

# Algoritmus, co vyhrál milión \$

Pavel Kordík

*Skupina výpočetní inteligence FEL a  
FIT ČVUT*



Computational Intelligence Group,  
Czech Technical University in Prague,  
Czech Republic

<http://cig.felk.cvut.cz>

# Anotace

- Během prezentace se, kromě toho, jaký že "algoritmus" to vyhrál v loňském roce Netflix prize, dozvíte o aktuálním výzkumu v oblastech neuroevoluce, metaučení, vytěžování znalostí z dat a optimalizace. Podrobně se budeme věnovat zejména projektu FAKE GAME ([fakegame.sourceforge.net](https://fakegame.sourceforge.net)), který pomocí metaučení automatizuje procesy předzpracování dat, modelování a extrakce znalostí.

# Netflix Prize

- Vyroberte mi klasifikátor, který je o 10% lepší než to co mám, a dostanete milion dolarů!
- <http://www.netflixprize.com/>
- Učení s učitelem
  - Trénovací data jsou množiny respondentů a jejich hodnocení (1,2,3,4,5 hvězdiček), které udělili jednotlivým filmům.
  - Vytvoř klasifikátor, který dostane na vstup identifikaci uživatele a jím ještě nehodnocený film a vyprodukuje počet hvězdiček.

# A kdo tedy vyhrál?

## Netflix Prize

[Home](#) [Rules](#) [Leaderboard](#) [Register](#) [Update](#) [Submit](#) [Download](#)

## Leaderboard

Display top  leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	<a href="#">The Ensemble</a>	0.8553	10.10	2009-07-26 18:38:22
2	<a href="#">BellKor's Pragmatic Chaos</a>	0.8554	10.09	2009-07-26 18:18:28
<b>Grand Prize - RMSE <math>\leq</math> 0.8563</b>				
3	<a href="#">Grand Prize Team</a>	0.8571	9.91	2009-07-24 13:07:49
4	<a href="#">Opera Solutions and Vandelay United</a>	0.8573	9.89	2009-07-25 20:05:52
5	<a href="#">Vandelay Industries I</a>	0.8579	9.83	2009-07-26 02:49:53
6	<a href="#">PragmaticTheory</a>	0.8582	9.80	2009-07-12 15:09:53
7	<a href="#">BellKor in BigChaos</a>	0.8590	9.71	2009-07-26 12:57:25
8	<a href="#">Dace</a>	0.8603	9.58	2009-07-24 17:18:43
9	<a href="#">Opera Solutions</a>	0.8611	9.49	2009-07-26 18:02:08
10	<a href="#">BellKor</a>	0.8612	9.48	2009-07-26 17:19:11
11	<a href="#">BigChaos</a>	0.8613	9.47	2009-06-23 23:06:52
12	<a href="#">Feeds2</a>	0.8613	9.47	2009-07-24 20:06:46
<b>Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos</b>				
13	<a href="#">xiangliang</a>	0.8633	9.26	2009-07-21 02:04:40
14	<a href="#">Gravity</a>	0.8634	9.25	2009-07-26 15:58:34
15	<a href="#">Ces</a>	0.8642	9.17	2009-07-25 17:42:38

# The Ensemble

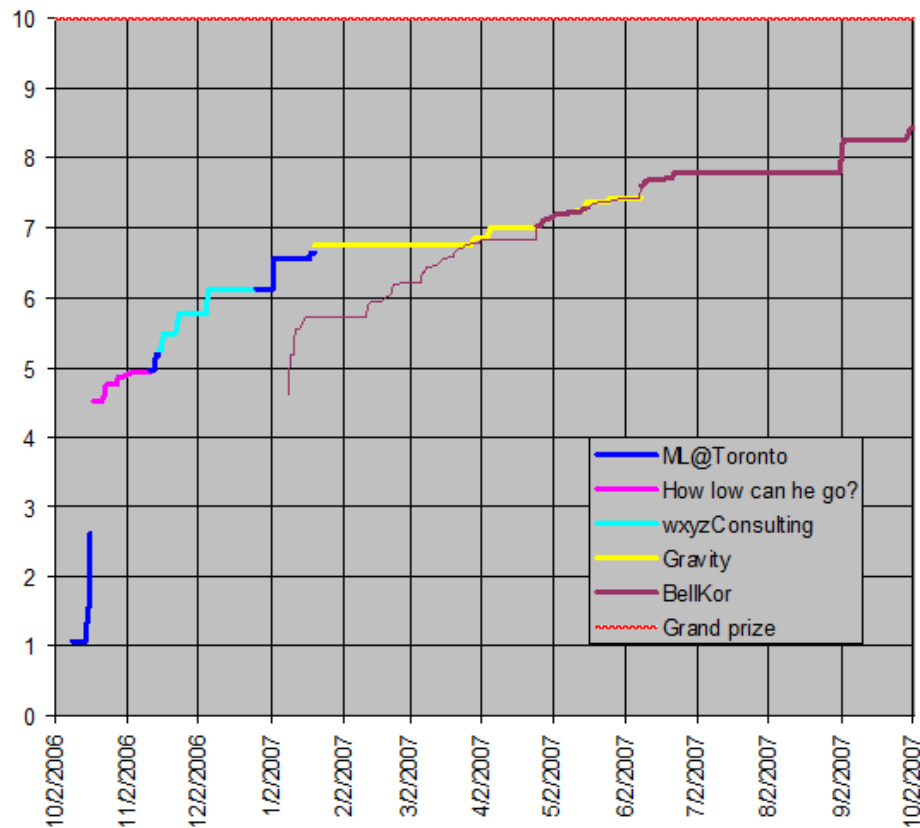
- And lo, as if powered by gravity, [Grand Prize Team](#) and [Vandelay Industries](#) ! began to draw in more and more members. And Vandelay went on to join forces with [Opera Solutions](#), and then Vandelay and Opera united with Grand Prize Team, and then ... and then ... well, things got so complex we decided just to call ourselves **The Ensemble**.
- <http://www.the-ensemble.com/content/meet-team>
- Ale i druhý vítěz BellKor's Pragmatic Chaos je vlastně ensemble!

# Takhle to vypadalo v květnu

Rank	Team Name	Best Score	% Improvement	Last Submit Time
--	No Grand Prize candidates yet	--	--	--
<b>Grand Prize - RMSE <math>\leq</math> 0.8563</b>				
1	<a href="#">BellKor in BigChaos</a>	0.8590	9.71	2009-05-13 08:14:09
2	<a href="#">PragmaticTheory</a>	0.8595	9.66	2009-05-19 13:23:12
3	<a href="#">Grand Prize Team</a>	0.8597	9.64	2009-05-08 14:10:33
4	<a href="#">Dace</a>	0.8604	9.56	2009-04-22 05:57:03
5	<a href="#">BigChaos</a>	0.8613	9.47	2009-05-09 21:06:09
<b>Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos</b>				
6	<a href="#">BellKor</a>	0.8621	9.39	2009-05-10 09:39:48
7	<a href="#">Gravity</a>	0.8634	9.25	2009-04-22 18:31:32
8	<a href="#">Opera Solutions</a>	0.8642	9.17	2009-05-18 05:53:59
9	<a href="#">Ces</a>	0.8642	9.17	2009-05-14 02:13:41
10	majia2	0.8642	9.17	2009-05-18 09:29:55
11	<a href="#">BruceDengDaoCiYiYou</a>	0.8642	9.17	2009-05-18 09:59:21
12	<a href="#">Feeds2</a>	0.8649	9.09	2009-05-17 10:47:21
13	<a href="#">Just a guy in a garage</a>	0.8651	9.07	2009-05-19 18:13:50
14	<a href="#">pengpengzhou</a>	0.8654	9.04	2009-05-05 18:18:03
15	NewNetflixTeam	0.8657	9.01	2009-05-13 01:01:36
16	<a href="#">J Dennis Su</a>	0.8658	9.00	2009-03-11 09:41:54
17	Vandelay Industries !	0.8658	9.00	2009-05-11 00:43:14
18	<a href="#">acmehill</a>	0.8659	8.99	2009-04-16 06:29:35
19	MonteCarlo	0.8661	8.97	2009-03-25 15:00:05
20	IDEA2	0.8661	8.97	2009-03-25 15:37:59
21	Newman, George, and Peterman !	0.8665	8.92	2009-05-07 05:04:09
22	xiangliang	0.8665	8.92	2009-05-19 14:19:10
23	Team ESP	0.8666	8.91	2009-05-19 14:33:15
24	My Brain and His Chain	0.8668	8.89	2008-09-30 02:19:47
25	<a href="#">xlvector</a>	0.8668	8.89	2009-05-19 07:33:38

# Vývoj soutěže

Top contenders for Progress Prize 2007



	No Progress Prize candidates yet		
<b>Progress Prize - RMSE &lt;= 0.8625</b>			
1	<a href="#">BellKor</a>	0.8705	8.50
<b>Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell</b>			
2	<a href="#">KorBell</a>	0.8712	8.43
3	<a href="#">When Gravity and Dinosaurs Unite</a>	0.8717	8.38
4	<a href="#">Gravity</a>	0.8743	8.10
5	<a href="#">basho</a>	0.8746	8.07
6	<a href="#">Dinosaur Planet</a>	0.8753	8.00
7	<a href="#">ML@UToronto A</a>	0.8787	7.64
8	<a href="#">Arek Paterek</a>	0.8789	7.62
9	<a href="#">NIPS Reject</a>	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	<a href="#">Ensemble Experts</a>	0.8841	7.07
12	<a href="#">mathematical capital</a>	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	<a href="#">The Thought Gang</a>	0.8849	6.99
15	<a href="#">Reel Ingenuity</a>	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	<a href="#">Rookies</a>	0.8872	6.75
22	<a href="#">Paul Harrison</a>	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	<a href="#">wxyzconsulting.com</a>	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	<a href="#">Kitty</a>	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	<a href="#">Birgit Kraft</a>	0.8885	6.61

# Rookies

“Thanks to Paul Harrison's collaboration, a simple mix of our solutions improved our result from 6.31 to 6.75”



--	No Progress Prize candidates yet	--	--
<b>Progress Prize - RMSE &lt;= 0.8625</b>			
1	<a href="#">BellKor</a>	0.8705	8.50
<b>Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell</b>			
2	<a href="#">KorBell</a>	0.8712	8.43
3	<a href="#">When Gravity and Dinosaurs Unite</a>	0.8717	8.38
4	<a href="#">Gravity</a>	0.8743	8.10
5	<a href="#">basho</a>	0.8746	8.07
6	<a href="#">Dinosaur Planet</a>	0.8753	8.00
7	<a href="#">ML@UToronto A</a>	0.8787	7.64
8	<a href="#">Arek Paterek</a>	0.8789	7.62
9	<a href="#">NIPS Reject</a>	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	<a href="#">Ensemble Experts</a>	0.8841	7.07
12	<a href="#">mathematical capital</a>	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	<a href="#">The Thought Gang</a>	0.8849	6.99
15	<a href="#">Reel Ingenuity</a>	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	<a href="#">Rookies</a>	0.8872	6.75
22	<a href="#">Paul Harrison</a>	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	<a href="#">wyzconsulting.com</a>	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	<a href="#">Kitty</a>	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	<a href="#">Birgit Kraft</a>	0.8885	6.61



# Arek Paterek

“My approach is to **combine the results of many methods** (also two-way interactions between them) using linear regression on the test set. The best method in my ensemble is regularized SVD with biases, post processed with kernel ridge regression”

[http://rainbow.mimuw.edu.pl/~ap/ap\\_kdd.pdf](http://rainbow.mimuw.edu.pl/~ap/ap_kdd.pdf)

--	No Progress Prize candidates yet	--	--
<b>Progress Prize - RMSE &lt;= 0.8625</b>			
1	<a href="#">BellKor</a>	0.8705	8.50
<b>Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell</b>			
2	<a href="#">KorBell</a>	0.8712	8.43
3	<a href="#">When Gravity and Dinosaurs Unite</a>	0.8717	8.38
4	<a href="#">Gravity</a>	0.8743	8.10
5	<a href="#">basho</a>	0.8746	8.07
6	<a href="#">Dinosaur Planet</a>	0.8753	8.00
7	<a href="#">ML@UToronto A</a>	0.8787	7.64
8	<a href="#">Arek Paterek</a>	0.8789	7.62
9	<a href="#">NIPS Reject</a>	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	<a href="#">Ensemble Experts</a>	0.8841	7.07
12	<a href="#">mathematical capital</a>	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	<a href="#">The Thought Gang</a>	0.8849	6.99
15	<a href="#">Reel Ingenuity</a>	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	<a href="#">Rookies</a>	0.8872	6.75
22	<a href="#">Paul Harrison</a>	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	<a href="#">wxyzconsulting.com</a>	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	<a href="#">Kitty</a>	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	<a href="#">Birgit Kraft</a>	0.8885	6.61

# U of Toronto

“When the predictions of **multiple** RBM models and **multiple** SVD models are linearly combined, we achieve an error rate that is well over 6% better than the score of Netflix’s own system.”

<http://www.cs.toronto.edu/~rsalakhu/papers/rbmcf.pdf>

--	No Progress Prize candidates yet	--	--
<b>Progress Prize - RMSE &lt;= 0.8625</b>			
1	<a href="#">BellKor</a>	0.8705	8.50
<b>Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell</b>			
2	<a href="#">KorBell</a>	0.8712	8.43
3	<a href="#">When Gravity and Dinosaurs Unite</a>	0.8717	8.38
4	<a href="#">Gravity</a>	0.8743	8.10
5	<a href="#">basho</a>	0.8746	8.07
6	<a href="#">Dinosaur Planet</a>	0.8753	8.00
7	<a href="#">ML@UToronto A</a>	0.8787	7.64
8	<a href="#">Arek Paterek</a>	0.8789	7.62
9	<a href="#">NIPS Reject</a>	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	<a href="#">Ensemble Experts</a>	0.8841	7.07
12	<a href="#">mathematical capital</a>	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	<a href="#">The Thought Gang</a>	0.8849	6.99
15	<a href="#">Reel Ingenuity</a>	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	<a href="#">Rookies</a>	0.8872	6.75
22	<a href="#">Paul Harrison</a>	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	<a href="#">wxyzconsulting.com</a>	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	<a href="#">Kitty</a>	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	<a href="#">Birgit Kraft</a>	0.8885	6.61

A co „tahoun“ celé soutěže?

## BellKor / KorBell

“Our final solution (RMSE=0.8712) consists of blending 107 individual results. “

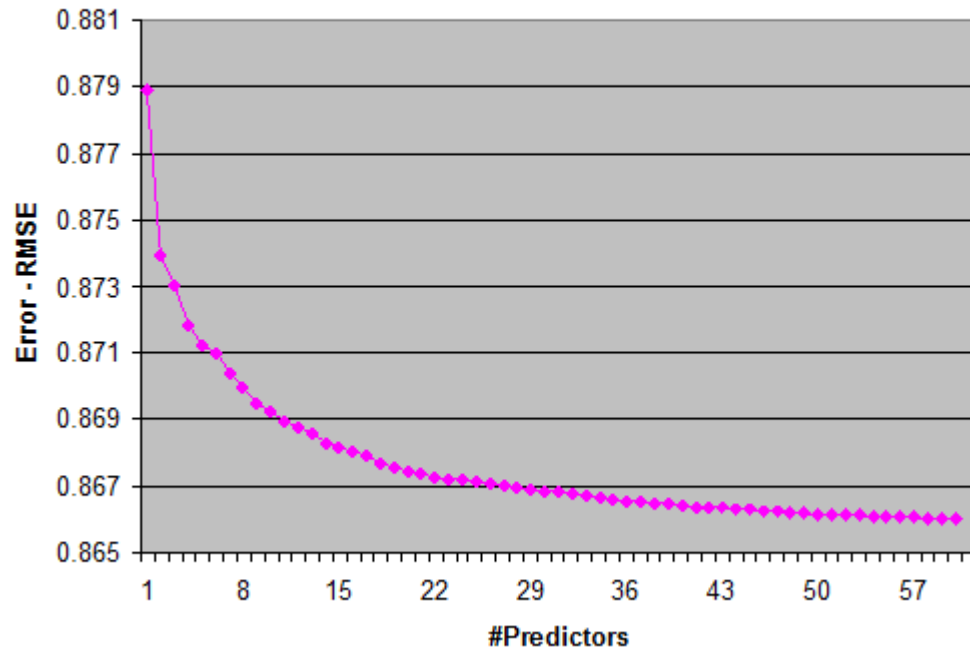
„blendingem“ rozuměj kombinování modelů

--	No Progress Prize candidates yet	--	--
<b>Progress Prize - RMSE &lt;= 0.8625</b>			
1	<a href="#">BellKor</a>	0.8705	8.50
<b>Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell</b>			
2	<a href="#">KorBell</a>	0.8712	8.43
3	<a href="#">When Gravity and Dinosaurs Unite</a>	0.8717	8.38
4	<a href="#">Gravity</a>	0.8743	8.10
5	<a href="#">basho</a>	0.8746	8.07
6	<a href="#">Dinosaur Planet</a>	0.8753	8.00
7	<a href="#">ML@UToronto A</a>	0.8787	7.64
8	<a href="#">Arek Paterek</a>	0.8789	7.62
9	<a href="#">NIPS Reject</a>	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	<a href="#">Ensemble Experts</a>	0.8841	7.07
12	<a href="#">mathematical capital</a>	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	<a href="#">The Thought Gang</a>	0.8849	6.99
15	<a href="#">Reel Ingenuity</a>	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	<a href="#">Rookies</a>	0.8872	6.75
22	<a href="#">Paul Harrison</a>	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	<a href="#">wxyzconsulting.com</a>	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	<a href="#">Kitty</a>	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	<a href="#">Birgit Kraft</a>	0.8885	6.61

# Chyba klesá s počtem modelů

■ BellKor in BigChaos

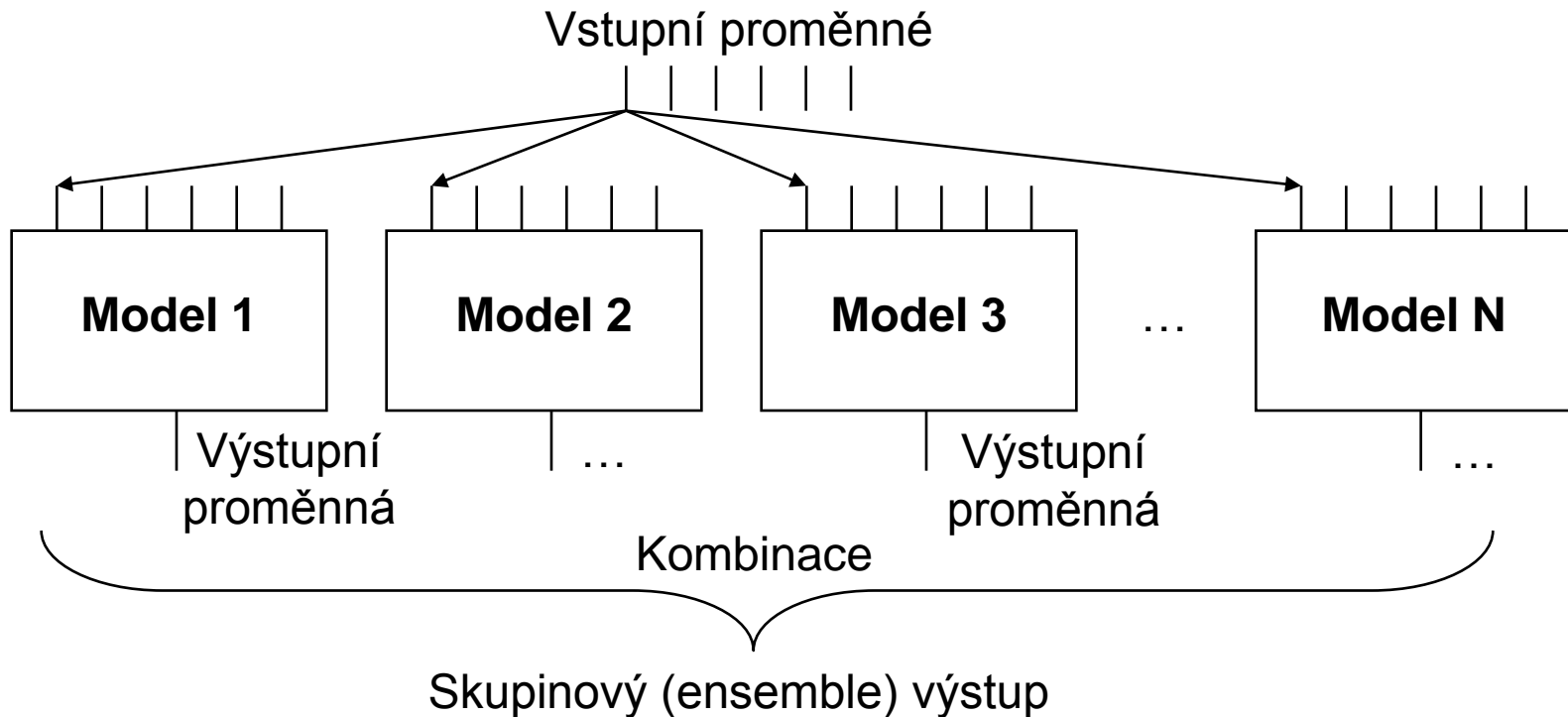
Effect of ensemble size



- Jak a jaké modely kombinovat?

# Princip kombinování modelů

- Skupina modelů (např. rozhodovacích stromů) se naučí na stejný (podobný) úkol.
- Výstupy naučených modelů se kombinují.



