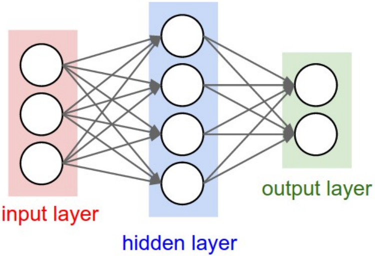
Лекция 2. Обучение нейронных сетей. Backpropagation

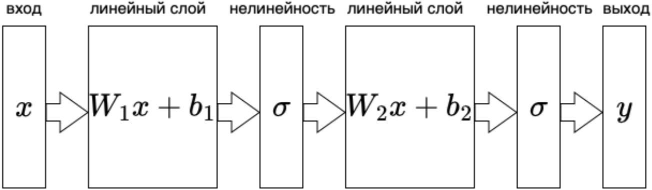
Глубинное обучение

Антон Кленицкий

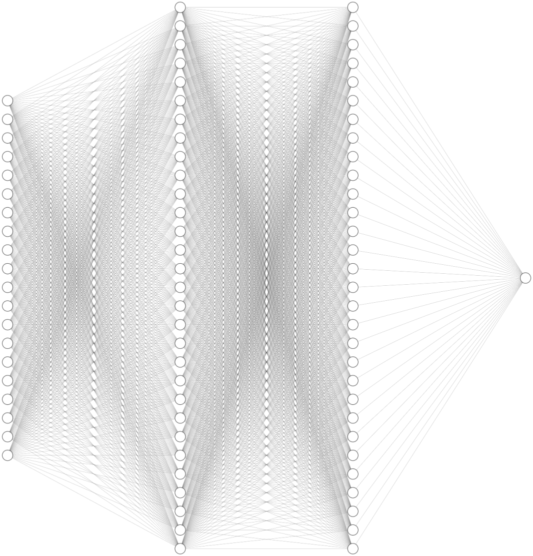
Multilayer Perceptron



$$y = \sigma(W_2\sigma(W_1x+b_1)+b_2)$$



В реальной жизни скорее так



Universal approximation theorem

Теорема об универсальной аппроксимации:

Любую непрерывную функцию можно с любой точностью приблизить нейросетью с одним скрытым слоем с сигмоидной функцией активации.

- Hornik "Multilayer feedforward networks are universal approximators"(1989)
- Cybenko "Approximation by superpositions of a sigmoidal function"(1989)

Universal approximation theorem

Но!

- Может понадобиться очень много (экспоненциально много) нейронов
- Неизвестно, сможем ли мы обучить нашими методами такую нейросеть
- Неизвестно, как это будет обобщаться на новые данные

Почему же "глубокое"обучение?

Глубокие нейронные сети

- >1 скрытого слоя
- Строят иерархию признаков
- Имеют большую выразительность при том же числе нейронов
- Работают на практике

Image credit

Convolutional neural networks (CNN) - Сверточные сети

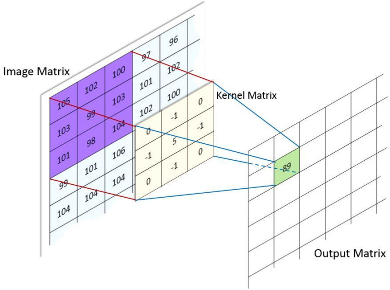


Image credit

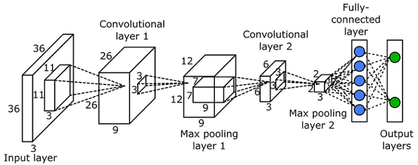


Image credit

- Могут быть действительно глубокими (100+ слоев)
- Изображения, звук, последовательности

