Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского
Институт информационных технологий, математики и механики

Образовательный курс
«Введение в глубокое обучение с использованием Intel® neon™ Framework»

Многослойные полностью связанные сети

При поддержке компании Intel

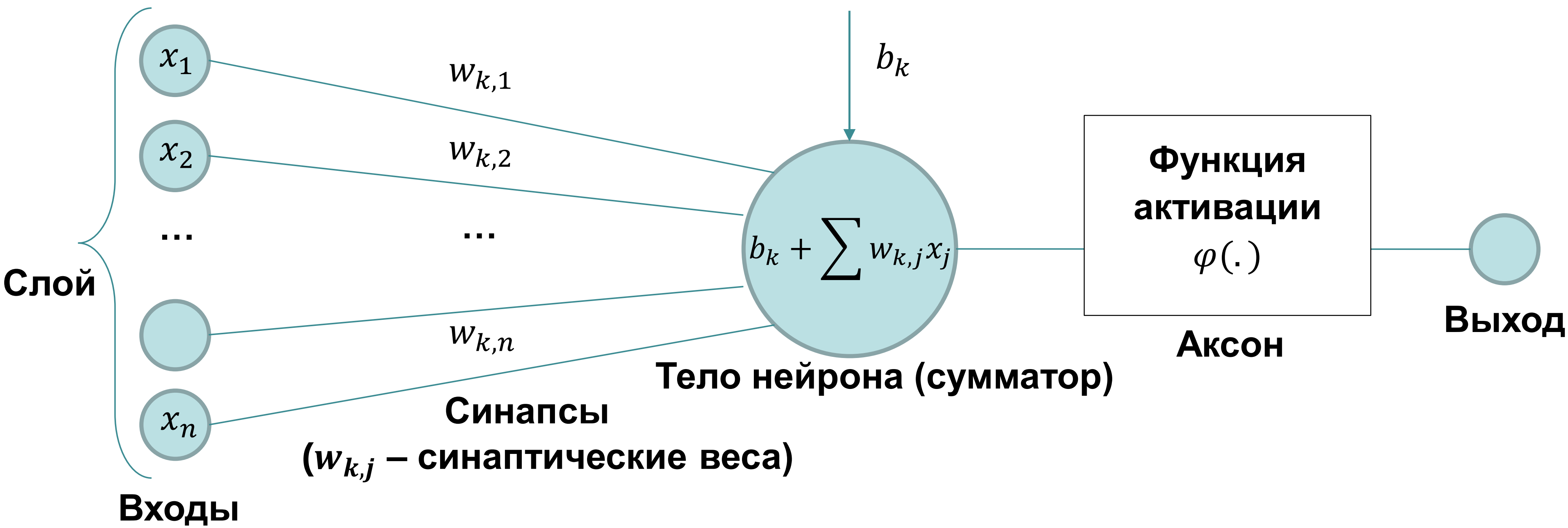
Кустикова Валентина,
к.т.н., ст.преп. каф. МОСТ ИИТММ,
ННГУ им. Н.И. Лобачевского

Содержание

- Детерминистская модель нейрона. Функции активации
- Общая структура модели многослойной полностью связанной нейронной сети
- Оптимизационная постановка задачи обучения многослойной полносвязной нейронной сети. Функция ошибки
- Метод обратного распространения ошибки
- Последовательный и пакетный режимы обучения
- Эвристические рекомендации по повышению производительности метода обратного распространения ошибки

ДЕТЕРМИНИСТСКАЯ МОДЕЛЬ НЕЙРОНА. ФУНКЦИИ АКТИВАЦИИ

Детерминистская модель нейрона (1)



Детерминистская модель нейрона (2)

- Модель нейрона состоит из трех основных компонент:
  – ***Синапсы*** – входные сигналы, каждый из которых характеризуется собственным ***весом***
  – ***Сумматор*** – компонент, который складывает входные сигналы с учетом установленных синаптических весов
  – ***Функция активации*** – компонент, который ограничивает амплитуду выходного сигнала, в результате чего выход нейрона, как правило, лежит в отрезке $[0,1]$ или $[-1,1]$

Детерминистская модель нейрона (3)

- Математическая модель нейрона:

$$u_k = \sum_{j=1}^{n} w_{k,j} x_j\,, \qquad y_k = \varphi(u_k + b_k) \qquad (1)$$

- В предположении, что $v_k = u_k + b_k$, указанная пара уравнений может быть записана следующим образом:

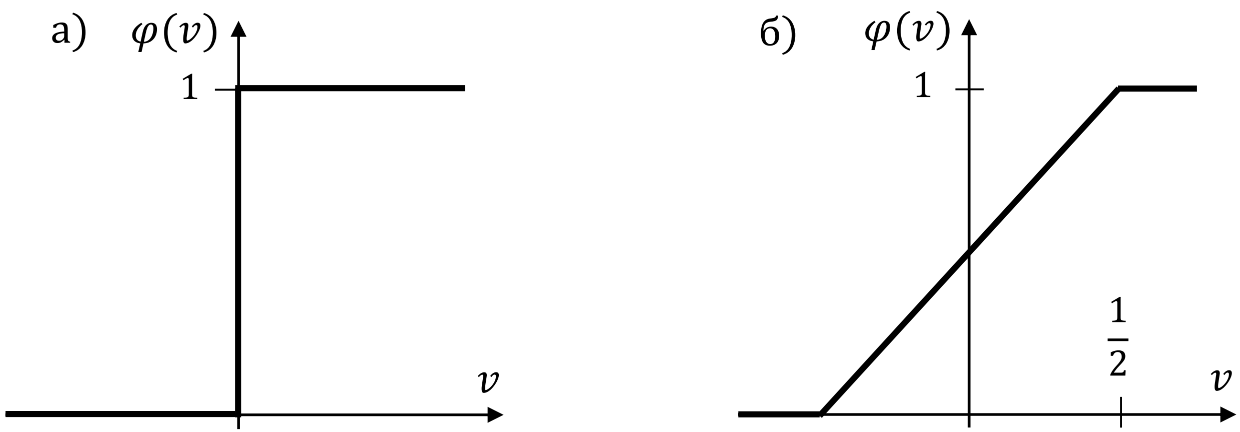
$$v_k = \sum_{j=0}^{n} w_{k,j} x_j\,, \qquad y_k = \varphi(v_k), \qquad (2)$$

где $x_0 = 1$ – новый синапс, а $w_{k,0} = b_k$ – его вес

- (1) и (2) – эквивалентные модели нейрона

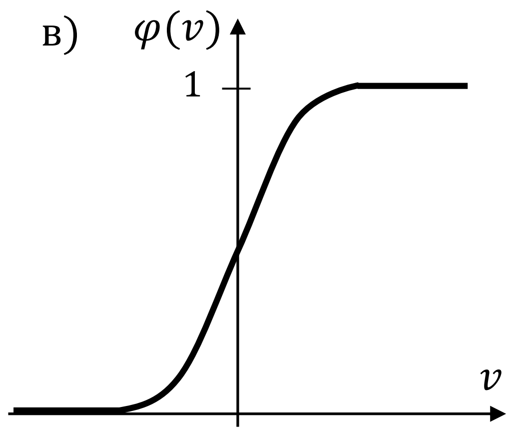
Функции активации (1)