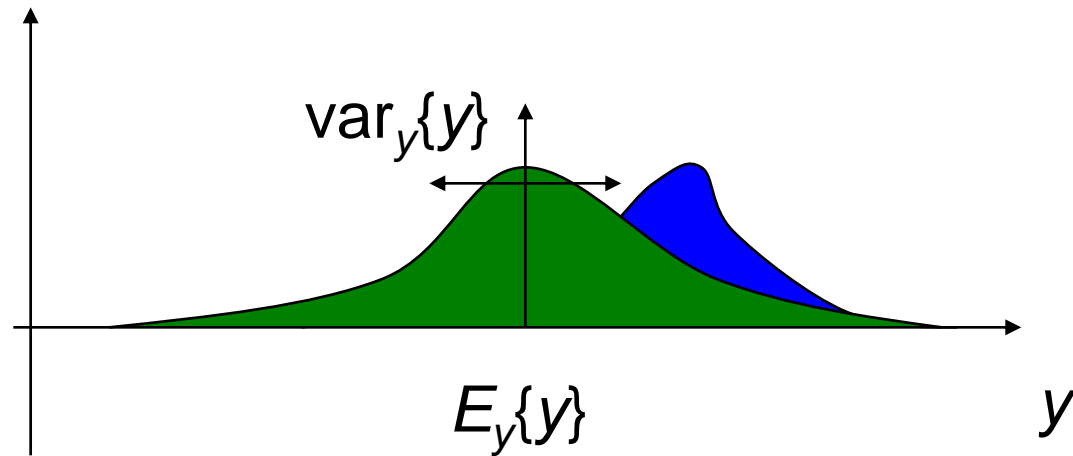
# Různorodost ensemble modelů

- Co se stane, když budou všechny modely totožné? => Degradace na jeden model.
- Jak zajistíme, aby byly modely různorodé?
  - Různé množiny trénovacích dat (počáteční podmínky)
  - Různé metody konstrukce modelů
- Jak se dá měřit různorodost modelů?
  - Odchyly výstupů na jednotlivých testovacích datech.
  - Strukturální odlišnosti

# Funguje to vůbec?

- Chceme zjistit, za jakých předpokladů se vyplatí modely kombinovat.
- Zajímá nás proč ensembling funguje.
- Potřebujeme k tomu analyzovat, čím je chyba modelů způsobena

# Dekompozice bias/variance (1)



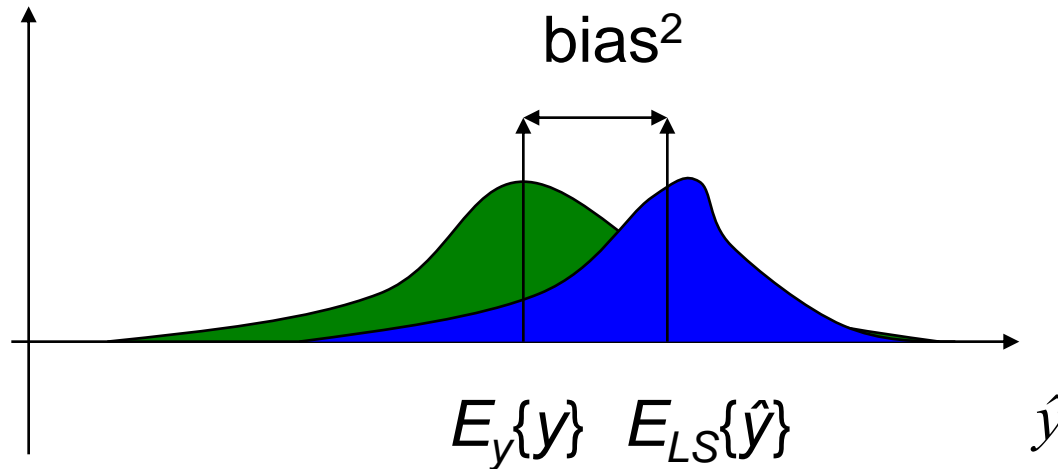
$$E = \underbrace{E_y\{(y - E_y\{y\})^2\}}_{\text{residuální chyba}} + E_{LS}\{(E_y\{y\} - \hat{y})^2\}$$

= residuální chyba = minimální dosažitelná chyba

$$= \text{var}_y\{y\}$$



# Dekompozice bias/variance (2)



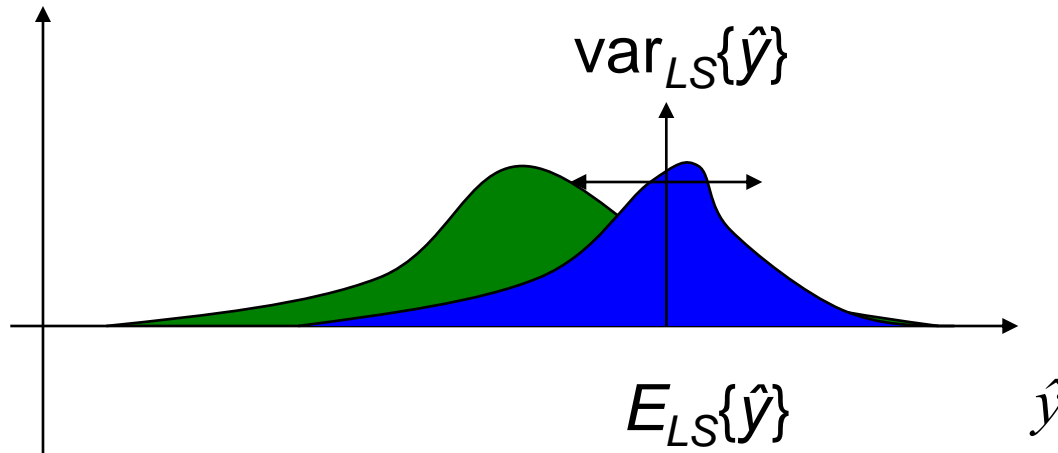
$$E = \text{var}_y\{y\} + (E_y\{y\} - E_{LS}\{\hat{y}\})^2 + \dots$$

$E_{LS}\{\hat{y}\}$  = průměrný model (přes všechny  $LS$ )

$\text{bias}^2$  = chyba průměrného modelu oproti modelu pana Bayese (optimálnímu)

Náš algoritmus učení

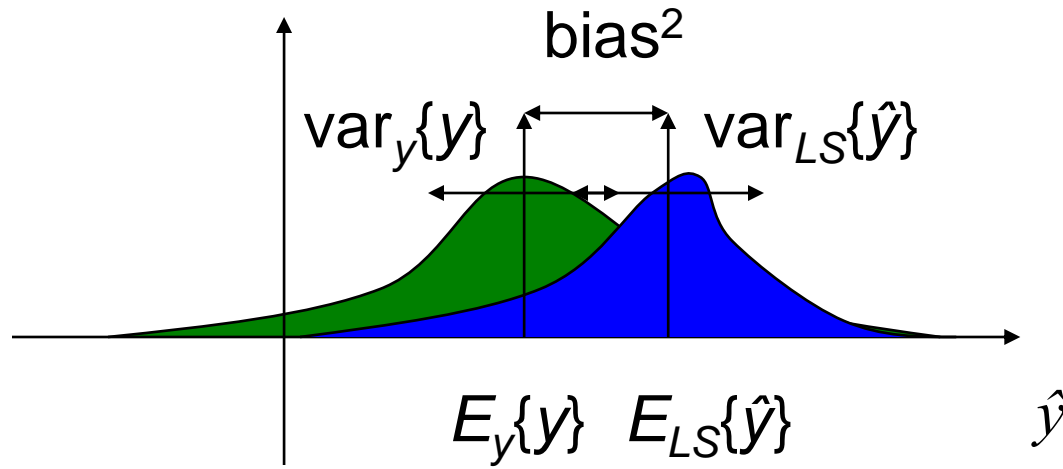
# Dekompozice bias/variance (3)



$$E = \text{var}_y\{y\} + \text{bias}^2 + E_{LS}\{(\hat{y} - E_{LS}\{\hat{y}\})^2\}$$

$\text{var}_{LS}\{\hat{\hat{y}}\}$  = variance našich modelů pro různé LS = důsledek přeučení

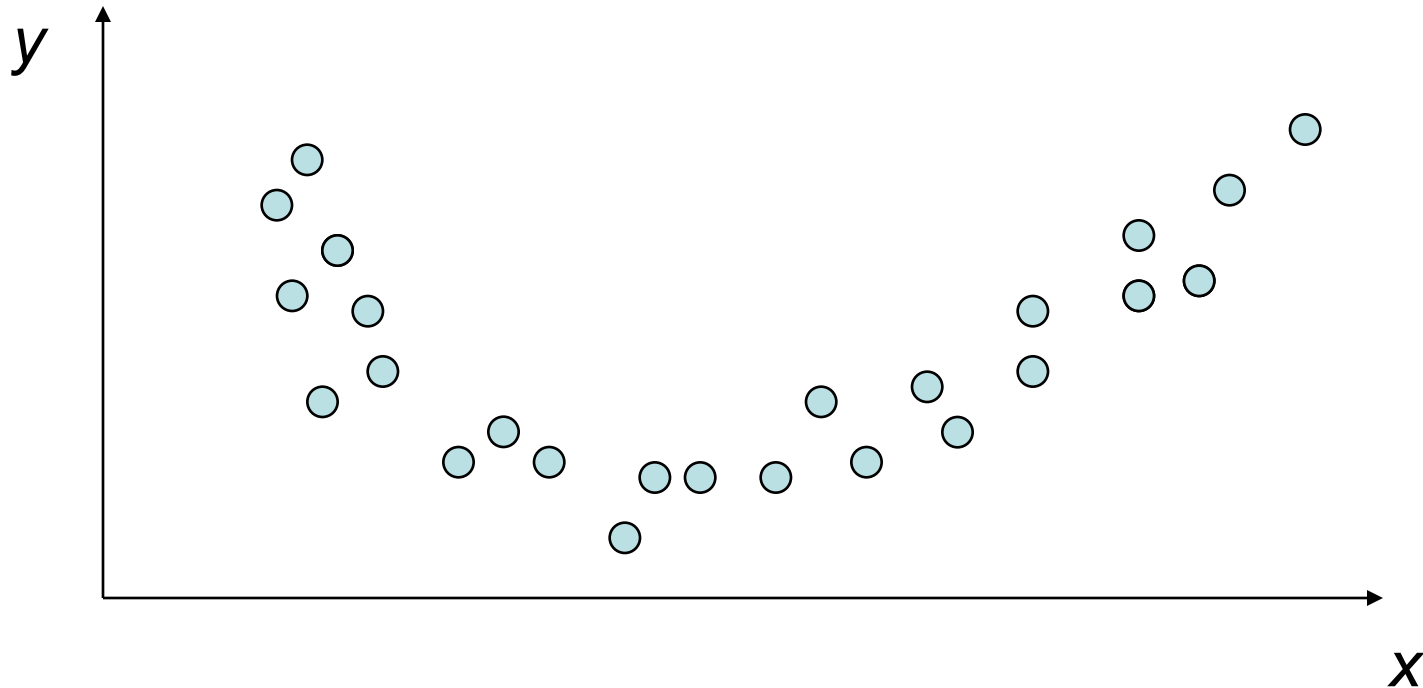
# Dekompozice bias/variance (4)



$$E = \text{var}_y\{y\} + \text{bias}^2 + \text{var}_{LS}\{\hat{y}\}$$

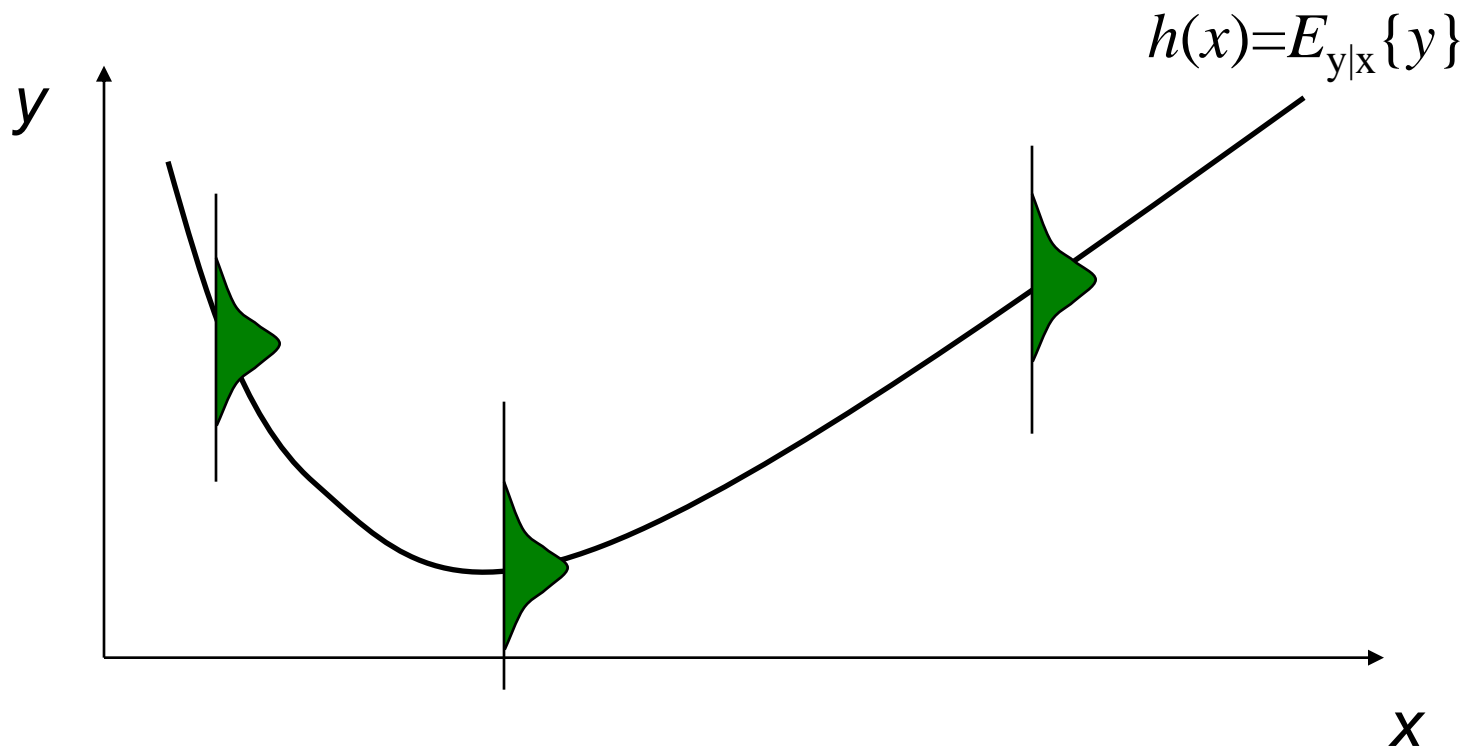
# Příklad (1)

- Najděte algoritmus produkující co nejlepší modely pro následující data:



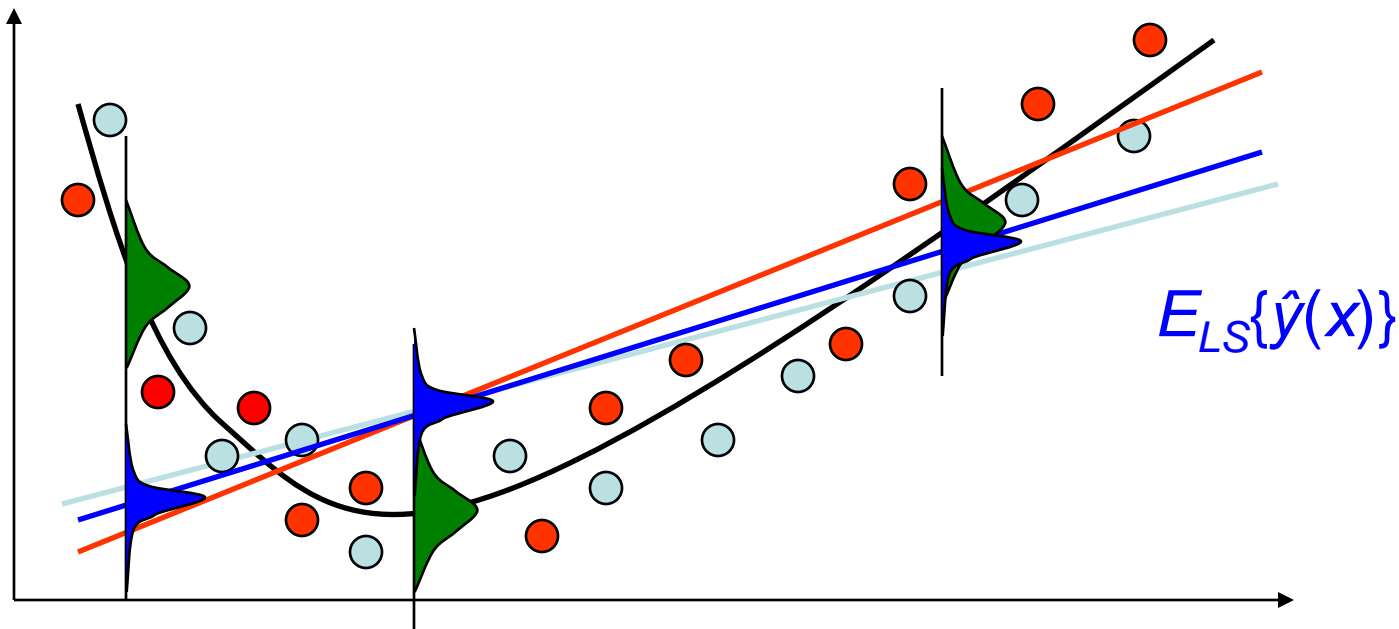
# Příklad (2)

- Optimální model:
  - Vstup  $x$ , náhodná proměnná rovnoměrně rozložená v intervalu  $[0,1]$
  - $y=h(x)+\varepsilon$ , kde  $\varepsilon\sim N(0,1)$  je Bayesovský model  $y$  a šum



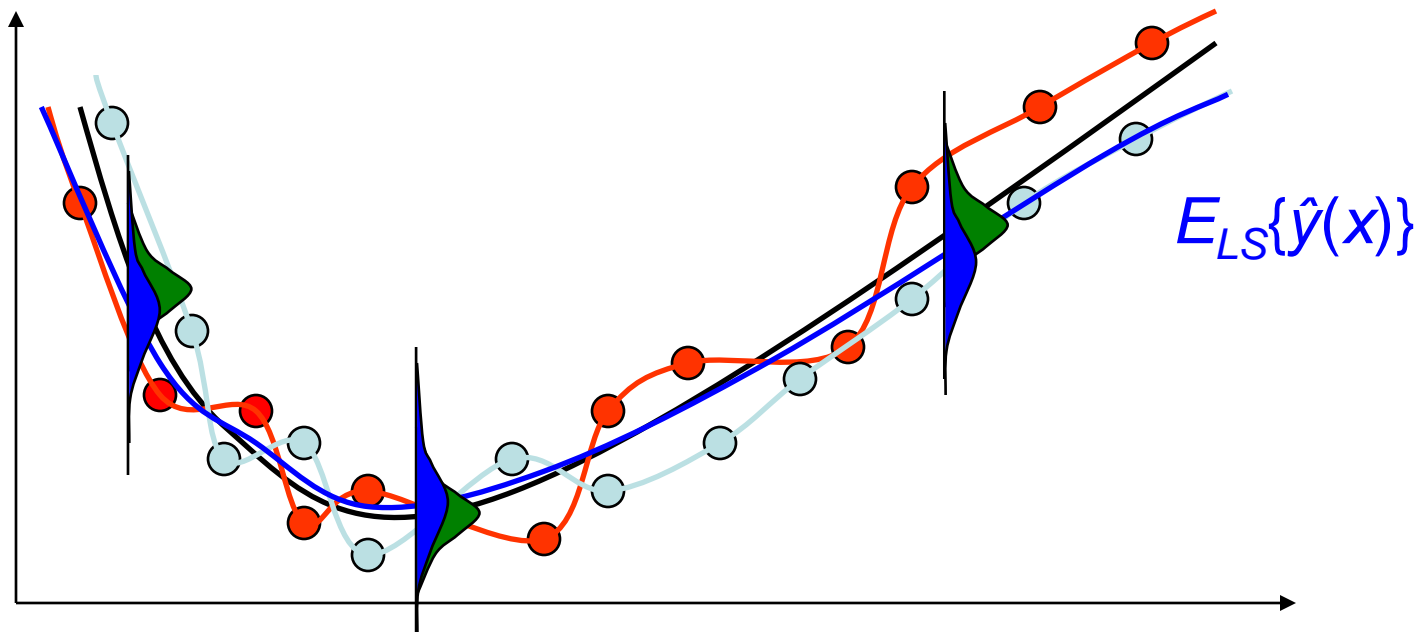
# Příklad – algoritmus lineární regrese

- Modely mají malý rozptyl (variance), ale velké zaujetí (bias)  $\Rightarrow$  nedoučení



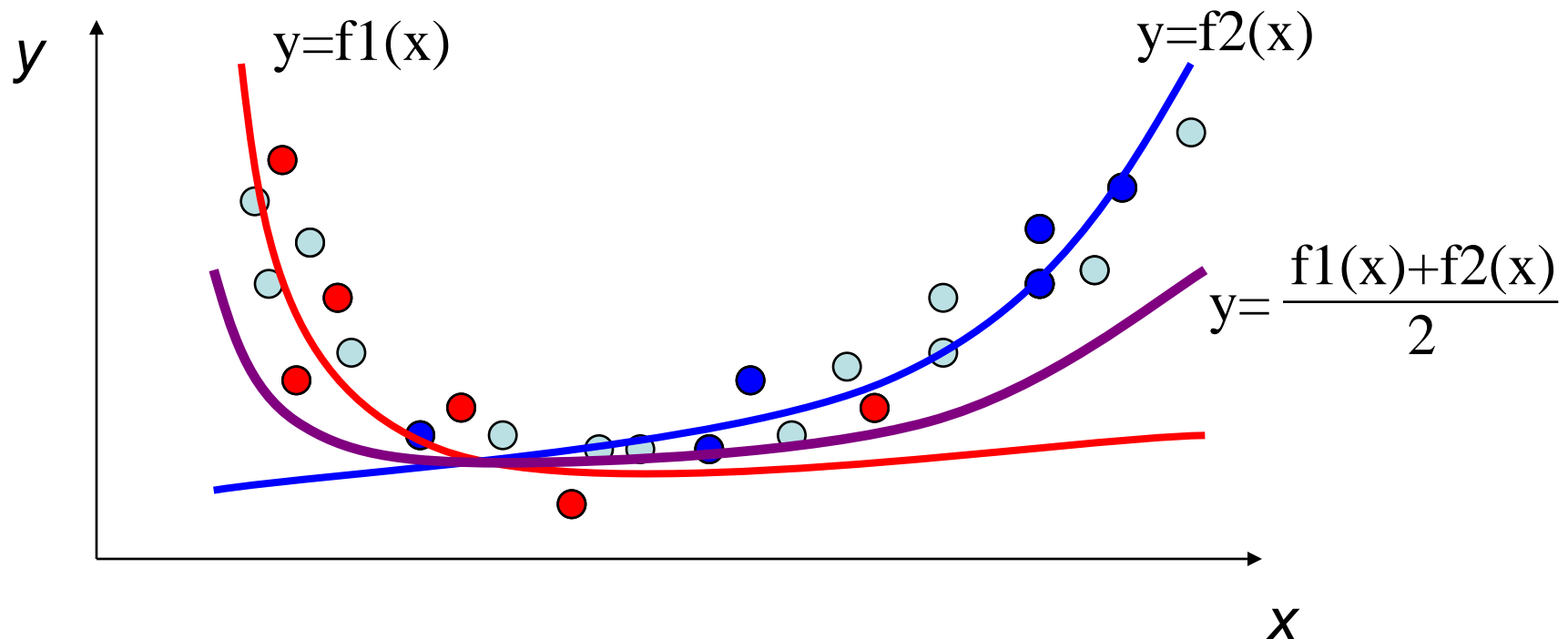
# Příklad – algoritmus RBFN s počtem neuronů = mohutnost LS