Основные архитектуры нейросетей

Recurrent neural networks (RNN) - Рекуррентные сети

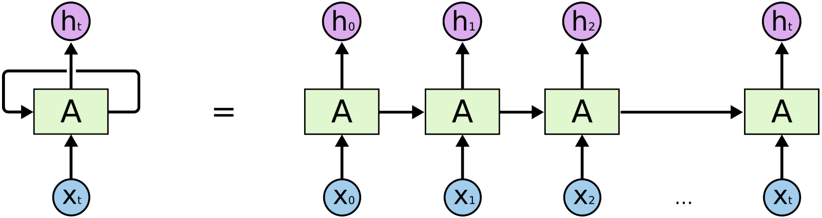


Image credit

Для работы с последовательностями

Transformers

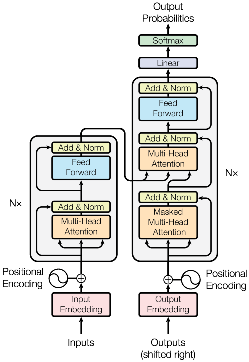


Image credit

NLP, последовательности, изображения, таблицы

Autoencoders

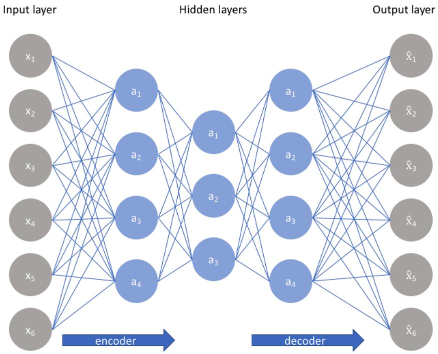
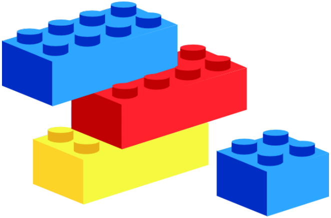
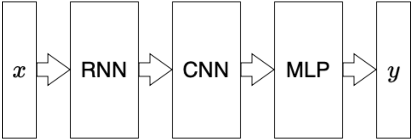


Image credit

Модель можно составлять из разных блоков





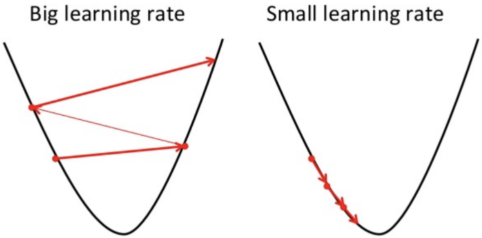
Как обучать нейросети?

Gradient descent

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i, \theta)) \to \min_{\theta}$$

$$\nabla_\theta L(\theta) = \left(\frac{\partial L}{\partial \theta_1}, \frac{\partial L}{\partial \theta_2}, \ldots\right)$$

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla_\theta L(\theta^{(t)})$$



Convex vs non-convex optimization

Выпуклая оптимизация | Невыпуклая оптимизация

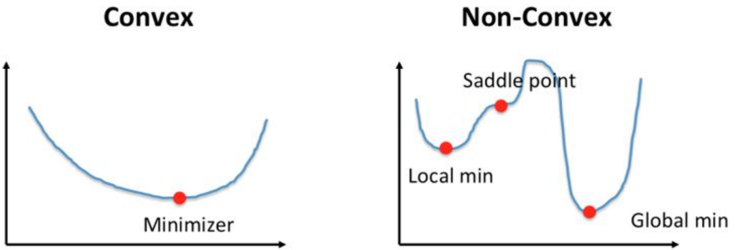
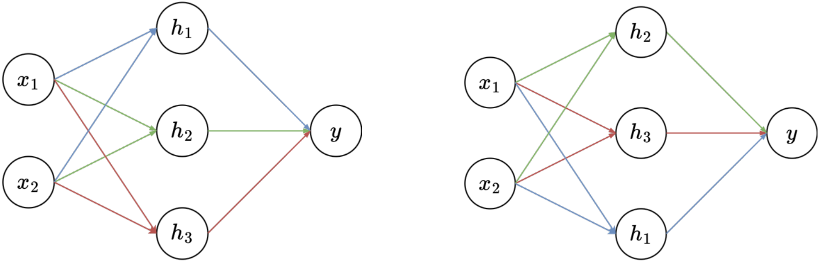


Image credit

Нейросети - сложные функции с огромным количеством локальных минимумов

Локальные минимумы у нейросетей

Изменится ли выход нейросети при перемешивании нейронов в скрытом слое?



Batch Gradient Descent

(Batch) gradient descent

$$\theta_{t+1} = \theta_t - \alpha \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta L(y_i, f(x_i, \theta))$$

Stochastic Gradient Descent

Stochastic gradient descent (SGD) - обновляем веса после каждого примера

$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta L(y_i, f(x_i, \theta))$$

- Несмещенная оценка полного градиента
- Если берем примеры из обучающей выборки в случайном порядке!