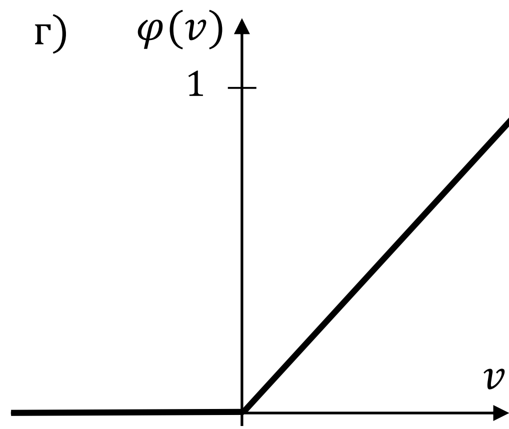
- ***Пороговая функция*** (а). Описывает состояние «все или ничего». Применяется в задачах, требующих бинарного ответа
- ***Кусочно-линейная функция*** (б). Может рассматриваться как аппроксимация нелинейного усилителя



Функции активации (2)

- ***Сигмоидальная функция*** (в). Примерами таких функций могут служить логистическая функция и гиперболический тангенс
- ***Положительная «срезка»*** (Rectified Linear Unit, ReLU, г)

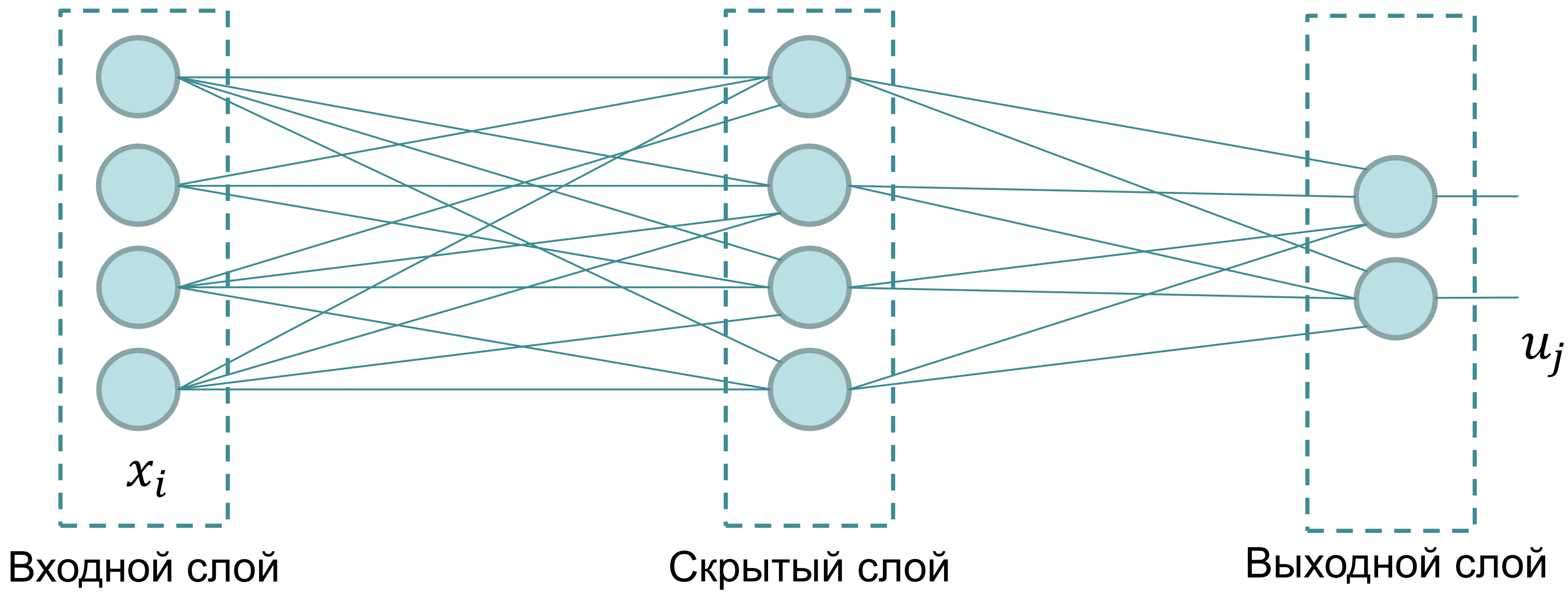




СТРУКТУРА МНОГОСЛОЙНОЙ ПОЛНОСВЯЗНОЙ СЕТИ

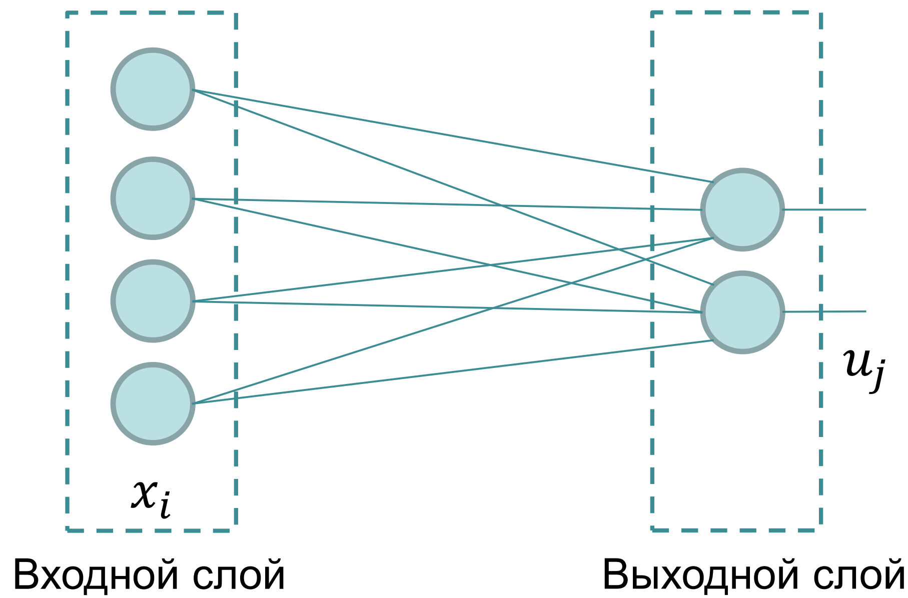
Общая структура модели многослойной полностью связанной нейронной сети (1)

- Многослойная нейронная сеть содержит нейроны, которые распределены по ***слоям***