- Nízké zaujetí (bias), velký rozptyl (variance) modelů  $\Rightarrow$  přeučení



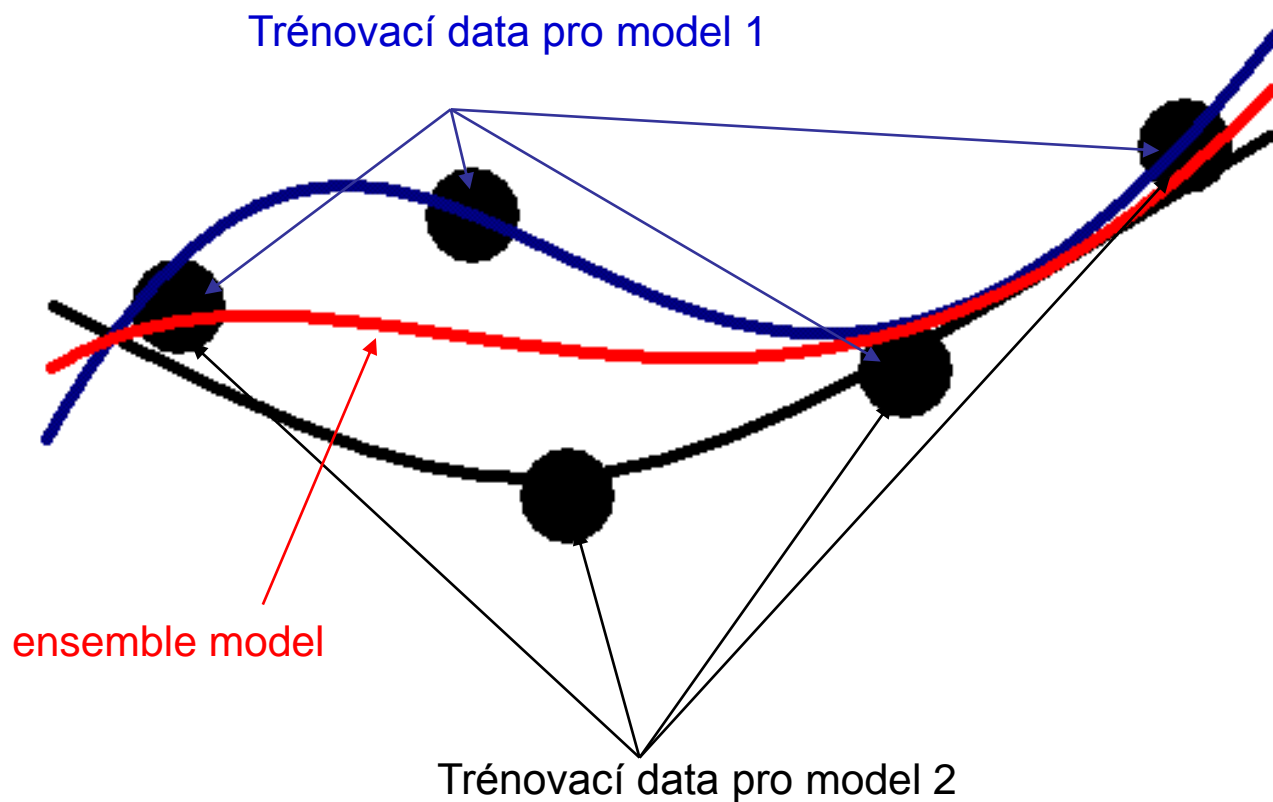
# Zpět ke kombinování modelů

- Co se stane, když naučím 2 jednoduché modely na různých podmnožinách LS?

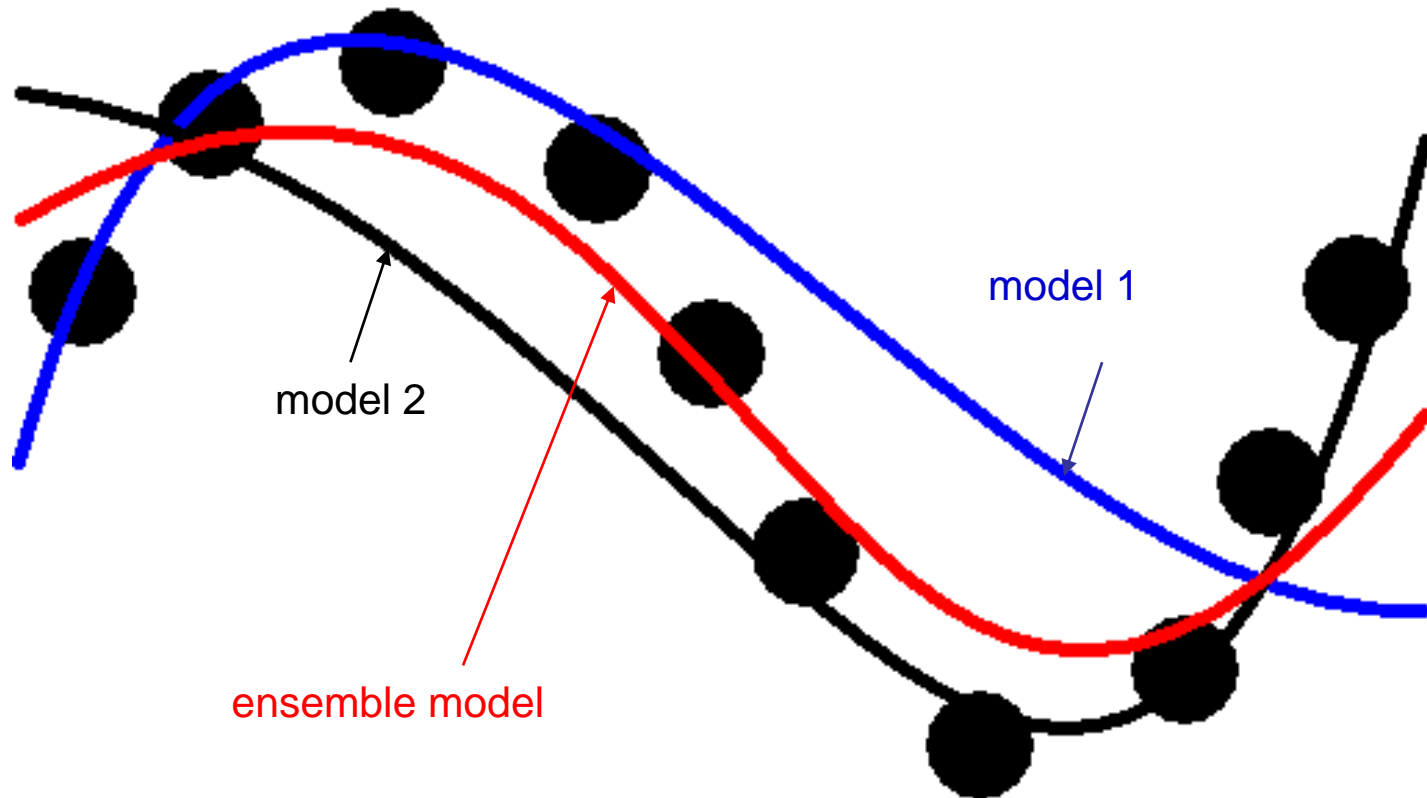




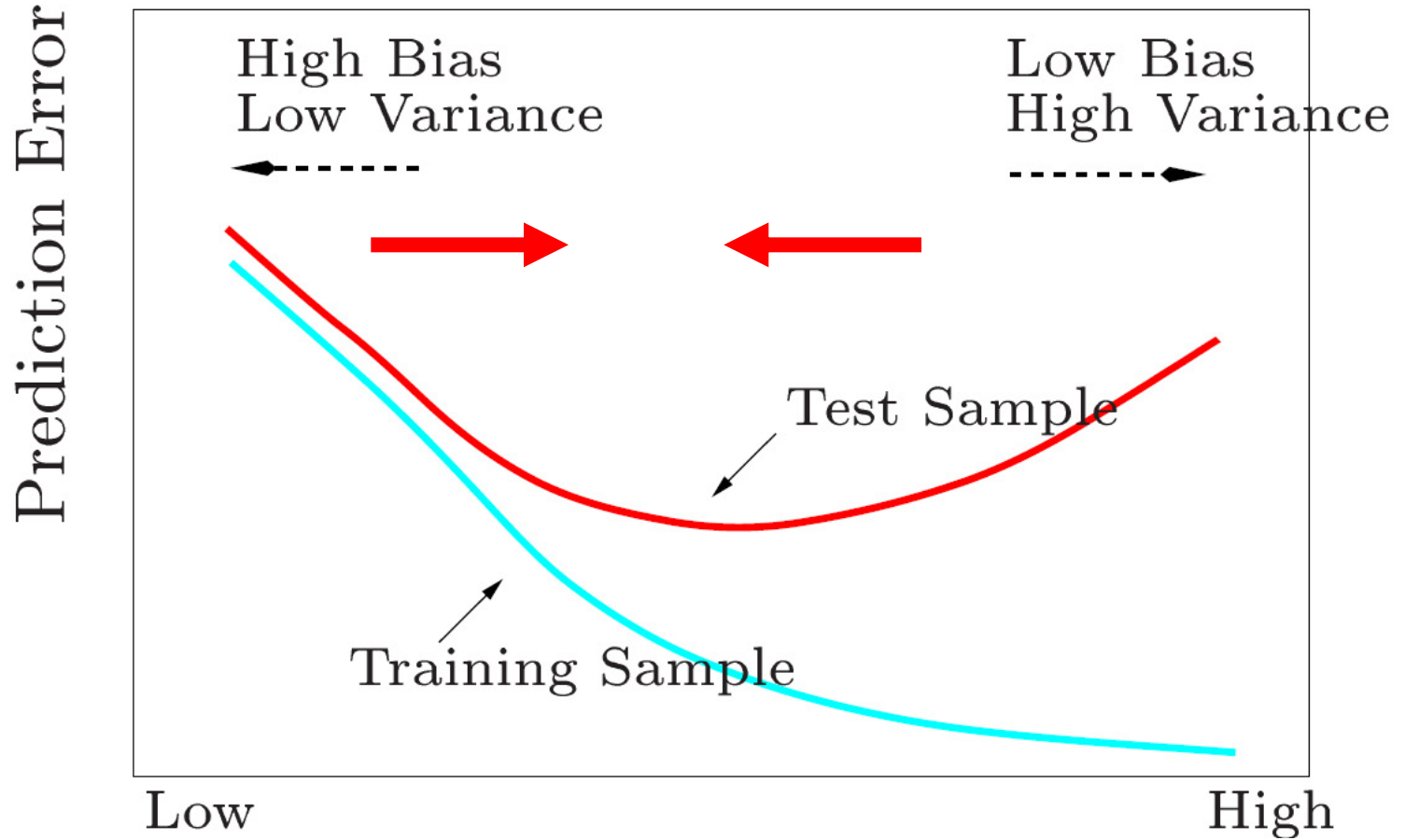
# Ensembling snižuje varianci



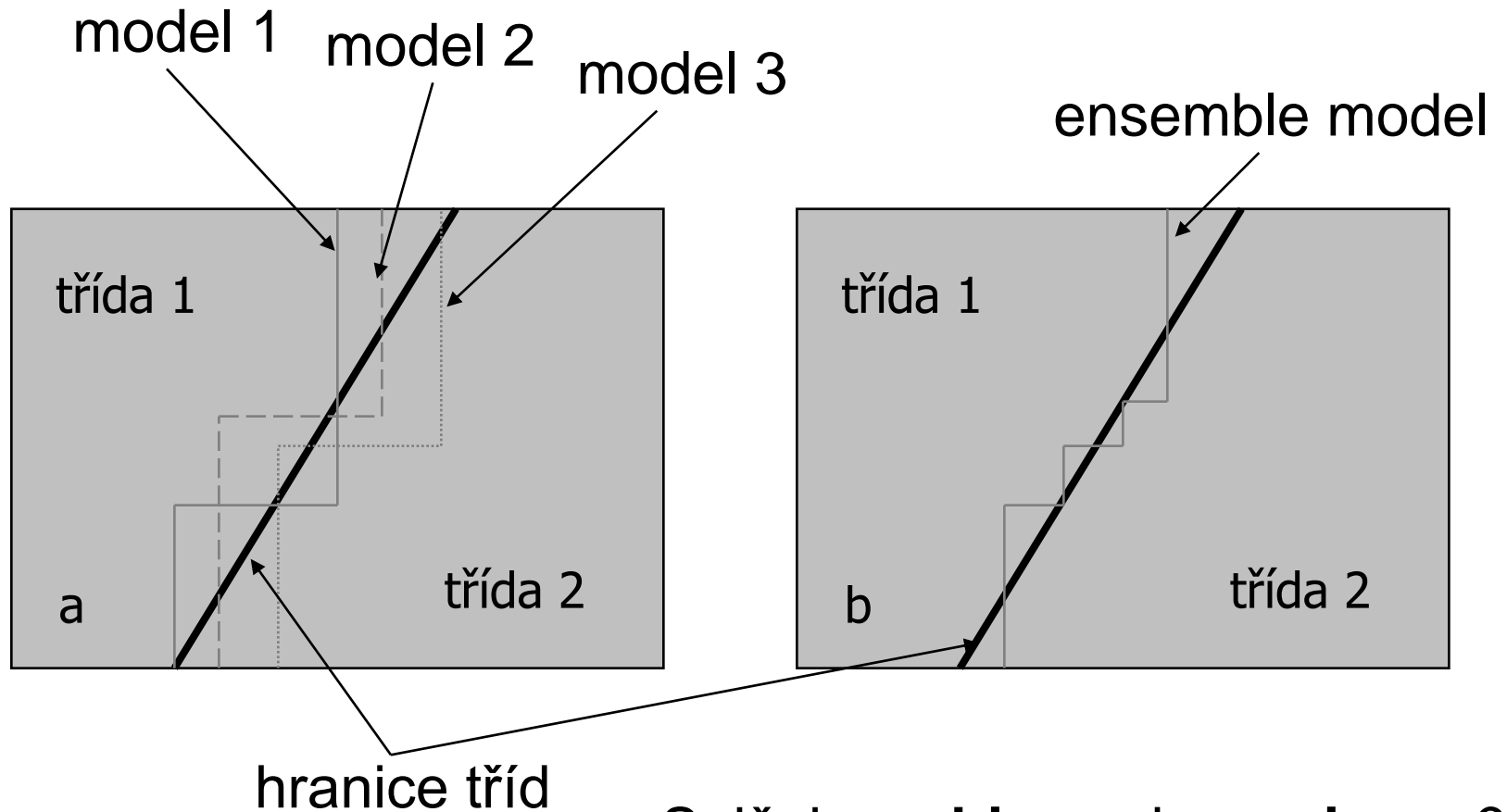
# Ensembling snižuje bias



# Tedy:



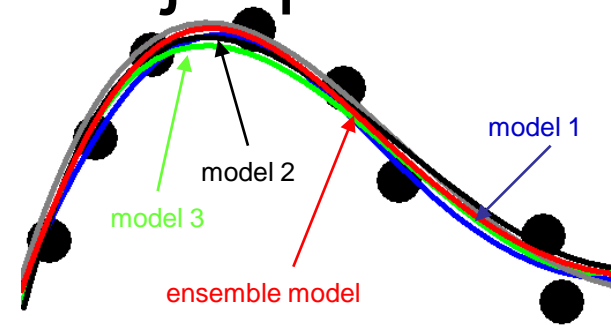
# Podobně pro klasifikaci?



Snižuje se **bias** nebo **variance**?

# Jaké modely kombinovat?

- Co se stane, když zkombinují optimálně naučené modely?



- Výhodně se dají kombinovat jednoduché modely (tzv. weak learners). redukujeme **bias**
- Modely musí být různorodé! Musejí vykazovat různé chyby na jednotlivých trénovacích vzorech. redukujeme **varianci**

# Populární ensemble metody

- Bagging (Bootstrap Aggregating)
  - Modely naučím nezávisle a jednoduše zkombinuji jejich výstup
- Boosting
  - Modely se učí sekvenčně, trénovací data jsou závislá na chybách předchozích modelů
- Stacking
  - Modely se učí nezávisle, kombinují naučením speciálního modelu

# Bagging (1)

$$E_{LS}\{Err(\underline{x})\} = E_{y|\underline{x}}\{(y - h_B(\underline{x}))^2\} + (h_B(\underline{x}) - E_{LS}\{\hat{y}(\underline{x})\})^2 + E_{LS}\{(\hat{y}(\underline{x}) - E_{LS}\{\hat{y}(\underline{x})\})^2\}$$

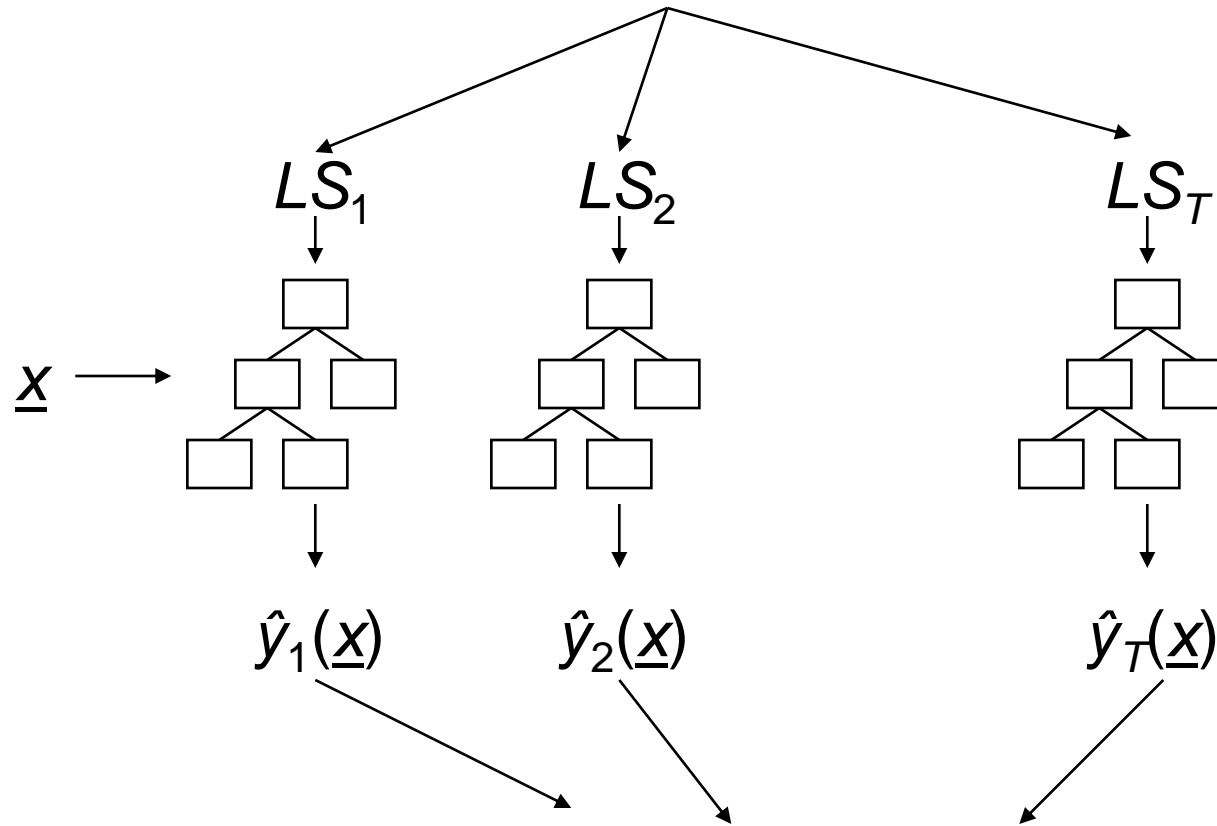
- **Myšlenka:** průměrný model  $E_{LS}\{\hat{y}(\underline{x})\}$  má stejný bias jako původní metoda, ale nulovou varianci
- **Bagging (Bootstrap AGGregatING) :**
  - K vypočtení  $E_{LS}\{\hat{y}(\underline{x})\}$ , potřebujeme nekonečně mnoho skupin  $LS$  (velikosti  $N$ )
  - Máme však jen jednu skupinu  $LS$ , musíme si sami nějak pomoc ...

