

Intro to Deep Learning

Boris Zubarev



@bobazooba

Classical Machine Learning



Deep Learning



Сравнение методов

Classical Machine Learning

- Основной упор на построение фичей
- В модели меняются только гиперпараметры
- В сложных задачах очень сложный процесс
- Очень большое количество данных практически не улучшает результат

Deep Learning

- Основной упор на задачу и архитектуру (сейчас реже)
- Возможность использовать претренированные модели для дообучения и даже БЕЗ дообучения
- Увеличение качества за счет большого объема данных
- Большая гибкость в решении задач

Сравнение методов

Classical Machine Learning

- Основной упор на построение фичей
- В модели меняются только гиперпараметры
- В сложных задачах очень сложный процесс
- Очень большое количество данных практически не улучшает результат

Deep Learning

- Основной упор на задачу и архитектуру (сейчас реже)
- Возможность использовать претренированные модели для дообучения и даже БЕЗ дообучения
- Увеличение качества за счет большого объема данных
- Большая гибкость в решении задач



Качество, как правило, лучше

Увеличение размеров моделей

Что умеют большие модели

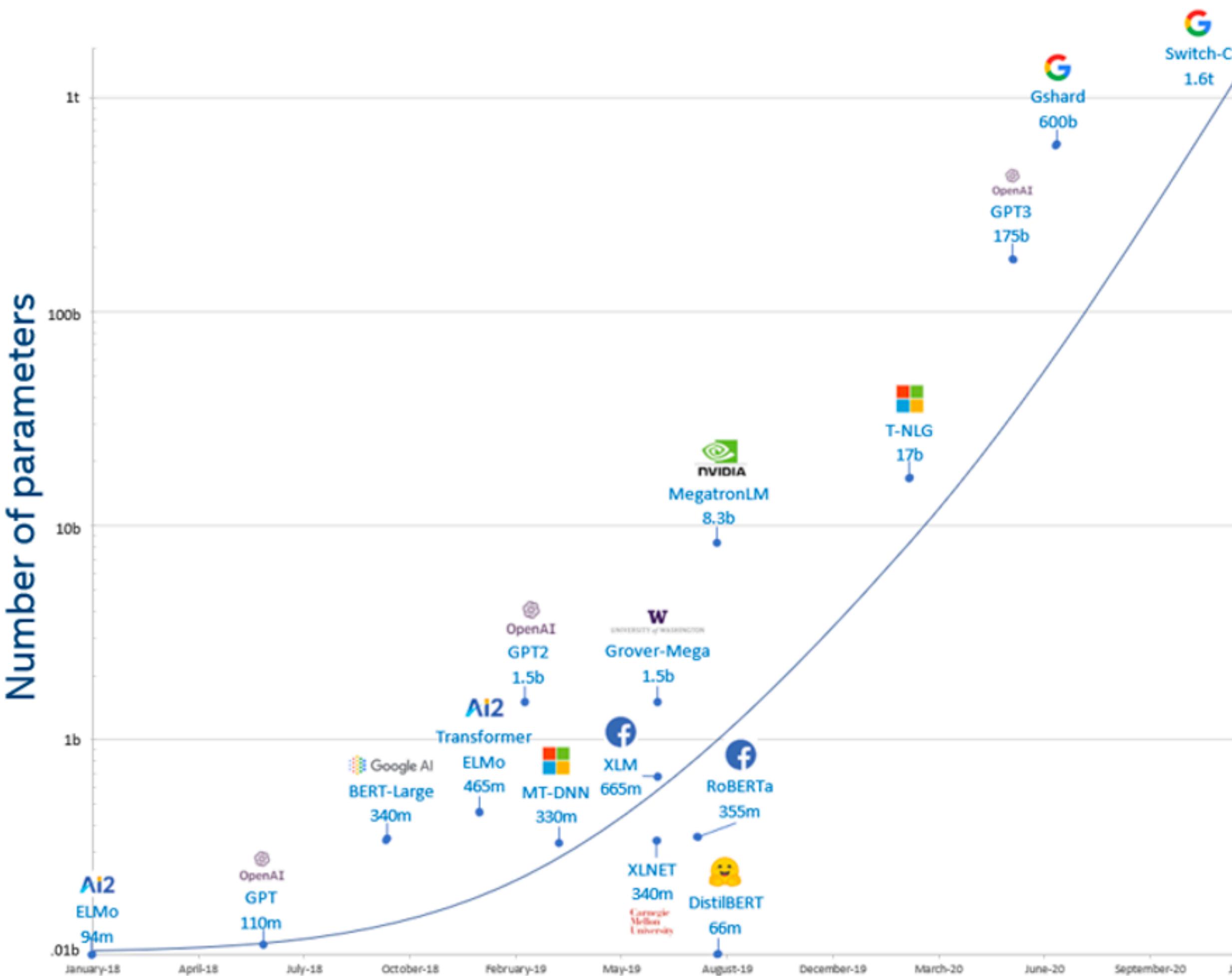
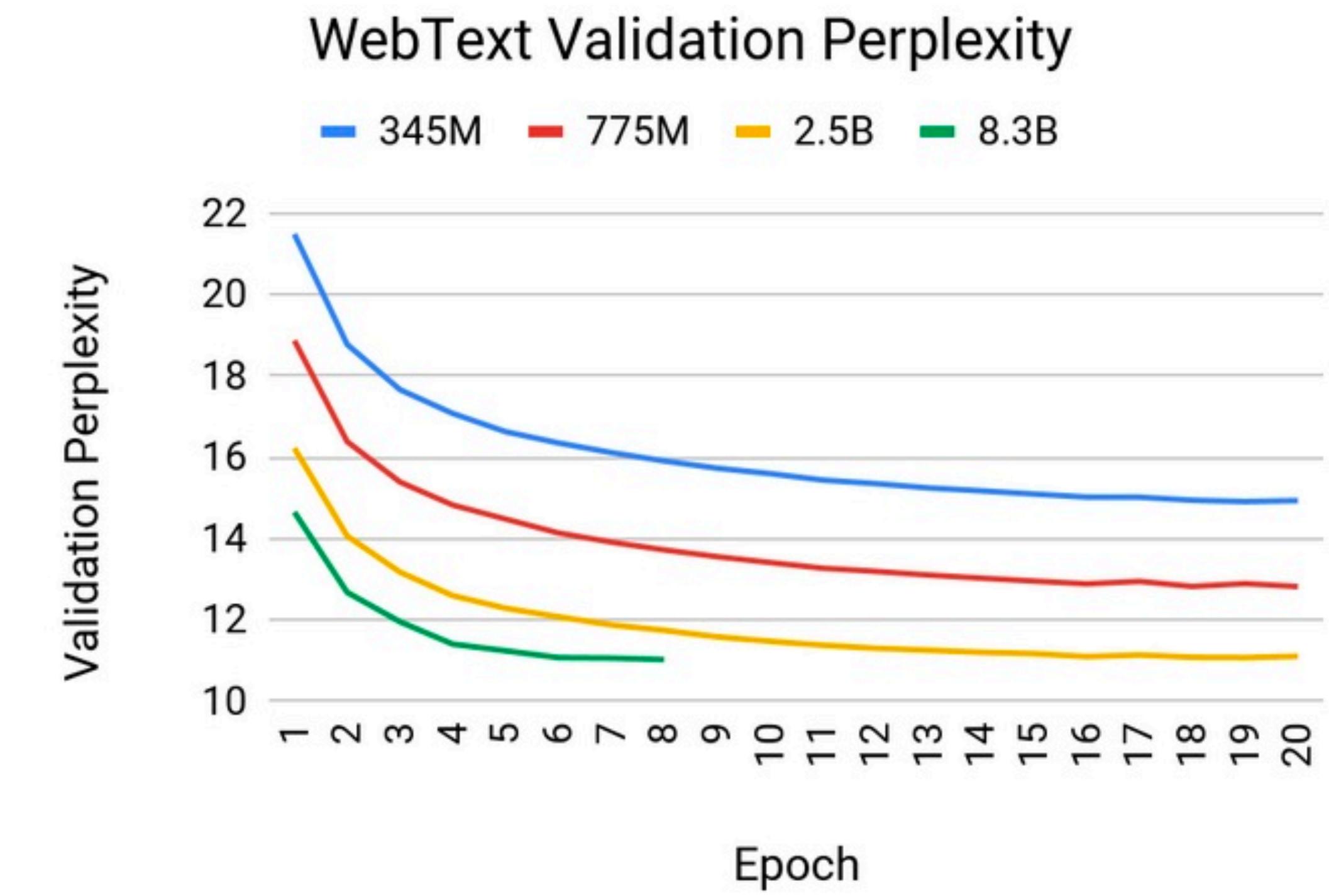
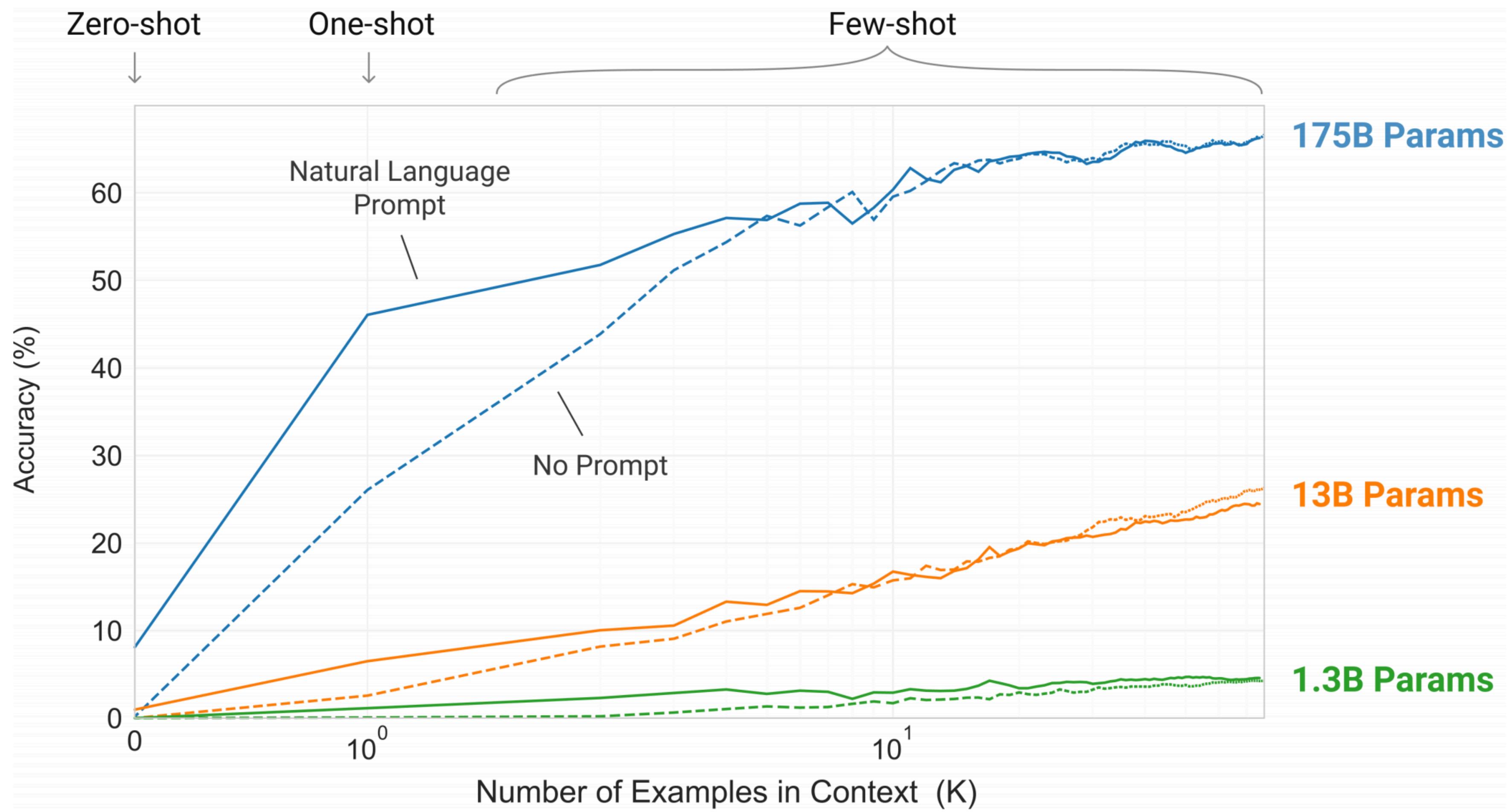


Figure 1: Exponential growth of number of parameters in DL models



Увеличение размеров моделей

Что умеют большие модели



Задачи

Основные темы исследований

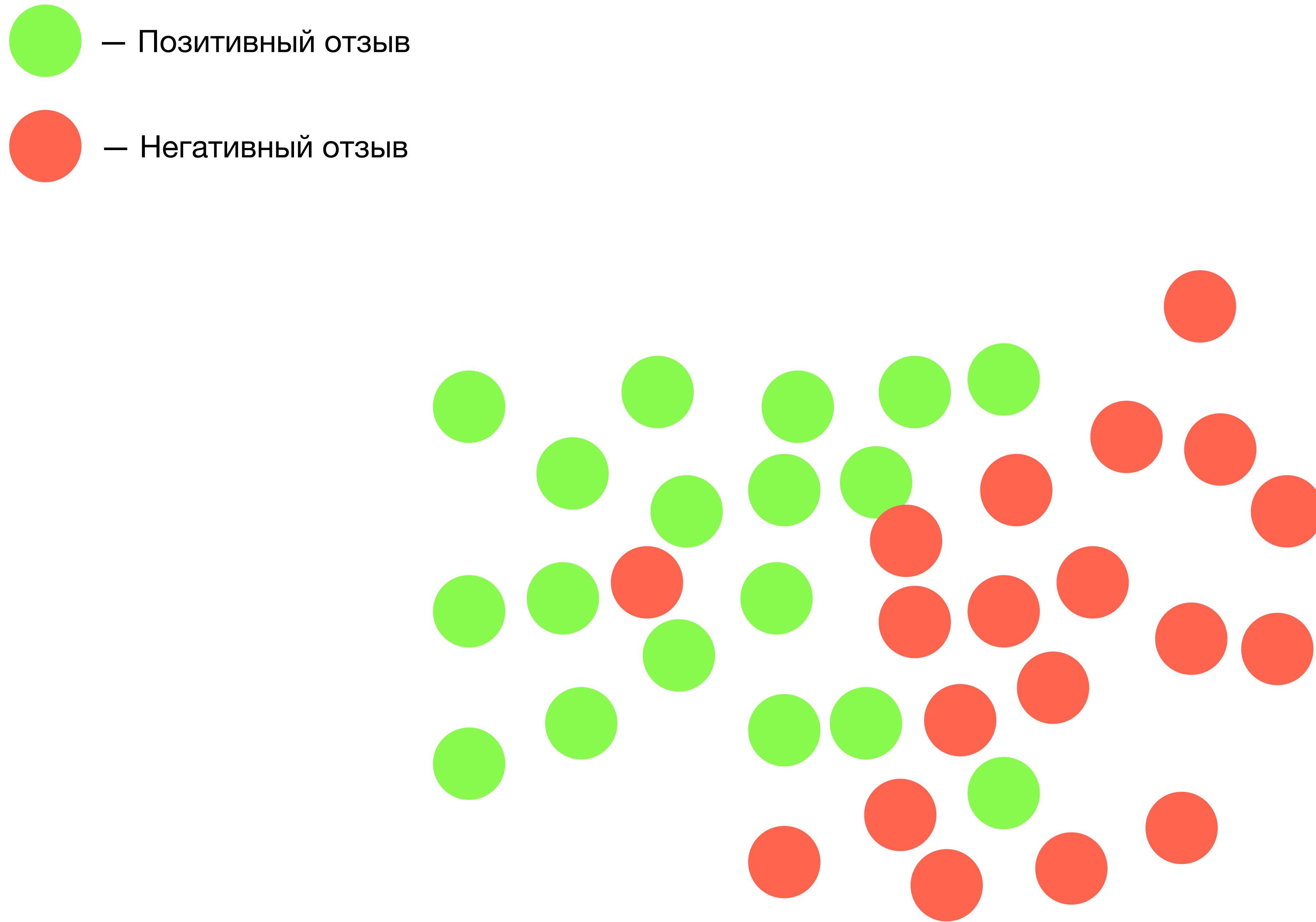
- Языковые модели
 - Их адаптация для других задач
- Диалоговые системы
- Вопросно-ответные системы
- Переводчики
- Суммаризация
- Мультимодальные системы (CV и NLP)
 - Natural Language Inference
 - Улучшение архитектур
 - Новые задачи для претренировки
 - Style Transfer
 - Syntax Parsing

Задачи

Основные темы исследований

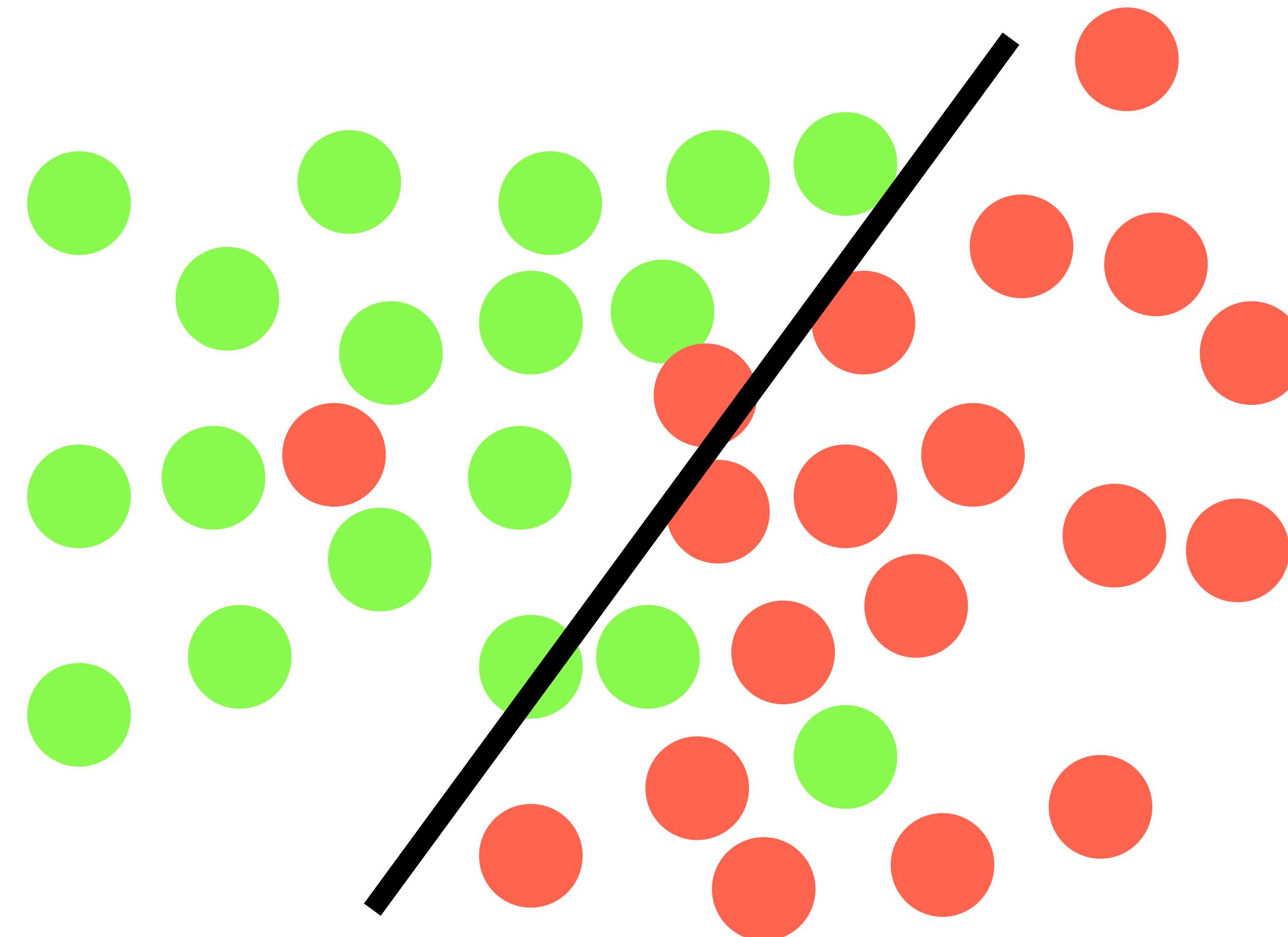
- Языковые модели
 - Их адаптация для других задач
- Диалоговые системы
- Вопросно-ответные системы
- Переводчики
- Суммаризация
- Мультимодальные системы (CV и NLP)
- Natural Language Inference
- Улучшение архитектур
- Новые задачи для претренировки
- Style Transfer
- Syntax Parsing

Machine Learning Recap



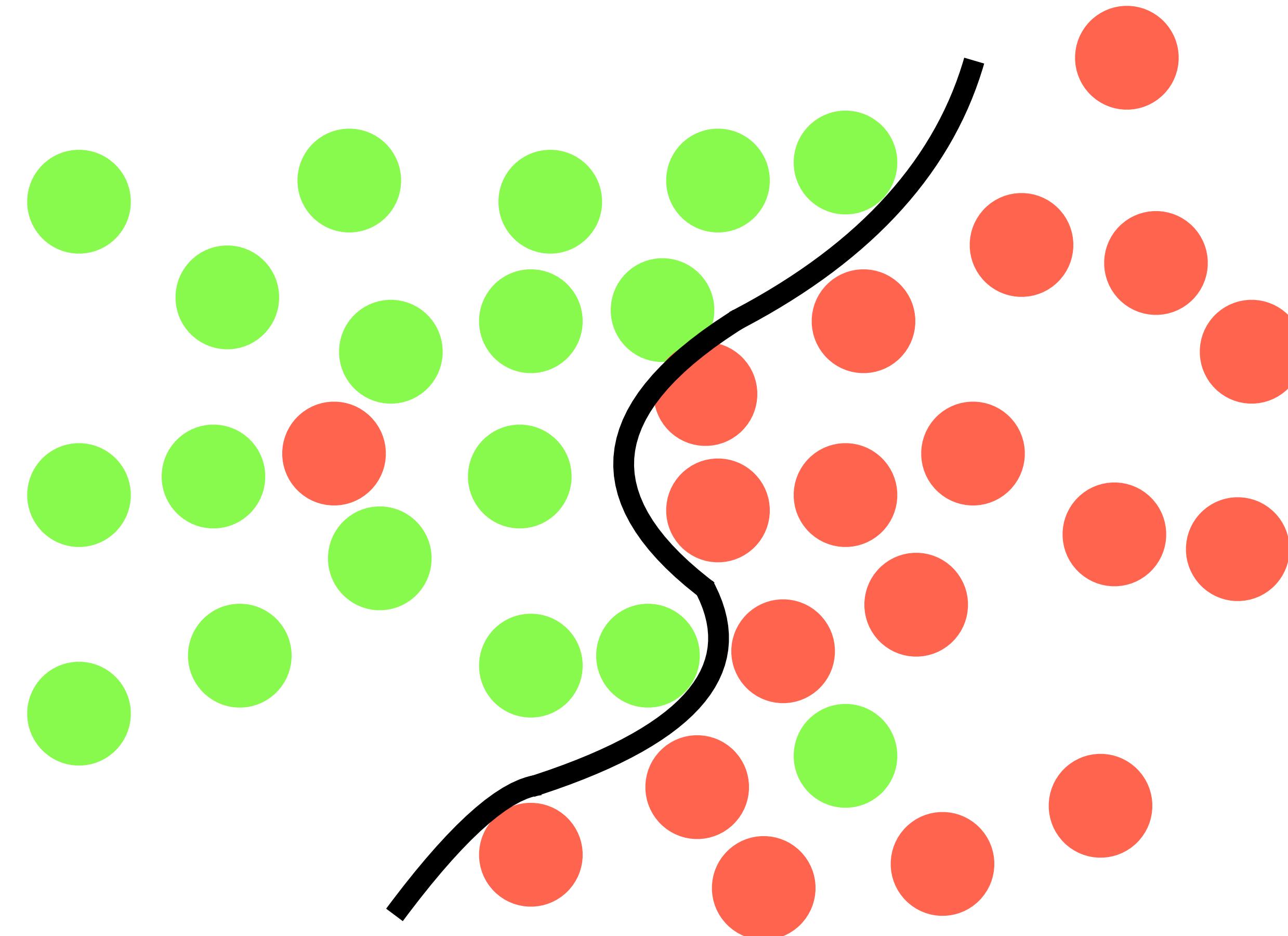
- Позитивный отзыв
- Негативный отзыв

Логистическая регрессия



- Позитивный отзыв
- Негативный отзыв

Нейронная сеть



* и другие нелинейные модели

Логистическая регрессия

Логистическая регрессия

Перевод данных в фичи

Логистическая регрессия

Перевод данных в фичи

Сырые данные

Логистическая регрессия

Перевод данных в фичи

Сырые данные → *Выделение фичей*

Логистическая регрессия

Перевод данных в фичи

Сырые данные → Выделение фичей

- Правила
- Статистики по словам
- Статистики по фразам
- Наличие именованных сущностей
- You name it

Логистическая регрессия

Перевод данных в фичи

Сырые данные → Выделение фичей →

- Правила
- Статистики по словам
- Статистики по фразам
- Наличие именованных сущностей
- You name it

	Фича 1	Фича 2	Фича 3	target
Пример 1	0.143986	0.905461	0.371917	1
Пример 2	0.453960	0.805769	0.617273	0
Пример 3	0.378570	0.121712	0.629929	0
Пример 4	0.684620	0.077442	0.137674	0
Пример 5	0.309269	0.710231	0.029111	1
Пример 6	0.456082	0.283086	0.856591	0
Пример 7	0.322638	0.395607	0.681704	1
Пример 8	0.922954	0.959641	0.953247	1
Пример 9	0.679032	0.149385	0.799034	1
Пример 10	0.606238	0.332173	0.725321	0

Логистическая регрессия

Перевод данных в фичи

Сырые данные → Выделение фичей →

- Правила
- Статистики по словам
- Статистики по фразам
- Наличие именованных сущностей
- You name it

	Фича 1	Фича 2	Фича 3	target
Пример 1	0.143986	0.905461	0.371917	1
Пример 2	0.453960	0.805769	0.617273	0
Пример 3	0.378570	0.121712	0.629929	0
Пример 4	0.684620	0.077442	0.137674	0
Пример 5	0.309269	0.710231	0.029111	1
Пример 6	0.456082	0.283086	0.856591	0
Пример 7	0.322638	0.395607	0.681704	1
Пример 8	0.922954	0.959641	0.953247	1
Пример 9	0.679032	0.149385	0.799034	1
Пример 10	0.606238	0.332173	0.725321	0

Логистическая регрессия

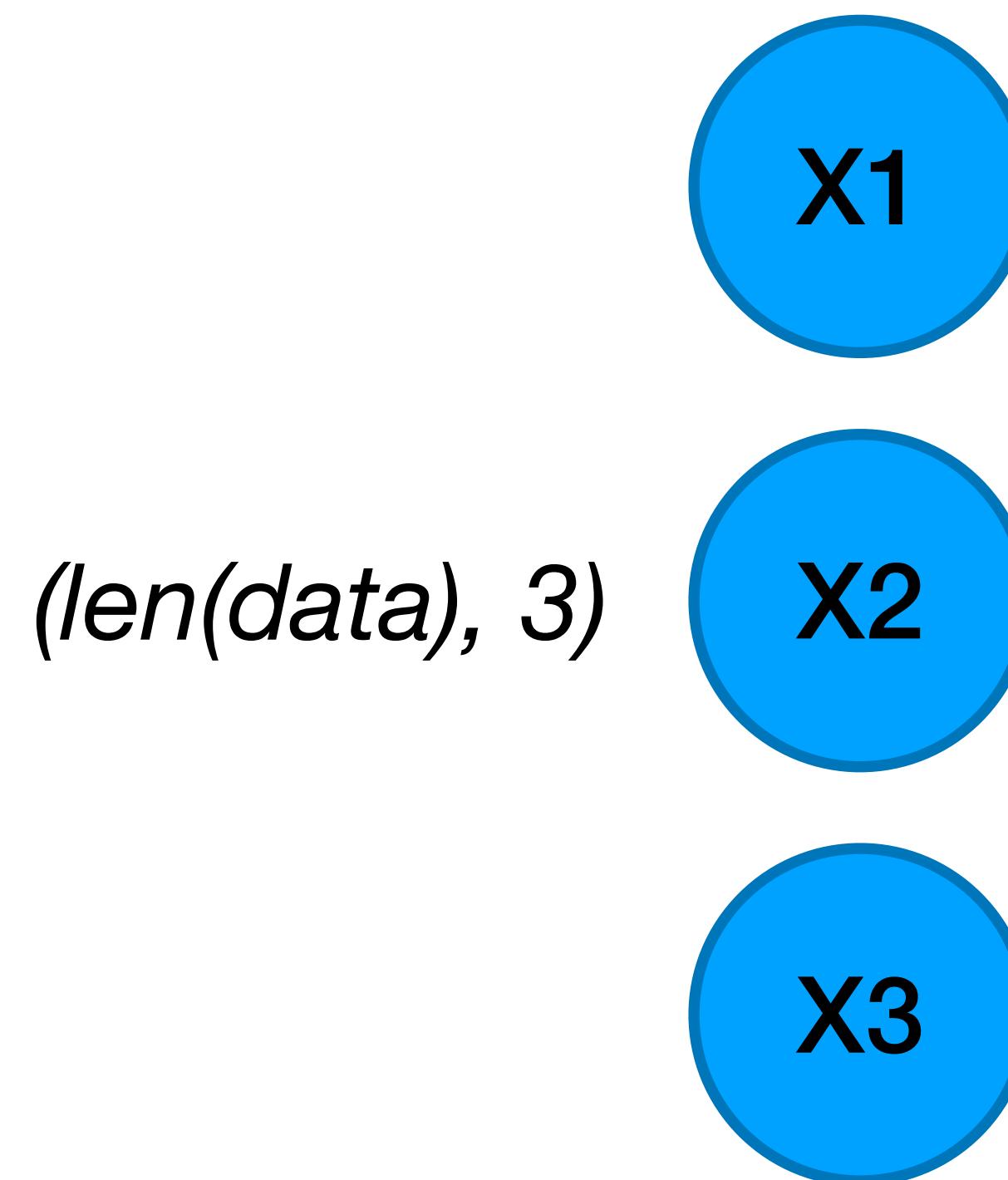
Inference

(len(data), 3)

	Фича 1	Фича 2	Фича 3
Пример 1	0.143986	0.905461	0.371917
Пример 2	0.453960	0.805769	0.617273
Пример 3	0.378570	0.121712	0.629929
Пример 4	0.684620	0.077442	0.137674
Пример 5	0.309269	0.710231	0.029111
Пример 6	0.456082	0.283086	0.856591
Пример 7	0.322638	0.395607	0.681704
Пример 8	0.922954	0.959641	0.953247
Пример 9	0.679032	0.149385	0.799034
Пример 10	0.606238	0.332173	0.725321

Логистическая регрессия

Inference



	Фича 1	Фича 2	Фича 3
Пример 1	0.143986	0.905461	0.371917
Пример 2	0.453960	0.805769	0.617273
Пример 3	0.378570	0.121712	0.629929
Пример 4	0.684620	0.077442	0.137674
Пример 5	0.309269	0.710231	0.029111
Пример 6	0.456082	0.283086	0.856591
Пример 7	0.322638	0.395607	0.681704
Пример 8	0.922954	0.959641	0.953247
Пример 9	0.679032	0.149385	0.799034
Пример 10	0.606238	0.332173	0.725321

