Градиентный спуск
Нейронные сети

Теория игр, 2023

Содержание

Содержание

Используемая литература

- Christopher Bishop. Pattern Recognition and Machine Learning
- Shai Shalev-Shwartz, Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms
- Jorge Nocedal, Stephen J. Wright. Numerical Optimization

Машинное обучение с учителем

Множество объектов $\mathcal{X} = \{\mathbf{x}\}$, представленных вектором из d признаков $\mathbf{x} = (x_1, \ldots, x_d)$
Множество значений зависимой переменой $\mathcal{Y} = \{y\}$
Множество моделей(гипотез) $\mathcal{H}$, описываемых в виде функции, которая каждому объекту ставит в соответствие одно значений зависимой переменой $h : \mathcal{X} \to \mathcal{Y}$.
Функция потерь $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$

Истинная ошибка

Множество объектов $\mathcal{X} = \{\mathbf{x}\}$, представленных вектором из d признаков $\mathbf{x} = (x_1, \ldots, x_d)$
Множество значений зависимой переменой $\mathcal{Y} = \{y\}$
Множество моделей(гипотез) $\mathcal{H}$, описываемых в виде функции, которая каждому объекту ставит в соответствие одно значений зависимой переменой $h : \mathcal{X} \to \mathcal{Y}$.
Функция потерь $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$
При заданном вероятностном распределении $\mathcal{D}$ над $\mathcal{X} \times \mathcal{Y}$ функция ℓ является случайной

Истинная ошибка модели $h \in \mathcal{H}$

математическое ожидание функции потери модели h:

$$L_D(h) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(h(\mathbf{x}), y)]$$

Эмпирическая ошибка

Множество объектов $\mathcal{X} = \{x\}$, представленных вектором из d признаков $x = (x_1, \ldots, x_d)$
Множество значений зависимой переменой $\mathcal{Y} = \{y\}$
Множество моделей(гипотез) $\mathcal{H}$, описываемых в виде функции, которая каждому объекту ставит в соответствие одно значений зависимой переменой $h : \mathcal{X} \to \mathcal{Y}$.
Функция потерь $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$
Так как в большинстве случаев распределение $\mathcal{D}$ неизвестно, то модель выбирается по имеющейся выборке из m объектов

$$S = ((\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)}))$$

Эмпирическая ошибка модели $h \in \mathcal{H}$

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h(\mathbf{x}^{(i)}), y^{(i)}) \tag{1}$$

