Deep Learning

Course

The purpose of this course is to acquaint you with the world of neural networks

You will learn:

- Different architectures
- Optimization techniques to learn them
- How to apply them

Team

Andrei Filatov

Olga Grebenkova

Anton Bishuk

Dmitry Kropotov

Plan

14 classes:

- Lecture + Break (10-15 minutes) + Seminar
- 5 Homeworks

Final grade:

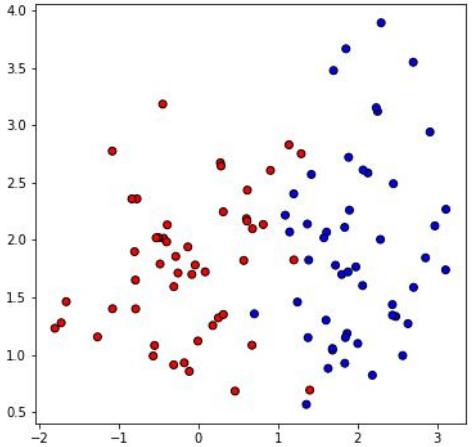
$$\max(0.7 * \mathrm{HW} + 0.3 * \mathrm{EXAM}, 10)$$

Cheating is prohibited!

Lecture 1: Multi-layer perceptron

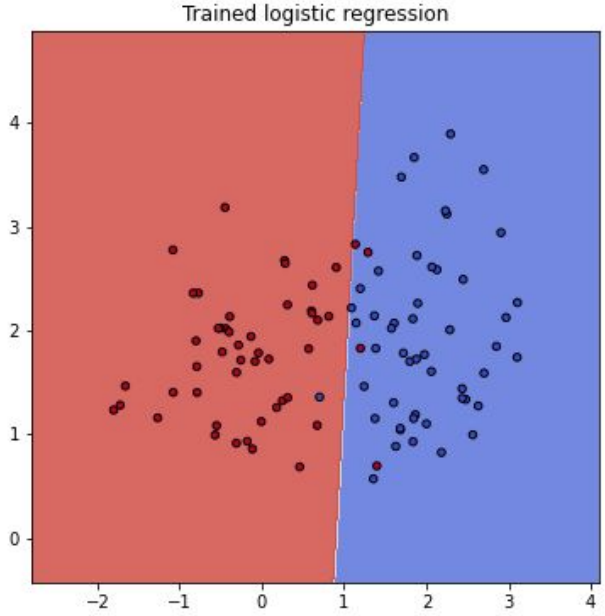
Consider some classification task:

$$\{x_i, y_i\}_{i=1}^n, \; x_i \in R^d, y_i \in \{-1, 1\};$$

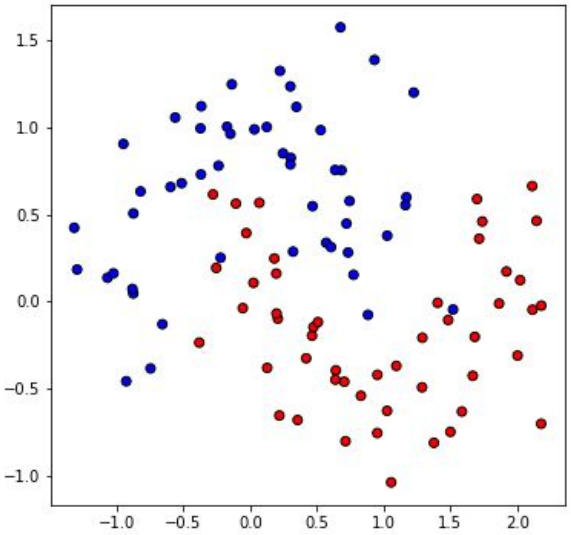


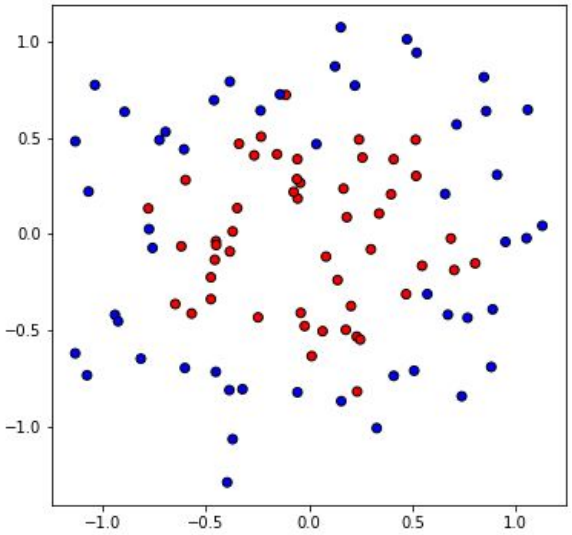
We want to train logistic regression models

