

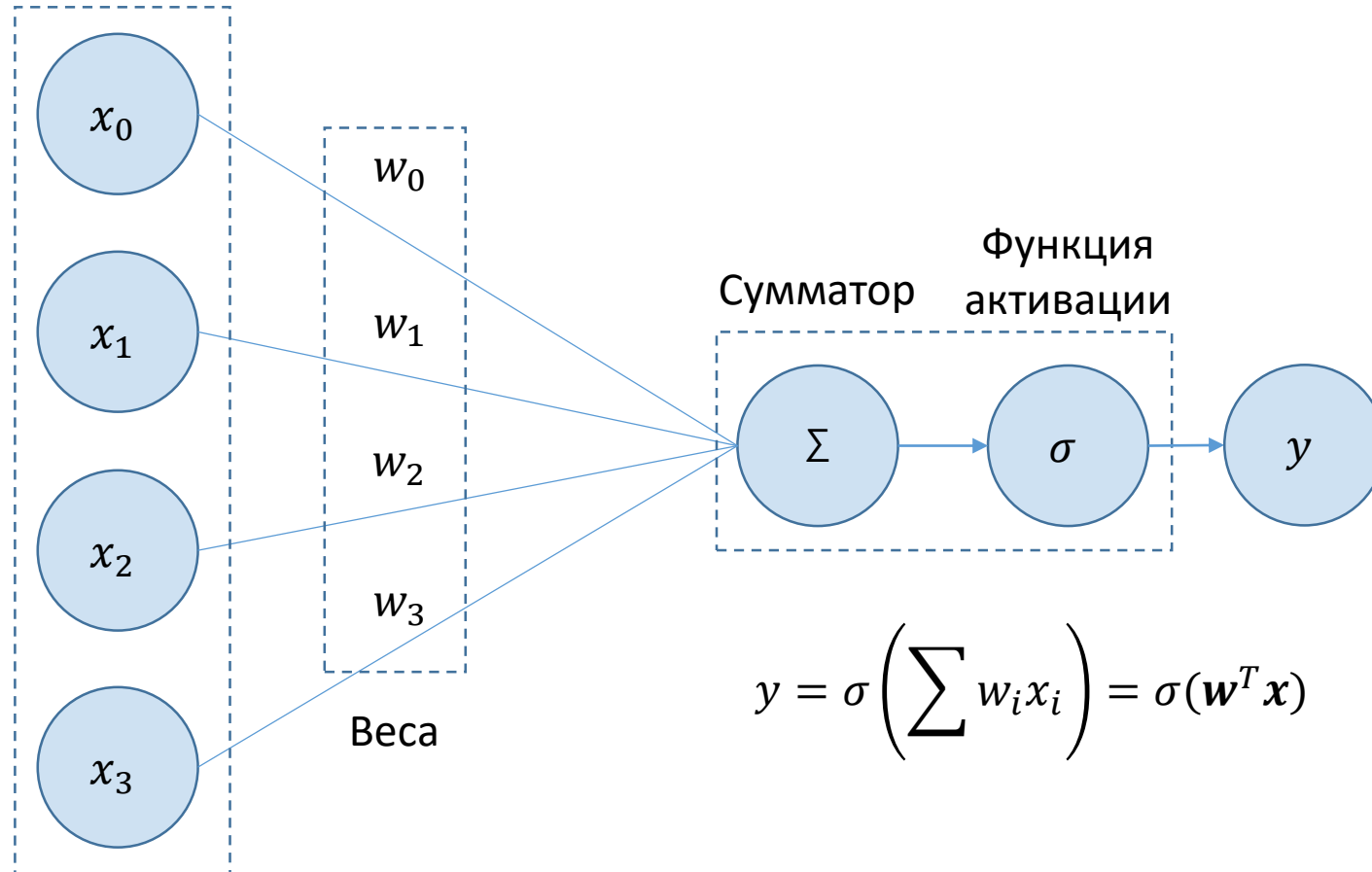
Курс “Анализ изображений”

Лекция#10.

