FEW-SHOT LEARNING

Because too much shouldn't be needed

May 22, 2018

Felipe Pérez

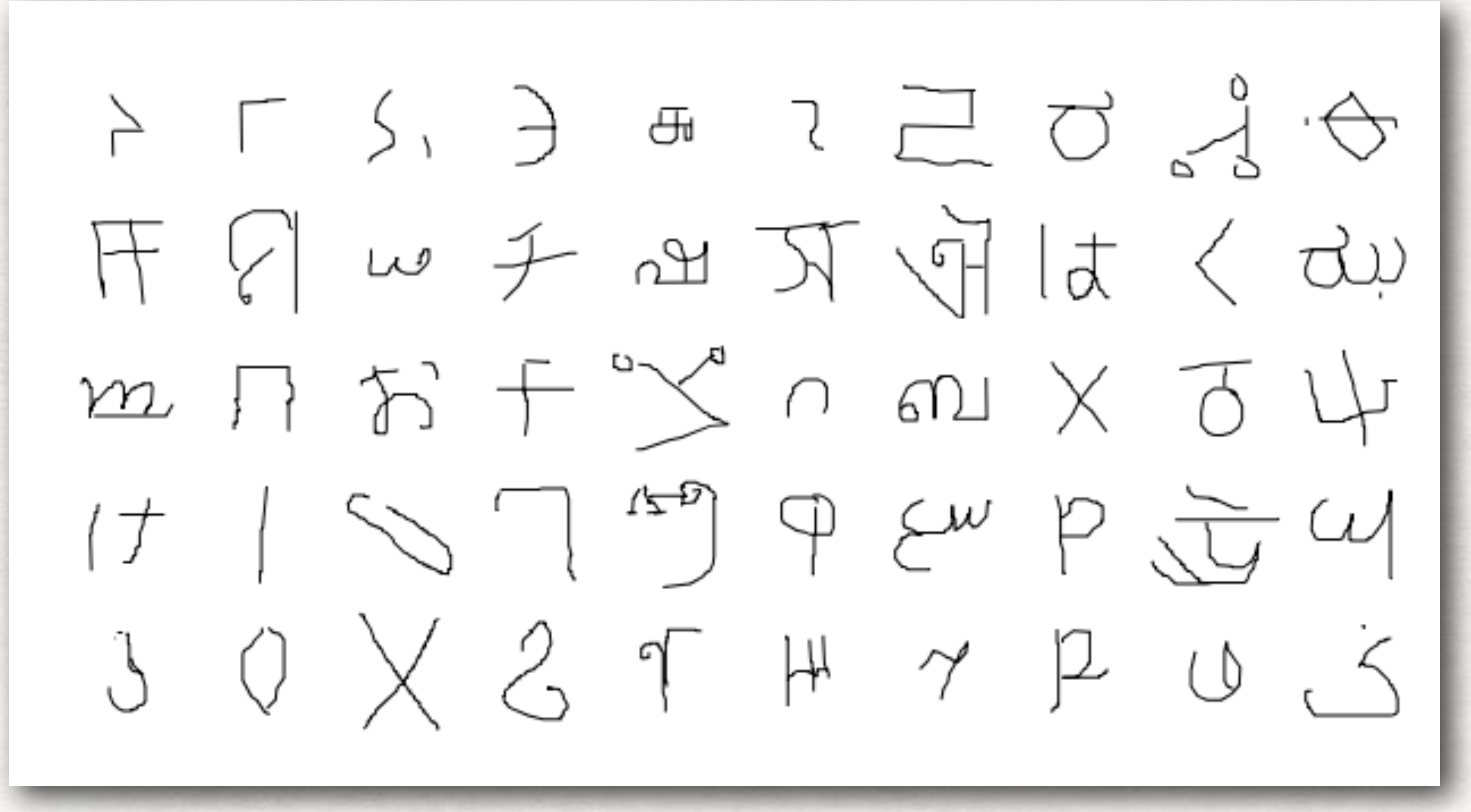
K-SHOT N-WAY CLASSIFICATION

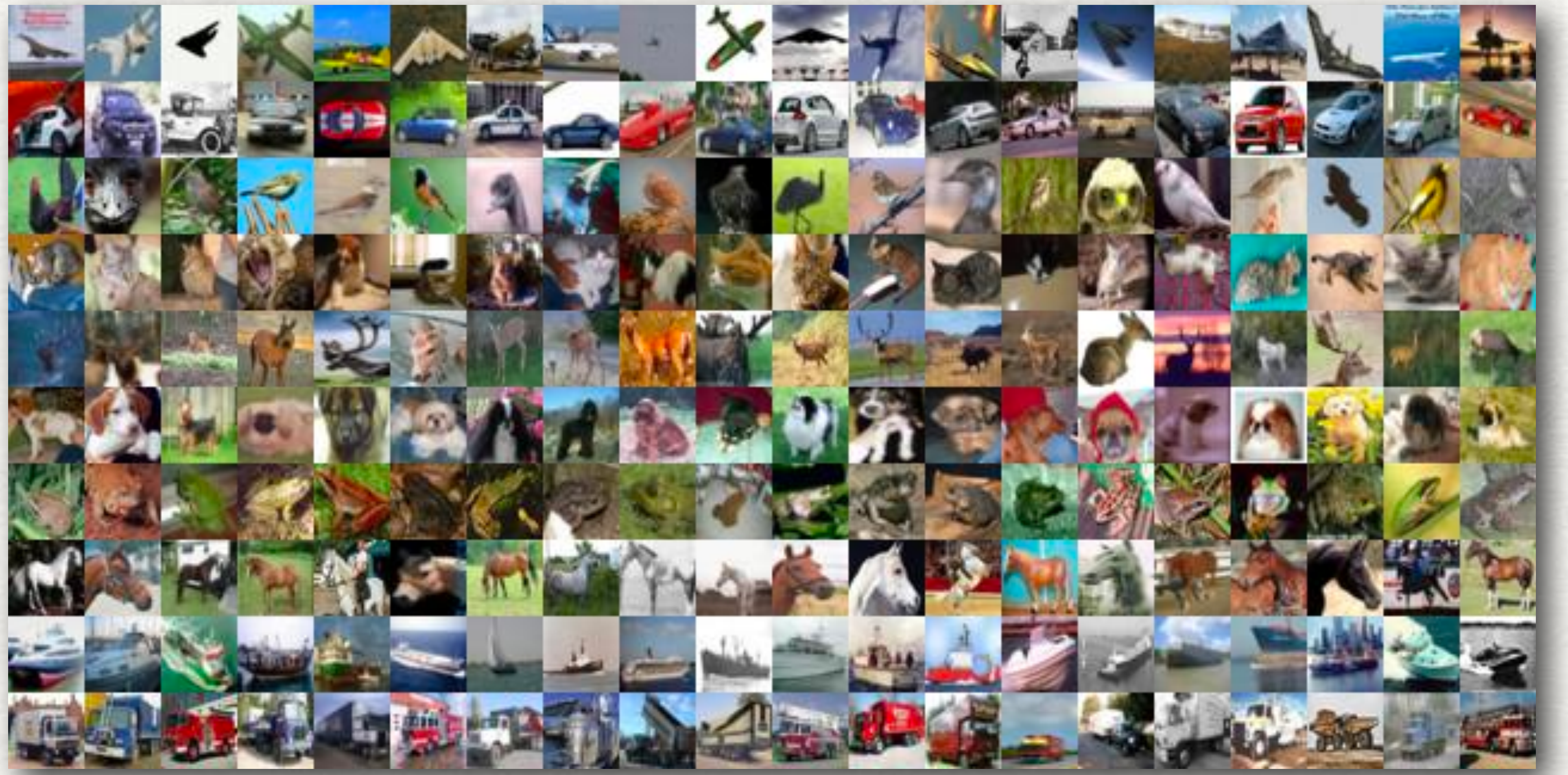
- **Classify** a data point into one of **N classes**.

- Only **K examples** of each class available.