$$p(y = 1|w, x) = \frac{1}{1 + \exp(-w^\top x)}$$

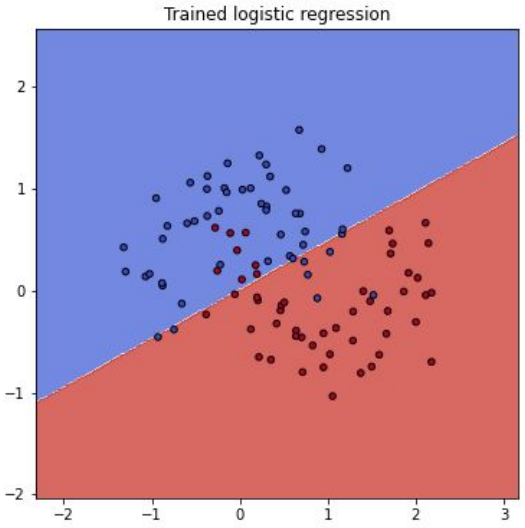

Trained logistic regression
4
3
2
1
0
−2
−1
0
1
2
3
4

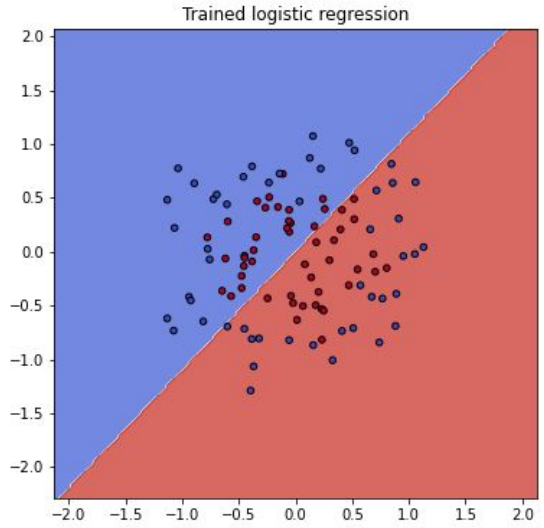
What does happen when our data has non-linear relationships