Нейронные сети. Слои. Алгоритм  
обратного распространения ошибки.  
Регуляризация.

# Модель нейрона Маккаллока-Питтса

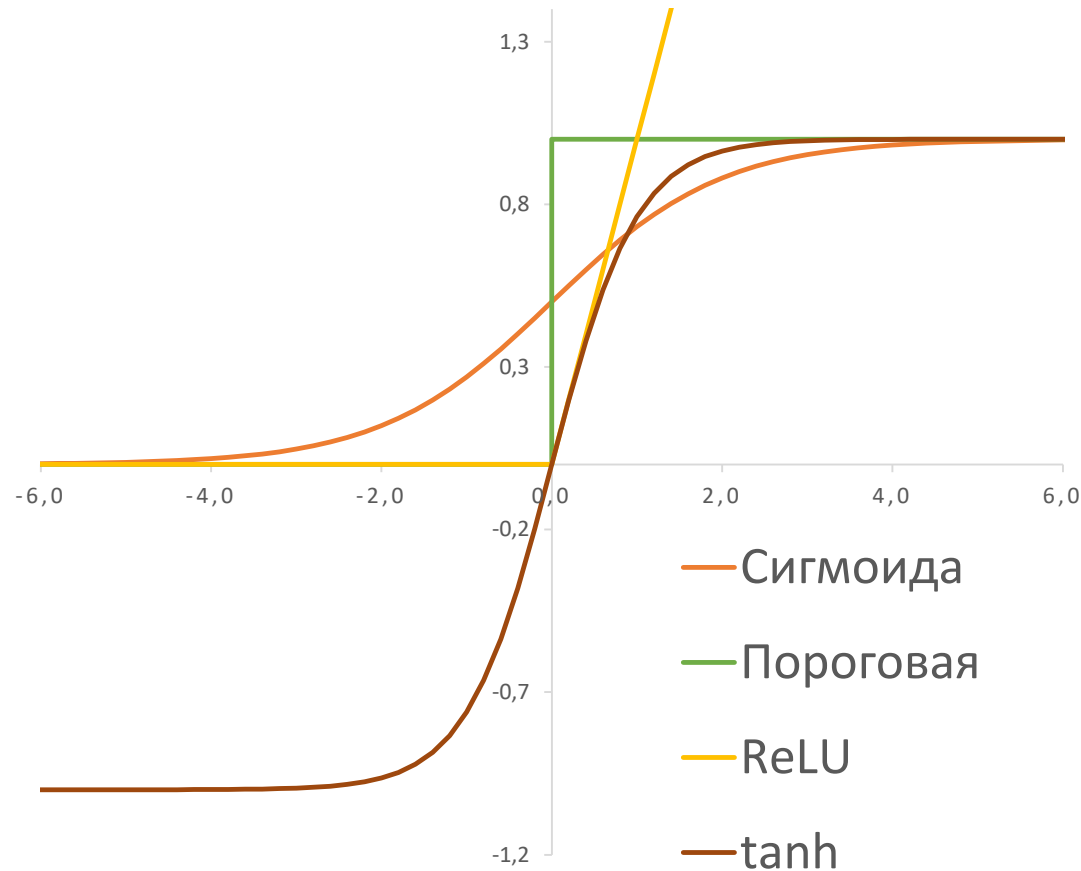
Входной сигнал



# Функции активации

- Линейная:  $\sigma(x) = x$
- Пороговая:  $\sigma(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$
- Логистическая сигмоида:  $\sigma(x) = \frac{1}{1+e^{-x}}$
- Гиперболический тангенс:  $\sigma(x) = \tanh(x) = \frac{e^{2x}-1}{e^{2x}+1} = 2\sigma(2x) - 1$
- ReLU (Rectified Linear Unit):  $\sigma(x) = \max(0, x)$

# Функции активации



$$y = \sigma(\alpha \mathbf{w}'^T \mathbf{x} + b), \|\mathbf{w}'\| = 1$$

- Множитель  $\alpha$  отвечает за растяжение по оси  $X$
- сдвиг  $b$  отвечает за сдвиг порога

