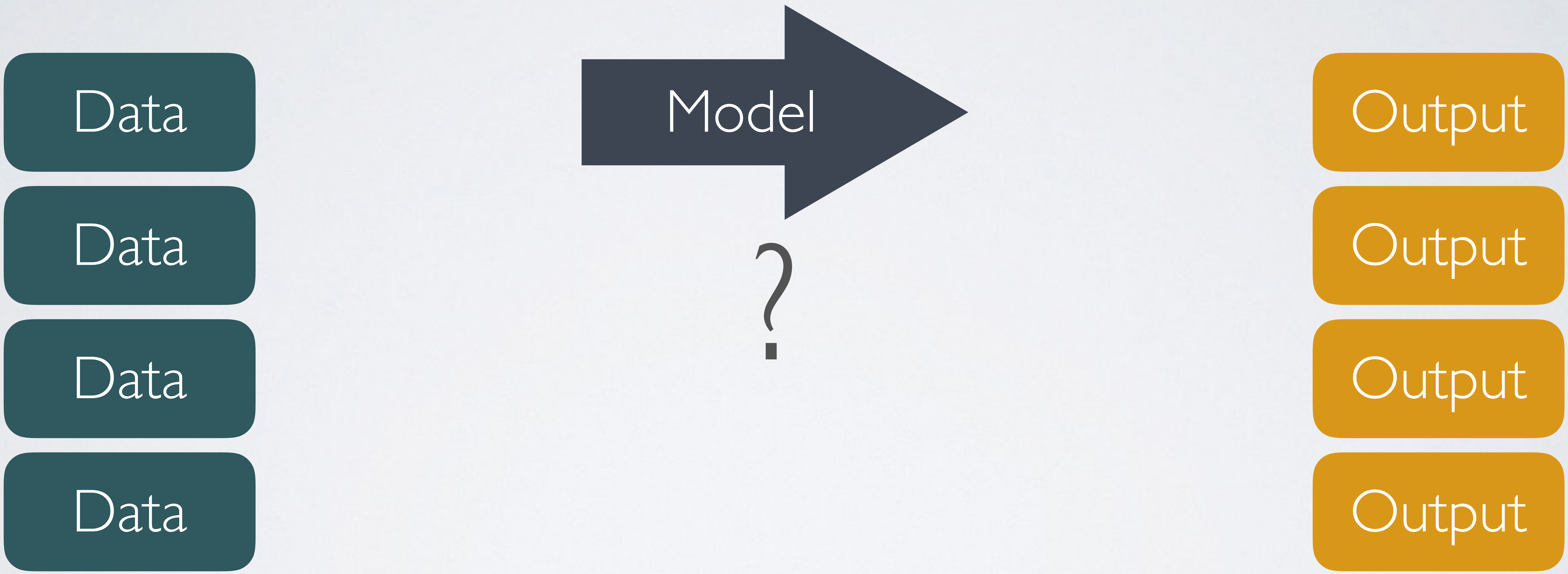
STOCHASTIC GRADIENT DESCENT

Philipp Krähenbühl

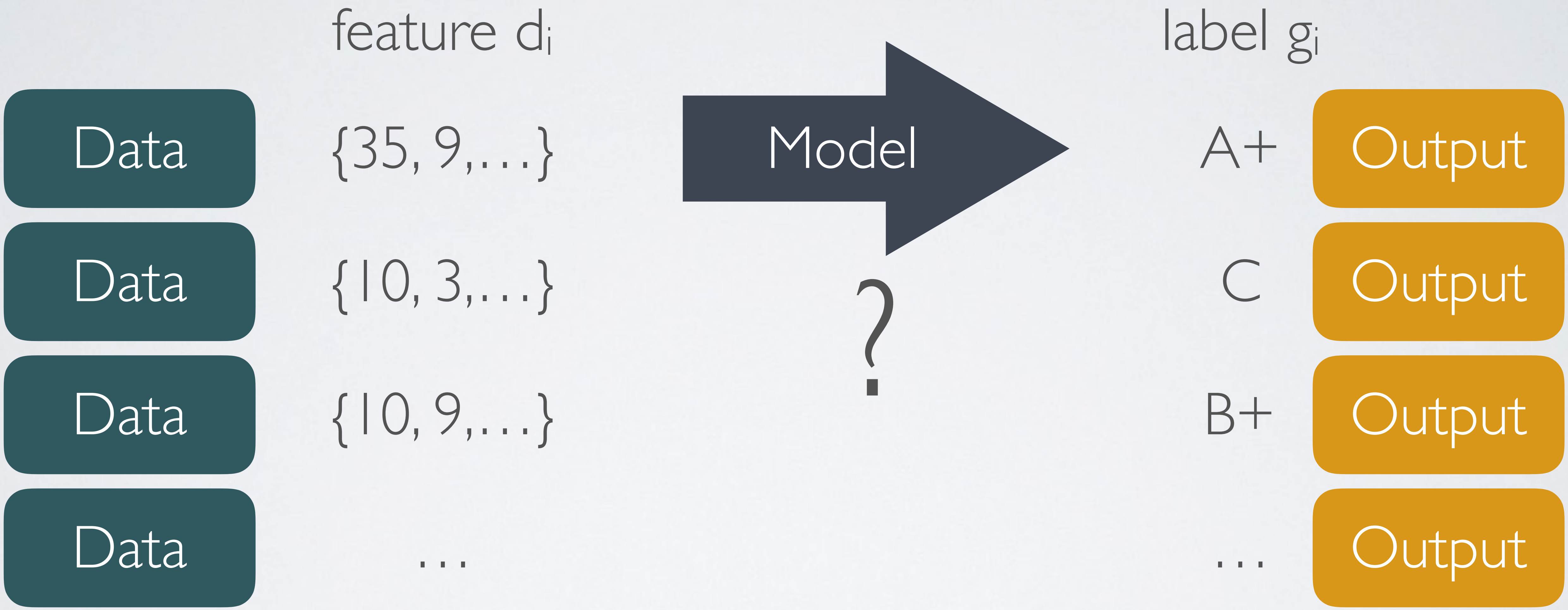
ADMINISTRATIVE

- List of papers due **Thursday 6am**
- Custer access (work in progress)
- Caffe tutorial for cs381v **Sep 12** 5pm-7pm
  - more details to follow

