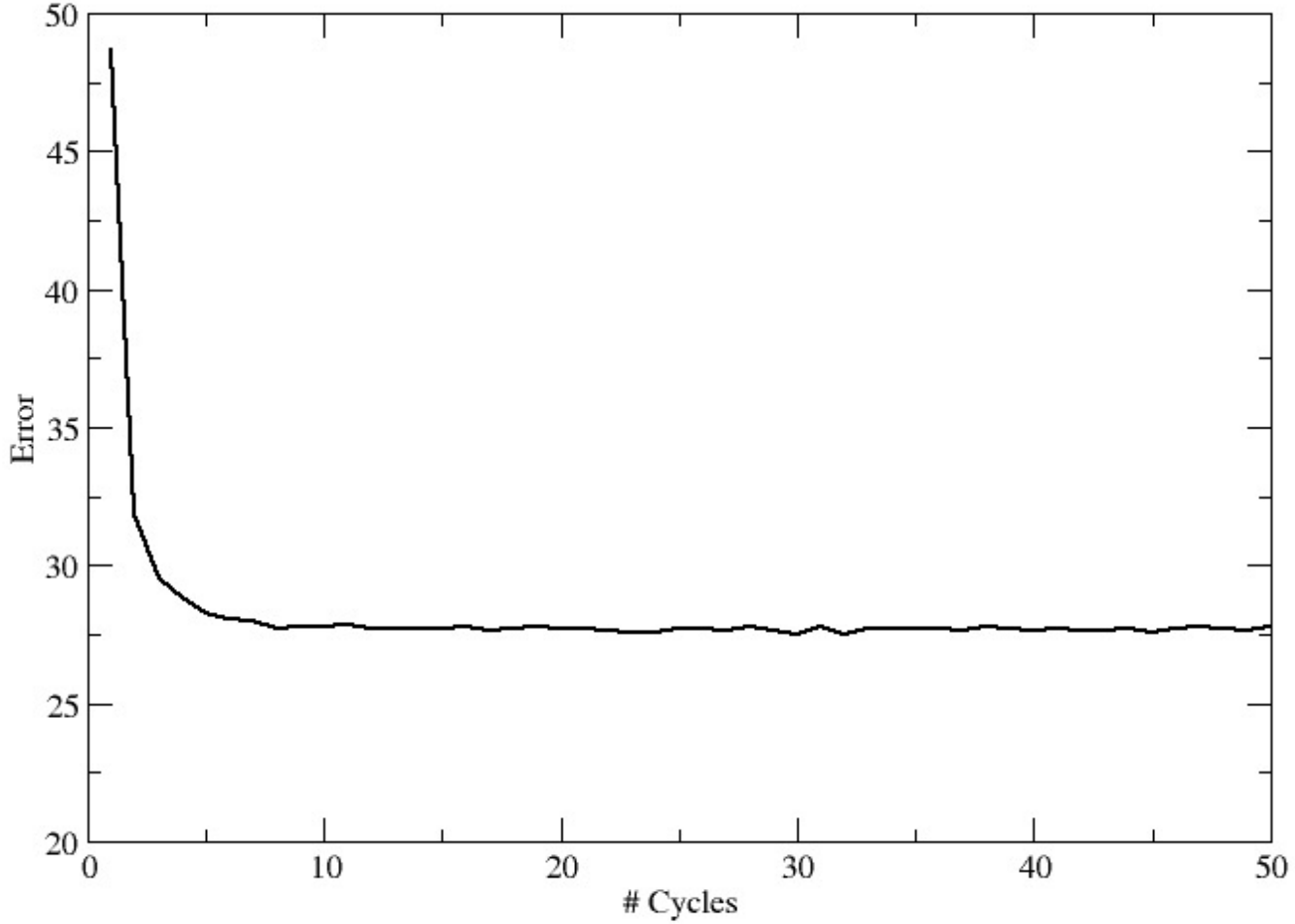
Linear function

$$O = I_1 \cdot w_1 + I_2 \cdot w_2$$

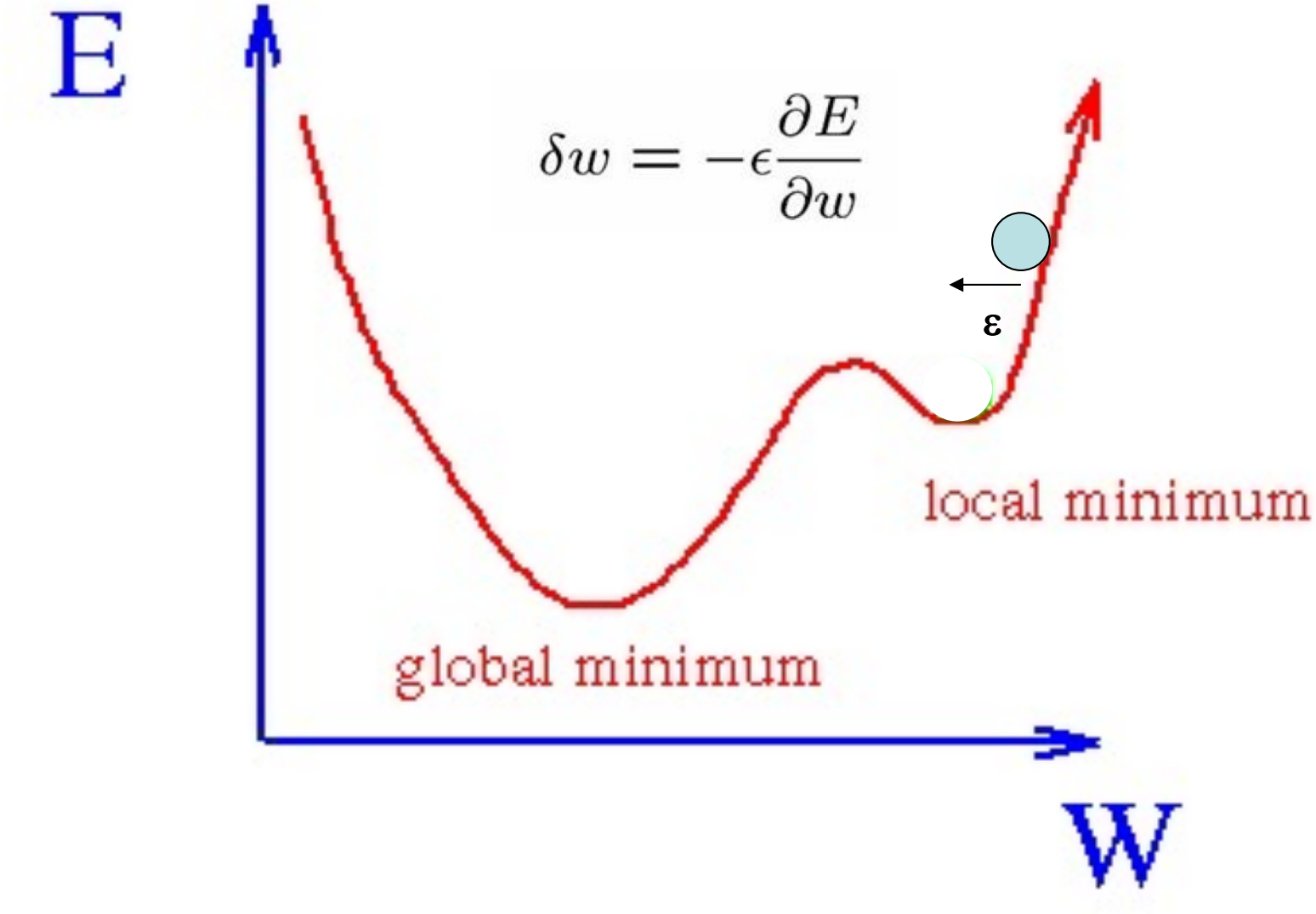


itr	W1	W2	O
0	0.1	0.1	0.1
1	0.19	0.1	0.19
2	0.27	0.1	0.27

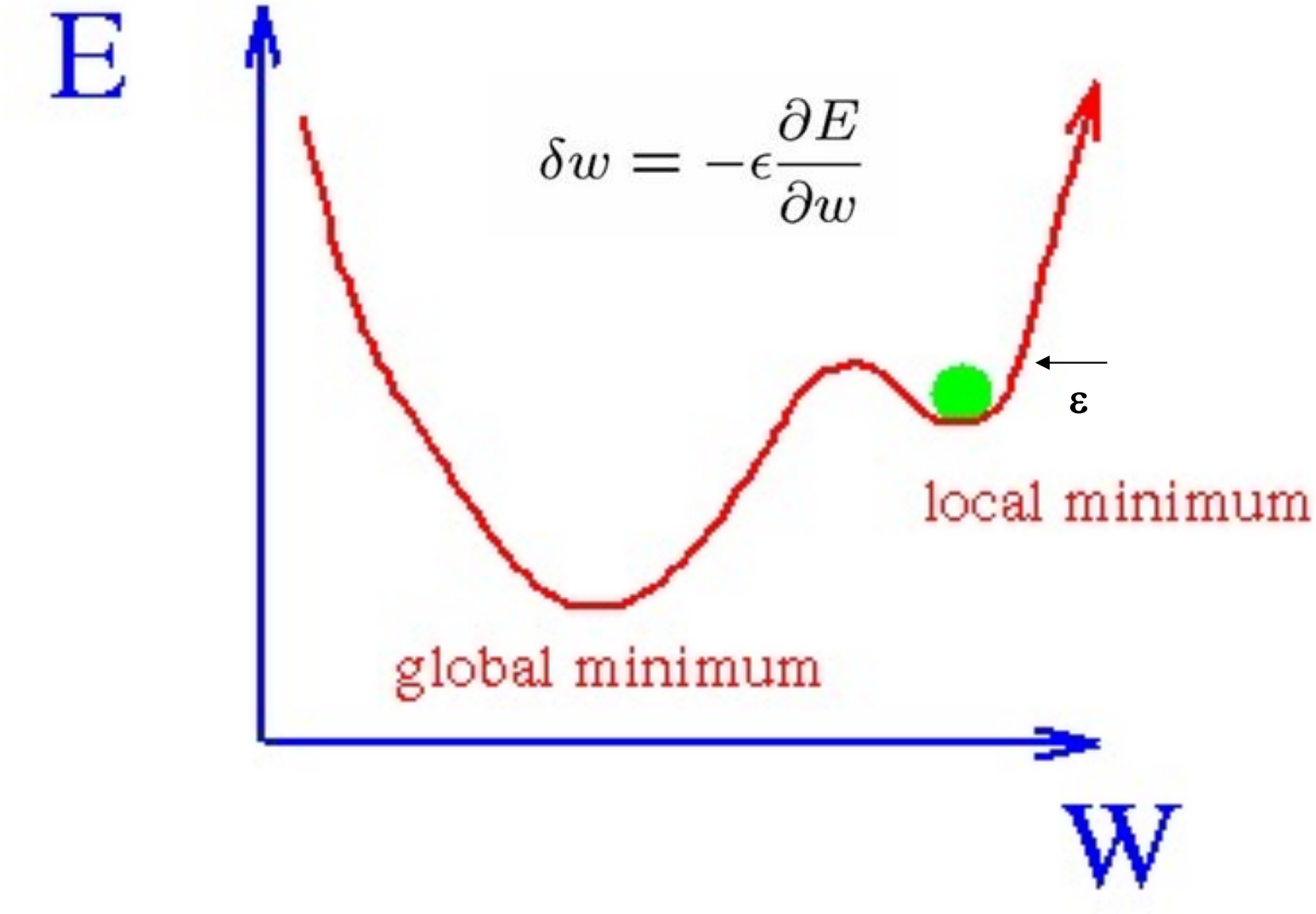
Gradient decent



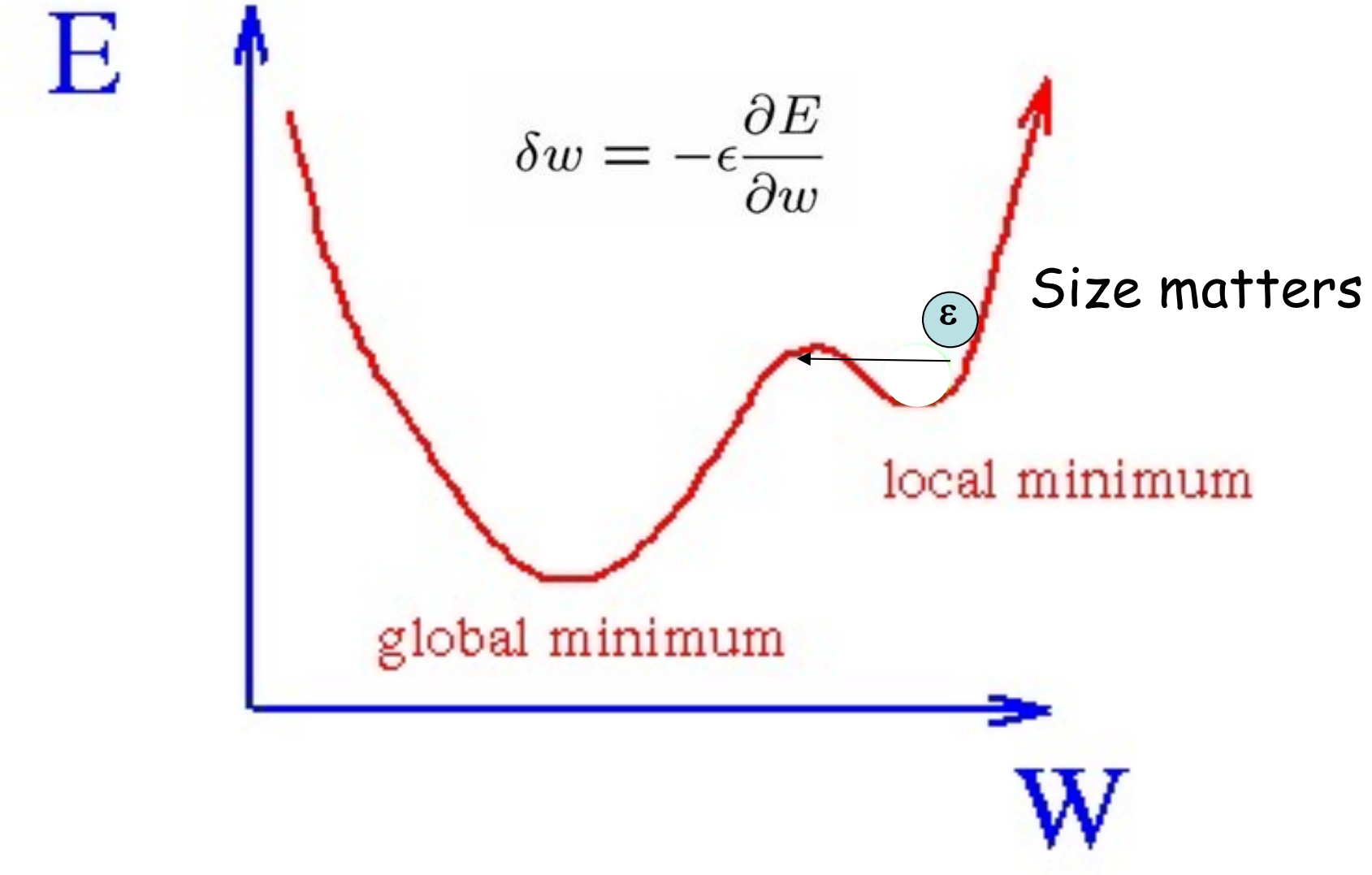
Training and error reduction



Training and error reduction



Training and error reduction



Is there anything beyond weight matrices?