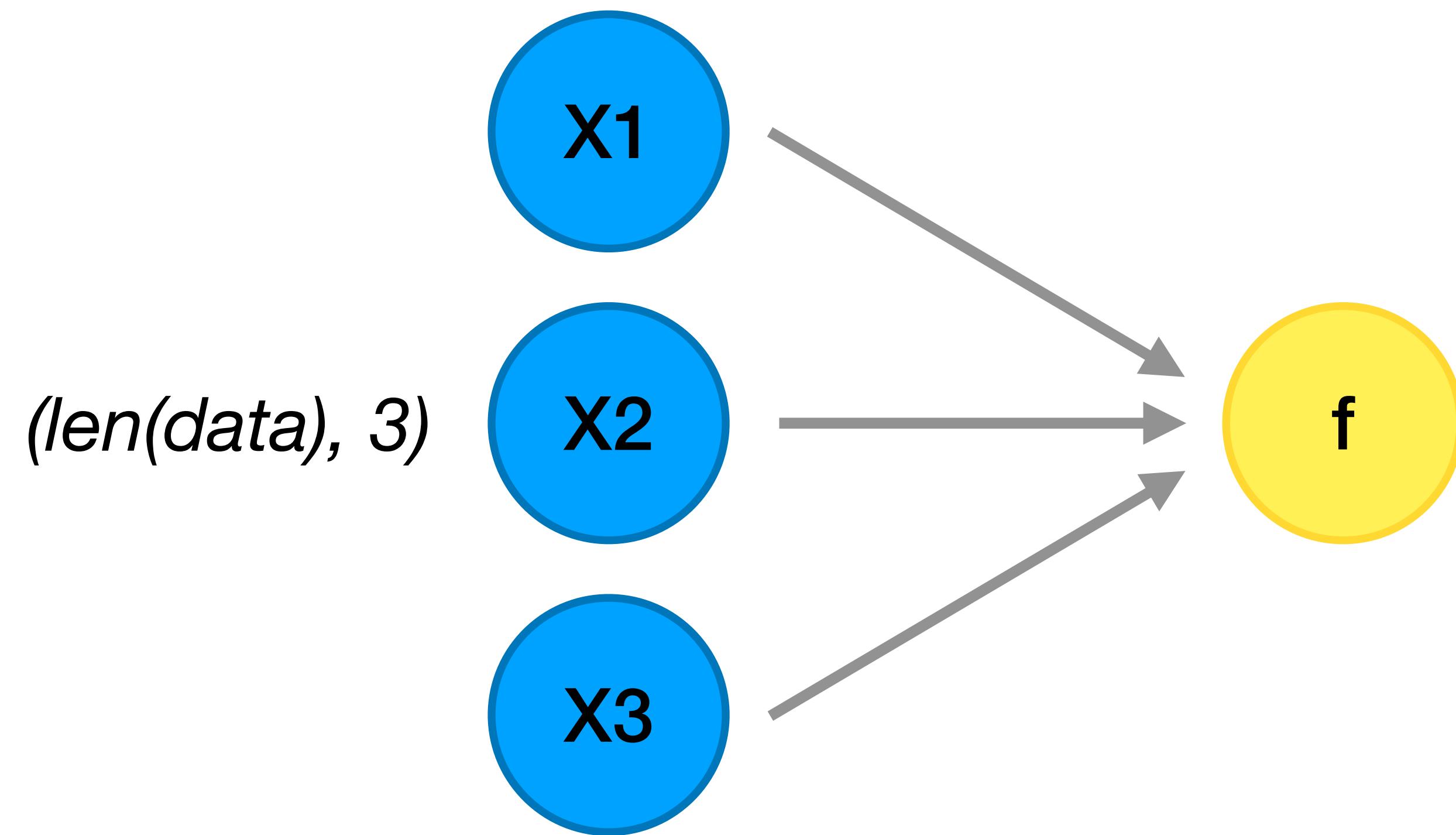
Логистическая регрессия

Inference



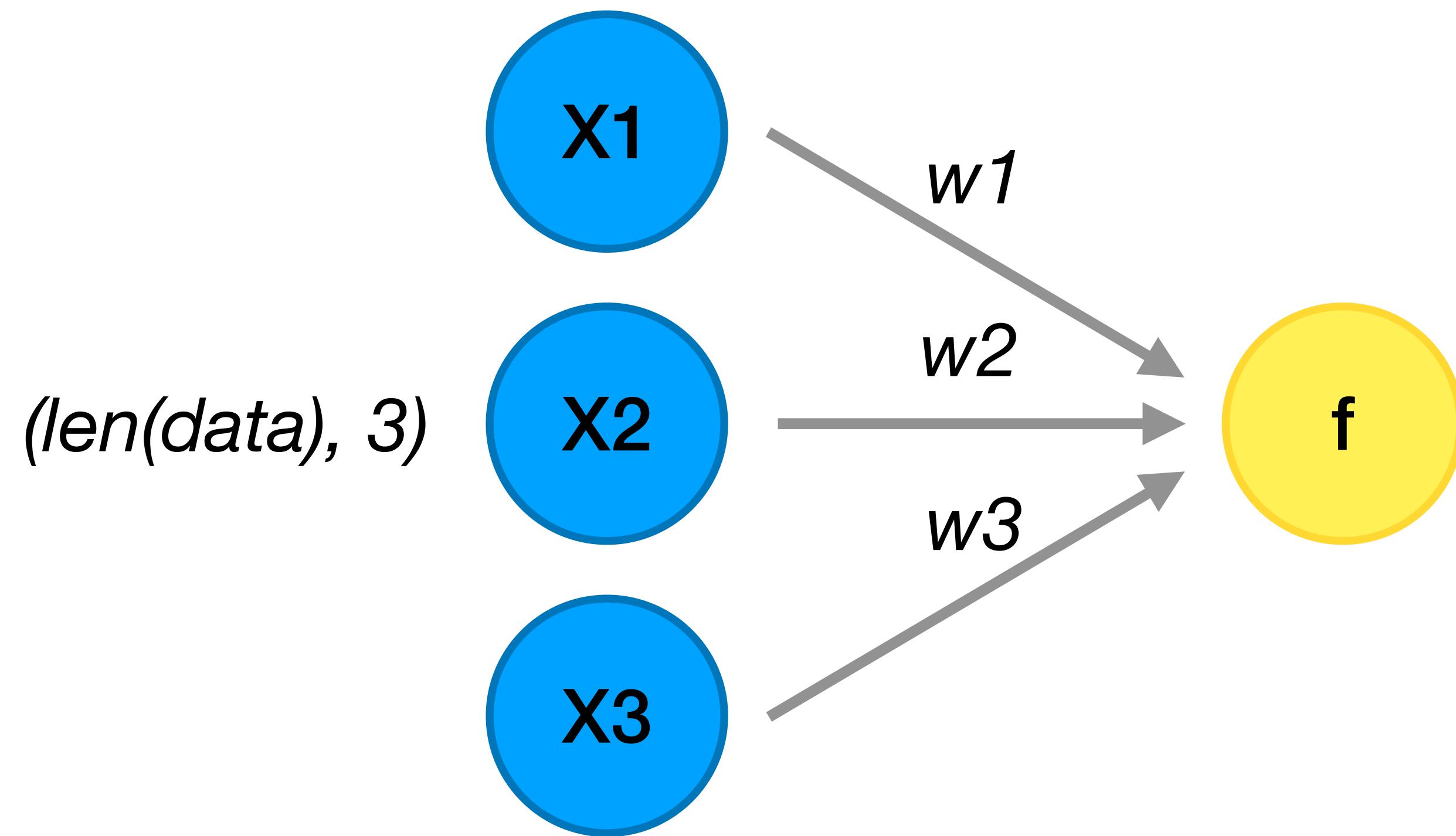
Логистическая регрессия

Inference



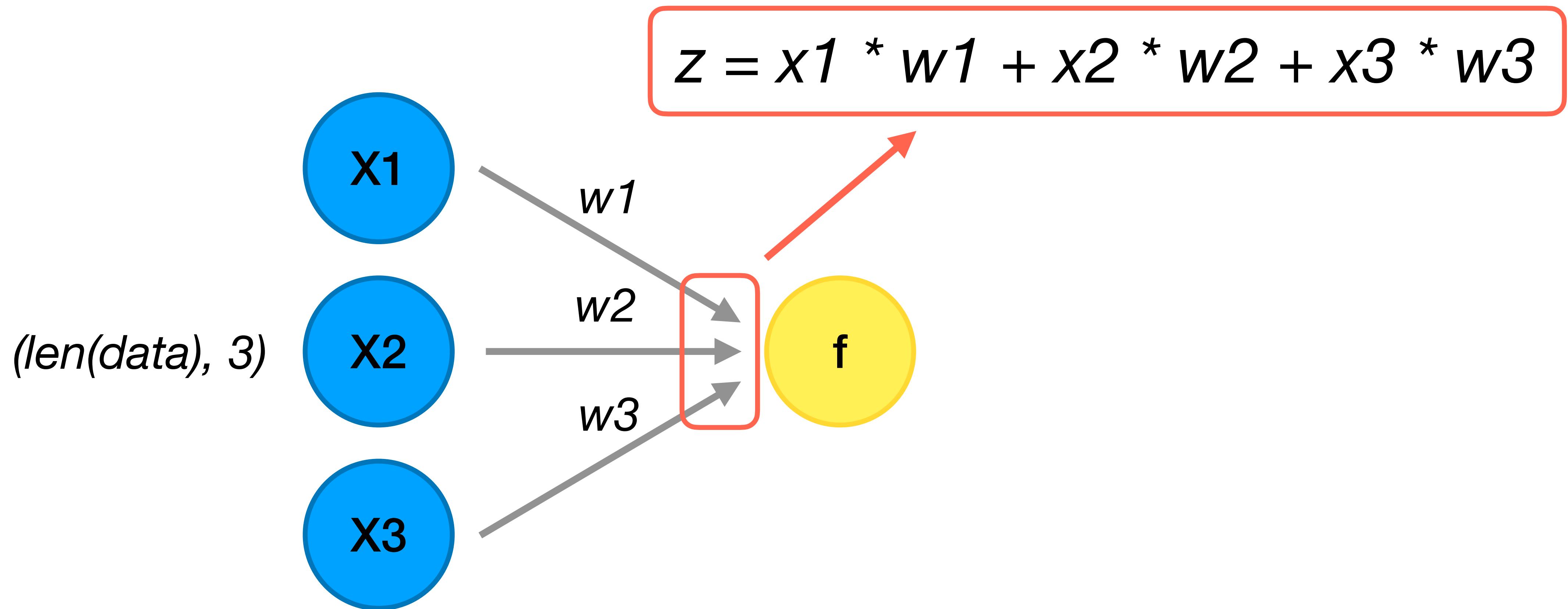
Логистическая регрессия

Inference



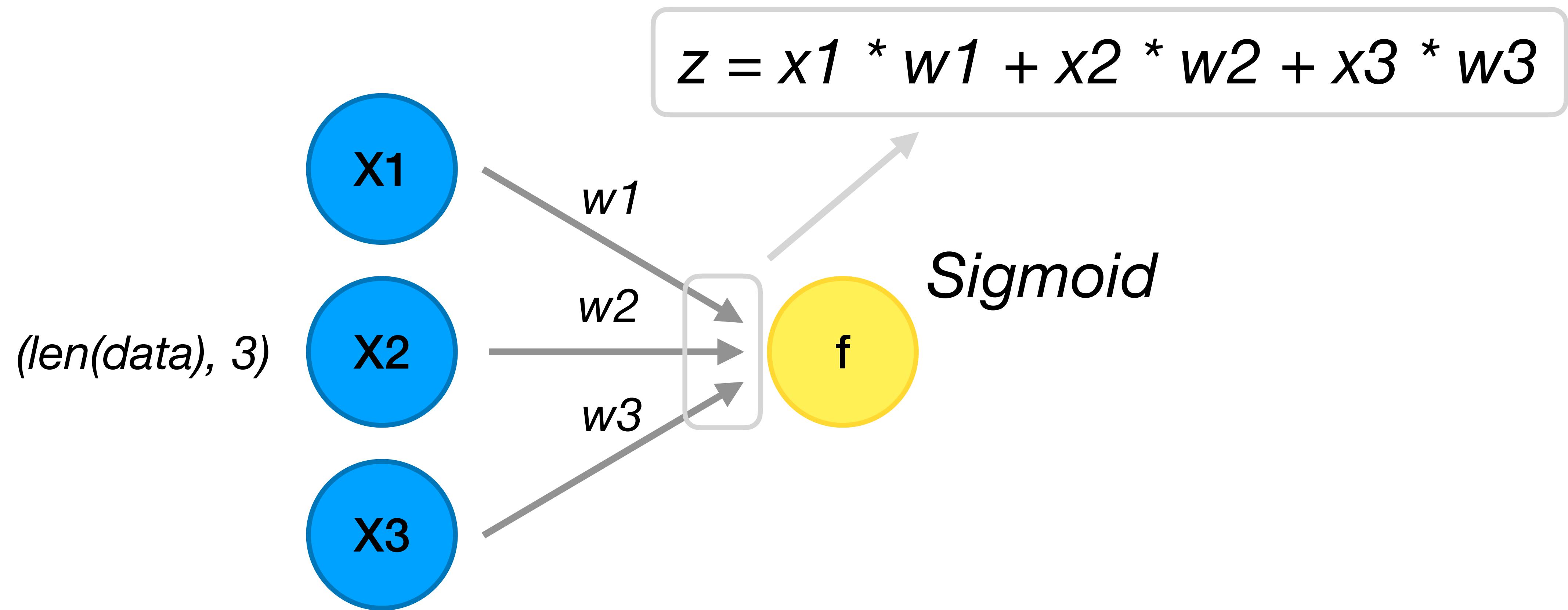
Логистическая регрессия

Inference



Логистическая регрессия

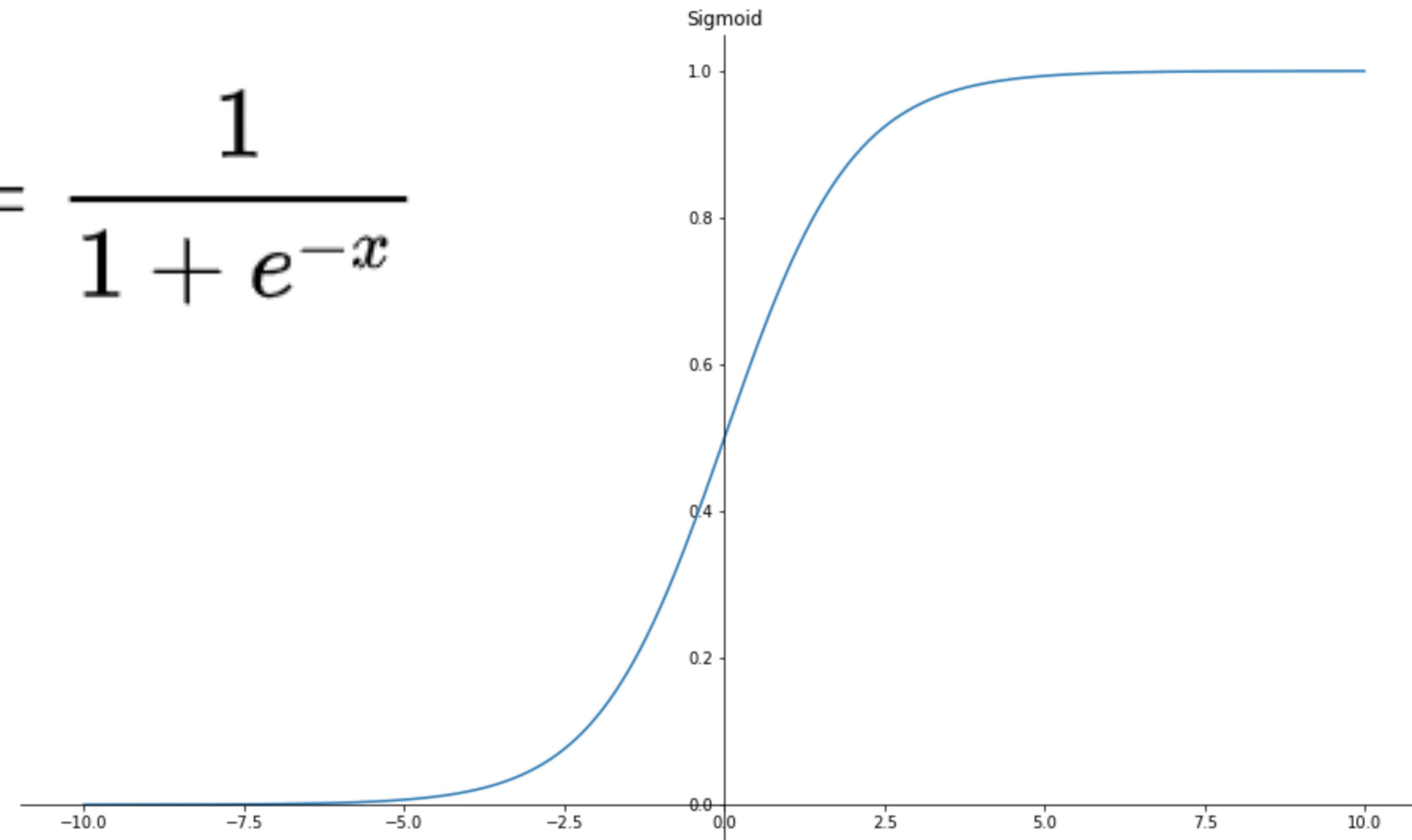
Inference



Логистическая регрессия

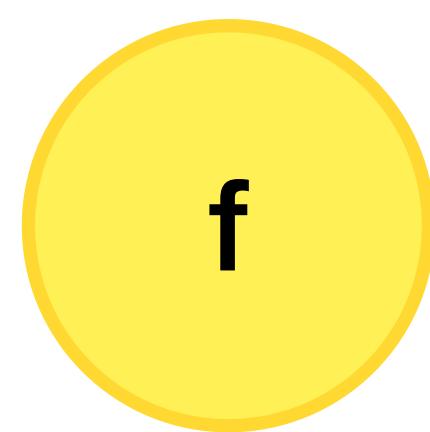
Inference – Sigmoid

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



Логистическая регрессия

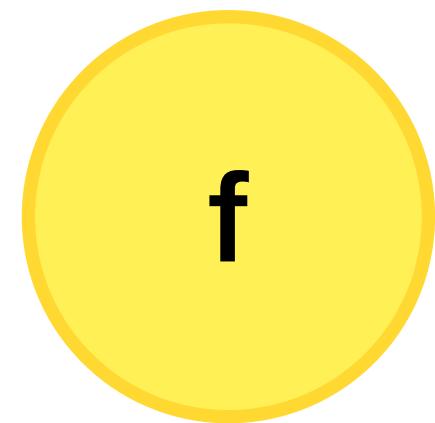
Inference — Sigmoid



Sigmoid

Логистическая регрессия

Inference — Sigmoid



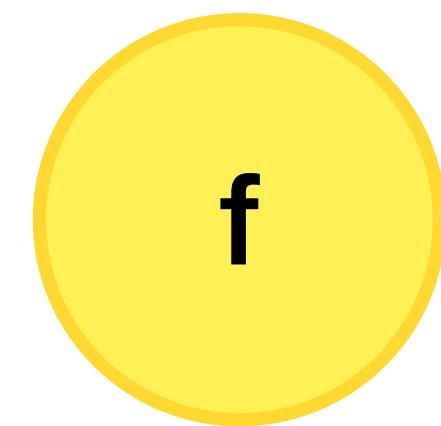
Sigmoid

$$= 1 / (1 + \exp(-x^* w + b))$$

Логистическая регрессия

Inference – Sigmoid

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



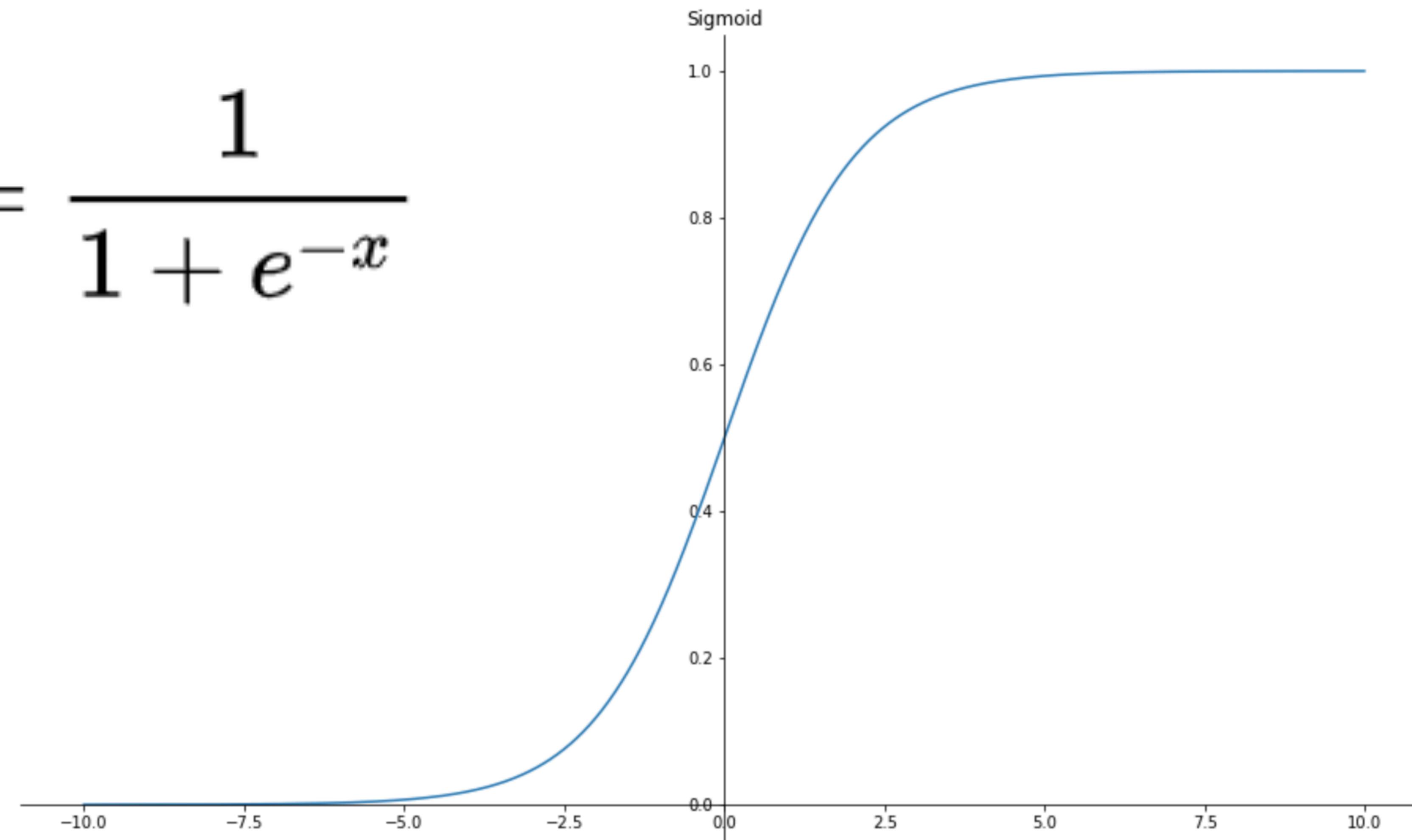
Sigmoid

$$= 1 / (1 + \exp(-x^* w + b))$$

Логистическая регрессия

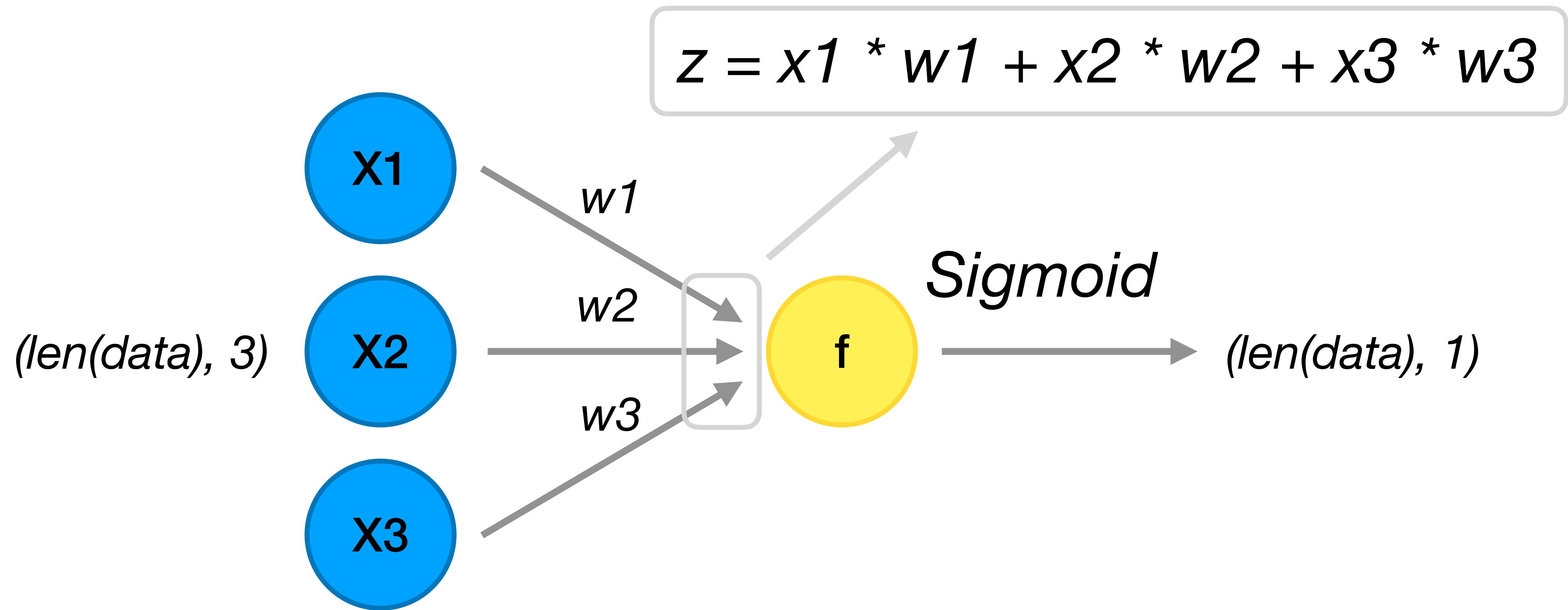
Inference – Sigmoid

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



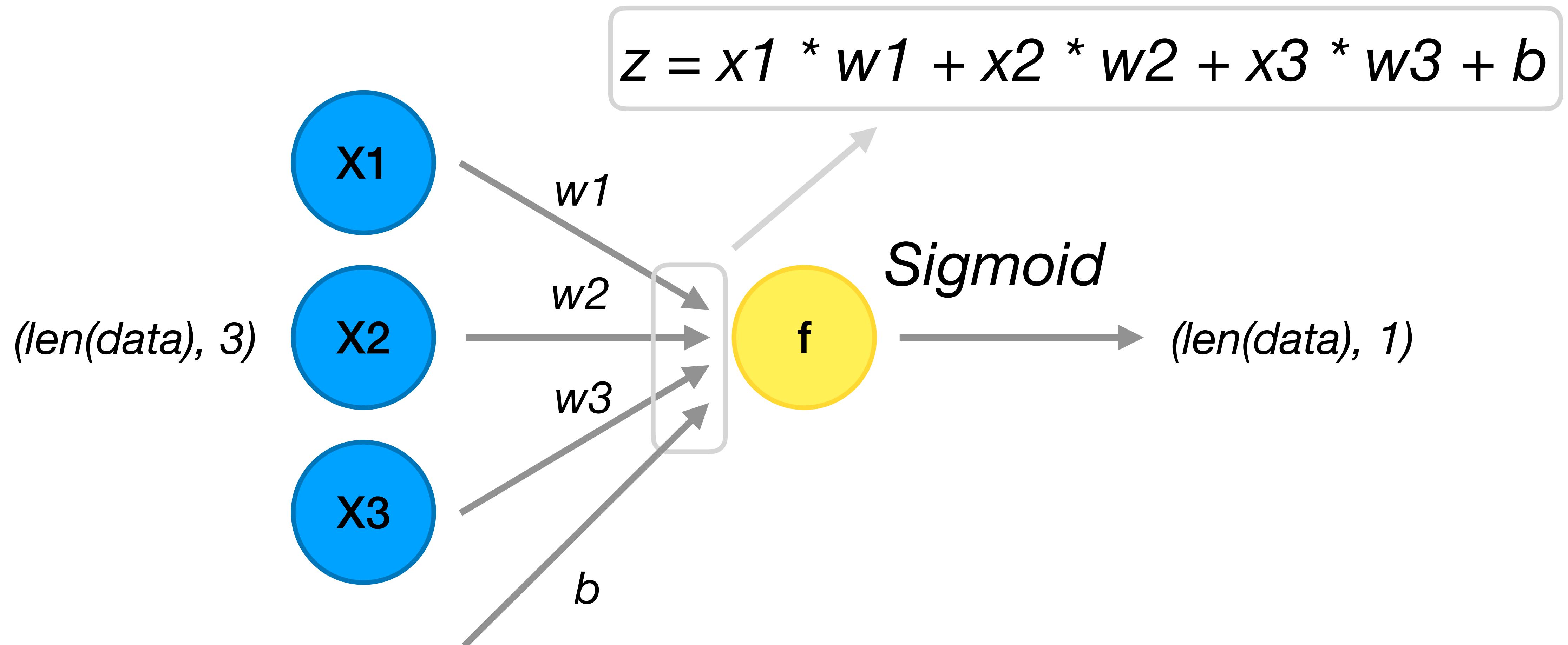
Логистическая регрессия

Inference



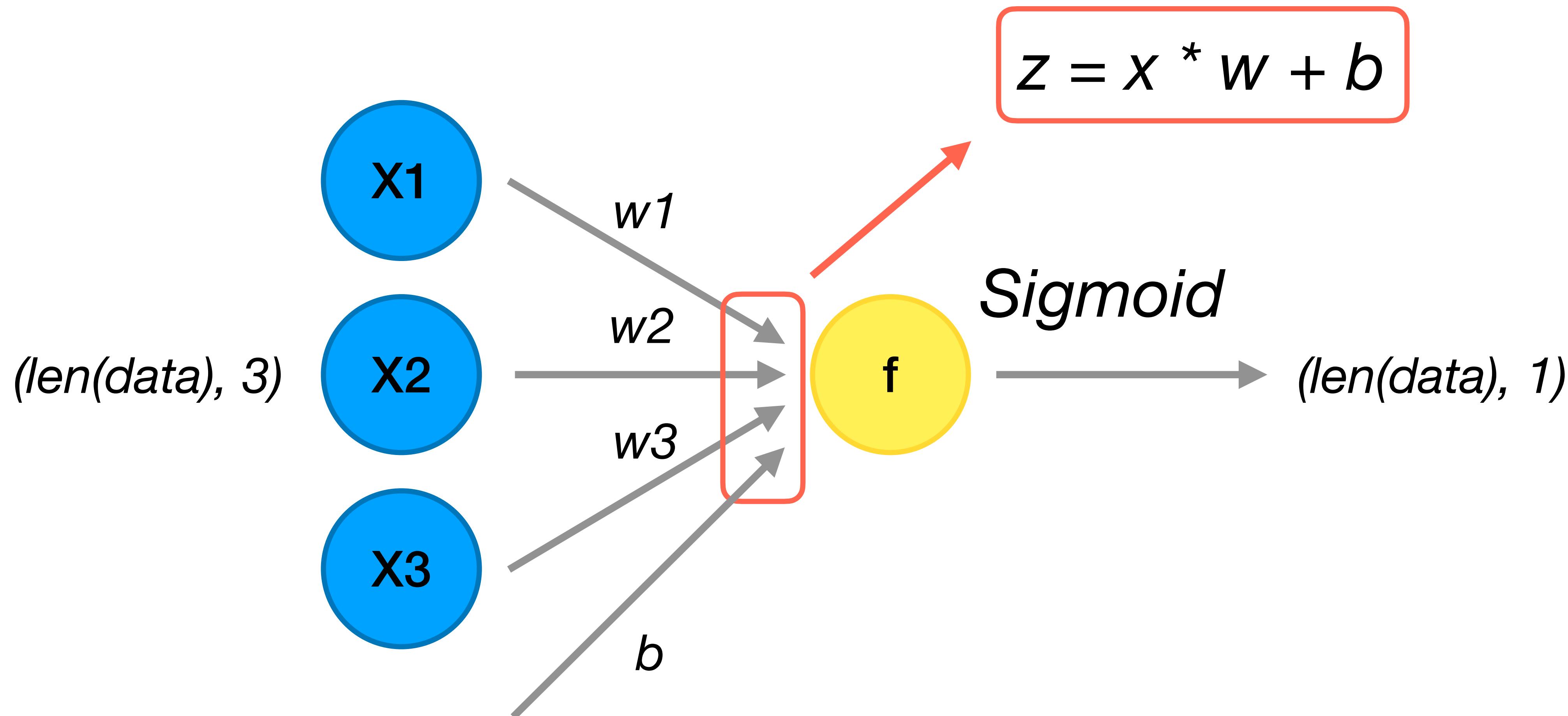
Логистическая регрессия

Inference



Логистическая регрессия

Inference



Логистическая регрессия

Dot Product

$$z = x^* w + b$$

Логистическая регрессия

Dot Product

$$z = x^* w + b = np.dot(x, w) + b$$

Логистическая регрессия

Dot Product

$$z = x^* w + b = np.dot(x, w) + b$$

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix}$$

$$AB = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} & a_{31}b_{13} + a_{32}b_{23} \end{pmatrix}$$

Логистическая регрессия

Dot Product

$$z = x^* w + b = np.dot(x, w) + b$$

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix}$$

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj} = \sum_{s=1}^n a_{sn}b_{sj}$$

$$AB = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} & a_{31}b_{13} + a_{32}b_{23} \end{pmatrix}$$

Логистическая регрессия

Dot Product

$$z = x^* w + b = np.dot(x, w) + b$$

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix}$$

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj} = \sum_{s=1}^n a_{sn}b_{sj}$$

$$AB = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} & a_{31}b_{13} + a_{32}b_{23} \end{pmatrix}$$

$$A.shape = (p, m)$$

$$B.shape = (n, k)$$

Логистическая регрессия

Dot Product

$$z = x^* w + b = np.dot(x, w) + b$$

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix}$$

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj} = \sum_{s=1}^n a_{sn}b_{sj}$$

$$AB = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} & a_{31}b_{13} + a_{32}b_{23} \end{pmatrix}$$

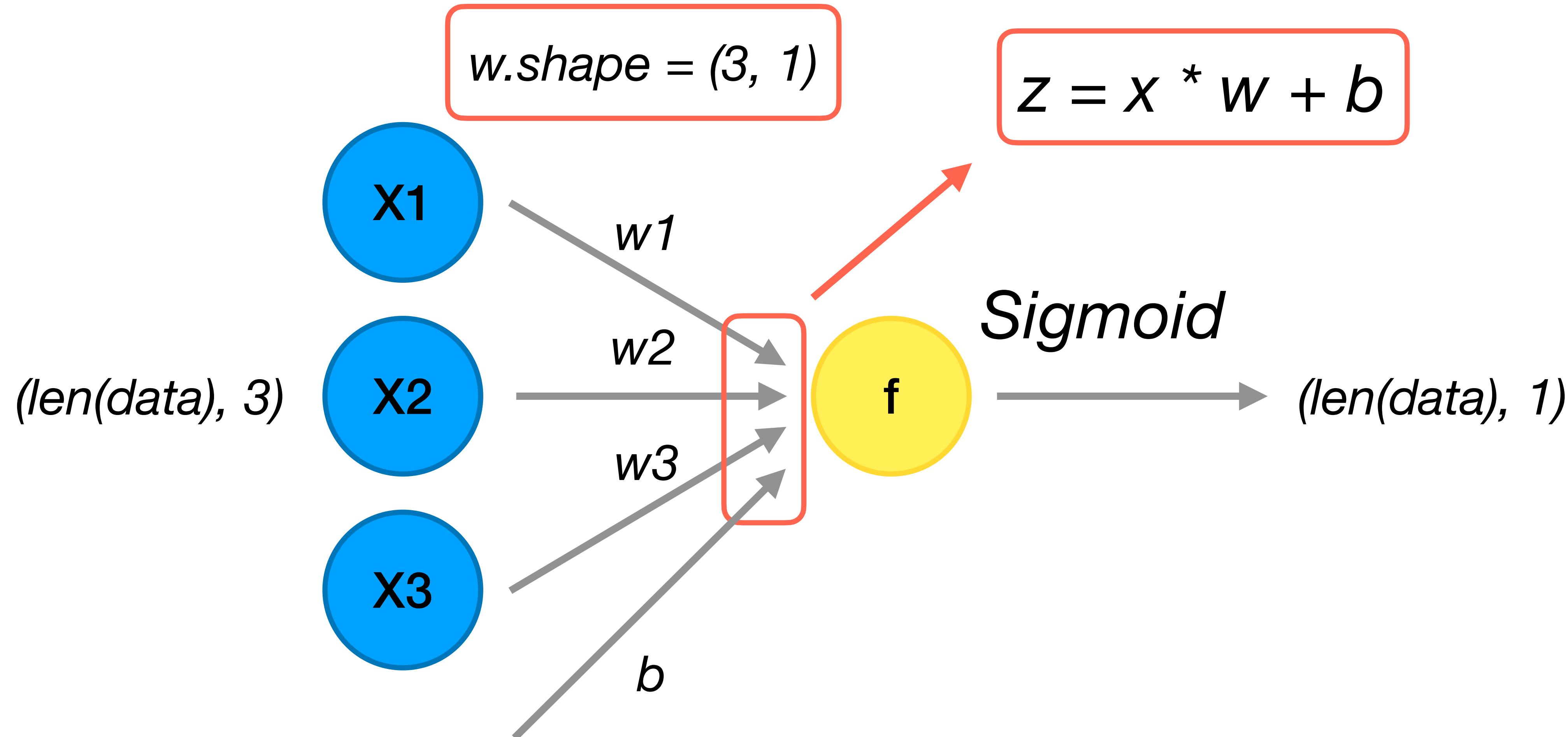
$A.shape = (p, m)$

$B.shape = (n, k)$

$np.dot(A, B).shape = (p, k) \text{ if } m == n$

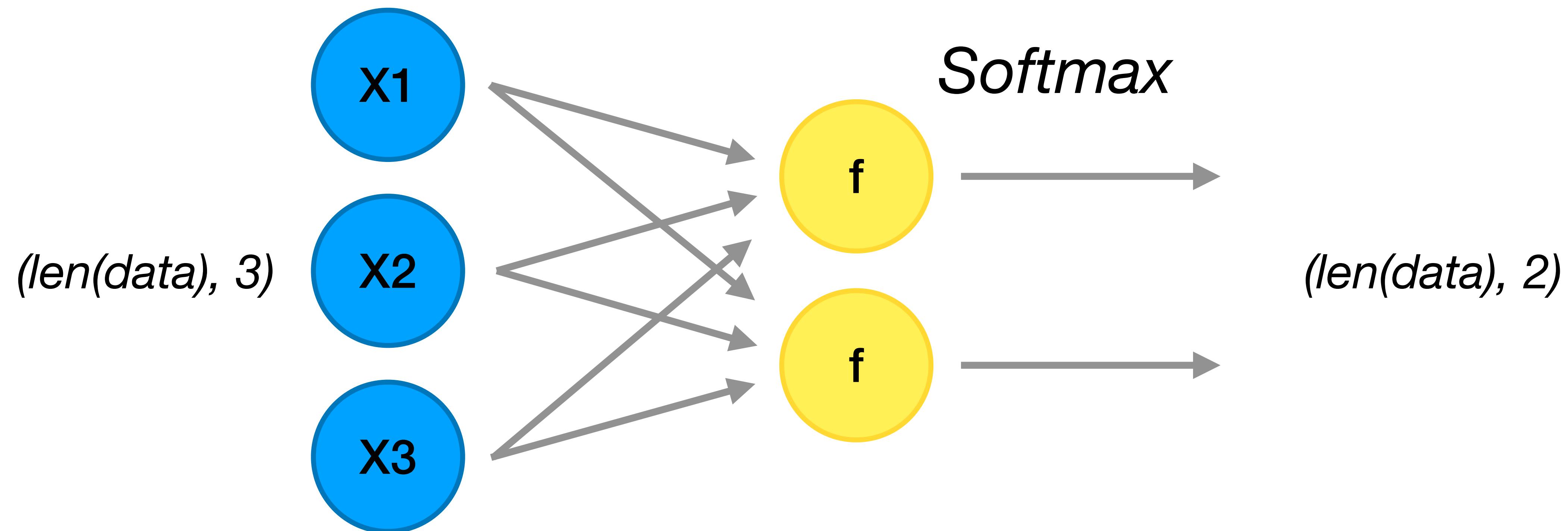
Логистическая регрессия

Inference



Логистическая регрессия

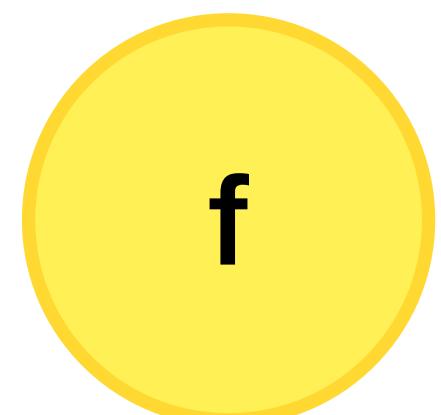
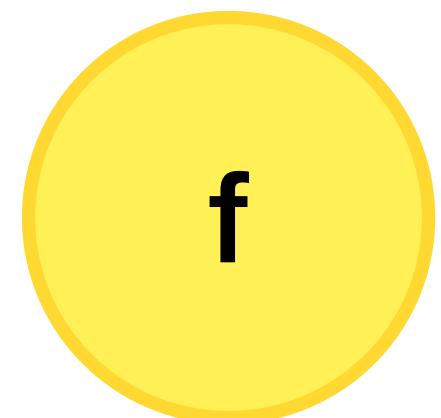
Inference



Логистическая регрессия

Inference — Softmax

Softmax



Логистическая регрессия

Inference — Softmax

Softmax

f

```
exp_scores = np.exp(pred)
```

f

```
softmax = exp_scores / exp_scores.sum()
```

Логистическая регрессия

Inference — Softmax

Softmax

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}}$$

f

```
exp_scores = np.exp(pred)
```

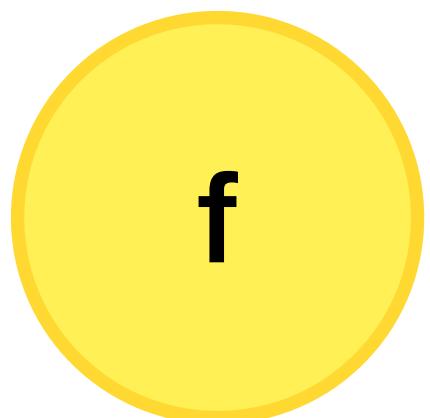
f

```
softmax = exp_scores / exp_scores.sum()
```