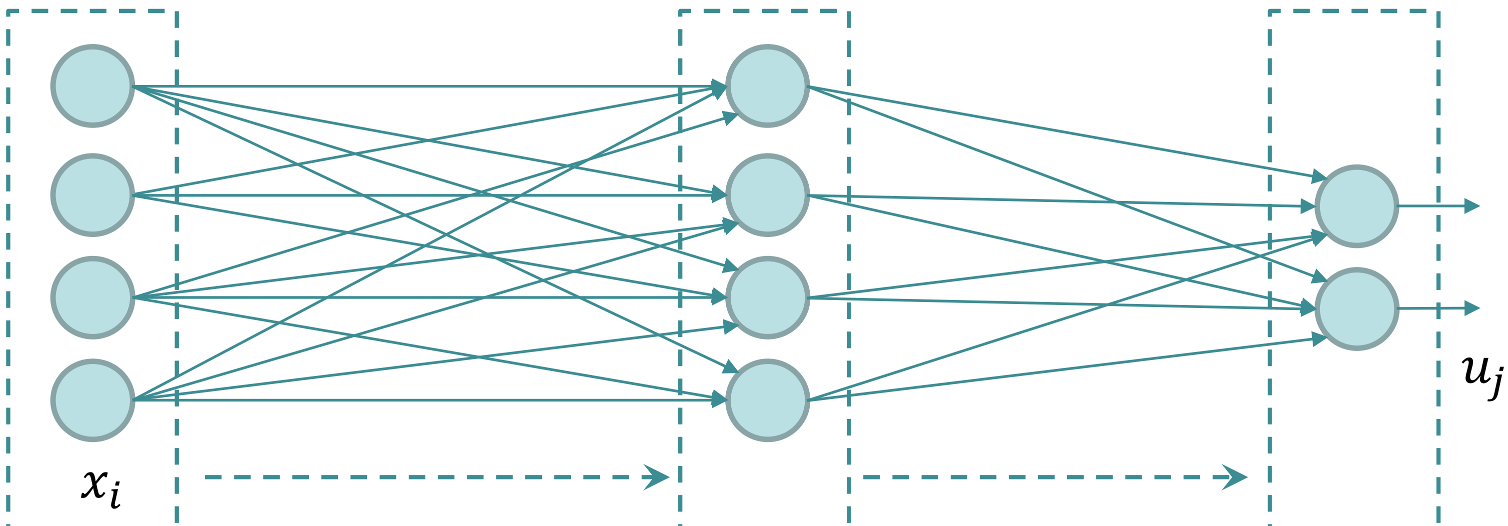
Общая структура модели многослойной полностью связанной нейронной сети (2)

- В простейшем случае в сети существует ***входной и выходной слои***, и сеть называется ***однослойной***



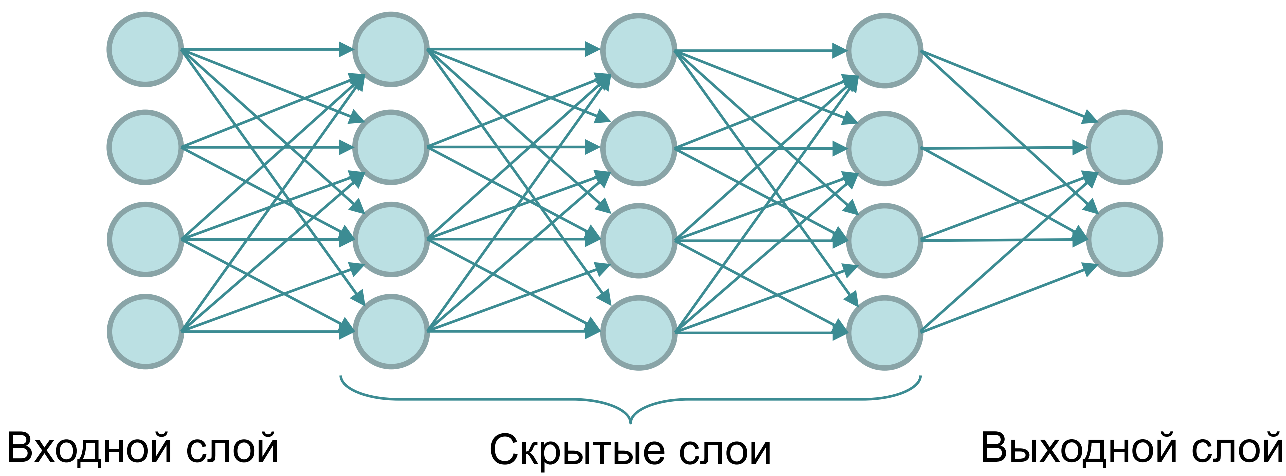
Общая структура модели многослойной полностью связанной нейронной сети (3)

- Если сигнал проходит от нейронов входного слоя к нейронам выходного, то такая сеть называется ***сетью прямого распространения***



Общая структура модели многослойной полностью связанной нейронной сети (4)

- Сеть может содержать множество скрытых слоев называется ***многослойной***
- Если все узлы слоя соединены с узлами следующего слоя, то слой называется ***полностью связным***
- Если данное условие выполнено для всех слоев сети, то сеть называется ***полностью связной*** или ***полносвязной*** (Fully-Connected Neural Network, FCNN)



ЗАДАЧА ОБУЧЕНИЯ НЕЙРОННОЙ ПОЛНОСВЯЗНОЙ СЕТИ

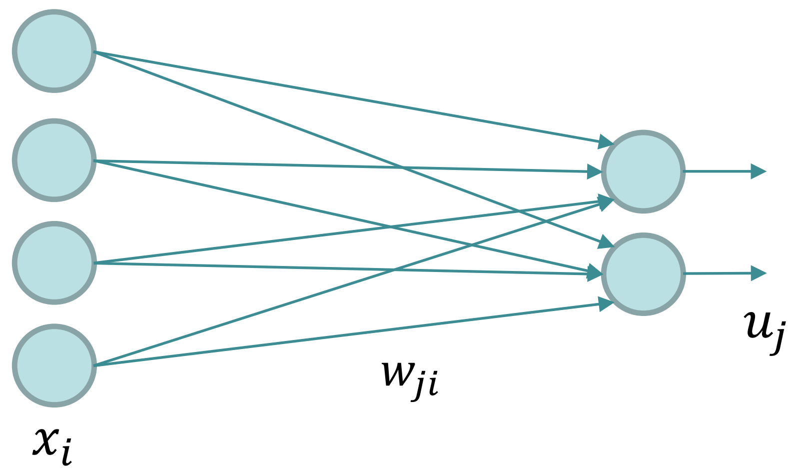
Задача обучения нейронной сети

- ***Цель обучения*** – подобрать веса

- ***Задача обучения*** – задача минимизации ***функции ошибки*** или ***функция стоимости*** (cost function), отражающей разницу ожидаемого сигнала, полученного на выходе сети и фактического сигнала, соответствующего текущему входу, по всей обучающей выборке

Квадратичная функция ошибки для однослойной нейронной сети (1)

- Рассмотрим однослойную нейронную сеть, содержащую N входных и M выходных нейронов



- Обучающая выборка $\langle X, Y \rangle$:
  - L – число примеров в выборке,
  - X – множество входных сигналов (векторов размерности N),
  - Y – множество векторов фактических выходных сигналов (векторов размерности M)

Квадратичная функция ошибки для однослойной нейронной сети (2)

- Квадратичная функции ошибки (используется для задачи восстановления регрессии):

$$E = \frac{1}{2}\sum_{k=1}^{L}\left\|y^k - u^k\right\|^2 = \frac{1}{2}\sum_{k=1}^{L}\sum_{j=1}^{M}\left(y_j^k - u_j^k\right)^2$$

$$= \frac{1}{2}\sum_{k=1}^{L}\sum_{j=1}^{M}\left(y_j^k - \varphi\left(\sum_{i=0}^{N} w_{ji}x_i^k\right)\right)^2,$$

где $y^k = \left(y_j^k\right)_{j=\overline{1,M}} \in Y$ – множество обучающих примеров, а $u^k = \left(u_j^k\right)_{j=\overline{1,M}}$ – выход сети для входа $x^k = \left(x_i^k\right)_{i=\overline{1,N}} \in X$