Weight matrices (PSSM)

- A weight matrix is given as

  $W_{ij} = \log(p_{ij}/q_j)$

  - where i is a position in the motif, and j an amino acid. q_j is the background frequency for amino acid j.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	0.6	0.4	-3.5	-2.4	-0.4	-1.9	-2.7	0.3	-1.1	1.0	0.3	0.0	1.4	1.2	-2.7	1.4	-1.2	-2.0	1.1	0.7
2	-1.6	-6.6	-6.5	-5.4	-2.5	-4.0	-4.7	-3.7	-6.3	1.0	5.1	-3.7	3.1	-4.2	-4.3	-4.2	-0.2	-5.9	-3.8	0.4
3	0.2	-1.3	0.1	1.5	0.0	-1.8	-3.3	0.4	0.5	-1.0	0.3	-2.5	1.2	1.0	-0.1	-0.3	-0.5	3.4	1.6	0.0
4	-0.1	-0.1	-2.0	2.0	-1.6	0.5	0.8	2.0	-3.3	0.1	-1.7	-1.0	-2.2	-1.6	1.7	-0.6	-0.2	1.3	-6.8	-0.7
5	-1.6	-0.1	0.1	-2.2	-1.2	0.4	-0.5	1.9	1.2	-2.2	-0.5	-1.3	-2.2	1.7	1.2	-2.5	-0.1	1.7	1.5	1.0
6	-0.7	-1.4	-1.0	-2.3	1.1	-1.3	-1.4	-0.2	-1.0	1.8	0.8	-1.9	0.2	1.0	-0.4	-0.6	0.4	-0.5	-0.0	2.1
7	1.1	-3.8	-0.2	-1.3	1.3	-0.3	-1.3	-1.4	2.1	0.6	0.7	-5.0	1.1	0.9	1.3	-0.5	-0.9	2.9	-0.4	0.5
8	-2.2	1.0	-0.8	-2.9	-1.4	0.4	0.1	-0.4	0.2	-0.0	1.1	-0.5	-0.5	0.7	-0.3	0.8	0.8	-0.7	1.3	-1.1
9	-0.2	-3.5	-6.1	-4.5	0.7	-0.8	-2.5	-4.0	-2.6	0.9	2.8	-3.0	-1.8	-1.4	-6.2	-1.9	-1.6	-4.9	-1.6	4.5

- W is a L x 20 matrix, L is motif length

NETtalk

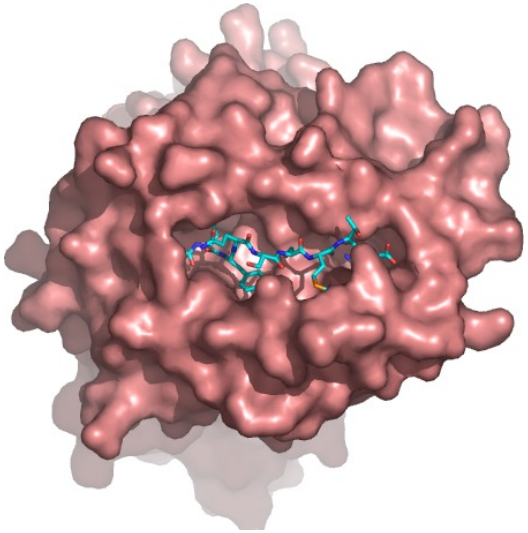
(T. Sejnowski and C. Rosenberg, 1987)

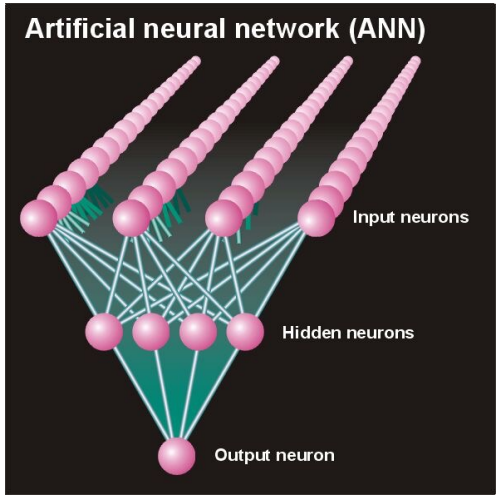
M**a**ry h**a**d **a** little l**a**mb

Three of the **a**'s must be pronounced differently! Reading aloud is a *context sensitive* cognitive skill.

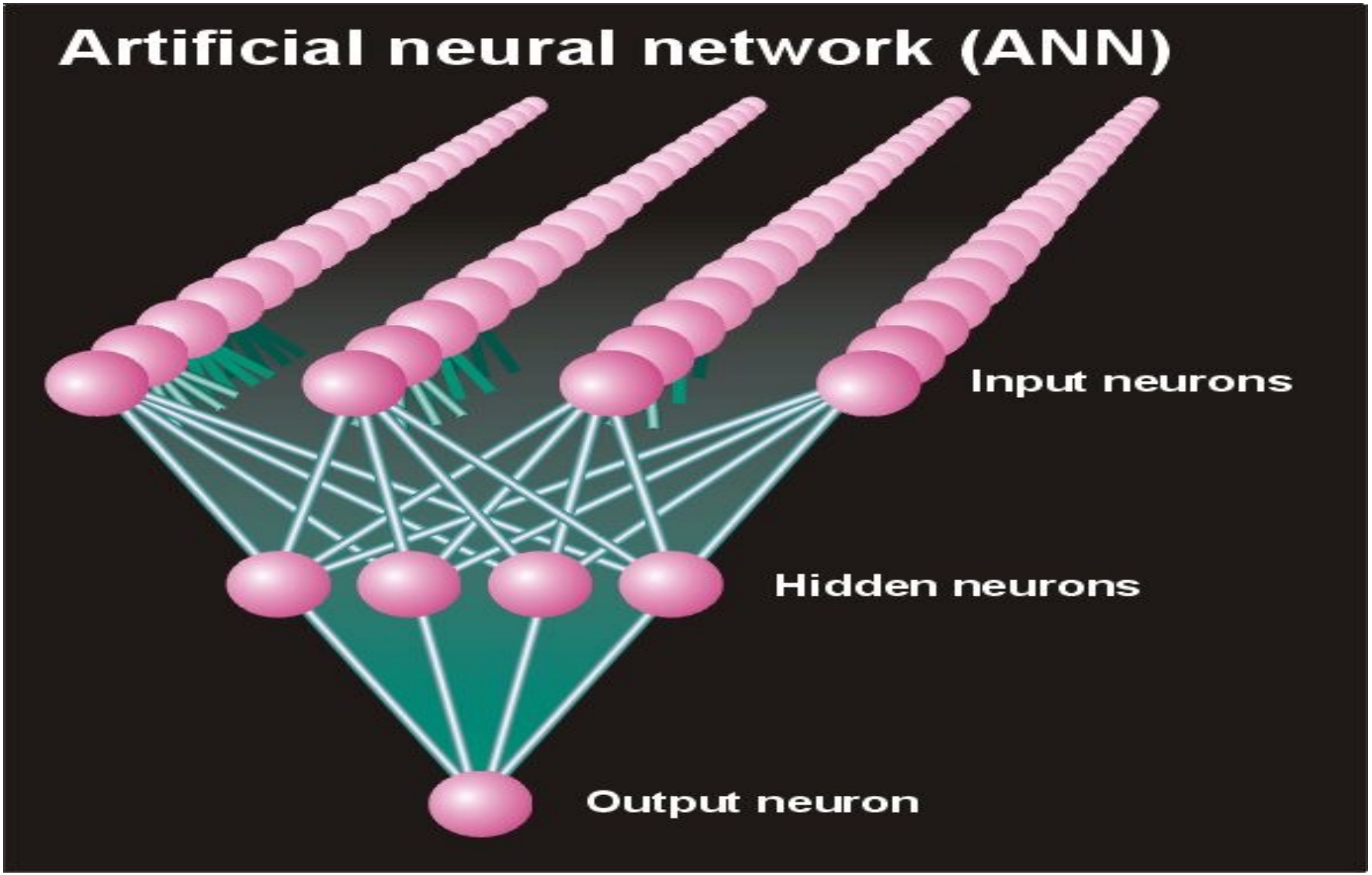
Is there anything beyond weight matrices

- The effect on the binding affinity of having a given amino acid at one position can be influenced by the amino acids at other positions in the peptide (sequence correlations).
  - Two adjacent amino acids may for example compete for the space in a pocket in the MHC molecule.
- Artificial neural networks (ANN) are ideally suited to take such correlations into account

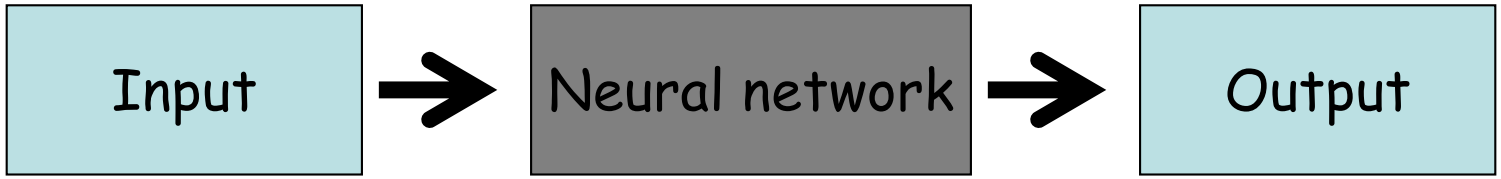




Data driven prediction methods

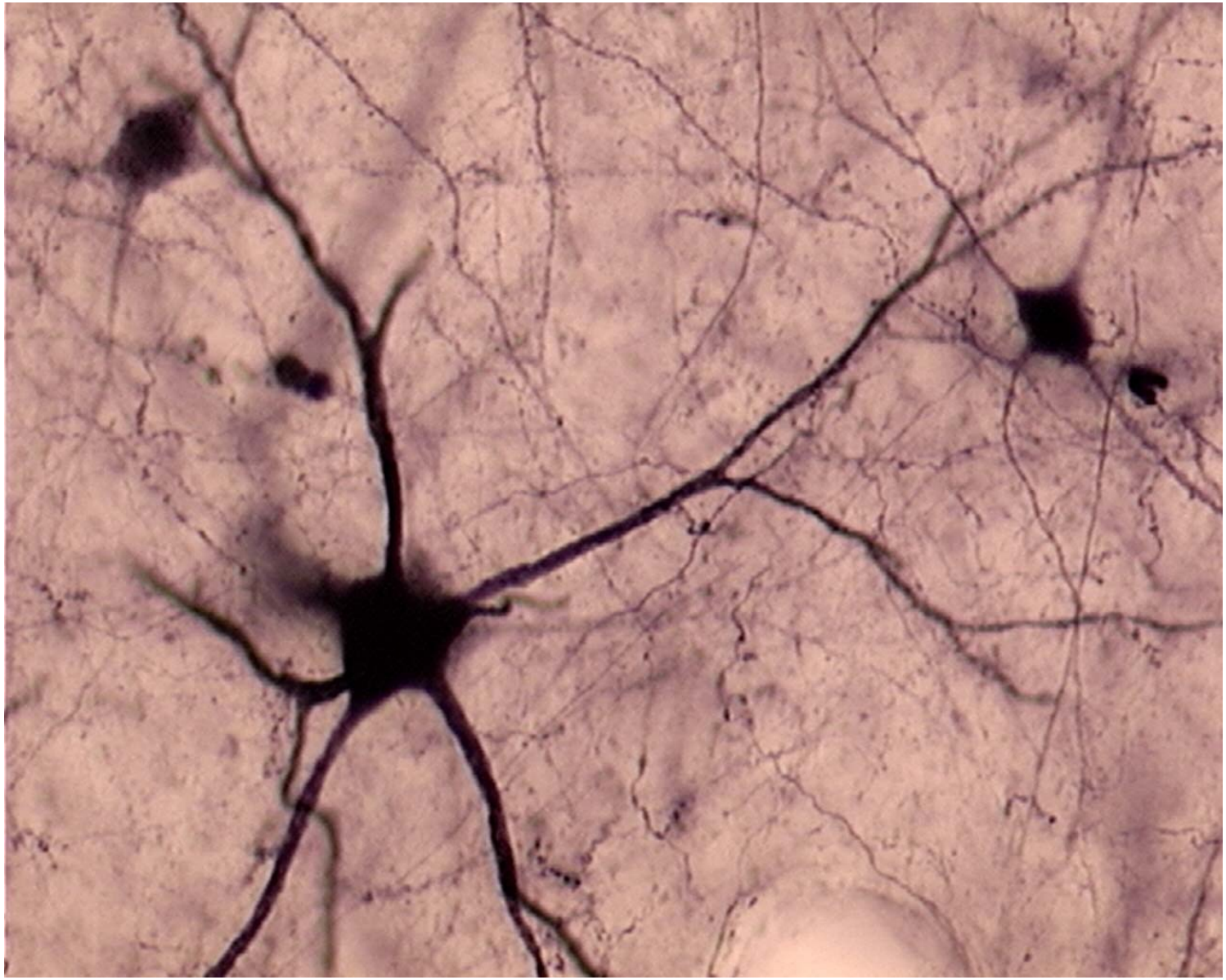


Objectives

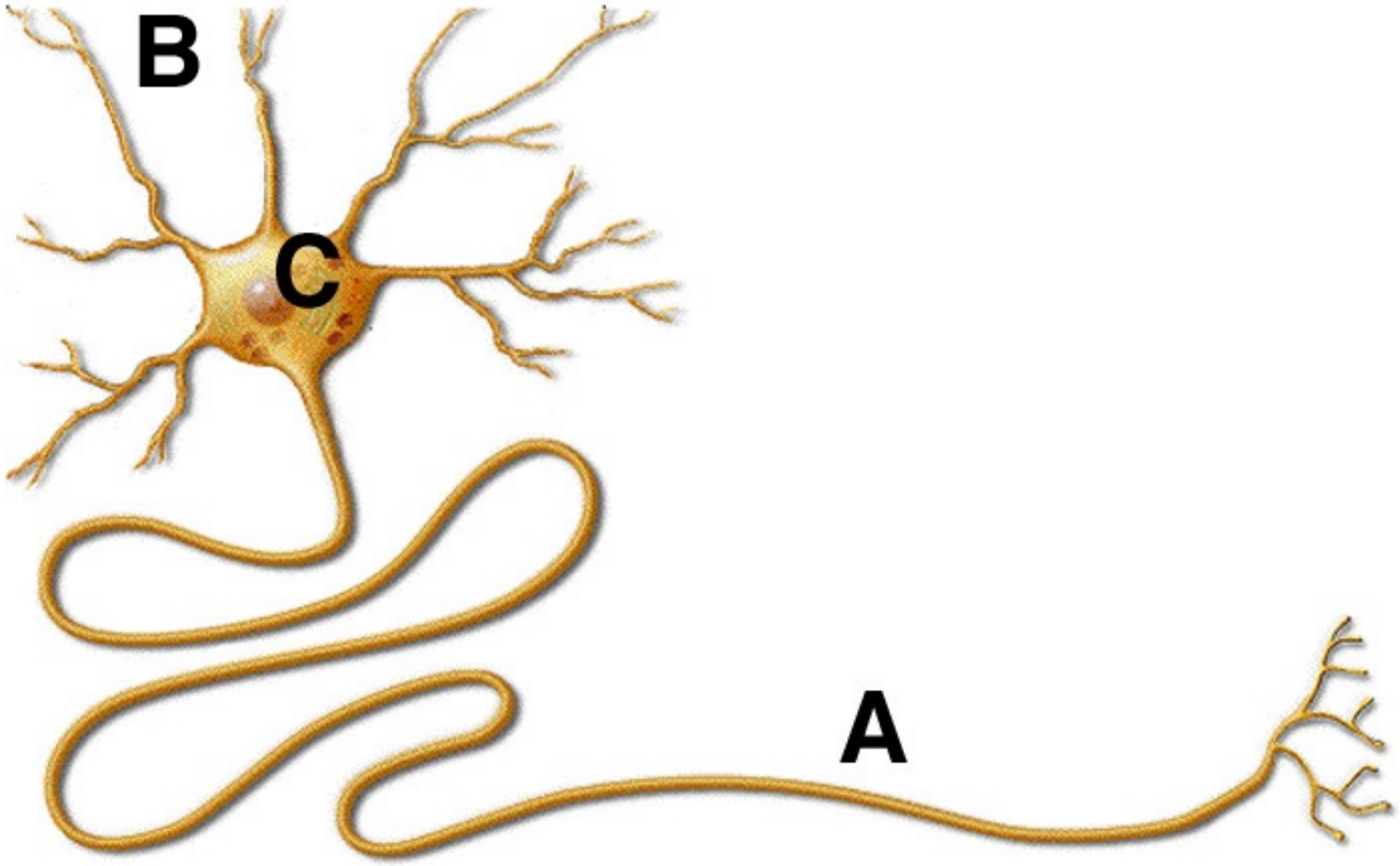


- Neural network:
  - is a black box that no one can understand
  - over-predicts performance
  - Overfitting - many thousand parameters fitted on few data