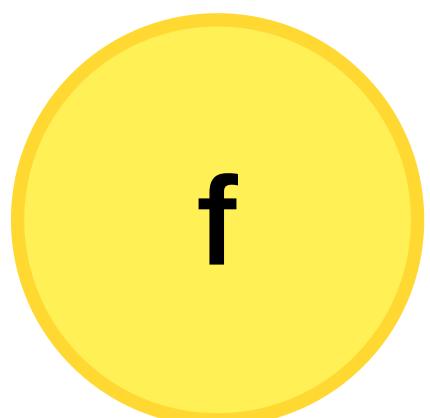
Логистическая регрессия

Inference – Softmax

Softmax

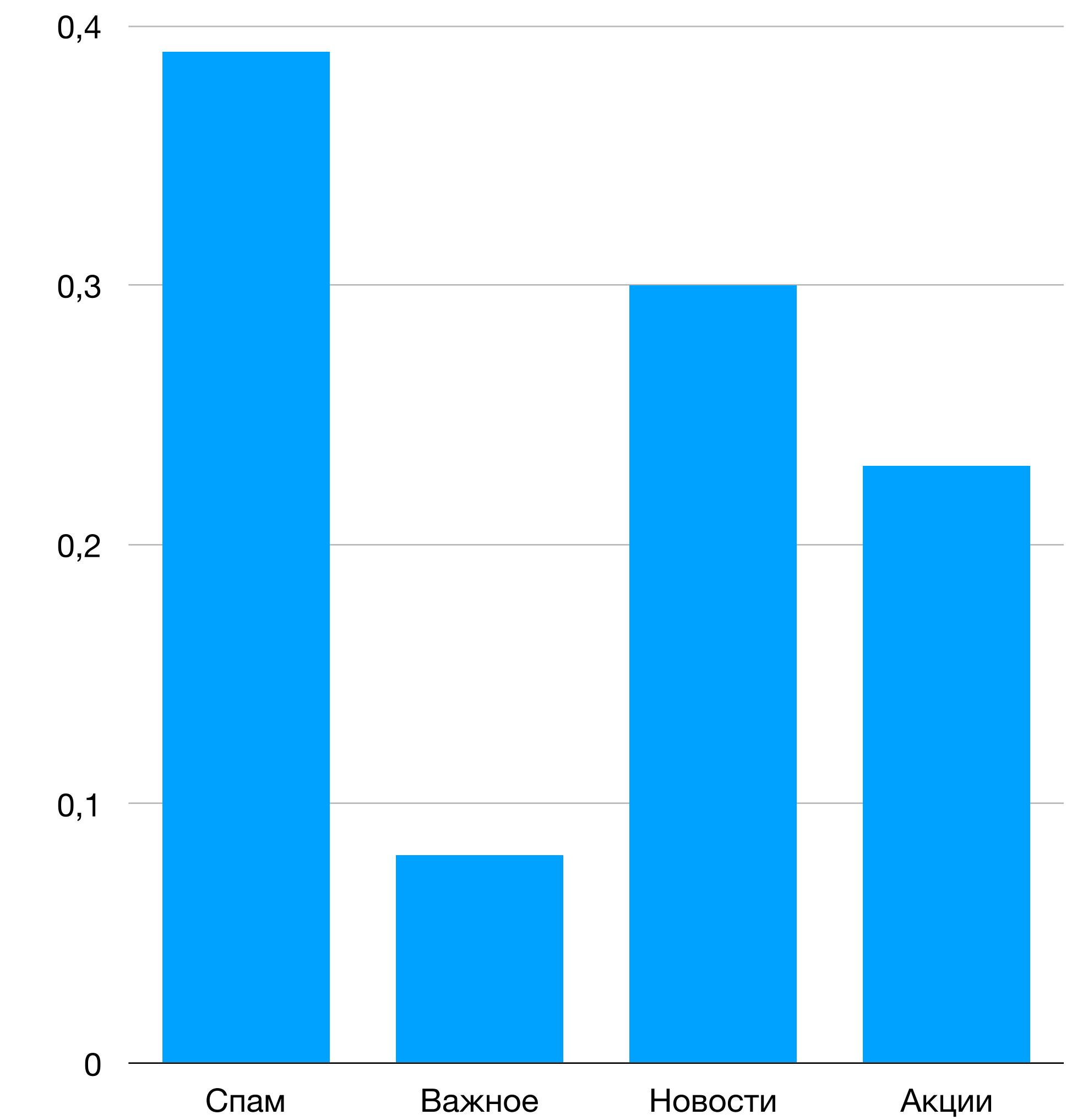


```
exp_scores = np.exp(pred)
```



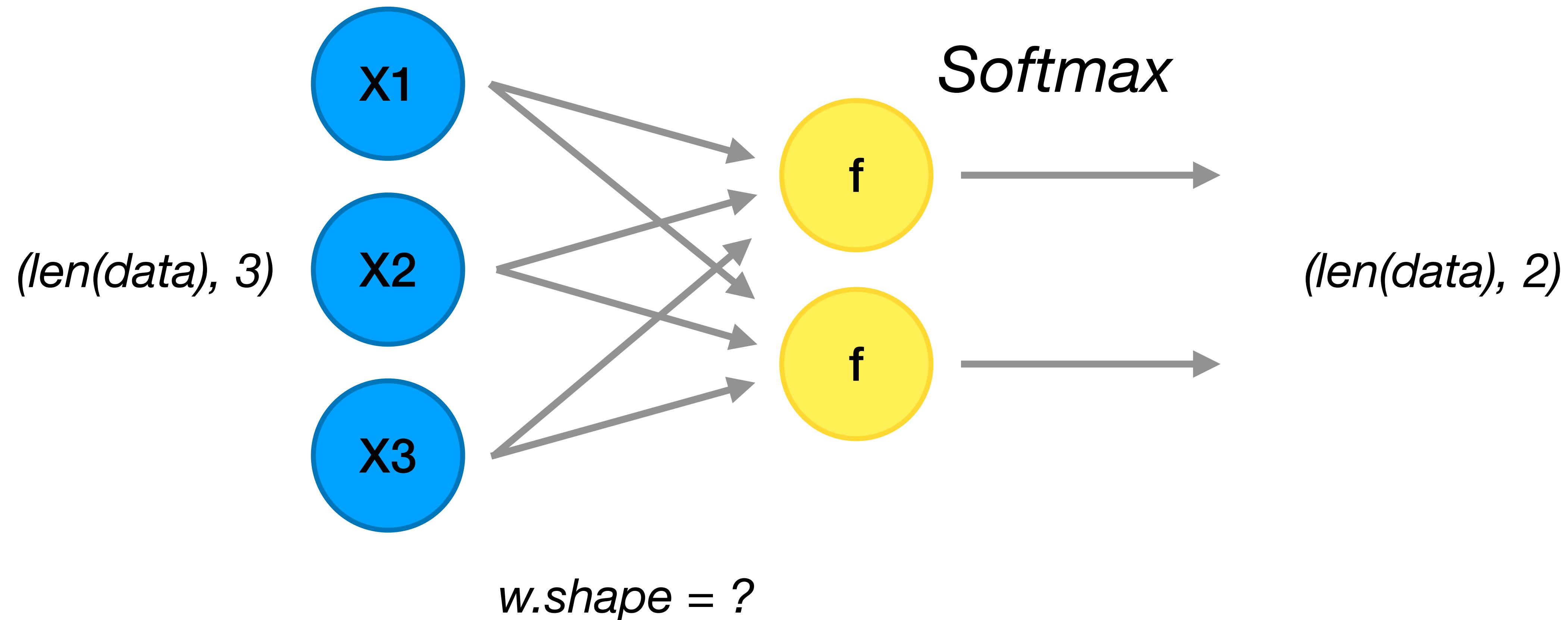
```
softmax = exp_scores / exp_scores.sum()
```

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}}$$



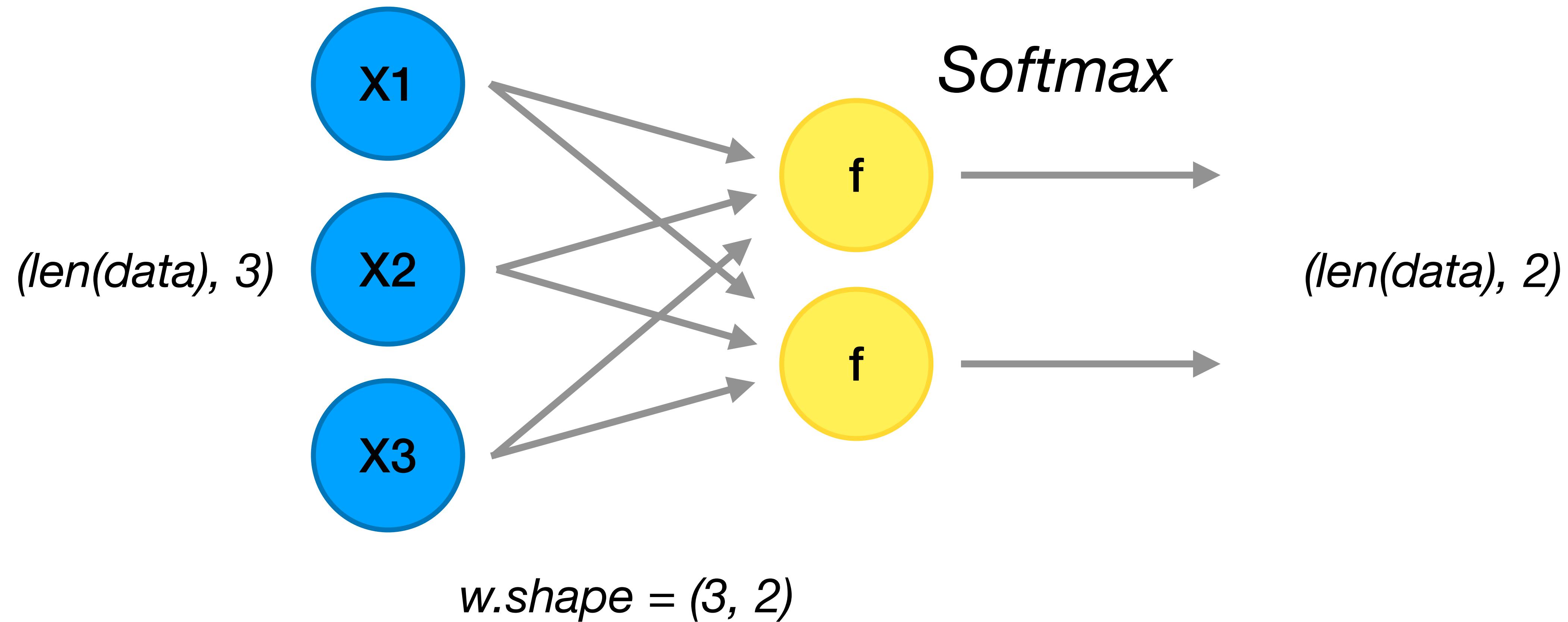
Логистическая регрессия

Inference



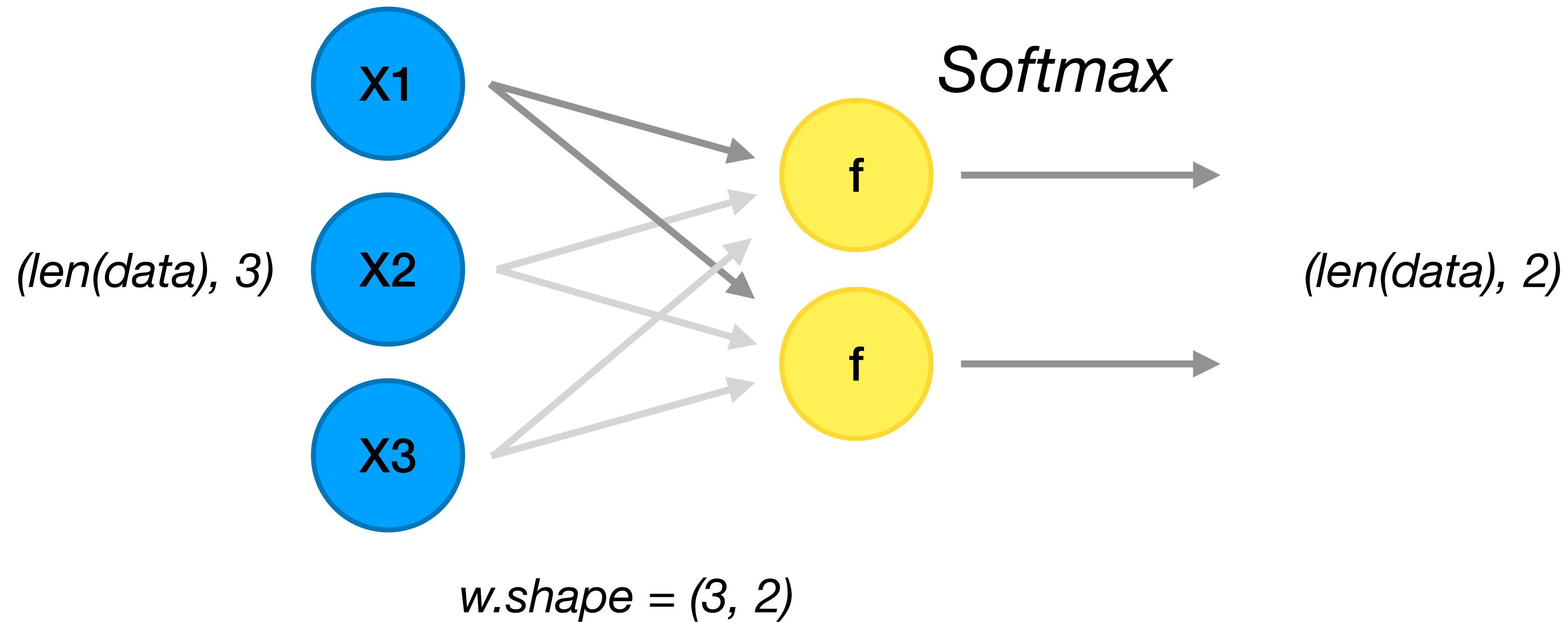
Логистическая регрессия

Inference



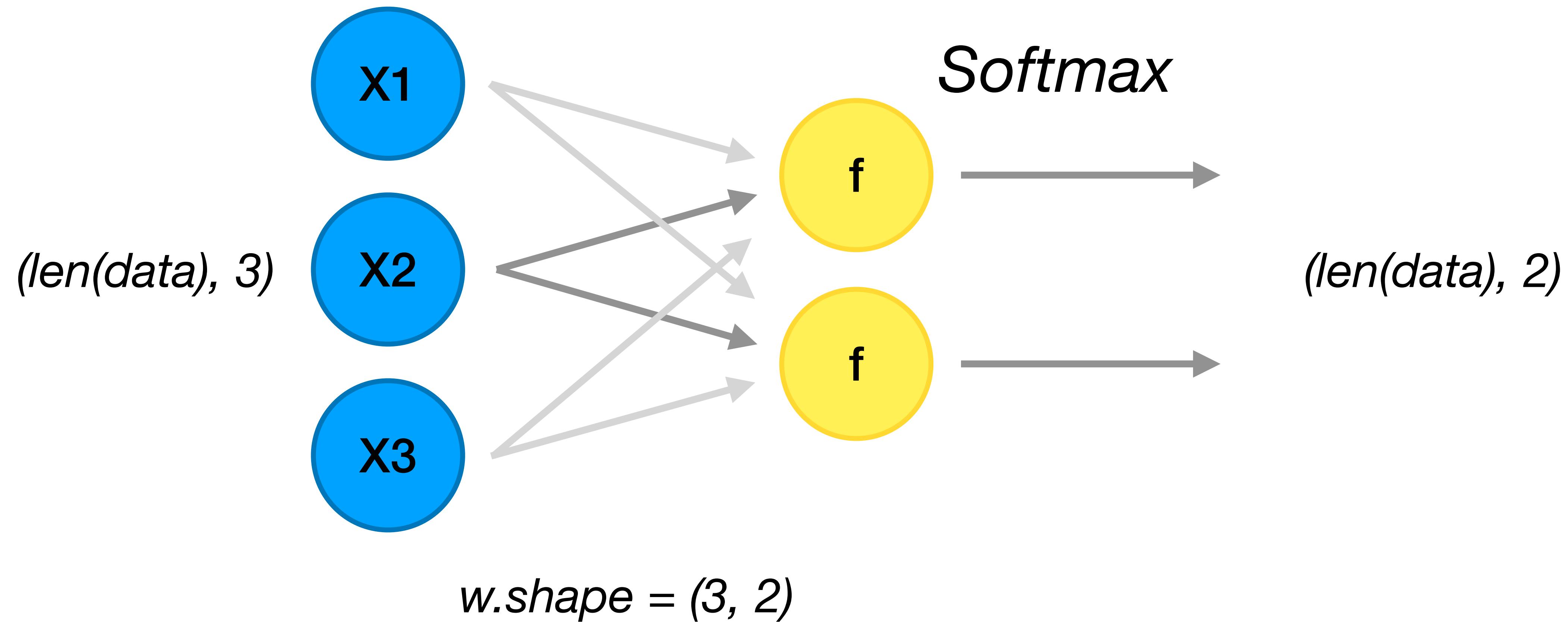
Логистическая регрессия

Inference



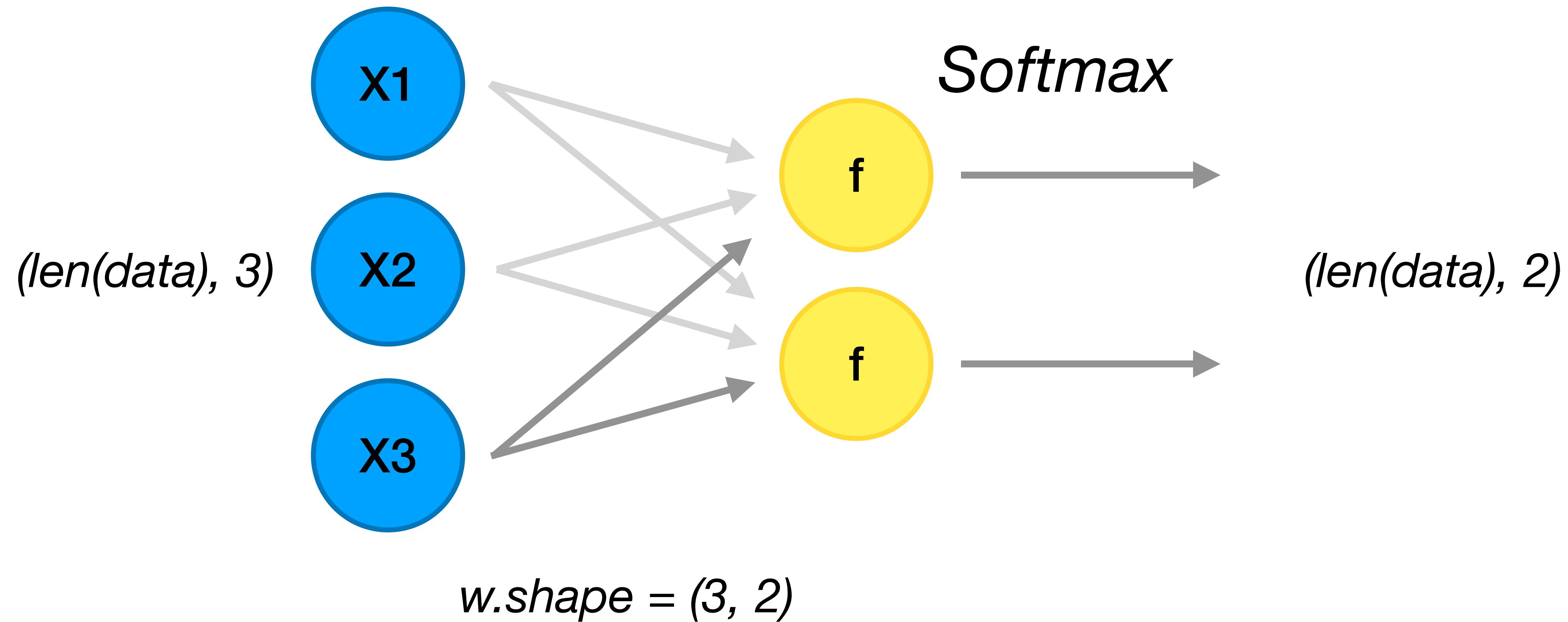
Логистическая регрессия

Inference



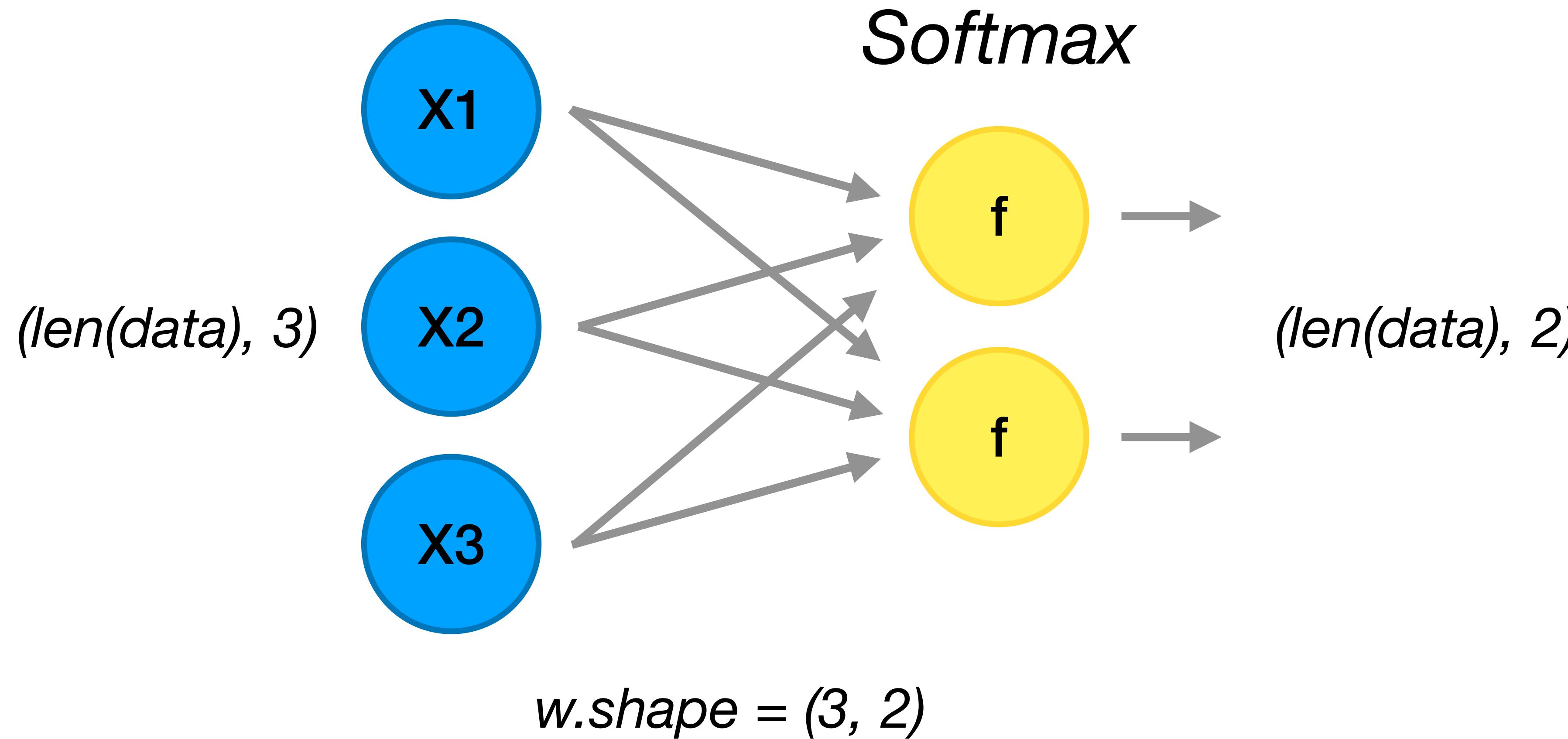
Логистическая регрессия

Inference



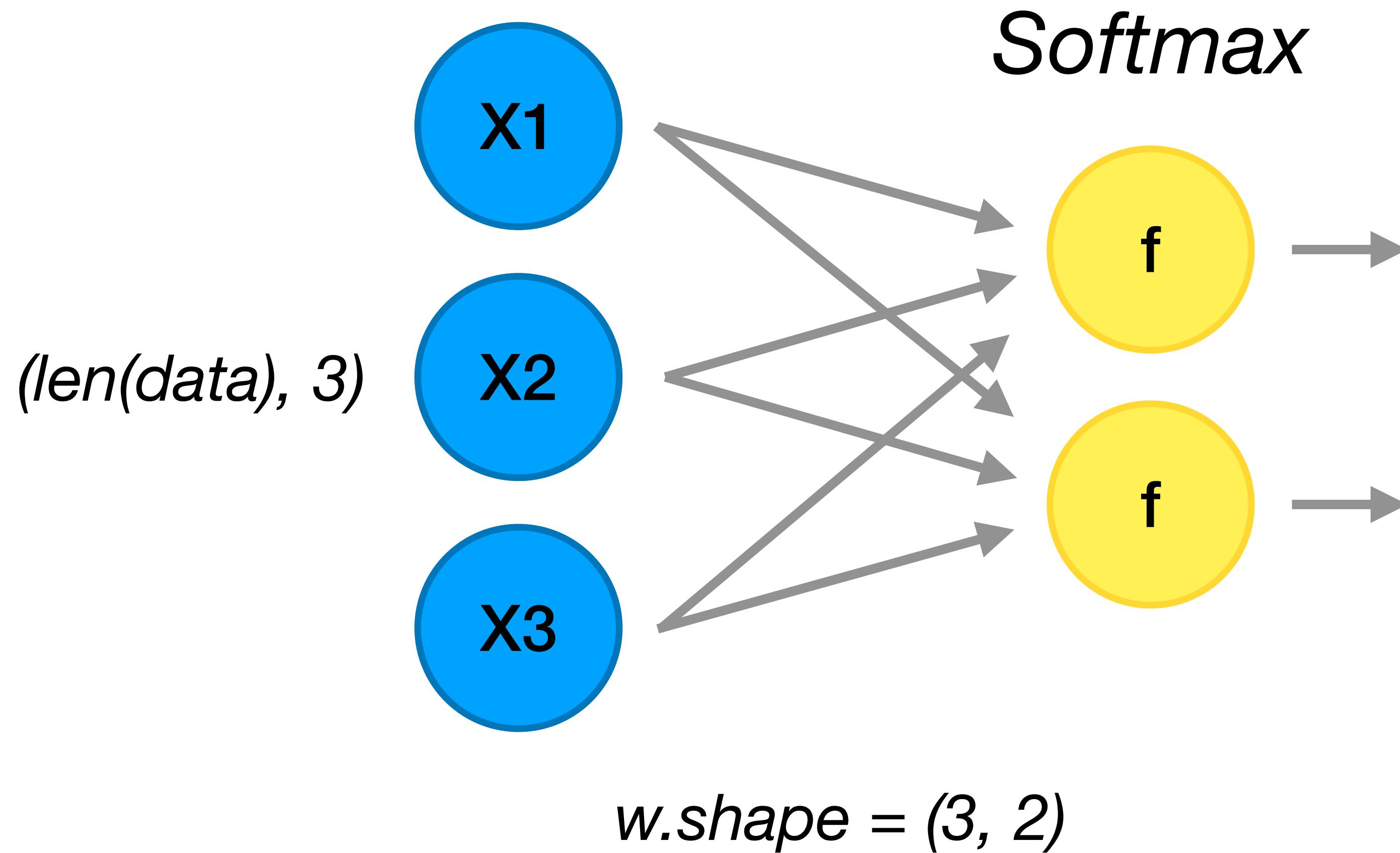
Логистическая регрессия

Train



Логистическая регрессия

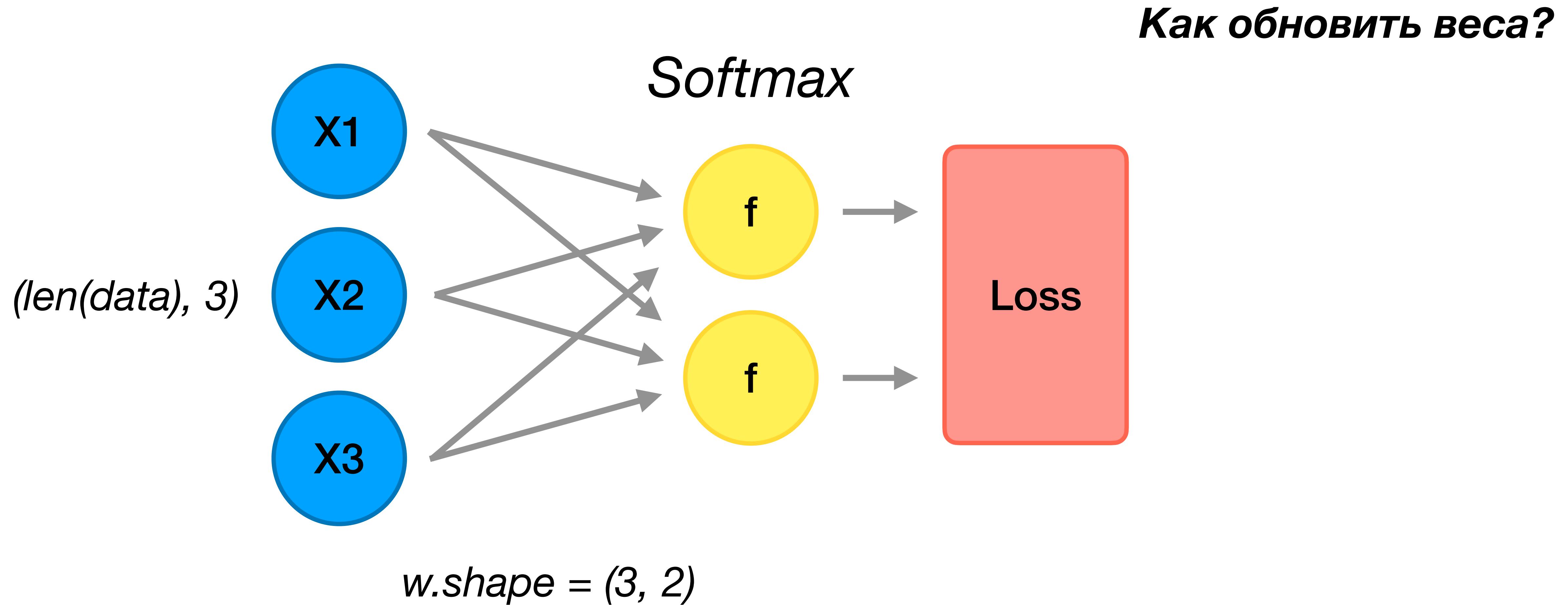
Train



Как обновить веса?

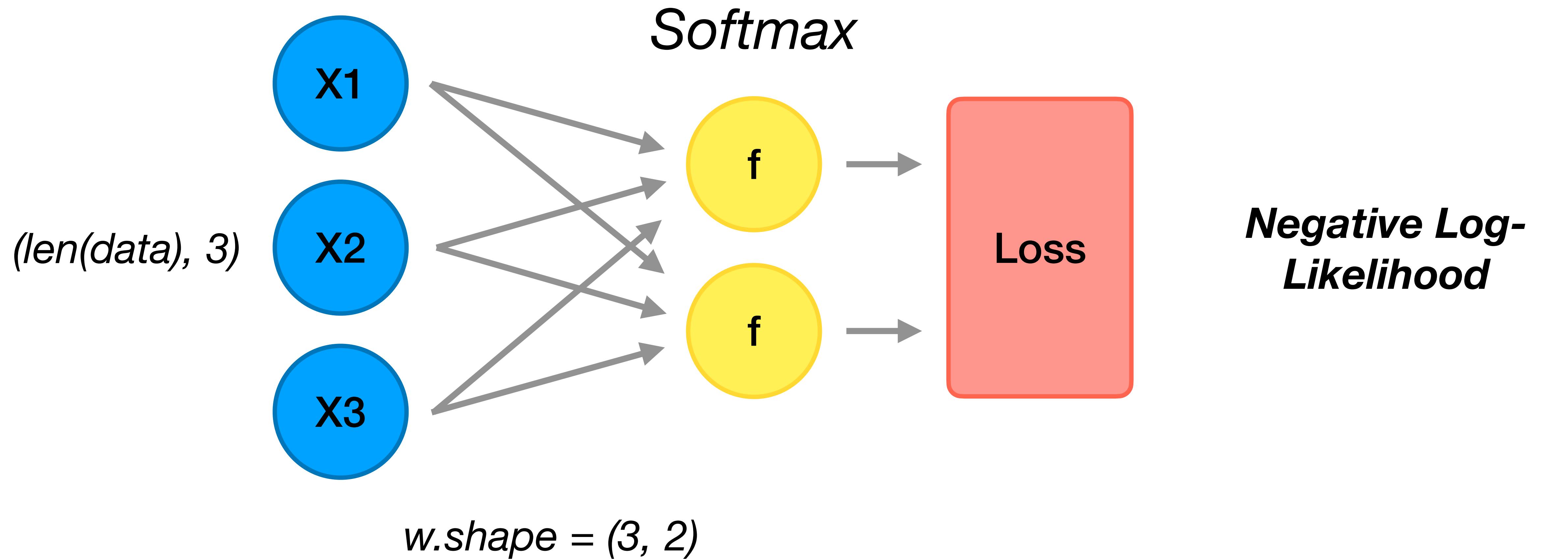
Логистическая регрессия

Train



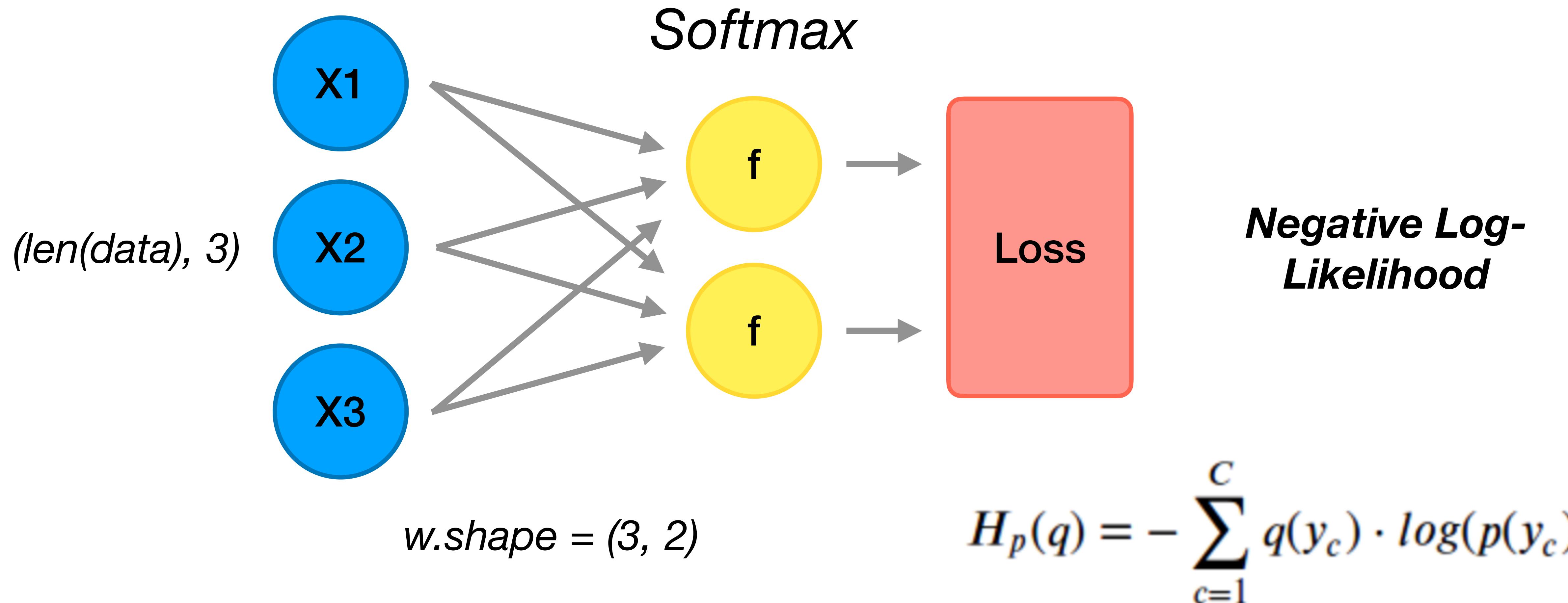
Логистическая регрессия

Train



Логистическая регрессия

Train



Логистическая регрессия

Negative Log-Likelihood



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1

Логистическая регрессия

Train

Логистическая регрессия

Train

$$L(y) = \prod_a^b P(\hat{y}|w, x)$$

Логистическая регрессия

Train

$$L(y) = \prod_a^b P(\hat{y}|w, x)$$

$$L(y) = \prod_a^b P(\hat{y}|w, x) = \sum_a^b logP(\hat{y}|w, x)$$

Логистическая регрессия

Train

$$L(y) = \prod_a^b P(\hat{y}|w, x)$$

$$L(y) = \prod_a^b P(\hat{y}|w, x) = \sum_a^b logP(\hat{y}|w, x)$$

$$H_p(q) = - \sum_{c=1}^C q(y_c) \cdot \log(p(y_c))$$

Логистическая регрессия

Train

$$L(y) = \prod_a^b P(\hat{y}|w, x) = \sum_a^b logP(\hat{y}|w, x)$$

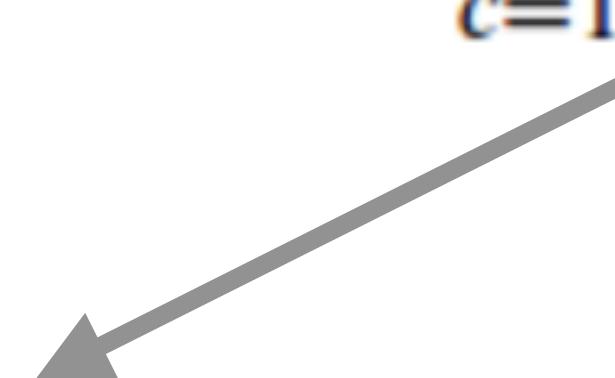
$$H_p(q) = - \sum_{c=1}^C q(y_c) \cdot \log(p(y_c))$$