Mini-batch stochastic gradient descent - обновляем веса после батча из B примеров

$$\theta_{t+1} = \theta_t - \alpha \frac{1}{B} \sum_{i=1}^{B} \nabla_\theta L(y_i, f(x_i, \theta))$$

- Можно эффективно использовать матричные вычисления

Stochastic Gradient Descent

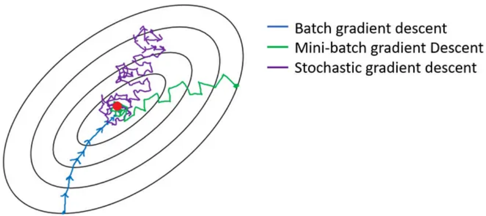


Image credit

- Шумная оценка полного градиента
- Может проскочить мимо неудачного локального минимума или седловой точки

Wide vs sharp minimum

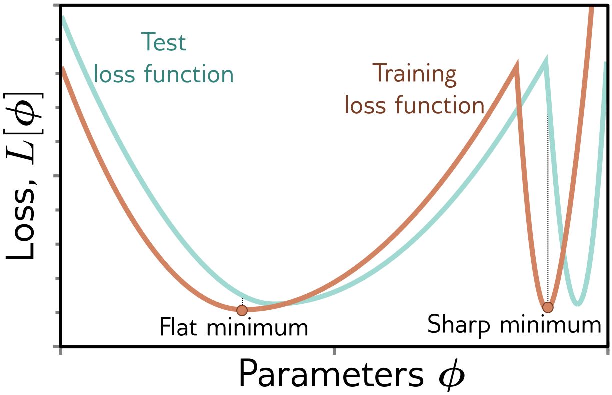
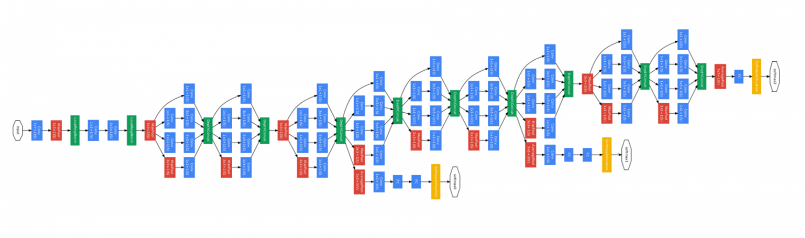


Image credit

Как считать градиенты?

Аналитически?



Но модель может быть очень большая и сложная :)

Как считать градиенты?

Численно?

Конечные разности

$$\frac{\partial f(x,\theta)}{\partial \theta} \approx \frac{f(x,\theta+\epsilon) - f(x,\theta-\epsilon)}{2\epsilon}$$

- Приближенный метод
- Очень долго - для каждого параметра по отдельности
- Можно использовать для проверки

Backpropagation

Алгоритм обратного распространения ошибки

https://karpathy.medium.com/yes-you-should-understand-backprop-e2f06eab496b

Правило дифференцирования сложной функции

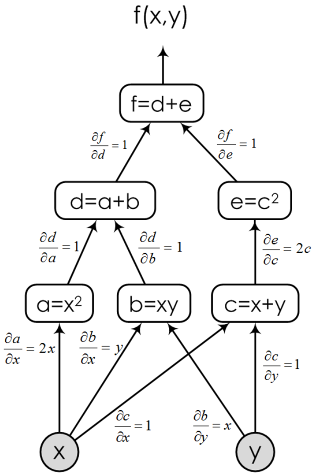
$$f = f(g(x))$$

$$\frac{df}{dx} = \frac{df}{dg}\frac{dg}{dx}$$

$$f = f(g_1(x), g_2(x), ..., g_k(x))$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial g_1}\frac{\partial g_1}{\partial x} + ... + \frac{\partial f}{\partial g_k}\frac{\partial g_k}{\partial x} = \sum_{i=1}^{k}\frac{\partial f}{\partial g_i}\frac{\partial g_i}{\partial x}$$