Biological Neural network



Biological neuron structure



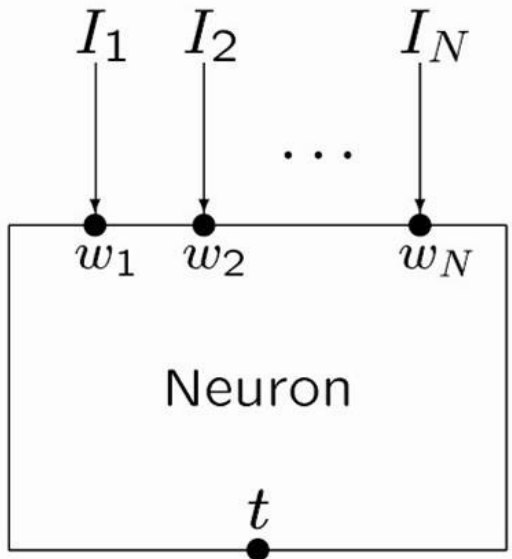
Artificial neuron

Input signals

Synaptic weights

Threshold

Output signal



$$O = \sigma\left(\sum_{n=1}^{N} w_n I_n - t\right)$$

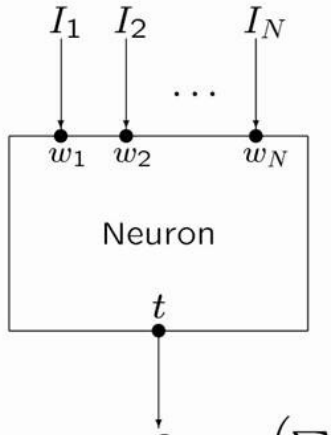
Artificial neuron

Input signals

Synaptic weights

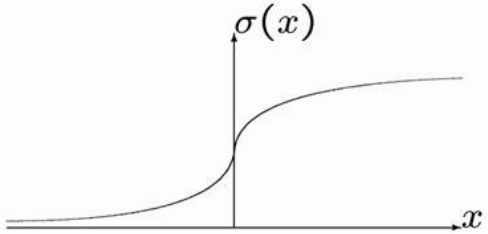
Threshold

Output signal

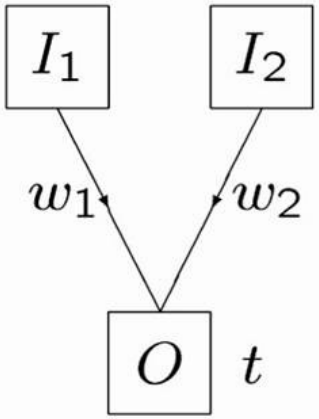


$$O = \sigma\left(\sum_{n=1}^{N} w_n I_n - t\right)$$

- Strongly simplified in relation to the biology.
- Dates back to McCulloch and Pitts, 1943.
- Linear combination af input, weights og threshold.
- Non-linear (sigmoid) response function.
- Typical choice: $\sigma(x) = 1/(1 + e^{-x})$



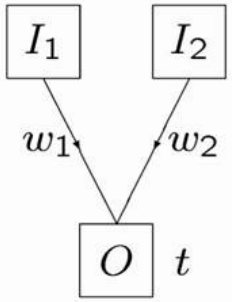
Linear separation by simple neural network



Two input features and one output.

$$O = \begin{cases} 1 & \text{for } w_1 I_1 + w_2 I_2 > t \\ 0 & \text{otherwise} \end{cases}$$

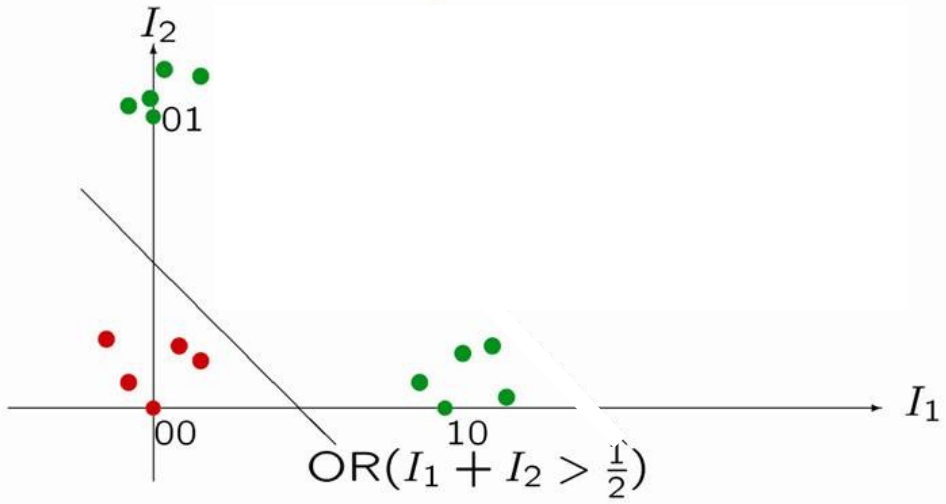
Linear separation by simple neural network



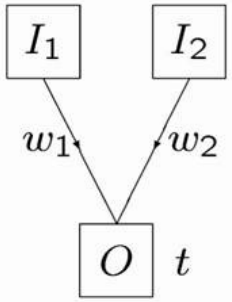
Two input features and one output.

$$O = \begin{cases} 1 & \text{for } w_1 I_1 + w_2 I_2 > t \\ 0 & \text{otherwise} \end{cases}$$

Equation $w_1 I_1 + w_2 I_2 = t$ is straight line in $I_1 I_2$-plane:



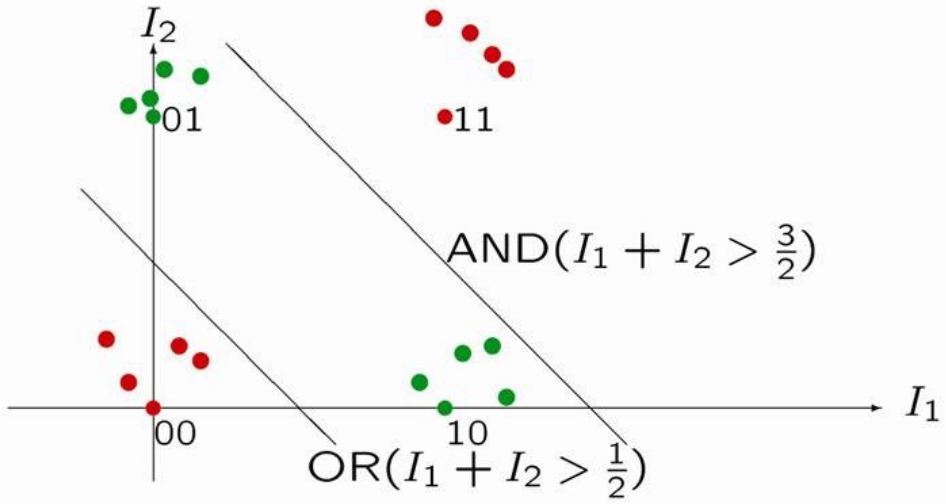
Linear separation by simple neural network



Two input features and one output.

$$O = \begin{cases} 1 & \text{for } w_1 I_1 + w_2 I_2 > t \\ 0 & \text{otherwise} \end{cases}$$

Equation $w_1 I_1 + w_2 I_2 = t$ is straight line in $I_1 I_2$-plane:



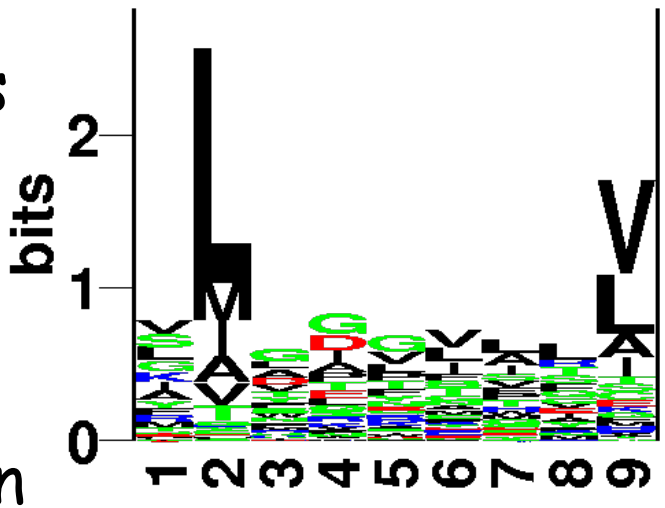
MHC peptide binding

CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

SLLPAIVEL YLLPAIVHI TLWVDPYEV GLVPFLVSV KLLEPVLLL LLDVPTAAV LLDVPTAAV LLDVPTAAV
LLDVPTAAV VLFRGGPRG MVDGTLLLL YMNGTMSQV MLLSVPLLL SLLGLLVEV ALLPPINIL TLIKIQHTL
HLIDYLVTS ILAPPVVKL ALFPQLVIL GILGFVFTL STNRQSGRQ GLDVLTAKV RILGAVAKV QVCERIPTI
ILFGHENRV ILMEHIHKL ILDQKINEV SLAGGIIGV LLIENVASL FLLWATAEA SLPDFGISY KKREEAPSL
LERPGGNEI ALSNLEVKL ALNELLQHV DLERKVESL FLGENISNF ALSDHHIYL GLSEFTEYL STAPPAHGV
PLDGEYFTL GVLVGVALI RTLDKVLEV HLSTAFARV RLDSYVRSL YMNGTMSQV GILGFVFTL ILKEPVHGV
ILGFVFTLT LLFGYPVYV GLSPTVWLS WLSLLVPFV FLPSDFFPS CLGGLLTMV FIAGNSAYE KLGEFYNQM
KLVALGINA DLMGYIPLV RLVTLKDIV MLLAVLYCL AAGIGILTV YLEPGPVTA LLDGTATLR ITDQVPFSV
KTWGQYWQV TITDQVPFS AFHHVAREL YLNKIQNSL MMRKLAILS AIMDKNIIL IMDKNIILK SMVGNWAKV
SLLAPGAKQ KIFGSLAFL ELVSEFSRM KLTPLCVTL VLYRYGSFS YIGEVLVSV CINGVCWTV VMNILLQYV
ILTVILGVL KVLEYVIKV FLWGPRALV GLSRYVARL FLLTRILTI HLGNVKYLV GIAGGLALL GLQDCTMLV
TGAPVTYST VIYQYMDDL VLPDVFIRC VLPDVFIRC AVGIGIAVV LVVLGLLAV ALGLGLLPV GIGIGVLAA
GAGIGVAVL IAGIGILAI LIVIGILIL LAGIGLIAA VDGIGILTI GAGIGVLTA AAGIGIIQI QAGIGILLA
KARDPHSGH KACDPHSGH ACDPHSGHF SLYNTVATL RGPGRAFVT NLVPMVATV GLHCYEQLV PLKQHFQIV
AVFDRKSDA LLDFVRFMG VLVKSPNHV GLAPPQHLI LLGRNSFEV PLTFGWCYK VLEWRFDSR TLNAWVKVV
GLCTLVAML FIDSYICQV IISAVVGIL VMAGVGSPY LLWTLVVLL SVRDRLARL LLMDCSGSI CLTSTVQLV
VLHDDLLEA LMWITQCFL SLLMWITQC QLSLLMWIT LLGATCMFV RLTRFLSRV YMDGTMSQV FLTPKKLQC
ISNDVCAQV VKTDGNPPE SVYDFFVWL FLYGALLLA VLFSSDFRI LMWAKIGPV SLLLELEEV SLSRFSWGA
YTAFTIPSI RLMKQDFSV RLPRIFCSC FLWGPRAYA RLLQETELV SLFEGIDFY SLDQSVVEL RLNMFTPYI
NMFTPYIGV LMIIPLINV TLFIGSHVV SLVIVTTFV VLQWASLAV ILAKFLHWL STAPPHVNV LLLLTVLTV
VVLGVVFGI ILHNGAYSL MIMVKCWMI MLGTHTMEV MLGTHTMEV SLADTNSLA LLWAARPRL GVALQTMKQ
GLYDGMEHL KMVELVHFL YLQLVFGIE MLMAQEALA LMAQEALAF VYDGREHTV YLSGANLNL RMFPNAPYL
EAAGIGILT TLDSQVMSL STPPPGTRV KVAELVHFL IMIGVLVGV ALCRWGLLL LLFAGVQCQ VLLCESTAV
YLSTAFARV YLLEMLWRL SLDDYNHLV RTLDKVLEV GLPVEYLQV KLIANNTRV FIYAGSLSA KLVANNTRL
FLDEFMEGV ALQPGTALL VLDGLDVLL SLYSFPEPE ALYVDSLFF SLLQHLIGL ELTLGEFLK MINAYLDKL
AAGIGILTV FLPSDFFPS SVRDRLARL SLREWLLRI LLSAWILTA AAGIGILTV AVPDEIPPL FAYDGKDYI
AAGIGILTV FLPSDFFPS AAGIGILTV FLPSDFFPS AAGIGILTV FLWGPRALV ETVSEQSNV ITLWQRPLV

Mutual information

- How is mutual information calculated?
- Information content was calculated as
  - Gives information in a single position



$$I = \sum_a p_a \log(\frac{p_a}{q_a})$$

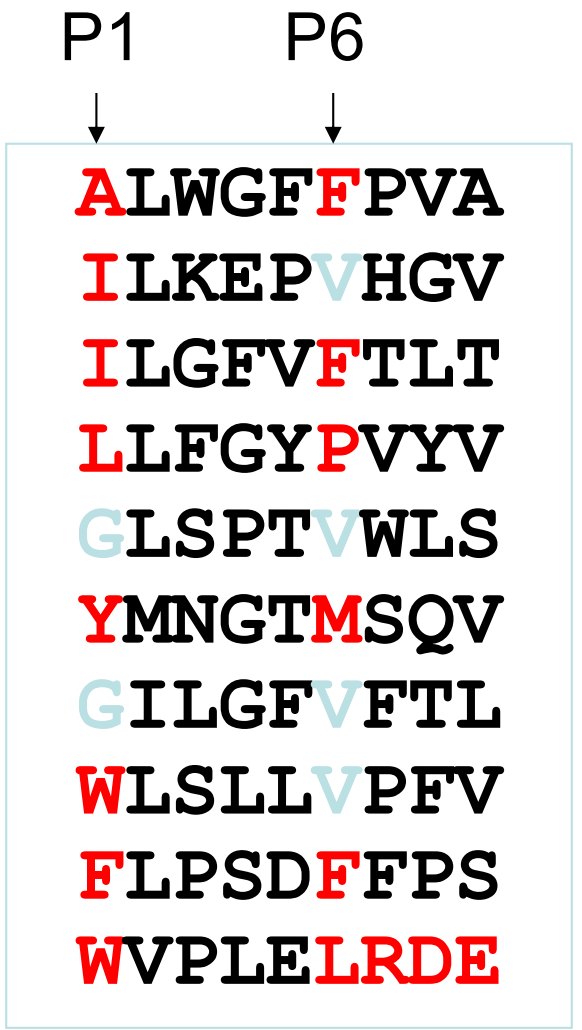
- Similar relation for mutual information
  - Gives mutual information between two positions

$$I = \sum_{a,b} p_{ab} \log(\frac{p_{ab}}{p_a \cdot p_b})$$

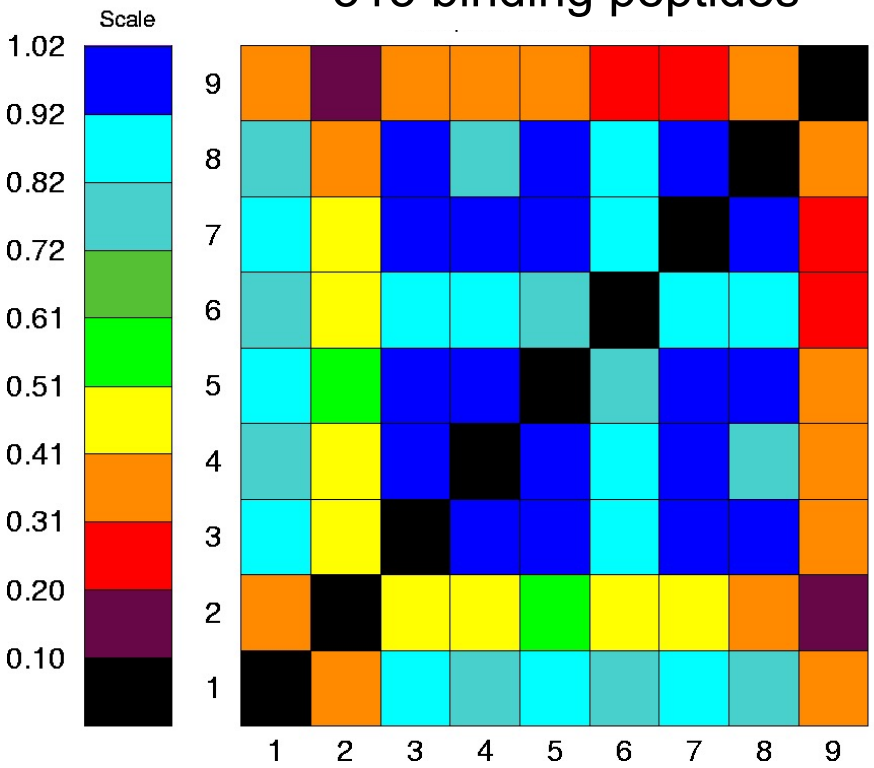
Mutual information. Example

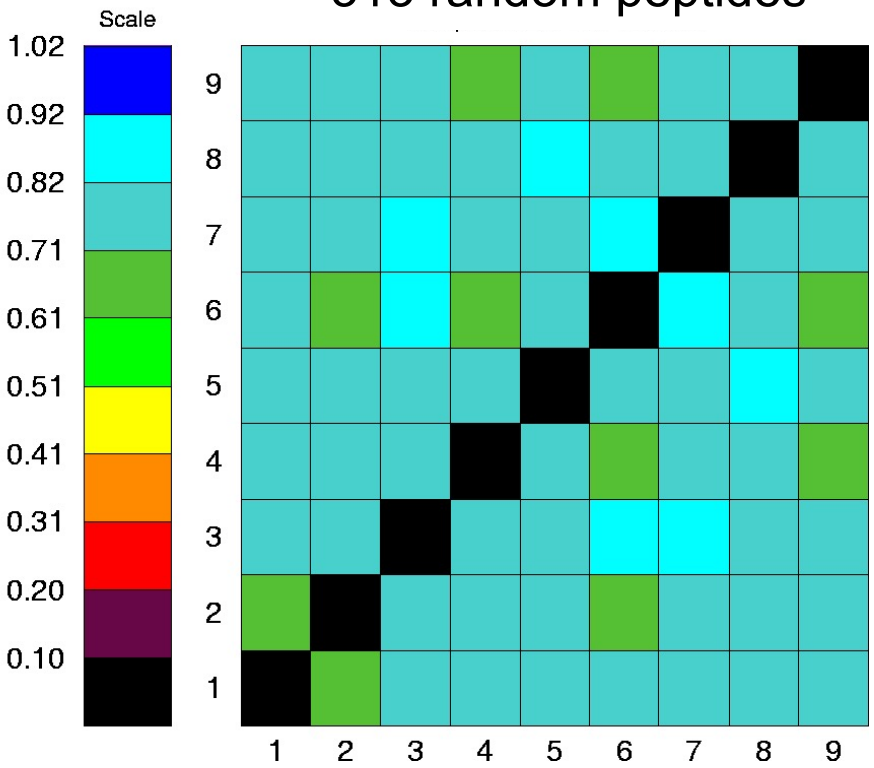
Knowing that you have G at P_1 allows you to make an educated guess on what you will find at P_6.
$P(V_6) = 4/10$. $P(V_6|G_1) = 1.0$!