Логистическая регрессия

Train

$$L(y) = \prod_a^b P(\hat{y}|w, x) = \sum_a^b logP(\hat{y}|w, x)$$

$$H_p(q) = - \sum_{c=1}^C q(y_c) \cdot \log(p(y_c))$$



Target

Логистическая регрессия

Train

$$L(y) = \prod_a^b P(\hat{y}|w, x) = \sum_a^b logP(\hat{y}|w, x)$$

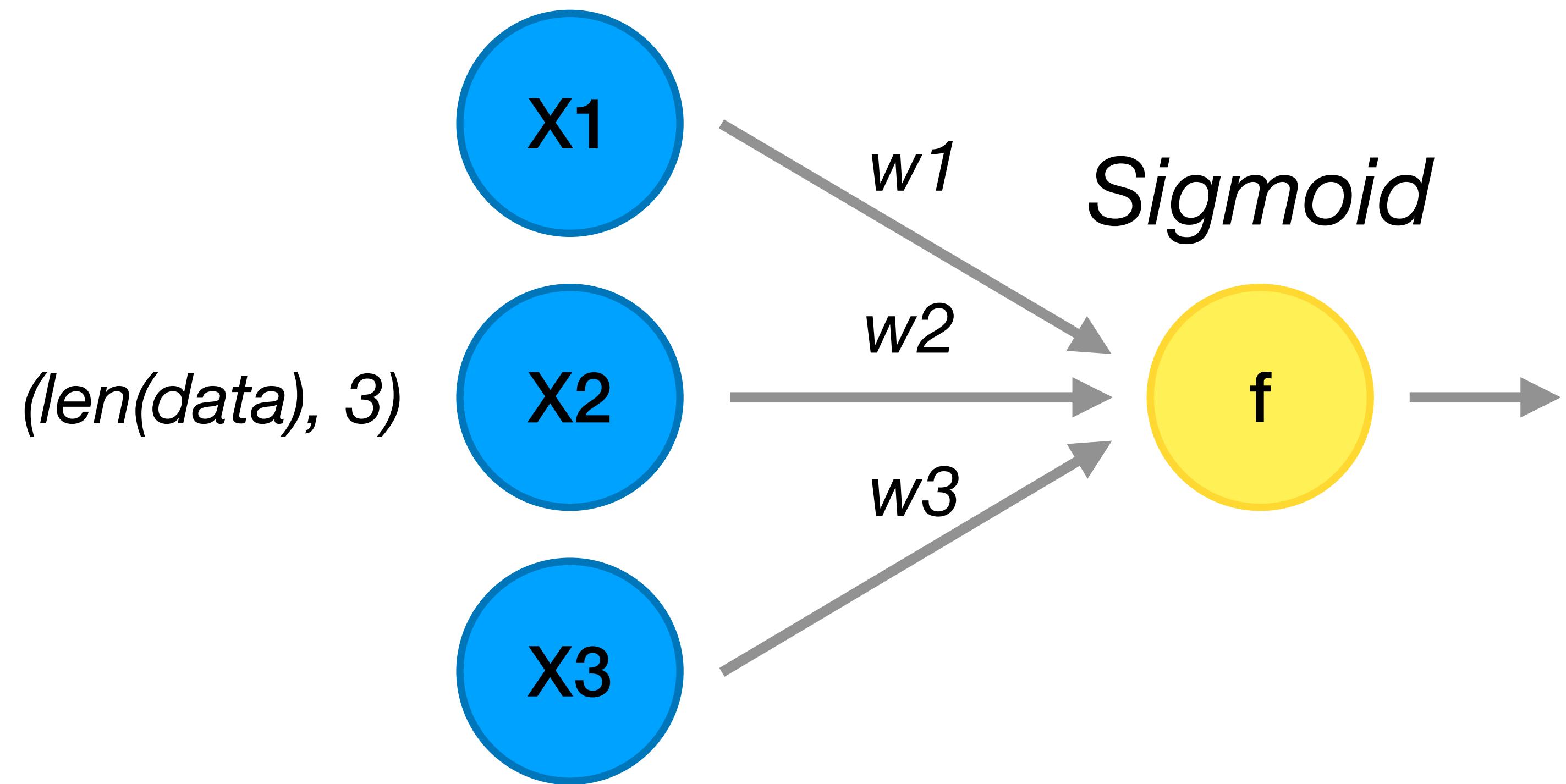
$$H_p(q) = - \sum_{c=1}^C q(y_c) \cdot \log(p(y_c))$$

Target

Prediction

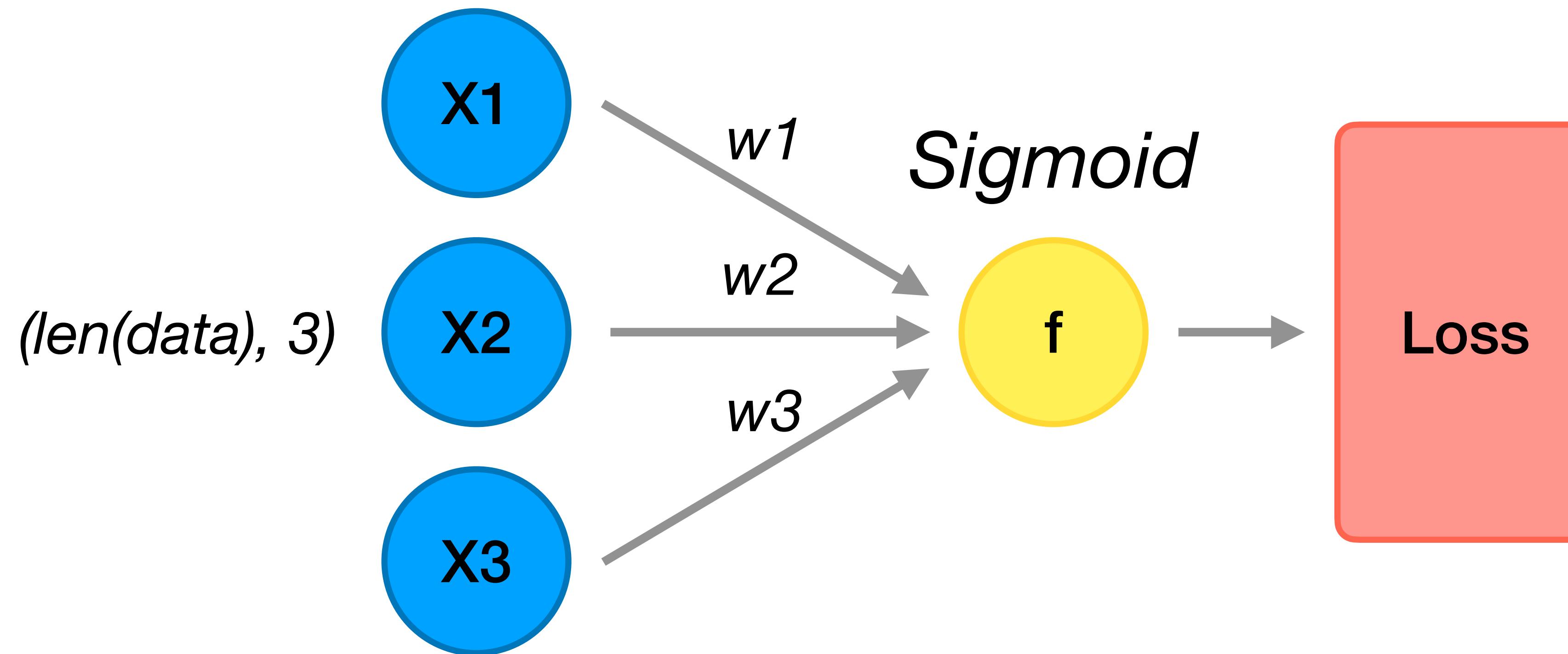
Логистическая регрессия

Train



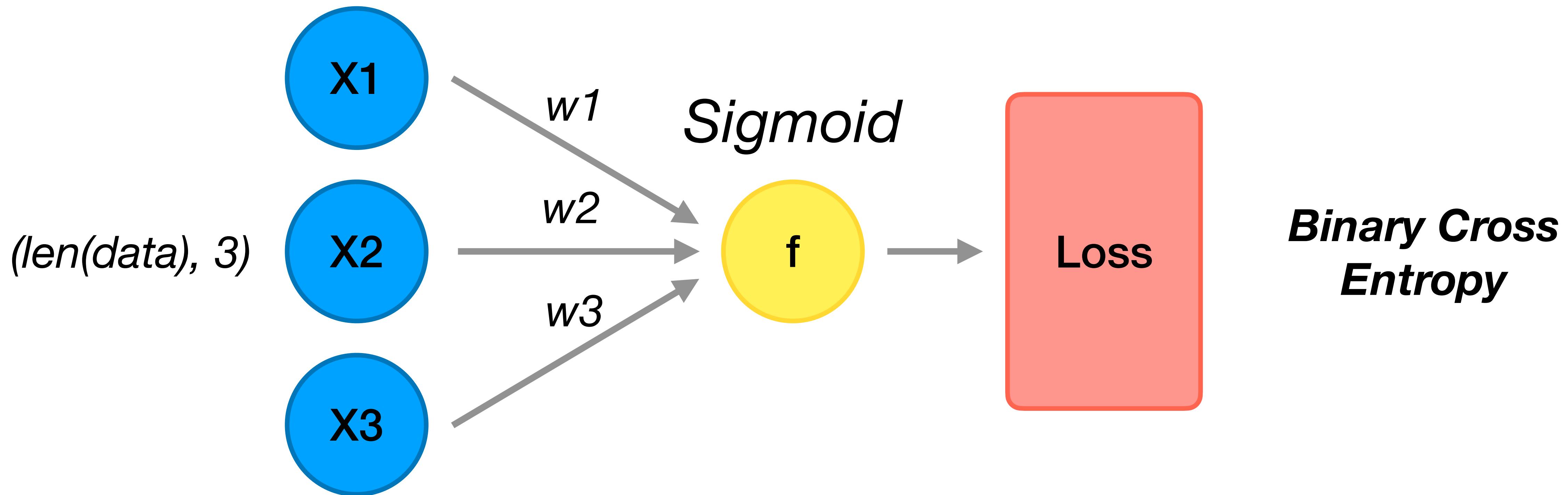
Логистическая регрессия

Train



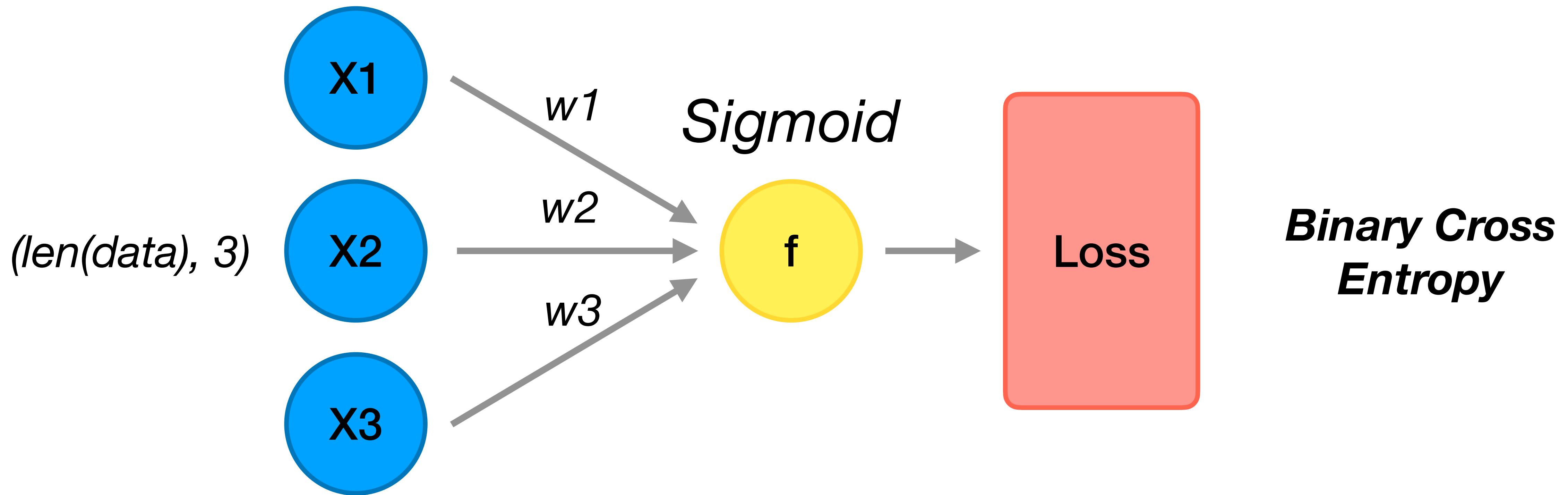
Логистическая регрессия

Train



Логистическая регрессия

Train



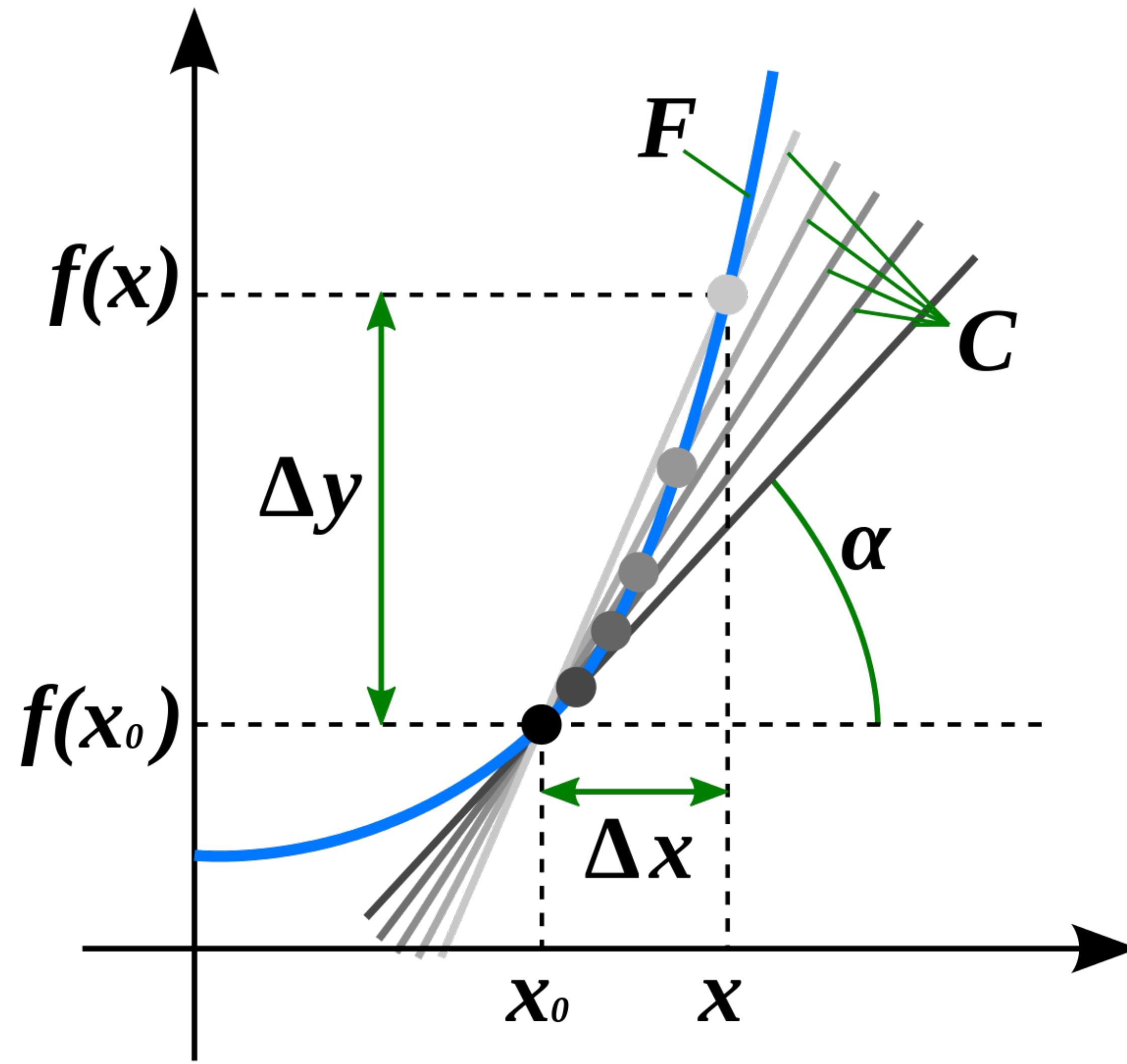
$$L(y, \hat{y}) = - (y * \log(\hat{y}_i) + (1 - y) * \log(1 - \hat{y}_i))$$

Gradient Descent



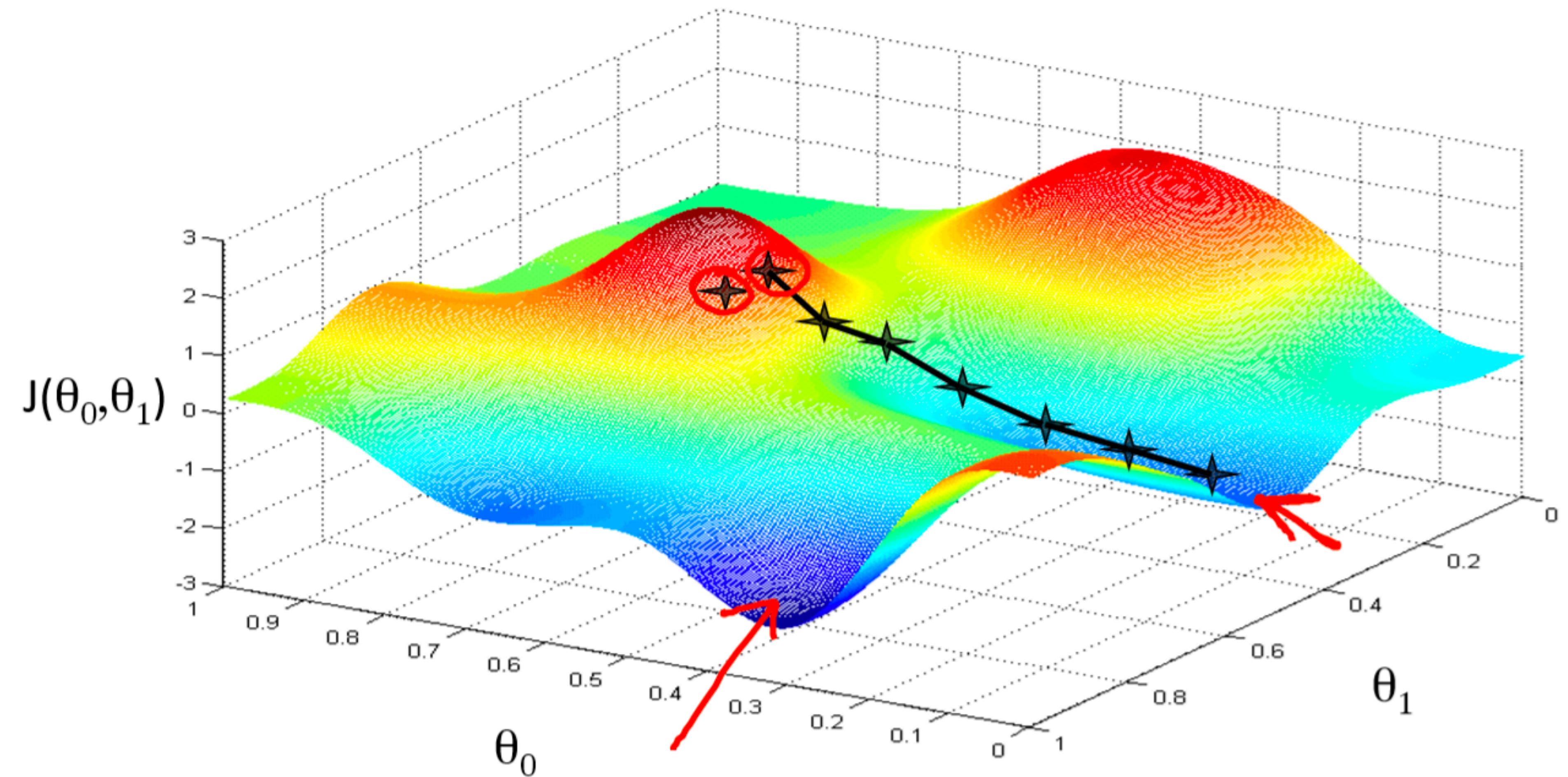
Gradient Descent

Производные



Gradient Descent

Производные



Gradient Descent

Производные

```
w = w - learning_rate * dLdW  
b = b - learning_rate * dLdb
```

Gradient Descent

Gradient Descent

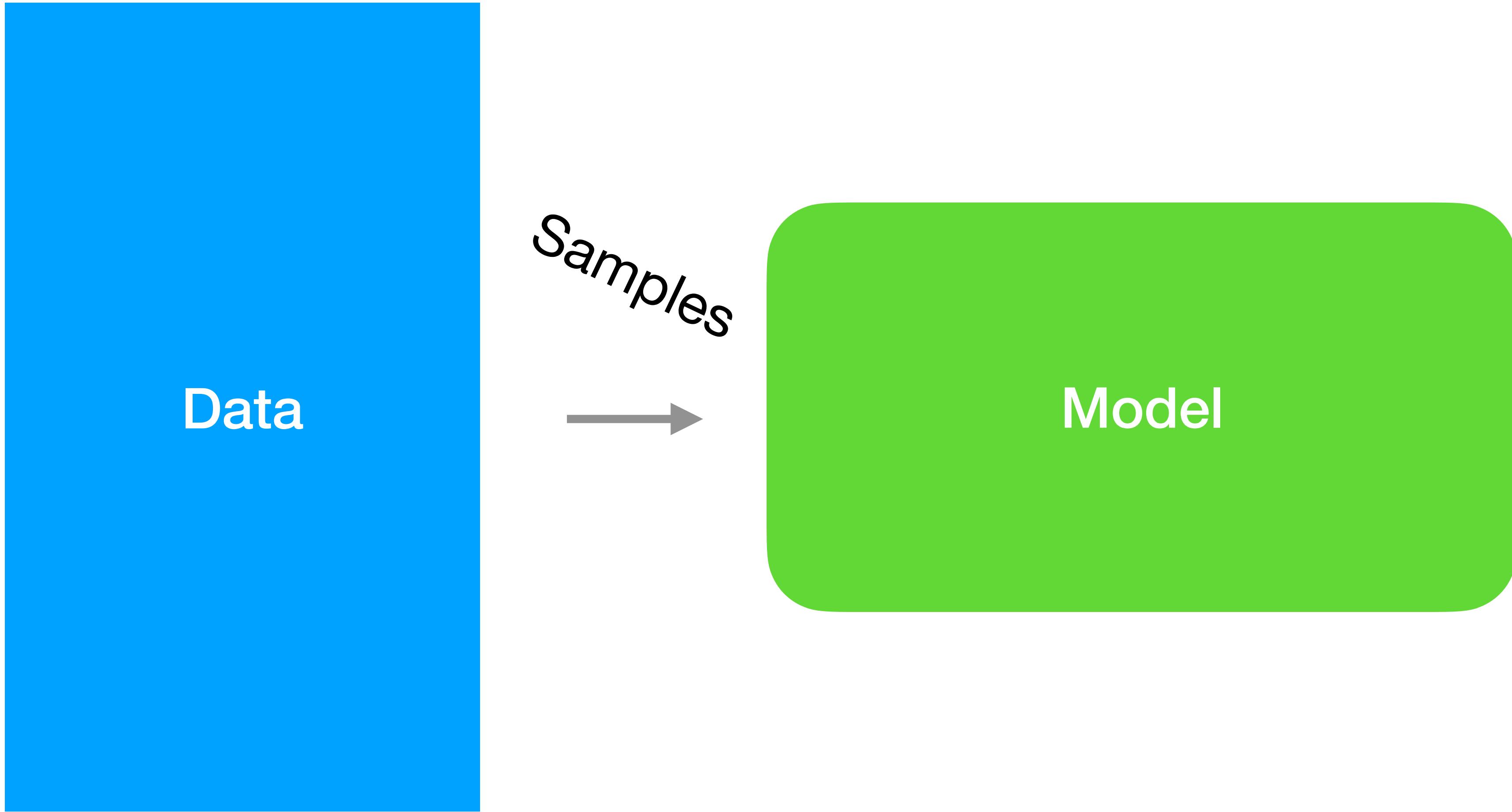
Features



Data

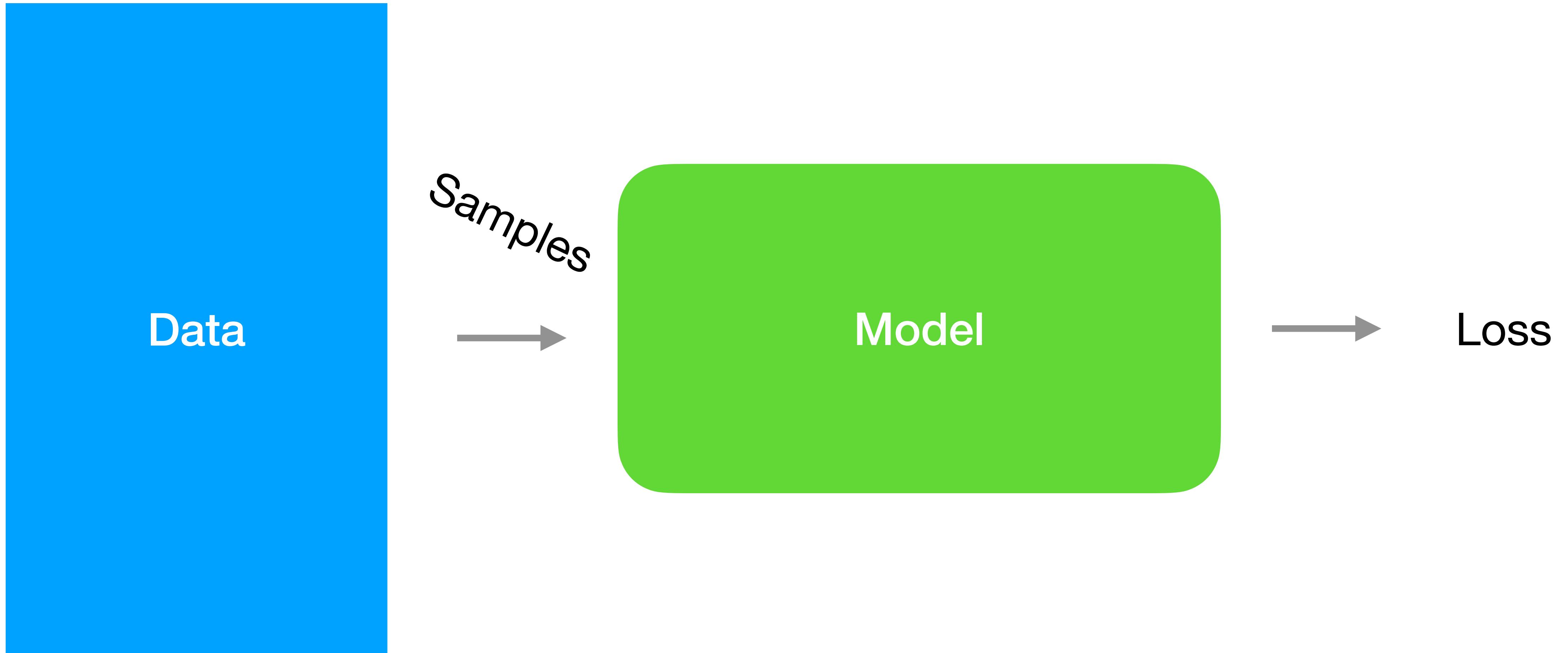
Gradient Descent

Features



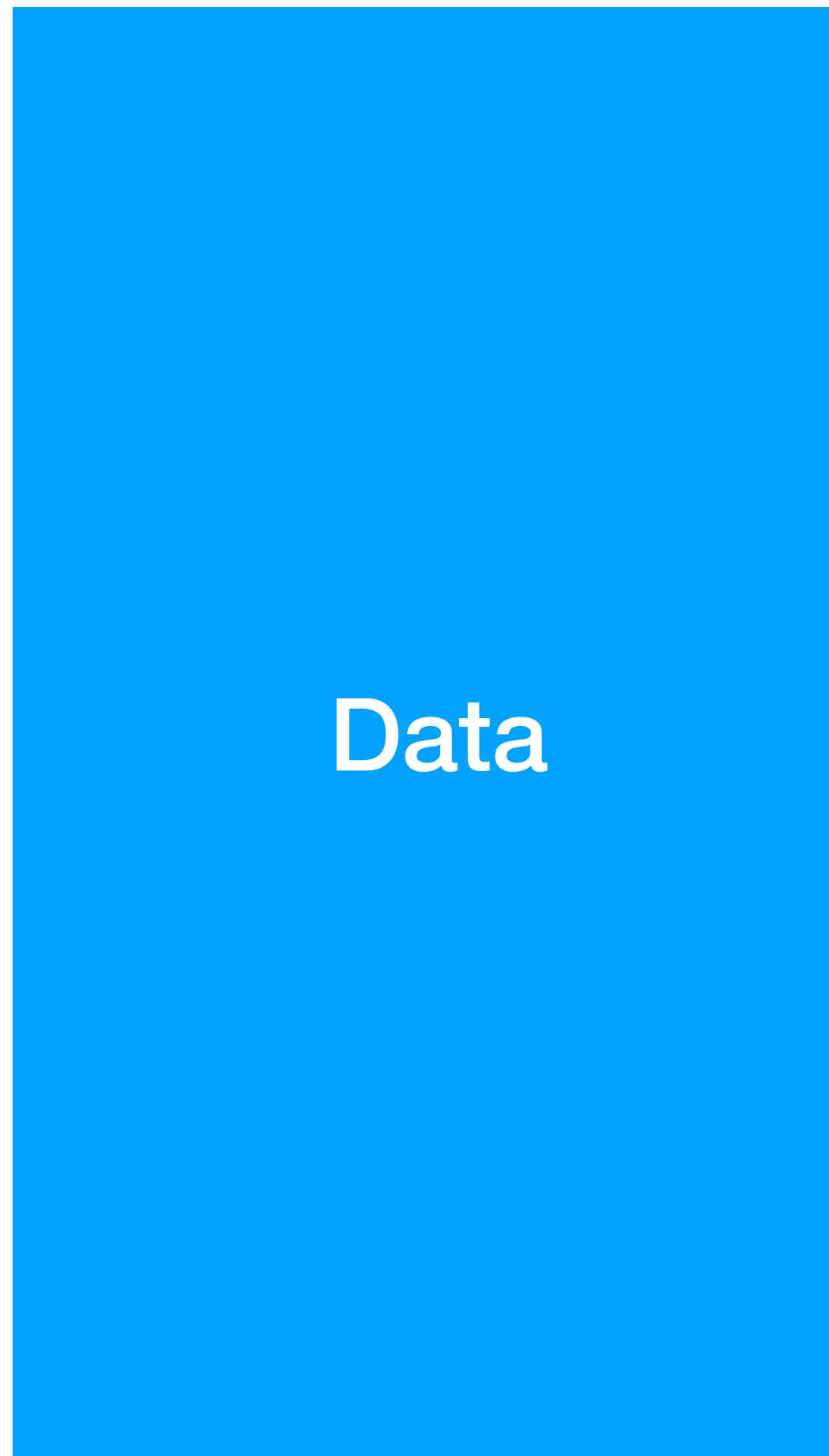
Gradient Descent

Features



Gradient Descent

Features



Samples



Data

Model

Loss

Update weights



Stochastic Gradient Descent

Features



Batch

Data

Stochastic Gradient Descent

Features



Batch

e.g. 32

Data

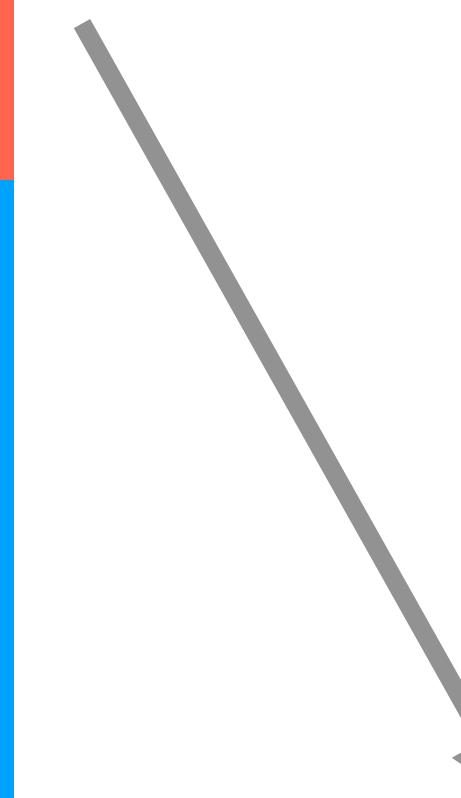
Stochastic Gradient Descent

Features

Batch

Data

Model

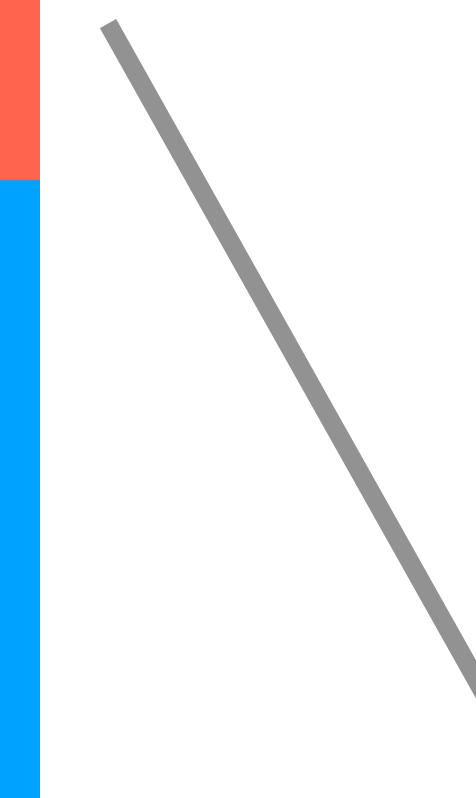


Stochastic Gradient Descent

Features

Batch

Data



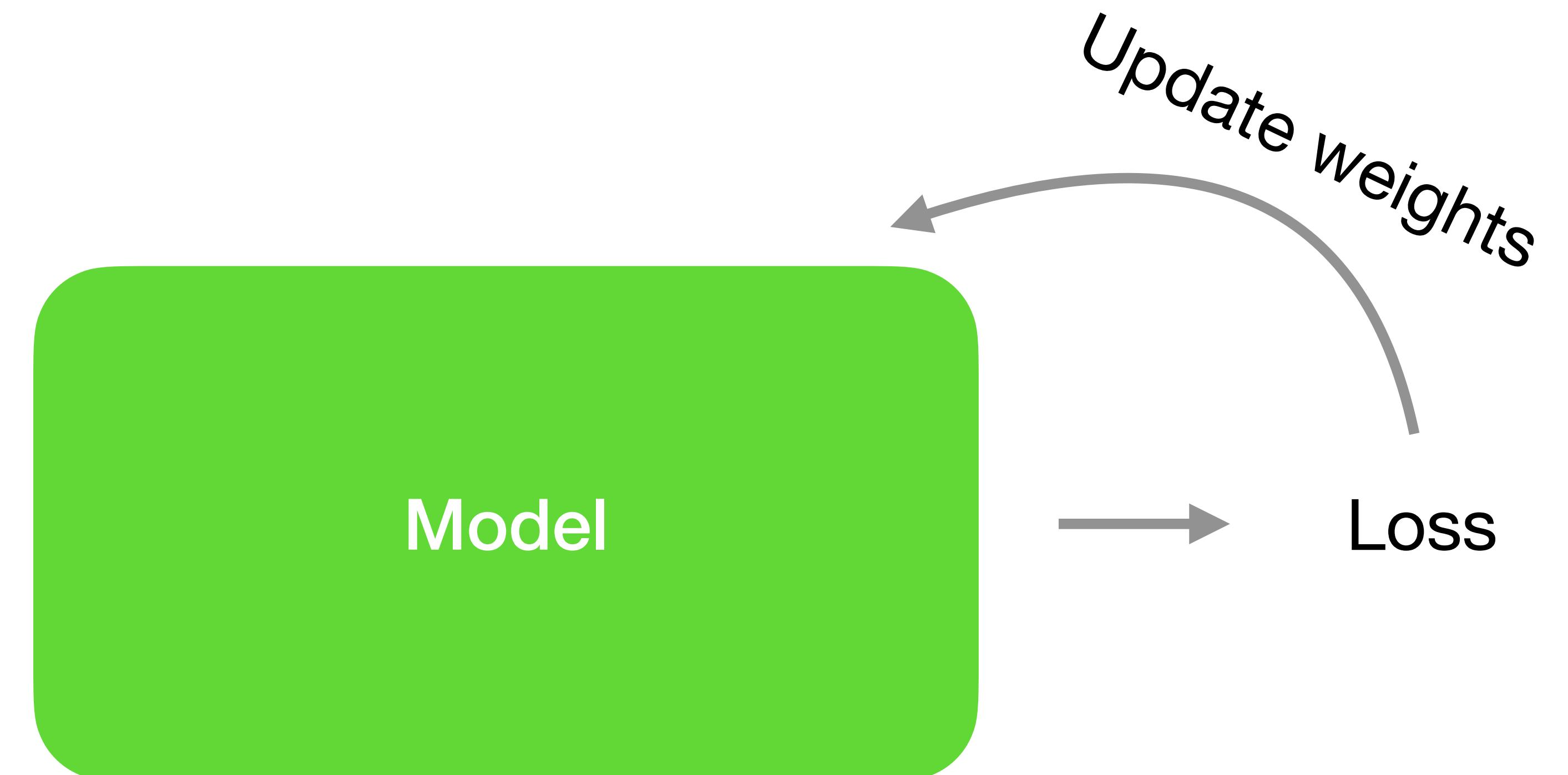
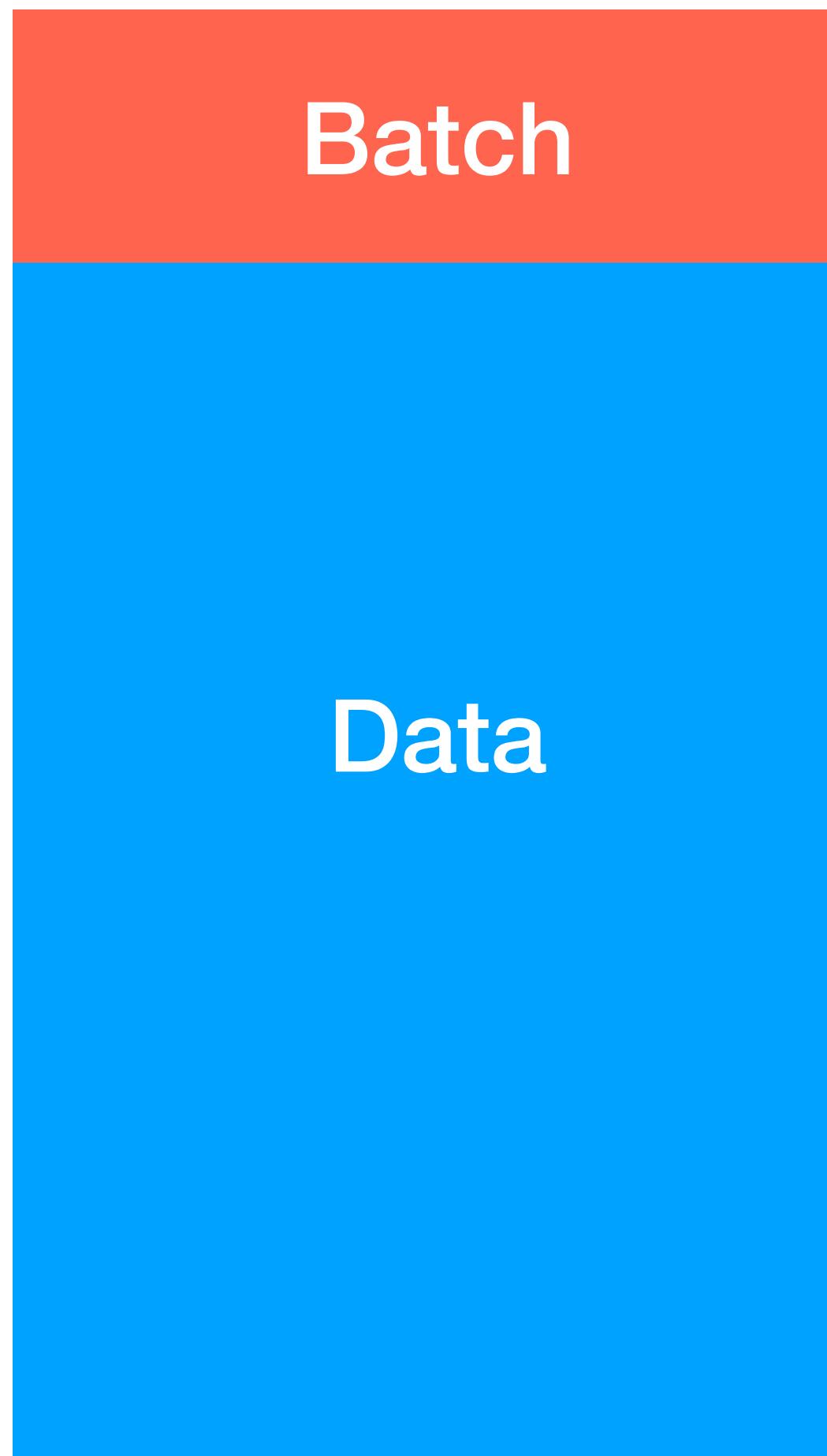
Model