RECAP

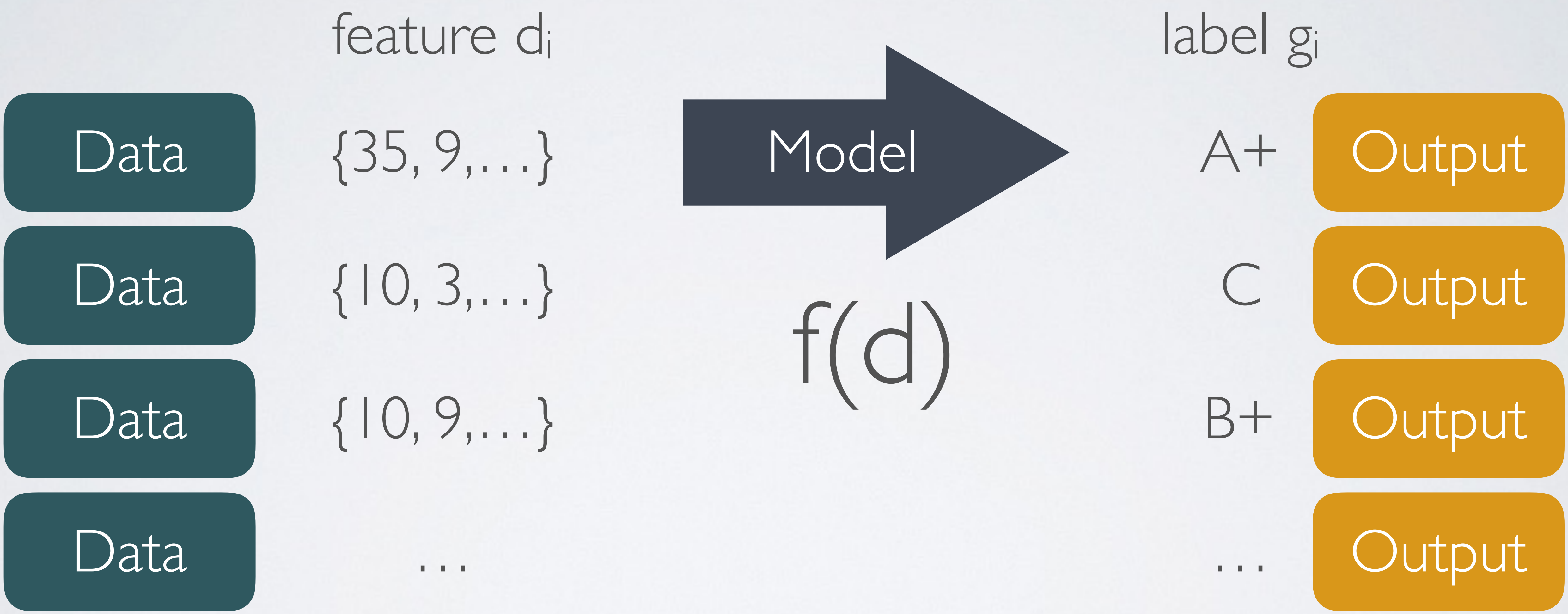


RECAP



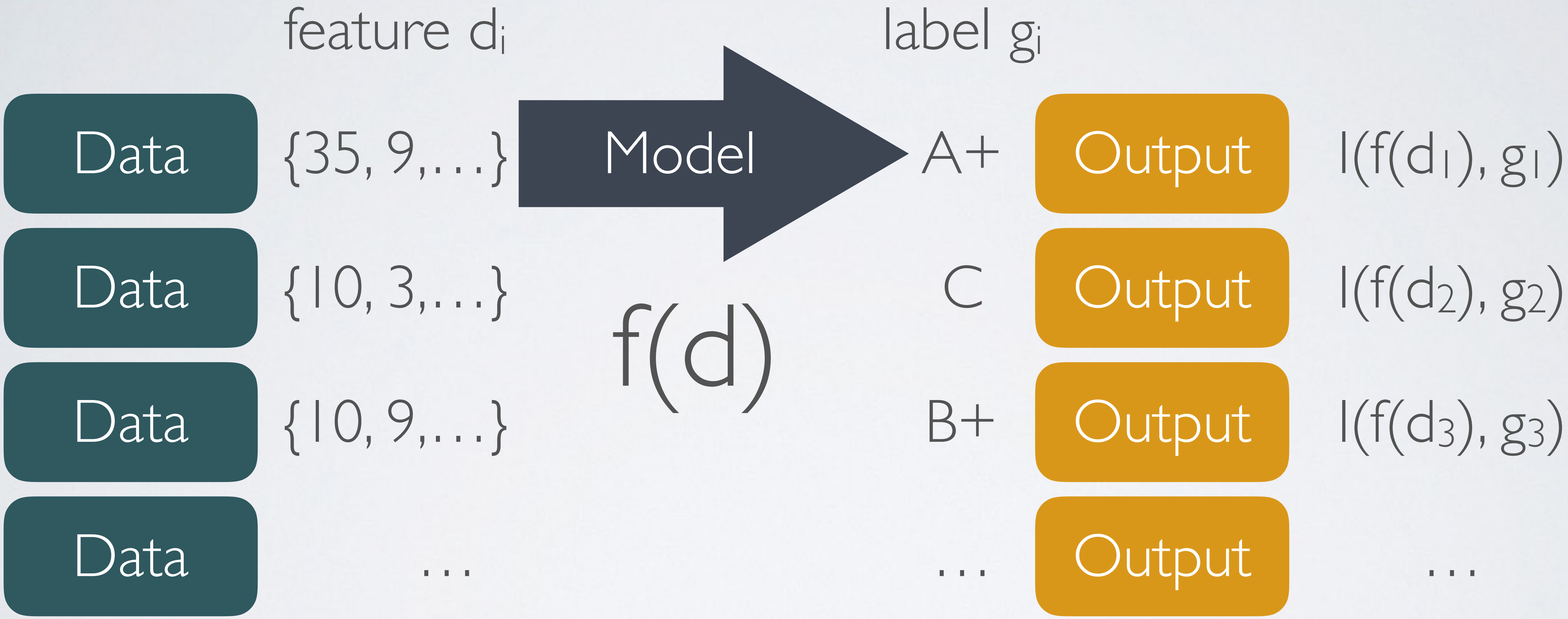
feature d_i
label g_i
Data
{35, 9,...}
Model
?
A+
Output
Data
{10, 3,...}
C
Output
Data
{10, 9,...}
B+
Output
Data
...
...
Output

RECAP



feature d_i
label g_i
Data
{35, 9,…}
Model
A+
Output
Data
{10, 3,…}
f(d)
C
Output
Data
{10, 9,…}
B+
Output
Data
…
…
Output

RECAP



feature d_i
label g_i
Data
{35, 9,…}
Model
A+
Output
$l(f(d_1), g_1)$
Data
{10, 3,…}
C
Output
$l(f(d_2), g_2)$
f(d)
Data
{10, 9,…}
B+
Output
$l(f(d_3), g_3)$
Data
…
…
Output
…