$$I = \sum_{a,b} p_{ab} \log\left(\frac{p_{ab}}{p_a \cdot p_b}\right)$$



Mutual information





Higher order sequence correlations

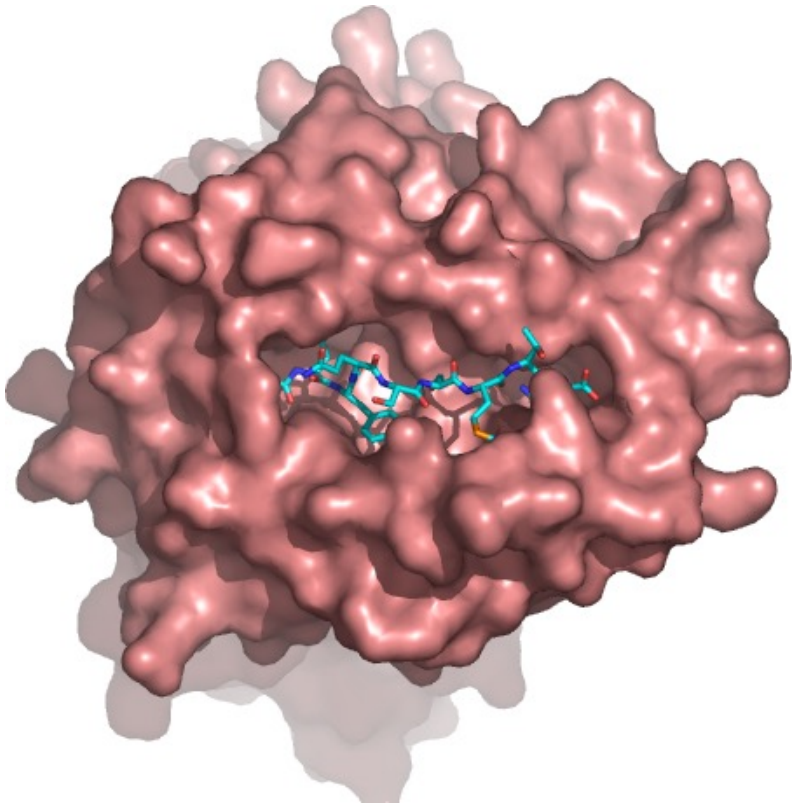
- Neural networks can learn higher order correlations!
  - What does this mean?

Say that the peptide needs one and only one large amino acid in the positions P3 and P4 to fill the binding cleft

How would you formulate this to test if a peptide can bind?

S S => 0
L S => 1
S L => 1
L L => 0

=> **XOR function**



Neural networks

- Neural networks can learn higher order correlations

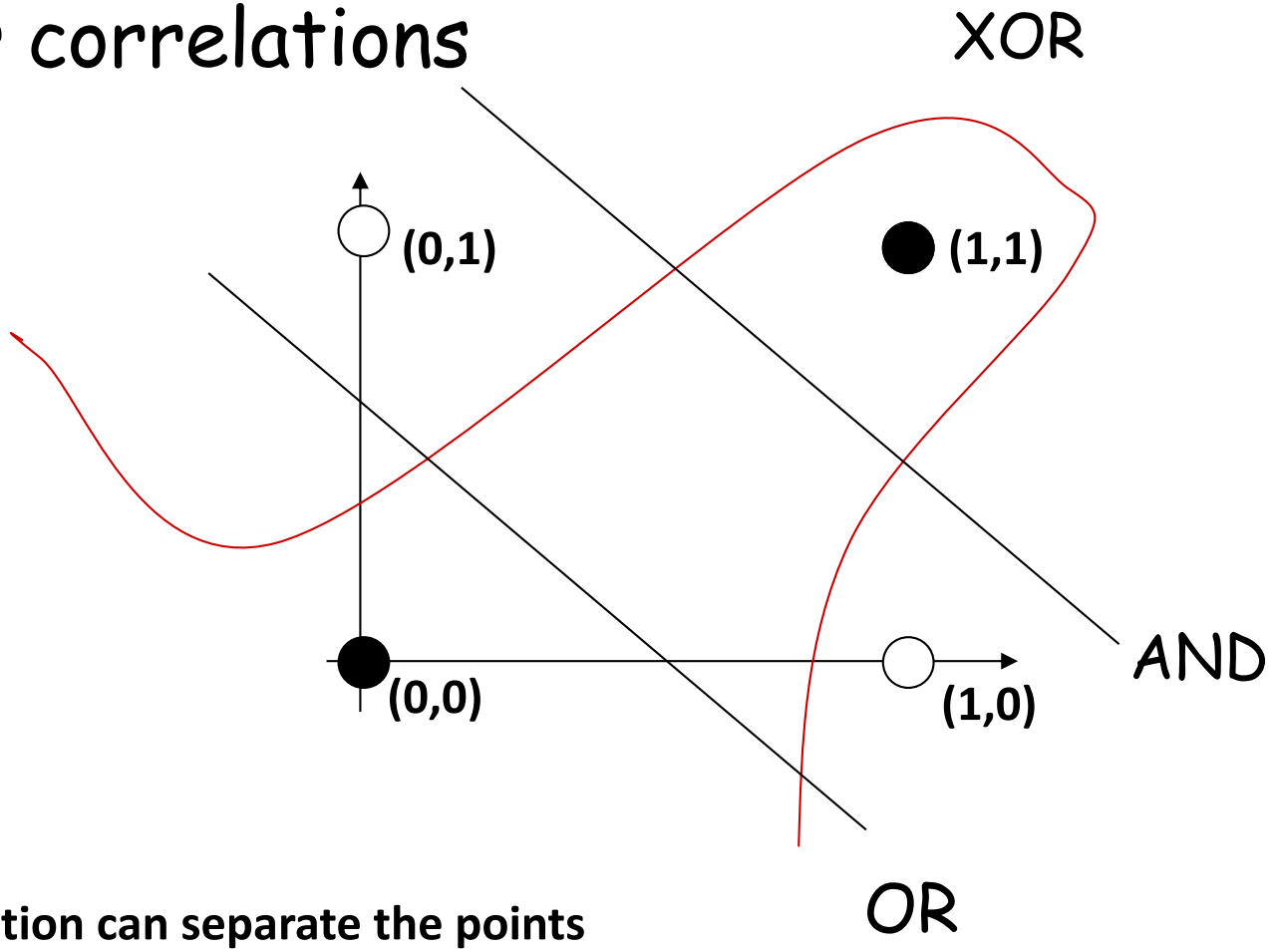
XOR function:

0 0 => 0

1 0 => 1

0 1 => 1

1 1 => 0

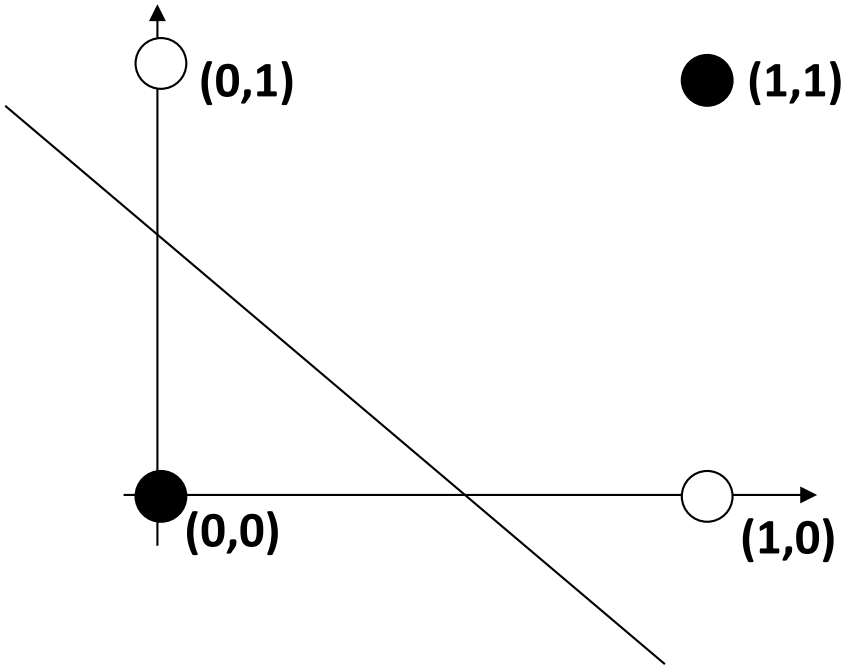


No linear function can separate the points

Error estimates

Encode L as 1 and S as 0

XOR	Predict	Error
S S (0 0) => 0	0	0
L S (1 0) => 1	1	0
S L (0 1) => 1	1	0
L L (1 1) => 0	1	1