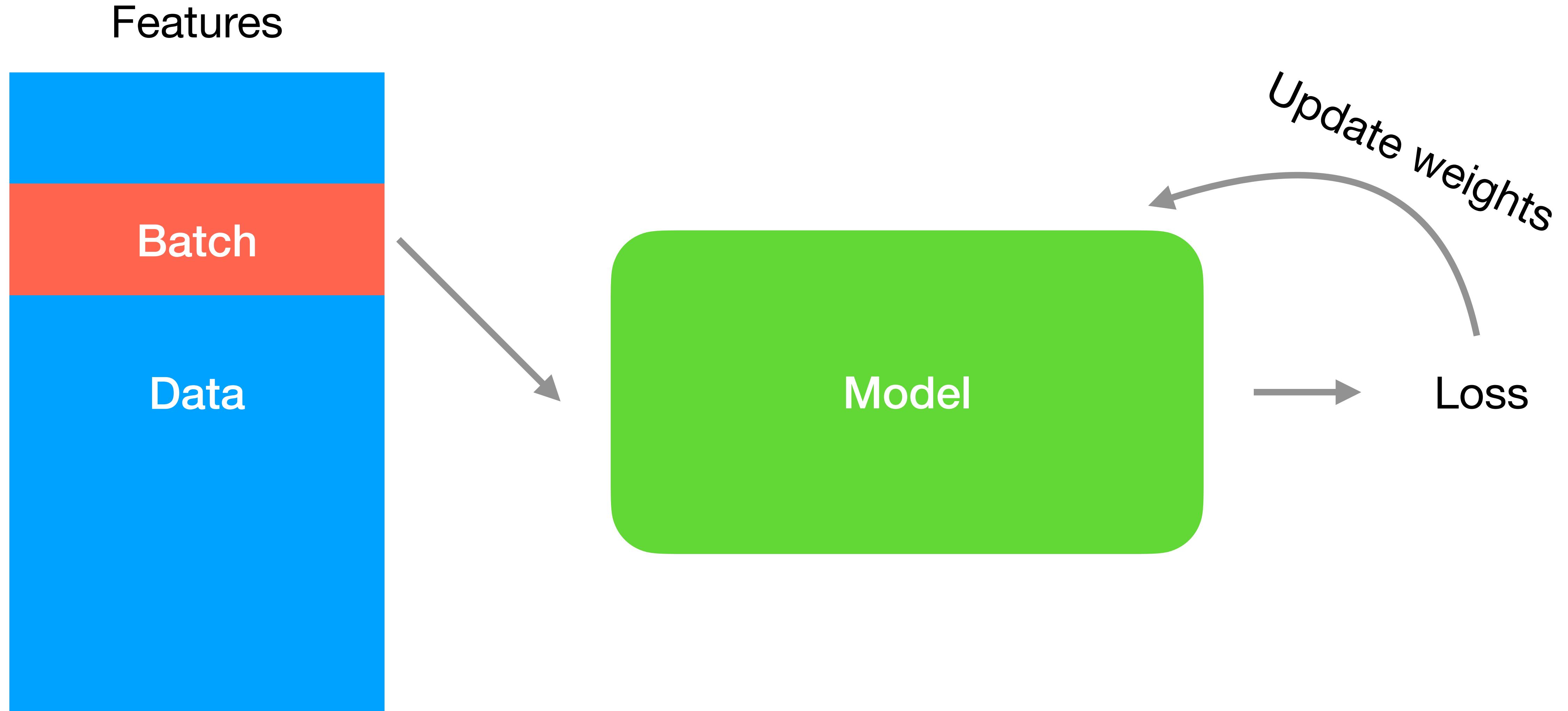
Loss

Stochastic Gradient Descent

Features

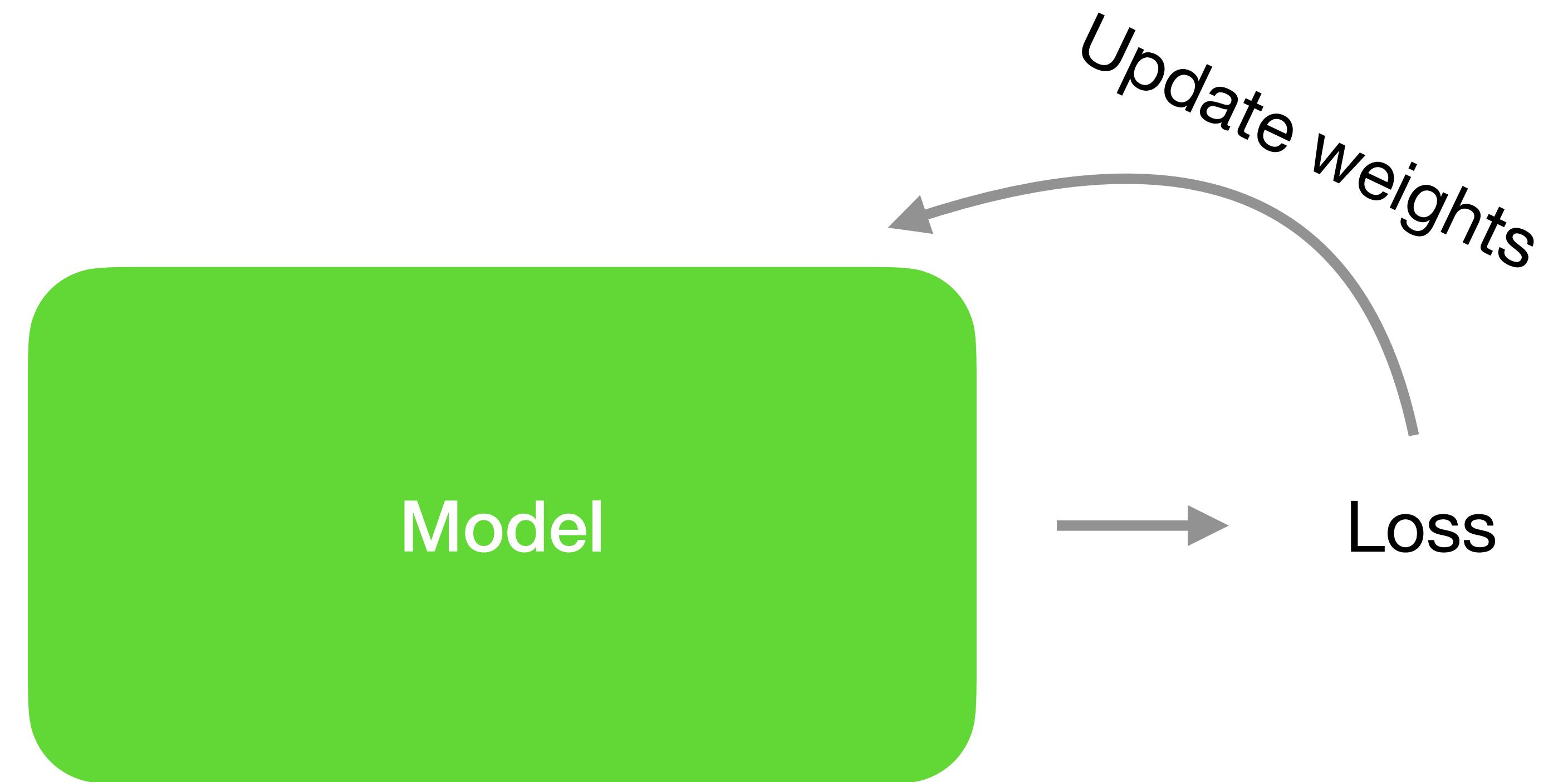
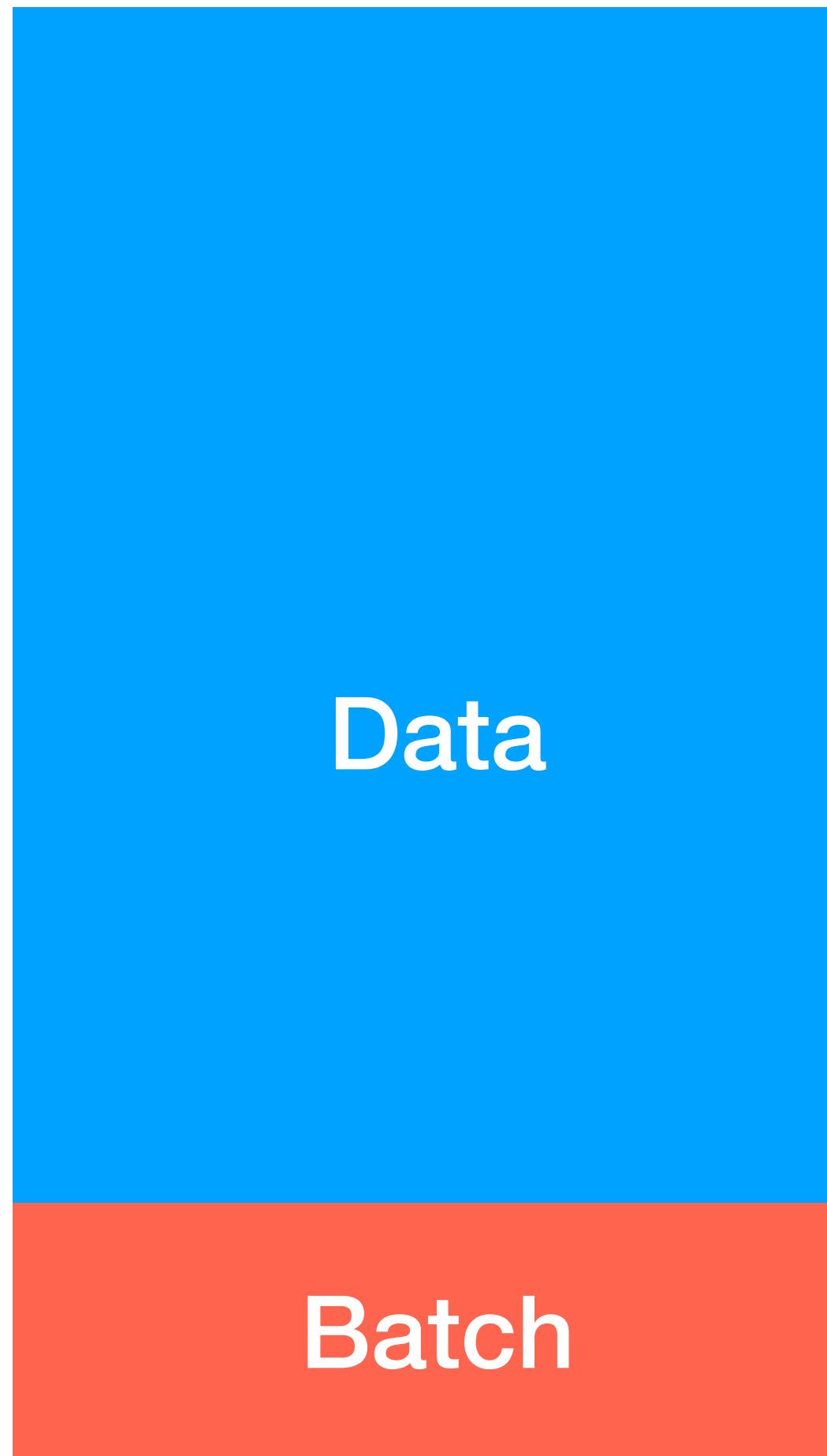


Stochastic Gradient Descent



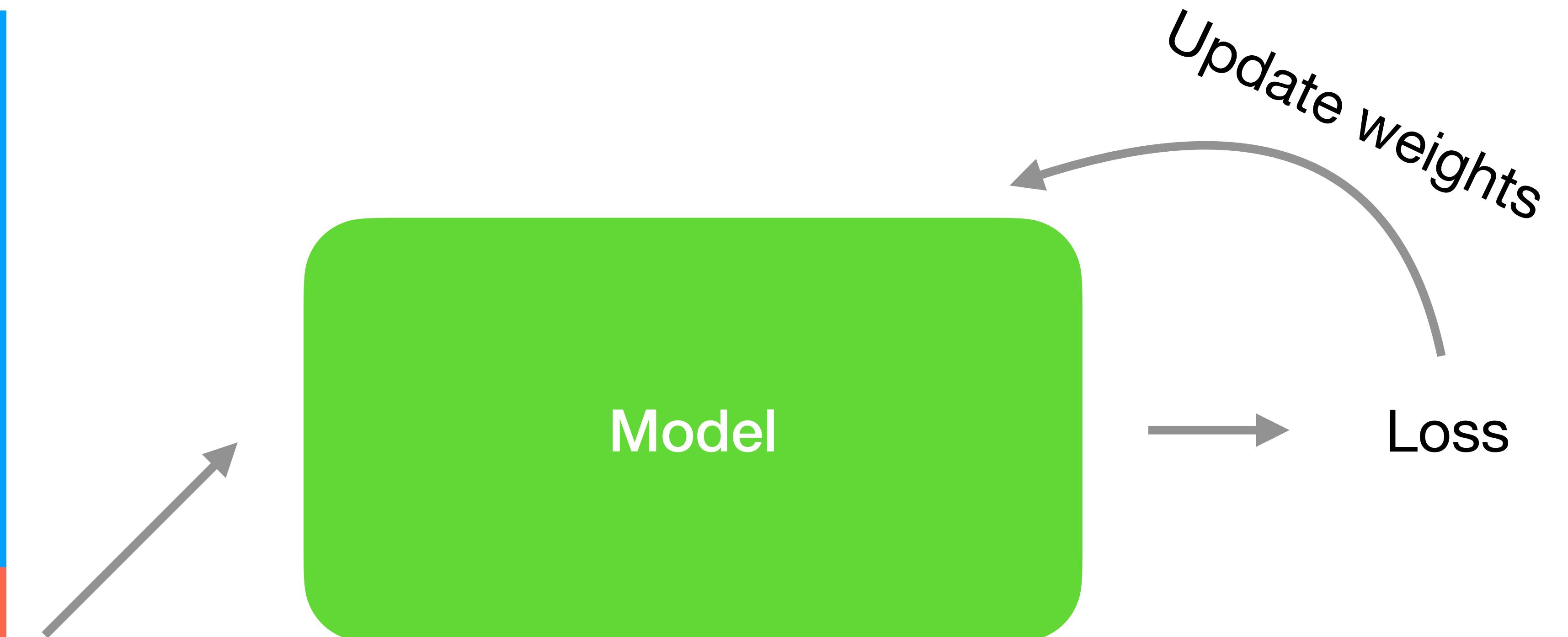
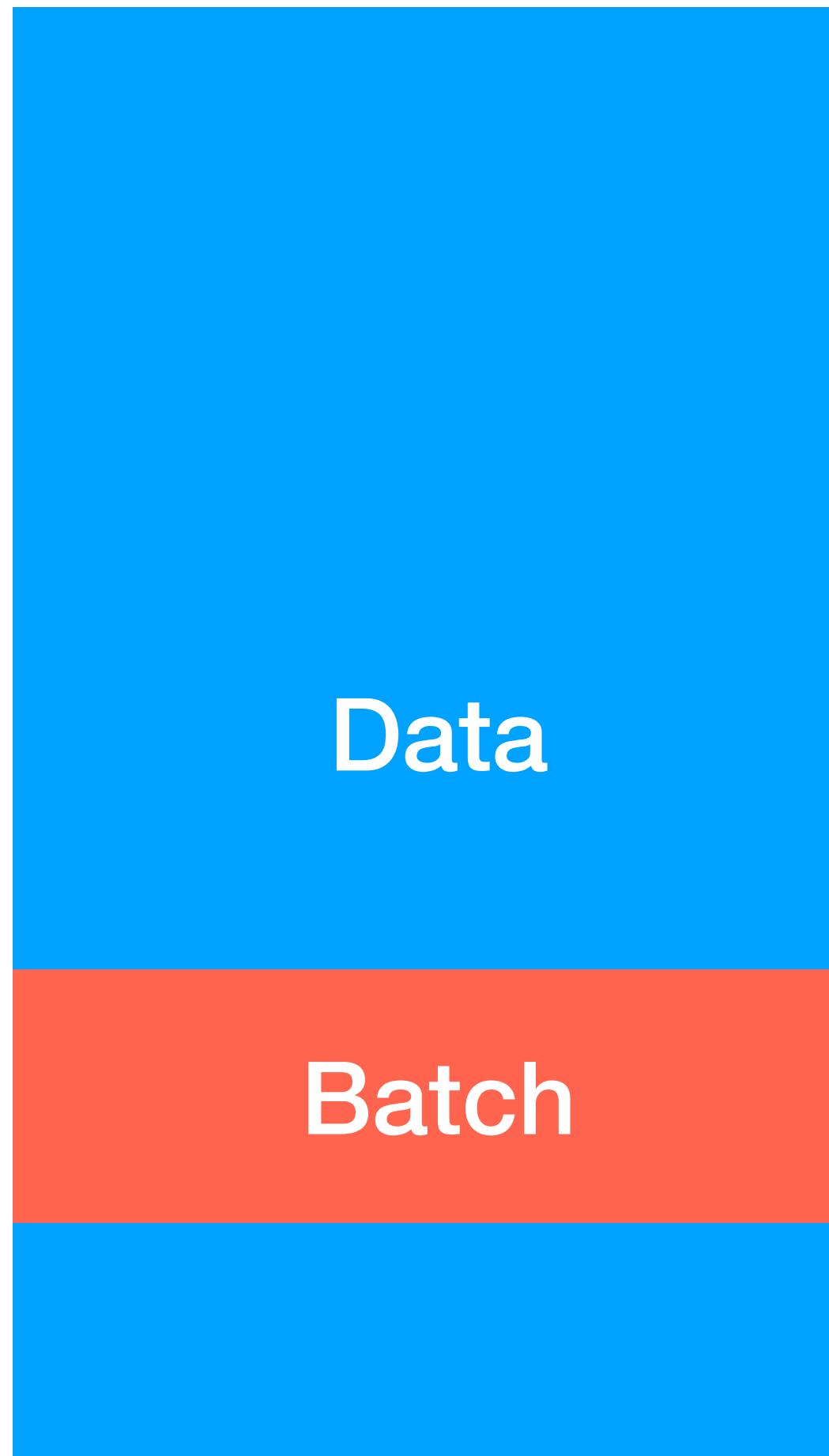
Stochastic Gradient Descent