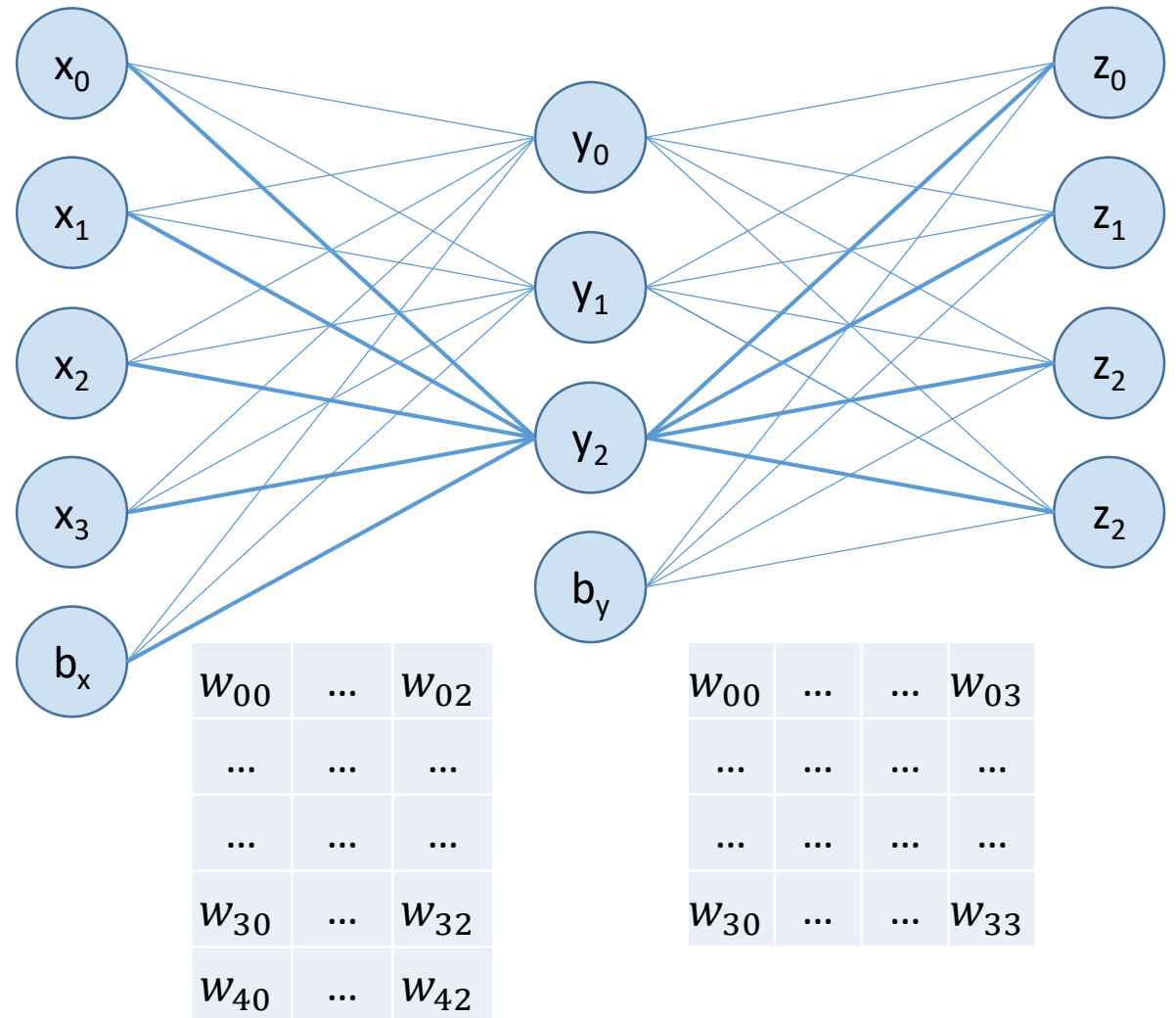
# Softmax

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^J e^{z_j}}$$

- Используется как выход классификатора для представления распределения вероятностей
- Реже – внутри модели, для выбора одного из вариантов внутренней переменной
- Компонента соответствует вероятности класса:
$$P(y = i|x) = \sigma(\mathbf{w}^T \mathbf{x})_i$$
- Сумма компонент вектора равна 1
- $\sigma(\mathbf{z} + c) = \sigma(\mathbf{z})$