RECAP

$$L = l(f(d_1), g_1) + l(f(d_2), g_2) + l(f(d_3), g_3) + \ldots$$

RECAP

$$L = E_{(d,g)\sim D}[\ l(\ f(\ d\ ),\ g\ )\ ]$$

GRADIENT DESCENT

$$\nabla L = E_{(d,g)\sim D}[\ \nabla l(\ f(\ d\ ), g\ )\ ]$$

$$f = f - \alpha \nabla L$$

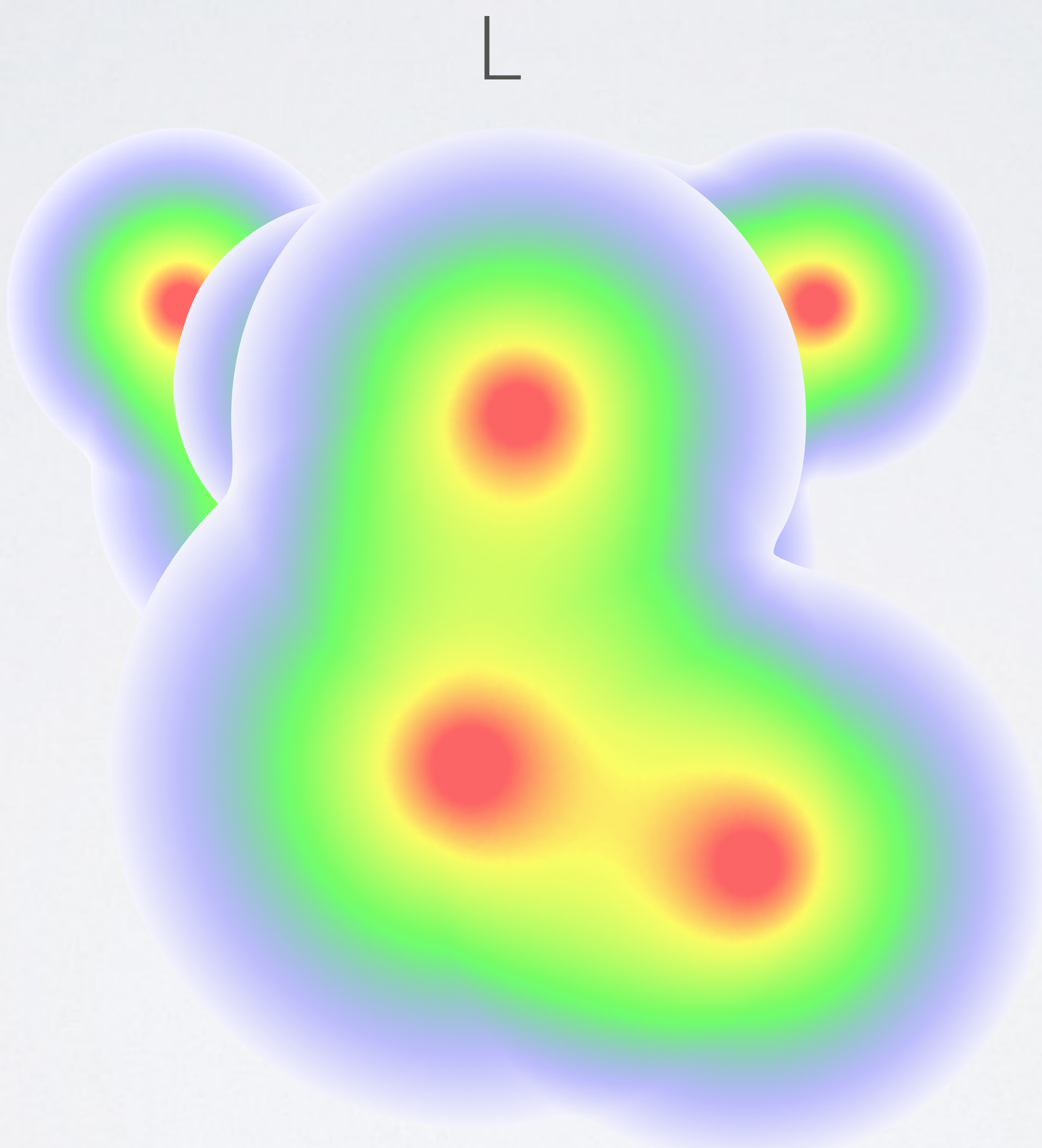
GRADIENT DESCENT

$l(f(d_1), g_1)$

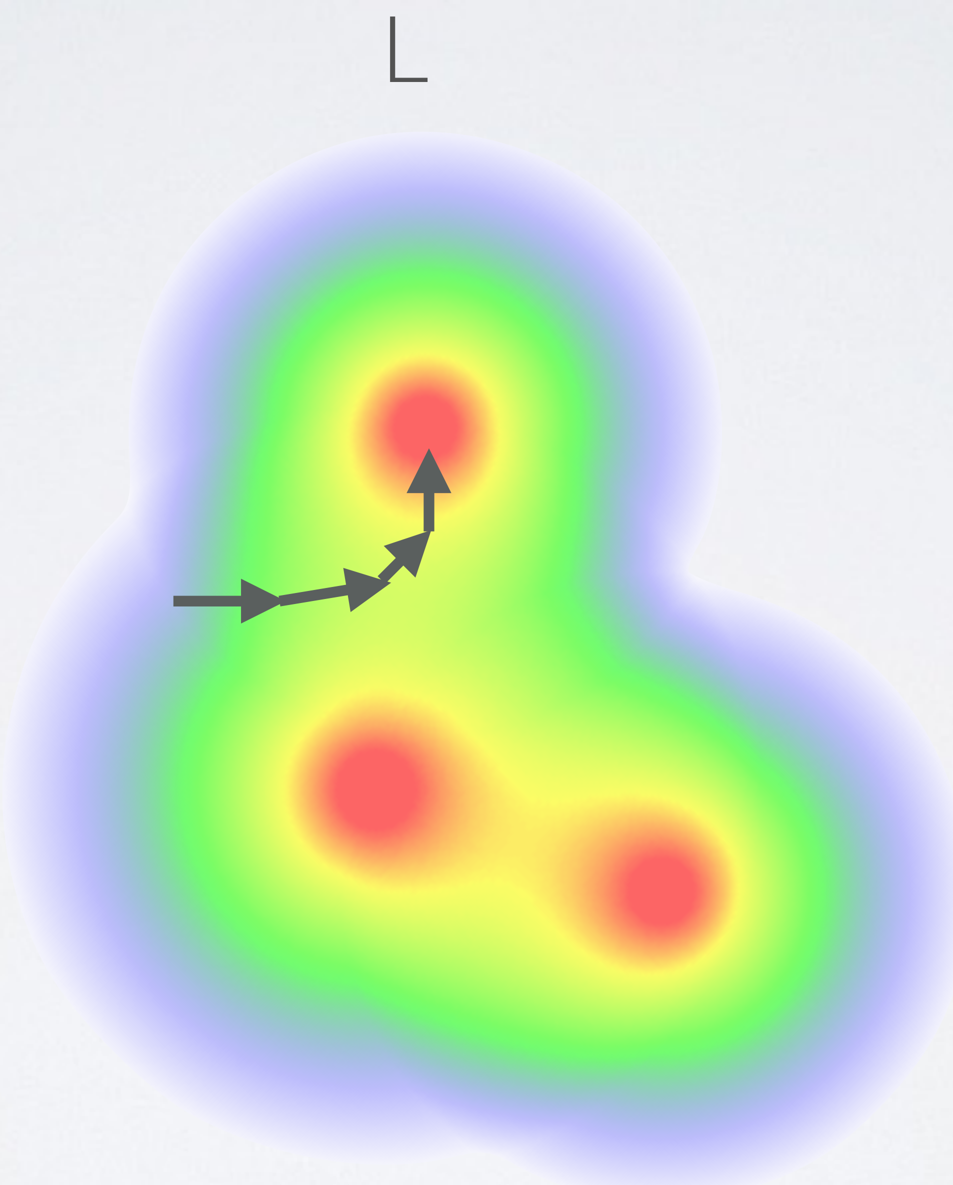
$l(f(d_2), g_2)$

$l(f(d_3), g_3)$

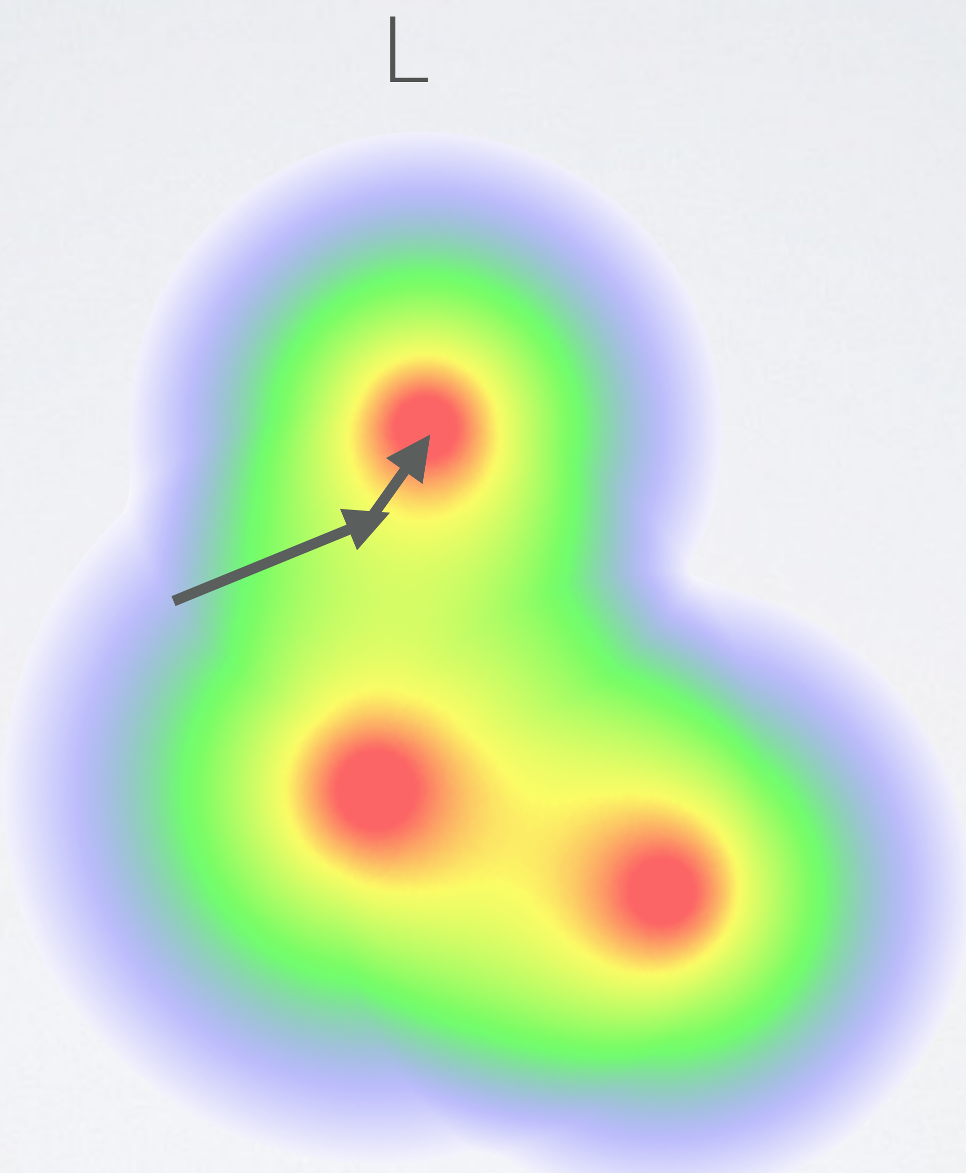
GRADIENT DESCENT



GRADIENT DESCENT

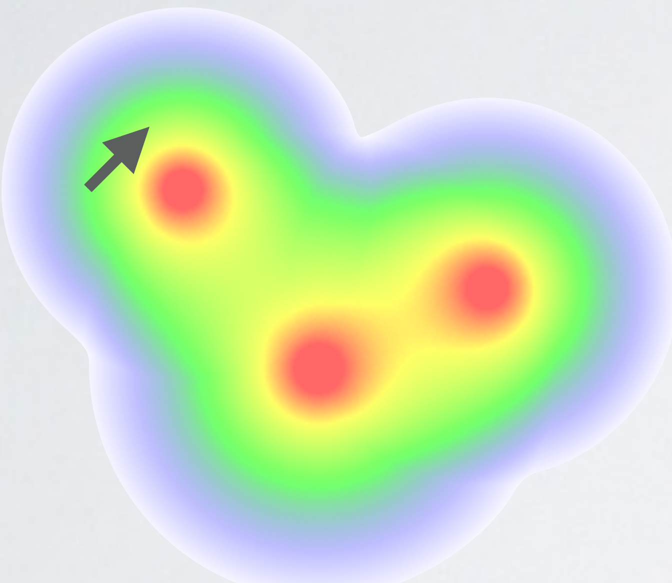


(2ND ORDER) GRADIENT DESCENT



STOCHASTIC GRADIENT DESCENT

$l(f(d_1), g_1)$



$l(f(d_2), g_2)$

$l(f(d_3), g_3)$

STOCHASTIC GRADIENT DESCENT

$$f = f - \gamma \nabla l(\ f(\ d\ ),\ g\ ) \quad \forall_{(d,g) \sim D}$$

SGD ANALYSIS

$$
\begin{aligned}
E(\tilde{f}_n) - E(f^*) &= E(f^*_{\mathcal{F}}) - E(f^*) && \text{Approximation error } (\mathcal{E}_{\mathrm{app}}) \\
&+ E(f_n) - E(f^*_{\mathcal{F}}) && \text{Estimation error } (\mathcal{E}_{\mathrm{est}}) \\
&+ E(\tilde{f}_n) - E(f_n) && \text{Optimization error } (\mathcal{E}_{\mathrm{opt}})
\end{aligned}
$$