Features



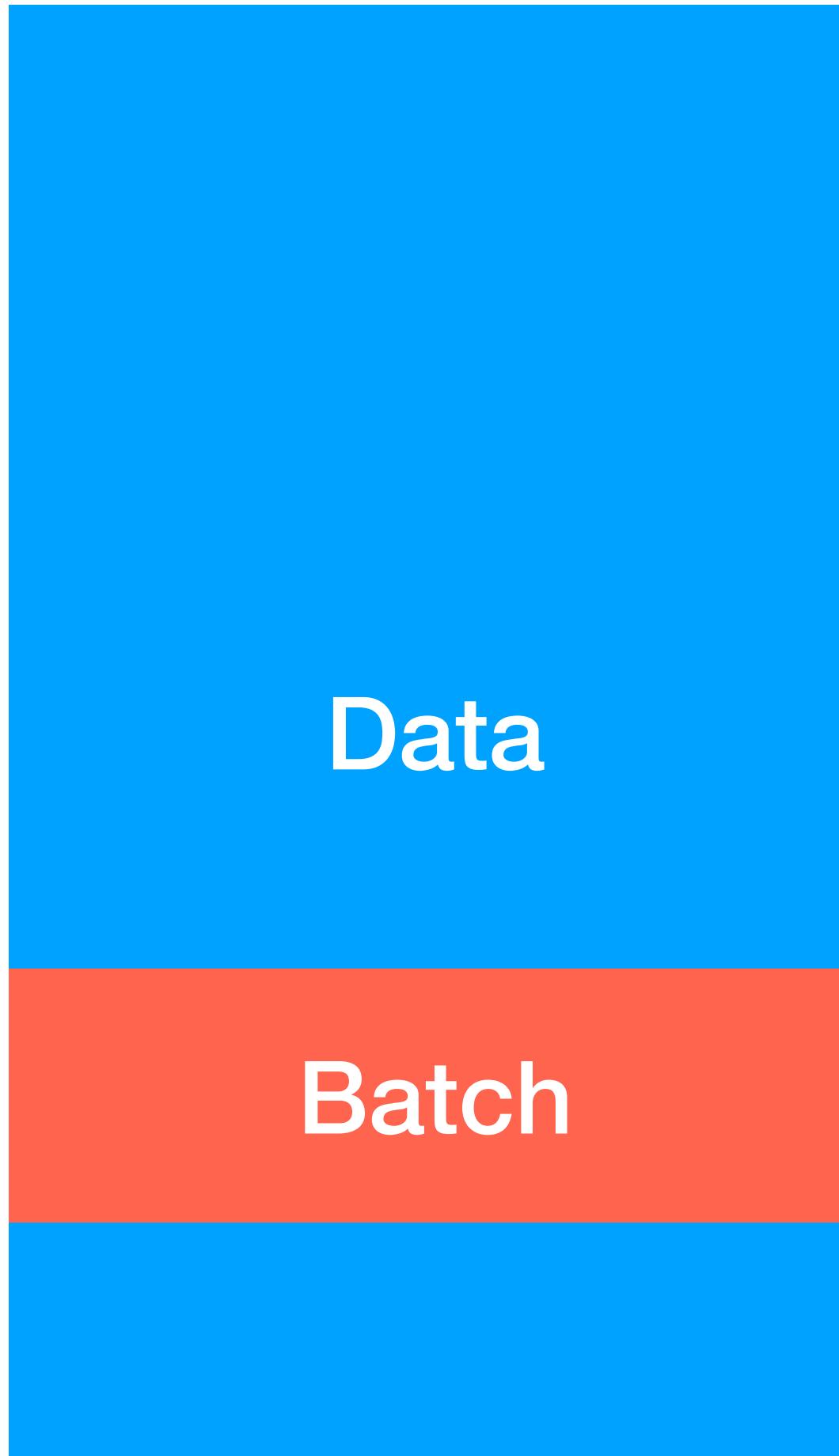
Stochastic Gradient Descent

Features



Stochastic Gradient Descent

Features

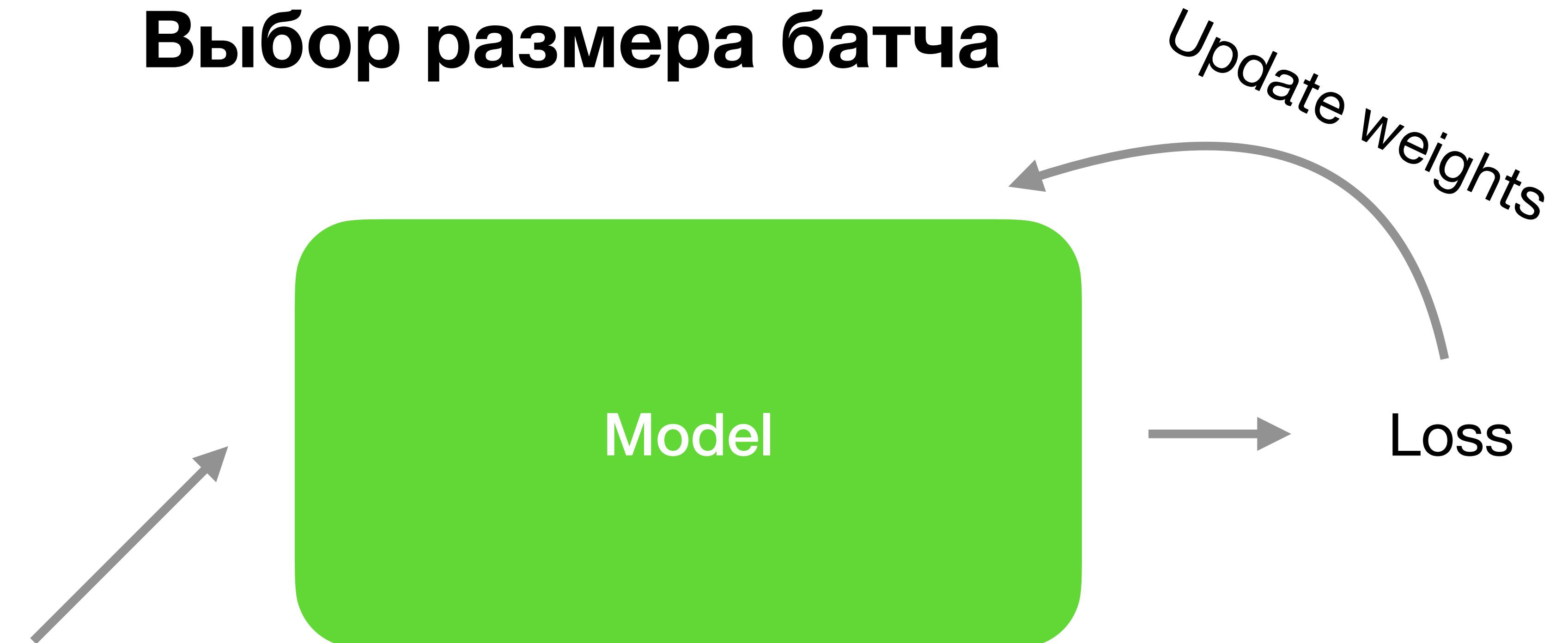


Выбор размера батча

Model

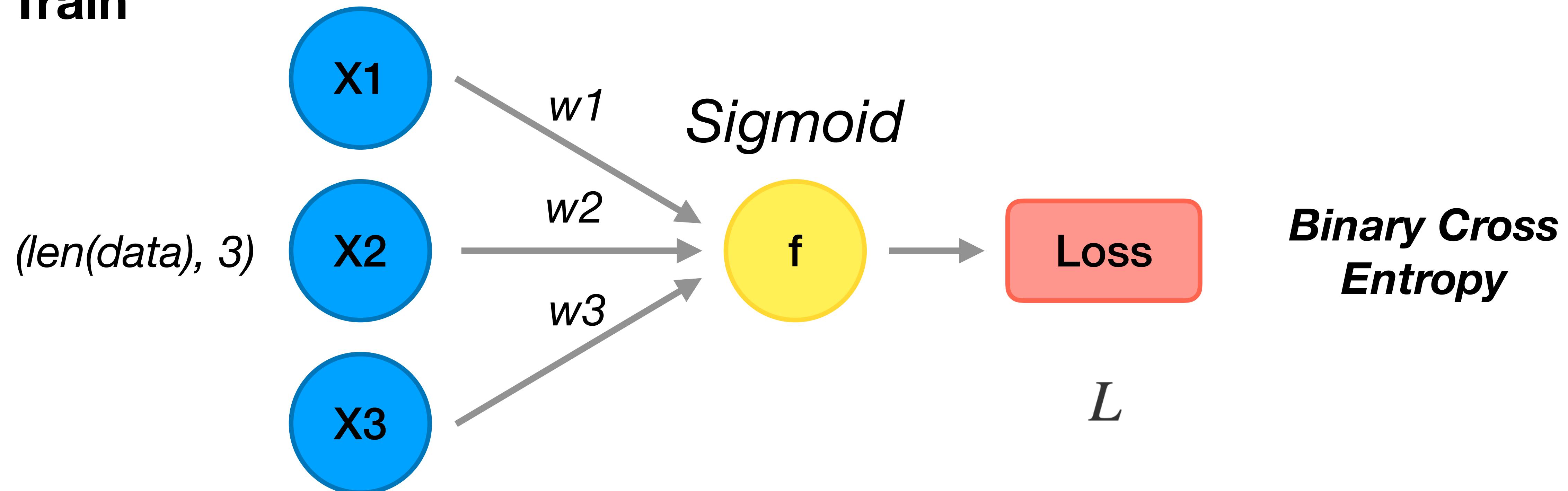
Loss

Update weights



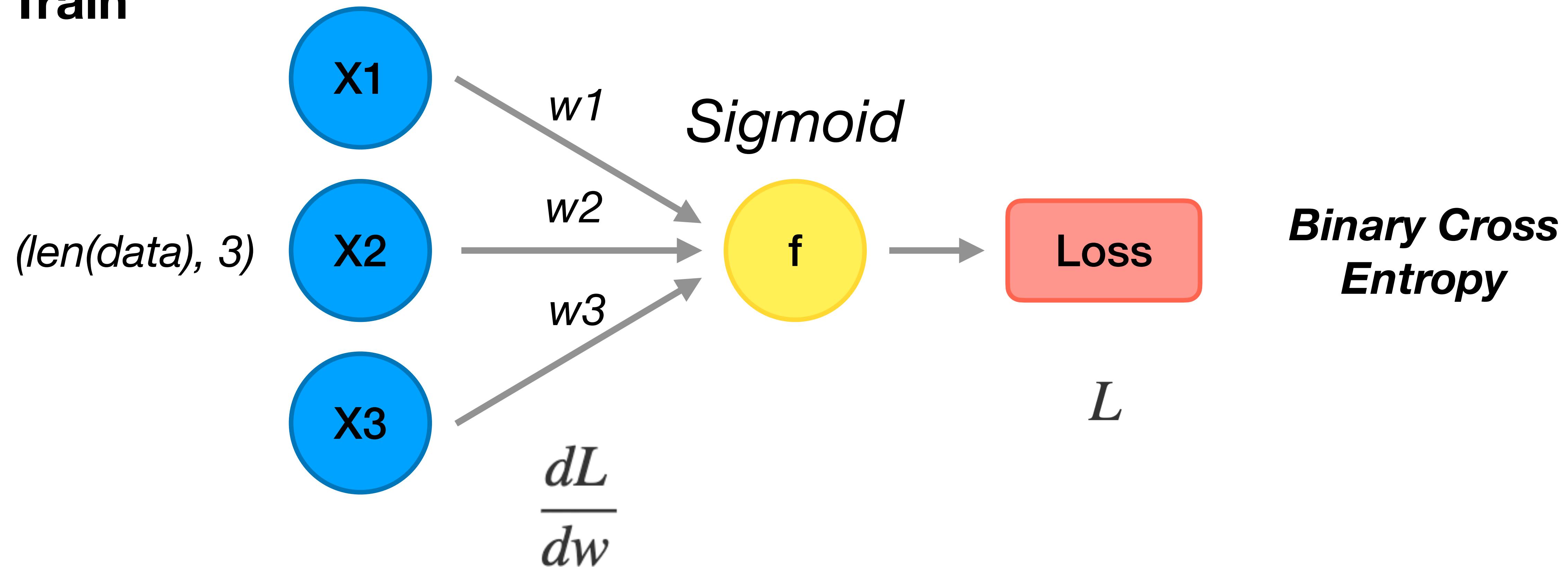
Логистическая регрессия

Train



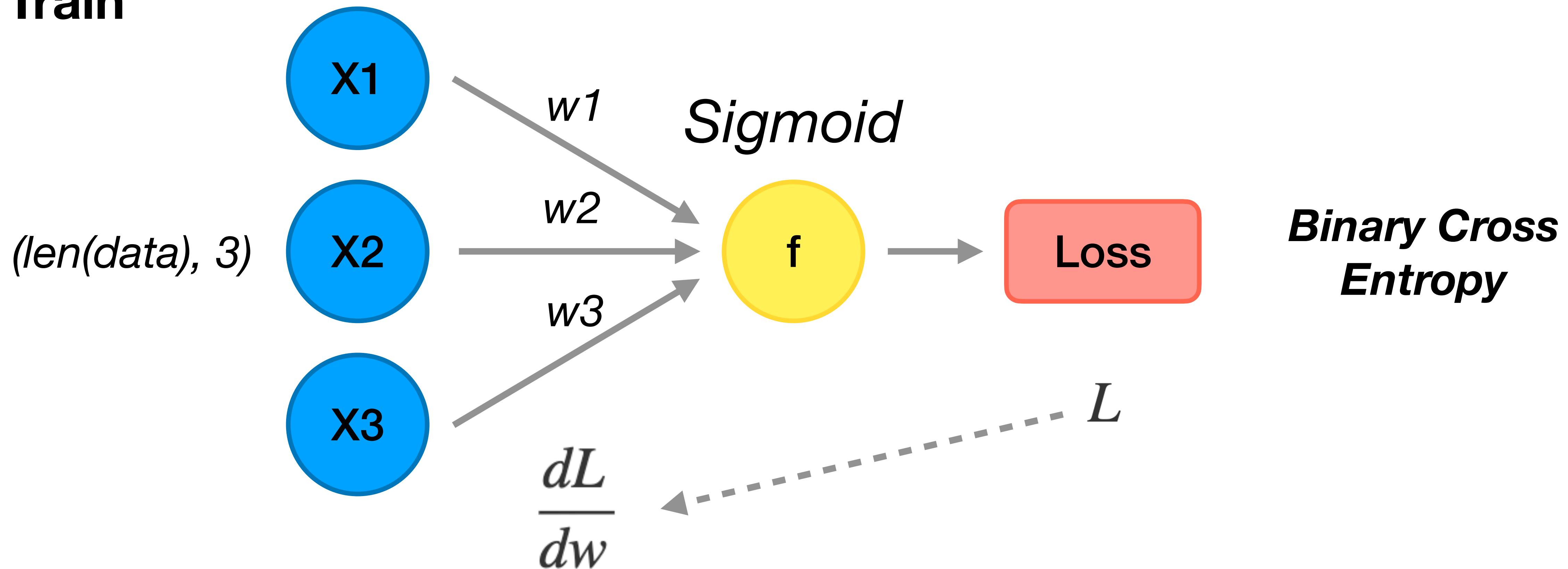
Логистическая регрессия

Train



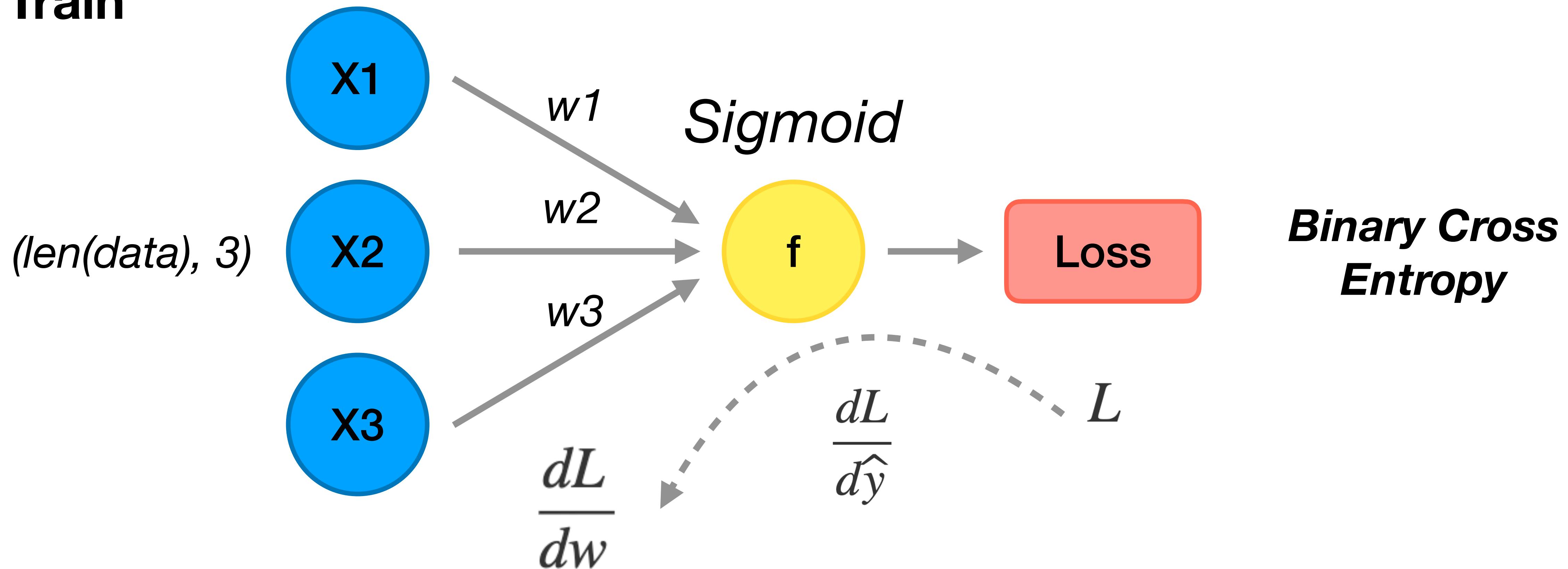
Логистическая регрессия

Train



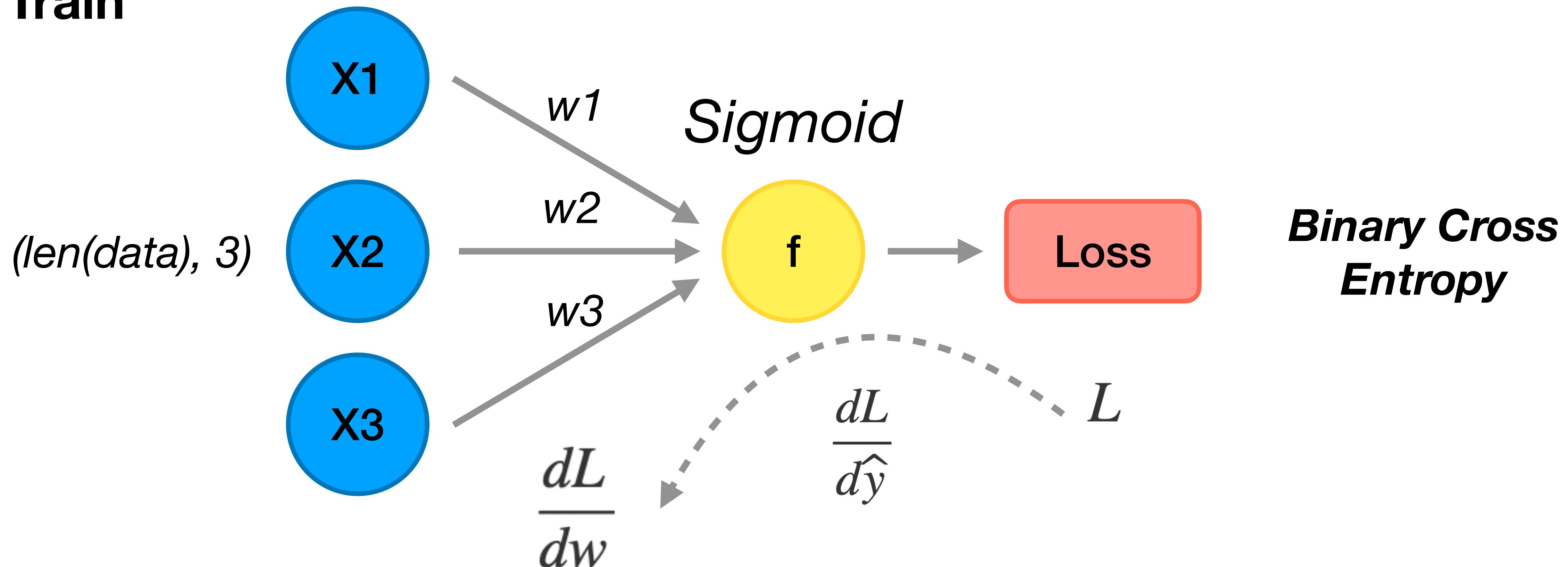
Логистическая регрессия

Train



Логистическая регрессия

Train



$$\frac{dL}{dw} = \frac{dL}{d\hat{y}} \frac{d\hat{y}}{dw}$$

Логистическая регрессия

Регуляризация

$$BCELoss = - (y * \log(pred) + (1 - y) * \log(1 - pred))$$

Логистическая регрессия

Регуляризация

$$BCELoss = - (y * \log(pred) + (1 - y) * \log(1 - pred)) + reg * R(w)$$

Логистическая регрессия

Регуляризация

$$BCELoss = - (y * \log(pred) + (1 - y) * \log(1 - pred)) + \boxed{\text{reg}} * R(w)$$



Гиперпараметр

Логистическая регрессия

Регуляризация

$$BCELoss = - (y * \log(pred) + (1 - y) * \log(1 - pred)) + \boxed{\text{reg}} * R(w)$$

$$R(W) = \sum_k \sum_l W_{k,l}^2$$

$$R(W) = \sum_k \sum_l |W_{k,l}|$$

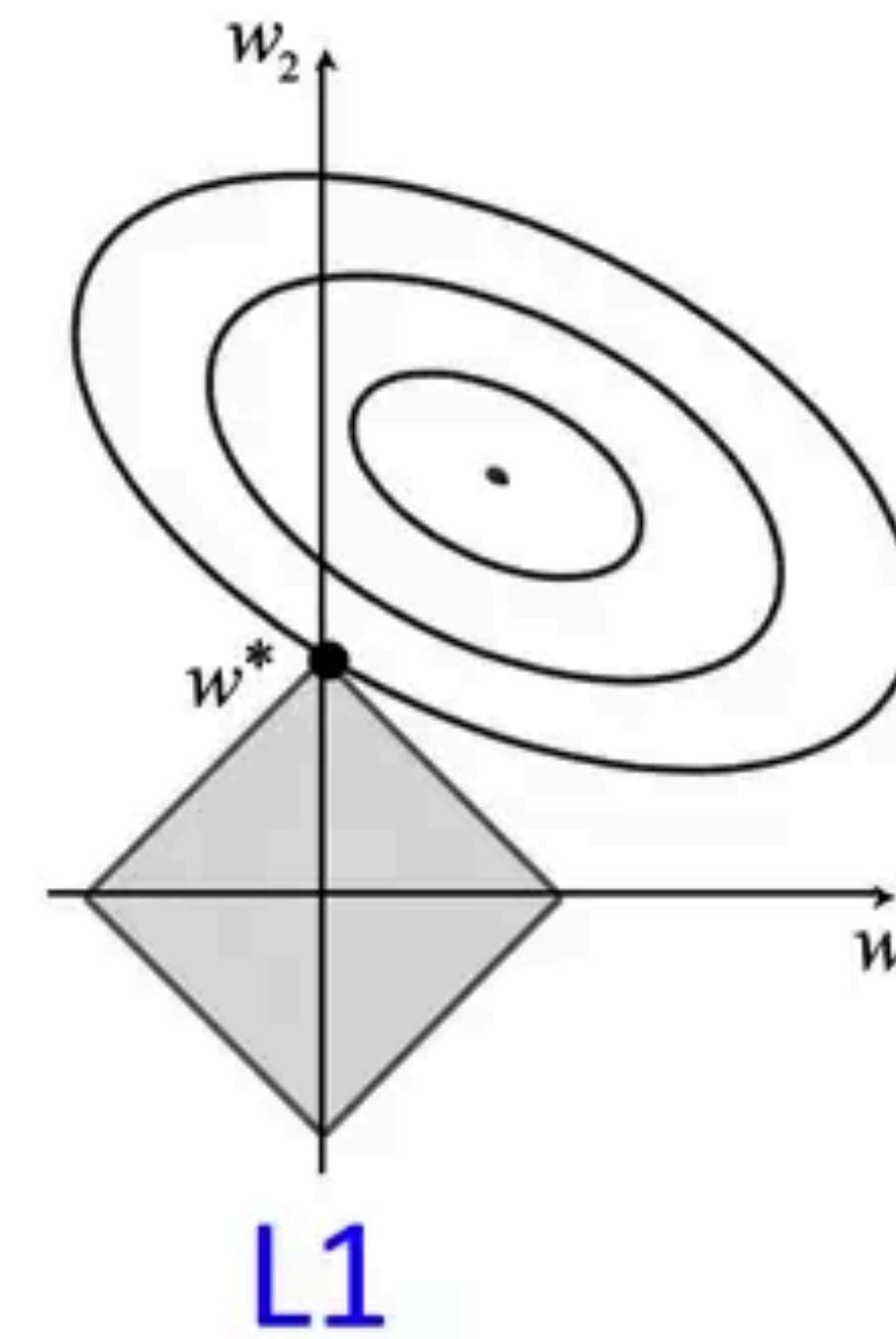
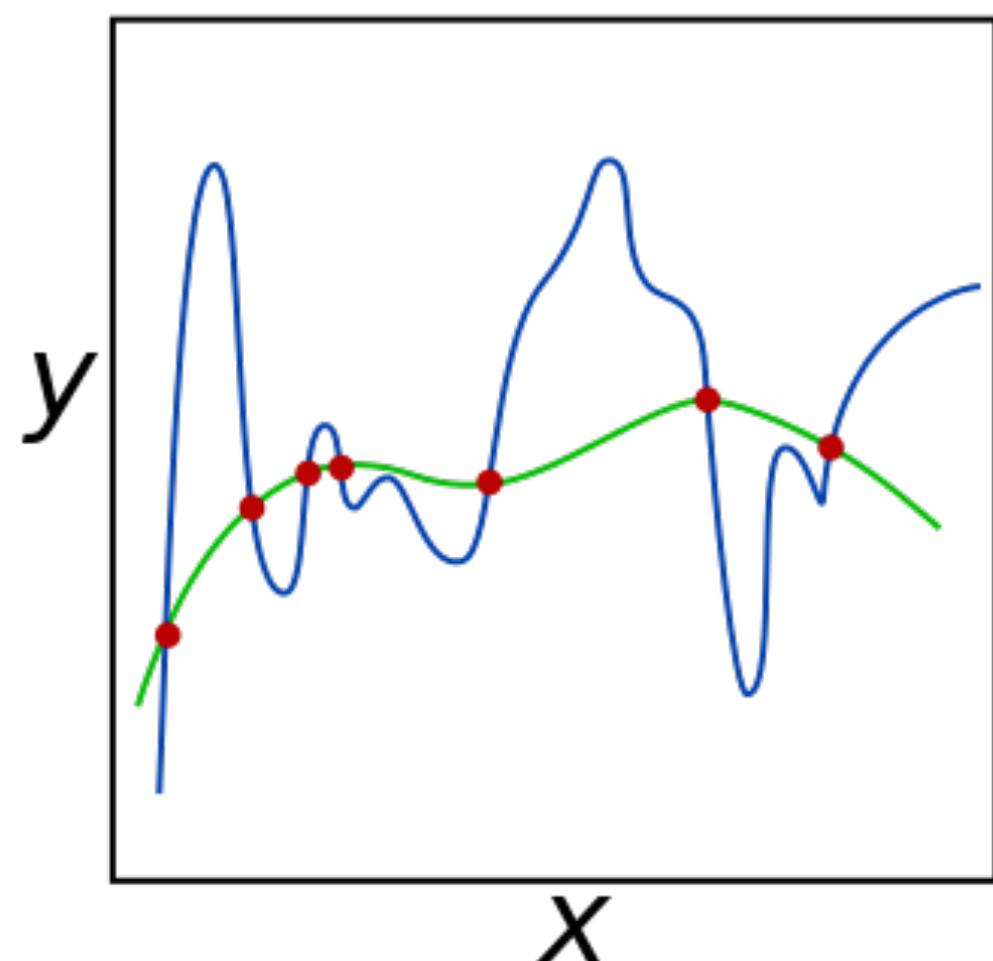
↓
Гиперпараметр