# Нейронная сеть. Полносвязный слой

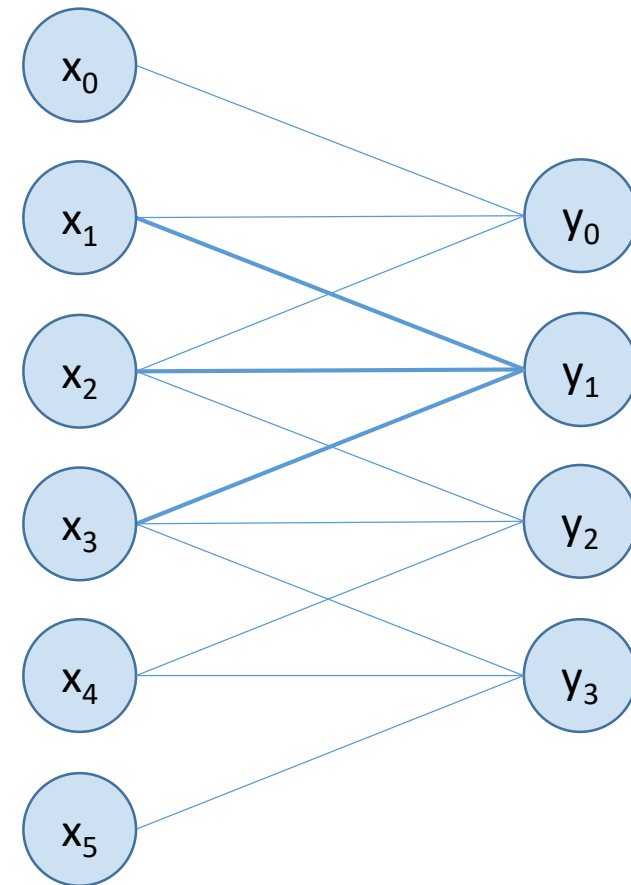
- Все нейроны предыдущего слоя связаны со всеми нейронами следующего
- Внутреннее представление слоя – матрица весов  
 $y_i = \sigma(\mathbf{w}^T \mathbf{x}) \rightarrow \mathbf{y} = \sigma(\mathbf{W}^T \mathbf{x})$
- Реализация слоя – перемножение матриц



# Локально связный слой

- Между собой связаны только соседние признаки (=пиксели)

$w_{00}$	0	0	0
$w_{10}$	$w_{11}$	0	0
$w_{20}$	$w_{21}$	$w_{22}$	0
0	$w_{31}$	$w_{32}$	$w_{33}$
0	0	$w_{42}$	$w_{43}$
0	0	0	$w_{53}$