THE DATA

WHAT THEY USE TO PLAY

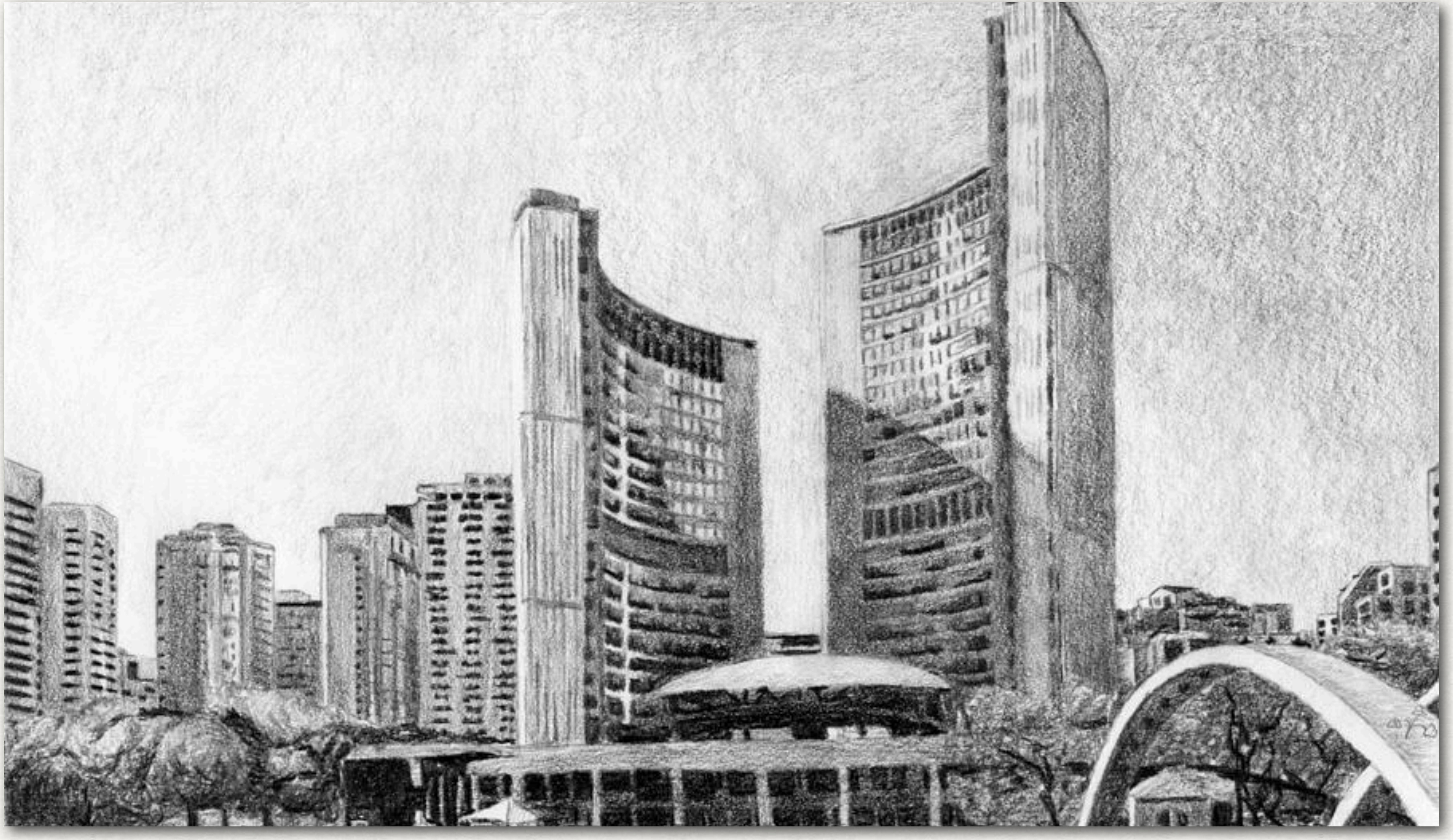




Omniglot:
1623 characters 32x32(?)
50 alphabets
20 different drawers

Mini-ImageNet:
60.000 84x84 Images
100 classes

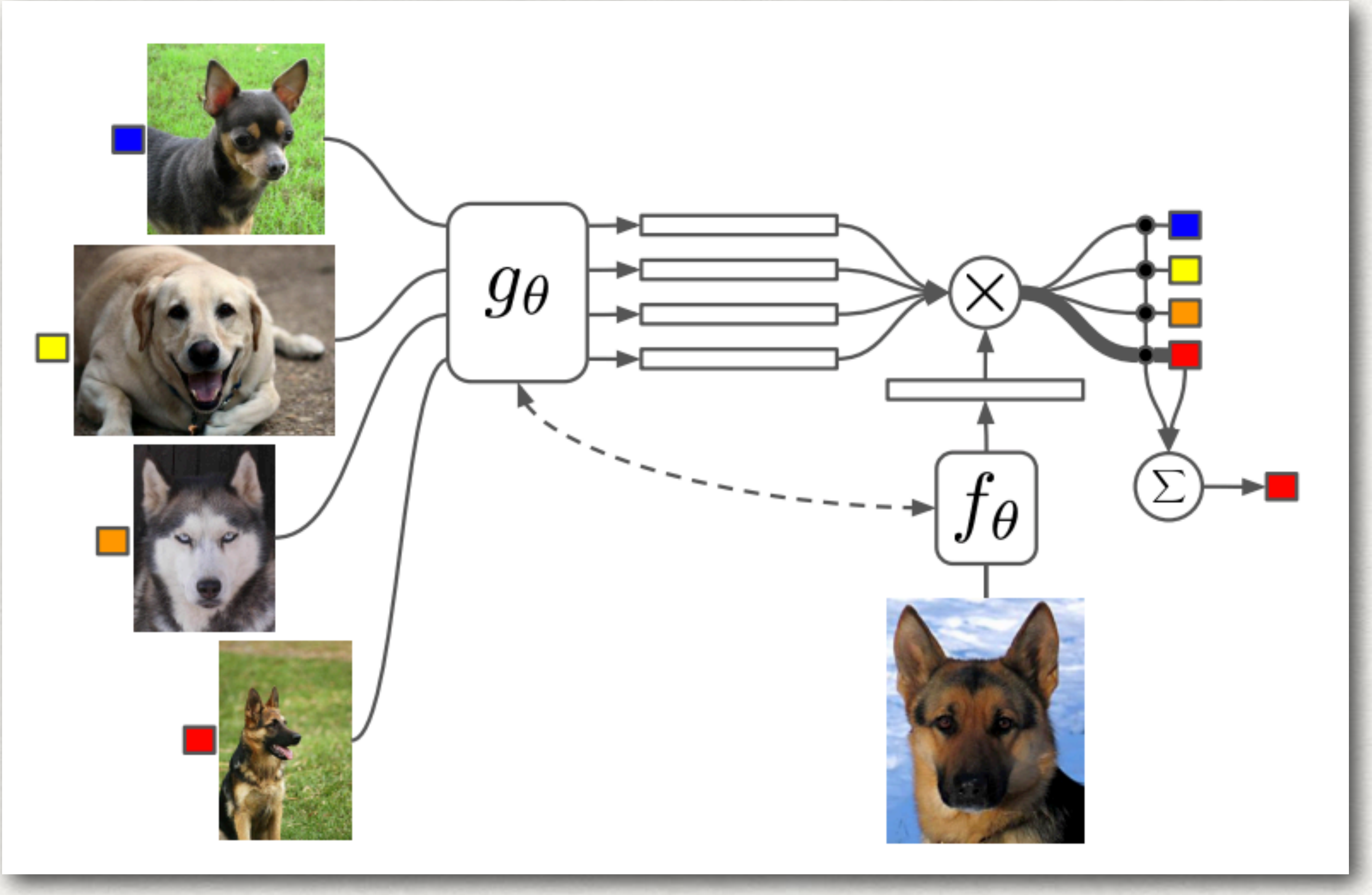
THE LANDSCAPE



Toronto City Hall I Study is a drawing by Duane Gordon

MATCHING NETWORKS

- Results are produced via **attention mechanisms.**
- Full context Embedding
- Clever **training strategy**



Omniglot

	1-Shot	5-Shot
5-Way	98.1	98.9
10-Way	93.8	98.5

Mini-Imagenet

	1-Shot	5-Shot
5-Way	46.6	60

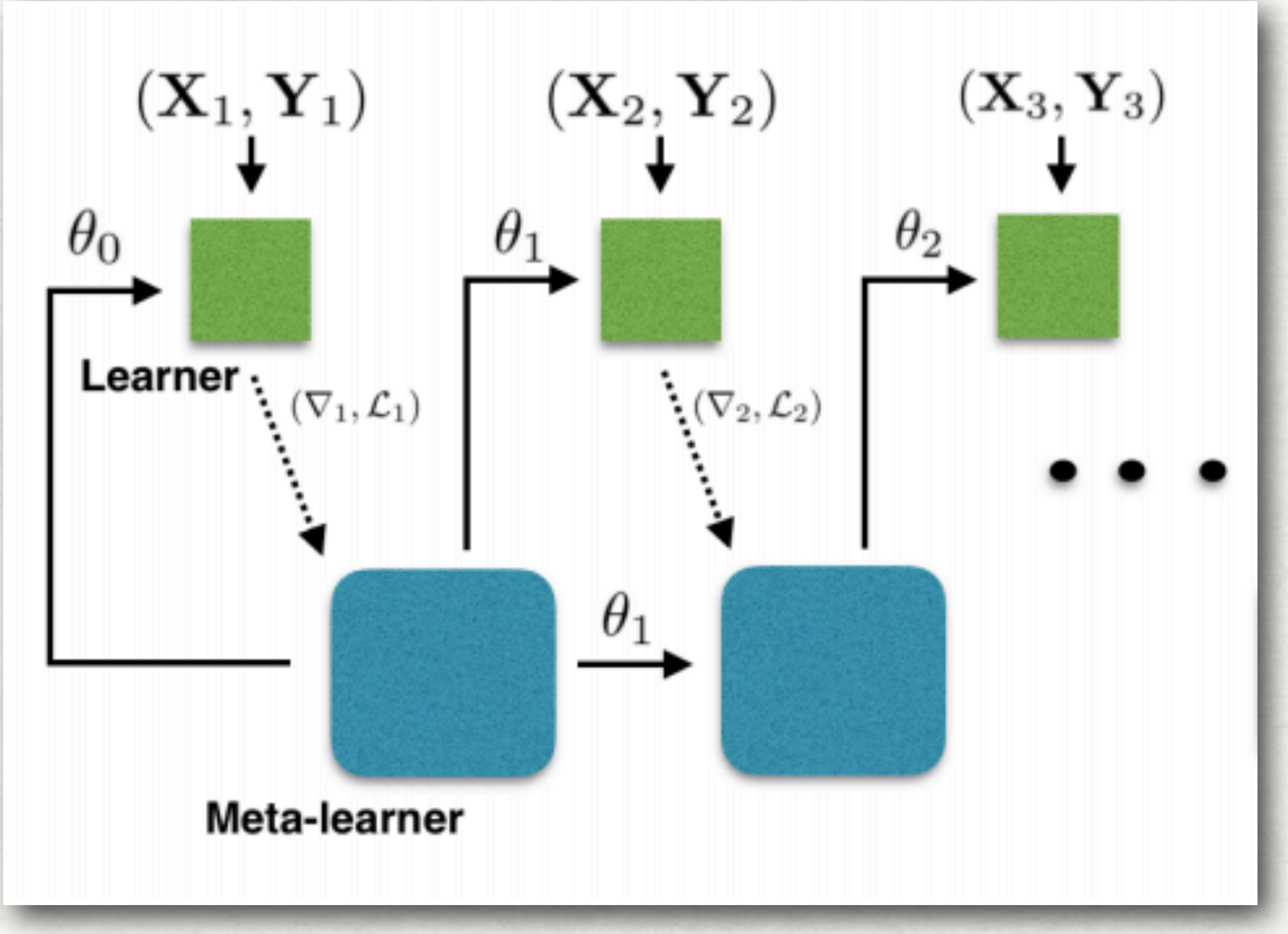
$$\hat{y} = \sum_{I=1}^{k} a(\hat{x}, x_i) y_i$$

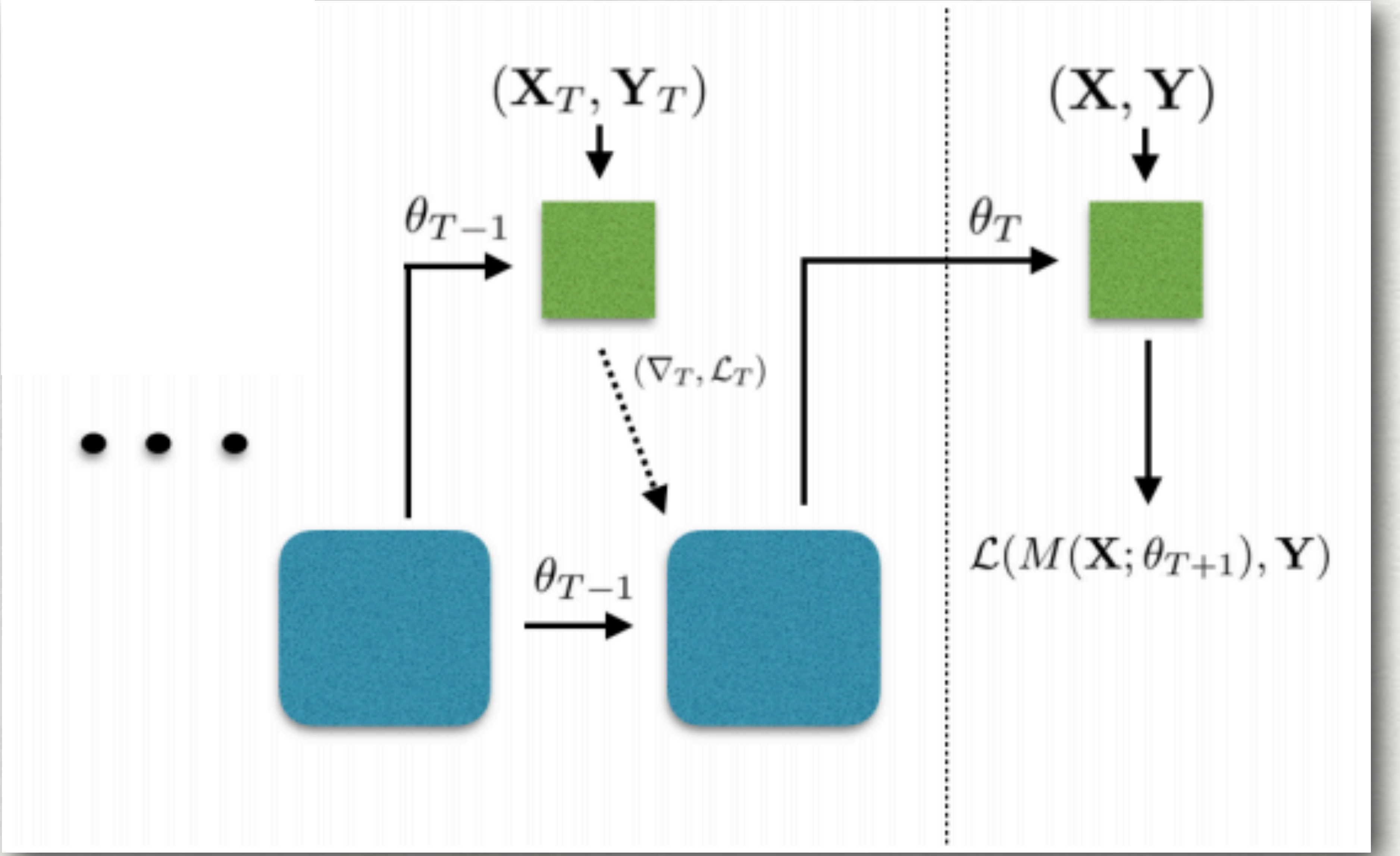
OPTIMIZATION AS MODEL

LEARNING TO LEARN

- LSTM meta-learner trained to optimize a learner neural network classifier.
- Key point:

Gradient descent
is similar to
update for cell state in LSTM





MINI-IMAGENET RESULTS

	1-Shot	5-Shot
5-Way	43.4	60.6

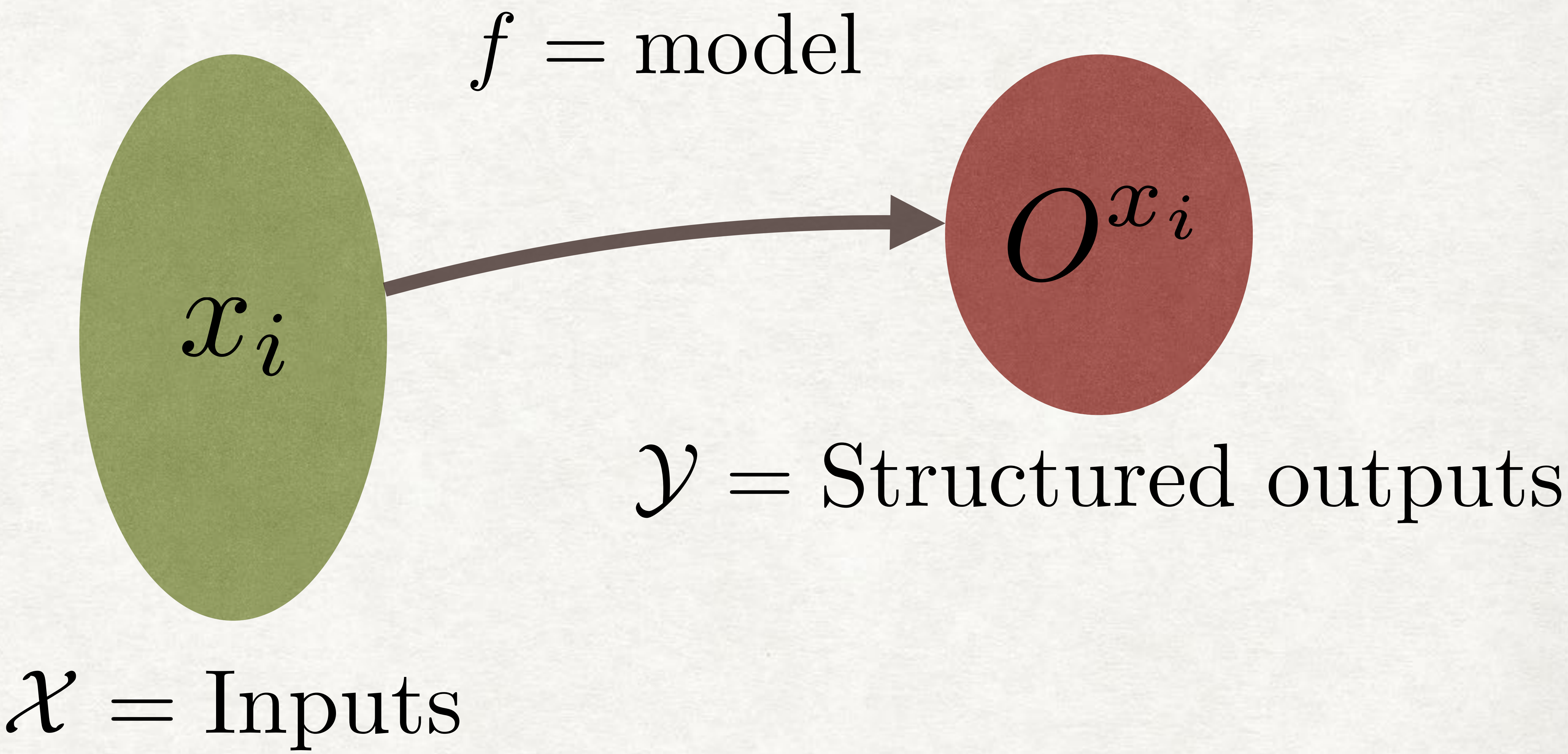
INFORMATION RETRIEVAL APPROACH

THE GOAL

RANKING FOR CLASSIFICATION

We want to classify the points by finding out which class are the most similar ones.

So, for each point we are going to rank all the other with respect to some similarity measure.



MEAN AVERAGE PRECISION

Suppose that we have a batch of points

$$\mathcal{B} = \{x_1, \ldots, x_n\}$$

with associated classes

$$c_1, \ldots, c_n$$

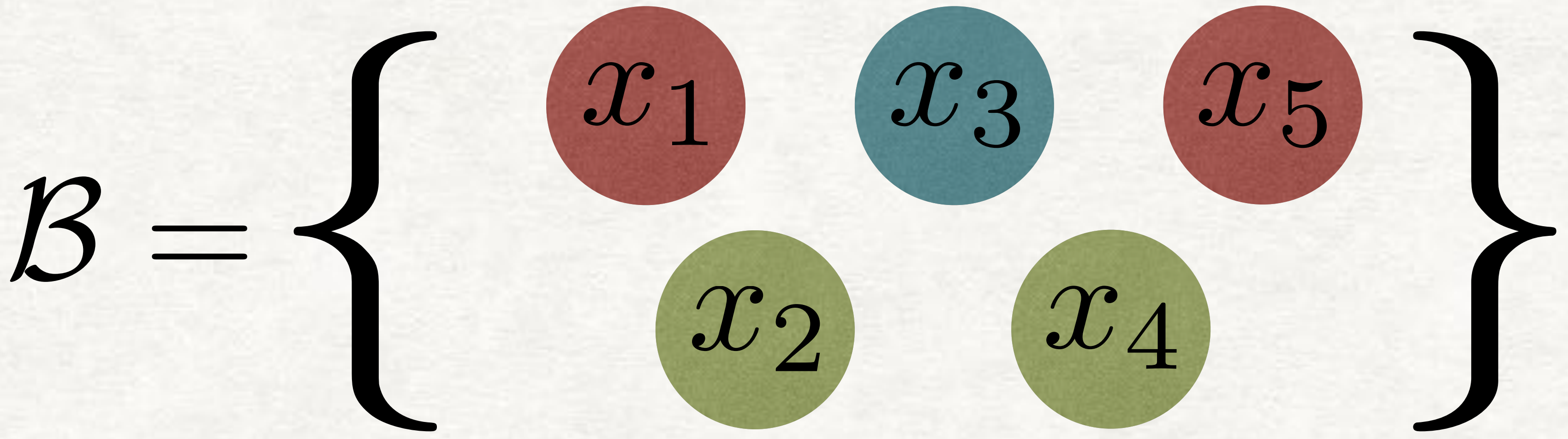
MEAN AVERAGE PRECISION

Suppose that we have a batch of points

$$\mathcal{B} = \{x_1, \dots, x_n\}$$

with associated classes

$$c_1, \dots, c_n$$



MEAN AVERAGE PRECISION

Suppose that we have a batch of points

$$\mathcal{B} = \{x_1, \dots, x_n\}$$

with associated classes

$$c_1, \dots, c_n$$

For each point we collect all the points with the same class

$$\mathrm{Rel}^{x_i} = \{x_j \in \mathcal{B} \mid c_i = c_j\}$$

$$\mathcal{B} = \left\{ \begin{matrix} x_1 & & x_3 & & x_5 \\ & x_2 & & x_4 & \end{matrix} \right\}$$

MEAN AVERAGE PRECISION

Suppose that we have a batch of points

$$\mathcal{B} = \{x_1, \dots, x_n\}$$

with associated classes

$$c_1, \dots, c_n$$

For each point we collect all the points with the same class

$$\mathrm{Rel}^{x_i} = \{x_j \in \mathcal{B} \mid c_i = c_j\}$$

$$\mathcal{B} = \left\{ \begin{array}{ccccc} x_1 & & x_3 & & x_5 \\ & x_2 & & x_4 & \end{array} \right\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1 \; x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2 \; x_4\}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \{x_1, \ldots, x_n\} \quad c_1, \ldots, c_n$$

$$\mathrm{Rel}^{x_i} = \{x_j \in \mathcal{B} \mid c_i = c_j\}$$

$$\mathcal{B} = \left\{ x_1 \; x_3 \; x_5 \; x_2 \; x_4 \right\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1 \; x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2 \; x_4\}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \{x_1, \ldots, x_n\} \quad c_1, \ldots, c_n$$

$$\mathrm{Rel}^{x_i} = \{x_j \in \mathcal{B} \mid c_i = c_j\}$$

$O^{x_i} =$ **Predicted** Ranking based on the similarity of x_i and x_j.

$$\mathcal{B} = \left\{ \begin{matrix} x_1 & x_3 & x_5 \\ x_2 & & x_4 \end{matrix} \right\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1 \; x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2 \; x_4\}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \{x_1, \dots, x_n\} \quad c_1, \dots, c_n$$

$$\mathrm{Rel}^{x_i} = \{x_j \in \mathcal{B} \mid c_i = c_j\}$$

O^{x_i} = **Predicted** Ranking based on the similarity of x_i and x_j.

$O^{x_i}[j]$ = **Predicted** j-th most similar point.

$$\mathcal{B} = \{x_1 \; x_3 \; x_5 \; x_2 \; x_4\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1 \; x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2 \; x_4\}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \{x_1, \ldots, x_n\} \quad c_1, \ldots, c_n$$

$$\mathrm{Rel}^{x_i} = \{x_j \in \mathcal{B} \mid c_i = c_j\}$$

O^{x_i} = **Predicted** Ranking based on the similarity of x_i and x_j.

$O^{x_i}[j]$ = **Predicted** j-th most similar point.

$$\mathcal{B} = \{ x_1 \;\; x_3 \;\; x_5 \;\; x_2 \;\; x_4 \}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{ x_1 \;\; x_5 \}$$

$$\mathrm{Rel}^{x_3} = \{ x_3 \}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{ x_2 \;\; x_4 \}$$

$$O^{x_2} = \begin{array}{|c|} \hline x_2 \\ x_1 \\ x_4 \\ x_5 \\ x_3 \\ \hline \end{array} \quad \begin{array}{l} \leftarrow O^{x_2}[0] \\ \leftarrow O^{x_2}[1] \\ \leftarrow O^{x_2}[2] \\ \leftarrow O^{x_2}[3] \\ \leftarrow O^{x_2}[4] \end{array}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \{ x_1 \;\; x_3 \;\; x_5 \;\; x_2 \;\; x_4 \}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{ x_1 \;\; x_5 \}$$

$$\mathrm{Rel}^{x_3} = \{ x_3 \}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{ x_2 \;\; x_4 \}$$

$$O^{x_2} = \boxed{\begin{matrix} x_2 \\ x_1 \\ x_4 \\ x_5 \\ x_3 \end{matrix}} \begin{matrix} \leftarrow O^{x_2}[0] \\ \leftarrow O^{x_2}[1] \\ \leftarrow O^{x_2}[2] \\ \leftarrow O^{x_2}[3] \\ \leftarrow O^{x_2}[4] \end{matrix}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \left\{ x_1 \; x_3 \; x_5 \; x_2 \; x_4 \right\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1 \; x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2 \; x_4\}$$

The precision at the j-th spot, is the proportion of relevant points found so far.

$$\mathrm{Prec@}j^{x_i} = \frac{|\{k \leq j : O^{x_i}[k] \in \mathrm{Rel}^{x_i}\}|}{j}$$

$$O^{x_2} = \boxed{\begin{array}{lcl} x_2 & \longleftarrow & O^{x_2}[0] \\ x_1 & \longleftarrow & O^{x_2}[1] \\ x_4 & \longleftarrow & O^{x_2}[2] \\ x_5 & \longleftarrow & O^{x_2}[3] \\ x_3 & \longleftarrow & O^{x_2}[4] \end{array}}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \{x_1, x_3, x_5, x_2, x_4\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1\ x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2\ x_4\}$$

The precision at the j-th spot, is the proportion of relevant points found so far.