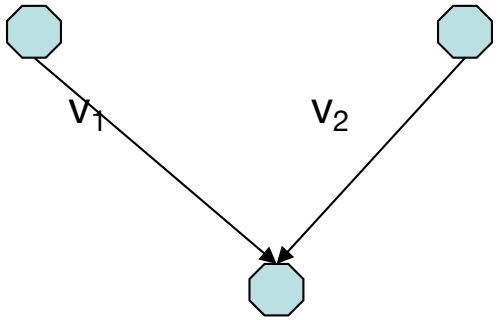


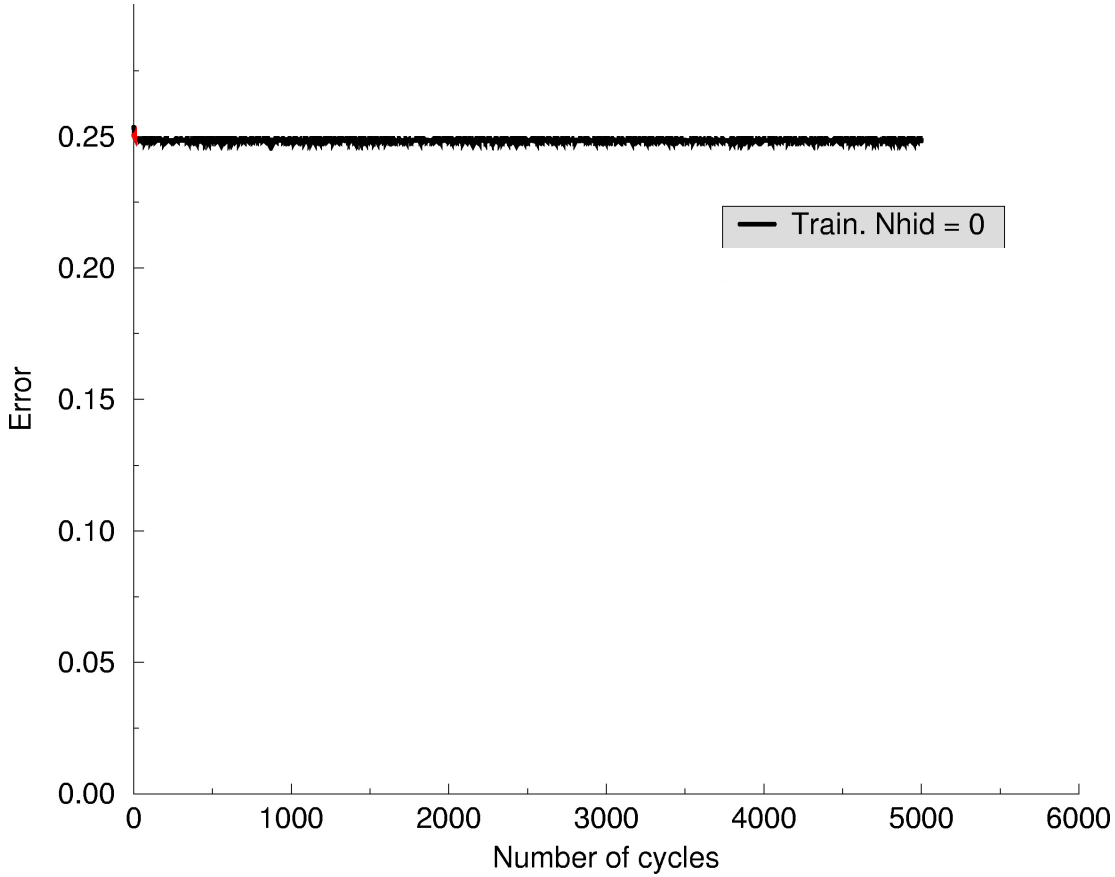
Mean error: 1/4

Neural networks

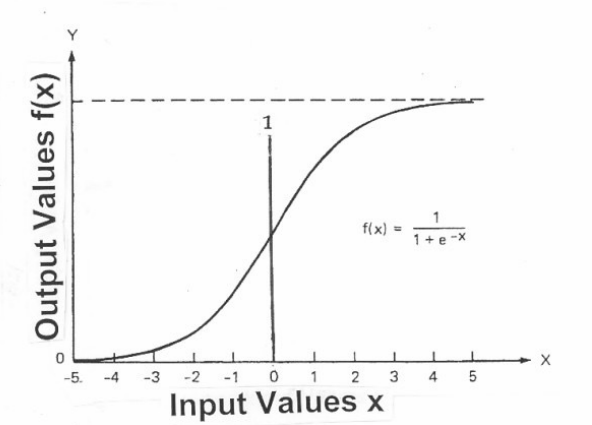
Linear function

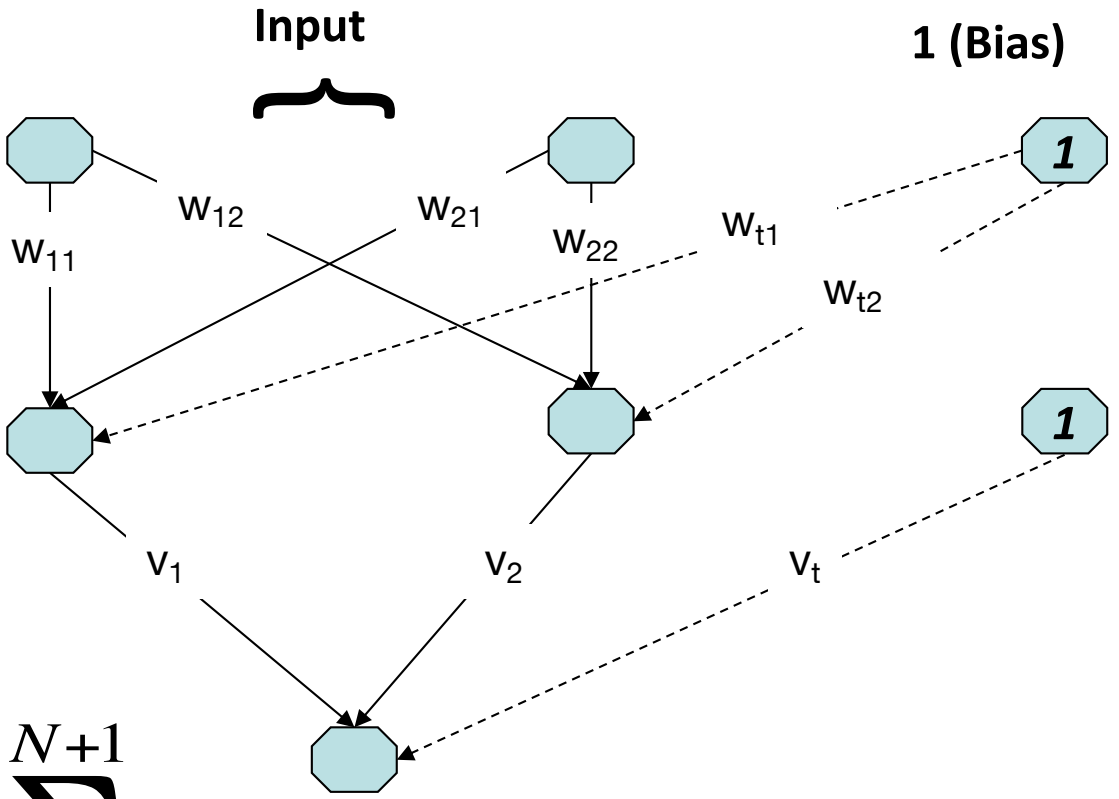
$$y = x_1 \cdot v_1 + x_2 \cdot v_2$$





Neural networks with a hidden layer



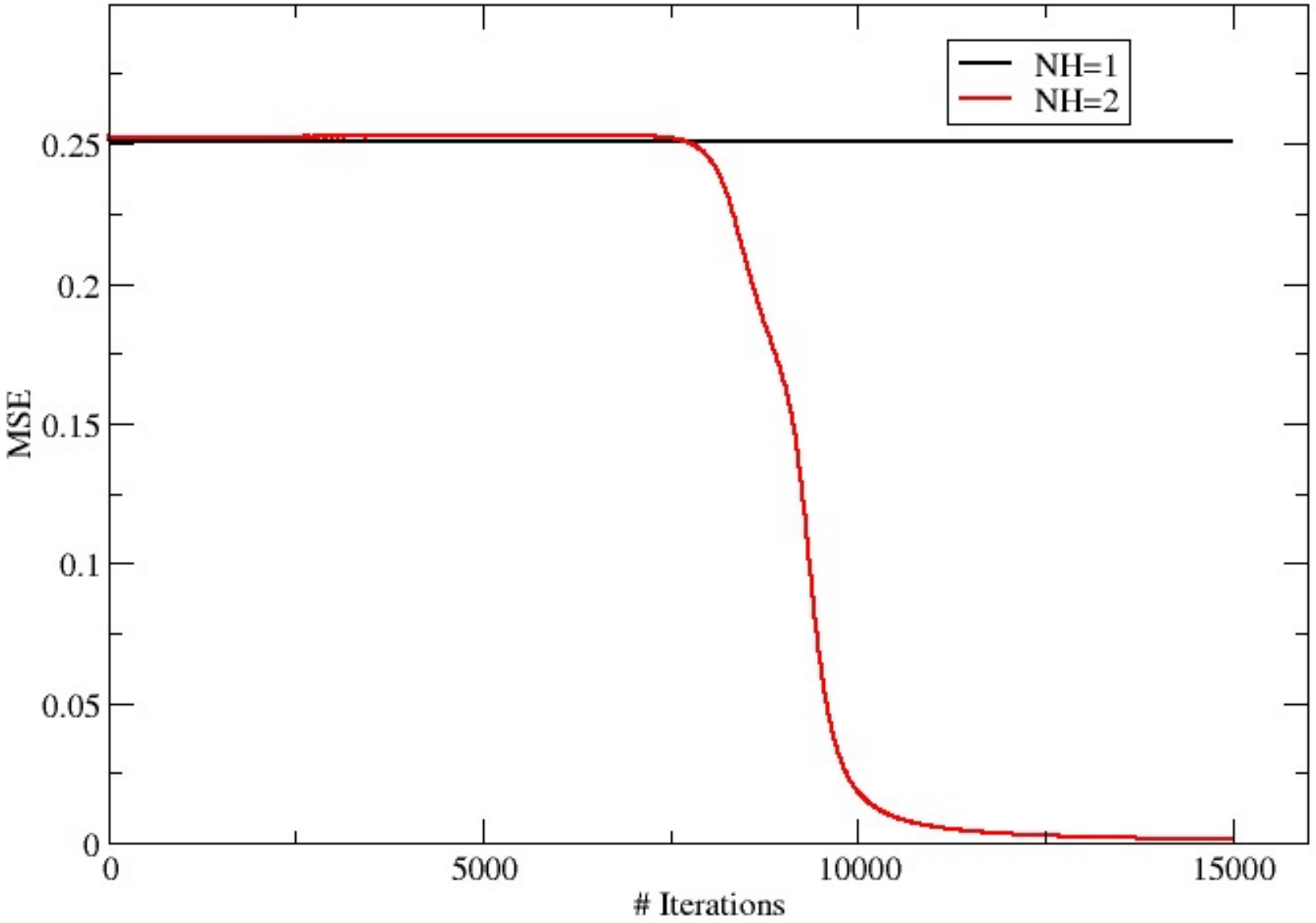


$$O = \frac{1}{1 + \exp(-o)}$$

$$o = \sum_{i=1}^{N} x_i \cdot w_i + t = \sum_{i=1}^{N+1} x_i \cdot w_i$$

$$x_{N+1} = 1 \quad (bias)$$

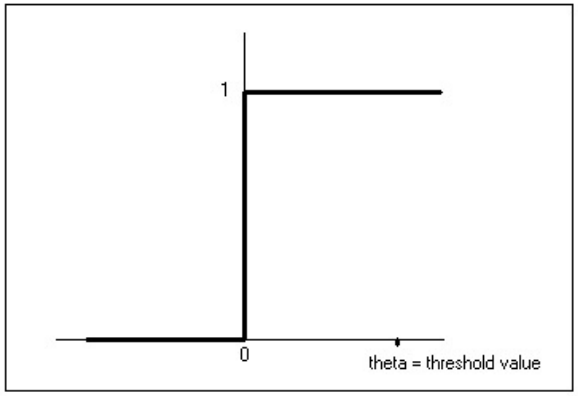
Neural networks

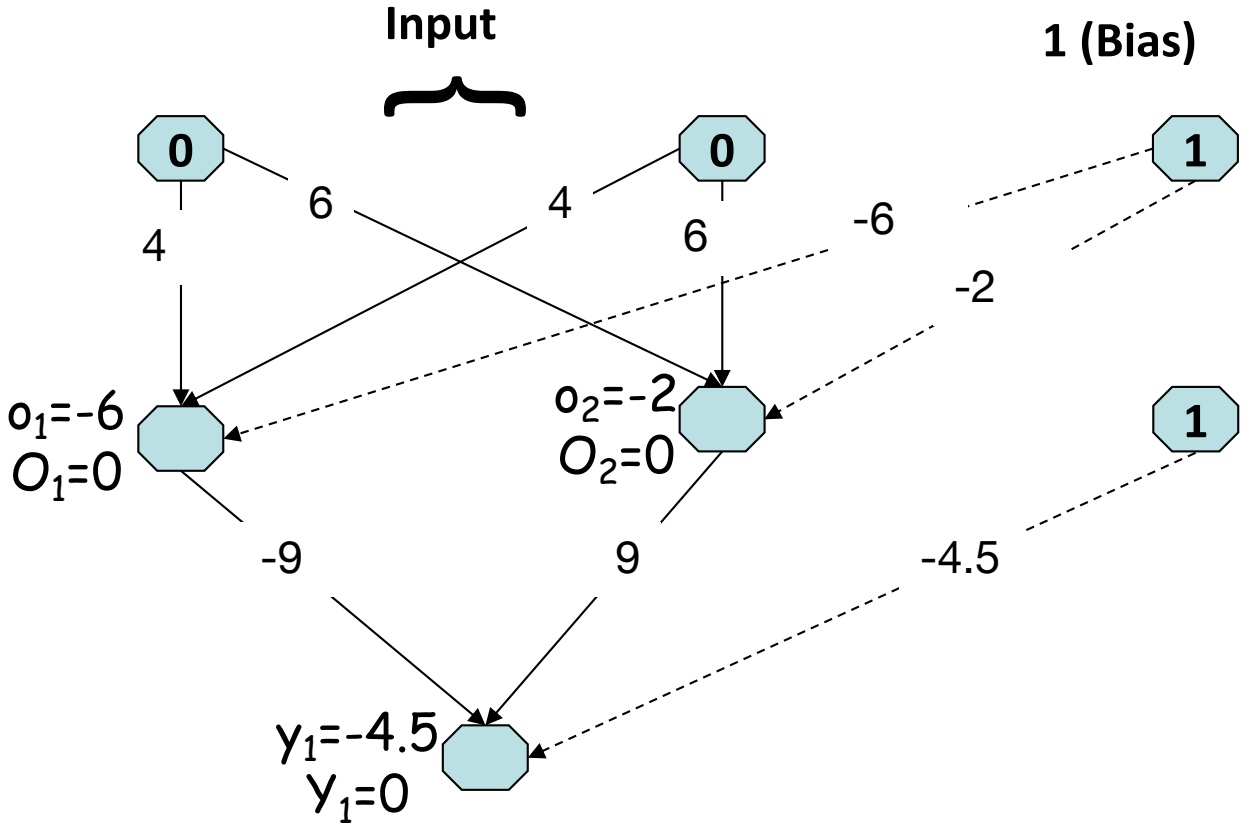


How does it work?
Ex. Input is (0 0)

$$O = \frac{1}{1 + \exp(-o)}$$

or





$$o = \sum x_i \cdot w_i$$

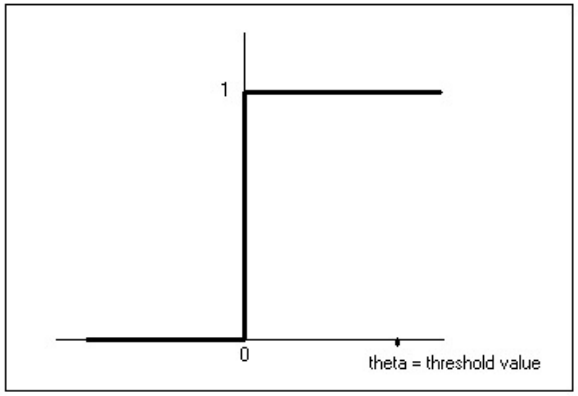
Neural networks. How does it work?

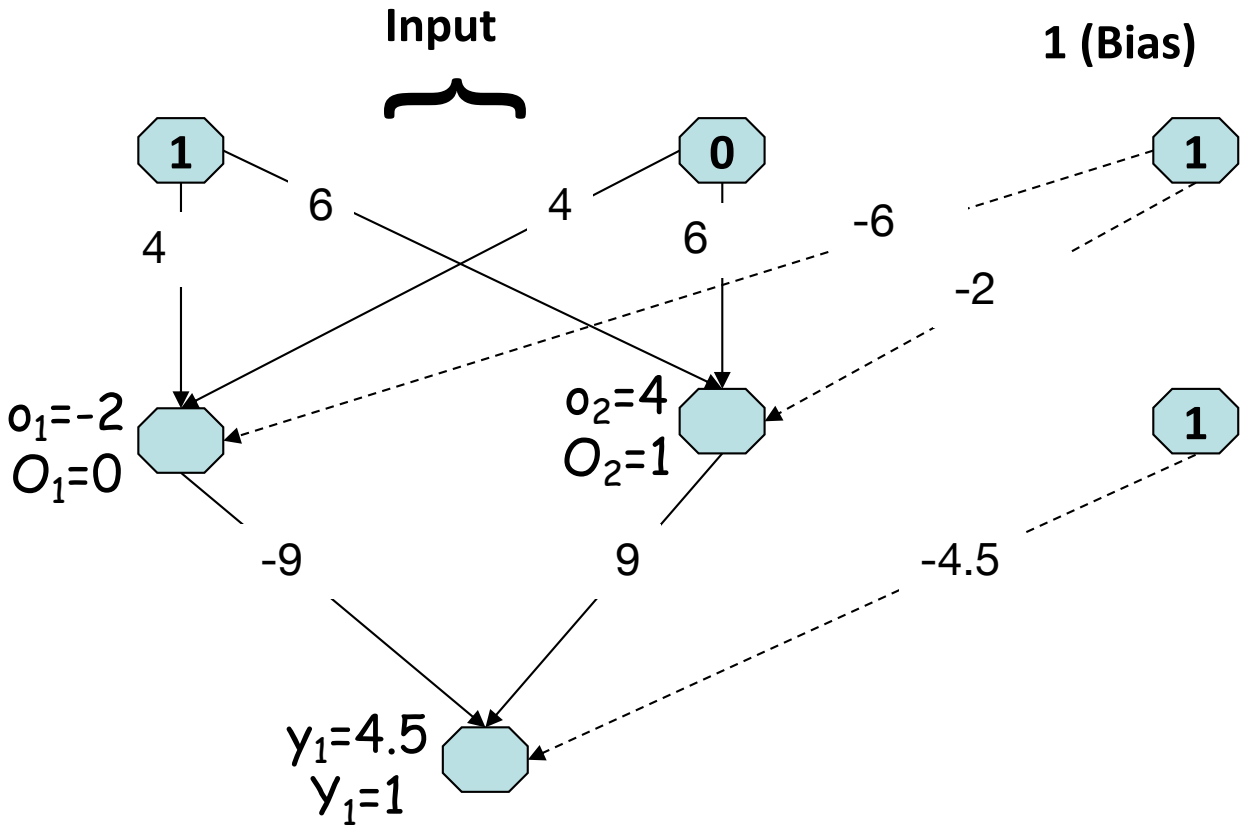
CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

Hand out

Neural networks (1 0 && 0 1)

$$O = \frac{1}{1 + \exp(-o)}$$

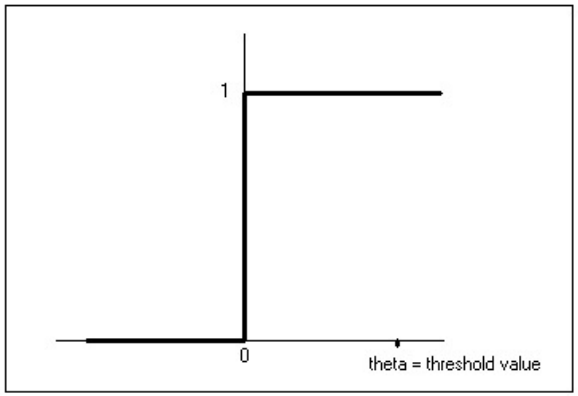




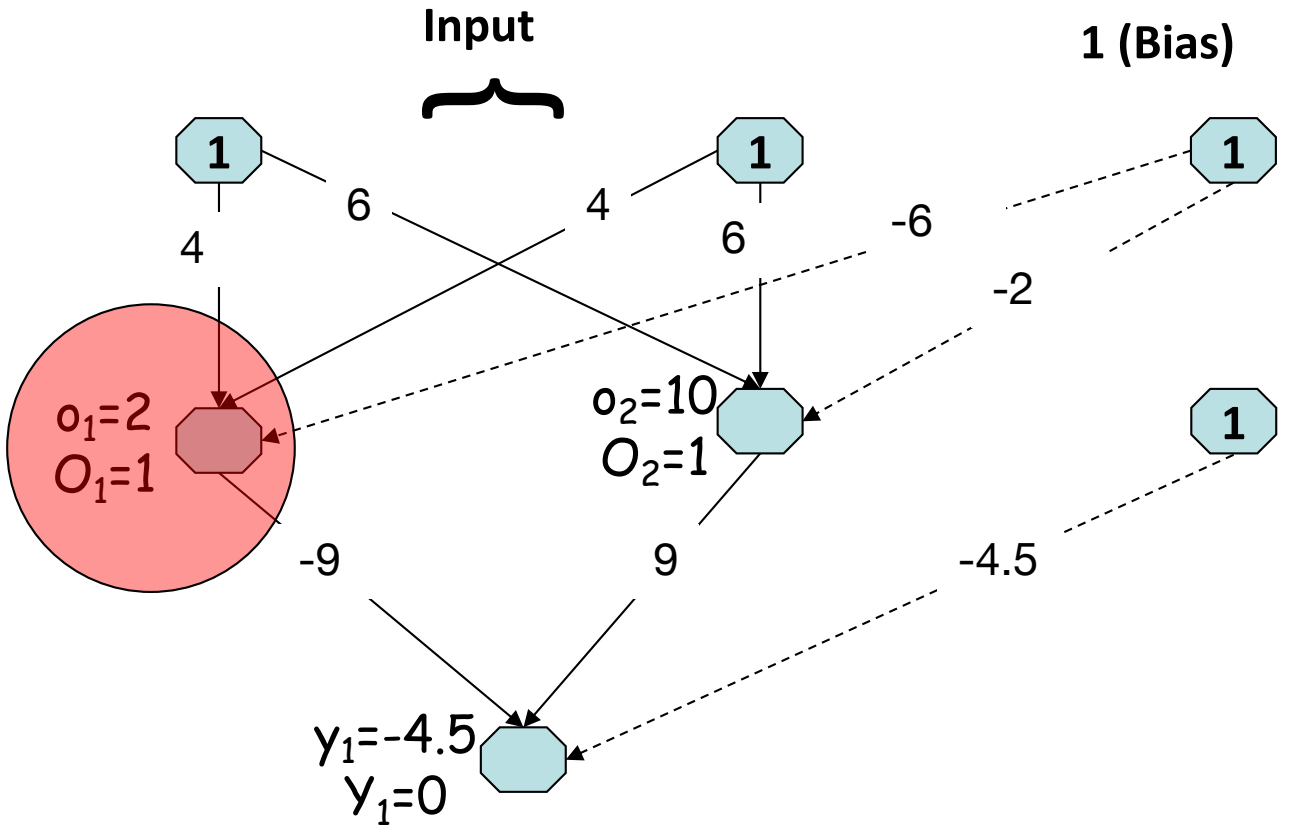
$$o = \sum x_i \cdot w_i$$

Neural networks (1 1)

$$O = \frac{1}{1 + \exp(-o)}$$



$$o = \sum x_i \cdot w_i$$



What is going on?