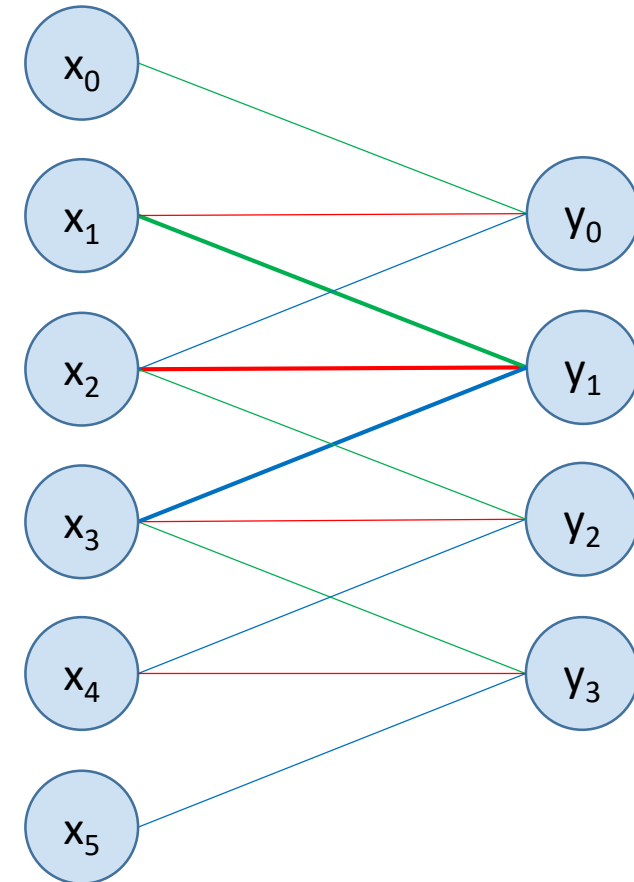
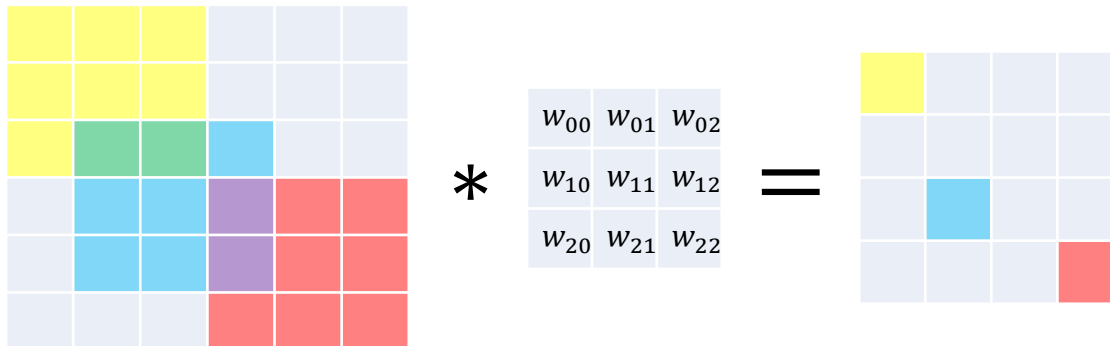


# Сверточный слой

- Реализует свертку изображения (или сигнала)  $x$  с ядром  $w$ :

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t - a)$$

- Инвариантен к параллельному переносу – искомый признак может находиться где угодно на изображении
- По сути – локально-связный слой с общими весами





# Pooling

- Реализует операцию локального прореживания
- В большинстве случаев используется усреднение или максимум по небольшому окну (2x2 или 3x3)