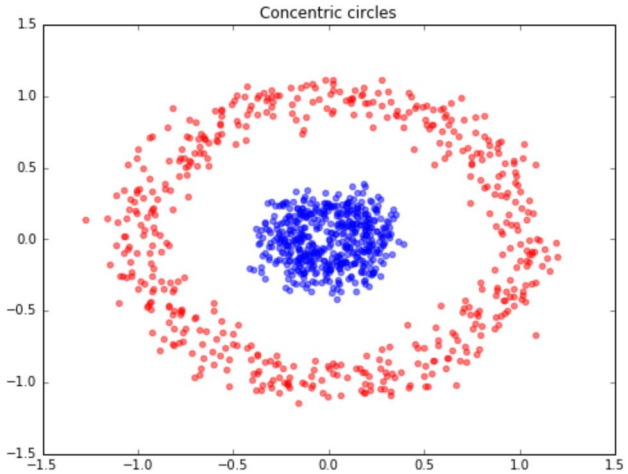


In this case logistic regression fails :(





The reason perceptron a.k.a logistic regression fails

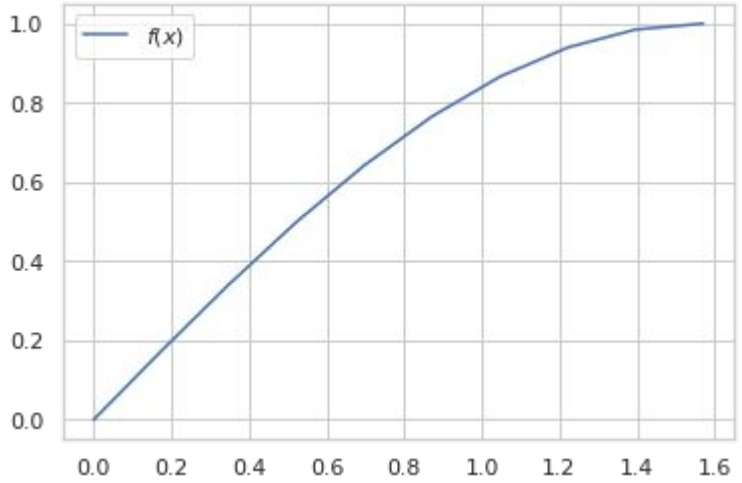


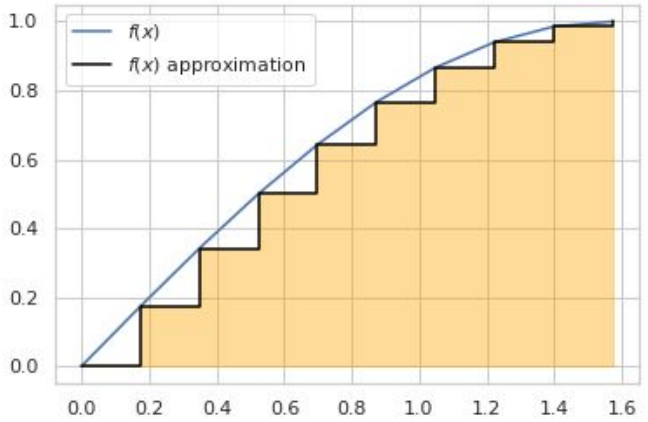
- Data can’t be splitted in linear way

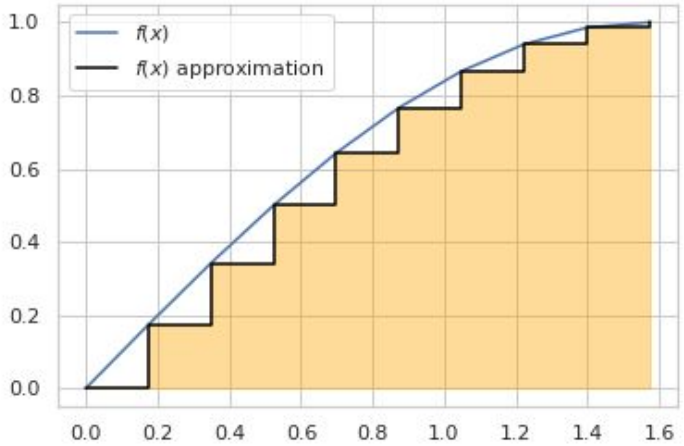
- Features itself are non-informative. But combination does.

https://thiscatdoesnotexist.com

MLP: universal approximator

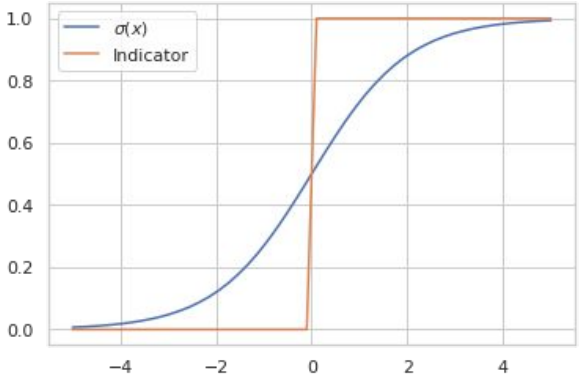


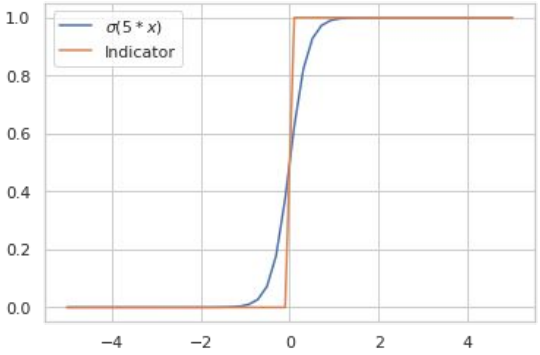


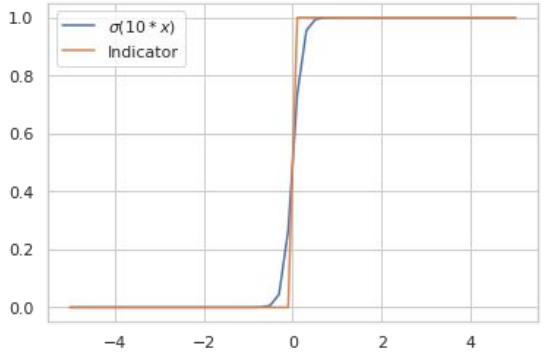


$$f(x) \approx \sum_{i=1}^{N} f(a_i)[x \in [a_i, b_i]]$$

Can we approximate the indicator with sigmoid?