$$f_{XOR}(x_1,x_2) = -2 \cdot x_1 \cdot x_2 + (x_1 + x_2) = -y_2 + y_1$$

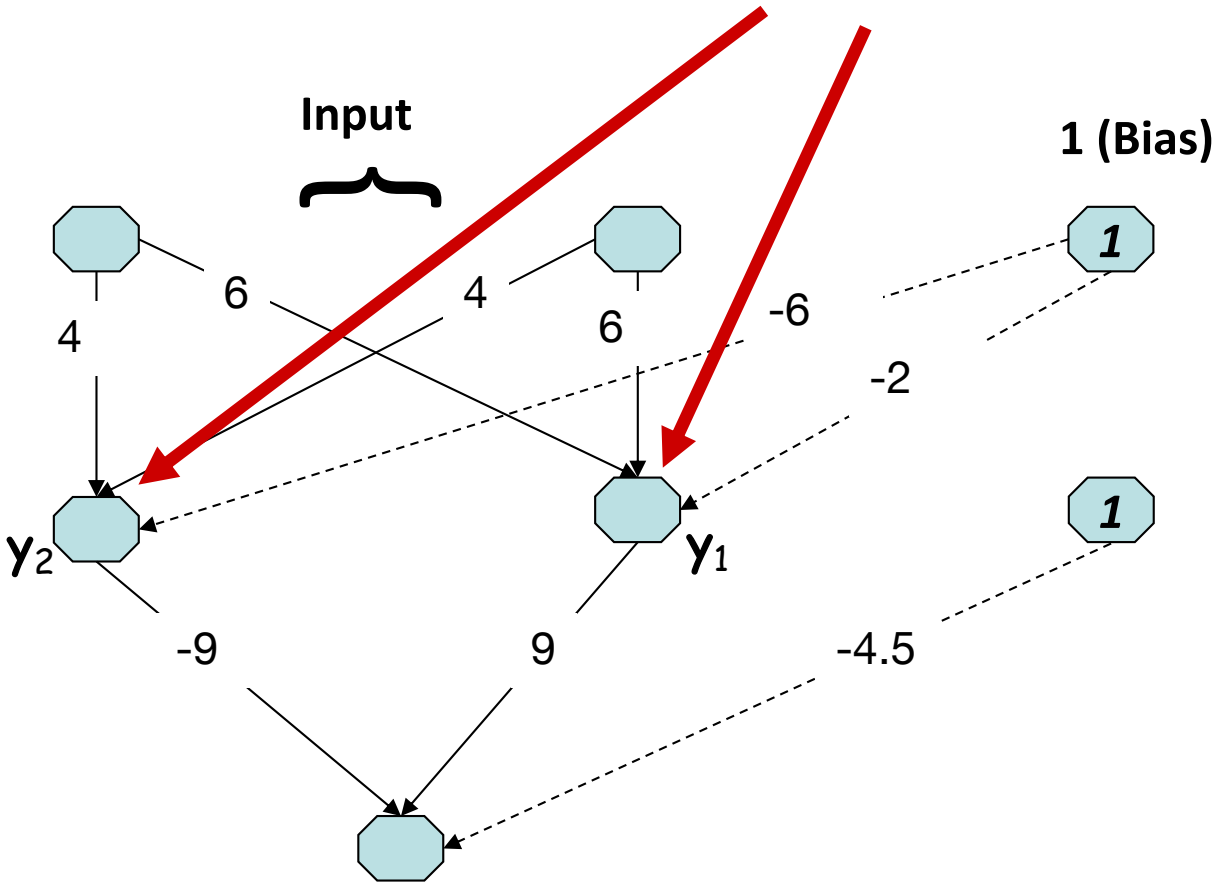
XOR function:

0 0 => 0

1 0 => 1

0 1 => 1

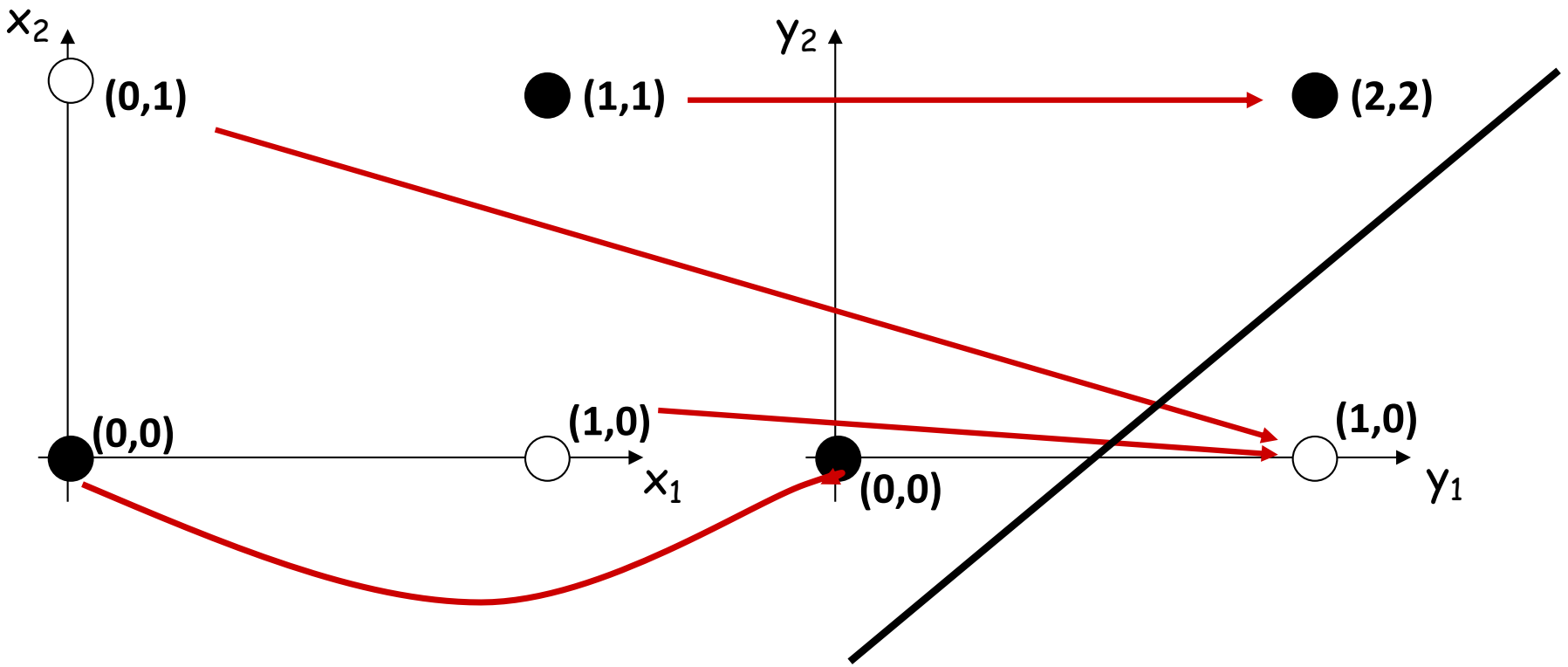
1 1 => 0



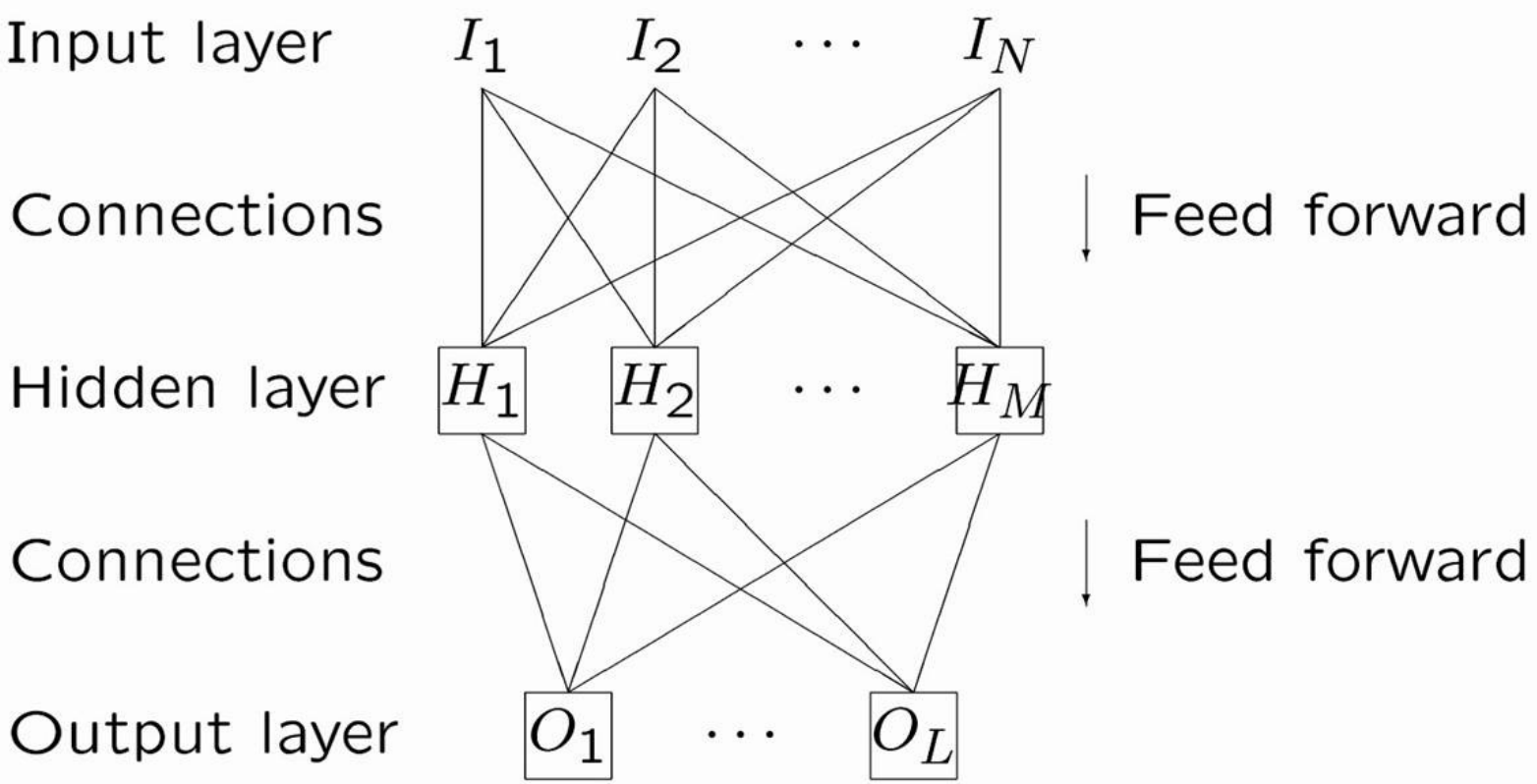
What is going on?