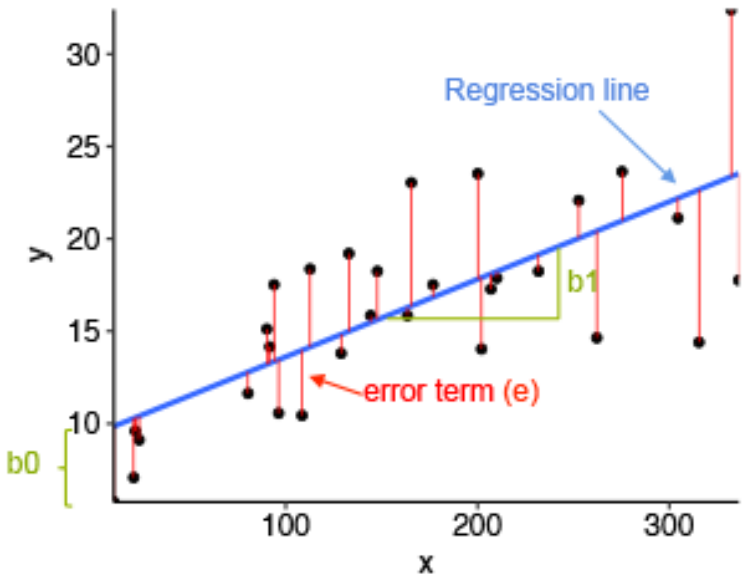
Задача регрессии

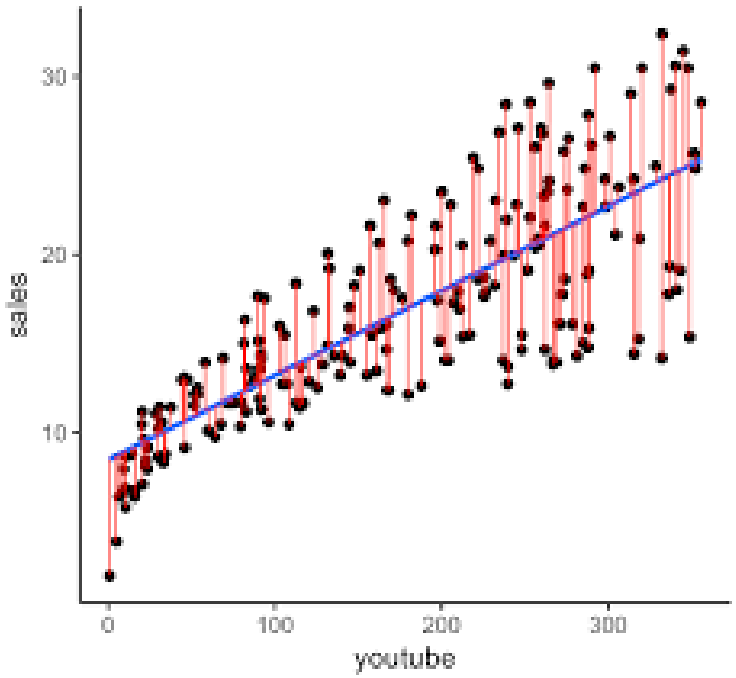
Множество объектов $\mathcal{X} = \{\mathbf{x}\}$, представленных вектором из d признаков $\mathbf{x} = (x_1, \ldots, x_d)$
Множество значений зависимой переменой $\mathcal{Y} = \mathbb{R}$
Множество моделей(гипотез) $\mathcal{H}$, описываемых в виде функции, которая каждому объекту ставит в соответствие одно значений зависимой переменой $h : \mathcal{X} \to \mathcal{Y}$.
Функция потерь $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$
$h, \mathbf{x}, y \mapsto (h(\mathbf{x}) - y)^2$

Пример задачи регрессии

Объект задается набором (вектором) d признаков $\mathbf{x} = (x_1, ..., x_d)$
Модель - линейная функция от вектора параметров $\mathbf{w}$

$$h_{\mathbf{w}}(\mathbf{x}) = h(\mathbf{x}, \mathbf{w}) = w_0 + x_1 * w_1 + ... + x_d * w_d = \langle \mathbf{x}, \mathbf{w} \rangle$$





Задача классификации

Множество объектов $\mathcal{X} = \{x\}$, представленных вектором из d признаков $\mathrm{x} = (x_1, \ldots, x_d)$
Множество значений зависимой переменой $\mathcal{Y} = y_1, \ldots, y_n$
Множество моделей(гипотез) $\mathcal{H}$, описываемых в виде функции, которая каждому объекту ставит в соответствие одно значений зависимой переменой $h : \mathcal{X} \to \mathcal{Y}$.
Функция потерь $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$

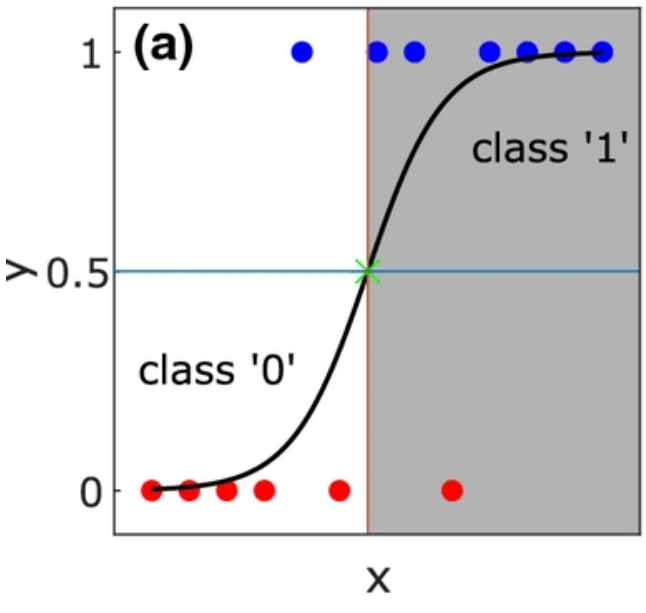
$$\ell(h(\mathbf{x}), y) \mapsto \begin{cases} 1, & \text{если } h(\mathbf{x}) \neq y \\ 0, & \text{иначе} \end{cases}$$