$$\mathrm{Prec@}j^{x_i} = \frac{|\{k \leq j : O^{x_i}[k] \in \mathrm{Rel}^{x_i}\}|}{j}$$

$$O^{x_2} = \boxed{\begin{array}{c} x_2 \\ x_1 \\ x_4 \\ x_5 \\ x_3 \end{array}} \begin{array}{l} \longleftarrow O^{x_2}[0] \\ \longleftarrow O^{x_2}[1] \\ \longleftarrow O^{x_2}[2] \\ \longleftarrow O^{x_2}[3] \\ \longleftarrow O^{x_2}[4] \end{array}$$

$$\mathrm{Prec@}0^{x_2} = \frac{1}{1} \qquad \mathrm{Prec@}1^{x_2} = \frac{1}{2}$$

$$\mathrm{Prec@}2^{x_2} = \frac{2}{3} \qquad \mathrm{Prec@}3^{x_2} = \frac{2}{4}$$

$$\mathrm{Prec@}4^{x_2} = \frac{2}{5}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \left\{ x_1 \quad x_3 \quad x_5 \quad x_2 \quad x_4 \right\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1 \; x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2 \; x_4\}$$

$$O^{x_2} = \begin{bmatrix} x_2 \\ x_1 \\ x_4 \\ x_5 \\ x_3 \end{bmatrix} \begin{array}{l} \leftarrow O^{x_2}[0] \\ \leftarrow O^{x_2}[1] \\ \leftarrow O^{x_2}[2] \\ \leftarrow O^{x_2}[3] \\ \leftarrow O^{x_2}[4] \end{array}$$

x_2

j-th	0	1	2	3	4
Precision	1/1	1/2	2/3	2/4	2/5

MEAN AVERAGE PRECISION

$$\mathcal{B} = \left\{ \begin{matrix} x_1 & x_3 & x_5 \\ x_2 & & x_4 \end{matrix} \right\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1 \; x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2 \; x_4\}$$

We take the Average Precision over all the positions that have relevant points.

$$\mathrm{AP}^{x_i} = \sum_{j:\, O^{x_i}[j] \in \mathrm{Rel}^{x_i}} \frac{\mathrm{Prec@}j^{x_i}}{|\mathrm{Rel}^{x_i}|}$$

$$O^{x_2} = \boxed{\begin{matrix} x_2 \\ x_1 \\ x_4 \\ x_5 \\ x_3 \end{matrix}} \quad \begin{matrix} \leftarrow O^{x_2}[0] \\ \leftarrow O^{x_2}[1] \\ \leftarrow O^{x_2}[2] \\ \leftarrow O^{x_2}[3] \\ \leftarrow O^{x_2}[4] \end{matrix}$$

x_2

j-th	0	1	2	3	4
Precision	1/1	1/2	2/3	2/4	2/5

MEAN AVERAGE PRECISION

$$\mathcal{B} = \left\{ x_1 \;\; x_3 \;\; x_5 \;\; x_2 \;\; x_4 \right\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1 \;\; x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2 \;\; x_4\}$$

We take the Average Precision over all the positions that have relevant points.

$$\mathrm{AP}^{x_i} = \sum_{j:\, O^{x_i}[j] \in \mathrm{Rel}^{x_i}} \frac{\mathrm{Prec@}j^{x_i}}{|\mathrm{Rel}^{x_i}|}$$

$$O^{x_2} = \begin{array}{|c|l} x_2 & \longleftarrow O^{x_2}[0] \\ x_1 & \longleftarrow O^{x_2}[1] \\ x_4 & \longleftarrow O^{x_2}[2] \\ x_5 & \longleftarrow O^{x_2}[3] \\ x_3 & \longleftarrow O^{x_2}[4] \end{array}$$

x_2

j-th	0	1	2	3	4
Precision	1/1	1/2	2/3	2/4	2/5

$$\mathrm{AP}^{x_2} = \frac{1/1 + 2/3}{2} = \frac{5}{6}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \left\{ x_1 \; x_3 \; x_5 \; x_2 \; x_4 \right\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1 \; x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2 \; x_4\}$$

x_2

j-th	0	1	2	3	4
Precision	1/1	1/2	2/3	2/4	2/5

$$\mathrm{AP}^{x_2} = \frac{1/1 + 2/3}{2} = \frac{5}{6}$$

$$O^{x_2} = \begin{bmatrix} x_2 \\ x_1 \\ x_4 \\ x_5 \\ x_3 \end{bmatrix} \begin{matrix} \leftarrow O^{x_2}[0] \\ \leftarrow O^{x_2}[1] \\ \leftarrow O^{x_2}[2] \\ \leftarrow O^{x_2}[3] \\ \leftarrow O^{x_2}[4] \end{matrix}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \left\{ \begin{matrix} x_1 & x_3 & x_5 \\ x_2 & & x_4 \end{matrix} \right\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1\ x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2\ x_4\}$$

x_2

j-th	0	1	2	3	4
Precision	1/1	1/2	2/3	2/4	2/5

$$\mathrm{AP}^{x_2} = \frac{1/1 + 2/3}{2} = \frac{5}{6}$$

$$O^{x_2} = \boxed{\begin{array}{l} x_2 \longleftarrow O^{x_2}[0] \\ x_1 \longleftarrow O^{x_2}[1] \\ x_4 \longleftarrow O^{x_2}[2] \\ x_5 \longleftarrow O^{x_2}[3] \\ x_3 \longleftarrow O^{x_2}[4] \end{array}}$$

Finally, we take the mean over all the averages:

$$\mathrm{mAP} = \frac{1}{|\mathcal{B}|} \sum_i \mathrm{AP}^{x_i}$$

MEAN AVERAGE PRECISION

$$\mathcal{B} = \left\{ x_1 \; x_3 \; x_5 \; x_2 \; x_4 \right\}$$

$$\mathrm{Rel}^{x_1} = \mathrm{Rel}^{x_5} = \{x_1 \; x_5\}$$

$$\mathrm{Rel}^{x_3} = \{x_3\}$$

$$\mathrm{Rel}^{x_2} = \mathrm{Rel}^{x_4} = \{x_2 \; x_4\}$$

$$O^{x_2} = \begin{array}{|c|l} \hline x_2 & \longleftarrow O^{x_2}[0] \\ x_1 & \longleftarrow O^{x_2}[1] \\ x_4 & \longleftarrow O^{x_2}[2] \\ x_5 & \longleftarrow O^{x_2}[3] \\ x_3 & \longleftarrow O^{x_2}[4] \\ \hline \end{array}$$

x_2

j-th	0	1	2	3	4
Precision	1/1	1/2	2/3	2/4	2/5

$$\mathrm{AP}^{x_2} = \frac{1/1 + 2/3}{2} = \frac{5}{6}$$

Finally, we take the mean over all the averages:

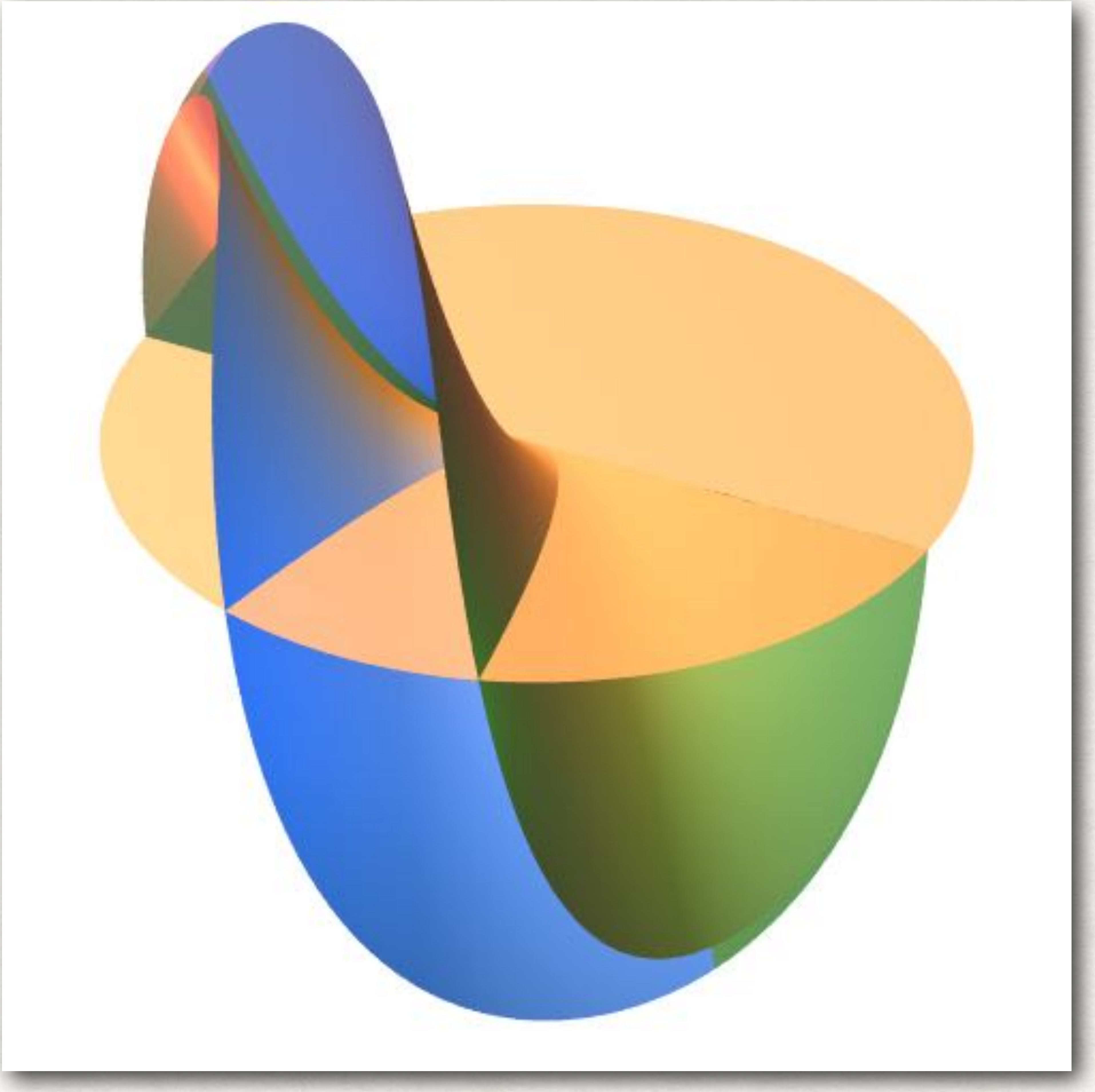
$$\mathrm{mAP} = \frac{1}{|\mathcal{B}|} \sum_i \mathrm{AP}^{x_i}$$

$$\mathrm{mAP} = \frac{\mathrm{AP}^{x_1} + 5/6 + \mathrm{AP}^{x_3} + \mathrm{AP}^{x_4} + \mathrm{AP}^{x_5}}{5}$$