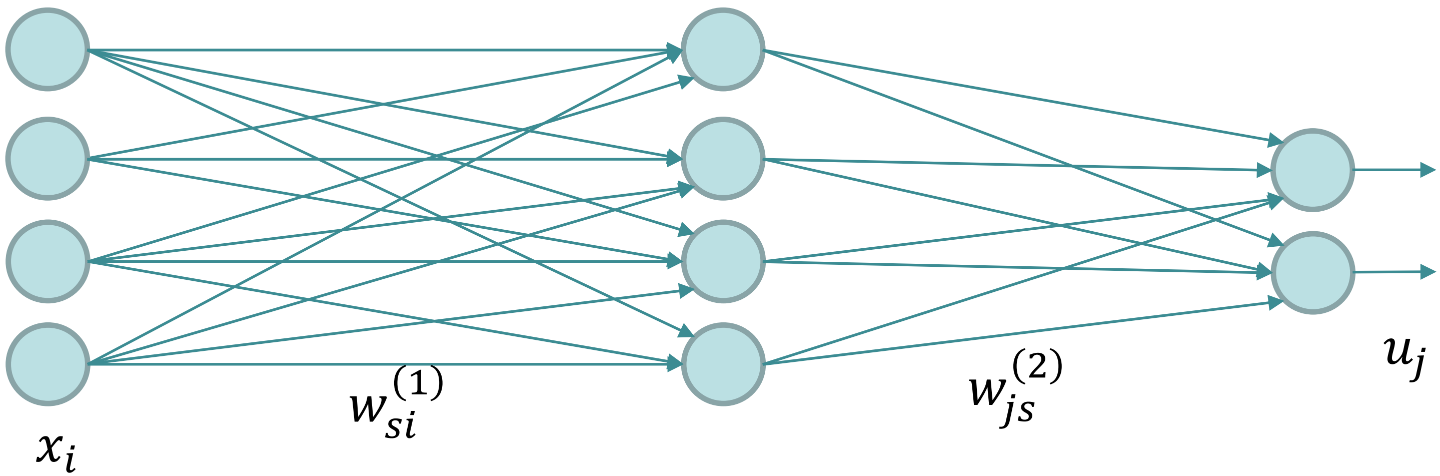
- Также вводится нормализация указанной метрики по числу примеров обучающей выборки L

Квадратичная функция ошибки для однослойной нейронной сети (3)

- Что изменится при переходе от однослойной полносвязной сети к двухслойной?

Квадратичная функция ошибки для двухслойной нейронной сети (1)

- Рассмотрим двухслойную нейронную сеть:

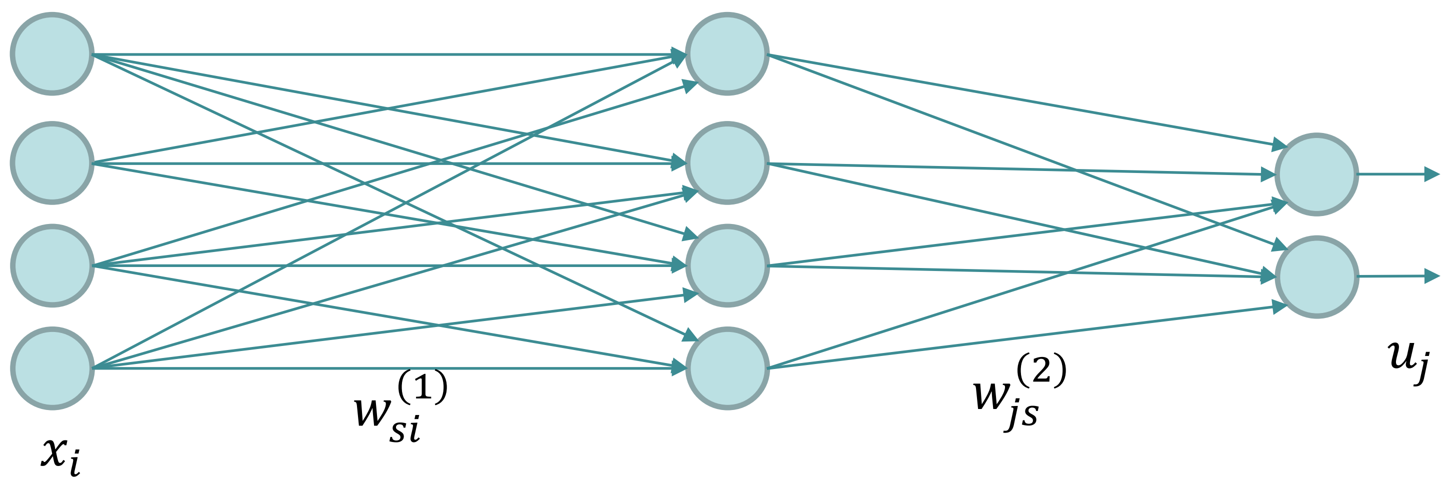


- $w_{si}^{(1)}, w_{js}^{(2)}$ – веса синаптических связей
- Выходной сигнал нейрона скрытого слоя описывается следующим образом:

$v_s = \varphi^{(1)}\left(\sum_{i=0}^{N} w_{si}^{(1)} x_i\right), s = \overline{0, K}$, где K – количество нейронов на скрытом слое

Квадратичная функция ошибки для двухслойной нейронной сети (2)

- Рассмотрим двухслойную нейронную сеть:



- Сигнал j-ого нейрона выходного слоя:

$$u_j = \varphi^{(2)}\left(\sum_{s=0}^{K} w_{js}^{(2)} v_s\right) = \varphi^{(2)}\left(\sum_{s=0}^{K} w_{js}^{(2)} \varphi^{(1)}\left(\sum_{i=0}^{N} w_{si}^{(1)} x_i\right)\right), j = \overline{1, M}.$$

Квадратичная функция ошибки для двухслойной нейронной сети (3)

- Квадратичная функция ошибки для набора тренировочных примеров имеет вид:

$$E(w) = \frac{1}{2}\sum_{k=1}^{L}\sum_{j=1}^{M}\left(y_j^k - u_j^k\right)^2$$

$$= \frac{1}{2}\sum_{k=1}^{L}\sum_{j=1}^{M}\left(y_j^k - \varphi^{(2)}\left(\sum_{s=0}^{K} w_{js}^{(2)}\varphi^{(1)}\left(\sum_{i=0}^{N} w_{si}^{(1)}x_i^k\right)\right)\right)^2$$

Оптимизационная постановка задачи обучения с квадратичной функцией ошибки

- Общая математическая постановка задачи обучения сети с квадратичной функцией ошибки:

$$\min_{w} E(w) = \min_{w} \left\{ \frac{1}{L} \sum_{k=1}^{L} \left\{ \frac{1}{2} \sum_{j=1}^{M} \left(y_j^k - u_j^k \right)^2 \right\} \right\}$$

- Квадратичная ошибка отражает разницу выхода сети разметки
- Квадратичная (Евклидова) ошибка применяется при решении ***задачи восстановления регрессии***

Оптимизационная постановка задачи обучения с кросс-энтропийной функцией ошибки

- Для ***задачи классификации*** в качестве функции ошибки выбирается ***кросс-энтропия***. Постановка задачи обучения сети:

$$\min_{w} E(w) = \min_{w} \left\{ -\frac{1}{L} \sum_{k=1}^{L} \sum_{j=1}^{M} y_j^k \ln u_j^k \right\}$$

где $y_j^k = 1 \leftrightarrow x^k$ принадлежит классу j, иначе $y_j^k = 0$

- Является дифференцируемой аппроксимацией функции ошибки классификации «0-1»
- В качестве функции активации на последнем слое рекомендуется выбирать ***функцию softmax***:

$$\varphi(u_j) = \frac{e^{u_j}}{\sum_{i=1}^{M} e^{u_i}}$$

МЕТОД ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

Метод обратного распространения ошибки (1)