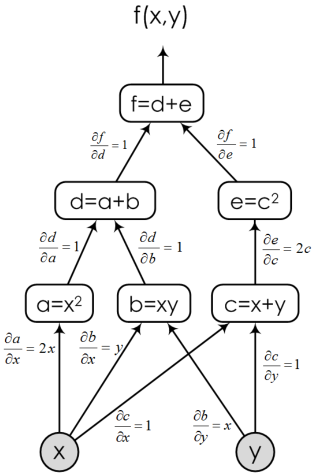
Простой пример



$$f(x) = x^2 + xy + (x+y)^2$$

Граф вычислений

Простой пример

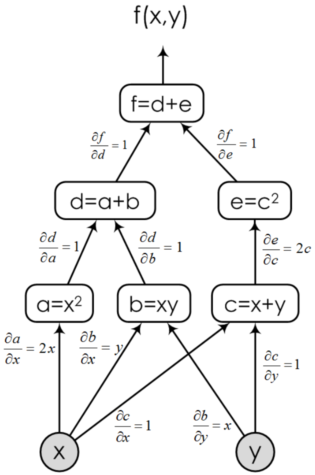


Граф вычислений

$$f(x) = x^2 + xy + (x+y)^2$$

$$\frac{\partial f}{\partial d} = 1; \frac{\partial f}{\partial e} = 1$$

Простой пример



Граф вычислений

$$f(x) = x^2 + xy + (x+y)^2$$

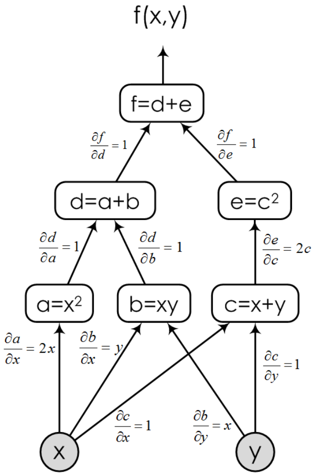
$$\frac{\partial f}{\partial d} = 1; \frac{\partial f}{\partial e} = 1$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial d}\frac{\partial d}{\partial a} = 1$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial d}\frac{\partial d}{\partial b} = 1$$

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial e}\frac{\partial e}{\partial c} = 2c$$

Простой пример



Граф вычислений

$$f(x) = x^2 + xy + (x+y)^2$$

$$\frac{\partial f}{\partial d} = 1; \frac{\partial f}{\partial e} = 1$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial d}\frac{\partial d}{\partial a} = 1$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial d}\frac{\partial d}{\partial b} = 1$$

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial e}\frac{\partial e}{\partial c} = 2c$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a}\frac{\partial a}{\partial x} + \frac{\partial f}{\partial b}\frac{\partial b}{\partial x} + \frac{\partial f}{\partial c}\frac{\partial c}{\partial x} = 2x + y + 2(x+y)$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial b}\frac{\partial b}{\partial y} + \frac{\partial f}{\partial c}\frac{\partial c}{\partial y} = x + 2(x+y)$$

Простая нейросеть

$$\hat{y} = w_3\sigma(w_2\sigma(w_1 x + b_1) + b_2) + b_3; \qquad L(y, \hat{y}) = (y - \hat{y})^2$$

Простая нейросеть

$$\hat{y} = w_3\sigma(w_2\sigma(w_1x + b_1) + b_2) + b_3; \qquad L(y, \hat{y}) = (y - \hat{y})^2$$

$$\boldsymbol{x} \to \boldsymbol{f_1} \to \boldsymbol{h_1} \to \boldsymbol{f_2} \to \boldsymbol{h_2} \to \boldsymbol{f_3} \to \boldsymbol{L}$$

$$f_1 = w_1x + b_1$$

$$h_1 = \sigma(f_1)$$

$$f_2 = w_2h_1 + b_2$$

$$h_2 = \sigma(f_2)$$

$$f_3 = w_3h_2 + b_3$$

$$L = (f_3 - y)^2$$

Простая нейросеть

$$\hat{y} = w_3\sigma(w_2\sigma(w_1x + b_1) + b_2) + b_3; \qquad L(y, \hat{y}) = (y - \hat{y})^2$$

$$x \to f_1 \to h_1 \to f_2 \to h_2 \to f_3 \to L$$

$$\frac{\partial L}{\partial f_3} = 2(f_3 - y)$$

$$f_1 = w_1x + b_1$$

$$h_1 = \sigma(f_1)$$

$$f_2 = w_2h_1 + b_2$$

$$h_2 = \sigma(f_2)$$

$$f_3 = w_3h_2 + b_3$$

$$L = (f_3 - y)^2$$

Простая нейросеть

$$\hat{y} = w_3\sigma(w_2\sigma(w_1x + b_1) + b_2) + b_3; \qquad L(y, \hat{y}) = (y - \hat{y})^2$$

$$x \to f_1 \to h_1 \to f_2 \to h_2 \to f_3 \to L$$

$$\frac{\partial L}{\partial f_3} = 2(f_3 - y)$$

$$\frac{\partial L}{\partial h_2} = \left(\frac{\partial L}{\partial f_3}\right)\frac{\partial f_3}{\partial h_2}$$