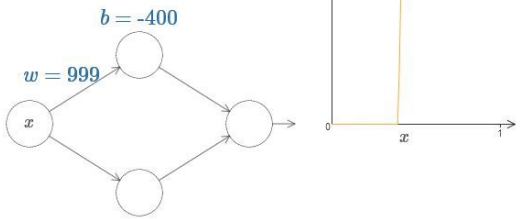


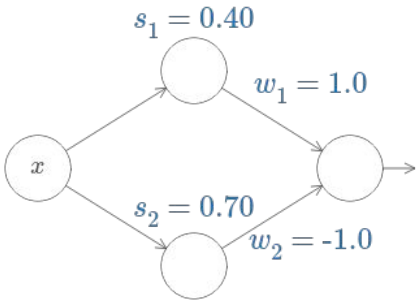


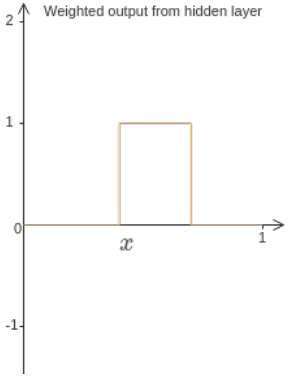


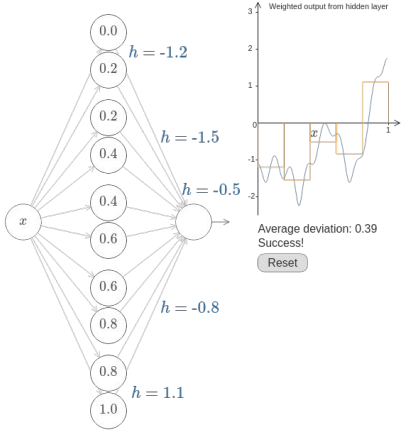
How to approximate interval indicator

$$[[x \geqslant a_i] - [x \geqslant b_i] > 0.5]$$









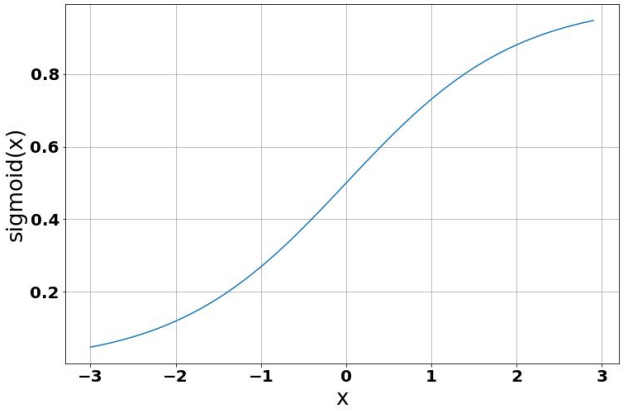
Universal approximation theorem

A sum of the form

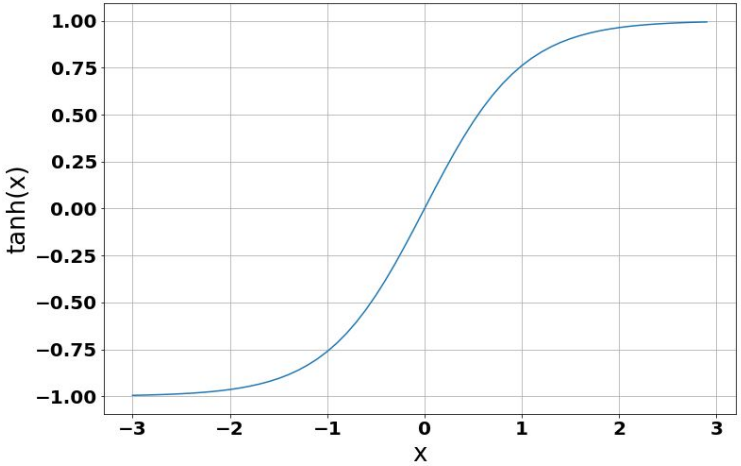
$$f(\mathbf{x}) = \sum_j w_j \sigma\left(b_j + \mathbf{v}_j \cdot \mathbf{x}\right)$$

can approximate any continuous function to any accuracy, but it might require any number of hidden neurons.

Activation function

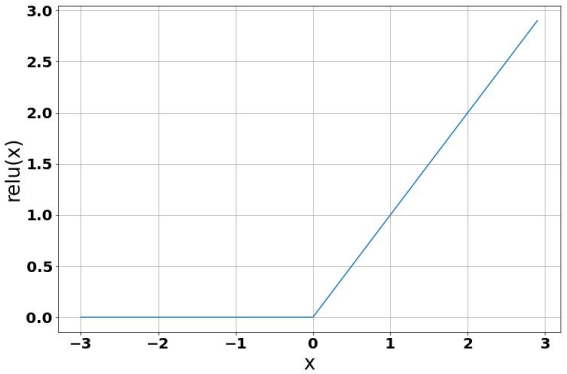


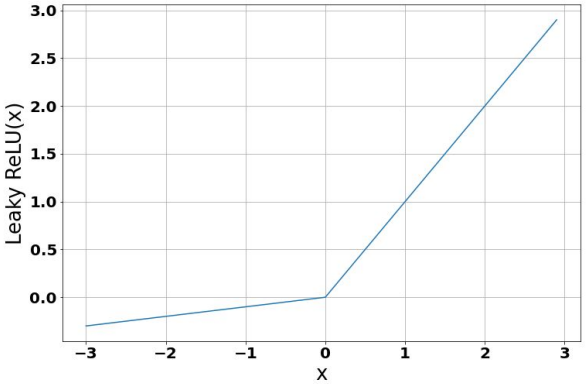
Sigmoid(x)

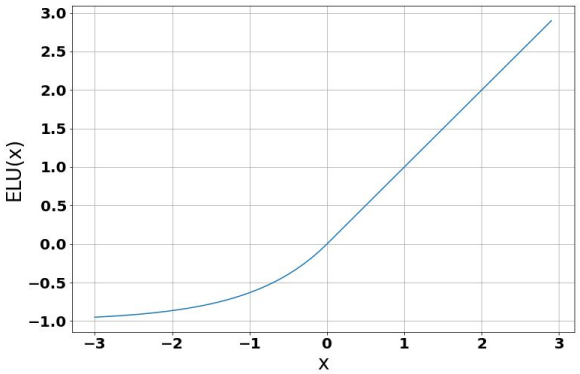


Tanh(x)