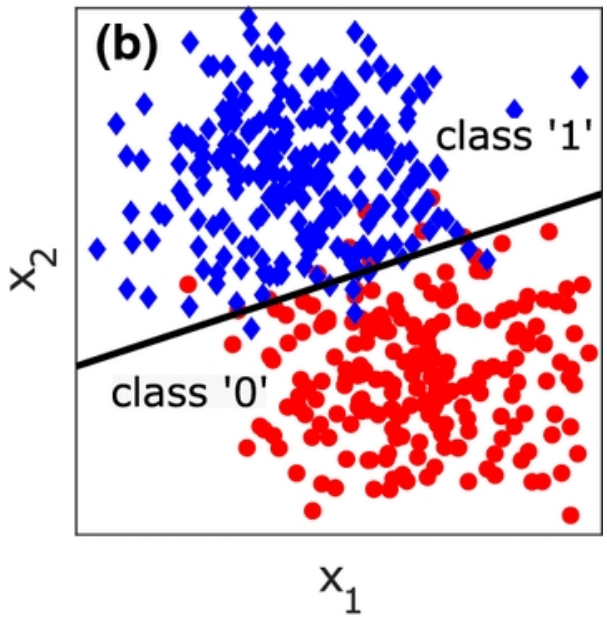
Пример задачи классификации

Объект задается набором (вектором) d признаков $\mathbf{x} = (x_1, ..., x_d)$
Множество значений зависимой переменой - два класса $\mathcal{Y} = \{0, 1\}$
Модель - логистическая функция от вектора параметров $\mathbf{w}$

$$h_{\mathbf{w}}(\mathbf{x}) = h(\mathbf{x}, \mathbf{w}) = \frac{1}{1 + e^{\langle \mathbf{x}, \mathbf{w} \rangle}}, \langle \mathbf{x}, \mathbf{w} \rangle = w_0 + x_1 * w_1 + ... + x_d * w_d$$

$$y^{\text{гипотеза}} = \begin{cases} 1, & \text{если } h_{\mathbf{w}}(\mathbf{x}) > 0.5 \\ 0, & \text{иначе} \end{cases}$$





Содержание

Задача оптимизации

Минимизация эмпирической ошибки модели $h \in \mathcal{H}$

$$L_S(h_{\mathbf{w}}) = L_S(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} \ell(h(\mathbf{x}^{(i)}, \mathbf{w}), y^{(i)}) \to \min_{\mathbf{w}} \tag{2}$$

Общая постановка задачи оптимизации для произвольной функции $f : \mathbb{R}^n \to \mathbb{R}$

$$f(\mathbf{w}) \to \min_{\mathbf{w}} \tag{3}$$

$\mathbf{w}*$ - минимум функции f, если $\forall \mathbf{w}\ f(\mathbf{w}^*) \leq f(\mathbf{w})$

Градиент

$f : \mathbb{R}^n \to \mathbb{R}$

$\mathbf{w} \mapsto f(\mathbf{w})$

Градиент функции f в точке p

$\nabla f : \mathbb{R}^n \to \mathbb{R}^n$

$$\nabla f_{\mathbf{w}}(p) = \begin{bmatrix} \frac{\partial f}{\partial w_1}(p) \\ \frac{\partial f}{\partial w_2}(p) \\ \vdots \\ \frac{\partial f}{\partial w_n}(p) \end{bmatrix}$$

Градиентный спуск

Нахождение локального минимуму функции $f : \mathbb{R}^n \to \mathbb{R}$

Алгоритм

Инициализация $\mathbf{w}^0$ произвольно
Для каждой итерации
выбор шага обновления α

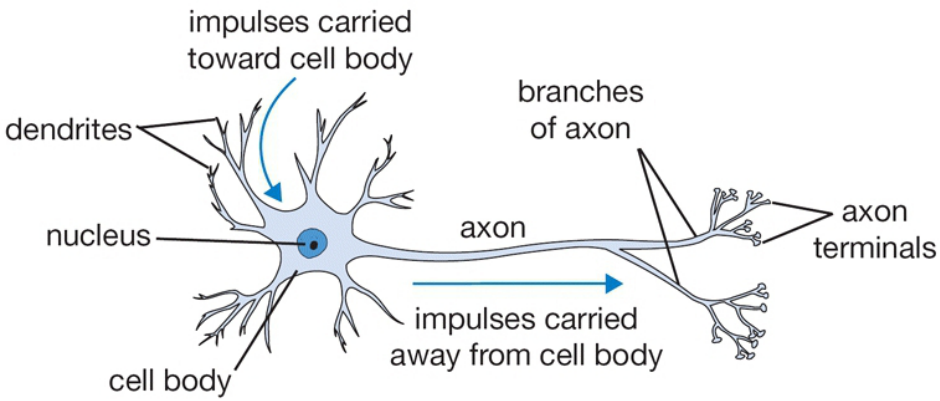
$$\mathbf{w}^{k+1} \leftarrow \mathbf{w}_k - \alpha \nabla f(\mathbf{w}_k)$$