$$f_1 = w_1x + b_1$$

$$h_1 = \sigma(f_1)$$

$$f_2 = w_2h_1 + b_2$$

$$h_2 = \sigma(f_2)$$

$$f_3 = w_3h_2 + b_3$$

$$L = (f_3 - y)^2$$

Простая нейросеть

$$\hat{y} = w_3\sigma(w_2\sigma(w_1x + b_1) + b_2) + b_3; \qquad L(y, \hat{y}) = (y - \hat{y})^2$$

$$x \rightarrow f_1 \rightarrow h_1 \rightarrow f_2 \rightarrow h_2 \rightarrow f_3 \rightarrow L$$

$$\frac{\partial L}{\partial f_3} = 2(f_3 - y)$$

$$\frac{\partial L}{\partial h_2} = \left(\frac{\partial L}{\partial f_3}\right)\frac{\partial f_3}{\partial h_2}$$

$$\frac{\partial L}{\partial f_2} = \left(\frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial h_2}\right)\frac{\partial h_2}{\partial f_2}$$

$$f_1 = w_1x + b_1$$

$$h_1 = \sigma(f_1)$$

$$f_2 = w_2h_1 + b_2$$

$$h_2 = \sigma(f_2)$$

$$f_3 = w_3h_2 + b_3$$

$$L = (f_3 - y)^2$$

Простая нейросеть

$$\hat{y} = w_3\sigma(w_2\sigma(w_1x + b_1) + b_2) + b_3; \qquad L(y, \hat{y}) = (y - \hat{y})^2$$

$$x \rightarrow f_1 \rightarrow h_1 \rightarrow f_2 \rightarrow h_2 \rightarrow f_3 \rightarrow L$$

$$f_1 = w_1x + b_1$$

$$h_1 = \sigma(f_1)$$

$$f_2 = w_2h_1 + b_2$$

$$h_2 = \sigma(f_2)$$

$$f_3 = w_3h_2 + b_3$$

$$L = (f_3 - y)^2$$

$$\frac{\partial L}{\partial f_3} = 2(f_3 - y)$$

$$\frac{\partial L}{\partial h_2} = \left(\frac{\partial L}{\partial f_3}\right)\frac{\partial f_3}{\partial h_2}$$

$$\frac{\partial L}{\partial f_2} = \left(\frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial h_2}\right)\frac{\partial h_2}{\partial f_2}$$

$$\frac{\partial L}{\partial h_1} = \left(\frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial h_2}\frac{\partial h_2}{\partial f_2}\right)\frac{\partial f_2}{\partial h_1}$$

Простая нейросеть

$$\hat{y} = w_3\sigma(w_2\sigma(w_1x + b_1) + b_2) + b_3; \qquad L(y, \hat{y}) = (y - \hat{y})^2$$

$$x \to f_1 \to h_1 \to f_2 \to h_2 \to f_3 \to L$$

$$\frac{\partial L}{\partial f_3} = 2(f_3 - y)$$

$$f_1 = w_1x + b_1$$

$$\frac{\partial L}{\partial h_2} = \left(\frac{\partial L}{\partial f_3}\right)\frac{\partial f_3}{\partial h_2}$$

$$h_1 = \sigma(f_1)$$

$$f_2 = w_2h_1 + b_2$$

$$\frac{\partial L}{\partial f_2} = \left(\frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial h_2}\right)\frac{\partial h_2}{\partial f_2}$$

$$h_2 = \sigma(f_2)$$

$$f_3 = w_3h_2 + b_3$$

$$\frac{\partial L}{\partial h_1} = \left(\frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial h_2}\frac{\partial h_2}{\partial f_2}\right)\frac{\partial f_2}{\partial h_1}$$

$$L = (f_3 - y)^2$$

$$\frac{\partial L}{\partial f_1} = \left(\frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial h_2}\frac{\partial h_2}{\partial f_2}\frac{\partial f_2}{\partial h_1}\right)\frac{\partial h_1}{\partial f_1}$$

Простая нейросеть

$$\hat{y} = w_3\sigma(w_2\sigma(w_1 x + b_1) + b_2) + b_3; \qquad L(y, \hat{y}) = (y - \hat{y})^2$$