Activation functions





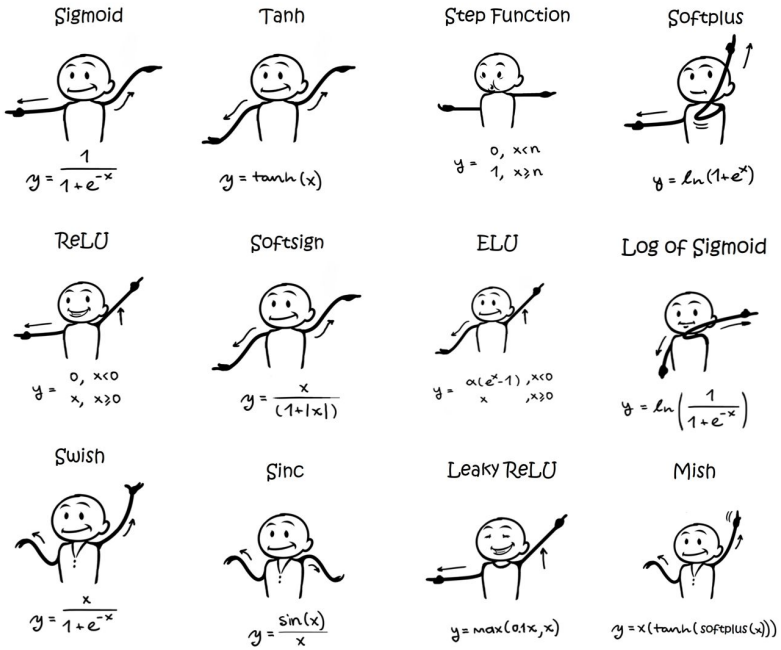


$$ReLU(x) = \begin{Bmatrix} x & x > 0 \\ 0 & x <= 0 \end{Bmatrix}$$

$$LeakyReLU(z) = \begin{Bmatrix} x & x > 0 \\ \alpha x & x <= 0 \end{Bmatrix}$$

$$ELU(x) = \begin{Bmatrix} x & x > 0 \\ \alpha(e^{\frac{x}{\alpha}} - 1) & x \leqslant 0 \end{Bmatrix}$$

Activation functions

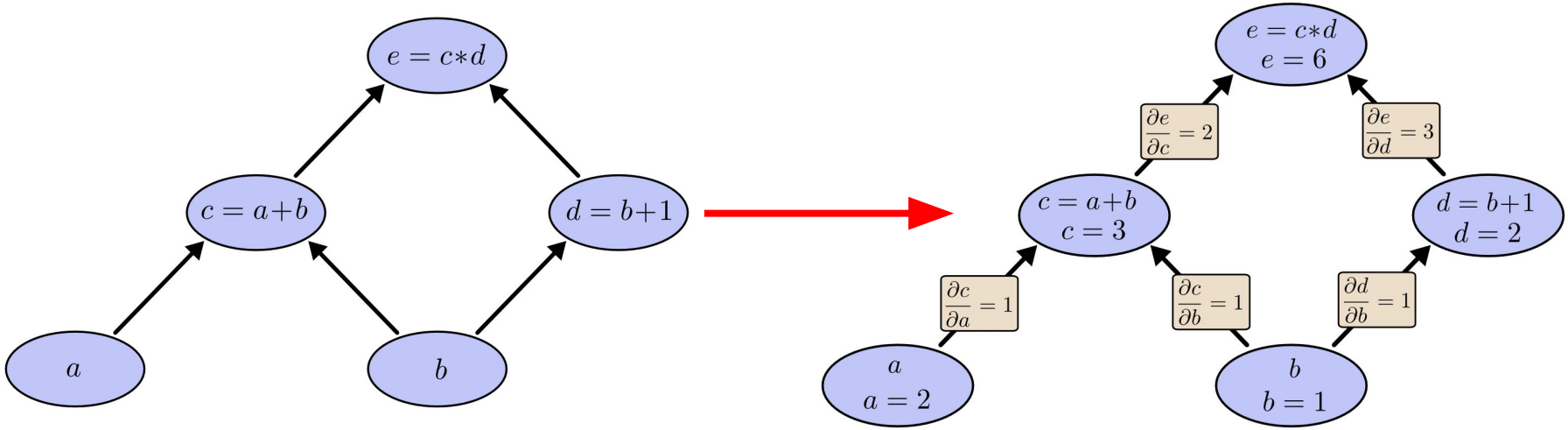


https://sefiks.com/2020/02/02/dance-moves-of-deep-learning-activation-functions/

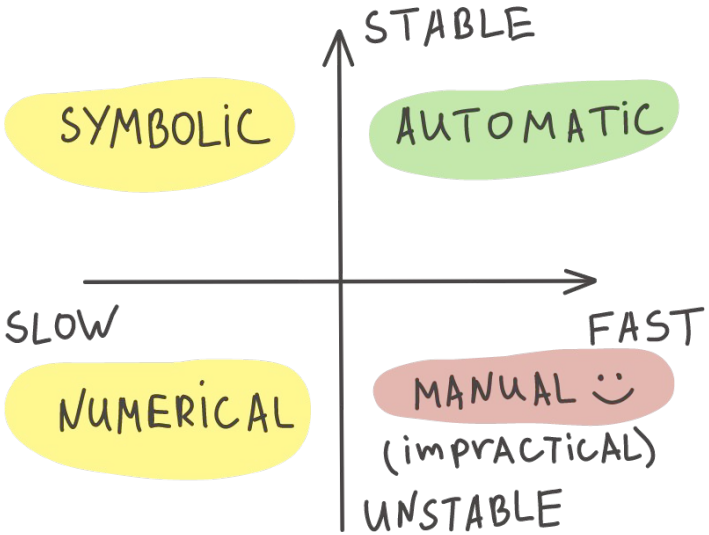
https://www.vincentsitzmann.com/siren/

Gradient calculation

How to calculate the gradient?