- Метод обратного распространения ошибки определяет стратегию изменения параметров сети w в ходе обучения с использованием градиентных методов оптимизации
- Градиентные методы на каждом шаге уточняют значения параметров:

$$w(k+1) = w(k) + \eta p(w)$$

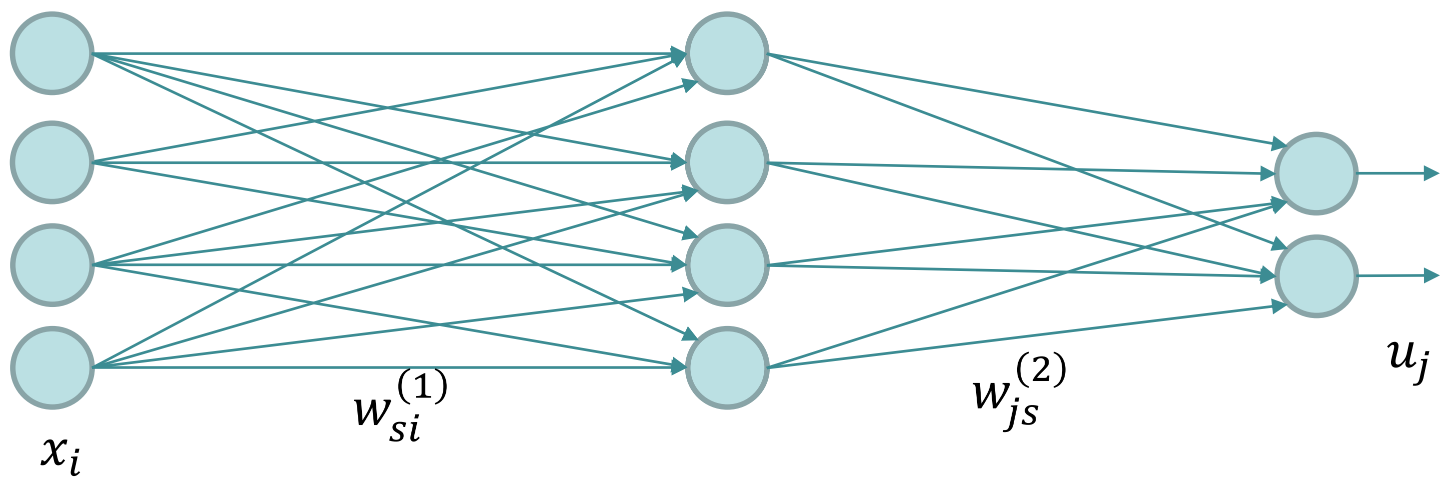
  - $\eta, 0 < \eta < 1$ – ***скорость обучения*** (learning rate) – «скорость» движения в направлении минимального значения функции,
  - $p(w)$ – направление в многомерном пространстве параметров нейронной сети
- В классическом методе обратного распространения ошибки направление движения совпадает с направлением антиградиента $p(w) = -\nabla E(w)$

Метод обратного распространения ошибки (2)

- ***Инициализация синаптических весов сети*** (случайным образом из некоторого распределения)
- Повторение следующих шагов для каждого примера тренировочного набора данных
  1. ***Прямой проход:***
     1. Вычисление значений выходных сигналов нейронов всех слоев
     2. Вычисление значений производных функций активации на каждом слое сети
  2. ***Обратный проход:***
     1. Вычисление значения целевой функции и ее градиента
     2. Корректировка синаптических весов
- ***Критерии остановки:*** число итераций метода (количество проходов по всему множеству примеров), достигнутая ошибка

Метод обратного распространения ошибки на примере двухслойной сети (1)

- Рассмотрим процедуру вычисления производных и значений функции ошибки на примере двухслойной нейронной сети



- Функция ошибки:

$$E(w) = \frac{1}{2}\sum_{j=1}^{M}\left(y_j - \varphi^{(2)}\left(\sum_{s=0}^{K} w_{js}^{(2)}\,\varphi^{(1)}\left(\sum_{i=0}^{N} w_{si}^{(1)} x_i\right)\right)\right)^2$$

Метод обратного распространения ошибки на примере двухслойной сети (2)

- Производная целевой функции по параметрам последнего слоя:

$$\frac{\partial E}{\partial w_{js}^{(2)}} = \frac{\partial \left(\frac{1}{2} \sum_{j=1}^{M} \left(y_j - \underbrace{\varphi^{(2)} \left(\sum_{s=0}^{K} \boldsymbol{w}_{\boldsymbol{js}}^{(\boldsymbol{2})} \varphi^{(1)} \left(\sum_{i=0}^{N} w_{si}^{(1)} x_i \right) \right)}_{u_j} \right)^2 \right)}{\partial \boldsymbol{w}_{\boldsymbol{js}}^{(\boldsymbol{2})}}$$

$$= -\left(y_j - u_j \right) \frac{d\varphi^{(2)}(g_j)}{d g_j} v_s = \delta_j^{(2)} v_s,$$

$$g_j = \sum_{s=0}^{K} w_{js}^{(2)} v_s, \qquad \delta_j^{(2)} = \frac{\partial E(w)}{\partial g_j}$$

Метод обратного распространения ошибки на примере двухслойной сети (2)

- Производная целевой функции по параметрам последнего слоя:

$$\frac{\partial E}{\partial w_{js}^{(2)}} = \frac{\partial\left(\frac{1}{2}\sum_{j=1}^{M}\left(y_j - \varphi^{(2)}\left(\sum_{s=0}^{K} \boldsymbol{w}_{js}^{(2)} \overbrace{\varphi^{(1)}\left(\sum_{i=0}^{N} w_{si}^{(1)} x_i\right)}^{v_s}\right)\right)^2\right)}{\partial \boldsymbol{w}_{js}^{(2)}}$$

$$= -(y_j - u_j)\frac{d\varphi^{(2)}(g_j)}{dg_j} v_s = \delta_j^{(2)} v_s,$$

$$g_j = \sum_{s=0}^{K} w_{js}^{(2)} v_s, \qquad \delta_j^{(2)} = \frac{\partial E(w)}{\partial g_j}$$

Метод обратного распространения ошибки на примере двухслойной сети (2)

- Производная целевой функции по параметрам последнего слоя:

$$\frac{\partial E}{\partial w_{js}^{(2)}} = \frac{\partial\left(\frac{1}{2}\sum_{j=1}^{M}\left(y_j - \varphi^{(2)}\left(\boxed{\sum_{s=0}^{K} \boldsymbol{w}_{js}^{(2)} \underbrace{\varphi^{(1)}\left(\sum_{i=0}^{N} w_{si}^{(1)} x_i\right)}_{v_s}}\right)\right)^2\right)}{\partial \boldsymbol{w}_{js}^{(2)}}$$

$$= -\left(y_j - u_j\right)\frac{d\varphi^{(2)}\left(g_j\right)}{dg_j}\boxed{v_s} = \delta_j^{(2)} v_s,$$

$$g_j = \sum_{s=0}^{K} w_{js}^{(2)} v_s, \qquad \delta_j^{(2)} = \frac{\partial E(w)}{\partial g_j}$$

Метод обратного распространения ошибки на примере двухслойной сети (3)