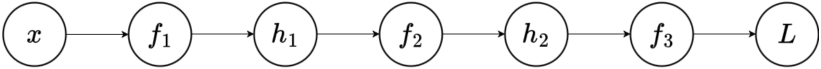
$$x \to f_1 \to h_1 \to f_2 \to h_2 \to f_3 \to L$$

$$f_1 = w_1 x + b_1$$
$$h_1 = \sigma(f_1)$$
$$f_2 = w_2 h_1 + b_2$$
$$h_2 = \sigma(f_2)$$
$$f_3 = w_3 h_2 + b_3$$
$$L = (f_3 - y)^2$$

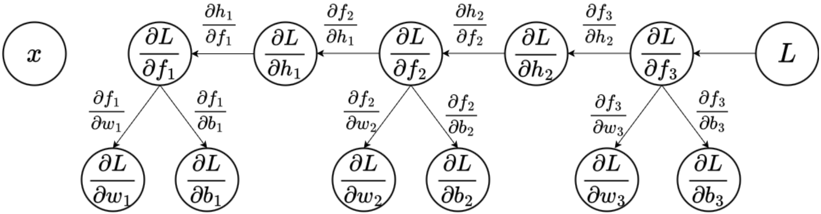
$$\frac{\partial L}{\partial w_k} = \left(\frac{\partial L}{\partial f_k}\right)\frac{\partial f_k}{\partial w_k} = \frac{\partial L}{\partial f_k} \cdot h_{k-1}$$

$$\frac{\partial L}{\partial b_k} = \left(\frac{\partial L}{\partial f_k}\right)\frac{\partial f_k}{\partial b_k} = \frac{\partial L}{\partial f_k} \cdot 1$$

Forward pass



Backward pass



Backpropagation summary

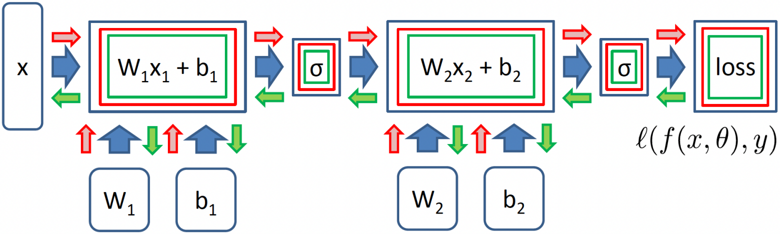


Image credit

- Нейросеть - граф вычислений
- Градиенты распространяются в обратную сторону по графу
- Умножаем на производную выхода по входу для каждого узла графа

Vanishing / exploding gradients

В процессе обратного распространения по сети градиенты могут "затухать"или "взрываться"

$$\frac{\partial L}{\partial f_1} = \left(\frac{\partial L}{\partial f_3} \frac{\partial f_3}{\partial h_2} \frac{\partial h_2}{\partial f_2} \frac{\partial f_2}{\partial h_1} \right) \frac{\partial h_1}{\partial f_1}$$

Let the gradient flow

Главное, чтобы градиент хорошо "протекал"по нейросети

Многое было придумано ради этого

- Хорошие функции активации
- Хорошая инициализация весов
- Skip residual connections
- Специальные архитектуры (LSTM/GRU)
- ...

DL Frameworks

DL Frameworks

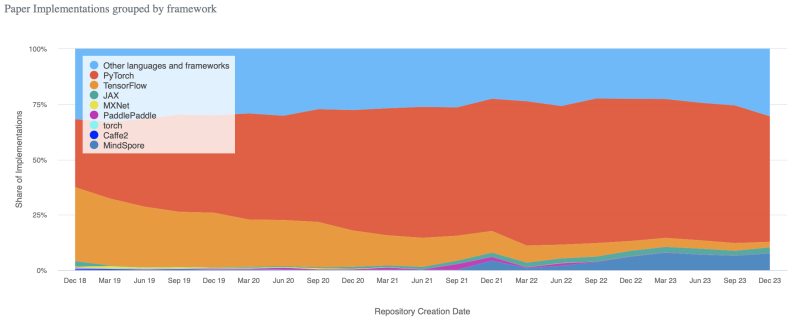
Автоматическое дифференцирование - расчет производных по графу вычислений

TensorFlow
Microsoft CNTK
theano
Keras
torch
PyTorch
mxnet
Chainer
Caffe
Caffe2

Image credit

DL Frameworks

`https://paperswithcode.com/trends`



В следующий раз

- Функции активации
- Функции потерь
- Инициализация весов