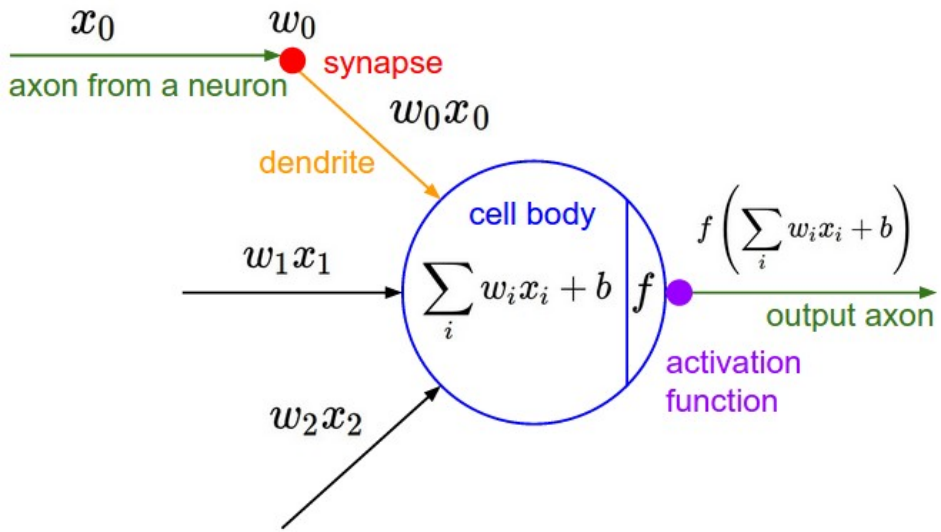
Для минимума w^* верно $\nabla f(\mathbf{w}^*) = 0$