# Příklad trénovacích množin

Trénovací vzory	1	2	3	4	5	6	7	8
vzorek 1	2	7	8	3	7	6	3	1
vzorek 2	7	8	5	6	4	2	7	1
...	...	...	...	...	...	...	...	...
vzorek M	4	5	1	4	6	4	3	8



# Bagging $LS$ (2)

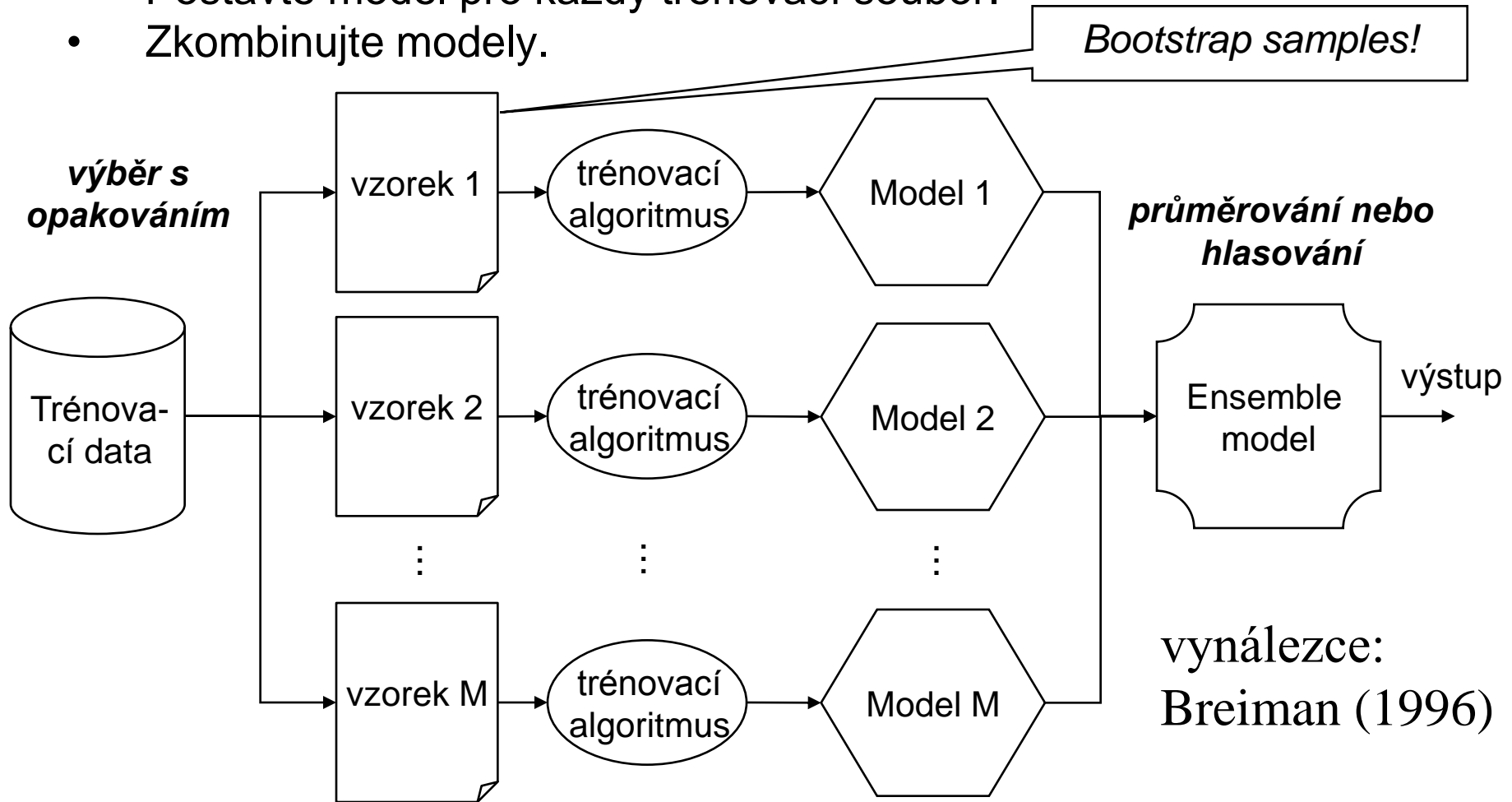


Pro regresi:  $\hat{y}(\underline{x}) = 1/k * (\hat{y}_1(\underline{x}) + \hat{y}_2(\underline{x}) + \dots + \hat{y}_T(\underline{x}))$

Pro klasifikaci:  $\hat{y}(\underline{x}) = \text{majoritní třída z } \{\hat{y}_1(\underline{x}), \dots, \hat{y}_T(\underline{x})\}$

# Bagging (Bootstrap aggregating)

- Výběrem s opakováním utvořte  $M$  trénovacích souborů o  $n$  vzorech (místo jednoho původního souboru o  $n$  vzorech).
- Postavte model pro každý trénovací soubor.
- Zkombinujte modely.



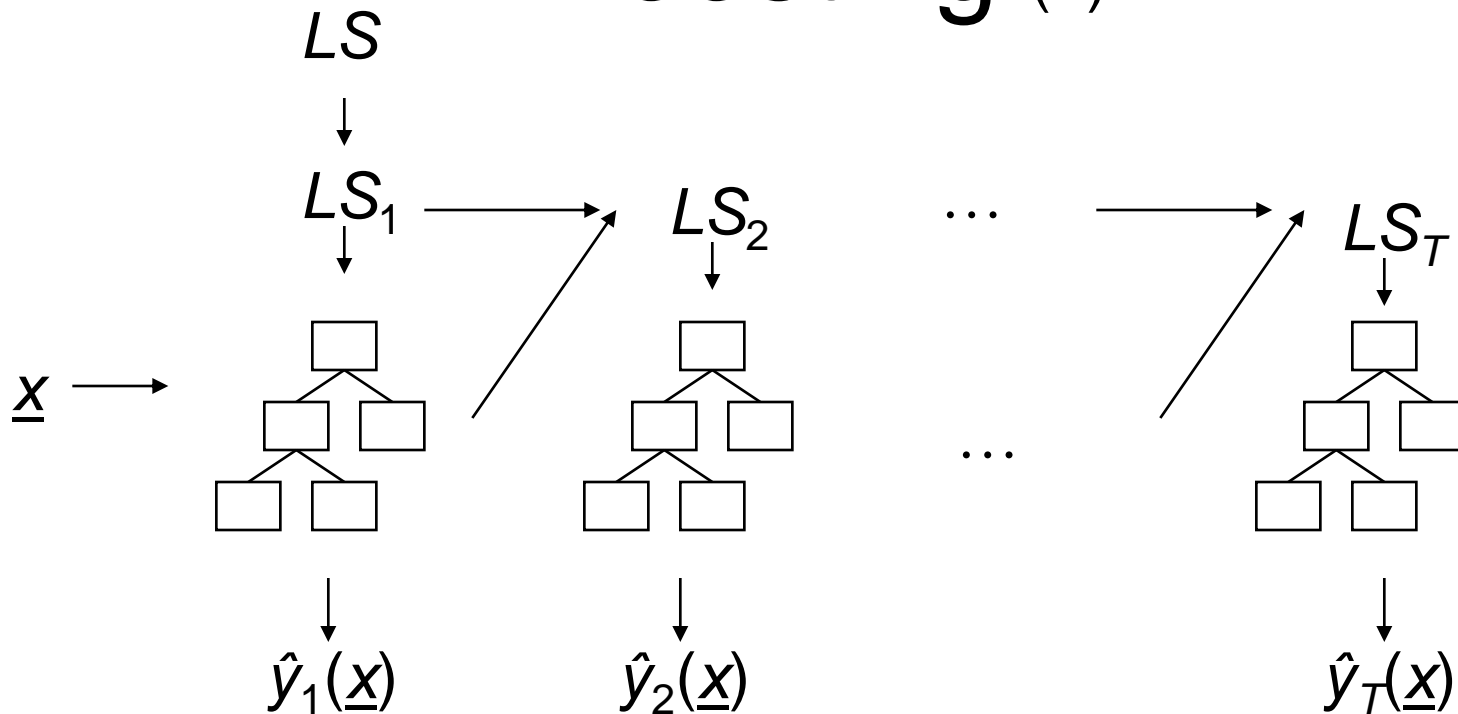
# Náhodné (rozhodovací) lesy – random forests

- K baggingu se ještě přidá to, že náhodně vybíráme podmnožinu vstupních atributů
- Tedy:
  - Postav rozhodovací strom z bootstrap vzorku
  - Najdi „best split“ mezi náhodnou podmnožinou  $k$  atributů, ne mezi všemi jako normálně(= bagging, když  $k$  je rovno počtu atributů)
  - Menší  $k$  redukuje varianci a zvyšuje bias

# Boosting

- Iterativní procedura adaptivně měnící rozložení učicích dat zvýrazňující špatně klasifikované vzory
- Používá se zejména ke kombinaci slabých modelů (weak learners), které mají velké zaujetí
- Výrazně redukuje bias – náchylnost k přeučení

# Boosting (2)



Pro regresi:  $\hat{y}(\underline{x}) = \beta_1 * \hat{y}_1(\underline{x}) + \beta_2 * \hat{y}_2(\underline{x}) + \dots + \beta_T * \hat{y}_T(\underline{x})$

Pro klasifikaci:  $\hat{y}(\underline{x}) = \text{majorita tříd z } \{\hat{y}_1(\underline{x}), \dots, \hat{y}_T(\underline{x})\}$   
s použitím vah  $\{\beta_1, \beta_2, \dots, \beta_T\}$

# Boosting (3)

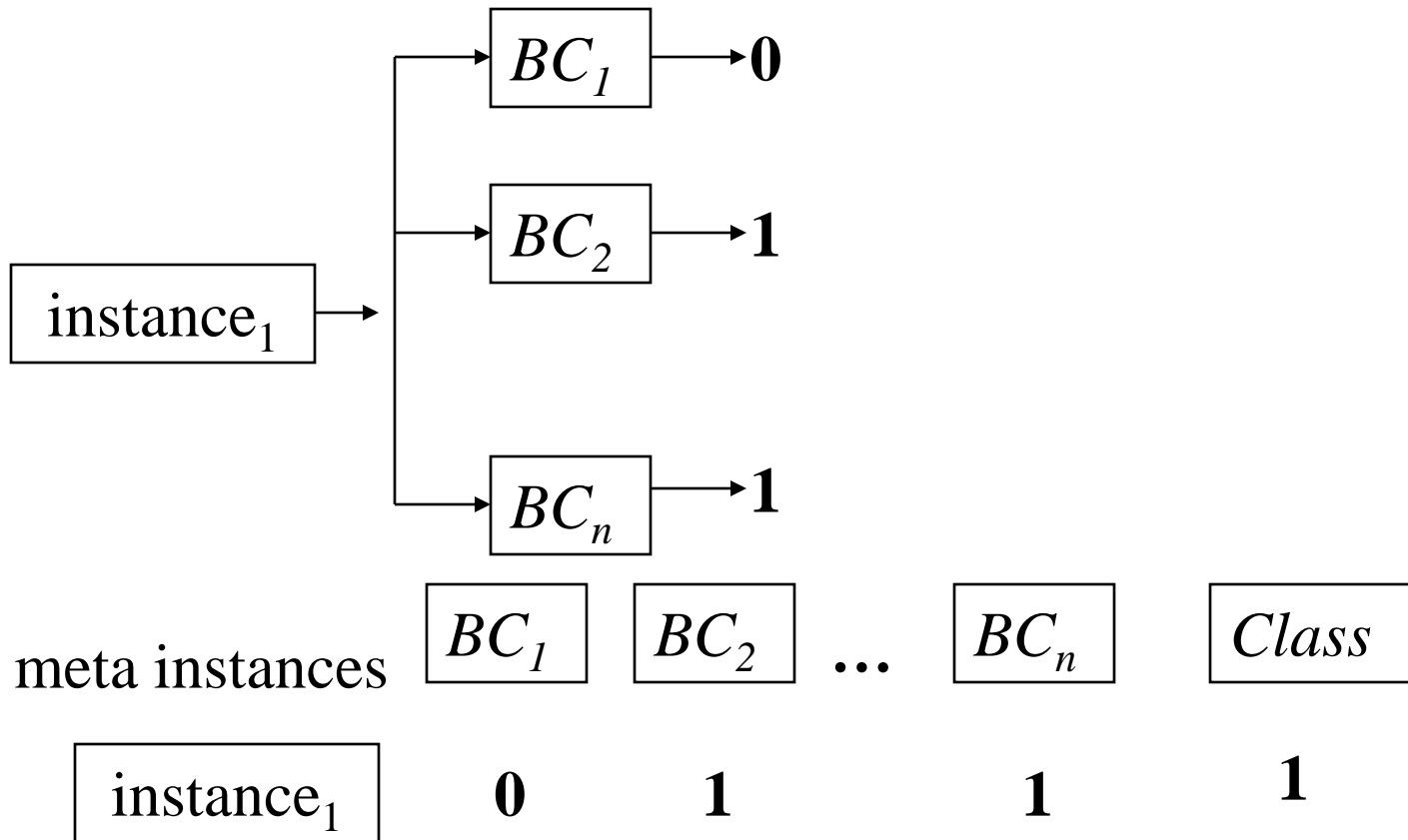
- Na počátku mají všechny vzory stejnou váhu
- Po naučení modelu zvýšíme váhu špatně klasifikovaným vzorům a snížíme ji dobře klasifikovaným
- **Příklad:** vzor 4 je špatně klasifikovatelný
- Postupně se zvyšuje jeho váha a tedy i pravděpodobnost zařazení do učicí množiny

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

# Stacking

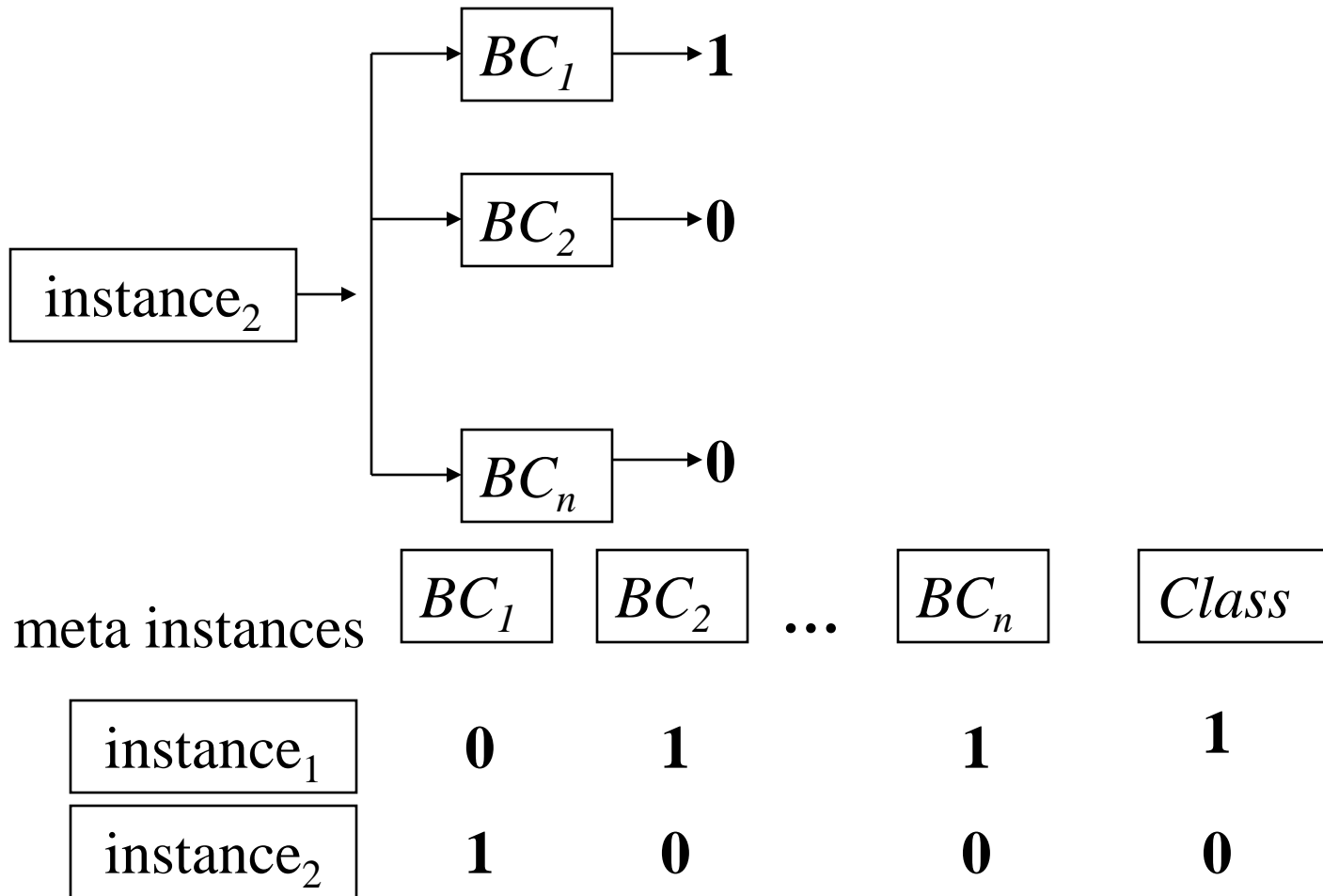
- Používá *meta model* pro kombinaci výstupů ensemble modelů (oproti jednoduchému průměrování, nebo hlasování)
  - Výstupy ensemble modelů jsou použity jako trénovací data pro meta model
- Ensemble modely jsou většinou naučeny různými algoritmy
- Teoretická analýza stackingu je obtížná, jedná se o “black magic”

# Stacking

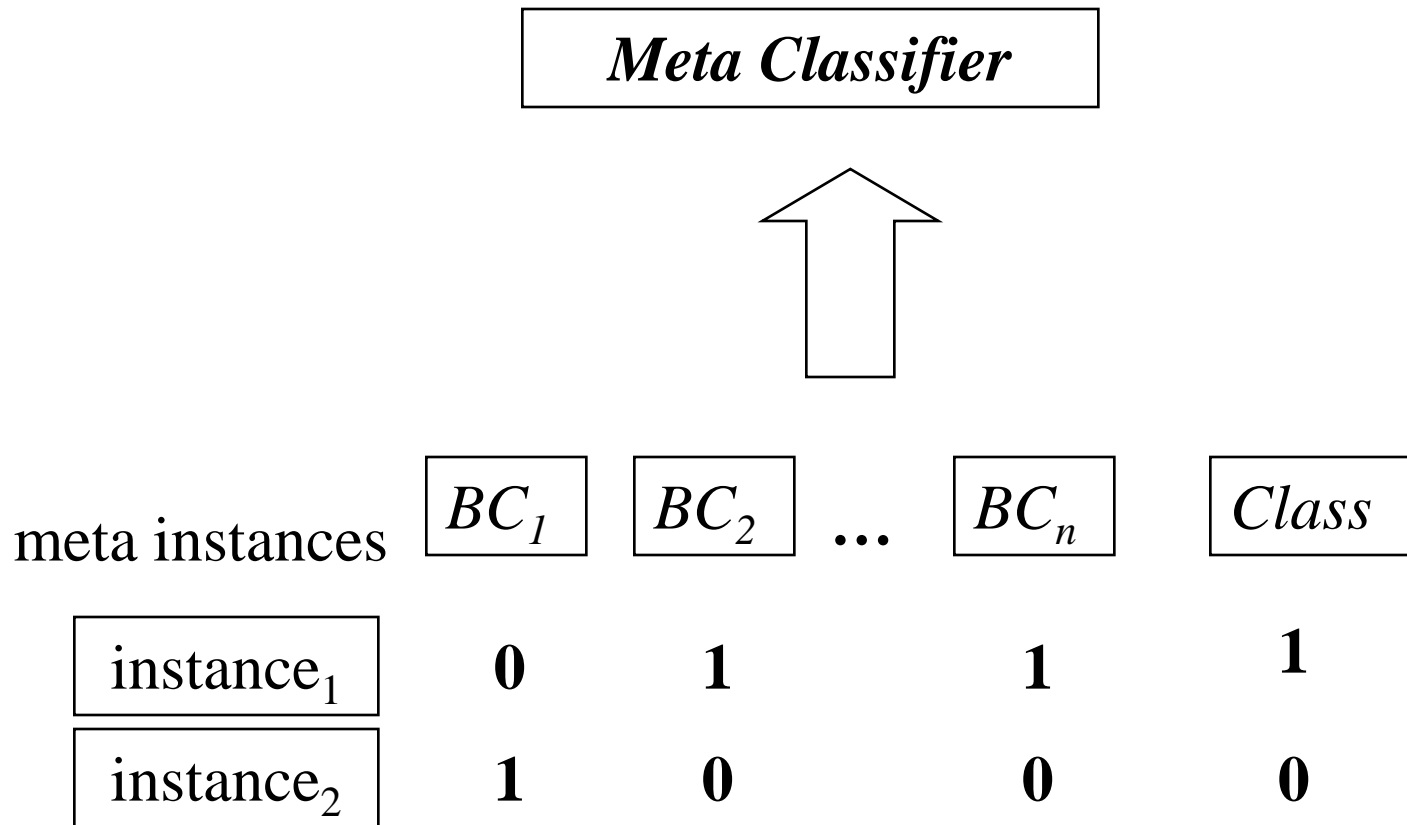




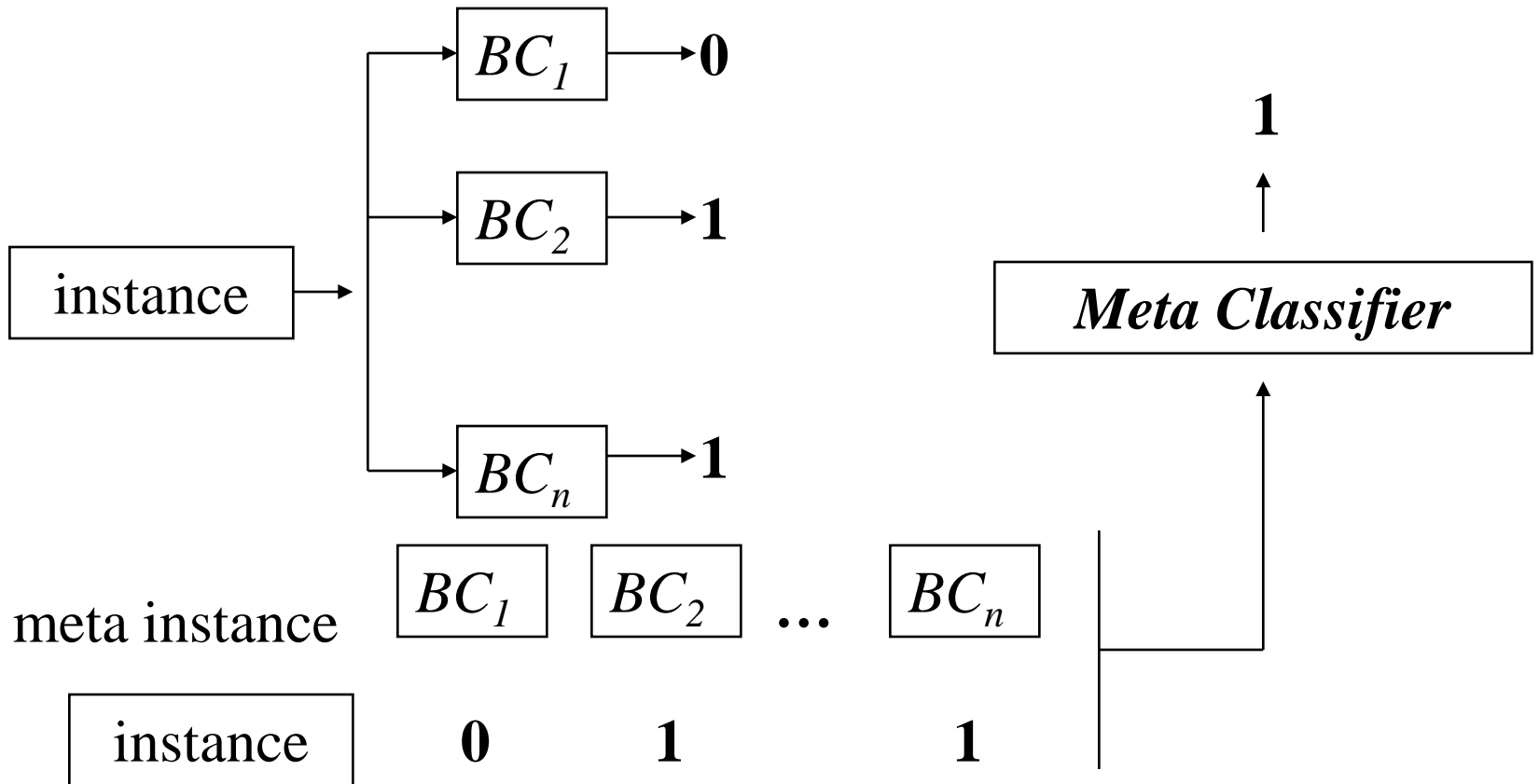
# Stacking



# Stacking



# Stacking



# Argumenty proti kombinování modelů?

- Okamova břitva – v jednoduchosti je síla
  - Je lepší mít jednoduchý optimální model, než kombinaci mnoha modelů
  - ... ale jak najít **optimální** model?
  - Domingos, P. Occam's two razors: the sharp and the blunt. KDD 1998.
- Kombinováním modelů se často kamufluje nedokonalost metod produkujících nedoučené nebo přeučené modely
- Kombinováním dostanu model s horšími výsledky na testovacích datech, než mají kombinované modely

# Argumenty pro kombinování

- Většinou zlepším výsledky na testovacích datech
  - Algoritmy jsou implicitně nastaveny, je třeba experimentovat s jejich konfigurací, aby produkováné modely byly optimální konkrétních datech
- Dostanu povědomí o jistotě modelu
  - když se pro jeden vstupní vektor jednotlivé modely hodně liší, zřejmě jsme mimo oblast trénovacích dat
- Netflix prize

# Co přesně použil vítěz?

- Víceúrovňový Stacking pomocí MLP
- Gradient Boosting rozhodovacích stromů
- A samozřejmě výborné base-klasifikátory:
  - KNN
  - SVD
  - RBM
  - ...