DIFFERENTIATION



Numerically

$$\frac{\partial f}{\partial x_i} = \frac{f(x + \varepsilon e_i) - f(x)}{\varepsilon}$$

Problems

- For large x dimension we should do a lot of function calls
- Numerically instability, optimal eps is equal to square root of machine precision

Numerically with any precision

Consider f - complex-valued function

$$f(x + i\varepsilon d) = f(x) + i\varepsilon \nabla_x f^\top d + O(\varepsilon^2)$$

So, we can calculate the gradient using only one function call

$$\nabla f(x)^\top d = \frac{Im[f(x + i\varepsilon d)]}{\varepsilon}$$

Matrix-vector differentiation

Conversion Rules

$$
\begin{aligned}
dA &= 0 \\
d(\alpha X) &= \alpha(dX) \\
d(AXB) &= A(dX)B \\
d(X+Y) &= dX + dY \\
d(X^T) &= (dX)^T \\
d(XY) &= (dX)Y + X(dY) \\
d\langle X, Y\rangle &= \langle dX, Y\rangle + \langle X, dY\rangle \\
d\left(\frac{X}{\phi}\right) &= \frac{\phi dX - (d\phi)X}{\phi^2}
\end{aligned}
$$

Standard derivatives table

$$
\begin{aligned}
d\langle A, X\rangle &= \langle A, dX\rangle \\
d\langle Ax, x\rangle &= \langle (A + A^T)x, dx\rangle \\
d\langle Ax, x\rangle &= 2\langle Ax, dx\rangle \qquad (\textbf{если } A = A^T) \\
d(\mathrm{Det}(X)) &= \mathrm{Det}(X)\langle X^{-T}, dX\rangle \\
d(X^{-1}) &= -X^{-1}(dX)X^{-1}
\end{aligned}
$$

Examples of calculating the differential of functions.
Example 1:

$$f(x) = (x^T Ax)(x^T Bx), \quad x \in \mathcal{R}^n, \quad A, B \in \mathcal{R}^{n \times n} \quad A = A^T, B = B^T$$

$$\begin{aligned} df &= d(x^T Ax)(x^T Bx) + (x^T Ax)d(x^T Bx) = \\ &= 2x^T Adx(x^T Bx) + (x^T Ax)2x^T Bdx \end{aligned}$$

Examples of calculating the differential of functions.
Example 2:

$$f(x) = \|x\|_2^3, \quad x \in \mathcal{R}^n$$

$$df = \frac{3}{2}(x^T x)^{\frac{1}{2}} d(x^T x) = \frac{3}{2}(x^T x)^{\frac{1}{2}} 2x^T dx = 3\|x\|_2 x^T dx$$

Examples of calculating the differential of functions.
Example 3:

$$f(X) = det(AXB), \quad X \in \mathcal{R}^{n \times n}$$

$$df = det(AXB)tr((AXB)^{-1}d(AXB)) =$$

$$= det(AXB)tr((AXB)^{-1}AdXB)$$

The canonical view of differentials

Вход \ **Выход**	Скаляр	Вектор	Матрица
Скаляр	$df(x) = f'(x)dx$ ($f'(x)$: скаляр; dx: скаляр)	–	–
Вектор	$df(x) = \langle \nabla f(x), dx \rangle$ ($\nabla f(x)$: вектор; dx: вектор)	$df(x) = J_f(x)dx$ ($J_f(x)$: матрица; dx: вектор)	–
Матрица	$df(X) = \langle \nabla f(X), dX \rangle$ ($\nabla f(X)$: матрица; dX: матрица)	–	–

Note: $\langle A, B \rangle = \sum_{i,j} A_{ij} B_{i,j} = tr(A^T B)$

Then the gradients of the functions from the examples.
Examples 1:

$$
\begin{aligned}
df &= d(x^T Ax)(x^T Bx) + (x^T Ax)d(x^T Bx) = \\
&= 2x^T Adx(x^T Bx) + (x^T Ax)2x^T Bdx = \\
&= 2(x^T Bx)x^T Adx + 2(x^T Ax)x^T Bdx = \\
&= (2(x^T Bx)x^T A + 2(x^T Ax)x^T B)dx = \nabla f(x)^T dx
\end{aligned}
$$

Then: $\nabla f(x) = 2((x^T Bx)xAx + 2(x^T Ax)Bx)$

Then the gradients of the functions from the examples.
Examples 2-3:

$$df = 3\|x\|_2 x^T dx = \nabla f(x)^T dx$$

Then:

$$\nabla f(x) = 3\|x\|_2 x$$

$$df = det(AXB)tr((AXB)^{-1}AdXB) =$$

$$= tr(det(AXB)B(AXB)^{-1}AdX) = tr(\nabla f(X)^T dX)$$

Then:

$$\nabla f(X) = det(AXB)A^T(AXB)^{-T}B^T$$

Hessian Calculation

$$f : U \to V$$

First differential: $df(x)[h]$

Second differential: $d^2 f(x)[h_1, h_2] = d(df(x)[h_1])(x)[h_2]$

If : $U = \mathcal{R}^n,\ V = \mathcal{R}$

Then: $d^2 f(x)[h_1, h_2] = h_1^T \nabla^2 f(x) h_2$

Show the calculation of the Hessian by an example

$$f(x) = \|x\|_2^3 \qquad df = 3\|x\|_2 x^T dx$$

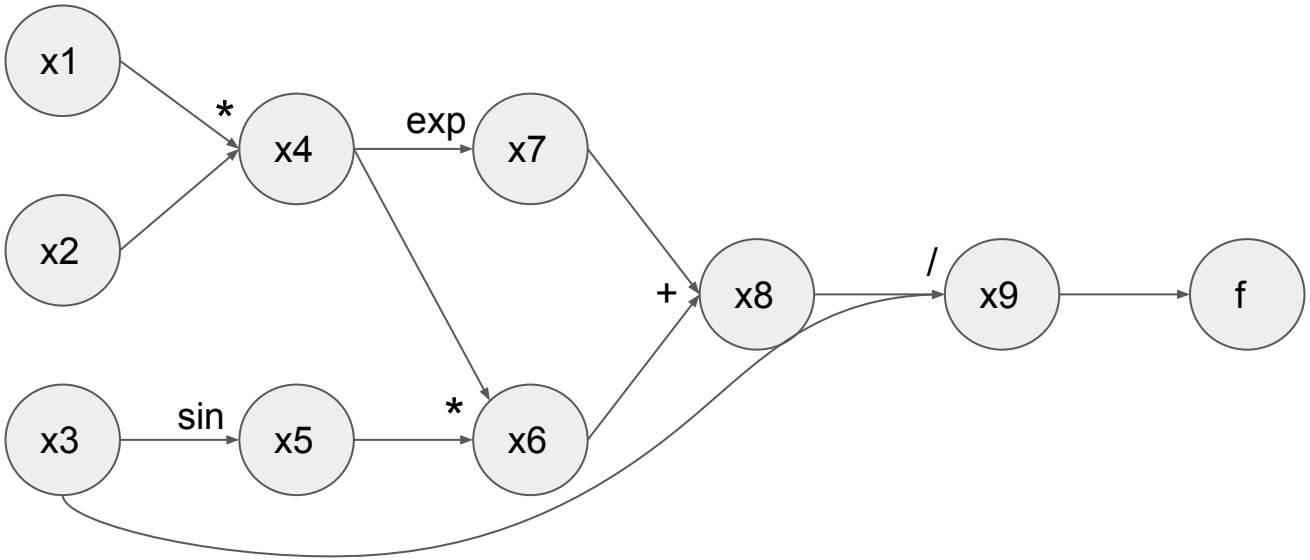
$$\begin{aligned} d^2 f(x)[dx_1, dx_2] &= d\left(3\|x\|_2 x^T dx_1\right) = \\ &= 3d((x^T x)^{\frac{1}{2}})x^T dx_1 + 3(x^T x)^{\frac{1}{2}} d(x^T dx_1) = \\ &= 3\frac{1}{2}(x^T x)^{-\frac{1}{2}} 2x^T dx_2 x^T dx_1 + 3(x^T x)^{\frac{1}{2}} dx_2^T dx_1 = \\ &= dx_1^T x 3(x^T x)^{-\frac{1}{2}} x^T dx_2 + dx_1^T 3(x^T x)^{\frac{1}{2}} I dx_2 \end{aligned}$$

$$\boxed{\nabla^2 f(x) = 3\frac{xx^T}{\|x\|} + 3\|x\| I}$$

Automatic differentiation

$$f(x_1, x_2, x_3) = \frac{x_1 x_2 \sin(x_3) + \exp(x_1 x_2)}{x_3}$$

Need: $\nabla f(x)$



Passing forward

At each step, the derivative $\frac{\partial x_j}{\partial x_i}$ will be calculated, where $i = 1, 2, 3$ are input variables, and $j = 4, 5, \ldots, 9$ are - nodes that will be sequentially calculated.

1. Initialization: $\frac{\partial x_1}{\partial x_1} = 1$, $\frac{\partial x_1}{\partial x_2} = 0$, $\frac{\partial x_1}{\partial x_3} = 0$
2. Calculate the value: $x_4 = x_1 x_2$
3. Calculate derivatives on the current layer: $\frac{\partial x_4}{\partial x_1} = \frac{\partial x_1}{\partial x_1} x_2 + x_1 \frac{\partial x_2}{\partial x_1}$, and we take the derivatives from the previous layer
4. Similarly, for the next layer, the value: $x_5 = \sin(x_3)$ and the derivative: $\frac{\partial x_5}{\partial x_1} = \cos(x_3) \frac{\partial x_3}{\partial x_1}$, where the derivative $\frac{\partial x_3}{\partial x_1}$ was calculated on the previous layer.
5. And so on.

Advantages of pass forward: No extra memory needed.
Disadvantages of pass forward: You need to traverse the calculation graph as many times as the input parameters.

Passing backwards

At each step, the derivative $\frac{\partial}{\partial x_j}$ will be calculated, where $j = 9, 8, \ldots, 1$ are the nodes that will be sequentially calculated.