PROBLEMS AHEAD

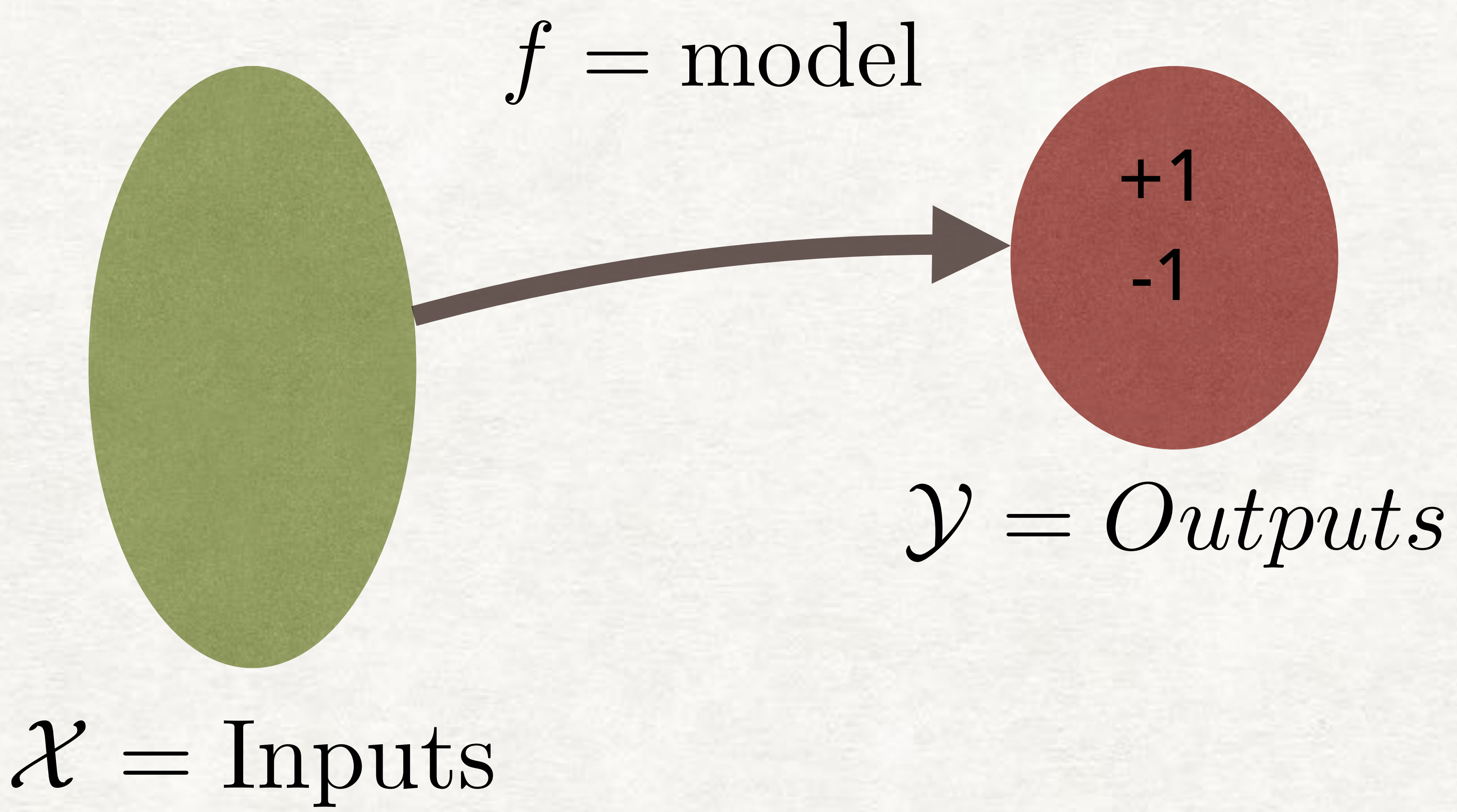
BECAUSE OTHERWISE IT WOULDN'T BE FUN

The mean Average Precision is a terrible loss function (for gradient descent purposes)



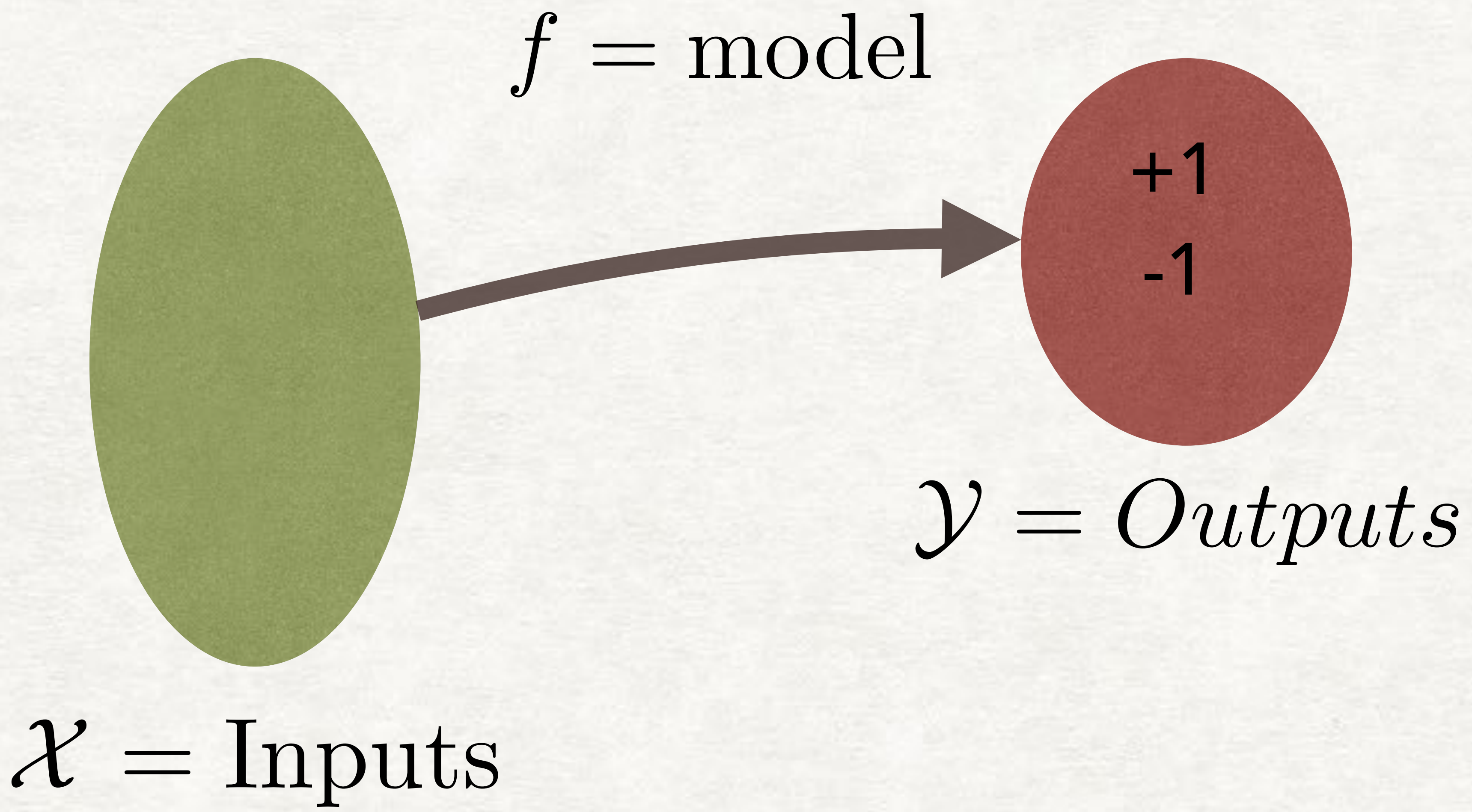
STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

USUAL SVM



STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

USUAL SVM



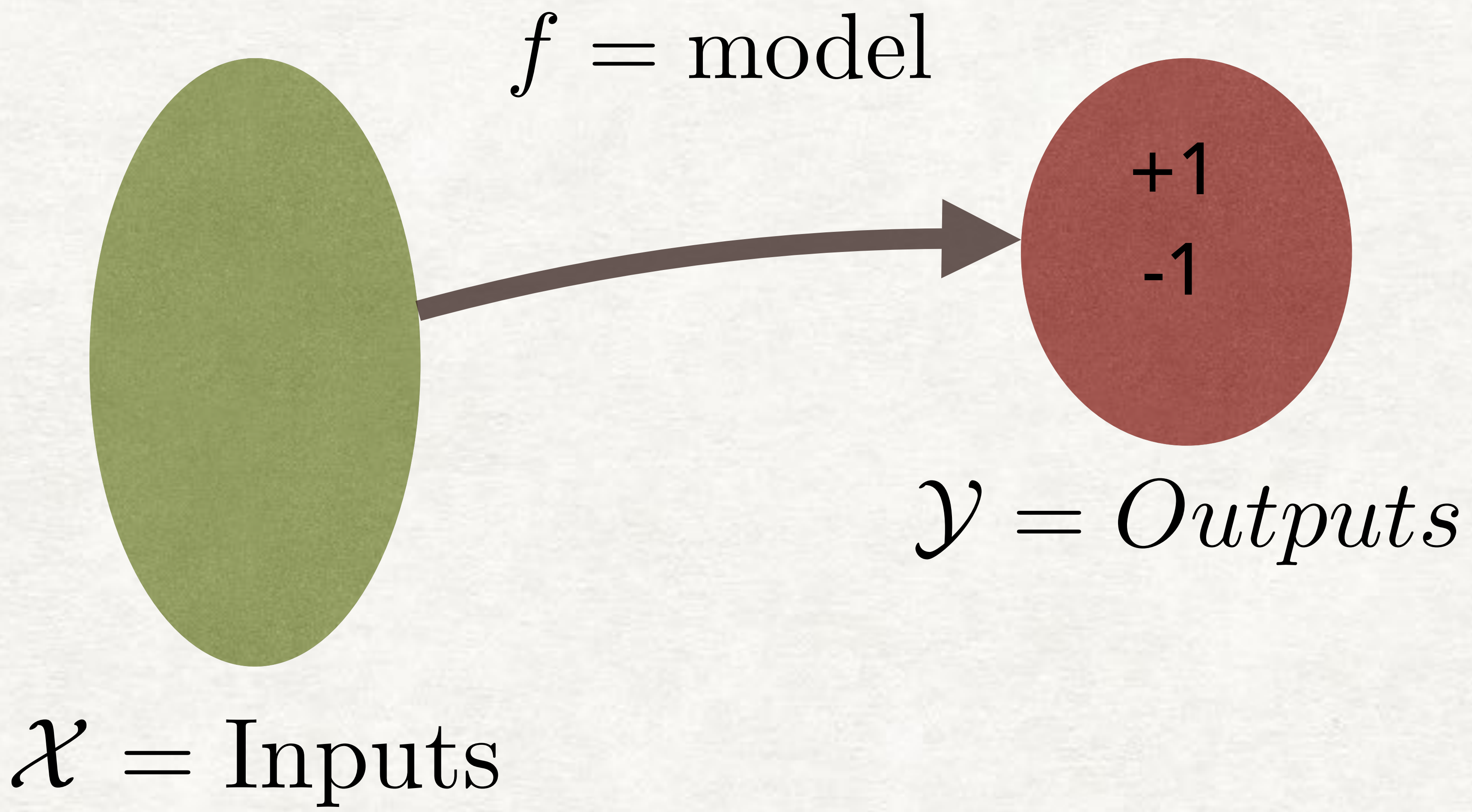
$$\min_{w,\zeta} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\zeta_i$$

$$y_i(wx_i) - (-y_i)(wx_i) \geq 1 - \zeta_i$$

$$\zeta_i \geq 0$$

STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

USUAL SVM



$$\min_{w,\zeta} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\zeta_i$$

$$y_i(wx_i) - (-y_i)(wx_i) \geq 1 - \zeta_i$$

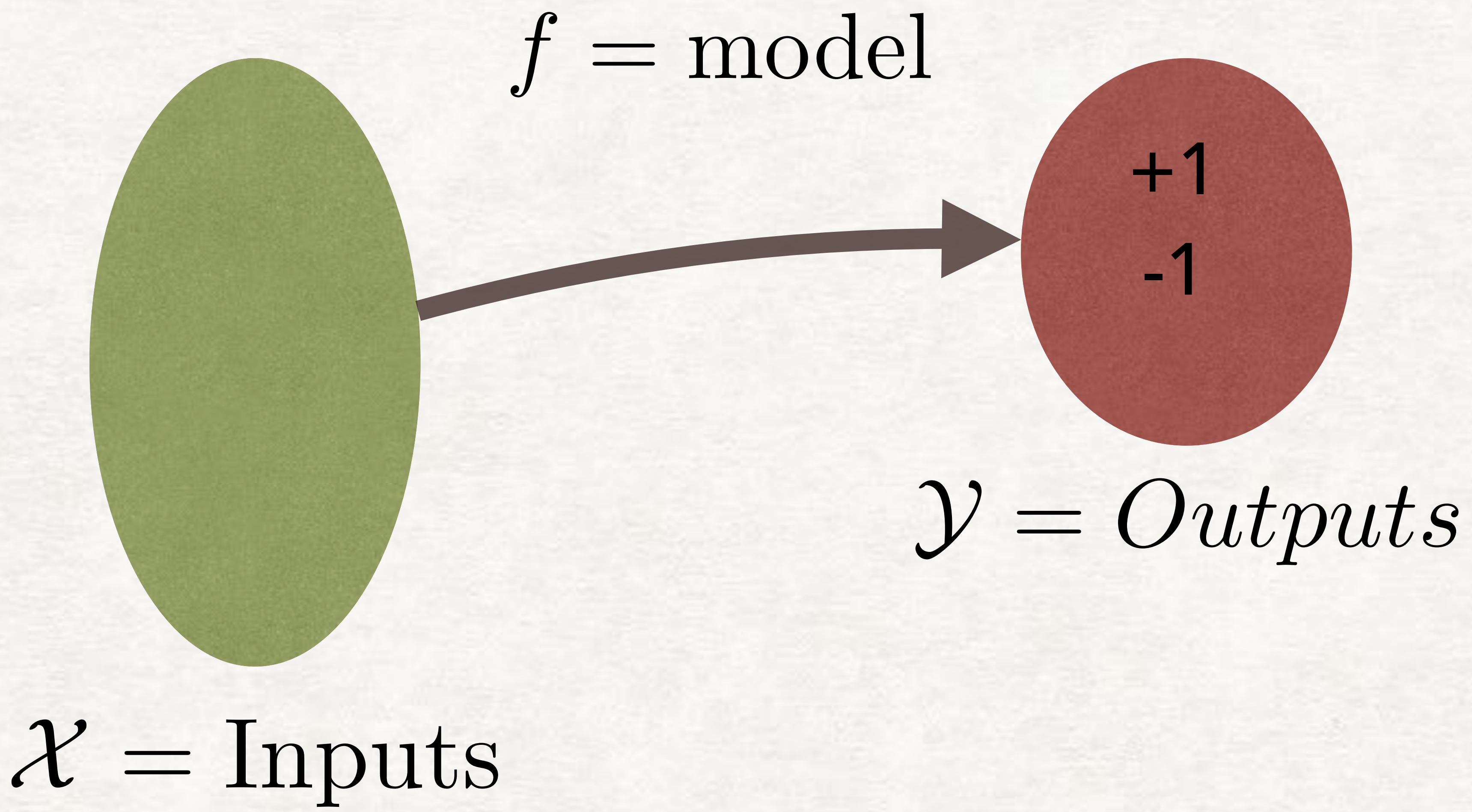
$$\zeta_i \geq 0$$

Note that there is a Scoring Function:

$$F(x, y; w) = y(wx)$$

STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

USUAL SVM



$$\min_{w,\zeta} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\zeta_i$$

$$y_i(wx_i) - (-y_i)(wx_i) \geq 1 - \zeta_i$$

$$\zeta_i \geq 0$$

Note that there is a Scoring Function:

$$F(x, y; w) = y(wx)$$

And we have a loss function:

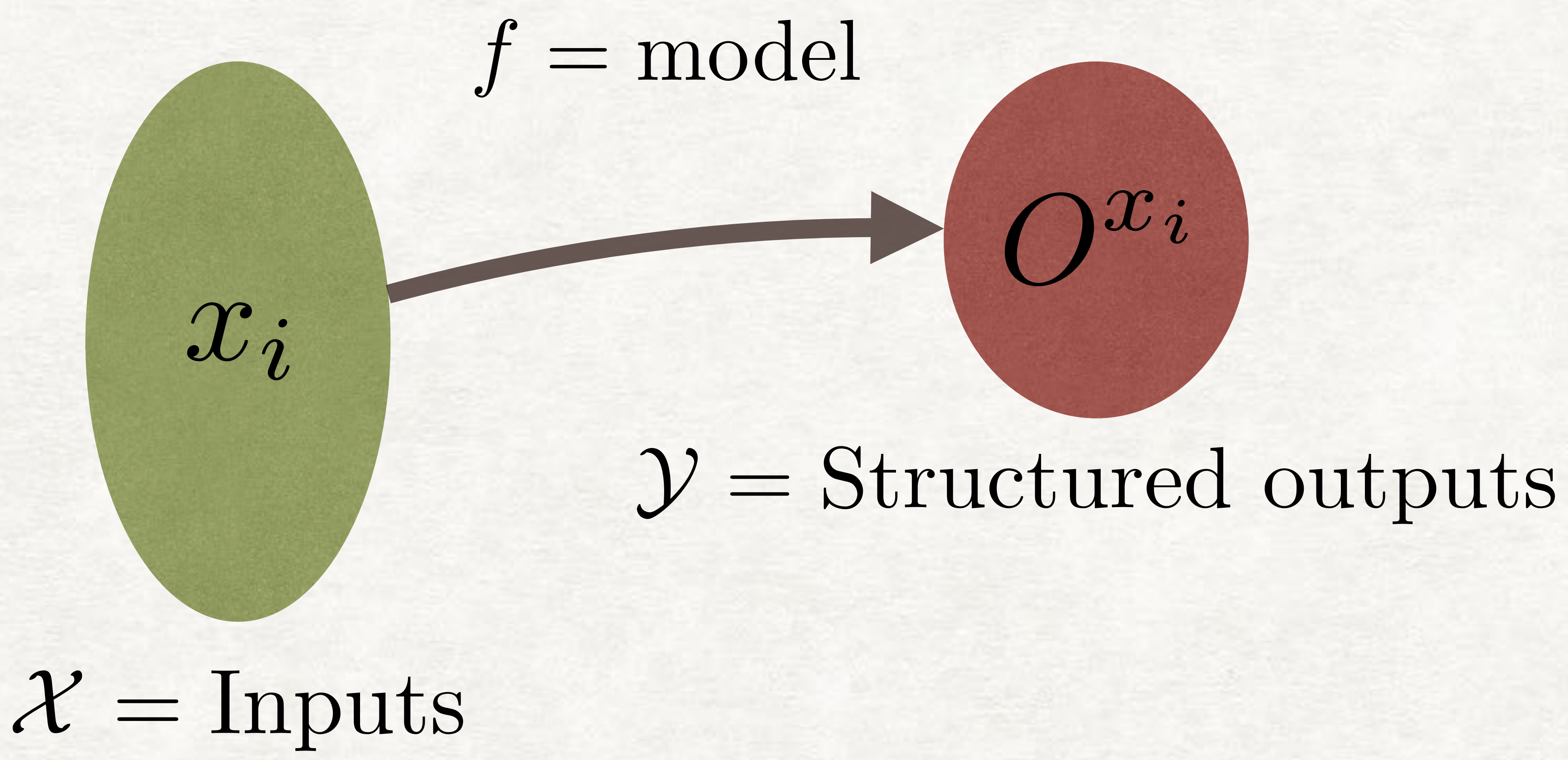
$$L(y_i, -y_i) = 1$$

STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

MARGIN RE-SCALING

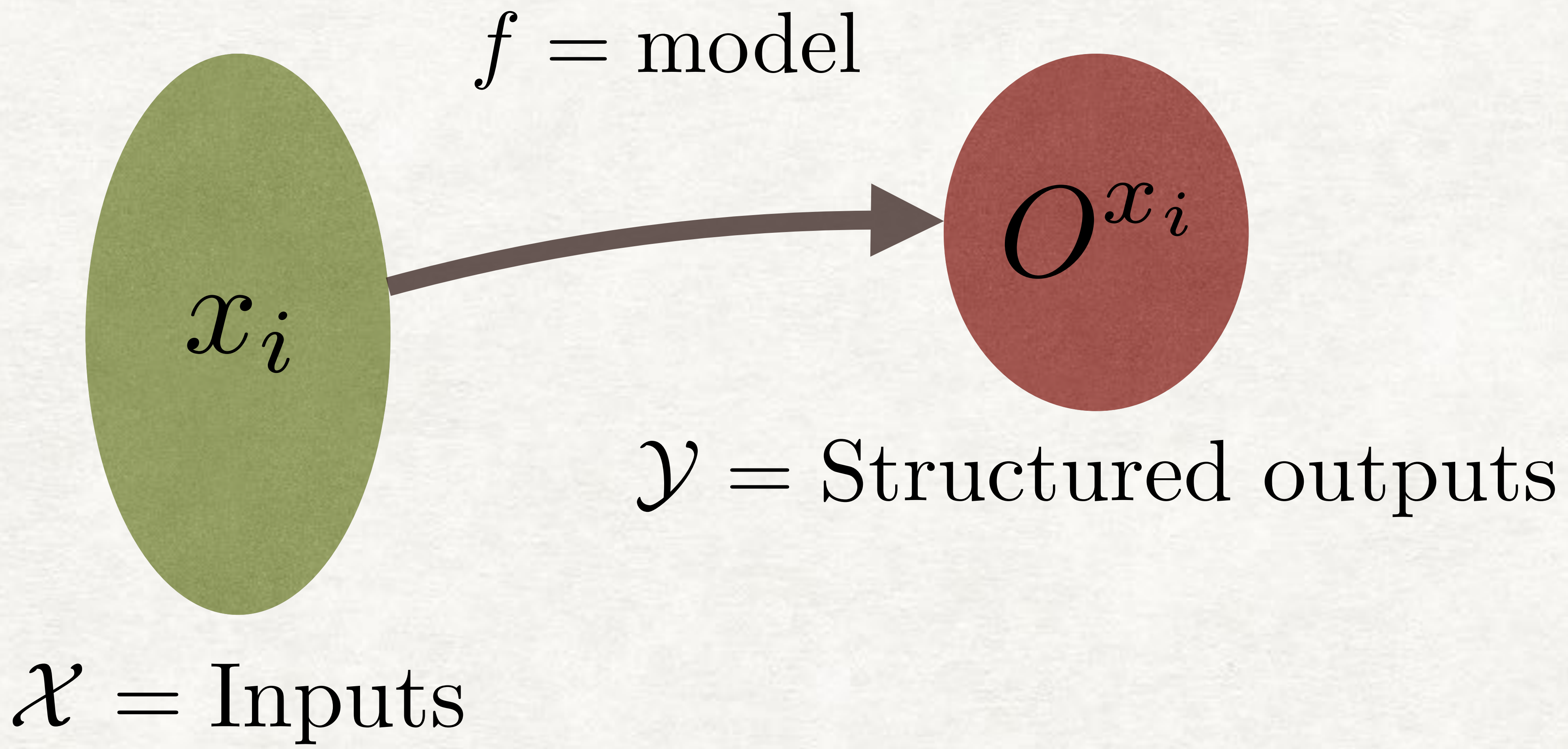
STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

MARGIN RE-SCALING



STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

MARGIN RE-SCALING

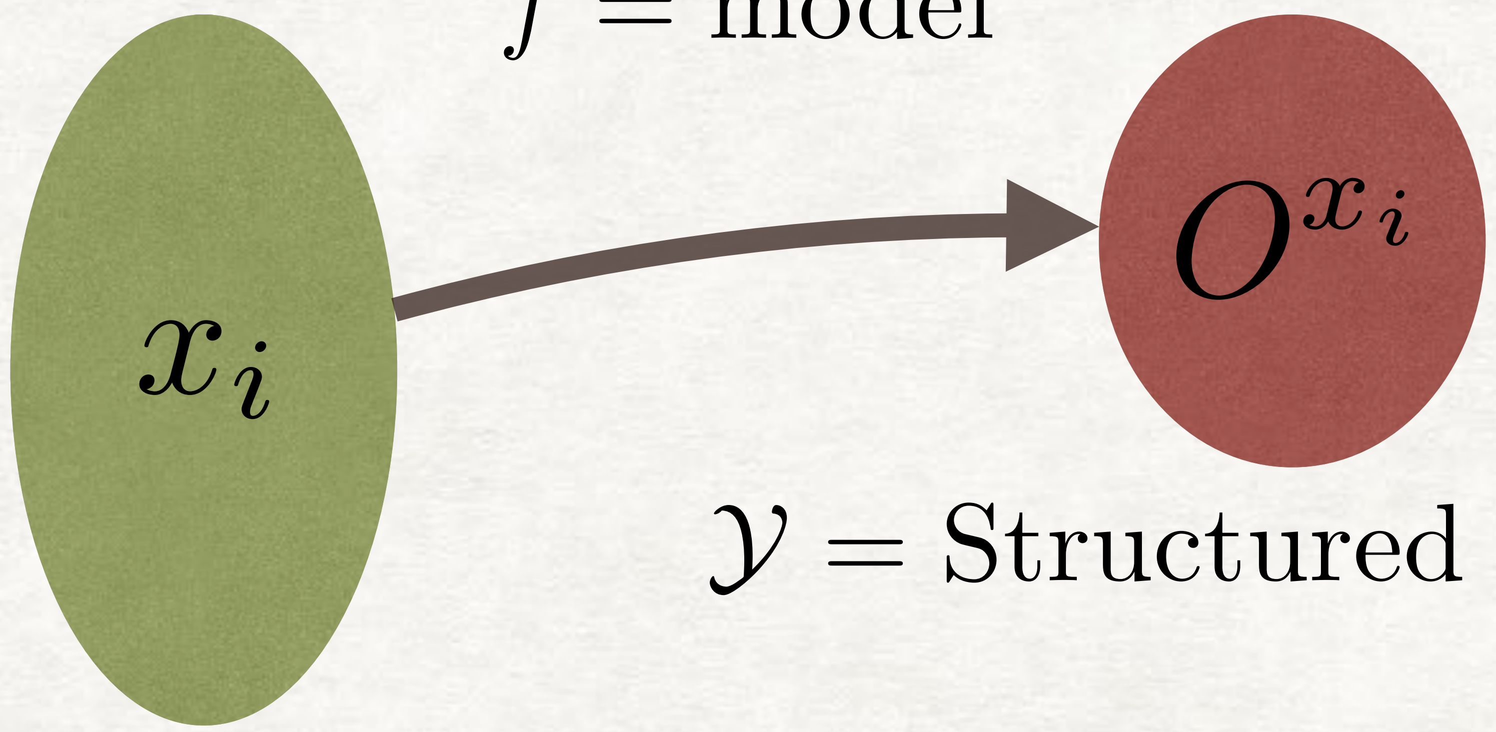


$$\min_{w,\zeta} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n} \zeta_i$$

STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

MARGIN RE-SCALING

$f = \text{model}$



$\mathcal{Y} = \text{Structured outputs}$

$\mathcal{X} = \text{Inputs}$

$$\min_{w,\zeta} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\zeta_i$$

We need a task loss function:

$$L(y, y') \geq 0$$

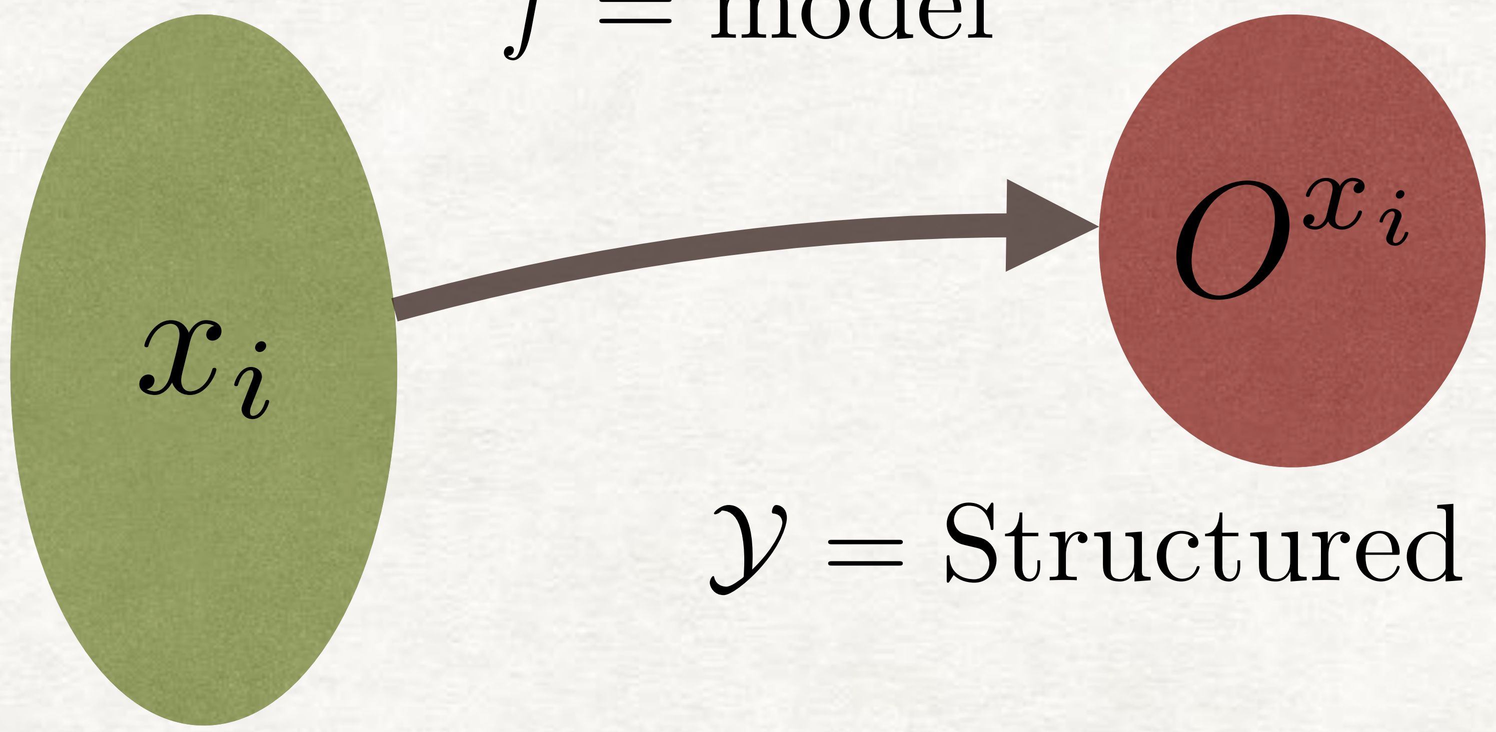
And a scoring function:

$$F(x, y; w)$$

STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

MARGIN RE-SCALING

f = model



$\mathcal{Y}$ = Structured outputs

$\mathcal{X}$ = Inputs

We need a task loss function:

$$L(y, y') \geq 0$$

And a scoring function:

$$F(x, y; w)$$

$$\min_{w,\zeta} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\zeta_i$$

$$F(x_i, y_i; w) - F(x_i, y; w) \geq L(y_i, y) - \zeta_i$$
$$\forall i, \forall y \in \mathcal{Y} \setminus \{y_i\}$$

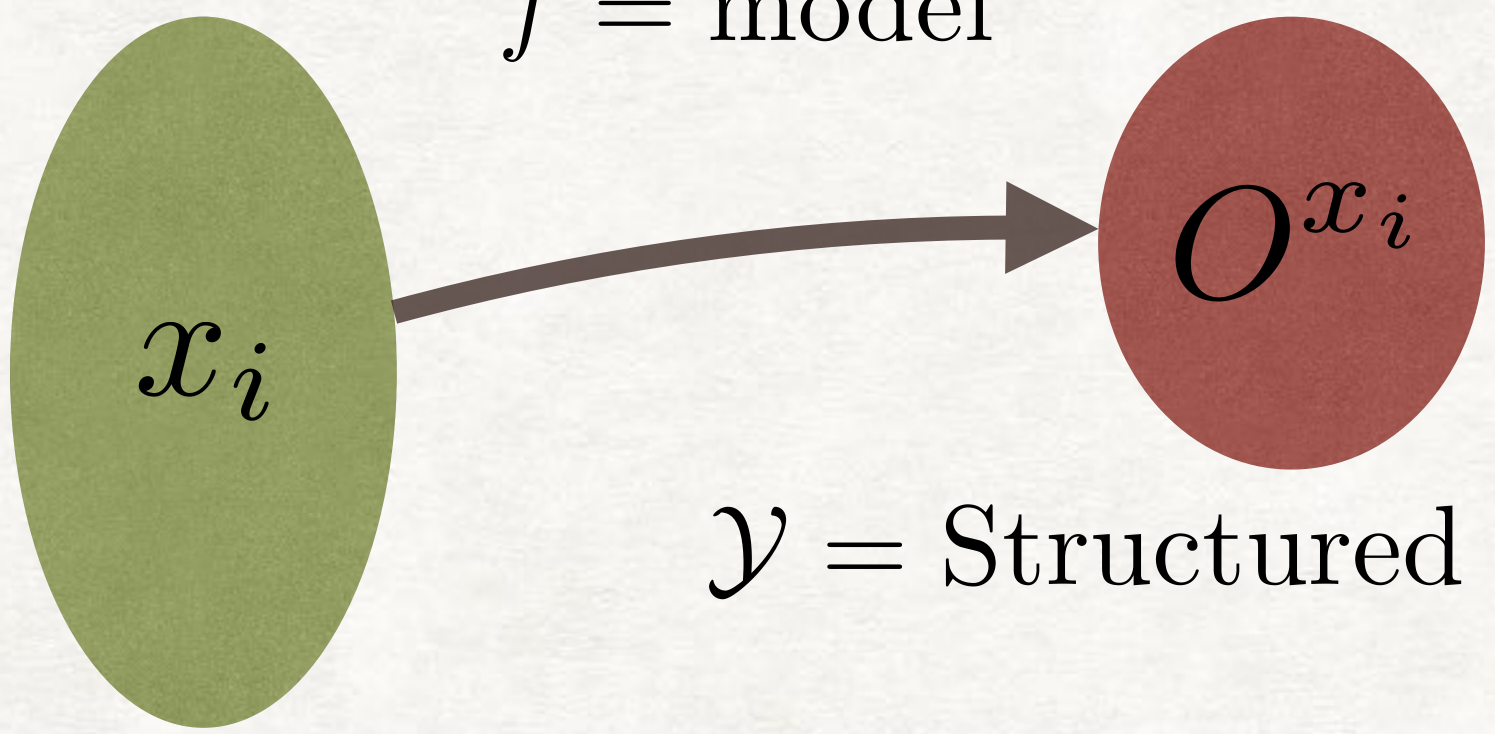
STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

MARGIN RE-SCALING

STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

MARGIN RE-SCALING

f = model



$\mathcal{Y}$ = Structured outputs

$\mathcal{X}$ = Inputs

$$\min_{w,\zeta} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\zeta_i$$

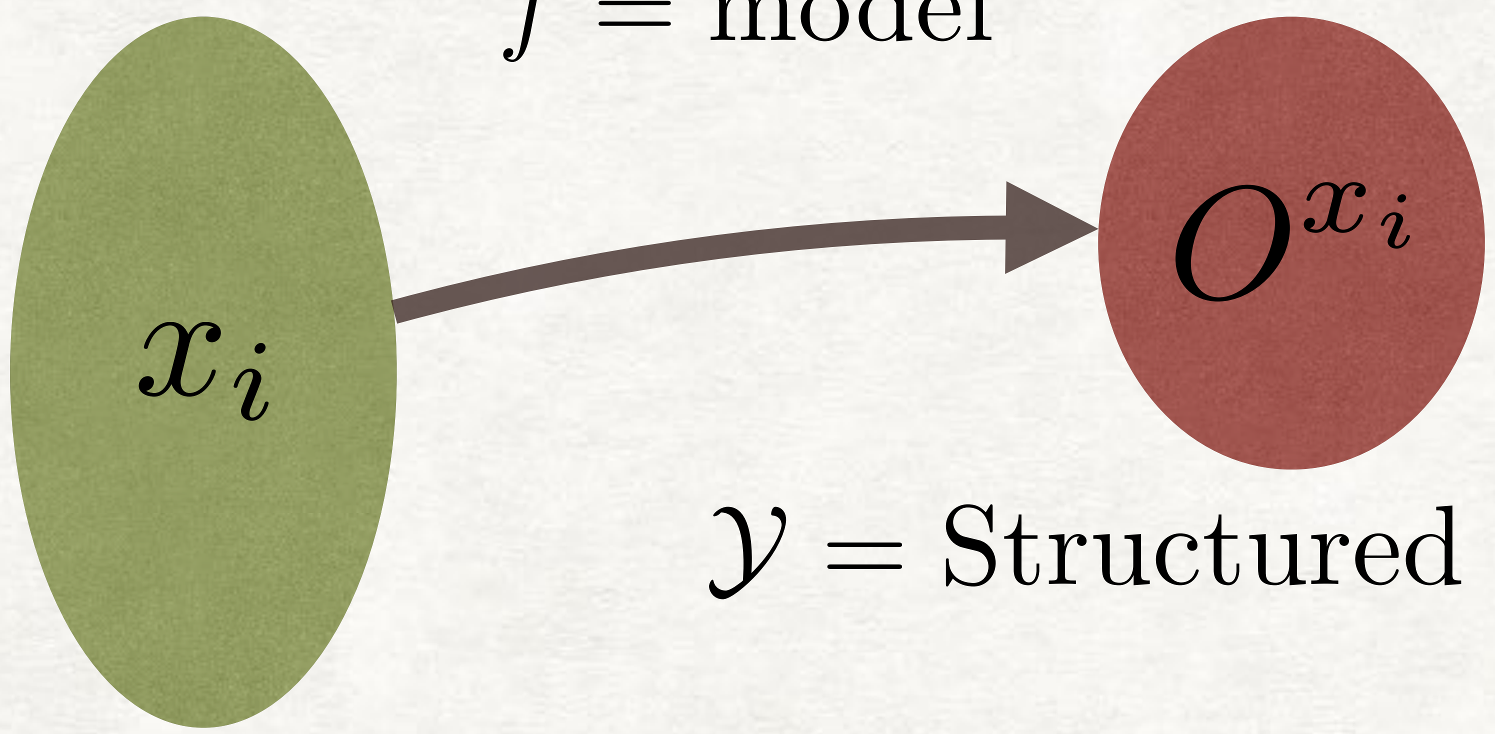
$$F(x_i, y_i; w) - F(x_i, y; w) \geq L(y_i, y) - \zeta_i$$