# Další info pro zájemce

- Podrobný popis vítězných algoritmů
  - [http://www.netflixprize.com/assets/GrandPrize2009\\_BPC\\_BellKor.pdf](http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf)
  - [http://www.netflixprize.com/assets/GrandPrize2009\\_BPC\\_BigChaos.pdf](http://www.netflixprize.com/assets/GrandPrize2009_BPC_BigChaos.pdf)
  - [http://www.netflixprize.com/assets/GrandPrize2009\\_BPC\\_PragmaticTheory.pdf](http://www.netflixprize.com/assets/GrandPrize2009_BPC_PragmaticTheory.pdf)





# Skupina výpočetní inteligence

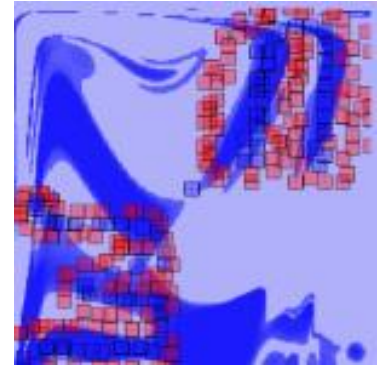
- Zdeněk Buk
- Ondřej Háva
- Miroslav Čepek
- Oleg Kovářík
- Jan Drchal
- Rudolf Marek
- Pavel Kordík
- Aleš Pilný
- Jan Koutník
- Petr Podhorský
- Miroslav Skrbek
- Tomáš Siegl
- Miroslav Šnorek
- Jakub Zahradník



# Vybrané projekty

- FAKE GAME

- Automatické zpracování dat



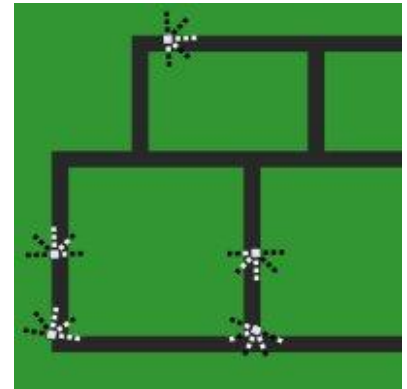
- JCool

- Open-source knihovna pro spojitou optimalizaci

- BlueCar

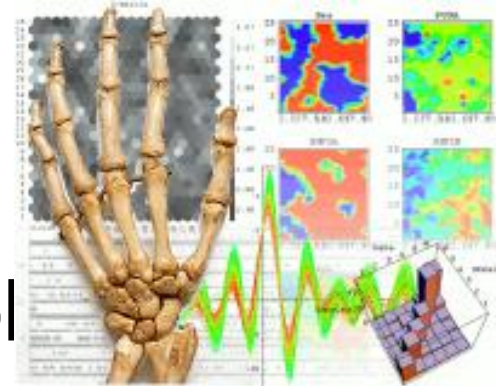
- ViVAE

- Fyzikální simulátor mobilních robotů



# Vybrané projekty

- Modelování stáří kosterních pozůstatků
- Tvorba modelu vývoje zubů české populace
- Diagnostika Parkinsonovy choroby
- Poruchy rovnováhy
- Papoušci



# Kdo, co...

- Zdeněk Buk

- Hybridní evoluce neuronových sítí
- Robotický simulátor (ViVAE)
- HyperGP – nepřímé kódování neuronových sítí
- Celulární kódování neuronových sítí

- Miroslav Čepek

- Počítačem podporované předzpracování dat pro data mining (algoritmus pro automatické předzpracování dat).

# Kdo, co...

- Jan Drchal
  - Nepřímé kódování neuronových sítí
  - Robotický simulátor (ViVAE)
  - JCool – optimalizační algoritmy
  - Vizualizace mnohadimenzionálních dat
- Ondřej Háva
  - Text mining

# Kdo, co...

- Oleg Kovářík
  - Optimalizace pomocí mravečích kolonií
  - Kombinování algoritmů
- Rudolf Marek
  - Paralelizace algoritmu THSOM
  - IBM Cell
- Aleš Pilný
  - Selektce nejvýznamnějších atributů v datech
  - Ohodnocování významnosti atributů

# Kdo, co...

- Petr Podhorský
  - Pulzní neuronové sítě
  - Třetí generace neuronových sítí, biologicky realistický model neuronu
  - Aktuální výzkum: evoluce pulzních sítí s použitím nepřímého kódování
  - Možné budoucí aplikace: ovládání a navigace robotů, klasifikace dat s časovou složkou

# Kdo, co...

- Tomáš Siegl – ensemble a statistické metody, negativní korelace
- Jakub Zahradník
  - Neuronové sítě s komplexními vahami
  - zobrazení stavu sítě na povrch 2D toroidu, pro sledování změn během učení
  - klasifikace časoprostorových dat



# Akce...

Každý pátek, schůzka skupiny CIG

- prezentace algoritmů, výzkumu
- otevřená debata o problémech souvisejících s výpočetní inteligencí

Místo – K205, katedra počítačů, FEL,  
Karlovo nám. 13

Čas – 9:15

Program se snažíme dávat na:

<http://cig.felk.cvut.cz>

Jste srdečně zváni!



**EUROSIM 2010**  
7<sup>th</sup> **EUROSIM Congress on Modelling and Simulation**  
**EUROSIM Congress the most important modelling and simulation event in Europe**

**September 2010, Prague, Czech Republic**

**Congress Venue**

The Congress will take place in Prague, the capital city of Czech Republic, at the Congress Center of Masaryk College, part of Czech Technical University, in cooperation with the Faculty of Electrical Engineering of CTU.

**About Czech Technical University in Prague**

Czech Technical University celebrates 300 years of its history in 2007. Under the name Estate Engineering Teaching Institute in Prague was founded by the rescript of the Emperor Josef I of 18 January 1707 on the basis of a petition of Christian Josef Willenberg (1678-1731). This school was reorganized in 1806 as the Prague Polytechnic, and, after the disintegration of the former Austro-Hungarian Empire in 1918, transformed in to the Czech Technical University in Prague.

**About EUROSIM**

EUROSIM, the federation of European simulation societies, was set up in 1969. Its purpose is to promote, especially through local simulation societies, the idea of modelling and simulation in different fields, industry, research and development. At present, EUROSIM has 14 full members and 4 observer members.

**Congress Scope and Topics**

The Congress scope includes all aspects of continuous, discrete (event) and hybrid modelling, simulation, identification and optimisation approaches. Contributions from both technical and non-technical areas are welcome. Two basic tracks will be organized: M&S Methods and Technologies and M&S Applications.

**Czech Republic - EUROSIM 2010 Host Country**

The Czech Republic is a country in the centre of Europe. It is interesting for its 1,000-year-long history, rich culture and diverse nature. The country is open to new influences and opportunities thanks to a high level of industrial infrastructure, safety measures and plural media. The location of the Czech Republic in the very heart of Europe contributes to the fact that one can get there easily and fast. Usually all it takes to enter the country is a valid passport. The need for a visas to enter the Czech Republic is very exceptional.

**Prague - EUROSIM 2010 Host City**

Prague is a magical city of bridges, cathedrals, gold-tipped towers and church spires, whose image has been mirrored in the surface of the Vltava River for more than a millennium. Walking through the city, you will quickly discover that the entire history of European architecture has left splendid representatives of various periods and styles. There are Romanesque, Gothic, Renaissance, Baroque and Classicist buildings, as well as more modern styles, such as Art Nouveau and Cubist. A poet once characterized Prague as a symphony of stones.

**About CSSS**

CSSS (The Czech and Slovak Simulation Society) has about 150 members in 2 groups connected to the Czech and Slovak national scientific and technical societies (Czech Society for Applied Cybernetics and Informatics, Slovak Society for Applied Cybernetics and Informatics). Since 1992 CSSS is a full member of EUROSIM.

**Invitation**

Czech and Slovak Simulation Society is greatly honored with the congress organisation and will do the best to organise an event with a high quality scientific programme with some other accompanied actions but also with some unforgettable social events.

**Prof. Miroslav Šnorek**, nominated for next EUROSIM president,  
**Prof. Miroslav Šnorek**, president of CSSS





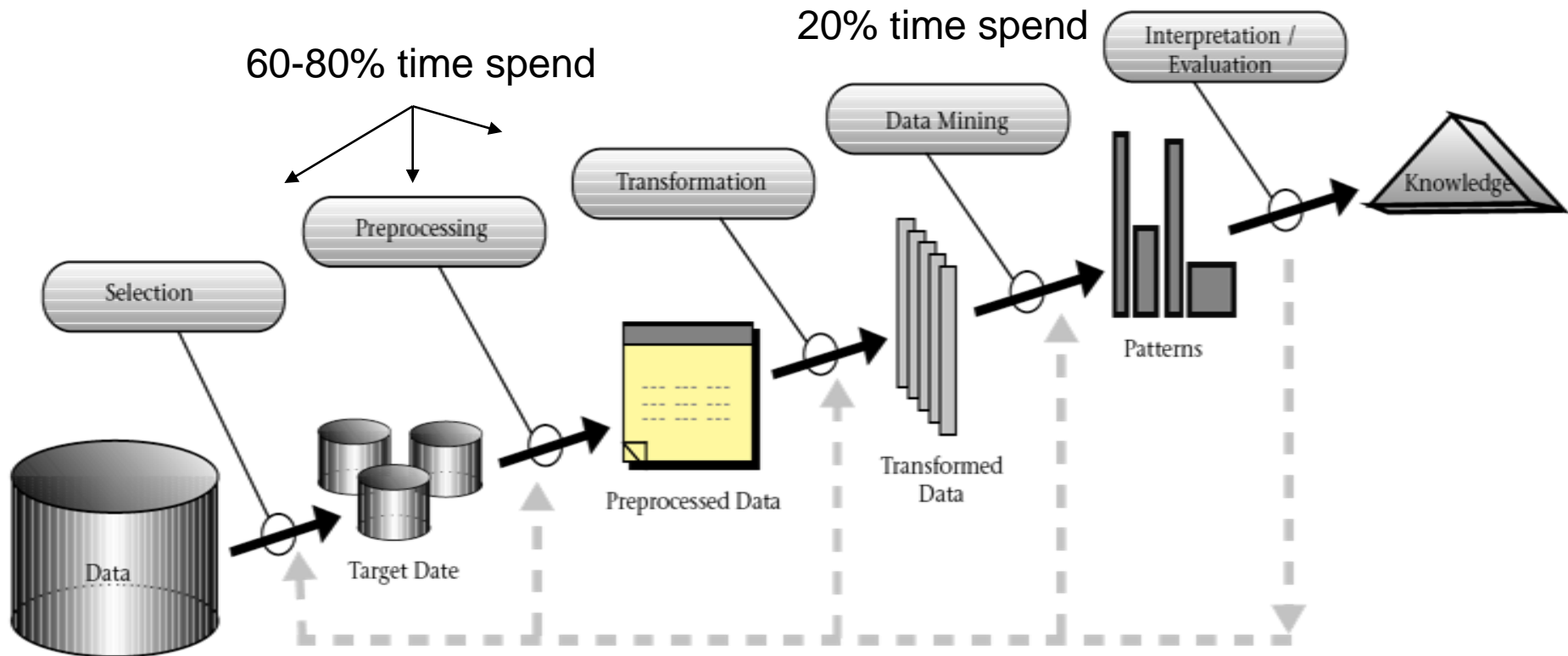
# FAKE GAME

**F**ully **A**utomated **K**nowledge **E**xtraction  
using  
**G**roup of **A**daptive **M**odels **E**volution

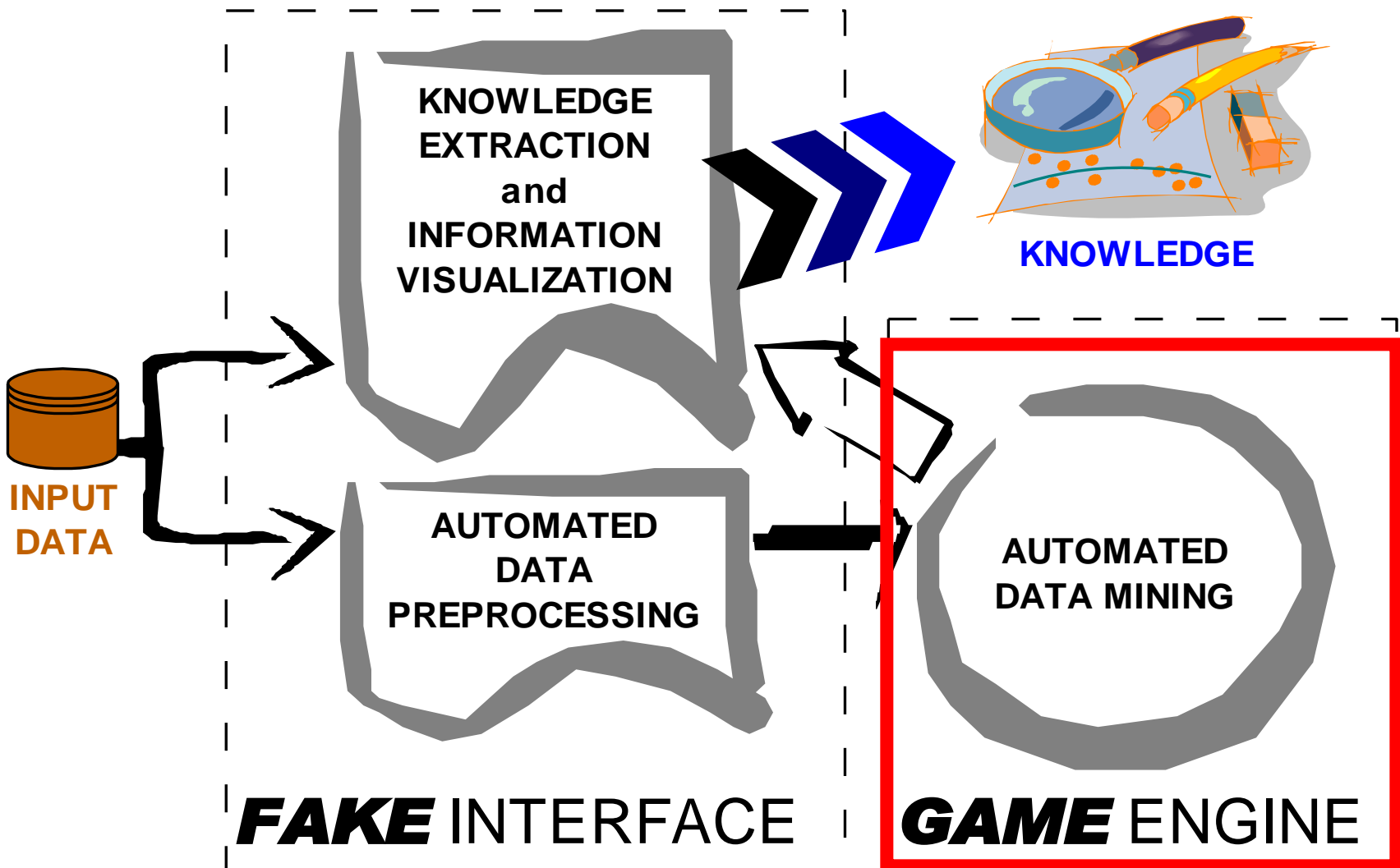
Open source java application hosted on sourceforge:

<http://sourceforge.net/projects/fakegame>

# The process of Knowledge Discovery from Databases



# Our approach: Automate some parts of the process

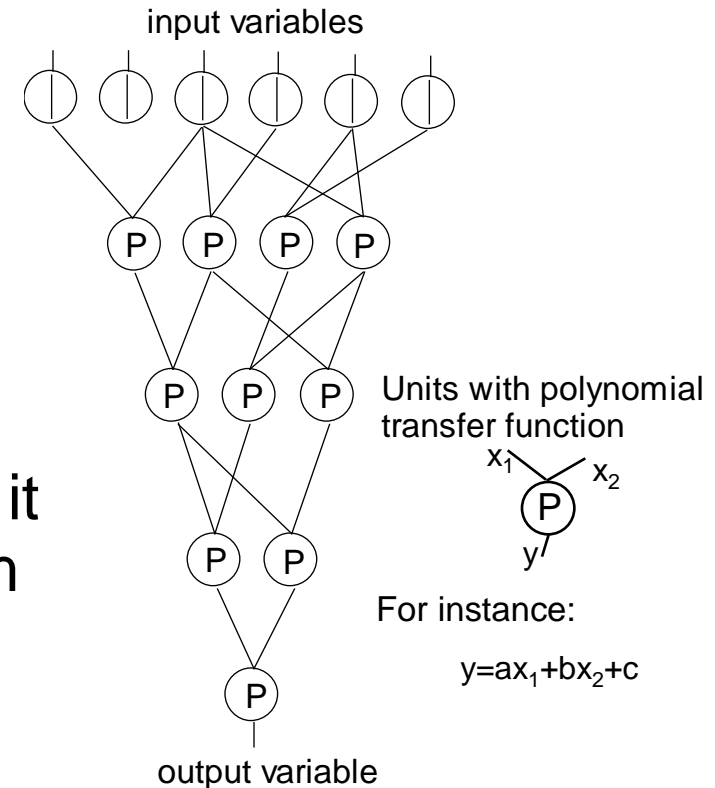


# Background (GMDH)

- **Group Method of Data Handling**  
GMDH – prof. Ivakhnenko in 1968

- The principle of ***INDUCTION*** employed:

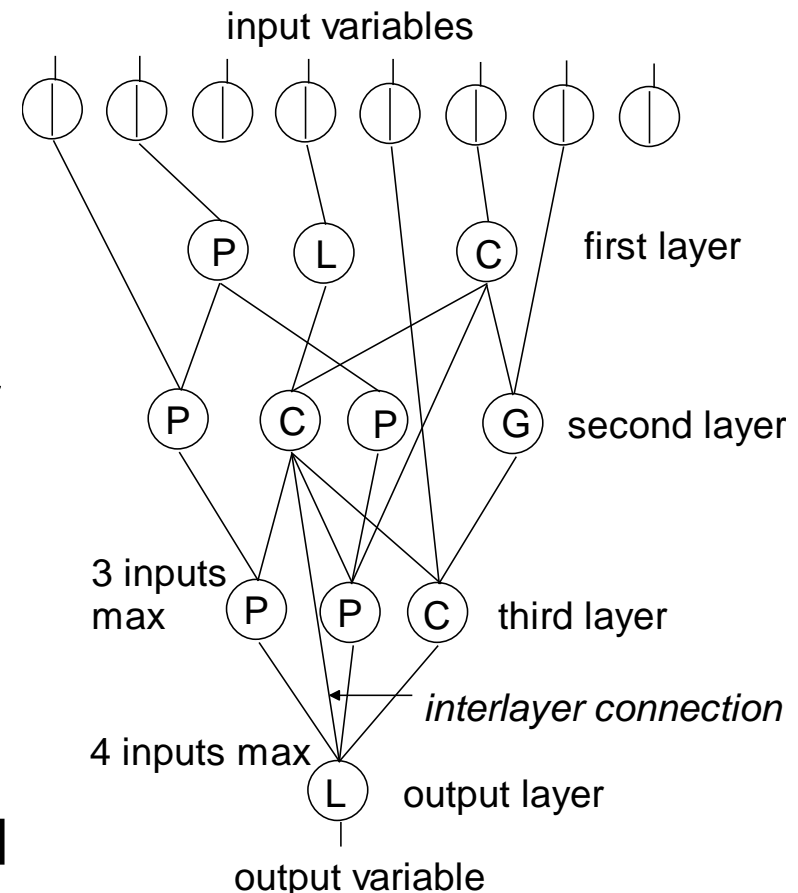
**Inductive model** grows from data, it decomposes a multivariate problem into subproblems, solves it in the space of lower dimensionality (*two dimensions for units with 2 inputs*), combines partial solutions to get the global solution of the problem.