Логистическая регрессия

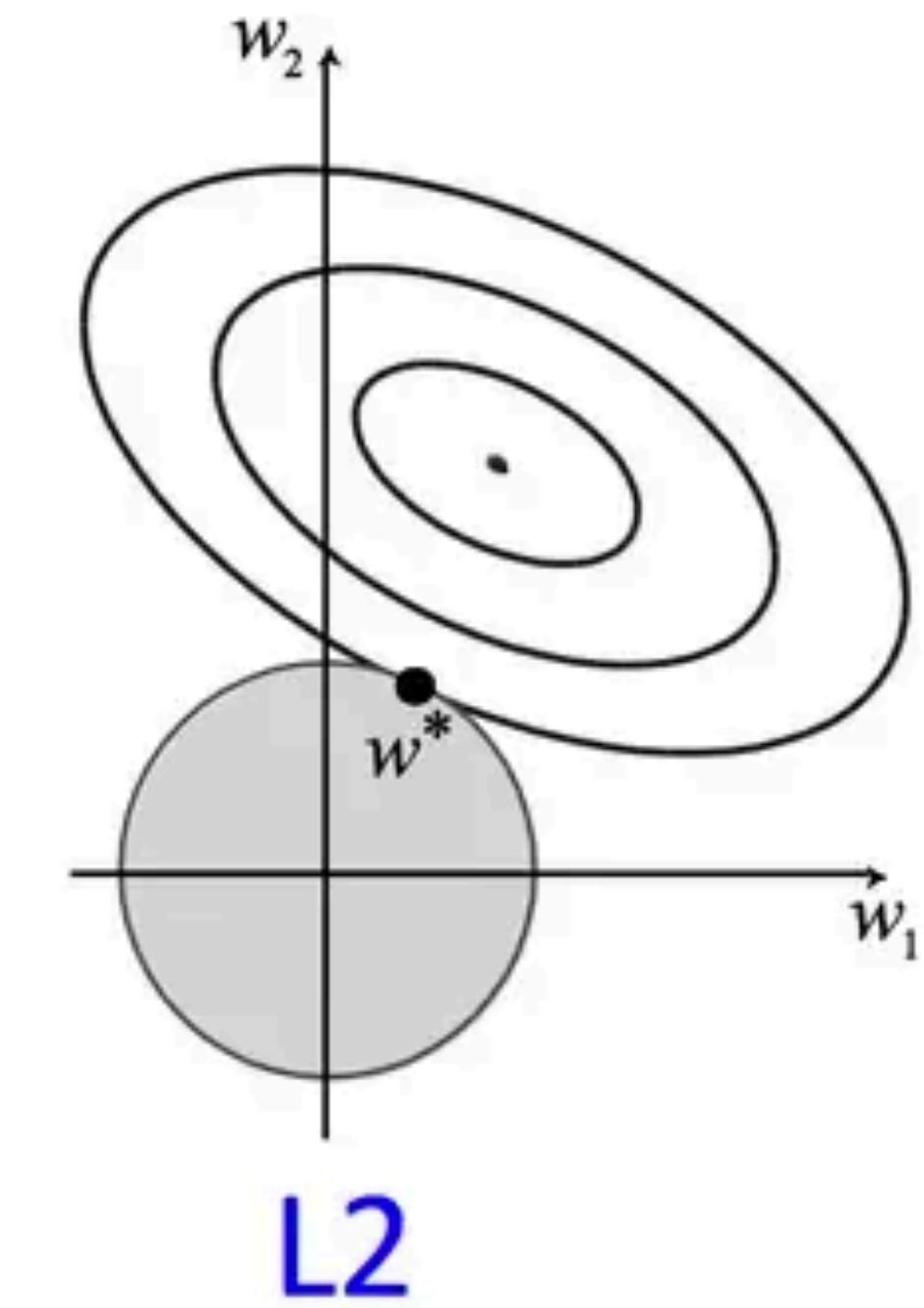
Регуляризация

$$R(W) = \sum_k \sum_l W_{k,l}^2$$

$$R(W) = \sum_k \sum_l |W_{k,l}|$$



L1



L2

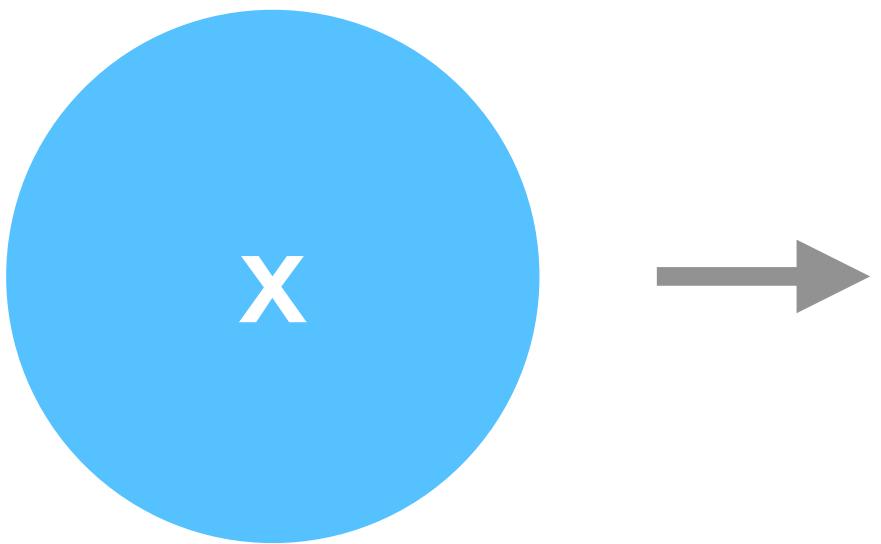
Нейронные сети

Нейронные сети

Forward

Нейронные сети

Forward



Нейронные сети

Forward



Нейронные сети

Forward



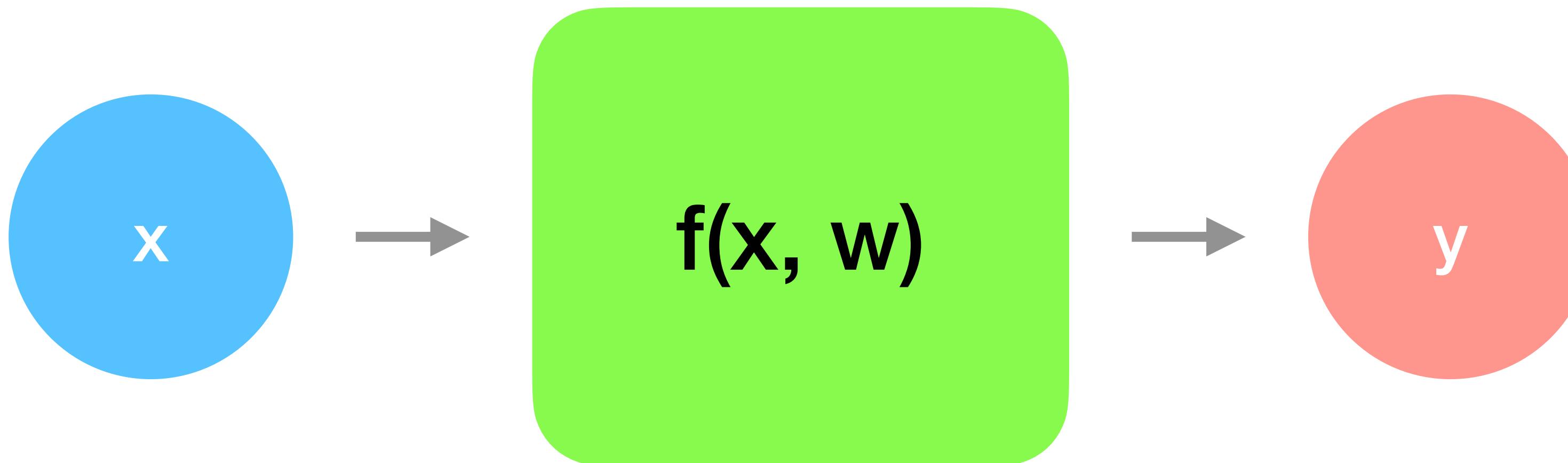
Нейронные сети

Forward



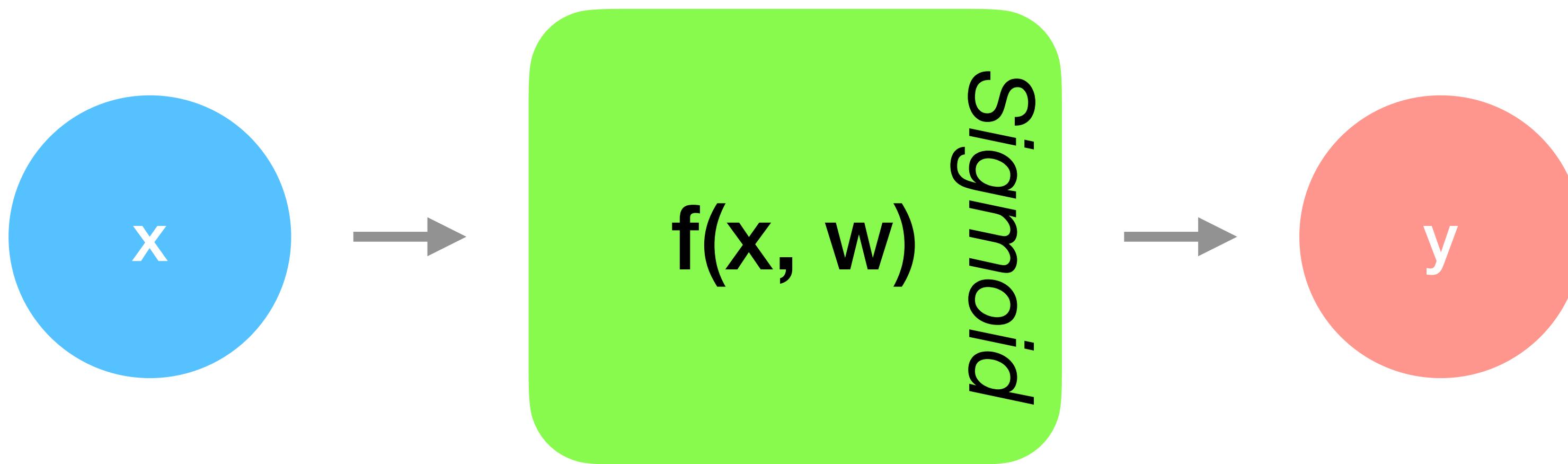
Нейронные сети

Forward



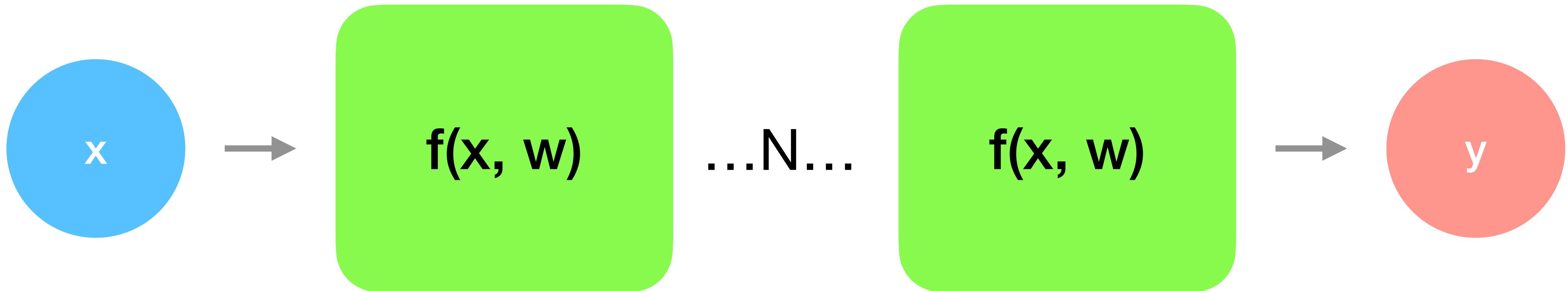
Нейронные сети

Forward



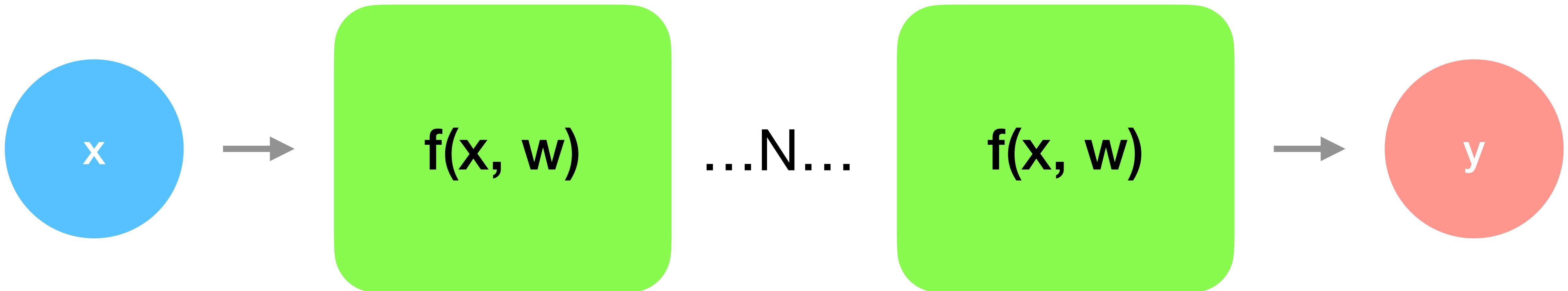
Нейронные сети

Forward



Нейронные сети

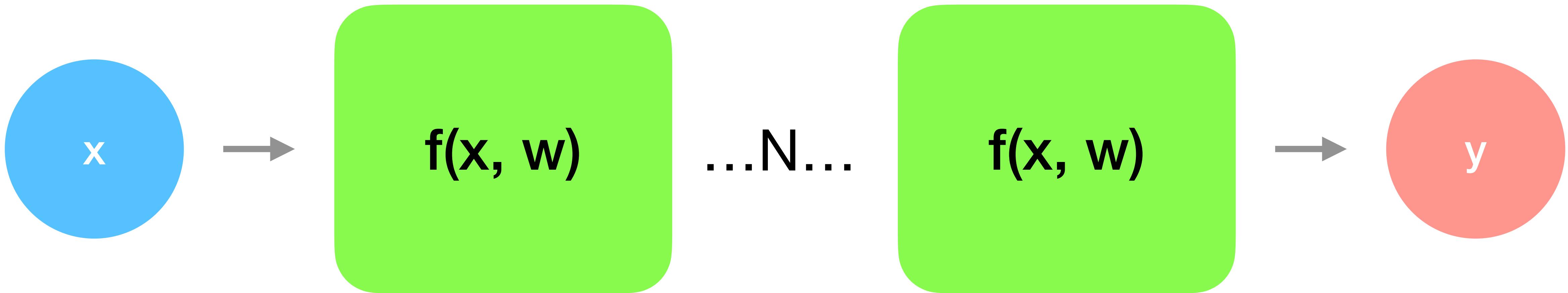
Forward



$$f_n(f_3(f_2(f_1(x, w_1), w_2), w_3), w_n)$$

Нейронные сети

Forward

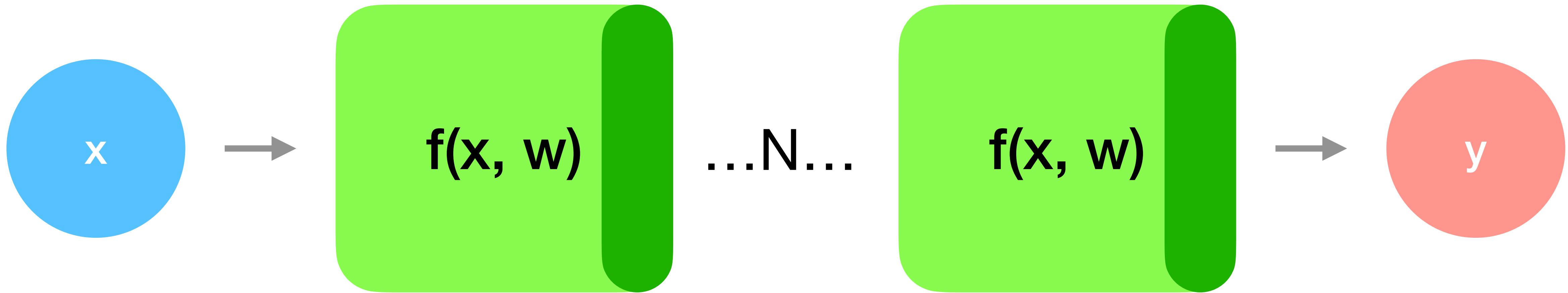


$$f_n(f_3(f_2(f_1(x, w_1), w_2), w_3), w_n)$$

$$f(x) = \text{non_linearity}(x * w + b)$$

Нейронные сети

Forward

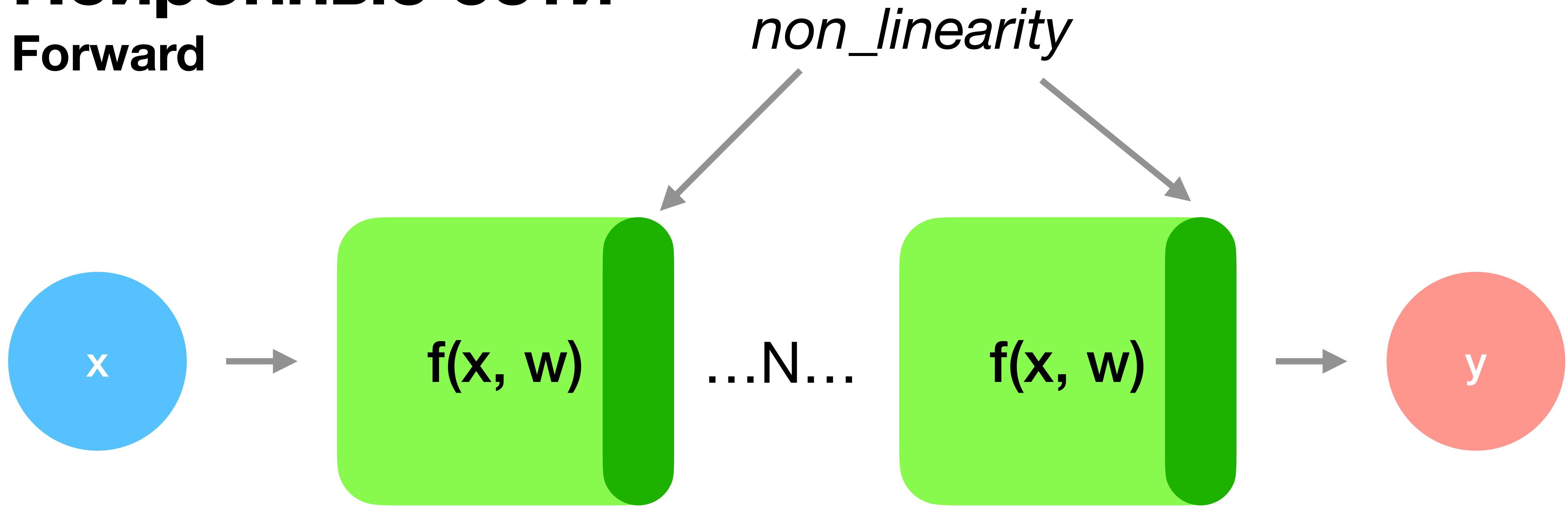


$$f_n(f_3(f_2(f_1(x, w_1), w_2), w_3), w_n)$$

$$f(x) = \text{non_linearity}(x * w + b)$$

Нейронные сети

Forward

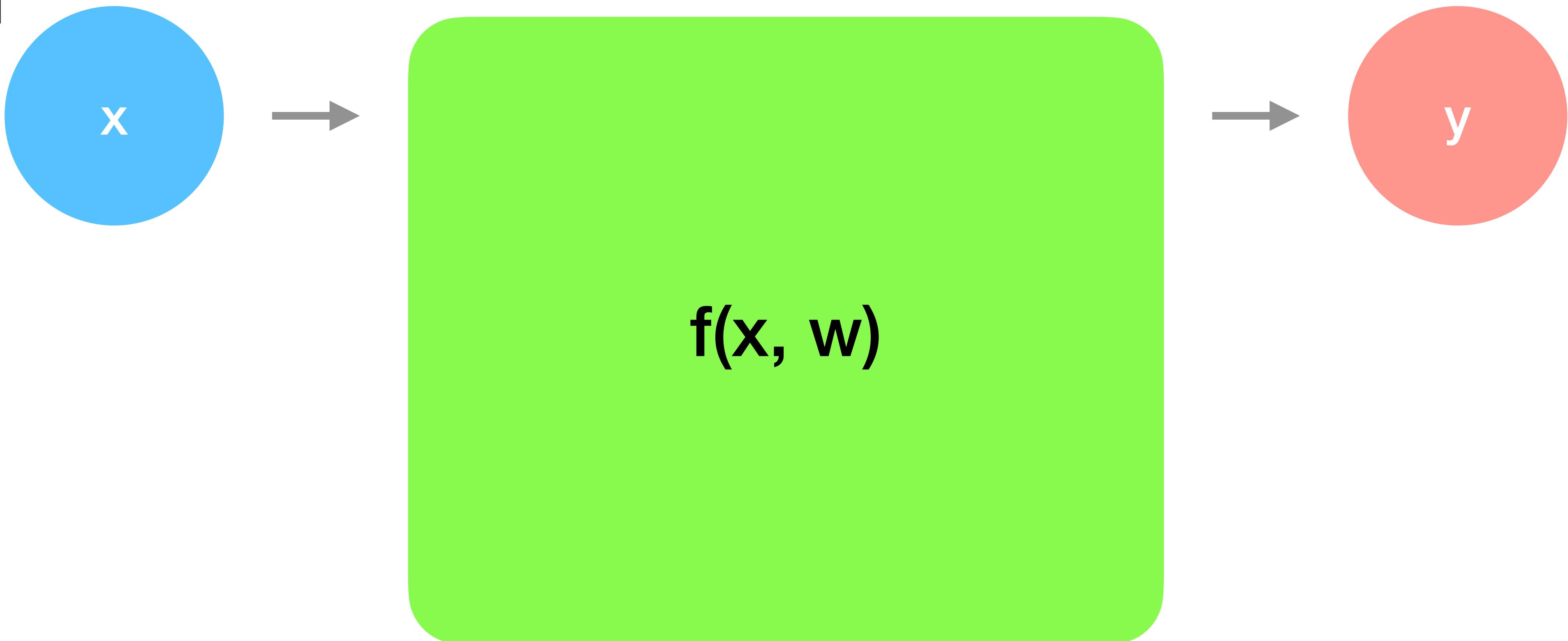


$$f_n(f_3(f_2(f_1(x, w_1), w_2), w_3), w_n)$$

$$f(x) = \text{non_linearity}(x * w + b)$$

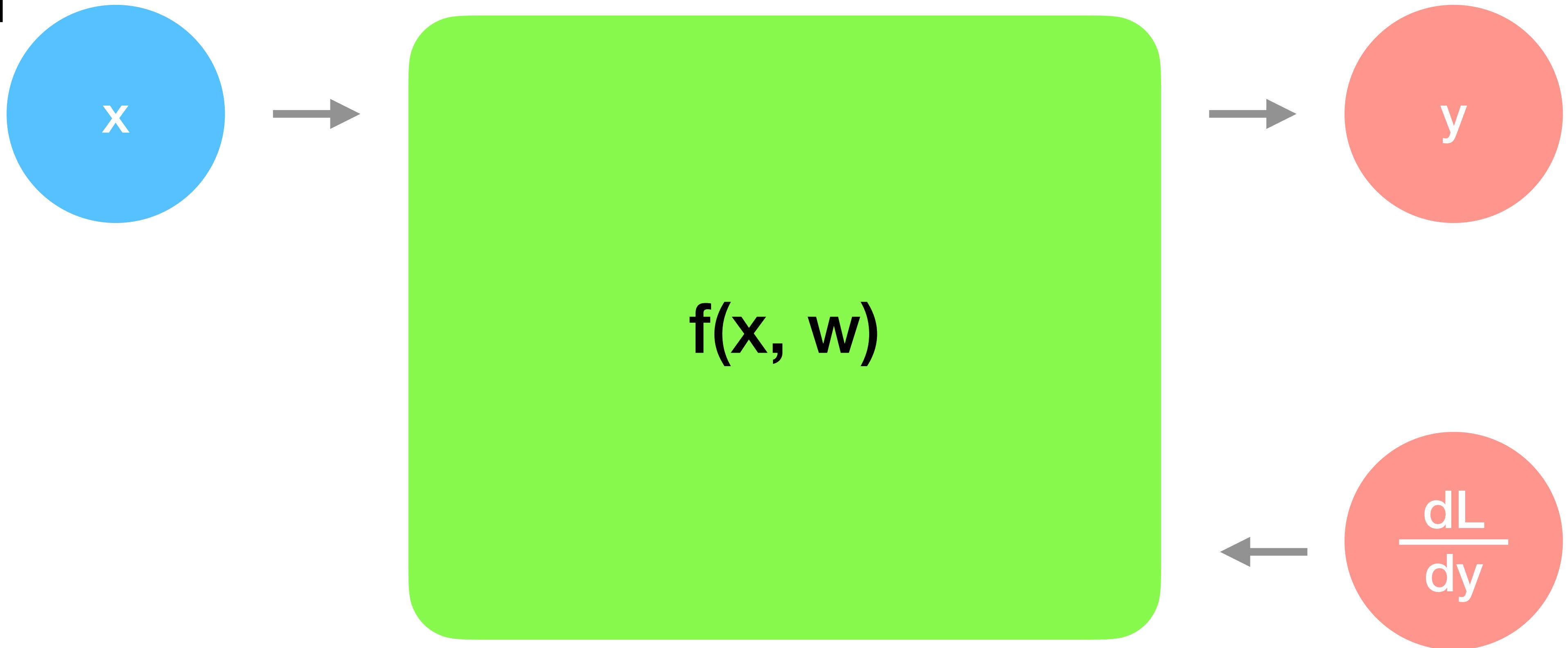
Нейронные сети

Backward



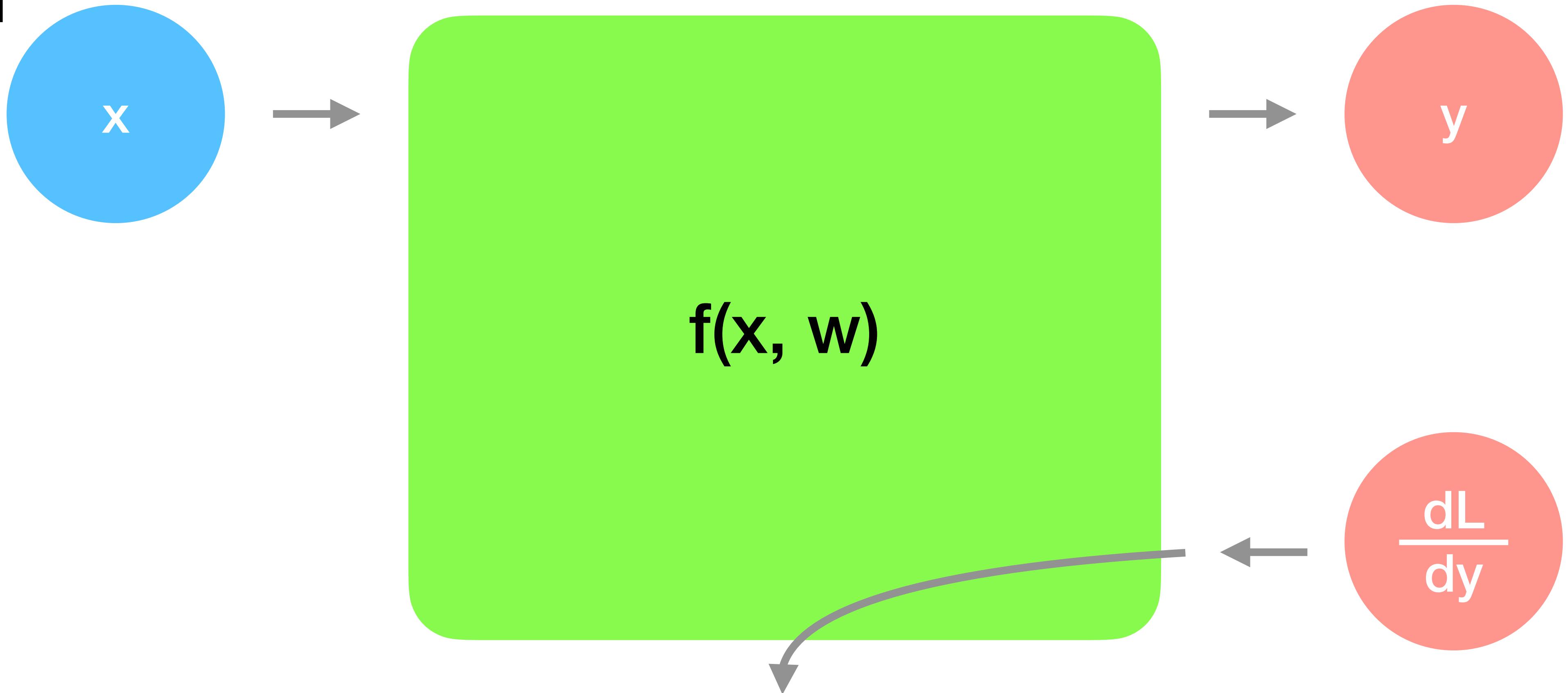
Нейронные сети

Backward



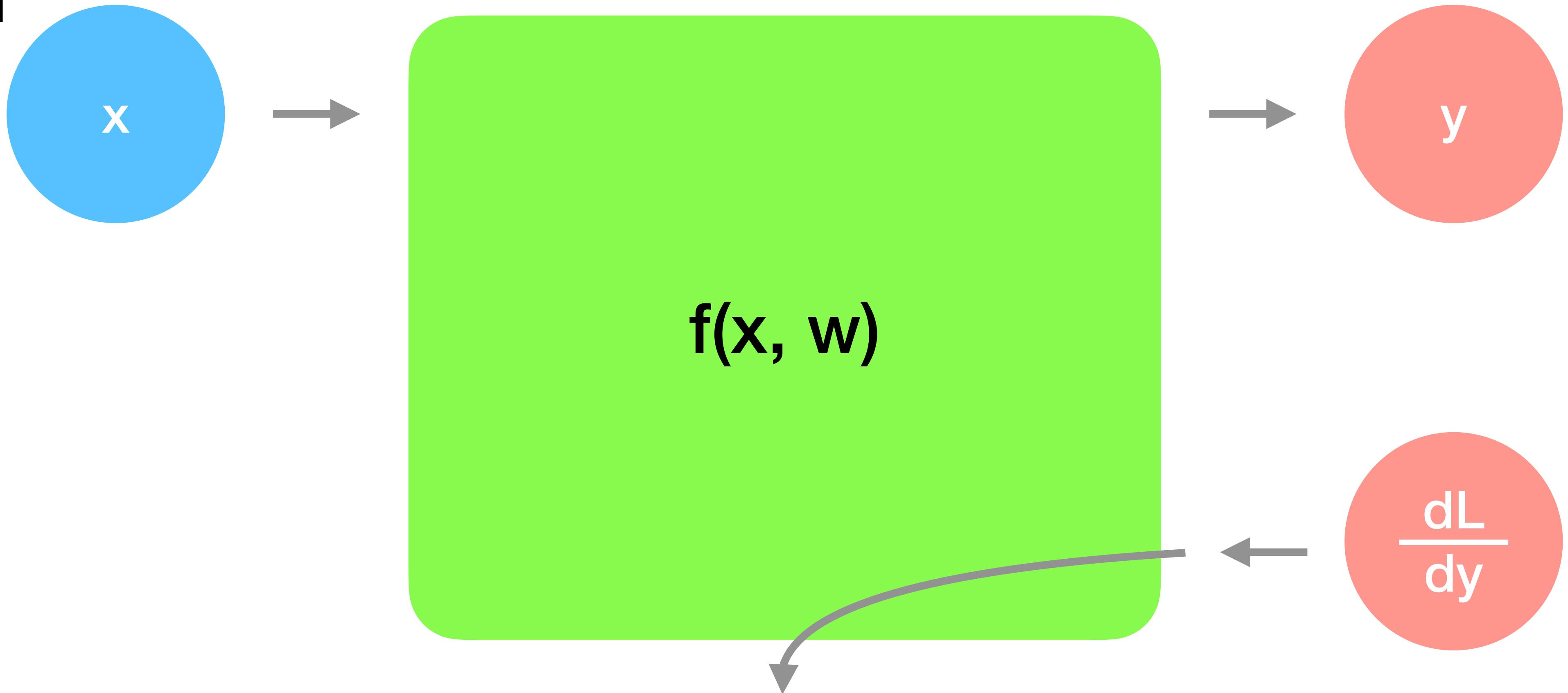
Нейронные сети

Backward



Нейронные сети

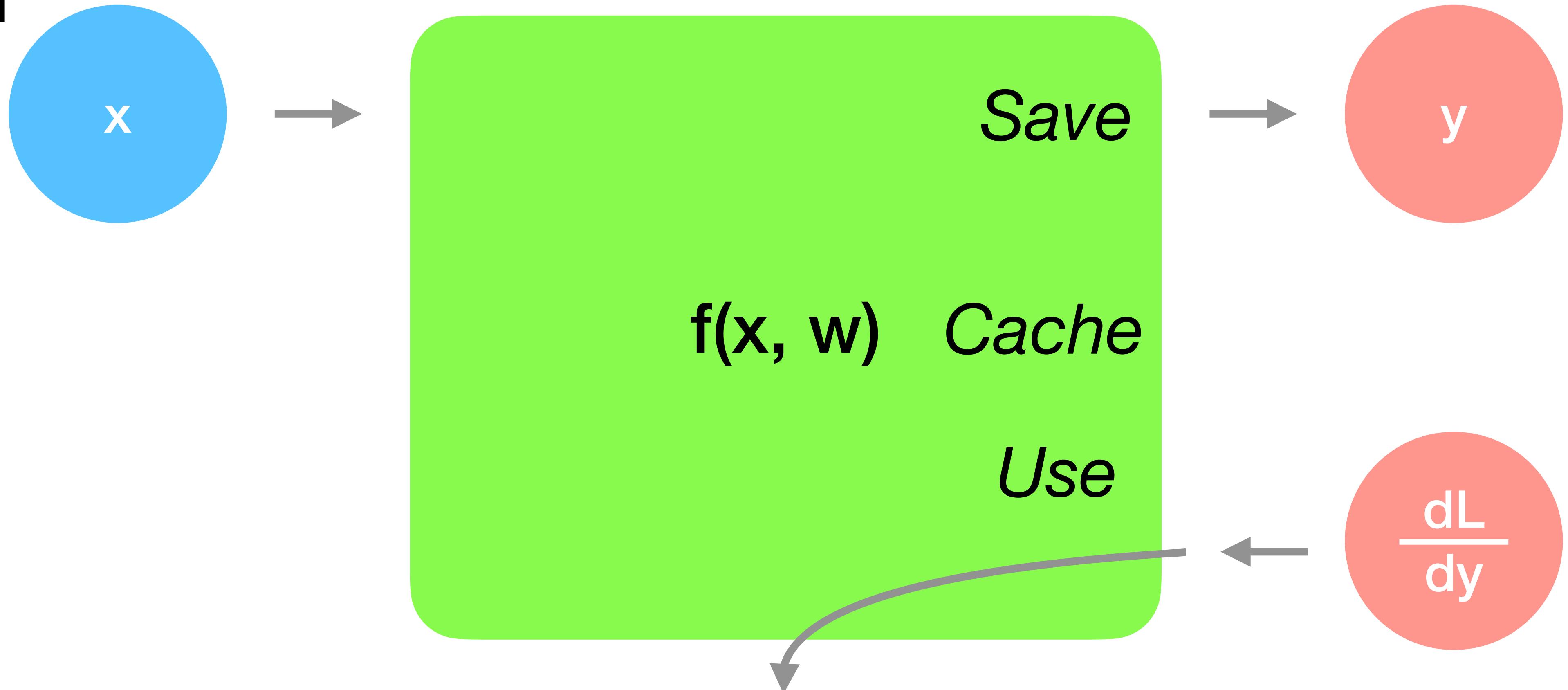
Backward



$$\frac{dL}{dw} = \frac{dL}{d\hat{y}} \frac{d\hat{y}}{dw}$$

Нейронные сети

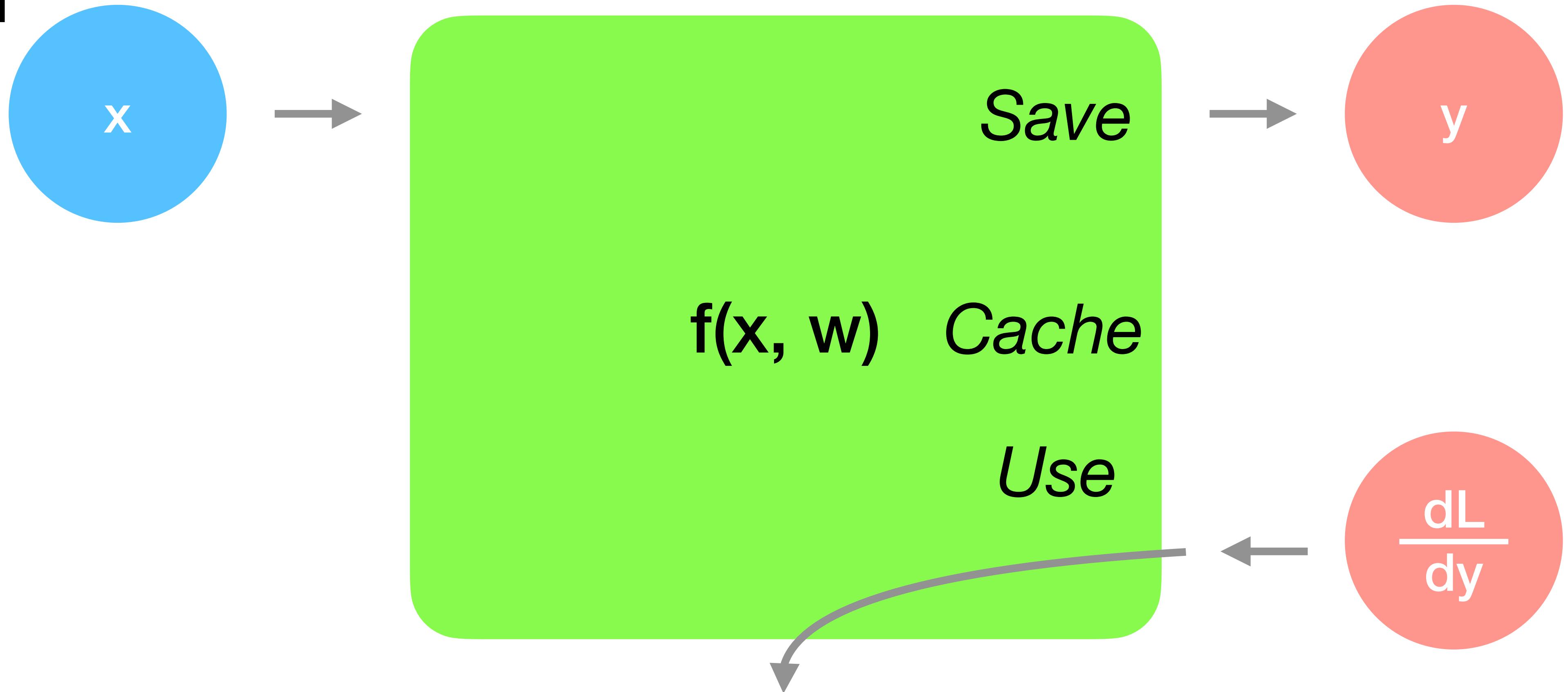
Backward



$$\frac{dL}{dw} = \frac{dL}{d\hat{y}} \frac{d\hat{y}}{dw}$$

Нейронные сети

Backward

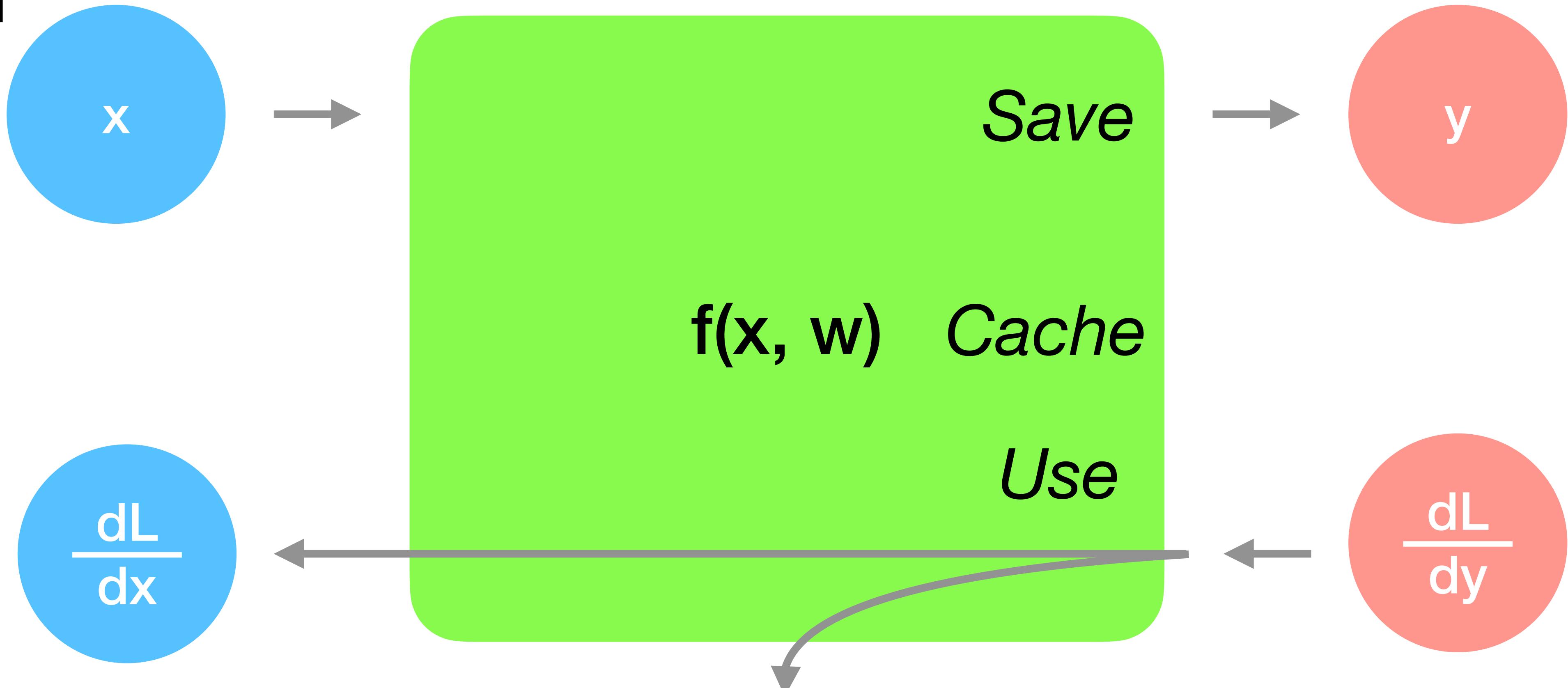


$$w = w - \alpha \frac{dL}{dw}$$

$$\frac{dL}{dw} = \frac{dL}{d\hat{y}} \frac{d\hat{y}}{dw}$$

Нейронные сети