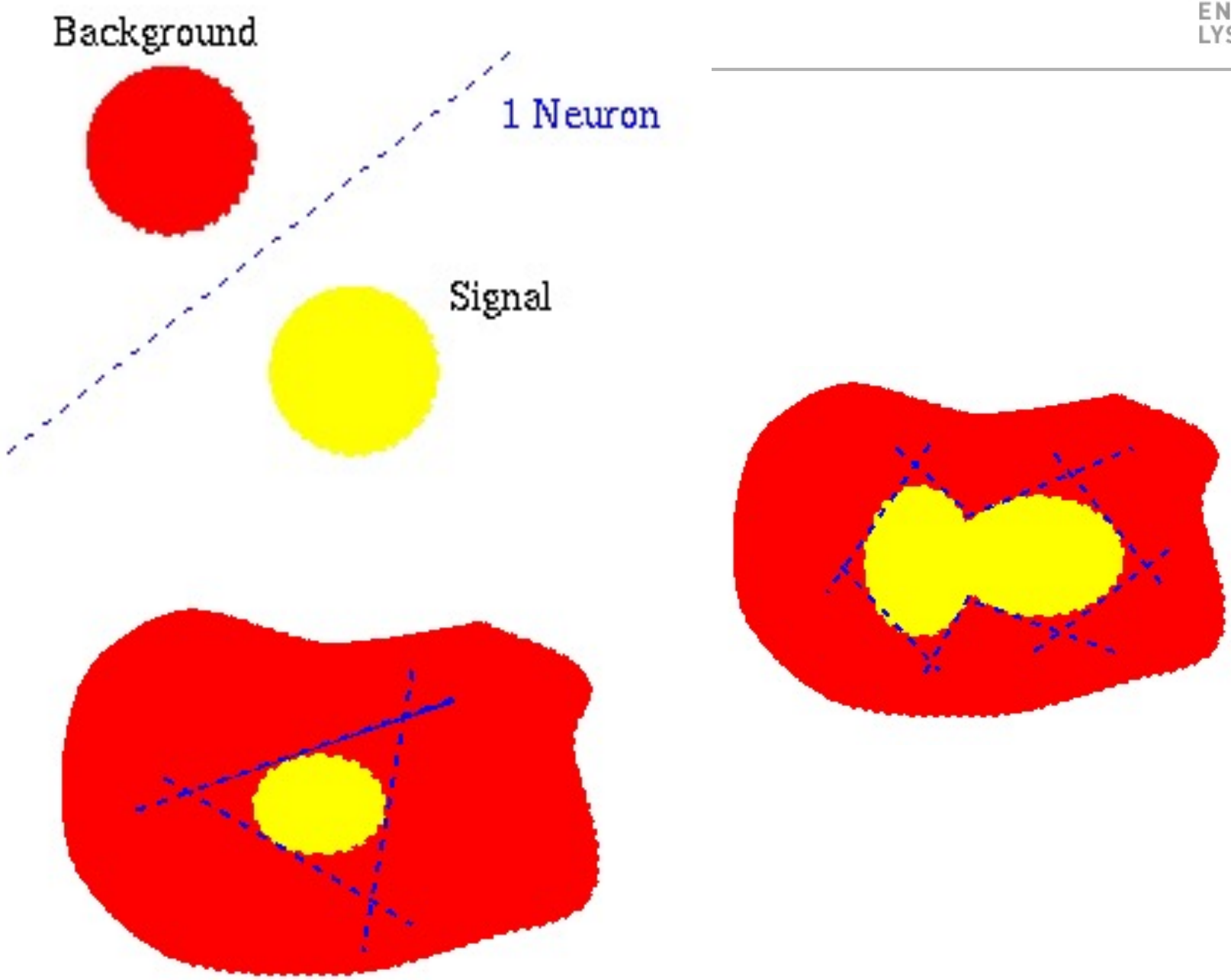
$$y_1 = x_1 + x_2$$

$$y_2 = 2 \cdot x_1 \cdot x_2$$



Network with more inputs and hidden units

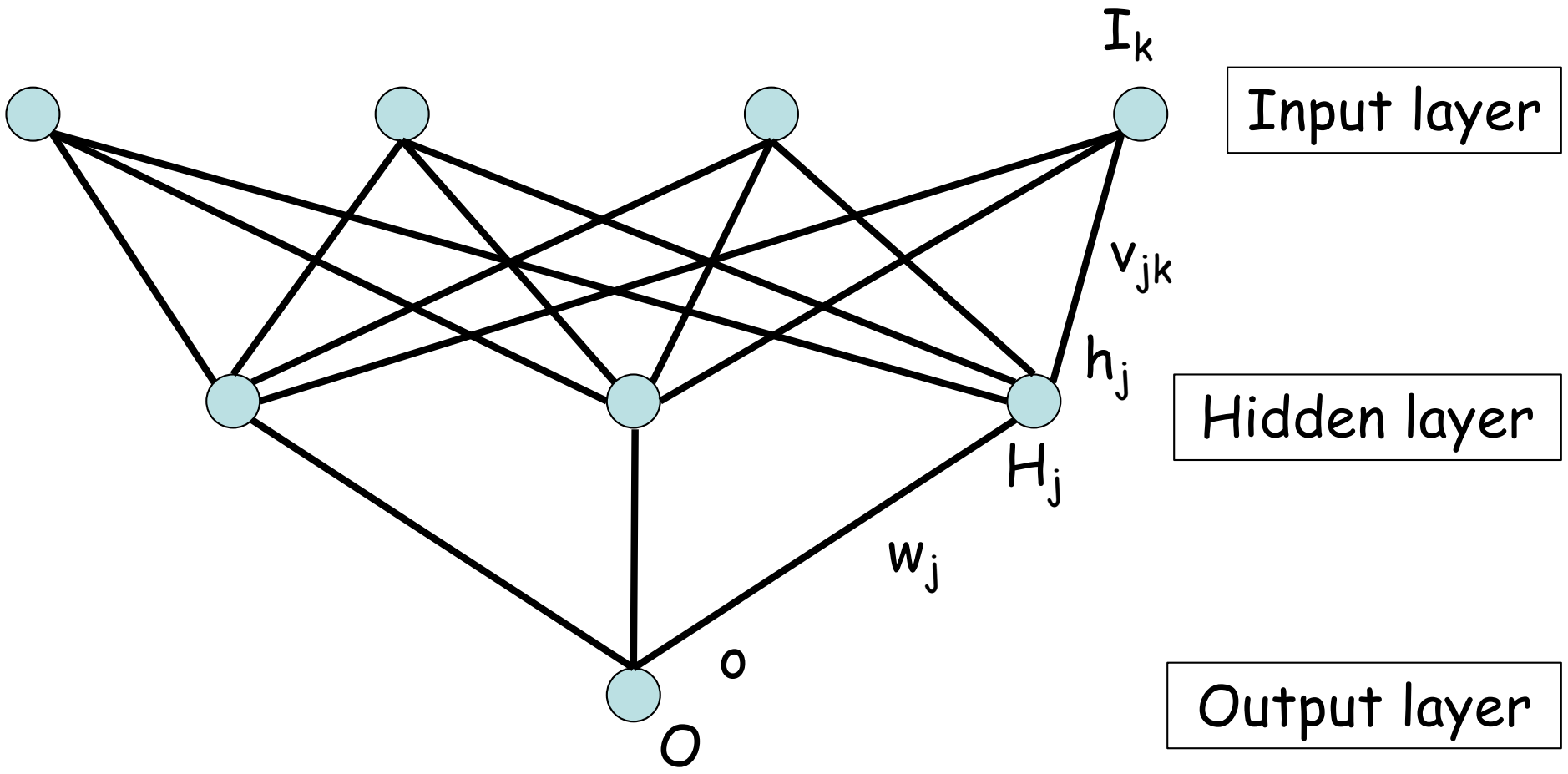



Background
1 Neuron
Signal

Demo

- http://playground.tensorflow.org/

Network architecture



What about the hidden layer?

$$\Delta w_i = -\varepsilon \cdot \frac{\partial E}{\partial w_i}$$

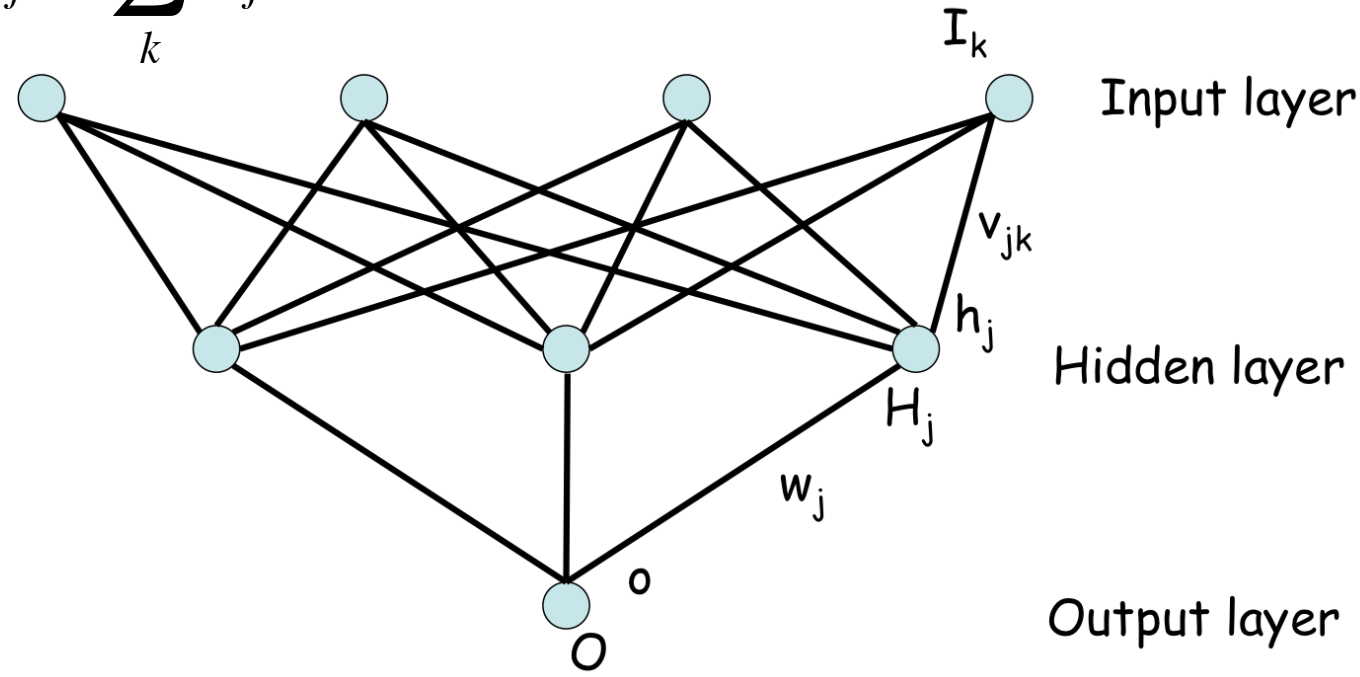
$$o_i = \sum_j w_{ij} \cdot H_j$$

$$h_j = \sum_k v_{jk} \cdot I_k$$

$$E = \tfrac{1}{2} \cdot (O - t)^2$$

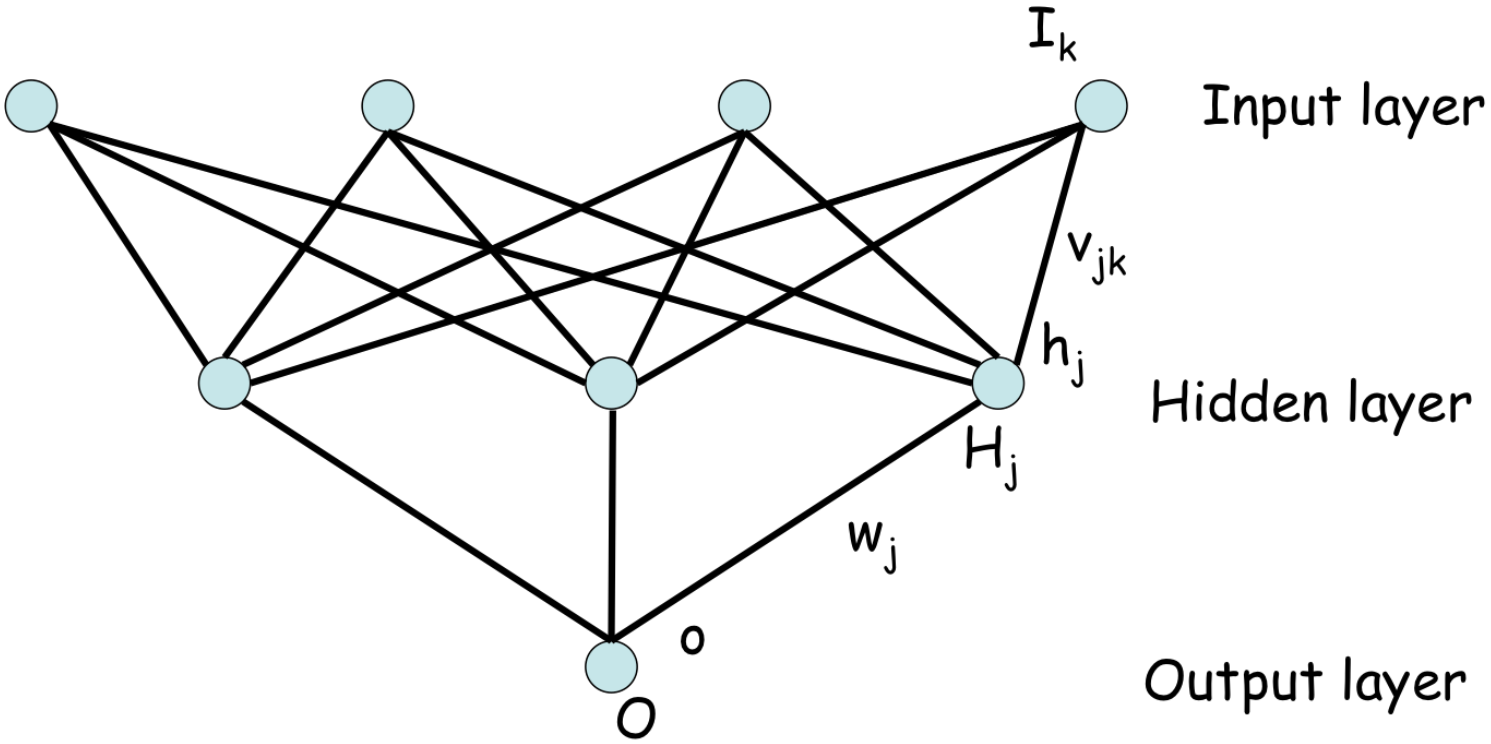
$$O = g(o), H = g(h)$$

$$g(x) = \frac{1}{1 + e^{-x}}$$



Hidden to output layer

$$\frac{\partial E}{\partial w_j} = \frac{\partial E(O(o(w_j)))}{\partial w_j} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial o} \cdot \frac{\partial o}{\partial w_j}$$



Hidden to output layer

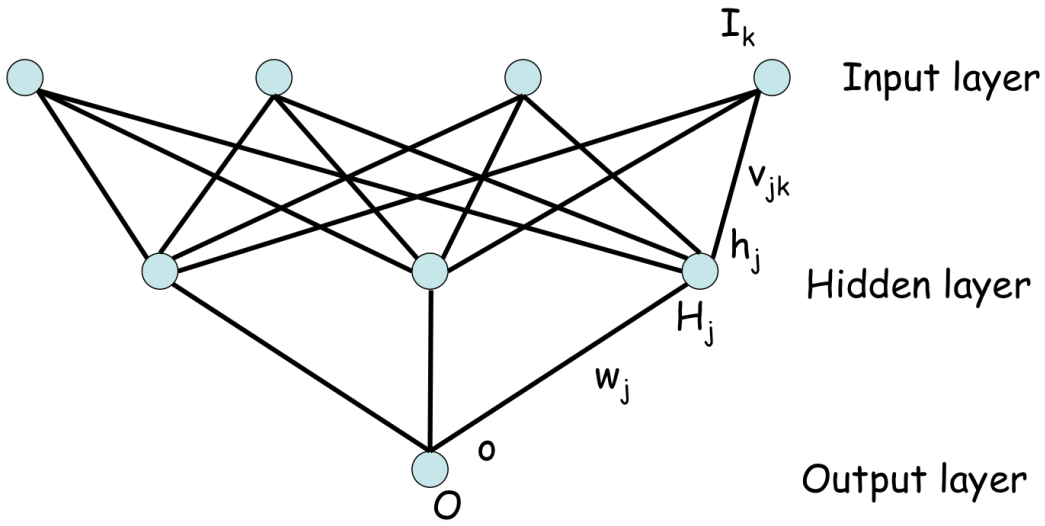
$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial o} \cdot \frac{\partial o}{\partial w_j} = (O - t) \cdot g'(o) \cdot H_j$$

$$= (O - t) \cdot (1 - O) \cdot O \cdot H_j$$

$$O = g(o)$$

$$g'(o) = (1 - g(o)) \cdot g(o)$$

$$= (1 - O) \cdot O$$

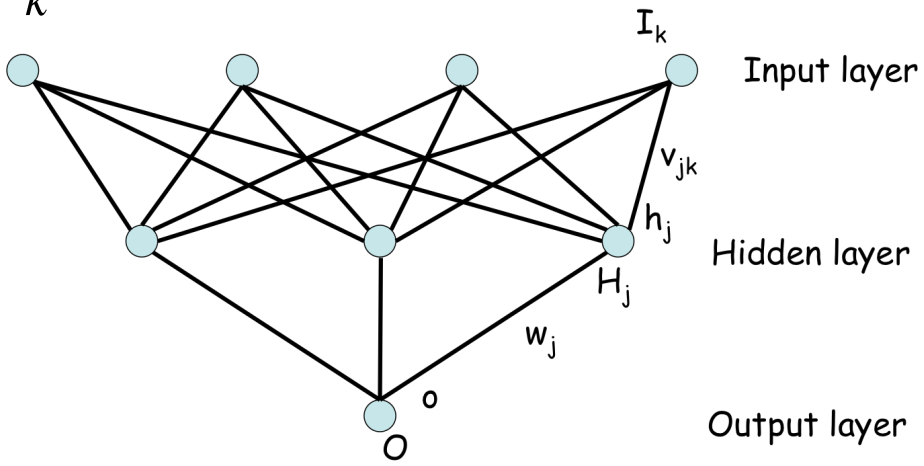


Input to hidden layer

$$\frac{\partial E}{\partial v_{jk}} = \frac{\partial E}{\partial H_j} \cdot \frac{\partial H_j}{\partial v_{jk}} = (O - t) \cdot g'(o) \cdot w_j \cdot g'(h_j) \cdot I_k$$

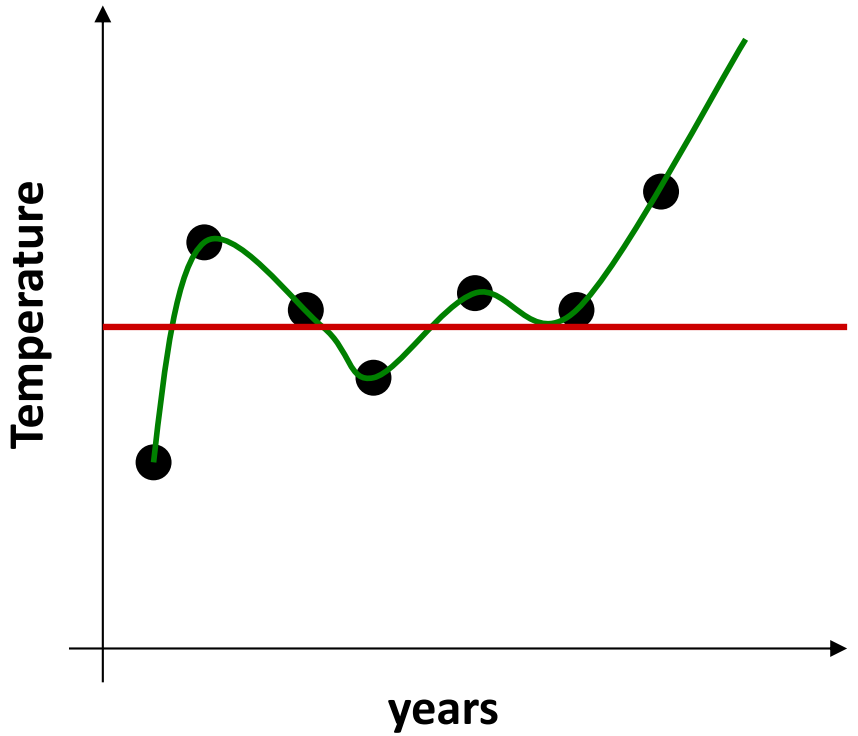
$$\frac{\partial E}{\partial H_j} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial o} \cdot \frac{\partial o}{\partial H_j} = (O - t) \cdot g'(o) \cdot w_j$$

$$\frac{\partial H_j}{\partial v_{jk}} = \frac{\partial H_j}{\partial h_j} \cdot \frac{\partial h_j}{\partial v_{jk}} = g'(h_j) \cdot I_k$$