Problem:

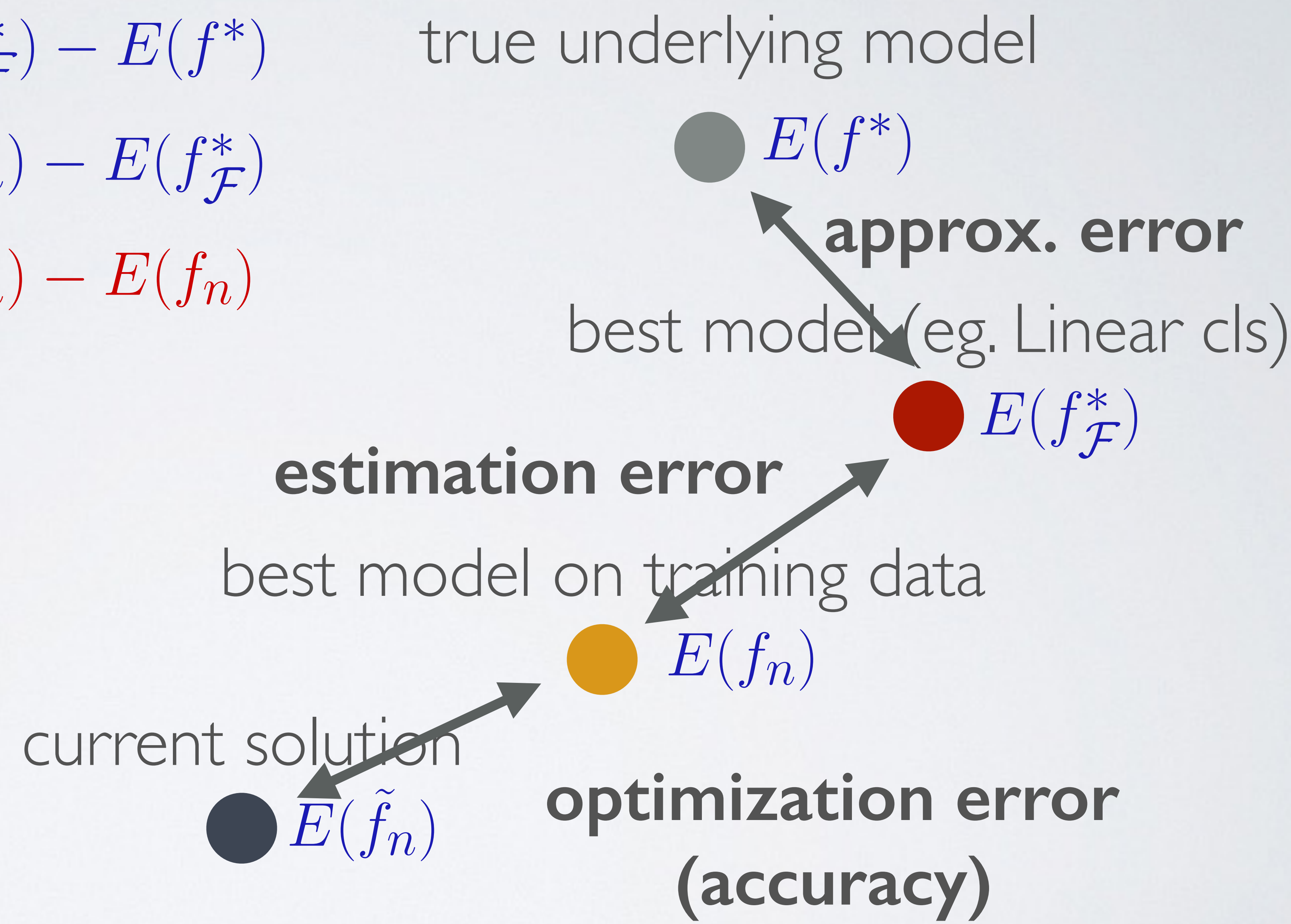
Choose $\mathcal{F}$, n, and ρ to make this as small as possible,

$$
\text{subject to budget constraints} \left\{ \begin{array}{l} \text{max number of examples } n \\ \text{max computing time } T \end{array} \right.
$$

slide credit Bottou 2010

SGD ANALYSIS

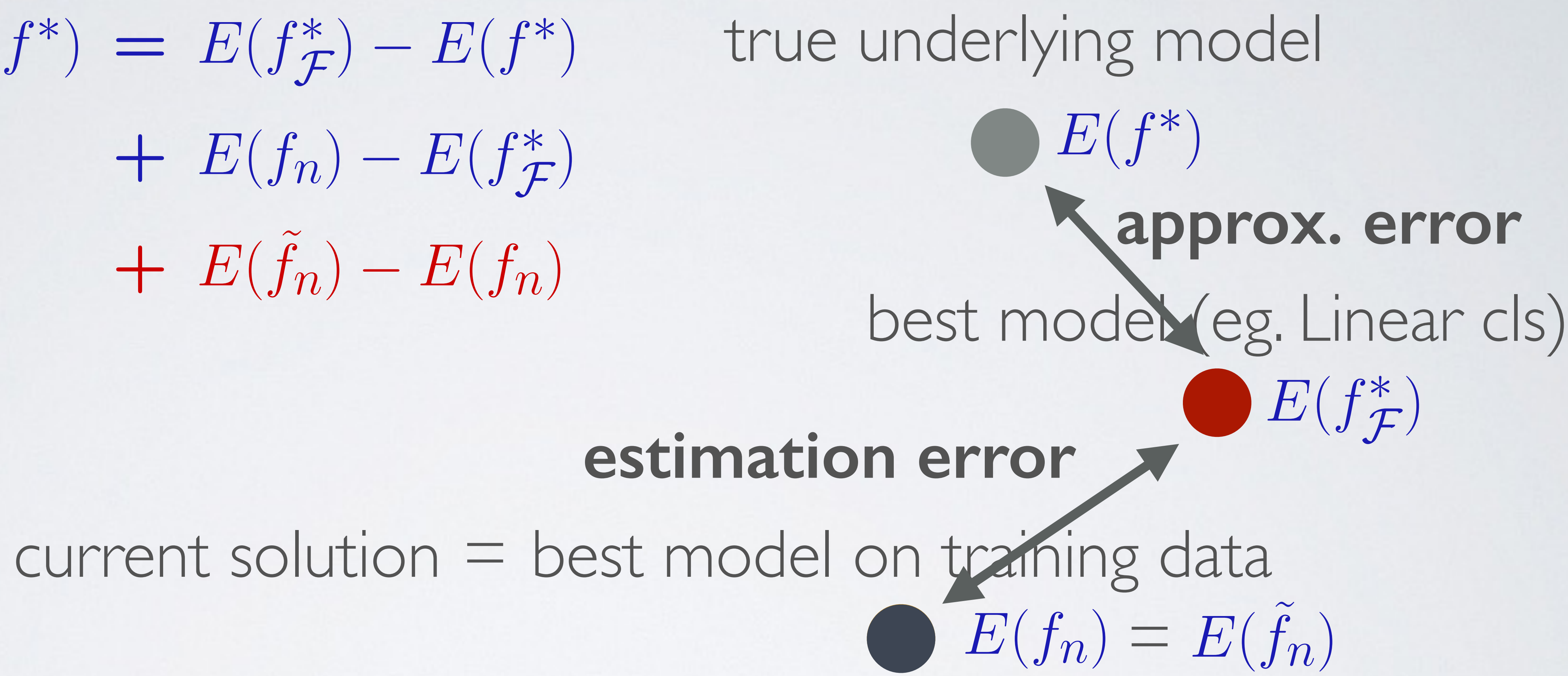
$$
\begin{aligned}
E(\tilde{f}_n) - E(f^*) &= E(f^*_{\mathcal{F}}) - E(f^*) \\
&+ E(f_n) - E(f^*_{\mathcal{F}}) \\
&+ E(\tilde{f}_n) - E(f_n)
\end{aligned}
$$