Backward

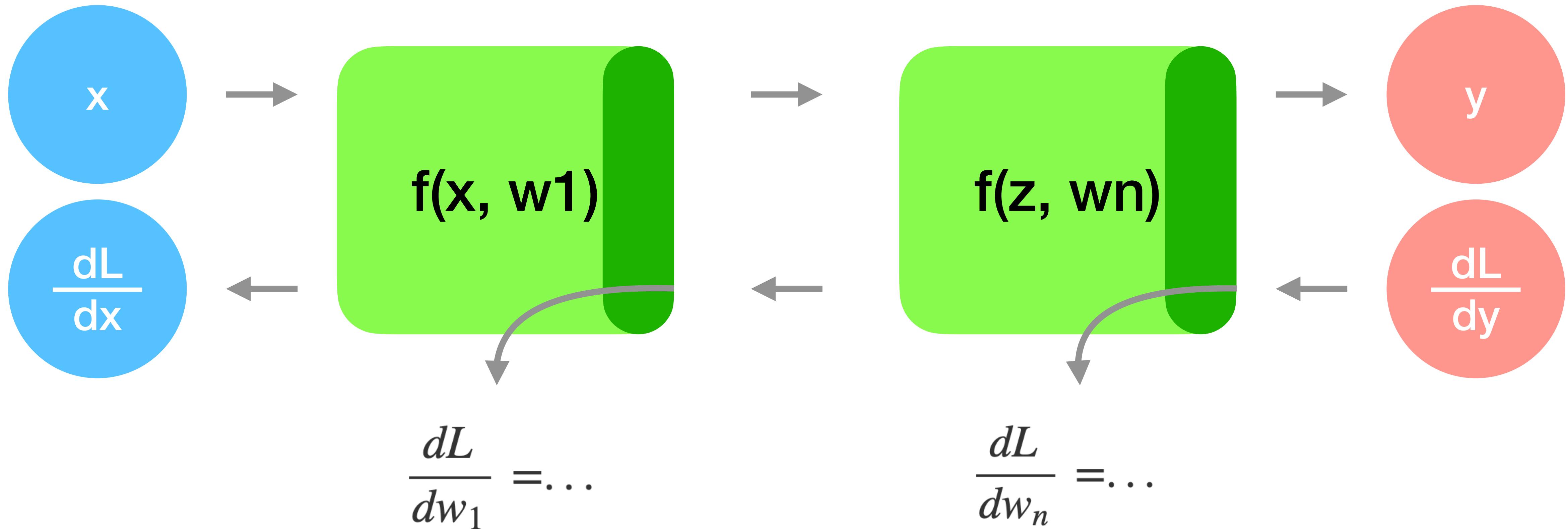


$$w = w - \alpha \frac{dL}{dw}$$

$$\frac{dL}{dw} = \frac{dL}{d\hat{y}} \frac{d\hat{y}}{dw}$$

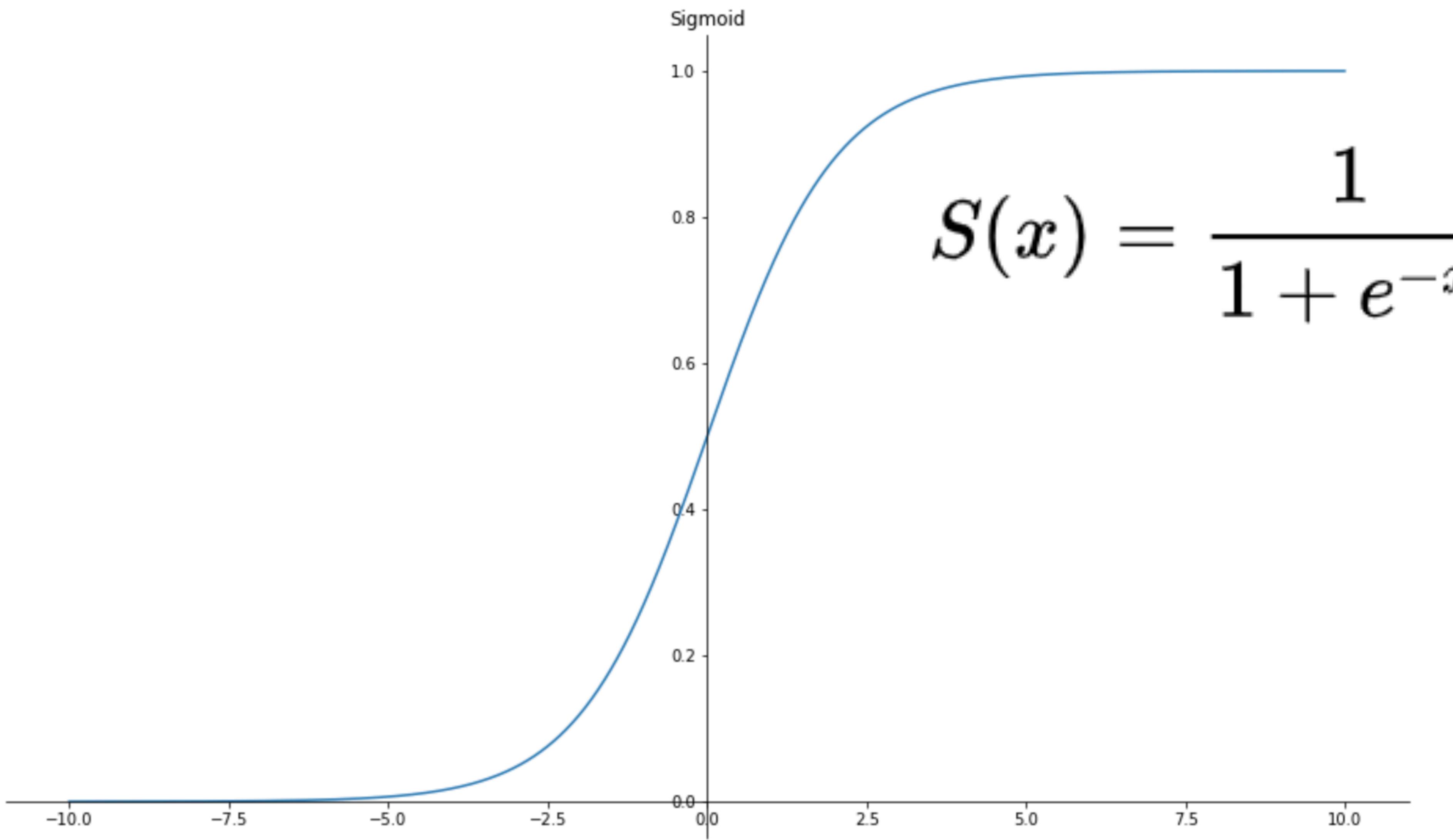
Нейронные сети

Backward

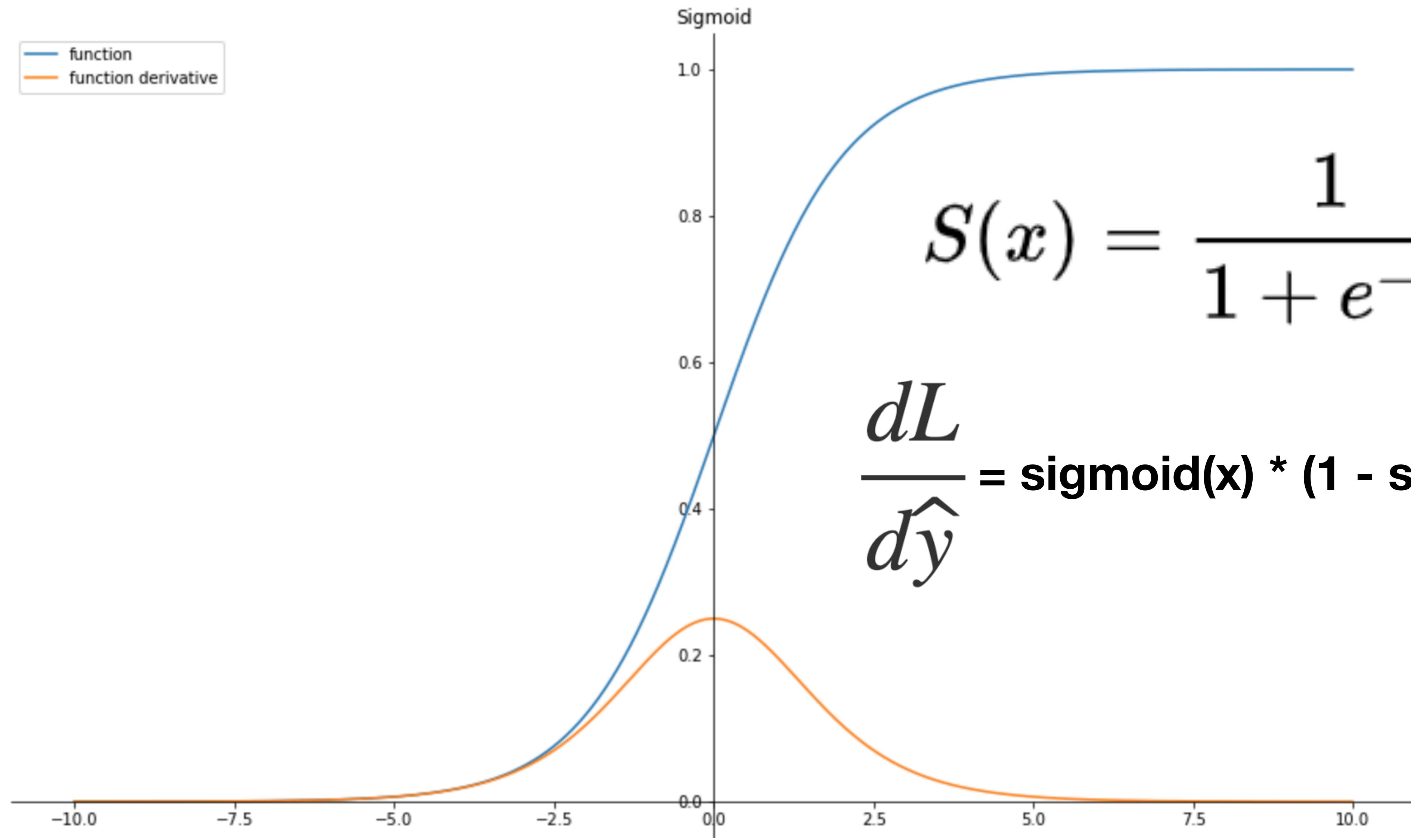


Функции активации

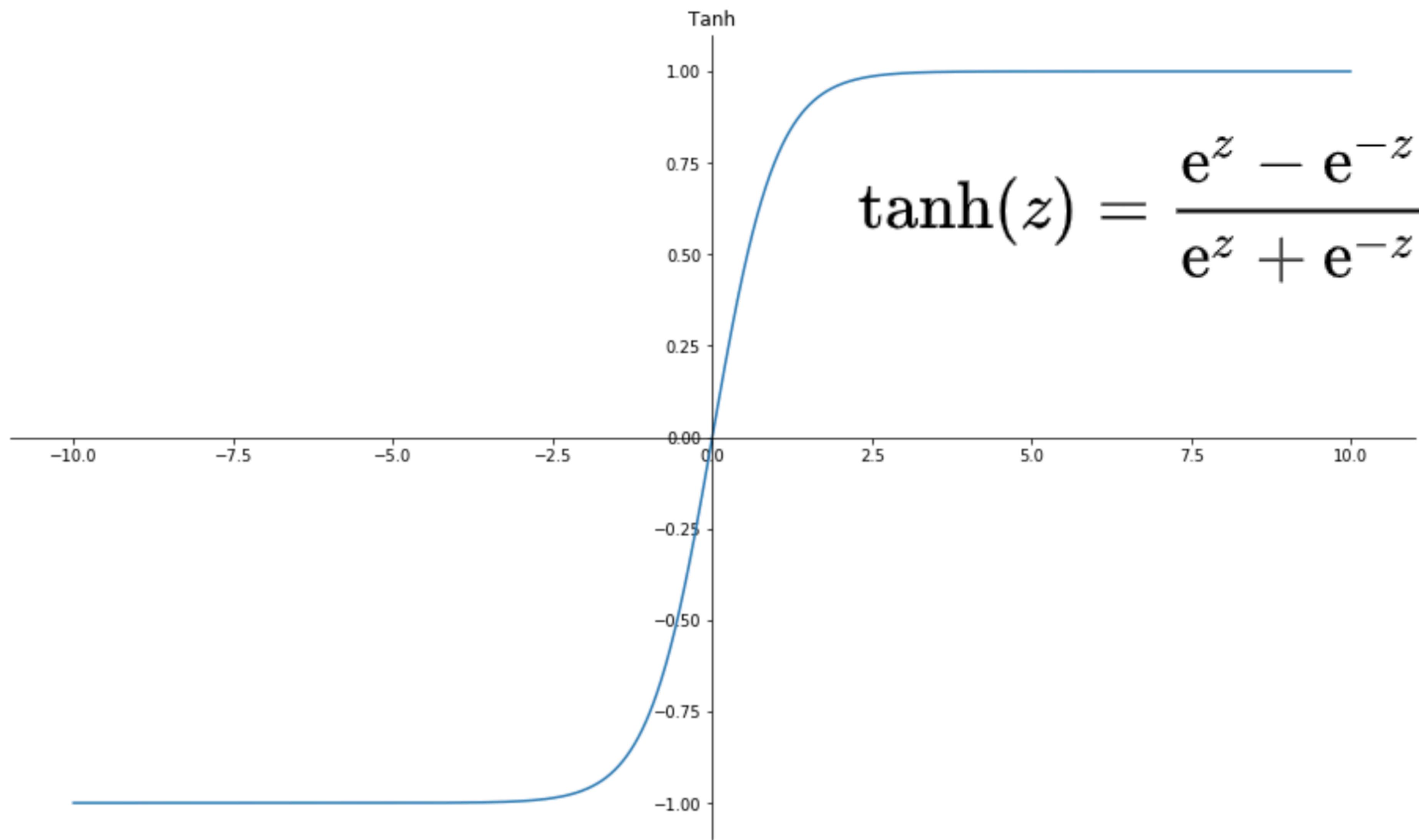
Sigmoid



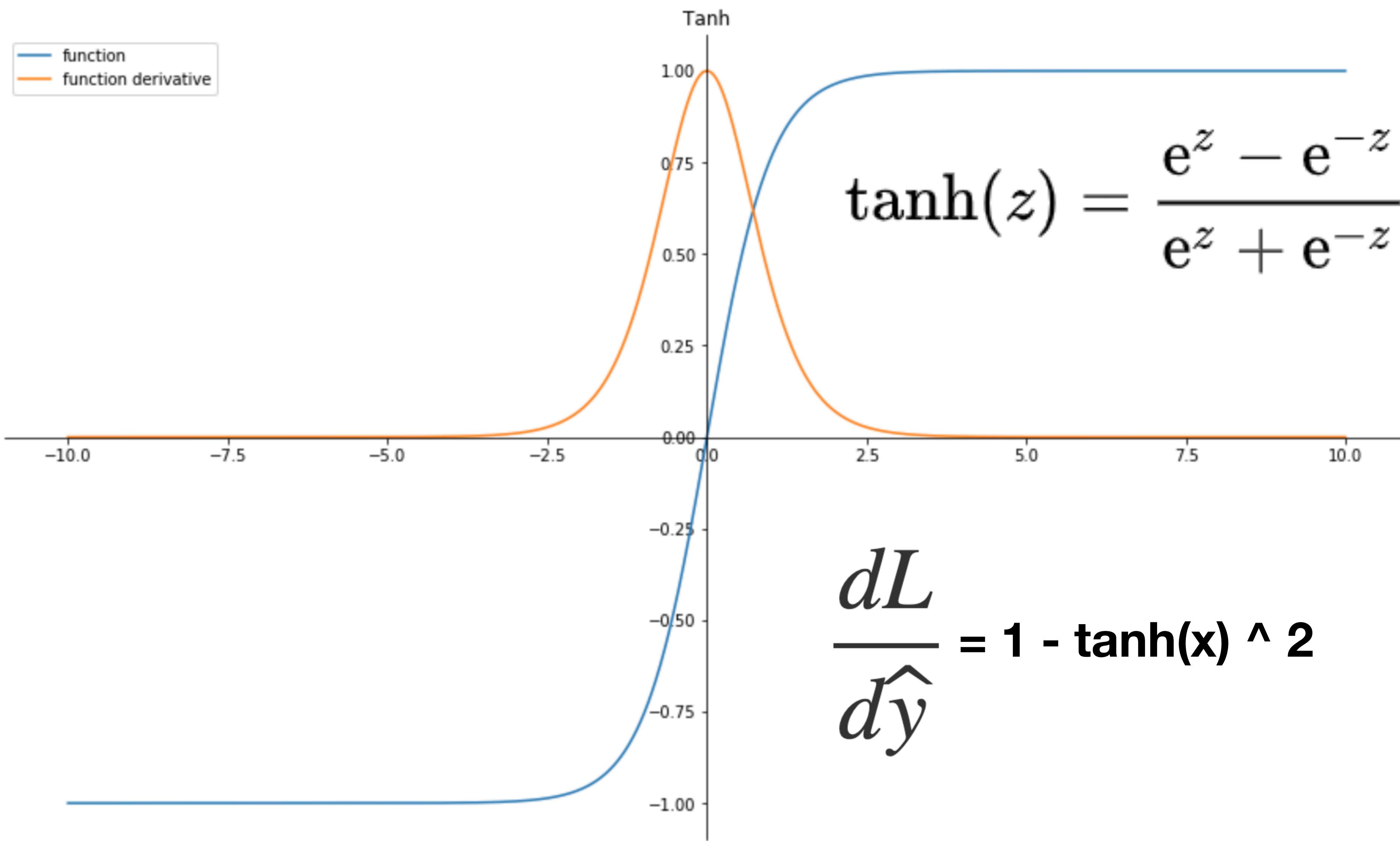
Sigmoid



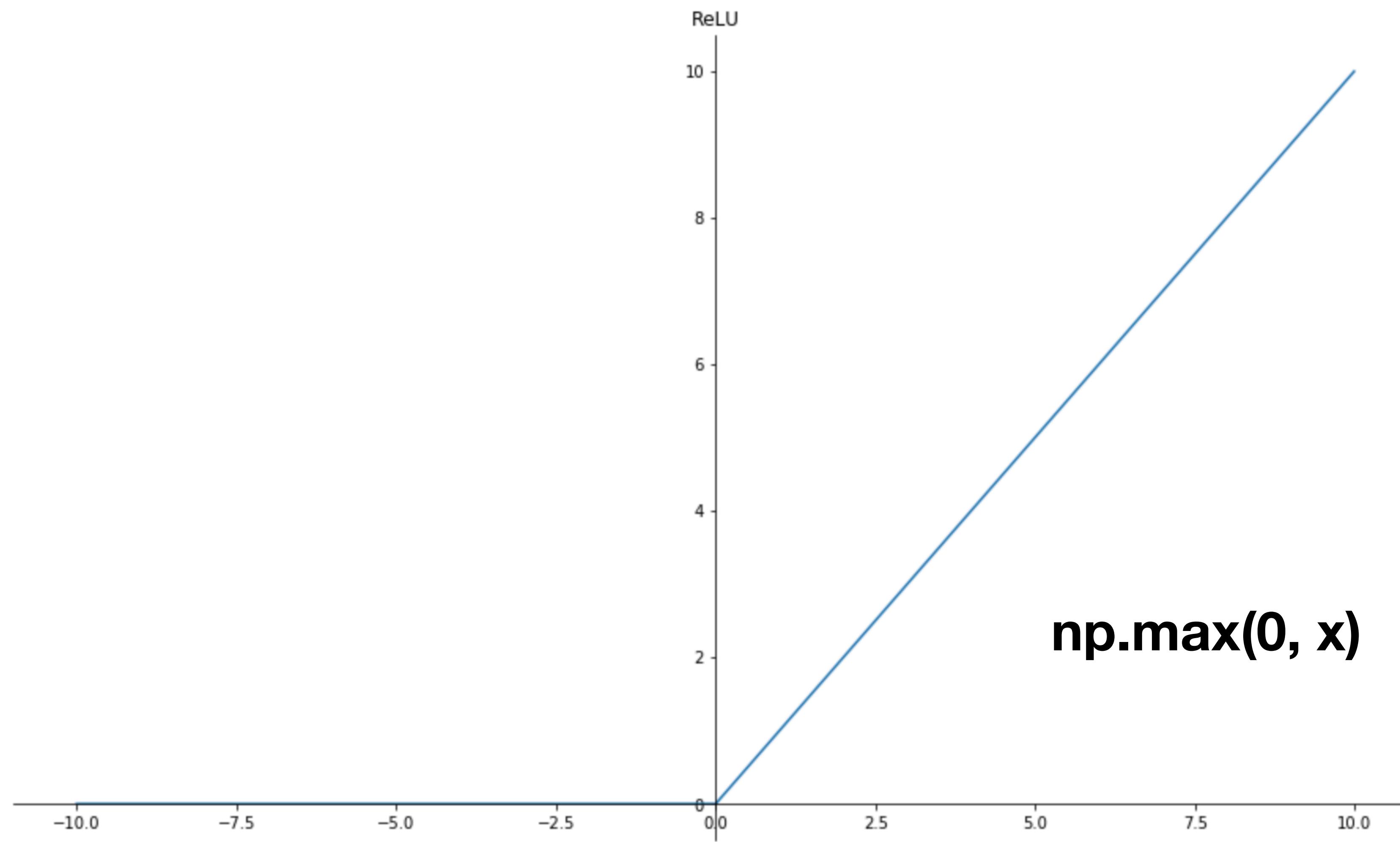
Tanh



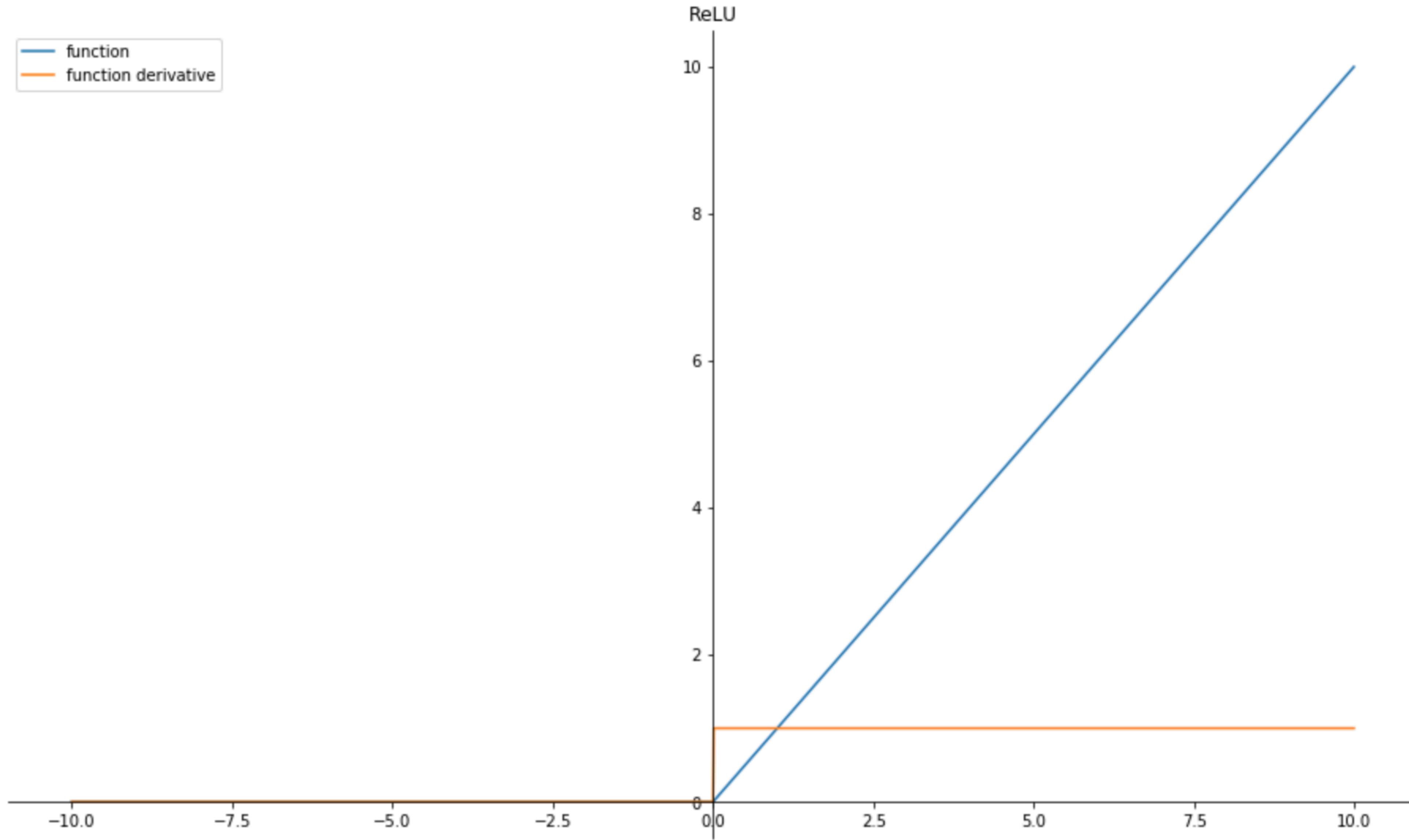
Tanh



ReLU



ReLU

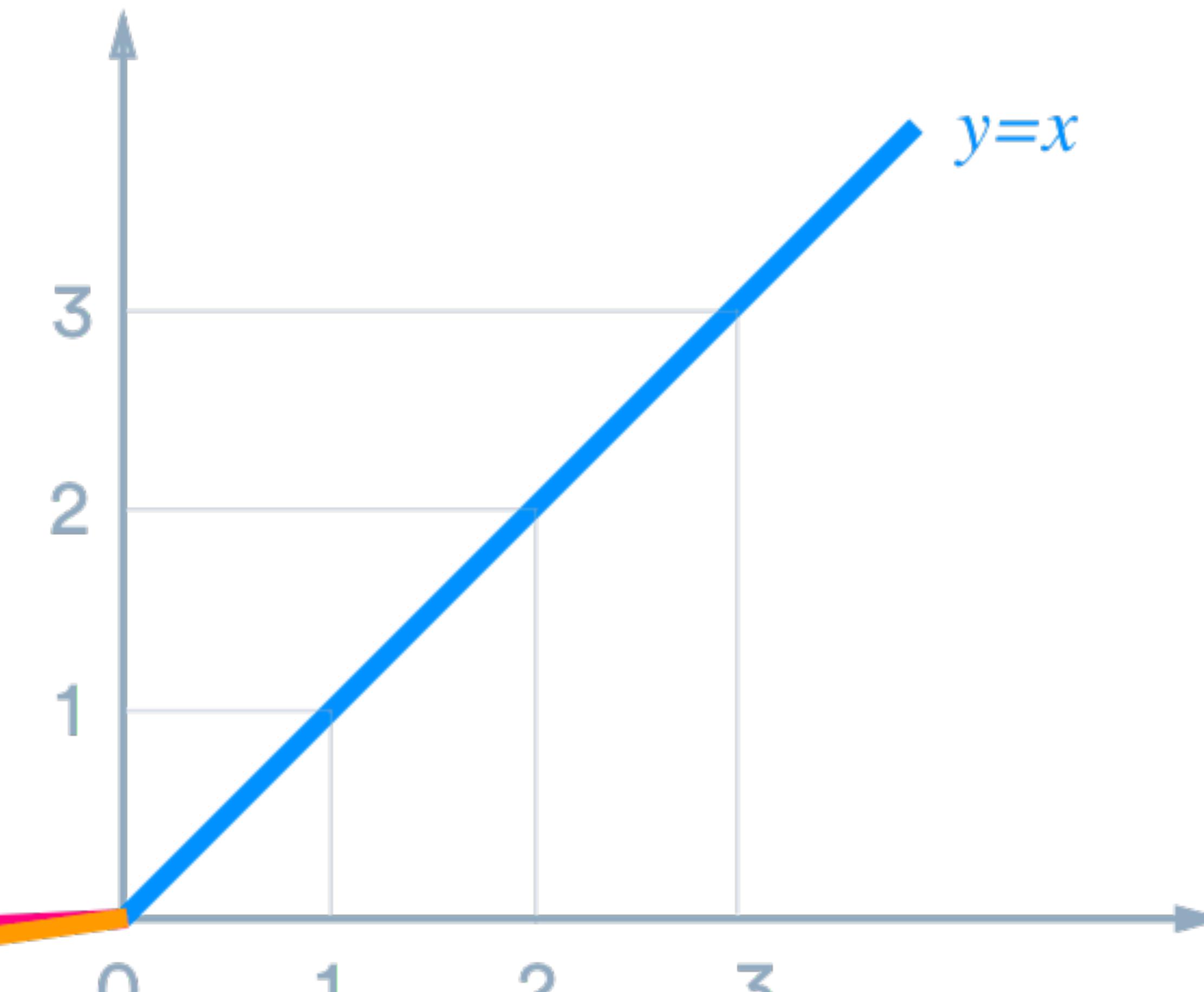


Leaky ReLU

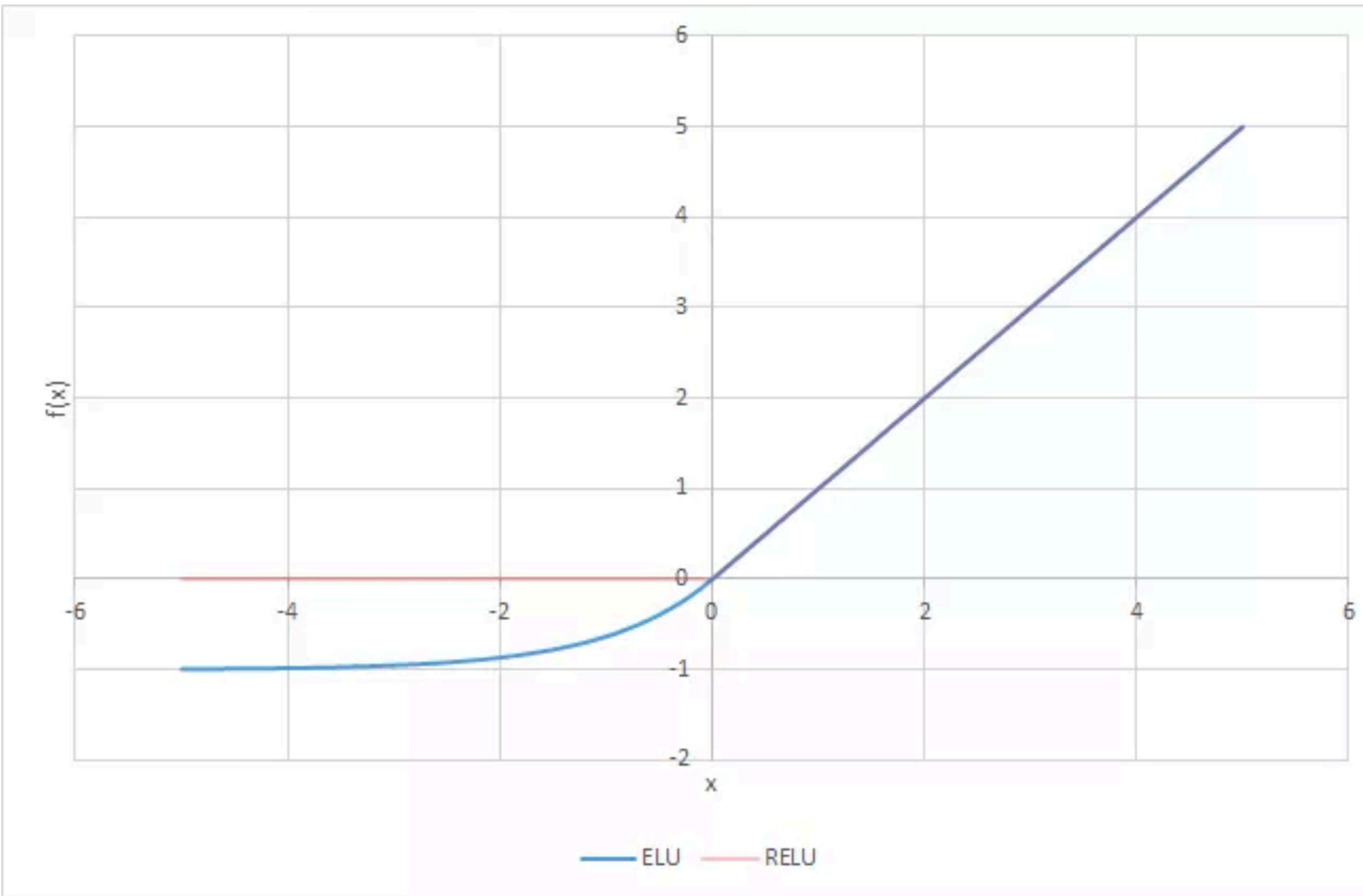
Leaky ReLU: $y=0.01x$



Parametric ReLU: $y=ax$



Leaky ReLU



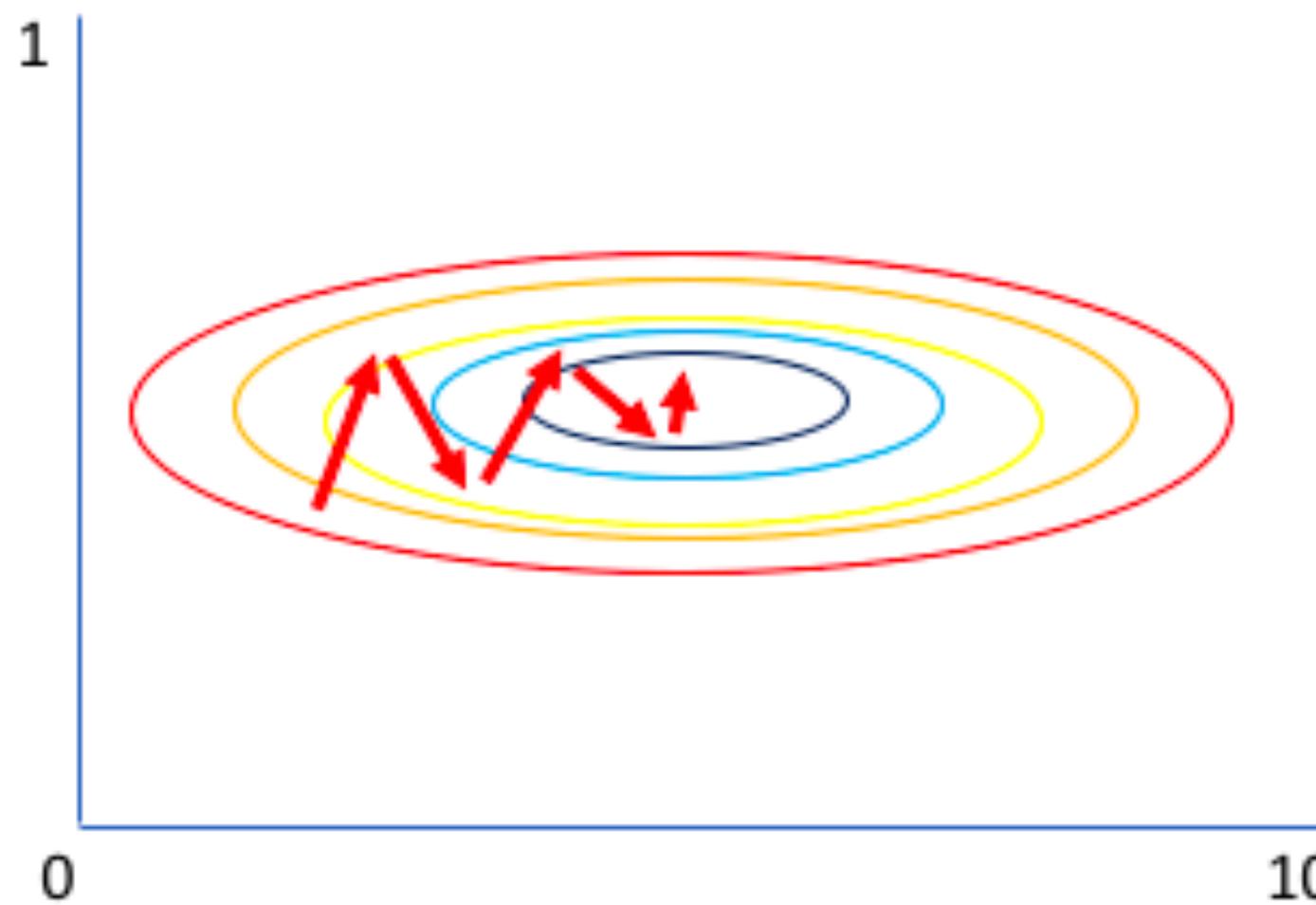
Важные замечания

Про функции активации

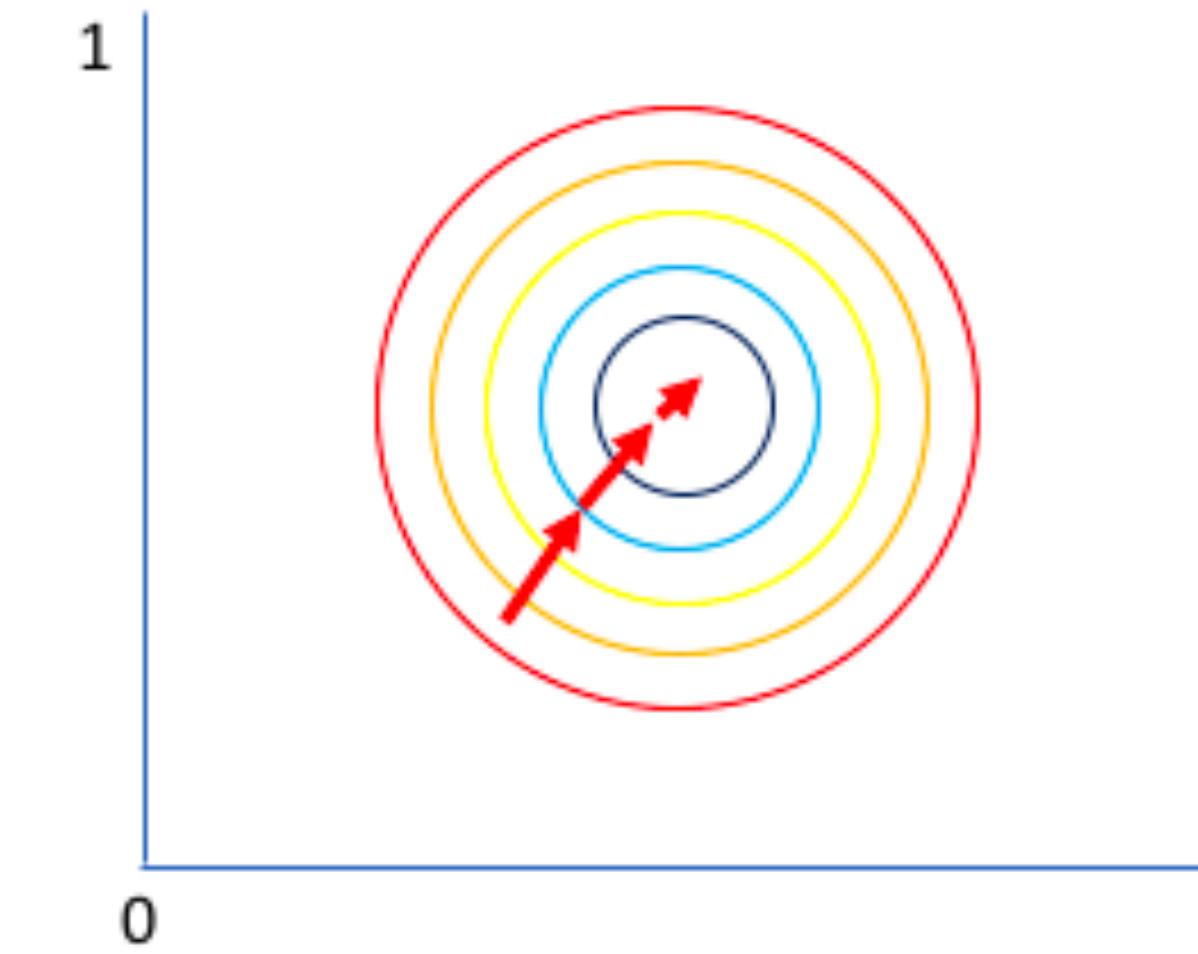
- Не используйте сигмоиду как функцию активации в промежуточных слоях MLP
- Но мы поговорим далее в курсе где вы все-таки можете и будете использовать сигмоиду в промежуточных слоях
- ReLU быстро считается и является стандартом для использования
- Сильно дальше в курсе мы поговорим про другие функции активации

Нормализация входных данных

Why normalize?



Gradient of larger parameter
dominates the update



Both parameters can be
updated in equal proportions

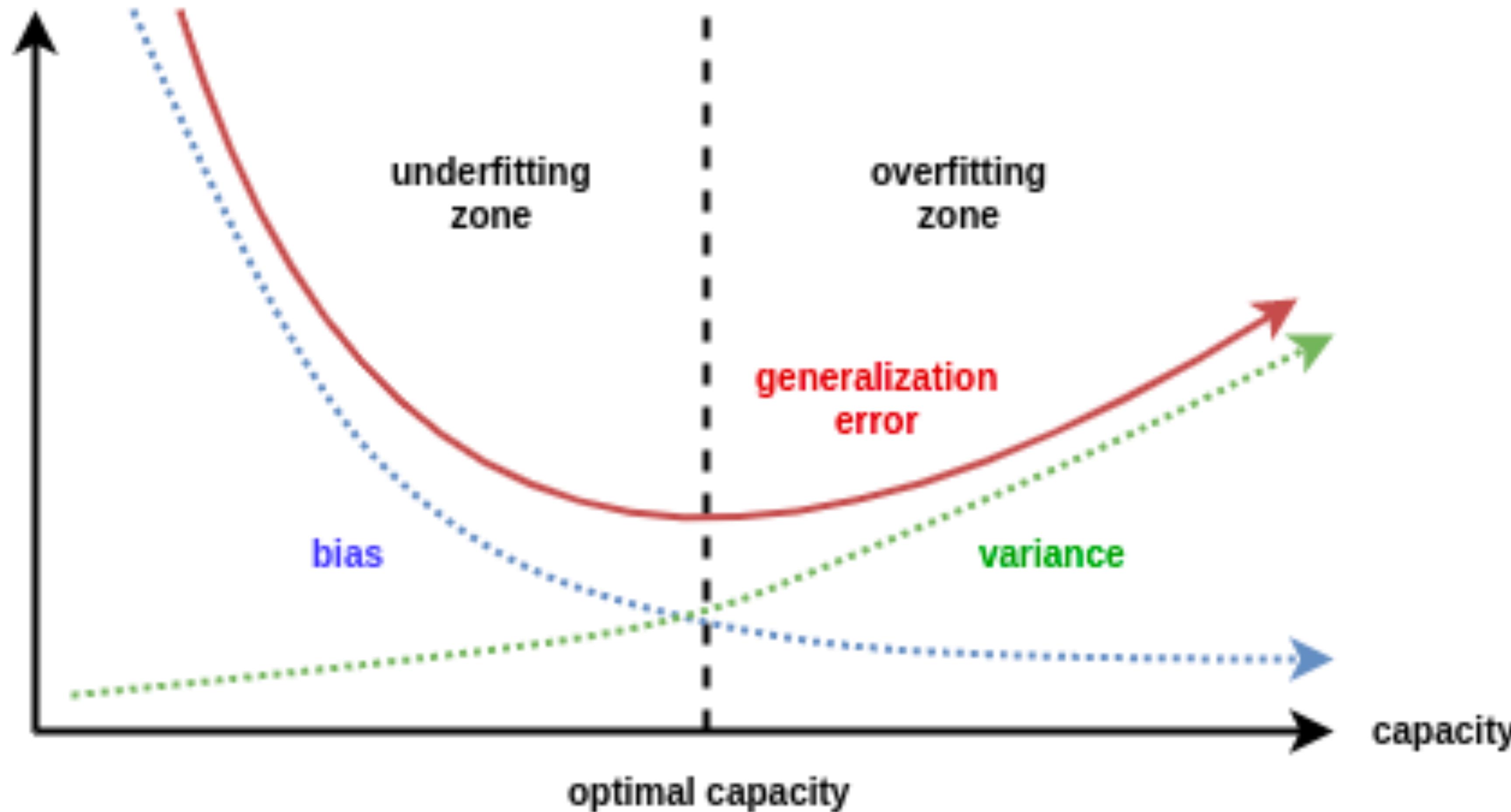
Нормализация входных данных

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$$

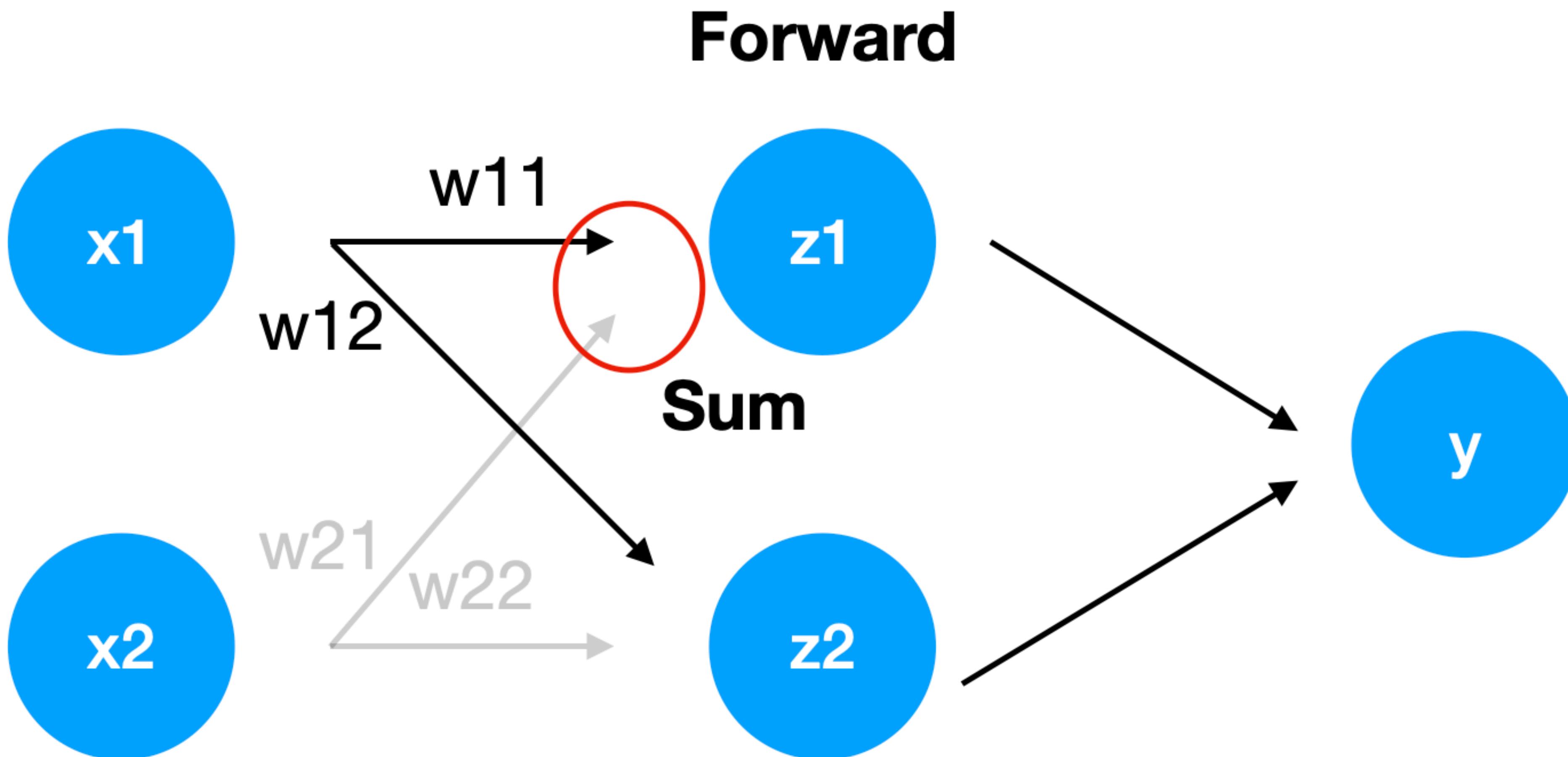
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$

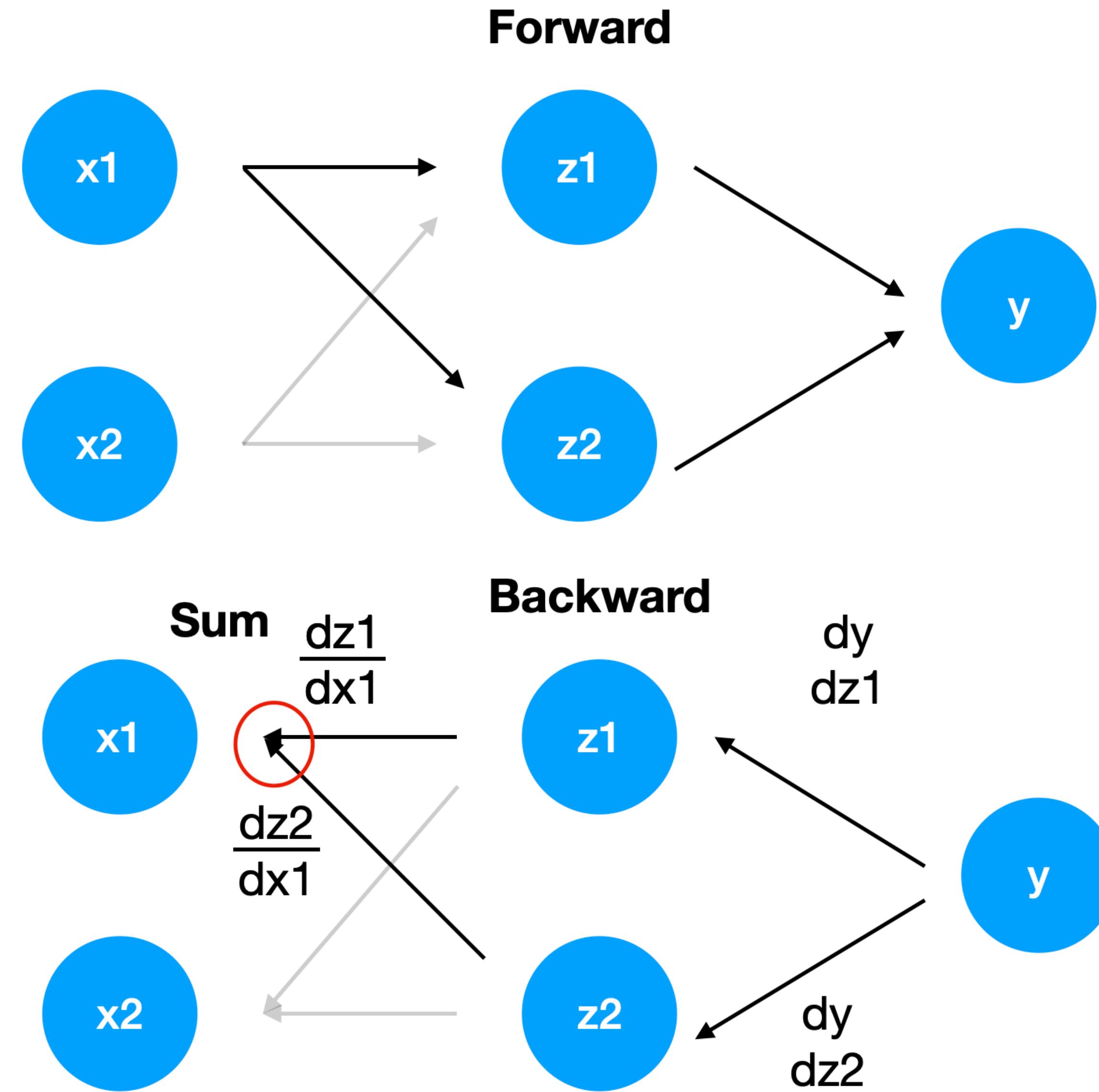
Early Stopping



Gradients

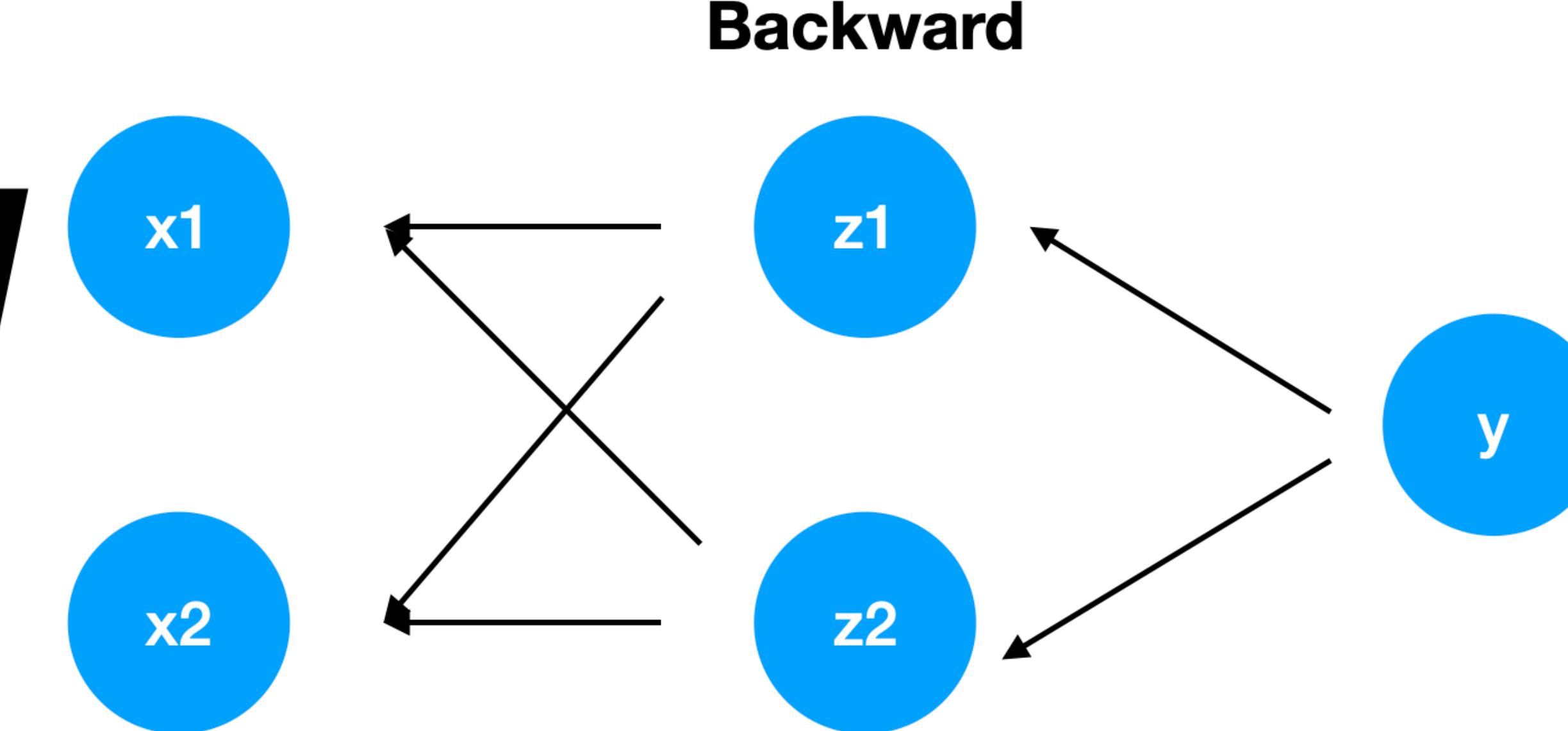


Gradients



Gradients

$$\left[\begin{array}{c} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \end{array} \right]$$



`dx = np.dot(prev_grad, w.T)`

`dw = np.dot(x.T, prev_grad)`

Thanks for your Attention!

Boris Zubarev



@bobazooba