MIA GMDH – the most commonly used algorithm

# Recent improvements (GAME)

- **Group of Adaptive Models Evolution (GAME)**
  - Heterogeneous units
  - Several competing training methods
  - Interlayer connections, number of inputs allowed increases
  - Niching genetic algorithm (will be explained) employed in each layer to optimize the topology of GAME networks.
  - Ensemble of models generated (will be explained)



# Example: Housing data

Input variables

CRIM ZN INDUS NOX RM AGE DIS RAD TAX PTRATIO B LSTA

Per capita crime rate by town

Weighted distances to five Boston employment centers

Proportion of owner-occupied units built prior to 1940

Median value of owner-occupied homes in \$1000's

MEDV

Output variable



# Housing data – records

Input variables										Output variable		
CRIM	ZN	INDUS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTA	MEDV
24	0.00632	18	2.31	53.8	6.575	65.2	4.09	1	296	15.3	396.9	4.98
21.6	0.02731	0	7.07	46.9	6.421	78.9	4.9671	2	242	17.8	396.9	9.14
...	...	...										

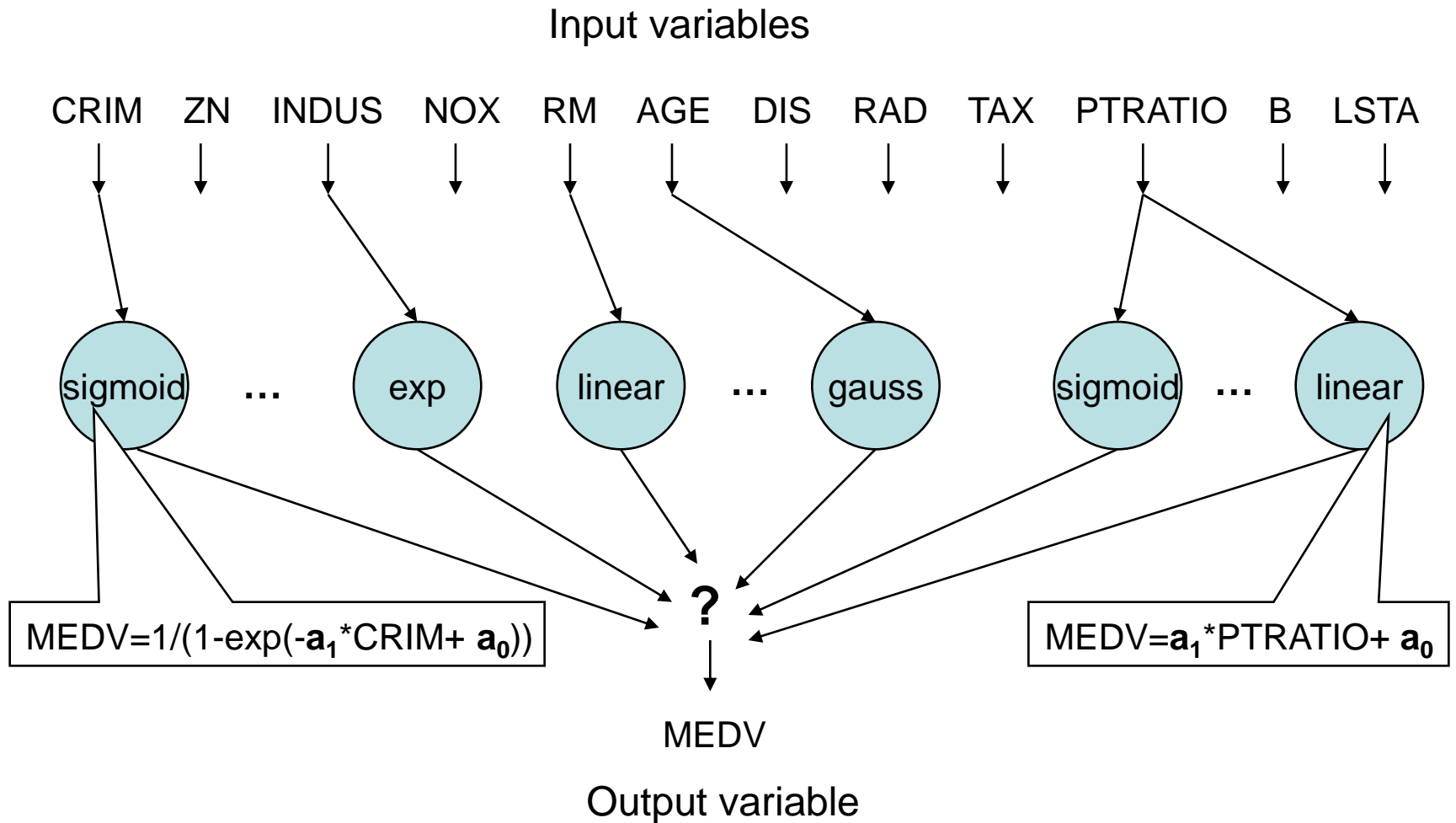
A {  
B {  
C {

A = Training set ... to adjust weights and coefficients of neurons

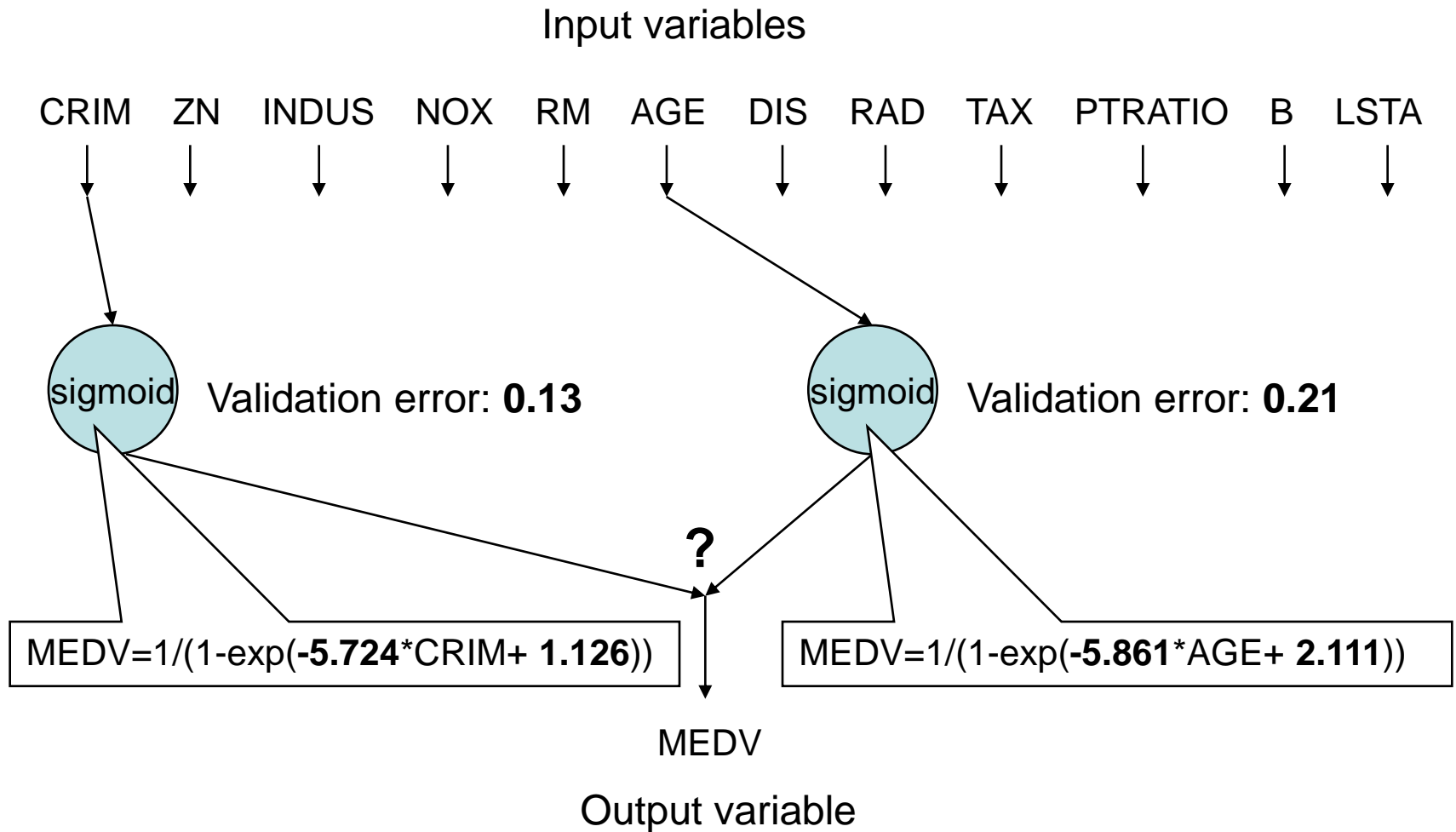
B = Validation set ... to select neurons with the best generalization

C = Test set ... not used during training

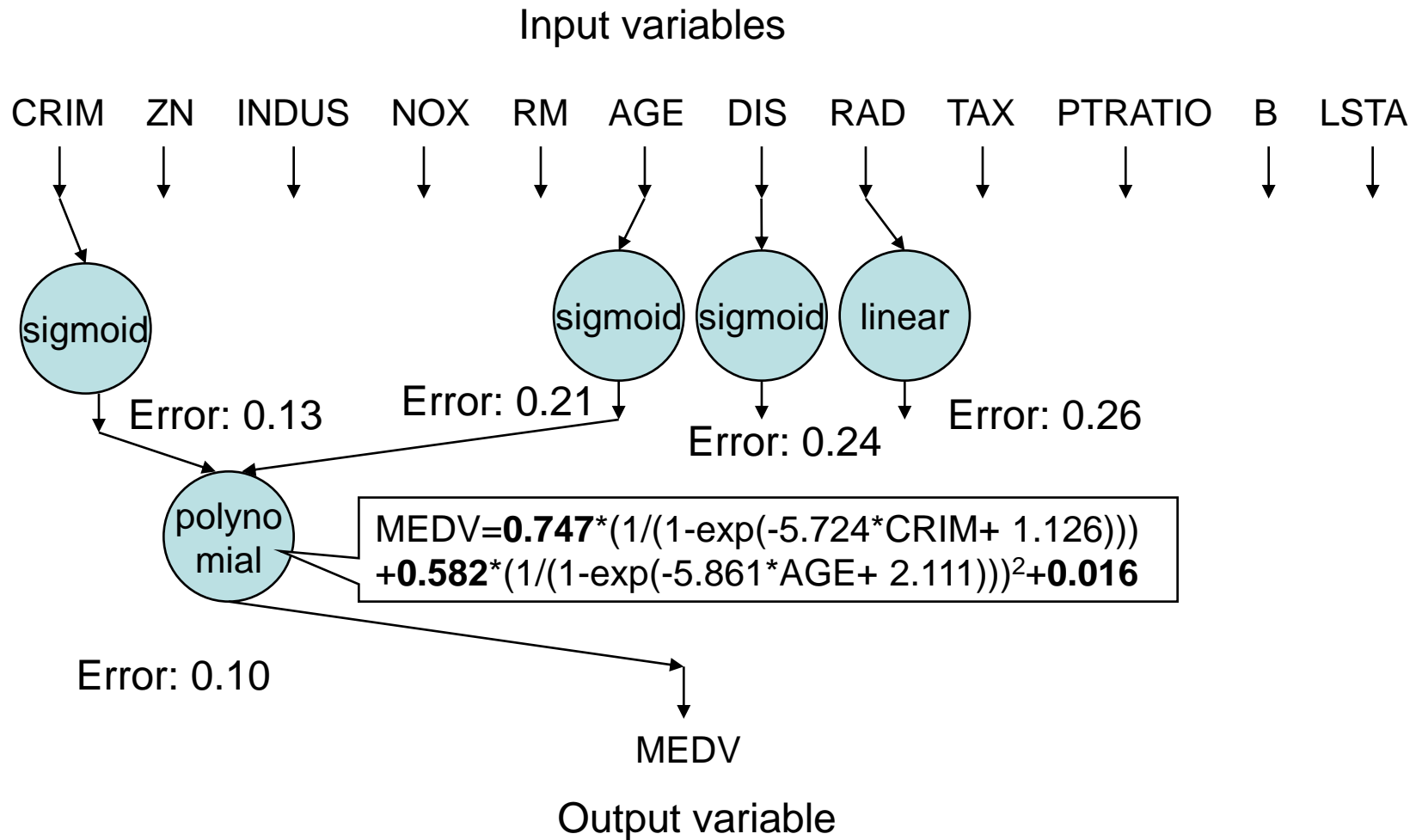
# Housing data – inductive model



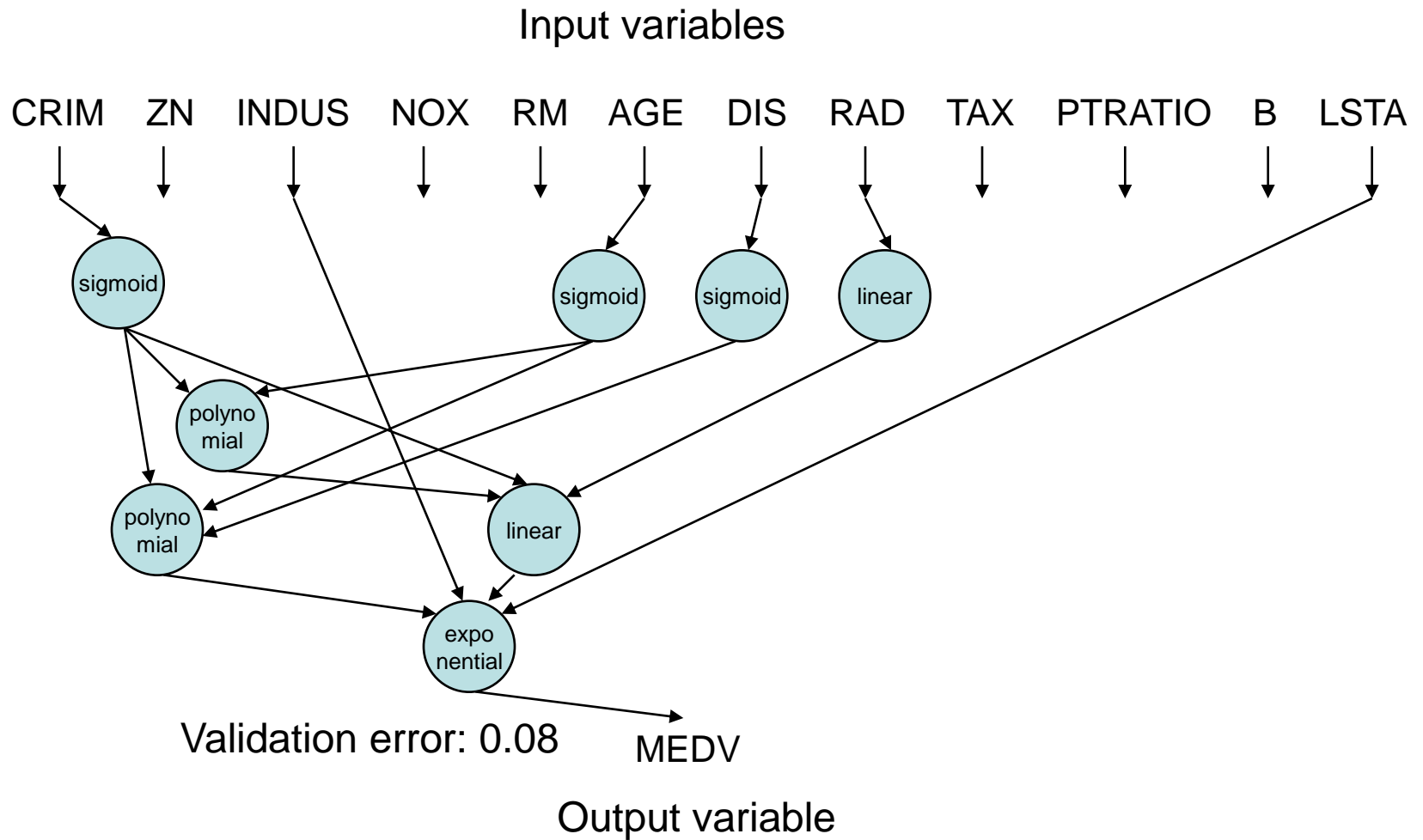
# Housing data – inductive model



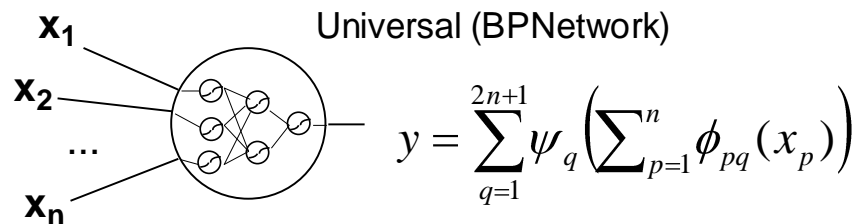
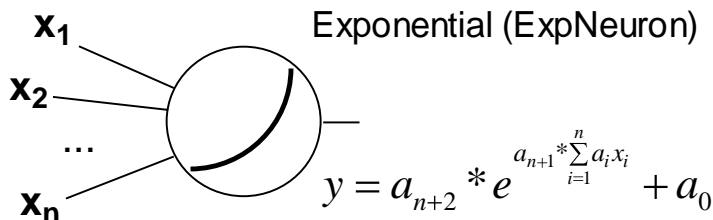
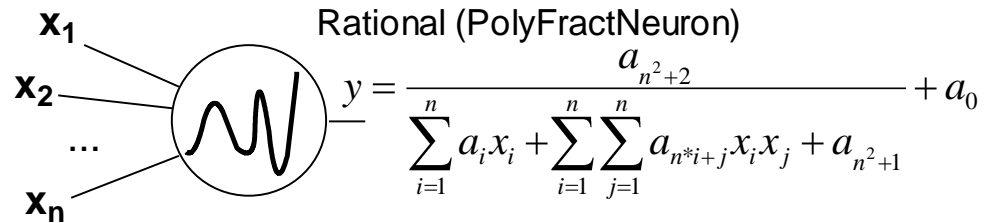
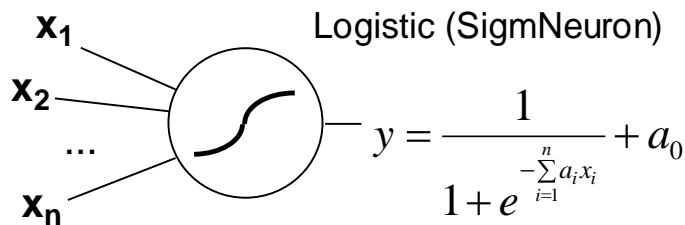
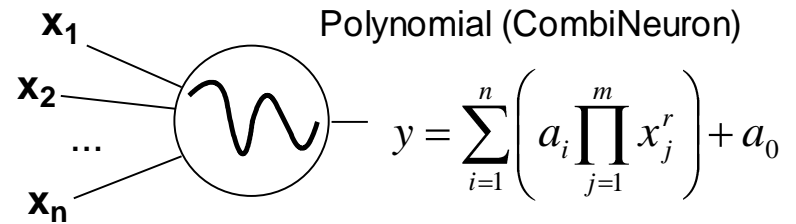
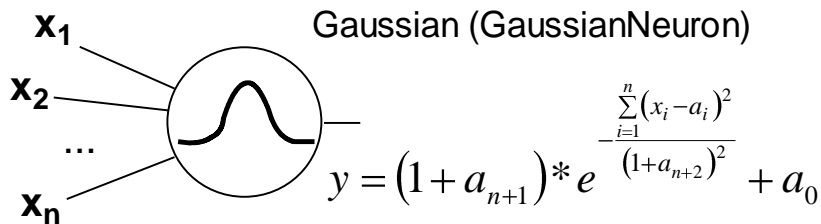
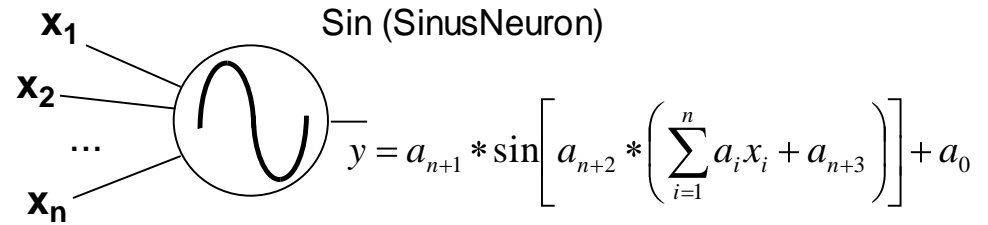
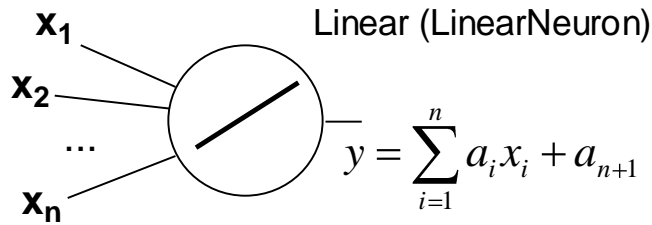
# Housing data – inductive model