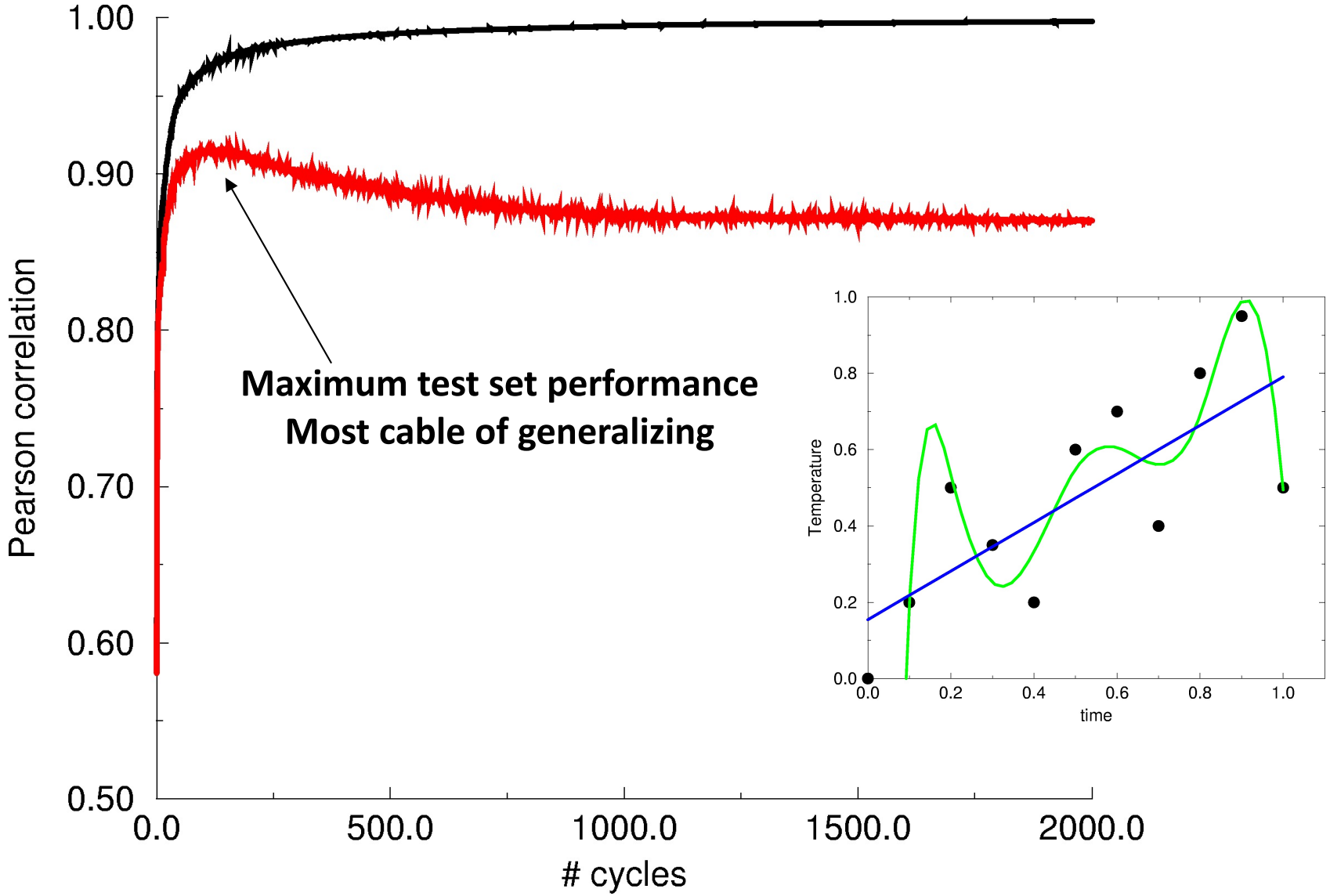


Neural network training (early stopping)

- A Network contains a very large set of parameters
  - A network with 5 hidden neurons predicting binding for 9meric peptides has 9x20x5=900 weights
  - 5 times as many weights as a matrix-based method
- Over fitting is a problem
  - The method can learn noise
- Stop training when test performance is optimal (use early stopping)



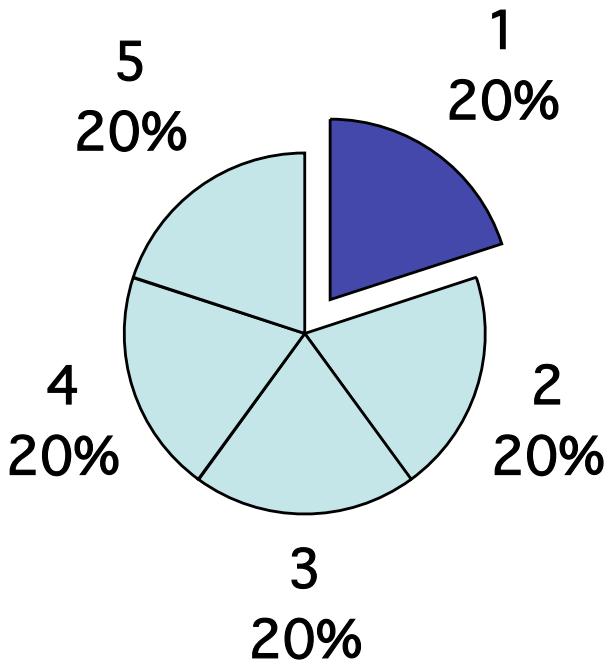
Neural network training curve



Neural network training. Cross validation

Cross validation

Train on 4/5 of data
Test on 1/5
=>
Produce 5 different neural networks each with a different prediction focus



Network training

- Encoding of sequence data
  - Sparse encoding
  - Blosum encoding
  - Sequence profile encoding

Sparse (or one hot) encoding

Inp Neuron	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
AAcid																				
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

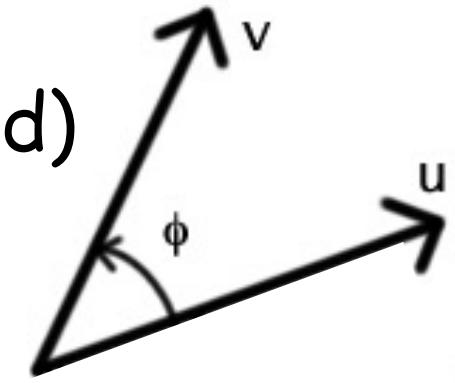
BLOSUM encoding (Blosum50 matrix)

CENTERFO
RBIOLOGI
CALSEQU
ENCEANA
LYSIS CBS

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

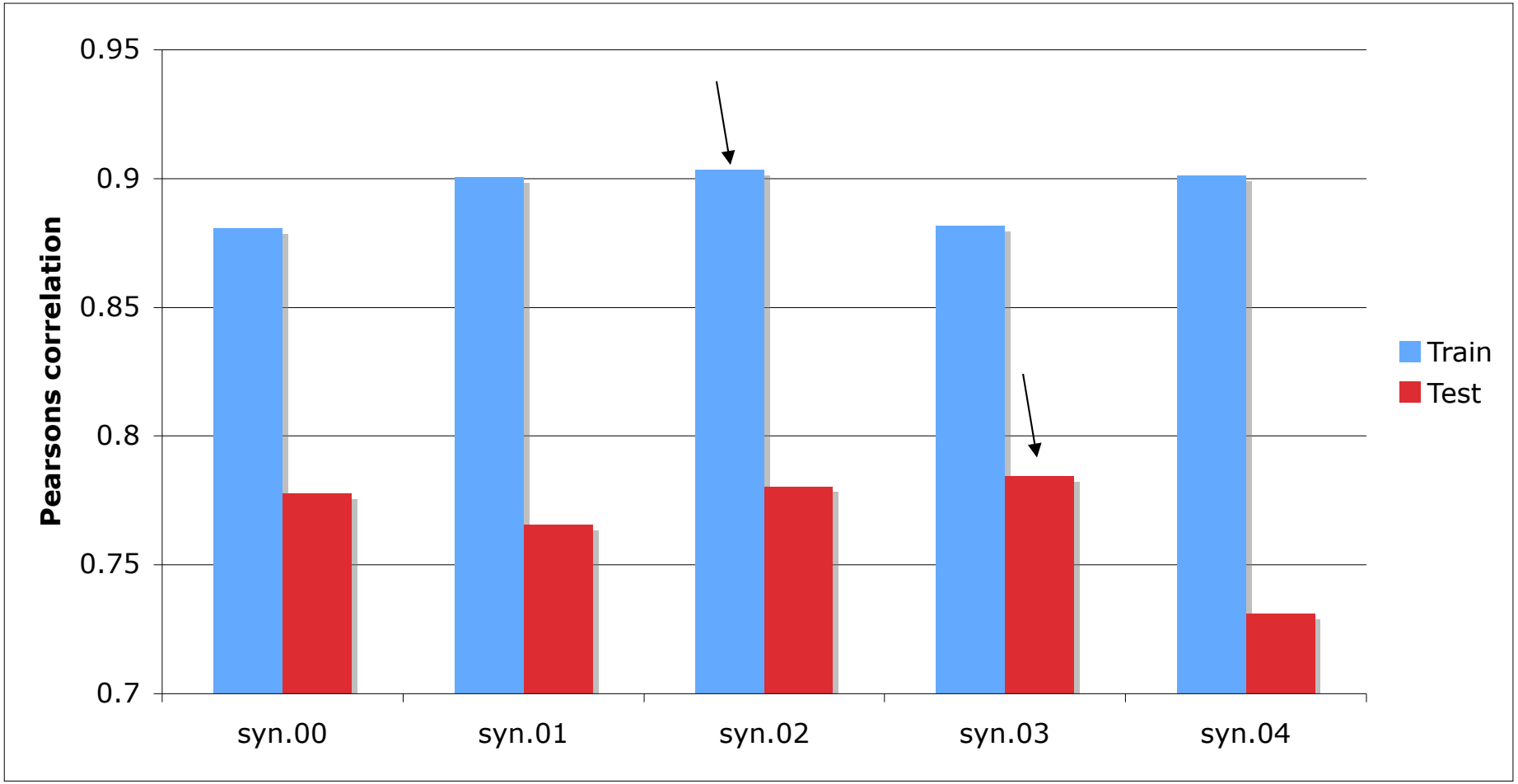
Sequence encoding (continued)

- Sparse encoding
  - V:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
  - L:0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
  - V·L=0 (unrelated)
- Blosum encoding
  - V: 0 -3 -3 -3 -1 -2 -2 -3 -3 3 1 -2 1 -1 -2 -2 0 -3 -1 4
  - L:-1 -2 -3 -4 -1 -2 -3 -4 -3 2 4 -2 2 0 -3 -2 -1 -2 -1 1
  - V·L = 0.88 (highly related)
  - V·R = -0.08 (close to unrelated)

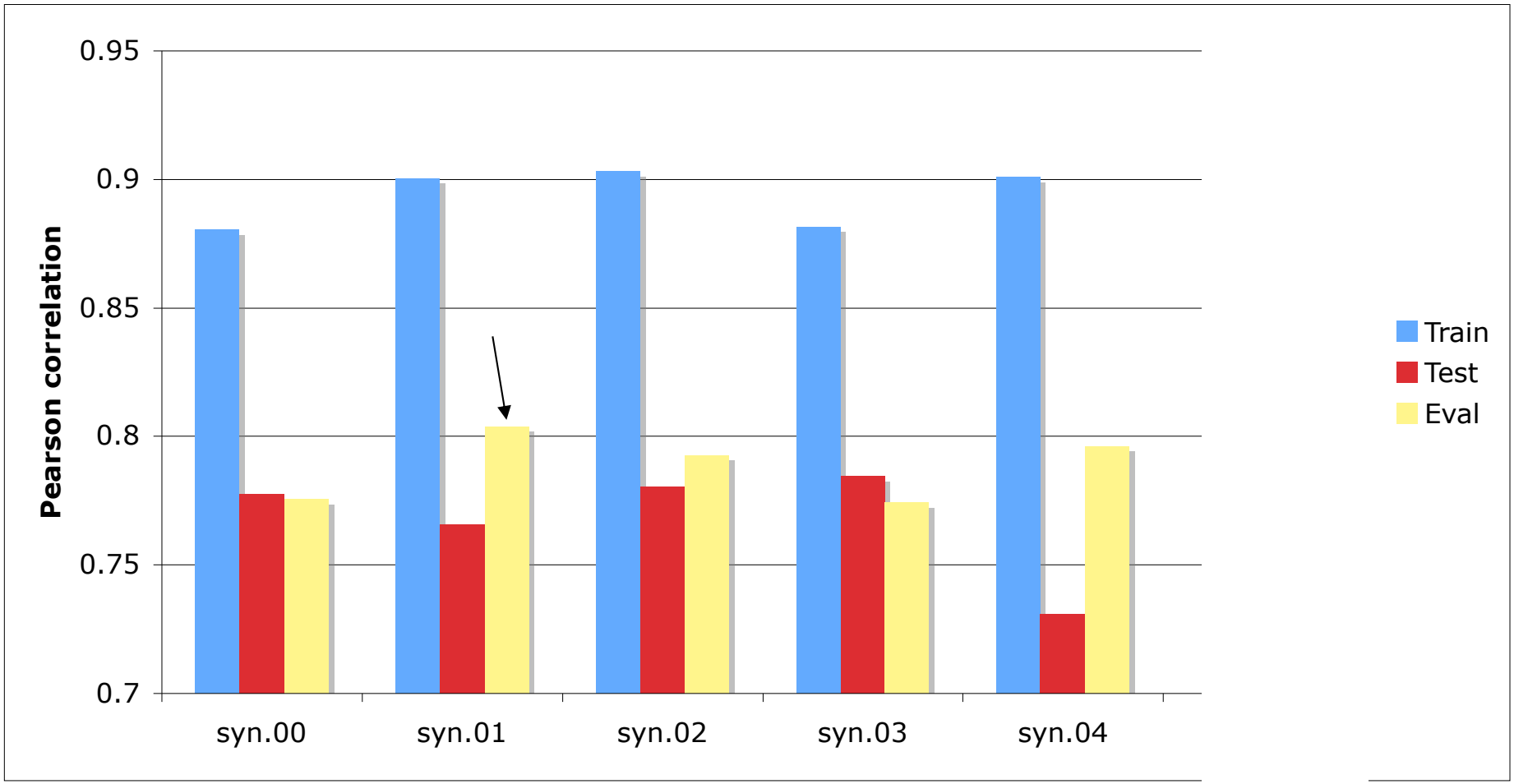


5 fold training

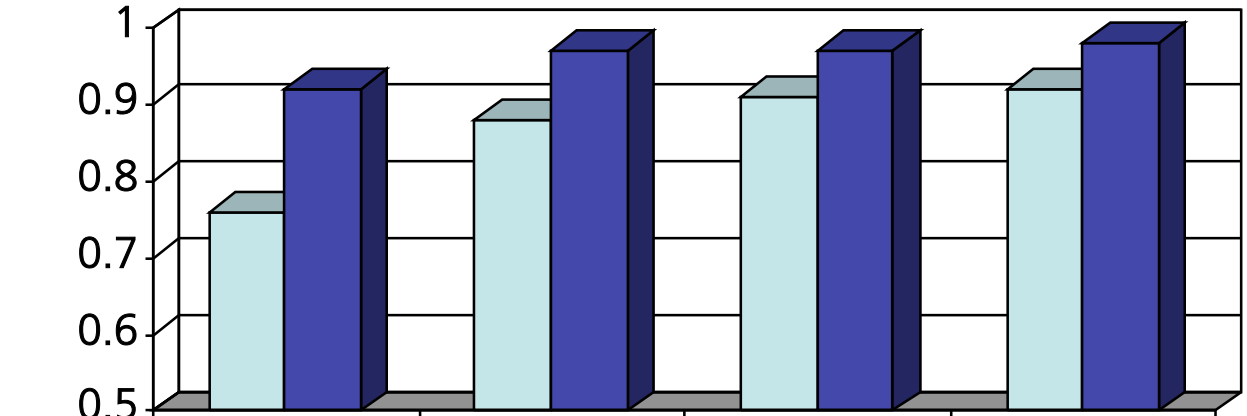
Which network to choose?



5 fold training



Windom of the growd



	Motif	Sparse	BLOSUM	ENS
Pear	0.76	0.88	0.91	0.92
Aroc	0.92	0.97	0.97	0.98

ENS: Ensemble of neural networks trained using sparse, Blosum, and weight matrix sequence encoding

Applications of artificial neural networks

- Talk recognition
- Prediction of protein secondary structure
- Prediction of Signal peptides
- Post translation modifications
  - Glycosylation
  - Phosphorylation
- Proteasomal cleavage
- MHC:peptide binding

What have we learned?

- Neural networks are not so bad as their reputation
- Neural networks can deal with higher order correlations
- Be careful when training a neural network
  - Overfitting is a critical issue
  - Always use cross validated training to assess performance