- Производная целевой функции по параметрам скрытого слоя:

$$\frac{\partial E}{\partial w_{si}^{(1)}} = \frac{\partial \left(\boxed{\frac{1}{2}\sum_{j=1}^{M}\left(y_j - \varphi^{(2)}\left(\sum_{s=0}^{K} w_{js}^{(2)}\, \varphi^{(1)}\left(\sum_{i=0}^{N} \boldsymbol{w_{si}^{(1)}} x_i\right)\right)\right)^2} \right)}{\partial \boldsymbol{w_{si}^{(1)}}}$$

$$= -\boxed{\sum_{j=1}^{M}(y_j - u_j)} \frac{d\varphi^{(2)}(g_j)}{dg_j} \frac{dg_j(v_s)}{dv_s} \frac{d\varphi^{(1)}(f_s)}{df_s} x_i =$$

$$= -\sum_{j=1}^{M}(y_j - u_j) \frac{d\varphi^{(2)}(g_j)}{dg_j} w_{js}^{(2)} \frac{d\varphi^{(1)}(f_s)}{df_s} x_i = \delta_s^{(1)} x_i,$$

$$f_s = \sum_{i=0}^{N} w_{si}^{(1)} x_i\,, \qquad \delta_s^{(1)} = \frac{\partial E(w)}{\partial f_s}$$

Метод обратного распространения ошибки на примере двухслойной сети (3)

- Производная целевой функции по параметрам скрытого слоя:

$$\frac{\partial E}{\partial w_{si}^{(1)}} = \frac{\partial\left(\frac{1}{2}\sum_{j=1}^{M}\left(y_j \boxed{- \varphi^{(2)}\left(\sum_{s=0}^{K} w_{js}^{(2)}\,\varphi^{(1)}\left(\sum_{i=0}^{N} \boldsymbol{w}_{\boldsymbol{si}}^{(\mathbf{1})} x_i\right)\right)}\right)^2\right)}{\partial \boldsymbol{w}_{\boldsymbol{si}}^{(\mathbf{1})}}$$

$$= \boxed{-}\sum_{j=1}^{M}(y_j - u_j)\boxed{\frac{d\varphi^{(2)}(g_j)}{dg_j}}\frac{dg_j(v_s)}{dv_s}\frac{d\varphi^{(1)}(f_s)}{df_s}x_i =$$

$$= -\sum_{j=1}^{M}(y_j - u_j)\frac{d\varphi^{(2)}(g_j)}{dg_j}w_{js}^{(2)}\frac{d\varphi^{(1)}(f_s)}{df_s}x_i = \delta_s^{(1)}x_i,$$

$$f_s = \sum_{i=0}^{N} w_{si}^{(1)} x_i\,, \qquad \delta_s^{(1)} = \frac{\partial E(w)}{\partial f_s}$$

Метод обратного распространения ошибки на примере двухслойной сети (3)

- Производная целевой функции по параметрам скрытого слоя:

$$\frac{\partial E}{\partial w_{si}^{(1)}} = \frac{\partial\left(\frac{1}{2}\sum_{j=1}^{M}\left(y_j - \varphi^{(2)}\left(\boxed{\sum_{s=0}^{K} w_{js}^{(2)}\,\varphi^{(1)}\left(\sum_{i=0}^{N} \boldsymbol{w}_{\boldsymbol{si}}^{(\mathbf{1})} x_i\right)}\right)\right)^2\right)}{\partial \boldsymbol{w}_{\boldsymbol{si}}^{(\mathbf{1})}}$$

$$= -\sum_{j=1}^{M}(y_j - u_j)\frac{d\varphi^{(2)}(g_j)}{dg_j}\boxed{\frac{dg_j(v_s)}{dv_s}}\frac{d\varphi^{(1)}(f_s)}{df_s}x_i =$$

$$= -\sum_{j=1}^{M}(y_j - u_j)\frac{d\varphi^{(2)}(g_j)}{dg_j}w_{js}^{(2)}\frac{d\varphi^{(1)}(f_s)}{df_s}x_i = \delta_s^{(1)}x_i,$$

$$f_s = \sum_{i=0}^{N} w_{si}^{(1)} x_i\,, \qquad \delta_s^{(1)} = \frac{\partial E(w)}{\partial f_s}$$

Метод обратного распространения ошибки на примере двухслойной сети (3)

- Производная целевой функции по параметрам скрытого слоя:

$$\frac{\partial E}{\partial w_{si}^{(1)}} = \frac{\partial\left(\frac{1}{2}\sum_{j=1}^{M}\left(y_j - \varphi^{(2)}\left(\sum_{s=0}^{K} w_{js}^{(2)} \boxed{\varphi^{(1)}\left(\sum_{i=0}^{N} \boldsymbol{w}_{\boldsymbol{si}}^{(\mathbf{1})} x_i\right)}\right)\right)^2\right)}{\partial \boldsymbol{w}_{\boldsymbol{si}}^{(\mathbf{1})}}$$

$$= -\sum_{j=1}^{M}(y_j - u_j)\frac{d\varphi^{(2)}(g_j)}{dg_j}\frac{dg_j(v_s)}{dv_s}\boxed{\frac{d\varphi^{(1)}(f_s)}{df_s}}x_i =$$

$$= -\sum_{j=1}^{M}(y_j - u_j)\frac{d\varphi^{(2)}(g_j)}{dg_j}w_{js}^{(2)}\frac{d\varphi^{(1)}(f_s)}{df_s}x_i = \delta_s^{(1)}x_i,$$

$$f_s = \sum_{i=0}^{N} w_{si}^{(1)} x_i\,, \qquad \delta_s^{(1)} = \frac{\partial E(w)}{\partial f_s}$$

Метод обратного распространения ошибки на примере двухслойной сети (3)

- Производная целевой функции по параметрам скрытого слоя:

$$\frac{\partial E}{\partial w_{si}^{(1)}} = \frac{\partial\left(\frac{1}{2}\sum_{j=1}^{M}\left(y_j - \varphi^{(2)}\left(\sum_{s=0}^{K} w_{js}^{(2)}\,\varphi^{(1)}\left(\boxed{\sum_{i=0}^{N} \boldsymbol{w}_{\boldsymbol{si}}^{(\mathbf{1})} x_i}\right)\right)\right)^2\right)}{\partial \boldsymbol{w}_{\boldsymbol{si}}^{(\mathbf{1})}}$$

$$= -\sum_{j=1}^{M}(y_j - u_j)\frac{d\varphi^{(2)}(g_j)}{dg_j}\frac{dg_j(v_s)}{dv_s}\frac{d\varphi^{(1)}(f_s)}{df_s}\boxed{x_i} =$$

$$= -\sum_{j=1}^{M}(y_j - u_j)\frac{d\varphi^{(2)}(g_j)}{dg_j}w_{js}^{(2)}\frac{d\varphi^{(1)}(f_s)}{df_s}x_i = \delta_s^{(1)}x_i,$$

$$f_s = \sum_{i=0}^{N} w_{si}^{(1)} x_i\,, \qquad \delta_s^{(1)} = \frac{\partial E(w)}{\partial f_s}$$

Метод обратного распространения ошибки на примере двухслойной сети (4)

- Описание градиента по параметрам выходного и скрытого слоев имеют идентичную структуру:

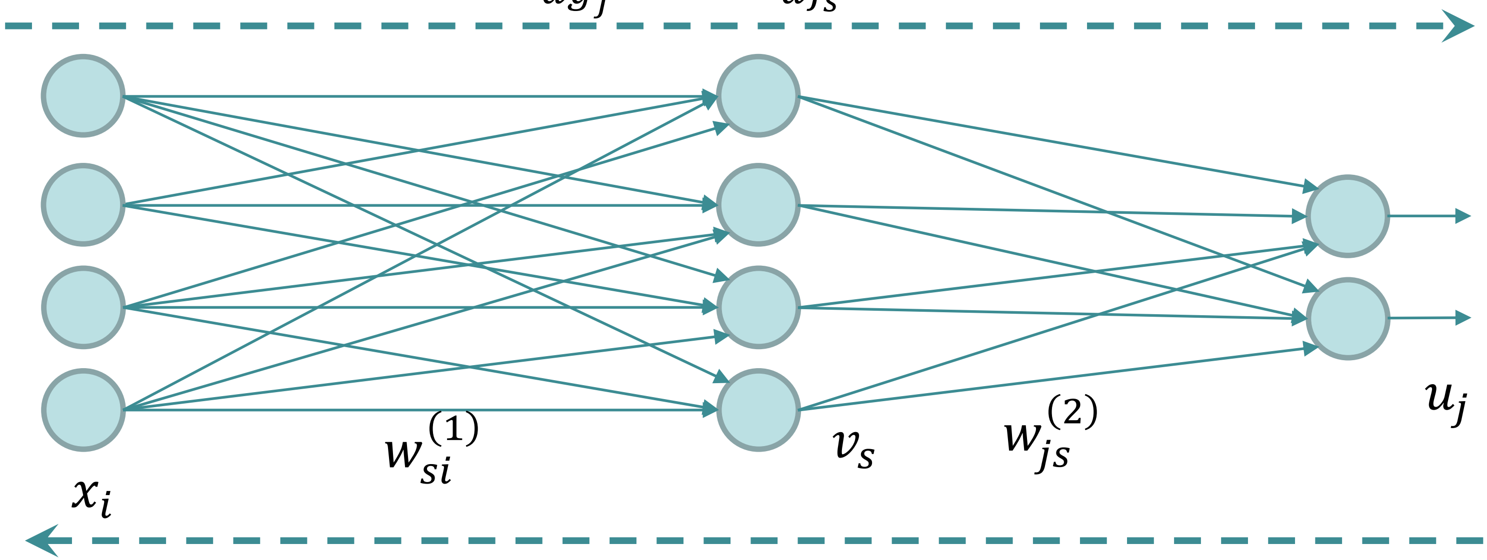
$$\frac{\partial E}{\partial w_{js}^{(2)}} = \delta_j^{(2)} v_s \qquad \frac{\partial E}{\partial w_{si}^{(1)}} = \delta_s^{(1)} x_i$$

Начальный нейрон взвешенной связи

Величина погрешности, перенесенной на узел, с которым эта связь установлена

Метод обратного распространения ошибки на примере двухслойной сети (5)

1. ***Прямой проход:***
   - Вычисление v_s и u_j
   - Вычисление $\frac{d\varphi^{(2)}(g_j)}{dg_j}$ и $\frac{d\varphi^{(1)}(f_s)}{df_s}$



2. ***Обратный проход:***
   - Вычисление целевой функции E и ее градиента $\frac{\partial E}{\partial w_{js}^{(2)}}$ и $\frac{\partial E}{\partial w_{si}^{(1)}}$
   - Коррекция весов $w(k+1) = w(k) - \eta \nabla E(w)$

Сходимость метода обратного распространения ошибки (1)

- Доказательства сходимости метода обратного распространения ошибки не существует

- Разумное условие остановки: евклидова норма вектора градиента достигла достаточно малых значений
- Для выполнения указанного критерия может потребоваться большое количество итераций метода обратного распространения

Сходимость метода обратного распространения ошибки (2)

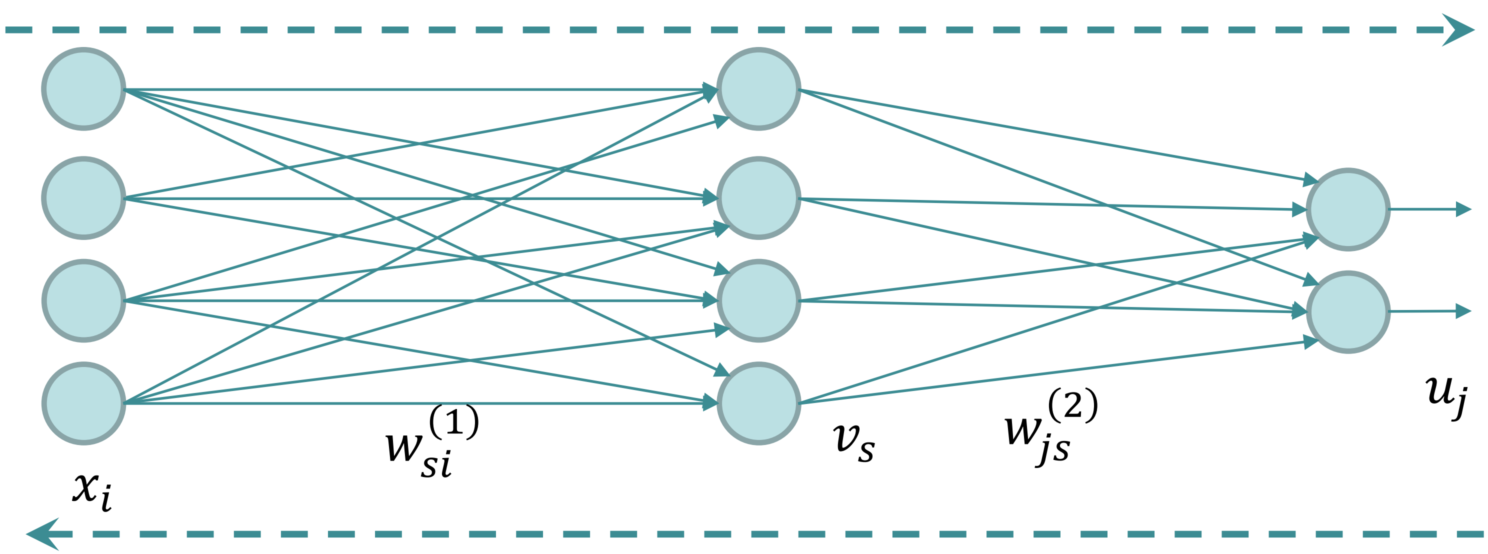
- Более слабое условие остановки: в течение полного цикла предъявления всех обучающих примеров абсолютное значение изменения ошибки достаточно мало (в пределах от 0.1 до 1%)
- Условие не гарантирует, что полученная сеть обладает хорошими обобщающими свойствами

Последовательный и пакетный режимы обучения (1)

- В процессе обучения многослойной полносвязной нейронной сети ей многократно предъявляется предопределенное множество обучающих примеров
- Один полный цикл предъявления полного набора примеров называется ***эпохой*** (epoch)
  – При обучении может выполняться несколько таких циклов до момента стабилизации синаптических весов, либо достижения минимального значения функции ошибки
  – При реализации эпох целесообразно изменять порядок примеров в обучающей выборке, обеспечивая стохастический поиск