1	2	7	2	5	2
9	5	4	5	5	4
1	2	3	1	4	6
1	2	8	9	8	3
8	3	3	7	4	6
3	7	2	5	6	8

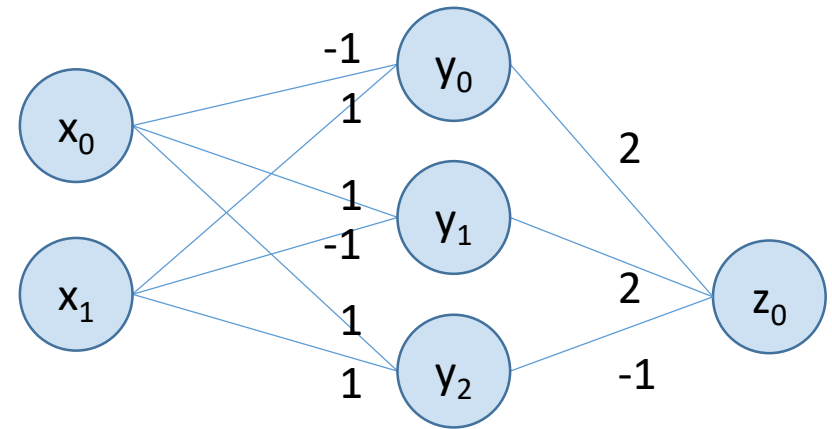
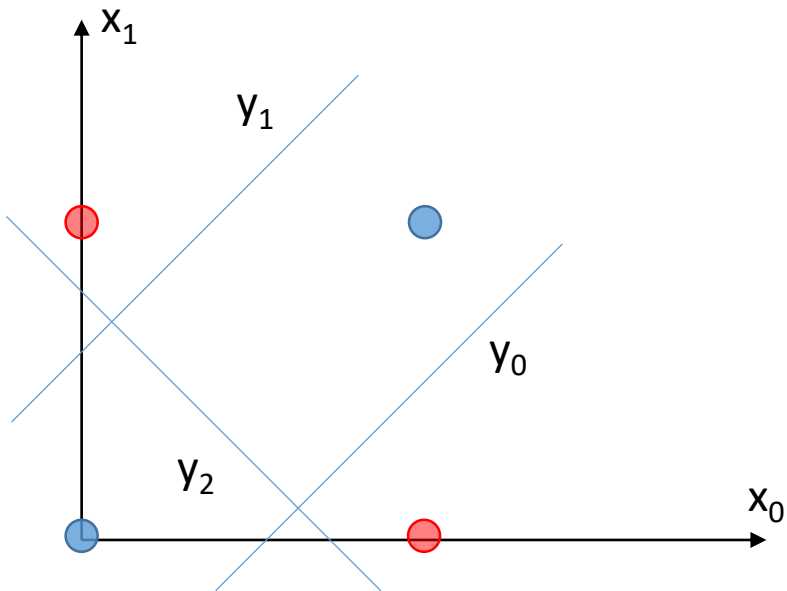


9	7	5
2	9	8
8	7	8

# Задача XOR

1	0
0	1

- Задача неразрешима с использованием только одного слоя
- Легко разрешается 2-мя слоями с нелинейными функциями активации (например, пороговой)



$x_0$	$x_1$	$y_0$	$y_1$	$y_2$	$z_0$
0	0	0	0	0	0
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	1	0

# Обучение нейронной сети

Обозначения:

- $E$  – функция потерь
- $l$  –  $l$ -ый слой сети
- $w_{ij}^l$  – веса  $l$ -го слоя
- $\sigma$  – функция активации
- $o_j^l = \sigma(x_j^l)$  – выход  $l$ -го слоя
- $x_j^l$  – преобразованный входной вектор

Хотим устроить градиентный спуск:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

$\eta$  – скорость обучения.

Для этого нужно посчитать градиенты  $\frac{\partial E}{\partial w_{ij}}$  для всех слоев

# Алгоритм обратного распространения ошибки

- Ответ слоя:

$$x_j^l = \sum_i w_{ij}^l o_i^{l-1} = f(w_{ij}^l)$$

- Функция потерь:

$$E = E(o_j, y_j) = E(\sigma(f(w_{ij}^l)), y_j)$$

- Градиент для последнего слоя:

$$\frac{\partial E}{\partial w_{ij}^l} = \frac{\partial E}{\partial o_j^l} \frac{\partial o_j^l}{\partial x_j^l} \frac{\partial x_j^l}{\partial w_{ij}^l}$$

$= \delta_j^l = \sigma'(x_j^l) = o_i^{l-1}$

- На внутренних слоях – считаем относительно предыдущего слоя:

$$\frac{\partial E}{\partial w_{ij}^l} = \sum_k \frac{\partial E}{\partial x_k^{l+1}} \frac{\partial x_k^{l+1}}{\partial x_j^l} \frac{\partial x_j^l}{\partial w_{ij}^l} =$$

$$= \frac{\partial x_j^l}{\partial w_{ij}^l} \sum_k \frac{\partial E}{\partial x_k^{l+1}} \frac{\partial x_k^{l+1}}{\partial o_j^l} \frac{\partial o_j^l}{\partial x_j^l}$$