# Housing data – inductive model



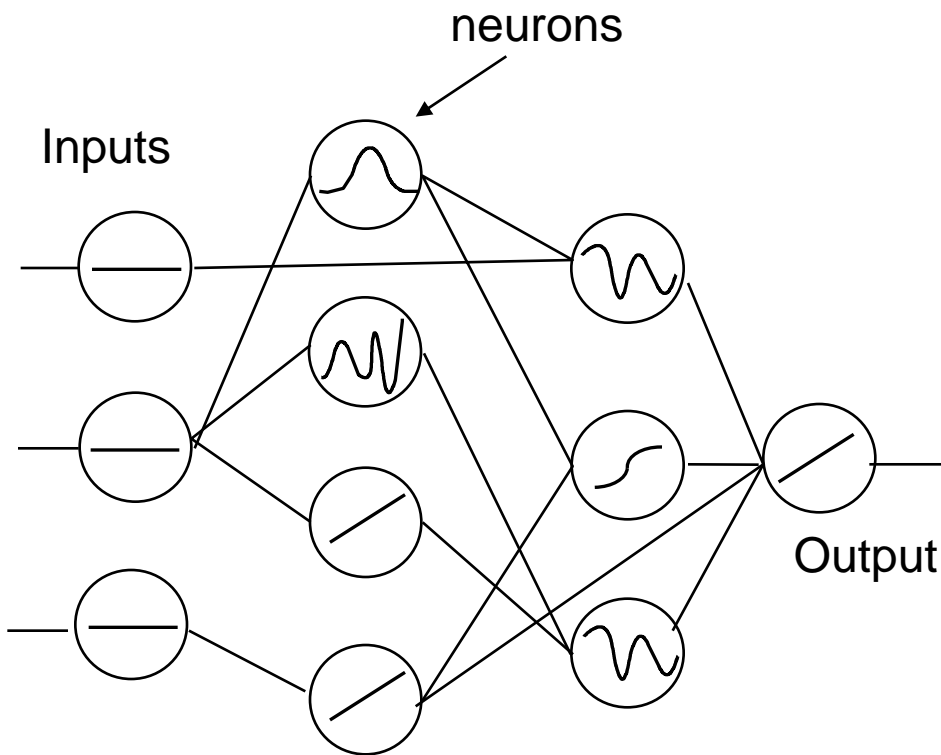
# Heterogeneous units in GAME



# Motivation

- Optimize neural networks with heterogeneous neurons
  - Optimize = adjust coefficients of transfer functions in order to minimize the error
  - Structural optimization (interconnection of neurons) not covered in this presentation
- Goal
  - Develop simple and efficient algorithm to train robust adaptive neural networks

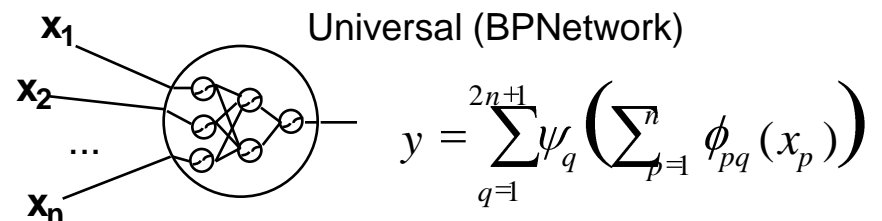
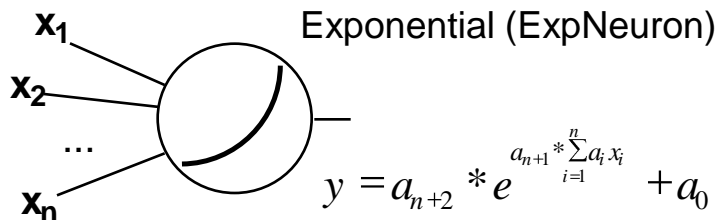
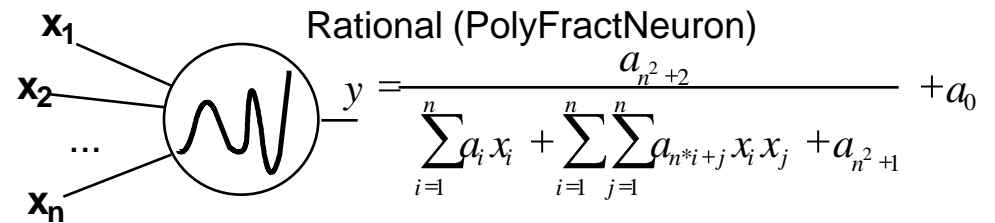
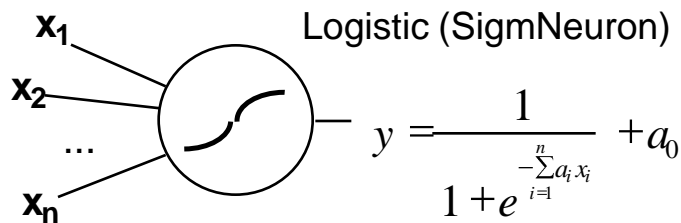
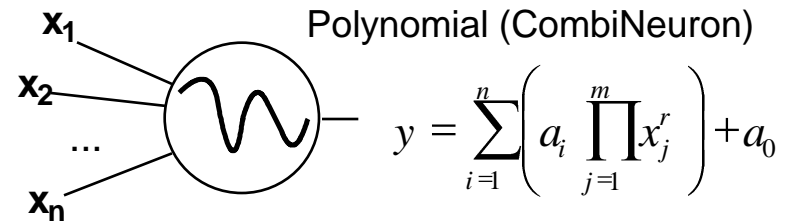
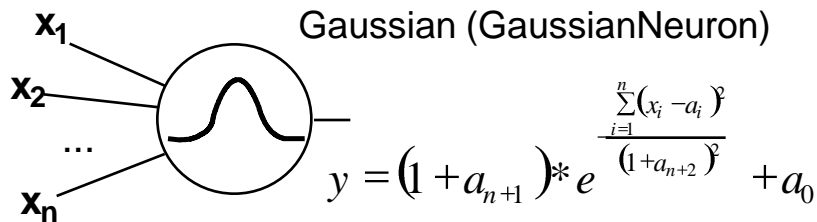
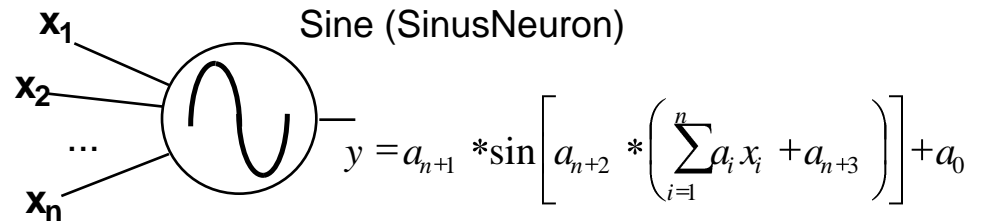
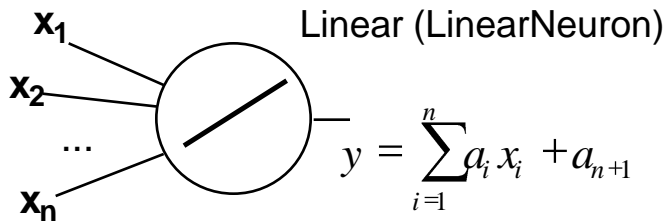
# Hybrid neural network



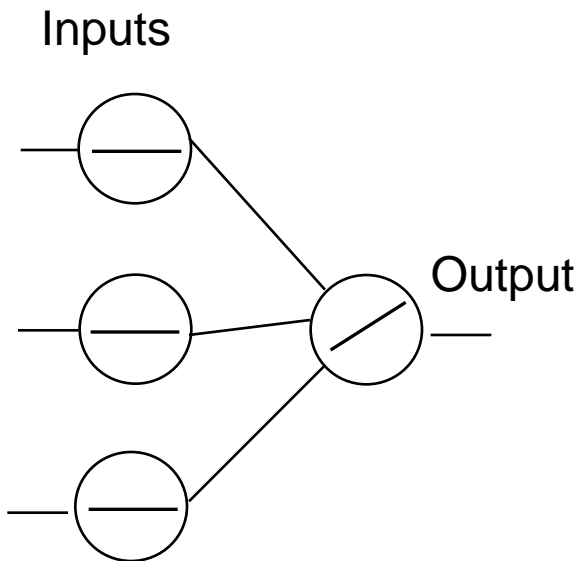
- Layered network of neurons
- Each neuron has a transfer function
- Feedforward
- Operates in two modes:
  - learning
  - recall



# Heterogeneous neurons



# Learning – a trivial network

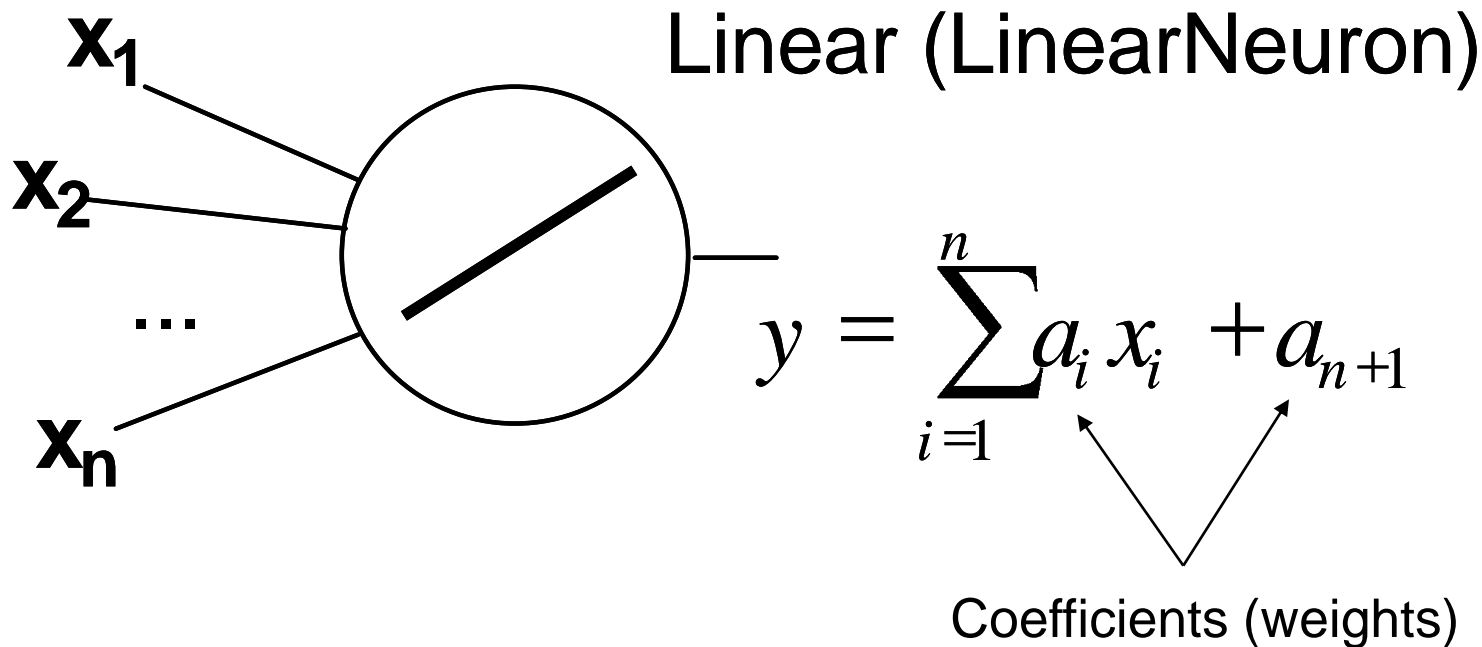


- Learning from data

$x_1$ , $x_2$ , $x_3$	$y$
0.1, 0.3, 0.2	0.6
0.2, 0.3, 0.1	0.8
...	

- Modify coefficients (weights) of neuron in order to minimize *error* of Output

# Optimization of coefficients $a_1, a_2, \dots, a_n$



Error of the Output is sum of residuals, coefficients can be estimated by linear least square method:

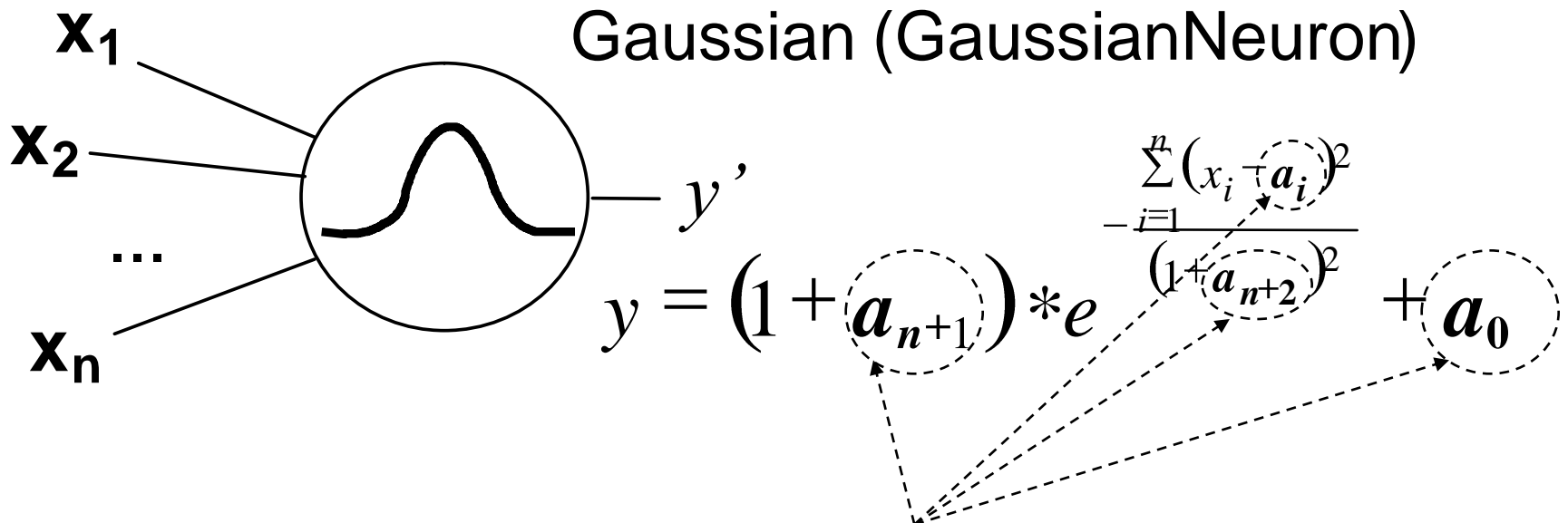
$\hat{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$  , where  $\mathbf{a}$  is vector of coefficients,  $\mathbf{X}$  input data matrix and  $\mathbf{y}$  is target output vector

# LMS algorithm

```
for (int j = 0; j < learning_vectors; j++) x[j][coef-1] = 1;
for (int i = 0; i < coef-1; i++) {
    for (int j = 0; j < learning_vectors; j++) {
        x[j][i] = inputVal[j][i];
    }
}
for (int j = 0; j < learning_vectors; j++) y[j][0] = cValue[j];
```

```
Matrix xx = new Matrix(x);
Matrix xt = xx.transpose();
Matrix yy = new Matrix(y);
xx = xt.times(xx);           //x=xT*x
if(xx.det()==0) return false; //matrix is singular!
xx = xx.inverse();          //power((xT*x):-1)
yy = xt.times(yy);          /// xT*y
Matrix res = xx.times(yy);  //a=(power((xT*x):-1))*xT*y
```

# Optimization of coefficients (non-linear neurons)



We are looking for optimal values of coefficients  $a_0, a_1, \dots, a_{n+2}$

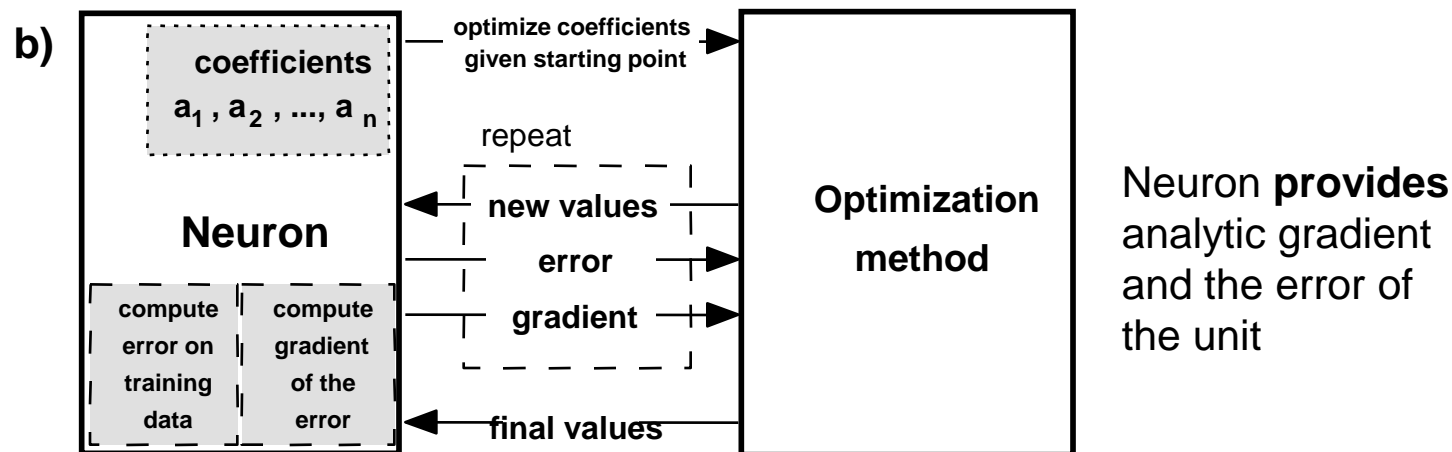
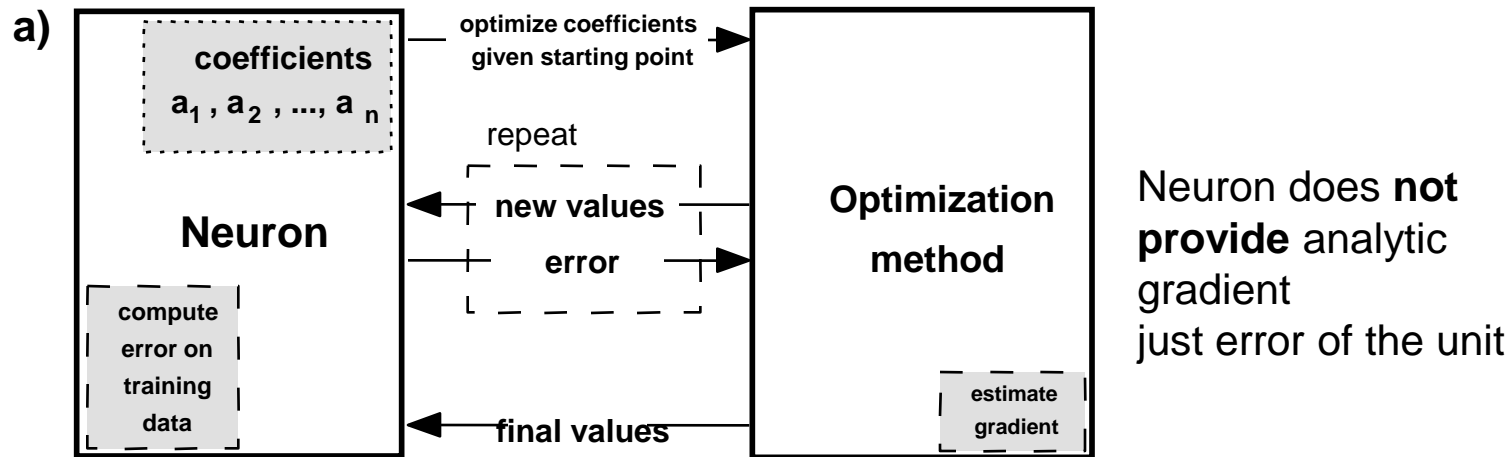
We have inputs  $x_1, x_2, \dots, x_n$  and target output  $y$  in the training data set

The difference between neuron output  $y'$  and the target value  $y$  should be minimal for all vectors from the training data set

Again, error is sum of residuals 
$$E = \sum_{i=1}^m (y' - y)^2$$

# Optimization of non-linear neurons

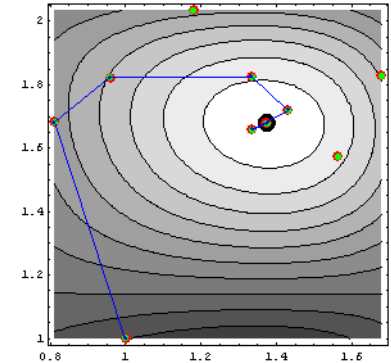
– an iterative method is needed



# What is an analytic gradient of error and how to derive it?

$$E = \sum_{j=0}^m (y_j - d_j)^2, \quad \text{Error of the neuron for training data (energy surface)}$$

$$\nabla \vec{E} = \left( \frac{\partial \vec{E}}{\partial a_1}, \frac{\partial \vec{E}}{\partial a_2}, \dots, \frac{\partial \vec{E}}{\partial a_n} \right), \quad \text{Gradient of the error}$$



$$y_j = (1 + a_{2n+1}) * e \underbrace{\frac{\sum_{i=1}^n (a_i * x_{ij} - a_{n+i})^2}{(1 + a_{2n+2})^2}}_{\rho_j} + a_0 \quad \text{Neuron with gaussian transfer function}$$

$$\frac{\partial \vec{E}}{\partial a_i} = \sum_{j=0}^m \left[ \frac{\partial \vec{E}}{\partial y_j} * \frac{\partial y_j}{\partial \rho_j} * \frac{\partial \rho_j}{\partial a_i} \right], \quad \text{Partial derivation of error in the direction of coefficient } a_i$$