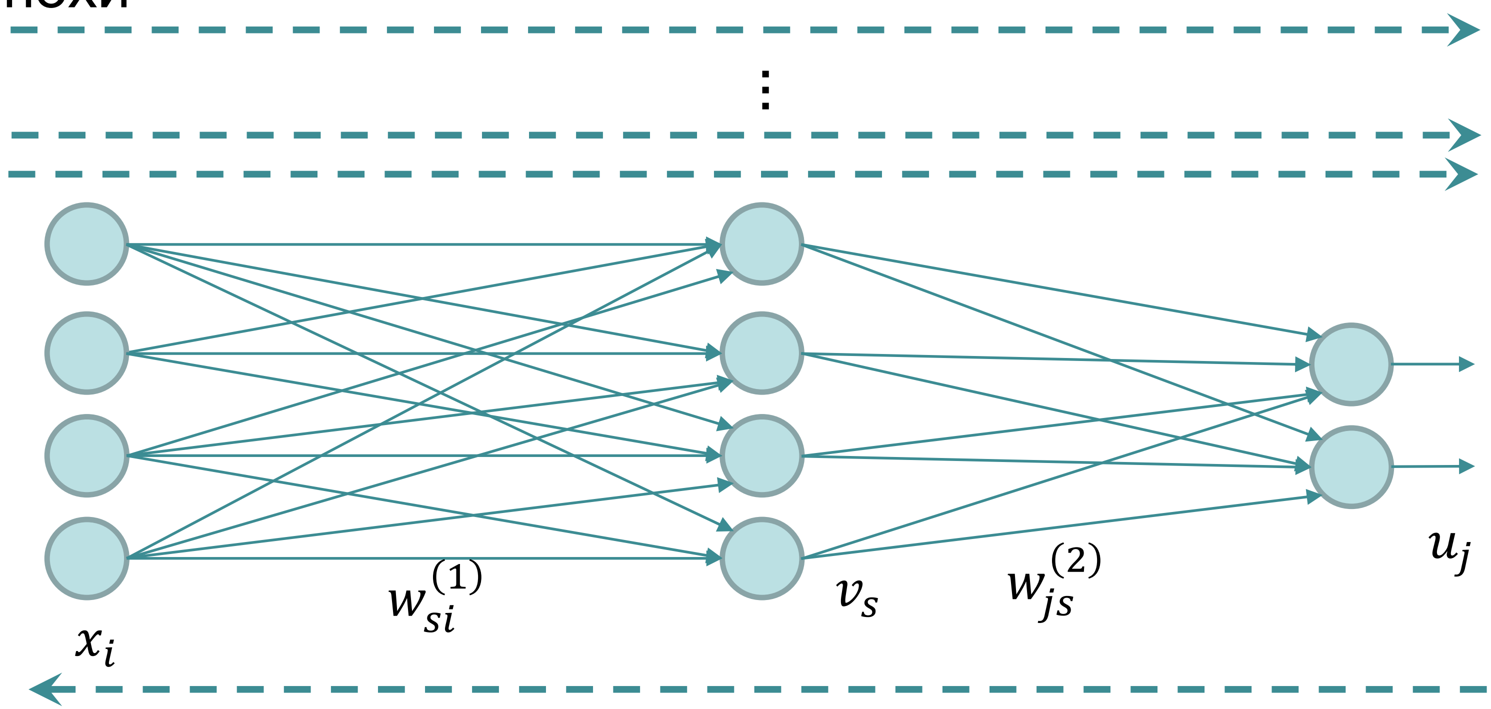
Последовательный и пакетный режимы обучения (2)

- Режимы реализации метода обратного распространения ошибки:
  - ***Последовательный (или стохастический) режим (sequential or stochastic mode)***. В указанном режиме корректировка весов выполняется после предъявления каждого примера обучающей выборки



Последовательный и пакетный режимы обучения (3)

- Режимы реализации метода обратного распространения ошибки:
  – ***Пакетный режим (batch mode).*** В данном режиме корректировка весов осуществляется по всем примерам эпохи



Последовательный и пакетный режимы обучения (4)

- Функция ошибки для всего набора тренировочных данных, нормированная по числу примеров выборки:

$$E(w) = \frac{1}{L}\sum_{k=1}^{L}\left\{\frac{1}{2}\sum_{j=1}^{M}\left(y_j^k - u_j^k\right)^2\right\}$$

- Корректировка весов проводится по всему набору данных. Для двухслойной полностью связанной сети нейронной сети формулы корректировки:

$$\Delta w_{js}^{(2)} = -\frac{\eta}{L}\sum_{k=1}^{L}\left(y_j^k - u_j^k\right)\left.\frac{d\varphi^{(2)}(g_j)}{dg_j}\right|_{x^k} \left.v_s\right|_{x^k},$$

$$\Delta w_{si}^{(1)} = -\frac{\eta}{L}\sum_{k=1}^{L}\sum_{j=1}^{M}\left(y_j - u_j\right)\left.\frac{d\varphi^{(2)}(g_j)}{dg_j}\right|_{x^k} \left.\frac{dg_j(v_s)}{dv_s}\right|_{x^k} \left.\frac{d\varphi^{(1)}(f_s)}{df_s}\right|_{x^k} x_i^k$$

Последовательный и пакетный режимы обучения (5)

- ***Последовательный режим обучения (sequential mode)***
  – Медленный

- ***Пакетный режим обучения (batch mode)***
  – Быстрый и стабильный
  – Может «застревать» в локальных минимумах

Последовательный и пакетный режимы обучения (6)

- Компромисс – применение мини-пакетов (mini-batch)
  - Тренировочное множество примеров разбивается на мини-пакеты
  - Прямой проход выполняется для всего множества примеров из мини-пакета
  - Обратный проход и коррекция весов осуществляется после обработки мини-пакета

Эвристические рекомендации по повышению производительности

- ***Максимизация информативности***
  - Наличие кардинально разных примеров в обучающей выборке (не только визуально воспринимаемые различия внешнего вида, но и различия в значениях функции ошибки)
- ***Функция активации***
  - Полносвязная сеть обучается быстрее, если функция активации является антисимметричной $\varphi(-x) = -\varphi(x)$
- ***Нормализация входов***
  - Все входные переменные следует предварительно нормализовать по всему обучающему множеству

Заключение

- Введена модель нейрона
- Рассмотрена общая схема построения полносвязных нейронных сетей
- Введена математическая постановка задачи поиска весов полносвязной сети
- Рассмотрена общая схема метода обратного распространения ошибки для обучения параметров сети
- Далее рассматривается пример использования глубоких полносвязных сетей для решения задачи компьютерного зрения средствами Intel® neon™ Framework

Основная литература

- Хайкин С. Нейронные сети. Полный курс. – М.: Издательский дом «Вильямс». – 2006. – 1104 с.
- Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика. – 2002. – 344 с.
- Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press. – 2016. – [http://www.deeplearningbook.org].

Авторский коллектив

- **Кустикова Валентина Дмитриевна**
  к.т.н., ст.преп. каф. МОСТ ИИТММ,
  ННГУ им. Н.И. Лобачевского
  valentina.kustikova@itmm.unn.ru
- **Жильцов Максим Сергеевич**
  магистрант каф. МОСТ ИИТММ,
  ННГУ им. Н.И. Лобачевского
  zhiltsov.max35@gmail.com
- **Золотых Николай Юрьевич**
  д.ф.-м.н., проф. каф. АГДМ ИИТММ,
  ННГУ им. Н.И. Лобачевского
  nikolai.zolotykh@itmm.unn.ru