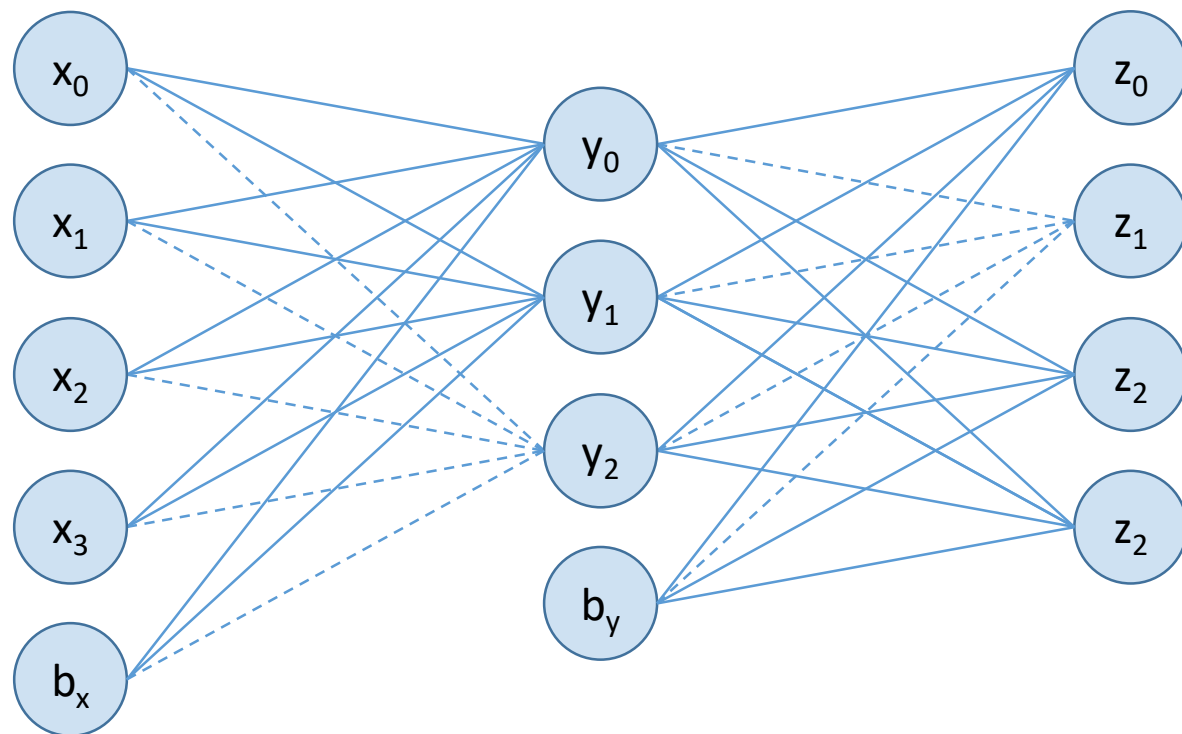
$= o_i^{l-1} \quad = \delta_k^{l+1} \quad = w_{jk}^{l+1} \quad = \sigma'(x_j^l)$

# Регуляризация

- Цель – уменьшение ошибки обобщения, не уменьшая при этом ошибку обучения
- Штраф по норме:
  - Задача оптимизации:  $\min \sum_{i=1}^N L(y_i, y'_i) + \lambda R(w)$
  - $L_2$  (ridge):  $R(w) = \sum_i w_i^2$
  - $L_1$  (lasso):  $R(w) = \sum_i |w_i|$  – реализует также отбор признаков
- Раздутие обучающей выборки
- Многозадачное обучение
- Ранняя остановка обучения
- Dropout

# Dropout

- Идея – в процессе обучения удалять из
- Реализация – умножение на 0 строк матрицы приращений
- Аппроксимирует ансамбль экспоненциального количества нейронных сетей с общими параметрами



# Автоэнкодер

- Задача – понизить размерность входного вектора
- Задача обучения – “скопировать входной вектор”, при этом пройдя через узкое горлышко
- Если  $L$  – среднеквадратичная ошибка, а декодер – линейная функция, то обучение эквивалентно PCA