# Partial derivatives of the Gauss neuron

$$y_j = (1 + a_{2n+1}) * e^{\underbrace{\frac{\sum_{i=1}^n (a_i * x_{ij} - a_{n+i})^2}{(1 + a_{2n+2})^2}}_{\rho_j}} + a_0$$

$$\frac{\partial \vec{E}}{\partial a_{2n+1}} = 2 \sum_{j=0}^m \left[ (y_j - d_j) * e^{-\frac{\sum_{i=1}^n (a_i * x_{ij} - a_{n+i})^2}{(1 + a_{2n+2})^2}} \right] \quad \frac{\partial \vec{E}}{\partial a_0} = 2 \sum_{j=0}^m (y_j - d_j)$$

$$\frac{\partial \vec{E}}{\partial a_{2n+2}} = 2 \sum_{j=0}^m \left[ (y_j - d_j) * (1 + a_{2n+1}) e^{\rho_j} * 2 \frac{\sum_{i=1}^n (a_i * x_{ij} - a_{n+i})^2}{(1 + a_{2n+2})^3} \right]$$

$$\frac{\partial \vec{E}}{\partial a_i} = 2 \sum_{j=0}^m \left[ (y_j - d_j) * (1 + a_{2n+1}) e^{\rho_j} * -2 \frac{a_i^2 * x_{ij}^2 - a_{n+i} * x_{ij}}{(1 + a_{2n+2})^2} \right]$$

$$\frac{\partial \vec{E}}{\partial a_{n+i}} = 2 \sum_{j=0}^m \left[ (y_j - d_j) * (1 + a_{2n+1}) e^{\rho_j} * -2 \frac{a_{n+i} - a_i * x_{ij}}{(1 + a_{2n+2})^2} \right].$$

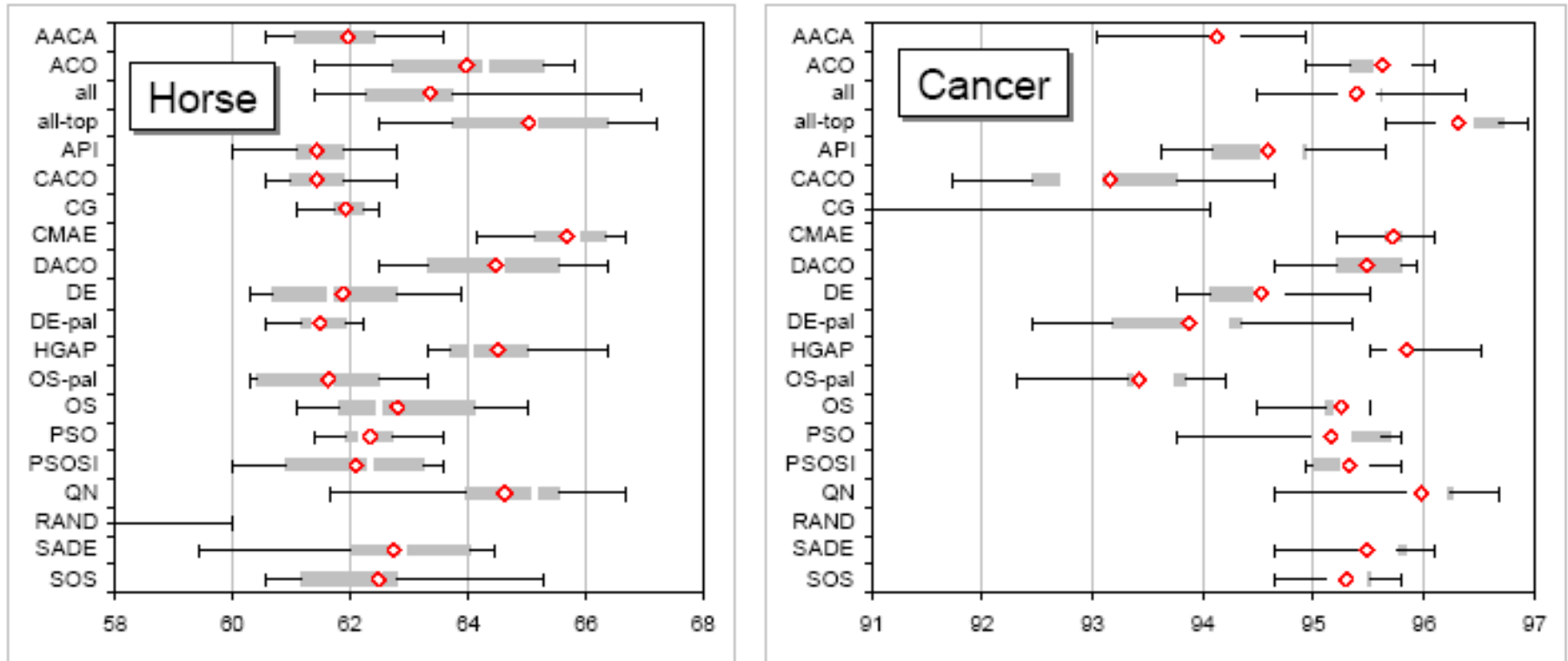


# Now, which optimization method should be used?

- Quasi Newton Method [QuasiNewtonTrainer]
- Quasi Newton - Limited Memory BFGS [QuasiNewtonLimitedMemoryBfgsTrainer]
- SADE - genetics method [SADETrainer]
- Particle Swarm Optimization [PSOTrainer]
- Hybrid of the GA and the PSO [HGAPSOTrainer]
- PAL: Differential Evolution [PALDifferentialEvolutionTrainer]
- Differential Evolution [DifferentialEvolutionTrainer]
- Stochastic Orthogonal Search [StochasticOSearchTrainer]
- Orthogonal Search [OrthogonalSearchTrainer]
- PAL : Conjugate Gradient [PALConjugateGradientTrainer]
- AACA: Adaptive Ant Colony Algorithm [AACATrainer]
- ACO\*: Ant Colony Optimization [ACOTrainer]
- API [APITrainer]
- CACO: Continuous Ant Colony Optimization [CACOTrainer]
- DACO: Direct Ant Colony Algorithm [DACOTrainer]
- Random Search Algorithm [RandomTrainer]
- Powell [PowellTrainer]

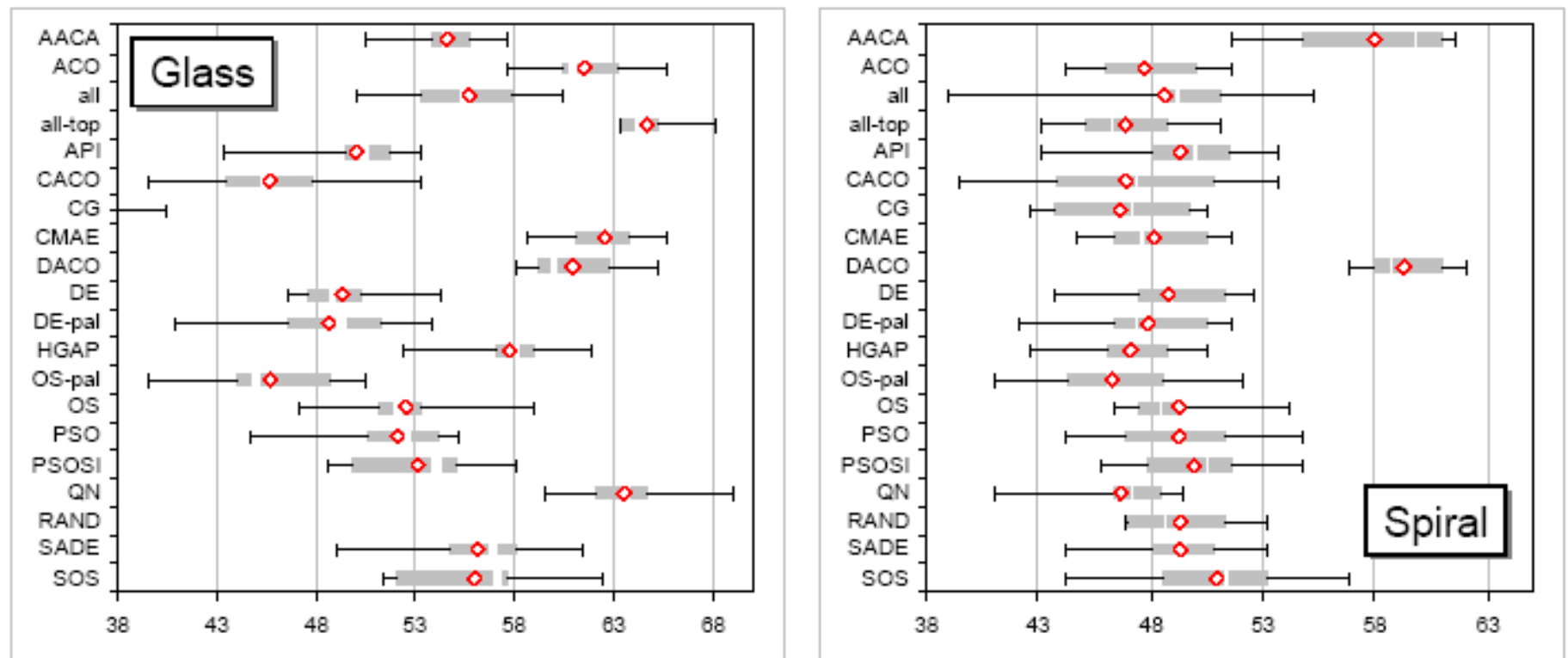
- It depends on
  - dataset
  - type of neuron
  - location of neuron within network
  - configuration of method

# Optimization methods benchmark



**Fig. 4.** Box plots showing classification accuracy of networks optimized by individual algorithms on Horse and Cancer data sets.

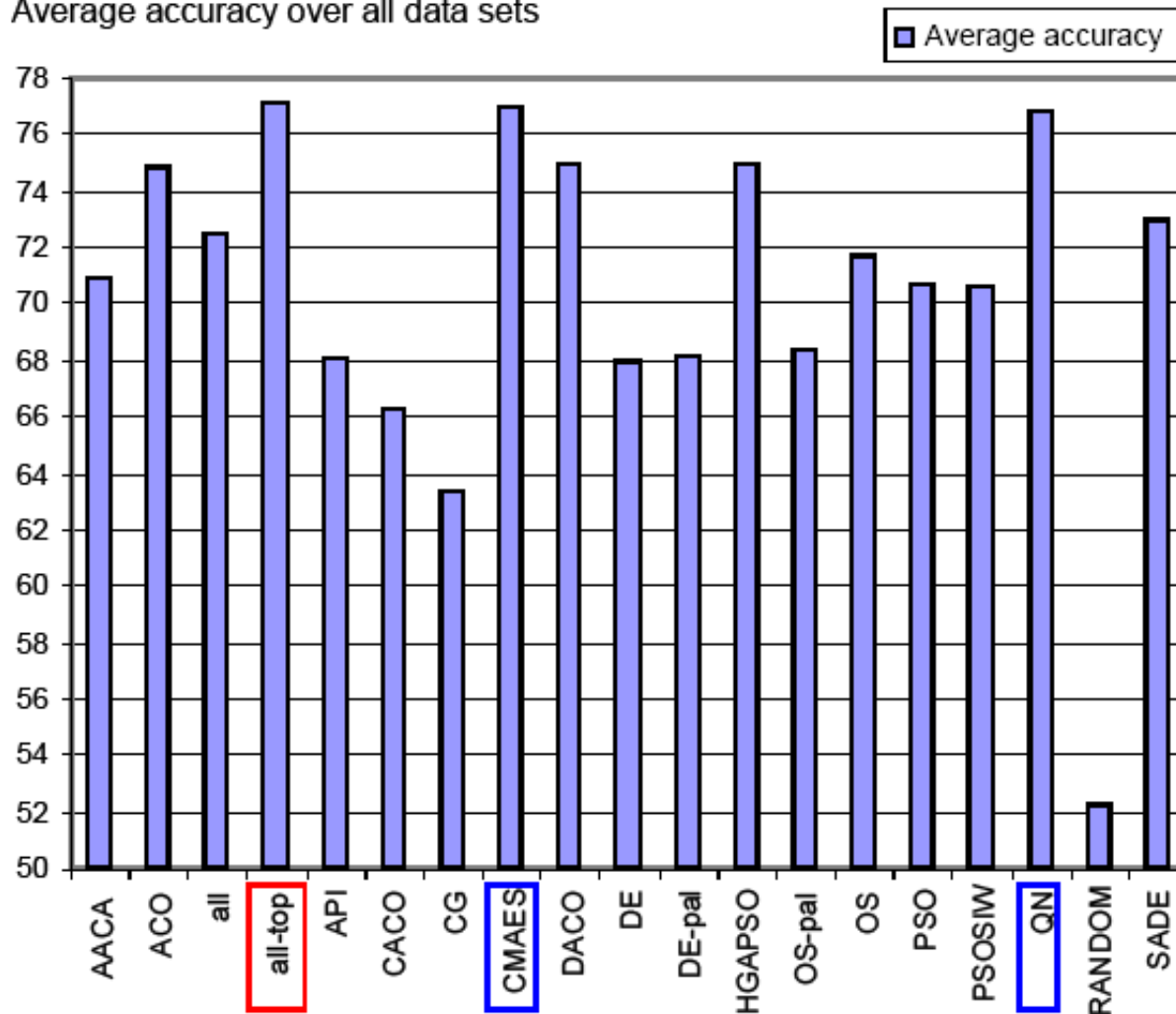
# Optimization methods benchmark, cont.



**Fig. 5.** Box plots showing classification accuracy of networks optimized by individual algorithms on Glass and Spiral data sets.

# Overall results of optimization methods (all data sets)

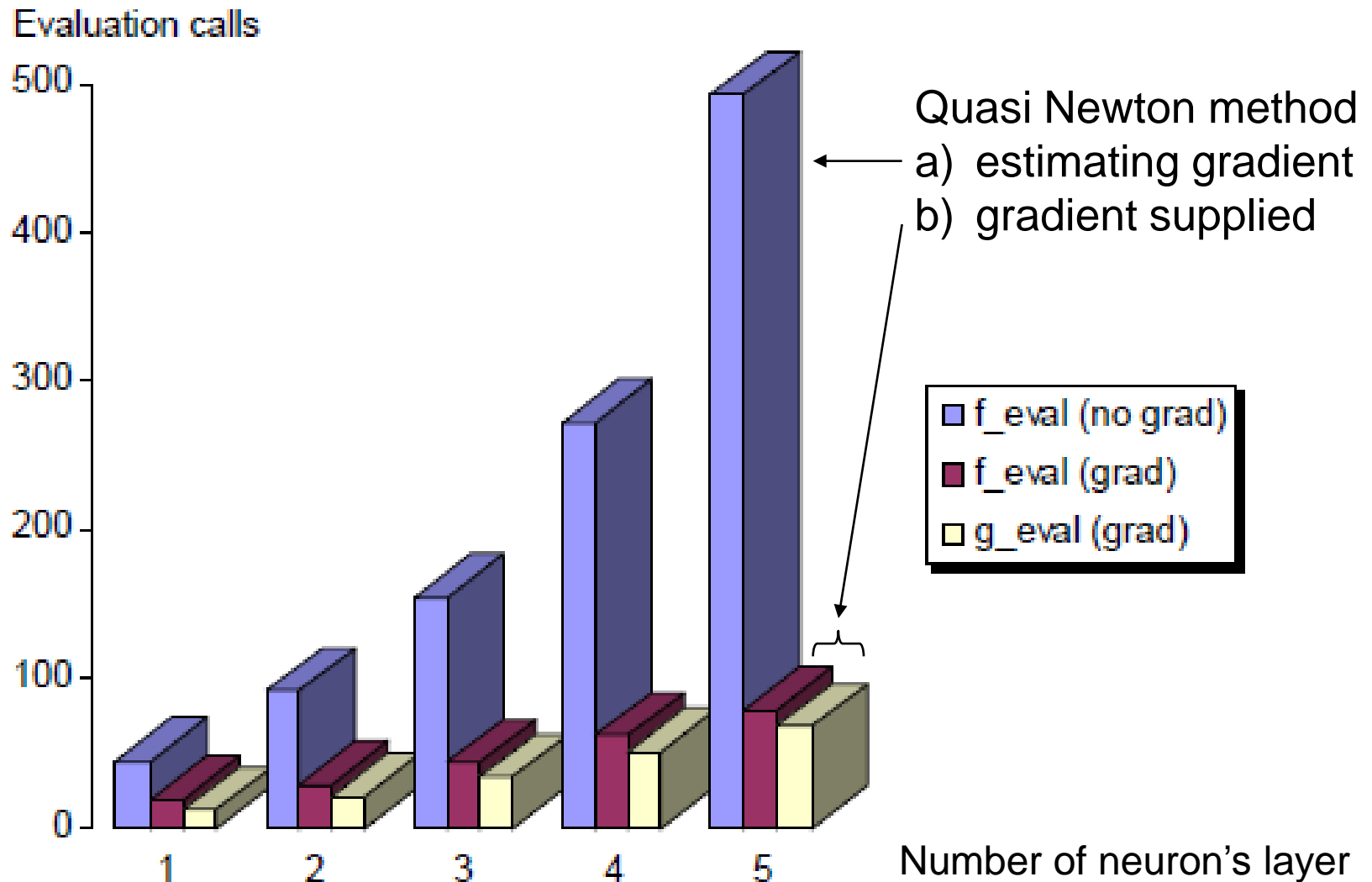
Average accuracy over all data sets



**QN** fastest,  
very accurate

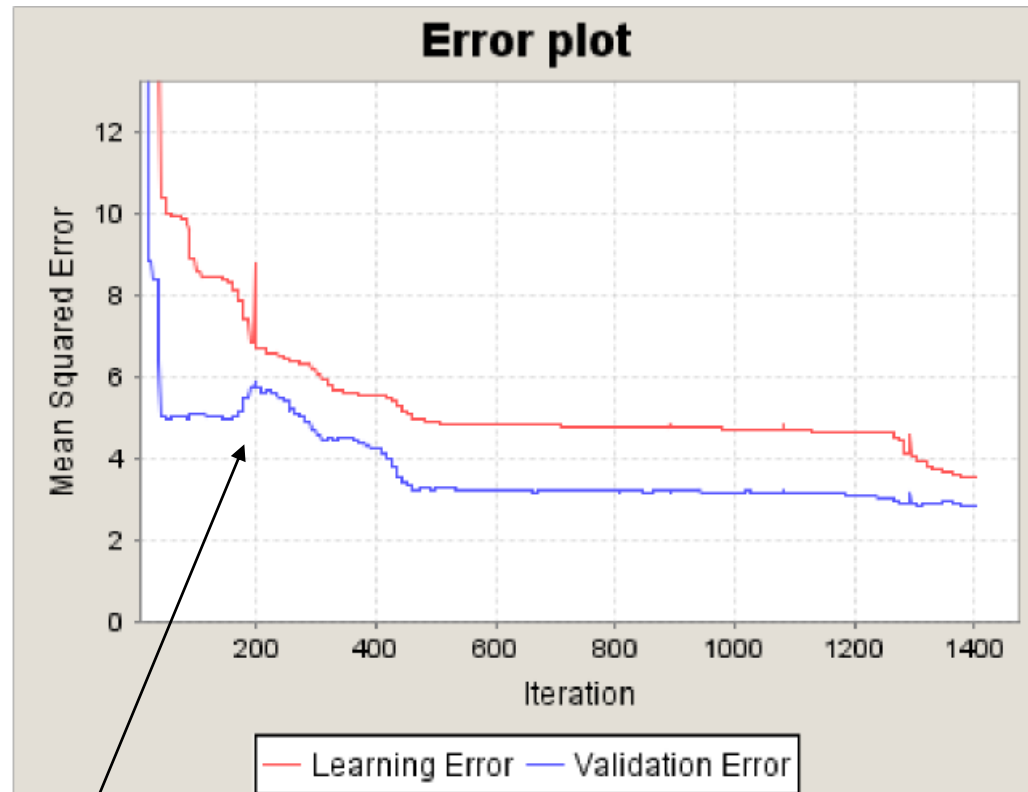
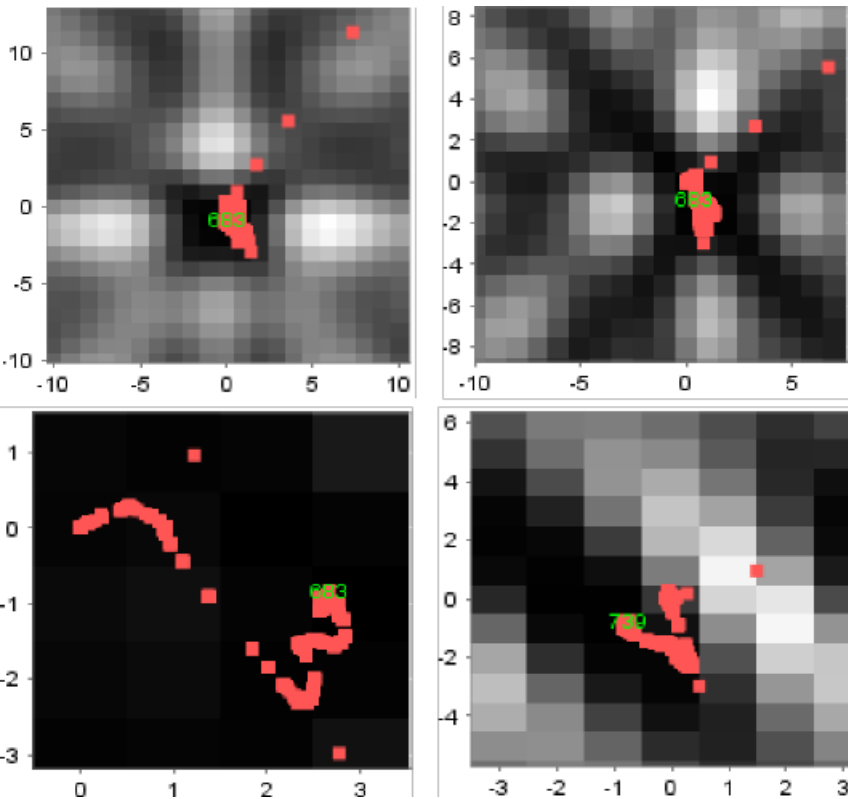
Covariance Matrix Adaptation Evolution Strategy, Quasi-Newton method

# Analytic gradient significantly reduces the number of error evaluation calls



# Monitoring optimization process

(Sine unit, QN numerical estimate of gradient)



Local minima overcame

# Some observations

- Starting point is very important
  - Unknown data distribution (MLE, LMS estimates not very accurate)
  - Better random numbers within an interval
  - Sigmoid neuron (-0.3,0.3) - linear slope
  - Gaussian neuron (variance  $>1$ , otherwise flat error surface)
- Analytic gradient not always better
  - QN from Uncmin library gets stuck in local minima when analytic gradient supplied and converges without

# Conclusion

- QN most efficient for training hybrid neural networks
- Analytic gradient and starting point interval estimate should be computed where possible
- There are still some unsolved problems with robustness of the QN method from Uncmin library, when analytic gradient is supplied