slide credit Bottou 2010

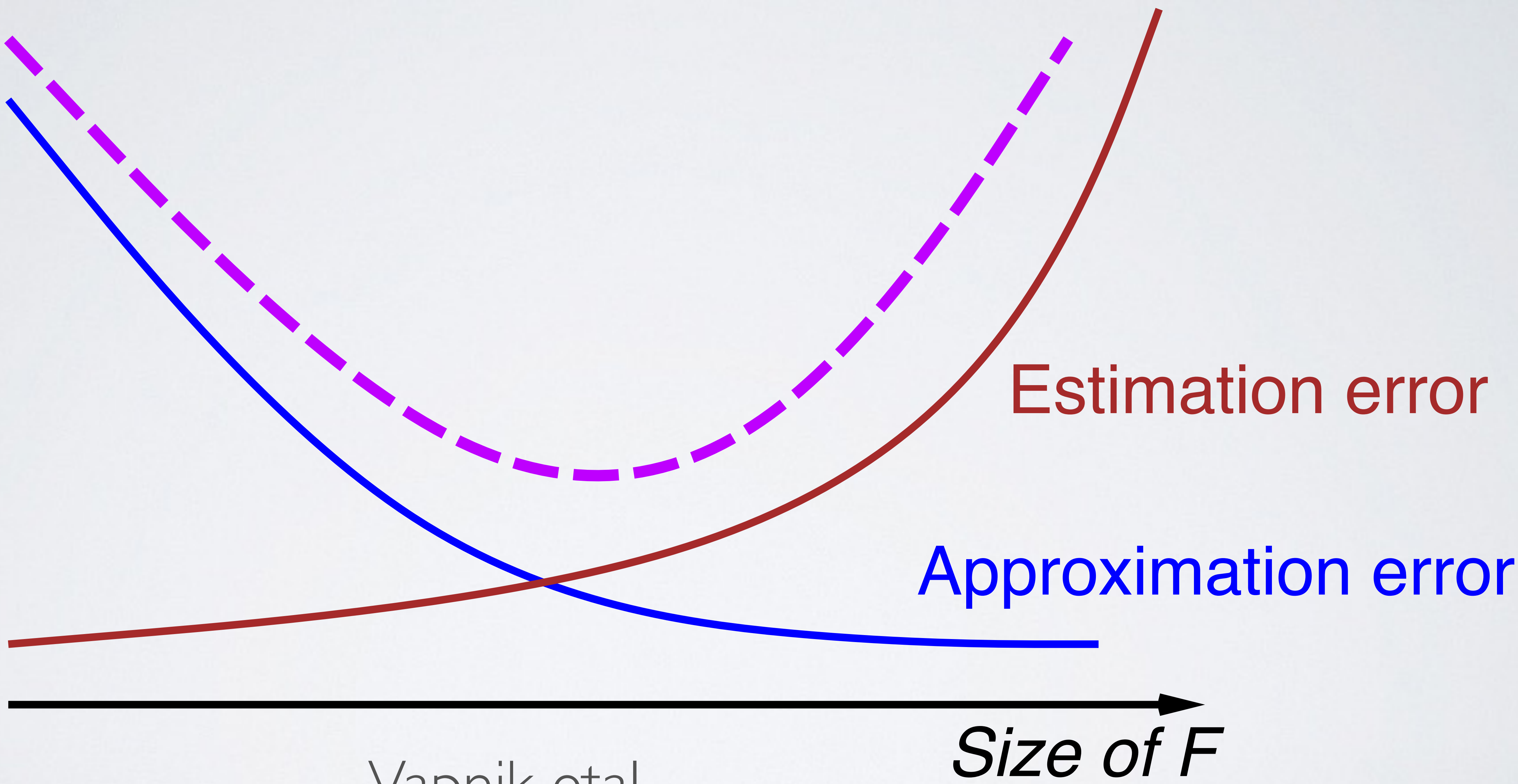
SMALL SCALE LEARNING

$$
\begin{aligned}
E(\tilde{f}_n) - E(f^*) = \; & E(f^*_{\mathcal{F}}) - E(f^*) \\
+ \; & E(f_n) - E(f^*_{\mathcal{F}}) \\
+ \; & E(\tilde{f}_n) - E(f_n)
\end{aligned}
$$



slide credit Bottou 2010

SMALL SCALE LEARNING

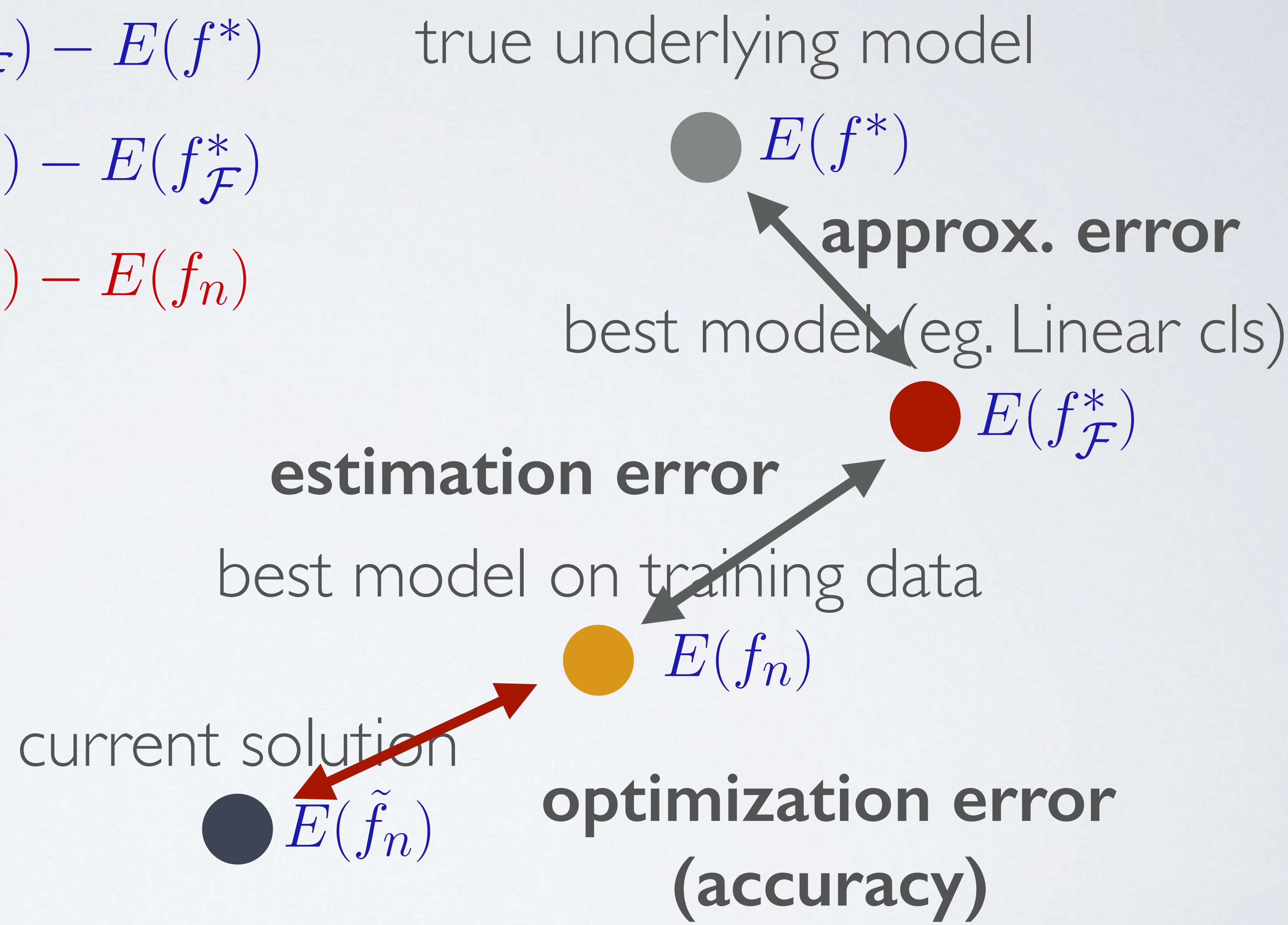


Vapnik etal.

slide credit Bottou 2010

SGD ANALYSIS

$$
\begin{aligned}
E(\tilde{f}_n) - E(f^*) &= E(f^*_{\mathcal{F}}) - E(f^*) \\
&+ E(f_n) - E(f^*_{\mathcal{F}}) \\
&+ E(\tilde{f}_n) - E(f_n)
\end{aligned}
$$



slide credit Bottou 2010

SGD ANALYSIS

$$
\begin{aligned}
E(\tilde{f}_n) - E(f^*) &= E(f^*_{\mathcal{F}}) - E(f^*) && \text{Approximation error } (\mathcal{E}_{\mathrm{app}}) \\
&+ E(f_n) - E(f^*_{\mathcal{F}}) && \text{Estimation error } (\mathcal{E}_{\mathrm{est}}) \\
&+ E(\tilde{f}_n) - E(f_n) && \text{Optimization error } (\mathcal{E}_{\mathrm{opt}})
\end{aligned}
$$