$$\forall i, \forall y \in \mathcal{Y} \setminus \{y_i\}$$

STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

MARGIN RE-SCALING

f = model



$\mathcal{Y}$ = Structured outputs

$\mathcal{X}$ = Inputs

$$\min_{w,\zeta} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\zeta_i$$

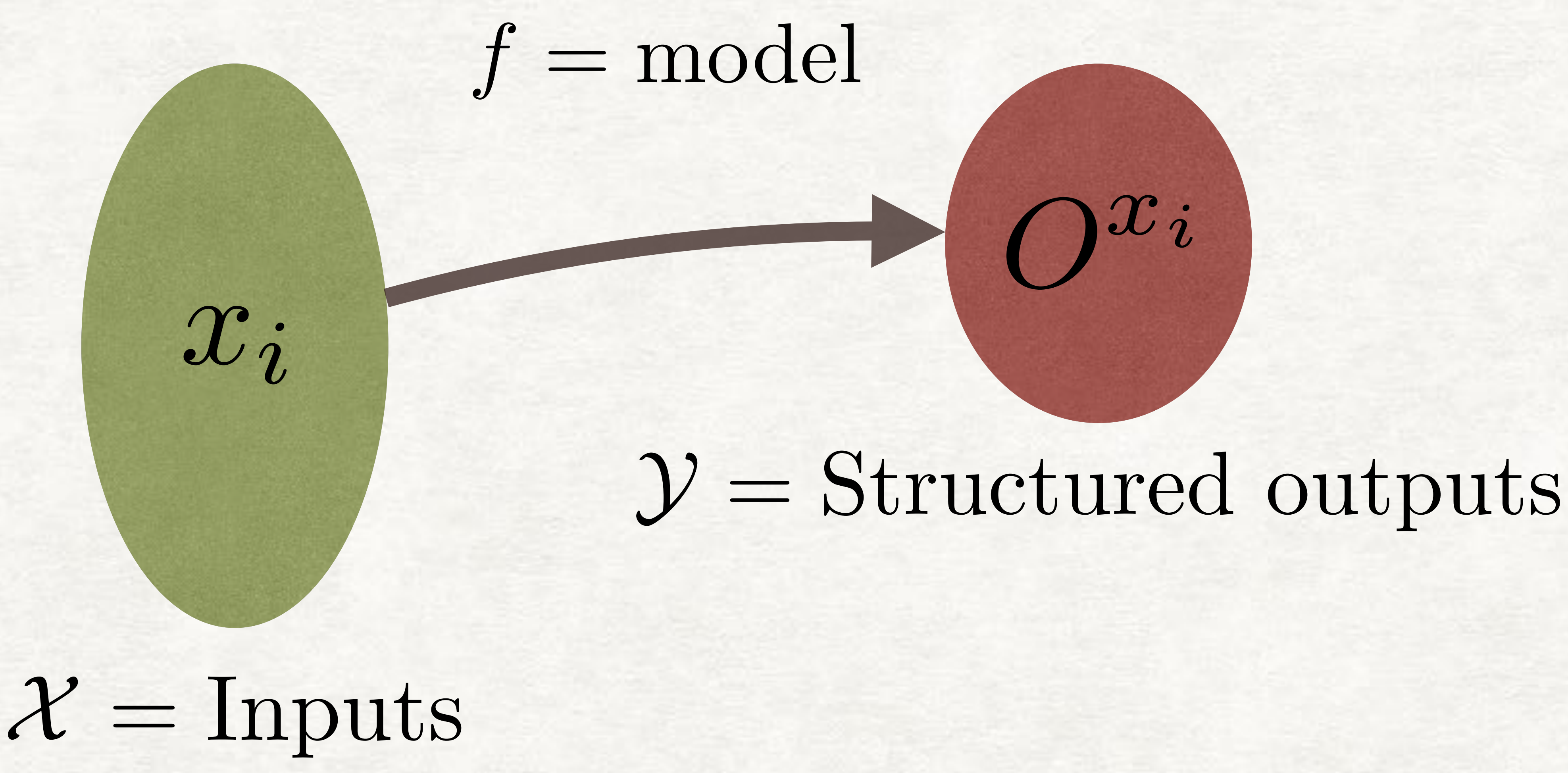
$$F(x_i, y_i; w) - F(x_i, y; w) \geq L(y_i, y) - \zeta_i$$

$$\forall i, \forall y \in \mathcal{Y} \setminus \{y_i\}$$

Result: Finding this minimum gives an upper bound to the average loss!

STRUCTURAL SUPPORT VECTOR MACHINES (SSVM)

MARGIN RE-SCALING



$$\min_{w,\zeta} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\zeta_i$$

$$F(x_i, y_i; w) - F(x_i, y; w) \geq L(y_i, y) - \zeta_i$$

$$\forall i, \forall y \in \mathcal{Y} \setminus \{y_i\}$$

Result: Finding this minimum gives an upper bound to the average loss!

TRAINING METHOD: Stochastic sub-gradient descent, by first finding the value

$$y_{\text{hinge}} = \text{argmax}_{y \in \mathcal{Y}} \{F(x_i, y, w) + L(y_t, y)\}$$

and then updating via

$$\nabla_w L(y) = \nabla_w F(x_t, y_{\text{hinge}}, w) - \nabla_w F(x_t, y_t, w)$$

FEW-SHOT LEARNING BY OPTIMIZING mAP

SETUP

$$\mathcal{B} = \{x_1, \ldots, x_N\}$$

$$\text{Classes} = \{1, 2, \ldots, \mathcal{C}\}$$

$$P^c = \text{points in class } c$$

$$N^c = \text{points not in class } c$$

We get our score function:

$$F = \sum_{i \in \mathcal{B}} \frac{1}{|P^{c_i}||N^{c_i}|} \sum_{k \in P^{c_i} \setminus \{i\}} \sum_{j \in N^{c_i}} y^i_{kj} (\varphi(x_i, x_k, w) - \varphi(x_i, x_j, w))$$

FEW-SHOT LEARNING BY OPTIMIZING mAP

SETUP

$$\mathcal{B} = \{x_1, \ldots, x_N\}$$

$$\text{Classes} = \{1, 2, \ldots, \mathcal{C}\}$$

$$P^c = \text{points in class } c$$

$$N^c = \text{points not in class } c$$

We get our score function:

$$F = \sum_{i \in \mathcal{B}} \frac{1}{|P^{c_i}|\,|N^{c_i}|} \sum_{k \in P^{c_i} \setminus \{i\}} \sum_{j \in N^{c_i}} y^i_{kj} \left(\boxed{\varphi(x_i, x_k, w)} - \varphi(x_i, x_j, w) \right)$$

Way of comparing embeddings

$$\boxed{\begin{gathered}\varphi(x_1, x_2, w) = \frac{f(x_1, w) \cdot f(x_2, w)}{|f(x_1, w)|\,|f(x_2, w)|} \\ f(x, w) = \text{embedding function, via neural network}\end{gathered}}$$

FEW-SHOT LEARNING BY OPTIMIZING mAP

SETUP

$$\mathcal{B} = \{x_1, \ldots, x_N\}$$

$$\text{Classes} = \{1, 2, \ldots, \mathcal{C}\}$$

$$P^c = \text{points in class } c$$

$$N^c = \text{points not in class } c$$

We get our score function:

$$F = \sum_{i \in \mathcal{B}} \frac{1}{|P^{c_i}||N^{c_i}|} \sum_{k \in P^{c_i} \setminus \{i\}} \sum_{j \in N^{c_i}} \boxed{y^i_{kj}} (\varphi(x_i, x_k, w) - \varphi(x_i, x_j, w))$$

Way of penalizing bad ranking

$$\boxed{y^i_{jk} = \begin{cases} 1 & k \text{ ranked higher than } j. \\ 0 & 0 \text{ if } j = k. \\ -1 & \text{else.} \end{cases}}$$

FEW-SHOT LEARNING BY OPTIMIZING mAP

SETUP

$$\mathcal{B} = \{x_1, \ldots, x_N\}$$

$$\text{Classes} = \{1, 2, \ldots, \mathcal{C}\}$$

$$P^c = \text{points in class } c$$

$$N^c = \text{points not in class } c$$

We get our score function:

$$F = \sum_{i \in \mathcal{B}} \frac{1}{|P^{c_i}||N^{c_i}|} \sum_{k \in P^{c_i} \setminus \{i\}} \sum_{j \in N^{c_i}} y^i_{kj} (\varphi(x_i, x_k, w) - \varphi(x_i, x_j, w))$$

And finally, our loss function:

$$L(y, y') = 1 - \text{mAP}$$

RESULTS

WHAT THEY FOUND

	Classification				Retrieval	
	1-shot		5-shot		1-shot	
	5-way	20-way	5-way	20-way	5-way	20-way
Siamese	**98.8**	95.5	-	-	98.6	95.7
Matching Networks [1]	98.1	93.8	98.9	98.5	-	-
Prototypical Networks [17]	**98.8**	**96.0**	99.7	**98.9**	-	-
MAML [18]	98.7	95.8	**99.9**	**98.9**	-	-
ConvNet w/ Memory [19]	98.4	95.0	99.6	98.6	-	-
mAP-SSVM (ours)	98.6	95.2	99.6	98.6	98.6	95.7
mAP-DLM (ours)	**98.8**	95.4	99.6	98.6	**98.7**	**95.8**

Omniglot

RESULTS

WHAT THEY FOUND

	Classification		Retrieval	
	5-way		5-way	20-way
	1-shot	5-shot	1-shot	1-shot
Baseline Nearset Neighbors*	41.08 ± 0.70 %	51.04 ± 0.65 %	-	-
Matching Networks* [1]	43.40 ± 0.78 %	51.09 ± 0.71 %	-	-
Matching Networks FCE* [1]	43.56 ± 0.84 %	55.31 ± 0.73 %	-	-
Meta-Learner LSTM* [2]	43.44 ± 0.77 %	60.60 ± 0.71 %	-	-
Prototypical Networks [17]	49.42 ± 0.78%	**68.20** ± 0.66 %	-	-
MAML [18]	48.70 ± 1.84 %	63.11 ± 0.92 %	-	-
Siamese	48.42 ± 0.79 %	-	51.24 ± 0.57 %	22.66 ± 0.13 %
mAP-SSVM (ours)	**50.32** ± 0.80 %	63.94 ± 0.72 %	52.85 ± 0.56 %	**23.87** ± 0.14 %
mAP-DLM (ours)	50.28 ± 0.80 %	63.70 ± 0.70 %	**52.96** ± 0.55 %	23.68 ± 0.13 %

Mini-Imagenet

THANKS!


© MARK ANDERSON
WWW.ANDERTOONS.COM
ANDERSON
"I'm here about the details."

REFERENCES

- Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-Shot Learning Through an Information Retrieval Lens, In Advances in Neural Information Processing Systems, 2252-2262, 2017. https://arxiv.org/abs/1707.02610
- Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In Advances in Neural Information Processing Systems, pages 3630–3638, 2016. https://arxiv.org/pdf/1606.04080.pdf
- Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In International Conference on Learning Representations, volume 1, page 6, 2017. https://openreview.net/pdf?id=rJY0-Kcll