1. First step: $\frac{\partial f}{\partial x_9} = 1$
2. Second step: $\frac{\partial f}{\partial x_8} = \frac{\partial f}{\partial x_9}\frac{\partial x_9}{\partial x_8} = 1 \cdot \frac{1}{x_3}$
3. And so on
4. In general: $\frac{\partial f}{\partial x_j} = \sum_{(j,k)\in\varepsilon} \frac{\partial f}{\partial x_k}\frac{\partial x_k}{\partial x_j}$, where $\frac{\partial f}{\partial x_k}$ is known from the previous step, and $\frac{\partial x_k}{\partial x_j}$ is computed directly from the graph .

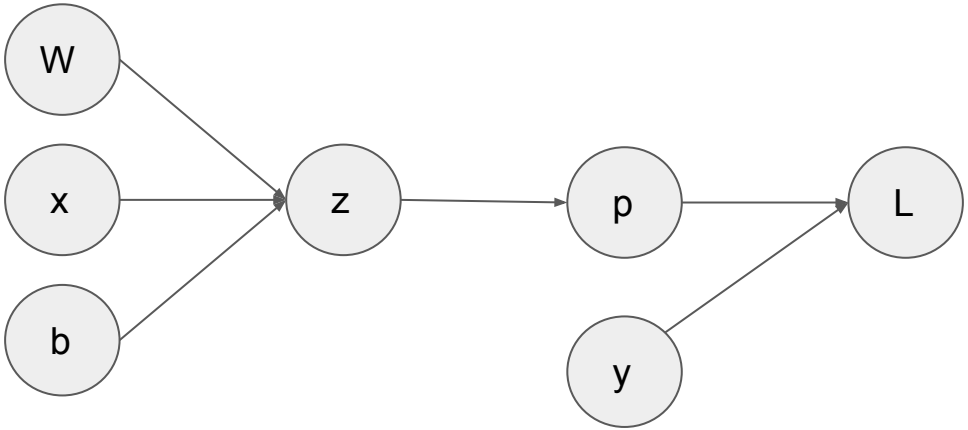
Advantages of pass backwards: Only one backtrack is required to compute the gradient.
Disadvantages of pass backwards: You need additional memory to store all intermediate vertices.

Gradient Calculation Examples: Logistic regression

$$z = Wx + b, \quad W \in \mathcal{R}^{K \times D}, \quad b \in \mathcal{R}^{D} \qquad\qquad x \in \mathcal{R}^{D}, \quad y \in \{1, 2, \ldots, K\}$$

$$p = softmax(z) = \frac{\exp(z)}{\sum_j \exp(z_j)}$$

$$L(p, y) = -\log p_y = -\log p^T \mathbf{1}_y, \quad \mathbf{1}_y = [0, \ldots, 0, \overset{y}{1}, 0, \ldots, 0]$$



Gradient Calculation Examples: Logistic regression

$$dL = d(-\log p^T \mathbf{1}_y) = -\frac{1}{p^T \mathbf{1}_y} d(p^T \mathbf{1}_y) = -\frac{1}{p^T \mathbf{1}_y} \mathbf{1}_y d(p)$$

then:

$$\nabla_p L = -\frac{1}{p^T \mathbf{1}_y} \mathbf{1}_y$$

Next, we need to calculate how p and z are related:

$$p = \frac{\exp(z)}{\exp(z)^T \mathbf{1}}$$

Gradient Calculation Examples: Logistic regression

$$dp = \frac{d(\exp(z))\exp(z)^T\mathbf{1} - \exp(z)d(\exp(z)^T\mathbf{1})}{(\exp(z)^T\mathbf{1})^2} =$$

$$= \frac{diag(\exp(z))dz\exp(z)^T\mathbf{1} - \exp(z)\mathbf{1}^T diag(\exp(z))dz}{(\exp(z)^T\mathbf{1})^2}$$

$$dL = -\frac{1}{p^T\mathbf{1}_y}\mathbf{1}_y dp =$$

$$= -\frac{1}{p^T\mathbf{1}_y}\frac{1}{\exp(z)^T\mathbf{1}}\mathbf{1}_y^T diag(\exp(z))dz + \frac{1}{p^T\mathbf{1}_y}\frac{\mathbf{1}_y\exp(z)\mathbf{1}_y^T diag(\exp(z))dz}{(\exp(z)^T\mathbf{1})^2}$$

Gradient Calculation Examples: Logistic regression

$$\nabla_z L = -\frac{1}{p^T \mathbf{1}_y} \frac{1}{\exp(z)^T \mathbf{1}} \mathbf{1}_y^T diag(\exp(z)) + \frac{1}{p^T \mathbf{1}_y} \frac{\mathbf{1}_y \exp(z) \mathbf{1}_y^T diag(\exp(z))}{(\exp(z)^T \mathbf{1})^2}$$

$$z = Wx + b$$

$$dz = dWx + Wdx + db$$

$$dL = \nabla_z L^T dz = \nabla_z L^T (dWx + Wdx + db) = tr(x \nabla_z L^T dW) + \nabla_z L^T W dx + \nabla_z L^T db$$

Then: $\nabla_W L = \nabla_z L x^T$ $\nabla_x L = W^T \nabla_z L$ $\nabla_b L = \nabla_z L$

Homework 1

Deadline: 19 september 23:59

Simple practicing in gradient calculation + backpropagation

Recap

- Multi-layer perceptron
  - Activations
  - Properties
- Gradient calculation
  - How to calculate gradient?