Problem:

Choose $\mathcal{F}$, n, and ρ to make this as small as possible,

subject to budget constraints $\begin{cases} \text{max number of examples } n \\ \text{max computing time } T \end{cases}$

slide credit Bottou 2010

STOCHASTIC GRADIENT DESCENT

	GD	**2GD**	**SGD**
Time per iteration :	n	n	1
Iterations to accuracy ρ :	$\log \frac{1}{\rho}$	$\log \log \frac{1}{\rho}$	$\frac{1}{\rho}$
Time to accuracy ρ :	$n \log \frac{1}{\rho}$	$n \log \log \frac{1}{\rho}$	$\frac{1}{\rho}$
Time to excess error $\mathcal{E}$:	$\frac{1}{\mathcal{E}^{1/\alpha}} \log^2 \frac{1}{\mathcal{E}}$	$\frac{1}{\mathcal{E}^{1/\alpha}} \log \frac{1}{\mathcal{E}} \log \log \frac{1}{\mathcal{E}}$	$\frac{1}{\mathcal{E}}$

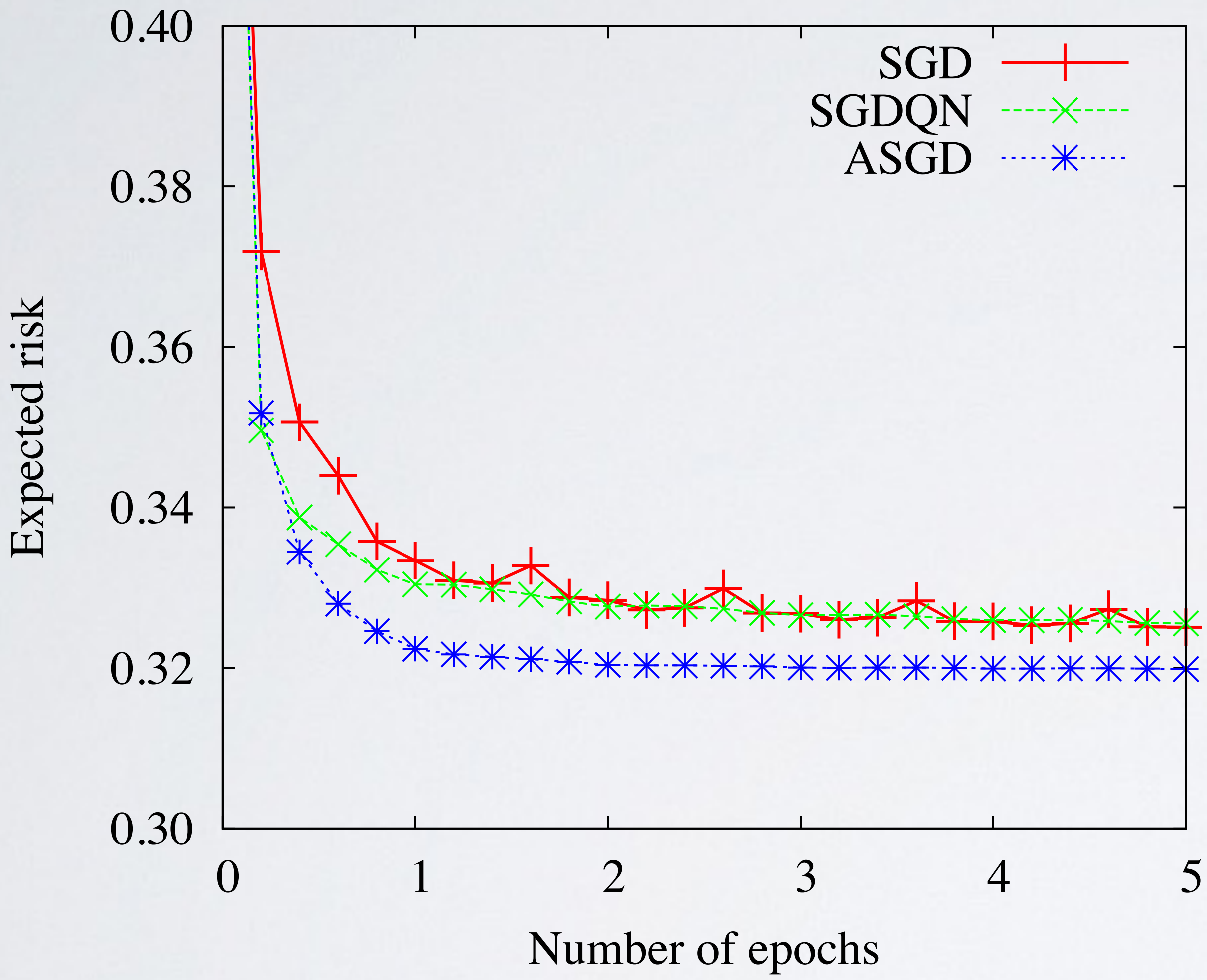
STOCHASTIC GRADIENT DESCENT - EXAMPLES

Loss	Stochastic gradient algorithm
Adaline (Widrow and Hoff, 1960) $Q_{\mathrm{adaline}} = \frac{1}{2}\left(y - w^\top \Phi(x)\right)^2$ $\Phi(x) \in \mathbb{R}^d,\ y = \pm 1$	$w \leftarrow w + \gamma_t \left(y_t - w^\top \Phi(x_t)\right) \Phi(x_t)$
Perceptron (Rosenblatt, 1957) $Q_{\mathrm{perceptron}} = \max\{0, -y\, w^\top \Phi(x)\}$ $\Phi(x) \in \mathbb{R}^d,\ y = \pm 1$	$w \leftarrow w + \gamma_t \begin{cases} y_t\, \Phi(x_t) & \text{if } y_t\, w^\top \Phi(x_t) \le 0 \\ 0 & \text{otherwise} \end{cases}$
K-Means (MacQueen, 1967) $Q_{\mathrm{kmeans}} = \min\limits_k \frac{1}{2}(z - w_k)^2$ $z \in \mathbb{R}^d,\ w_1 \dots w_k \in \mathbb{R}^d$ $n_1 \dots n_k \in \mathbb{N},\ \text{initially } 0$	$k^* = \arg\min_k (z_t - w_k)^2$ $n_{k^*} \leftarrow n_{k^*} + 1$ $w_{k^*} \leftarrow w_{k^*} + \frac{1}{n_{k^*}}(z_t - w_{k^*})$