Содержание

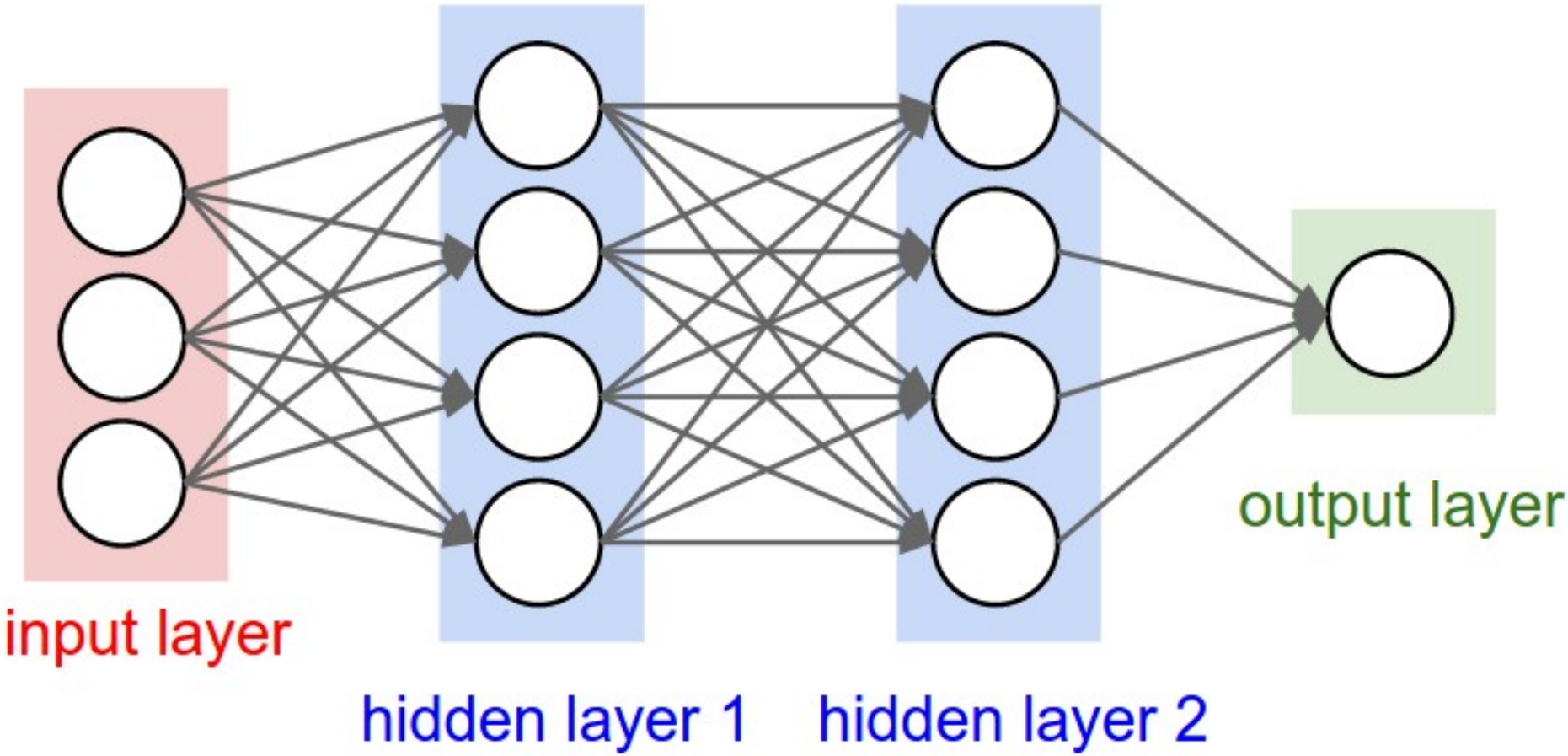
Нейрон

https://cs231n.github.io





Нейронная сеть

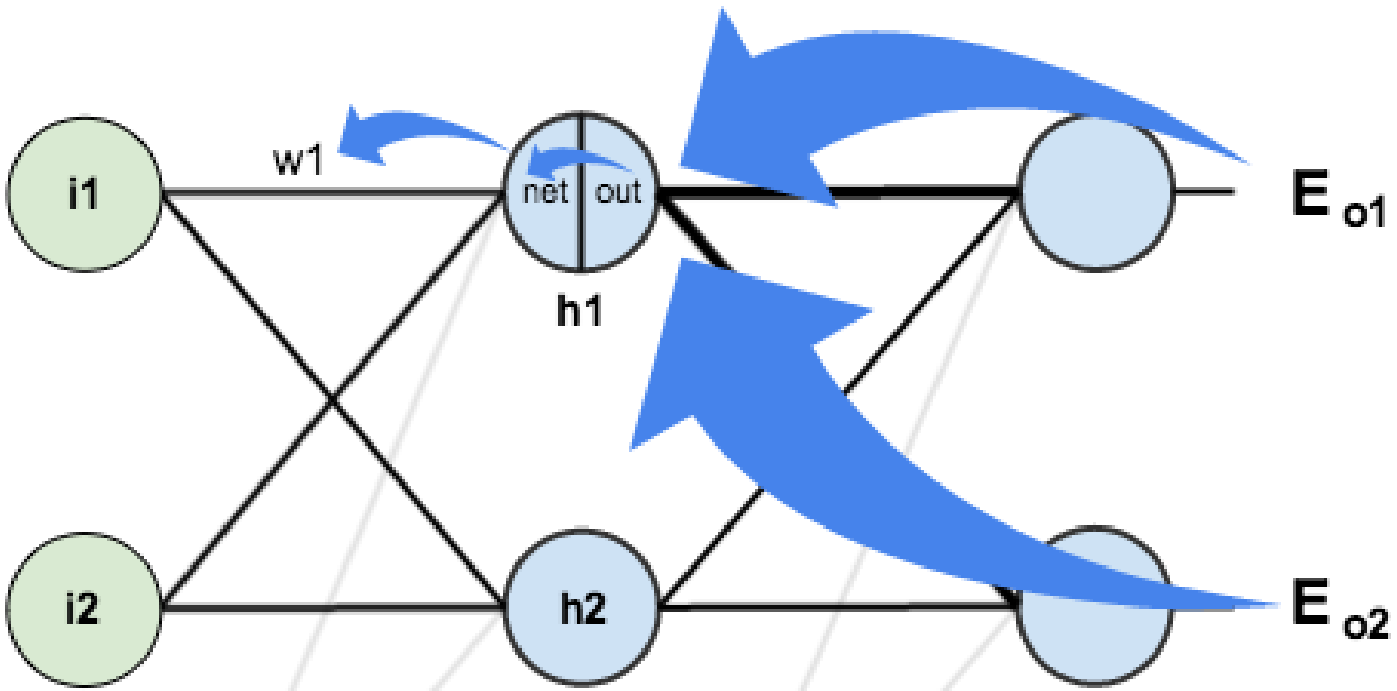


Алгоритм обратного распространения

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\downarrow$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$$E_{total} = E_{o1} + E_{o2}$$