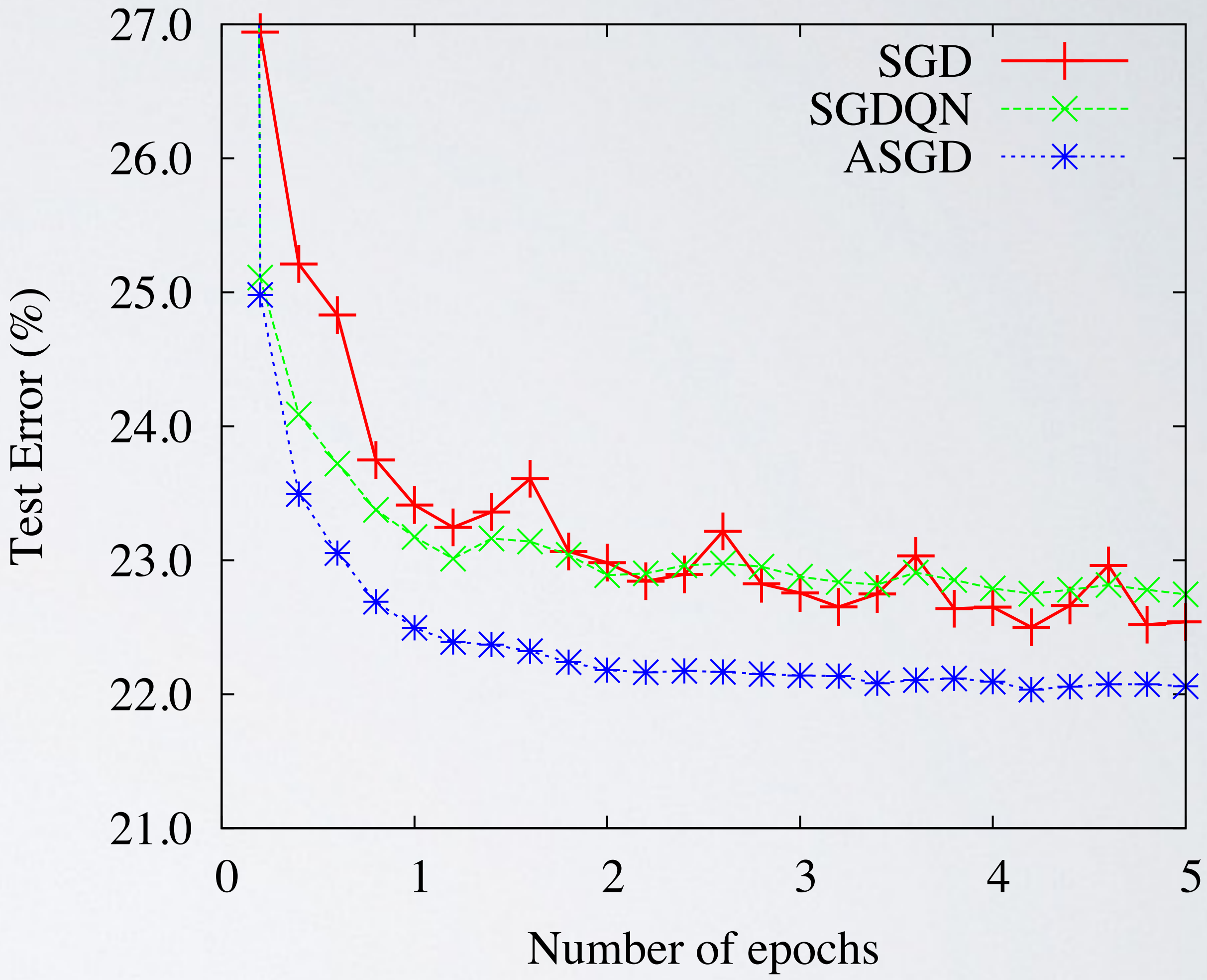
STOCHASTIC GRADIENT DESCENT - EXAMPLES

SVM (Cortes and Vapnik, 1995) $Q_{\mathrm{svm}} = \lambda w^2 + \max\{0, 1 - y\, w^\top \Phi(x)\}$ $\Phi(x) \in \mathbb{R}^d,\ y = \pm 1,\ \lambda > 0$	$w \leftarrow w - \gamma_t \begin{cases} \lambda w & \text{if } y_t\, w^\top \Phi(x_t) > 1, \\ \lambda w - y_t\, \Phi(x_t) & \text{otherwise.} \end{cases}$
Lasso (Tibshirani, 1996) $Q_{\mathrm{lasso}} = \lambda \lvert w \rvert_1 + \frac{1}{2}\big(y - w^\top \Phi(x)\big)^2$ $w = (u_1 - v_1, \ldots, u_d - v_d)$ $\Phi(x) \in \mathbb{R}^d,\ y \in \mathbb{R},\ \lambda > 0$	$u_i \leftarrow \big[u_i - \gamma_t\big(\lambda - (y_t - w^\top \Phi(x_t))\Phi_i(x_t)\big)\big]_+$ $v_i \leftarrow \big[v_i - \gamma_t\big(\lambda + (y_t - w_t^\top \Phi(x_t))\Phi_i(x_t)\big)\big]_+$ with notation $[x]_+ = \max\{0, x\}$.

RESULTS



Expected risk
0.40
0.38
0.36
0.34
0.32
0.30
0
1
2
3
4
5
Number of epochs
SGD
SGDQN
ASGD



Test Error (%)
27.0
26.0
25.0
24.0
23.0
22.0
21.0
0
1
2
3
4
5
Number of epochs
SGD
SGDQN
ASGD

RESULTS

Algorithm	**Time**	**Test Error**
Hinge loss SVM, $\lambda = 10^{-4}$.		
SVMLight	23,642 s.	6.02 %
SVMPerf	66 s.	6.03 %
SGD	**1.4** s.	6.02 %
Log loss SVM, $\lambda = 10^{-5}$.		
TRON (`-e0.01`)	30 s.	5.68 %
TRON (`-e0.001`)	44 s.	5.70 %
SGD	**2.3** s.	5.66 %

PRACTICAL CONSIDERATIONS

- Mini - batches
  - reduces variance
- Momentum
  - further reduces variance

SGD FOR DEEP LEARNING

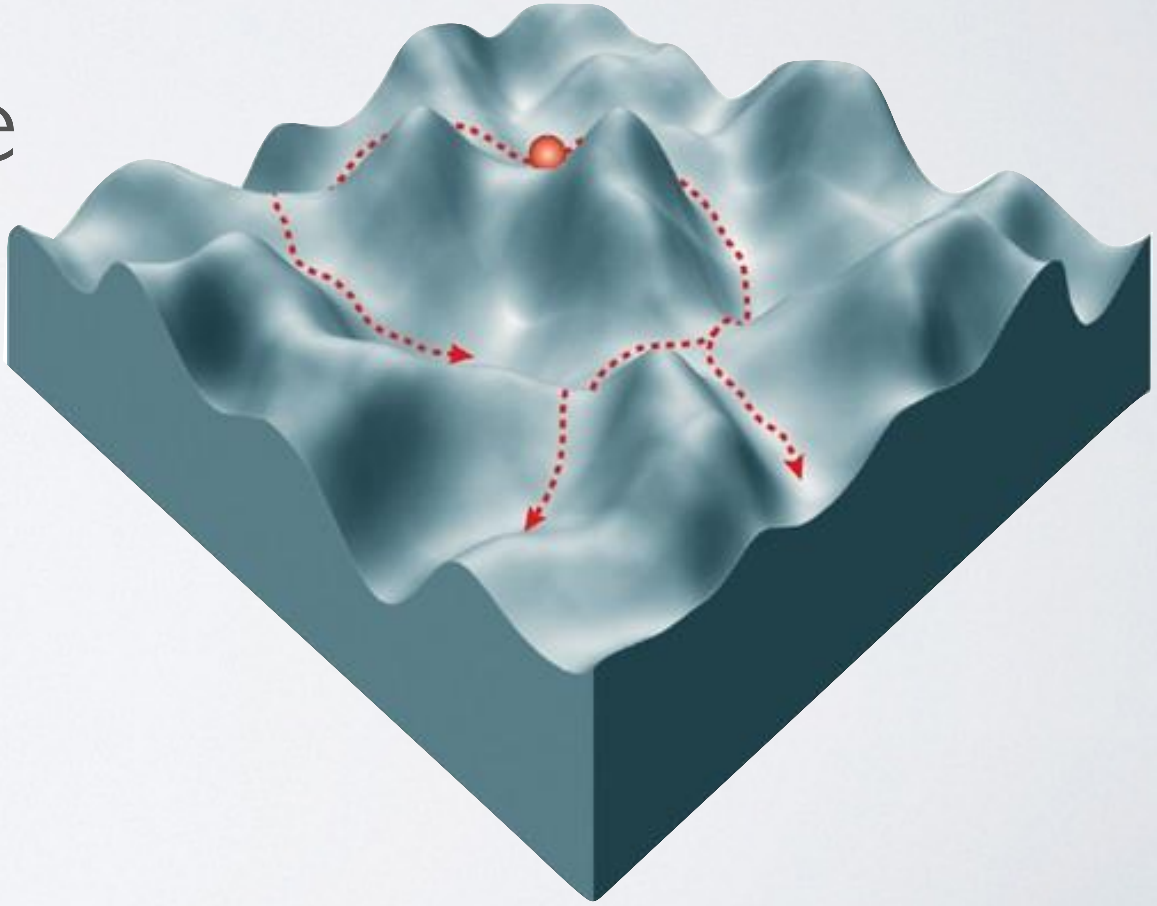
- Faster training
  - a lot of data!

SGD FOR DEEP LEARNING

- Less overfitting
  - shorter training
  - fewer passes over data

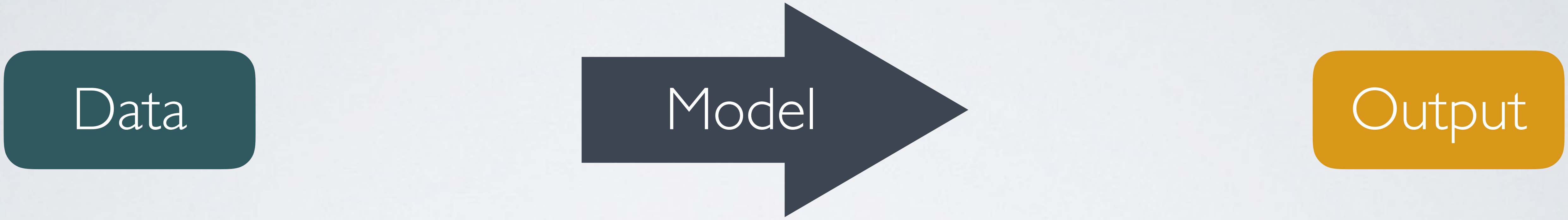
SGD FOR DEEP LEARNING

- Saddle points
  - exponentially more saddle points than local minima
  - GD gets stuck on saddle points



DISCUSSION

PAPER ASSIGNMENT



PAPER ASSIGNMENT

Data

PAPER ASSIGNMENT

Data

S1	S2	S3	S4
P4	P3	P1	P7
P6	P6	P2	P6
P8	P7	P3	P5
...	...	...	...

PAPER ASSIGNMENT

Data

	S1	S2	S3	S4
P1	3	3	1	5
P2	4	2	2	3
P3	1	1	3	2
…	…	…	…	…

PAPER ASSIGNMENT

Output

PAPER ASSIGNMENT

Output

S1	S2	S3	S4
P4	P3	P1	P7

PAPER ASSIGNMENT

Output

	S1	S2	S3	S4
P1	1	0	0	0
P2	0	0	0	1
P3	0	1	0	0
...				

PAPER ASSIGNMENT

Output

	S1	S2	S3	S4
P1	1	0	0	0
P2	0	0	0	1
P3	0	1	0	0
…				

sum up to 1

PAPER ASSIGNMENT

Output

	S1	**S2**	**S3**	**S4**	
P1	1	0	0	0	sum up to 1
P2	0	0	0	1	
P3	0	1	0	0	
...					

RELAXATION

- Keep summation constraints
- Relax $\{0,1\}$ to $[0,1]$

PAPER ASSIGNMENT

Output

	S1	S2	S3	S4	
P1	0.9	0	0	0.1	sum up to 1
P2	0	0.1	0	0.9	
P3	0.1	0.9	0	0	
...					

sum up to 1

DOUBLY STOCHASTIC

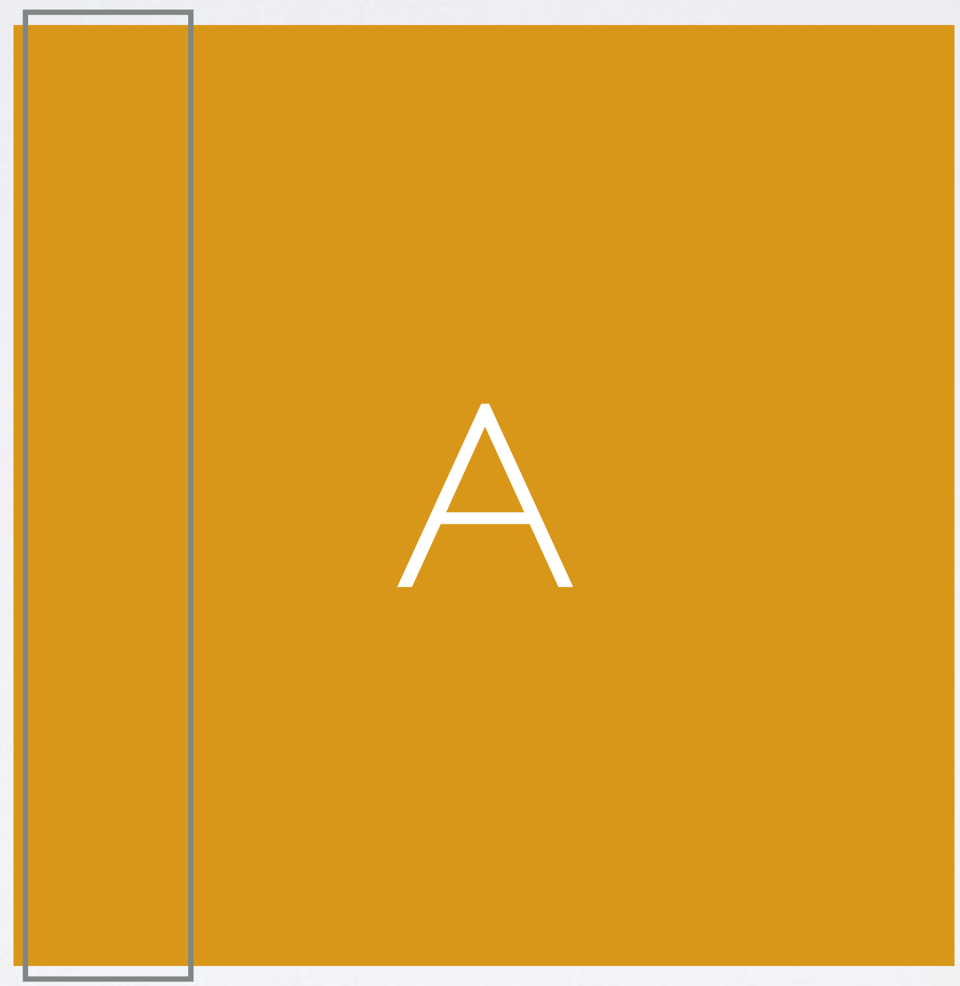
- Solution 1: Constraints
  - linear program
  - not differentiable
  - how to train?

DOUBLY STOCHASTIC

- Solution 2: Approximation

DOUBLY STOCHASTIC

- Solution 2: Approximation



sum and divide

DOUBLY STOCHASTIC

- Solution 2: Approximation
  - Differentiable
  - Trainable with SGD

PAPER ASSIGNMENT

Data

	S1	S2	S3	S4
P1	3	3	1	5
P2	4	2	2	3
P3	1	1	3	2
…	…	…	…	…

Input matrix D

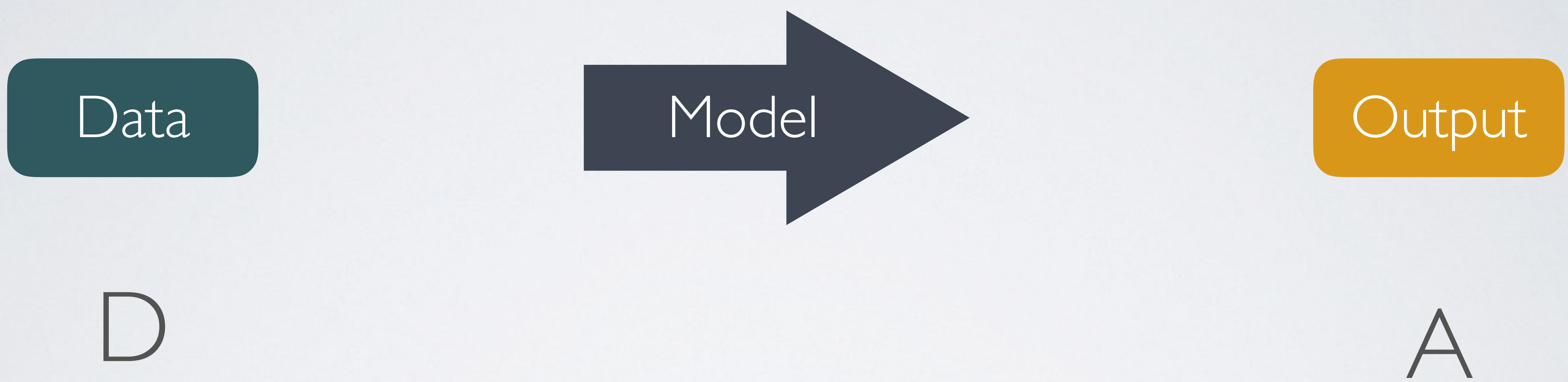
PAPER ASSIGNMENT

Output

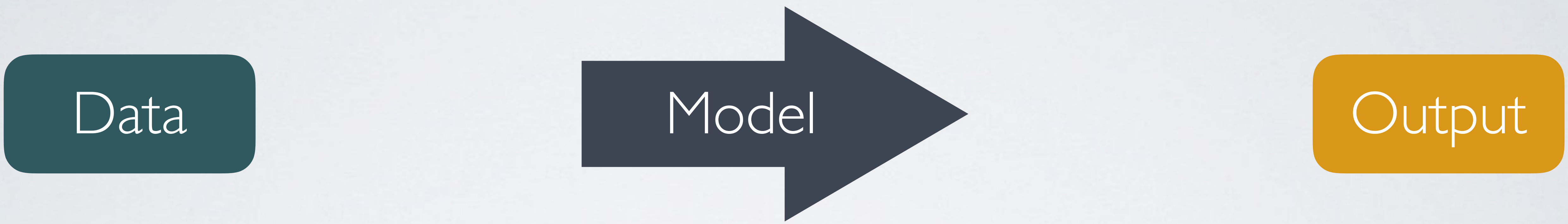
	S1	S2	S3	S4
P1	0.9	0	0	0.1
P2	0	0.1	0	0.9
P3	0.1	0.9	0	0
…				

Output matrix A

TRAINING LOSS



TRAINING LOSS



D

$\mathrm{sum}(A \cdot D)$

A

TENSORFLOW IMPLEMENTATION